



UNIVERSIDADE
DE ÉVORA

1º Trabalho de Sistemas Operativos II



Trabalho realizado por:
Miguel Azevedo nº36975
Vasco Crespo nº37913

Professor:
José Saias

Introdução

O nosso trabalho tinha como objetivo a implementação de um sistema de informação de determinados produtos em determinadas lojas, de forma a que o utilizador possa notificar o sistema da necessidade crítica que tem de um determinado produto, e quando este for visto por outro interveniente, o utilizador é notificado.

Para isto utilizamos uma base de dados que irá ser modificada consoante as várias operações que o utilizador efetuar. Estas operações incluem um registo da necessidade crítica de um produto por parte de um utilizador, da consulta de necessidades de todos os utilizadores do sistema e também o reporte da disponibilidade de um produto numa determinada loja.

O nosso sistema também contém um sistema de *login*.

Desenvolvimento

Cada utilizador terá de efetuar um login para entrar usufruir do sistema. Estes também iram ter um ID único de forma a que o sistema consiga distinguir cada utilizador e as necessidades que cada um contém. Tendo em conta isto, foram criadas seis tabelas.

A primeira denominada utilizador contém o email de cada utilizador, e é gerado pelo SQL um *ID* único, para cada utilizador de forma a ajudar o sistema a distinguir cada utilizador. De seguida, temos as tabelas loja, produto e lojaproduto que são tabelas cujo os seus INSERTS forma introduzidos por nós, programadores. No caso da tabela loja, contém todas as lojas do nosso sistema, nas quais o utilizador pode pesquisar os produtos que necessita. Na tabela produtos, contém todos os produtos que o podem existir no sistema e no caso da tabela lojaproduto é uma tabela que relaciona as lojas aos produtos indicando quais os produtos que a loja X tem.

A tabela necessidades tem como objetivo registar as necessidades que cada utilizador tem, tendo assim o *ID* único de cada utilizador e o produto que o mesmo necessita. Por fim na tabela respostas, vão ser

registados os produtos e a loja em que se encontra para quando houver alguma necessidade critica de um produto, esse utilizador ser notificado.

Para as diferentes operações do sistema, foram criados métodos para resolver cada um:

- *int login* (String email) – este método irá registar um utilizador na base de dados, verificando se este email já se encontra na tabela utilizador retornando 0 caso exista algum email igual a esse, ou retornando 1 caso o registo tenha sido efetuado com sucesso.
- *ArrayList<String> consultaNecessidades ()* – este método irá percorrer a tabela necessidades e irá adicionar para uma ArrayList todas as linhas dessa tabela, correspondendo ao ID de cada utilizador e ao produto que necessita. No fim é retornada a ArrayList.
- *int registoNecessidade* (int id, String nome_Produto) – este método irá inserir na tabela *necessidades*, a necessidade correspondente ao utilizador com o ID e o produto que este necessita. Caso o utilizador já tenha efetuado o registo da sua necessidade, e ainda não tenha recebido a notificação que o produto ficou disponível irá ser retornado 0, caso o utilizador ainda não tenha registado a necessidade irá registar com sucesso e retornar 1.
- *int getId* (String email) – este método tem apenas como função retornar o *ID* associado ao email.
- *ArrayList<String> getProdutos ()* – este método irá pesquisar na tabela produtos e adicionar todos os produtos existentes no nosso sistema para uma ArrayList que irá ser retornada.
- *ArrayList<String> getLojas ()* – este método irá pesquisar na tabela loja e adicionar todas as lojas existentes no nosso sistema para uma ArrayList que irá ser retornada.
- *boolean disponibilidade* (String produto, String loja) – este método tem como objetivo verificar se um determinado produto está naquela loja retornando *true* se isso acontecer ou *false* caso contrario.

- *int reportaDisponibilidade* (String nome_loja, String produto) – este método irá adicionar a tabela *respostas* que o produto foi visto numa determinada loja, para mais tarde ser enviada notificação ao utilizador interessado. Caso este reporte já tenha sido efetuado vai ser retornado valor -1, caso contrario return 1.
- *ArrayList<String> getReports* (String nome_produto) – este método vai pesquisar na tabela *respostas* e adicionar todos os reports de disponibilidade de um determinado produto e em que loja é que foram vistos a uma *ArrayList* que vai ser retornada.
- *ArrayList<String> getProdutosLoja* (String nome_loja) – este método vai pesquisar na tabela *lojaproduto* e adicionar a uma *ArrayList* todos os produtos de uma determinada loja, retornando-a quando a tabela for vista por completo.
- *void RemoveNecessidade* (int id, String produto) - este método irá a tabela *necessidades* e remover a necessidade associada àquele ID com este produto. Este método é utilizado após o utilizador receber a notificação que o produto que necessidade ficou disponível, e sendo assim remove a sua necessidade critica.

Conclusão

Ao longo deste trabalho, foram encontradas algumas dificuldades, tais como, a ligação do Servidor à base de dados e a funcionalidade da base de dados.

Com este trabalho, deu-nos uma ideia mais preceptiva da importância do *middleware* e como alguns aspetos devem ficar transparentes ao utilizador.