

UNIVERSIDADE SÃO JUDAS TADEU

Gabriela Alves Rodrigues - 82311687

Henryk Bagdanovicius Roza - 823135401

Emilio Gabriel dos Santos Palacios - 823112595

GESTÃO E QUALIDADE DE SOFTWARE

PROJETO A3 – DOCUMENTO DE DESENVOLVIMENTO DE BIBLIOTECA  
VIRTUAL

São Paulo

2025

## SUMARIO

1. INTRODUÇÃO.....	3
2. CONTEXTUALIZAÇÃO DO PROJETO .....	3
3. PLANEJAMENTO DE TESTES DE SOFTWARE.....	3
3.1. CRONOGRAMA DE ATIVIDADES .....	3
3.2. ALOCAÇÃO DE RECURSOS.....	4
3.3. MARCOS DO PROJETO .....	4
4. DOCUMENTOS DE DESENVOLVIMENTO DE SOFTWARE .....	4
4.1 PLANO DE PROJETO .....	4
4.1.1 PLANEJAMENTO DE PROJETO .....	4
4.1.2 ESCOPO .....	5
4.1.3 RECURSOS.....	5
4.2 DOCUMENTO DE REQUISITOS .....	7
4.3 PLANEJAMENTO DE TESTES .....	9
4.3.1 PLANO DE TESTES .....	9
4.3.1.1 INTRODUÇÃO.....	9
4.3.1.2 ESCOPO .....	9
4.3.1.3 OBJETIVOS .....	9
4.3.1.4 REQUISITOS A SEREM TESTADOS.....	10
4.3.1.5 ESTRATÉGIAS E FERRAMENTAS.....	10
4.3.1.6 RECURSOS.....	10
4.3.1.7 CRONOGRAMA .....	11
4.3.1.8 MARCOS DO PROJETO .....	11
4.3.2. CASOS DE TESTE.....	12
4.3.3. ROTEIRO DE TESTES .....	13
5. GESTÃO E CONFIGURAÇÃO DE SOFTWARE.....	13
6. REPOSITÓRIO DE GESTÃO DE CONFIGURAÇÃO DE SOFTWARE .....	14
7. CONCLUSÃO.....	15

## **1. INTRODUÇÃO**

Esta documentação tem como objetivo descrever o desenvolvimento do projeto A3 de desenvolvimento do sistema BiblioTech, uma aplicação web destinada à gestão de bibliotecas escolares. O sistema será utilizado por instituições de ensino para gerenciar o cadastro de livros, controle de empréstimos e devoluções, além da geração de relatórios sobre os livros e o uso de suas bibliotecas. O objetivo é automatizar tarefas que geralmente são feitas de forma manual, promovendo maior controle e gestão, redução de falhas e melhora na experiência dos usuários. Este documento serve como base para o planejamento, execução e testes do projeto, descrevendo os requisitos funcionais, não funcionais, escopo, recursos e estratégias de qualidade a serem aplicadas.

## **2. CONTEXTUALIZAÇÃO DO PROJETO**

Em muitas escolas, o gerenciamento da biblioteca é feito manualmente, o que dificulta a organização dos livros, o controle de empréstimos e a emissão de relatórios. Esse cenário pode gerar atrasos na devolução de livros, perdas de exemplares e dificuldade no acesso às informações. O BiblioTech tem como objetivo esse processo, fornecendo uma solução acessível, prática e segura para a realidade das instituições de ensino. O sistema permitirá:

- Cadastro e consulta de livros com informações detalhadas;
- Registro de usuários com diferentes perfis de acesso;
- Empréstimos e devoluções com controle de prazos;
- Geração de relatórios sobre movimentações da biblioteca;
- Alertas para livros em atraso e controle do histórico de uso.

