



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

05 - ATIVIDADE COMPLEMENTAR (JPA E GIT)

1. Importação de Bibliotecas

- 1.1. Abra a pasta [rede]\Rogerio\PCII\lib-final e copie o conteúdo da pasta.
- 1.2. Acesse o Eclipse e navegue através do seguinte caminho: WebContent/WEB-INF.
- 1.3. Cole os JARs copiados dentro da pasta lib (crie a pasta, se necessário).

2. Mapeamento

- 2.1. Crie a classe **Endereco** abaixo dentro do pacote **model** e anote a classe com @Getter e @Setter.

Endereco
- id : Long
- logradouro : String
- numero : Long
- complemento : String
- bairro : String
- cidade : String
- estado : String
- cep : String

- 2.2. Mapear a classe com as anotações @Entity e @Table, conforme **EXEMPLO** abaixo:

```
@Entity
@Table(name="TBL_VEICULO")
public class Veiculo {
```

- 2.3. Mapear o atributo **id** como chave primária da tabela e auto-incrementável, conforme **EXEMPLO** abaixo:

```
@Id
@GeneratedValue
@Column(name="ID_VEICULO")
private Long id;
```

- 2.4. Mapear os outros atributos com a anotação @Column.
- 2.5. Salvar a classe (Ctrl + S).



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

- 2.6. Acesse o menu **Window** → **Web Browser** → **Default system web browser** ou selecione o navegador desejado para executar a aplicação.
- 2.7. Execute a página validacao.xhtml e clique no botão **Validar** e veja se o resultado é "**Conexão OK**".
- 2.8. **Delete a pasta lib** do projeto através do seguinte caminho: **WebContent/WEB-INF/**.
- 2.9. Acesse o local de gravação do seu projeto (via **Prompt de Comando**) e entre na pasta **exercicio3**.
- 2.10. Verifique a situação dos arquivos no repositório Git.

```
git status
```

- 2.11. Faça com que os arquivos sejam rastreados pelo Git.

```
git add .
```

- 2.12. Verifique a situação dos arquivos no repositório Git novamente.
- 2.13. Execute o comando para gravar as mudanças no repositório com a mensagem "**Atualização do exercício**".

```
git commit -m "Atualização do exercício"
```

- 2.14. Verifique a situação do arquivo no repositório Git novamente.

- 2.15. Envie todos os *commits* do *branch* local para o GitHub:

```
git push
```

- 2.16. Abra a página web do **Github** e verifique se o repositório foi atualizado.

- 2.17. Envie o endereço do **Git**

Para: rogeriodemoraes@etec.sp.gov.br, maria.marques34@etec.sp.gov.br

Assunto: Git – Seu nome (3Q)