

SISTEMA DE GESTÃO DE BIBLIOTECA ESCOLAR

Especificação de Requisitos

Versão 1.3 — Documento de Referência

Íuri Santos de Souza

Tecnologias: PHP 7.4+ (procedural) · MySQL 5.7+ · HTML5 · CSS3 · JavaScript (ES6)

Design System: Material Design 3

Ambiente: MAMP (Apache + MySQL local)

Data: Abril de 2026

1. Introdução

Este documento especifica os requisitos funcionais e não funcionais do Sistema de Gestão de Biblioteca Escolar na sua versão 1.3. O sistema foi desenvolvido com tecnologias web procedurais (PHP, MySQL, HTML, CSS e JavaScript), seguindo o Design System Material Design 3, para ser executado localmente no ambiente MAMP.

1.1 Objetivo

Prover uma plataforma de gerenciamento completa para bibliotecas escolares, contemplando controle do acervo de livros, cadastro de estudantes, registro e acompanhamento de empréstimos, geração de relatórios e administração de funcionários com acesso autenticado.

1.2 Escopo

O sistema atende às seguintes necessidades operacionais da biblioteca escolar:

- Gestão do acervo bibliográfico com controle de disponibilidade de exemplares
- Cadastro e manutenção de estudantes por série e turma
- Registro de empréstimos com controle de prazos, renovações e devoluções
- Relatórios operacionais com exportação em PDF
- Controle de acesso por autenticação de funcionários com perfis de permissão
- Dashboard com indicadores e gráficos estatísticos em tempo real

1.3 Convenções de Prioridade

Prioridade	Critério
Alta	Funcionalidade central; impede operação sem ela
Média	Funcionalidade importante; degrada a experiência se ausente
Baixa	Melhoria de usabilidade; sistema funciona sem ela

2. Requisitos Funcionais

Os requisitos funcionais descrevem o comportamento e as funcionalidades que o sistema deve executar. Estão organizados por módulo, refletindo a estrutura de navegação do sistema.

2.1 Autenticação e Controle de Acesso

ID	Módulo	Descrição	Prioridade	Status
RF01	Autenticação	O sistema deve exibir uma tela de login (login.php) que solicita e-mail e senha antes de permitir qualquer acesso às páginas protegidas.	Alta	Implementado
RF02	Autenticação	O sistema deve validar as credenciais contra a tabela funcionarios do banco de dados, verificando a senha com password_verify() (bcrypt).	Alta	Implementado
RF03	Autenticação	Após login bem-sucedido, o sistema deve regenerar o ID de sessão (session_regenerate_id) e armazenar nome, e-mail e perfil do funcionário na sessão.	Alta	Implementado
RF04	Autenticação	O sistema deve registrar a data e hora do último acesso do funcionário (campo ultimo_login) a cada login.	Média	Implementado
RF05	Autenticação	O sistema deve redirecionar automaticamente para login.php qualquer tentativa de acesso direto a páginas protegidas sem sessão ativa (guard auth_check).	Alta	Implementado
RF06	Autenticação	O sistema deve permitir logout com destruição completa da sessão (logout.php), redirecionando para a tela de login.	Alta	Implementado
RF07	Autenticação	O sistema deve suportar dois perfis de acesso: Administrador (acesso total) e Bibliotecário (sem acesso ao módulo de funcionários).	Alta	Implementado

ID	Módulo	Descrição	Prioridade	Status
RF08	Autenticação	A tela de login deve exibir botão para alternar a visibilidade da senha (mostrar/ocultar).	Baixa	Implementado Com erros

