

Memorial Descritivo – Projeto Interdisciplinar (PI) – 3º Semestre DSM

1. Tema

O Projeto Interdisciplinar (PI) 3º semestre DSM consiste no desenvolvimento da plataforma OpenPages, um sistema web destinado ao gerenciamento de livros acadêmicos em bibliotecas universitárias. O sistema busca otimizar o acesso de estudantes e professores ao acervo, permitindo consulta, sugestão de novos títulos, empréstimos online e gestão de acervo pelos administradores. A evolução do projeto contempla a migração do banco de dados relacional (SQL) para um modelo não relacional (MongoDB), integrando novas entidades, relações e funcionalidades de login e autenticação.

2. Justificativa

No cotidiano acadêmico, o acesso rápido a livros é essencial para o acompanhamento de disciplinas, desenvolvimento de pesquisas e cumprimento de prazos de trabalhos. Entretanto, estudantes e professores frequentemente enfrentam dificuldades em verificar a disponibilidade de livros antes de se deslocarem até a biblioteca.

A implementação da plataforma OpenPages justifica-se pelos seguintes fatores:

- Eficiência: agiliza o processo de consulta e empréstimo de livros.
- Segurança: introduz login com perfis distintos para usuários e administradores.
- Usabilidade: melhora a experiência de acesso ao acervo por meio de interface intuitiva.
- Controle: possibilita à biblioteca maior organização e confiabilidade no gerenciamento de livros.

Assim, a plataforma não se limita a um simples catálogo, mas evolui para um sistema robusto de gestão de acervo acadêmico.

3. Objetivos

3.1 Objetivo Geral

Desenvolver e implantar a plataforma OpenPages em modelo NoSQL (MongoDB), com funcionalidades de autenticação, consulta e empréstimo online de livros acadêmicos, além de gerenciamento administrativo do acervo.

3.2 Objetivos Específicos

- Migrar a modelagem relacional para uma modelagem não relacional (MongoDB).

- Implementar sistema de login e autenticação com perfis distintos (usuário e administrador).
- Criar funcionalidades de visualização, sugestão e empréstimo de livros.
- Estruturar coleções e documentos no MongoDB garantindo escalabilidade e segurança.
- Estabelecer relações de associação, agregação e composição entre as entidades.

4. Entidades e Campos do Modelo NoSQL

4.1 Usuário

Representa os atores do sistema (estudantes e administradores).

Campos:

- - id: ObjectId → Identificador único.
- - nome: String → Nome completo do usuário.
- - email: String → E-mail institucional, usado como login.
- - senha: String → Senha armazenada de forma criptografada.
- - perfil: Enum("admin", "usuario") → Define o perfil do usuário.
- - curso_id: ObjectId → Referência ao curso do usuário.

Relação e Cardinalidade:

Curso 1 --- N Usuários (associação).

Justificativa: um curso pode ter vários usuários, mas cada usuário pertence a apenas um curso.

4.2 Curso

Representa os cursos vinculados à instituição.

Campos:

- - id: ObjectId → Identificador único.
- - nome: String → Nome do curso.
- - descricao: String → Breve descrição do curso.

Relação e Cardinalidade:

Curso 1 --- N Livros (agregação).

Curso 1 --- N Usuários (associação).

Justificativa: o curso agrega livros de sua área e é associado a estudantes matriculados.

4.3 Livro

Representa cada item do acervo.

Campos:

- - id: ObjectId → Identificador único.
- - titulo: String → Título da obra.
- - autor: String → Nome do autor.
- - editora: String → Nome da editora.
- - edicao: String → Edição da obra.
- - materia: String → Área ou disciplina relacionada.
- - palavras_chave: [String] → Termos que facilitam a busca.
- - curso_id: ObjectId → Referência ao curso relacionado.
- - disponibilidade: Boolean → Indica se o livro está disponível.
- - data_cadastro: Date → Data de inserção no sistema.

Relação e Cardinalidade:

Livro 1 --- N Empréstimos (composição).

Justificativa: o empréstimo só existe vinculado a um livro, logo a relação é de composição.

4.4 SugestaoDeLivro

Armazena as sugestões feitas por usuários para inclusão no acervo.

Campos:

- - id: ObjectId → Identificador único.
- - usuario_id: ObjectId → Referência ao usuário que fez a sugestão.
- - livro_sugerido: String → Nome do livro sugerido.
- - autor: String → Nome do autor.
- - editora: String → Editora da obra.
- - motivo: String → Justificativa da sugestão.
- - curso_id: ObjectId → Curso relacionado ao livro sugerido.
- - status: Enum("pendente", "aprovada", "rejeitada") → Situação da sugestão.
- - data_sugestao: Date → Data em que a sugestão foi feita.

Relação e Cardinalidade:

Usuário 1 --- N Sugestões (associação).

Justificativa: cada sugestão pertence a um usuário, mas um usuário pode sugerir vários livros.

4.5 Emprestimo

Gerencia o controle de retirada e devolução de livros.

Campos:

- - id: ObjectId → Identificador único.
- - livro_id: ObjectId → Referência ao livro emprestado.
- - usuario_id: ObjectId → Referência ao usuário que solicitou.
- - data_emprestimo: Date → Data de retirada.

- - data_devolucao_prevista: Date → Prazo estipulado para devolução.
- - data_devolucao_real: Date | null → Data real de devolução.
- - status: Enum("ativo", "concluido", "atrasado") → Estado do empréstimo.

Relação e Cardinalidade:

Usuário 1 --- N Empréstimos (associação).

Livro 1 --- N Empréstimos (composição).

Justificativa: o empréstimo depende tanto do usuário que solicita quanto do livro a ser retirado; sem livro ou usuário, não há empréstimo.