

FIAP	DOMAIN DRIVEN DESIGN
ALUNO:	TURMA:
PROFESSOR: Benefrancis do Nascimento	31/08/2022
SIMULADO	Check Point 1

Sumário

Estudo de caso	1
Modelo de dados	2
Script para criação do Banco de dados	2
Projeto Java	2
O que deverá ser feito.....	2
Critério para correção e pontuação:	4

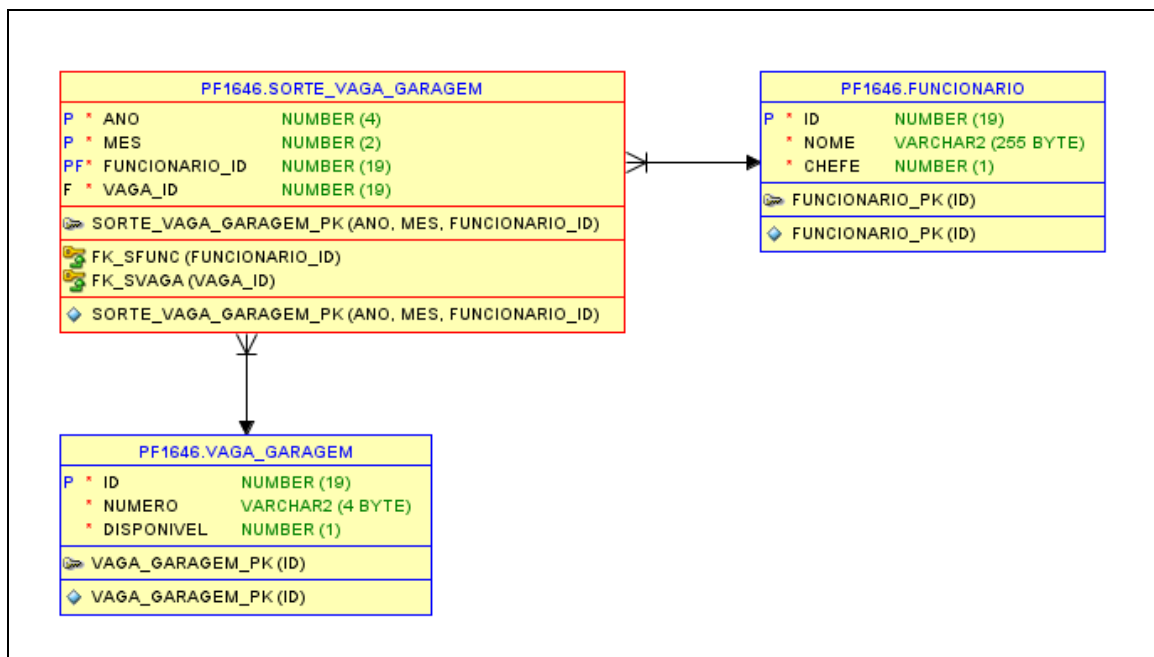
Estudo de caso

A Prefeitura do fictício Município de *Fiaplandia* possui no centro da cidade um prédio com 30 vagas de garagem. A prefeitura tem 50 funcionários neste prédio. Desses 50 funcionários 20 são do alto escalão (são chefes) e, por este motivo, têm direito a uma vaga de garagem cada um. Os outros 30 funcionários disputam as 10 vagas restantes. Entretanto, isso vem trazendo muita dor de cabeça para a administração.

No corrente mês, o prefeito decidiu que essas 10 vagas remanescentes devem ser sorteadas entre os 30 funcionários (aqueles que não são chefes) para que haja uma ocupação mais racional das vagas de garagem.

A ideia do prefeito foi muito bem recebida entre os funcionários e com isso nasceu a demanda para a equipe de TI desenvolver um programa para realização deste sorteio e gravação no banco de dados dos funcionários sorteados de cada mês.

