



Tutorial de instalação da Plataforma For no Windows

Desenvolvido por: Vitor N. Muniz

ID	Software	Versão	Formato	Comentários
01	MySQL	5.7	EXE	-
02	WildFly	9.0.2	ZIP	Necessária deszipagem para a pasta C:\ do computador
03	NodeJS	10.15.3	EXE	-
04	Projeto	-	ZIP	-
05	Apache	2.4.39	ZIP	Necessária deszipagem para a pasta C:\ do computador
06	JDK	1.8	ZIP	Necessária deszipagem e execução do arquivo <code>jdk-8u181-windows-x64.exe</code> para a instalação do software
07	Maven	3.6.1	ZIP	Necessária deszipagem para a pasta C:\ do computador

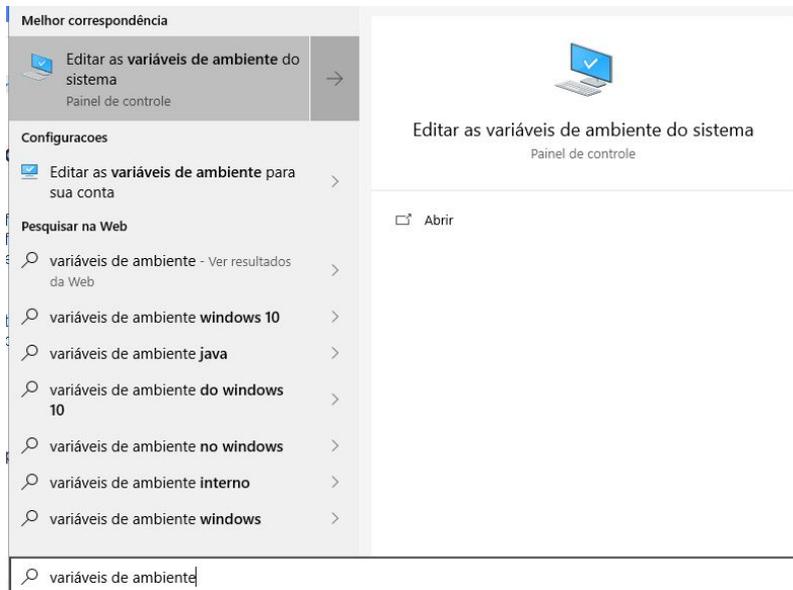
Tabela 1: Softwares e seus respectivos links para download/versões utilizados neste tutorial.

OBS: As versões dos softwares foram baseadas em um sistema operacional com arquitetura x64.

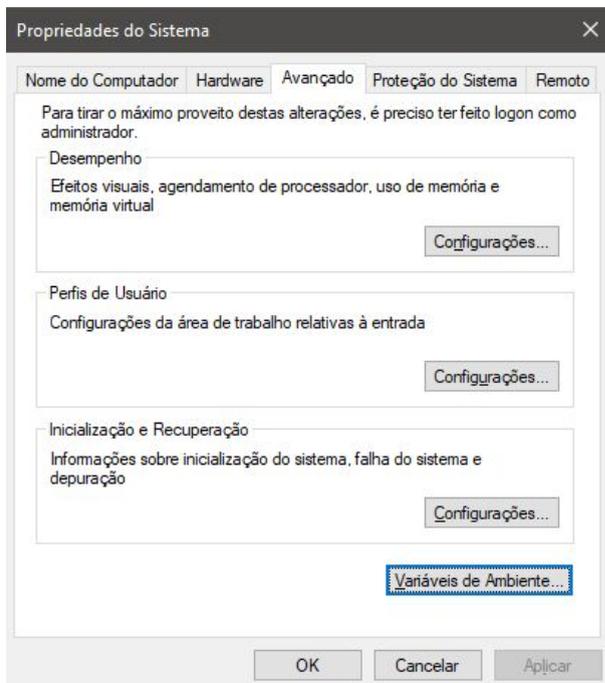
Construindo o frontend

O front-end do projeto foi feito utilizando a ferramenta Webpack para building e o framework React.js utilizando a arquitetura Flux. Para realizar o build do código do frontend instale o NodeJS presente na Tabela 1 e adicione os executáveis à sua variável de ambiente `PATH`:

