

Curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Componente Curricular: Desenvolvimento de Sistemas I

Professor: Ricardo Luis dos Santos

Aluno: \_\_\_\_\_

Data: 12.07.2023 - Assinatura: \_\_\_\_\_

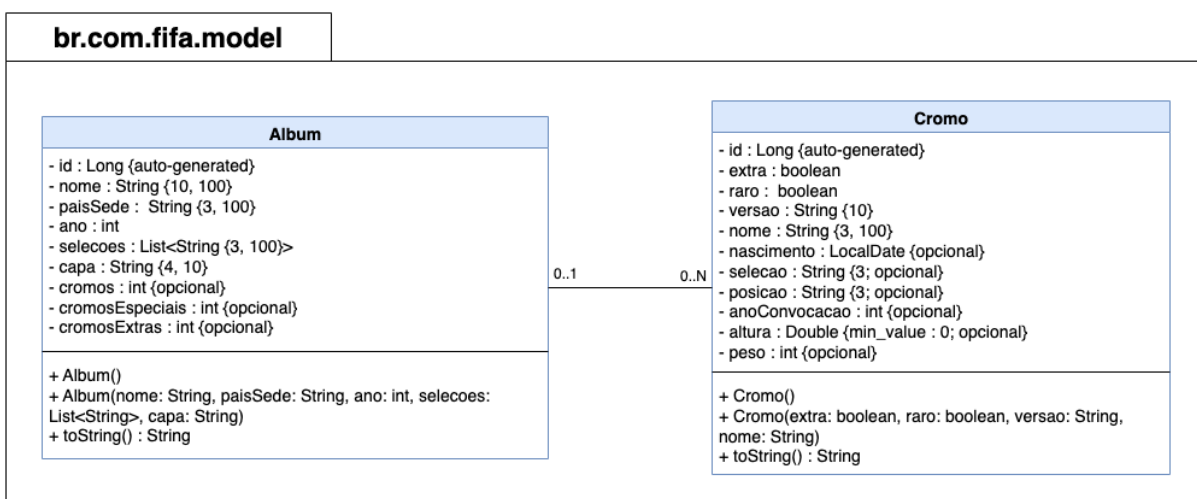
Nota:

## TRABALHO PRÁTICO

Implemente o cenário abaixo, utilizando as tecnologias de acordo com o visto em aula até o momento.

1. **(10 pontos, especificados conforme abaixo)** Considere o diagrama de classes abaixo. Crie as implementações para uma aplicação considerando o Java e o framework Spring (JPA e Hibernate), bem como todas as boas práticas vistas durante o componente curricular. Lembre-se que todas as implementações devem ser funcionais e inclua o padrão MVC em seu projeto. Ademais, considere as seguintes especificações:

**NOTE:** Ao submeter a prova no *moodle* você deverá compactar toda a pasta do projeto e submetê-la. Não esqueça de documentar classes e métodos.



Você deverá implementar o esboço de um sistema que gerencia um álbum de figurinhas colecionáveis da copa do mundo, também chamadas de cromos. No âmbito deste sistema, os modelos acima são essenciais. Você deverá implementar os modelos com as devidas restrições implementadas na entidade (utilizando as *annotations* vista durante o componente), bem como as restrições dentro dos métodos utilizados. Por exemplo, o modelo **Album** possui um atributo nome que deve ser do tipo *String* e conter no mínimo 3 e no máximo 100 caracteres. Essa restrição deve ser implementada na *annotation* utilizada e no método **setName**. Observe também que você deve implementar apenas os métodos construtores definidos na UML acima e com a devida assinatura do método (nome do método, a quantidade, ordem e tipo dos parâmetros).