## **3. PLANEJAMENTO DE TESTES DE SOFTWARE**

### **3.1. CRONOGRAMA DE ATIVIDADES**

O cronograma de testes do sistema BiblioTech será dividido em cinco fases:

- Semana 1: Planejamento e definição da estratégia de testes.
- Semana 2: Elaboração dos casos e roteiros de teste.
- Semana 3: Execução dos testes unitários.
- Semana 4: Execução dos testes de integração.
- Semana 5: Execução dos testes de sistema e testes de aceitação.

As atividades serão acompanhadas através de um quadro Kanban para garantir visibilidade e controle do progresso.

### 3.2. ALOCAÇÃO DE RECURSOS

Para este projeto, os seguintes recursos serão alocados:

- 2 Analistas de Teste (QA) responsáveis por criar, executar e documentar os testes.
- 1 Analista de Requisitos para apoiar a definição dos critérios de aceitação e validação de funcionalidades.
- Ambiente de homologação com banco de dados espelhado do ambiente real, configurado com dados fictícios.
- Ferramentas utilizadas: Jira (gestão de tarefas), Selenium (testes automatizados), Postman (testes de API) e TestLink (documentação de testes).

### 3.3. MARCOS DO PROJETO

- Finalização do planejamento de testes (final da Semana 1);
- Aprovação dos casos de teste pela equipe de desenvolvimento (Semana 2);
- Conclusão dos testes unitários (Semana 3);
- Integração contínua com execução de testes automatizados (a partir da Semana 4);
- - Validação final com stakeholders e usuários-chave no ambiente de homologação (Semana 5).

## 4. DOCUMENTOS DE DESENVOLVIMENTO DE SOFTWARE

### 4.1 PLANO DE PROJETO

#### 4.1.1 PLANEJAMENTO DE PROJETO

**Metodologia:** Utilização de Scrum com sprints de 2 semanas

**Duração total:** 6 meses (12 sprints)

**Equipe:**

- 1 Product Owner
- 1 Scrum Master
- 4 Desenvolvedores Full-Stack
- 1 QA Engineer
- 1 Designer UX/UI

#### **Riscos identificados:**

- Integração com sistemas legados das escolas
- Capacitação de usuários finais
- Disponibilidade de infraestrutura tecnológica nas escolas

#### **4.1.2 ESCOPO**

O escopo do projeto inclui as seguintes funcionalidades principais do sistema:

- Cadastro de usuários (alunos, professores, funcionários)
- Gestão de acervo com categorização
- Sistema de empréstimos/devoluções automatizado
- Relatórios de uso em tempo real
- Notificações automatizadas

#### **4.1.3 RECURSOS**

Os recursos necessários para o desenvolvimento do sistema BiblioTech abrangem tecnologia, infraestrutura e equipe humana. Abaixo está o detalhamento e o orçamento total:

##### **Tecnológicos (Software e Serviços)**

- Frontend: React.js (licença gratuita)
- Backend: Node.js + NestJS (licença gratuita)
- Banco de Dados: MySQL (licença gratuita - uso comunitário)
- Hospedagem em Nuvem: AWS EC2
  - Total para 6 meses: R\$ 2.880
- **Ferramentas utilizadas:**
  - Jira:  $\text{R\$ } 43/\text{usuário/mês} \times 8 \text{ usuários} \times 6 \text{ meses} = \text{R\$ } 2.064$
  - Selenium, GitHub, Figma: Gratuitos (versões educacionais ou open-source)

**Infraestrutura (Hardware)**

Para garantir um ambiente de desenvolvimento eficiente, serão utilizados 5 computadores padrão desenvolvedor e 1 computador para testes com as seguintes configurações:

**Desenvolvedores (5 máquinas)**

- **Processador:** Intel Core i5-12400
- **Memória RAM:** 16GB DDR4
- **Armazenamento:** SSD 512GB
- **Sistema Operacional:** Windows 11 Pro
- **Custo por unidade:** R\$ 4.500
- **Total (5 máquinas): R\$ 22.500**

