

LABORATÓRIO DE APLICAÇÕES WEB E BASES DE DADOS

RELATÓRIO

DO PROJETO

Francisco Ruano, al78474
Maria Inês Cardoso, al78222
Miguel Teixeira, al78321
Rui Pedro Madureira, al78282

Sumário

INTRODUÇÃO	4
1- FASE 1	5
1.1 Análise dos Requisitos de Dados do Sistema	5
1.1.1 Entidades	5
1.1.2 Relacionamentos.....	6
1.2 Especificação do Modelo Conceptual de Dados do Sistema.....	7
1.3 Análise dos Requisitos Funcionais e Não Funcionais do Sistema	8
1.3.1 Requisitos Funcionais (RF).....	8
1.3.2 Requisitos Não Funcionais (RNF).....	11
1.4 Especificação do Modelo Funcional.....	13
2- FASE 2	14
2.1 Mapeamento entre o Modelo Conceptual de Dados e o Modelo Relacional	14
2.2 Implementação do Modelo físico da base de dados recorrendo à linguagem SQL.....	16
2.3 Especificação das interfaces com o utilizador para o <i>Backoffice</i> e <i>Frontoffice</i> recorrendo a <i>Mockups</i>	20
2.3.1 <i>Frontoffice</i>	20
2.3.2 <i>Backoffice</i> (Bibliotecário)	31
2.3.3 <i>Backoffice</i> (Administrador).....	39
3- FASE 3	46
3.1 Autenticação no Sistema.....	47
3.1.1 Registo de Utilizadores.....	47
3.1.2 <i>Login</i>	48
3.1.3 Gestão de Acessos	50
3.2 <i>Frontoffice</i>	51
3.2.1 Requisição de Livros	51

3.3	<i>Backoffice</i>	52
3.3.1	Bibliotecário	52
3.3.2	Administrador	54
3.4	Integração com a Base de Dados	55
CONSIDERAÇÕES FINAIS		56
ANEXOS		57
1.1	Anexo A – Tabela ‘<i>AspNetUsers</i>’	57
1.2	Anexo B – Tabela ‘<i>AspNetRoles</i>’	57
1.3	Anexo C – Tabela ‘<i>AspNetUserRoles</i>’	57

INTRODUÇÃO

No âmbito da unidade curricular de Laboratório de Aplicações *Web* e Base de Dados foi-nos proposto o desenvolvimento de uma aplicação *web* capaz de suportar a gestão da Biblioteca LEI-UTAD.

O presente relatório tem como objetivo descrever a execução das diferentes fases do projeto a desenvolver que permitiram uma evolução progressiva desde a análise de requisitos até à implementação completa do sistema.

Na primeira fase, o foco esteve na análise e modelação dos requisitos do sistema. Foram identificadas as entidades, relacionamentos e funcionalidades principais, culminando na criação do diagrama Entidade-Relacionamento (E-R) e na especificação dos requisitos funcionais e não funcionais. Esta etapa estabeleceu uma base teórica sólida para o desenvolvimento subsequente.

Na segunda fase, o projeto avançou para o mapeamento do modelo conceptual para o modelo físico de dados, utilizando *SQL* para implementar a Base de Dados. Além disso, foram criados *Mockups* das interfaces de utilizador para o *frontoffice* e o *backoffice*, permitindo visualizar e planear a interação dos utilizadores com o sistema de forma clara e intuitiva.

Finalmente, na terceira fase, realizou-se a implementação prática do sistema, integrando os módulos de *frontoffice*, *backoffice* e Base de Dados. Esta etapa envolveu o desenvolvimento de interfaces funcionais, a configuração de autenticação personalizada e a validação rigorosa das funcionalidades implementadas, garantindo um sistema robusto, fiável e escalável, adequado para todos os tipos de utilizadores.

Este relatório detalha as principais etapas do projeto, destacando os desafios superados, as soluções adotadas e os resultados alcançados, com o objetivo de proporcionar uma visão abrangente do desenvolvimento e da aplicação prática das competências adquiridas.

1- FASE 1

1.1 Análise dos Requisitos de Dados do Sistema

1.1.1 Entidades

Autenticado (Username, Password, Contacto, Contacto2, Contacto3, Nome)

Bibliotecario (Username_Bib)

- Username_Bib referencia Autenticado

Leitores (Username_Lei, Morada, Email)

- Username_Lei referencia Autenticado

Administradores (Username_Admin, Nome, Email, Password)

Biblioteca (ID_Biblioteca, HorarioI, HorarioF, Localizacao, Nome, Email, Telefone)

Livro (ISBN, Preco, Titulo, N_Exemplares, Genero, Username_BibAdd, ID_Autor, ID_Biblioteca)

- Username_BibAdd referencia Bibliotecario
- ID_Autor referencia Autor
- ID_Biblioteca referencia Biblioteca

Autor (ID_Autor, Name_Autor)

Requisicao (ID_Requisicao, Username_Bib, Username_Lei, ISBN, Data_Requisicao, Data_Devolucao, Estado)

- UsernameBib referencia Bibliotecario
- UsernameLei referencia Leitores
- ISBN referencia Livro

HistoricoRequ (ID_Historico, ID_Requisicao)

- ID_Requisicao referencia Requisicao

EntregasAtraso (ID_Entregas, ID_Requisicao)

- ID_Requisicao referencia Requisicao

MotivoBloq (ID_MotivoBloq, Descricao)

Bloquear (ID_Bloqueio, ~~ID_MotivoBloq~~, ~~Username_Admin~~, Username, Data_Bloqueio, Bloqueado)

- ID_MotivoBloq referencia MotivoBloq
- Username_Admin referencia Administrador
- Username referencia Autenticado

1.1.2 Relacionamentos

Cria_Admin (~~Username_Admin~~, ~~Username_NovoAdmin~~)

- Username_Admin referencia Administradores
- Username_NovoAdmin referencia Administradores

ValidarRegisto (ID_Valid, Data_Valid, Validado, ~~Username_Admin~~, ~~Username_Bib~~)

- Username_Bib referencia Bibliotecario
- Username_Admin referencia Administradores

1.2 Especificação do Modelo Conceptual de Dados do Sistema

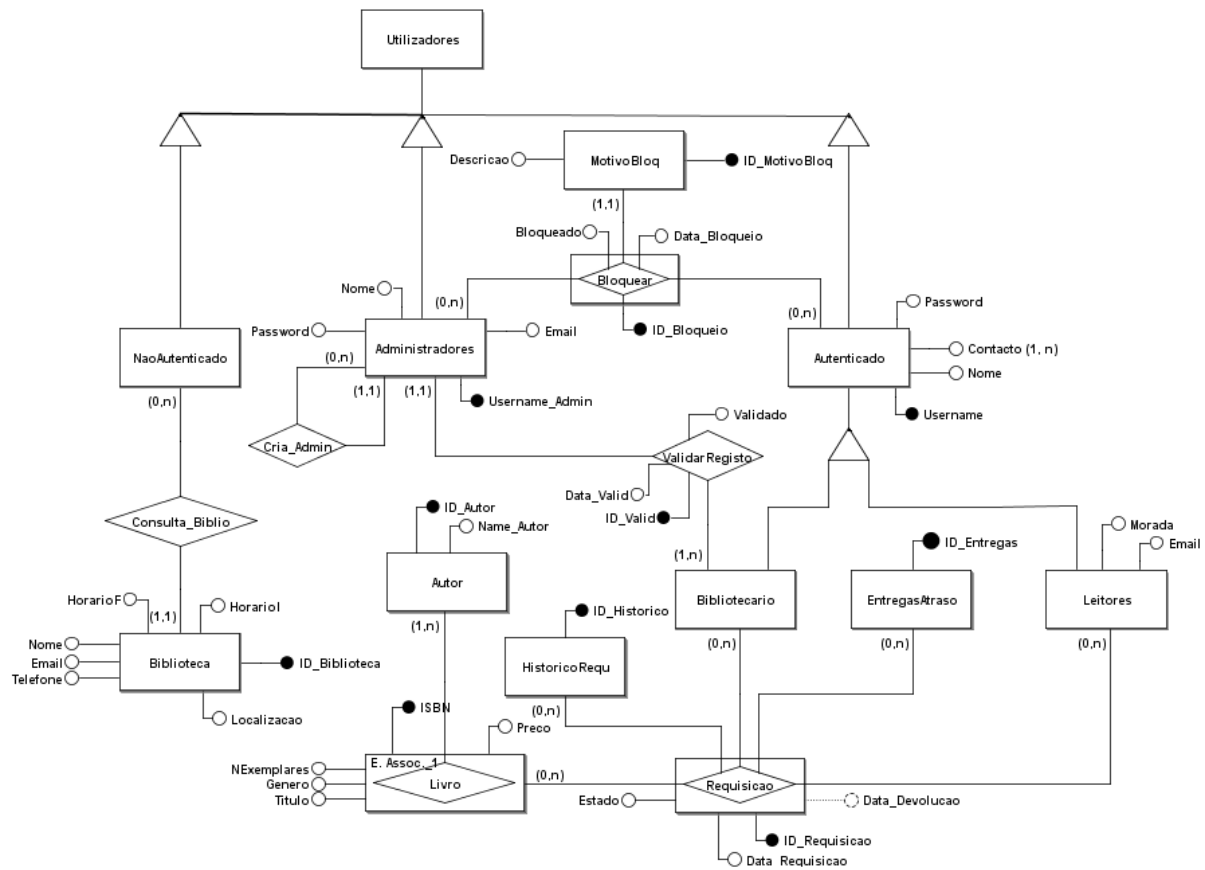


Figura 1- Diagrama Entidade-Relacionamento (E-R)

1.3 Análise dos Requisitos Funcionais e Não Funcionais do Sistema

1.3.1 Requisitos Funcionais (RF)

1.3.1.1 Autenticação

RF1 → Registo de Utilizador

- **Descrição:** O sistema deve permitir o registo de utilizadores através da introdução de informações pessoais, como nome, e-mail, *username*, palavra-passe, morada e contacto. Os leitores recebem um e-mail de confirmação, enquanto os bibliotecários requerem aprovação de um administrador.

RF2 → Autenticação de Credenciais

- **Descrição:** O sistema deve verificar a identidade do utilizador com base nas credenciais fornecidas, como *username* e palavra-passe.

RF3 → Recuperação de Palavra-Passe

- **Descrição:** Os utilizadores devem poder redefinir a sua palavra-passe em caso de esquecimento ou perda da mesma.

RF4 → Gestão de Sessão

- **Descrição:** O sistema deve gerir as sessões de utilizador de forma segura, garantindo que os utilizadores não permanecem ligados indefinidamente, fornecendo assim opções para encerrar a sessão.

RF5 → Notificações de Segurança

- **Descrição:** O sistema deve enviar notificações de segurança, via e-mail ou pela própria aplicação, quando se detetarem atividades suspeitas, como tentativas falhadas de *login* repetidas, ativação ou bloqueio de contas.

1.3.1.2 Parte Pública (Utilizadores Não Autenticados)

RF6 → Consulta de Informações da Biblioteca

- **Descrição:** O sistema deve permitir que os utilizadores não autenticados consultem informações públicas sobre a biblioteca, como nome, localização, e-mail, telefone e horário de funcionamento.

RF7 → Listagem de Livros

- **Descrição:** O sistema deve disponibilizar a visualização de uma listagem de livros com detalhes como título, autor, género, preço, ISBN e número de exemplares disponíveis.

1.3.1.3 Parte Privada (Utilizadores Autenticados)

RF8 → Consulta histórico e livros requisitados

- **Descrição:** O sistema deve permitir aos utilizadores consultem o seu histórico de livros requisitados.

RF9 → Pesquisa de Livros

- **Descrição:** O sistema deve permitir que os leitores, quando autenticados, pesquisem livros por título, autor e género.

RF10 → Requisição de Livros

- **Descrição:** O sistema deve permitir que os leitores requisitem livros, com um prazo máximo de devolução de quinze dias.

RF11 → Histórico de Requisições

- **Descrição:** O sistema deve disponibilizar aos leitores o seu histórico de livros requisitados.

RF12 → Gestão de Livros

- **Descrição:** O sistema deve permitir que os bibliotecários adicionem, editem ou removam livros do catálogo da biblioteca.

RF13 → Gestão de Empréstimos

- **Descrição:** O sistema deve permitir que os bibliotecários acompanhem o estado dos empréstimos, sabendo que leitores têm livros emprestados, bem como a sua data de entrega.

RF14 → Notificações de Atraso

- **Descrição:** Aquando do *login* na aplicação, o sistema deve notificar os bibliotecários sobre a listagem de leitores com entregas de livros em atraso.

RF15 → Gestão de Utilizadores

- **Descrição:** O sistema deve permitir que os administradores visualizem, criem, atualizem, ativem ou bloqueiem outros utilizadores.

RF16 → Bloqueio de Utilizadores

- **Descrição:** O sistema deve permitir que os administradores bloqueiem leitores ou bibliotecários e que o utilizador bloqueado seja informado do motivo do seu bloqueio ao tentar aceder ao sistema.