2.2 Gestão do Acervo de Livros

ID	Módulo	Descrição	Prioridade	Status
RF09	Acervo	O sistema deve permitir cadastrar novos livros com os campos: título, autor, ISBN, editora, ano de publicação, gênero, quantidade de exemplares e descrição/sinopse.	Alta	Implementado
RF10	Acervo	O sistema deve garantir unicidade do ISBN; tentativas de cadastro com ISBN duplicado devem ser rejeitadas com mensagem de erro.	Alta	Implementado
RF11	Acervo	O sistema deve manter automaticamente o campo quantidade_disponível decrementando ao emprestar e incrementando ao devolver.	Alta	Implementado
RF12	Acervo	O sistema deve permitir editar todos os campos de um livro, recalculando quantidade_disponível ao alterar quantidade_total (preservando exemplares em empréstimo).	Alta	Implementado
RF13	Acervo	O sistema deve impedir a exclusão de livros com empréstimos ativos ou atrasados, exibindo mensagem de erro.	Alta	Implementado
RF14	Acervo	O sistema deve exibir a listagem de livros com paginação (12 por página) e busca por título, autor, ISBN ou gênero.	Média	Implementado
RF15	Acervo	O sistema deve exibir visualmente a disponibilidade de cada livro: chip verde para disponível, chip vermelho para indisponível (0 exemplares).	Média	Implementado

2.3 Gestão de Estudantes

ID	Módulo	Descrição	Prioridade	Status
RF16	Estudantes	O sistema deve permitir cadastrar estudantes com os campos: nome, matrícula, turma, série, e-mail, telefone, responsável e telefone do responsável.	Alta	Implementado
RF17	Estudantes	O sistema deve garantir unicidade da matrícula; duplicatas devem ser rejeitadas com mensagem de erro.	Alta	Implementado
RF18	Estudantes	O sistema deve permitir editar todos os dados cadastrais de um estudante.	Alta	Implementado
RF19	Estudantes	O sistema deve impedir a exclusão de estudantes com empréstimos ativos ou atrasados.	Alta	Implementado
RF20	Estudantes	O sistema deve exibir listagem com paginação (15 por página) e busca por nome, matrícula, turma ou série.	Média	Implementado

2.4 Gestão de Empréstimos

ID	Módulo	Descrição	Prioridade	Status
RF21	Empréstimos	O sistema deve permitir registrar um empréstimo selecionando livro disponível, estudante, data do empréstimo e data prevista de devolução.	Alta	Implementado
RF22	Empréstimos	A data prevista de devolução deve ser preenchida automaticamente com base no prazo padrão configurado (configuracoes: prazo_emprestimo), sendo editável pelo funcionário.	Alta	Implementado
RF23	Empréstimos	O sistema deve bloquear o empréstimo se o estudante já atingiu o limite de livros simultâneos configurado (configuracoes: limite_livros), exibindo mensagem de erro.	Alta	Implementado
RF24	Empréstimos	O sistema deve bloquear empréstimo duplicado: mesmo livro para o mesmo estudante com status diferente de devolvido.	Alta	Implementado

ID	Módulo	Descrição	Prioridade	Status
RF25	Empréstimos	O sistema deve bloquear empréstimo de livro com quantidade_disponivel igual (ou menor) a zero.	Alta	Implementado
RF26	Empréstimos	O sistema deve atualizar automaticamente o status de empréstimos para atrasado quando a data_prevista_devolucao for anterior à data atual (executado a cada carregamento de página relevante).	Alta	Implementado
RF27	Empréstimos	O sistema deve exibir na listagem de empréstimos, para os atrasados, o número de dias de atraso.	Média	Implementado
RF28	Empréstimos	O sistema deve permitir renovar o prazo de um empréstimo ativo ou atrasado com um clique, estendendo a data_prevista_devolucao pelo prazo padrão configurado e revertendo o status para ativo.	Média	Implementado
RF29	Empréstimos	O sistema deve suportar filtros na listagem: por status (todos, ativos, atrasados, devolvidos) e busca textual por livro, estudante ou matrícula.	Média	Implementado

2.5 Gestão de Devoluções

ID	Módulo	Descrição	Prioridade	Status
RF30	Devoluções	O sistema deve permitir registrar a devolução de um empréstimo informando a data real de devolução.	Alta	Implementado
RF31	Devoluções	Ao registrar devolução, o sistema deve atualizar o status do empréstimo para devolvido, gravar data_devolucao e incrementar quantidade_disponivel do livro, em transação atômica.	Alta	Implementado
RF32	Devoluções	A tela de devoluções deve listar apenas empréstimos pendentes (status ativo ou atrasado), ordenados com atrasados primeiro.	Média	Implementado

