

Trabalho Prático 2: Desenvolvimento Seguro de Tela de Login e Registro

Objetivo

Desenvolver uma tela de login e registro aplicando conceitos básicos de segurança da informação. O aluno pode utilizar qualquer tecnologia (PHP, Python, Node.js, Java, C#, etc.) ou framework (React, Angular, Vue, Django, Flask, Laravel, etc.), desde que atenda aos requisitos de segurança propostos.

Enunciado

Você foi contratado para implementar um sistema de autenticação (login e registro) para uma aplicação web. Além da funcionalidade básica, **é essencial que a solução siga boas práticas de segurança.**

Requisitos Funcionais



Tela de Registro:

- Campos: Nome, E-mail, Senha, Confirmação de Senha.
- Validação de e-mail (formato válido).
- Comparação entre senha e confirmação.



Tela de Login:

- Campos: E-mail e Senha.
- Feedback claro em caso de erro (sem revelar detalhes técnicos).



Funcionalidades Extras (Opcionais):

- Recuperação de senha (envio de e-mail com token temporário).
 - Verificação em duas etapas (2FA).
-

Entrega



O que deve ser entregue?

1. Código Fonte (pode ser um repositório Git ou arquivo compactado).
2. Relatório Técnico (PDF ou documento) contendo:
 - Tecnologia/Framework utilizado.
 - Explicação de como cada requisito de segurança foi implementado.
 - Dificuldades encontradas e como foram resolvidas.



Prazo: 25/04/2025

Forma de envio: via e-mail professorjdefora@gmail.com com as seguintes informações:

ASSUNTO: Trabalho 1

CONTEÚDO: a escolha do sistema e os nomes completos de até 5 integrantes do grupo

ANEXO: qualquer formato desde que acessível sem necessidade de confirmação.

Critérios de Avaliação

Critério	Pontuação
-----	-----
Funcionalidade (Login/Registro)	30%
Implementação de Segurança	50%
Organização do Código	10%
Relatório Técnico	10%

Dicas Extras

◆ Ferramentas Úteis:

- OWASP Cheat Sheet (guia de segurança).
- Postman/Insomnia (testar APIs).
- Burp Suite (testar vulnerabilidades).

◆ Exemplos de Implementação Segura:

- Node.js: Usar ``bcrypt`` para senhas, ``helmet`` para headers.
- PHP: Usar ``password_hash()``, ``PDO`` para SQL.
- Python (Flask/Django): Usar ``Werkzeug.security``, ``flask-talisman``.

Bom trabalho! 🚀🔒