RF17 → Registo de Leitores

- **Descrição:** O sistema deve permitir o registo de leitores através do portal, enviando um e-mail de confirmação para validar o registo.

RF18 → Registo de Bibliotecários

- **Descrição:** O sistema deve permitir o registo de bibliotecários, sujeito a aprovação de um administrador, com o registo da aprovação.

RF19 → Registo de Administradores

- **Descrição:** O sistema deve permitir que apenas os administradores possam criar/registar outros administradores na aplicação.

1.3.2 Requisitos Não Funcionais (RNF)

1.3.2.1 Utilidade

RNF1 → Utilidade

- **Descrição:** A interface da aplicação deve ser intuitiva, fácil de usar e acessível para utilizadores com diferentes níveis de conhecimento técnico.

1.3.2.2 Desempenho e Escalabilidade

RNF2 → Desempenho

- **Descrição:** O sistema deve ser capaz de responder às solicitações de consulta e pesquisa de livros em tempo real.

RNF3 → Escalabilidade

- **Descrição:** O sistema deve suportar um aumento no número de utilizadores e requisições sem comprometer o desempenho.

1.3.2.3 Segurança

RNF4 → Segurança de Dados

- **Descrição:** O sistema deve garantir a proteção dos dados pessoais dos utilizadores através de encriptação e padrões de segurança adequados.

1.3.2.4 Disponibilidade e Compatibilidade

RNF5 → Disponibilidade

- **Descrição:** O sistema deve estar disponível 24/7, com o menor tempo de inatividade possível.

RNF6 → Compatibilidade

- **Descrição:** A aplicação deve ser compatível com os principais navegadores web e dispositivos.

1.3.2.5 Manutenibilidade

RNF7 → Manutenibilidade

- **Descrição:** O sistema deve ser de fácil manutenção, permitindo atualizações e correções de *bugs*, sem alterações significativas no código.

1.3.2.6 Conformidade

RNF8 → Conformidade

- **Descrição:** O sistema deve estar em conformidade com as leis e regulamentos sobre proteção de dados, como o RGPD.

1.4 Especificação do Modelo Funcional

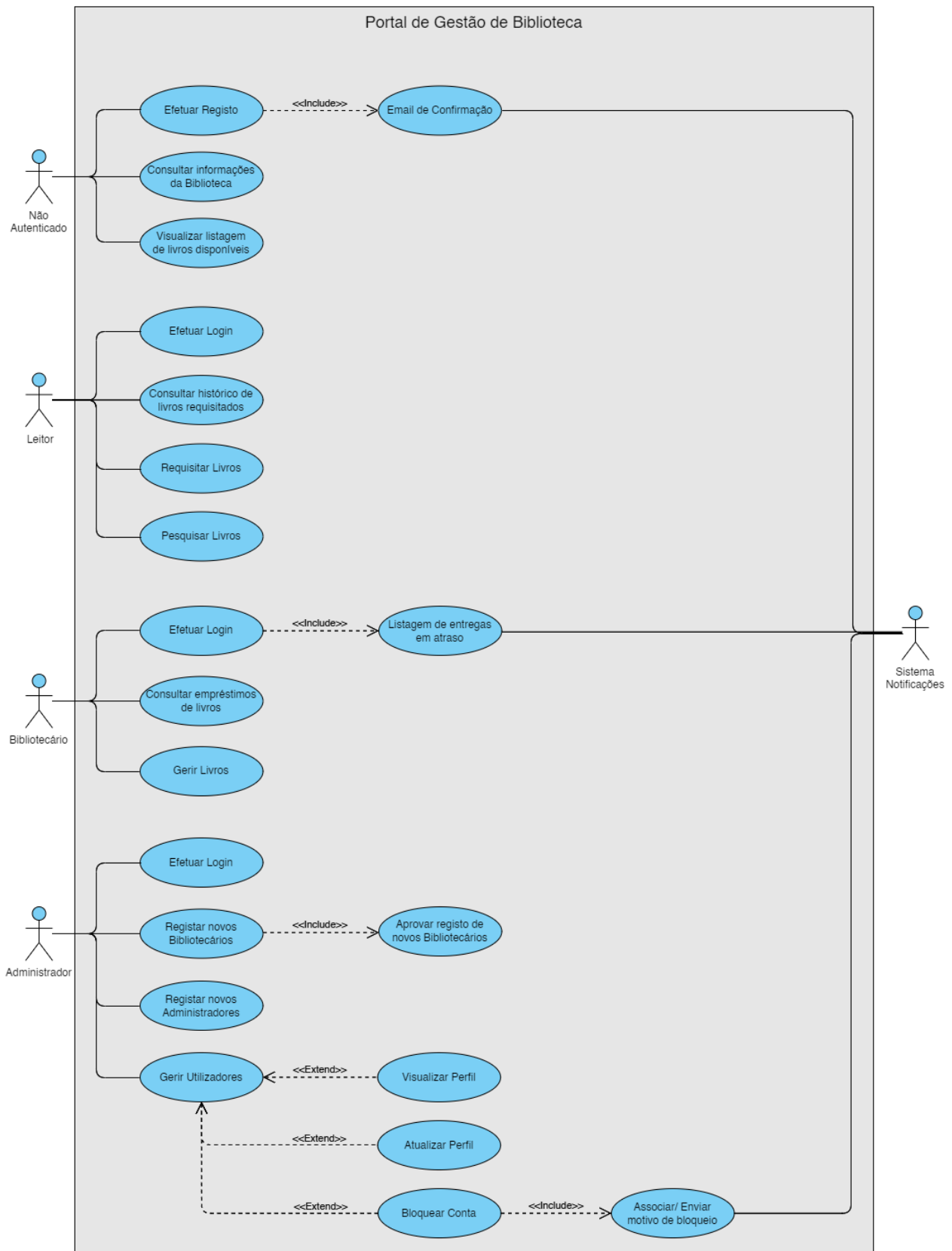


Figura 2- Diagrama de Casos-de-Uso

2- FASE 2

2.1 Mapeamento entre o Modelo Conceptual de Dados e o Modelo Relacional

Autenticado (Username, Password, Contacto, Contacto2, Contacto3, Nome)

- Relacionamentos:
 - Bloquear: Relaciona-se com Administradores
 - Cardinalidade: (0,n)

Bibliotecario (Username, Bib)

- Relacionamentos:
 - ValidarRegisto: Relaciona-se com Administrador
 - Cardinalidade: (1,1)

Leitores (Username, Lei, Morada, Email)

- Relacionamentos:
 - Requisicao: Relaciona-se com Livro
 - Cardinalidade: (0,n)

Autor (ID_Autor, Name_Autor)

Livro (ISBN, Preco, Titulo, N_Exemplares, Genero, Username, BibAdd, ID_Autor, ID_Biblioteca)

- Relacionamentos:
 - Requisicao: Relaciona-se com Leitores
 - Cardinalidade: (0,n)

Biblioteca (ID_Biblioteca, HorarioI, HorarioF, Localizacao, Nome, Email, Telefone)

- Relacionamentos:
 - Consulta_Biblio: Relaciona-se com NaoAutenticado
 - Cardinalidade: (0,n)

NaoAutenticado

- Relacionamentos:
 - Consulta_Biblio: Relaciona-se com Biblioteca
 - Cardinalidade: (1,1)

Requisicao (ID_Requisicao, ~~Username_Bib~~, ~~Username_Lei~~, ~~ISBN~~, Data_Requisicao, Data_Devolucao, Estado)

- Relacionamentos:
 - Requisicao: Relaciona-se com HistoricoRequ
 - Cardinalidade: (0,n)
 - Requisicao: Relaciona-se com EntregasAtraso
 - Cardinalidade: (0,n)
 - Requisicao: Relaciona-se com Bibliotecário
 - Cardinalidade: (0,n)

HistoricoRequ (ID_Historico, ~~ID_Requisicao~~)

EntregasAtraso (ID_Entregas, ~~ID_Requisicao~~)

Bloquear (ID_Bloqueio, ~~ID_MotivoBloq~~, ~~Username_Admin~~, Username, Data_Bloqueio, Bloqueado)

- Relacionamentos:
 - Bloquear: Relaciona-se com MotivoBloq
 - Cardinalidade: (1,1)
 - Bloquear: Relaciona-se com Administradores
 - Cardinalidade: (0,n)
 - Bloquear: Relaciona-se com Autenticado
 - Cardinalidade: (0,n)

MotivoBloq (ID_MotivoBloq, Descricao)

Administradores (Username_Admin, Nome, Email, Password)

- Relacionamentos:
 - Bloquear: Relaciona-se com Autenticado
 - Cardinalidade: (0,n)
 - Cria_Admin: Relaciona-se com Administrador
 - Cardinalidade: (0,n) e (1,1)
 - ValidarRegisto: Relaciona-se com Bibliotecario
 - Cardinalidade: (1,n)

2.2 Implementação do Modelo físico da base de dados recorrendo à linguagem SQL

```

USE master
GO

CREATE DATABASE LAB2
GO

USE LAB2
GO

CREATE TABLE Autenticado(
    Username          VARCHAR(50)      NOT NULL,
    Password          VARCHAR(10)     NOT NULL,
    Contacto          BIGINT           NOT NULL,
    Contacto2         BIGINT,
    Contacto3         BIGINT,
    Nome              VARCHAR(50)      NOT NULL,

    CHECK (Contacto LIKE '9[1236][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CHECK (Contacto2 LIKE '9[1236][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CHECK (Contacto3 LIKE '9[1236][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (Username)
)

CREATE TABLE Bibliotecario(
    Username_Bib      VARCHAR(50)      NOT NULL,

    PRIMARY KEY (Username_Bib),
    FOREIGN KEY (Username_Bib) REFERENCES Autenticado(Username)
)

CREATE TABLE Leitores(
    Username_Lei      VARCHAR(50)      NOT NULL,
    Morada            VARCHAR(50)      NOT NULL,
    Email             VARCHAR(50)      NOT NULL,

    PRIMARY KEY (Username_Lei),
    FOREIGN KEY (Username_Lei) REFERENCES Autenticado(Username)
)

CREATE TABLE Administradores(
    Username_Admin    VARCHAR(50)      NOT NULL,
    Nome              VARCHAR(50)      NOT NULL,
    Email             VARCHAR(50)      NOT NULL,
    Password          VARCHAR(10)     NOT NULL,

    PRIMARY KEY (Username_Admin)
)

CREATE TABLE MotivoBloq(
    ID_MotivoBloq     INT              NOT NULL,
    Descricao         VARCHAR(100)     NOT NULL,

    CHECK (ID_MotivoBloq LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_MotivoBloq)
)

```



```

CREATE TABLE Bloquear(
    ID_Bloqueio          INT          NOT NULL,
    ID_MotivoBloq        INT          NOT NULL,
    Username_Admin       VARCHAR(50)  NOT NULL,
    Username             VARCHAR(50)  NOT NULL,
    Data_Bloqueio        SMALLDATETIME NOT NULL DEFAULT GETDATE(),
    Bloqueado            BIT          NOT NULL DEFAULT 0, -- nao bloqueado

    CHECK (ID_Bloqueio LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Bloqueio),
    FOREIGN KEY (Username_Admin) REFERENCES Administradores(Username_Admin),
    FOREIGN KEY (Username) REFERENCES Autenticado(Username),
    FOREIGN KEY (ID_MotivoBloq) REFERENCES MotivoBloq(ID_MotivoBloq)
)

CREATE TABLE ValidarRegisto(
    ID_Valid            INT          NOT NULL,
    Username_Admin     VARCHAR(50)  NOT NULL,
    Username_Bib       VARCHAR(50)  NOT NULL,
    Data_Valid         SMALLDATETIME NOT NULL,
    Validado           BIT          NOT NULL DEFAULT 0, -- não validado

    CHECK (ID_Valid LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Valid),
    FOREIGN KEY (Username_Admin) REFERENCES Administradores(Username_Admin),
    FOREIGN KEY (Username_Bib) REFERENCES Bibliotecario(Username_Bib)
)

CREATE TABLE Autor(
    ID_Autor           INT          NOT NULL,
    Name_Autor        VARCHAR(50)  NOT NULL,

    CHECK (ID_Autor LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Autor)
)

CREATE TABLE Biblioteca(
    ID_Biblioteca      INT          NOT NULL,
    HorarioI          TIME          NOT NULL,
    HorarioF          TIME          NOT NULL,
    Localizacao       VARCHAR(500)  NOT NULL,
    Nome              VARCHAR(50)   NOT NULL,
    Email             VARCHAR(50)   NOT NULL,
    Telefone          BIGINT        NOT NULL,

    CHECK (ID_Biblioteca LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    CHECK (Telefone LIKE '2[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CHECK (HorarioF > HorarioI),
    PRIMARY KEY (ID_Biblioteca)
)

