

IPV – Instituto Politécnico de Viseu  
ESTGL – Escola Superior de Tecnologia e Gestão de Lamego

Sistemas de Bases de Dados  
Computação Cloud

Ano Letivo – 2024/2025

Miguel Pereira nº11177

# ÍNDICE

Introdução .....	1
Capítulo 1 – Estrutura e Modelagem da Base de Dados .....	2
1.1 - Diagrama de Entidade-Relacionamento (ERD) .....	2
1.2 - Relacionamentos .....	3
Capítulo 2 – Criação da base de Dados .....	3
2.1 - Otimização e Abstração com Views .....	4
2.2 - Encapsulamento da Lógica com Stored Procedures .....	5
Capítulo 3 – Interface e Funcionalidades .....	6
3.1 – Gestão de Clientes .....	7
3.2 - Gestão de Filmes .....	9
3.3 – Gestão de Alugueres .....	12
3.4 – Histórico de Alugueres .....	14
3.5 – Relatórios Estatísticos e Consultas Avançadas .....	15
3.5.1 - Top Filmes Mais Alugados .....	15
3.5.2 - Filmes Nunca Alugados .....	17
3.5.3 - Número de Alugueres por Cliente .....	17
3.5.4 – Filmes Alugados por Intervalo de Datas .....	19
3.5.5 – Clientes com Alugueres em Atraso .....	21

Figura 1 - Data modeler.....	2
Figura 2 – Dashboard .....	6
Figura 3 - Gestão de Clientes .....	7
Figura 4 - Histórico Individual .....	8
Figura 5 - Gestão de Filmes.....	9
Figura 6 - Chamada à API .....	10
Figura 7 - Recomendação após três letras .....	11
Figura 8 - Preenchimento automático graças à API .....	12
Figura 9 - Gestão de alugueres .....	13
Figura 10 - Formulário Histórico de Alugueres .....	14
Figura 11 - Top Filmes mais Alugados .....	16
Figura 12 - Query Top5 Filmes .....	16
Figura 13 - Filmes nunca Alugados .....	17
Figura 14 - Alugueres por Cliente .....	18
Figura 15 - Alugueres por data .....	19
Figura 16 - Clientes com Aluguer em Atraso .....	21
Figura 17 - Contactar Cliente .....	23

# Introdução

Este relatório técnico detalha o desenvolvimento de um Sistema de Gestão de Videoclube, uma aplicação desktop desenvolvida em C# utilizando Windows Forms com ligação a uma base de dados SQL Server. O projeto foi dividido em duas fases principais, abordando desde a gestão básica de dados até funcionalidades avançadas de relatórios e consultas.

Na primeira fase do projeto, o foco principal foi a implementação de um sistema robusto de gestão de dados que permitisse operações CRUD (Criar, Ler, Atualizar, Eliminar) para as entidades essenciais de um videoclube: Clientes, Filmes e Alugueres. Esta fase incluiu a modelagem da base de dados no Oracle SQL Data Modeler, a criação das tabelas necessárias, e a utilização de ADO.NET para a interação com a base de dados SQL Server. Adicionalmente, foram desenvolvidas stored procedures e views para otimizar operações e demonstrar boas práticas de desenvolvimento de base de dados. A aplicação foi desenhada seguindo princípios de separação de camadas para garantir a organização e a manutenibilidade do código, com uma interface gráfica clara e funcional e validação de dados obrigatória.

A segunda fase expandiu significativamente a aplicação, incorporando funcionalidades de consultas SQL avançadas e relatórios estatísticos. O objetivo foi aprofundar a manipulação de dados, permitindo análises mais detalhadas do funcionamento do videoclube. Novas funcionalidades foram integradas na interface, utilizando queries SQL com operações de agregação, ordenação e filtragem. Entre as implementações chave destacam-se a capacidade de identificar os filmes mais alugados, o número de alugueres por cliente, filmes alugados num intervalo de datas, filmes nunca alugados e a listagem de clientes com alugueres em atraso. Estas funcionalidades foram construídas utilizando ADO.NET, com views e stored procedures parametrizadas para demonstração de requisitos técnicos avançados. Os resultados são apresentados em DataGridViews, garantindo uma interface de utilizador intuitiva e eficiente.

Este relatório irá descrever em detalhe as funcionalidades implementadas, as tecnologias utilizadas e a arquitetura geral da aplicação, complementado por capturas de ecrã que ilustram a interface e o comportamento do sistema em tempo real.

# Capítulo 1 – Estrutura e Modelagem da Base de Dados

A base de dados do Sistema de Gestão de Videoclube foi concebida para gerir eficientemente informações sobre clientes, filmes e alugueres. A modelagem foi realizada utilizando o Oracle SQL Data Modeler, garantindo uma estrutura relacional otimizada e a integridade dos dados. O modelo é composto por três entidades principais: Cliente, Filme e Aluguer, interligadas por relacionamentos que refletem as regras de negócio do videoclube.

## 1.1 - Diagrama de Entidade-Relacionamento (ERD)

O diagrama abaixo ilustra a estrutura da base de dados, com as tabelas, os seus atributos e os relacionamentos estabelecidos:

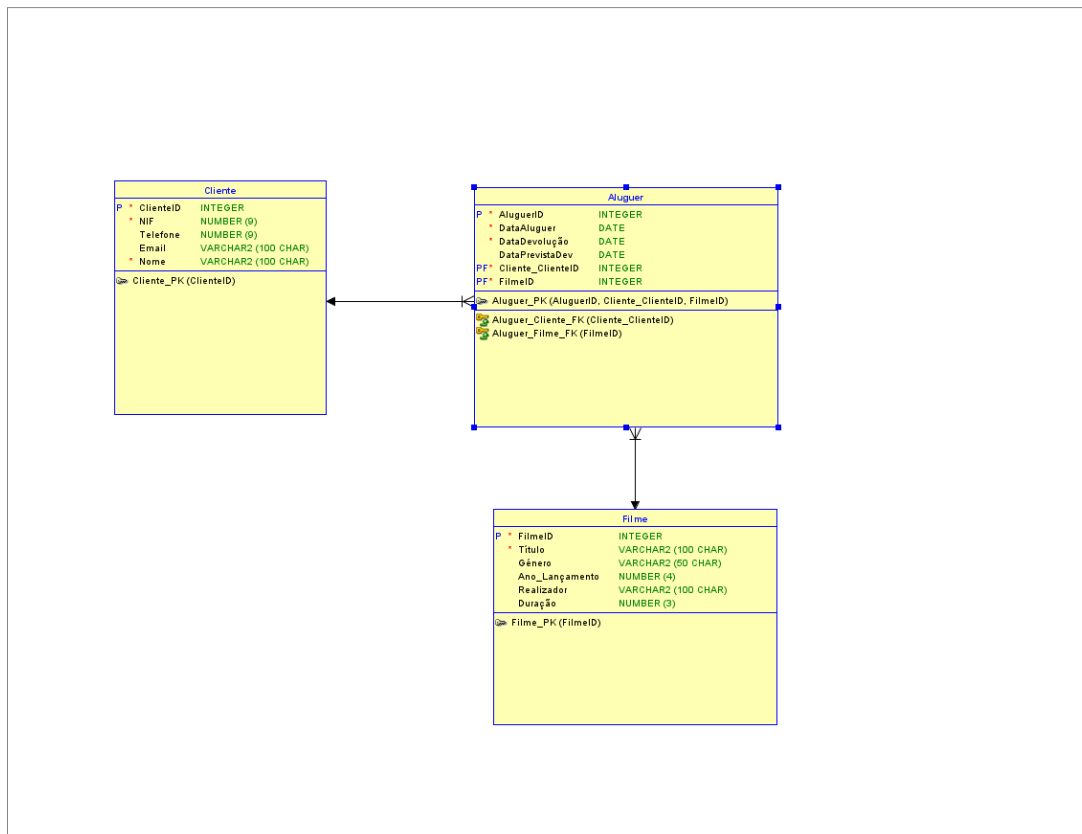


Figura 1 - Data modeler

## 1.2 - Relacionamentos

Cliente para Aluguer (Um-para-Muitos): Um cliente pode realizar múltiplos alugueres, mas cada aluguer está associado a apenas um cliente. Este relacionamento é estabelecido através da chave estrangeira `ClienteID` na tabela `Aluguer`.

