

Disciplinas: Programação Orientada a Objetos, Projeto e Arquitetura de Software, Fundamentos de Programação e Verificação e Validação de Software

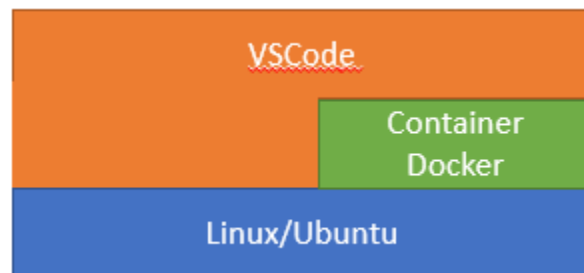
Parte 2 – Usando os containers

Usando Containers Docker com o VSCode

Quando se trabalha usando um container a ideia é que se use um ambiente virtual rodando sobre o sistema nativo da máquina que estamos usando. Instalamos todos os softwares necessários neste ambiente e desenvolvemos o trabalho nele. A vantagem é que podemos “voltar” para o ambiente nativo da máquina hospedeira, entrar em outros containers etc.

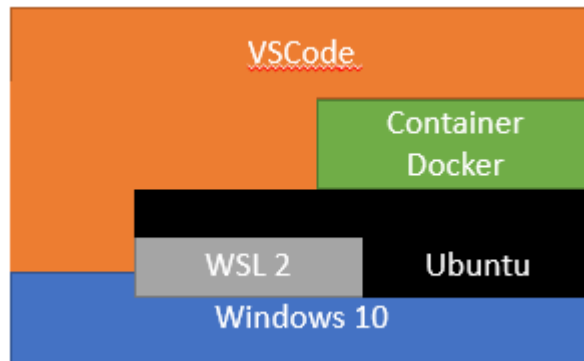
Uma das desvantagens do uso de containers Docker é que eles não emulam o ambiente de janelas do sistema operacional, trabalham apenas em modo comando de linha. Isso permite que sejam mais leves e que atendam os propósitos originais dos containers, isto é, hospedar múltiplos serviços simples para a realização de uma ou mais tarefas complexas. Para trabalhar com desenvolvimento, entretanto, isso se torna um problema porque precisamos do ambiente visual da IDE para trabalhar.

As extensões “remote” do VSCode, indicadas na parte 1 deste roteiro, tem exatamente este papel. Elas servem para que o VSCode execute o ambiente de edição visual na máquina hospedeira, mas utilize o ambiente do container (onde instalamos a versão correta de Java e todos os demais softwares necessários) durante a compilação e execução dos projetos. O ambiente do console também é executado no contexto do container. A figura que segue demonstra estes conceitos.



Resumindo: os arquivos fonte (código dos programas) são armazenados no sistema de arquivos da máquina hospedeira. Quando o VSCode é acionado ele carrega estes arquivos e executa o sistema de edição (IDE visual) na máquina hospedeira. Na hora de compilar e executar, porém, ele usa o ambiente virtual do container.

No caso de máquinas Windows a diferença é que temos a distribuição de Ubuntu executando sobre o WSL como na figura que segue.



Neste caso precisamos sempre nos conectar no ambiente Ubuntu para poder localizar nossos arquivos antes de poder abrir os projetos. Este passo é desnecessário no ambiente Linux porque, neste caso o VSCode já está executando diretamente sobre o Linux (sistema hospedeiro). No caso do uso de Mac o funcionamento é igual ao do Linux.

Como o VSCode sabe que deve usar um container e qual container usar?

Para que o VSCode saiba que determinado projeto deve ser compilado/executado em um determinado container, uma pasta chamada “.devcontainer” (o container do desenvolvedor) deve ser colocada na raiz do projeto ou do workspace. No momento em que o VSCode abre o projeto ele percebe a presença desta pasta e pergunta se deve abrir o projeto naquele contexto. Responda que sim de maneira que o projeto possa ser adequadamente compilado e executado.

