

Relatório

Projeto de POO - Fase 2

Cauê Haucke Porta Guerra NUSP: 5450871

Leandro de Moraes NUSP: 5382062

Descrição das Classes

StInvite

Representa um convite. Contém basicamente 3 atributos:

- Usuário que enviou o convite.
- Tarefa envolvida no convite.
- Status do convite, que pode ser “aceito” ou “rejeitado” dependendo da resposta do usuário que recebeu o convite.

StInviteStatusViewer

Classe que representa o componente para visualização dos convites feitos pelo usuário. O status de um convite feito só é mostrado quando o usuário que o recebeu aceita ou rejeita o convite. Uma lista dos convites feitos e seus status é mostrada, e o usuário tem a opção de remover um item dessa lista.

StInviteViewer

Componente que fornece a visualização dos convites recebidos pelo usuário. É mostrado um convite por linha. Cada convite tem-se a opção de aceita-lo ou rejeita-lo. Ao aceitar ou rejeitar, é enviado uma resposta ao usuário que enviou o convite.

StListComponent

Componente utilizado para listar as tarefas de um usuário. Com esse componente é possível fazer a filtragem das tarefas, e a partir das tarefas filtradas, ordená-las de uma forma pré-definida. Tanto o modo de filtragem como o modo de ordenação podem ser definidas em tempo de execução.

StLoggedInComponent

É o principal componente. É o componente renderizado quando o usuário efetua o login. É neste componente que o usuário interage e tem acesso as funcionalidades do sistema. Este componente é constituído basicamente por 3 componentes: StMessageComponent, StListComponent e StMenuComponent.

StLoginComponent

Componente que representa a tela de login. Apresenta um campo para que se digite o email e outro para que se digite a senha. É mostrado também um link para que usuários novos possam se registrar. Possui um StMessageComponent que é utilizado para enviar mensagens ao usuário.

StMagmaDatabase

Classe da base de dados Magma. Na inicialização é criado 3 entradas na base de dados.

- #users – Coleção onde é armazenado todos os usuários cadastrados do sistema.

- #invites – Dicionário onde mapeia um usuário para uma coleção, ou seja, esta coleção é o conjunto dos convites recebidos pelo usuário.
- #invitesStatus – Dicionário que mapeia um usuário para uma coleção. Esta coleção representa o conjunto de convites feitos pelo usuário com seus devidos status.

StMenuComponent

Componente utilizado para renderizar um menu. As entradas do menu podem ser definidas a partir de uma associação chave->valor, onde a chave é o texto , e o valor é o link de execução. As associações são armazenadas em uma coleção.

StMessageComponent

Componente Utilizado para enviar mensagens ao usuário. Pode-se definir também o tipo de mensagem, se é um erro ou só uma informação.

StRegisterComponent

Componente responsável pelo cadastramento de um novo usuário no sistema. É composto por uma StMessageComponent. Garante que a senha do usuário é encriptada antes de ser armazenada no sistema. Usa-se o algoritmo SHA para isto.

StRootTask

Controla o fluxo do sistema. É o componente central. Subclasse de WATask.

StSession

Componente que representa uma sessão. É subclasse de WASession. Possui como variáveis de sessão a base de dados(db), usuário(user , usuário logado) e a fachada(instancia da FachadaDoElefante).

StTaskEditor

Componente que representa o editor de uma tarefa. É este componente chamado quando se quer editar ou criar uma nova tarefa. Possui um StMessageComponent. É através dele que o usuário é capaz de enviar convites para outros usuários.

StTestingComponent

Componente usado quando se precisa fazer perguntas do tipo verdadeiro ou falso ao usuário. Possui dois botões que devolvem true ou false. A pergunta que é feita ao usuário pode ser definida em tempo de execução.

StUnregisterComponent

Componente utilizado para desregistrar um usuário.

Funcionalidades implementadas

Persistir os dados usando o banco de dados orientado a objetos Magma.

Testes

Durante o desenvolvimento do projeto, tomamos algumas decisões que nos obrigou a modificar alguns testes.

Decidimos por exibir uma tela de confirmação sempre que o usuário desejar excluir uma tarefa, o que fez com que incluíssemos essa etapa nos testes que envolviam exclusão de tarefas; E utilizamos a descrição no enunciado dessa fase, que pedia que ao ser feito um convite, o usuário convidado pudesse aceitar ou recusar uma tarefa. Então, alteramos os testes de compartilhamento de tarefas para incluir essa etapa de aceitação da tarefa compartilhada.

Segue em anexo os testes alterados, e que foram utilizados pelo nosso grupo.

Pacotes

Serão criados os seguintes pacotes:

POO

StTutTodoApp

Preparando o Ambiente

Instalando o magma:

```
Installer squeaksource project: 'Installer'; install: 'Installer-Core'.
```

```
Installer universe  
  answer: 'username' with: 'admin';  
  answer: 'password' with: 'seaside';  
  install: 'Magma seasideHelper';  
  install.
```

Na configuração da aplicação:

adicionar **WAMagmaConfiguration** como ancestral.

Magma Helper Class: WAMagmaSoloAuto

Location path: magma-elefante

Root Component: StRootTask

Session Class: StSession

Digite e execute as seguintes linhas no workspace:

```
MagmaRepositoryController  
  create: 'magma-elefante'  
  root: Dictionary new.
```

WAKomEncoded startOn: 8080

Referências:

http://www.swa.hpi.uni-potsdam.de/seaside/tutorial?_k=ClSkggne&_s=NseVOIFiwjhKvVEb

<http://wiki.squeak.org/squeak/6021/>