Filme para Aluguer (Um-para-Muitos): Um filme pode ser alugado múltiplas vezes, mas cada registo de aluguer refere-se a um único filme. Este relacionamento é estabelecido através da chave estrangeira `FilmeID` na tabela `Aluguer`.

Esta modelagem assegura a integridade referencial dos dados e facilita a execução de consultas complexas e a gestão das operações do videoclube. A partir deste modelo, procedeu-se à criação da base de dados no SQL Server.

## Capítulo 2 – Criação da base de Dados

A transição do modelo conceptual para a implementação física da base de dados é um passo crítico no desenvolvimento de qualquer sistema de informação. Para o Sistema de Gestão de Videoclube, esta fase foi concretizada através da criação da base de dados Videoclube no SQL Server, utilizando um script SQL detalhado que não só definiu a estrutura das tabelas, mas também incorporou mecanismos essenciais para garantir a integridade dos dados e otimizar o desempenho das operações.

A estrutura da base de dados foi concebida para suportar as operações fundamentais de um videoclube, centrando-se em três entidades principais: `Cliente`, `Filme` e `Aluguer`. Cada tabela foi cuidadosamente definida com tipos de dados apropriados para armazenar as informações específicas, como identificadores únicos (`ClienteID`, `FilmeID`, `AluguerID`) definidos com a propriedade `IDENTITY` para auto-incremento, simplificando a gestão de novos registos e delegando a geração de chaves primárias à base de dados.

Para assegurar a consistência e a validade dos dados, foram aplicadas diversas estratégias de integridade diretamente ao nível da base de dados, prevenindo a inserção de dados inválidos e mantendo a coerência relacional:

- **Integridade Referencial:** A interligação entre as tabelas é mantida através de chaves estrangeiras. Estas ligações garantem que cada aluguer está sempre associado a um cliente e um filme existentes. Este mecanismo

impede a criação de registos inconsistentes e protege contra a eliminação acidental de dados associados, reforçando as regras de negócio do videoclube.

- **Validação de Dados:** Foram definidas restrições para validar a lógica de negócio e a validade dos atributos. Uma dessas restrições assegura que a data prevista de devolução de um aluguer nunca seja anterior à data em que o filme foi alugado, evitando inconsistências temporais. Outras restrições validam a duração e o ano de lançamento dos filmes, como garantir que a duração seja um valor positivo e dentro de um intervalo realista (entre 1 e 499 minutos), e que o ano de lançamento esteja dentro de um período cronologicamente válido (entre 1888 e o ano corrente). Adicionalmente, o número de identificação fiscal (NIF) na tabela Cliente possui uma restrição que impede a duplicação, garantindo a unicidade de cada cliente registado na base de dados.

## 2.1 - Otimização e Abstração com Views

A utilização de VIEWS no sistema tem como principal objetivo simplificar a recuperação de dados e criar uma camada de abstração sobre as tabelas subjacentes. Isto melhora a manutenibilidade do código SQL e, potencialmente, a segurança, ao permitir que a aplicação aceda apenas a conjuntos de dados pré-definidos.

As views desenvolvidas desempenham papéis específicos na funcionalidade do sistema:

- **Dados de Cliente e Filme:** Existem views para fornecer uma maneira padronizada e simplificada de aceder a todos os dados de clientes e filmes, respetivamente, reduzindo a complexidade de consultas diretas às tabelas.
- **Dados de Aluguer:** Uma view centraliza as informações das tabelas de Aluguer, Cliente e Filme, consolidando os detalhes de cada aluguer (incluindo o nome do cliente e o título do filme). O seu propósito é apresentar estes dados de forma completa e eficiente, sem exigir junções complexas na camada de aplicação.
- **Alugueres em Atraso:** Uma view dedicada foi implementada para identificar alugueres pendentes. Esta view filtra os alugueres cuja data prevista de devolução já passou e que ainda não foram devolvidos,

retornando detalhes essenciais do cliente e do filme. É uma ferramenta de gestão vital para ações proativas sobre alugueres em atraso.

- Alugueres por Cliente: Essencial para relatórios estatísticos, esta view agrupa alugueres por cliente e calcula o número total de alugueres que cada um realizou. Apresenta o identificador, nome e NIF do cliente, bem como o total de alugueres, fornecendo insights sobre a atividade dos clientes.

## 2.2 - Encapsulamento da Lógica com Stored Procedures

A implementação de STORED PROCEDURES na base de dados encapsula a lógica de negócio no servidor, promovendo a reutilização de código, a segurança e a otimização de desempenho através de planos de execução pré-compilados.

As stored procedures desenvolvidas abrangem desde as operações CRUD básicas até funcionalidades de consulta mais complexas:

- Operações CRUD Básicas: Procedures foram criadas para padronizar as operações de criação, leitura, atualização e eliminação de registos para clientes, filmes e alugueres. Por exemplo, a criação de novos clientes valida a unicidade do NIF.
- Registo de Devolução: Uma procedure permite registar a devolução de um filme, atualizando a data de devolução de um aluguer específico e marcando-o como devolvido.
- Alugueres Atuais: Uma procedure é utilizada para popular a dashboard da aplicação, retornando os alugueres ativos (ainda não devolvidos), fornecendo uma visão em tempo real dos filmes que estão em posse dos clientes.
- Alugueres por Intervalo de Datas: Uma stored procedure parametrizada permite listar filmes alugados num intervalo de datas específico, recebendo datas inicial e final. Esta funcionalidade é crucial para análises temporais e filtragem do histórico de alugueres.
- Top Filmes: Para identificar os filmes mais populares, uma procedure agrupa os alugueres por filme e os ordena pelo total de alugueres, fornecendo insights sobre as preferências dos utilizadores.
- Outras Procedures de Apoio: Incluem procedimentos para consultar o histórico de alugueres de um cliente específico, listar todos os alugueres com detalhes



combinados, e verificar rapidamente se um filme específico está atualmente emprestado.

## Capítulo 3 – Interface e Funcionalidades

A aplicação "Sistema de Gestão de Videoclube", desenvolvida em C# com Windows Forms, proporciona uma interface gráfica de utilizador (GUI) projetada para a gestão eficiente das operações diárias. A sua arquitetura visa uma interação fluida e a separação de responsabilidades entre as camadas, o que facilita a manutenção e futuras expansões.

Para estabelecer a comunicação com a base de dados SQL Server, a aplicação utiliza uma connection string que se encontra configurada e armazenada no ficheiro App.config. Esta prática é fundamental para a manutenibilidade do sistema, pois permite que os detalhes de conexão à base de dados (como o servidor, nome da base de dados, credenciais) sejam alterados sem a necessidade de recompilar a aplicação. Além disso, contribui para a segurança, evitando que credenciais sensíveis estejam hardcoded no código-fonte da aplicação.

A interface principal funciona como uma dashboard, oferecendo uma visão geral e acesso rápido às áreas chave do sistema.