```

```

CREATE TABLE Livro(
    ISBN                BIGINT                NOT NULL,
    Preço               MONEY                NOT NULL,
    Titulo              VARCHAR(50)          NOT NULL,
    N_Exemplares        INT                  NOT NULL,
    Genero              VARCHAR(50)          NOT NULL,
    Username_BibAdd     VARCHAR(50)          NOT NULL,
    ID_Autor            INT                  NOT NULL,
    ID_Biblioteca       INT                  NOT NULL,

    CHECK (ISBN LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ISBN),
    FOREIGN KEY (Username_BibAdd) REFERENCES Bibliotecario(Username_Bib),
    FOREIGN KEY (ID_Autor) REFERENCES Autor(ID_Autor),
    FOREIGN KEY (ID_Biblioteca) REFERENCES Biblioteca(ID_Biblioteca)
)

CREATE TABLE Requisicao(
    ID_Requisicao        INT                  NOT NULL,
    Username_Bib        VARCHAR(50)          NOT NULL,
    Username_Lei        VARCHAR(50)          NOT NULL,
    ISBN               BIGINT                NOT NULL,
    Data_Requisicao      SMALLDATETIME        NOT NULL    DEFAULT GETDATE(),
    Data_Devolucao      SMALLDATETIME,
    Estado              BIT                  NOT NULL    DEFAULT 0, -- não requisitado

    CHECK (ID_Requisicao LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    CHECK (Data_Devolucao > Data_Requisicao),
    PRIMARY KEY (ID_Requisicao),
    FOREIGN KEY (Username_Bib) REFERENCES Bibliotecario(Username_Bib),
    FOREIGN KEY (Username_Lei) REFERENCES Leitores(Username_Lei),
    FOREIGN KEY (ISBN) REFERENCES Livro(ISBN)
)

CREATE TABLE Cria_Admin(
    Username_Admin      VARCHAR(50)          NOT NULL,
    Username_NovoAdmin  VARCHAR(50)          NOT NULL,

    FOREIGN KEY (Username_Admin) REFERENCES Administradores(Username_Admin),
    FOREIGN KEY (Username_NovoAdmin) REFERENCES Administradores (Username_Admin)
)

CREATE TABLE EntregasAtraso(
    ID_Entregas         INT                  NOT NULL,
    ID_Requisicao        INT                  NOT NULL,

    CHECK (ID_Entregas LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Entregas),
    FOREIGN KEY (ID_Requisicao) REFERENCES Requisicao(ID_Requisicao)
)

CREATE TABLE HistoricoRequ(
    ID_Historico        INT                  NOT NULL,
    ID_Requisicao        INT                  NOT NULL,

    CHECK (ID_Historico LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Historico),
    FOREIGN KEY (ID_Requisicao) REFERENCES Requisicao(ID_Requisicao)
)

```

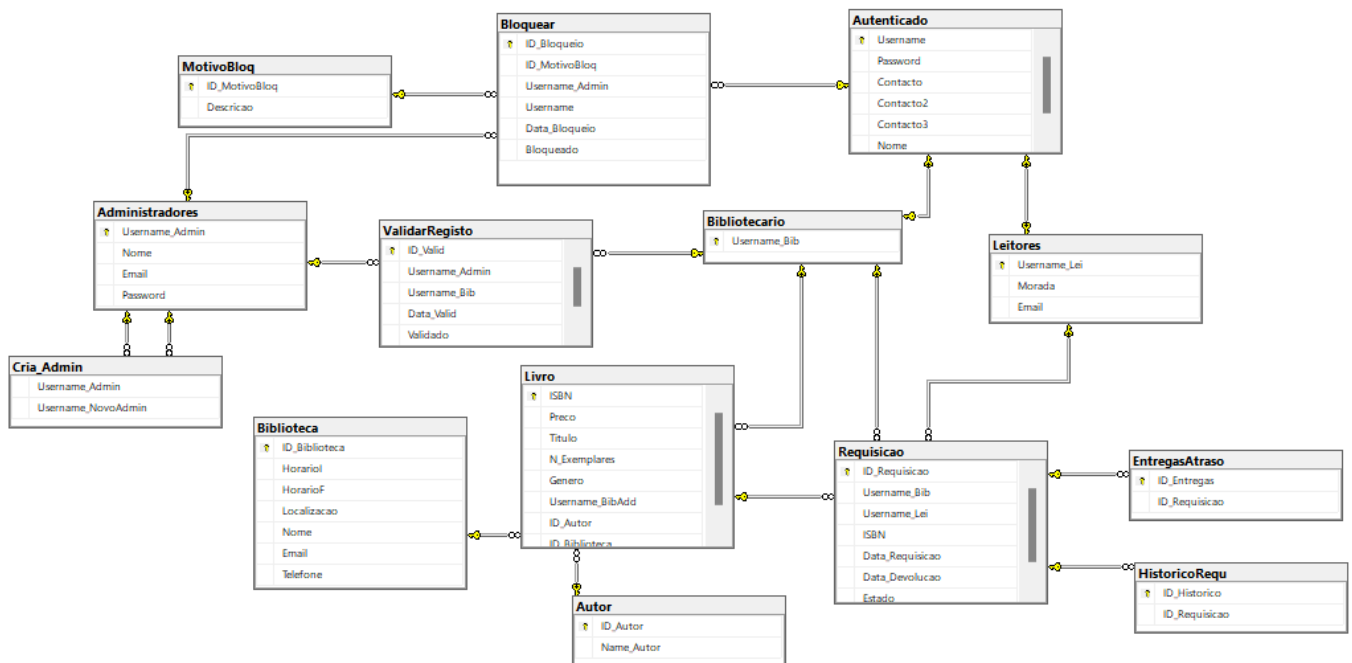


Figura 3- Diagrama de Modelo Relacional

2.3 Especificação das interfaces com o utilizador para o *Backoffice* e *Frontoffice* recorrendo a *Mockups*

2.3.1 *Frontoffice*

A experiência agradável do utilizador final deve ser uma prioridade. Assim, procuramos criar uma interface intuitiva, garantindo uma navegação facilitada no sistema.

2.3.1.1 Página de Apresentação e Menu Inicial



Figura 4- Página de Apresentação

Quando se executa a aplicação web, é exibida uma página de Apresentação (Figura 4). Ao clicar no botão 'Start', o utilizador é redirecionado para uma página que contém o Menu Inicial da aplicação (Figura 5).

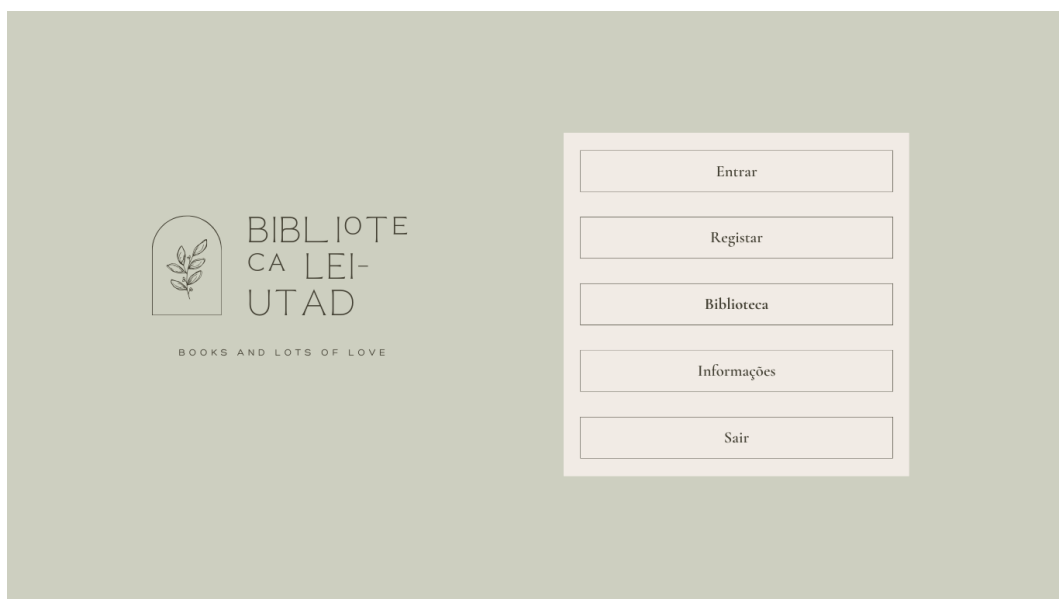


Figura 5- Menu Inicial

Nesta página, o utilizador pode seleccionar uma das cinco opções apresentadas. Os dois primeiros botões, 'Entrar' e 'Registar', permitem realizar o *Login* e o Registo na aplicação, respetivamente. Caso o utilizador não se deseje autenticar na aplicação, é possível aceder à lista de livros disponíveis, através do botão 'Biblioteca', ou consultar as informações gerais da biblioteca, através do botão 'Informações'. Por fim, o botão 'Sair' encerra a aplicação.

2.3.1.2 Consultar Informações da Biblioteca

Ao seleccionar o botão 'Informações' na página da Figura 5, o utilizador é redireccionado para uma página onde pode consultar as informações gerais sobre a Biblioteca LEI-UTAD (Figura 6).



Figura 6- Página de Informações

2.3.1.3 Registo de Utilizadores na Aplicação

Ao seleccionar o botão 'Registar' na página da Figura 5, o utilizador é redireccionado para a página de registo (Figura 7).

Nesta página, o utilizador deverá seleccionar que tipo de registo deseja fazer (Leitor ou Bibliotecário) e, dependendo da opção escolhida, alguns dos parâmetros de informações são ocultados.

Quando o registo é realizado, o utilizador irá receber uma notificação da tela. Se for um Leitor, a mensagem de confirmação será a exibida na Figura 8. No caso de ser Bibliotecário, a mensagem apresentada é a que se encontra na Figura 9.

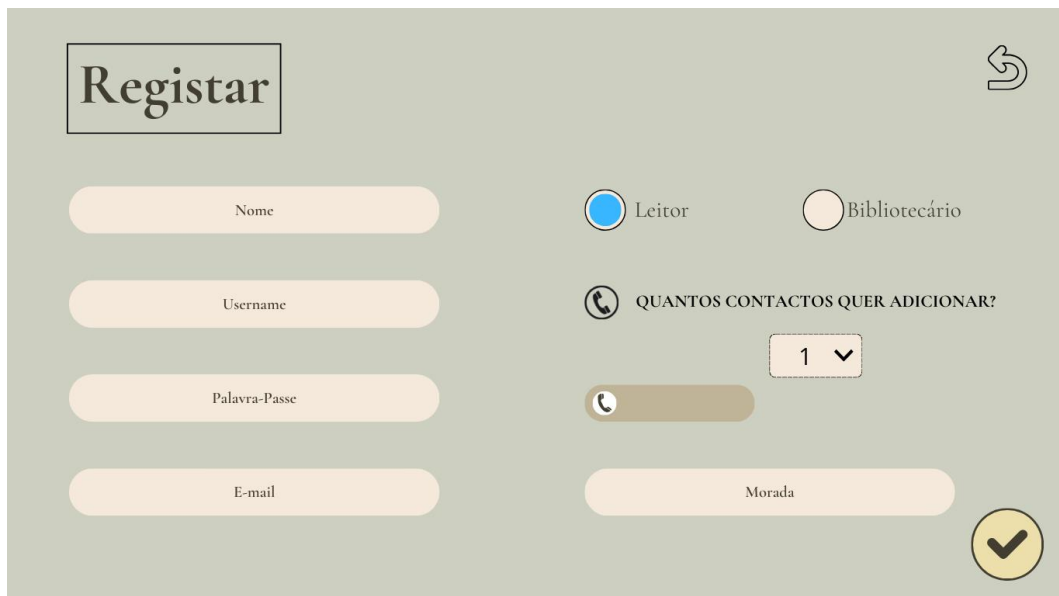


Figura 7- Página de Registo

The image shows a registration form titled "Registar" in a box at the top left. In the top right corner, there is a circular arrow icon. The form contains several input fields: "Nome", "User", "Palavra", "E-mail", and "Morada". Between the "Nome" and "User" fields, there are two radio buttons: "Leitor" (which is selected with a blue dot) and "Bibliotecário". To the right of the "User" field, there is a label "TOS QUER ADICIONAR?" and a dropdown menu showing the number "1". A yellow message box with a dashed border is overlaid on the form, containing the text: "Será enviado um email de confirmação para validar o seu registo." At the bottom right of the form, there is a circular button with a checkmark icon.

Figura 8- Mensagem de confirmação para o Leitor

The image shows the same registration form as in Figure 8, but with the "Bibliotecário" radio button selected. The yellow message box now contains the text: "É necessária a aprovação por parte de um administrador para continuar... Por favor, aguarde enquanto a sua solicitação é analisada e autorizada." All other elements of the form, including the "Registar" title, input fields, and icons, are identical to the previous figure.

Figura 9- Mensagem para o Bibliotecário

2.3.1.4 Login Utilizadores Autenticados

Ao seleccionar o botão 'Entrar' na página da Figura 5, o utilizador é redireccionado para a página de Login (Figura 10).

Esta página, permite ao utilizador entrar na aplicação, após o registo devidamente concluído. Para fazer o *login* deverá apenas preencher os campos com o seu *Username* e Palavra-Passe, e clicar no botão 'Entrar' apresentado abaixo.

