



Bianca Viana \_CP300547X

Carolina Zanotto \_CP3003817

Deive Leal \_CP3003825



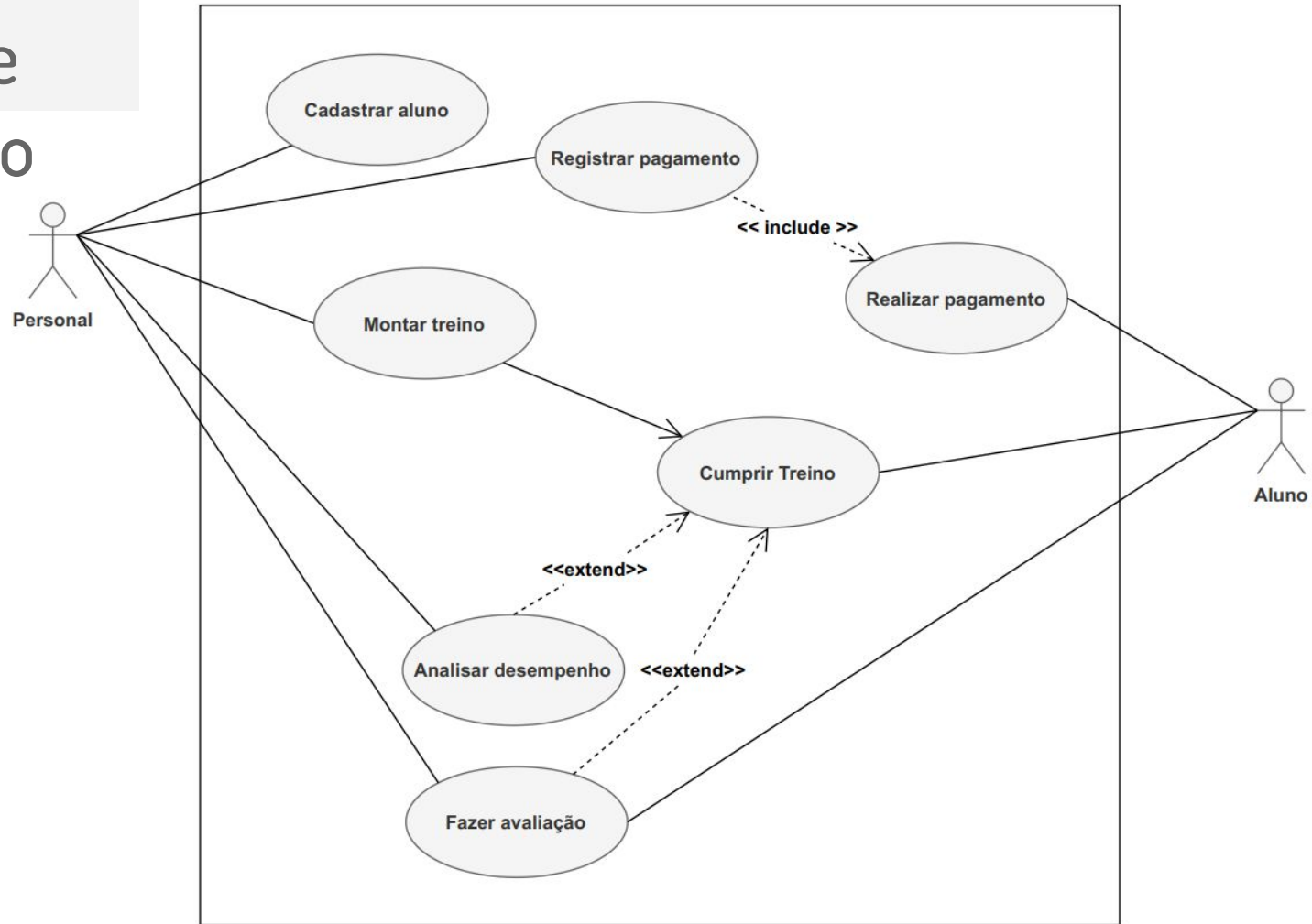
Gerenciador de Carteira de Alunos para Personal Trainer

