Plataformas de Desenvolvimento
16/17

Mini-relatório de estado de Meta1/Trabalho prático

Identificação do tema e do grupo **TP / meta 1**

ID Grupo	**G14**			
Aluno 1:	Número	**21260825**	Nome:	Diogo Gomes
Aluno 2:	Número	**9805008.**	Nome:	Eugénio Santos

EJB Existentes

ClientVisitante

- Stateless, remoto
- Tem como objectivo permitir o acesso remoto à funcionalidade da aplicação por clientes remotos, sem autenticação.
- Funções mais salientes:
    - boolean inscreveUtilizador(String nome, String morada, String username, String password). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para inscrever um utilizador, ficando a aguardar aprovação do Admin para poder fazer login. Devolve true se for realizado com sucesso.
    - boolean existeUsername(String username). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para verificar a disponibilidade do username único durante a inscrição de utilizador. Devolve true se existe.
    - boolean loginUtilizador(String username, String password). Esta função, utiliza uma função do sistema (Leiloeira - EJB Local) para autenticar o utilizador (ou admin) no sistema (Leiloeira - EJB local). Devolve true se login com sucesso.

ClientAdmin

- Singleton, remoto
- Tem como objetivo permitir o acesso às funcionalidades de administração por cliente remoto, autenticado como “admin”.
- Funções mais salientes:
    - ArrayList getUtilizadoresPedidos(). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para devolver a lista de utilizadores inscritos, a aguardar aprovação para entrar.
    - boolean ativaUtilizador(String username). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para aprovar um utilizador inscrito para poder entrar. Devolve true se ativação com sucesso.
    - ArrayList getPedidosSuspensao(). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para mostrar a lista de utilizadores que pediram suspensão, com respetiva razão do pedido.
    - boolean suspendeUsername(String username). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para suspender um utilizador. Devolve true se a suspensão for com sucesso.
    - void setLastAction(). Esta função é chamada sempre que outra função seja executada, para através de uma função do sistema (Leiloeira - EJB Local) fazer reset ao timeout definido no sistema para logout automático ao fim de 4 minutos sem atividade.

Plataformas de Desenvolvimento
16/17

Mini-relatório de estado de Meta1/Trabalho prático

Plataformas de Desenvolvimento
16/17

Mini-relatório de estado de Meta1/Trabalho prático

ClientUtilizador

- Stateful, remoto
- Tem como objetivo permitir o acesso às funcionalidades de utilizador autenticado por cliente remoto.
- Funções mais salientes:
    - boolean setMyName(String username, String password). Esta função efetua o login em sessão de utilizador registanto o username no EJB statefull e impede que haja outra sessão do mesmo utilizador em simultâneo. Devolve true se for login único e false se o utilizador já tiver sessão ativa.
    - boolean logOff(). Esta função, efetua o logoff da sessão do utilizador através de uma função do sistema (Leiloeira - EJB Local). Devolve True se tiver sucesso.
    - Double addSaldo(Double valor). Esta função permite adicionar saldo à conta do utilizador. Devolve o valor do saldo após a execução.
    - Double getSaldo(). Esta função devolve o valor de saldo atual do utilizador.
    - String getDados(). Esta função devolve os dados pessoais do utilizador em formato String.
    - boolean atualizaDados(String nome, String morada). Esta função permite ao utilizador alterar os seus dados pessoais atraves de uma funçao no sistema (Leiloeira - EJB Local). Devolve true se tiver sucesso.
    - boolean pedirSuspensao(String razao). Esta função utiliza uma função do sistema (Leiloeira - EJB Local) para permitir ao utilizador, pedir a suspensão da sua conta, indicando a razão do pedido.
    - boolean sendMensagem(String destinatario, String texto,String assunto). Esta função usa uma função do sistema (Leiloeira - EJB Local) para permitir enviar mensagens a utilizadores. Verifica se o username destinatario existe e devolve false caso nao exista.
    - ArrayList<Mensagem> consultarMensagens(). Esta função através de uma função do sistema (Leiloeira - EJB Local), permite obter uma lista de objetos Mensagem do utilizador em sessão.
    - void setLastAction(). Esta função é chamada sempre que outra função seja executada, para através de uma função do sistema (Leiloeira - EJB Local) fazer reset ao timeout definido no sistema para logout automático ao fim de 4 minutos sem atividade.