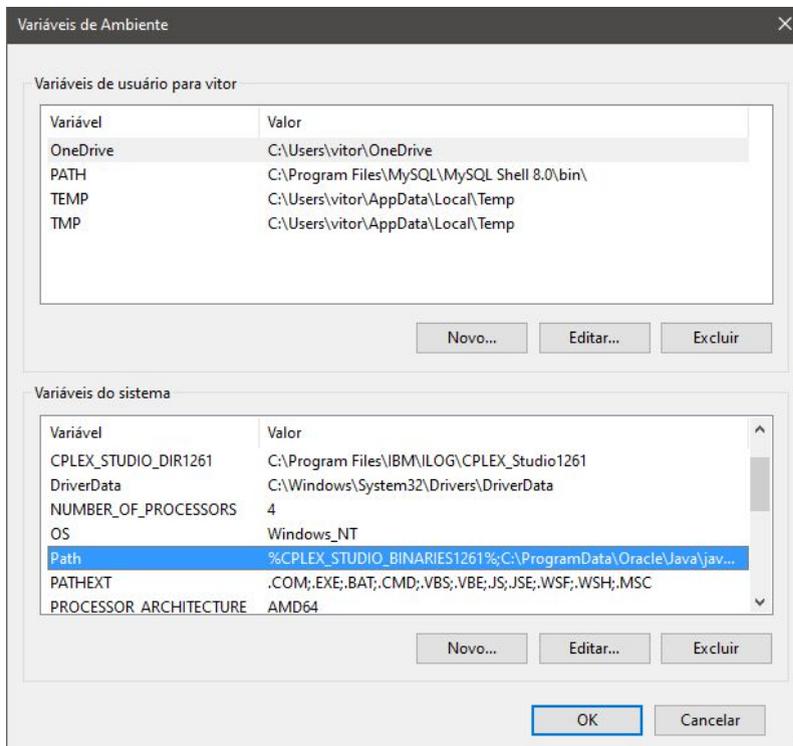
- Pesquise por variáveis de ambiente:



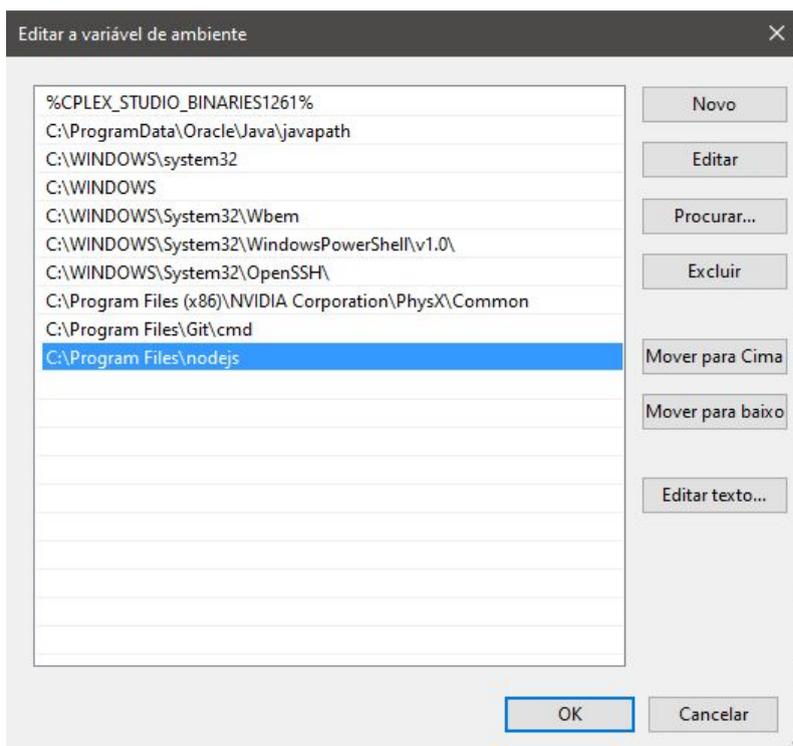
- Clique em variáveis de ambiente no canto inferior da janela:



- Na janela seguinte, em Variáveis do sistema procure por 'PATH', selecione e clique em Editar...:



- Feito isso, clique em Novo e adicione o caminho da pasta a qual você salvou o NodeJS e clique em OK:



Em seguida abra o cmd e navegue até a pasta `..\plataforma-for\frontend-web\` do projeto também presente na Tabela 1. Essa pasta contém todo o código do frontend. Execute o seguinte comando para instalar as bibliotecas do Node.js necessárias:

```
npm install
```

Caso apareça a seguinte mensagem:

```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules\fsevents):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})  
  
added 1103 packages from 629 contributors and audited 12991 packages in 50.395s  
found 502 vulnerabilities (7 moderate, 495 high)  
run `npm audit fix` to fix them, or `npm audit` for details
```

Digite o seguinte comando para corrigir as vulnerabilidades encontradas:

```
npm audit fix
```

Feito, você pode realizar o build para versão de produção. Basta executar o comando abaixo:

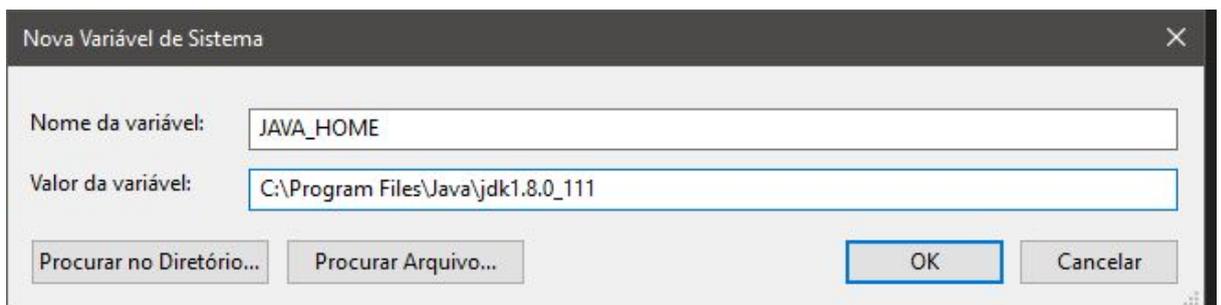
```
npm run build
```

Após executar este comando seu frontend pronto para publicação estará disponível na pasta `..\plataforma-for\frontend-web\dist.`

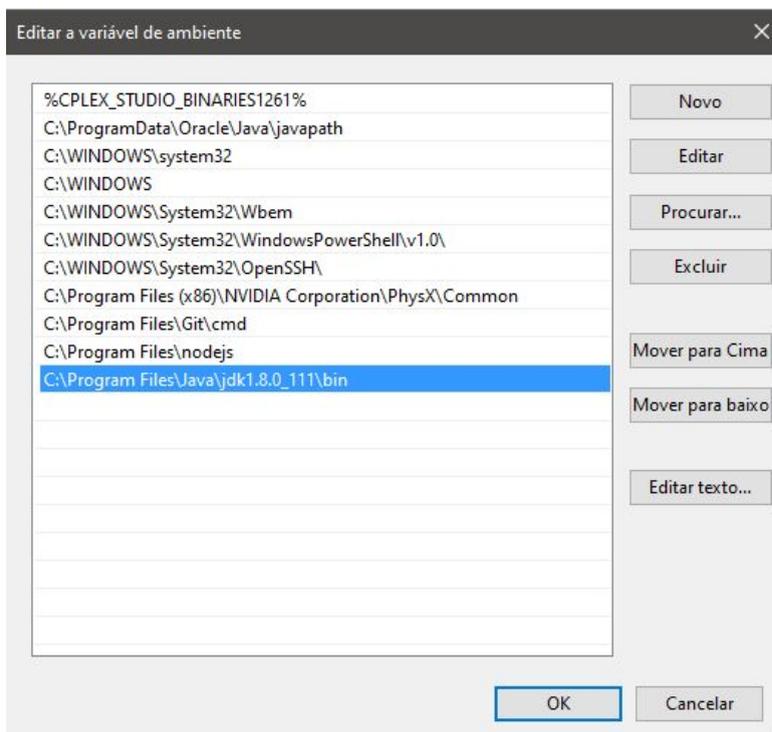
Construindo o .war do backend

O ForPDI utiliza o Apache Maven para realizar o processo de packaging do backend da aplicação em um arquivo `.war` que pode ser implantado em um servidor de aplicação que suporta Java. Para esta etapa, considera-se que você já instalou o JDK 1.8 da Tabela 1 e configurou corretamente a variável de ambiente `JAVA_HOME`:

