

# ReservaLab

Michael Pinho, João Felipe Guedes, João Vitor Fahning, Tiago Trindade, Alisson Gabriel

[Link para o Repositório Github](#)

## Histórico de Revisões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor(es)</b>
14/08/2025	0.01	Populando o documento pela primeira vez, montando as seções a partir do Template: Introdução, Descrição, Glossário, Análise de Requisitos - versões básicas	Tiago Trindade João Felipe Guedes
14/08/2025	0.01	Criação do sumário após definição do Template Inicial	João Vitor Fahning
15/08/2025	0.01	Elicitação de Requisitos	Michael Pinho
16/08/2025	0.01	Aprimoradas a clareza e a padronização das seções da Parte 1 (Introdução, Descrição, Glossário e Análise de Requisitos), com ajustes no texto e maior detalhamento das informações	Alisson Gabriel
03/09/2025	0.2	Pontos 6 e 7, dado a tarefa Projeto 2.	Tiago Trindade
18/09/2025	0.3	Laboratório Modelo 2 - DAO	João Vitor Fahning

# Sumário

## 1. Introdução

### 1.1. Propósito do documento

### 1.2. Visão geral do documento

### 1.3. Documentos relacionados

## 2. Descrição Geral

### 2.1. Motivação

### 2.2. Problemas identificados

### 2.3. Visão geral do sistema

### 2.4. Usuários do sistema

### 2.5. Suposições e restrições gerais

## 3. Glossário

## 4. Elicitação de Requisitos

### 4.1. Técnicas Utilizadas

### 4.2. Conclusão da Elicitação

## 5. Análise de Requisitos

### 5.1. Requisitos Funcionais

### 5.2. Requisitos Não Funcionais

# 1. Introdução

Este documento apresenta a especificação de requisitos do sistema **ReservaLab**, uma aplicação de linha de comando (CLI) desenvolvida para o Centro de Informática da Universidade Federal da Paraíba (CI/UFPB). O documento é produzido como parte das atividades da disciplina Métodos de Projeto de Software (período 2025.1).

O objetivo é formalizar o processo de reserva de laboratórios e salas, garantindo organização, transparência e uso eficiente dos recursos. Ele também serve como instrumento de comunicação entre a equipe de desenvolvimento, os professores e os alunos que utilizarão a aplicação, registrando de forma clara as decisões e expectativas.

## 1.1 Propósito do documento

O propósito deste documento é fornecer uma visão clara e estruturada dos requisitos do ReservaLab, servindo como referência para desenvolvedores, professores, alunos e demais partes interessadas. Ele organiza os objetivos, funcionalidades esperadas, restrições e padrões de qualidade que devem ser atendidos pelo sistema.

Além disso, este documento será utilizado para alinhar expectativas entre a equipe e os usuários finais, evitando ambiguidades e servindo como registro oficial das decisões tomadas durante a fase inicial do projeto.

## 1.2 Visão geral do documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto, além de conter uma lista de referências para outros documentos relacionados. As demais seções apresentam a especificação do sistema ReservaLab.

Este documento está organizado da seguinte forma:

- Elementos preliminares: capa, histórico de revisões e sumário, que identificam o projeto, registram suas versões e fornecem uma visão geral da estrutura.
- Seção 1 – Introdução: apresenta os objetivos do documento, suas convenções, escopo de utilização e documentos relacionados.
- Seção 2 – Descrição Geral: apresenta os objetivos do sistema, motivação, problemas identificados, visão geral, perfis de usuários e restrições.
- Seção 3 – Glossário: define os principais termos técnicos e específicos utilizados no contexto do projeto.
- Seção 4 – Elicitação de Requisitos: descreve as técnicas utilizadas para levantar informações com os stakeholders e apresenta os resultados obtidos.
- Seção 5 – Análise de Requisitos: detalha os requisitos funcionais e não funcionais que guiarão o desenvolvimento da aplicação.

## 1.3 Documentos relacionados

Os documentos e referências utilizados na elaboração deste material incluem:

- Documento base da disciplina: Modelo de Documento de Requisitos (fornecido em sala).
- Glossário do RUP:  
<http://josepaulopapo.blogspot.com.br/2007/08/o-glossario-do-rup-este-ser.html>.
- Especificação inicial do projeto ReservaLab (descrição e requisitos discutidos pelo grupo).