Leiloeira

- Singleton, local
- Tem como objetivo ser o núcleo do servidor. Agrupa a lista de utilizadores, lista de itens, categorias, mensagens, etc.
- Tem definida a lógica principal da aplicação e restringe o acesso a utilizadores e itens.
- Os EJBs remotos só interagem com os utilizadores ou itens através das funções deste EJB local.
- Funções principais:
    - boolean existeUtilizador(String username). Verifica existe o username no sistema. Devolve true se existir.
    - boolean registaUtilizador(String nome, String morada, String username, String password). Regista um novo utilizador no sistema. Devolve false se o username ja existir.
    - boolean loginUtilizador(String username, String password). Faz o login de um utilizador no sistema. Devolve false se o username nao existe ou se não estiver em estado “ativo”, ou se já tiver a sessão ativa.
    - boolean logOff(String username). Efetua o logoff de um utilizador. Devolve false se o utilizar não tiver sessão ativa.
    - void checkInactivity(). @Schedule(second = "*/5", minute = "*", hour = "*") Executado de 5 em 5 minutos, faz logoff de utilizadores com 4 minutos de inatividade.
    - void loadstate(). @PostConstruct. Carrega dados registados em ficheiro.
    - void saveState(). @PreDestroy. Regista dados em ficheiro.

- ArrayList getUsernameInscritos(). Devolve a lista de todos os utilizadores inscritos.
- ArrayList getUsernamesOnline(). Devolve a lista de todos os utilizadores com sessão no sistema.
- Double addSaldo(Double valor,String username). Adiciona saldo a um utilizador. Devolve o valor após operação.
- Double getSaldo(String username). Devolve o saldo de um utilizador.
- boolean ativaUtilizador(String username). Ativa um utilizador para poder fazer login. Devolve false se o utilizar já estive em estado "Ativo".
- ArrayList getUtilizadoresEstado(UtilizadorEstado estado). Devolve uma lista de usernames de utilizadores com determinado estado.
- String getDadosUtilizador(String username). Devolve os dados de um utilizador, em texto legível.
- boolean atualizaDadosUtilizador(String username, String nome, String morada). Atualiza os dados: nome e morada de um utilizador identificado pelo username utilizando uma função do objeto Utilizador. Devolve fase se nao tiver sucesso.
- boolean pedirSuspensaoUtilizador(String username,String razao). Regista o pedido de suspensao de utilizador, com a respetiva razao justificativa. Este pedido altera o estado de um utilizador para "Suspensao Pedida". Devolve sempre true.
- HashMap getPedidosSuspensao(). Devolve uma Hashmap dos pedidos de suspensão, com a razão de cada pedido associado ao username respetivo.
- boolean suspendeUtilizador(String username). Altera para o estado "Suspenso" um determinado utilizador. Devolve sempre true.
- void setLastAction(String username). Efetua o reset ao timeout de um utilizador através de uma função identica no objeto Utilizador.
- boolean addMensagem(String remetente, String destinatario, String texto,String assunto). Regista uma mensagem de um utilizador para outro, com o estado "Enviada".
- ArrayList getMensagensUtilizador(String username). Devolve a lista de mensagens recebidas de um utilizador.

- Dados mais salientes, persistidos nas funções (saveState/@PreDestroy e loadstate/@PostConstruct)
  - HashMap de utilizadores. Username como chave, objeto Utilizador como valor.
  - ArrayList de mensagens.
  - ArrayList de categorias.

Ligação entre os vários módulos no servidor (EJB)

ClientVisitante, ClientAdmin e ClientUtilizador, todos têm @EJB LeiloeiraLocal leiloeira para aceder ao EJB Local Singleton que reune todos os dados do sistema.

Plataformas de Desenvolvimento
16/17

Mini-relatório de estado de Meta1/Trabalho prático

Outros aspectos relevantes

Nas referências, em vez do IP do Glassfish, foi usado o hostname “glassfixe”, pelo que para que a ligação funcione deve-se associar o IP ao hostname “glassfixe”. Para isso, na máquina cliente, em ambientes *nix adicionar a linha ‘<IP servidor> glassfixe’ ao ficheiro /etc/hosts.

Na aplicação cliente, para adicionar uma opção a um menu, adicionar uma linha deste género:

```
opcoes.add(new OpcaoMenu("Contas", () -> controlador.contas()));
```

“Contas” = designação da opção no menu

controlador.contas() = função lambda que executa a função do menu. (neste caso, é a função contas())

Para adicionar um sub-menu, a função que chama o sub-menu é algo do género:

```
menu = new MenuAdminCategorias(ligacao, (ControladorAdministrador) controlador);
```

em que a nova classe MenuAdminCategorias tem de ter uma estrutura similar à dos outros menus.