PROGRAMAÇÃO ORIENTADA A OBJETOS

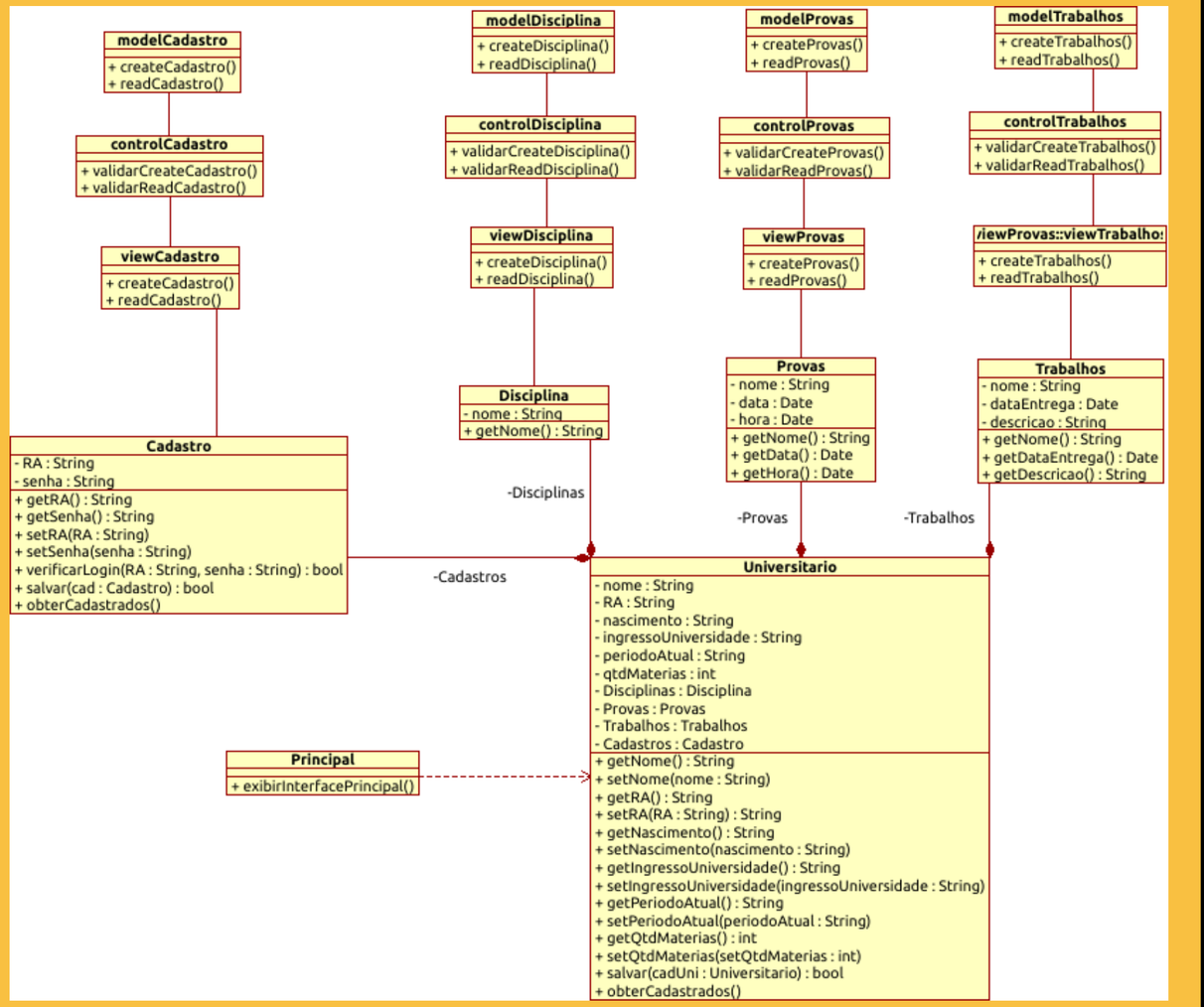
Meu PLANNER

TRABALHO MULTIDISCIPLINAR

Luis Henrique Ferracciu Pagotto Mendes
Maria Eduarda Soares Romana Silva
Tiago Fernandes Soucek