Caso o utilizador se encontre bloqueado, não será possível realizar o *Login*. Nesta eventualidade, é exibida na tela uma notificação com o motivo do seu bloqueio, tal como é apresentado na Figura 11.

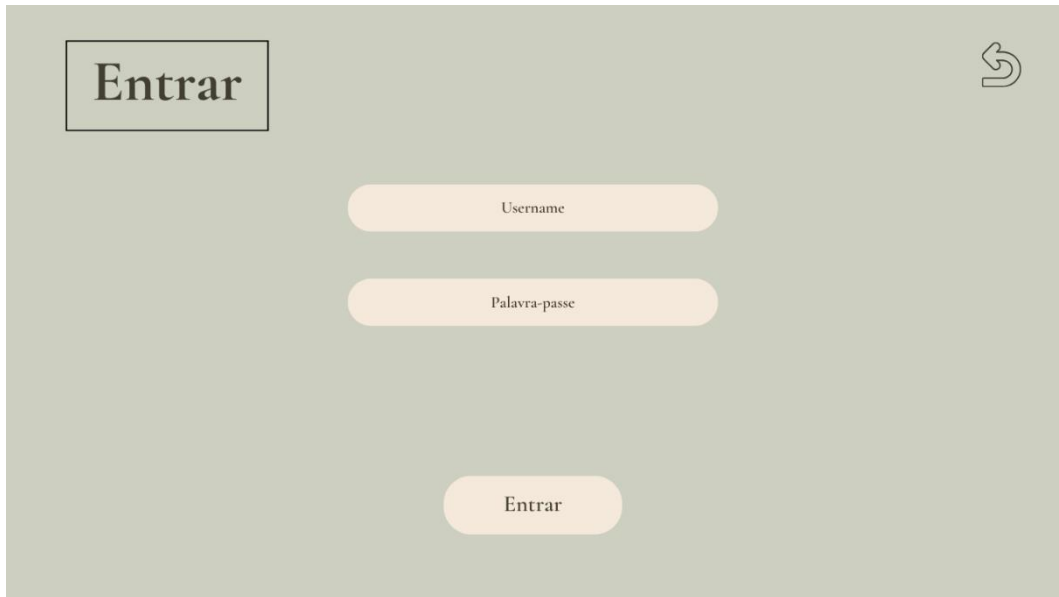
A screenshot of a login page with a light green background. In the top left, there is a box containing the word 'Entrar'. In the top right, there is a circular arrow icon. The center of the page features two stacked, rounded rectangular input fields. The first field is labeled 'Username' and the second is labeled 'Palavra-passe'. Below these fields is a rounded rectangular button labeled 'Entrar'.

Figura 10- Página de *Login*

A screenshot of the same login page as in Figure 10, but with a dark grey background. A yellow rectangular message box with a dashed border is centered on the screen. The message box contains the text 'A sua conta foi bloqueada por um administrador.' and a label 'Motivo:' followed by a small, empty rectangular input field. The 'Entrar' button at the bottom is now greyed out. The 'Entrar' label in the top left and the circular arrow icon in the top right remain visible.

Figura 11- Mensagem de bloqueio para Utilizadores

2.3.1.5 Página Principal do Leitor

Após efetuar o *Login* com sucesso, os Leitores têm acesso à página principal da aplicação (Figura 12), onde podem explorar os livros disponíveis na Biblioteca, consultar informações sobre a mesma (Figura 6) ou terminar a sua sessão.

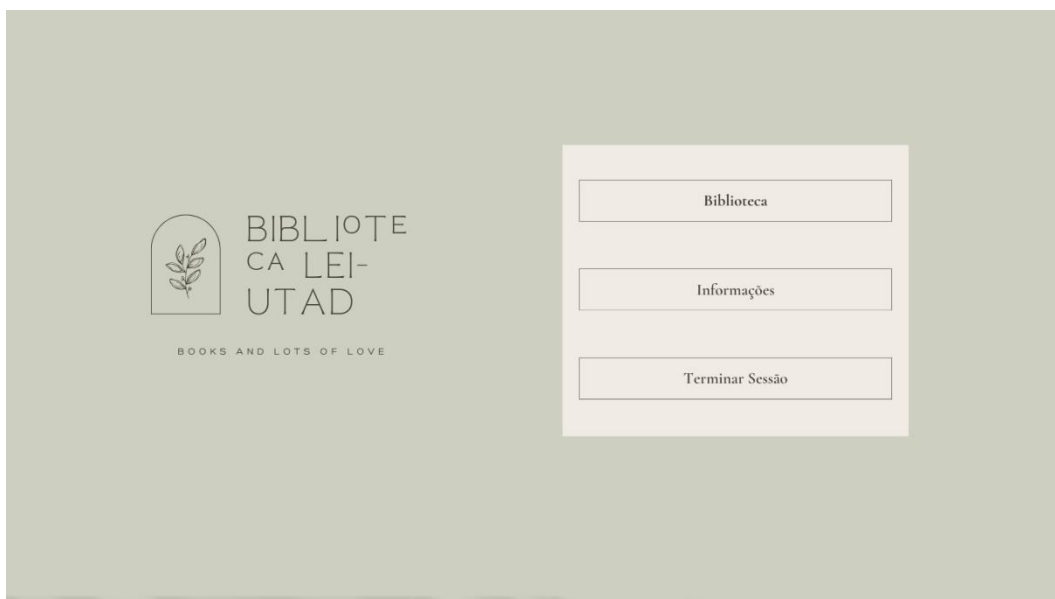


Figura 12- Página Principal do Leitor

2.3.1.6 Consultar Livros

Ao selecionar o botão 'Biblioteca' na página da Figura 12, o Leitor é redirecionado para a página da Biblioteca (Figura 13).

Nesta página, o Leitor poderá explorar/consultar informações sobre os livros disponíveis na Biblioteca, bem como efetuar pedidos de requisição dos mesmos.

Para além disso, é possível visualizar detalhes como o preço dos livros e o número de exemplares disponíveis para requisição, pesquisar livros específicos e consultar o seu histórico de requisições.



The screenshot shows a web application titled 'Biblioteca'. It features a search bar with the placeholder 'Procurar' and a magnifying glass icon. To the right of the search bar are buttons for 'Requisitar', a clock icon, and a refresh icon. Below the search bar is a table with the following columns: 'Título', 'Autor', 'Gênero', 'Preço', 'Nº DE EXEMPLARES', and 'ISBN'. The table contains four rows of data, each with a checkbox in the first column.

	Título	Autor	Gênero	Preço	Nº DE EXEMPLARES	ISBN
<input type="checkbox"/>	Isto Acaba Aqui	Colleen Hoover	Romance	19,99€	17	978-0-7334-2609-4
<input type="checkbox"/>	Pai Rico, Pai Pobre	Robert Kiyosaki, Sharon L. Lechter	Finanças pessoais	16,99€	20	978-0-7334-2609-5
<input type="checkbox"/>	O segredo do nº9	Paulo Pires	Thriller	14,99€	0	978-0-7334-2609-6
<input type="checkbox"/>	Steve Jobs	Walter Isaacson	Biografia	21,00€	2	978-0-7334-2609-7
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						

Figura 13- Página da Biblioteca para Leitores

2.3.1.7 Pesquisa de Livros

Ao seleccionar a barra de pesquisa, é exibido um menu com a filtragem disponível para a pesquisa de livros na Biblioteca (Figura 14). Nesta secção, o Leitor pode pesquisar o livro que deseja visualizar, inserindo apenas os campos que achar relevantes, através da seleção dos mesmos.



The screenshot shows the same 'Biblioteca' web application interface as Figure 13, but with a search menu overlay. The menu is titled 'Biblioteca' and contains the following sections:

- Título**: A text input field with the placeholder 'Insira o Título do Livro'.
- Gênero**: A text input field with the placeholder 'Insira o Gênero do Livro'.
- Autor(es)**: A section with a dropdown menu labeled 'Quantos Autores tem o Livro?' showing the value '3'. Below this are three text input fields, each with the placeholder 'Insira o Autor do Livro'.
- Pesquisar**: A yellow button at the bottom of the menu.

The background table from Figure 13 is partially visible behind the menu.

Figura 14- Menu de pesquisa de livros

2.3.1.8 Requisição de Livros

Após encontrar o livro desejado, o Leitor pode requisitá-lo. Para isso, deve seleccionar o livro, tal como é apresentado na Figura 15, e seleccionar o botão 'Requisitar'.



The screenshot shows a web application interface for a library. At the top left is a logo with the word "Biblioteca". To its right is a search bar with the placeholder text "Procurar" and a magnifying glass icon. Further right is a yellow button labeled "Requisitar", followed by a clock icon and a circular arrow icon. Below these elements is a table with the following columns: "Título", "Autor", "Género", "Preço", "Nº DE EXEMPLARES", and "ISBN". The table contains five rows of data, with the first row having a small square icon in the first column. The rest of the table is empty.

	Título	Autor	Género	Preço	Nº DE EXEMPLARES	ISBN
	Isto Acaba Aqui	Colleen Hoover	Romance	19,99€	17	978-0-7334-2609-4
	Pai Rico, Pai Pobre	Robert Kiyosaki, Sharon L. Lechter	Finanças pessoais	16,99€	20	978-0-7334-2609-5
	O segredo do nº9	Paulo Pires	Thriller	14,99€	0	978-0-7334-2609-6
	Steve Jobs	Walter Isaacson	Biografia	21,00€	2	978-0-7334-2609-7
						
						
						
						
						

Figura 15- Seleção do livro pretendido

De seguida, é exibida uma mensagem de confirmação do pedido de requisição (Figura 16). Caso o Leitor, selecione a opção 'Sim', a mensagem da Figura 17 é apresentada na tela, caso contrário apresenta a mensagem da Figura 18.