**Testes (1 máquina para QA)**

- **Processador:** Intel Core i7-12700
- **Memória RAM:** 32GB DDR4
- **Armazenamento:** SSD 1TB
- **Sistema Operacional:** Windows 11 Pro
- **Custo por unidade:** R\$ 6.000

Total de Infraestrutura (hardware): R\$ 28.500

**Recursos Humanos**

Para os profissionais alocados em tempo integral por 6 meses:

Função	Qtde	Custo Mensal	Duração	Total
Product Owner	1	R\$ 10.000	6 meses	R\$ 60.000
Scrum Master	1	R\$ 8.000	6 meses	R\$ 48.000
Desenvolvedores	4	R\$ 7.000	6 meses	R\$ 168.000
QA Engineer	1	R\$ 7.000	6 meses	R\$ 42.000
UX/UI Designer	1	R\$ 6.500	6 meses	R\$ 39.000

Total Recursos Humanos: R\$ 357.000

# Orçamento Consolidado do Projeto

Categoria	Valor (R\$)
Recursos Humanos	R\$ 357.000
Infraestrutura	R\$ 28.500
Serviços em Nuvem	R\$ 2.880
Licenças (Jira)	R\$ 2.064
<b>Total Geral</b>	<b>R\$ 390.444</b>

## 4.1.4 ESTIMATIVA DE PROJETO

Fase	Duração	Custo
Requisitos	3 semanas	R\$ 45k
Desenvolvimento	14 semanas	R\$ 210k
Testes	4 semanas	R\$ 50k
Implantação	1 semana	R\$ 15k

## 4.2 DOCUMENTO DE REQUISITOS

### Requisitos funcionais:

**RF-001:** O sistema deve permitir o cadastro de usuários (alunos, professores e bibliotecários), com validação obrigatória de um documento oficial (ex: RG ou CPF).

**Prioridade:** Alta

**RF-002:** O sistema deve permitir a pesquisa de livros no acervo utilizando múltiplos critérios, como título ou autor.

**Prioridade:** Alta

**RF-003:** O sistema deve permitir ao usuário reservar exemplares com antecedência, desde que disponíveis, e notificar quando o exemplar estiver pronto para retirada.

**Prioridade:** Média

**RF-004:** O sistema deve gerar automaticamente relatórios mensais de movimentações (empréstimos, devoluções, reservas e usuários ativos).

**Prioridade:** Média

**RF-005:** O sistema deve permitir o cadastro, edição e remoção de livros no acervo, incluindo quantidade de exemplares.

**Prioridade:** Alta

**RF-006:** O sistema deve registrar empréstimos e devoluções com controle de prazos e cálculo de atraso.

**Prioridade:** Alta

**RF-007:** O sistema deve enviar alertas de devolução pendente por e-mail ou notificação interna.

**Prioridade:** Média

**RF-008:** O sistema deve manter o histórico de empréstimos de cada usuário.

**Prioridade:** Média

**RF-009:** O sistema deve permitir o acesso com autenticação (login e senha), com diferenciação de permissões por perfil.

**Prioridade:** Alta

**RF-010:** O sistema deve permitir ao administrador gerar relatórios personalizados por período, status e usuário.

**Prioridade:** Baixa

### **Requisitos Não Funcionais:**

**RNF-001:** O sistema deve garantir tempo de resposta inferior a 2 segundos para consultas com até 1.000 usuários concorrentes.

