

Documento de Requisitos — Sistema de Reserva de Laboratórios de Informática

1) Escopo do MVP

Sistema web básico multiplataforma, desenvolvido em React JS (frontend) e Node.js com MongoDB (backend), para cadastro e gerenciamento de horários de aulas em laboratórios de informática.

Cadastros (CRUD)

- Instituição: nome, sigla, CNPJ (opcional), endereço (opcional), status.
- Curso: instituição, nome, turno(s), status.
- Professor: nome, e-mail institucional, telefone (opcional), status.
- Disciplina: curso, nome, carga horária, professor responsável (opcional), status.
- Laboratório: nome/identificador, capacidade, localização (opcional), status.
- Blocos de horário: turno, dia da semana, início, fim, ordem.

Horário de Aulas

Permite criar alocações de disciplinas em laboratórios conforme dia da semana e bloco de horário. O sistema impede conflitos de horários de laboratório e professor.

Consultas

Permite visualizar horários por laboratório, curso, disciplina ou professor.

2) Modelo de Dados (MongoDB — esboço)

```
instituicoes: { nome, sigla, cnpj, endereco, status }  
cursos: { instituicaoId, nome, turnos, status }  
professores: { nome, email, telefone, status }  
disciplinas: { cursoId, nome, cargaHoraria, professorId, status }  
laboratorios: { nome, capacidade, local, status }  
blocos: { turno, diaSemana, inicio, fim, ordem }  
aulas: { semestre, cursoId, disciplinaId, professorId, laboratorioId, diaSemana, blocos, dataInicio, dataFim }
```

3) API (REST, Node.js/Express)

Cadastros:

- /api/instituicoes
- /api/cursos
- /api/professores

- /api/disciplinas
- /api/laboratorios
- /api/blocos

Horários (Aulas):

- POST /api/aulas (criar alocação)
- GET /api/aulas (consultar)
- PUT /api/aulas/:id (editar)
- DELETE /api/aulas/:id (remover)

4) Telas (React JS)

- Login (opcional).
- Dashboard com atalhos.
- CRUDs de cadastros.
- Formulário de criação de horário de aula.
- Consulta de horários por filtros e grade semanal.

5) Requisitos Funcionais (RF)

RF01: CRUD de Instituição, Curso, Professor, Disciplina, Laboratório e Blocos.

RF02: Cadastro de Aulas com validação de conflito.

RF03: Consulta de horários.

RF04: Edição e remoção de alocações.

RF05: Upload/seed de blocos por turno (opcional).

6) Requisitos Não Funcionais (RNF)

RNF01: Stack — React, Node.js (Express), MongoDB.

RNF02: Performance — resposta média < 500 ms.

RNF03: Qualidade — validações no backend.

RNF04: Segurança — JWT simples.

RNF05: UX — responsivo, teclado navegável.

7) Critérios de Aceite (CA)

CA01: Criação de aula em laboratório ocupado deve falhar.

CA02: Criação de aula com conflito de professor deve falhar.

CA03: Consulta por laboratório exibe todos os blocos alocados.

CA04: CRUDs salvam e exibem dados corretamente.