Configuração Api de Treinamento de Automação de Testes Backend

Introdução

Este documento tem por objetivo guiar o leitor no processo de configuração e instalação das ferramentas que serão usadas durante o treinamento de automação de testes *backend*, bem como a inicialização do sistema que iremos automatizar.

Docker

Docker é uma tecnologia que permite aos desenvolvedores empacotar, entregar e executar aplicações em containers leves e autossuficientes.

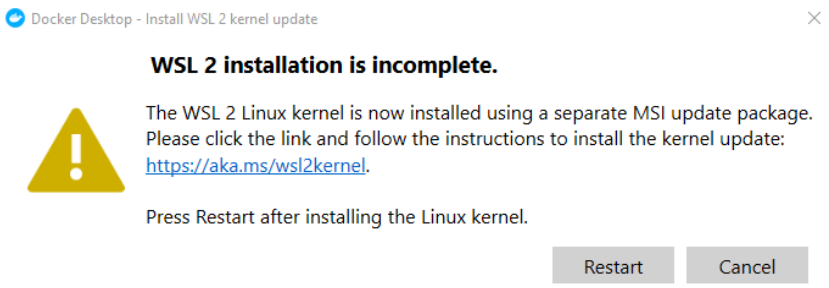
Instalação

Faça o download do docker para o sistema operacional de sua preferência e prossiga com a instalação, em caso de dúvidas consulte o guia de instalação.

Ao final da instalação se solicitado reinicie o sistema operacional.

Atualização de *kernel*

Durante o primeiro acesso ao docker pode ser necessário a atualização do WSL 2 *Kernel*.



WSL 2 *Kernel Update* - Imagem

Para resolver este problema será necessário fazer o download e instalação do pacote de atualização do *kernel*.

Depois de instalar o pacote de atualização, abra o *PowerShell* ou o *prompt de comando* e execute o comando abaixo:

```
wsl --set-default-version 2
```

Reinicie o docker.

** O passo a passo completo para atualização do *kernel* pode ser acessado através do link.

Aplicações

Para o treinamento de automação de testes, usaremos uma *Web Api* desenvolvida em .Net5 com o banco de dados *Postgres* conteinerizados em um ambiente docker.

Iniciando as aplicações

Baixe o fonte da aplicação disponibilizada durante o treinamento em uma pasta de sua preferência.

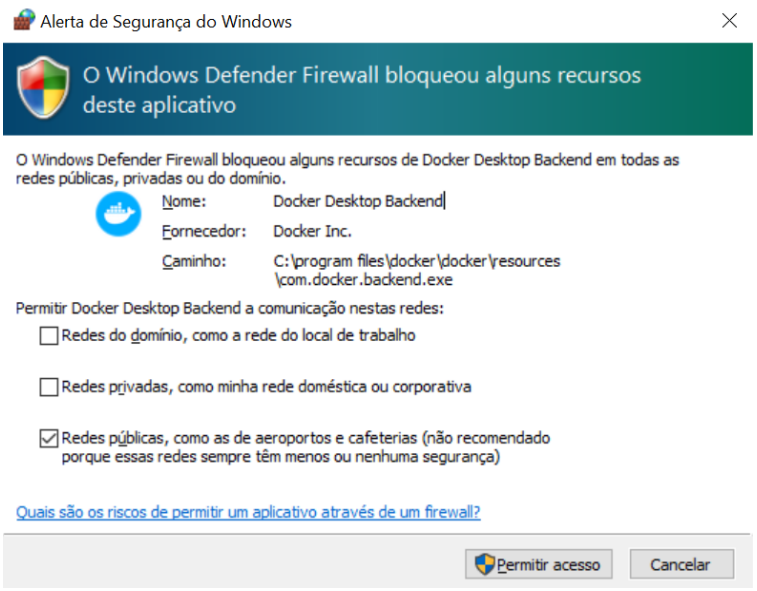
Abra o *PowerShell* ou *prompt de comando* na pasta ***/ci*** e execute o comando abaixo para gerar as imagens necessárias das aplicações que serão utilizadas:

```
docker-compose build
```

Após a conclusão do *build*, execute o comando abaixo para iniciar os *containers*.

```
docker-compose up -d
```

Se solicitado, conceda ao *docker* acesso a rede.



