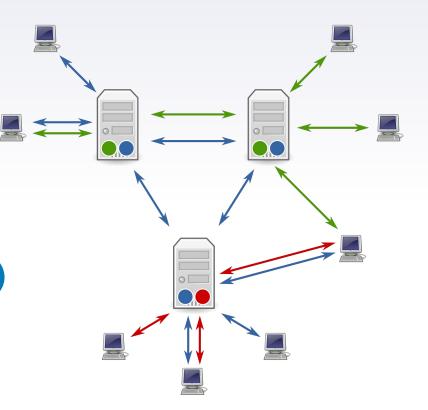
SISTEMA BANCÁRIO DISTRIBUÍDO



Alunos

- Isabela Fontes de Araújo
- Kéthlyn Campos Silva
- Kamilla Cristina Silva Martins
- Fabrício Lopes Mendonça
- João Victor Santos Alves e Silva



Introdução

- O projeto desenvolvido é um Sistema Bancário Distribuído (SBD);
- Requisitos do Sistema:
 - Abertura de conta;
 - Consulta de saldo e extrato da conta
 - Transferências entre contas
 - Saque
 - Depósito



Metodologia de Trabalho

- Java;
- Java Remote Method Invocation (RMI);
- Java Database Connectivity (JDBC).







Vantagens

- Sistema Distribuído
 - Disponibilidade de serviços
 - Heterogeneidade
 - Transparência
 - Independência
- Java RMI
 - Atende aos requisitos
 - Comunicação distribuída

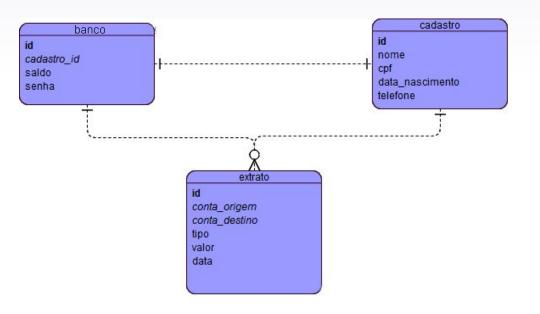


Divisão de etapas

Conexão do Modelo banco com o entidade-relacionamento sistema Criação e Realização de alimentação testes do banco

Arquitetura do Software

Modelo entidade-relacionamento



Código SQL

Criar tabelas:

```
CREATE TABLE cadastro (
id SERIAL PRIMARY KEY,
nome VARCHAR (50) NOT NULL,
cpf VARCHAR (11) UNIQUE NOT NULL,
telefone VARCHAR (13) NOT NULL,
data_nascimento DATE NOT NULL
);
```



Código SQL

Criar tabelas:

```
CREATE TABLE banco (
id SERIAL PRIMARY KEY,
cadastro_id INT REFERENCES cadastro (id),
saldo numeric(15,2) default 0,
senha VARCHAR NOT NULL
);
```



Código SQL

data DATE NOT NULL

Criar tabelas: CREATE TABLE extrato (id SERIAL PRIMARY KEY, conta_origem INT REFERENCES cadastro (id), conta_destino INT REFERENCES cadastro (id), tipo VARCHAR NOT NULL, valor numeric (15,2) default 0,

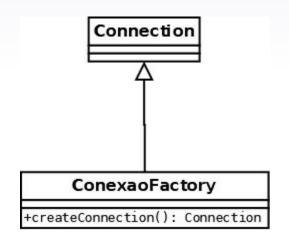


Diagramas de Classes

- ConexaoFactory;
- AbstractRepository;
- ExtratoRepository;
- CadastroRepository;
 - SqlBuilder.
- BancoRepository.
 - Banco.
 - BancoController.



