

Banco de Dados 2 - Relatório EP

Maria Eduarda Garcia - 11796621

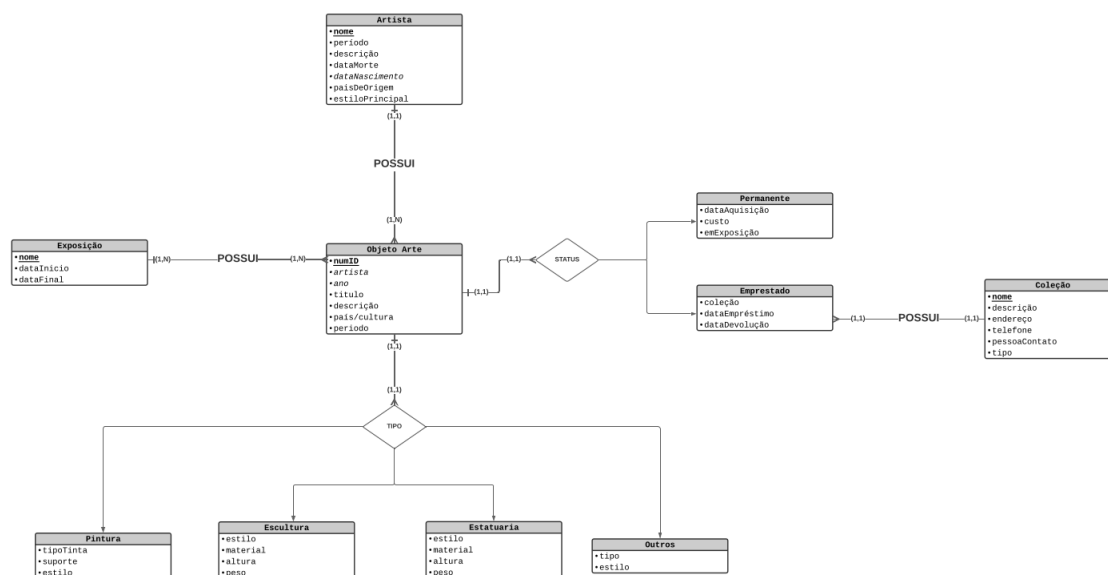
Mirela Mei - 11208392

- Projeto

O programa simula um modelo de Museu, com classes para artista, objetos de arte, exposição, tipo (pintura/escultura/outros) e status (emprestado/permanente). Por meio de uma interface no terminal, ele gera relatórios, imprime as consultas desejadas e faz o CRUD (Create, Read, Update, Delete) de todos os objetos mencionados.

Foi implementado na linguagem Java, utilizando JDBC (Java Database Connectivity), PostgreSQL, Maven e o Padrão de Projeto MVC (Model-View-Controller).

Modelo Entidade-Relacionamento:



Os atributos que estão representados em negrito/sublinhados são atributos-chave das entidades; como é o caso do nome em Exposição, Coletor e Artista, e numID em Objeto Arte. Os atributos que estão em itálico são opcionais, como artista e ano em Objeto Arte, e dataNascimento em Artista. A cardinalidade está exemplificada nas linhas, juntamente com as relações entre as entidades. A entidade Objeto Arte

possui duas especializações: TIPO, podendo ser Pintura, Escultura, Estatuaria ou Outros. E Status, podendo ser Permanente ou Emprestado.

- Estrutura

Estrutura das pastas:

1. **src:** códigos fonte Java
2. **target:** arquivos compilados
3. **pom:** dependências maven

Projeto MVC:

A utilização do padrão MVC traz como benefício isolar as regras de negócios da lógica de apresentação/interface com o usuário. Isso proporciona mais flexibilidade e oportunidades de reuso das classes, além do baixo acoplamento entre elas.

1. **connection:** realiza a conexão do programa com o SGBD PostgreSQL
2. **model:** camada que possui a lógica da aplicação, responsável pelas regras de negócios, persistência com o banco de dados e as classes de entidades
 - a. **bean:** uma classe JavaBean, também conhecida como POJO (Plain Old Java Objects), possui um construtor, atributos privados e os métodos getters e setters públicos.
 - b. **dao:** responsável por trocar informações com o SGBD e fornecer operações CRUD e de pesquisas
3. **main:** camada de visualização que representa a parte do sistema que interage com o usuário, haverá a entrada dos dados inseridos pelo usuário e também a saída de informações que serão exibidas para ele
4. **bd:** tabelas e inserções que foram utilizadas para modelar e popular o banco

O arquivo `/model/dao/ConsultasDAO.java` possui as consultas extra que foram necessárias para o EP.