ID	Módulo	Descrição	Prioridade	Status
RF33	Devoluções	O sistema deve destacar visualmente empréstimos atrasados na listagem de devoluções.	Média	Implementado
RF34	Devoluções	O link "Devolver" na listagem de empréstimos deve direcionar ao formulário de devolução pré-selecionando o empréstimo correspondente.	Baixa	Implementado

2.6 Relatórios

ID	Módulo	Descrição	Prioridade	Status
RF35	Relatórios	O sistema deve gerar o relatório de Devoluções Atrasadas listando todos os empréstimos com status atrasado, exibindo livro, estudante, turma/série, contato do responsável e dias de atraso.	Alta	Implementado
RF36	Relatórios	O sistema deve gerar o relatório de Empréstimos por Período, filtrável por data de início e data de fim, com contagem por status.	Alta	Implementado
RF37	Relatórios	O sistema deve gerar o relatório de Devoluções por Período, filtrável por data, indicando para cada devolução se ocorreu no prazo ou com atraso.	Alta	Implementado
RF38	Relatórios	O sistema deve gerar o relatório de Acervo Completo listando todos os livros com quantidades total, disponível, emprestada e total histórico de empréstimos.	Média	Implementado
RF39	Relatórios	Todos os relatórios devem ser exportáveis em PDF via página dedicada (pdf_relatorio.php), com layout profissional contendo cabeçalho institucional, tabelas formatadas e rodapé configurável.	Alta	Implementado
RF40	Relatórios	A página de relatórios deve apresentar abas de navegação entre os quatro tipos de relatório.	Baixa	Implementado

2.7 Histórico do Estudante

ID	Módulo	Descrição	Prioridade	Status
RF41	Histórico	O sistema deve permitir buscar um estudante pelo nome ou matrícula para visualizar seu histórico completo de empréstimos.	Alta	Implementado
RF42	Histórico	O sistema deve exibir, para cada estudante, estatísticas individuais: total de empréstimos, devolvidos, em aberto, atrasados e devolvidos com atraso.	Média	Implementado
RF43	Histórico	O sistema deve exibir os três livros mais emprestados pelo estudante (favoritos).	Baixa	Implementado
RF44	Histórico	O histórico do estudante deve ser exportável em PDF (pdf_historico.php) com dados de perfil, estatísticas e tabela completa de empréstimos.	Média	Implementado

2.8 Dashboard

ID	Módulo	Descrição	Prioridade	Status
RF45	Dashboard	O dashboard deve exibir 6 indicadores em cards: total de exemplares, total de títulos, estudantes ativos, empréstimos abertos, devoluções atrasadas e empréstimos no mês corrente.	Alta	Implementado
RF46	Dashboard	O dashboard deve exibir gráfico de barras com a quantidade de empréstimos nos últimos 7 meses.	Média	Implementado
RF47	Dashboard	O dashboard deve exibir gráfico de rosca com a distribuição dos empréstimos por status (ativo, atrasado, devolvido).	Média	Implementado
RF48	Dashboard	O dashboard deve exibir gráfico de barras horizontais com os gêneros literários mais emprestados (top 6).	Baixa	Implementado
RF49	Dashboard	O dashboard deve exibir gráfico de barras com as séries escolares que mais realizam empréstimos (top 6).	Baixa	Implementado

ID	Módulo	Descrição	Prioridade	Status
RF50	Dashboard	O dashboard deve exibir lista dos últimos 6 empréstimos registrados e o ranking dos 5 livros mais emprestados.	Média	Implementado

2.9 Notificações de Prazo

ID	Módulo	Descrição	Prioridade	Status
RF51	Notificações	O sistema deve exibir um ícone de sino na barra superior em todas as páginas, com ponto laranja quando houver empréstimos com vencimento dentro do prazo de alerta configurado.	Média	Implementado
RF52	Notificações	Ao clicar no sino, deve-se abrir painel listando os empréstimos próximos do vencimento com mensagens diferenciadas: "Vence hoje!", "Vence amanhã" ou "Vence em X dias".	Média	Implementado
RF53	Notificações	O badge de notificação de atrasos no menu lateral de Empréstimos deve exibir a contagem de empréstimos atrasados.	Média	Implementado