# Conceito do Projeto

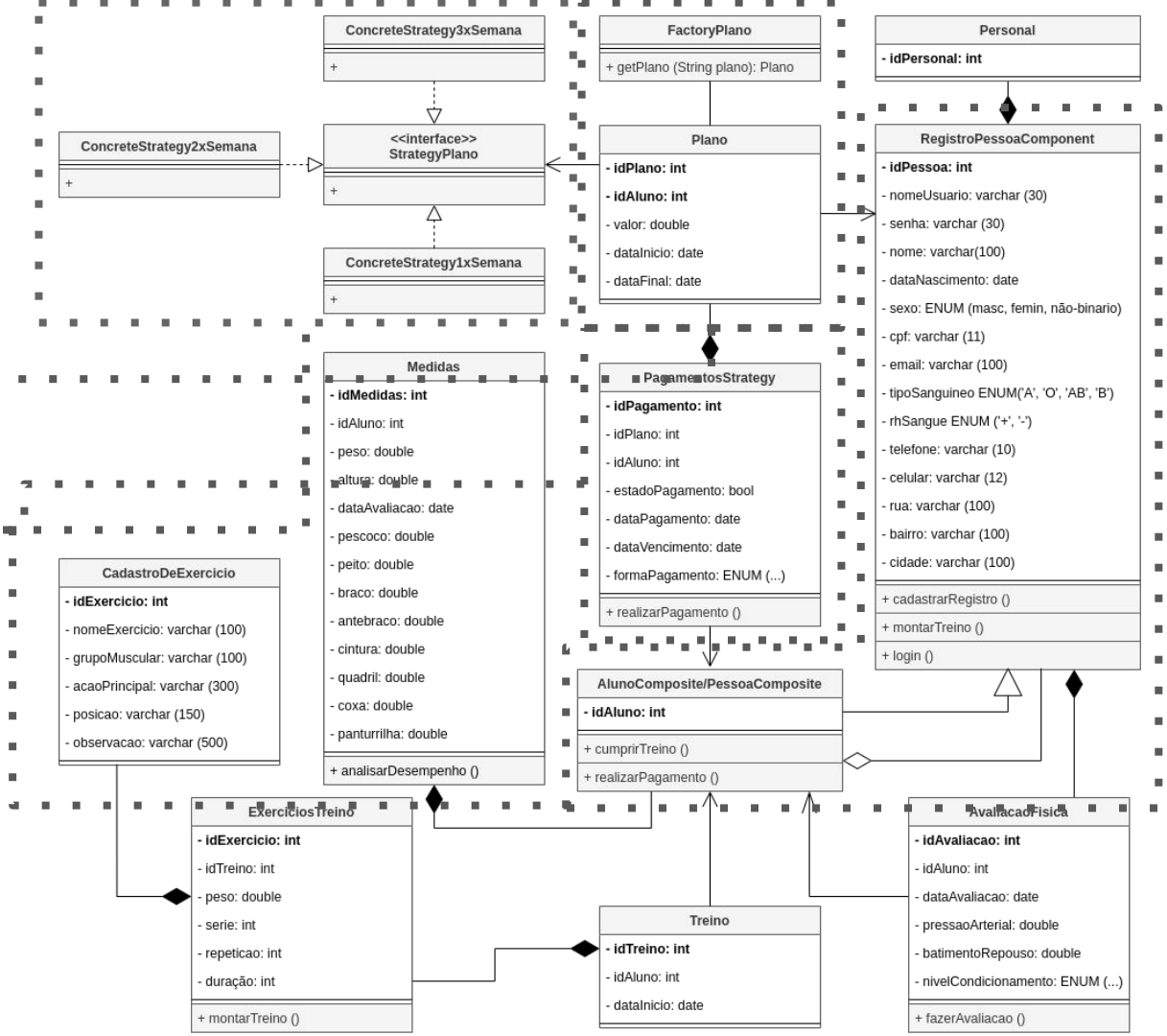
Sistema Efeito tem como premissa assistir treinadores e profissionais liberais, à gerenciar sua carteira de alunos em relação a treinos e desempenho físico. Possibilitar uma visualização de resultados de forma efetiva e funcional, com o adicional de controle financeiro.



