

Mãos a obra: Usando o CDI

Vamos alterar nosso `Managedbeans` para serem gerenciados pelo **CDI**

Vamos começar pela classe `LoginBean`

```
@ManagedBean
public class LoginBean implements Serializable {
    ...
}
```

Vamos remover a anotação `@ManagedBean` e vamos adicionar a anotação `@Named`, para que essa classe fique acessível via `Expression Language`. Além disso precisamos definir o escopo dessa classe, como a classe `LoginBean` não tinha nenhum escopo definido o JSF assumia que o escopo seria de `Request`. Então vamos manter esse escopo `@RequestScoped`.

Vale lembrar que as anotações de escopo vem do pacote `javax.enterprise.context`.

Após a alteração nossa classe `LoginBean` deve estar assim:

```
@Named
@RequestScoped
public class LoginBean implements Serializable{
    ...
}
```

Faremos o mesmo processo para as classes `VendasBean` e `AutorBean`.

Na classe `TemaBean` vamos adicionar as anotações `@Named` e `@SessionScoped`.

```
@Named
@SessionScoped //do pacote javax.enterprise.context
public class TemaBean implements Serializable {
    ...
}
```

Por fim vamos alterar nossa classe `LivroBean`, essa por sua vez está usando um escopo do JSF `ViewScoped` e precisamos manter esse escopo quando alterarmos para que o **CDI** gerencie essa classe.

Para isso vamos remover a anotação `@ManagedBean` e adicionar a anotação `@Named`, e na anotação `@ViewScoped` vamos alterar o pacote dessa anotação para `javax.faces.view.ViewScoped` esse é um escopo que o JSF para se integrar com o **CDI**.

```
@Named
@ViewScoped // do pacote javax.faces.view.ViewScoped
public class LivroBean implements Serializable {
    ...
}
```

Rode a aplicação novamente e veja se tudo está funcionando.