08-03-2024

Jorge Fernandes e Hugo Oliveira

ISTEC

Gestão Biblioteca

Introdução à Programação

Índice

[Índice de Imagens 1](#_Toc192241765)

[Introdução 1](#_Toc192241766)

[Descrição do Problema 1](#_Toc192241767)

[Funcionalidades Implementadas 1](#_Toc192241768)

[Implementação 1](#_Toc192241769)

[Ambiente de Desenvolvimento 1](#_Toc192241770)

[Organização do Código 2](#_Toc192241771)

[Principais Desafios 2](#_Toc192241772)

[Testes 2](#_Toc192241773)

[Conclusão 3](#_Toc192241774)

[Código – Fonte 3](#_Toc192241775)

## Índice de Imagens

[Figura 1 - Bibliotecas 3](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362235)

[Figura 2 – inicializarTabuleiro() 3](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362236)

[Figura 3 - mostrarTabuleiro() 3](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362237)

[Figura 4 - verificarVitoria() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362238)

[Figura 5 - verificarEmpate() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362239)

[Figura 6 - jogadaValida() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362240)

[Figura 7 - jogar() 4](file:///C:\Users\Utilizador\Downloads\jogo_velha_IP.docx#_Toc190362241)

# Introdução

O objetivo deste projeto é desenvolver um sistema de gestão de biblioteca que permita aos utilizadores realizar operações comuns, como adicionar livros, pesquisar livros, emprestar e devolver livros, e gerar relatórios.

# Descrição do Problema

A gestão manual de bibliotecas pode ser ineficiente e suscetível a erros. Um sistema automatizado melhora a organização, rastreamento de livros e experiência do utilizador.

# Funcionalidades Implementadas

* **Registo de livros**: Permite adicionar informações sobre os livros, como título, autor, ISBN, gênero.
* **Pesquisa de livros**: Possibilita a busca por diferentes critérios, como título, autor ou gênero.
* **Empréstimo e devolução**: Registra empréstimos e devoluções com controle de datas.
* **Gestão de utilizadores**: Registra e gere utilizadores da biblioteca e os seus históricos.
* **Relatórios**: Gera relatórios de livros emprestados, disponíveis e histórico de empréstimos.
* Persistência de dados: Armazena informações em ficheiros CSV.

# Implementação

## Ambiente de Desenvolvimento

* Linguagem de programação utilizada: C
* Ferramentas: Dev-C++ (IDE)
* Ambiente: Windows

# Organização do Código

O código foi dividido em funções principais para garantir modularidade e organização:

* **salvarLivros()**: Salva a lista de livros no arquivo *livros.csv*.
* **carregarLivros()**: Carrega a lista de livros a partir do arquivo *livros.csv*.
* **salvarUtilizadores()**: Salva a lista de utilizadores no arquivo *utilizadores.csv*.
* **carregarUtilizadores()**: Carrega a lista de utilizadores a partir do arquivo *utilizadores.csv*.
* **adicionarLivro()**: Adiciona um novo livro à lista e salva os dados.
* **pesquisarLivro()**: Pesquisa livros.
* **registarUtilizador()**: Regista um novo utilizador.
* **emprestarLivro()**: Empresta um livro a um utilizador.
* **devolverLivro()**: Devolve um livro emprestado.
* **relatorios()**: Gera relatórios de livros emprestados, disponíveis ou históricos de utilizadores.
* **menu()**: Exibe o menu principal e permite escolher a ação a ser realizada.
* **main()**: Carrega dados de livros e utilizadores e exibe o menu principal.

# Principais Desafios

* Implementar a persistência dos dados nos ficheiros CSV.
* Garantir que um livro não seja emprestado mais de uma vez simultaneamente.
* Criar uma interface intuitiva e eficiente para os utilizadores.

# 

# Testes

Foram realizados vários testes para garantir o funcionamento correto da gestão da biblioteca, incluindo:

* Registo e remoção de livros.
* Consulta de livros por título, autor e gênero.
* Registo de empréstimos e devoluções.
* Geração de relatórios.

Todos os testes foram concluídos com sucesso, demonstrando que o sistema é capaz de lidar corretamente com todas as funcionalidades implementas para o trabalho.

# Conclusão

O projeto de Sistema de Gestão de Biblioteca foi concluído com sucesso, permitindo uma gestão organizada e eficiente de livros e utilizadores. Ao longo do desenvolvimento, adquirimos conhecimento em manipulação de ficheiros CSV, organização de estrutura de software e boas práticas de programação.

# Código – Fonte

Define o tamanho do tabuleiro como 3x3 e declara uma matriz 3x3 para armazenar o estado do tabuleiro.