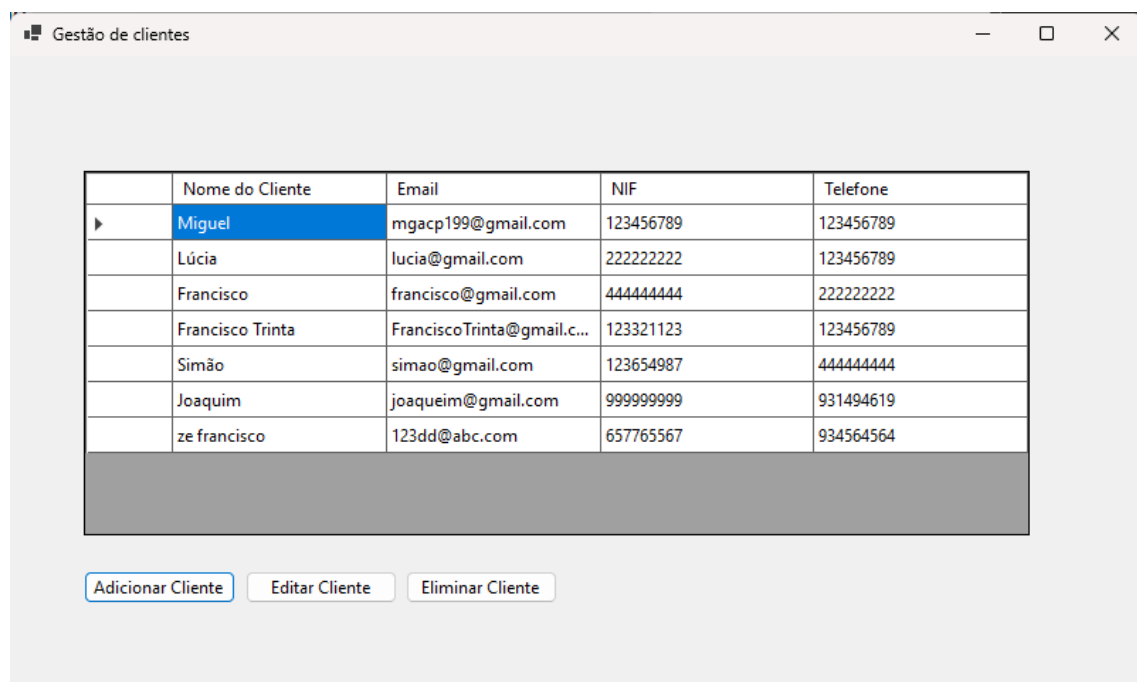


Figura 2 – Dashboard

Esta dashboard não só facilita a navegação para a gestão de clientes, filmes, alugueres, histórico e relatórios, como também exibe em tempo real uma lista de "Alugueres Atuais". Esta grelha é preenchida dinamicamente, permitindo aos operadores visualizar imediatamente os filmes que estão alugados, as datas de aluguer e de devolução prevista, e oferece uma opção direta para registar a devolução. O objetivo é otimizar o acompanhamento diário e agilizar o processo de devolução.

### 3.1 – Gestão de Clientes

A gestão de clientes é uma funcionalidade central para manter os registos dos utilizadores do videoclube atualizados. Esta secção permite executar operações essenciais sobre os dados dos clientes: criar registos, consultar existentes, modificar informações e eliminar dados.



*Figura 3 - Gestão de Clientes*

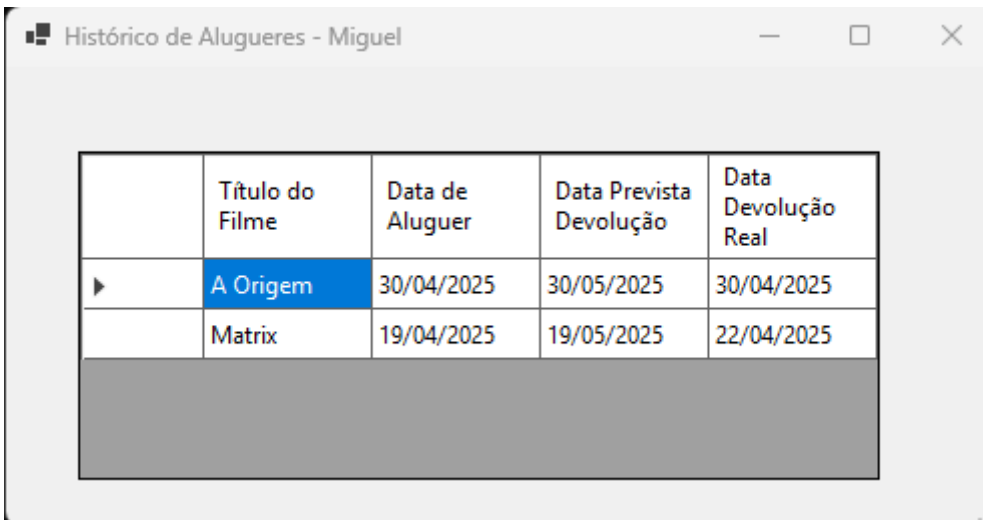
O formulário de gestão de clientes exibe uma grelha (DataGridView) com todos os clientes registados (Nome, NIF, Telefone, Email), permitindo uma visualização clara e seleção rápida. As ações disponíveis, acessíveis através de botões dedicados, incluem:

- Adicionar Novo Cliente: Para registar novos utilizadores no sistema, coletando informações como Nome, NIF, Telefone e Email.
- Editar Cliente Existente: Para atualizar os dados de um cliente selecionado, corrigindo ou complementando informações.

- **Eliminar Cliente:** Para remover um registo. Esta operação é protegida pela base de dados, que impede a eliminação de clientes com alugueres associados, mantendo a integridade dos dados históricos.

A comunicação com a base de dados para estas operações é efetuada através de Stored Procedures dedicadas, como CreateClient, UpdateCliente, DeleteCliente. Esta abordagem centraliza a lógica de acesso e manipulação de dados no servidor, melhorando a segurança e a consistência das operações.

Além das operações CRUD, o sistema permite, a partir da listagem principal de clientes, visualizar o histórico de alugueres de um cliente individual. Ao selecionar um cliente na grelha, uma nova janela é apresentada, detalhando todos os filmes que esse cliente alugou. Esta funcionalidade é crucial para o acompanhamento do perfil de aluguer de cada cliente, permitindo uma análise rápida do seu histórico de interações com o videoclub. A consulta para este histórico é realizada através da Stored Procedure GetHistoricoByClienteID.



	Título do Filme	Data de Aluguer	Data Prevista Devolução	Data Devolução Real
▶	A Origem	30/04/2025	30/05/2025	30/04/2025
	Matrix	19/04/2025	19/05/2025	22/04/2025

*Figura 4 - Histórico Individual*

Na camada de aplicação, foram implementadas validações rigorosas para garantir a qualidade dos dados introduzidos pelo utilizador:

- **Formato de Contacto:** Validações asseguram que os números de telefone correspondam ao formato numérico português (9 dígitos) e que os endereços de email contêm a estrutura @ e um domínio válido (ex: exemplo@dominio.com). O propósito é garantir a fiabilidade dos contactos para comunicação futura e evitar erros de registo.

- Campos Obrigatórios: A aplicação exige o preenchimento de campos essenciais (Nome, NIF, Telefone) antes de permitir o registo, prevenindo a criação de registos incompletos.

### 3.2 - Gestão de Filmes

A secção de Gestão de Filmes é fundamental para a manutenção e expansão do catálogo de filmes do videoclube. Esta funcionalidade permite operações CRUD completas sobre os registos de filmes, garantindo que a base de dados de títulos esteja sempre atualizada e acessível.