Figura 16- Confirmação do pedido de requisição

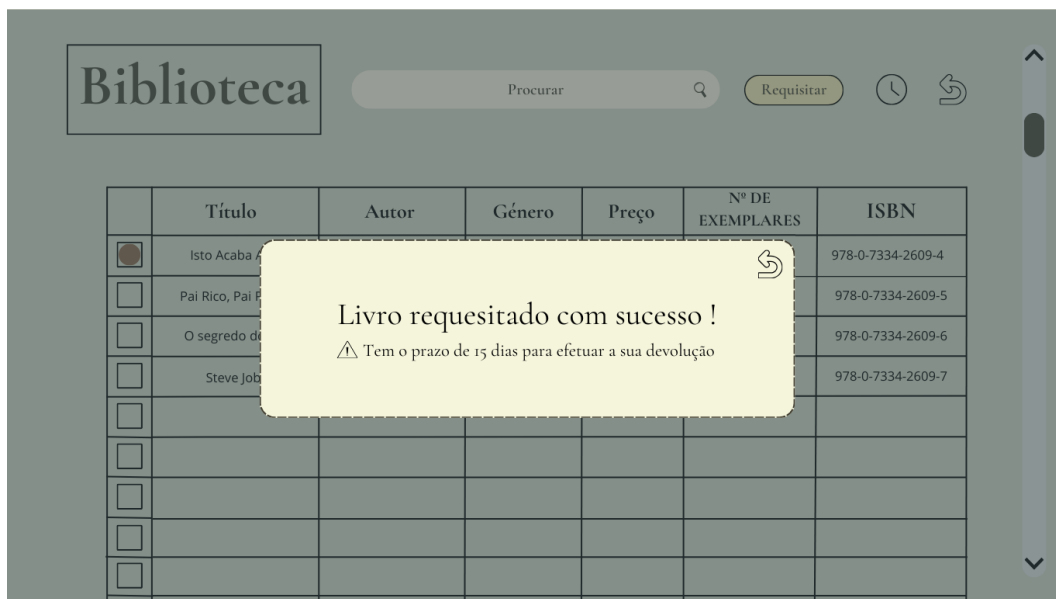


Figura 18- Mensagem de confirmação da requisição

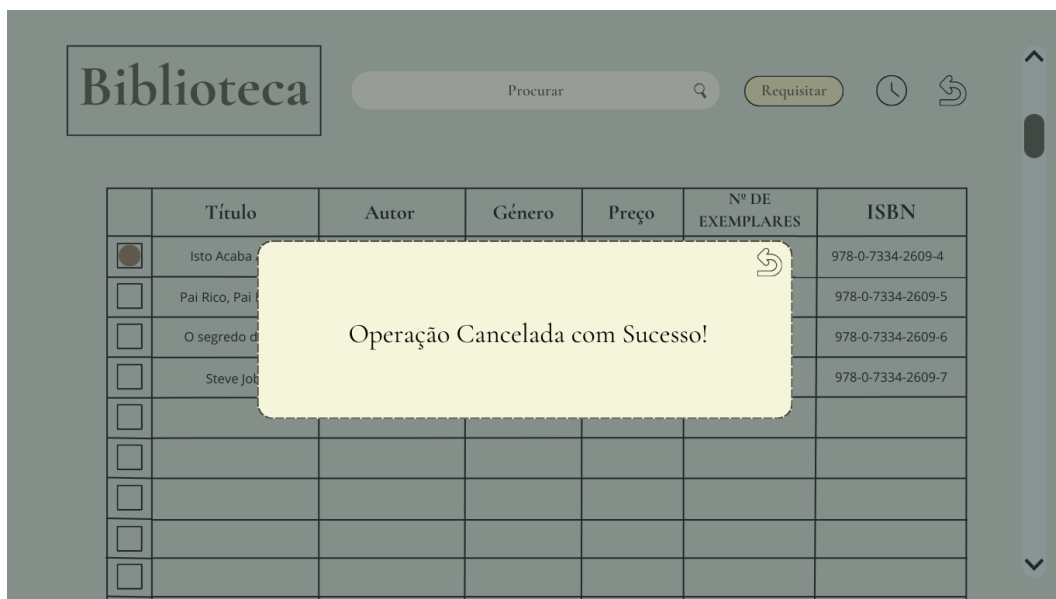


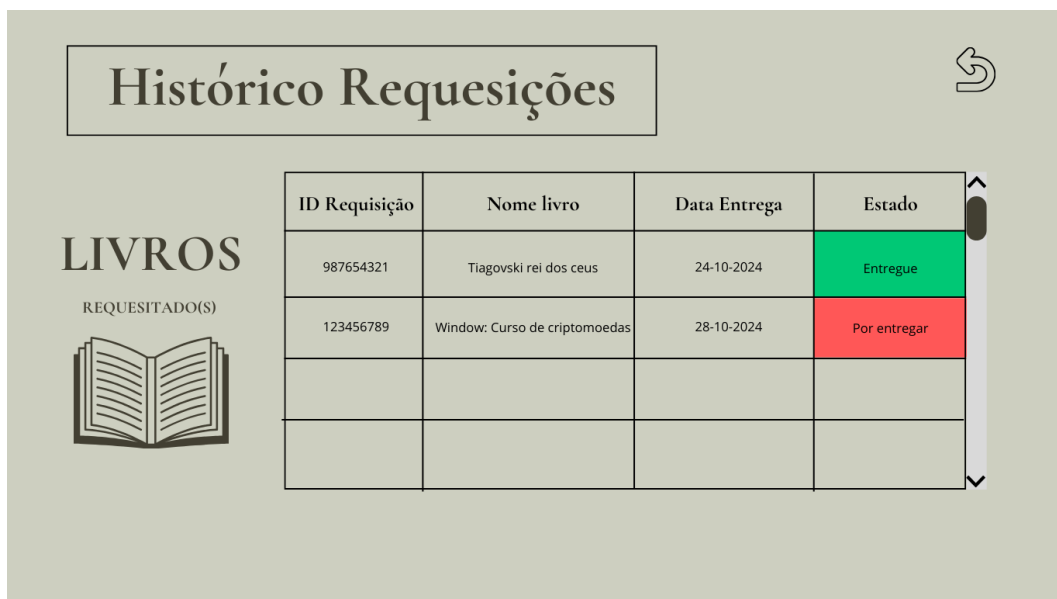
Figura 17- Mensagem de cancelamento da requisição

Caso o Leitor pretenda realizar a requisição de mais de um livro, deve repetir este processo quantas vezes forem desejadas.

2.3.1.9 Consultar Histórico de Requisições

Ao selecionar o segundo botão localizado no canto superior direito da Figura 5, o Leitor é redirecionado para a página da Figura 19.

Nesta secção, é possível consultar todo o histórico de livros requisitados até ao momento pelo Leitor, bem como visualizar o estado das requisições e a data de entrega dos livros. Estas informações são importantes para que o Leitor acompanhe as suas requisições e datas de entregas futuras.



ID Requisição	Nome livro	Data Entrega	Estado
987654321	Tiagovski rei dos ceus	24-10-2024	Entregue
123456789	Window: Curso de criptomoedas	28-10-2024	Por entregar

Figura 19- Página de consulta do histórico

2.3.2 *Backoffice* (Bibliotecário)

2.3.2.1 Página Principal do Bibliotecário

Após efetuar o *Login* com sucesso, os Bibliotecários têm acesso à página principal da aplicação (Figura 20), onde podem explorar a Biblioteca, consultar informações sobre a mesma (Figura 6) ou terminar a sua sessão.

O Bibliotecário tem acesso a funcionalidades que o Leitor comum não tem. Para além de visualizar a listagem de livros e as suas informações, é também responsável por gerir a Biblioteca e garantir o bom funcionamento da mesma.

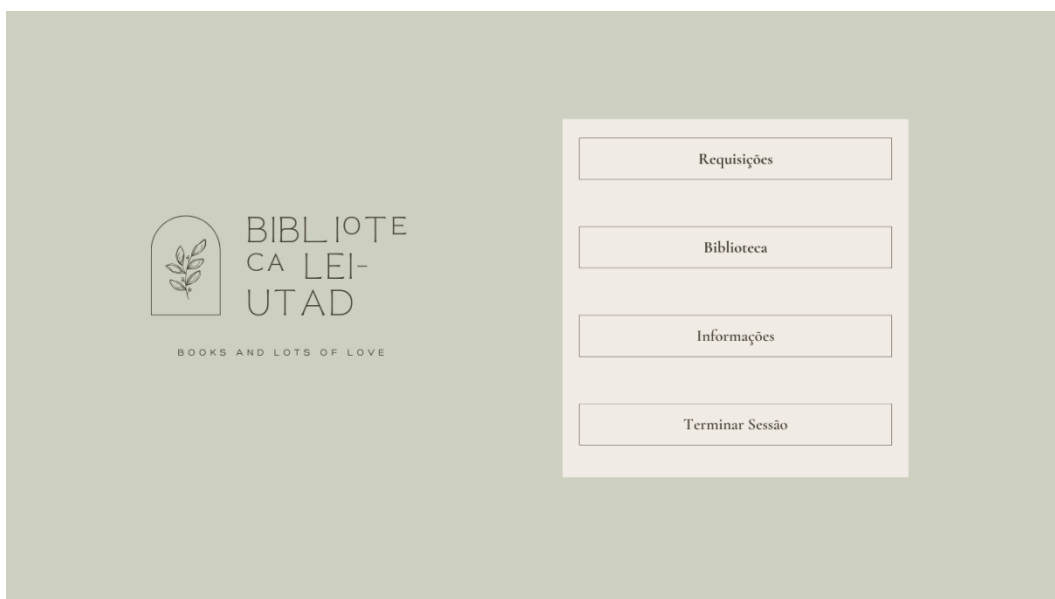


Figura 20- Página Principal do Bibliotecário

2.3.2.2 Gestão de Requisições

Ao selecionar o botão 'Requisições' na página da Figura 20, o Bibliotecário é redirecionado para uma página de gestão das requisições de livros (Figura 21).

Nesta página, o Bibliotecário tem acesso a informações sobre os empréstimos de livros. Cada empréstimo tem associado um ID, que permite ao Bibliotecário saber do que se trata essa requisição. O livro requisitado está identificado pelo seu ISBN.

	ISBN	ID Requisição	Data Requisição	Data Entrega	Estado
<input type="checkbox"/>	978-0-7334-2609-4	112	16-10-2024	24-10-2024	Entregue
<input type="checkbox"/>	978-0-7334-2609-9	123	17-10-2024	30-10-2024	Por entregar
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Figura 21- Página de Requisições

2.3.2.3 Consultar Empréstimos de Livros

Ao selecionar o botão destacado a vermelho na coluna 'Data Requisição', o Bibliotecário consegue visualizar o nome do Leitor que efetuou determinada requisição de um livro, tal como podemos observar na Figura 22.

	ISBN	ID Requisição	Data Requisição	Data Entrega	Estado
<input type="checkbox"/>	978-0-7334-2609-4	112	16-10-2024	24-10-2024	Entregue
<input type="checkbox"/>	978-0-7334-2609-9	123		30-10-2024	Por entregar
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Figura 22- Consulta da Requisição de um Leitor

2.3.2.4 Modificar Estado de Requisição

Para além de monitorizar os empréstimos de livros, o Bibliotecário pode modificá-los seleccionando a requisição pretendida, tal como podemos observar na Figura 23.

Deste modo, quando um Leitor entrega um livro que requisitou, o Bibliotecário pode alterar o estado de requisição do mesmo para 'Entregue'.

	ISBN	ID Requisição	Data Requisição	Data Entrega	Estado
<input type="checkbox"/>	978-0-7334-2609-4	112	16-10-2024	24-10-2024	Entregue
<input checked="" type="checkbox"/>	978-0-7334-2609-9	123	17-10-2024	30-10-2024	Por entregar
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Figura 23- Alteração do estado de requisição de um livro

2.3.2.5 Consultar Leitores com Entregas em Atraso

Ao seleccionar o segundo botão localizado no canto superior direito da Figura 21, o Bibliotecário é redireccionado para a página da Figura 24.

Nesta secção, é possível consultar a listagem de Leitores com entregas de livros em atraso. Esta informação é importante para que o Bibliotecário acompanhe os empréstimos de livros e os seus prazos de entrega, garantindo a circulação organizada de livros e facilitando o cumprimento das normas da Biblioteca.

Entregas em atraso			
ID Requisição	Leitor	Livro	Data Prevista de Entrega
112	nimadururu.tkd	Tiagovski rei dos céus	15-10-2024
123	SennaLenda1	Window: Curso de criptomoedas	14-10-2024
162	PhelpsOuro28	Guinness World Records 2025	19-10-2024
144	LewisRitmo44	Furacão Hamilton	17-10-2024

Figura 24- Listagem de Leitores com entregas em atraso

2.3.2.6 Gestão de Biblioteca

Ao selecionar o botão 'Biblioteca' na página da Figura 20, o Bibliotecário é redirecionado para uma página de gestão da Biblioteca (Figura 25).

Nesta página, o Bibliotecário tem acesso a todos os livros existentes na biblioteca, podendo editar informações dos mesmos, adicionar novos livros à Biblioteca ou remover livros já existentes na mesma.





Biblioteca						
						
	Título	Autor	Género	Preço	Nº DE EXEMPLARES	ISBN
<input type="checkbox"/>	Isto Acaba Aqui	Colleen Hoover	Romance	19,99€	17	978-0-7334-2609-4
<input type="checkbox"/>	Pai Rico, Pai Pobre	Robert Kiyosaki, Sharon L. Lechter	Finanças pessoais	16,99€	20	978-0-7334-2609-5
<input type="checkbox"/>	O segredo do nº9	Paulo Pires	Thriller	14,99€	0	978-0-7334-2609-6
<input type="checkbox"/>	Steve Jobs	Walter Isaacson	Biografia	21,00€	2	978-0-7334-2609-7
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						

Figura 25- Página da Biblioteca

2.3.2.7 Adicionar Livro

Caso um Bibliotecário pretenda adicionar um novo livro à Biblioteca, deve seleccionar o quarto botão localizado no canto superior direito da página da Figura 25.

Imediatamente, é redireccionado para a página de inserção de livros (Figura 26). Nesta secção, deve preencher todos os campos relativos ao novo livro. Se o número de autores for diferente de um, o Bibliotecário deve indicar esse parâmetro no campo destinado para o efeito e, em seguida, irão ser adicionados mais campos para inserir os restantes autores, tal como ilustra a figura.

Após a conclusão desta etapa, o novo livro passa a fazer parte da lista da Biblioteca.

A imagem mostra a interface de uma página web para adicionar um novo livro. No topo, há um título "Adicionar Livro" dentro de um retângulo. À direita do título, há um ícone de seta curva para cima. Abaixo do título, há dois colunas de campos de entrada. A primeira coluna contém quatro campos: "(Insira o título do livro)", "(Insira o ISBN)", "(Insira o género do livro)" e "(Insira o preço do livro)". A segunda coluna começa com o texto "Quantos autores tem ?" seguido de um menu suspenso com o valor "3" e um ícone de seta para baixo. Abaixo disso, há três campos idênticos: "(Insira o autor do livro)". No canto inferior direito da interface, há um botão circular amarelo com um ícone de seta verde para cima.

Figura 26- Página de adição de um livro

2.3.2.8 Editar Livro

Caso um Bibliotecário necessite de alterar pormenores em qualquer um dos livros disponíveis na Biblioteca, deve seleccionar o livro pretendido (Figura 27) e, em seguida, o terceiro botão localizado no canto superior direito da página da Figura 25.

Desta forma, é redireccionado para a página de edição de livros (Figura 28). Nesta secção, é-lhe concebida permissão para alterar certos parâmetros relativos ao livro, tais como o seu preço, uma vez que os restantes pormenores devem ser definitivos.

