Aluno: PAULO ROBERTO QUEIROZ DE OLIVEIRA

Disciplina: Programação com frameworks e componentes

1 - Código fonte

Disponibilizado no Github:

https://github.com/paulorgo/projeto-rmi-alfa-paulo/

2 - Conexão com o banco de dados:

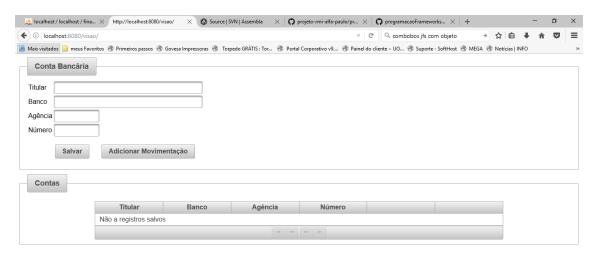
2 – No projeto "projeto-persistencia" executar a classe "PersistenciaModeloImpl.java" que sobe na porta 2001;

```
package br.com.financas.rmi.impl;
mport java.rmi.RemoteException;
 public class PersistenciaModeloImpl extends UnicastRemoteObject implements PersistenciaModelo {
     private static final long serialVersionUID = 181353194560607760L;
     private ContaDao contaDao = new ContaDao();
     private MovimentacaoDao movimentacaoDao = new MovimentacaoDao();
     private EntidadeConverter entidadeConverter = new EntidadeConverter();
     public PersistenciaModeloImpl() throws RemoteException {
          super();
     public static void main(String args[]) {
              PersistenciaModeloImpl persistencia = new PersistenciaModeloImpl();
               // Naming.rebind("OlaServidor", obj);
              Registry registry = LocateRegistry.createRegistry(2001); registry.rebind("OlaServidor", persistencia); System.out.println("Servidor carregado no registry");
          } catch (Exception e) {
              System.out.println("OlaImpl erro: " + e.getMessage());
          }
     3
     @Override
     public ContaVO salvar(ContaVO contaVO) throws RemoteException {
         Conta conta = entidadeConverter.converterVOparaEntidade(contaVO);
          conta = contaDao.salvar(conta);
          return entidadeConverter.converterEntidadeParaVO(conta);
     }
     @Override
     public ContaVO alterar(ContaVO contaVO) throws RemoteException {
          Conta conta = entidadeConverter.converterVOparaEntidade(contaVO);
          conta = contaDao.alterar(conta);
          return entidadeConverter.converterEntidadeParaVO(conta);
```

3 – Acessar o projeto da view "visao" pelo seguinte endereço:

Fazer o deploy do war "visao.war" ou executar o projeto na IDE e acessar o endereço:

http://localhost:8080/visao/

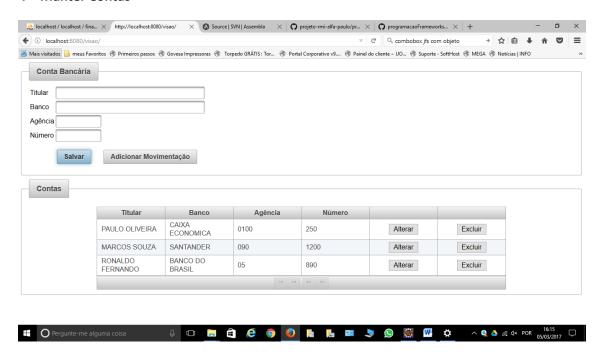




Possui dois controladores "ContaMB.java" e "MovimentacaoMB.java" aonde após rodar o construtor assume como padrão 127.0.0.1 e porta 2001 para comunicar com a persistência por RMI.

```
package controle;
import java.io.IOException;[]
@ManagedBean
@SessionScoped
public class MovimentacaoMB {
     public MovimentacaoMB() {
          super();
     }
     private PersistenciaModelo persistenciaModelo;
     private Registry registry;
     private MovimentacaoVO movimentacaoVO;
List<MovimentacaoVO movimentacoes = new ArrayList<>();
List<ContaVO> contas = new ArrayList<>();
     private ContaVO contaVO;
     @PostConstruct
     public void popular() {
    movimentacaoV0 = new MovimentacaoV0();
               registry = LocateRegistry.getRegistry("127.0.0.1", 2001);
persistenciaModelo = (PersistenciaModelo) registry.lookup("0laServidor");
                movimentacaoVO.setTipoMovimentacao(TipoMovimentacao.ENTRADA);
                movimentacaoVO.setData(new Date());
                movimentacaoVO.setValor(BigDecimal.ZERO);
          pesquisar();
  pesquisarContas();
} catch (RemoteException | NotBoundException e) {
                e.printStackTrace();
```

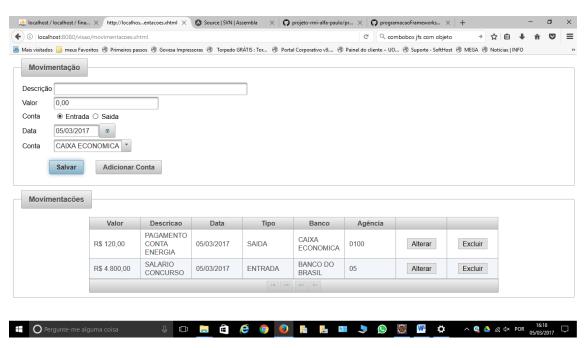
4 - Manter Contas



Realiza as operações de Salvar, Alterar e Excluir uma Conta;

Se Clicar em Adicionar Movimentação redireciona para pagina de Manter Movimentação;

5 – Manter Movimentação



Realiza as operações de Salvar, Alterar e Excluir uma Movimentação;

Se Clicar em Adicionar Conta redireciona para pagina de Manter Conta;