## 2. Descrição Geral

O sistema ReservaLab tem como objetivo centralizar e formalizar o processo de reserva de salas e laboratórios no Centro de Informática da UFPB. Atualmente, esse processo é pouco estruturado, o que gera conflitos de horários, falta de transparência e dificuldades para professores e alunos acessarem os espaços.

A aplicação será uma ferramenta de linha de comando (CLI) que combina simplicidade, rapidez e baixo consumo de recursos, garantindo que qualquer usuário — mesmo em máquinas mais limitadas — possa utilizar o sistema de forma eficiente.

Com o ReservaLab, espera-se melhorar a organização do uso dos recursos físicos do CI, facilitar o agendamento de aulas e projetos, evitar conflitos de reservas e oferecer relatórios úteis para a gestão acadêmica.

### 2.1. Motivação

A motivação para o desenvolvimento do ReservaLab vem da necessidade de:

- Tornar o processo de reserva de salas e laboratórios mais transparente e confiável.
- Reduzir conflitos de agendamento que prejudicam aulas e atividades de pesquisa.
- Facilitar a vida de professores e alunos, oferecendo uma ferramenta simples e acessível.
- Permitir à administração do CI gerenciar melhor os recursos, com relatórios que apoiem decisões.

### 2.2. Problemas identificados

Foram observados os seguintes problemas no processo atual de reservas:

- Ausência de um sistema formal para registro de solicitações.
- Conflitos de horários por comunicação falha ou registros duplicados.
- Dificuldade de professores e alunos em verificar a disponibilidade das salas em tempo real.

- Falta de relatórios para a coordenação acompanhar a taxa de ocupação dos laboratórios.
- Dependência de processos manuais ou informais (como planilhas ou mensagens avulsas).

## 2.3. Visão geral do sistema

O ReservaLab permitirá:

- Autenticação de usuários (login e senha).
- Cadastro e gerenciamento de salas (nome, capacidade, recursos).
- Agendamento e cancelamento de reservas por professores (e eventualmente alunos).
- Visualização de disponibilidade de salas (aberta inclusive para usuários não autenticados).
- Relatórios gerenciais para administradores.

Escopo negativo (o que não fará):

- O sistema não terá interface gráfica (web ou desktop), sendo restrito à CLI.
- Não fará integração inicial com sistemas externos de gestão acadêmica (SIGAA, por exemplo).
- Não oferecerá notificações por e-mail ou SMS nesta primeira versão (podendo ser previsto para futuro).

Independência:

O sistema é auto-contido, não dependendo de outros softwares institucionais para funcionar. Futuras integrações poderão ser adicionadas, mas não fazem parte do escopo atual.

## 2.4. Usuários do sistema

O sistema será utilizado por três perfis principais:

- Administrador: responsável por cadastrar, editar e remover salas, gerenciar usuários e visualizar/cancelar qualquer reserva.
- Professor: poderá reservar salas para aulas, projetos de pesquisa e extensão, além de consultar a disponibilidade e suas próprias reservas.
- Aluno: terá acesso à visualização de disponibilidade e, sob certas regras (a definir), poderá solicitar reservas para estudo ou projetos acadêmicos.

Esses usuários representam diferentes níveis de permissão, de modo que a interface de comandos apresentada será adaptada ao perfil.

## 2.5. Suposições e restrições gerais

- O sistema será desenvolvido inicialmente como uma aplicação de linha de comando.
- Deve ser compatível com Windows, Linux e macOS.

- O acesso dependerá de autenticação (login e senha).
- Os dados serão armazenados em banco de dados seguro (com senhas criptografadas).
- Por se tratar de um projeto acadêmico, o sistema pode ser expandido posteriormente para outras interfaces (web ou mobile).
- Deve responder rapidamente (em até 2 segundos) aos comandos.

### 3. Glossário

Este glossário reúne os termos técnicos e específicos do sistema ReservaLab, servindo como referência para todos os envolvidos no projeto.

