

Desenvolvimento *de um* **SISTEMA** de gestão de uma **LIVRARIA** *em c#*

Edgar Rafael Stratulat de Sousa (28557)

Renato Filipe Carneiro Azevedo Azevedo (28549)

Linguagens de Programação I – 1º Semestre

Curso Técnico Superior Profissional em Desenvolvimento Web e Multimédia

Vila Nova de Famalicão, 22 de Janeiro de 2024

RESUMO

O presente relatório descreve o desenvolvimento de um programa em C# para gerir uma livraria, desde a definição de metas até à implementação das funcionalidades. O sistema inclui autenticação de utilizadores, registo e atualização de livros, gestão de stock, compras, vendas e consultas de dados específicos. Surgiram desafios na criação do menu principal e na gestão de privilégios de utilizador, mas garante uma experiência funcional para o utilizador. Apesar dos obstáculos, o sistema atingiu as expectativas, mostrando eficiência e robustez. Destaca-se a aplicação de conceitos teóricos, como ciclos, condições e tratamento de exceções aprendidos através das aulas e de trabalho de pesquisa. Reconhece-se áreas a melhorar, mas o projeto foi valioso para aprender e consolidar conhecimentos em C#, contribuindo para a compreensão aprofundada do desenvolvimento de software. Em conclusão, destaca-se a capacidade de resposta do sistema às necessidades de uma livraria, deixando espaço para melhorias futuras.

Palavras-Chave: GESTÃO DE LIVRARIA; C# ; DESENVOLVIMENTO DE SOFTWARE;
FUNCIONALIDADES DO SISTEMA;

ÍNDICE GERAL

1. INTRODUÇÃO.....	1
2. CARATERIZAÇÃO.....	3
3. ATIVIDADES REALIZADAS.....	5
a) Autenticação no sistema	5
b) Registar Livros	7
c) Consultar o Stock.....	9
d) Atualizar Livro	10
e) Consultar Livros através do código	11
f) Consultar Livros através do género literário	12
g) Consultar Livros através do autor.....	13
h) Comprar Livros - Acrescentar ao stock.....	14
i) Venda de Livros	15
j) Consultar o total de livros vendidos e a sua receita.....	16
k) Criar/Remover Utilizadores	18
l) Listar utilizadores do Sistema.....	19
5. CONCLUSÃO	21
6. REFERÊNCIAS BIBLIOGRÁFICAS	VII
ANEXOS.....	VIII
Anexo I - DIAGRAMA DE CASO DE USO	VIX

ÍNDICE DE FIGURAS

Figura 1 - Login na aplicação	5
Figura 2 - Erro no login	5
Figura 3 - Menu de seleção	5
Figura 4 - Permissões de acesso à informação.....	6
Figura 5 - Registar Livros.....	7
Figura 6 - Registar Livros.....	8
Figura 7 - Consultar o Stock.....	9
Figura 8 - Atualizar Livro	10
Figura 9 - Consultar Livros através do código	11
Figura 10 - Consultar Livros através do código - interação.....	11
Figura 11 - Consultar Livros através do género literário	12
Figura 12 - Consultar Livros através do Autor	13
Figura 13 - Comprar Livros	14
Figura 14 - Comprar Livros	15
Figura 15 - Consultar o total de livros vendidos e a sua receita	16
Figura 16 - Consultar o total de livros vendidos e a sua receita (erro)	17
Figura 17 – Criação de novo utilizador	18
Figura 18 – Remover Utilizador.....	18
Figura 19 – Listar utilizadores do sistema	19

ÍNDICE DE TABELAS

Tabela 1 - Permissões de Utilizador	7
--	----------

[ESTA PÁGINA ENCONTRA-SE EM BRANCO PROPOSITADAMENTE]

1. INTRODUÇÃO

O presente relatório descreve o processo de desenvolvimento de um programa em linguagem C#. Este projeto foi elaborado no âmbito da Unidade Curricular de Linguagens de Programação I, com o principal objetivo de consolidar os conhecimentos adquiridos durante as aulas e desafiar a capacidade de pesquisa e de autoaprendizagem.

Ao definirmos as expectativas iniciais, o foco centra-se na construção de um sistema robusto e intuitivo capaz de realizar diversas operações, desde o registo de livros até à gestão de utilizadores e transações de venda. O objetivo principal consiste em criar uma representação fidedigna das complexidades inerentes ao ambiente de uma livraria, distribuindo responsabilidades específicas a cada categoria de funcionário. Na construção do programa, as expectativas delineadas orientam as escolhas de implementação, motivam a exploração de conceitos específicos de C# e de lógica de programação, proporcionando assim uma visão clara dos objetivos a serem atingidos. Tudo para assegurar a eficiência na manipulação de dados e um controlo adequado de acesso aos mesmos.