- Copie o caminho da pasta a qual você salvou o JDK 1.8. Em seguida, em variáveis de ambiente, clique em Novo e adicione o caminho copiado:



- Em seguida, adicione o caminho da pasta bin presente dentro da pasta do JDK 1.8 à variável 'PATH', como feito anteriormente para o NodeJS:



A próxima etapa é realizar o download do Maven. Após o download do zip da Tabela 1, descompacte o arquivo em uma pasta. Neste passo a passo será considerado que o Maven foi descompactado na pasta `C:\`. Em seguida, configure a variável de ambiente `MAVEN_HOME` da mesma maneira feita para o `JAVA_HOME`. Também coloque na sua variável `PATH` o caminho da pasta `bin` do Maven (ex: `C:\apache-maven-3.6.1\bin`). Caso você possua Netbeans e o mesmo já possua o maven instalado, você provavelmente encontrará a pasta em `..\Netbeans\java\maven`.

Também é necessário fechar e abrir novamente o cmd para as alterações fazerem efeito.

Com os procedimentos de instalação prontos, você já pode realizar o build e packaging da aplicação. O arquivo `..\plataforma-for\backend-java\pom.xml` descreve todas as configurações do Maven para o projeto. O ForPDI está configurado com alguns *profiles* iniciais, o profile de desenvolvimento (perfil padrão) já vem pronto para uso, com as configurações no arquivo `..\plataforma-for\backend-java\dev.properties`:

```
# dev.properties
backendUrl=http://localhost:8080/forpdi/

db.host=localhost
db.port=3306
db.name=forpdi_db
db.username=root
db.password=

mail.smtp.from.name=ForPDI
mail.smtp.from.email=noreply@forpdi.org
mail.smtp.url=localhost
mail.smtp.port=25
mail.smtp.username=
mail.smtp.password=
mail.smtp.ssl=false
mail.smtp.tls=false
```

Note que o perfil de desenvolvimento vem configurado com SMTP local (localhost na porta 25) e o usuário do banco de dados é o root sem senha. Caso seu ambiente de desenvolvimento tenha outras configurações você pode trocá-las nesse arquivo. Os outros perfis disponíveis são o de teste (`test`), espelho (`mirror`) e produção (`prd`). Os arquivos de propriedades desses ambientes não são sincronizados pelo Git, já que eles contêm informações de senhas e usuários. Para realizar o build você terá que criar um arquivo com o nome do ambiente cujo qual você quer realizar build (`test.properties`, `mirror.properties` ou `prd.properties`). Em seguida, edite o arquivo e insira os dados de conexão do banco de dados e SMTP de seu ambiente. Por exemplo, para gerar um arquivo WAR para *produção* navegue até a pasta `..\plataforma-for\backend-java` e digite o seguinte comando:

```
notepad prd.properties
```

Na janela seguinte, copie e cole o conteúdo abaixo modificando os campos necessários, que se trata de um exemplo de arquivo para produção:

```
# prd.properties
backendUrl=http://localhost:8080/forpdi/

db.host=localhost
db.port=3306
db.name=forpdi_prd
db.username=SeuUsuarioDoBancoDeDados
db.password=SuaSenhaDoBancoDeDados

mail.smtp.from.name=ForPDI
mail.smtp.from.email=noreply@forpdi.org
mail.smtp.url=smtp.gmail.com
mail.smtp.port=587
mail.smtp.username=seuemail@gmail.com
mail.smtp.password=SuaSenhaDoEmail
mail.smtp.ssl=false
mail.smtp.tls=true
```

Ainda na pasta backend-java, após configurar o arquivo podemos realizar o packaging do maven para o perfil desejado:

```
mvn clean package -P prd
```

Após o packaging, o arquivo WAR estará disponível no caminho `..\plataforma-for\backend-java\target\forpdi.war`. Esse arquivo é o backend da sua aplicação, pronto para publicação no Wildfly.

Runtime

Considerando que o WildFly 9.0.2 já tenha sido baixado e descompactado na pasta `c:\` do seu computador, para publicar o backend é necessário a instalação do MySQL e a criação do banco de dados que irá conter as tabelas do ForPDI. Esse banco de dados deve possuir o nome definido no arquivo `prd.properties` na hora do build (propriedade `db.name`). A codificação de caracteres deve ser o *UTF-8*.

