

# Bibliotex

Apresentação de Andamento  
Trabalho de Laboratório  
ACH2025 Banco de Dados II

André Palacio NUSP: 13835534 - Turma 94  
Giovanna Couto NUSP: 13729042 - Turma 94  
Luan Pinheiro NUSP: 13672471 - Turma 04  
Matheus Rocha NUSP: 11885102 - Turma 94  
Tiago Fraga NUSP: 13859638 - Turma 04

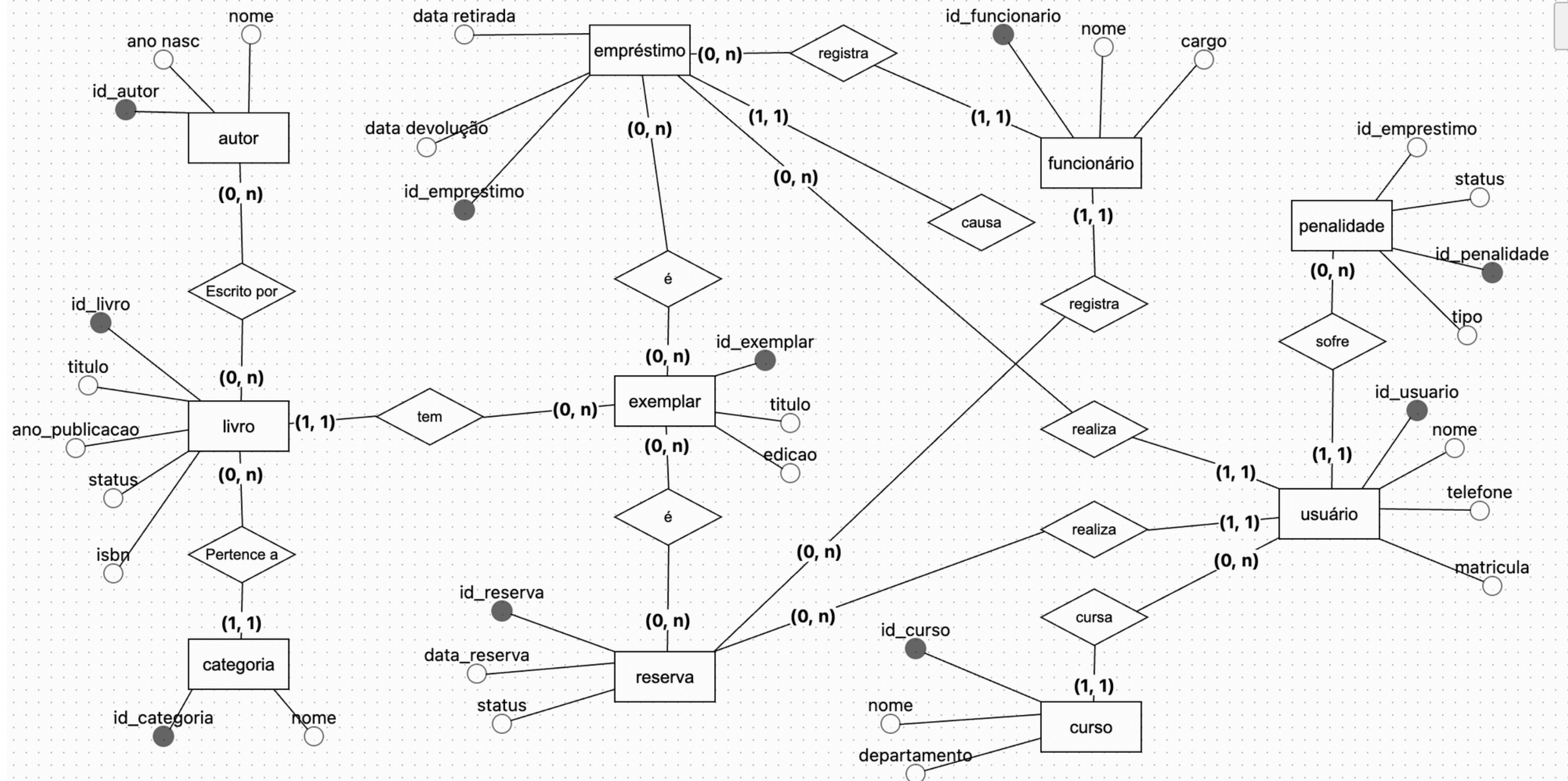
# Sistema Gerencial de Biblioteca Universitária

## Justificativa:

- Proximidade com a realidade acadêmica
- Crescente volume de acervos demanda que as operações sejam cada vez mais eficientes
- Campo com pouca modernização
- Muitas entidades a serem exploradas e demanda para otimização
- Possibilidade de exploração de queries com alto volume de operações
- Alta volumetria na gestão de livros e empréstimos

