

(Paula Prata)

1 – Timer Programado

a) Construir uma “enterprise application” com um module EJB. Definir um “Session Stateless Bean” com um método remoto à escolha. Definir uma aplicação cliente e testar a aplicação invocando o método.

b) Pretende-se construir um Timer programado. Para isso, adicionar ao Bean construído o seguinte código, após o cabeçalho da classe:

```
@Resource
TimerService timerService;
```

No interior da classe definir um método local anotado com @Timeout. Por Exemplo:

```
@Timeout
public void metodoTimeout(Timer timer) {
    System.out.println (“Ocorreu um timeout” + new Date());
}
```

- Para testar o método criar, num método remoto do Bean, o objeto do tipo Timer, que provocará a execução do método “metodoTimeout” ao fim do tempo especificado. Para isso usar a instrução:

```
Timer timer = timerService.createTimer(intervalDuration, “Created new timer”);
```

- O valor para o tempo após o qual ocorrerá um timeout (intervalDuration) pode ser dado como parâmetro do método que cria o Timer.

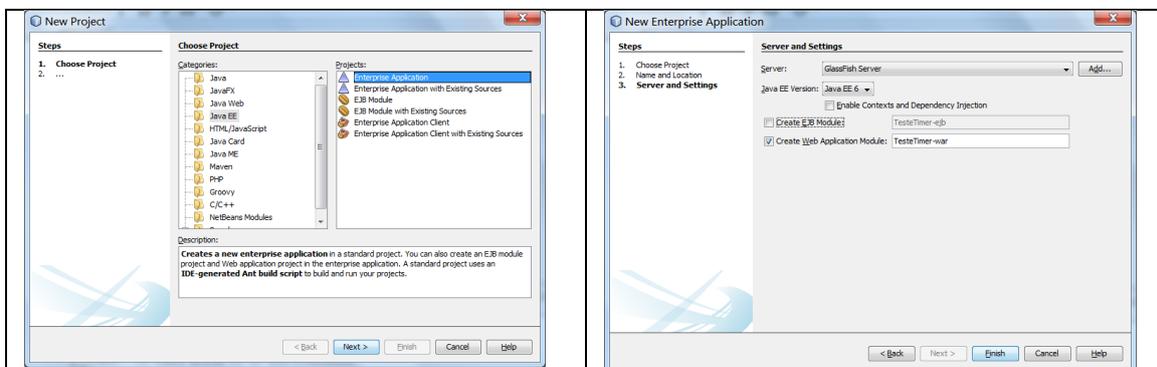
- Modificar a aplicação cliente de forma a invocar o método que cria o timer.

- Testar o exemplo pedindo ao utilizador vários valores para a duração do *timeout*.

Nota: É possível ver o output do método “metodoTimeout” na consola do servidor GassFish.

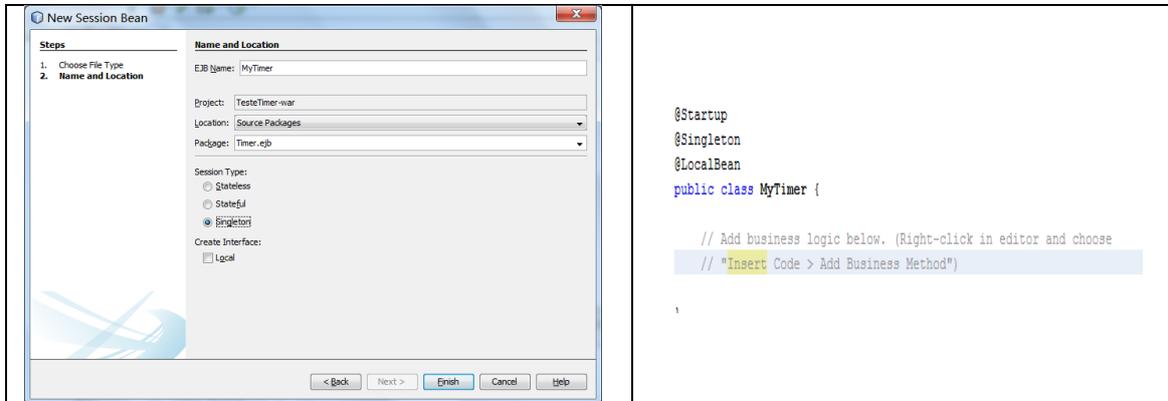
2 – Timer Automático

a) Criar uma “enterprise application” (nome = TesteTimer) com um module “web Application”.



(Paula Prata)

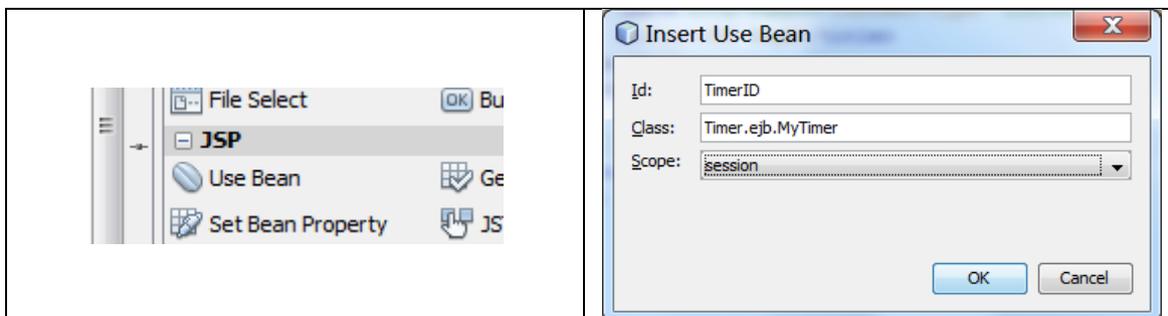
b) No módulo TesteTimer-war, Source Packages, criar um Session Singleton Bean com o nome MyTimer. Na classe do Bean, inserir a anotação @Startup (terá de fazer “Fix imports” para corrigir os erros que surgirem) e o código do método AutomaticTimeout() listado abaixo.



```
@Schedule(minute = "*/1", hour = "*")
public void automaticTimeout() {
    System.out.println ("Automatic timeout occurred " + new Date());
}
```

c) Após corrigir os erros, editar o ficheiro index.jsp em TesteTime-war, Web Pages. No menu Window selecione a Palette caso não tenha essa janela aberta.

- Na linha a seguir a “<h1>Hello World!</h1>” inserir o item UseBean da secção JSP da Palette. Dar um nome ao item, indicar o nome do Bean que construiu e escolher o scope “session”.



d) Executar a aplicação TesteTimer e observar o que acontece no Browser e na consola do servidor GlassFish.

e) Testar outros calendários (Schedules).

f) Modifique a aplicação para ter um Timer programado cuja duração é dada pelo utilizador.

g) No Bean, comentar a anotação @Startup, executar a aplicação e perceber a diferença!!

h) Na página index.jsp, comentar a linha “use Bean” e perceber o que acontece!!

i) Terminar o Timer em “Services” “Server” GlassFish Server” “Applications” em TesteTimer fazer undeploy.