Esta fora do escopo deste documento entender como os arquivos “.devcontainer” são criados. O professor irá fornecer o arquivo correto para a sua disciplina

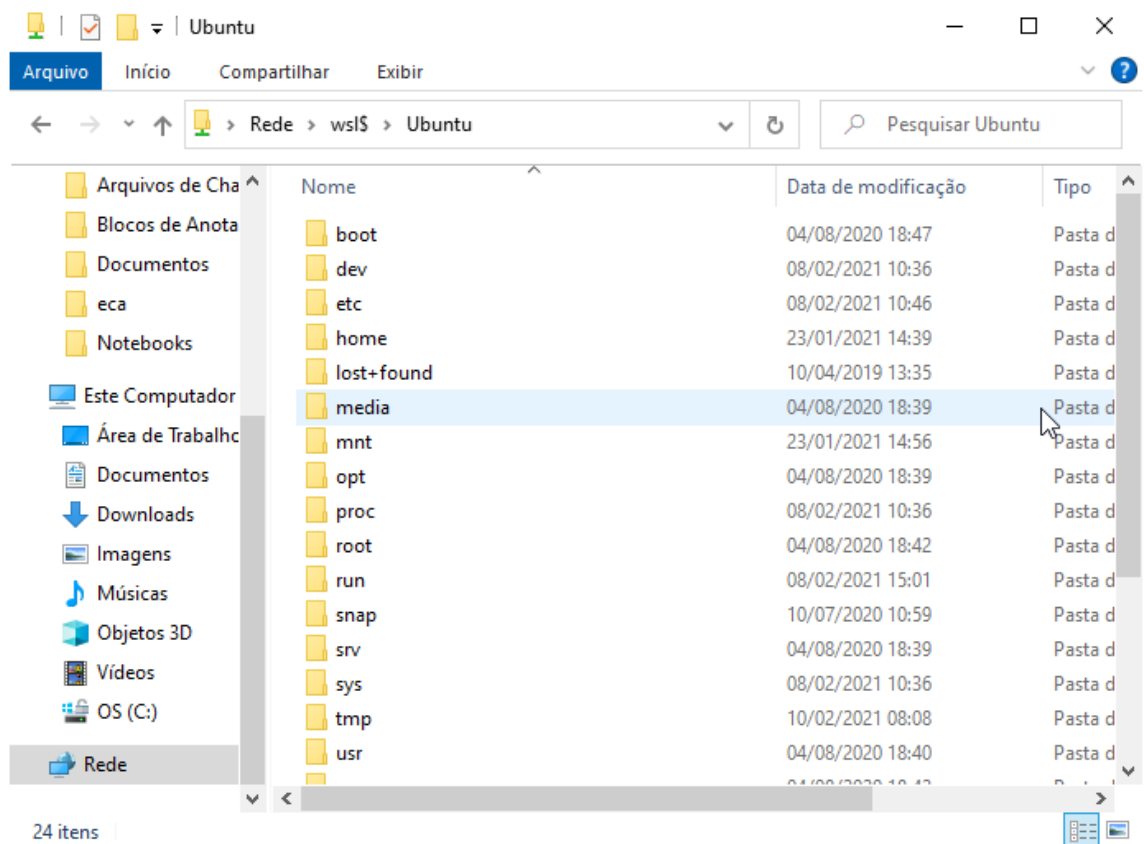
Como acessar o Ubuntu a partir de uma máquina com Windows

Se você instalou adequadamente o WSL2 e o cliente Ubuntu como indicado anteriormente, existem duas formas de ter acesso ao sistema de arquivos do Ubuntu:

1. A partir do console do Ubuntu: abra o aplicativo “Terminal Ubuntu” instalado anteriormente. Ele irá abrir uma console na raiz do sistema de arquivos Linux. A partir daí você tem acesso a todos os arquivos a partir do terminal (console).

```
bcpstein@DESKTOP-90MME79: ~  
bcpstein@DESKTOP-90MME79:~$  
bcpstein@DESKTOP-90MME79:~$ ls -las  
total 44  
4 drwxr-xr-x 7 bcpstein bcpstein 4096 Feb 10 08:05 .  
4 drwxr-xr-x 3 root      root      4096 Jan 23 14:39 ..  
0 lrwxrwxrwx 1 bcpstein bcpstein   23 Jan 23 14:56 .aws -> /mnt/c/Users/berna/.aws  
0 lrwxrwxrwx 1 bcpstein bcpstein   25 Jan 23 14:56 .azure -> /mnt/c/Users/berna/.azure  
4 -rw----- 1 bcpstein bcpstein 1132 Feb 10 08:12 .bash_history  
4 -rw-r--r-- 1 bcpstein bcpstein 220 Jan 23 14:39 .bash_logout  
4 -rw-r--r-- 1 bcpstein bcpstein 3771 Jan 23 14:39 .bashrc  
4 drwxr-xr-x 4 bcpstein bcpstein 4096 Jan 23 15:29 .docker  
4 drwxr-xr-x 2 bcpstein bcpstein 4096 Jan 23 14:39 .landscape  
0 -rw-r--r-- 1 bcpstein bcpstein   0 Feb 10 08:02 .motd_shown  
4 -rw-r--r-- 1 bcpstein bcpstein  807 Jan 23 14:39 .profile  
0 -rw-r--r-- 1 bcpstein bcpstein   0 Feb  8 10:39 .sudo_as_admin_successful  
4 drwxr-xr-x 5 bcpstein bcpstein 4096 Jan 23 15:22 .vscode-server  
4 drwxr-xr-x 6 bcpstein bcpstein 4096 Feb  9 09:17 Projetos  
4 drwxr-xr-x 2 root      root      4096 Jan 26 11:35 ctrlcorridasbasico  
bcpstein@DESKTOP-90MME79:~$
```

2. A partir do “Explorador de Arquivos do Windows”: para ter acesso aos arquivos armazenados na “partição Linux” através do software “Explorador de Arquivos” do Windows basta mapear esta partição. O caminho para ela é: [\\wsl\\$\\Ubuntu](#).



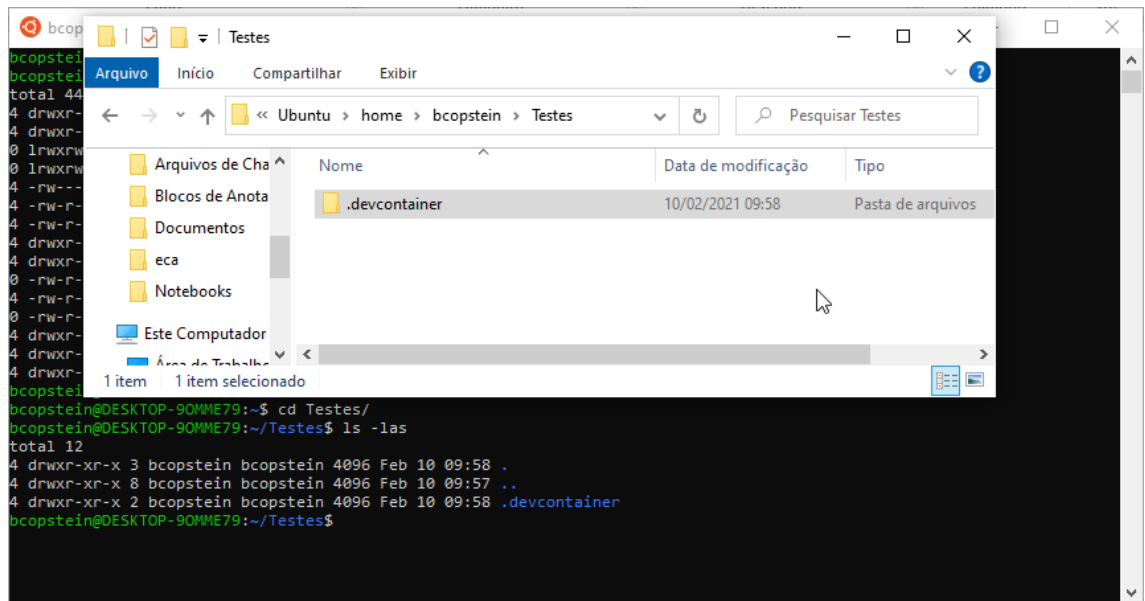
Usando os Containers Docker para criar projetos Java

Quando se trabalha com projetos grandes, o correto é incluir o arquivo “.devcontainer” na raiz do projeto. Isso fará com que o Docker crie uma imagem associada aquele projeto e um container correspondente.