O relatório que se apresenta está dividido em 3 capítulos: a caracterização, onde serão descritas detalhadamente as funcionalidades do sistema; as atividades realizadas, onde se explora o desenvolvimento do programa e suas funcionalidades específicas; e a conclusão, onde se estabelece uma comparação entre o que foi planeado e o que foi implementado, fazendo referência às principais dificuldades enfrentadas durante o trabalho.

[ESTA PÁGINA ENCONTRA-SE EM BRANCO PROPOSITADAMENTE]

2. CARATERIZAÇÃO

A criação do sistema de gestão da livraria exige uma análise cuidadosa das funcionalidades essenciais para atender às necessidades dos utilizadores e assegurar a eficiência nas operações. Nesta secção, apresentamos as principais funcionalidades do sistema. Estes requisitos funcionais são cruciais para construir um programa robusto e intuitivo, proporcionando uma experiência agradável para os colaboradores da livraria. Desde o login dos utilizadores, passando pela gestão do stock até às vendas, cada requisito foi pensado para refletir os objetivos do sistema e as expectativas dos utilizadores. Compreender estes requisitos é vital para a implementação bem-sucedida do sistema, assegurando a sua capacidade de dar suporte a todas as operações diárias da livraria. Posto isto, temos (ver anexo I – diagrama de caso de uso):

- O sistema deve permitir que os colaboradores realizem login com suas credenciais (nome de usuário e palavra-passe).
- O sistema deve possuir critérios de acesso à informação, de acordo com a função de cada utilizador;
- O sistema deve permitir o registo/remoção de novos funcionários ao sistema.
- O sistema deve permitir o registo de novos livros e incluir informações específicas sobre os mesmos.
- O sistema deve permitir a atualização dos dados de livros já registados no sistema.
- O programa deve permitir a consulta das informações do livro através do código, do autor, ou do género literário do livro.
- O utilizador gerente deve conseguir visualizar todos os utilizadores registados no sistema.
- O sistema deve permitir aumentar a quantidade de livros e todos os funcionários deverão ter acesso ao stock geral da livraria ou de um livro em específico.
- O sistema deve permitir efetuar vendas de livros, efetuando o respetivo cálculo do valor total em euros. Também deverá ter a capacidade de aplicar um desconto de 10% caso ultrapasse os 50 euros de receita.
- O sistema deve permitir consultar as receitas geradas pela venda de um livro em particular.

[ESTA PÁGINA ENCONTRA-SE EM BRANCO PROPOSITADAMENTE]

3. ATIVIDADES REALIZADAS

a) Autenticação no sistema

Quando o sistema é iniciado, surge uma mensagem de boas-vindas e um campo para o utilizador se autenticar com as suas credenciais, como vemos na figura seguinte.

Figura 1 - Login na aplicação

```
-----Bem-vindo ao Sistema de Registo da Livraria 'Páginas do Saber'-----  
Número de Utilizador:  
1  
Password:
```

Caso usemos um número de utilizador que não se encontre registado, uma mensagem de erro surge e permite que o utilizador tente novamente. (figura 2).

Figura 2 - Erro no login

```
-----Bem-vindo ao Sistema de Registo da Livraria 'Páginas do Saber'-----  
Número de Utilizador: 1  
Password: 222d  
Falha no Login. Número de Utilizador ou password incorretos. TENTE NOVAMENTE!  
Número de Utilizador:
```

Depois de o utilizador iniciar sessão, o sistema identifica a função do utilizador autenticado, através de uma mensagem de boas-vindas, e logo abaixo surge o menu para que o utilizador escolha o que pretende fazer (figura 3).

Figura 3 - Menu de seleção

```
Bem-vindo, Gerente!  
  
:::::POR FAVOR ESCOLHA UMA DAS SEGUINTE OPÇÕES:::::  
1- Registar Livros  
2- Atualizar Livro  
3- Consultar informações do livro, usando código  
4- Consultar livro por género  
5- Consultar livro por Autor  
6- Compra de Livros  
7- Vender Livros  
8- Consultar stock geral  
9- Consultar o total de livros vendidos e a sua receita  
10- Listar utilizadores do sistema  
11- Criar/Remover Utilizadores  
12- Sair  
:::::OPÇÃO:
```

Vamos agora mostrar os conceitos teóricos aplicados para conseguirmos este sistema de autenticação:

1. Criamos uma classe chamada “Livraria” e definimos os atributos da nossa classe.
2. Desenvolvemos dois construtores parametrizados, um para os livros e outro para utilizadores.
3. A partir dos construtores, criamos 2 listas na classe livraria: a lista Utilizador e a lista Livros.
4. De seguida, instanciamos estas mesmas listas para criação de objetos e possibilitar o armazenamento dos livros e dos utilizadores no sistema.
5. Foram criados 3 utilizadores (gerente, repositor, caixa) e adicionados 3 livros ao programa.

