**Documentação de Desenvolvimento - QTS Trabalho Biblioteca**

**Controle de Versão**

| **Versão** | **Data** | **Autor** | **Alterações Realizadas** |
| --- | --- | --- | --- |
| 1.0 | 05/08/25 | Matheus Felipe da Silva | Criação da documentação inicial e levantamento dos requisitos. |
| 1.1 | 05/08/25 | Matheus Felipe da Silva | Criação da classe LoginDAO. |

**Sistema de Gerenciamento de Biblioteca (QTS Trabalho Biblioteca)**

**1. Requisitos Funcionais**

**1.1 Usuários**

|  | **Descrição** |
| --- | --- |
| RF001 | O sistema deve permitir o cadastro de usuários (nome, e-mail, telefone, senha). |
| RF002 | O sistema deve validar o login de usuários com e-mail e senha. |
| RF003 | O sistema deve permitir a busca de usuários por e-mail ou ID. |

**1.2 Livros**

|  | **Descrição** |
| --- | --- |
| RF004 | O sistema deve permitir o cadastro de livros (título, ISBN, autor, ano). |
| RF005 | O sistema deve permitir a consulta de livros por id, título, autor ou ISBN. |

**1.3 Segurança e Validação**

|  | **Descrição** |
| --- | --- |
| RF006 | O sistema deve sanitizar entradas de dados para evitar SQL Injection e XSS. |
| RF007 | O sistema deve armazenar senhas de forma criptografada (hash). |

**2. Requisitos Não Funcionais**

**2.1 Desempenho**

|  | **Descrição** |
| --- | --- |
| RNF001 | O sistema deve responder a requisições sem atraso. |

**2.2 Segurança**

|  | **Descrição** |
| --- | --- |
| RNF002 | O sistema deve usar conexão segura (PDO) para interação com o banco de dados. |
| RNF003 | O sistema deve validar e sanitizar todas as entradas do usuário. |

**2.3 Usabilidade**

|  | **Descrição** |
| --- | --- |
| RNF004 | O sistema deve ter mensagens claras de erro para o usuário. |

**2.4 Manutenibilidade**

|  | **Descrição** |
| --- | --- |
| RNF005 | O código deve seguir boas práticas de POO e ser modularizado. |

**3. Classes e Funcionalidades**

**3.1**Database**(**Conexao.php**)**

* **Responsabilidade**: Gerenciar a conexão com o banco de dados MySQL.
* **Métodos**:
  + getConnection(): Retorna uma conexão PDO configurada.

**3.2**Livro**(**Livro.php**)**

* **Responsabilidade**: Representar um livro no sistema.
* **Atributos**: id, titulo, isbn, autor, ano\_publicacao.
* **Métodos**: Getters e Setters.

**3.3**Usuario**(**Usuario.php**)**

* **Responsabilidade**: Representar um usuário do sistema.
* **Atributos**: id, email, nome, telefone, senha.
* **Métodos**: Getters e Setters.

**3.4**UsuarioDAO**(**UsuarioDAO.php**)**

* **Responsabilidade**: Gerenciar operações de banco de dados relacionadas a usuários.
* **Métodos**:
  + buscarPorEmail($email): Busca usuário por e-mail.
  + cadastrar($email, $senha, $nome, $telefone): Cadastra novo usuário.
  + buscarPorId($id): Busca usuário por ID.

**3.5**Sanitizacao**(**Sanitizacao.php**)**

* **Responsabilidade**: Sanitizar dados de entrada para evitar ataques.
* **Métodos**:
  + sanitizar($dado): Remove espaços, barras e converte caracteres especiais.

**3.6**LoginDAO**(**LoginDAO.php**)**

* **Responsabilidade**: Autenticar dados de login, efetuar logout.
* **Métodos**:
  + validarLogin($email, $senha): Valida credenciais de login.
  + logout(): Encerra sessão do usuário.

**4. Considerações Finais**

* **Banco de Dados**: MySQL (biblioteca\_db).
* **Tecnologias**: PHP (POO) ,HTML ,MySQL.