

Objetivos

- Praticar a utilização do padrão Table Module na organização da camada de negócio, consolidando os conceitos apresentados na aula teórica.
- Usando este padrão e uma camada de acesso aos dados fornecida, em que foi usado o padrão Table Data Gateway, implementar a camada de negócio do SaleSys.

Preparar a sessão de trabalho

Importe via *maven* o projeto eclipse `table-module-v1-alunos` disponibilizado na página da disciplina.

Analisar o projeto importado

O projeto Eclipse que acabou de importar contém um esqueleto da implementação a fazer. Como o objetivo é a implementação da camada de negócio, o projeto inclui `/src/main/resources/dataaccess-tdgw.jar` com a implementação da camada de acesso a dados seguindo o padrão *Table Data Gateway*.

A camada de acesso a dados fornecida desta vez não faz o armazenamento de dados apenas em memória como no projeto anterior, mas sim numa base de dados relacional. Mais precisamente, a implementação fornecida faz o armazenamento dos dados numa base de dados Derby embebida, sendo a comunicação da aplicação com o SGBD realizada através de um conector JDBC. A dependência do projeto numa versão embebida do Derby é visível por um lado no pom do projeto e, por outro lado, nos scripts de criação e reset das tabelas existentes em `/data/scripts`. Para saber como pode interagir com esta camada deve consultar o *javadoc* incluído na pasta doc. Abra o `index.html` e siga a documentação.

De forma a poder experimentar o seu código é fornecida uma pequena classe cliente (`SimpleClient`). É fornecida uma classe `CreateDatabase` que serve para mandar criar as tabelas necessárias e inserir os dados necessários para arrancar com a aplicação e `ResetTables` para eliminar os dados das tabelas em que os nossos casos de uso fazem inserções.

Completar a implementação das camadas de aplicação e negócio

1. Analise a forma como a operação `addCustomer`, cujo desenho foi discutido nas aulas teórico-práticas, se encontra concretizada no código fornecido (olhe para as classes `CustomerService`, `CustomerModule` e `TableModule`).
2. Ajuste a classe `SimpleClient` de forma a exercitar apenas a criação de um cliente e execute-a. Não se esqueça que antes tem de mandar criar a base de dados, através da execução de `CreateDatabase`.
3. Implemente a operação `newSale` de acordo com o desenho que foi realizado nas aulas teórico-práticas.
4. Ajuste a classe `SimpleClient` de forma a passar a exercitar também a criação de uma venda e execute-a. Não se esqueça que antes tem de executar `ResetTables`. Note que é a classe `SaleTableDataGateway` quem conhece o nome das colunas da tabela `Sale`, os quais são precisos para aceder à informação dos registos de um `ResultSet` produzido pela `SaleTableDataGateway`.
5. Repita os passos anteriores para a operação `addProductToSale`. Note que é a classe `SaleTableDataGateway` quem conhece o valor usado para representar o status de uma venda na tabela `Sale`.

Submeter trabalho

Crie um repositório `lab04` na sua área do `git.alunos.di.fc.ul.pt/`. Faça clone do repositório vazio para o seu computador, coloque aí o projeto Eclipse com o trabalho realizado e faça *push* para o repositório remoto. Não inclua ficheiros desnecessários (por exemplo, inclua a diretoria `target/` no `.gitignore`).