# Diagrama de Casos de Uso



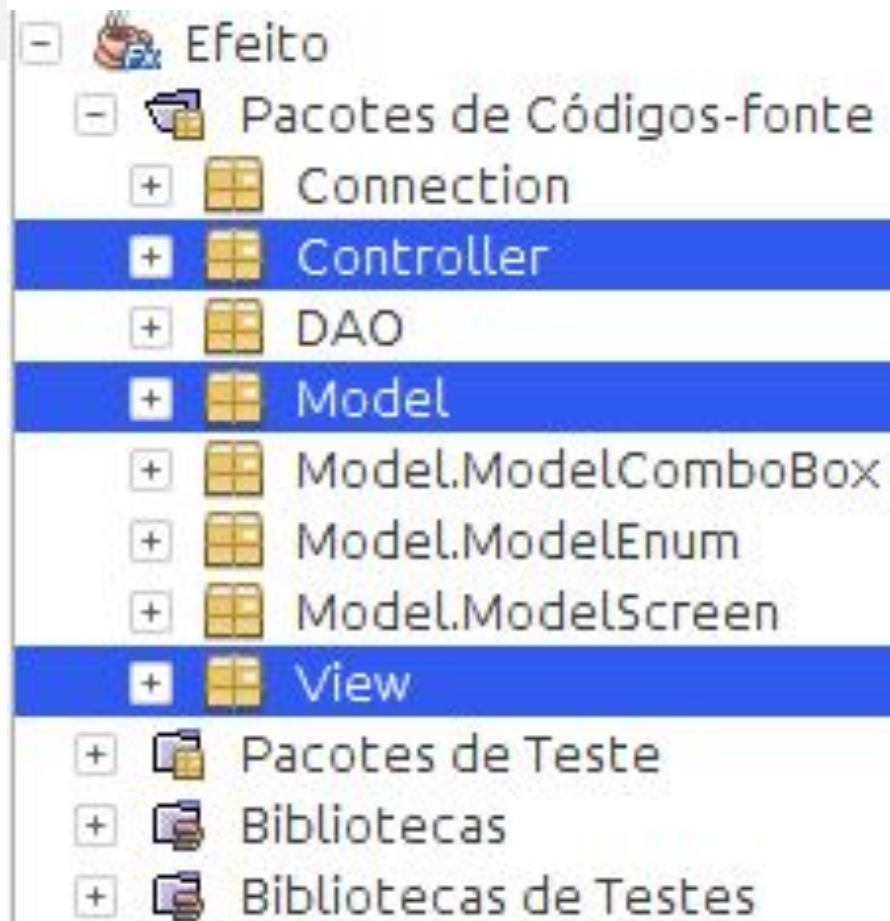
## +Padrões de Projeto



Padrão Single-site  
Orçamento por Aluno  
Acumulado de Pagamento

# Padrão de Arquitetura

MVC



# Padrão de Arquitetura

## MVC

### - Connection

- ConnectionFactoryInterface.java
- ConnectionFactoryMysqlSingleton.java

### - Controller

- CadastaAlunoController.java
- CadastaPlanoController.java
- DadosAlunoCadastradoController.java
- HistoricoAfericoesMedidasController.java
- HistoricoAvaliacaoFisicaController.java
- LoginController.java
- MenuAlunoController.java
- MenuFinanceiroController.java
- MenuPrincipalController.java
- RegistroAvaliacaoFisicaController.java
- RegistroMedidasController.java

### - DAO

- LoginDAO.java
- PlanoDAO.java
- RegistraAlunoDAO.java
- RegistroAvaliacaoFisicaDAO.java
- RegistroMedidasDAO.java

### - Model.ModelComboBox

- QuantidadeMesesCB.java

### - Model.ModelEnum

- CondicionamentoFisicoEnum.java
- RhSangueEnum.java
- SexoEnum.java
- TipoSanguineoEnum.java

