

RELATÓRIO

Disciplina: Programação de Sistemas de Informação

Tema-Projeto: Módulo 9 - Sistema de Gestão de Biblioteca

CURSO TÉCNICO DE PROGRAMAÇÃO E GESTÃO DE SISTEMAS INFORMÁTICOS

Professor: Breno Sousa

Nomes dos Alunos:  Tiago Guerreiro, Rafael Belchior

Nº Alunos: 2480, 2474

29/05/2025

O relatório encontra-se em condições para ser apresentado \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ciclo de Formação 2023/2026  
Ano Letivo 2024/202

ÍNDICE

[INTRODUÇÃO 4](#_Toc200376369)

[PLANEAMENTO DO PROJETO 5](#_Toc200376370)

[DESENVOLVIMENTO DO PROJETO 6](#_Toc200376371)

[Classe Biblioteca 6](#_Toc200376372)

[Geração de Relatórios 6](#_Toc200376373)

[Armazenamento de Dados com JSON 6](#_Toc200376374)

[A Classe de Gestão da Biblioteca. 6](#_Toc200376375)

[Interface do Utilizador: De Tkinter para Consola 7](#_Toc200376376)

[FUNCIONAMENTO 8](#_Toc200376377)

[OBJETIVO 14](#_Toc200376378)

[CÓDIGO 15](#_Toc200376379)

ÍNDICE DE FIGURAS

[Imagem 1 - Esboço da GUI com Tkinter 8](#_Toc200376396)

[Imagem 2 - Menu 9](#_Toc200376397)

[Imagem 3 - Opção 1. Cadastrar Livro 9](#_Toc200376398)

[Imagem 4 - Opção 2. Cadastrar Aluno 9](#_Toc200376399)

[Imagem 5 - Opção 3. Consultar Livros Disponíveis 10](#_Toc200376400)

[Imagem 6 - Opção 4. Emprestar Livro 10](#_Toc200376401)

[Imagem 7 - Opção 4. ( Parte 2 ) 11](#_Toc200376402)

[Imagem 8 - Opção 5. Devolver livro 11](#_Toc200376403)

[Imagem 9 - Opção 6 - Criar Relatório – Opções 12](#_Toc200376404)

[Imagem 10 - Opção 6 - (Parte 2) 12](#_Toc200376405)

[Imagem 11 - Opção 6 - (Parte 3) 12](#_Toc200376406)

[Imagem 12 - Opção 7. Abrir Relatório 13](#_Toc200376407)

[Imagem 13 - Opção 7. (Parte 2) 13](#_Toc200376408)

[Imagem 14 - Opção 7. (Parte 3) 13](#_Toc200376409)

[Imagem 15 - Opção 8. Listar Relatórios 14](#_Toc200376410)

[Imagem 16 - Opção 9. Listar Livros 14](#_Toc200376411)

[Imagem 17 - Opção 10. Listar Alunos 14](#_Toc200376412)

INTRODUÇÃO

Este relatório detalha o planeamento e o desenvolvimento de um sistema de gestão de biblioteca. Onde o principal objetivo foi criar uma aplicação capaz de gerir livros e alunos, gerar relatórios e armazenar dados. Este documento descreve as etapas e as decisões tomadas, desde a conceção inicial das classes até à implementação das funcionalidades e à adaptação da interface de utilizador.

PLANEAMENTO DO PROJETO

No início do projeto, o foco foi na definição das classes neceesárias para uma biblioteca. Identificámos as duas classes principais que seriam a base do sistema:

* Livro: Para gerir os atributos de cada livro, e com métodos para por exemplo devolver e emprestar.
* Aluno: Para armazenar as informações dos alunos, e métodos para poder pegar emprestado um livro e devolvê-lo.

Com estas classes base definidas, o passo seguinte foi conceber a classe principal para executar todas as operações da biblioteca.

DESENVOLVIMENTO DO PROJETO

O desenvolvimento foi feito de forma modular, criando classes específicas de acordo com as necessidades:

# Classe Biblioteca

A classe Biblioteca foi criada como o centro do sistema. Esta classe armazena as listas de objetos Livro e Aluno e inclui métodos para:

* Adicionar, remover e pesquisar livros.
* Adicionar, remover e pesquisar alunos.
* Gerir o empréstimo e a devolução de livros, associando-os aos alunos.

# Geração de Relatórios

Para possibilitar a visualizam de dados, criarmos método para a criação de relatórios, o que permite exportar os dados formatados para um ficheiro de texto (.txt), fornecendo uma visão geral dos livros, alunos e registos de empréstimos, podendo escolher quais as informações pretendidas.