No link presente na tabela, instale somente o MySQL Server 5.7 dentre as opções presentes no instalador.

Feito isso, para criar o banco de dados navegue pelo cmd até a pasta bin presente em `C:\MySQL\MySQL Server 5.7\bin` e digite os seguintes comandos:

```
mysql -u root -p # ao digitar esse comando, será necessário inserir a senha de root que você configurou na instalação do MySQL Server 5.7  
CREATE DATABASE forpdi_prd CHARSET=utf8;
```

O processo de build do arquivo WAR é coberto nas seções anteriores. Isso inclui dados de conexão com o banco de dados que devem ser passados para o sistema no momento do build. Nessa etapa a única coisa que precisa ser feita é a publicação do arquivo WAR no Wildfly. Existem várias maneiras de fazer isso e o Wildfly possui várias configurações de execução, caso você queira otimizar essas configurações consulte a documentação do Wildfly. A maneira mais simples de executar o sistema é rodar o Wildfly em modo standalone e colocar o seu arquivo WAR na pasta de deployments.

Navegue até a pasta `..\plataforma-for\backend-java\target` e execute o seguinte comando:

```
copy forpdi.war C:\wildfly-9.0.2.Final\standalone\deployments\
```

Antes de inicializar o Wildfly, é necessário que habilitemos o conector AJP para que o Apache HTTPD possa posteriormente atuar como proxy reverso utilizando este protocolo. Edite o arquivo `standalone.xml` para incluir este conector navegando até `C:\wildfly-9.0.2.Final\standalone\configuration` e digitando o comando abaixo:

```
notepad standalone.xml
```

Na janela seguinte procure pela tag

`<subsystem xmlns="urn:jboss:domain:undertow:2.0">` e insira a linha indicada:

```
...
<subsystem xmlns="urn:jboss:domain:undertow:2.0">
  <buffer-cache name="default"/>
  <server name="default-server">
    <ajp-listener name="default-ajp" socket-binding="ajp" redirect-socket="http"/> <!-- Inclua esta tag
    <http-listener name="default" socket-binding="http" redirect-socket="https"/>
    <host name="default-host" alias="localhost">
      <filter-ref name="server-header"/>
      <filter-ref name="x-powered-by-header"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
    <websockets/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="{jboss.home.dir}/welcome-content"/>
  </handlers>
  <filters>
    <response-header name="server-header" header-name="Server" header-value="WildFly/9"/>
    <response-header name="x-powered-by-header" header-name="X-Powered-By"
header-value="Undertow/1"/>
  </filters>
</subsystem>
...
```

Feito isso, basta iniciar o WildFly. Navegue até `C:\wildfly-9.0.2.Final\bin` e digite o seguinte comando:

```
standalone.bat &
```

Por padrão o backend do sistema estará disponível em: `http://ip-do-seu-servidor:8080/forpdi/`. Após a publicação do backend, é necessário configurar um servidor web para servir os arquivos de frontend e para atuar como um proxy reverso para as chamadas ao backend. Você pode usar o Apache HTTPD ou o NGINX. Neste tutorial iremos utilizar o Apache HTTPD, porém se você preferir o NGINX é só realizar a configuração equivalente neste servidor web.

Você precisa instalar o Apache HTTPD (usaremos a versão 2.4, porém é possível realizar a mesma configuração na versão 2.2, basta consultar a documentação do Apache HTTPD). Instale também os módulos (se já não vierem junto com o pacote da sua distribuição) `mod_rewrite`, `mod_proxy` e `mod_proxy_ajp`. Além de instalar, é preciso ativá-los nos arquivos de configuração do Apache HTTPD:

- Abra outra janela do cmd, navegue até a pasta `C:\Apache24\conf` e digite o seguinte comando:

```
notepad httpd.conf
```

- Na janela que abrir, pressione Ctrl+F e procure pelas linhas que contêm `mod_rewrite`, `mod_proxy` e `mod_proxy_ajp` e retire a '#' do início de cada uma delas para ativar os módulos:

```
#LoadModule reqtimeout_module modules/mod_reqtimeout.so  
LoadModule rewrite_module modules/mod_rewrite.so  
#LoadModule sed_module modules/mod_sed.so
```

```
LoadModule negotiation_module modules/mod_negotiation.so  
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```

