

Fase 2 do EP - Relatório

Renato Coelho, Saulo de Oliveira, Vitor Sakaguti

28 de maio de 2008

Na implementação dessa segunda fase do projeto, onde tínhamos que escolher duas funcionalidades de um conjunto de seis para implementarmos, decidimos tentar trabalhar com o banco de dados orientado a objetos Magma. Essa decisão, no entanto, mostrou-se ser um grande erro, visto que encontramos muitos problemas com esse utilitário. Entre esses problemas, podemos citar que a instalação do Magma, quando da instalação do EP, não dava certo, fazendo com que mensagens passadas a objetos do banco de dados gerassem erros na execução do programa.

Visto que perdemos muito tempo tentando resolver os problemas com o banco de dados, precisamos usar partes que fizemos da fase um. No entanto a grande limitação de tempo com a qual nos deparamos quando finalmente desistimos do Magma não nos permitiu corrigir alguns problemas que havia na nossa implementação dessa fase, frutos da nossa então inexperiência. Como desde o princípio estávamos tentando implementar o projeto com o Magma, executamos os testes com a implementação supondo que o banco de dados estava funcionando, o que se reverteu em uma armadilha quando resolvemos usar uma implementação mais singela, fazendo com que nosso programa falhasse em todos os testes, o que conseguimos corrigir parcialmente.

Os testes foram levemente modificados para se adequarem melhor à nossa implementação. As chamadas "open elefante" foram trocadas para "open seaside/elefante". Além disso, no teste de manipulação de tarefas, removemos o campo que verificava o valor da data, pois para colocar a data usamos um `dateInput`, com o intuito de garantir consistência. A suite de testes está incluída no SAR.

Há também o caso da ordenação, que verificamos estar funcionando, mas que não está passando nos testes, por razões ainda desconhecidas.

Outra consequência do insucesso com o Magma foi que não pudemos nos dedicar a implementar as outras funcionalidades extras sugeridas.

A arquitetura que utilizamos é baseada em uma tarefa, subclasse de WATask. Essa tarefa escolhe quais serão os componentes mostrados. Temos três componentes principais: um para ser usado quando do registro de um usuário, um para ser usado quando o usuário vai se logar no sistema e outro para ser usado quando o usuário já está logado. Os dois primeiros apenas criam formulários, o que é natural, uma vez que as funcionalidades do programa são para o usuário logado. No componente do usuário logado, criamos mais dois componentes, um com a responsabilidade de listar as tarefas e o outro com a de criar menus.

No mais, um último comentário é que para inicializar o nosso programa, é preciso configurar a Session Class da aplicação para Sessao, pois não conseguimos fazer isso a partir do preâmbulo. Consultar o LEIAME incluído no SAR para maiores informações.