Os diferentes utilizadores possuem níveis de privilégios diferentes e é isso que vamos analisar a seguir. De acordo com os privilégios definidos, a responsabilidade de registar livros é do utilizador Repositor, por isso, quando o Gerente tenta aceder surge uma mensagem de ausência de permissões, tal como a figura 4 representa. Depois desta mensagem, é apresentado novamente este menu de opções para o utilizador seleccionar outra opção.

Figura 4 - Permissões de acesso à informação

```
Bem-vindo, Gerente!

:::::POR FAVOR ESCOLHA UMA DAS SEGUINTE OPÇÕES:::::
1- Registar Livros
2- Atualizar Livro
3- Consultar informações do livro, usando código
4- Consultar livro por género
5- Consultar livro por Autor
6- Compra de Livros
7- Vender Livros
8- Consultar stock geral
9- Consultar o total de livros vendidos e a sua receita
10- Listar utilizadores do sistema
11- Criar/Remover Utilizadores
12- Sair
:::::
OPÇÃO: 1
Não tem permissões para aceder a este módulo. Consulte o administrador do sistema.

-----
```

Para bloquearmos o acesso indesejado a algum menu, recorreremos ao switch-case e, dentro de cada caso, utilizamos a expressão condicional if. A tabela seguinte (tabela 1) pretende representar as permissões de acesso concebidas a cada utilizador.

Tabela 1 - Permissões de Utilizador

	Gerente	Repositor	Caixa
Registrar Livros			
Atualizar Livros			
Consultar Livros usando código			
Consultar Livros pelo género literário			
Consultar Livros pelo Autor			
Comprar Livros			
Vender Livros			
Consultar Stock			
Consultar Vendas e Receitas			
Listar Utilizadores			
Criar Remover Utilizador			

Legenda:

	Acesso permitido
	Acesso negado

b) Registrar Livros

A funcionalidade de Registrar Livros é da responsabilidade do Repositor, por isso, apenas este possui acesso. Depois do mesmo iniciar sessão e seleccionar a opção 1 do menu de selecção surge um campo para inserimos o código do livro e os dados do mesmo. Contudo, se o ID do livro já constar na lista Livros, uma mensagem é gerada, reiniciando assim o loop e voltando a pedir novamente o código do livro (figura 5). Se o utilizador digitar '0', voltamos ao menu de selecção.

Figura 5 - Registrar Livros

```
OPÇÃO: 1
Opção 1 Escolhida | REGISTRAR LIVRO
Insira o código do livro (ou '0' para sair):
Z04
Qual o seu id:
3
Já existe um livro com esse ID. Por favor, tente novamente.
Insira o código do livro (ou '0' para sair):
```

Depois de inserirmos todos os dados, é gerada uma mensagem que o livro foi adicionado ao sistema com sucesso (Figura 6).

Figura 6 - Registrar Livros

```
OPÇÃO: 1

Opção 1 Escolhida | REGISTRAR LIVRO
Insira o código do livro (ou '0' para sair):
Z04
Qual o seu id:
4
Insira o título:
Branca de Neve e os 7 Anões
Insira o autor:
Wilhelm Grimm
ISBN:
33333
Insira o género:
Contos Infantis
Qual o preço?
7,5
Qual a taxa de Iva(6%, 23%)?
0,06
Qual a quantidade a registar?
5
Livro registado com sucesso!
```

Depois de o livro estar registado, isso leva-nos a uma outra opção do menu de seleção, a opção 8, que será abordado a seguir.

Antes de avançarmos para a próxima etapa, convém enumerar os pressupostos teóricos invocados no processo de registar o livro no sistema:

1. Instanciamos um novo objeto da classe Livraria;
2. Utilização do loop While (enquanto o código do livro inserido for diferente de "0", o utilizador pode continuar a adicionar livros).
3. Uso de condições e expressões condicionais para controlar os dados inseridos pelo utilizador, como:
 - a. `Livros.Any (livro => livro.Codigo == registrarlivro.Codigo)`: devolve 'true' se a condição for verificada e informa o utilizador que o código inserido já existe.
 - b. `Livros.Add(registarlivro)` ⇒ Adiciona o livro registado à lista Livros.
4. Informamos o utilizador do sucesso da operação.

c) Consultar o Stock

O registo do livro feito anteriormente, pode ser consultado nesta secção e daí não seguirmos a ordem do menu de seleção. Quando voltamos ao menu e selecionamos a opção 8 é nos apresentado o stock disponível. E realmente confirmamos que o livro que adicionamos anteriormente ficou inserido na lista (Figura 7).