**Prioridade:** Alta

**RNF-002:** O sistema deve ter disponibilidade mínima de 99,9% durante o horário escolar (07h às 18h, dias úteis).

**Prioridade:** Alta

**RNF-003:** O sistema deve ser compatível com os principais navegadores modernos (Chrome, Firefox, Edge, Safari).

**Prioridade:** Média

**RNF-004:** Os dados devem ser armazenados em banco de dados relacional com backups automáticos diários.

**Prioridade:** Alta

**RNF-005:** O sistema deve permitir acesso via dispositivos móveis e desktop, com design responsivo.

**Prioridade:** Alta

**RNF-006:** O sistema deve implementar criptografia de senhas utilizando algoritmo seguro.

**Prioridade:** Alta

**RNF-007:** As operações críticas devem possuir logs de auditoria com data, hora, usuário e ação executada.

**Prioridade:** Média



**RNF-008:** O sistema deve suportar ao menos 5.000 cadastros de usuários e 10.000 livros no acervo sem degradação de desempenho.  
Prioridade: Média

## **4.3 PLANEJAMENTO DE TESTES**

### **4.3.1 PLANO DE TESTES**

#### **4.3.1.1 INTRODUÇÃO**

A estratégia de testes do projeto BiblioTech será contínua e integrada ao pipeline de integração e entrega contínuas (CI/CD), assegurando que cada nova funcionalidade seja validada automaticamente antes da sua liberação. O foco será garantir a estabilidade do sistema, a segurança dos dados e a facilidade de uso.

#### **4.3.1.2 ESCOPO**

O plano de testes abrange:

- 100% dos requisitos funcionais (RF-001 a RF-010), com foco em garantir que todas as funcionalidades essenciais do sistema operem conforme o esperado em diferentes cenários (fluxo ideal, fluxos alternativos e casos de erro).
- 85% dos requisitos não funcionais (RNF-001 a RNF-008), incluindo testes de desempenho, compatibilidade, segurança e responsividade.
- Exceções: Testes de carga com mais de 10.000 usuários simultâneos não serão realizados nesta fase, por excederem os limites de infraestrutura do ambiente de testes atual.

#### **4.3.1.3 OBJETIVOS**

Os objetivos do plano de testes são:

- Eliminar bugs críticos na versão de produção, garantindo estabilidade desde o primeiro uso;
- Atingir ao menos 95% de cobertura de código, com testes automatizados unitários e de integração;
- Obter aprovação de usabilidade por 90% dos usuários-piloto, validando que as funcionalidades são intuitivas e o sistema é acessível a usuários com pouca familiaridade com tecnologia;

- Assegurar conformidade com os critérios de aceite definidos nos requisitos e casos de uso.

#### 4.3.1.4 REQUISITOS A SEREM TESTADOS

Serão testados integralmente os seguintes componentes:

- **Requisitos Funcionais (RF):**
  - RF-001 ao RF-010 – funcionalidades como cadastro de usuários e livros, processos de empréstimo e devolução, geração de relatórios, autenticação, notificações e permissões de acesso.
- **Requisitos Não Funcionais (RNF):**
  - RNF-001 ao RNF-008 – desempenho, disponibilidade, segurança de dados, responsividade e escalabilidade.

Os testes garantirão que o sistema atenda às necessidades dos usuários e esteja preparado para uso em ambiente real e seja validado técnica e funcionalmente de forma robusta.

#### 2.3.1.5 ESTRATÉGIAS E FERRAMENTAS

Tipo de teste	Ferramenta	Cobertura
Unitário	Jest	Componentes críticos
Integração	Postman	APIs REST
Usabilidade	UserTesting.com	Fluxos principais
Carga	JMeter	500 usuários conectados

#### 4.3.1.6 RECURSOS

- **Ambiente Dedicado de Testes:**  
Um ambiente exclusivo, espelhado ao de produção, será mantido ao longo de todo o ciclo de desenvolvimento. Ele permitirá a execução de testes funcionais, de regressão e automação em condições controladas e representativas da realidade dos usuários finais.

- **Equipe de Testes:**
  - 2 testadores manuais, responsáveis pela validação exploratória e execução de casos de teste com foco em usabilidade, fluxos alternativos e cenários de exceção.
  - 1 analista de automação, responsável pela implementação e manutenção dos testes automatizados utilizando Selenium integrado ao pipeline CI/CD. Esse profissional garantirá a eficiência na regressão contínua e a cobertura dos principais fluxos do sistema.
- **Ferramentas de Suporte:**
  - Jira para controle de bugs e acompanhamento de testes;
  - Selenium para automação de testes funcionais;
  - Figma para validação da aderência às interfaces projetadas;
  - Postman para testes de API durante as fases de integração.