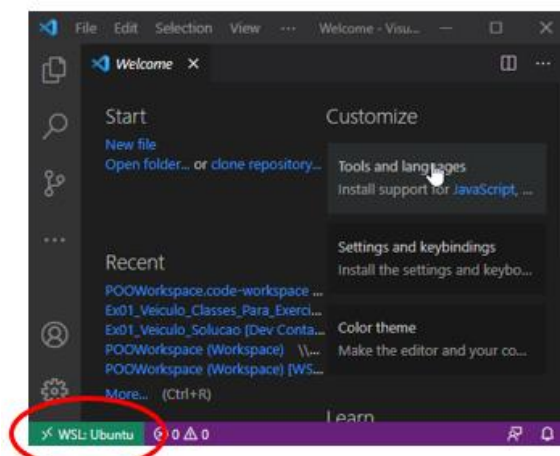
No contexto de uma disciplina, porém, é comum criarmos vários projetos. Neste caso serão criados uma imagem e um container para cada projeto tornando as coisas mais difíceis de serem gerenciadas. Para contornar esse problema iremos trabalhar com o conceito de “Workspace” do VSCode. Um “Workspace” é simplesmente um agrupamento de diversos projetos. Tipicamente o conjunto de projetos com os quais você está trabalhando. Então iremos associar o “.devcontainer” com o “Workspace” de maneira que teremos uma única imagem para todos os projetos da disciplina.

Os passos a seguir descrevem como criar o primeiro projeto, como criar um workspace, como associar o workspace com um container e como compilar e executar seus programas. A primeira vez o processo é trabalhoso, mas permitirá que você disponha de um ambiente igual ao do professor para executar os projetos da disciplina. A criação dos projetos seguintes é bem simples.

- 1) **Criando a pasta que irá abrigar os projetos:** crie a pasta que irá abrigar seus projetos e copie a pasta “.devcontainer” fornecida pelo professor para dentro dela. Veja a figura que segue.



- 2) **Conecte o VSCode no Ubuntu (passo exclusivo para Windows):** para conectar o VSCode no Ubuntu clique no ícone verde no canto inferior esquerdo da tela. No menu que irá abrir selecione “New Window with distro” e então escolha Ubuntu. Após a seleção a tela deverá ficar como na figura.



O VSCode abre uma nova janela ao se conectar no Ubuntu, logo a janela anterior pode ser fechada. Usuários de Linux podem pular este passo porque o VSCode já está rodando naturalmente sobre a distribuição Linux.

- 3) **Criando o primeiro projeto:** para criar o primeiro projeto use a opção “View | Command Pallet”. Na janela de diálogo que irá se abrir digite “Java: create java Project” (note que à medida que você digita as opções começam a surgir). Na sequência escolha o tipo de projeto. Se for um projeto Java simples escolha “No build tools”. Finalmente selecione a pasta onde deseja criar o projeto (indique a mesma onde está o arquivo “.devcontainer”) e, finalmente, indique o nome do projeto. O VSCode irá criar uma pasta com o nome do projeto no mesmo nível da pasta “.devcontainer”. Note que por enquanto o VSCode não está associando o projeto com o container descrito no arquivo “.devcontainer”.
- 4) **Criando um “Workspace” para abrigar seus projetos:** para criar um workspace use a opção “File | Add folder to workspace” e selecione o projeto recém criado. Esta operação irá adicionar o projeto ao workspace padrão.