# Armazenamento de Dados com JSON

Para garantir que os dados não fossem perdidos entre as sessões, desenvolvemos uma classe dedicada aos ficheiros json. Esta classe é responsável por:

* Salvar os dados da biblioteca (livros, alunos, empréstimos) em um ficheiro JSON.
* Carregar esses dados ao iniciar a aplicação.

# A Classe de Gestão da Biblioteca.

Foi criada uma classe adicional para a gestão da biblioteca, que inclui métodos, como o preenchimento de dados padrão. Esta funcionalidade permite preencher a biblioteca com um conjunto inicial de livros e alunos, para fins de testes, também armazena as classes responsáveis pelos relatórios e pelos ficheiros json.

# Interface do Utilizador: De Tkinter para Consola

Originalmente, planeámos uma interface gráfica (GUI) com Tkinter, chegando a esboçar o seu layout. No entanto, devido a restrições de tempo, optámos por uma solução mais prática: uma interface com console. Para isso, desenvolvemos um sistema de menu no terminal, que permite aceder á maioria das funcionalidades através de texto.

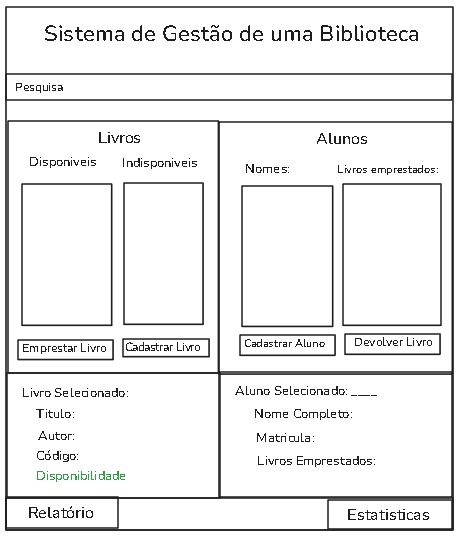


Imagem 1 - Esboço da GUI com Tkinter

FUNCIONAMENTO

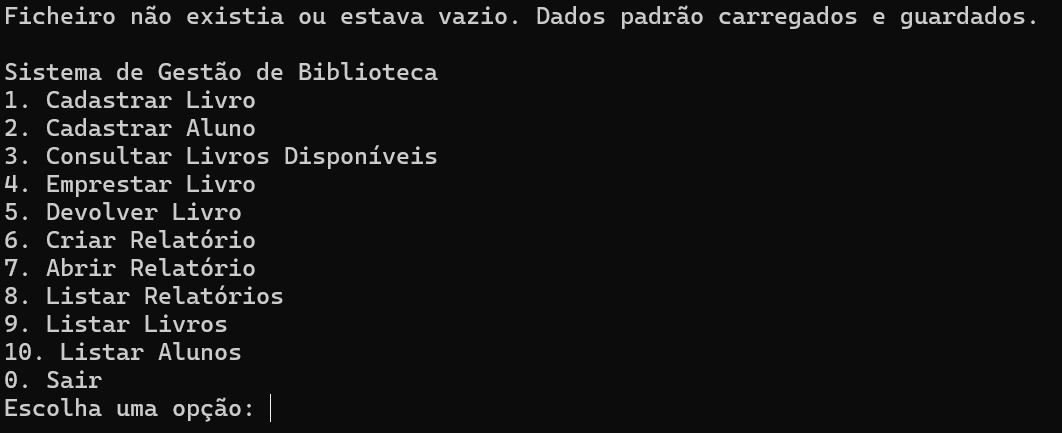


Imagem 2 - Menu

Quando se abre o código compilado, aparece um console com um menu em loop até que a opção seja 0.

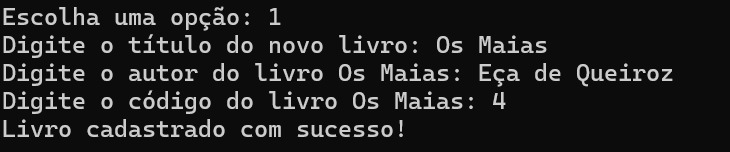


Imagem 3 - Opção 1. Cadastrar Livro

Na opção 1 cadastrar-se um livro na opção 1, onde é necessário definir um título, um autor e um código que não têm uma formatação especifica, pois, a ideia é não ser especifico pois assim pode ser usado em qualquer contexto, seja 0003 ou fe34-rw32, etc.