2.10 Busca Global

ID	Módulo	Descrição	Prioridade	Status
RF54	Busca	O sistema deve disponibilizar campo de busca global na barra superior (presente em todas as páginas autenticadas), com dropdown de sugestões em tempo real (debounce de 280ms).	Média	Implementado
RF55	Busca	O dropdown de busca deve apresentar resultados agrupados por categoria: livros (máx. 4), estudantes (máx. 4) e empréstimos (máx. 3), com ícone e subtítulo identificadores.	Média	Implementado
RF56	Busca	Pressionar Enter ou clicar em "Ver todos os resultados" deve redirecionar para a página de resultados	Média	Implementado

ID	Módulo	Descrição	Prioridade	Status
		completos (busca.php) com listagem filtrada por categoria.		

2.11 Gestão de Funcionários

ID	Módulo	Descrição	Prioridade	Status
RF57	Funcionários	O módulo de funcionários deve ser acessível somente ao perfil Administrador (guard auth_admin); funcionários com perfil Bibliotecário são redirecionados ao tentar acessar.	Alta	Implementado
RF58	Funcionários	O administrador deve poder cadastrar, editar e remover funcionários, definindo nome, e-mail, senha (bcrypt) e perfil (admin ou bibliotecário).	Alta	Implementado
RF59	Funcionários	O sistema deve impedir que um funcionário remova sua própria conta.	Alta	Implementado
RF60	Funcionários	Na edição, o campo senha deve ser opcional; deixá-lo em branco mantém a senha atual.	Média	Implementado
RF61	Funcionários	A listagem de funcionários deve exibir perfil, status (ativo/inativo) e data do último login.	Baixa	Implementado

2.12 Perfil do Funcionário

ID	Módulo	Descrição	Prioridade	Status
RF62	Perfil	Cada funcionário deve poder atualizar seu próprio nome e e-mail na página Meu Perfil.	Média	Implementado
RF63	Perfil	O funcionário deve poder alterar sua própria senha informando a senha atual, a nova senha e a confirmação; a nova senha deve ter mínimo de 6 caracteres.	Alta	Implementado
RF64	Perfil	O sistema deve validar em tempo real (JavaScript) se a nova senha e a	Baixa	Implementado

ID	Módulo	Descrição	Prioridade	Status
		confirmação conferem, com feedback visual no campo.		
RF65	Perfil	A página de perfil deve exibir data do último login e data de cadastro do funcionário.	Baixa	Implementado

2.13 Configurações do Sistema

ID	Módulo	Descrição	Prioridade	Status
RF66	Configurações	O sistema deve permitir configurar o nome da escola e o nome da biblioteca, refletindo essas informações no cabeçalho da navegação, no título do navegador e nos PDFs gerados.	Alta	Implementado
RF67	Configurações	O sistema deve permitir configurar o prazo padrão de empréstimo em dias, aplicado automaticamente no formulário de novo empréstimo e nas renovações.	Alta	Implementado
RF68	Configurações	O sistema deve permitir configurar o limite máximo de livros por estudante, aplicado na validação do registro de empréstimos.	Alta	Implementado
RF69	Configurações	O sistema deve permitir configurar o número de dias de antecedência para alertas de vencimento no sino de notificações.	Média	Implementado
RF70	Configurações	O sistema deve permitir configurar o texto do rodapé exibido nos relatórios PDF.	Baixa	Implementado
RF71	Configurações	As configurações devem ser persistidas na tabela configuracoes do banco de dados e carregadas com cache de sessão PHP para evitar consultas repetidas.	Média	Implementado

3. Requisitos Não Funcionais

Os requisitos não funcionais definem restrições e características de qualidade que o sistema deve satisfazer, independentemente das funcionalidades específicas.

