



Imersão em GenAI para Desenvolvedores

Projeto 1: Conhecendo o ambiente

Preâmbulo: Nesse módulo, você irá interagir com o ambiente da 42 e conhecer o sistema de avaliação

Versão: 1.0

Sumário

I	Preâmbulo	2
II	Instruções gerais	4
III	Ex00 - Echo do Z	6
IV	Ex01 - Caminho para a automação	8
V	Entrega e Avaliação entre pares	10
V.1	Processo de Entrega	10
V.2	Avaliação entre pares	10
V.3	Dicas para uma avaliação bem-sucedida	10

Capítulo I

Preâmbulo

"A filosofia Unix não é uma metodologia formal de design. Ela não foi entregue do alto das montanhas da ciência da computação teórica como uma forma de produzir software teoricamente perfeito. Nem é o eterno espelhismo executivo, alguma maneira de extrair magicamente software inovador, mas confiável, de programadores desmotivados, mal gerenciados e mal pagos em prazos muito curtos."

— Eric S. Raymond¹

A essência do Unix, e por extensão, do desenvolvimento em shell, reside em um conjunto de princípios fundamentais que têm guiado gerações de programadores:

1. **Simplicidade e clareza:** Prefira soluções simples e claras.
2. **Modularidade:** Construa sistemas complexos a partir de partes simples.
3. **Composição:** Projete programas para trabalhar juntos.
4. **Separação de mecanismo e política:** Mantenha as ferramentas flexíveis.
5. **Economia e geração de código:** Valorize o tempo do programador.
6. **Portabilidade e abertura:** Escreva uma vez, execute em qualquer lugar.

Estes princípios se manifestam no shell através de:

- Scripts legíveis e compreensíveis à primeira vista.
- Funções focadas e reutilizáveis.
- Uso efetivo de pipes e redirecionamentos.

¹Adaptado de Raymond, E. S. (2003). The Art of Unix Programming. Addison-Wesley Professional.
<http://www.catb.org/esr/writings/taoup/html/ch01s06.html>

- Separação de configuração e lógica.
- Automação de tarefas repetitivas.
- Códigos que funcionam em diversos ambientes Unix.

"Quando em dúvida, use força bruta."

— Ken Thompson

"Escreva programas que façam uma coisa e a façam bem feito."

— Doug McIlroy

Capítulo II


Instruções gerais

- Esta página é sua única referência oficial. Não confie em informações não verificadas.
- Os exercícios estão organizados em ordem crescente de complexidade. É essencial dominar cada exercício antes de prosseguir para o próximo.
- Preste atenção às permissões de seus arquivos e pastas.
- Siga rigorosamente o procedimento de entrega para todos os exercícios.
- Seus exercícios serão avaliados por colegas da Imersão.
- Para exercícios em Shell, utilize `/bin/zsh`.
- Mantenha em sua pasta apenas os arquivos explicitamente solicitados nos enunciados.
- Em caso de dúvidas, consulte seus colegas à direita ou à esquerda.
- Utilize recursos como Google, manuais online e a Internet como referência.
- Leia os exemplos com atenção. Eles podem conter requisitos não explicitamente mencionados no enunciado.
- Para exercícios em Python:
 - Use a versão do Python especificada no exercício de configuração do ambiente.
 - Utilize os modelos e provedores sugeridos para garantir tempos de resposta adequados e consistência nos testes.
- Esteja atento a erros em todos os exercícios. Eles raramente são tolerados durante a avaliação.
- **Aviso sobre o uso de ferramentas de AI (como ChatGPT):**
 - O uso de ferramentas como o ChatGPT não deve ser encarado como um substituto para seu próprio esforço e entendimento.
 - O aprendizado efetivo ocorre quando você interage ativamente com o conteúdo: pesquisando, refletindo e aplicando o que aprendeu.
 - Nas avaliações, serão feitas perguntas para avaliar sua compreensão real sobre o assunto.

- E durante as avaliações, seus colegas também avaliarão seu nível de conhecimento.