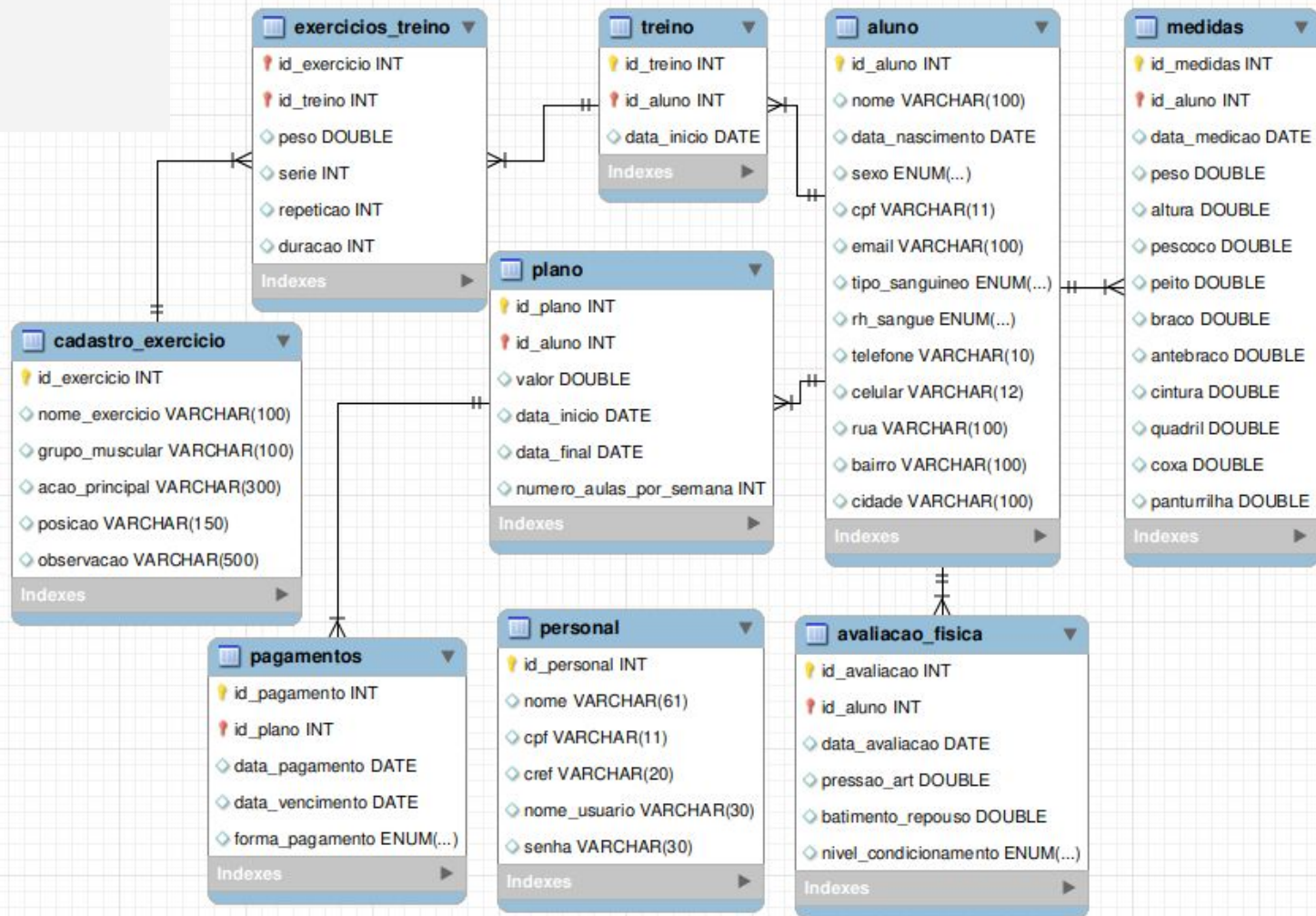
### - Model.ModelScreen

- CadastaAlunoScr.java
- CadastaPlanoScr.java
- DadosAlunoCadastradoScr.java
- HistoricoAfericoesMedidasScr.java
- HistoricoAvaliacaoFisicaScr.java
- LoginScr.java
- MenuAlunoScr.java
- MenuFinanceiroScr.java
- MenuPrincipalScr.java
- RegistroAvaliacaoFisicaScr.java
- RegistroMedidasScr.java

