

Universidade Federal de São Carlos  
Centro de Ciências Exatas e de Tecnologia  
Departamento de Computação

Trabalho 4 - Compiladores - **Linguagem: Portfólios 3000**  
UFSCar - ENPE 2020.2 - Bloco A  
Prof. Daniel Lucrédio

Caio César de Moraes Sales - 760722

## Introdução

A linguagem Portfólios 3 foi pensada justamente para atender as necessidades de usuários que desejam criar seus próprios PORTFÓLIOS DE PROJETOS EM FORMATO HTML. Possibilitando que esses usuários possam hospedar suas páginas HTML, totalmente responsivas, em qualquer plataforma de hospedagem; sem a necessidade de escrever uma linha de código HTML/CSS se quer. Parece bom demais para ser verdade, mas com a linguagem Portfólios 3000, é totalmente possível!

## Descrição da Linguagem

```
1 informacoes Nome, Telefone, Email, Linkedin, Github
2
3 habilidades android, java, javascript, html
4
5 projetos _p1, _p2, _p3
6
7 monta_portfolio
8     Nome <- "Caio Cesar de Moraes Sales"
9     Telefone <- "3598459-2516"
10    Email <- "caio@meuemail.com"
11    Linkedin <- "https://www.linkedin.com/in/caiosales"
12    Github <- "https://github.com/caiosales35"
13
14    _p1("Site responsivo para divulgacao de eventos", html)
15    _p2("Aplicativo mobile TO-DO-LIST nativo", android, html)
16    _p3("Calculadora cientifica mobile", javascript, java)
17
18    {comentario}
19 fim_portfolio
20
```

Figura 01. Exemplo de algoritmo da linguagem Portfólios 3000

Inicialmente, como na linha 1 da figura 01, declaramos nossas informações pessoais. Iniciando com a palavra reservada “informacoes”, seguida de quais informações pessoais desejamos incluir. As informações pessoais possíveis são: Nome, Telefone, Email, Linkedin, Github e Site; devem ser separadas por vírgula e iniciadas com letra maiúscula.

Seguindo, temos a declaração das nossas habilidades, como na linha 3 da figura 01. Iniciam com a palavra reservada "habilidades", seguida de quais habilidades queremos incluir. As habilidades devem iniciar com letra minúscula e são separadas por vírgulas, é possível incluir qualquer cadeia de caracteres formada por letras e dígitos nas habilidades (respeitando a inicial minúscula).

Ainda no bloco de declarações, temos a declaração de projetos, como na linha 5 da figura 01. Iniciam com a palavra reservada “projetos”, seguida de um identificador para nossos projetos. Os identificadores de projetos podem ser qualquer cadeia de caracteres formada

por letras e dígitos, desde que sejam iniciados com “\_” (carácter de sublinhado); devem ser separados por vírgula.

O bloco de comandos deve ser envolvido pelas palavras reservadas “monta\_portfolio” e “fim\_portfolio”, respectivamente, como nas linhas 7 e 19 da figura 01. Os comandos possíveis são:

- Comando atribuição: utilizando o operador de atribuição “<-”, realiza uma atribuição de uma cadeia de caracteres (entre aspas duplas) à uma informação pessoal. Por exemplo, na linha 10 da figura 01; a informação pessoal “Email” recebe a cadeia de caracteres “caio@meuemail.com”.
- Comando projeto: com parênteses, vincula uma cadeia de caracteres, ou melhor, uma descrição do projeto (entre aspas duplas) e uma lista de habilidades a um identificador de projeto. Por exemplo, na linha 16 da figura 01; o identificador de projeto “\_p3”, está sendo vinculado à descrição “Calculadora científica mobile”, e às habilidades “javascript” e “java”. A descrição e a lista de habilidades devem ser separadas por vírgula (pode-se vincular quantas habilidades forem desejadas).

Convém lembrar que é possível incluir comentários no código, como na linha 18 da figura 01. Os comentários são cadeias de caracteres (letras) e números envolvidos por chaves, “{” (abre comentário) e “}” (fecha comentário).

## Considerações Iniciais

Antes de prosseguir, certifique-se de ter o [Java](#) devidamente instalado e configurado no seu computador. Para verificar a instalação, pode-se rodar o seguinte comando: “java -version”, sem as aspas, em um terminal; a saída deve ser algo parecido com:

```
java version "1.8.0_121"  
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
```

Se tudo correr bem, pode-se prosseguir.

## Instruções para Rodar com Exemplos de Casos de Testes

1. Após baixar os arquivos, a partir da raiz do projeto, entre no diretório “trabalho04\target” e localize o arquivo “trabalho04-1.0-SNAPSHOT-jar-with-dependencies”.
2. Acesse, a partir da raiz do projeto, a pasta “casos-de-testes”. Nela, é possível encontrar oito arquivos prontos de casos de testes; arquivos com a terminação “-erro”, são casos de testes que apresentam algum tipo de erro (é possível abrir o caso de teste em algum editor de arquivo e visualizar as linhas onde são esperadas mensagens de erro).

3. Agora, abra um terminal e execute: java -jar <ARG1 - diretório do arquivo localizado no passo 1> <ARG 2 - diretório do caso de teste desejado, de acordo com o passo 2> <ARG 3 - diretório completo para salvar a saída>

**Exemplo de comando completo:** java -jar

C:\Users\caio\Documents\trabalho04\target\trabalho04-1.0-SNAPSHOT-jar-with-dependencies.jar C:\Users\caio\Documents\casos-de-teste\1-algoritmo.txt C:\temp

**\*Obs:** certifique-se de separar os diretórios (<ARG 1>, <ARG 2> e <ARG 3>), por um espaço em branco, na hora de montar o comando executado no passo 3. Da mesma forma, não é necessário incluir uma barra no último diretório do comando (diretório para salvar a saída, <ARG 3>).

4. Após rodar o comando no passo 3, o terminal exibirá “Fim da geração”, em caso de sucesso. Caso algum tipo de erro seja detectado no arquivo de entrada, ele será listado no terminal.
5. Para ver a saída HTML gerada (de acordo com o caso de entrada), acesse o diretório passado no ARG 3, no comando montado no passo 3. Pode-se abrir esse arquivo com qualquer navegador HTML (recomenda-se acesso à internet para uma melhor experiência).

## Instruções para Compilar

Para realizar a compilação do projeto, é necessário ter instalado e configurado o [NetBeans](#), juntamente com o plugin do [Maven](#).

1. Para iniciar, abra o NetBeans, clique em “Arquivo, “Abrir projeto”, e selecione a pasta “trabalho04”, a partir da raiz do projeto baixado.
2. Com o projeto aberto no NetBeans, clique na opção “Executar”, e depois em “Limpar e Construir Projeto”; ou se preferir, pode-se utilizar o atalho do teclado, as teclas Shift+F11; ou ainda, clicar no ícone de “martelo e vassoura”.
3. Se tudo correr bem, ou seja, se nenhuma mensagem de erro for gerada, significa que o projeto foi compilado com sucesso.