Biblioteca

	Título	Autor	Género	Preço	Nº DE EXEMPLARES	ISBN
	Isto Acaba Aqui	Colleen Hoover	Romance	19,99€	17	978-0-7334-2609-4
<input type="checkbox"/>	Pai Rico, Pai Pobre	Robert Kiyosaki, Sharon L. Lechter	Finanças pessoais	16,99€	20	978-0-7334-2609-5
<input type="checkbox"/>	O segredo do nº9	Paulo Pires	Thriller	14,99€	0	978-0-7334-2609-6
<input type="checkbox"/>	Steve Jobs	Walter Isaacson	Biografia	21,00€	2	978-0-7334-2609-7
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						

Figura 27- Seleção do livro pretendido

Editar Livro

Título :
Autor:

ISBN:
Preço:

Género:

Figura 28- Página de edição de livros

2.3.2.9 Remover Livro

Caso um Bibliotecário pretenda remover um livro da Biblioteca, deve seleccionar o livro pretendido (Figura 27) e, em seguida, o segundo botão localizado no canto superior direito da página da Figura 25.

De seguida, é exibida uma mensagem de confirmação do pedido de remoção (Figura 29). Caso o Bibliotecário, selecione a opção 'Sim', a mensagem da Figura 30 é apresentada na tela, caso contrário apresenta a mensagem da Figura 31.



Figura 29- Confirmação do pedido de remoção do livro



Figura 30- Mensagem de confirmação da remoção



Figura 31- Mensagem de cancelamento da remoção

2.3.3 Backoffice (Administrador)

2.3.3.1 Página Principal do Administrador

Após efetuar o *Login* no sistema, os Administradores têm acesso à página principal da aplicação (Figura 32), onde podem gerir todo o tipo de utilizadores da aplicação (Leitores, Bibliotecários e outros Administradores) ou encerrar a sua sessão através do botão ‘Sair’.

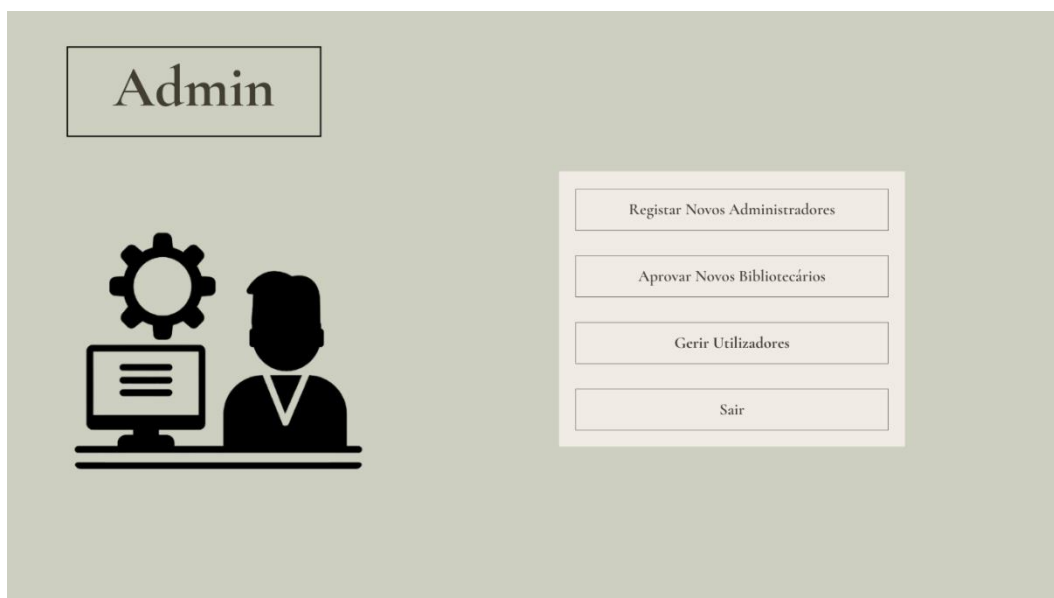


Figura 32- Página Principal do Administrador

2.3.3.2 Registar Novos Administradores

Ao selecionar o botão ‘Registar Novos Administradores’ na página da Figura 32, o Administrador é redirecionado para a página da Figura 33, onde poderá efetuar o registo de Administradores da aplicação (Figura 34).

Para isso, deve apenas preencher os campos necessários e concluir o registo. De seguida, será exibida na tela uma mensagem de confirmação do novo registo, tal como é apresentado na Figura 34.

Os Administradores são os únicos utilizadores com permissão para criar novos Administradores do sistema.



The image shows a registration form titled "Registar Administradores" in a dark box at the top left. To the right of the title is a circular icon with a right-pointing arrow. Below the title are four light orange rounded rectangular input fields, each with a label: "Nome", "Username", "E-mail", and "Palavra-Passe". In the bottom right corner, there is a circular icon with a checkmark.

Figura 33- Página de Registo de Administradores



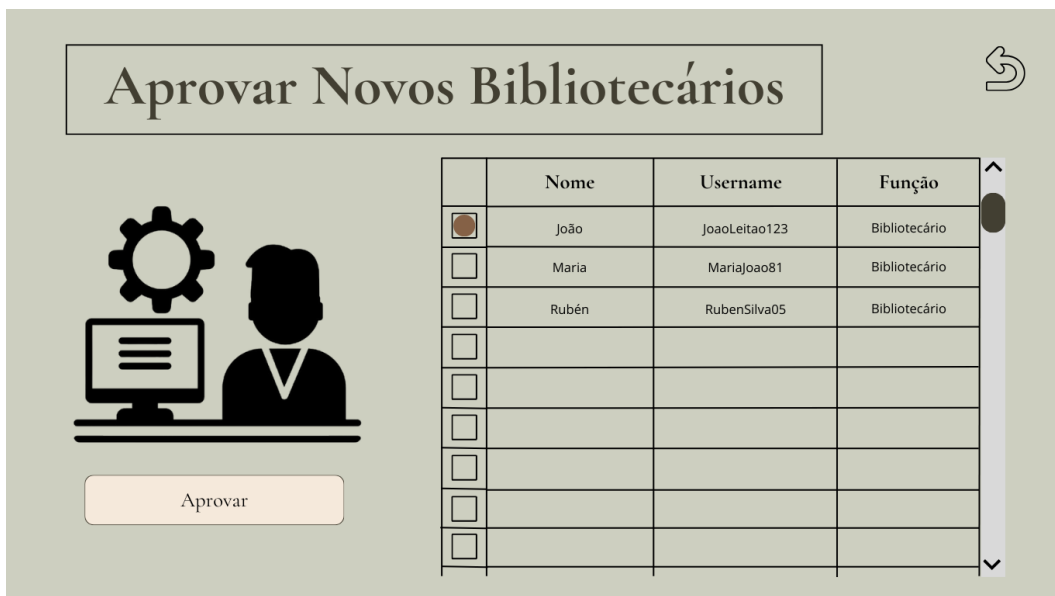
The image shows the same registration form as in Figure 33, but with a confirmation message overlay. The message is contained in a yellow rounded rectangle with a dashed border and a small circular icon with a right-pointing arrow in its top right corner. The text inside the message box reads "Novo Administrador criado com sucesso!". Below this text is a yellow button labeled "Continuar". The input fields for "Nome" and "Palavra-Passe" are visible behind the message box. The title "Registar Administradores" and the circular arrow icon are still present at the top, and the circular checkmark icon is in the bottom right corner.

Figura 34- Mensagem de confirmação do registo

2.3.3.3 Aprovar Registo de Bibliotecários

Ao seleccionar o botão 'Aprovar Novos Bibliotecários' na página da Figura 32, o Administrador é redireccionado para uma página onde poderá proceder à aprovação do registo de utilizadores com funções de Bibliotecário (Figura 35).

Nesta página, é apresentada uma listagem de Bibliotecários que aguardam aprovação para concluir o seu registo na aplicação. O Administrador deve seleccionar um Bibliotecário e, em seguida, seleccionar no botão 'Aprovar'. Posteriormente, é exibida uma mensagem de confirmação, tal como indicado na Figura 36.



	Nome	Username	Função
<input checked="" type="checkbox"/>	João	JoaoLeitao123	Bibliotecário
<input type="checkbox"/>	Maria	MariaJoao81	Bibliotecário
<input type="checkbox"/>	Rubén	RubenSilva05	Bibliotecário
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Figura 35- Página de Aprovação de Bibliotecários

2.3.3.4 Gestão de Utilizadores

Ao seleccionar o botão ‘Gerir Utilizadores’ na página da Figura 32, o Administrador é redireccionado para a página da Figura 37.

Nesta página, é apresentada uma listagem com todos os Leitores e Bibliotecários registados na aplicação, possibilitando ao Administrador efetuar a gestão do perfil de qualquer utilizador, através da seleção dos mesmos.



Figura 37- Página de Gestão de Utilizadores

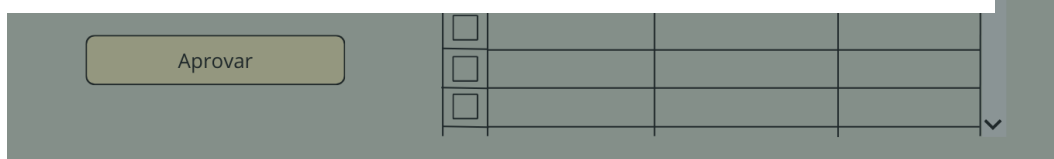


Figura 36- Mensagem de confirmação do registo

Após escolher o utilizador que pretende gerir, o Administrador é redireccionado para a página da Figura 38, onde lhe é possibilitado atualizar dados de perfil, bloquear ou ativar a conta.

Gerir Perfil

Nome: Sebastião

Username: SebVet05

Email: sebvettel5@gmail.com

Morada: Rua das ruas, Vila Real

Contacto: 987654321

Função: Leitor

Estado: Ativo

Atualizar Perfil

Ativar Conta

Bloquear Conta

Figura 38- Página de Gestão de Perfil de um Utilizador

2.3.3.5 Atualizar Perfil de Utilizador

Ao seleccionar o botão 'Atualizar Perfil' na página da Figura 38, o Administrador poderá alterar alguns parâmetros do perfil do Utilizador em questão, nomeadamente a sua morada e contacto telefónico (Figura 39). De seguida, deve concluir a edição, seleccionando o botão localizado no canto inferior direito da tela.

Gerir Perfil

Nome: Sebastião

Username: SebVet05

Email: sebvettel5@gmail.com

Morada: Rua das ruas, Vila Real

Contacto: 987654321

Função: Leitor

Estado: Ativo

Atualizar Perfil

Ativar Conta

Bloquear Conta

Figura 39- Atualização de Perfil

2.3.3.6 Bloquear Conta

Ao selecionar o botão 'Bloquear Conta' na página da Figura 38, o Administrador poderá bloquear a conta de determinado utilizador, indicando o motivo de bloqueio associado (Figura 40).

Após a seleção do motivo, o estado da conta é alterado para 'Bloqueado' e é exibida na tela uma mensagem de confirmação do bloqueio (Figura 41).

The screenshot shows the 'Gerir Perfil' (Manage Profile) page. The user's name is 'Sebastião'. A modal dialog is open, prompting the user to 'Indique o motivo do bloqueio:' (Indicate the reason for blocking). The dialog lists five reasons: 'Vários atrasos na devolução de livros' (Various delays in book return), 'Perda ou danos a livros' (Loss or damage to books), 'Conduta inadequada' (Inadequate behavior), 'Violação de políticas da biblioteca' (Violation of library policies), and 'Compartilhamento não autorizado de credenciais' (Unauthorized sharing of credentials). The background form shows fields for Username, Email, Morada (Address), and Contacto (Contact), along with an 'Atualizar Perfil' (Update Profile) button and a confirmation checkmark icon.

Figura 40- Seleção do motivo de bloqueio

The screenshot shows the 'Gerir Perfil' page after the account has been blocked. A modal dialog displays the message 'Utilizador Bloqueado com Sucesso!' (User blocked successfully) with a 'Continuar' (Continue) button. The background form shows the user's profile information, including the name 'Sebastião', email 'sebastiao@leilao.com', address 'Rua das ruas, Vila Real', and contact number '987654321'. The 'Estado' (Status) field is now 'Bloqueado' (Blocked). The 'Atualizar Perfil' button and the confirmation checkmark icon are still visible.

Figura 41- Mensagem de confirmação de bloqueio

2.3.3.7 Ativar Conta

Caso o Administrador pretenda reativar a conta de determinado utilizador que se encontre bloqueado, deve seleccionar o botão 'Ativar Conta' na página da Figura 38.

Nesse instante, o estado da conta é alterado para 'Ativo' e é exibida na tela uma mensagem de confirmação da ativação (Figura 42).



The screenshot displays the 'Gerir Perfil' (Manage Profile) interface. A yellow confirmation dialog box is centered on the screen, displaying the text 'Conta Ativa com sucesso!' (Account Active successfully!) and a 'Continuar' (Continue) button. The background form contains the following fields and values:

- Nome: Sebastião
- Username: [redacted]
- Email: sebv...
- Morada: Rua das ruas, Vila Real
- Contacto: 987654321
- Função: Leitor
- Estado: Ativo

Additional UI elements include an 'Atualizar Perfil' (Update Profile) button with a refresh icon, a 'Conta' button, and a green checkmark icon in the bottom right corner.