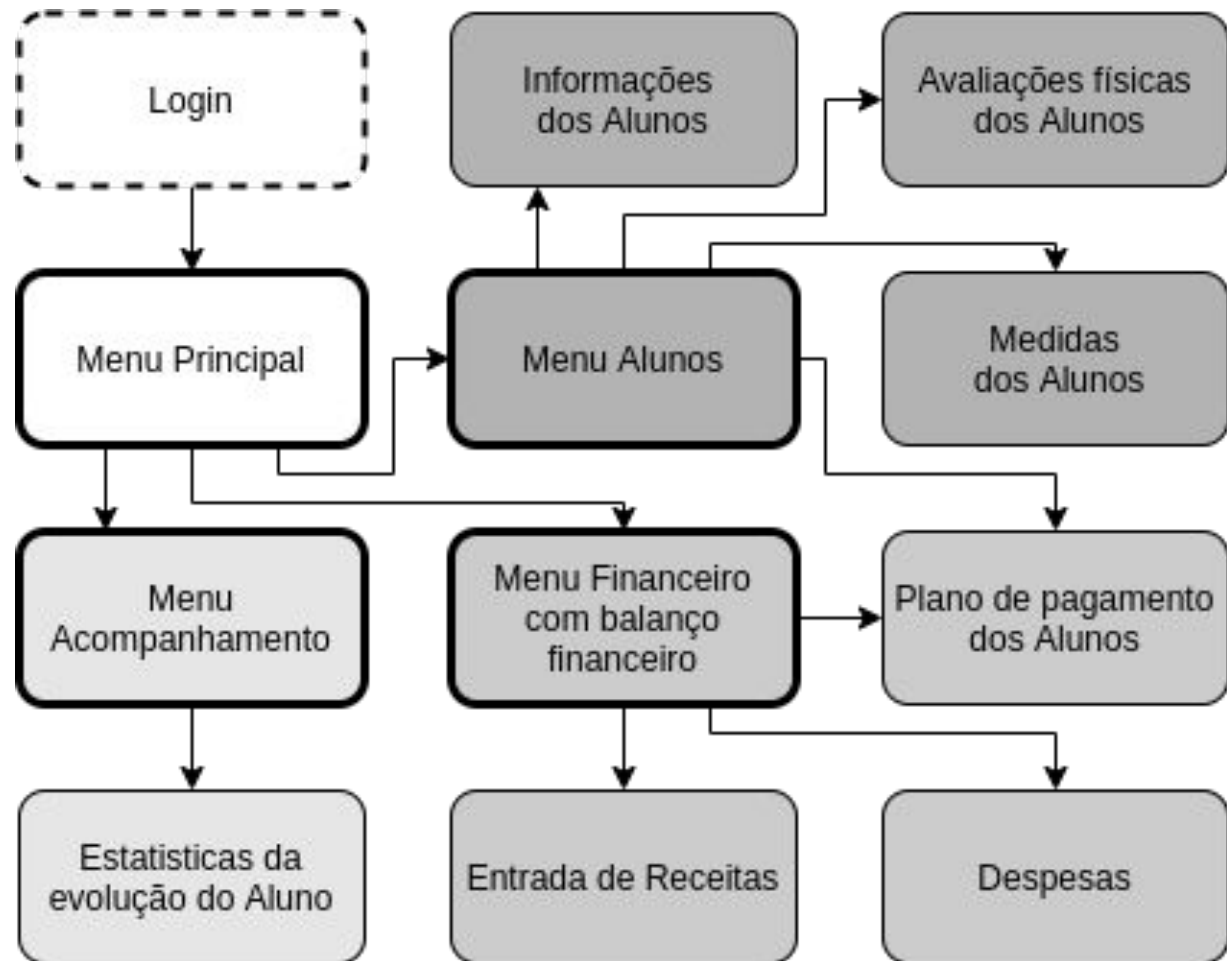
### - View

- CadastaAluno.fxml
- CadastaPlano.fxml
- DadosAlunoCadastrado.fxml
- HistoricoAfericoesMedidas.fxml
- HistoricoAvaliacaoFisica.fxml
- Login.fxml
- MenuAluno.fxml
- MenuFinanceiro.fxml
- MenuPrincipal.fxml
- RegistroAvaliacaoFisica.fxml
- RegistroMedidas.fxml

