

→ JPA – Java Persistence API

A – Criar JPA “entities” a partir de uma base de dados já existente

1 – Gerar JPA “entities”, criar um EJB para aceder à Base de Dados e criar uma Servlet para testar o Bean.

a) Criar uma Web Application (JPAfirst) como descrito anteriormente (folha 8, exercício B - 1.).

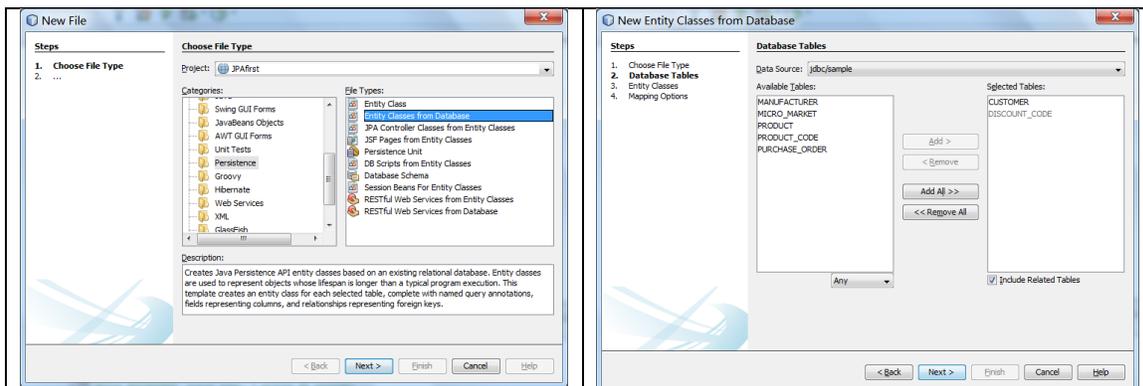
[New project / Java Web / Web application / Next/ atribuir o nome ao projecto/ Next / seleccionar versão do JavaEE e o servidor Glassfish / Finish].

- Executar a aplicação e observar que a página index.html é aberta no Browser definido por omissão.

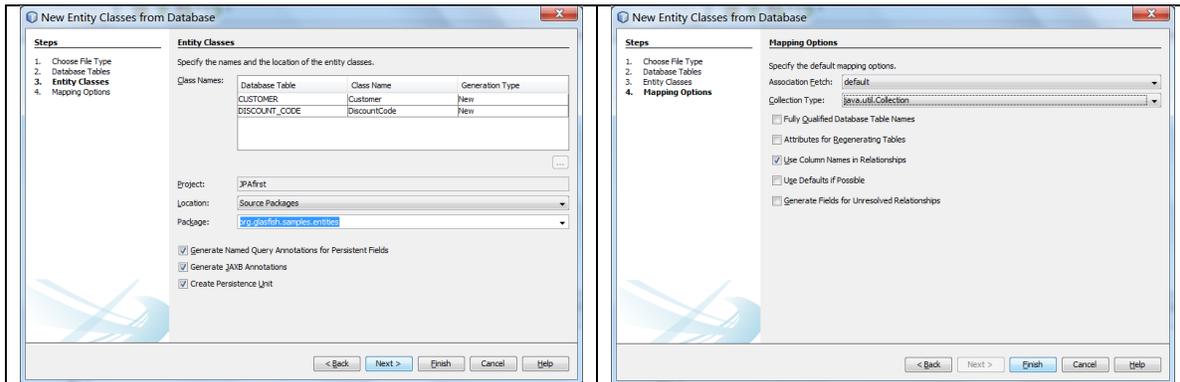
b) Criar “entities” para aceder a uma base de dados já existente.

- Fazer right-click no projeto e seleccionar: New / Other ... / Persistence / **Entity classes from database** /Next... (Nota: o percurso depende da versão do NetBens)

- Seleccionar no campo Data Source o valor “jdbc/sample”. As tabelas da base de dados exemplo aparecerão em “Available Tables”). Seleccionar a tabela CUSTOMER e fazer Add.