ConexaoFactory



```
package br.com.banco.server;

import java.sql.*;

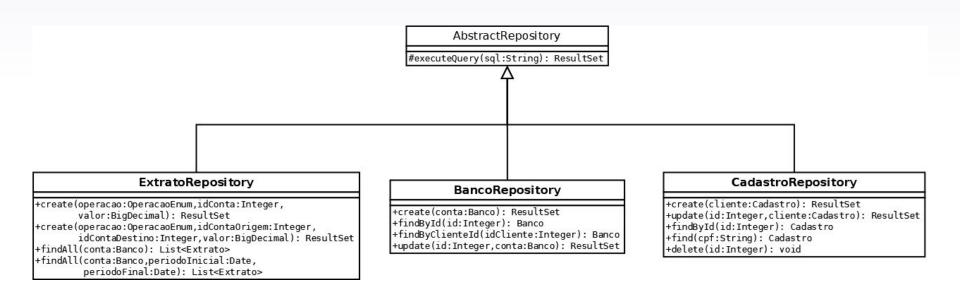
public class ConexaoFactory {

public static Connection createConnection() throws SQLException{
    String url = "jdbc:postgresql://172.16.56.190/postgres";
    String user = "postgres";
    String password = "123456";

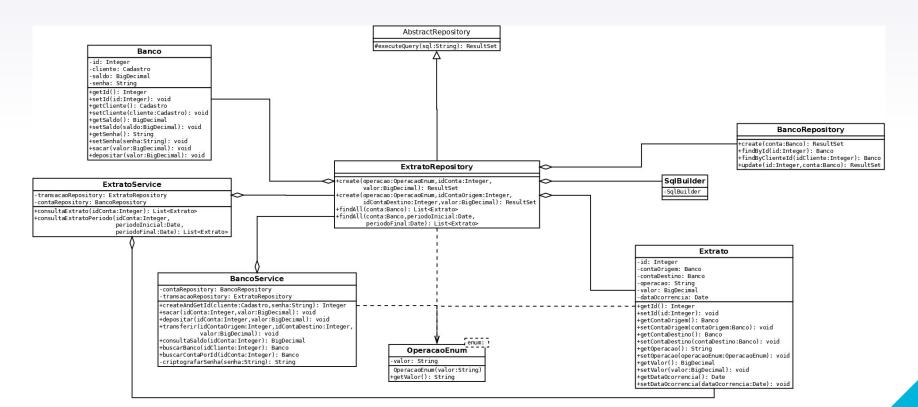
return DriverManager.getConnection(url, user, password);
}

}
```