Alerta de segurança do Windows - Imagem

Se todos os passos forem executados com sucesso, o sistema usado para o treinamento estará disponível através da url http://localhost:8080/swagger.

Encerrando as aplicações

Abra o *PowerShell* ou *prompt de comando* na pasta */***ci** e execute o comando abaixo para encerrar os *containers* das aplicações:

```
docker-compose down
```

O comando acima irá parar os *containers* usados no treinamento, mas manterá as *images* e os *volumes* das aplicações para reuso no futuro, caso queira remover por completo os recursos usados pelas aplicações, execute o comando abaixo:

```
docker-compose down -v --rmi all
```

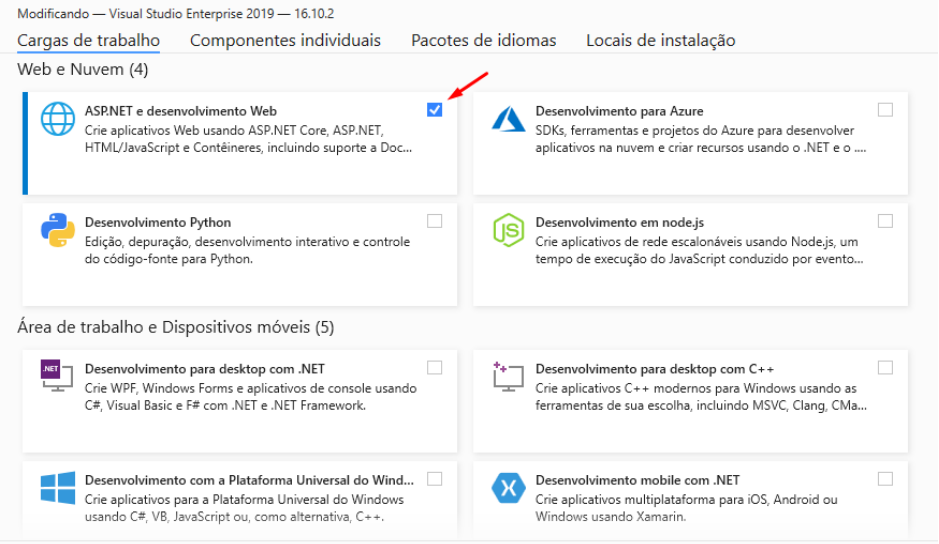
IDE (Ambiente de Desenvolvimento Integrado)

No guia abaixo você encontrará instruções para instalação e configuração de duas opções de IDE, **você não precisa instalar as duas**, basta escolher a de sua preferência e seguir com o treinamento.

Obs.:
Para a gravação do treinamento foi utilizado o Visual Studio 2019 Enterprise (requer licença).