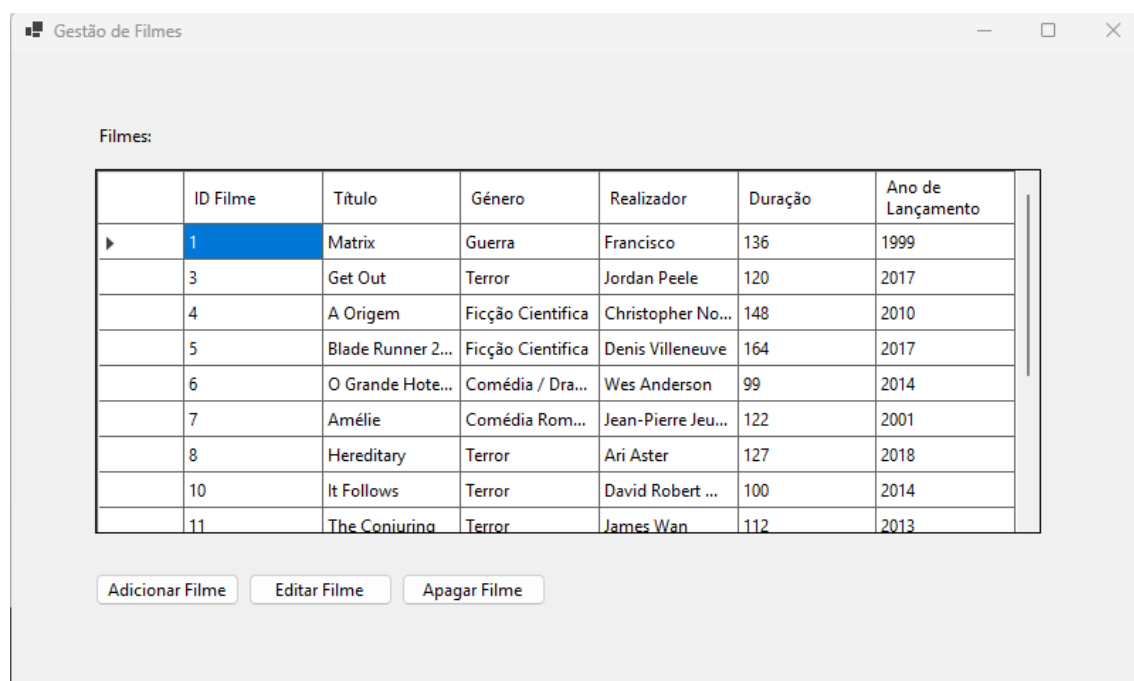


Figura 5 - Gestão de Filmes

O formulário de gestão de filmes apresenta uma grelha (DataGridView) com o catálogo completo, exibindo detalhes como Título, Género, Realizador, Duração e Ano de Lançamento. A interface facilita a consulta e a manipulação dos registos através de botões para:

- Adicionar Novo Filme: Permite a inserção de um novo título no catálogo. Esta funcionalidade inclui uma integração com uma API externa (TMDB). Ao iniciar o processo de adição, a aplicação permite pesquisar por filmes e, ao selecionar um título, os campos de "Título", "Género", "Realizador", "Duração" e "Ano de Lançamento" são preenchidos automaticamente com os dados obtidos da API,

otimizando e acelerando o processo de registo e garantindo a precisão das informações.

```
private const string TMDB_API_KEY = "b74ed907fd01a8d31740e4676e5d67b4";  
private static readonly HttpClient _httpClient = new HttpClient();  
private bool _isUpdatingTextBoxProgrammatically = false;
```

*Figura 6 - Chamada à API*

- Editar Filme Existente: Habilita a modificação dos detalhes de um filme selecionado.
- Apagar Filme: Remove um registo de filme. Esta operação é protegida pela base de dados, que impede a eliminação de filmes que tenham alugueres associados, mantendo a integridade referencial dos dados históricos.

A interação com a base de dados é gerida por Stored Procedures como CreateFilme (que verifica a duplicação de títulos antes da inserção, conforme presente no script\_final.sql ), UpdateFilme, e DeleteFilme.

Adicionar Novo Filme

Título:

asd

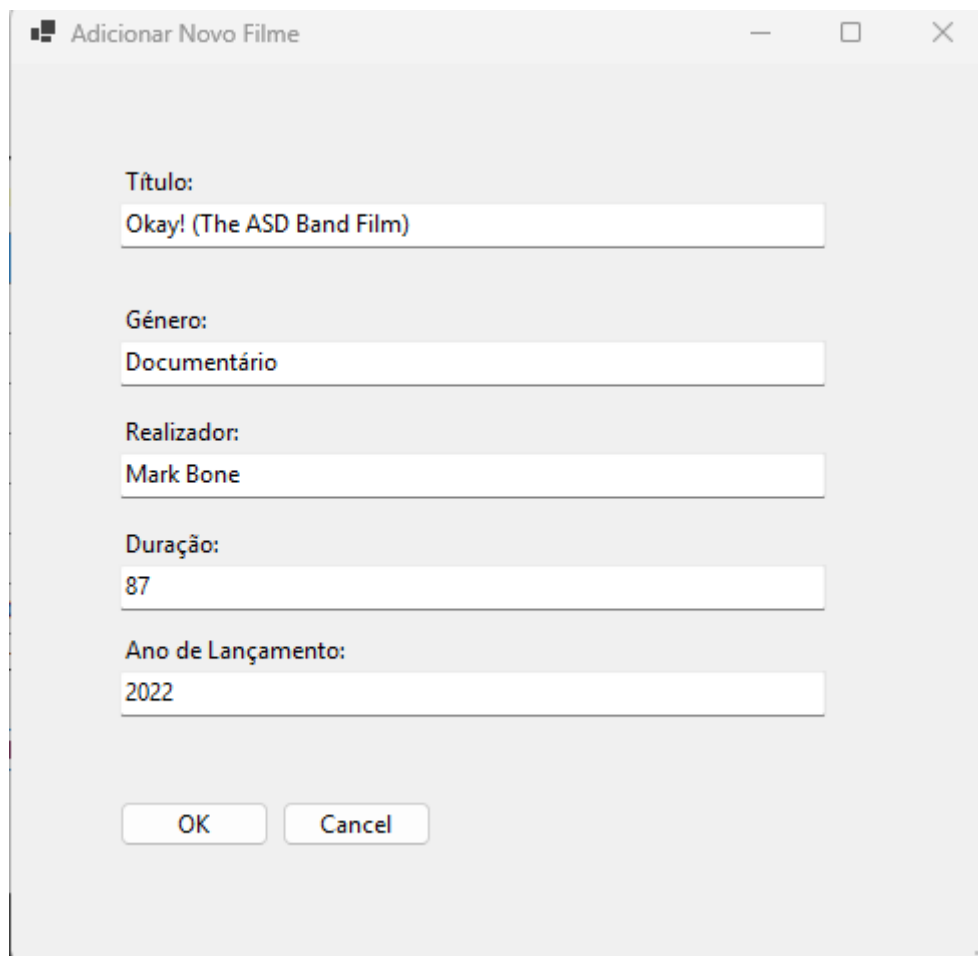
Okay! (The ASD Band Film) (2022)  
Inside Asda: Bigger, Better, Cheaper? (2020)  
Asdrúbal Trouxe o Trombone (2017)  
ASD - Alma sin dueño (Sin alma) (2010)  
Friends Jokes (2006)  
Asdrúbal, O que há com seu peru? (1978)  
The Three Friends (1966)  
Raising ASD

Duração:

Ano de Lançamento:

OK Cancel

Figura 7 - Recomendação após três letras



Adicionar Novo Filme

Título:  
Okay! (The ASD Band Film)

