

 <b>Serviço Nacional de Aprendizagem Industrial</b> Santa Catarina	RECUPERAÇÃO PRÁTICA	Desempenho
	Data: 07/08/2025	
	Docente: Vinicius Trindade	
	Aprendizagem Industrial em informática para internet	
	Unidade Curricular: Lógica e paradigmas da programação	
	Turma: T TIIN 2025/1 INT1	
	Estudante:	

#### Escala de Desenvolvimento:

**0,0 à 6,9 - Inapto / 7,0 à 8,4 - Em Desenvolvimento / 8,5 à 10,0 - Apto**

#### CAPACIDADES

C1 - Aplicar técnicas de programação na elaboração de algoritmos inerentes aos sistemas de TI.  
 C2 - Empregar estruturas de controle e dados, como variáveis, constantes, operadores, estrutura.

#### CONTEXTUALIZAÇÃO

A WEG, multinacional brasileira de referência global em automação, energia e tecnologia industrial, está ampliando sua frente de atuação no setor de soluções digitais personalizadas. A empresa está investindo fortemente no desenvolvimento de sistemas baseados em lógica computacional, com foco na automação de atividades operacionais, organização de informações e otimização da produtividade em empresas de diferentes segmentos — como logística, varejo, educação e serviços.

Para isso, a WEG conta com uma equipe de desenvolvedores responsáveis por prototipar sistemas com base em fluxos lógicos bem definidos, utilizando estruturas como listas, mapas e algoritmos para simular funcionalidades reais do dia a dia empresarial. Esses protótipos funcionais ajudam a entender as necessidades dos clientes, testar soluções e preparar o terreno para sistemas maiores e mais complexos.

Você foi contratado como desenvolvedor júnior para auxiliar nesse processo. Sua missão é aplicar técnica de programação de algoritmos para implementar um protótipo funcional em Java, simulando um sistema que possa representar uma das soluções que a WEG poderá oferecer a seus parceiros. O sistema deve seguir os princípios da lógica de programação, fazer uso estruturado de List e Map, e apresentar um menu interativo com múltiplas funcionalidades.

Esse protótipo poderá servir de base para a construção futura de plataformas web, sistemas automatizados e aplicações corporativas.

# Desafio: Sistema de Empréstimo de Livros

## Descrição Geral:

Aplicar técnicas de programação estruturada em Java para desenvolver um sistema de gerenciamento de empréstimos de livros em uma biblioteca. O sistema deve permitir o cadastro de usuários, autenticação por CPF e senha, e a realização de operações como empréstimo, visualização e devolução de livros por meio de um menu interativo. Os dados devem ser armazenados em listas paralelas.

---

## Requisitos Obrigatórios

### 1. Cadastro de Usuários

- Cada usuário deve ser identificado por um CPF único e possuir uma senha.
  - Antes de cadastrar um novo usuário, o sistema deve verificar se o CPF já está registrado.
  - Armazenar em listas paralelas:
    - `List<String> cpfsUsuarios`
    - `List<String> senhasUsuarios`
- 

### 2. Login de Usuários

- O sistema deve permitir que o usuário informe CPF e senha para autenticação.
  - Caso o CPF não exista ou a senha esteja incorreta, uma mensagem de erro apropriada deve ser exibida.
  - Após login bem-sucedido, o usuário deve acessar o menu de operações da conta.
- 

### 3. Menu Principal (fora da conta)

- Cadastrar novo usuário
  - Fazer login
  - Sair do sistema
- 

### 4. Menu de Operações (após login)

- Fazer novo empréstimo
- Listar meus empréstimos
- Devolver um livro
- Listar todos os empréstimos (exclusivo para CPF "conf")
- Sair da conta (retorna ao menu principal)

---

## Funcionalidades do Sistema

### 1. Fazer Novo Empréstimo

O usuário poderá emprestar um livro informando:

- Título do livro
- Autor do livro
- Data de devolução prevista (ex: "10/08/2025")

O sistema deve gerar automaticamente um código de empréstimo exclusivo, no formato **E1**, **E2**, **E3**..., utilizando uma variável global `int proximoCodigo = 1` com incremento a cada empréstimo.