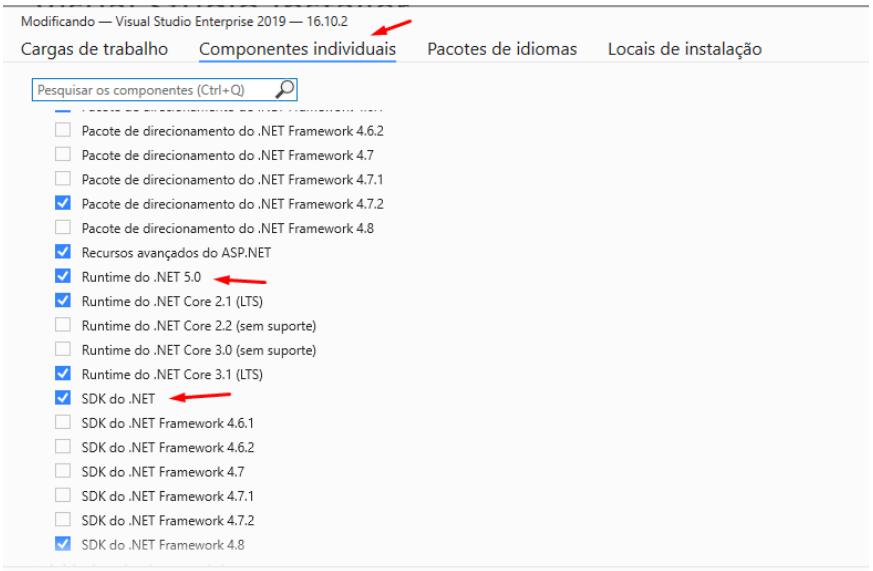
Visual Studio

Baixe e instale o Visual Studio 2019 ou superior, durante a instalação selecione o pacote para desenvolvimento de aplicações “Asp.Net e Desenvolvimento Web”, conforme imagem abaixo.



Instalação Visual Studio - Imagem

Na aba “Componentes Individuais”, verifique se as opções SDK do .NET e Runtime do .NET 5.0 estão selecionadas, caso não estejam selecione-as.



Instalação Visual Studio - Imagem 2

O idioma padrão da IDE utilizado durante o treinamento é o inglês, para evitar dificuldades em encontrar menus por causa da diferença entre linguagens, sugerimos que você selecione e instale o idioma “Inglês” na aba “Pacote de idiomas”.

Em caso de dúvida em alguma etapa do processo de instalação, consulte o guia completo de instalação do Visual Studio.

Após concluir a instalação, faça o download e instale a extensão do *SpecFlow* que facilitará a criação e escrita dos cenários de teste.

Visual Studio Code

Baixe e instale o dotnet 5.0 SDK, necessário para o desenvolvimento e execução do projeto de teste.

Baixe e instale o Visual Studio Code.

Extensões úteis para o desenvolvimento do projeto de teste:

- C# extension
- Cucumber Language Support

Configurando o acesso ao banco de dados

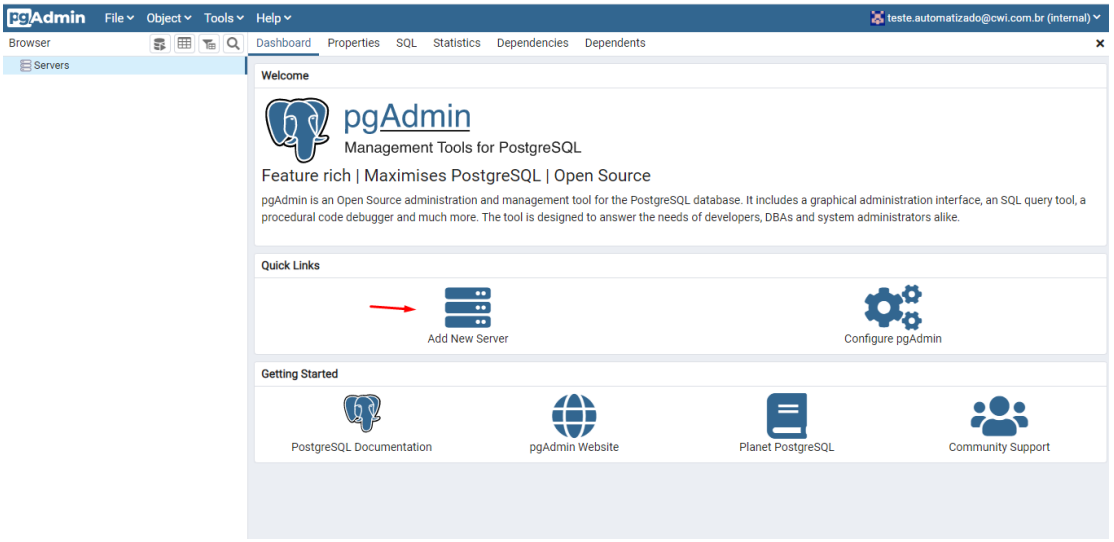
A aplicação criada para o treinamento faz uso de uma instância do banco de dados *Postgres*. Para tornar mais fácil o acesso a base usaremos a aplicação *PgAdmin* já disponível na estrutura de *containers* criada nos passos anteriores e que pode ser acessada através da url http://localhost:16543/login.

