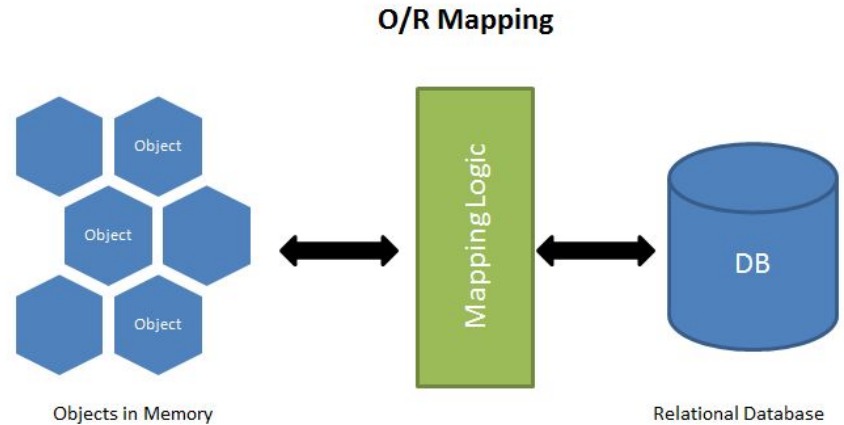


# Mapeamento objeto-relacional

*Análise de suas  
capacidades e  
limitações*

Alexandre Kenji Okamoto - 11208371  
Fernanda Cavalcante Nascimento - 11390827  
Gabriel Felix de Souza Lopes - 11295682  
Guilherme Mutschele Sena - 11208304  
Lucas Tatsuo Nishida - 11208270  
Mirela Mei - 11208392

# Análise do modelo lógico



- Como o ORM representa 1 para N?

```
@OneToMany
private List<Obra> pinturas;
```

- Como o ORM representa N para N?

```
@ManyToMany
private List<Obra> pinturas;
```

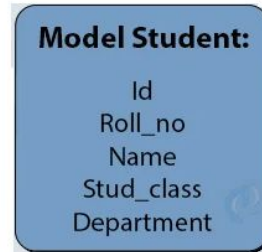
- Se o ORM é capaz de representar herança, como funciona?

```
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
```

```
create table pinturas (id bigint not null, suporte varchar(255), tipoTinta varchar(255),
primary key (id))
```

# Análise qualitativa

## Django ORM MODEL



Creation  
& storage of  
data

id	roll_no	name	stud_class	department
1	010	DataFlair	CS	Computer Science
2	020	DataFlair	CS	Computer Science
3	030	DataFlair	Django Tutorial	Python Language

Table & Database

- As queries geradas são expressivas?

**CREATE TABLE:**

```
create table artista (nome varchar(255) not null, dataMorte timestamp, dataNasc timestamp, descricao varchar(255), estiloPrincipal varchar(255), paisOrig varchar(255), periodoArt varchar(255), primary key (nome))
```

```
create table artista_Obra (Artista_nome varchar(255) not null, pinturas_id bigint not null)
```

```
create table esculturas (id bigint not null, altura integer, material varchar(255), peso integer, primary key (id))
```

```
create table exposicoes (id integer not null, dataFinal timestamp, dataInicio timestamp, nomeExposicao varchar(255), primary key (id))
```

```
create table exposicoes_Obra (Exposicoes_id integer not null, pinturas_id bigint not null)
```

```
create table pinturas (id bigint not null, suporte varchar(255), tipoTinta varchar(255), primary key (id))
```

