

UNIVERSIDADE PAULISTA (UNIP)

CIÊNCIA DA COMPUTAÇÃO

**TRABALHO DE APLICAÇÕES DA LINGUAGEM DE PROGRAMAÇÃO
ORIENTADA A OBJETOS (ALPOO)**

ANDREICK GOMES MAIA - F044289

BEATRIZ ALMEIDA DE LIMA SOUZA - N459JB4

MAICOLN VINICIUS DOS S. SILVA - F0551I7

MATEUS CARVALHO DA SILVA - N508620

SÃO PAULO

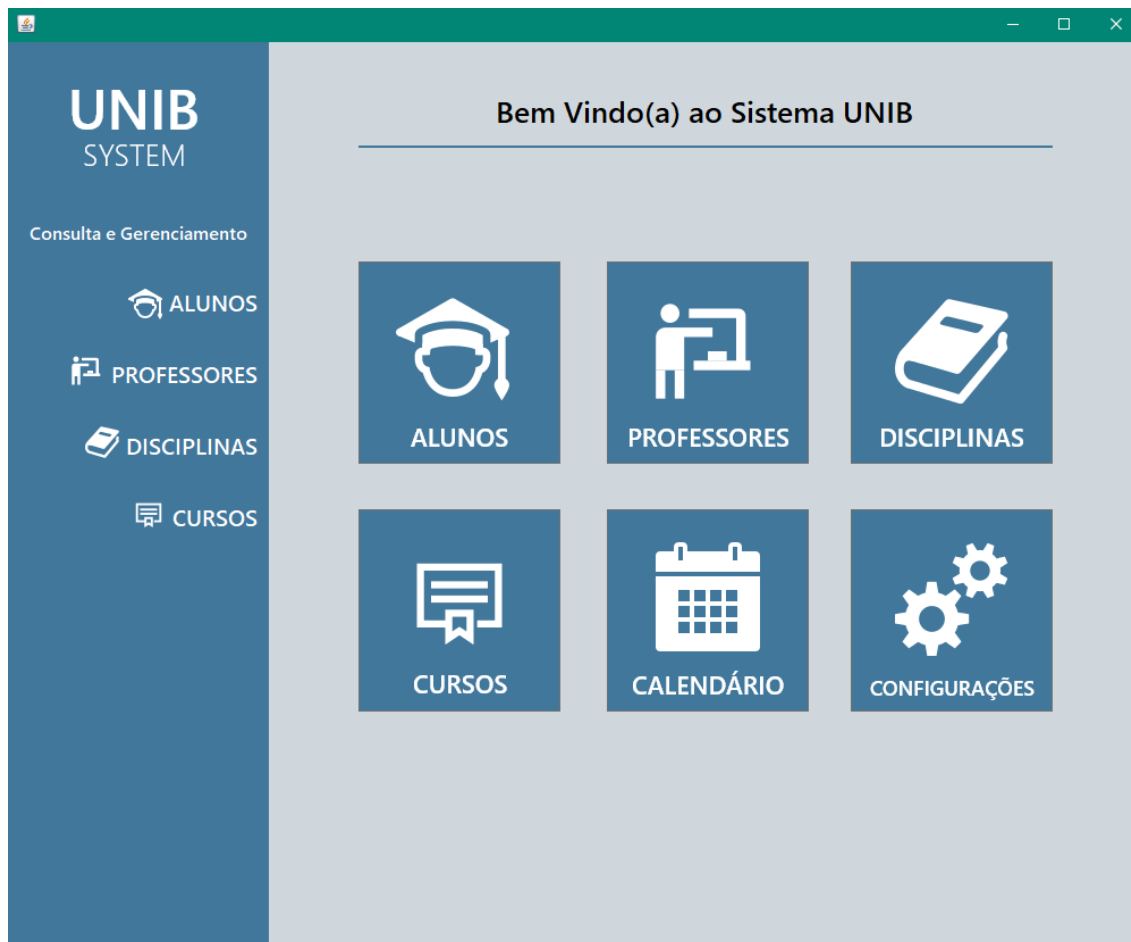
2020

OBJETIVO

Criar um sistema para gerenciamento escolar, utilizando a linguagem Java. O sistema deve apresentar telas como aluno, professor, curso e disciplina, bem como uma tela de buscas integrada com um banco de dados.