Figura 7 - Consultar o Stock

```
OPÇÃO: 8
Opção 8 Escolhida | CONSULTAR O STOCK
-----
ID: 1 | Título: O Rapaz do Pijama às Riscas | Autor: John Boyne | Stock: 100
-----
ID: 2 | Título: O Último Grimm | Autor: Álvaro Magalhães | Stock: 4
-----
ID: 3 | Título: As fadas Verdes | Autor: Matilde Rosa Araújo | Stock: 5
-----
ID: 4 | Título: Branca de Neve e os 7 Anões | Autor: Wilhelm Grimm | Stock: 5
-----
```

Para conseguirmos listar os elementos da Lista livros, recorreremos à estrutura *'foreach'* para iterar os elementos da nossa lista que se encontra dentro da nossa classe *'Livraria'*. Por isso, obtivemos uma estrutura assim: *'foreach (Livraria livros in Livros)'*. Durante a iteração, cada objeto é representado por *'livros'*. Quando termina, voltamos ao menu de seleção.

d) Atualizar Livro

Supondo agora que queremos alterar o preço de um livro. O livro com ID 1 sofreu um aumento. É aqui que entra esta função. Seleccionando a opção 2, é nos questionado que livro precisamos de alterar e são mostrados os parâmetros para substituir as informações. Como queremos apenas alterar apenas o preço do livro, mantemos todas as informações à exceção do preço. No final surge a lista com o preço do nosso livro atualizado.

Figura 8 - Atualizar Livro

```
Opção 2 Escolhida | ATUALIZAR LIVRO
Qual o ID que pretende alterar? (ou '0' para sair)1
Novo código: Z001
Novo Título: O Rapaz do Pijama às Riscas
Novo Autor: John Boyne
Novo Género: Ação
Novo ISBN: 978972415
Novo Stock: 100
Novo Preço: 15,00
Novo Iva: 0,06

Livro atualizado com sucesso!
Código: Z001 | Título: O Rapaz do Pijama às Riscas | Autor: John Boyne | Género: Ação | ISBN: 978972415 | Stock: 100 |
Preço: 15 | Iva: 0,06
Código: Z02 | Título: O Último Grimm | Autor: Álvaro Magalhães | Género: Romance | ISBN: 978976567 | Stock: 4 | Preço: 1
4,4 | Iva: 0,06
Código: Z03 | Título: As fadas Verdes | Autor: Matilde Rosa Araújo | Género: Poesia | ISBN: 987345678 | Stock: 5 | Preço
: 8,85 | Iva: 0,23
Qual o ID que pretende alterar? (ou '0' para sair)
```

Para esta etapa, usamos:

1. Utilização de um loop infinito "while(true)" para que as nossas condições sejam cumpridas até serem interrompidas pela instrução "break".
2. Pedimos ao utilizador para inserir o ID do livro que pretende atualizar e usamos a expressão "`int.TryParse(Console.ReadLine(), out id`" para converter para um número inteiro a variável introduzida e armazenar na variável ID. Caso digite '0', o sistema volta ao menu e terá de repetir o processo.
3. Uso da função *Find* encontramos os livros na lista Livros.
4. Verificamos se o ID inserido está na lista, através da expressão "`if (Livro != null)`"
5. Recurso ao bloco try-catch para tratar exceções ao converter o ISBN
6. No final dos dados inseridos, utilizamos o *foreach* para percorrer todos os livros da lista "Livros" e os exibi-los na consola após as atualizações.

Agora queremos verificar se realmente, o preço deste livro foi alterado. Para isso, recorreremos à opção 3 do menu para consultar um livro através do código.

e) Consultar Livros através do código

Para verificarmos, se de facto, o livro ficou registado vamos explorar a opção 3 do menu e fazer uma pesquisa do livro através do seu código. Quando entramos nesta secção e introduzimos o código do livro 'Z001' verificamos que, de facto, este tem um novo preço (figura 9).

Figura 9 - Consultar Livros através do código

```
OPÇÃO: 3
Opção 3 Escolhida | CONSULTAR LIVROS ATRÁVES DO CÓDIGO
Introduza o código do livro que pretende consultar: (ou 'sair' para encerrar)
Z001

Código: Z001
Título: O Rapaz do Pijama às Riscas
Autor: John Boyne
Género: Ação
ISBN: 978972415
Stock: 100
Preço: 15?
Iva: 0,06%

Introduza o código do livro que pretende consultar: (ou 'sair' para encerrar)
```

Caso o utilizador insira, um código de um livro que não esteja no stock, o programa informa e pede uma nova tentativa (figura 10).