**ALTER TABLE:**

```
alter table exposicoes_Obra add constraint UK_ipqyiafj395f2bl3nigixqga6 unique (pinturas_id)
```

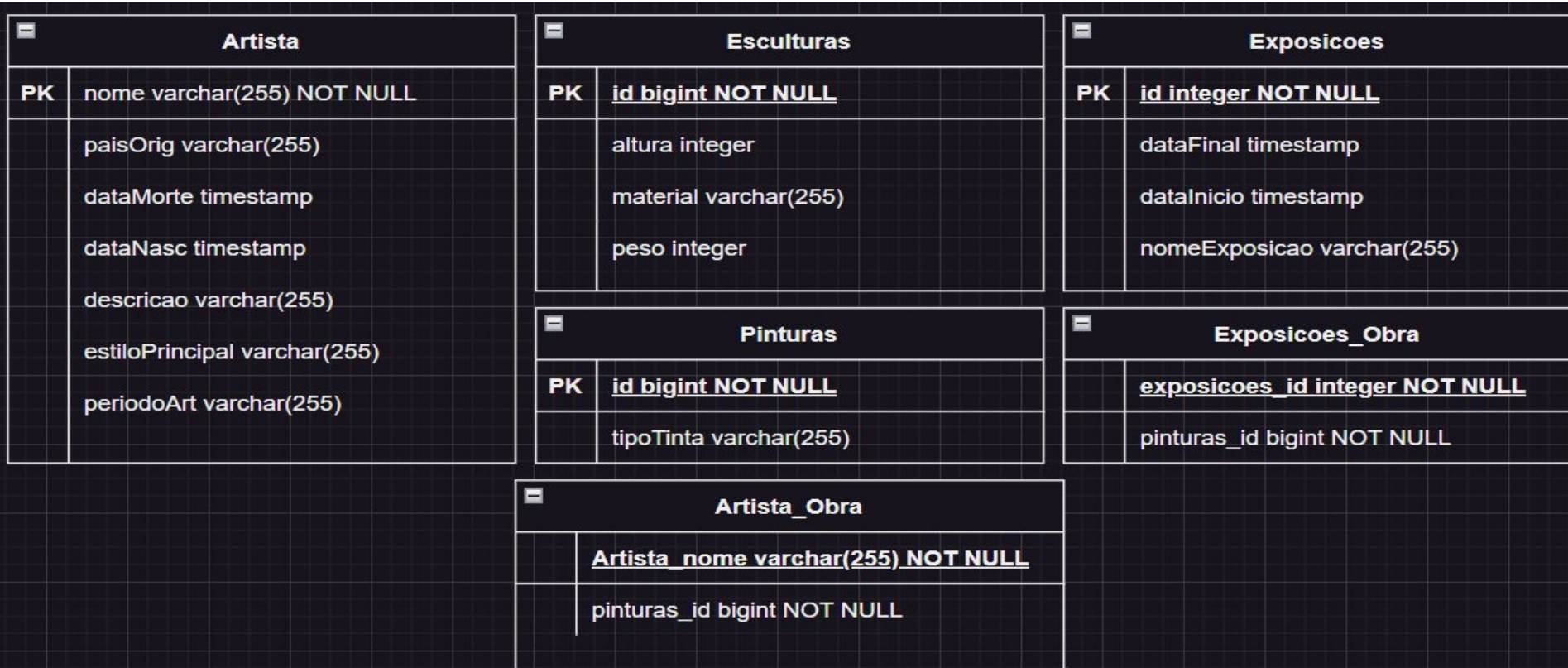
```
alter table artista_Obra add constraint FKg4yyq3g4xvq091dg2906rcatc foreign key (Artista_nome) references artista
```

```
alter table exposicoes_Obra add constraint FKpnm7efxyfjy45am89lxtj0nwn foreign key (Exposicoes_id) references exposicoes
```

**INSERT:**

```
insert into pinturas (suporte, tipoTinta, id) values (?, ?, ?)
```

- O modelo lógico é bom em termos de performance e organização?



- A experiência do desenvolvedor foi boa?
  - Configuração inicial um pouco complexa, exigindo o conhecimento de conceitos como anotações e arquivos de configuração XML.
  - A documentação oficial é extensa e abrangente, fornecendo uma variedade de exemplos e explicações detalhadas. No entanto, pode parecer um pouco técnica demais para iniciantes.
  - Reduz a quantidade de código de baixo nível para manipular operações de persistência.
  - Hibernate oferece recursos de log detalhados que podem ajudar a identificar e solucionar erros relacionados ao mapeamento objeto-relacional, consultas JPQL, configuração do Hibernate e outros aspectos.

- Quais as vantagens e desvantagens que herdamos do modelo de Objetos?

## **Vantagens**

- Abstração orientada a objetos;
- Encapsulamento e modularidade;
- Herança e polimorfismo;
- Produtividade do desenvolvedor.

## **Desvantagens**

- Complexidade do mapeamento;
- Desempenho;
- Limitações do modelo relacional;
- Necessidade de configuração e ajustes.

- Quais as vantagens e desvantagens que herdamos do modelo Relacional?

## **Vantagens**

- Estrutura organizada e consistência dos dados;
- Consultas poderosas;
- Desempenho otimizado;
- Conformidade com padrões e interoperabilidade.

## **Desvantagens**

- Complexidade de modelagem;
- Rigidez da estrutura;
- Dificuldades com objetos complexos;
- Restrições de escalabilidade horizontal.



**Obrigado!**