**1. Implementação**

**Projeto**: Geartech Project

**GitHub:**

* **Backend:** <https://github.com/geartech/geartech-back>
* **Frontend**: <https://github.com/geartech/geartech-front>

**Atividade**:

Implementar um cadastro de usuários **CRUD** (criar, ler, atualizar).

**3. Requisitos Funcionais**

* RF1: **Listar usuários** com paginação, ordenação e filtros (placa, modelo, ano, status).
* RF2: **Criar** usuário com validações de campos obrigatórios como login, nome, email e senha.
* RF3: **Editar** dados do uauário preservando histórico de alterações.
* RF4: **Nunca** **Excluir** usuário ele deve ser somente ativado/desativado.
* RF5: **Visualizar detalhe** do usuário.
* RF6: **Exportar** resultados da listagem em Excel e CSV.
* RF7: **Anexos** opcionais (foto) até 1 MB por arquivo.

**4. Regras de Negócio**

* RN1: **Login, email** devem ser únicos na base de dados.
* RN2: Usuário deve ter status de **ativo, expiração de senha e conta**.
* RN4: A **senha** deve ser revalidada e obrigatoriamente senha forte
* RN5: Registros e atualizações devem ser aarmazenados em tabela de auditoria com a informação de quem e quando.

**5. Campos (mínimos)**

* login, nome, email, telefone, senha, data Expiracao da senha, e status de ativação.

**7. API (REST – exemplo)**

* GET /api/usuarios (consulta com ordenação e filtros)
* POST /api/usuarios
* GET /api/usuarios/{id}
* PUT /api/usuarios/{id}
* DELETE /api/usuarios/{id}
* GET /api/usuarios/export.csv
* GET /api/usuarios/export.xlsx

**8. Validações**

* Email, telefone: regex do padrão pt-br.
* Campos obrigatórios login, nome, email, status.

**12. Integrações (opcional)**

* N/A

**14. Critérios de Aceite**

* CA1: CRUD completo funcionando com as regras de negócio definidas.
* CA2: Filtros por placa, status e ano; exporta Excel do resultado.
* CA4: Testes: unitários ≥ 70% nas regras.

**Observações de implementação (opinião)**

* N/A