Figura 42- Mensagem de confirmação de ativação

3- FASE 3

A terceira e última fase deste projeto consiste na materialização de todas as propostas delineadas nas fases anteriores. Nesta etapa, procede-se à implementação das funcionalidades previamente descritas, aplicando as técnicas e metodologias discutidas ao longo de todo o relatório. Este processo é fundamental para consolidar a transição entre a conceção teórica e a aplicação prática.

Desta forma, o projeto foi desenvolvido utilizando a plataforma *Visual Studio Community*, seguindo o padrão *ASP.NET Model-View-Controller (MVC)*. Foi implementado com a versão 8.0 do *.NET Framework*, aproveitando os recursos mais recentes para garantir maior desempenho e compatibilidade. A autenticação foi configurada para contas individuais, proporcionando uma gestão segura e personalizada dos utilizadores. Por forma a facilitar a colaboração em grupo, utilizou-se o *GitHub* como plataforma de versionamento.

A implementação da terceira fase está organizada em quatro pilares fundamentais:

- **Desenvolvimento do *Frontend*:** Construção de interfaces que proporcionem uma experiência de utilizador eficiente e intuitiva, atendendo às necessidades específicas dos Leitores, Bibliotecários e Administradores;
- **Desenvolvimento do *Frontoffice* e *Backoffice*:** Implementação de ferramentas robustas de gestão tanto para os utilizadores com funções administrativas e operacionais, como para os utilizadores finais, assegurando o controlo eficaz do sistema;
- **Integração com a Base de Dados:** Estabelecimento de ligações consistentes entre a aplicação e a base de dados do sistema, definida na segunda fase, garantindo a fiabilidade e coerência dos dados tratados;
- **Testes e Validação:** Realização de um conjunto de testes rigorosos que permitam verificar o cumprimento dos requisitos funcionais e a integração correta entre os diferentes módulos do sistema.

3.1 Autenticação no Sistema

3.1.1 Registo de Utilizadores

Para a implementação do registo de novos utilizadores, adotámos uma abordagem estruturada. Inicialmente, incorporámos *scaffold items* para facilitar as funcionalidades relacionadas com o registo, *login* e confirmação de registo. Adicionalmente, na criação da classe 'AppUser.cs', configurámos a herança de atributos da classe 'IdentityUser', garantindo a integração com o sistema de autenticação da aplicação web.

Os dados introduzidos durante o processo de registo são armazenados na tabela 'AspNetUsers', pertencente ao esquema de base de dados gerido pelo *framework* de identidade. Esta tabela armazena as informações essenciais de autenticação e outros atributos dos utilizadores.

Quando o utilizador conclui o seu registo, todos os dados introduzidos são simultaneamente introduzidos nas seguintes tabelas: 'AspNetUsers', 'Autenticado' e 'Leitor' ou 'Bibliotecário', conforme o caso. Este método otimiza a gestão de dados e facilita futuras operações, garantindo uma integração eficiente e consistente em todo o sistema.

Nas Figuras 43 e 44, é implementada a lógica de inserção de dados do registo nas tabelas 'Autenticado', 'Bibliotecário' e 'Leitor', assegurando que o sistema possa aceder de forma centralizada e eficiente às informações armazenadas.

```
var autenticado = new Autenticado
{
    Username = user.Username,
    Password = Input.Password,
    Contacto = user.PhoneNumber,
    Nome = user.Nome,
    Contacto2 = user.PhoneNumber2,
    Contacto3 = user.PhoneNumber3
};

_context.Autenticados.Add(autenticado);
await _context.SaveChangesAsync();

if (Input.Role == "Bibliotecario")
{
    var bibliotecario = new Bibliotecario
    {
        UsernameBib = user.Username
    };
    _context.Bibliotecarios.Add(bibliotecario);
}
```

Figura 43- Inserção de dados 'Autenticado' e 'Bibliotecário'

```
else if (Input.Role == "Leitor")
{
    var leitor = new Leitore
    {
        UsernameLei = user.Username,
        Morada = Input.Address,
        Email = Input.Email,
    };
    _context.Leitores.Add(leitor);
}

await _context.SaveChangesAsync();
```

Figura 44- Inserção de dados 'Leitor'

3.1.1.1 Diferenciação Tipos de Utilizador

A distinção entre o registo de Leitores e de Bibliotecários é implementada no método ‘OnPost’, conforme ilustrado na Figura 45.

No caso do registo de um Bibliotecário, foram definidos menos campos obrigatórios, simplificando o processo e permitindo uma diferenciação mais evidente em relação ao registo de Leitores. Este mecanismo assegura uma personalização adaptada às necessidades específicas de cada tipo de utilizador.

```
if (ModelState.IsValid)
{
    var user = new AppUser
    {
        Nome = Input.Nome,
        Role = Input.Role,
        PhoneNumber = Input.PhoneNumber,
        PhoneNumber2 = Input.PhoneNumber2,
        PhoneNumber3 = Input.PhoneNumber3,
        UserName = Input.Username,
        Email = Input.Email,
    };

    if (Input.Role == "Bibliotecario")
    {
        ModelState.Remove("Input.Email");
        ModelState.Remove("Input.Address");
        user.EmailConfirmed = true;
    }
    else
    {
        user.Address = Input.Address;
        user.Email = Input.Email;
    }
}
```

Figura 45- Distinção de Utilizadores

3.1.2 Login

3.1.2.1 Registo Pendente para Bibliotecários

Depois de efetuar o registo na aplicação web como Bibliotecário, este estará impedido de aceder ao sistema até que um Administrador aprove o seu registo.

O código apresentado verifica o valor do campo ‘Validado’ na tabela ‘validarRegisto’. Caso a validação ainda esteja pendente, o acesso à aplicação é impedido até que uma aprovação seja realizada por um Administrador (Figura 46). Após essa aprovação, o Bibliotecário poderá aceder à aplicação normalmente ao efetuar *login*.

```
else if (bibliotecario != null && (validarRegisto == null || !validarRegisto.Validado))
{
    await _signInManager.SignOutAsync();
    ModelState.AddModelError(string.Empty, "O seu registo está pendente... Aguarde pela aprovação.");
    return Page();
}
```

Figura 46- Verificação do Estado de Validação de Registo

3.1.2.2 Bloqueio de Utilizadores

Aquando da tentativa de *login* na aplicação *web*, numa situação ideal, o fluxo de autenticação ocorre com sucesso e o utilizador acede ao *site*. No entanto, um utilizador que esteja bloqueado pelo sistema não deve conseguir ter acesso ao mesmo (Figura 47).

Esta funcionalidade é essencial para restringir o acesso de utilizadores cuja conta tenha sido previamente bloqueada por um Administrador, garantindo assim a segurança e a integridade do sistema.

```
var bloqueio = await _context.Bloquears
    .Where(b => b.Username == Input.Username && b.Bloqueado)
    .FirstOrDefaultAsync();

if (bloqueio != null)
{
    var motivoBloqueio = bloqueio.IdMotivoBloqNavigation;

    if (motivoBloqueio != null)
    {
        ModelState.AddModelError(string.Empty, $"O acesso está bloqueado. Motivo: {motivoBloqueio.Descricao}");
        return Page();
    }
    else
    {
        ModelState.AddModelError(string.Empty, "Erro ao verificar o motivo do bloqueio.");
        return Page();
    }
}
```

Figura 47- Restrição de Acesso a Utilizadores Bloqueados

O processo consiste nos seguintes passos:

I. Consulta da Base de Dados

É realizada uma consulta à tabela '*Bloquears*' para verificar se existe um registo correspondente ao nome de utilizador fornecido e se o campo 'Bloqueado' se encontra ativado.

II. Validação do Motivo de Bloqueio

Caso o utilizador esteja bloqueado, o sistema tenta aceder ao motivo do seu bloqueio, através da chave estrangeira '*IdMotivoBloqNavigation*'.

Desta forma é apresentada uma mensagem de erro ao utilizador, indicando que o seu acesso está bloqueado e especificando a descrição do motivo. Se por alguma razão o motivo de bloqueio não se encontrar disponível, é exibida uma mensagem de erro genérica, informando o erro na tentativa de acesso ao motivo.

3.1.3 Gestão de Acessos

A gestão de acessos foi reforçada com a implementação de estruturas fundamentais. Em primeiro lugar, adicionámos todas as funções (*roles*) possíveis de utilizadores, juntamente com os respetivos *IDs*, à tabela 'AspNetRoles' na base de dados. De seguida, verificamos o tipo de utilizador que se encontra autenticado na aplicação *web*, atribuindo-lhe o papel correspondente. Por exemplo, caso seja identificado como Leitor, será atribuído esse papel específico (Figura 48).

```
if (leitor != null && !await _userManager.IsInRoleAsync(user, "Leitor"))
{
    await _userManager.AddToRoleAsync(user, "Leitor");
    _logger.LogInformation($"Role 'Leitor' atribuída ao utilizador {user.UserName}");
}
```

Figura 48- Exemplo de Atribuição de *Roles* aos Utilizadores

Após esta atribuição, o utilizador é redirecionado para a página inicial associada ao seu papel. Por fim, para garantir que cada utilizador apenas acede às páginas permitidas, é adicionada ao *Controller* de cada tipo de utilizador uma linha de código que define o seu role do seguinte tipo: *[Authorize(Roles = "papel do utilizador")]*.

Caso um utilizador não autorizado tente aceder a uma página que não lhe pertence, será exibida uma mensagem de aviso apropriada, indicando a negação de acesso ao mesmo (Figura 49).

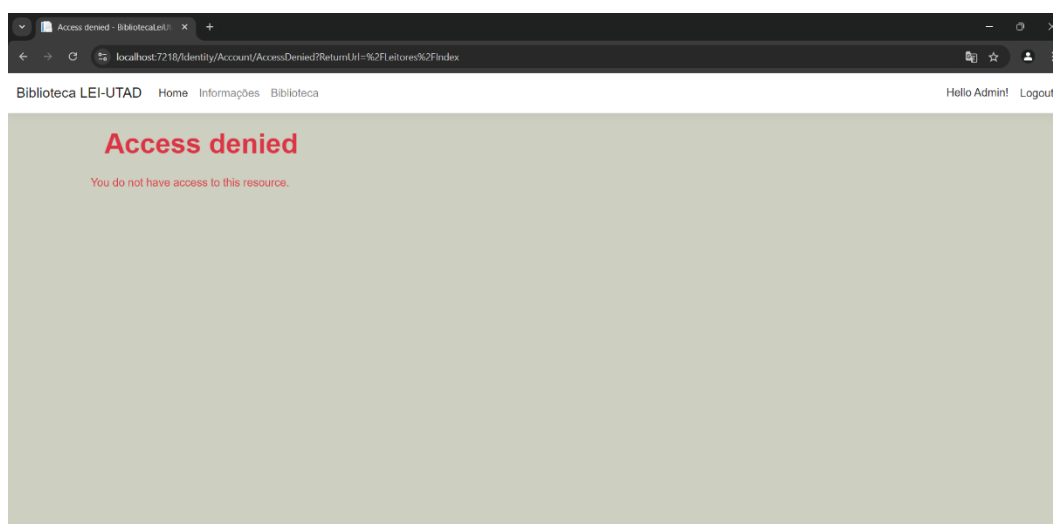


Figura 49- Página de "Acesso Negado"

3.2 Frontoffice

3.2.1 Requisição de Livros

Sempre que um Leitor desejar efetuar a requisição de um determinado livro, este deve selecionar a opção 'Requisitar' na sua biblioteca. Ao clicar nesta opção, é acionada a função 'confirmarRequisicao()', permitindo ao Leitor rever a sua solicitação e concluir o processo (Figura 50).

Esta função captura o campo 'ISBN' do livro selecionado e faz uma requisição *AJAX* para o servidor, enviada através do método *POST*, com o *ISBN* em formato *JSON*. Isso permite que a informação seja transmitida de forma eficiente para o servidor, sem a necessidade de recarregar a página. Após o envio da requisição, o servidor responde com um objeto *JSON* contendo um campo sucesso que indica se a operação foi realizada com êxito.

A partir deste momento, o Leitor poderá aceder às suas requisições na página 'Leitores/History' que, para além de conter os livros requisitados pelo mesmo, possui também a data prevista de entrega e o estado de cada requisição.

Se ocorrer algum erro durante a requisição, como problemas de rede ou falhas na comunicação, a função captura o erro e exibe uma mensagem de alerta ao Leitor, além de registar o erro no terminal para facilitar a depuração.

