PROJETO 03 – TESTES AUTOMATIZADOS, AUTENTICAÇÃO E AUTORIZAÇÃO
PROFs. Guilherme e Victor

EDITAL DO ENTREGÁVEL FINAL

SISTEMA DO PROJETO INTEGRADOR E MENSAL

ENTREGA SEXTA FEIRA (28/11)

---

1) REQUISITOS TÉCNICOS OBRIGATÓRIOS

Para este entregável, o foco será na implementação de ponta a ponta de segurança e no deploy da aplicação em um ambiente de nuvem (AWS).

1. Segurança de Back-End (Spring Security):

- **Criptografia:** As senhas dos usuários no banco de dados devem ser criptografadas (ex: BCrypt ou similar).
- **Gerenciamento de Chaves:** O segredo do JWT (secret) deve estar em um arquivo de configuração (ex: .env ou application.properties), não "hardcoded" no código-fonte.
- **Autenticação:** Deve existir um endpoint de login (ex: /login) que valide as credenciais e retorne um token JWT válido.
- **Proteção de Rotas:** O Spring Security deve estar configurado para bloquear o acesso a rotas protegidas se um token JWT válido não for fornecido no header Authorization.
- **Autorização (Roles):** Rotas que exigem perfis específicos (ex: ROLE_ADMIN) devem estar bloqueadas para usuários sem a permissão adequada.

2. Segurança de Front-End (Angular):

- **Route Guards:** O Angular deve utilizar CanActivate (Guardas de Rota) para impedir que um usuário não autenticado acesse URLs de páginas internas colando o link diretamente no navegador.
- **Token Handling:** O Angular deve utilizar um HttpInterceptor para anexar automaticamente o token JWT (ex: Bearer ...) em todas as requisições para a API protegida.

3. Deployment (AWS):

- **Aplicação na Nuvem:** A aplicação (front-end e back-end) deve estar funcional e acessível em um servidor na nuvem (AWS), conforme demonstrado nos vídeos de referência.

1. REQUISITOS QUE SERÃO AVALIADOS PELO PROFESSOR

---

2) REQUISITOS QUE SERÃO AVALIADOS PELO PROFESSOR:

PROJETO MENSAL 04 - - ENTREGA FINAL (entrega 28/11 (SEXTA) - apres. 03 e	
CRITÉRIO	**PESO**
Participação	20%
As senhas no banco de dados devem estar criptografadas usando BCrypt (ou similar).	10%
O segredo do JWT deve estar em um arquivo .env	5%
O endpoint de login funciona, valida as credenciais e retorna um token JWT válido	15%
O Spring Security está configurado para bloquear automaticamente o acesso a rotas protegidas se um token JWT válido não for fornecido no header	15%
Rotas que exigem perfis específicos (ex: ROLE_ADMIN) também estão bloqueadas para usuários sem a permissão adequada.	5%
O Angular utiliza CanActivate para impedir que um usuário não autenticado acesse URLs de páginas internas colando o link diretamente no navegador.	10%
Utiliza um HttpInterceptor para anexar automaticamente o token JWT (ex: Bearer ...) em todas as requisições para a API protegida.	10%
Aplicação Funcionando na NUVEM	10%
TOTAL	**100%**

** lembrando*

Abaixo está o detalhamento dos critérios que compõem a ficha de avaliação:

- **As senhas no banco de dados devem estar criptografadas:** Avalia se as senhas dos usuários estão armazenadas no banco de dados usando um hash forte (BCrypt), e não em texto plano.
- **O segredo do JWT deve estar em um arquivo .env:** Verifica se chaves sensíveis (como o segredo do JWT) estão externalizadas em arquivos de configuração e não escritas diretamente no código ("hardcoded").
- **O endpoint de login funciona...:** Confirma se a API possui um endpoint de autenticação que recebe credenciais, valida-as corretamente e retorna um token JWT válido em caso de sucesso.
- **O Spring Security está configurado para bloquear...:** Analisa se os filtros do Spring Security estão corretamente configurados para interceptar requisições, validar o token JWT e bloquear o acesso a rotas protegidas.
- **Rotas que exigem perfis específicos...:** Verifica a implementação da autorização baseada em papéis (Roles), garantindo que um usuário (ex: ROLE_USER) não possa acessar rotas de ROLE_ADMIN.
- **O Angular utiliza CanActivate...:** Confirma se o front-end possui guardas de rota (CanActivate) que impedem o acesso a páginas protegidas colando a URL diretamente no navegador.
- **Utiliza um HttpInterceptor...:** Avalia se um interceptor HTTP foi implementado para anexar o token Bearer automaticamente em todas as chamadas para a API.

Aplicação funcionando NA NUVEM: Valida o sucesso do processo de deploy, verificando se a aplicação está totalmente funcional, integrada (front e back) e acessível publicamente na AWS.

---

3) FORMA DE ENTREGA E PRAZOS:

ENTREGA:

- **Data**: **28/11/2025**. (SEXTA-FEIRA)
- Repositório Git: Apenas um do grupo envia a URL do repositório por meio da BlackBoard, desde que este liste os nomes de todos os membros do grupo.
- Arquivo zip do Projeto
- Mesmo grupo do PI.

APRESENTAÇÕES:

- **Data**: **03/12 e 04/12/2025**
- Mostrar e explicar a solução.
- Todos os membros devem falar/explicar algo (a avaliação é individual).
- Tempo médio de apresentação: entre 10 e 15 minutos.

5) ORIENTAÇÕES:

- **Dias de apresentações não são dias para tirar dúvidas ou corrigir bugs**! Portanto, iniciem desde já o desenvolvimento e tirem suas dúvidas até quinta-feira (27/11).