Género:  
Documentário

Realizador:  
Mark Bone

Duração:  
87

Ano de Lançamento:  
2022

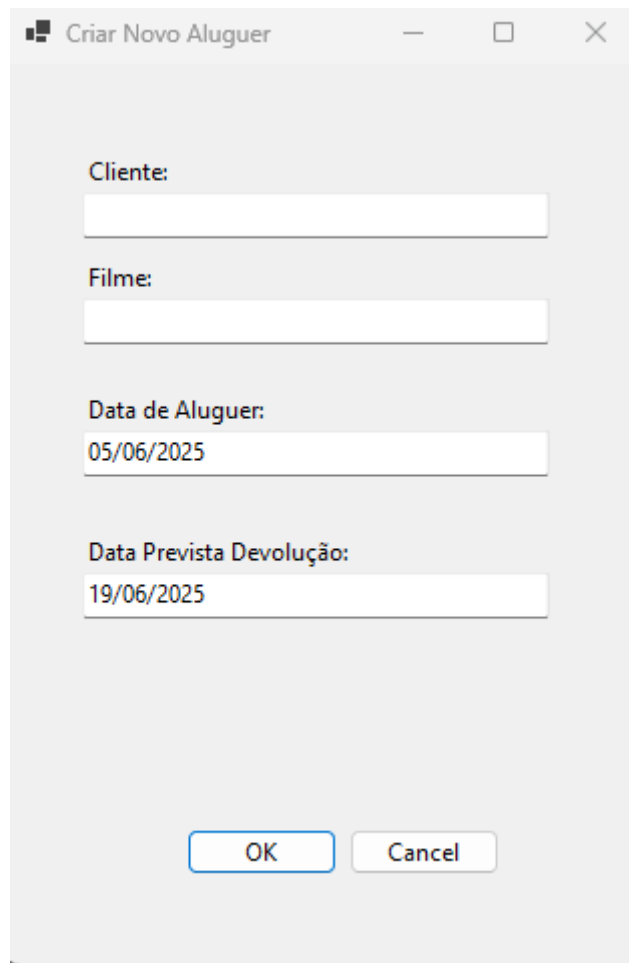
OK Cancel

*Figura 8 - Preenchimento automático graças à API*

No que diz respeito às validações de dados na camada de aplicação, foram implementadas verificações para garantir a consistência e a validade dos dados inseridos para os filmes. Isto inclui a validação de campos obrigatórios e a verificação de formatos para a duração (deve ser um valor numérico positivo) e o ano de lançamento (deve ser um ano válido). Estas validações complementam as restrições (CHECK CONSTRAINT) definidas na base de dados que garantem a validade da duração (entre 1 e 499 minutos) e do ano de lançamento (entre 1888 e o ano corrente, até ao ano atual).

### 3.3 – Gestão de Alugueres

A secção de Gestão de Alugueres é crucial para o controlo do fluxo de filmes, desde a sua saída do videoclube até à sua devolução. Permite registar novos alugueres e gerir o processo de devolução, assegurando que o estado de cada filme é corretamente atualizado.



*Figura 9 - Gestão de alugueres*

O formulário para Criar Novo Aluguer é concebido para simplificar o processo, exigindo a seleção de um cliente e de um filme através de listas suspensas populadas dinamicamente com dados existentes na base de dados. Esta abordagem garante que apenas clientes registados e filmes presentes no catálogo possam ser selecionados para um aluguer, reforçando a integridade referencial ao nível da interface. Para otimizar a usabilidade, a Data de Aluguer e a Data Prevista de Devolução são automaticamente preenchidas com base na data atual. A data de aluguer é definida para o dia corrente, e a data prevista de devolução é calculada adicionando um período padrão (e.g., 14 dias) à data de aluguer, embora permitindo ajustes manuais se necessário.

Uma validação crucial implementada no sistema impede que um filme seja alugado se já estiver em posse de outro cliente (ou do mesmo cliente, caso ainda não tenha sido devolvido). Esta verificação é realizada utilizando a Stored Procedure `IsFilmeAlugado` antes da criação do aluguer, garantindo que não ocorram alugueres duplos de um mesmo exemplar e mantendo a integridade do inventário.

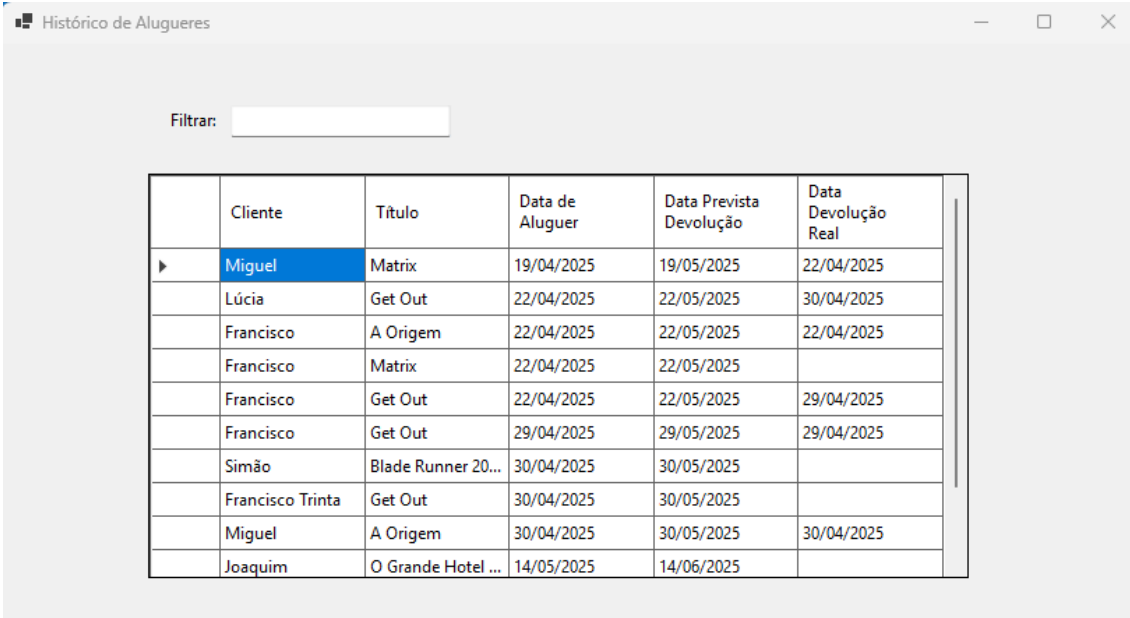


A funcionalidade de Registrar Devolução é acessível diretamente da Dashboard Principal, na grelha de "Alugueres Atuais". Ao clicar no botão "Devolver" para um aluguer específico, o sistema atualiza a DataDevolucao para a data corrente e remove o aluguer da lista de alugueres pendentes. Este processo é gerido pela Stored Procedure MarkAluguerAsReturned.

As operações de gestão de alugueres utilizam Stored Procedures como CriarAluguer para registar novos alugueres e MarkAluguerAsReturned para processar devoluções. A Stored Procedure GetOngoingAlugueres é utilizada para popular a grelha de "Alugueres Atuais" na dashboard. A base de dados, através de restrições (CHECK CONSTRAINT), garante que a data prevista de devolução nunca seja anterior à data de aluguer, reforçando a lógica de negócio temporal.

### 3.4 – Histórico de Alugueres

A secção de Histórico de Alugueres oferece uma visão abrangente de todos os alugueres realizados no videoclube, independentemente do seu estado (ativos ou já devolvidos). Esta funcionalidade é essencial para consultas e análises do histórico de transações.



The screenshot shows a window titled "Histórico de Alugueres" with a search filter labeled "Filtrar:" and a table of rental transactions. The table has six columns: an empty column for selection, "Cliente", "Título", "Data de Aluguer", "Data Prevista Devolução", and "Data Devolução Real". The first row is highlighted with a blue background.

	Cliente	Título	Data de Aluguer	Data Prevista Devolução	Data Devolução Real
►	Miguel	Matrix	19/04/2025	19/05/2025	22/04/2025
	Lúcia	Get Out	22/04/2025	22/05/2025	30/04/2025
	Francisco	A Origem	22/04/2025	22/05/2025	22/04/2025
	Francisco	Matrix	22/04/2025	22/05/2025	
	Francisco	Get Out	22/04/2025	22/05/2025	29/04/2025
	Francisco	Get Out	29/04/2025	29/05/2025	29/04/2025
	Simão	Blade Runner 20...	30/04/2025	30/05/2025	
	Francisco Trinta	Get Out	30/04/2025	30/05/2025	
	Miguel	A Origem	30/04/2025	30/05/2025	30/04/2025
	Joaquim	O Grande Hotel ...	14/05/2025	14/06/2025	

Figura 10 - Formulário Histórico de Alugueres

O formulário exibe uma grelha (DataGridView) com as informações mais relevantes para cada aluguer: o Nome do Cliente, o Título do Filme, a Data de Aluguer, a Data Prevista de Devolução e a Data de Devolução Real. Os identificadores internos

(ClienteID e FilmeID) são ocultados nesta interface para uma visualização mais clara e orientada ao utilizador. A grelha é populada pela Stored Procedure GetTodosAlugueres, que junta dados das tabelas de Aluguer, Cliente e Filme para apresentar uma visão consolidada de cada transação.

