

O QUE FAZ	BANCO DE DADOS	O QUE FAZ	VALIDAÇÃO	O QUE FAZ	PEGAR DADOS NO HTML
cria o bd	<pre>engine = create_engine("mysql://root:fatec2021@localhost/fatec") if not database_exists(engine.url): create_database(engine.url) app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:fatec2021@localhost/fatec' app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True db = SQLAlchemy(app)</pre>	VALIDA O LOGIN	<pre>if usuário: #session['logged_in'] = True session['nome'] = usuário['nome'] session['senha'] = usuário['senha'] msg = 'Logged in successfully!' #flash(msg) return redirect(url_for('show')) # return render_template('show.html', msg=msg) else: msg = 'Nome Usuário e/ou Senha Incorretos!' flash(msg) #return 'Nome e ou senha invalidos' return render_template('login.html', msg=msg)</pre>	PEGA O CONTEUDO DO FRONT PARA O LOGIN	<pre>@app.route('/') @app.route('/login', methods=['GET', 'POST']) def login(): msg = "" if request.method == 'POST':# and 'nome' in request.form and 'pwd' in request.form: nome = request.form['nome'] senha = request.form['pwd'] #CONEXAO COM O BD"</pre>
cria tabela user	<pre>class user(db.Model): id = db.Column(db.Integer, primary_key=True, autoincrement=True) matricula = db.Column(db.String(30), primary_key=True) nome = db.Column(db.String(30)) email = db.Column(db.String(30)) senha = db.Column(db.String(15))</pre>	FAZ O LOGOUT	<pre>@app.route('/logout') def logout(): #session.pop('logged_in', None) session.pop('nome', None) return redirect(url_for('login'))</pre>	PEGA O CONTEUDO DO FRONT PARA O CADASTRO	<pre>@app.route('/cadastro', methods=['GET', 'POST']) def cadastro(): msg = "" if request.method == 'POST': matricula = request.form['matricula'] nome = request.form['nome'] email = request.form['email'] senha = request.form['pwd']</pre>
cria tabela postagem	<pre>class postagem(db.Model): id = db.Column(db.Integer, primary_key=True, autoincrement=True) data = db.Column(db.DateTime(timezone=True), server_default=func.now()) titulo = db.Column(db.String(100)) conteudo = db.Column(db.String(2000)) nome = db.Column(db.String(30)) db.create_all()</pre>	VALIDA SENHA DIGITADA	<pre>msg = "" if senha != request.form['pwd']: msg = "SENHA DIGITADA NÃO CONFERE" flash(msg) return render_template('cadastro.html') #return "SENHA DIGITADA NÃO CONFERE"</pre>	PEGA O CONTEUDO DAS MENSAGENS	<pre>@app.route('/postar', methods=['POST', 'GET']) def postar(): if request.method == 'POST': nome = session['nome'] titulo = request.form['titulo'] conteudo = request.form['conteudo']</pre>
AUTENTICAÇÃO DO BD	<pre>app.secret_key = 'your secret key' app.config['MYSQL_HOST'] = 'localhost' app.config['MYSQL_USER'] = 'root' app.config['MYSQL_PASSWORD'] = 'fatec2021' app.config['MYSQL_DB'] = 'fatec'</pre>	VALIDAÇÃO DOS CAMPOS DA TELA DE CADASTRO	<pre>if user: msg = 'Conta já cadastrada!' flash(msg) return render_template('cadastro.html') elif not re.match("^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,6}\$", email): msg = 'Endereço de email inválido!' flash(msg) return render_template('cadastro.html') elif not re.match("^[a-zA-Z0-9]+", nome): msg = 'Nome do usuário deve conter somente caracteres e numeros!' flash(msg) return render_template('cadastro.html') elif not nome or not senha or not email: msg = 'Preencha o formulario!' flash(msg) return render_template('cadastro.html')</pre>		
CONSULTA TUDO DA TABELA USER ONDE O NOME E SENHA É IGUAL AO DIGITADO	<pre>cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor) cursor.execute('SELECT * FROM user WHERE nome = %s AND senha = %s', (nome, senha,)) usuário = cursor.fetchone()</pre>	index	<pre>@app.route('/login') def index(): return render_template('login.html')</pre>		
SELECIONA MATRICULA NA TABELA USER ONDE MATRICULA É IGUAL AO DIGITADO	<pre>cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor) cursor.execute('SELECT matricula FROM user WHERE matricula = %s', (matricula,)) user = cursor.fetchone()</pre>	VERIFICA SE O TITULO DA MENSAGEM É TRUE	<pre>if not titulo: flash("Titulo é obrigatorio") #INSERIR MENSAGENS FLASH NO HTML return render_template('postar.html')</pre>		
INSERE NA TABELA USER OS VALORES DENTRO PARENTeses. CASO A VALIDAÇÃO FEITA ANTERIORMENTE SEJA TRUE	<pre>cursor.execute('INSERT INTO user (nome,email,matricula,senha) values (%s, %s, %s, %s)', (nome, email, matricula, senha,)) mysql.connection.commit() msg = 'Cadastro Efetuado com Sucesso!' flash(msg) return redirect(url_for('login')) elif request.method == 'POST': msg = 'Por favor preencha o formulario' flash(msg) return redirect(url_for('login'))</pre>	BOTAO CADASTRO DA TELA DE LOGIN	<pre>@app.route('/cadastro') def cadastro(): return render_template('cadastro.html')</pre>		
INSERE NA TABELA POSTAGEM OS VALORES DENTRO DO PARENTeses, ESTANDO O TITULO DA MENSAGEM OK.	<pre>cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor) cursor.execute('INSERT INTO postagem (titulo,conteudo,nome) values (%s, %s, %s)', (titulo, conteudo,nome,)) mysql.connection.commit() return redirect(url_for('show'))</pre>	Chamar a tela de postar.html pelo botao de +.	<pre>@app.route('/leitura') def leitura(): return render_template('postar.html')</pre>		
CONSULTA AS MENSAGENS DA TABELA POSTAGEM	<pre>@app.route('/tela_cheia') def show(): nome = session['nome'] cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor) cursor.execute('SELECT * FROM postagem ') postagem = cursor.fetchall() cursor.close() return render_template('visualizacao.html', postagem=postagem)</pre>				