3.1 Desempenho

ID	Categoria	Descrição	Referência
RNF01	Desempenho	Todas as páginas do sistema devem carregar em menos de 2 segundos em rede local (LAN), considerando banco de dados com até 10.000 empréstimos registrados.	Ambiente MAMP local
RNF02	Desempenho	As consultas de busca global (busca.php?ajax=1) devem retornar resultados em menos de 500ms, com debounce de 280ms no front-end para evitar requisições excessivas.	busca.php + app.js
RNF03	Desempenho	O carregamento dos gráficos do dashboard (Chart.js) não deve bloquear a renderização das demais informações da página; o Chart.js é carregado via CDN ao final do body.	dashboard.php
RNF04	Desempenho	As configurações do sistema devem ser cacheadas em variável global PHP (\$_CONFIG_CACHE) durante o ciclo de vida da requisição, evitando múltiplas consultas ao banco por página.	config.php

3.2 Segurança

ID	Categoria	Descrição	Referência
RNF05	Segurança	Senhas de funcionários devem ser armazenadas exclusivamente como hash bcrypt (PASSWORD_BCRYPT), nunca em texto plano.	auth.php + funcionarios.php
RNF06	Segurança	Todas as entradas de usuário devem ser sanitizadas com htmlspecialchars() + strip_tags() + trim() antes de uso; consultas ao banco devem usar exclusivamente prepared statements PDO.	functions.php (sanitize())

ID	Categoria	Descrição	Referência
RNF07	Segurança	O ID de sessão deve ser regenerado (<code>session_regenerate_id(true)</code>) imediatamente após autenticação bem-sucedida para mitigar session fixation.	login.php
RNF08	Segurança	Toda página protegida deve chamar <code>auth_check()</code> antes de qualquer output; páginas exclusivas de administrador devem chamar <code>auth_admin()</code> .	auth.php
RNF09	Segurança	URLs de redirecionamento pós-login devem ser validadas para aceitar apenas caminhos internos, prevenindo redirecionamento aberto (open redirect).	login.php
RNF10	Segurança	Saídas dinâmicas em HTML devem usar <code>htmlspecialchars()</code> com <code>ENT_QUOTES</code> para prevenir Cross-Site Scripting (XSS).	Todas as views

3.3 Usabilidade

ID	Categoria	Descrição	Referência
RNF11	Usabilidade	A interface deve seguir os princípios do Material Design 3, utilizando tokens CSS (variáveis) para cores, sombras e raios de borda, garantindo consistência visual em todos os módulos.	style.css
RNF12	Usabilidade	A navegação lateral deve ter altura fixa (100vh) com <code>overflow-y: auto</code> na área de itens, permitindo rolagem do menu quando o conteúdo exceder a altura da tela sem necessidade de reduzir o zoom.	style.css (.nav-drawer)
RNF13	Usabilidade	A área de conteúdo principal (.page-content) deve ter <code>overflow-y: auto</code> , permitindo rolagem independente da navegação lateral em todas as resoluções.	style.css (.page-content)
RNF14	Usabilidade	Todas as ações destrutivas (exclusão de livros, estudantes, funcionários) devem exigir confirmação em dialog modal antes de serem executadas.	app.js (confirmDialog)
RNF15	Usabilidade	O sistema deve exibir feedback visual imediato para todas as operações de CRUD através de mensagens coloridas (success/error) com ícone correspondente.	Todas as páginas de CRUD

ID	Categoria	Descrição	Referência
RNF16	Usabilidade	A interface deve ser responsiva; em telas menores que 900px a navegação lateral deve recolher-se e ser acessível via botão de menu (hambúrguer).	style.css + app.js

3.4 Manutenibilidade

ID	Categoria	Descrição	Referência
RNF17	Manutenibilidade	O código deve seguir estrutura modular com separação clara de responsabilidades: includes/ (helpers e layout), pages/ (controllers + views) e assets/ (CSS e JS).	Estrutura de arquivos
RNF18	Manutenibilidade	Atualizações de schema de banco de dados devem ser entregues em arquivos de migração versionados e incrementais (migracao_v1.x.sql), sem necessidade de recriar o banco completo.	migracao_v1.x.sql
RNF19	Manutenibilidade	Toda lógica de conexão ao banco deve estar centralizada em includes/db.php, usando PDO com ERRMODE_EXCEPTION e charset UTF-8mb4.	db.php
RNF20	Manutenibilidade	O cabeçalho (header.php) e rodapé (footer.php) devem ser compartilhados por todas as páginas, centralizando navegação, autenticação, notificações e inclusão de assets.	header.php + footer.php