Para facilitar a análise e a pesquisa em casos de grande volume de dados, o sistema permite filtrar o histórico de alugueres. A interface oferece campos de pesquisa que permitem ao utilizador filtrar os registos pelo nome do cliente ou pelo título do filme. Esta funcionalidade é implementada diretamente na camada de aplicação, aplicando filtros sobre os dados carregados, se tal for mais performático e otimizado ao nível da base de dados.

### 3.5 – Relatórios Estatísticos e Consultas Avançadas

Esta secção detalha as funcionalidades de relatórios e consultas avançadas, que foram implementadas para fornecer insights sobre as operações do videoclub. É importante notar que, para otimizar a experiência do utilizador e a performance, os diferentes relatórios não abrem novos formulários. Em vez disso, a aplicação utiliza um sistema de painéis, onde o conteúdo da área de relatório é dinamicamente atualizado e exibido no mesmo formulário principal, dependendo do botão de relatório que é acionado. Este design promove uma navegação mais fluida e reduz o consumo de recursos.

Estas funcionalidades utilizam VIEWS e STORED PROCEDURES no SQL Server, encapsulando a lógica de negócio e otimizando o desempenho das consultas. Os resultados são apresentados de forma clara em grelhas, permitindo uma análise rápida dos dados.

#### 3.5.1 - Top Filmes Mais Alugados

Este relatório tem como objetivo identificar os filmes mais populares do catálogo, listando os títulos que foram alugados com maior frequência. É uma ferramenta de gestão valiosa para compreender as preferências dos clientes e auxiliar na tomada de decisões relativas à aquisição de novos filmes.

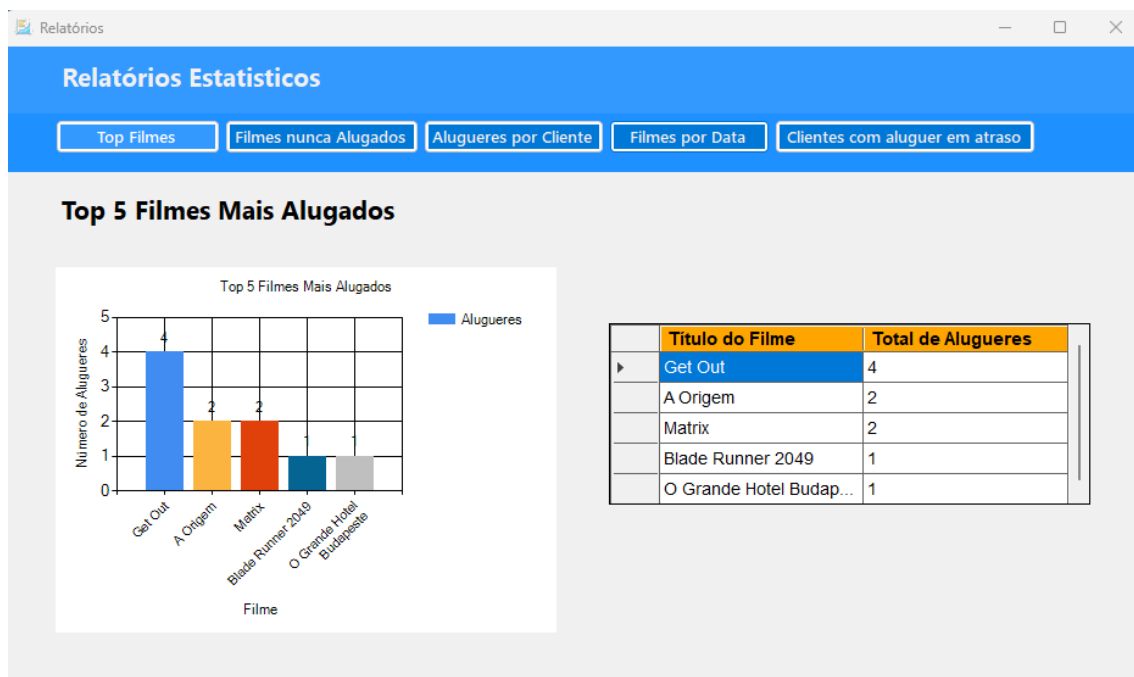


Figura 11 - Top Filmes mais Alugados

O relatório exibe o Título de cada filme e o Número Total de Alugueres que este teve, ordenados de forma decrescente pelo número de alugueres para destacar os mais procurados. Além da apresentação tabular, este relatório é complementado por um gráfico de barras, que oferece uma representação visual imediata da popularidade dos filmes, facilitando a comparação e a identificação dos títulos de maior destaque. A lógica de consulta para este relatório é implementada diretamente no código da aplicação (camada de dados em C#), utilizando comandos ADO.NET para executar uma query SQL que agrega os dados da tabela de Alugueres por FilmeID, conta as ocorrências e limita os resultados aos 5 filmes mais alugados, conforme os requisitos do trabalho prático 2. Os dados obtidos são então utilizados para popular tanto a grelha quanto o componente de gráfico de barras.

```

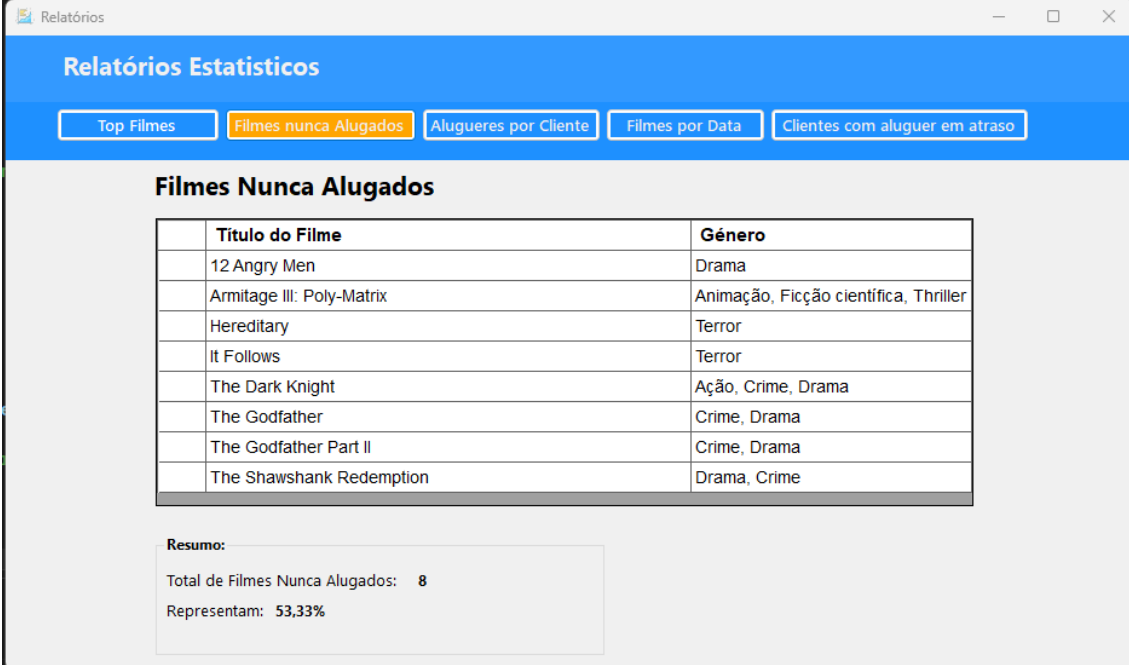
string query = @"
SELECT TOP (@QtdFilmes)
    f.FilmeID,
    f.Titulo,
    COUNT(a.AluguerID) AS TotalAlugueres
FROM Filme f
JOIN Aluguer a ON f.FilmeID = a.Filme_FilmeID
GROUP BY f.FilmeID, f.Titulo
ORDER BY COUNT(a.AluguerID) DESC";

```

Figura 12 - Query Top5 Filmes

### 3.5.2 - Filmes Nunca Alugados

Este relatório lista todos os filmes presentes no catálogo do videoclub e que, até à data, nunca foram alugados.



