



**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.*

## 15 - DAOS, SERVICES E MANAGED BEANS

### 1. ClasseDAO

- 1.1. Crie o pacote `br.com.etechoracio.training.dao`.
- 1.2. Crie a classe `AlunoDAO` no pacote recém-criado.
- 1.3. Altere de **class** para **interface** e herde de `BaseDAO`.

```
public interface AlunoDAO extends BaseDAO<Aluno>{  
}
```

- 1.4. Salve a classe ( `CTRL` + `S` ).

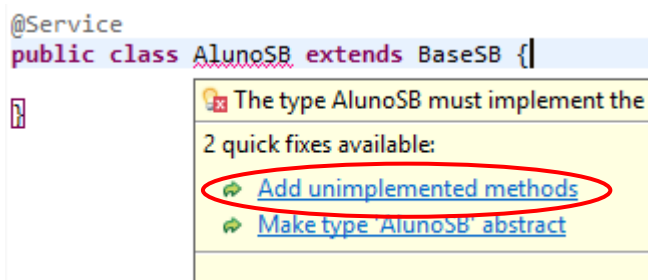
### 2. Classe Business

- 2.1. Crie o pacote `br.com.etechoracio.training.business`.
- 2.2. Crie a classe `AlunoSB` no pacote recém-criado.
- 2.3. A classe deve herdar de `BaseSB` e anotada com `@Service`.

```
@Service  
public class AlunoSB extends BaseSB {  
}
```

- 2.4. A classe estará com marcação de erro. Faça a correção adicionando os métodos não-implementados:

```
@Service  
public class AlunoSB extends BaseSB {  
}
```





**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.*

**2.5.** Aparecerá o método chamado `postConstructImpl()`.

```
@Override
protected void postConstructImpl() {
    // TODO Auto-generated method stub
}
```

**2.6.** Crie o atributo `alunoDAO` conforme abaixo:

```
@Service
public class AlunoSB extends BaseSB {

    private AlunoDAO alunoDAO;

    @Override
    protected void postConstructImpl() {

    }

}
```

**2.7.** Faça a instanciação dentro do método `postConstructImpl()` via método `getDAO()`:

```
@Override
protected void postConstructImpl() {
    alunoDAO = getDAO(AlunoDAO.class);
}
```

**2.8.** Crie o método `save()` abaixo:

```
public void save(Aluno aluno) {
    alunoDAO.save(aluno);
}
```

**2.9.** Anote-o com `@Transactional`:

```
@Transactional(propagation = Propagation.REQUIRED)
public void save(Aluno aluno) {
    alunoDAO.save(aluno);
}
```

**2.10.** Idente ( `CTRL` + `SHIFT` + `F` ) e salve a classe ( `CTRL` + `S` ).

## 3. Managed Bean

**3.1.** Crie o pacote `br.com.etechoracio.training.view`.

**3.2.** Crie a classe `AlunoMB` no pacote recém-criado.



**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.*

**3.3.** A classe deve herdar de BaseMB e anotada conforme abaixo :

```
@Getter
@Setter
@Controller
@Scope("view")
public class AlunoMB extends BaseMB {

}
```

**3.4.** Crie o atributo `alunoSB` anotado com `@Autowired` conforme abaixo:

```
@Getter
@Setter
@Controller
@Scope("view")
public class AlunoMB extends BaseMB {

    @Autowired
    private AlunoSB alunoSB;

}
```

**3.5.** Crie o atributo `edit` para a gravação dos dados do **Aluno** na tela:

```
@Getter
@Setter
@Controller
@Scope("view")
public class AlunoMB extends BaseMB {

    @Autowired
    private AlunoSB alunoSB;

    private Aluno edit = new Aluno();

}
```

**3.6.** Crie o método `onSave()` abaixo:

```
public void onSave(){

}
```

**3.7.** Inclua a edição do endereço:

```
public void onSave(){
    edit.getEndereco().setAluno(edit);
}
```



**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.*

- 3.8.** Salve o objeto `edit` e exiba a mensagem **"Registro inserido com sucesso."** através do método `showInsertMessage()`:

```
public void onSave(){  
    edit.getEndereco().setAluno(edit);  
    alunoSB.save(edit);  
    showInsertMessage();  
}
```

- 3.9.** Idente ( `CTRL` + `SHIFT` + `F` ) e salve a classe ( `CTRL` + `S` ).