Caso o utilizador opte por cancelar a operação, a página é redirecionada para a tela principal da biblioteca, e uma mensagem de confirmação de cancelamento é exibida. Assim, o fluxo é interrompido, garantindo que o utilizador retorne à página inicial sem realizar a requisição.

```
<script>
function confirmarRequisicao() {
  const selectedLivro = document.querySelector('input[name="selectedLivro"]:checked');
  if (!selectedLivro) {
    alert("Por favor, selecione um livro para requisitar.");
    return;
  }

  const confirmacao = confirm("Tem certeza que deseja requisitar este livro?");
  if (confirmacao) {
    const isbn = selectedLivro.value;
    fetch('@Url.Action("Requisitar", "Leitores")', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ isbn: isbn })
    })
    .then(response => response.json())
    .then(data => {
      if (data.sucesso) {
        alert("Livro requisitado com sucesso!\nDispõe de um prazo de 15 dias para efetuar a sua devolução");
      } else {
        alert(data.mensagem);
      }
    })
    .catch(error => {
      alert("Ocorreu um erro. Tente novamente.");
      console.error("Erro:", error);
    });
  } else {
    alert("Operação cancelada com sucesso!");
    window.location.href = "@Url.Action("Biblioteca", "Leitores)";
  }
}
```

Figura 50- Confirmação de Requisições

3.3 Backoffice

3.3.1 Bibliotecário

Através de operações *CRUD* nos *Controllers* 'Livro' e 'Autor', o Bibliotecário é capaz de gerir os livros e autores disponíveis no sistema da biblioteca, adicionado, editando ou removendo parâmetros quando necessário.

3.3.1.1 Consulta de Requisições

O Bibliotecário deve conseguir aceder a todas as requisições de livros existentes na biblioteca. Desta forma, quando algum Leitor devolver um livro à biblioteca, a sua requisição é marcada como 'Entregue', sendo guardada a data de devolução.

3.3.1.2 Alteração de Estado de Requisições

A alteração de estado é feita com o auxílio de uma requisição *AJAX*, conforme apresentado na Figura 51. Ao clicar no botão de alteração de estado, a função captura o ID da requisição e exibe uma confirmação do Bibliotecário. Caso a confirmação seja positiva, é enviada uma requisição *AJAX* ao servidor que a processa e retorna um objeto *JSON*, incluindo um campo sucesso.

Se a operação for bem-sucedida, uma mensagem de alerta é exibida, informando que o estado foi alterado com sucesso, e a página é recarregada para refletir as mudanças. Caso haja algum erro, uma mensagem com a descrição do problema é exibida ao Bibliotecário.

```
<script>
  document.querySelectorAll('.alterar-estado').forEach(button => {
    button.addEventListener('click', function () {
      const idRequisicao = this.getAttribute('data-id');
      const confirmar = confirm("Tem certeza de que deseja alterar o estado desta requisição?");
      if (!confirmar) return;

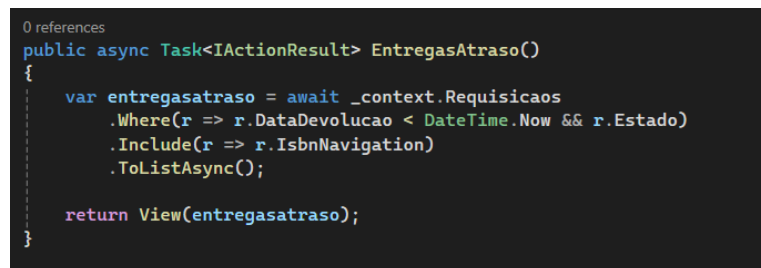
      fetch('/Livro/marcarComoEntregue', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({ IdRequisicao: idRequisicao })
      })
        .then(response => response.json())
        .then(data => {
          if (data.sucesso) {
            alert("Estado alterado com sucesso!");
            location.reload();
          } else {
            alert(data.mensagem);
          }
        })
        .catch(error => console.error('Erro ao alterar estado:', error));
    });
  });
</script>
```

Figura 51- Devolução de Livros

3.3.1.3 Consulta de Entregas em Atraso

Após a ultrapassagem do prazo de devolução de 15 dias, as requisições em atraso são exibidas nesta página, de forma a facilitar a identificação das mesmas por parte dos Bibliotecários.

A filtragem deste tipo de requisições é feita na chamada da função ‘EntregasAtraso’ no *Controller* ‘Livro’, onde o critério é a data de devolução ser anterior à data atual do sistema e o estado de entrega seja ‘true’, ou seja, ‘Por Entregar’ (Figura 52).

A screenshot of a code editor showing the implementation of the 'EntregasAtraso' method. The code is in C# and is part of an asynchronous action result. It filters a list of requisitions based on the return date being earlier than the current date and the status being 'true'. The filtered list is then converted to a list of objects and passed to a view.

```
0 references
public async Task<IActionResult> EntregasAtraso()
{
    var entregasatraso = await _context.Requisicaoos
        .Where(r => r.DataDevolucao < DateTime.Now && r.Estado)
        .Include(r => r.IsbnNavigation)
        .ToListAsync();

    return View(entregasatraso);
}
```

Figura 52- Critérios para Visualização de Entregas em Atraso

3.3.2 Administrador

3.3.2.1 Registo de Novos Administradores:

Conforme o nome sugere, esta funcionalidade permite criar uma entidade do tipo Administrador na tabela correspondente da base de dados, armazenando informações como 'Username', 'Nome', 'Email' e 'Password'.

3.3.2.2 Aprovação de Bibliotecários:

A aprovação de Bibliotecários consiste em apresentar uma lista de profissionais que ainda não receberam acesso à aplicação. Ao seleccionar o Bibliotecário desejado, o Administrador poderá aprovar o registo, alterando o campo Validado para True na base de dados.

3.3.2.3 Gestão de Perfis:

Na funcionalidade de gestão de perfis, o Administrador tem acesso a uma lista de todos os utilizadores presentes da base de dados.

```
[HttpGet]
[Route("api/Administradores/Utilizadores")]
// references
public async Task<IActionResult> ObterUtilizadores()
{
    try
    {
        // Retorna os utilizadores da base de dados
        var utilizadores = await _context.Users
            .Select(u => new
            {
                Id = u.Id,
                Nome = u.Nome,
                Username = u.UserName,
                Role = u.Role
            })
            .ToListAsync();

        if (utilizadores == null || utilizadores.Count == 0)
        {
            return NotFound("Nenhum utilizador encontrado.");
        }

        return Ok(utilizadores);
    }
    catch (Exception ex)
    {
        return StatusCode(500, $"Erro ao obter utilizadores: {ex.Message}");
    }
}
```

Figura 53 - Obter utilizadores

Após selecionar o utilizador desejado, o Administrador é redirecionado para uma nova interface que lhe permite a gestão detalhada do perfil do mesmo. Essa interface apresenta as informações relativas ao próprio utilizador .

Além disso, a funcionalidade oferece a possibilidade de bloqueá-lo, justificando o bloqueio com base num motivo, sendo em seguida registados na base de dados na tabela 'MotivoBloq' e na tabela 'Bloquear'.

3.4 Integração com a Base de Dados

Além das tabelas previamente definidas na Fase 2 do projeto, foi necessária a utilização de tabelas '*AspNetIdentity*', geradas automaticamente pelo *Entity Framework* no *Visual Studio*, para facilitar a implementação do sistema de autenticação e autorização na aplicação web.

Apesar do elevado número de tabelas criadas pelo *Identity*, apenas aquelas que atendiam às nossas necessidades foram integradas na base de dados. Estas incluem as tabelas '*AspNetUsers*', '*AspNetRoles*' e '*AspNetUserRoles*', cujas estruturas completas podem ser consultadas no **Anexo A** do presente relatório.

A tabela '*AspNetUsers*' foi personalizada para incluir informações específicas de cada tipo de utilizador do sistema. Por sua vez, os 'roles' definidos na tabela '*AspNetRoles*' foram usados para categorizar os diferentes níveis de acesso e permissões dentro sistema, enquanto a tabela '*AspNetUserRoles*' fez o mapeamento das relações entre os utilizadores e as suas funções.

A integração destas tabelas na nossa base de dados, consolidou a gestão de autenticação e controlo de acessos, garantindo escalabilidade e segurança ao sistema e simplificando a conexão entre utilizadores e as entidades do sistema.

CONSIDERAÇÕES FINAIS

O projeto desenvolvido ao longo de três fases reflete uma abordagem progressiva e estruturada, resultando num sistema completo e funcional.

Na primeira fase, a criação dos diagramas Entidade-Relacionamento (E-R) e a especificação de tanto requisitos funcionais como não funcionais foram essenciais para assegurar uma base sólida. Estes elementos proporcionaram uma compreensão clara das necessidades do sistema e anteciparam possíveis desafios, criando as fundações para um desenvolvimento alinhado com as expectativas dos utilizadores.

Na segunda fase, avançou-se para o mapeamento do Modelo Conceptual de Dados com o Modelo Físico, um passo crucial para a transição entre a parte teoria e a prática do desenvolvimento do projeto. A implementação do Modelo Físico da base de dados, utilizando SQL, permitiu construir a estrutura real do sistema, garantindo que os dados fossem armazenados de forma eficiente em conformidade com as especificações iniciais. A criação de *Mockups* para as interfaces de utilizador, tanto para o *frontoffice* como para o *backoffice*, deu uma visão antecipada da interação dos utilizadores com o sistema, assegurando que o design fosse funcional, intuitivo e cumprisse com todos os requisitos presentes no protocolo, garantindo assim uma experiência de utilização alinhada com as necessidades identificadas.

Deste modo, as duas fases complementam-se, reforçando a base técnica e operacional do projeto. O trabalho realizado até agora estabelece uma estrutura consistente para a fase seguinte que consiste na implementação da aplicação.

A conclusão da terceira fase consolidou o projeto, traduzindo as propostas iniciais em funcionalidades práticas e operacionais. A integração entre *frontoffice*, *backoffice* e base de dados garantiu uma comunicação eficiente entre os módulos, enquanto interfaces intuitivas e ferramentas robustas atenderam às necessidades de Leitores, Bibliotecários e Administradores. Os testes realizados validaram o cumprimento dos requisitos funcionais e não funcionais, assegurando a robustez e fiabilidade do sistema. Este processo destacou a importância da colaboração e da aplicação de tecnologias modernas, como o *ASP.NET MVC*, no desenvolvimento de soluções eficazes.

Em suma, o sistema final não apenas reflete a visão inicial delineada durante a análise de requisitos, mas também supera as expectativas ao oferecer um produto funcional, escalável e alinhado com as boas práticas de desenvolvimento. Esta experiência constitui um marco importante no desenvolvimento académico e prepara-nos para desafios futuros no campo das tecnologias de informação.

ANEXOS

1.1 Anexo A – Tabela ‘AspNetUsers’

```
CREATE TABLE [dbo].[AspNetUsers] (

    [Id] NVARCHAR (450) NOT NULL,
    [UserName] NVARCHAR (256) NULL,
    [NormalizedUserName] NVARCHAR (256) NULL,
    [Email] NVARCHAR (256) NULL,
    [NormalizedEmail] NVARCHAR (256) NULL,
    [EmailConfirmed] BIT NOT NULL,
    [PasswordHash] NVARCHAR (MAX) NULL,
    [SecurityStamp] NVARCHAR (MAX) NULL,
    [ConcurrencyStamp] NVARCHAR (MAX) NULL,
    [PhoneNumber] NVARCHAR (MAX) NULL,
    [PhoneNumberConfirmed] BIT NOT NULL,
    [AccessFailedCount] INT DEFAULT ((0)) NOT NULL,
    [LockoutEnabled] BIT DEFAULT ((0)) NOT NULL,
    [LockoutEnd] DATETIME NULL,
    [TwoFactorEnabled] BIT DEFAULT ((0)) NOT NULL,
    [Address] NVARCHAR (MAX) NULL,
    [Nome] NVARCHAR (MAX) DEFAULT (N'') NOT NULL,
    [PhoneNumber2] NVARCHAR (MAX) NULL,
    [PhoneNumber3] NVARCHAR (MAX) NULL,
    [Role] NVARCHAR (MAX) DEFAULT (N'') NOT NULL,

    CONSTRAINT [PK_AspNetUsers] PRIMARY KEY CLUSTERED ([Id] ASC)

);
```

1.2 Anexo B – Tabela ‘AspNetRoles’

```
CREATE TABLE [dbo].[AspNetRoles] (

    [Id] NVARCHAR (450) NOT NULL,
    [Name] NVARCHAR (256) NULL,
    [NormalizedName] NVARCHAR (256) NULL,
    [ConcurrencyStamp] NVARCHAR (MAX) NULL,

    CONSTRAINT [PK_AspNetRoles] PRIMARY KEY CLUSTERED ([Id] ASC)

);
```

1.3 Anexo C – Tabela ‘AspNetUserRoles’

```
CREATE TABLE [dbo].[AspNetUserRoles] (

    [UserId] NVARCHAR (450) NOT NULL,
    [RoleId] NVARCHAR (450) NOT NULL,

    CONSTRAINT [PK_AspNetUserRoles] PRIMARY KEY CLUSTERED ([UserId] ASC, [RoleId] ASC),
    CONSTRAINT [FK_AspNetUserRoles_AspNetRoles_RoleId] FOREIGN KEY ([RoleId]) REFERENCES [dbo].[AspNetRoles] ([Id]) ON DELETE CASCADE,
    CONSTRAINT [FK_AspNetUserRoles_AspNetUsers_UserId] FOREIGN KEY ([UserId]) REFERENCES [dbo].[AspNetUsers] ([Id]) ON DELETE CASCADE

);
```