Relatório do Projeto: Controle de Dispositivos via Socket

Luis Guilherme Godim da Fonseca - 21.2.8007

5 Fevereiro, 2024

Resumo

Este relatório detalha a implementação de um sistema de controle remoto de dispositivos inteligentes, especificamente lâmpadas e ar-condicionados. A comunicação entre os componentes é estabelecida via sockets, com o uso da linguagem Python. Em sintese, a aplicação consiste em uma ferramenta para modificação, criação e gestão de dispositivos por comunicação local com salvamento de configurações de usuário por meio de arquivos CSV.

Estrutura do Projeto

1. Servidor Central

Descrição:

Gerencia as conexões dos clientes e valida suas autenticações. Utiliza threads para suportar múltiplos clientes simultaneamente.

Implementação:

Escrito em Python utilizando a biblioteca socket e configurado para operar na porta 20001 com o IP '0.0.0.0' (ouvindo todas as interfaces disponíveis), este script tem como objetivo primordial validar a existência do usuário. Essa validação é realizada por meio de uma tupla de booleanos. Caso o usuário não exista, o script cria uma nova entrada utilizando os dados fornecidos e os registra em dois arquivos distintos: 'client.csv' e 'client-ar.csv'.

Ambos os arquivos servem ao mesmo propósito, mas cada um é designado para armazenar informações específicas sobre um tipo de dispositivo. O arquivo denominado "client-ar" guarda os dados do cliente que possui pelo menos um ar condicionado registrado, enquanto o arquivo "client" mantém informações sobre lâmpadas. As colunas presentes nesses arquivos são (user-login, ip, e devices-list), representando respectivamente o nome de usuário, endereço IP e a lista de dispositivos associados.

Cabe destacar que os dados registrados consistem em identificadores únicos criados para cada dispositivo, proporcionando uma organização eficiente e facilitando a gestão dos dispositivos associados a cada usuário.

2. Servidor da Lâmpada

Descrição:

Responsável pelo controle exclusivo de lâmpadas. Aceita comandos do cliente para ligar/desligar, alterar cores e obter informações sobre dispositivos. A cada iteração e mudança do cliente os dados de arquivo são atualizados para que configuração não seja perdida.

Implementação:

Estruturado em classes, com uma classe `Lampada` para cada dispositivo. Utiliza arquivos CSV (`lamp.csv`) para armazenar informações sobre dispositivos. Utiliza a Porta: 20003 e o IP Dummy: '0.0.0.0' para poder receber conexões de multiplas faixas. Vale ressaltar que as faixas de IP's liberados para acessar a aplicação são restringidas para funcionar dentro do intervalo (192.168.0.0 a 192.168.255.255).

3. Servidor do Ar-Condicionado

Descrição:

Encarregado do controle exclusivo de ar-condicionados. Aceita comandos do cliente para ligar/desligar, alterar temperatura e obter informações sobre dispositivos.

Implementação:

Similar à implementação do servidor da lâmpada, mas com funcionalidades específicas para o ar-condicionado. Utiliza arquivos CSV (`ar.csv`) para armazenar informações sobre dispositivos. Utiliza a Porta: 20002 e o IP Dummy: '0.0.0.0' para poder receber conexões de multiplas faixas.

4. Interface do Cliente

Descrição:

Interface interativa para que os usuários controlem dispositivos. Solicita informações como nome de usuário, dispositivo a ser controlado e operação desejada.

Implementação:

Escrita em Python, utiliza a biblioteca socket para se comunicar com os servidores. A implementação atual utiliza a porta: 20001 e o endereço IPv4 da minha máquina: '192.168.0.101'. Vale ressaltar que, o IP selecionado na aplicação como Host deve ser o mesmo que o endereço IPv4 da máquina servidor para que a aplicação funcione localmente. Idealmente, seria mais interessante obter o endereço IPv4 da máquina cliente automaticamente para facilitar uso do usuário, porém não consegui obter esse resultado.

Comunicação

A comunicação entre os componentes é estabelecida por meio de sockets. Os servidores central, da lâmpada e do ar-condicionado escutam em portas específicas e estão associados a um IP de exemplo ('192.168.0.101'). A interface do cliente se conecta ao servidor central na mesma porta e IP. Os clientes são autenticados, e, uma vez validados, podem enviar comandos para os servidores dedicados.

Resultados Obtidos

Autenticação de Usuário:

Os clientes são validados com sucesso, evitando colisões de nomes de usuário.

Controle de Dispositivos:

Operações como ligar/desligar, alterar configurações e criar novos dispositivos foram implementadas com sucesso.

Tratamento de Exceções:

Implementado tratamento de exceções para garantir entrada de dados válida dos usuários (o tipo dos dados é muito importante para comunicação via socket e foi um desafio devido à característica da linguagem Python de não tipagem de variáveis).

Relatórios de Status:

Mensagens informativas e relatórios de status são disponibilizados aos usuários para proporcionar uma experiência interativa. No entanto, essas interações ocorrem via terminal, o que pode ser pouco intuitivo. Idealmente, o projeto deveria ser implementado com interfaces, abrindo perspectivas interessantes para o desenvolvimento futuro do projeto.

Conclusões

Este projeto teve como objetivo a aplicação prática de conceitos de Redes de Computadores por meio da implementação de um sistema de controle de dispositivos inteligentes. Embora tenha alcançado êxito em muitos aspectos, algumas áreas carecem de aprimoramento. A comunicação via socket, por exemplo, poderia ser mais amigável, evitando depender excessivamente do conhecimento técnico do usuário. Além disso, a interface via terminal, embora funcional, pode apresentar desafios de acessibilidade para determinados usuários. Apesar desses pontos a serem refinados, a realização do projeto proporcionou valiosas observações sobre a integração prática de conceitos de redes em ambientes de controle de dispositivos inteligentes. O próximo passo sugerido seria a consideração de interfaces mais intuitivas e acessíveis, visando aprimorar a experiência do usuário e expandir a usabilidade do sistema.