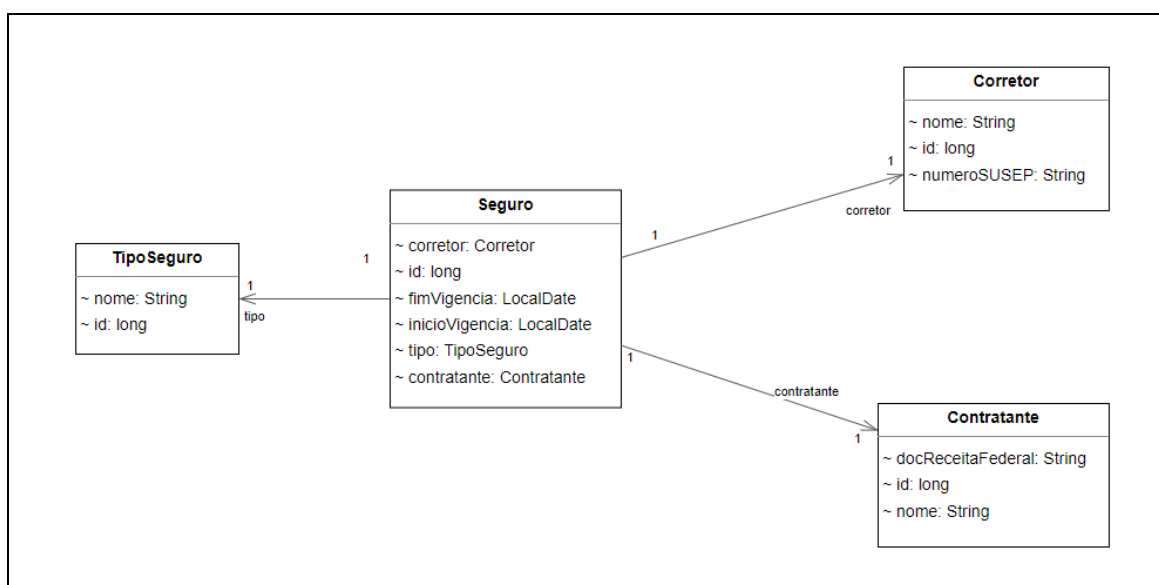


FIAP		JAVA
ALUNO:		TURMA:
PROFESSOR: Benefrancis do Nascimento		
CHECKPOINT I		VALE 10 PONTOS

A Holding Benezinho deseja entrar no concorrido mercado de seguros, para tal está desenvolvendo um revolucionário sistema para cadastramento de seguros.

Contratamos você e uma equipe de Arquitetos de Software para projetarem e construir um sistema capaz de registrar os seguros contratados no banco de dados.

Na sprint atual, você foi incumbido de fazer o Mapeamento Objeto Relacional das primeiras classes envolvidas neste projeto de software, da criação automatizada das tabelas no banco de dados Oracle, da persistência dos dados, e da criação de dois métodos capazes de realizar consultas aos dados persistidos previamente. Veja o Diagrama de Classes abaixo (as classes já foram criadas):



Importe o projeto do github :

<https://github.com/Benefrancis/seguro-holding-benezinho.git>

Caso o github esteja indisponível, você deverá pegar o projeto no diretório compartilhado.

Você deverá:

- (0,5 Ponto)** acessar o arquivo persistence.xml e alterar as configurações da **persistence-unit** para que seja possível conectar-se ao banco de dados Oracle da FIAP

com o seu usuário e senha (manter o seu usuário e senha ativo é sua responsabilidade). Não utilize o usuário e senha de outro aluno. Caso tenha problema para autenticar, comunique o professor.

- b) **(1 ponto)** na classe Main, dentro do método main, você deverá criar a instância da **EntityManagerFactory** e, em seguida, obter a **EntityManager** com as configurações da persistence-unit;
- c) **(1 Ponto)** adicionar corretamente as anotações JPA na classe **TipoSeguro**. Lembre-se de adicionar uma *constraint* para que não seja possível ter mais de um tipo de seguro com o mesmo nome na tabela do banco de dados;
- d) **(1 Ponto)** adicionar corretamente as anotações JPA na classe **Corretor**. Lembre-se de adicionar uma *constraint* para que não seja possível ter mais de um **Corretor** de seguro com o mesmo número de cadastro na SUSEP (SUSEP - Superintendência de Seguros Privados) na tabela do banco de dados;
- e) **(2 Pontos)** adicionar corretamente as anotações JPA na classe **Contratante**. Lembre-se de adicionar uma *constraint* para que não seja possível ter mais de um Contratante com o mesmo documento da Receita Federal na tabela do banco de dados;
- f) **(3 Pontos)** adicionar corretamente as anotações JPA na classe **Seguro**. Lembre-se que existe relação Muitos para Um entre:
 - a. Seguro e Contratante;
 - b. Seguro e Corretor e;
 - c. Seguro e TipoSeguro;
- g) **(0,5 Ponto)** criar um método capaz de persistir **Seguros** na correspondente tabela no banco de dados (lembre-se de cadastrar ou buscar os objetos relacionados [**Contratante, Corretor e TipoSeguro**] no SGBD para que eles estejam no estado Gerenciado - **Managed**);
- h) **(0,5 Ponto)** criar um método capaz de consultar um **Seguro** pelo seu identificador na correspondente tabela no banco de dados;
- i) **(0,5 Ponto)** criar um método capaz de consultar todos os **Seguros** na correspondente tabela no banco de dados;

A avaliação é individual e sem consulta.

Boa avaliação.