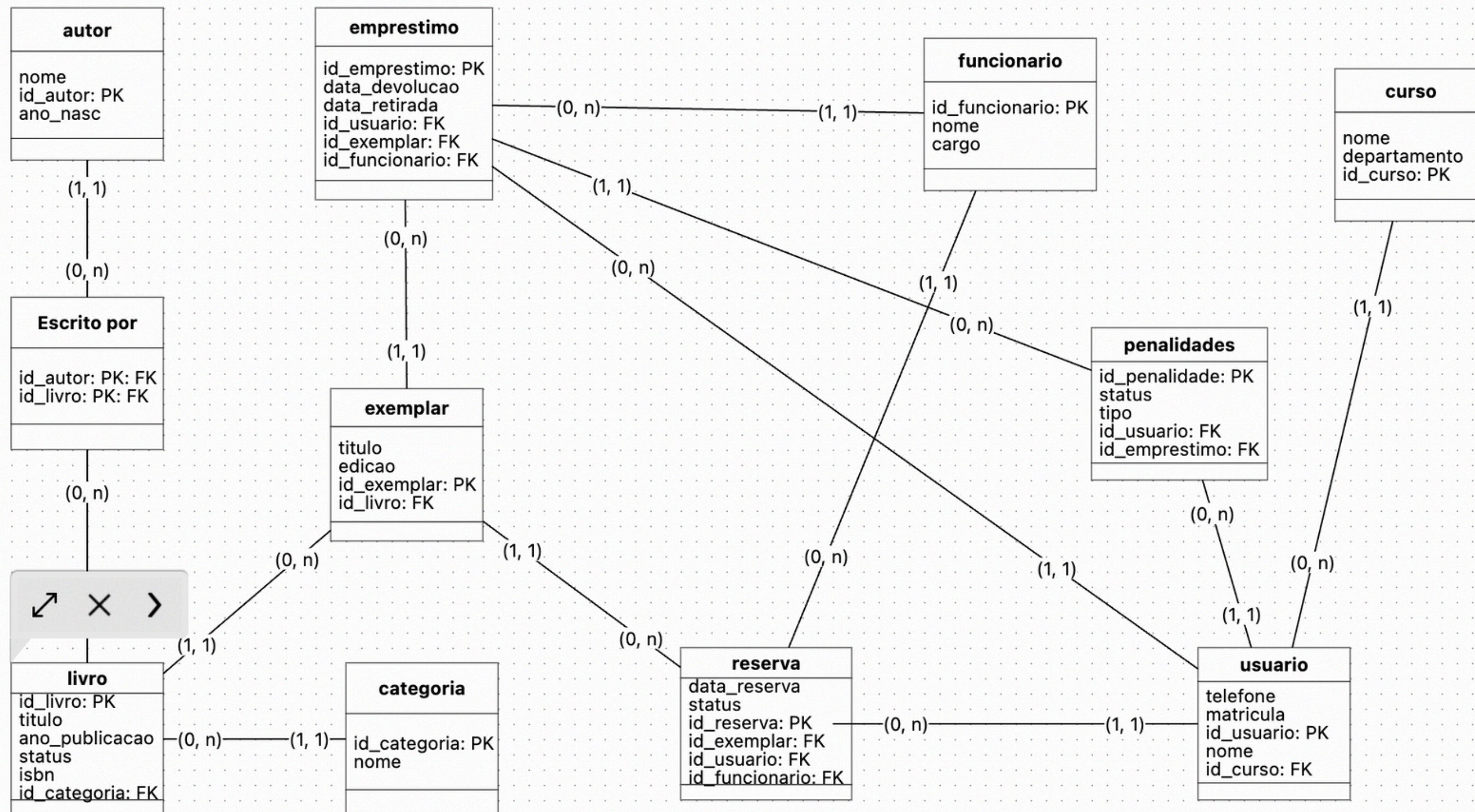


# Modelo Conceitual





# Modelo Lógico





# Operações críticas

- **Índice composto para acelerar JOIN de categoria**

```
CREATE INDEX IF NOT EXISTS idx_livro_categoria  
ON livro(id_categoria);
```

- **Índice parcial para status “disponível”**

```
CREATE INDEX IF NOT EXISTS idx_livro_status_disp  
ON livro(status)  
WHERE status = 'disponível';
```

- **Índice composto para histórico de empréstimo**

```
CREATE INDEX IF NOT EXISTS idx_emprestimo_usuario_data  
ON emprestimo(id_usuario, data_retirada DESC);
```

# Operações críticas

- **View materializada para livros disponíveis**

```
CREATE MATERIALIZED VIEW IF NOT EXISTS mv_livros_disponiveis AS
SELECT
  l.id_livro,
  l.titulo,
  c.nome AS categoria
FROM livro l
JOIN categoria c ON l.id_categoria = c.id_categoria
WHERE l.status = 'disponível'
WITH DATA;
```

