Relatório de Automação — Central de Comandos via Telnet para Consulta Acadêmica

Discentes: Carolina Soares Freitas e Júlia Pereira Hallal

📝 Tarefa escolhida e motivação para automatizá-la

A proposta inicial deste projeto era automatizar o processo de **consulta de faltas no Portal do Aluno do Senac/RS**, que exige navegação manual e constante atenção a atualizações frequentes. Ao longo do desenvolvimento, percebemos o potencial de expansão dessa automação para outras funcionalidades relevantes para o acompanhamento acadêmico — como a **verificação da situação curricular**.

Com isso, surgiu a ideia de criar uma solução mais robusta: uma **interface central de comandos via Telnet**, que permite acessar, executar e visualizar os resultados das automações a partir de qualquer terminal na rede, **sem depender da execução manual via script**.

Essa central tem como objetivo **agilizar o acesso às informações acadêmicas**, reduzindo fricções no uso das automações e facilitando o controle pessoal dos dados diretamente por linha de comando.

🎯 Objetivos específicos da automação

- **Centralizar o uso das automações** em uma única interface via Telnet.
- **Facilitar o acesso remoto** às informações de faltas e situação curricular.
- **Permitir a execução sob demanda** dos scripts, sem necessidade de reabertura ou modificação.
- **Oferecer uma experiência interativa** por meio de menu simples e responsivo no terminal.
- **Exibir o horário atual** do sistema como utilitário adicional.

🔧 Arquitetura e funcionamento

O projeto é composto por três elementos principais:

1. **Script de Consulta de Faltas**
   Utiliza Python e Selenium WebDriver (em modo headless) para acessar o boletim do Portal do Aluno, identificar as faltas por disciplina, gerar alertas e salvar logs.
2. **Script de Situação Curricular**
   Também desenvolvido com Python e Selenium, extrai dados como disciplinas aprovadas, optativas cumpridas e carga horária realizada, gerando um relatório do progresso no curso.
3. **Servidor Telnet (servidor_telnet.py)**
   Implementado com a biblioteca `socket`, cria um servidor local na porta `1234`, permitindo conexão via protocolo Telnet. Ao se conectar, o usuário tem acesso a um **menu interativo**, com as seguintes opções:
   - Executar script de consulta de faltas

   - Executar script de situação curricular

   - Ler os logs de execução

   - Exibir o **horário atual do servidor**

   - Encerrar a conexão

   A comunicação com os scripts é feita via `subprocess.run`, garantindo que a execução ocorra em tempo real, e os logs resultantes são convertidos para ASCII e enviados ao cliente conectado.

🧩 Desafios enfrentados e soluções adotadas

- **Transmissão correta de caracteres acentuados** nos logs → padronizamos a conversão para ASCII com substituição, garantindo que as mensagens sejam legíveis no terminal Telnet.
- **Tratamento de erros em tempo real**, exibindo mensagens claras ao cliente caso algo falhe durante a execução.
- **Encerramento seguro da conexão**, garantindo que o servidor continue rodando sem travamentos mesmo após múltiplas conexões.
- **Deixar as saídas dos scripts organizadas e legíveis no terminal do cliente Telnet**. Como o conteúdo retornado pelos scripts contém múltiplas linhas, acentos

e caracteres especiais, foi necessário implementar ajustes na codificação e formatação para evitar quebras inesperadas, garantindo uma experiência de leitura clara e fluida no menu interativo da CLI.

🔮 Melhorias futuras

- **Proteção por senha ao iniciar a sessão Telnet**, garantindo mais segurança.
- **Logs unificados por data e tipo de operação**.
- **Envio automático dos logs via e-mail ou Telegram**.