Relatório Técnico – Portal de Atividades Complementares

# 📌 Objetivo do Sistema

Criar um portal web onde alunos possam:  
- Fazer login (simulado);  
- Enviar atividades complementares;  
- Visualizar histórico de envios;  
- A secretaria possa avaliar os documentos e gerar relatórios.

# 🗂️ Estrutura do Projeto

Portal\_Atividades/  
├── src/  
│ ├── app.js # Arquivo principal da aplicação  
│ ├── routes/  
│ │ ├── auth.js # Rota para autenticação (simulada)  
│ │ └── dashboard.js # Rota principal protegida  
│ ├── views/  
│ │ └── dashboard.ejs # Página inicial do painel  
│ ├── public/ # Arquivos estáticos (CSS, imagens)  
│ └── database/  
│ └── connection.js # (Futuro) Conexão com banco de dados  
├── .env # Variáveis de ambiente  
├── package.json  
└── README.md

# ⚙️ Principais Dependências

- express: servidor web.  
- passport, passport-oauth2: autenticação (será simulada).  
- express-session: gerenciamento de sessões.  
- dotenv: carrega variáveis de ambiente do .env.  
- ejs: motor de renderização de templates HTML.

# 📂 .env – Simulação Local

PORT=3000  
SESSION\_SECRET=segredo\_super\_secreto  
  
# Comentado: Integração real com Moodle  
# MOODLE\_BASE\_URL=https://moodle.pucrs.br  
# CLIENT\_ID=seu\_client\_id  
# CLIENT\_SECRET=seu\_client\_secret  
# CALLBACK\_URL=http://localhost:3000/auth/moodle/callback  
# AUTH\_URL=https://moodle.pucrs.br/admin/oauth2authorize.php  
# TOKEN\_URL=https://moodle.pucrs.br/admin/oauth2/token.php

# 🧠 app.js – Explicação Comentada

require('dotenv').config();  
const express = require('express');  
const session = require('express-session');  
const path = require('path');  
  
const app = express();  
  
// Configuração de sessões  
app.use(session({  
 secret: process.env.SESSION\_SECRET,  
 resave: false,  
 saveUninitialized: false  
}));  
  
// Middleware para simulação de login  
app.use((req, res, next) => {  
 req.user = {  
 id: 1,  
 name: 'Usuário Simulado',  
 email: 'simulado@teste.com'  
 };  
 next();  
});  
  
// Configuração do motor de visualização  
app.set('view engine', 'ejs');  
app.set('views', path.join(\_\_dirname, 'views'));  
  
// Arquivos estáticos (CSS, JS etc.)  
app.use(express.static(path.join(\_\_dirname, 'public')));  
  
// Rota principal  
app.get('/', (req, res) => {  
 res.render('dashboard', { user: req.user });  
});  
  
const PORT = process.env.PORT || 3000;  
app.listen(PORT, () => {  
 console.log(`Servidor rodando em http://localhost:${PORT}`);  
});

## 🧾 Comentários:

- req.user é definido manualmente, simulando um login.  
- A autenticação real com Moodle está desabilitada para testes.  
- O dashboard é carregado com o usuário simulado.

# 🖼️ views/dashboard.ejs

<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
 <meta charset="UTF-8">  
 <title>Portal de Atividades</title>  
</head>  
<body>  
 <h1>Bem-vindo, <%= user.name %></h1>  
 <p>Email: <%= user.email %></p>  
 <p>Aqui será exibido o histórico de envios e o formulário.</p>  
</body>  
</html>

# 🧪 Simulação de Uso

1. Execute com:  
 node src/app.js  
  
2. Acesse: http://localhost:3000  
  
3. Você verá a mensagem com o nome e email do usuário simulado.

# ✅ Próximos Passos

- Criar formulário de envio de atividades.  
- Salvar dados em um banco (MySQL).  
- Tela para secretarias visualizarem e aprovarem.  
- Habilitar autenticação real com o Moodle (OAuth2).