- **Índice na view para acelerar buscas por título**

```
CREATE INDEX IF NOT EXISTS idx_mv_livros_disp_titulo
ON mv_livros_disponiveis(titulo);
```

# Preenchimento dos dados

## Dataset Kaggle

- <https://www.kaggle.com/datasets/saurabhbhagchi/books-dataset>
- Dataset de livros reais registrados no site da Amazon
- 271.379 livros registrados

## Dados artificiais para testes

- Desenvolvemos scripts para popular os bancos de dados em volumes controlados
- Os testes de otimizações serão feitos em tabelas com muitos registros (atualmente 100 milhões)
- Trazer insights para entender quais otimizações trouxeram maiores resultados em larga escala

# Script populando livros

```
2
3 import os
4 import psycopg2
5 from psycopg2.extras import execute_values
6 import random
7
8 # ———— CONSTANTES DE CONFIGURAÇÃO ————
9 BOOK_COUNT = 100_000_000      # Total de livros a inserir
10 BATCH_SIZE = 100_000         # Quantos inserir por transação
11 NUM_CATEGORIES = 10          # Quantas categorias pré-criar, se necessário
12
13 def main():
14     DATABASE_URL = os.getenv(
15         "DATABASE_URL",
16         "postgresql://bibliodex:bibliodex@db:5432/bibliodex_db"
17     )
18
19     conn = psycopg2.connect(DATABASE_URL)
20     cur = conn.cursor()
21
22     # Garante que existam categorias (de 1 a NUM_CATEGORIES)
23     cur.execute("SELECT id_categoria FROM categoria")
24     categorias = [row[0] for row in cur.fetchall()]
25     if not categorias:
26         for i in range(1, NUM_CATEGORIES + 1):
27             cur.execute(
28                 "INSERT INTO categoria (nome) VALUES (%s) RETURNING id_categoria;",
29                 (f"Categoria {i}",)
30             )
31             categorias.append(cur.fetchone()[0])
32     conn.commit()
33
34     statuses = ["disponível", "emprestado", "reservado"]
```



# Script populando livros

```
34     statuses = ["disponível", "emprestado", "reservado"]
35
36     for offset in range(0, BOOK_COUNT, BATCH_SIZE):
37         batch = []
38         upper = min(offset + BATCH_SIZE, BOOK_COUNT)
39         for i in range(offset, upper):
40             title = f"Livro Teste {i+1}"
41             year = random.randint(1900, 2025)
42             status = random.choice(statuses)
43             cat = random.choice(categorias)
44             batch.append((title, year, status, cat))
45
46         execute_values(
47             cur,
48             """
49             INSERT INTO livro (titulo, ano_publicacao, status, id_categoria)
50             VALUES %s
51             """,
52             batch
53         )
54         conn.commit()
55         print(f"Inserted {upper} / {BOOK_COUNT} livros")
56
57     cur.close()
58     conn.close()
59     print("População concluída.")
60
61 if __name__ == "__main__":
62     main()
```

# Tecnologias

## Back end:

- Python3
- Fast API
- PostgreSQL (visualização com DBeaver)
- Docker

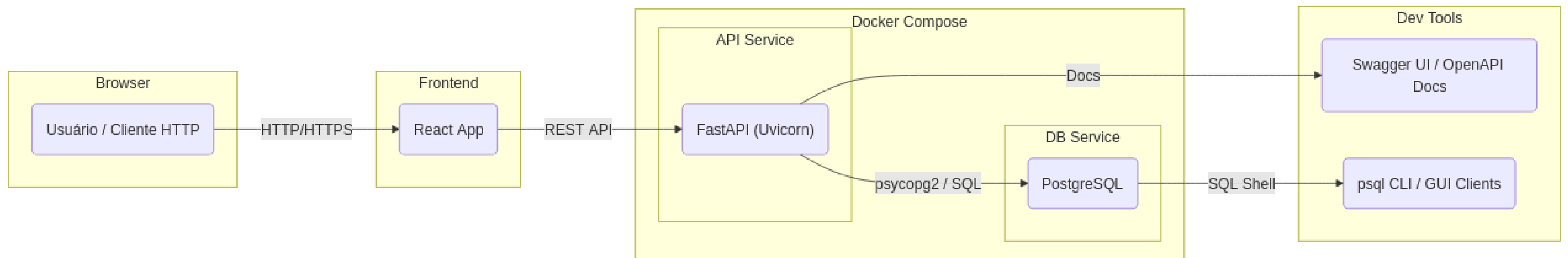


## Front end:

- React
- Next.js
- TypeScript
- TailwindCSS




# Arquitetura





# Front-end Atual

 **Bibliodex**  
Sistema de Gerenciamento de Biblioteca Universitária

[Home](#) [Usuários](#) [Empréstimos](#) [Reservas](#) [Devoluções](#) [Relatórios](#) [Administração](#) [Login](#)

