

1 Projeto: Controle de acesso via aplicativo móvel

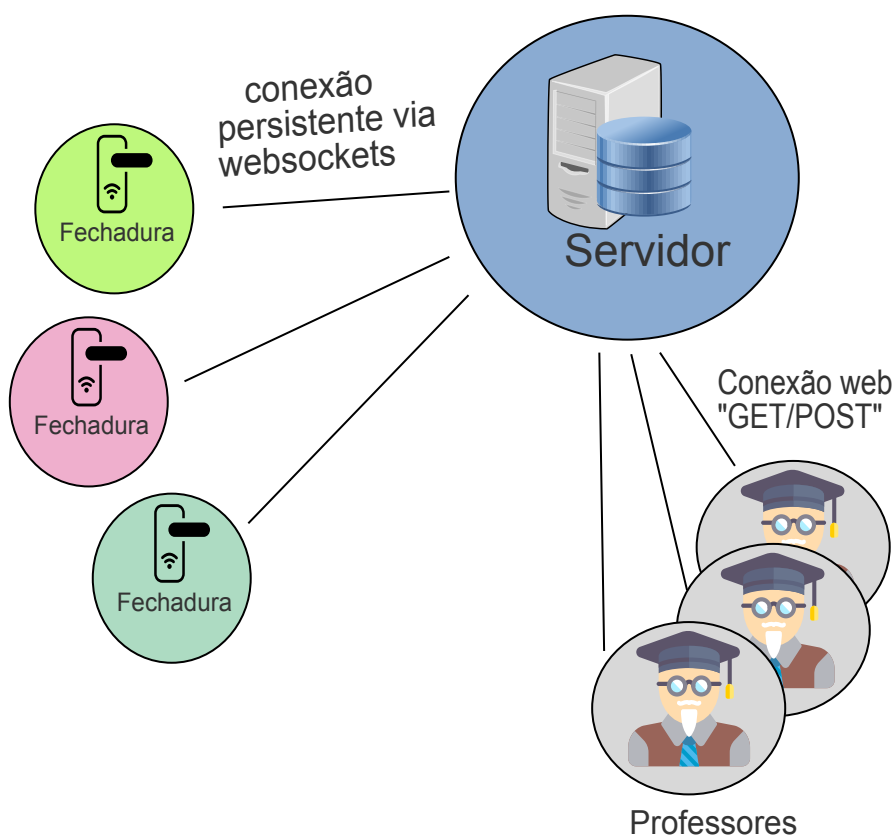
1.1 Introdução

Na universidade existem muitos professores e salas de aula, laboratórios e de permanência. O acesso às salas é feito utilizando chaves em posse dos professores ou mediante solicitação à segurança do campus. O processo é muitas vezes demorado, ineficiente e propenso a falhas com salas sendo deixadas abertas sem fiscalização, professores aguardando a abertura da sala ou mesmo chaves sendo perdidas.

O objetivo deste trabalho é realizar o controle de acesso às salas utilizando um aplicativo móvel, um servidor de aplicação e fechaduras inteligentes.

1.1.1 Visão geral

O sistema é dividido em fechaduras, usuários (usando aplicativos móveis) e o servidor de aplicação. As fechaduras tem conexão persistente com o servidor via websockets e apenas aguardam a ordem do servidor para abrir a porta. Os usuários por sua vez utilizam um aplicativo móvel para fazer uma requisição de abertura de porta (que pode ser aceita ou negada pelo servidor, dependendo se este possui autorização).



1.1.2 Fechaduras Inteligentes

São dispositivos que conectam na rede WiFi da universidade e após obter um IP, conectam no servidor de aplicação via websockets (digamos na porta 10.000) e enviam suas credenciais (`id_fechadura`, `senha_fechadura`). Após autorizada, a fechadura pode receber do servidor o comando `ABRE` para então para abrir a porta (acionamento de motor ou solenoide).

Uma forma de implementar a fechadura é utilizando o microcontrolador ESP8266/ESP32 visto que estes já possuem módulo WiFi embutido. Mas para propósitos de implementação neste trabalhos as fechaduras serão codificadas como processos em nodejs. A idéia é via linha de comando chamar: `node fechadura.js sala_c120 senha123` e neste caso a o programa `fechadura.js` recebe os parâmetros ID da sala e senha da sala para serem utilizados junto ao servidor (exemplo de `process.argv` abaixo). Podemos abrir diversas janelas e disparar diversas fechaduras inteligentes, cada qual com seu ID e senha.

```
// node fechadura.js id_sala senha_sala

if (process.argv.length !== 4)
{
    console.log('forma de usar:')
    console.log('node fechadura ID_FECHADURA SENHA_FECHADURA');
    process.exit();
}

let id = process.argv[2];
let passwd = process.argv[3];
```

Em <https://www.npmjs.com/package/websocket> existe um pacote para criar um cliente websocket in node. Utilize essa biblioteca para conectar no servidor de aplicação e enviar (empacotado como um objetivo JSON) as credenciais da fechadura.

1.1.3 Servidor de aplicação

O servidor de aplicação é ao mesmo tempo servidor de websockets para as conexões das fechaduras e servidor WEB para as requisições dos aplicativos móveis. O servidor também armazena (mongodb) os dados dos usuários e das fechaduras em coleções distintas. A coleção fechaduras deve possuir um ID, senha (da fechadura) e a coleção usuários deve possuir o ID e senha (do usuário) e a lista de fechaduras que ele é capaz de acessar (pode ser uma lista dos IDs das fechaduras ou mesmo ID da fechadura e o horário que ele pode acessar a sala).

1.1.4 Aplicativo móvel

O aplicativo móvel acessa o servidor e envia as credenciais do usuário via POST. Caso o usuário seja validado ele recebe a listagem das salas em que está apto a acessar. O aplicativo cria dinamicamente um botão para representar cada sala dessa lista. Ao pressionar um desses botões o aplicativo envia via GET uma requisição para abrir a sala. O servidor retorna para o aplicativo móvel o status da requisição (Sucesso ou Falha) que é apresentado na parte inferior do aplicativo.

PS: Utilize a biblioteca axios para gerenciar a conexão e também o vue para criar o frontend.

Estude o exemplo 7_render_cliente_vue da unidade 1. O aplicativo pode ser construído utilizando apenas **cordova** ou **PWA**.

1.2 Apresentação

O trabalho deve ser defendido para o professor até a data de 19/12 em horário e local marcado com o professor. As equipes podem ter até 3 integrantes.