# Documentação do Sistema de Biblioteca

## 1. Estrutura de Pastas

project\_root/  
├─ config/  
│ └─ db.php # Conexão com PostgreSQL  
├─ controller/  
│ ├─ LivroController.php # CRUD de livros  
│ ├─ LoginController.php # Autenticação e integração Google  
│ ├─ NotificacaoController.php # Envio de notificações de atrasos e reservas  
│ └─ ReservaController.php # Gerenciamento de reservas  
├─ model/  
│ ├─ Emprestimo.php # Model de empréstimos  
│ ├─ Livro.php # Model de livros  
│ ├─ Reserva.php # Model de reservas  
│ └─ Usuario.php # Model de usuários  
├─ view/  
│ ├─ livros.php # Listagem de livros  
│ ├─ livro\_form.php # Cadastro/Edição de livros  
│ ├─ login.php # Tela de login  
│ ├─ reservas.php # Listagem de reservas para usuário  
│ ├─ reservas\_admin.php # Listagem de reservas para admin/bibliotecário  
│ └─ logout.php # Logout do sistema  
├─ uploads/  
│ └─ capas/ # Armazenamento de capas de livros  
├─ assets/  
│ ├─ css/  
│ └─ js/  
└─ vendor/ # Bibliotecas externas (PHPMailer, Google OAuth)

## 2. Tabelas Principais

### Usuários

* id (PK)
* nome, sobrenome
* telefone, whatsapp
* email, senha, foto
* tipo (admin, bibliotecário, aluno, público externo)

### Livros

* id (PK)
* título, autor, editora, ano, ISBN, gênero, capa
* dados complementares (edição, páginas, idioma, sinopse, localização, palavras-chave, código de barras, ficha catalográfica)

### Empréstimos

* id (PK)
* livro\_id (FK -> livros.id)
* leitor\_id (FK -> usuarios.id)
* data\_emprestimo, data\_devolucao\_prevista, data\_devolucao\_real
* status (Emprestado, Devolvido)

### Reservas

* id (PK)
* livro\_id (FK -> livros.id)
* usuario\_id (FK -> usuarios.id)
* data\_reserva
* status (ativa, cancelada, concluida)

## 3. Fluxo de Cadastro e Edição de Livros

1. View livro\_form.php coleta dados e capa
2. Controller LivroController.php valida e chama Model Livro.php
3. Model realiza INSERT/UPDATE via PDO
4. Upload de capa salvo em uploads/capas/
5. Verificação de duplicidade de ISBN

## 4. Empréstimos

* Model Emprestimo.php gerencia empréstimos e devoluções
* Controller (a ser criado) manipula requisições CRUD
* Funções importantes: listarPorUsuario, cadastrar, devolver, listarAtivos, listarAtrasados

## 5. Reservas

* Model Reserva.php: cadastra, cancela e lista reservas
* Controller ReservaController.php: valida duplicidade e permissões de usuário
* Views reservas.php e reservas\_admin.php exibem reservas por perfil
* Notificação de reservas disponíveis via NotificacaoController.php

## 6. Autenticação e Login

* Tela login.php + Controller LoginController.php
* Suporta integração futura com Google OAuth
* Logout via logout.php
* Sessões PHP para controle de usuário logado

## 7. Notificações

* NotificacaoController.php envia e-mails de:
  + Empréstimos atrasados
  + Reservas disponíveis
* Reutiliza funções de Models Emprestimo.php e Reserva.php
* Utiliza PHPMailer configurável via SMTP

## 8. Boas Práticas e Observações

* Toda manipulação de banco de dados via PDO com prepared statements
* Check constraints em tabelas evitam dados inconsistentes (ex: status)
* Evitar duplicidade em reservas e ISBNs
* Separação clara entre Model, View e Controller
* Upload de arquivos com verificação de extensão
* Código modular facilita manutenção e futuras extensões

## 9. Extensões Futuras

* Integração completa com Google OAuth
* Relatórios avançados de empréstimos e reservas
* Dashboard administrativo com alertas e estatísticas
* Sistema de notificações via SMS ou WhatsApp