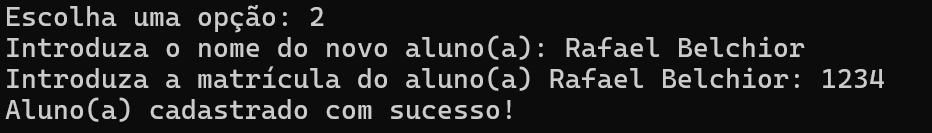


Imagem 4 - Opção 2. Cadastrar Aluno

Na opção 2, cadastra-se um aluno, com o nome e matricula que como o código dos livros não está limitado a formatos, podendo ser 1, L2305, as4-fs23, etc.

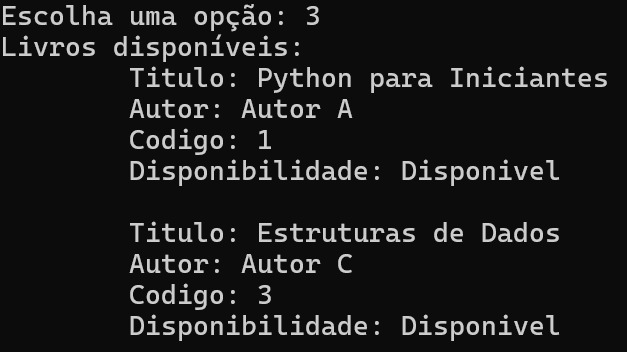


Imagem 5 - Opção 3. Consultar Livros Disponíveis

Na opção 3, apenas os livros que estão disponíveis para emprestar são listados.

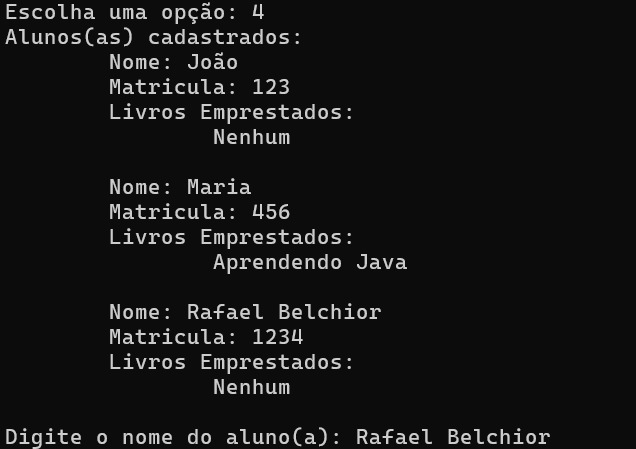


Imagem 6 - Opção 4. Emprestar Livro

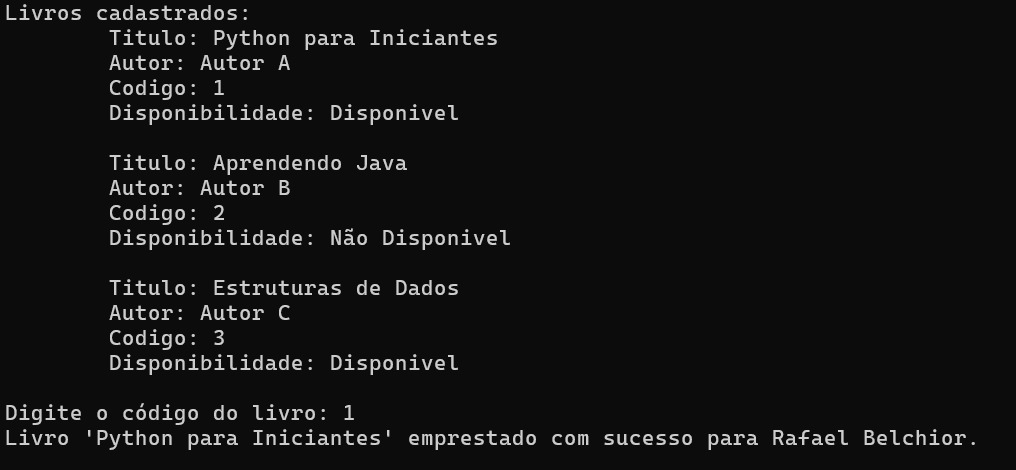


Imagem 7 - Opção 4. ( Parte 2 )

Na opção 4, os alunos são listados para que o utilizador introduza o nome do aluno, e caso acham 2 alunos com o mesmo nome pergunta qual a matricula, e depois lista os livros e pede para que se introduza o código do livro que será emprestado para o aluno escolhido.