Credenciais para acesso ao *PgAdmin*:

- Usuário: teste.automatizado@cwi.com.br
- Senha: treinamento123

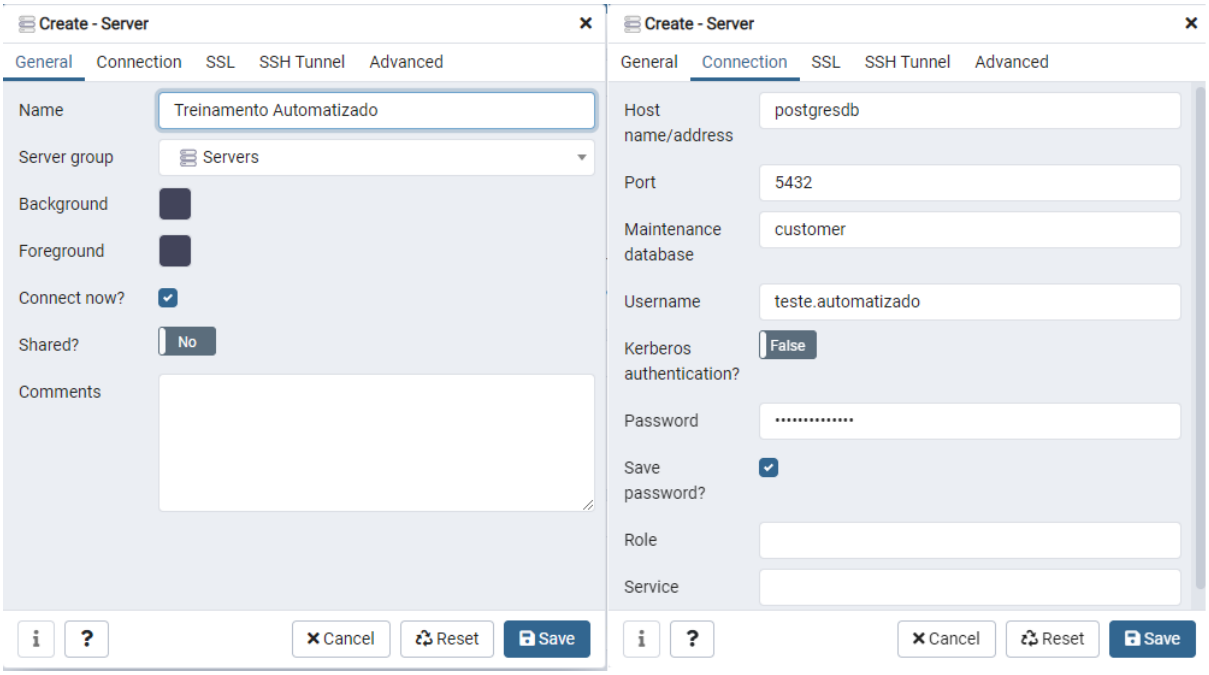
Após logar na aplicação, caso o servidor não apareça na lista de *servers*, você pode adicionar uma nova conexão seguindo os passos abaixo.

1. Clique no botão "*Add Server*"



Tela Inicial *PgAdmin* - Imagem

2. Na aba "*General*" preencha o campo "*Name*" e avance a aba "*Connection*"
3. Na aba "*Connection*" preencha os campos a seguir
   3.1. *Host name/address*: **postgresdb**
   3.2. *Maintenance database*: **employee**
   3.3. *Username*: **teste.automatizado**
   3.4. *Password*: **treinamento123**



Exemplo Inclusão do servidor - Imagem

4. Clique no botão “*Save*”

Depois de configurado o servidor aparecerá na lista de “*Servers”* e estará pronto para uso.

Links de apoio

Docker
PowerShell
Prompt de comando
Postgres
PgAdmin
Visual Studio
Visual Studio Code
SpecFlow