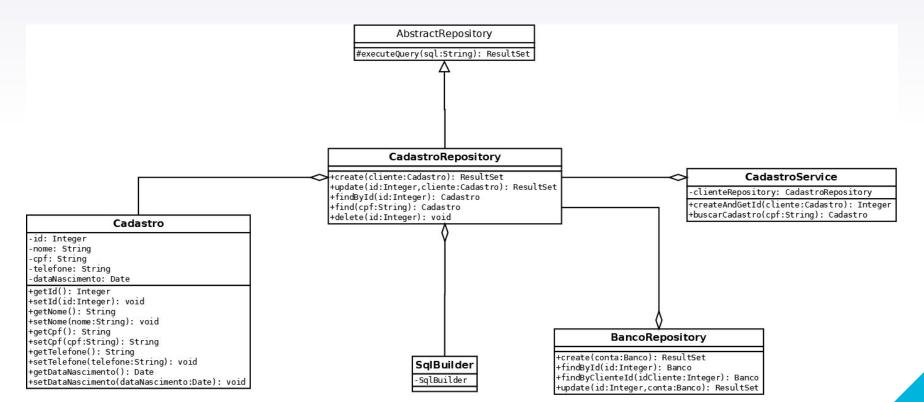
AbstractRepository



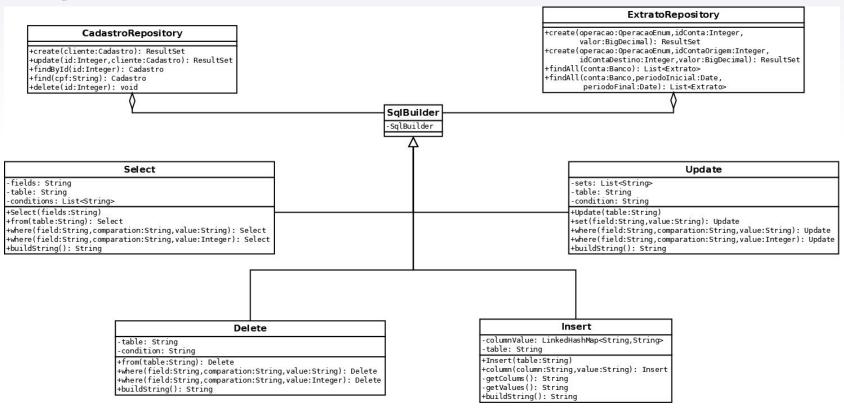
ExtratoRepository



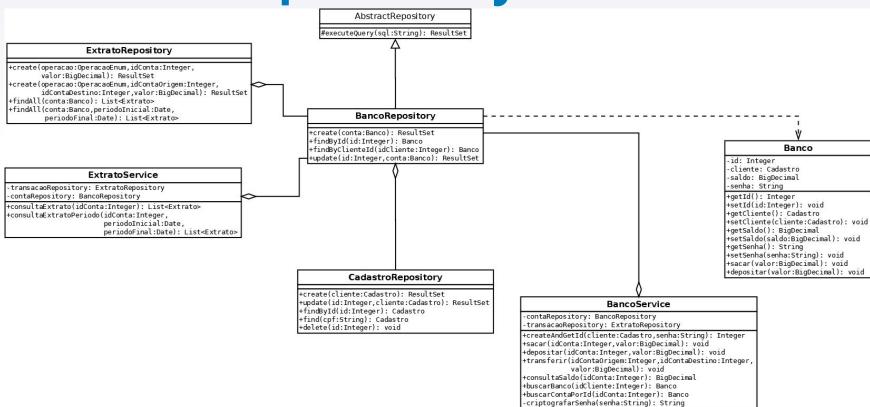
CadastroRepository



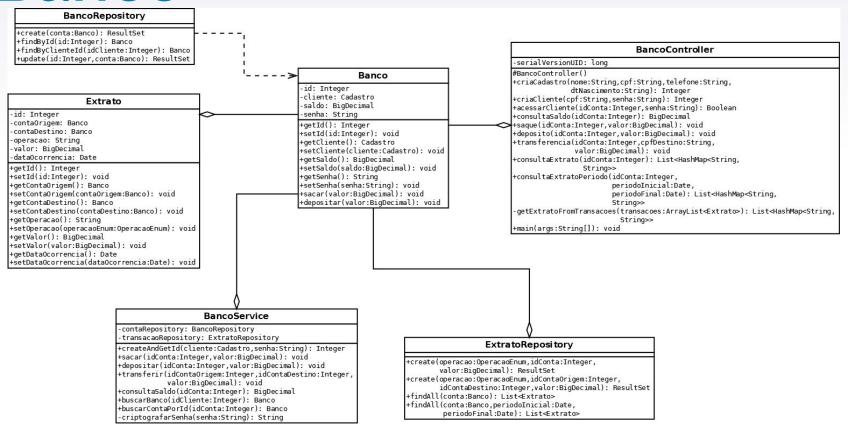
SqlBuilder



BancoRepository



Banco



BancoController

Cadastro

-id: Integer
-nome: String
-cpf: String
-telefone: String
-dataNascimento: Date
+getId(): Integer
+setId(id:Integer): void
+detNome(): String

+setNome(nome:String): void +getCpf(): String +setCpf(cpf:String): String +getTelefone(): String

+setTelefone(telefone:String): void +getDataNascimento(): Date

+setDataNascimento(dataNascimento:Date): void

BancoController

#BancoController()