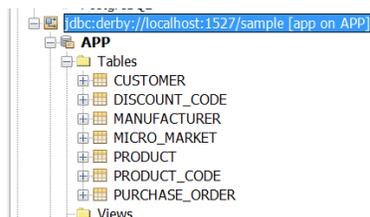
- Após Next, introduzir “org.glassfish.samples.entities” para o nome do package. Na coluna Class names aparecem as classes mapeadas da base de dados (isto é, a classe CUSTOMER e as classes que tenham alguma associação com a primeira). Após Next, observar as opções de mapeamento usadas por omissão.



c) Estude as classes que foram criadas em “Source Packages” “org.glassfish.samples.entities”. Para cada tabela mapeada foi criada uma classe.

- Verifique que as classes geradas usam JPQL (Java Persistence Query Language) para definir queries à base de Dados. Para cada tabela, foram gerados queries para consultar a tabela por cada um dos seus campos e um query para consultar todas as linhas da tabela. Podem ser adicionados novos queries.

- **Pode observar a estrutura das tabelas seleccionadas da base de dados em Services / Databases / Drivers / ...**

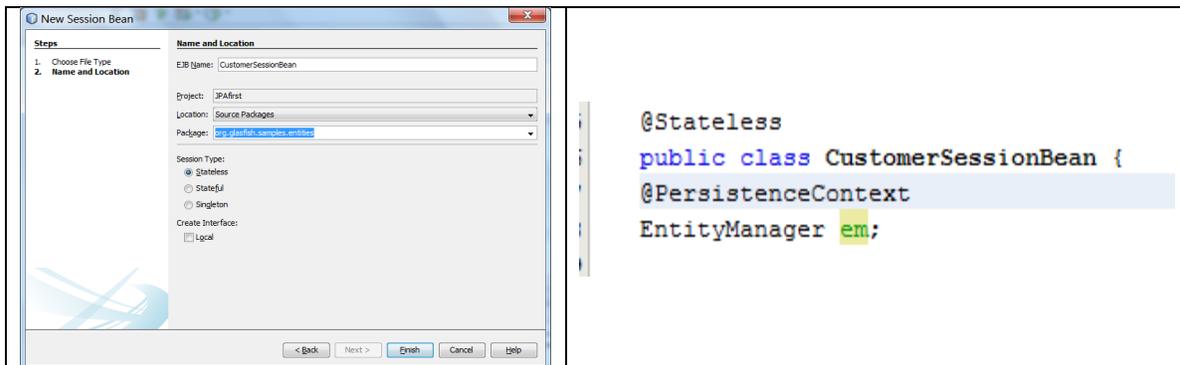


- Verifique que foram criadas regras de validação “BeanValidation Constraints” para cada campo, baseadas na definição do esquema relacional. Essas restrições serão usadas sempre que uma instância da entity é gravada, atualizada ou removida da base de dados. Restrições baseadas em expressões regulares podem ser usadas para validar os valores. Ver por exemplo o campo phone em Customer.java.

- Observando os campos das duas entidades, DISCOUNT_CODE e CUSTOMER, pode ver-se que existe uma associação (um para vários) em que um desconto está associado a vários clientes e um cliente está associado a um desconto. Esta associação está representada com as anotações @ManyToOne (em Customer) e @OneToMany (em DiscountCode). O campo discountCode permite fazer a junção das tabelas.

d) Criar um Bean para fazer *queries* à base de dados.

- Fazer Right-click em `org.glassfish.samples.entities` package e criar um Stateless EJB com o nome `CustomerSessionBean`. Na classe criada, injetar uma instância de `EntityManager` como ilustrado na figura abaixo. Fazer "Fix Imports".



