

# Trabalho 1: Inventário Simples

Objetivo: Criar um inventário básico para livros, inspirado no sistema apresentado nos exercícios 9.3 a 9.5 do manual. Apesar da base ser semelhante ao exemplo do catálogo de livros, o sistema deve ser mais rico, aproximando-se de um inventário básico, incluindo campos como **referência**, **preço**, **descrição** e **quantidade**.

## Critérios de Avaliação:

### 1. Nível Básico (Nota de Partida, entre 10 a 13 valores):

- Reproduzir o modelo dos exercícios 9.3 a 9.5 usando HTML, CSS e JavaScript simples, com os campos adicionais exigidos (referência, preço, descrição e quantidade).
- Este é o mínimo esperado para aprovação neste trabalho. Não esquecer que haverá vídeos a explicar como atingir estes mínimos.

### 2. Nível Intermédio (Notas de 14 a 16 valores):

- Introduzir melhorias funcionais, como permitir a edição e remoção de itens no inventário ou implementar filtros para facilitar a busca (aqui já se deixa ao formando o trabalho de fazer as suas próprias pesquisas, quer no Manual e bibliografia recomendados, quer nos sites das frameworks usadas).
- Estilizar a interface com maior atenção ao design e à usabilidade (usando HTML, CSS e/ou JS).
- Adicionar validações básicas, como garantir que todos os campos obrigatórios estão preenchidos e que valores numéricos (ex.: preço e quantidade) não aceitem entradas inválidas.

### 3. Nível Avançado (Notas de 17 a 20 valores):

- Implementar segurança básica com **Spring Security**, incluindo:
  - Uma página de login simples.
  - Garantir que apenas utilizadores autenticados podem aceder ao inventário.
- Explorar ferramentas no frontend à escolha, desde que o **backend seja implementado em Java (Spring Boot)**:
  - Os alunos podem usar HTML/CSS/JavaScript, React, Angular, Vue ou até Java no frontend, utilizando **Vaadin Flow**.
- Apresentar uma interface profissional, funcional e intuitiva, valorizando a experiência do utilizador.

## Nota:

Este trabalho é a sua oportunidade de consolidar o que aprendeu até cerca de meio do Curso. Começa com o básico e constrói gradualmente algo mais completo e interessante. Saiba que tem liberdade criativa no frontend e no design do inventário. Mas lembre-se: quanto mais cuidadoso for nos detalhes, mais o seu esforço será valorizado. Não desista, porque cada passo conta para o teu sucesso!

## Trabalho 2: Inventário Completo e Aplicação Opcional (para notas de 17-20)

Proposta aberta a discussão até sessão de 8 de janeiro

### Objetivo para Inventário Completo:

Criar um sistema de inventário funcional, seguro e mais avançado, com novas funcionalidades e foco em segurança e experiência do utilizador. Este trabalho vai além do Trabalho 1, introduzindo requisitos mais exigentes e abrindo possibilidades para o uso de tecnologias modernas de frontend.

### Requisitos Gerais para o Inventário:

- Implementar um **sistema de autenticação** com Spring Security.
- Adicionar **níveis de autorização baseados em "roles"**, diferenciando permissões de utilizadores (ex.: admin e utilizadores regulares).
- Funcionalidades obrigatórias no inventário:
  - **Adicionar, listar, editar e remover itens.**
  - Implementar **campos mais detalhados**, como:
    - Referência
    - Nome do produto
    - Preço
    - Descrição
    - Quantidade em stock
    - Categoria
  - Relatórios básicos, como **listagens filtradas por categoria** ou **produtos com stock insuficiente**.
- Utilizar **Spring Boot para o backend**, enquanto o frontend pode ser desenvolvido em qualquer ferramenta à escolha:
  - **HTML/CSS/JavaScript simples**
  - Frameworks como **React, Angular, Vue** ou **Vaadin Flow**.

## **Critérios de Avaliação:**

### **1. Nível Básico (Nota de Partida: 10 a 13 valores):**

- Seguir o vídeo fornecido para implementar um sistema funcional com Spring Boot e **Vaadin Flow**.
- Incluir as funcionalidades de autenticação (login simples) e gestão básica do inventário (adicionar, listar e editar itens).

### **2. Nível Intermédio (Nota de 14 a 16 valores, com muito apoio de vídeo):**

- Implementar autorizações baseadas em "roles", permitindo:
  - Acesso diferenciado para utilizadores regulares e administradores.
- Adicionar funcionalidades práticas, como:
  - **Filtros avançados** para listagens de inventário.
  - **Relatórios básicos** (ex.: produtos com stock baixo).
- Melhorar a interface para ser mais intuitiva e visualmente apelativa.

### **3. Nível Avançado:**

- Ver secção seguinte
  - Proposta de Desenvolvimento de Aplicação à Escolha

---

## Projeto Para os Mais Ambiciosos (Opcional, para notas 17-20):

### Introdução

Como o que estava previsto era a realização de um Inventário, com segurança, autenticação e autorização (diferentes atores poderem fazer coisas diferentes consoante os seus "roles"), é importante que mesmo quem opta por "brilhar" (tentar alcançar notas de 17 a 20) comece por fazer um inventário simples, como o Vaadin permite, e como eu explico no vídeo para levar as pessoas até ao nível Intermédio de avaliação (notas 14-16).

