



UNIVERSIDADE DE SÃO PAULO

APRESENTAÇÃO FINAL

BANCO DE DADOS II

Turma 04

Dimitri Prado 14746022

Drielly de Moraes Guerreiro 14746040

Gabriela Carvalho Vitorino 14749616

Júlia Du Bois Araújo Silva 14584360

Rodrigo Lyusei Suguimoto 14670877



SISTEMA DE GERENCIAMENTO DE UMA

ACADEMIA DE DANÇA



ACADEMIA DE DANÇA HAVE DREAMS

PARTICIPAÇÃO INDIVIDUAL

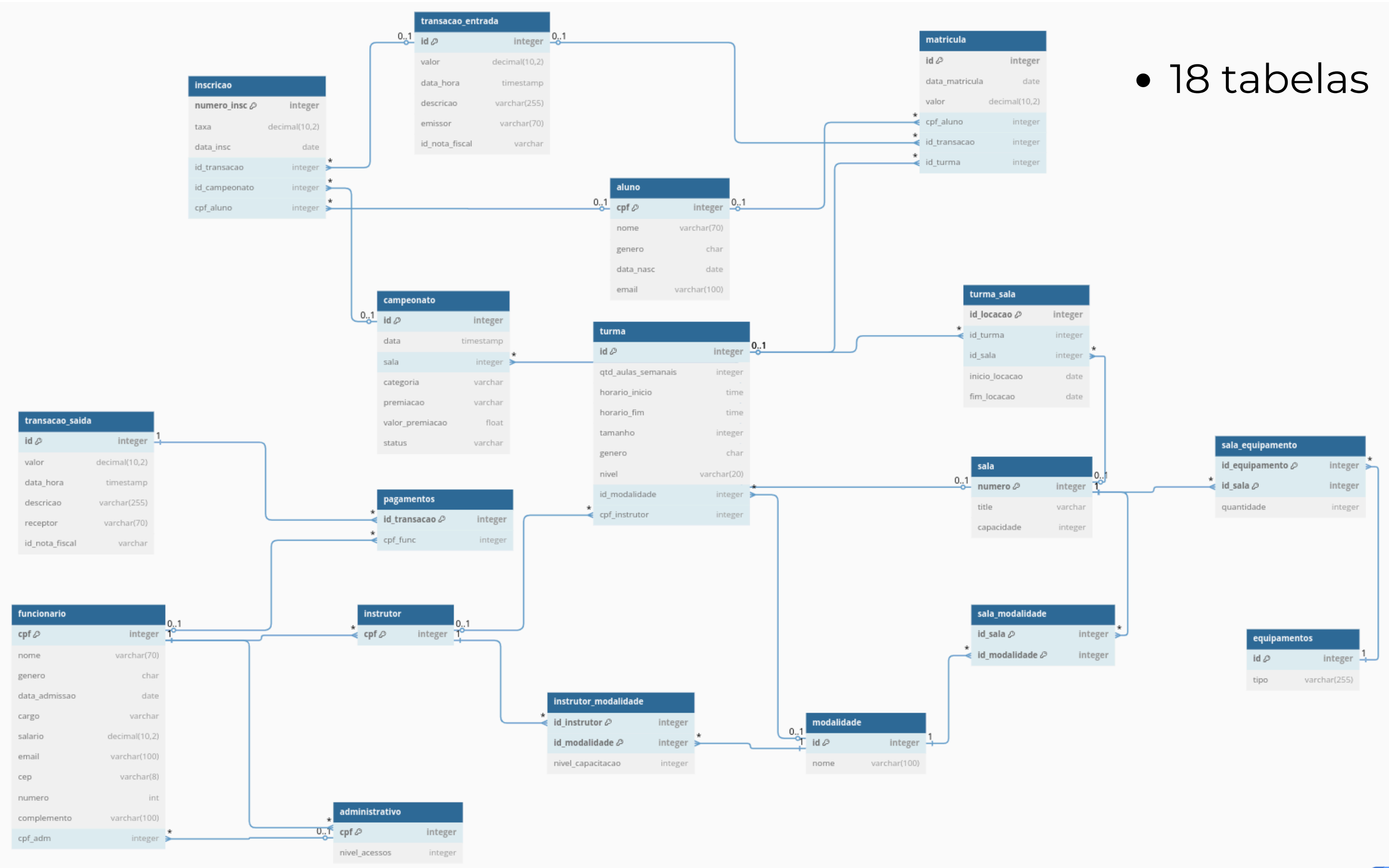
- Dimitri: modelagem do banco de dados, banco de dados físico, desenvolvimento do frontend, deploy da implementação.
- Drielly: modelagem do banco de dados, banco de dados físico e população do banco de dados.
- Gabriela: modelagem do banco de dados, slides de andamento, métricas de otimização, suporte na população do banco de dados.
- Júlia: modelagem do banco de dados, revisão do modelo lógico do BD, mapeamento e criação dos CRUD's das entidades.
- Rodrigo: modelagem do banco de dados, desenvolvimento do frontend para o CRUD.

