

# DESAFIO DO JPQL

ALGUNS MÉTODOS PARA SEREM DESENVOLVIDO PARA O EXERCICIO 05

# QUERIES OBRIGATÓRIAS

- ▶ Buscar motorista por código. OK
- ▶ Listar todas as corridas por código de motorista. OK
- ▶ Buscar corridas a partir da data da corrida, determinando duas datas(Início e Fim). OK
- ▶ Contar veículos a partir do código do motorista. OK
- ▶ Contar as corridas do passageiro a partir do código do passageiro. OK

# Buscar motorista por código

## ► MotoristaDAO

```
Motorista buscarMotoristaPorCodigo(int codigo);
```

## ► MotoristaDAOImpl

```
@Override
public Motorista buscarMotoristaPorCodigo(int codigo) {

    return em.createQuery("from Cliente c where c.codigo = :codigo", Motorista.class)
        .setParameter("codigo", codigo).getSingleResult();

}
```

# Listar todas as corridas por código de motorista.

## ► CorridaDAO

```
List<Corrida> listarCorridasPorCodigoMotorista(int codigo);
```

## ► CorridaDAOImpl

```
@Override
public List<Corrida> listarCorridasPorCodigoMotorista(int codigo) {
    //Navegando entre atributos, para acessar o código do motorista
    return em.createQuery("from Corrida c where c.motorista.codigo = :codigoMoto", Corrida.class)
        .setParameter("codigoMoto", codigo).getResultList();
}
```

# Buscar corridas a partir da data da corrida, determinando duas datas(Inicio e Fim).

## ► CorridaDAO

```
List<Corrida> listarCorridasPorData(Calendar dataInicio, Calendar dataFim);
```

## ► CorridaDAOImpl

```
@Override
public List<Corrida> listarCorridasPorData(Calendar dataInicio, Calendar dataFim) {
    return em.createQuery("from Corrida c where c.data between :dataInicio and :dataFim", Corrida.class)
        .setParameter("dataInicio", dataInicio)
        .setParameter("dataFim", dataFim).getResultList();
}
```

# Contar veículos a partir do código do motorista

## ► VeiculoDAO

```
// A função count() dentro do Banco de Dados SEMPRE RETORNA UM tipo de dado long  
long contarVeiculosPorCodigoMotorista(int codigo);
```

## ► VeiculoDAOImpl

```
@Override  
public long contarVeiculosPorCodigoMotorista(int codigo) {  
    // O mesmo apelido que vc der para a classe vc precisa colocar entre os parentese da funcao count(),  
    // nesse caso o apelido que adicionei para classe Veiculo foi v  
    return em.createQuery("select count(v) from Veiculo v where v.motoristas.codigo = :codigoMoto", Long.class)  
        .setParameter("codigoMoto", codigo).getSingleResult();  
}
```

# Contar as corridas do passageiro a partir do código do passageiro.

## ► CorridaDAO

```
long contarCorridaPorCodigoPassageiro(int codigo);
```

## ► CorridaDAOImpl

```
@Override
public long contarCorridaPorCodigoPassageiro(int codigo) {
    //quando for retorna um dado do tipo long, apos o codigo SQL, precisa colocar a classe Long
    return em.createQuery("select count(c) from Corrida c where c.passageiro.codigo = :codigoPass", Long.class)
        .setParameter("codigoPass", codigo).getSingleResult();
}
```