```
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so  
#LoadModule proxy_balancer module modules/mod_proxy_balancer.so
```

- Feito isso, adicione `'Include conf/forpdi.conf'` na última linha do arquivo para fazer a ligação do arquivo `forpdi.conf` que será criado nos próximos passos com o `httpd.conf`.

Agora é preciso criar o arquivo `forpdi.conf` para inserir as configurações do ForPDI/ForRisco. Para isso, você precisará definir alguns parâmetros que utilizaremos na configuração do sistema:

- `FRONTEND_DIR` -> Deve ser substituído pelo caminho de uma pasta do servidor onde está a versão gerada pelo build da aplicação (disponível na pasta `..plataforma-for\frontend-web\dist`), ex: `C:/Apache24\htdocs/forpdi`
- `FORPDI_DOMAIN` -> Domínio que será utilizado para acessar o ForPDI/ForRisco, ex: `www.forpdi.org` ou `localhost:80` se a instalação for local

Com essas definições, podemos criar o arquivo na pasta apropriada. Em `C:\Apache24\conf` e digite o seguinte comando:

```
notepad forpdi.conf
```

Na janela seguinte, copie e cole o trecho abaixo:

```
# Arquivo forpdi.conf
<VirtualHost *:80>
    ServerName FORPDI_DOMAIN
    UseCanonicalName Off
    ServerAdmin contato@forpdi.org
    DocumentRoot FRONTEND_DIR

    ErrorLog logs/forpdi-error.log
    CustomLog logs/forpdi-access.log combined

    <Location /forpdi>
        ProxyPreserveHost on
        ProxyPass ajp://127.0.0.1:8009/forpdi
    </Location>
</VirtualHost>
<Directory FRONTEND_DIR>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
    <IfModule mod_rewrite.c>
        RewriteEngine On
        RewriteBase /
        RewriteRule ^index\.html$ - [L]
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteCond %{REQUEST_FILENAME} !-l
        RewriteRule . /index.html [L]
```

```
</IfModule>
</Directory>

# As configurações abaixo são opcionais mas são recomendadas para performance em produção
KeepAlive on
KeepAliveTimeout 15
MaxKeepAliveRequests 0
Header append Vary User-Agent
AddOutputFilterByType DEFLATE text/html text/css application/json application/javascript
text/javascript
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
DeflateCompressionLevel 4 # 1 a 9
DeflateMemLevel 9 # 1 a 9
DeflateWindowSize 15 # 1 a 15
# As configurações abaixo servem para habilitar acesso via HTTPS:
<VirtualHost *:443>
    ServerName FORPDI_DOMAIN
    UseCanonicalName Off
    ServerAdmin contato@forpdi.org
    DocumentRoot FRONTEND_DIR

    SSLEngine on
    # Caminho dos arquivos de certificado digital fictícios, deve trocar pelo
    # caminho dos arquivos em seu servidor
    SSLCertificateFile "/etc/httpd/ssl/forpdi/cert.crt"
    SSLCertificateKeyFile "/etc/httpd/ssl/forpdi/private.key"
    SSLCACertificateFile "/etc/httpd/ssl/forpdi/ca-bundle.crt"
    SSLCompression Off
    SSLHonorCipherOrder On
    SSLCipherSuite
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-
AES256-SHA384:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA
-AES256-SHA256:DHE-RSA-AES256-SHA:!LOW:!MD5:!aNULL:!eNULL:!3DES:!EXP:!PS
K:!SRP:!DSS