Figura 10 - Consultar Livros através do código - interação

```
Introduza o código do livro que pretende consultar: (ou 'sair' para encerrar)
5
O código inserido não é válido. Tente novamente!
Introduza o código do livro que pretende consultar: (ou 'sair' para encerrar)
```

Para conseguirmos esta interatividade, usamos os seguintes pressupostos teóricos:

1. Colocamos o loop infinito 'While' dentro de um try-catch para lidar com exceções;
2. Recurso ao loop while(true)
3. Criação de variável code "vazia" para armazenar o código que o utilizador digitar
4. Solicitar ao utilizador para introduzir o código do livro que pretende consultar através do console ReadLine e converter para string o digitado (*code = Convert.ToString(Console.ReadLine())*)
5. Uso da função Find para encontrar o livro na lista Livros (*var Livro = Livros.Find(livro => livro.Codigo == code)*). A função Find destaca-se pela eficiência na procura de um único elemento numa lista, interrompendo prontamente a busca após encontrar o primeiro elemento que atende à condição do predicado.
6. Recurso à expressão condicional if (se o código inserido corresponder a algum livro da lista, são mostradas as informações do livro, senão o programa pede para introduzir novamente um código (até que o utilizador digite 'sair')).

f) Consultar Livros através do género literário

Neste momento, queremos consultar se a nossa livraria possui em stock, livros do género literário Romance. Para isso seleccionamos a opção 4 e escrevemos 'Romance'. E os resultados mostram que, temos 4 livros do mesmo autor do género literário pretendido (Figura 11).

Figura 11 - Consultar Livros através do género literário

```
OPÇÃO: 4
Opção 4 Escolhida | CONSULTAR LIVROS ATRÁVES DO GÉNERO
Insira o género que deseja consultar: (ou escreva 'sair' para encerrar o programa)Romance

Código: Z02
Título: O Último Grimm
Autor: Álvaro Magalhães
Género: Romance
ISBN: 978976567
Stock: 4
Preço: 14,4?
Iva: 0,06%

Insira o género que deseja consultar: (ou escreva 'sair' para encerrar o programa)_
```

Para conseguirmos consultar por género literário, baseamo-nos no seguinte:

- Utilização de um loop while(true) para permitir consultas consecutivas de livros por gênero até que o utilizador decida sair;
- Questionar o utilizador qual o género que pretende consultar;
- Converter o que o utilizador inseriu para uma variável do tipo *string* através da linha (*genero = Convert.ToString(Console.ReadLine());*);
- Recurso ao 'if' para encerrar o programa, caso o utilizador digite 'sair'
- Uso da função Find para encontrar o livro na lista
- Recurso à expressão condicional IF para tratamento de resultados: se o género inserido corresponder a algum livro da lista, são mostradas as informações do livro, senão o programa pede para introduzir novamente o género.
- Em caso afirmativo, são mostrados todos os livros daquele género literário.

g) Consultar Livros através do autor

Do nosso repositório, queremos averiguar se existe um livro da autora 'Matilde Rosa Araújo'. Vamos experimentar... para isso recorreremos à opção 5 do menu de seleção (figura 12).

Figura 12 - Consultar Livros através do Autor

```
OPÇÃO: 5
Opção 5 Escolhida | CONSULTAR LIVROS ATRÁVES DO AUTOR

Escreva o nome do autor que pretende consultar: (ou escreva 'sair' para encerrar o programa)Matilde Rosa Araújo

Código: Z03
Título: As fadas Verdes
Autor: Matilde Rosa Araújo
Género: Poesia
ISBN: 987345678
Stock: 5
Preço: 8,85?
Iva: 0,23%

Escreva o nome do autor que pretende consultar: (ou escreva 'sair' para encerrar o programa)Renato Azevedo
O nome do autor introduzido não foi encontrado na lista. Tente novamente!

Escreva o nome do autor que pretende consultar: (ou escreva 'sair' para encerrar o programa)_
```

Neste caso, encontramos o livro 'As Fadas Verdes' desta autora. Se introduzir um autor que não conste na lista, então o sistema alerta e pede para tentar novamente (figura 12). Nesta secção, foram usados pressupostos teóricos idênticos aos usados na secção anterior e que analisamos anteriormente.

h) Comprar Livros - Acrescentar ao stock

O modo comprar livros permite acrescentar stock aos livros já existentes na nossa livraria. Posto isto, o programa pede primeiramente o ID dos livros que pretende acrescentar. Caso o livro não conste no nosso repositório o sistema pede uma nova tentativa. No caso de existir um livro com o ID que foi digitado, o sistema prossegue com a compra (Figura 13).