- 5) **Atribuindo um nome para o “Workspace” padrão e disparando o container:** para atribuir um nome para o Workspace use a opção “File|Save Workspace As”. ATENÇÃO: cuidado para armazenar a descrição do “Workspace” na mesma pasta que está armazenada a pasta “.devcontainer”. Na hora de renomear o nome do arquivo que guarda a descrição do Workspace, cuide para renomear apenas a parte do nome do arquivo que o VSCode indica. Por exemplo, se o nome do seu “workspace” for “Poo”, o nome do arquivo descritor do “Workspace” deverá ser “Poo.code-workspace”. Ao concluir esta operação o VSCode irá notar a presença da pasta “.devcontainer” ao lado do arquivo descritor do “Workspace” e irá perguntar se deve reabrir o projeto no contexto do container (este diálogo costuma aparecer no canto inferior direito da tela). Responda que sim. Na primeira vez esta operação pode demorar um pouco porque o container será criado. Quando tudo estiver concluído na área em verde no canto inferior esquerdo da tela deverá aparecer o nome do container (exs: “Dev Container POO: Java”, “Dev Container PAS:JavaNG” etc). Pronto: agora o VSCode está conectado ao container e irá executar e rodar seu projeto de dentro deste contexto.
- 6) **Executando um programa:** Ao criar um projeto Java com a opção “No build tools” o VSCode sempre cria um programa do tipo “Hello World” no arquivo “App.java”. Este estará localizado na pasta “src” (source – de código fonte). Abra este arquivo e após alguns momentos a opção “run|debug” irá aparecer em cima da função “main”. Uma alternativa é usar o menu “Run” do próprio VSCode. Altere e execute o programa quantas vezes quiser.
- 7) **Verificando se o ambiente está correto:** Para verificar se você está executando o VSCode sobre o container correto pode usar os seguintes comandos na janela do terminal:
- a. `javac -version` → deve responder 15.0.2
 - b. `java -version` → deve responder 15.0.2
 - c. `mvn -version` → deve responder Apache Maven 3.6.3 além de uma série de outras informações sobre o ambiente.
- 8) **Encerrar o ambiente de trabalho:** Para encerrar uma sessão de trabalho você pode:
- a. Simplesmente fechar o VSCode. Neste caso quando você o abrir novamente será restaurada a mesma sessão de trabalho dentro do container.
 - b. Fechar o projeto usando a sequência de teclas: “Ctrl+k+f”. Neste caso o VSCode fecha o projeto, mas permanece no mesmo container. Como este container agora está “vazio” você não conseguirá achar o seu “Workspace” nele para reabrir a sessão de trabalho. Neste caso existem duas opções: na lista de trabalhos recentes estiver aparecendo o nome do Workspace com a indicação “[Dev Container]” do lado, basta clicar nela. Caso contrário clique na área verde e então selecione a opção “Close remote connection”. Após esta operação o VSCode voltará a executar sobre o sistema operacional hospedeiro. Usuários do Windows irão necessitar reconectar com o Ubuntu para localizar seus projetos.
 - c. Reabrindo o “Workspace”. Se o VSCode está executando no Linux ou conectado no Ubuntu do Windows, você pode retornar para seus projetos simplesmente reabrindo o “Workspace”. Ao reabrir o “Workspace” ele irá identificar a pasta “.devcontainer” e restaurar o container que contém as configurações da disciplina. Não abra os projetos diretamente. Se fizer isso o VSCode não irá “enxergar” a pasta “.devcontainer” e não irá associar o projeto ao mesmo.

- 9) **Criando projetos novos:** para criar projetos novos basta abrir o “Workspace”, criar o projeto no mesmo nível dos demais e, em seguida, adicionar o novo projeto ao “Workspace”. Desta forma todos os projetos ficarão associados ao mesmo container.

Exercício:

Crie um projeto que consiste em um programa capaz de exibir na tela a versão de Java na qual ele está executando. O código da função “main” deste programa segue:

```
public static void main(String[] args) throws Exception {  
    System.out.println("\nHi!");  
    String version = System.getProperty("java.version");  
    System.out.println("Running Java Version "+version+"\n");  
}
```

Seja curioso

Este texto permite que você execute o ambiente necessário para a disciplina. Porém ele envolve vários conceitos que foram explicados superficialmente. Pesquise mais a fundo sobre todos os temas abordados aqui e compartilhe suas descobertas com seus colegas através do fórum da disciplina. Suas contribuições através do fórum serão consideradas na avaliação da disciplina.