3.5 Portabilidade e Compatibilidade

ID	Categoria	Descrição	Referência
RNF21	Portabilidade	O sistema deve operar corretamente no ambiente MAMP (Mac e Windows) sem necessidade de configuração adicional além da importação do banco de dados e ajuste de credenciais em db.php.	db.php + LEIA-ME.md
RNF22	Portabilidade	O sistema deve ser compatível com PHP 7.4 ou superior e MySQL 5.7 ou superior.	Requisito de ambiente
RNF23	Portabilidade	O sistema deve funcionar corretamente nos navegadores modernos: Google Chrome,	Front-end

ID	Categoria	Descrição	Referência
		Mozilla Firefox, Microsoft Edge e Safari (versões dos últimos 2 anos).	
RNF24	Portabilidade	O sistema não deve depender de frameworks PHP externos (Laravel, Symfony, etc.) nem de gestores de pacotes PHP (Composer); todo o código deve ser procedural puro.	Restrição de projeto

3.6 Confiabilidade

ID	Categoria	Descrição	Referência
RNF25	Confiabilidade	Operações que afetam múltiplas tabelas simultaneamente (registro de empréstimo e decremento de estoque; devolução e incremento de estoque) devem ser executadas em transações PDO (beginTransaction / commit / rollback).	emprestimos.php + devolucoes.php
RNF26	Confiabilidade	O sistema deve atualizar automaticamente o status de empréstimos vencidos para atrasado em toda requisição às páginas de empréstimos, devoluções e dashboard, garantindo consistência dos dados.	Múltiplas páginas
RNF27	Confiabilidade	Chaves estrangeiras (FK) devem ser definidas com ON DELETE RESTRICT para impedir remoção de livros ou estudantes com empréstimos vinculados, preservando a integridade referencial no banco de dados.	database.sql

3.7 Internacionalização

ID	Categoria	Descrição	Referência
RNF28	Internacionalização	Toda a interface, mensagens de erro e labels devem estar em Português do Brasil (pt-BR).	Todas as views
RNF29	Internacionalização	O banco de dados deve utilizar charset utf8mb4 com collation utf8mb4_unicode_ci para suporte completo a caracteres especiais e acentuação.	database.sql
RNF30	Internacionalização	Datas devem ser armazenadas no formato ISO (YYYY-MM-DD) e exibidas no formato	functions.php

ID	Categoria	Descrição	Referência
		brasileiro (DD/MM/YYYY) via função formatDate().	

3.8 Impressão e Exportação

ID	Categoria	Descrição	Referência
RNF31	Exportação	As páginas de exportação PDF (pdf_relatorio.php e pdf_historico.php) devem renderizar layout otimizado para impressão, ocultando controles de interface via CSS @media print.	pdf_layout.php
RNF32	Exportação	Os PDFs gerados devem incluir cabeçalho com nome da instituição e biblioteca (lidos das configurações), tabelas zebreadas, chips de status coloridos e rodapé com texto configurável.	pdf_layout.php
RNF33	Exportação	O botão "Imprimir / Salvar PDF" nas páginas de exportação deve acionar window.print(), permitindo salvar como PDF pelo diálogo nativo do navegador.	pdf_layout.php

4. Resumo Quantitativo

Categoria	Total	Alta	Média	Baixa
Requisitos Funcionais (RF01–RF71)	71	36	23	12
Requisitos Não Funcionais (RNF01–RNF33)	33	—	—	—
TOTAL	104	36	23	12

Distribuição dos RF por módulo:

Autenticação (RF01–RF08): 8 · Acervo (RF09–RF15): 7 · Estudantes (RF16–RF20): 5
Empréstimos (RF21–RF29): 9 · Devoluções (RF30–RF34): 5 · Relatórios (RF35–RF40): 6
Histórico (RF41–RF44): 4 · Dashboard (RF45–RF50): 6 · Notificações (RF51–RF53): 3
Busca Global (RF54–RF56): 3 · Funcionários (RF57–RF61): 5 · Perfil (RF62–RF65): 4 ·
Configurações (RF66–RF71): 6