DESENVOLVIMENTO

TELAS



UNIB
SYSTEM

Consulta e Gerenciamento

ALUNOS

PROFESSORES

DISCIPLINAS

CURSOS

Consulta e Gerenciamento

🔍

CADASTRAR NOVO

MATRÍCULA	NOME	DATA NASCIMENTO	CURSO
23264	JANAINA SOARES	26 / MARÇO / 2000	CIÊNCIA DA COMPUTAÇÃO
23265	ROBERTO ALMEIDA	17 / JANEIRO / 1989	BIOMEDICINA
23266	NAIARA LIMA	05 / FEVEREIRO / ...	ADMINISTRAÇÃO
23267	LINDOVALDO PEREIRA	25 / MARÇO / 1999	CIÊNCIAS BIOLÓGICAS
23268	PATRÍCIA CARVALHO	25 / DEZEMBRO / 2...	EDUCAÇÃO FÍSICA
23269	JOSÉ DA SILVA	07 / OUTUBRO / 1...	FARMACOLOGIA
23270	BEATRIZ LIMEIRA	02 / NOVENBRO / ...	DIREITO
23271	JENIFER PEREIRA	05 / JULHO / 2002	REDES DE COMPUTADORES
23272	KLEBER CARDOSO	24 / AGOSTO / 1989	SISTEMA DE INFORMAÇÕES
23273	LUIZ CARLOS	05 / Junho / 1977	EDUCAÇÃO FÍSICA
23274	LARISSA SANTIAGO	01 / Janeiro / 1996	CIÊNCIA DA COMPUTAÇÃO
23276	PAO DE BATATA	06 / Maio / 1904	CIÊNCIA DA COMPUTAÇÃO

UNIB
SYSTEM

Consulta e Gerenciamento

ALUNOS

PROFESSORES

DISCIPLINAS

CURSOS

Consulta e Gerenciamento

⬅ Voltar

MATRÍCULA

23267

DATA NASCIMENTO

25 ▾

Março ▾

1999 ▾

NOME

LINDOVALDO PEREIRA

CÓDIGO DO CURSO

NOME DO CURSO

CIÊNCIAS BIOLÓGICAS

DISCIPLINAS

EXCLUIR

ATUALIZAR

UNIB
SYSTEM

Consulta e Gerenciamento

ALUNOS

PROFESSORES

DISCIPLINAS

CURSOS

Consulta e Gerenciamento

🔍

CADASTRAR NOVO

ID	NOME	DATA NASC	TITULO	ESPECIALIDADE
3	JOÃO MARCELO	26/JULHO/2000	MESTRADO	MATEMÁTICA
4	EMÍLIA MARTINS	15/MARÇO/1985	ESPECIALISTA	MEDICINA
5	NÁDIA DE SOUZA	25/FEVEREIRO/...	DOCTORADO	INFORMÁTICA
6	GUSTAVO NOGUEIRA	10/JUNHO/1980	MESTRADO	MATEMÁTICA

UNIB
SYSTEM

Consulta e Gerenciamento

ALUNOS

PROFESSORES

DISCIPLINAS

CURSOS

Consulta e Gerenciamento

< Voltar

IDENTIFICAÇÃO

3

DATA NASCIMENTO

26

Janeiro

2000

NOME

JOÃO MARCELO

ESPECIALIDADE

MATEMÁTICA

TÍTULO

MESTRADO

ENDEREÇO

RUA DAS PEDRAS, 10

CEP

02505-151

BAIRRO

BAIRRO BARROSO

CIDADE

CIDADOSA

ESTADO

CE

TELEFONE

(15) 1561-6516

EMAIL

JOÃO@EMAIL.COM

EXCLUIR

ATUALIZAR

UNIB
SYSTEM

Consulta e Gerenciamento

ALUNOS

PROFESSORES

DISCIPLINAS

CURSOS

Consulta e Gerenciamento

Q

CADASTRAR NOVO

CÓD. DISCIPLI...	NOME	CARGA HORÁRIA	AULAS SEMANAIS	CÓD. CURSO
1	PROGRAMAÇÃO	100.0	5	1
2	BANCO DE DADOS	100.0	3	1
3	CÁLCULO	50.0	2	2
4	ARTE	50.0	2	5
5	CIÊNCIAS	100.0	5	6
6	TCC	100.0	3	7

