

## Trabalho 2: Inventário Completo (para notas 10-16)

### **e** Aplicação Opcional (para notas de 17-20)

Prazo: **9 de fevereiro** (sugestão: não deixe de ver **apoio no YouTube**)

#### 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:
  - **Vaadin Flow (muito recomendado)**, pelo foco dado no curso. As instruções serão dadas para esta framework. Se fizer com outras, deve sentir-se capaz de trabalhar de forma independente, com base nos seus conhecimentos e pesquisas).
  - Frameworks como **React, Angular, Vue**.
  - **HTML/CSS/JavaScript** simples (não muito recomendado, porque estará a ter muito mais trabalho e preocupação com pormenores que as Frameworks referidas anteriormente tentam poupar aos programadores).

## **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).
- Nota: iremos usar o Vaadin Start, e componentes pré-concebidos, pelo que o processo será extremamente simples e visual.
- Seguir os passos de um vídeo de menos de uns 2 minutos permite atingir o objetivo

### **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.
- Nota: uma vez mais, iremos usar, em boa parte, mesmo no NÍvel Intermédio, o Vaadin Start, e o processo será praticamente fazer configuração visual (também aqui, um vídeo de 2 a 3 minutos deve ser suficiente para ver como tudo o que aprendeu no curso lhe permite perceber o que está “subjacente” a esta forma rápida de fazer aplicações para a Web modernas, elegantes, seguras e robustas)

### **3. NÍvel Avançado (17+ valores):**

- Ver secção seguinte
  - Desenvolvimento Livre de Aplicação à EscolhaAqui não haverá vídeo; o trabalho será guiado pelo Manual, exemplos no GitHub, Referências Bibliográficas e documentação oficial das frameworks.

---

## 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).

Depois, desenvolva uma aplicação, usando obrigatoriamente como backend Spring Boot. Para o frontend, pode escolher a framework que preferir, mas recomenda-se muito o uso de Vaadin Flow, que foi o foco do curso, e permite fazer todo o projeto em Java, a menos que esteja bem familiarizado/a com outras tecnologias / frameworks. Mas a sua nota não será desvalorizada se usar outra tecnologia de frontend.

## 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.

---