06

Mãos à obra: Injeção o UsuarioDao e FacesContext

Vamos expandir o uso de CDI e o gerenciamento de beans por este contêiner em nossa aplicação. Segue as linhas gerais do que deve ser feito.

1 - Altere a classe `UsuarioDao` para que faça uso do EntityManager gerenciado pelo CDI (aproveite para implementar serializable). Essa mudança acarretará na remoção da dependência `JPAUtil` e a criação de um atributo do tipo `EntityManager` anotado com `@Inject`. Não esqueça de remover de `UsuarioDao` todas as chamadas para `em.close()`.

2 - Precisamos alterar também `LoginBean` para que receba `UsuarioDao` via injeção com a anotação `@Inject`. Lembre-se que isso acarreta na remoção de `new UsuarioDao()` em `LoginBean`.

3 - Ainda em `LoginBean`, injete também o `FacesContext` no novo atributo `contexto`, mas sabendo que isso por enquanto não funcionará. Ainda lembra a razão? Um `FacesContext` não é criado através `new`, mas através de uma chamada de um método estático e por isso o CDI não conseguirá injetá-lo para nós. Precisamos criar alguém que saiba produzir um `FaceContext` para nós da mesma forma que criamos uma classe que sabe fornecer um `EntityManager`.

4 - Crie a classe `JsfUtil`. Ela será uma produtora de `FacesContext` para podermos injetá-lo em nosso `LoginBean`. Só fique atento que o escopo do `FacesContext` produzido será `@RequestScoped`.

5 - Agora é só testar a aplicação.