Cada usuário poderá manter no máximo 3 empréstimos ativos.

Armazenar os dados em listas paralelas:

java

CopiarEditar

```
List<String> emprestimosCpf           // CPF do usuário
List<String> emprestimosTitulo       // Título do livro
List<String> emprestimosAutor        // Autor do livro
List<String> emprestimosDataDevolucao // Data prevista para devolução
List<String> codigosEmprestimos      // Código do empréstimo gerado
```

---

### 2. Listar Meus Empréstimos

Exibir todos os empréstimos ativos do usuário logado, listando:

- Código do empréstimo
- Título do livro
- Autor
- Data de devolução prevista

---

### 3. Devolver um Livro

- O sistema deve exibir os empréstimos do usuário logado.
  - O usuário informa o código do empréstimo que deseja devolver.
  - O sistema verifica se o código pertence ao CPF logado.
  - Se for válido, o sistema remove os dados das listas paralelas nos respectivos índices.
- 

### 4. Listar Todos os Empréstimos (acesso restrito)

Disponível apenas para o usuário com CPF "conf" e senha "conf".

O sistema deve listar todos os empréstimos ativos no sistema, incluindo:

- CPF do usuário
  - Código do empréstimo
  - Título do livro
  - Autor
  - Data de devolução prevista
- 

### Regras de Negócio

- O CPF deve ser único no momento do cadastro.
- Cada usuário pode manter no máximo 3 empréstimos ativos.
- O código do empréstimo deve ser gerado automaticamente e sequencialmente, no formato E1, E2, E3..., com base em uma variável global.
- A senha deve ser validada corretamente no momento do login.
- Um empréstimo só pode ser devolvido pelo usuário que o realizou.

RESULTADOS E ENTREGAS
<p>1 - O sistema permite realizar todas as funcionalidades obrigatórias conforme descritas no desafio.</p> <p>2 - Todas as operações (cadastro, login, empréstimo, devolução, listagem, saída) funcionam corretamente em sequência, sem falhas.</p> <p>3 - O programa trata corretamente situações como: CPF duplicado, limite de empréstimos, devolução inválida, login inexistente, entre outras.</p> <p>4 - Apresentar o código-fonte de forma legível e organizada, com indentação adequada, nomes de variáveis significativos e métodos bem estruturados.</p>

5 - Garantir que o código compile e execute corretamente, sem erros de sintaxe ou falhas lógicas, possibilitando o uso contínuo do sistema até a escolha explícita pela finalização.

Tempo estimado 120 minutos

#### LISTA DE VERIFICAÇÃO

Avaliação Prática -

ATIVIDADE 01:

CRITÉRIOS DE AVALIAÇÃO	CAP	Desenvolvimento (Nota)	JUSTIFICATIVA
O aluno aplicou técnicas de programação para implementar o <b>cadastro de usuários</b> corretamente e de forma funcional?	1 e 2 1,0		
O aluno aplicou técnicas de programação para implementar o <b>login</b> de clientes corretamente e de forma funcional?	1 e 2 1,0		
O aluno aplicou técnicas de programação para implementar o <b>menu de operações</b> corretamente e de forma funcional?	1 e 2 1,0		
O aluno aplicou técnicas de programação para implementar a funcionalidade para <b>fazer novo empréstimo de livro</b> corretamente e de forma funcional?	1 e 2 1,0		
O aluno aplicou técnicas de programação para implementar a funcionalidade para <b>listar os empréstimos do usuário</b> corretamente e de forma funcional?	1 e 2 1,0		
O aluno aplicou técnicas de programação para implementar a funcionalidade para <b>devolver livro</b> corretamente e de forma funcional?	1 e 2 2,0		

O aluno aplicou técnicas de programação para implementar a funcionalidade para <b>fazer login com usuário “conf”</b> corretamente e de forma funcional?	1 e 2  1,0		
O aluno aplicou técnicas de programação para implementar a funcionalidade para <b>listar todos os empréstimos do sistema</b> corretamente e de forma funcional?	1 e 2  2,0		