### Consulta de Livros

**O Senhor dos Anéis**  
Autor: J.R.R. Tolkien  
Categoria: Fantasia  
[Emprestar](#) [Reservar](#)


**1984**  
Autor: George Orwell  
Categoria: Distopia  
[Emprestar](#) [Reservar](#)

**Dom Quixote**  
Autor: Miguel de Cervantes  
Categoria: Clássico  
[Emprestar](#) [Reservar](#)

**A Revolução dos Bichos**  
Autor: George Orwell  
Categoria: Sátira Política  
[Emprestar](#) [Reservar](#)

**O Pequeno Príncipe**  
Autor: Antoine de Saint-Exupéry  
Categoria: Infantil  
[Emprestar](#) [Reservar](#)

**Orgulho e Preconceito**  
Autor: Jane Austen  
Categoria: Romance  
[Emprestar](#) [Reservar](#)

 **Bibliodex**  
Sistema de Gerenciamento de Biblioteca Universitária


[Home](#) [Usuários](#) [Empréstimos](#) [Reservas](#) [Devoluções](#) [Relatórios](#) [Administração](#) [Login](#)

### Consulta de Livros

**1984**  
Autor: George Orwell  
Categoria: Distopia  
[Emprestar](#) [Reservar](#)

**A Revolução dos Bichos**  
Autor: George Orwell  
Categoria: Sátira Política  
[Emprestar](#) [Reservar](#)

# Front-end Atual



**Bibliodex**  
Sistema de Gerenciamento de Biblioteca Universitária

Home

Usuários

Empréstimos

Reservas

Devoluções

Relatórios

Administração

Login

### Gestão de Empréstimos

Buscar usuário


Buscar livro

Filtrar por autor

Filtrar por categoria

Filtrar por usuário


Filtrar por livro

Todos os status 

Livro	Usuário	Data de Empréstimo	Data de Devolução	Status	Ações
Livro A (Autor A)	Usuário A (Aluno)	2023-10-01	—	<div>Em andamento</div>	<div>Detalhes</div> <div>Registrar Devolução</div>
Livro B (Autor B)	Usuário B (Professor)	2023-09-15	2023-09-30	<div>Devolvido</div>	<div>Detalhes</div>

© 2025 Bibliodex - Projeto BDII USP

# Front-end Atual



**Bibliodex**  
Sistema de Gerenciamento de Biblioteca Universitária

Home

Usuários

Empréstimos

Reservas

Devoluções

Relatórios

Administração

Login

### Gestão de Reservas

Filtrar por usuário

Filtrar por livro

Filtrar por autor

Filtrar por categoria

Todos os status ▾

Nova Reserva

Livro	Usuário	Data da Reserva	Status	Ações
Livro A (Autor A)	Usuário A (Aluno)	2023-10-01	Ativa	Cancelar
Livro B (Autor B)	Usuário B (Professor)	2023-09-15	Concluída	

© 2025 Bibliodex - Projeto BDII USP



# API Atual

## Bibliodex API 0.1.0 OAS 3.1

[/openapi.json](#)

### Livros

GET	/livros/	Listar Livros	✓
POST	/livros/	Criar Livro	✓
GET	/livros/{livro_id}	Obter Livro	✓
DELETE	/livros/{livro_id}	Excluir Livro	✓

### Categorias

GET	/categorias/	Listar Categorias	✓
POST	/categorias/	Criar Categoria	✓
GET	/categorias/{categoria_id}	Obter Categoria	✓
DELETE	/categorias/{categoria_id}	Excluir Categoria	✓

### default

GET	/	Root	✓
-----	---	------	---

# Uso de Inteligência Artificial Generativa

## Bons momentos:

- Utilizamos o GPT-3/4, alternando para o o4-mini-high, que diz ser melhor para código
- Tentamos manter ao máximo a IA em controle do desenvolvimento (“vibe coding”)
- Aplicar práticas de Prompt Engineering foi essencial conforme a codebase aumentava
- O setup inicial do projeto foi onde ela se mostrou mais valiosa
- Auxílio forte em tarefas além do código, como instalar bibliotecas e dependências
- Questionar a sua própria opinião sobre uma decisão demonstrou uma eficiente forma de validação

# Uso de Inteligência Artificial Generativa

## Maus momentos:

- Constante alteração de lógicas que não necessariamente impactavam o que foi pedido
- Pouco cuidado com edge-cases
- Precisão razoável em diagnósticos de bugs, mas com muitas sugestões sem sentido
- Mudanças de opinião baseadas no prompt
- Baixa confiança em terceirizar decisões totalmente para a IA



# Obrigado!