#### 4.3.1.7 CRONOGRAMA

Sprint	Atividades
1-4	Testes unitários/integração
5-8	Testes sistema/usabilidade
9-12	Testes aceitação/performance

#### 4.3.1.8 MARCOS DO PROJETO

Os marcos do projeto representam pontos críticos de acompanhamento e validação ao longo do ciclo de desenvolvimento e testes do sistema BiblioTech. Eles permitem mensurar o progresso, identificar riscos antecipadamente e garantir que os objetivos definidos para cada fase sejam atingidos.

- **Kick-off (Final da Sprint 1):**  
Reunião inicial de alinhamento entre todos os membros do time e stakeholders. Nessa etapa, são apresentados os objetivos do projeto, o escopo inicial, a metodologia adotada (Scrum) e a definição clara dos papéis e responsabilidades. Também será aprovado plano de testes preliminar e o ambiente de homologação é iniciado.
- **MVP Funcional (Sprint 6):**  
Entrega do Produto Mínimo Viável, contendo as principais funcionalidades do sistema, como cadastro de usuários e livros, processo básico de empréstimo/devolução e autenticação de acesso. A partir dessa versão, é

possível realizar testes funcionais e demonstrações com os usuários-piloto para validação dos fluxos principais.

- **Beta Testing (Sprint 10):**

Lançamento da versão beta do sistema em ambiente de homologação para um grupo selecionado de usuários reais (como bibliotecários e professores). Nessa fase, são executados testes de usabilidade, testes exploratórios e coleta de feedback qualitativo. Também são avaliadas questões de desempenho, estabilidade e aderência aos requisitos não funcionais.

- **Go-Live (Sprint 12):**

Lançamento oficial do sistema no ambiente de produção. Após a validação final, correções de bugs e ajustes finos, o sistema é disponibilizado para uso completo nas instituições de ensino. Também é realizada uma revisão geral da documentação, fechamento dos testes automatizados e entrega do pacote de deploy com manual técnico e de usuário.

#### 4.3.2. CASOS DE TESTE

##### Exemplo 1:

**Caso de teste:** Realizar empréstimo de livro

**Objetivo:** Verificar se o sistema permite o empréstimo de um livro disponível.

**Pré-condições:** Usuário e livro cadastrados no sistema.

**Entradas:** Código do livro, ID do usuário.

**Passos:**

2. Acessar o sistema.
3. Realizar login como usuário.
4. Buscar livro disponível.
5. Clicar em 'Solicitar empréstimo'.

**Resultado esperado:** O sistema deve registrar o empréstimo com a data de retirada e a data prevista de devolução.

##### Exemplo 2:

**Caso de teste:** Devolução de livro atrasado

**Objetivo:** Verificar se o sistema registra multa ao devolver um livro após o prazo.

**Pré-condições:** Empréstimo com prazo vencido.

**Entradas:** Código do livro, ID do usuário.

**Passos:**

1. Realizar login.
2. Acessar menu de devoluções.
3. Selecionar o livro emprestado.
4. Confirmar devolução.

**Resultado esperado:** O sistema deve registrar a devolução e aplicar a multa conforme as regras definidas (ex: R\$1,00 por dia de atraso).

### 4.3.3. ROTEIRO DE TESTES

#### Roteiro 1 - Fluxo ideal:

1. Acessar o sistema;
2. Realizar login como administrador;
3. Cadastrar novo usuário;
4. Cadastrar novo livro;
5. Realizar empréstimo do livro para o usuário;
6. Consultar status do empréstimo;
7. Realizar devolução do livro;
8. Gerar relatório de empréstimos ativos e devolvidos.