- Interface

A interface gráfica possui o seguinte formato:

```

---- Consultas básicas ----
Artistas: 1 para cadastrar, 2 para listar, 3 para alterar e 4 para excluir
Objetos de arte: 5 para cadastrar, 6 para listar, 7 para alterar e 8 para excluir
Coleção: 9 para cadastrar, 10 para listar, 11 para alterar e 12 para excluir
Exposição: 13 para cadastrar, 14 para listar, 15 para alterar e 16 para excluir
Pintura: 17 para cadastrar, 18 para listar, 19 para alterar e 20 para excluir
Escultura: 21 para cadastrar, 22 para listar, 23 para alterar e 24 para excluir
Outros: 25 para cadastrar, 26 para listar, 27 para alterar e 28 para excluir
Empréstimo: 29 para cadastrar, 30 para listar, 31 para alterar e 32 para excluir
Permanente: 33 para cadastrar, 34 para listar, 35 para alterar e 36 para excluir

---- Consultas especiais ----
37 para listar objetos comprados por tipo
38 para listar objetos comprados por classe (emprestado ou próprio)
39 para listar as coleções com maiores números de empréstimo por mês e por ano
40 para listar as compras de objetos de arte por mês e por ano
41 para listar a quantidade de objetos emprestados por coleção, por mês e por ano
42 para sair

```

Nela, o usuário decide qual ação pretende realizar a partir do número pré definido para esta. As consultas básicas englobam o cadastro, a listagem, alteração e exclusão de cada uma das tabelas, conforme mencionado anteriormente. As consultas avançadas, por sua vez, se referem às consultas especificadas para o trabalho em questão. Ao pressionar 37, por exemplo, para listar os objetos comprados por tipo, tem-se a seguinte saída:

Objetos								
ID	Título	Nome do artista		Descrição	Ano de criação	Período da Arte		Pais da cultura
	Estilo	Tipo	Status	Custo				
6	Auto-Retrato	Vicent Van Gogh	PERMANENTE	600.0	1889-01-01	Modernismo	Holanda	óleo sobre tela
	PINTURA							
Objetos								
ID	Título	Nome do artista		Descrição	Ano de criação	Período da Arte		Pais da cultura
	Estilo	Tipo	Status	Custo				
7	A Noite Estrelada	Vicent Van Gogh	PERMANENTE	700.0	1889-01-01	Modernismo	Holanda	óleo sob
re tela	PINTURA							
Objetos								
ID	Título	Nome do artista		Descrição	Ano de criação	Período da Arte		Pais da cultura
	Estilo	Tipo	Status	Custo				
8	Os Girassóis	Vicent Van Gogh	PERMANENTE	800.0	1889-01-01	Modernismo	Holanda	óleo sobre tela
	PINTURA							
Objetos								
ID	Título	Nome do artista		Descrição	Ano de criação	Período da Arte		Pais da cultura
	Estilo	Tipo	Status	Custo				
9	O Grande Masturbador	Salvador Dali	PERMANENTE	900.0	1929-01-01	Surrealismo	Espanha	óleo sobre tela
	PINTURA							
Objetos								
ID	Título	Nome do artista		Descrição	Ano de criação	Período da Arte		Pais da cultura
	Estilo	Tipo	Status	Custo				
10	A Persistência da Memória	Salvador Dali	PERMANENTE	1000.0	1931-01-01	Surrealismo	Espanha	óleo sob
re tela	PINTURA							

Como pode-se observar, o sistema requisitou a entrada do parâmetro tipo, estabeleceu a conexão, buscou os dados, a fechou e os imprimiu na tela. Tem-se que, em razão do tipo escolhido ser 'pintura', todos os objetos arte tem essa característica e todos tem o status 'permanente', como exigido na

consulta. Todas as queries de todas as consultas disponíveis se encontram na pasta **dao**.

- Rodar

Para compilar/rodar, é necessário abrir o projeto Maven e fazer um Realod Project para carregar as dependências. Como o banco de dados é local, deve-se ter PostgreSQL instalado, com o mesmo nome e senha da conexão realizada em `/connection/ConnectionBD.java`, além das tabelas devidamente populadas por meio do arquivo `src/bd.sql`.

Após isso, a interação se dá pelo Main, que irá mostrar a saída com todas as operações CRUD e de consultas.

```
$ lsb_release -a
```

```
No LSB modules are available.
```

```
Distributor ID: Pop
```

```
Description: Pop!_OS 20.04 LTS
```

```
Release: 20.04
```

```
Codename: focal
```

```
$ java -version
```

```
openjdk version "11.0.15" 2022-04-19
```

```
OpenJDK RuntimeEnvironment (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1)
```

```
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1,  
mixed mode, sharing)
```