# DER



# Diagrama de Navegabilidade





# Conceitos P00

## Herança

```
package Model;

import java.util.Date;

public class RegistroPessoa {

    private int id_pessoa;
    private String nome;
    private String nomeUsuario;
    private String senha;
    private Date dataNascimento;
    private Enum sexo;
    private String cpf;
    private String email;
    private Enum tipoSanguineo;
    private Enum rhSangue;
    private String telefone;
    private String celular;
    private String rua;
    private String bairro;
    private String cidade;

    public int getId_pessoa() {
        return id_pessoa;
    }

    public void setId_pessoa(int id_pessoa) {
        this.id_pessoa = id_pessoa;
    }
}

package Model;

import java.util.Date;

public class RegistroAlunos extends RegistroPessoa {

    public RegistroAlunos() {
    }

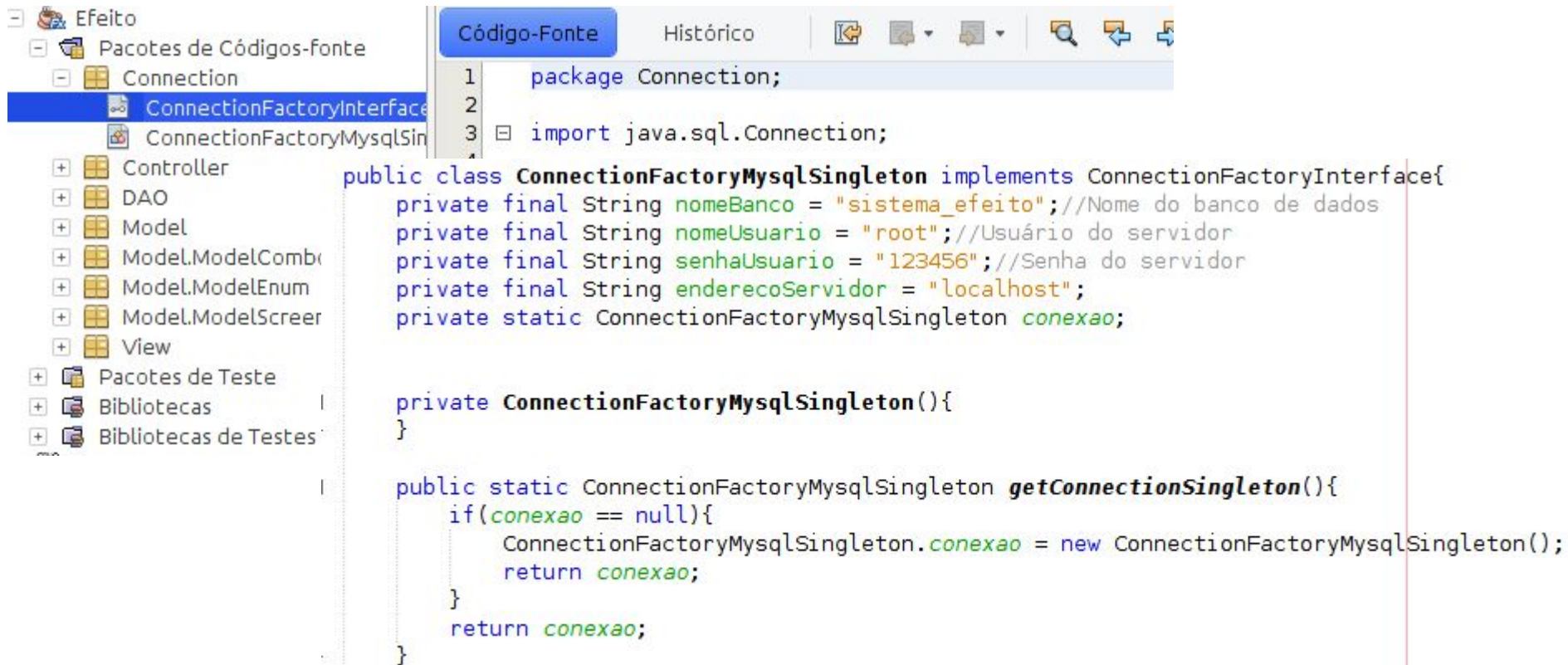
    public RegistroAlunos(String nome, Date dataNascimento,
        Enum rhSangue, String telefone, String celular) {
        super.setNome(nome);
        super.setDataNascimento(dataNascimento);
        super.setSexo(sexo);
        super.setCpf(cpf);
        super.setEmail(email);
        super.setTipoSanguineo(tipoSanguineo);
        super.setRhSangue(rhSangue);
        super.setTelefone(telefone);
        super.setCelular(celular);
        super.setRua(rua);
        super.setBairro(bairro);
        super.setCidade(cidade);
    }

    ;

    public RegistroAlunos(int id_aluno, Date data_medica)
```