Modelo de dados:



Script para criação do Banco de dados

Você deverá rodar o script para criação do banco de dados. Segue o link, basta rodar o script contido na pasta do banco de dados que você está utilizando. O professor poderá ajudá-lo nesta etapa:

<https://github.com/Benefrancis/fiap/tree/main/ADS/DDD/Check%20Point/A>

Projeto Java

Você deverá baixar o projeto Java para realização do simulado.

<https://github.com/Benefrancis/fiap/tree/main/ADS/DDD/Check%20Point/A/projeto>

O que deverá ser feito

O programa que você recebeu já é capaz de se conectar ao banco de dados desde que você preencha corretamente os parâmetros disponíveis no arquivo **application.properties**. Basta,

portanto, preencher os parâmetros nesse arquivo e escolher o driver JDBC correto na pasta lib do projeto para se conectar ao SGBD de sua escolha ou o que esteja disponível.

Lembre-se de:

- 1º) Baixar o projeto no link que fornecemos acima;
- 2º) Rodar o script para criação das tabelas no banco de dados;
- 2º) Adicionar ao build path do projeto o driver JDBC correspondente ao SGBD.

Você deverá atualizar o projeto para que, preferencialmente, se conecte ao banco de dados ORACLE.

O programa, por meio da classe **FuncionarioRepository** já é capaz de consultar a tabela **FUNCIONARIO** e trazer apenas aqueles que não são chefes. Cada funcionário ocupa uma posição na Lista de funcionários.

Você deverá implementar método na classe **VagaRepository** para que o programa também seja capaz de consultar a tabela **VAGA_GARAGEM** e trazer apenas as vagas disponíveis. Cada vaga deverá ocupar uma posição na Lista de vagas. Essas serão as vagas que os funcionários sorteados receberão.

Você deverá criar, no repositório para persistência dos sorteios realizados (**SorteioRepository**), um método para salvar o resultado de um sorteio na tabela **SORTE_VAGA_GARAGEM**.

No método **main**, o programa percorre a Listagem de vagas que contém vagas que estão disponíveis. E, em cada volta do looping, ele tenta inserir um registro na tabela **SORTE_VAGA_GARAGEM**, após a realização do sorteio entre os funcionários. Você deverá atualizar o método responsável pelo sorteio. Para isso: Altere o método **sortear** da classe **SorteioService** para realizar o sorteio entre os funcionários.

Atente-se ao requisito de que não poderá ter duas vagas para o mesmo funcionário no mês/Ano. Garanta que isso não ocorra no programa que você está implementando. Para isso, implemente o método **jaSorteado** na classe **SorteioRepository**.

Não se esqueça de criar comentários capazes de gerar o **JavaDoc** da aplicação e efetivamente criar os arquivos JavaDoc.

Critério para correção e pontuação:

CRITÉRIO	PONTOS
O Aluno foi capaz de inserir os parâmetros corretos no arquivo application.properties , configurou o build path para que utilize o driver JDBC correto e o programa está se conectando ao banco de dados.	1,5
Foi criado um método na classe VagaRepository para consultar as vagas de garagem disponíveis no banco de dados e, o programa está trazendo, da tabela VAGA_GARAGEM , apenas as vagas disponíveis para sorteio.	2
Foi capaz de alterar o método sortear da classe SorteioService para realizar o sorteio entre os funcionários.	0,5
O aluno foi capaz de criar um método na classe SorteioRepository para salvar o sorteio fazendo a inserção na tabela SORTE_VAGA_GARAGEM com os campos: ID da vaga; id do funcionário sorteado bem como o ano e o mês corrente.	2
O aluno foi capaz de evitar que um mesmo funcionário seja sorteado no mesmo ano e mês para mais de uma vaga implementado, para este fim, o método jaSorteado na classe SorteioRepository	1,5
O aluno criou o JAVADOC de todas as classes e métodos da aplicação	0,5

A prova é individual.

Boa prova.