- a. **(1.50 pontos)** Implementar o modelo Album
  - i. **(0.50 ponto)** Implementar os atributos com as devidas restrições detalhadas no diagrama
  - ii. **(0.50 ponto)** Implementar os métodos com as devidas restrições detalhadas no diagrama
  - iii. **(0.50 ponto)** Implementar a relação de associação com o modelo Cromo
- b. **(1.50 pontos)** Implementar o modelo Cromo
  - i. **(0.50 ponto)** Implementar os atributos com as devidas restrições detalhadas no diagrama
  - ii. **(0.50 ponto)** Implementar os métodos com as devidas restrições detalhadas no diagrama
  - iii. **(0.50 ponto)** Implementar a relação de associação com o modelo Album
- c. **(4 pontos)** Criar o *package* DAO, com um repositório genérico (disponibilizado nos slides) e implementar os repositórios específicos com os seguintes métodos de consulta
  - i. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de álbuns que **contenham** uma determinada *String nome* (a ser repassada como parâmetro) em seu atributo **nome**. O método deverá consultar as instâncias do modelo *Album* persistidos na base de dados, e retornar aquelas que **contenham** em seu nome a *string* informada e o retorno deve ser **ordenado** em ordem alfabética **crecente**.
  - ii. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de álbuns que **possuam** uma determinada *String pais* (a ser repassada como parâmetro) em seu atributo **paisSede**. O método deverá consultar as instâncias do modelo *Album* persistidos na base de dados, e retornar aquelas que **possuam** em seu atributo **paisSede** a *string* informada e o retorno deve ser **ordenado** em ordem alfabética **crecente**.
  - iii. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de álbuns que **sejam** de mais recentes que um determinado *int ano* (a ser repassado como parâmetro), para tanto considerar o atributo **ano**. O método deverá consultar as instâncias do modelo *Album* persistidos na base de dados, e retornar aquelas que sejam mais **recentes** que o **ano** passado como parâmetro.
  - iv. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de álbuns que **possuam** de mais **cromosEspeciais** que um determinado *int quantidade* (a ser repassado como parâmetro), para tanto considerar o atributo **cromosEspeciais**. O método deverá consultar as instâncias do modelo *Album* persistidos na base de dados, e retornar aquelas que possuam um **número maior** de **cromosEspeciais** que a **quantidade** passada como parâmetro.
  - v. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de cromos que **sejam** raros, ou seja, que seu atributo **raro** seja igual a **true**. O método deverá consultar as instâncias do modelo *Cromo* persistidos na base de dados, e retornar aquelas que **sejam** consideradas raras.
  - vi. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de cromos que **possuam** uma determinada *String selecao* (a ser repassada como parâmetro)

em seu atributo **selecao**. O método deverá consultar as instâncias do modelo *Cromo* persistidos na base de dados, e retornar aquelas que **possuam** em seu atributo **selecao** a *string* informada como parâmetro e o retorno deve ser **ordenado** em ordem alfabética **decrecente**.

- vii. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de cromos que **sejam** de uma determinada *String posicao* (a ser repassada como parâmetro), utilizar para tanto o atributo **posicao**. O método deverá consultar as instâncias do modelo *Cromo* persistidos na base de dados, e retornar aquelas que **possuam** em seu atributo **posicao** a *string* informada como parâmetro e o retorno deve ser **ordenado** em ordem alfabética **decrecente**.
  - viii. **(0.5 ponto)** Implementar uma consulta que retorne uma lista de cromos que **sejam** de uma determinada *String posicao* e de uma determinada *String selecao* (ambas a serem repassadas como parâmetros), utilizar para tanto os atributos **posição** e **selecao**. O método deverá consultar as instâncias do modelo *Cromo* persistidos na base de dados, e retornar aquelas que **possuam** em seus atributos **posição** e **selecao** as *strings* informadas como parâmetros e o retorno deve ser **ordenado** em ordem alfabética **decrecente** de **selecao**.
- d. **(1.5 pontos)** Implementar os seguintes **métodos de consulta** utilizando para tanto queries JPQL
- i. **(0.50 ponto)** Implementar o método **getCromosByAlbum** no DAO do modelo *Cromo*, o qual ao receber um *Album* album, deverá retornar uma lista com todas as instâncias do modelo *Cromos* associados a este **album** que estão persistidos na base de dados. É obrigatório utilizar JPQL para implementar esta consulta.
  - ii. **(0.50 ponto)** Implementar o método **getCromosByAlbumAndSelecao** no DAO do modelo *Cromo*, o qual ao receber um *Album* album e uma *String selecao*, deverá retornar uma lista com todas as instâncias do modelo *Cromos* associados a este **album** que estão persistidos na base de dados e que sejam da **selecao** informada como parâmetro na consulta. É obrigatório utilizar JPQL para implementar esta consulta.
  - iii. **(0.50 ponto)** Implementar o método **getAlbumByCromo** no DAO do modelo *Album*, o qual ao receber um *Cromo* cromos, deverá retornar a instância do modelo *Album* associada a este **cromos** que esteja persistida na base de dados. É obrigatório utilizar JPQL para implementar esta consulta.
- e. **(1.5 pontos)** Implementar um *script* para testar a aplicação, pode ser implementado utilizando um método **CommandLineRunner** (como visto em aula)
- i. **(0.25 ponto)** Criar pelo menos 2 álbuns e persisti-los na base de dados
  - ii. **(0.25 ponto)** Criar pelo menos 10 cromos e persisti-los na base de dados
  - iii. **(0.25 ponto)** Fazer os relacionamentos entre todos os objetos criados
  - iv. **(0.25 ponto)** Imprimir os objetos persistidos na base de dados tanto para o modelo Album quanto para o modelo Cromo
  - v. **(0.25 ponto)** Utilizar uma das consultas desenvolvidas no item c e imprimir os resultados
  - vi. **(0.25 ponto)** Utilizar uma das consultas desenvolvidas no item d e imprimir os resultados