

# Faltas Checker

## Relatório de Automação — Consulta de Faltas no Portal do Aluno

**Discentes: Carolina Soares Freitas e Júlia Pereira Hallal**



### **Tarefa escolhida e motivação para automatizá-la**

A tarefa escolhida foi a automação do processo de verificação de faltas no Portal do Aluno do Senac/RS. Esta atividade, quando realizada manualmente, exige vários cliques e carregamentos de página, além de uma atenção constante para verificar atualizações no boletim — especialmente em períodos de fechamento de notas e frequência.

A motivação principal para automatizar essa tarefa surgiu da necessidade de poupar tempo e esforço em um processo repetitivo, e também da dificuldade pessoal em acompanhar com frequência a quantidade de faltas acumuladas por disciplina, o que pode impactar diretamente na aprovação ou reprovação por frequência.



### **Objetivos específicos da automação**

- Poupar tempo com consultas manuais no portal do aluno.
- Melhorar o controle pessoal sobre a quantidade de faltas em cada disciplina.
- Gerar visibilidade rápida sobre o status de presença, por meio de logs, mensagens de alerta automáticas e capturas de tela.
- Salvar capturas de tela do boletim e das páginas acessadas, possibilitando a verificação visual do conteúdo acessado pelo browser controlado via Selenium, garantindo rastreabilidade e transparência no processo automatizado.



### **Desafios enfrentados e soluções adotadas**

Durante o desenvolvimento do script, um dos principais desafios foi lidar com a estrutura do portal do aluno, que é desenvolvido em PHP. Isso resultou em

dificuldades para identificar e interagir com certos elementos da página, especialmente na tabela do boletim.

Outro ponto complexo foi garantir que os elementos estivessem devidamente carregados antes de interagir com eles, já que o portal possui carregamentos dinâmicos que, se ignorados, quebram o fluxo do script. Para isso, foi necessário utilizar estratégias de espera explícita ( `WebDriverWait` ) com condições específicas de visibilidade e interatividade dos elementos.

Além disso, a seleção do semestre correto na tela do boletim envolvia clicar em uma seta e depois escolher manualmente o período desejado, o que exigiu atenção para os seletores corretos e aplicação de `scrollIntoView` para evitar falhas.

A solução geral adotada envolveu o uso da biblioteca Selenium WebDriver em modo headless, com tratamento de exceções e registros visuais (screenshots) em pontos estratégicos, ajudando no monitoramento e depuração do script.



## Possíveis melhorias e extensões futuras

- **Envio automático por e-mail ou WhatsApp:** Integrar o script com uma API de e-mail ou bot do WhatsApp para enviar as faltas e alertas automaticamente ao final da execução.
- **Agendamento automático:** Usar um agendador de tarefas para rodar o script periodicamente, sem necessidade de execução manual.
- **Extração de outras informações:** Ampliar o script para capturar médias, notas finais, situações de aprovação/reprovação ou dados financeiros.
- **Geração de relatórios PDF:** Ao invés de logs em `.txt` , permitir exportação em PDF com visual mais amigável.
- **Tratamento de exceções mais robusto:** Adicionar logs de erro mais detalhados e captura de traceback para facilitar debugging.