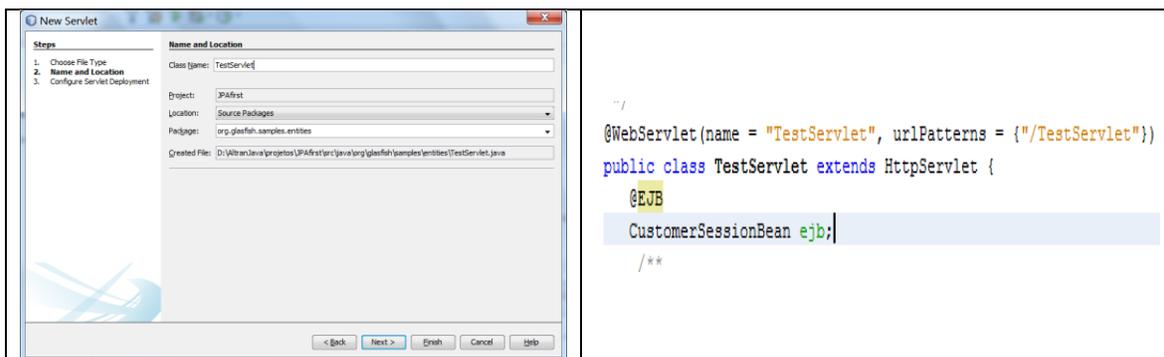
- Adicionar à classe `CustomerSessionBean` o seguinte método:

```
public List<Customer> getCustomer () {  
    return (List<Customer>) em.createNamedQuery("Customer.findAll").getResultList();  
}
```

//(Atenção: ao fazer Fix imports, o tipo `List` pertence ao package `java.util`)

- Este método está a usar um dos *queries* gerados anteriormente para obter todos os clientes (`customers`) da base de dados e devolvê-los num objeto do tipo lista de objetos do tipo `Customer`.

e) Criar uma servlet (de nome `TestServlet`) para testar o Bean. Observar, na Servlet criada, a anotação `@WebServlet`. Para invocar métodos do Bean criado, injetar a classe `CustomerSessionBean` na Servlet.



No Body do método processRequest da Servlet introduzir a invocação do método que definiu no bean CustomerSessionBean, isto é, a chamada do método getCustomer ().

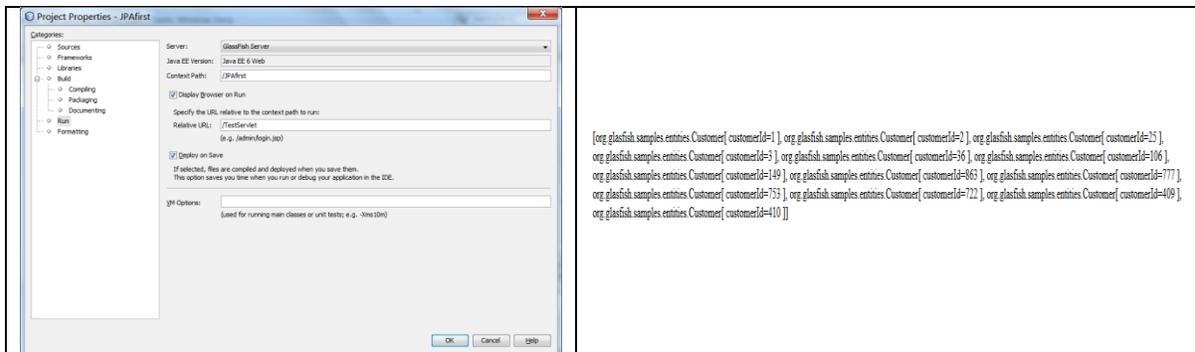
Essa invocação corresponde à instrução a bold da seção de código ilustrada abaixo.

```
out.println("<h1>Servlet TestServlet at " + request.getContextPath() + "</h1>");  
out.println (ejb.getCustomer());  
out.println("</body>");
```

Testar a servlet com right-click em TestServlet.java e seleccionar Run File.

[Alternativamente pode configurar o projeto para iniciar na servlet: Seleccionar projeto, em Properties / run / relative URL inserir /TestServlet (figura da esquerda)]

- Ao executar o projeto o Browser mostra a seguinte lista de identificadores (figura da direita):



2 - Modificar a aplicação para que mostre outros atributos de cada Customer (porque só aparece o ID?).

3 – Mostrar numa servlet os valores da tabela Product.

4 – Explore a construção de outros queries, nomeadamente queries que recebam valores através de parâmetros.

5 – Explore como criar uma base de dados simples (usando o JavaDB). Depois de criar a base de dados, proceda como anteriormente para gerar as “entities” correspondentes.