Atividade Avaliativa - Lista de Exercícios

**Curso:** Pós em Web, Mobile e Embarcados

**Componente Curricular:** Desenvolvimento de Aplicações Web com Java

**Forma de entrega:** Moodle

**Regras:**

* Esta atividade deve ser realizada **individualmente**.
* Não é permitido nenhum tipo de plágio (cópia de colegas ou outras fontes).
* Não haverá prorrogação da data limite para entrega desta atividade.
* Para a entrega, criar um repositório público no github ou bitbucket e enviar o link pelo moodle.
* Incluir no repositório um arquivo LEIAME contendo instruções sobre a organização dos exercícios no repositório.
* Os exercícios teóricos podem ser respondidos em um único arquivo de texto.

### Exercícios Teóricos

**ET1)** Descreva com suas palavras o funcionamento dos protocolos HTTP e HTTPS. Explique quais são as principais diferenças entre os dois protocolos.

**HTTP: nesse protocolo o cliente solicita uma informação para o servidor, o qual apenas retorna a informação, caso exista. Por exemplo ao acessar um site, o cliente informa o endereço desejado ao servidor, e o mesmo retorna a pagina estática, caso exista.**

**HTTPS: funciona da mesma forma que o HTTP, porém, possui um nível de segurança maior, baseado nos protocolos SSL e TLS que criptografam as informações, exigindo um certificado digital para que se possa acessá-las.**

**ET2)** Descreva com suas palavras o que é *Representational State Transfer* (REST).

**Rest é um modelo de arquitetura de informação, de como o sistema é construído para fornecer informações para qualquer que seja a integração necessária. Constituído de regras que o tornam simples, leve e ao mesmo tempo bem estruturado.**

**ET3)** Escolha um dos ataques catalogados pela OWASP Foundation (<https://owasp.org/www-community/attacks/>). Pesquise sobre o ataque escolhido e sobre quais medidas devem ser tomadas para proteger um sistema de tal ataque.

**Brute Force Attack.**

**Como o próprio nome diz é um ataque de força bruta, onde o invasor utiliza de ferramentas para realizar solicitações a um servidor, testando com uma quantidade gigantesca de tipos de usuários e senhas, normalmente disponíveis em um dicionário criado pelo mesmo, analisando assim as respostas do servidor.**

**Existem algumas formas de bloqueio deste ataque que seriam a utilização de senhas fortes, limitação do numero de logins, autenticação em dois fatores, login multinível e colocação de lista negra onde apenas determinados IPs teriam acesso.**

### Exercícios Práticos

**EP1)** Crie uma aplicação Spring Boot que seja capaz de atender requisições HTTP, do tipo GET, no caminho “/contador”. A resposta para este caminho deve ser uma página HTML contendo a quantidade de vezes que o caminho foi acessado (i.e., “Número de Acessos: XX”). O contador não precisa ser persistido, basta que seja armazenado em memória.

**EP2)** Considerando as entidades *Pessoa, Produto, Pedido* e *PedidoItem* apresentadas abaixo, realiza a implementação de uma aplicação Spring Boot, contendo as seguintes funcionalidades: (1) operações CRUD para Pessoa e Produto, e (2) movimento de realizar pedido. Deve-se realizar a persistência de dados com Spring Data. Além disso, deve-se prover acesso às funcionalidades do sistema por meio de uma interface web (e.g., usando o Thymeleaf). O valor do produto está em Float, mas pode-se usar outra forma mais adequada para valores monetários, tal como o uso de BigDecimal.