Assim, deve começar por fazer um inventário com backend em Spring Boot mas frontend na framework que preferir (recomendo Vaadin), até ao nível intermédio. Pode usar o Vaadin Start, e componentes pré-feitos (seguindo ver o vídeo que farei para ajudar os colegas que preferirem fazer o Inventário no Trabalho 2).

Mas depois faço esta sugestão de Trabalho, em substituição de trabalhar apenas no tema do Inventário:

### Proposta Livre:

- Elaborar um **relatório breve** justificando a relevância do projeto para o curso e o desafio pessoal envolvido.
- Desenvolver um sistema que vá além do escopo básico do inventário, integrando conceitos de **Engenharia de Software, Design de Interfaces e Segurança**.

## Exemplos de Projetos Possíveis (Mas pode Fazer Sobre Outro Tema):

### 1. Sistema de Gestão de Carteiras de Ativos Financeiros

- **Descrição:** Uma aplicação para registar e consultar uma carteira de investimentos.
- **Funcionalidades:**
  - Adicionar, editar e eliminar ativos financeiros (ações, criptomoedas, etc.).
  - Calcular automaticamente o valor total da carteira.
  - Apenas uma integração com uma API de preços (ex.: para consultar o preço de um ativo específico).
  - Interface simples com tabelas e formulários.

---

### 2. Aplicação de Monitorização Meteorológica

- **Descrição:** Uma aplicação que apresenta a previsão meteorológica de uma cidade selecionada.
- **Funcionalidades:**
  - Selecionar uma cidade e mostrar o estado do tempo atual (ex.: temperatura, humidade).
  - Integração com uma API de meteorologia (ex.: OpenWeatherMap).
  - Histórico simples com as últimas 5 consultas feitas pelo utilizador.
  - Interface com um campo de texto e uma área de resultados.

---

### 3. Plataforma de Informação sobre Aquecimento Global

- **Descrição:** Uma aplicação que apresenta um resumo simples sobre o impacto do aquecimento global.
  - **Funcionalidades:**
    - Apresentar um gráfico básico com dados de emissões de CO<sub>2</sub> de uma API externa (ex.: dados fixos ou mockados).
    - Secção de explicação com um texto estático sobre o tema.
    - Permitir aos utilizadores registar notas ou comentários pessoais sobre o tema.
    - Interface minimalista, com um gráfico e uma área de texto.
- 

### 4. Sistema de Gestão de Consultas Médicas

- **Descrição:** Um sistema básico para registar médicos, pacientes e consultas.
  - **Funcionalidades:**
    - Criar, visualizar e editar consultas (médico, paciente, data).
    - Registar perfis simples de médicos e pacientes.
    - Permitir consultas apenas aos utilizadores autenticados.
    - Interface simples com tabelas para listar consultas.
- 

### 5. Sistema de Gestão de Biblioteca

- **Descrição:** Uma aplicação para registar livros e gerir empréstimos básicos.
  - **Funcionalidades:**
    - Adicionar, editar e eliminar livros.
    - Registar quem emprestou um livro (apenas nome do utilizador e data).
    - Listar os livros disponíveis e os emprestados.
    - Não requer funcionalidades avançadas como notificações.
- 

### 6. Sistema de Reserva de Viagens e Turismo

- **Descrição:** Uma aplicação para criar e listar pacotes turísticos.
  - **Funcionalidades:**
    - Adicionar e visualizar pacotes turísticos (nome, destino, preço, descrição).
    - Permitir que os utilizadores registem um interesse num pacote.
    - Sistema de autenticação básico para separar administradores de utilizadores comuns.
    - Interface com tabelas para pacotes e reservas.
- 

### 7. Sistema de Gestão de Eventos

- **Descrição:** Uma aplicação para gerir eventos de forma básica.
- **Funcionalidades:**
  - Criar e listar eventos (nome, local, data e descrição).
  - Registar participantes para eventos.
  - Diferenciar utilizadores: organizadores podem criar eventos, participantes só se podem registar.
  - Interface com tabelas simples e formulários.

---

## 8. Plataforma de Gestão de Projetos com Kanban

- **Descrição:** Uma aplicação para gerir tarefas de um projeto com o estilo Kanban.
  - **Funcionalidades:**
    - Criar tarefas com nome, descrição e estado (pendente, em progresso, concluída).
    - Alterar o estado das tarefas manualmente.
    - Listar tarefas em diferentes colunas, conforme o estado.
    - Não requer integrações ou notificações automáticas.
- 

## 9. Sistema de Gestão de Clubes Desportivos

- **Descrição:** Uma aplicação para registar atletas e equipas.
  - **Funcionalidades:**
    - Adicionar atletas e atribuí-los a equipas.
    - Listar atletas e as respetivas equipas.
    - Criar jogos e associar equipas a cada jogo.
    - Não requer cálculos de estatísticas ou permissões avançadas.
- 

## 10. Plataforma Educativa Personalizada

- **Descrição:** Uma aplicação para gerir cursos e alunos.
  - **Funcionalidades:**
    - Criar cursos com nome e descrição.
    - Associar alunos a cursos.
    - Listar os cursos de cada aluno.
    - Não requer avaliações automáticas ou relatórios avançados.
- 

## 11. ... o que a sua imaginação quiser, e for compatível com os objetivos do curso.

---