UNIB
SYSTEM

Consulta e Gerenciamento

ALUNOS

PROFESSORES

DISCIPLINAS

CURSOS

Consulta e Gerenciamento

Voltar

DISCIPLINA

ARTE

CÓDIGO

4

CARGA HORÁRIA

50.0

AULAS SEMANAIS

2

CÓDIGO CURSO

5

EXCLUIR

ATUALIZAR



CÓDIGO – PRINCIPAIS MÉTODOS

```
private void cadastraAluno() {
    String nome = tf_nome_aluno.getText();
    String dia = tf_dia_aluno.getSelectedItem().toString();
    String mes = tf_mes_aluno.getSelectedItem().toString();
    String ano = tf_ano_aluno.getSelectedItem().toString();
    String dataNasc = (dia + " / " + mes + " / " + ano );
    int codCurso = Integer.parseInt(tf_codCurso_aluno.getText());
    String nomeCurso = tf_nomeCurso_aluno.getText();

    try{
        Aluno aluno = new Aluno(nome, dataNasc, codCurso, nomeCurso);
        AlunoDAO dao = new AlunoDAO();
        int matricula = dao.create(aluno);
        if(matricula > 0) aluno.setMatricula(matricula);
        System.out.println("Aluno Cadastrado");
        System.out.println(aluno.toString());

    } catch (Exception ex) {
        System.out.println("Erro no cadastro do Aluno");
    }
}

private void updateAluno() {
    String nome = tf_nome_aluno.getText();
    String dia = tf_dia_aluno.getSelectedItem().toString();
    String mes = tf_mes_aluno.getSelectedItem().toString();
    String ano = tf_ano_aluno.getSelectedItem().toString();
    String dataNasc = (dia + " / " + mes + " / " + ano );
    int codCurso = Integer.parseInt(tf_codCurso_aluno.getText());

    try{
        Aluno aluno = new Aluno(nome, dataNasc, codCurso);
        aluno.setMatricula(Integer.parseInt(tf_matricula_aluno.getText()));
        dao.AlunoDAO.getInstance().update(aluno);
        System.out.println("Aluno Atualizado");
        System.out.println(aluno.toString());

    } catch (Exception ex) {
        System.out.println("Erro na atualização do Aluno");
    }
}
```

```

private void deleteAluno() {
    dao.AlunoDAO.getInstance().delete((int)
        tb_busca_aluno.getModel().getValueAt(tb_busca_aluno.getSelectedRow(), 0));
}

public List<Aluno> read() {
    Connection con = ConnectionFactory.getConnection();
    PreparedStatement pst = null;
    ResultSet rs = null;

    List<Aluno> alunos = new ArrayList<>();
    try {
        pst = con.prepareStatement("SELECT A.Matricula, A.Nome, A.DataNasc, C.NomeC "
            + "FROM Alunos A "
            + "INNER JOIN Cursos C "
            + "ON A.CodCurso = C.CodCurso");

        rs = pst.executeQuery();

        while (rs.next()) {
            Aluno aluno = new Aluno(0, "", "", "");

            aluno.setMatricula(rs.getInt("Matricula"));
            aluno.setNome(rs.getString("Nome"));
            aluno.setDataNasc(rs.getString("DataNasc"));
            aluno.setNomeCurso(rs.getString("NomeC"));
            alunos.add(aluno);
        }
    } catch (SQLException ex) {
        Logger.getLogger(AlunoDAO.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        ConnectionFactory.closeConnection(con, pst, rs);
    }
    return alunos;
}

```



```
public void update(Aluno a){
    Connection con = ConnectionFactory.getConnection();
    PreparedStatement pst = null;
    ResultSet rs = null;

    String sql = "UPDATE Alunos SET Nome = ?, DataNasc = ?, CodCurso = ? where Matricula = ?";

    try{
        pst = con.prepareStatement(sql);

        pst.setString(1, a.getNome());
        pst.setString(2, a.getDataNasc());
        pst.setInt(3, a.getCodCurso());
        pst.setInt(4, a.getMatricula());

        pst.execute();
        JOptionPane.showMessageDialog(null, "Aluno atualizado com sucesso!");

    } catch(SQLException ex) {
        throw new RuntimeException("Erro no update");
    } finally{
        ConnectionFactory.closeConnection(con, pst, rs);
    }
}
```

```
public void delete (int id) {
    Connection con = ConnectionFactory.getConnection();
    PreparedStatement pst = null;
    ResultSet rs = null;

    String sql = "DELETE FROM Alunos WHERE Matricula = ?";

    try{
        pst = con.prepareStatement(sql);
        pst.setInt(1, id);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Aluno excluido com sucesso!");

    } catch(SQLException ex){
        throw new RuntimeException("Erro no Select");
    } finally{
        ConnectionFactory.closeConnection(con, pst, rs);
    }
}
```