#### Roteiro 2 - Fluxo alternativo e de erro:

1. Tentar login com credenciais inválidas;
2. Tentar realizar empréstimo de livro indisponível;
3. Tentar devolver livro não emprestado;
4. Consultar relatórios com perfil de usuário sem permissão;
5. Verificar mensagens de erro e tratamento adequado de exceções.

## 5. GESTÃO E CONFIGURAÇÃO DE SOFTWARE

#### Itens de configuração:

- Código-fonte
  - Módulo de cadastro de usuários, livros e categorias
  - Módulo de controle de estoque (quantidade disponível)
  - Módulo de registro de empréstimos e devoluções
  - Módulo de relatórios de uso
  - Módulo de notificações de atraso
- Database
  - Estruturas de tabelas para livros, usuários, empréstimos, devoluções, categorias
- Documentação

- Manual do usuário
  - Documentação técnica (diagramas e APIs)
  - Políticas de segurança
  - Script de implantação
  - Scripts de criação e atualização do database
  - Scripts de backup e restauração
  - Configurações do sistema
  - Parâmetros de prazos de empréstimo
  - Configurações de envio de notificações
- Versionamento

Para o versionamento do projeto, o uso do GitHub para o controle de versão do projeto é essencial, sendo cada atualização acompanhada de mensagem descritiva e seu respectivo autor.

- Controle de mudanças
  - Qualquer modificação deve estar registrada por solicitação, devendo ser aprovada somente pelo gerente de projeto
  - As modificações devem ser testadas, e depois implementadas

Para o sistema requerido, será necessário a elaboração de testes unitários para cada módulo do software, bem como para o banco de dados. Sendo assim, será necessário a realização de testes dos módulos de usuários, livros e categorias; do módulo do controle de estoque; do módulo de registro de empréstimos e devoluções; do módulo de relatórios de uso e do módulo de notificações de atraso. Após os testes unitários, será feito o teste de integração entre os módulos, e depois o teste de validação e verificação, isto é, verificar se foi implementado corretamente e validar com o cliente se foi este software pedido.

Para o repositório, o proprietário do repositório do GitHub ficará encarregado do versionamento do projeto, e de suas atualizações, ficando a seu encargo a aprovação ou reprovação de possíveis alterações

## 6. REPOSITÓRIO DE GESTÃO DE CONFIGURAÇÃO DE SOFTWARE

O repositório oficial do projeto BiblioTech será hospedado no GitHub, garantindo versionamento, rastreabilidade e colaboração entre os membros da equipe. A organização do repositório seguirá os seguintes critérios:

- **Estrutura de pastas:**
  - /frontend: Código-fonte do React.js
  - /backend: Código-fonte do Node.js/NestJS
  - /docs: Documentação técnica e funcional
  - /scripts: Scripts de automação, backup, implantação

- /tests: Scripts de testes unitários e automatizados
- **Branches:**
  - main: Branch principal (versão estável)
  - develop: Integração de funcionalidades em desenvolvimento
  - feature/\*: Desenvolvimento de novas funcionalidades
  - hotfix/\*: Correções emergenciais
  - release/\*: Versões prontas para homologação

## 7. CONCLUSÃO

O projeto BiblioTech foi é uma proposta de solução tecnológica voltada para a gestão eficiente de bibliotecas escolares, que busca atender a uma necessidade real de modernização e automação. Além da documentação do desenvolvimento do sistema, a proposta central deste trabalho foi a elaboração completa da documentação de desenvolvimento de software, A documentação inclui desde o planejamento do projeto e definição de requisitos até o plano de testes, roteiros de validação, gestão de configuração e controle de versões. Através deste trabalho, foi possível simular o ciclo completo de desenvolvimento de um software real, reforçando a importância de boas práticas de gestão e qualidade de software aprendidos ao longo do semestre.