# Conceitos P00

## Interface



```
1 package Connection;
2
3 import java.sql.Connection;

public class ConnectionFactoryMysqlSingleton implements ConnectionFactoryInterface{
    private final String nomeBanco = "sistema_efeito";//Nome do banco de dados
    private final String nomeUsuario = "root";//Usuário do servidor
    private final String senhaUsuario = "123456";//Senha do servidor
    private final String enderecoServidor = "localhost";
    private static ConnectionFactoryMysqlSingleton conexao;

    private ConnectionFactoryMysqlSingleton(){
    }

    public static ConnectionFactoryMysqlSingleton getConnectionSingleton(){
        if(conexao == null){
            ConnectionFactoryMysqlSingleton.conexao = new ConnectionFactoryMysqlSingleton();
            return conexao;
        }
        return conexao;
    }
}
```

# Conceitos P00

## Polimorfismo

- Model
  - Pagamentos.java
  - Plano.java
  - RegistroAlunos.java
  - RegistroAvaliacaoFisica.java
  - RegistroMedidas.java
  - RegistroPersonal.java
  - RegistroPessoa.java
  - Util.java
- Model.ModelComboBox
  - QuantidadeMesesCB.java**
- Model.ModelEnum
  - CondicionamentoFisicoEnum.java
  - RhSangueEnum.java
  - SexoEnum.java
  - TipoSanguineoEnum.java
- Model.ModelScreen
  - CadastraAlunoScr.java
  - CadastraPlanoScr.java
  - DadosAlunoCadastradoScr.java
  - HistoricoAfericoesMedidasScr.java
  - HistoricoAvaliacaoFisicaScr.java
  - LoginScr.java
  - MenuAlunoScr.java
  - MenuFinanceiroScr.java
  - MenuPrincipalScr.java
  - RegistroAvaliacaoFisicaScr.java
  - RegistroMedidasScr.java

```
4
5 private int quantidadeMeses;
6 private double valorPlano;
7 private String tipoPlano;
8
9 public QuantidadeMesesCB(int quantidadeMeses, double valorPlano, String tipoPlano) {
10     this.quantidadeMeses = quantidadeMeses;
11     this.valorPlano = valorPlano;
12     this.tipoPlano = tipoPlano;
13 }
14
15 public QuantidadeMesesCB() {...5 linhas }
16
17 public double getValorPlano() {...3 linhas }
18
19 public void setValorPlano(double valorPlano) {...3 linhas }
20
21 public int getQuantidadeMeses() {...3 linhas }
22
23 public void setQuantidadeMeses(int quantidadeMeses) {...3 linhas }
24
25 public String getTipoPlano() {...3 linhas }
26
27 public void setTipoPlano(String tipoPlano) {...3 linhas }
28
29 @Override
30 public String toString() {
31     return getTipoPlano();
32 }
33 }
```