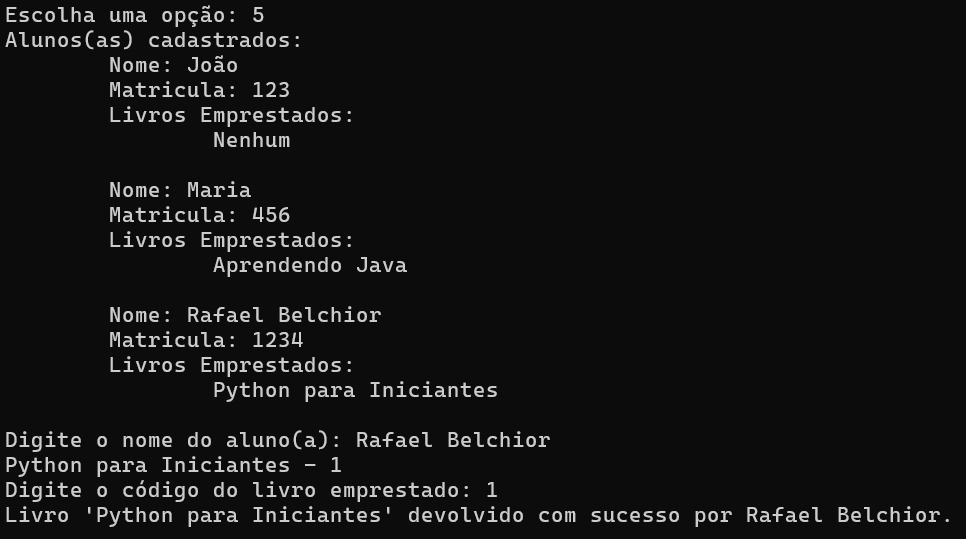


Imagem 8 - Opção 5. Devolver livro

Na opção 5, o código é parecido ao da opção 4, mas ao invés de escolher um livro entre todos os disponíveis. Aqui depois de ter-se escolhido um aluno, lista os livros emprestados ao aluno escolhido, para que um deles seja devolvido.

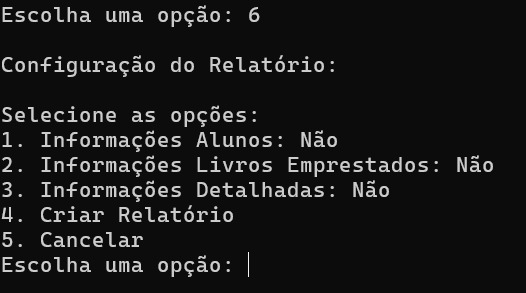


Imagem 9 - Opção 6 - Criar Relatório – Opções

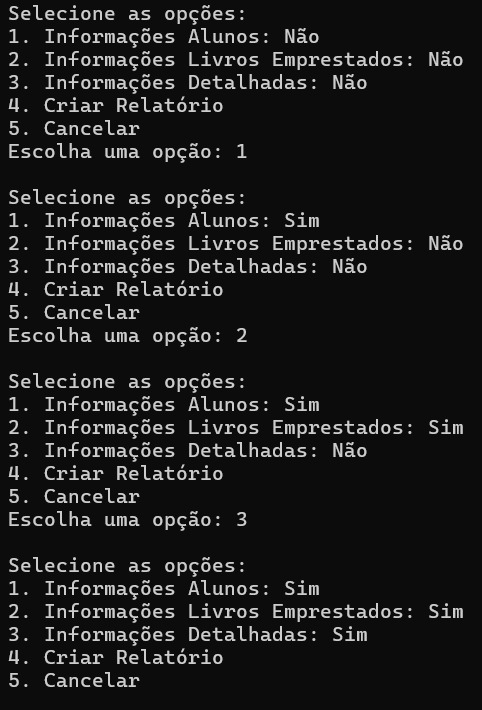


Imagem 10 - Opção 6 - (Parte 2)

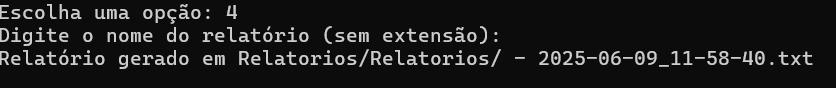


Imagem 11 - Opção 6 - (Parte 3)

Na opção 6, antes da criação do ficheiro é possível escolher quais informações é que seram salvas, e também qual será o nome do ficheiro.