Título do Filme	Gênero
12 Angry Men	Drama
Armitage III: Poly-Matrix	Animação, Ficção científica, Thriller
Hereditary	Terror
It Follows	Terror
The Dark Knight	Ação, Crime, Drama
The Godfather	Crime, Drama
The Godfather Part II	Crime, Drama
The Shawshank Redemption	Drama, Crime

**Resumo:**  
Total de Filmes Nunca Alugados: 8  
Representam: 53,33%

Figura 13 - Filmes nunca Alugados

O relatório exibe em formato tabular numa DataGridView o Título e o Gênero dos filmes que não possuem qualquer registo na tabela de alugueres. Complementarmente, a interface apresenta labels estatísticas que fornecem uma visão rápida da situação: o Número Total de Filmes Nunca Alugados e a Percentagem que estes representam em relação ao catálogo total de filmes. A consulta para obter os filmes sem aluguer é realizada através da VIEW [dbo].[FilmesNuncaAlugados], que efetua um LEFT JOIN entre a tabela Filme e a tabela Aluguer, filtrando pelos filmes onde não há correspondência (A.AluguerID IS NULL). O cálculo do total de filmes nunca alugados e da sua percentagem em relação ao catálogo total é realizado na camada de aplicação, após a obtenção dos dados da VIEW.

### 3.5.3 - Número de Alugueres por Cliente

Este relatório fornece uma visão detalhada da atividade de aluguer de cada cliente, permitindo identificar os clientes mais ativos no videoclub.



Nome do Cliente	NIF do Cliente	Total de Alugueres
Francisco	444444444	4
Lúcia	222222222	2
Miguel	123456789	2
Simão	123654987	1
ze francisco	657765567	1
Francisco Trinta	123321123	1
Joaquim	999999999	1

Figura 14 - Alugueres por Cliente

O relatório exhibe em formato tabular numa DataGridView o Nome do Cliente, o NIF e o Número Total de Alugueres realizados por cada um. Os resultados são ordenados de forma decrescente pelo número de alugueres, destacando os clientes mais assíduos. Para facilitar a localização de clientes específicos, a interface inclui uma funcionalidade de pesquisa, que permite filtrar os resultados por Nome do Cliente ou por NIF, através de campos de texto dedicados e um botão "Pesquisar".

A base para a obtenção dos dados exibidos neste relatório reside na VIEW vw\_AlugueresPorCliente. A utilização de uma view neste contexto é uma prática que promove a abstração e a reutilização da lógica de negócio na base de dados.

A vw\_AlugueresPorCliente é construída para agregar as informações de aluguer por cliente. A sua lógica interna é a seguinte:

- A view seleciona o ClienteID, o NomeCliente, o NIFCliente e, crucialmente, o COUNT(a.AluguerID) (aliás TotalAlugueres). Este COUNT é responsável por contabilizar o número total de alugueres associados a cada cliente.
- A operação é efetuada através de um LEFT JOIN entre a tabela Cliente e a tabela Aluguer na coluna ClienteID. A escolha do LEFT JOIN é intencional, pois garante que todos os clientes sejam incluídos no resultado, mesmo aqueles que não possuem qualquer registo de aluguer.

Para estes clientes, o COUNT(a.AluguerID) resultará em zero, assegurando que o relatório apresenta um panorama completo de toda a base de clientes, e não apenas dos clientes que já realizaram alugueres.

Os resultados são agrupados (GROUP BY) pelo ClienteID, Nome e NIF do cliente. Esta agregação é essencial para que a função COUNT retorne o número total de alugueres por cada cliente distinto.

### 3.5.4 – Filmes Alugados por Intervalo de Datas

Este relatório é fundamental para a gestão de um videoclube, permitindo uma análise temporal precisa do fluxo de alugueres. Do ponto de vista técnico, a sua inclusão visa demonstrar a capacidade do sistema em realizar consultas parametrizadas sobre dados históricos, o que é um requisito comum em aplicações de gestão de bases de dados. Permite que os administradores identifiquem tendências, avaliem o impacto de campanhas em períodos específicos ou monitorizem a atividade geral do negócio.

The image shows a web application window titled 'Relatórios'. Inside, there's a blue header bar with the text 'Relatórios Estatísticos'. Below this header is a row of five buttons: 'Top Filmes', 'Filmes nunca Alugados', 'Alugueres por Cliente', 'Filmes por Data' (which is highlighted in orange), and 'Clientes com aluguer em atraso'. Below the buttons, the main content area is titled 'Filmes Alugados por Intervalo de Datas'. This section contains a form with two date pickers: 'Data Inicial:' followed by a dropdown showing '6 de junho de 2025', and 'Data Final:' followed by another dropdown showing '6 de junho de 2025'. To the right of these pickers is a green button labeled 'Pesquisar'. Below the form is a large, empty gray rectangular box, likely intended for a chart or table of results.

*Figura 15 - Alugueres por data*

A implementação desta funcionalidade assenta na interação contínua entre a camada de interface e a camada de dados (SQL Server).

Na camada de interface, o formulário Relatórios Estatísticos dispõe de um painel específico para este relatório, desenhado para uma interação intuitiva. O utilizador define

o período de interesse selecionando uma Data Inicial e uma Data Final através de controlos de seleção de data.

Após a definição do intervalo temporal, um botão "Pesquisar" é acionado para desencadear a lógica de consulta. No evento associado a este botão, a aplicação recolhe os valores das datas. É crucial que, a nível da aplicação, seja realizada uma validação inicial para assegurar que a Data Inicial não seja posterior à Data Final. Esta validação preventiva, executada antes de qualquer comunicação com a base de dados, é vital para otimizar o desempenho e evitar erros de lógica, uma vez que impede chamadas desnecessárias ou inválidas ao servidor de base de dados. Se a validação for bem-sucedida, os valores das datas são então formatados e passados como parâmetros para a invocação da stored procedure correspondente na camada de dados, utilizando ADO.NET.

Na camada de dados (SQL Server), a lógica central desta funcionalidade é encapsulada numa stored procedure chamada "GetAlugueresPorDataSP" parametrizada. Esta abordagem é tecnicamente preferível à construção de queries dinâmicas na aplicação por diversas razões: primeiramente, contribui significativamente para a segurança, mitigando riscos de ataques de injeção de SQL, uma vez que os parâmetros são tratados de forma isolada da query principal. Em segundo lugar, promove a otimização de desempenho, pois as stored procedures são pré-compiladas, resultando num plano de execução mais eficiente e rápido em chamadas repetidas. Por fim, permite o encapsulamento da lógica de negócio diretamente no servidor de base de dados, centralizando a regra de seleção de filmes por intervalo de datas e promovendo a reutilização e a manutenibilidade do código SQL. Esta stored procedure é responsável por selecionar os filmes da tabela Aluguer (realizando, se necessário, junções com as tabelas Filme e Cliente para obter os detalhes completos) onde a Data de Aluguer se enquadra entre a Data Inicial e a Data Final fornecidas.

A stored procedure executa uma consulta que:

- Seleciona os detalhes essenciais do aluguer: Recolhe o nome do cliente, o título do filme e as datas de aluguer e devolução.
- Combina dados de múltiplas tabelas: Realiza junções (JOINS) entre as tabelas Aluguer, Cliente e Filme para consolidar toda a informação relevante de cada aluguer num único resultado.

- Filtra por intervalo de datas: Aplica um filtro (WHERE) para incluir apenas os alugueres cuja DataAluguer esteja dentro do intervalo definido pelos parâmetros @DataInicial e @DataFinal.
- Ordena os resultados: Apresenta os dados ordenados cronologicamente pela DataAluguer, facilitando a análise.