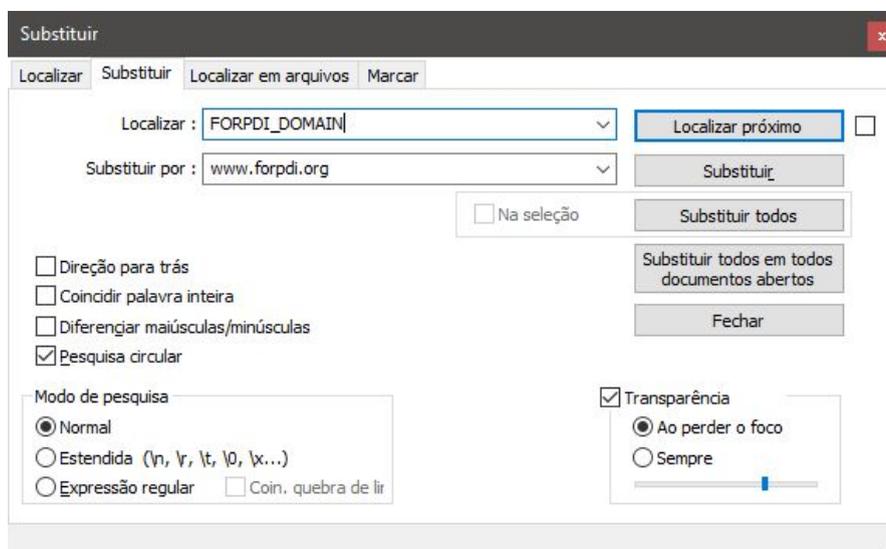
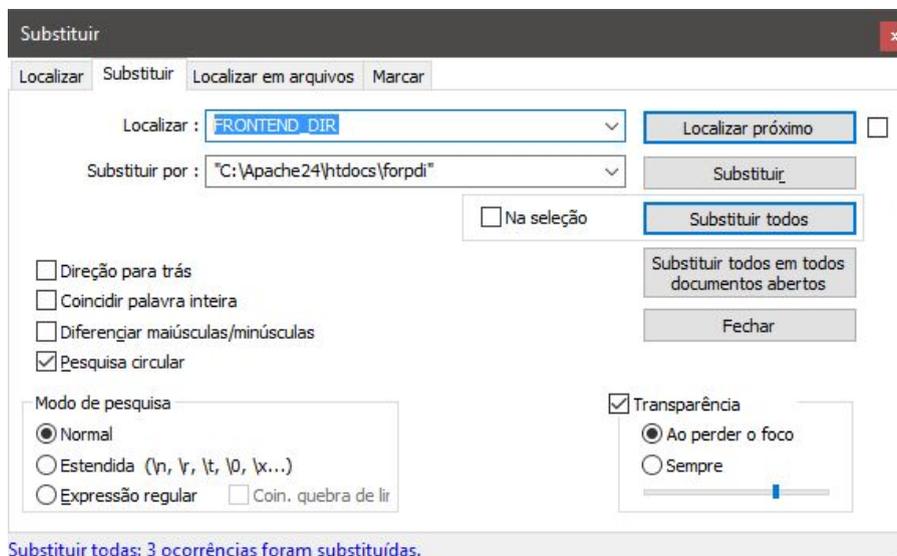
    ErrorLog logs/forpdi-ssl-error.log
    CustomLog logs/forpdi-ssl-access.log combined

    <Location /forpdi>
        ProxyPreserveHost on
        ProxyPass ajp://0.0.0.0:8009/forpdi
        Order allow,deny
        Allow from all
    </Location>
</VirtualHost>
# Para permitir apenas acesso SSL, pode trocar o virtual host da porta 80 para:
```

```
<VirtualHost *:80>
  ServerName FORPDI_DOMAIN
  UseCanonicalName Off
  RewriteEngine On
  RewriteCond %{HTTPS} !=on
  RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
</VirtualHost>
```

Para realizar as substituições descritas para o FORPDI_DOMAIN e FORPDI_DIR, dentro do arquivo o qual você colou o trecho acima pressione Ctrl + H e substitua de acordo com o solicitado, clicando em 'Substituir todos' após preencher os campos.

- Exemplo utilizando o Notepad++:



Após realizar a configuração do arquivo para o HTTPD, basta copiar o conteúdo da pasta `..\plataforma-for\frontend-web\dist` após o build do frontend para a pasta `FRONTEND_DIR` colocada na configuração:

- Navegue até a pasta `..\plataforma-for\frontend-web\` e digite o seguinte comando:

```
robocopy dist\ C:\Apache24\htdocs\forpdi /e
```

Em seguida basta iniciar o HTTPD e a aplicação estará disponível em `http://FORPDI_DOMAIN/`: navegue até a pasta `C:\Apache24\bin` e reinicie o servidor pela aplicação 'ApacheMonitor'.

O primeiro acesso deve ser feito utilizando o usuário administrador de sistema:

E-mail: `admin@forpdi.org`

Senha: 12345