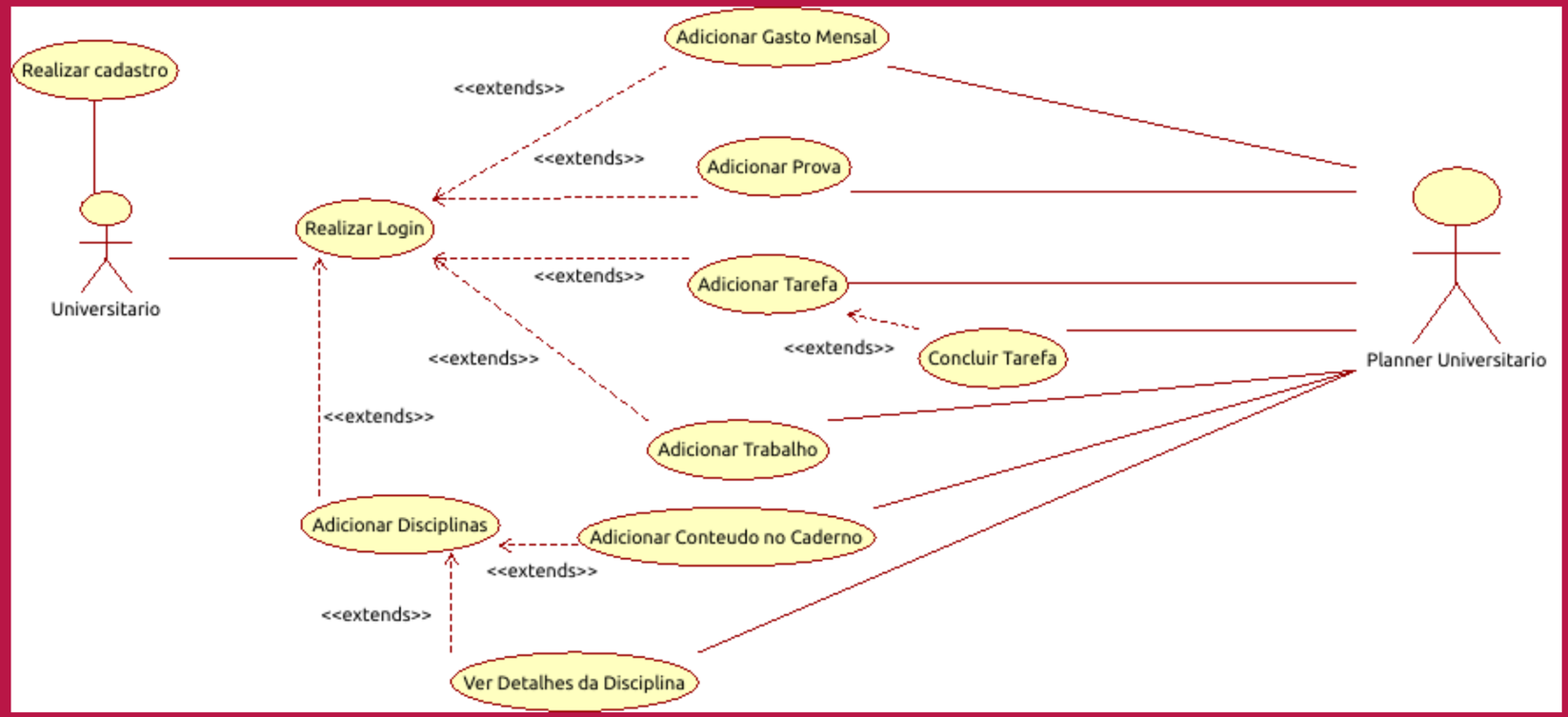
PROGRAMAÇÃO ORIENTADA A OBJETOS

DIAGRAMA DE CLASSES



PROGRAMAÇÃO ORIENTADA A OBJETOS

DIAGRAMA DE CASOS DE USO



POLIMORFISMO

A interface Gasto estabelece um contrato que as classes que a implementam devem seguir. Neste contexto, a única exigência é a implementação do método calcularTotalGasto().

```
public interface IGasto {
    double calcularTotalGasto();
}
```

```java
public class Aluguel implements Gasto {
    private double valor;

    public Aluguel(double valor) {
        this.valor = valor;
    }

    @Override
    public double calcularTotalGasto() {
        return valor;
    }
}

public class Luz implements Gasto {
    private double valor;

    public Luz(double valor) {
        this.valor = valor;
    }

    @Override
    public double calcularTotalGasto() {
        return valor;
    }
}
```

POLIMORFISMO

Cada classe que implementa a interface Gasto, como Aluguel, Luz, Internet, etc., oferece uma implementação específica para o método calcularTotalGasto(). Cada tipo de gasto, portanto, tem conhecimento sobre como calcular seu próprio total