### 3.5.5 – Clientes com Alugueres em Atraso

A sua implementação técnica visa demonstrar a capacidade do sistema em realizar consultas de estado e alertas, um componente vital em qualquer sistema de gestão que lide com prazos e obrigações contratuais (neste caso, o contrato de aluguer). Permite uma ação proativa por parte da gestão, como o contacto com o cliente para a recuperação do filme.

	Ação	Nome do Cliente	NIFCliente	Filme Alugado	Data de Aluguer	Data Prevista Devolução	Dias em Atraso
▶	Contactar	Francisco	444444444	Matrix	22/04/2025	22/05/2025	15
	Contactar	Simão	123654987	Blade Runner 2049	30/04/2025	30/05/2025	7
	Contactar	Francisco Trinta	123321123	Get Out	30/04/2025	30/05/2025	7
	Contactar	Lúcia	222222222	Amélie	21/05/2025	04/06/2025	2
	Contactar	ze francisco	657765567	The Conjuring	22/05/2025	05/06/2025	1

Figura 16 - Clientes com Aluguer em Atraso

Para além da apresentação tabular, a interface foi enriquecida com funcionalidades que melhoram significativamente a usabilidade e a capacidade de resposta da gestão. Na parte superior da grelha, existem campos de pesquisa dedicados para Cliente e para Filme. Estes permitem ao utilizador introduzir termos específicos para filtrar dinamicamente os resultados apresentados, facilitando a localização rápida de alugueres em atraso



associados a um determinado cliente ou filme, sem necessidade de recarregar a lista completa.

Complementarmente, uma legenda de cores é visível na interface. Esta legenda serve como um sistema visual de alerta, onde diferentes cores aplicadas aos registos na grelha distinguem distintos estados ou níveis de criticidade do atraso, como por exemplo, atrasos recentes versus atrasos prolongados. Esta sinalização visual permite uma rápida compreensão da situação geral dos alugueres pendentes.

Uma funcionalidade de interatividade fundamental reside na capacidade de contactar o cliente diretamente a partir da grelha. Ao seleccionar um registo de aluguer em atraso, o sistema permite acionar uma funcionalidade que abre automaticamente um formulário de contacto. Este formulário de contacto (ilustrado na Figura 17) surge já pré-preenchido com as informações relevantes do cliente em questão, como o nome, NIF, telefone e email, recolhidas diretamente da base de dados sem necessidade de introdução manual. Após o preenchimento da mensagem, ao carregar no botão "Enviar", o programa irá utilizar o cliente de email predefinido no sistema operativo do utilizador para enviar a mensagem. Esta automação agiliza drasticamente o processo de follow-up e comunicação, otimizando a recuperação de filmes e a gestão de clientes.

**Contactar Cliente - Aluguer em Atraso**

**Notificação de Aluguer em Atraso**

**INFORMAÇÕES DO CLIENTE**

Nome: Francisco NIF: 444444444  
Filme: Matrix Dias em Atraso: 15

**Método de Contacto:**

☒ Email ☐ SMS

**Email:**  
francisco@gmail.com

**Mensagem:**  
Estimado(a) Francisco,  
  
Gostaríamos de relembrar que o filme "Matrix" alugado em 22/04/2025 deveria ter sido devolvido em 22/05/2025. Atualmente encontra-se com 15 dias de atraso.  
  
Por favor, proceda à devolução o mais breve possível para evitar custos adicionais.  
Agradecemos a sua atenção.  
  
Com os melhores cumprimentos,

**Enviar** **Cancelar**

*Figura 17 - Contactar Cliente*

A atualização dos dados na grelha principal é dinâmica e reflete o estado atual dos alugueres pendentes, garantindo que a lista de atrasos é sempre a mais recente e que a gestão dispõe de informações em tempo real para as suas ações.

Para a obtenção dos dados apresentados neste relatório, a aplicação interage com a base de dados utilizando a VIEW `vw_AlugueresEmAtraso`. A opção por uma view neste cenário otimiza a separação de preocupações: a lógica de identificação de alugueres em atraso reside no servidor de base de dados, enquanto a aplicação cliente apenas consulta essa view como se fosse uma tabela, simplificando o código na camada de apresentação.

A lógica interna da `vw_AlugueresEmAtraso` é a seguinte:

A view seleciona os campos necessários para o relatório, incluindo o Nome, NIF, Email e Telefone do cliente, o Título do filme, e as datas de `DataAluguer` e `DataPrevistaDev`. Adicionalmente, calcula a coluna `DiasEmAtraso` utilizando a função `DATEDIFF(day, a.DataPrevistaDev, GETDATE())`, que indica quantos dias se passaram desde a `DataPrevistaDev` até à data atual do sistema.

A seleção é realizada através de junções (JOINS) entre as tabelas Cliente, Aluguer e Filme para consolidar todas as informações relevantes de cada transação.

A filtragem (WHERE) é crucial para a finalidade da view:

- `a.DataPrevistaDev < GETDATE()`: Garante que apenas os alugueres cuja data prevista de devolução já passou são considerados.
- `AND a.DataDevolucao IS NULL`: Assegura que apenas os filmes que ainda não foram devolvidos são incluídos, evitando registos de alugueres que já foram regularizados mas que estiveram em atraso.

Esta view encapsula, portanto, a complexa lógica de identificação e agregação de dados de alugueres em atraso, tornando a recuperação de dados pela aplicação mais eficiente e direta.

## Conclusão

O desenvolvimento do Sistema de Gestão de Videoclube representou um projeto abrangente que demonstrou a aplicação prática de princípios fundamentais de engenharia de software e gestão de bases de dados. Desde a fase de modelagem conceptual até à implementação de uma interface funcional e responsiva, o objetivo central foi criar uma ferramenta robusta e eficiente para a gestão de um videoclube moderno.

A arquitetura do sistema foi concebida com uma clara separação de preocupações entre a camada de interface (Windows Forms em C#) e a camada de dados (SQL Server). Esta abordagem não só promove a modularidade e a manutenibilidade, mas também otimiza o desempenho e a segurança. A decisão de armazenar a connection string no ficheiro App.config ilustra uma boa prática de desenvolvimento, permitindo flexibilidade na configuração e protegendo credenciais sensíveis.

No nível da base de dados, a implementação da estrutura tabelar, a aplicação rigorosa de integridade referencial e validações de dados (CHECK CONSTRAINTS) asseguraram a consistência e a fiabilidade da informação. A utilização estratégica de Views (como `vw_AlugueresEmAtraso` e `vw_AlugueresPorCliente`) e Stored Procedures (como `GetAlugueresPorDataSP` para alugueres por intervalo de datas e as operações CRUD) foi crucial. Estas permitiram encapsular a lógica de negócio complexa, otimizar

a execução de queries através de pré-compilação, e fortalecer a segurança, minimizando a exposição a ataques de injeção de SQL.

A interface da aplicação foi desenhada a pensar na usabilidade e na eficiência operacional. Funcionalidades como a gestão abrangente de clientes e filmes (incluindo a integração com a API TMDb para preenchimento automático de dados de filmes), a gestão detalhada de alugueres e a visualização do histórico individual por cliente, foram implementadas para agilizar as operações diárias. Os relatórios estatísticos e as consultas avançadas – abrangendo os filmes mais alugados, filmes nunca alugados, alugueres por intervalo de datas, e a atividade de aluguer por cliente – fornecem à gestão insights valiosos para a tomada de decisões estratégicas. O relatório de "Clientes com Alugueres em Atraso", em particular, demonstra a capacidade de monitorização em tempo real e a funcionalidade de contacto automatizado, realçando a proatividade do sistema na gestão de pendências.

Em suma, este Sistema de Gestão de Videoclube representa uma solução completa e tecnicamente sólida, capaz de suportar as operações essenciais de um videoclube, fornecendo ferramentas eficazes para a gestão de dados, análise de desempenho e interação com o cliente. O projeto valida as metodologias de desenvolvimento aplicadas e a integração eficaz entre as diferentes camadas da aplicação.