

Fase 2 – Programação Orientada a Objetos

Nome	Número USP
Pedro Paulo de Souza Bento da Silva	5434813
Rafael de Oliveira Lopes Gonçalves	5122902
Rafael Issao Miyagawa	3728716
Vivian Lababde Cury	3672228

Introdução

Nesta fase foi implementada uma interface gráfica para nosso usuário usando um aplicativo Web usando o framework Seaside. As funcionalidades obrigatórias implementadas foram: Banco de dados Magma e Magritte.

Requisitos da imagem

É necessário ter o Magma 1.0 server r4 instalado na imagem. Após a instalação do magma, deve ser instalado Fase2.sar

Considerações sobre o Desenvolvimento

A arquitetura tenta seguir o padrão Model-View-Controller. Com o Magritte, as definições de modelo fazem gerar mais facilmente a visualização destes, separando a visualização dos dados da semântica deles. Com o Magma, foi possível persistir os dados. Mas este poderia ser melhor integrado com o Magritte para restringir a consistência dos dados. Com o Magritte diversas tarefas repetitivas como a criação da visualização de objetos próprios foi automatizada. Houve complicações para integrar os componentes gerados pelo Magritte com os componentes envolvidos no sistema. Um dos problemas foi que não conseguimos implementar o mesmo layout para os componentes gerados pelo Magritte. Outro foi a dificuldade em criar atributos nos elementos gerados pelo Magritte.

Por fim, erramos em ver os testes somente quando o sistema ficou pronto, pois foi neste momento que surgiu a dificuldade de passar nos testes.

Arquitetura do sistema

Houve mudanças em relação a fase 1 na arquitetura do sistema. A seguir a lista das classes e suas respectivas modificações:

- **Usuario:** Adicionado o campo usuário e descrição para os atributos.
- **Tarefa:** Adicionado as descrições dos atributos.
- **Convite:** Classe nova. A instância desta classe é usada para guardar informações do convite feito para participação de uma tarefa. Atributos são remetente, destinatário e a tarefa.
- **BancoDeDados:** Classe nova. Responsável por persistir os dados criados no sistema.
- **RepositorioDe*:** Não está mais em uso.

- **FachadaDoElefante:** Possui novos métodos para satisfazer a necessidade da interface como salvar e recuperar dados do banco. Cada instância desta classe possui uma instância da classe bancoDeDados.

Arquitetura da Interface

A arquitetura da interface segue a mesma idéia do tutorial do seaside. Algumas classes foram criadas para facilitar o desenvolvimento:

- **EvLayout:** Super classe dos componentes que são responsáveis em criar uma interface completa. O layout da página é definido nesta classe. As subclasses devem implementar 'renderizarConteudoEm: html' no lugar de 'renderContentOn: html'

- **EvTaskReport:** Uma subclasse de MAREport. Esta classe foi criada para inserir alguns atributos nos elementos html gerado. Foi sobrescrito o método 'renderTableContentOn: html'

- **EvSession:** Uma subclasse de WASession. Criada para guardar a instância do usuário conectado e uma instância da fachadaDoElefante. Seguimos a idéia de uma conexão de banco por usuário. A sessão também guarda o email do usuário que foi removido e também se o usuário realizou logout.

O resto dos componentes seguem a idéia do tutorial.

Comportamento de compartilhamento de tarefa.

O usuário tem a opção de convidar um outro usuário a participar de uma certa tarefa. Para isso é enviado um convite para esta tarefa. O usuário destinatário tem a opção de aceitar ou rejeitar o convite. O usuário que envia convites tem a opção de remover o convite. Se um usuário envia um convite de uma tarefa e logo depois remove a tarefa da sua lista, o convite continua válido.

Sobre os testes selenium

A aplicação passa por testes quatro testes. Ele não passa do teste de compartilhamento de tarefas. Não tivemos tempo de modificar este teste.

Por causa do Magritte tivemos que mudar os testes que tentavam clicar em algum link ou botão criado pelo Magritte. Para contornar esse problema, muitos testes que eram clickAndWait viraram runAndWait com javascript que procura o elemento e realiza a ação. Os testes modificados estão na pasta Teste-Fase2.

Problemas conhecidos na aplicação

A aplicação possui dois problemas conhecidos. A data da tarefa é interpretado como sendo MM/dd/AAAA na hora da criação. Dependemos do Magritte para mudar esta interpretação.

Outro problema é o layout do componente gerado pelo Magritte. Até agora não conseguimos vincular o layout principal nos componentes Magritte.

Conclusão

Usar o Magritte foi o mais complicado nesta fase 2. O Magritte realmente facilita na geração de componentes, mas é necessário um estudo maior para podermos customizá-lo. Ao contrário, o banco de dados Magma foi realmente fácil de implementar.