Figura 1 - Bibliotecas

Preenche todas as posições da matriz tabuleiro com ' ', indicando que todas as casas estão vazias.

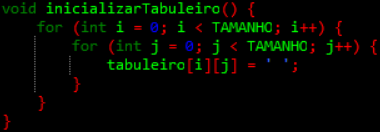


Figura 2 – inicializarTabuleiro()

Cria o tabuleiro.

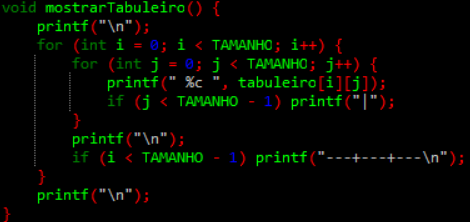


Figura 3 - mostrarTabuleiro()

Verifica se um jogador venceu, verificando:

* Linhas: Se todos os três elementos de alguma linha são iguais.
* Colunas: Se todos os três elementos de alguma coluna são iguais.
* Diagonais: Se os três elementos de uma diagonal são iguais.

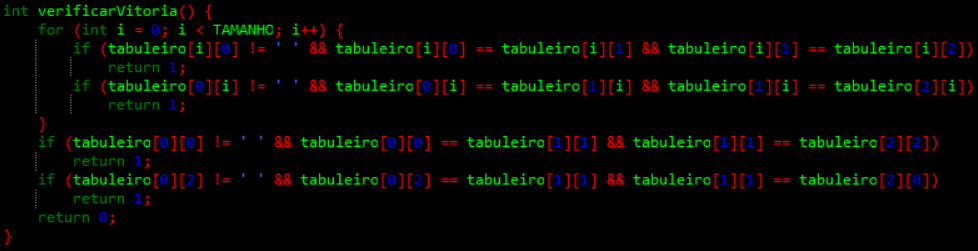
Retorna 1 (true) se houver um vencedor, caso contrário retorna 0.

Figura 4 - verificarVitoria()

Percorre toda a matriz tabuleiro e se houver alguma célula vazia, retorna 0 (não é empate).

Se todas as células estiverem preenchidas e ninguém venceu, retorna 1, ou seja, empate.

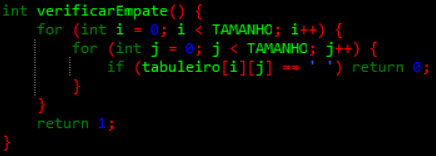


Figura 5 - verificarEmpate()

Verifica se a posição escolhida está dentro dos limites do tabuleiro e se está vazia.

Figura 6 - jogadaValida()

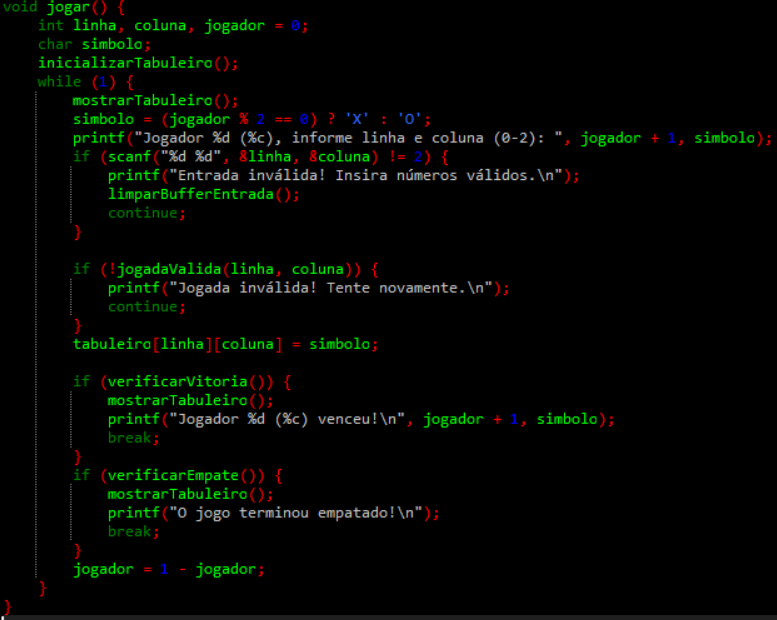
O jogo começa por inicializar o tabuleiro e, em seguida, solicita que o jogador atual (X ou O) faça a sua jogada. Após receber a entrada, verifica se a jogada é válida; caso contrário, exibe uma mensagem de erro e solicita uma nova tentativa. Se a jogada for válida, o tabuleiro é atualizado com o símbolo correspondente. Em seguida, o sistema verifica se há um vencedor e finaliza o jogo se necessário. Se todas as posições estiverem preenchidas e nenhum jogador tiver vencido, a partida é encerrada como empate.

Figura 7 - jogar()