

Departamento de Computação

Disciplina	Curso	Turno	Período	Valor
Tópicos Especiais em L. P.	Engenharia da Computação	Manhã	10	40 Pontos
Aluno(a)	Henrique Coelho Mendes		Matrícula	20193017733
Ao entregar esse exame eu dou minha palavra que eu o fiz sozinho, entendendo que eu posso consultar qualquer material publicamente disponível, exceto aqueles disponibilizados por colegas que estão fazendo esse curso ou que já o fizeram no passado.				

## Prova

Instruções
<ul style="list-style-type: none"><li>A entrega deve ser feita exclusivamente via sistema acadêmico (SIGAA) em formato PDF. Este arquivo deve conter as respostas para os três itens: a), b) e c).</li><li>O programa do item b) também pode ser enviado em um arquivo próprio com extensão .sh.</li></ul>

**Questão)** Desenvolva um programa em Shell Script para resolver algum problema interessante. O script deve ter pelo menos 30 linhas de código (usando cloc: <https://github.com/AIDanial/cloc>).

a) Descreva qual problema está sendo resolvido com uma ou duas frases.

Para o meu tcc, foi preciso coletar métricas de qualidade de software orientados a objeto. Para tanto, a ferramenta “ck” foi aplicada em uma lista de repositórios do github a partir de um arquivo json.

b) Mostre o programa desenvolvido.

```
#!/bin/bash

json_file="C:\Users\h8men\OneDrive\Documentos\Cefet\TCC\result.json"
log_file="C:\Users\h8men\OneDrive\Documentos\Cefet\TCC\shellErrors.log"
metrics_repo="C:\Users\h8men\OneDrive\Documentos\Cefet\TCC\metrics\"

# Verifica se o arquivo JSON existe
if [ ! -f "$json_file" ]; then
    echo "Arquivo JSON não encontrado!" >> "$log_file"
    exit 1
fi

# Verifica se o diretório de métricas existe, caso contrário cria
if [ ! -d "$metrics_repo" ]; then
    echo "Diretório de métricas não encontrado, criando..." >> "$log_file"
    mkdir -p "$metrics_repo"
fi

# Itera sobre as URLs do arquivo JSON
for row in $(jq -r '.[ ] | .[2]' "$json_file"); do
    echo "URL encontrada: ${row}"

    github_url=$row
    repo_name=$(echo "$github_url" | awk -F '/' '{print $(NF-0)}' | cut
-d'.' -f1)

    echo "Clonando o repositório do GitHub $repo_name"
    git clone "$github_url" "$repo_name"

    # Verifica se o clone foi bem-sucedido
    if [ "$?" -ne 0 ]; then
        echo "Erro ao clonar o repositório do GitHub. ${row}" >>
```

```
"$log_file"
    continue
fi

    echo "Executando análise de métricas no repositório $repo_name"
    java -jar ck-0.7.1-SNAPSHOT-jar-with-dependencies.jar "$repo_name" true
0 false "${metrics_repo}${repo_name}"

    if [ "$?" -ne 0 ]; then
        echo "Erro ao executar o CK no repositório ${repo_name}" >>
"$log_file"
        rm -rf "$repo_name"
        continue
    fi

    echo "Análise concluída. Deletando o diretório $repo_name"
    rm -rf "$repo_name"
done

# Limpeza dos logs mais antigos
find "$log_file" -type f -mtime +30 -exec rm -f {} \;

echo "Processo finalizado com sucesso."
```

c)

#	Trecho de código	Conceito	Aula
01	<code>rm -rf "\$repo_name"</code>	Remoção de arquivos sem solicitar confirmação	Sistemas de arquivos
02	<code>if [ "\$?" -ne 0 ]; then</code>	Variáveis posicionais de interação com a shell via '\$?'	Parâmetros e variáveis
03	<code>repo_name=\$(echo "\$github_url"   awk -F'/' '{print \$(NF-0)}'   cut -d'.' -f1</code>	Expansão de comandos via \$()	Expansões
04	<code>json_file="C:\Users\h8men\OneDrive\Documents\Cefet\TCC\result.json"</code>	Atribuição de variáveis	Parâmetros e variáveis
05	<code>echo "Arquivo JSON não encontrado!" &gt;&gt; "\$log_file"</code>	Uso de >> para concatenar a saída ao arquivo	Entradas, Saídas e Redirecionamentos
06	<code>if [ "\$?" -ne 0 ]; then</code>	Controle de fluxo condicional	Estruturas de repetição
07	<code>for row in \$(jq -r '.[[]   .[2]' "\${json_file}"); do</code>	Uso do comando de repetição for	Estruturas de repetição
08	<code>repo_name=\$(echo "\$github_url"   awk -F'/' '{print \$(NF-0)}'   cut -d'.' -f1)</code>	Processamento de dados utilizando AWK, separando campos por '/'	AWK
09	<code>for row in \$(jq -r '.[[]   .[2]' "\${json_file}"); do</code>	Uso de   (pipes) para filtrar a saída de um comando	Entradas, saídas e redirecionamentos
10	<code>for row in \$(jq -r '.[[]   .[2]' "\${json_file}"); do</code>	Aplicação de agrupamentos com expressões regulares	Expressões regulares