

The image features a dark blue background with a decorative border of stylized books and a quill. The books are in various colors (red, orange, teal) and are shown in different orientations, some open and some closed. A white quill with a blue feather is positioned in the upper center. The title 'LIVRARIA' is written in large, bold, white capital letters with a red shadow effect.

# LIVRARIA

E-commerce marketplace

Participantes: Biatrice Fontes,  
Gustavo Enrique,  
Luiza Helena Pitzer e  
Maria Luisa Almeida.

# INTRODUÇÃO



O comércio eletrônico revolucionou a forma como produtos são ofertados e adquiridos, especialmente no setor de livros. Através de plataformas digitais, tornou-se possível conectar leitores a milhares de obras com apenas alguns cliques.




Este sistema foi pensado no formato de marketplace, um tipo de e-commerce em que diversos vendedores oferecem seus produtos em um mesmo ambiente virtual. Essa estrutura permite variedade, concorrência saudável de preços e maior alcance de público.



No contexto de uma livraria, o marketplace oferece vantagens como: catálogo diversificado, facilidade de comparação entre ofertas e maior acessibilidade ao conteúdo literário.








# ESTRUTURA DO CÓDIGO


com explicação


# LOGIN E LOGOUT

-  Durante o processo de login, o sistema solicita ao usuário seu nome e senha, e armazena essas informações como uma tupla em uma estrutura de pilha. A cada novo login, os dados são adicionados ao topo da pilha, representando o usuário ativo no momento.
-  No processo de logout, o sistema verifica se há algum usuário logado. Se houver, ele remove do topo da pilha os dados do último usuário que acessou, encerrando sua sessão.
-  Esse comportamento ilustra a lógica da estrutura de dados do tipo pilha, onde o último a entrar é o primeiro a sair (LIFO – Last In, First Out). É uma forma simples e eficiente de controlar sessões ativas em um sistema.

[https://colab.research.google.com/drive/1sLnPuNv5Y\\_pFAU7VtSPD2cdd9Rykiumq?usp=sharing](https://colab.research.google.com/drive/1sLnPuNv5Y_pFAU7VtSPD2cdd9Rykiumq?usp=sharing)

# LISTA DE PRODUTOS

 O código utiliza uma estrutura composta por uma lista principal que armazena outras listas internas, formando um vetor de vetores. Cada uma dessas listas internas representa um livro, contendo suas informações organizadas na mesma ordem: um número identificador, o título, o autor, o preço e a quantidade em estoque.

 Essa estrutura facilita a organização dos dados e permite percorrê-los de maneira eficiente usando laços de repetição. O uso de vetores é adequado porque permite armazenar múltiplos conjuntos de informações de forma compacta e estruturada, além de possibilitar o acesso direto aos dados por meio de índices.

<https://colab.research.google.com/drive/1nzTafoRAMsiQqAxajElVrqCxY-TWJcGT?usp=sharing>



# CARRINHO DE COMPRAS



O código simula o funcionamento de um carrinho de compras de uma livraria, onde o usuário pode escolher entre dois livros disponíveis e adicionar quantidades desejadas ao carrinho. O sistema permite que o usuário faça múltiplas escolhas, até decidir finalizar a compra. Ao final, o carrinho exibe todos os itens adicionados, incluindo informações como tipo de livro, autor, quantidade e valor total.



Para organizar os itens no carrinho, foi utilizada a estrutura de fila, que segue o princípio FIFO (First In, First Out), ou seja, o primeiro item que entra é o primeiro que sai. Isso representa bem situações reais, como uma fila de atendimento ou de pedidos, onde os itens são processados na ordem em que foram adicionados.

[https://colab.research.google.com/drive/1ikbd3okf\\_vBnuAmWDLA60XoOGHqIWDnP?usp=sharing](https://colab.research.google.com/drive/1ikbd3okf_vBnuAmWDLA60XoOGHqIWDnP?usp=sharing)

# MÉTODO DE PAGAMENTO



O código cria uma lista com vários métodos de pagamento, onde cada método tem um código numérico, um nome e um status que indica se está ativo (True) ou inativo (False). Ele define duas funções: uma para listar todos os métodos que estão ativos, mostrando seu código e nome, e outra para buscar um método pelo código informado.



O programa começa listando os métodos ativos, depois pede para o usuário digitar o código de um método desejado e busca esse método na lista. Se encontrado, exibe o nome e se ele está ativo ou inativo; caso contrário, avisa que o método não foi encontrado.

[https://colab.research.google.com/drive/1ikbd3okf\\_vBnuAmWDLA60XoOGHqIWDnP?usp=sharing](https://colab.research.google.com/drive/1ikbd3okf_vBnuAmWDLA60XoOGHqIWDnP?usp=sharing)

# DADOS DO USUÁRIO



O código começa com uma lista vazia para armazenar usuários logados. A função principal recebe nome e e-mail, cria um dicionário com essas informações representando o usuário e adiciona esse dicionário à lista. Sempre que um usuário é "logado", aparece uma mensagem de confirmação.



O programa adiciona dois usuários com nome e e-mail e, no final, imprime a lista completa de usuários logados, mostrando o nome e e-mail de cada um.

[https://colab.research.google.com/drive/1DY\\_E\\_cVY7Mg2USbSl8j0QOZ-agm0dNwG#scrollTo=\\_lxwA2m8AiBH&line=33&uniqifier=1](https://colab.research.google.com/drive/1DY_E_cVY7Mg2USbSl8j0QOZ-agm0dNwG#scrollTo=_lxwA2m8AiBH&line=33&uniqifier=1)



# CONCLUSÃO



O controle de login e logout foi implementado com uma pilha, que gerencia as sessões seguindo a lógica LIFO, onde o último usuário a entrar é o primeiro a sair. A lista de produtos foi utilizada para armazenar os dados dos livros, facilitando o acesso e a organização dessas informações. O carrinho de compras foi implementado como uma fila, garantindo que os itens sejam processados na ordem em que foram adicionados (FIFO). Para os métodos de pagamento, utilizou-se uma matriz, permitindo armazenar múltiplos dados sobre cada método, como código, nome e status, facilitando a busca e listagem de opções ativas. Por fim, os dados dos usuários foram armazenados em uma lista simples, registrando informações de forma clara. Essas implementações mostram como estruturas de dados básicas são fundamentais para criar sistemas eficientes e funcionais.

<https://colab.research.google.com/drive/144oU5Qo7Yw1uM-IOkqQeMBCZEEfq4E7g?usp=sharing>





**OBRIGADA!**