- Reserva: Ação de agendar uma sala ou laboratório em uma data e horário específicos.
- Sala/Laboratório: Espaço físico do Centro de Informática (CI/UFPB) que pode ser reservado para aulas, pesquisas, projetos ou estudos.
- Usuário: Pessoa que interage com o sistema. Pode ser administrador, professor ou aluno.
- Administrador: Usuário com privilégios máximos, responsável por gerenciar salas, usuários e reservas.
- Professor: Usuário que pode solicitar e gerenciar suas próprias reservas, além de consultar disponibilidade.
- Aluno: Usuário com acesso à consulta de disponibilidade e, em algumas situações, autorização para solicitar reservas.
- Disponibilidade: Estado de uma sala/laboratório em determinado dia e horário, indicando se está livre ou ocupada.
- Agendamento: Registro no sistema que confirma a reserva de uma sala/laboratório.
- Cancelamento: Ação de remover uma reserva previamente registrada.
- Conflito de agendamento: Situação em que dois usuários tentam reservar o mesmo espaço para o mesmo horário.
- CLI (Command Line Interface): Interface de linha de comando, por meio da qual os usuários interagem com o sistema usando instruções digitadas no terminal.
- Relatório: Documento ou saída gerada pelo sistema, apresentando dados de uso (ex.: taxa de ocupação das salas).
- Login: Processo de autenticação do usuário no sistema, utilizando credenciais (nome de usuário e senha).
- Permissões: Conjunto de regras que definem quais ações cada perfil de usuário pode executar no sistema.
- CI (Centro de Informática): Unidade acadêmica da Universidade Federal da Paraíba (UFPB) onde o sistema será implantado para gerenciar reservas de espaços.
- UFPB (Universidade Federal da Paraíba): Instituição de ensino superior pública, localizada na Paraíba, responsável pelo ambiente acadêmico em que o sistema será utilizado.

## 4. Elicitação de Requisitos

### 4.1. Técnicas Utilizadas

#### **Técnica 1:** Entrevistas com Stakeholders

Entrevistas realizadas com alunos e professores do CI/UFPB (usuários que precisam reservar salas), e coordenadores/administradores do CI (gerenciadores dos espaços).

Objetivo:

- Levantar os problemas atuais no processo de reservas e expectativas em relação ao sistema (o que devemos esperar dele).

Resultados da entrevista:

- Professores relataram dificuldade em verificar a disponibilidade em tempo real.
- Administradores apontaram problemas de comunicação e conflitos de agendamento.
- Usuários querem simplicidade (desejam usar comandos rápidos e realizar reservas rapidamente).
- A necessidade de relatórios foi considerada importante, mas não essencial na primeira versão (desejável para versões futuras).

#### **Técnica 2:** Questionário Online com Alunos e Professores

Perguntas simples (Google Forms) enviadas para alunos e professores. Público: 15 respondentes (10 alunos, 5 professores).

Principais resultados (conclusões):

- 80% dos respondentes consideram “visualizar disponibilidade antes de reservar” como funcionalidade essencial.
- 65% apontaram que o maior problema atual é o conflito de horários.
- 70% dos professores desejam relatórios de uso (mas aceitam que fiquem para versões futuras).
- 90% acham aceitável que a aplicação seja CLI, desde que exista uma ajuda simples (`--help`).



## 4.2 Conclusão da Elicitação

A partir das duas técnicas (entrevistas e questionários), foi possível identificar que:

- O núcleo do sistema deve ser a reserva de salas com prevenção de conflitos (foi tomado como base para a escolha dos requisitos funcionais mais importantes).
- Autenticação e perfis de usuário são necessários para dar legitimidade às reservas e segurança ao sistema.
- Consulta pública de disponibilidade é indispensável para evitar sobreposição de horários.
- Funcionalidades adicionais (relatórios, manual, etc.) são importantes, mas podem ser entregues em versões futuras.

## 5. Análise de Requisitos

### 5.1. Requisitos Funcionais

#### [RF001] Autenticação de Usuário

- Descrição: O sistema deve permitir que os usuários se autenticuem através de login e senha.
- Prioridade: Essencial

#### [RF002] Gerenciamento de Usuários

- Descrição: O administrador deve poder listar, editar e remover usuários existentes.
- Prioridade: Importante

#### [RF003] Cadastro e Gerenciamento de Salas

- Descrição: O administrador deve poder cadastrar novas salas, informando nome, capacidade e recursos disponíveis. Ele deve também poder editar e remover salas já cadastradas.
- Prioridade: Essencial

#### [RF004] Agendamento de Sala

- Descrição: Usuários autenticados (professores e, sob regras específicas, alunos) devem poder reservar salas em uma data e horário.
- Prioridade: Essencial

#### [RF005] Cancelamento de Agendamento

- Descrição: Usuários devem poder cancelar reservas próprias; administradores podem cancelar qualquer reserva.
- Prioridade: Essencial

#### [RF006] Visualização de Disponibilidade de Salas

- Descrição: Usuários (mesmo sem login) devem poder visualizar a disponibilidade de salas e laboratórios.
- Prioridade: Essencial