MOTIVAÇÃO

- Busca por um sistema que necessita de recorrentes inserções, remoções consultas e atualizações, gerando um banco de dados dinâmico.
- Criação de um sistema escalável.
- Escolha de um sistema que pudesse ter aplicações reais.
- Escolha de um sistema o qual o grupo tivesse conhecimentos de negócio suficiente para saber dos requisitos funcionais do mesmo, para criar uma base de dados confiável

RELEMBRANDO O BANCO DE DADOS

- 18 tabelas



Tables (18)

- > administrativo
- > aluno
- > campeonato
- > equipamentos
- > funcionario
- > inscricao
- > instrutor
- > instrutor_modalidade
- > matricula
- > modalidade
- > pagamentos
- > sala
- > sala Equipamento
- > sala_modalidade
- > transacao_entrada
- > transacao_saida
- > turma
- > turma_sala

BD FÍSICO E POPULAÇÃO

O Banco de Dados físico foi implementado no SGBD PostgreSQL

Tabela: Matriculas

	Id [PK] integer	data_matricula date	valor numeric (10,2)	cpf_aluno character varying (11)	id_transacao integer	id_turma integer
1	1	2025-01-10	200.00	100000000001	1	1
2	2	2025-01-11	200.00	100000000002	2	2
3	3	2025-01-12	200.00	100000000003	3	3
4	4	2025-01-13	200.00	100000000004	4	4
5	5	2025-01-14	200.00	100000000005	5	5

Tabela: Aluno

cpf [PK] character varying (11)	nome character varying (70)	genero character (1)	data_nasc date	email character varying (100)
100000000001	João Pereira	M	2000-01-15	joao.pereira@email.com
100000000002	Maria Oliveira	F	1999-05-23	maria.oliveira@email.com
100000000003	Pedro Santos	M	2001-08-12	pedro.santos@email.com
100000000004	Ana Costa	F	2002-02-20	ana.costa@email.com
100000000005	Lucas Almeida	M	1998-11-30	lucas.almeida@email.com

ESTRATÉGIAS DE OTIMIZAÇÃO

Criação de índice em CPF na tabela Matricula, tendo em vista as frequentes buscas por matriculas de alunos com base em seus CPFs

	QUERY PLAN text
1	Seq Scan on aluno (cost=0.00..1.62
2	Filter: ((cpf)::text = '100000000008'::
3	Rows Removed by Filter: 49
4	Planning Time: 0.102 ms
5	Execution Time: 0.058 ms

Sem índice

	QUERY PLAN text
1	Seq Scan on aluno (cost=0.00..
2	Filter: ((cpf)::text = '100000000
3	Rows Removed by Filter: 49
4	Planning Time: 0.217 ms
5	Execution Time: 0.037 ms

Com índice

**36,21% mais rápida
em termos de
tempo de execução**

ESTRATÉGIAS DE OTIMIZAÇÃO

Criação de índice em data_admissão na tabela funcionario, tendo em vista que o controle de funcionarios é frequentemente acessado pela administração da empresa.

	QUERY PLAN text
1	Seq Scan on funcionario (cost=
2	Filter: (data_admissao = '2020
3	Rows Removed by Filter: 49
4	Planning Time: 0.356 ms
5	Execution Time: 0.136 ms

Sem índice

	QUERY PLAN text
1	Seq Scan on funcionario (cos
2	Filter: (data_admissao = '202
3	Rows Removed by Filter: 49
4	Planning Time: 0.335 ms
5	Execution Time: 0.106 ms

Com índice

**22% mais rápida em
termos de tempo
de execução**

ESTRATÉGIAS DE OTIMIZAÇÃO

Outra estratégia de **otimização** considerada para o sistema da escola de dança foi o **particionamento** da tabela matricula por data de matrícula. Essa técnica seria útil para melhorar o desempenho das consultas, especialmente com o crescimento contínuo do número de alunos ao longo dos meses e anos. Com o particionamento, o banco acessaria apenas os dados relevantes em cada busca, tornando as operações mais eficientes. No entanto, a estratégia não foi implementada neste momento, já que o volume atual de dados ainda é pequeno e não justifica a complexidade adicional.

IMPLEMENTAÇÃO

- Back end: Java SpringBoot utilizando Jakarta Persistence para montar API's CRUD para as tabelas
- Front end: Angular
- Deploy: foram feitas imagens do Docker de cada uma das aplicações e foram executadas na nuvem (AWS) usando a ferramenta docker-compose.

DEMONSTRAÇÃO DO SISTEMA



[Demonstração]

USO DA IAG

- Utilizamos a IAG em diversas etapas do desenvolvimento do projeto
- Suporte nos estudos sobre otimização, população do banco de dados
- Suporte na implementação do sistema, tanto no backend quanto no frontend
 - Criação de um Gemini Gem para auxiliar na criação dos componentes
- Pontos positivos: otimização de tempo em tarefas repetitivas
- Pontos negativos: quando utilizado para auxiliar nas etapas de desenvolvimento de código, erros cometidos atrapalhavam e custavam mais tempo



OBRIGADO PELA ATENÇÃO!

Turma 04

Dimitri Prado 14746022

Drielly de Moraes Guerreiro 14746040

Gabriela Carvalho Vitorino 14749616

Júlia Du Bois Araújo Silva 14584360

Rodrigo Lyusei Suguimoto 14670877