Capítulo III

Ex00 - Echo do Z

	Exercício : 00
Criar um arquivo que exiba o caractere 'Z' quando lido pelo comando 'cat'.	
Pasta de entrega : <i>ex00/</i>	
Arquivos para entregar : <i>z</i>	

Como desenvolvedor, você inicia uma jornada para explorar o mundo da GenAI. Seu primeiro passo é criar um script simples que exibe a letra 'Z'. Esse momento marca o começo da sua exploração e serve como um teste da sua configuração básica.

Instruções

1. Crie um arquivo chamado 'z'.
2. O arquivo deve conter apenas o caractere "Z"seguido por uma nova linha.
3. Ao usar o comando 'cat' no arquivo, deve-se ver o seguinte resultado:

```
$ cat z
Z
$
```



O comando 'cat' é usado para exibir o conteúdo de arquivos.



Lembre-se da importância dos caracteres de nova linha em arquivos de texto.




Você pode criar este arquivo sem usar um editor de texto tradicional. Pesquise sobre redirecionamento de saída no shell.



A pesquisa é sua aliada. Não hesite em buscar informações adicionais sobre comandos do shell e manipulação de arquivos.

Capítulo IV

Ex01 - Caminho para a automação

	Exercício : 01
Criar um script que busque todos os arquivos com extensão ".sh" no diretório atual e subdiretórios, exibindo seus nomes sem a extensão	
Pasta de entrega : <i>ex01/</i>	
Arquivos para entregar : <i>find_sh.sh</i>	

Ao organizar suas ferramentas, você cria um utilitário para buscar arquivos .sh, já pensando em como essa habilidade será útil para automatizar outras tarefas. Este exercício destaca a importância da organização e eficiência no seu fluxo de trabalho de desenvolvimento.

Instruções

1. Crie um script shell chamado 'find_sh.sh'.
2. O script deve buscar todos os arquivos que terminam com ".sh" no diretório atual e em todos os subdiretórios.
3. Deve exibir apenas os nomes dos arquivos, sem a extensão ".sh".
4. A saída deve ser compatível com o comando 'cat -e', mostrando caracteres de fim de linha.

Exemplo de uso:

```
\$ ./find_sh.sh | cat -e
find\_sh\$
file1\$
file2\$
file3\$
\$
```



Arquivos `.sh` são scripts shell, comumente usados em sistemas Unix/Linux para automatizar tarefas.



Pesquise sobre comandos de busca de arquivos no sistema, como `'find'`.



Considere como você pode manipular strings para remover a extensão `".sh"` dos nomes dos arquivos.

Capítulo V

Entrega e Avaliação entre pares

V.1 Processo de Entrega

- Submeta seu trabalho no repositório Git gerado na página principal do projeto.
- Certifique-se de que todos os arquivos necessários estejam incluídos e organizados conforme as instruções do projeto.
- Respeite o prazo de entrega estabelecido.

V.2 Avaliação entre pares

- Seu projeto será avaliado por um dos seus colegas.
- A avaliação focará na qualidade do seu código e na aderência aos requisitos do projeto.
- Critérios de avaliação podem incluir:
 1. Funcionalidade: O código atende a todos os requisitos especificados?
 2. Legibilidade: O código é claro e bem estruturado?
 3. Eficiência: As soluções implementadas são otimizadas e seguem boas práticas?
 4. Organização: Os arquivos e estrutura do projeto estão bem organizados?
- Feedback detalhado é esperado, mas pode variar em extensão e detalhamento.

V.3 Dicas para uma avaliação bem-sucedida

- Revise seu código antes da submissão final.

- Teste exaustivamente todas as funcionalidades implementadas.
- Se entender necessário para clareza, documente claramente qualquer decisão ou suposição feita.
- Esteja preparado para explicar suas escolhas de implementação.



A avaliação entre pares é uma oportunidade para aprendizado e crescimento pessoal e profissional. Esteja aberto ao feedback recebido e use-o para aprimorar suas habilidades.