#### [RF007] Controle de Conflitos de Reserva

- Descrição: O sistema deve impedir reservas conflitantes e notificar o usuário.
- Prioridade: Essencial

#### [RF008] Geração de Relatórios

- Descrição: O administrador deve poder gerar relatórios de utilização das salas.
  - Prioridade: Desejável
- 

## 5.2. Requisitos Não Funcionais

### Usabilidade

#### [RNF001] Interface Clara e Intuitiva

- Descrição: A CLI deve oferecer comandos fáceis de lembrar, com mensagens claras e opção de ajuda (*reservaLab --help*).
- Prioridade: Essencial

#### [RNF002] Manual do Usuário

- Descrição: Deve existir um guia básico de comandos para orientar novos usuários.
- Prioridade: Desejável

### Desempenho

#### [RNF003] Tempo de Resposta

- Descrição: O sistema deve responder aos comandos em até 2 segundos.
- Prioridade: Essencial

### Segurança

#### [RNF004] Armazenamento Seguro de Senhas

- Descrição: Senhas devem ser armazenadas de maneira segura.
- Prioridade: Essencial

## **Confiabilidade**

### **[RNF005] Alta Disponibilidade**

- Descrição: O sistema deve estar disponível pelo menos 95% do tempo durante o período letivo.
- Prioridade: Essencial

### **[RNF006] Cópias de Segurança**

- Descrição: O sistema deve manter backups regulares do banco de dados de reservas.
- Prioridade: Importante

## **Portabilidade**

### **[RNF007] Compatibilidade Multiplataforma**

- Descrição: O sistema deve funcionar em Windows, Linux e macOS.
- Prioridade: Essencial

## **Manutenibilidade**

### **[RNF008] Código-fonte Estruturado**

- Descrição: O código deve ser bem documentado e modular, facilitando manutenção futura.
- Prioridade: Importante

## **Escalabilidade**

### **[RNF009] Suporte à Expansão Futura**

- Descrição: A arquitetura deve permitir a adição de novas funcionalidades sem perda significativa de desempenho.
- Prioridade: Desejável

## **6. Especificação de Requisitos**

### **[UC001] Autenticação de Usuário**

**Ator:** Usuário (administrador, professor, aluno)

#### **Entradas e Pré-condições:**

- Usuário deve informar login e senha.
- Credenciais devem existir no sistema.

#### **Saídas e Pós-condições:**

- Usuário autenticado tem acesso às funcionalidades permitidas por seu perfil.
- Em caso de falha, exibe mensagem de erro.

**Fluxo Principal:**

1. Usuário executa comando *login*.
2. Sistema solicita credenciais.
3. Usuário informa login e senha.
4. Sistema valida dados no banco.
5. Sistema libera acesso.

**Fluxos Alternativos:**

- Senha incorreta → Mensagem de erro.
- Usuário inexistente → Registro não encontrado.

**[UC002] Agendamento de Sala**

**Ator:** Professor e Aluno

**Prioridade:** Essencial

**Entradas e Pré-condições:**

1. Usuário deve estar autenticado.
2. Deve existir sala cadastrada.
3. Data e horário livres.

**Saídas e Pós-condições:**

4. Reserva criada com sucesso.
5. Sala marcada como ocupada no período.

**Fluxo Principal:**

1. Usuário executa comando *reservar*.
2. Sistema solicita: sala, data e horário.
3. Usuário informa os dados.
4. Sistema valida disponibilidade.
5. A reserva é registrada.
6. A mensagem de confirmação é exibida.

**Fluxos Alternativos:**

- Sala indisponível → Mensagem de conflito.
- Dados inválidos → Solicitação rejeitada.

## [UC003] Visualização de Disponibilidade de Salas

**Ator:** Qualquer usuário, mesmo não autenticado

**Prioridade:** Essencial

### **Entradas e Pré-condições:**

- Usuário solicita disponibilidade para data/horário.

### **Saídas e Pós-condições:**

- O sistema apresenta as salas livres/ocupadas no período.

### **Fluxo Principal:**

- Usuário executa comando *listar-salas* ou *disponibilidade*.
- Sistema solicita data/horário (opcional).
- Usuário informa parâmetros.
- Sistema consulta reservas existentes.
- Resultado é exibido.

### **Fluxos Alternativos:**

- Data inválida → Solicita novamente.
- Nenhuma sala encontrada → Mensagem informativa.

## 7. Análise de Casos de Uso

### 7.1. Diagrama de Casos de Uso



## 7.1. Diagrama de Classes de Análise