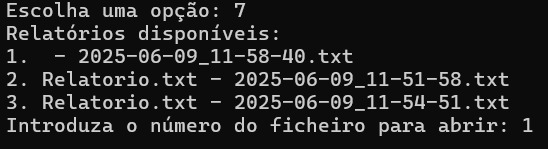


Imagem 12 - Opção 7. Abrir Relatório

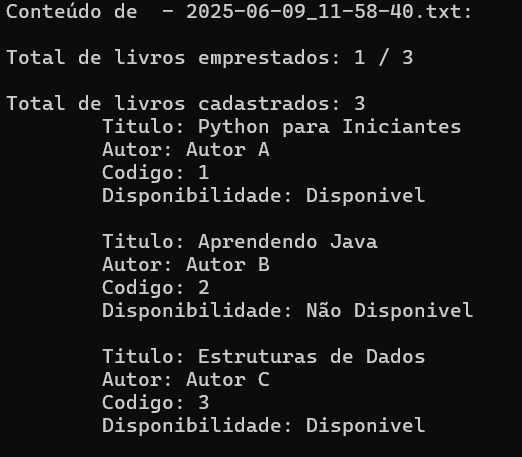


Imagem 13 - Opção 7. (Parte 2)

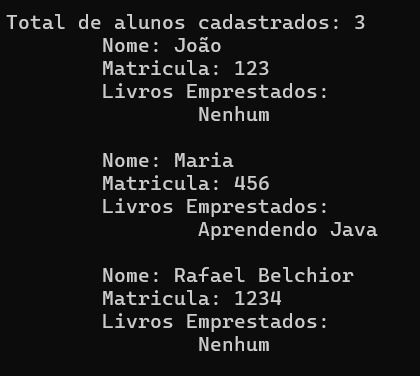


Imagem 14 - Opção 7. (Parte 3)

Na opção 7, primeiro escolhe-se um relatório para que este seja aberto e as informações exibidas no console.

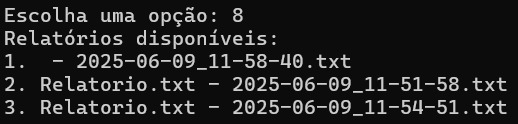


Imagem 15 - Opção 8. Listar Relatórios

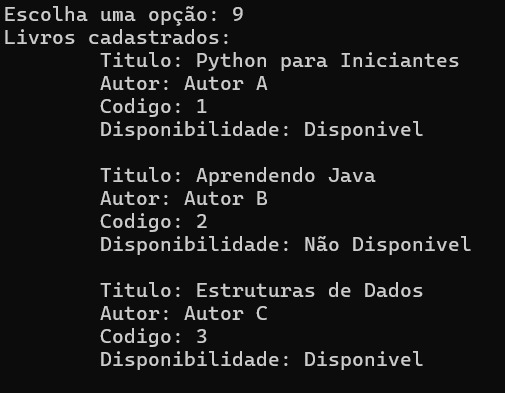


Imagem 16 - Opção 9. Listar Livros

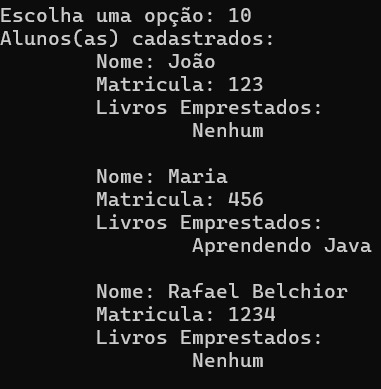


Imagem 17 - Opção 10. Listar Alunos

As opções 8 / 9 / 10 listam os relatórios / livros / alunos, e estas partes usam os mesmo métodos que são usados em outras opções do menu.

OBJETIVO

O objetivo deste projeto foi desenvolver um sistema de gestão de biblioteca escolar, utilizando a linguagem Python. O sistema permite o cadastro de livros e alunos, o controle de empréstimos e devoluções de livros, a consulta de livros disponíveis e a geração de relatórios, para além de guardar e carregar os dados em ficheiros json,

CÓDIGO

Biblioteca: Classe principal que gerencia os livros e alunos da biblioteca.

cadastrar\_livro(livro): Adiciona um novo livro à biblioteca, se o código não existir.

cadastrar\_aluno(aluno): Adiciona um novo aluno à biblioteca, se a matrícula não existir.

consultar\_livros\_disponiveis(): Retorna a lista de livros disponíveis.

emprestar\_livro(aluno, livro): Empresta o livro para o aluno.