Figura 13 - Comprar Livros

```
OPÇÃO: 6
Opção 6 Escolhida | COMPRAR LIVROS - ACRESCENTAR STOCK

Qual o ID do livro que pretende? (ou escreva '0' para encerrar o programa)
2
Que quantidade deseja comprar para O Último Grimm? 10

Compra bem-sucedida! Stock atualizado para 14 unidades.

Temos então:
Código: Z02
Título: O Último Grimm
Autor: Álvaro Magalhães
Gênero: Romance
ISBN: 978976567
Stock: 14
Preço: 14,4?
Iva: 0,06%

Qual o ID do livro que pretende? (ou escreva '0' para encerrar o programa)
```

O sistema adiciona ao stock existente a quantidade adquirida. A título de exemplo, caso previamente tenhamos 4 unidades em stock do livro 'O Último Grimm', se comprarmos 10, o sistema atualiza para 14 unidades desse mesmo livro. Ao acrescentarmos livros ao stock, isso vai -se refletir na secção 7 do menu de seleção - a venda de livros. Mas isso, abordaremos a seguir.

Primeiro, vamos enumerar os conceitos aplicados neste método:

1. Utilização do loop `while(true)`;
2. Questionar o utilizador qual o id do livro que pretende comprar;
3. Armazenar o ID inserido pelo utilizador e convertê-lo para um número inteiro - `'int id = Convert.ToInt32(Console.ReadLine())'`.
4. Se o ID introduzido for '0' interrompe o loop e volta ao menu inicial, caso exista o
5. Utilização da função Find para encontrar os livros por ID na nossa lista de livros
6. Se o livro com o ID fornecido não for encontrado (`livro == null`), o programa exibe uma mensagem indicando que o ID é inválido e solicita novamente ao utilizador que insira um ID válido.
7. Questionamos a quantidade e aqui o sistema, converte para um número inteiro positivo e verifica se a quantidade introduzida é maior do que zero. No caso de corresponder, adiciona a quantidade ao stock do livro existente e mostra essas informações na tela:

8. O stock do livro é atualizado somando a quantidade adquirida ($livro.Stock += Quantidade$), refletindo com precisão a nova quantidade disponível.

i) Venda de Livros

Vimos na secção anterior que ficamos com 14 unidades do livro com ID 2. Agora este modo possibilita a venda dos mesmos e faz o respetivo cálculo do valor a pagar, ou seja, remove livros do stock. Vamos ver se realmente estes livros estão disponíveis para venda. Caso o colaborador queira vender mais unidades do que as registadas em sistema, surge uma mensagem a alertar do sucedido (Figura 14).

Outro aspeto, a política interna da livraria define a aplicação de 10% em compras superiores a 50€.O sistema reconheceu que o valor da compra excedeu os 50 euros e aplicou automaticamente esse desconto, discriminando o valor do IVA, o desconto efetuado e o total.

Figura 14 - Comprar Livros

```
OPÇÃO: 7

Opção 7 Escolhida | MODO DE VENDA DE LIVROS

Qual o ID do livro que deseja vender? (ou '0' para encerrar
2
Quantidade a ser vendida: 12
::::::::::::::::::::RESUMO DE VENDA::::::::::::::::::::
Subtotal: 172,8 euros
Valor do IVA: 10,37 euros
Desconto: 10%
Total: 164,85 euros
::::::::::::::::::::VENDA CONCLUÍDA COM SUCESSO::::::::::::::::::::

Qual o ID do livro que deseja vender? (ou '0' para encerrar
2
Quantidade a ser vendida: 3
ERRO! A quantidade a ser vendida excede o número de livros em stock

Qual o ID do livro que deseja vender? (ou '0' para encerrar
```

Fundamentando, temos:

1. Utilização de um loop while(true) para permitir operações de venda consecutivas até que o utilizador decida encerrar.
2. Pede ao utilizador para inserir o ID do livro que deseja vender.
3. Utilização da função Find para localizar o livro na lista com base no ID fornecido pelo utilizador. Se o livro com o ID não for encontrado, o programa exibe uma mensagem de ID inválido e pede novamente ao utilizador.