PROGRAMAÇÃO ORIENTADA A OBJETOS

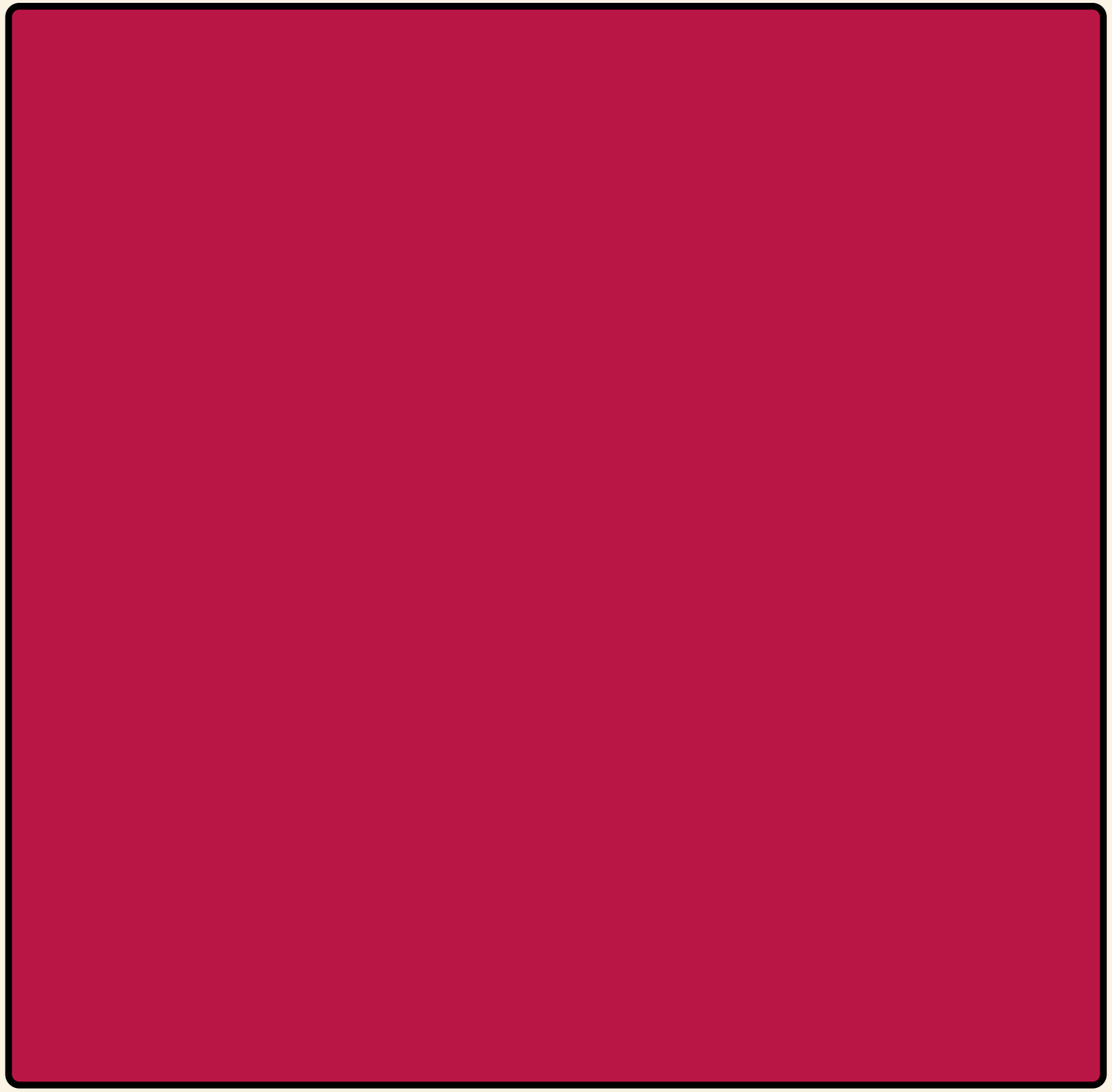
TRATAMENTO EXCEÇÃO PERSONALIZADO

```
        } catch (CamposVaziosException | RAJaCadastradoException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }

    private void validarCampos() throws CamposVaziosException {
        if (tfRACadastro.getText().isEmpty() || pfSenhaCadastro.getPassword().length == 0) {
            throw new CamposVaziosException("Todos os campos devem ser preenchidos.");
        }
    }
    private static class CamposVaziosException extends Exception {
        public CamposVaziosException(String message) {
            super(message);
        }
    }
    public static class RAJaCadastradoException extends Exception {
        public RAJaCadastradoException(String message) {
            super(message);
        }
    }
    public void validarRACadastrado() throws RAJaCadastradoException {
        String raDigitado = tfRACadastro.getText();
        for (Cadastro usuario : Cadastro.obterCadastrados()) {
            if (usuario.getRA().equals(raDigitado)) {
                throw new RAJaCadastradoException("RA já cadastrado. Escolha outro RA.");
            }
        }
    }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

SISTEMA EM EXECUÇÃO



Obrigado !

Luis Henrique Ferracciu Pagotto Mendes
Maria Eduarda Soares Romana Silva
Tiago Fernandes Soucek