-serialVersionUID: long

+criaCadastro(nome:String,cpf:String,telefone:String, dtNascimento:String): Integer

+criaCliente(cpf:String,senha:String): Integer +acessarCliente(idConta:Integer,senha:String): Boolean +consultaSaldo(idConta:Integer): BigDecimal

+saque(idConta:Integer,valor:BigDecimal): void
+deposito(idConta:Integer,valor:BigDecimal): void
+transferencia(idConta:Integer,cofDestino:String.

valor:BigDecimal): void +consultaExtrato(idConta:Integer): List<HashMap<String,

String>> +consultaExtratoPeriodo(idConta:Integer,

periodoInicial:Date,
periodoFinal:Date): List<HashMap<String,</pre>

.getExtratoFromTransacoes(transacoes:ArrayList≪xtrato>): List∢HashMap<String, String>>

+main(args:String[]): void

Banco

-id: Integer -cliente: Cadastro -saldo: BigDecimal -senha: String +getId(): Integer

+setId(id:Integer): void +getCliente(): Cadastro +setCliente(cliente:Cadastro): void +getSaldo(): BigDecimal +setSaldo(saldo:BigDecimal): void

+getSenha(): String
+setSenha(senha:String): void
+sacar(valor:BigDecimal): void
+depositar(valor:BigDecimal): void

+criarCadastro(nome:String,cpf:String,telefone:String,dtNascimento:Date): Integer +criarCliente(cpf:String,senha:String): Integer +acessarCliente(adconta:Integer,senha:String): Boolean +consultaSaldo(idconta:Integer,BigDecimal): void +saque(idcliente:Integer,valor:BigDecimal): void +deposito(idcliente:Integer,valor:BigDecimal): void +transferencia(idcliente:Integer,cpfDestino:String,valor:BigDecimal): void +consultaExtrato(idcliente:Integer): List<HashMap<String,String>> +consultaExtratoPeriodo(idcliente:Integer,periodoFinal:Date,periodoFinal:Date): List<HashMap<String,String>>

BancoInterface

BancoClient

+in: Scanner +main(args:String[]): void

-getBancoInterface(ids:String[]): BancoInterface

Interface:

BancoController

```
public static void main(String[] args) {
107⊖
108
                                      try {
                                                   BancoController obj = new BancoController();
109
                                                   Registry registry = LocateRegistry.createRegistry(2001);
110
                                                   registry.rebind("BancoServer", obj);
111
                                                   System.out.println("Servidor carregado no registry");
112
                                      } catch (Exception e) {
113
114
                                                   System.out.println("Banco RMI erro: " + e.getMessage());
116
                                                                                      117 }
                                                                                                         😑 🗣 👂 🕆 ° ° 🗆 🔯 BancoControllerjana II 🖟 Conessofactoryjana 💢 BancoSenicajana 🖟 AbstractRepositoryjana

→ G. BancoController

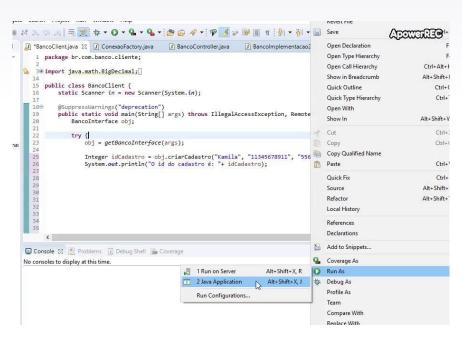
                                                                                       ≥ Bi conectaPostore
                                                                                                                                                                                                JF seria/VersionUID : long
                                                                                                                                                                                                of BancoController()
                                                                                       Frontas-idboxip expanded
118
                                                                                                                         public class SancoController extends UnicastSemoteObject implements SancoInterface (
                                                                                                                                                                                                . criarCadastro(String, String, String, Date)
                                                                                                                                                                                                . crierCliente(String, String) : Integer
                                                                                                                           private static final long serial VersionUZD = -1400220616929807673L;
                                                                                        w 🥌 professionfava
                                                                                                                                                                                                . acessarCliente(Integer, String): Boole
                                                                                           AbstractRepositoryjava
                                                                                                                                                                                                e - transferencialistener String BigDecimali-
                                                                                                                          public Integer crier(adastro(String nose, String cof, String telefone, Date diffusciments) throws Remote
                                                                                          2 P. Rancolnterface laws
                                                                                                                                                                                                . consultaEutratoPeriodo/Intener, Date, Dat.
                                                                                          > III BancoService Inva
                                                                                                                                                                                                • * main(String(II) : wold
                                                                                                                             cliente.setDataNascimento(dtNascimento);
                                                                                          > I/I CadastroRepositors.iava
                                                                                                                             return CadastroService.createAndGetTd(cliente):
                                                                                          > II) Conexagfactory ava
                                                                                                                           public Integer criarCliente(String cof, String senha) throws RemoteException (
                                                                                          > II ExtratoRepository and

    EstratoService.java
    El OperacaoEnum.java

                                                                                                                             Cadastro cliente = CadastroService.buscarCadastro(cpf);
return BancoService.createAndGetId(cliente, senha);
                                                                                          > IZ SqlBuilder,java
                                                                                        M IRE System Library Llava/SE-130
                                                                                                                           public Boolean acessarCliente(Integer idConta, String senha) throws RemoteException (
                                                                                        Mayen Dependencies
                                                                                                                             Banco conta = BancoService, buscarContaPorId(idConta);
                                                                                                                                                                                            ■ X % | B, XI B (## | # 0 - m - * n
                                                                                                                     BancoController (lava Application) Ci-Program Filesi Javanije 18.0,1771/binijavaw.exe (28 de nov de 2019 19:17:06;
Servidor carregado no registry
```

Writable Smart Insert 31:39:914

Alguns testes Criação de contas



Alguns testes Depósito e Saque

```
    ■ BancoClient.j... 
    □ ConexaoFactor...
    □ ConexaoFactor...

                                       J BancoControll...
                                                                                                   AbstractRepo...
                   Integer idConta = 1;
  30
                  if (obj.acessarCliente(idConta, "123456")) {
                       obj.deposito(idConta, new BigDecimal(75.3));
                       System.out.println("O depósito foi realizado.");
                       obj.saque(idConta, new BigDecimal(50.2));
                      System.out.println("O saque foi realizado");
                   obj.criarCadastro("João", "11343378911", "5563941918191", new Date());
                   Integer idContaJoao = obj.criarCliente("11343378911", "123");
                  if (obj.acessarCliente(idContaJoao, "123")) {
  40
                      obj.deposito(idContaJoao, new BigDecimal(75.6));
  41
                      obj.saque(idContaJoao, new BigDecimal(50.8));
  42
                      obj.transferencia(idConta, "11343378911", new BigDecimal(20));
  43
  44
  45
                   List<HashMap<String, String>> extrato = obj.consultaExtrato(idConta);
                   for (int i = 0; i < extrato.size(); i++) {
  47
                      System.out.println(extrato.get(i));
  49
                   List<HashMap<String, String>> extrato1 = obj.consultaExtrato(idContaJoao);
  50
                   for (int i = 0; i < extrato.size(); i++)
  51
                       System.out.println(extrato.get(i));
Console 23 Problems (3) Debug Shell Coverage
<terminated> BancoClient (3) [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (3 de dez de 2019 23:40:37)
O id do cadastro é: 1
O número da conta é: 1
                                                          7 2 Java Application
```

Conclusão

- Sistema distrubuído lógica e fisicamente
- Requisitos do sistema:
 - Abertura de conta
 - Consulta de saldo e extrato da conta
 - Transferências entre contas
 - ▶ Saque ✓
 - Depósito

Obrigado pela atenção!