4. Questiona o utilizador sobre a quantidade desejada para a venda. Verifica se a quantidade desejada não excede o número de livros em stock. Se exceder, solicita novamente a quantidade.
5. Reduz o stock do livro na lista com base na quantidade vendida.
6. Apresenta um resumo detalhado da venda, incluindo subtotal, valor do IVA, desconto aplicado (se houver) e o total final.

```
livro.Total = (livro.Quantidade * livro.Preco);  
livro.Novoiva = (livro.Total * livro.Iva);  
livro.Novototal = (livro.Total + livro.Novoiva);
```

7. Aplica um desconto de 10% ao total se o valor da venda for igual ou superior a 50 euros.
8. Conclui a venda com sucesso.

j) Consultar o total de livros vendidos e a sua receita

Supondo agora que o colaborador caixa vendeu 4 exemplares do livro com ID 2. No final do dia, queremos o resumo de vendas desse livro. Para isso, basta irmos à opção 9 do menu e verificarmos a faturação. Vamos testar... (figura 15).

Figura 15 - Consultar o total de livros vendidos e a sua receita

```
OPÇÃO: 9  
Opção 9 Escolhida | CONSULTAR O TOTAL DE LIVROS VENDIDOS E A SUA RECEITA  
  
Insira o id do livro que deseja consultar o total vendido e a respetiva receita: (0 para encerrar)  
2  
Livro ID: 2  
Título: O Último Grimm  
Quantidade Vendida: 4  
Receita Total: 57,6 euros  
Insira o id do livro que deseja consultar o total vendido e a respetiva receita: (0 para encerrar)
```

E sim, através daqui comprovamos que foram vendidos 4 livros com ID 2. Para além da quantidade é apresentada a receita gerada por esse livro em euros. Na consulta de um novo livro, o colaborador enganou-se e inseriu um ID de um livro que não existia na base de dados e o sistema alertou quando a essa situação, pedindo novamente para inserir um id (figura 16).

Figura 16 - Consultar o total de livros vendidos e a sua receita (erro)

```
Insira o id do livro que deseja consultar o total vendido e a respetiva receita: (0 para encerrar)
4
Livro não encontrado.
Insira o id do livro que deseja consultar o total vendido e a respetiva receita: (0 para encerrar)
```

Portanto, em termos teóricos temos:

1. Utilização de loop while(true);
2. Questionar o utilizador sobre o ID do livro desejado para consulta da receita;
3. O ID inserido pelo utilizador é convertido para um número inteiro usando 'Convert.ToInt32(Console.ReadLine()).';
4. Utilização da função Find para localizar o livro na lista;
5. Se o ID inserido corresponder a algum livro da lista, o código atualiza a quantidade total de livros vendidos (totallivrosvendidos) ao adicionar a quantidade vendida do livro.

```
int totallivrosvendidos = 0; // variável iniciada com zero e armazena a quantidade de livros vendidos.
totallivrosvendidos += Livro.Quantidade; // Adicionar à variável a quantidade vendida
```

6. No momento da consulta, são exibidas na consola informações pormenorizadas sobre o livro em questão, abrangendo o ID, título, quantidade vendida e receita total.

É aplicado um bloco try-catch para lidar com situações excepcionais, mostrando uma mensagem de erro caso ocorra alguma exceção durante a execução do método.

k) Criar/Remover Utilizadores

O gerente decidiu contratar um contabilista para a equipa da Livraria 'Páginas do Saber' e necessitou de criar uma autenticação em sistema para que este consiga aceder aos dados do sistema e efetuar o seu trabalho contabilístico. O gerente teve de usar a opção 11 do menu de seleção e inserir os dados pedidos pelo sistema (número do utilizador, função e password), como representa a figura seguinte (figura 17).

Figura 17 – Criação de novo utilizador

```
OPÇÃO: 11
Opção 11 Escolhida | CRIAR / REMOVER UTILIZADORES
Escolha uma opção ('Criar' 'Remover' e Sair para encerrar):
Criar
Número de Utilizador: 4
Função: Contabilista
Password: 12contas

Utilizador criado com sucesso!
Número de Utilizador: 4 | Função: Contabilista | Password: 12contas
```

Nota: Na eventualidade de querer remover um dos utilizadores do sistema, teria também essa possibilidade (Figura 18)

Figura 18 – Remover Utilizador

```
OPÇÃO: 11
Opção 11 Escolhida | CRIAR / REMOVER UTILIZADORES
Escolha uma opção ('Criar' 'Remover' e Sair para encerrar):
Remover
Insira o ID do utilizador que deseja remover:
2
Utilizador removido com sucesso!

Opção 11 Escolhida | CRIAR / REMOVER UTILIZADORES
Escolha uma opção ('Criar' 'Remover' e Sair para encerrar):
```

Vamos compreender agora, os pressupostos teóricos aplicados anteriormente:

1. Utilização de um loop *while* (*true*) para manter o utilizador no menu até que a opção de "Sair" seja escolhida.
2. O utilizador é convidado a inserir uma opção, que pode ser "Criar", "Remover" ou "Sair".
3. Criação de um novo utilizador,
 - o *if* (*Opcao == "Criar"*)
 - ii. Solicita informações como número de utilizador, função e password.
 - iii. Realiza uma validação para garantir que o número de utilizador (ID) inserido seja diferente de zero. Caso seja, ele volta também ao menu de seleção;

- iv. Adiciona o novo utilizador à lista chamada Utilizador através do excerto de código: `' Utilizador.Add(novoutilizador) '` ;
- o *else if (Opcao == "Remover")*
 - i. Se o utilizador escolher "Remover", solicita o ID do utilizador a ser removido.
 - ii. Itera sobre a lista Utilizador procurando um utilizador com o ID correspondente e remove-o se encontrado. (`Utilizador.RemoveAt(i)`);

Se o utilizador escolher "Sair", o loop é quebrado, e o método retorna ao menu principal.

I) Listar utilizadores do Sistema

Vimos anteriormente, que foi adicionado um acesso ao sistema de um novo utilizador com função de contabilista. O gerente agora, quer confirmar se realmente houve sucesso na criação do novo utilizador. Para isso, tem de escolher a opção 10 do menu e listar os utilizadores do sistema. Se fizermos isso, verificamos que, realmente, o utilizador com ID número 4 foi adicionado, e por isso um novo acesso foi criado (Figura 19).

Figura 19 – Listar utilizadores do sistema

```
OPÇÃO: 10
Opção 10 Escolhida | LISTAR UTILIZADORES DO SISTEMA
ID de utilizador: 1 | Password: 123gerente | Função: Gerente.
ID de utilizador: 2 | Password: 222repositor | Função: Repositor.
ID de utilizador: 3 | Password: 999caixa | Função: Caixa.
ID de utilizador: 4 | Password: 12contas | Função: Contabilista.
```

Para isto acontecer, foram seguidos os seguintes passos:

1. Utilizamos a estrutura *foreach* para iterar através da lista de utilizadores (Utilizador) presente na classe Livraria.
2. Para cada utilizador no sistema, são apresentadas informações relevantes: ID de utilizador, a password e a função.
3. No final, o menu é apresentado novamente ao utilizador.

[ESTA PÁGINA ENCONTRA-SE EM BRANCO PROPOSITADAMENTE]

5. CONCLUSÃO

Este relatório detalhou as fases do desenvolvimento do software em C#, desde a definição dos objetivos até à implementação das diversas funcionalidades.

Ao longo do desenvolvimento deste projeto, enfrentamos desafios significativos que contribuíram para o nosso crescimento e compreensão aprofundada do processo de programação em C#. A identificação e aplicação de conceitos teóricos previamente desconhecidos mostraram-se como uma etapa desafiadora, exigindo pesquisa aprofundada e dedicação para assimilar estes novos conhecimentos.

No que diz respeito aos conceitos teóricos, a utilização de ciclos e condições, a forma de organizar dados (como por exemplo, a organização do nosso código-fonte por regiões) e o tratamento de situações excecionais foi crucial para tornar o sistema eficiente e robusto.

Um dos desafios na conceção do sistema foi a elaboração do menu principal, especialmente no que diz respeito à manutenção dos privilégios de utilizador. A complexidade associada à criação de um sistema que preservasse esses privilégios ao regressar ao menu inicial revelou-se uma tarefa desafiadora, sendo identificada como uma área que exigiu considerável esforço. No entanto, reconhecemos que esse ponto específico pode ser aprimorado em futuras versões do sistema.

A expectativa inicial de criar um sistema que conseguisse responder às necessidades de um ambiente de livraria e atribuisse funções a cada colaborador no sistema, podemos dizer que foi superada com sucesso, uma vez que o sistema é funcional. Contudo, não descartamos a possibilidade de realizar melhorias no código-fonte, a fim de melhorar a interatividade com o utilizador e a resposta do sistema perante algumas situações.

Apesar do código fonte e do presente relatório não serem perfeitos, tornaram-se objetos de aprendizagem bastante eficazes para melhorarmos e compreendermos a linguagem C#, consolidar os conhecimentos que foram lecionados na unidade curricular durante todo o semestre e fomentar o trabalho em equipa.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Carlucci, A., et al. (2021). *C# para Iniciantes*. Obtido em 29 de Dezembro de 2023, de <https://livrocsharp.com.br/wp-content/uploads/dae-uploads/CSharpIniciantes.pdf>

Tavares, A. et al. (2015). *C# Essencial* (Material de Apoio IPCA).

ANEXOS

Anexo I - DIAGRAMA DE CASO DE USO