ProcurarLivroPeloCodigo(codigo): Procura um livro pelo código.

devolver\_livro(aluno, livro): Devolve o livro de um aluno.

listar\_Livros(): Retorna a lista de todos os livros registrados.

listar\_Alunos(): Retorna a lista de todos os alunos registrados.

Procurar\_Aluno(nome, matricula): Busca um aluno pelo nome e matrícula.

Procurar\_Aluno\_Pelo\_Nome(nome): Busca alunos pelo nome.

to\_dict(): Converte os dados da biblioteca para dicionário.

from\_dict(data): Carrega os dados da biblioteca a partir de um dicionário.

Livro: Representa um livro da biblioteca.

\_\_init\_\_(titulo, autor, codigo): Inicializa um livro com título, autor e código.

emprestar(): Marca o livro como emprestado, se disponível.

devolver(): Marca o livro como disponível, se estava emprestado.

\_\_str\_\_(): Retorna uma representação em string do livro.

to\_dict(): Converte os dados do livro para dicionário.

from\_dict(data): Carrega os dados do livro a partir de um dicionário.

Aluno: Representa um aluno da biblioteca.

\_\_init\_\_(nome, matricula): Inicializa um aluno com nome e matrícula.

listar\_livros\_emprestados(): devolve um lista dos livros atualmente emprestados pelo aluno.

pegar\_livro(livro): Adiciona um livro à lista de livros emprestados do aluno, se permitido.

devolver\_livro(livro): Remove um livro da lista de livros emprestados do aluno.

\_\_str\_\_(): Retorna uma representação em string do aluno.

to\_dict(): Converte os dados do aluno para dicionário.

from\_dict(data): Carrega os dados do aluno a partir de um dicionário.

Relatorio: Gera relatórios sobre o estado da biblioteca.

ObterInformacoes(InformacoesLivrosEmprestados, InformacoesAluno, InformacoesLivro): Retorna informações selecionadas sobre a biblioteca.

ObterInformacoesAlunos(): Retorna informações detalhadas dos alunos.

ObterInformacoesLivros(): Retorna informações detalhadas dos livros.

Gerar\_ficheiro\_relatorio(InformacoesLivrosEmprestados, InformacoesAluno, InformacoesLivro): Gera e salva um relatório em arquivo.

Biblioteca\_JSON: Gerencia os dados da biblioteca em JSON.

GuardarDados(): Salva os dados da biblioteca em arquivo JSON.

CarregarDados(): Carrega os dados da biblioteca do arquivo JSON.

Verificar\_Se\_Dados\_Existem(): Verifica se existem dados salvos no arquivo JSON.

GestorDeDadosBiblioteca: Gerencia o carregamento, salvamento e inicialização dos dados da biblioteca.

\_\_init\_\_(): Inicializa os gestores de JSON e relatório.

CarregarDadosPadrão(): Carrega dados padrão na biblioteca.

GuardarDados(): Salva os dados atuais da biblioteca.

CarregarDados(): Carrega os dados salvos da biblioteca.

Criar\_Relatorio(nome\_arquivo): Cria um relatório completo da biblioteca.

VerificarFicheiroJSON\_ContemDados\_E\_CarregarOuCriar(): Carrega dados do JSON ou cria dados padrão se necessário.

Console\_Biblioteca: Interface de linha de comando para interação com o usuário.

menu(): Exibe o menu principal e gerencia as opções do usuário.

\_\_cadastrar\_livro(): Solicita dados e cadastra um novo livro.

\_\_cadastrar\_aluno(): Solicita dados e cadastra um novo aluno.

\_\_consultar\_livros\_disponiveis(): Exibe os livros disponíveis para emprestar.

\_\_emprestar\_livro(): Empresta de um livro para um aluno.

\_\_devolver\_livro(): Realiza a devolução de um livro por um aluno.

\_\_ListarLivros(): Lista todos os livros cadastrados.

\_\_ObterLivroDeLista(): Solicita e retorna um livro pelo código.

\_\_ObterLivro\_Emprestado\_De\_Lista(aluno): Solicita e retorna um livro emprestado pelo aluno.

\_\_ListarAlunos(): Lista todos os alunos cadastrados.

\_\_ObterAlunoDeLista(): Solicita e retorna um aluno pelo nome (e matrícula, se necessário).

\_\_ListarRelatorios(): Lista todos os relatórios gerados.

\_\_CriarRelatorio(): Configura e gera um novo relatório.

\_\_AbrirRelatorio(): Abre e exibe o conteúdo de um relatório existente.