

Prof. Evandro César Freiberg

Tutorial VSCode para Java

Brasil

16 de fevereiro de 2023

Sumário

1	Preparação Básica do Ambiente	2
1.1	Download e Instalação	2
1.2	Instalação de Extensões	2
1.3	Instalação e Configuração do Maven	4
2	Criando um Projeto Maven Básico	5
2.1	Criando a Estrutura de Pacotes	10
2.2	Codificando os Elementos de Código	11

1 Preparação Básica do Ambiente

Nesta seção serão apresentadas as ações para tornar a IDE VSCode apta ao desenvolvimento de software com a linguagem Java. O ambiente será configurado em etapas, conforme as necessidades de cada tipo de projeto Java. Assim, iniciaremos com projetos Java Básico utilizando a ferramenta Maven para a gestão e construção do projeto.

1.1 Download e Instalação

Neste tutorial foi utilizado a versão portátil, que não requer instalação. O download pode ser feito por meio da URL: <https://code.visualstudio.com/download> (.zip ou .tar.gz). Nessa opção basta realizar o download e executar o arquivo Code.exe (no caso de Windows).

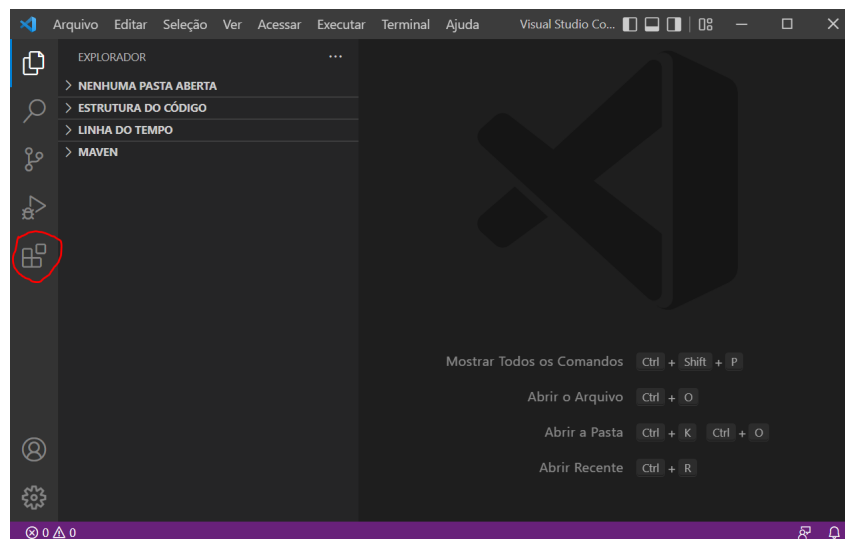
1.2 Instalação de Extensões

Nesta seção serão abordadas algumas extensões que são úteis para a IDE VSCode. Algumas são de uso geral e outras específicas para a linguagem Java.

Portuguese (Brazil) Language Pack - Pacote de Idioma Português Brasileiro para VS Code.

Extension Pack for Java - Reune um conjunto de extensões que habilitam o VS Code para a linguagem Java, depuração e Maven.

Figura 1 – Botão de Extensões do VS Code



Instalação de uma extensão para habilitar o idioma Português para o VS Code, é opcional.

Instalação de uma extensão Java. Essa extensão é importante pois dá suporte à linguagem, ao depurador de erros e suporte a construção e gerenciamento de projetos com o Maven.

Figura 2 – Instalando a Extensão Idioma Português

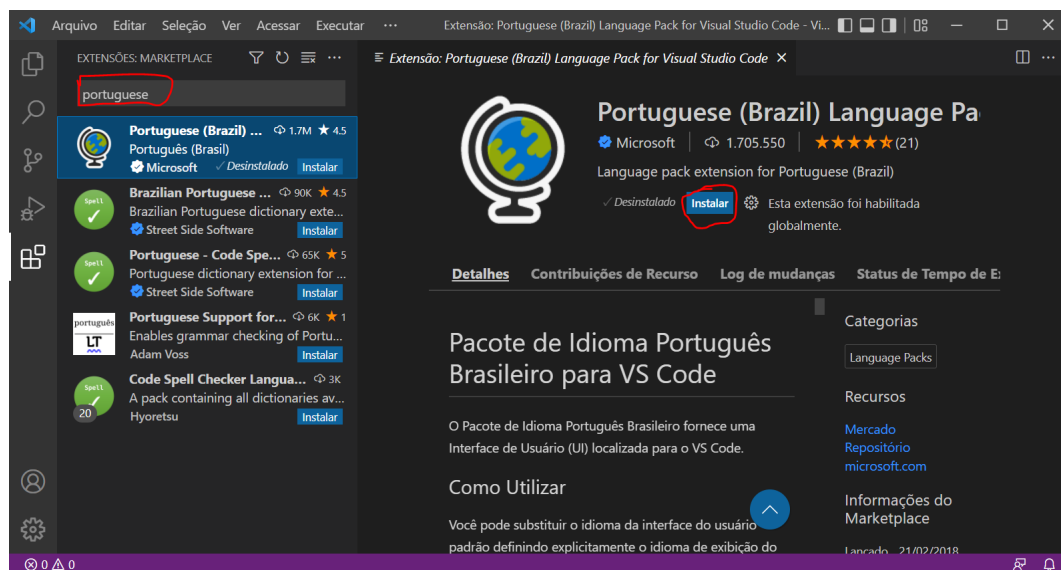
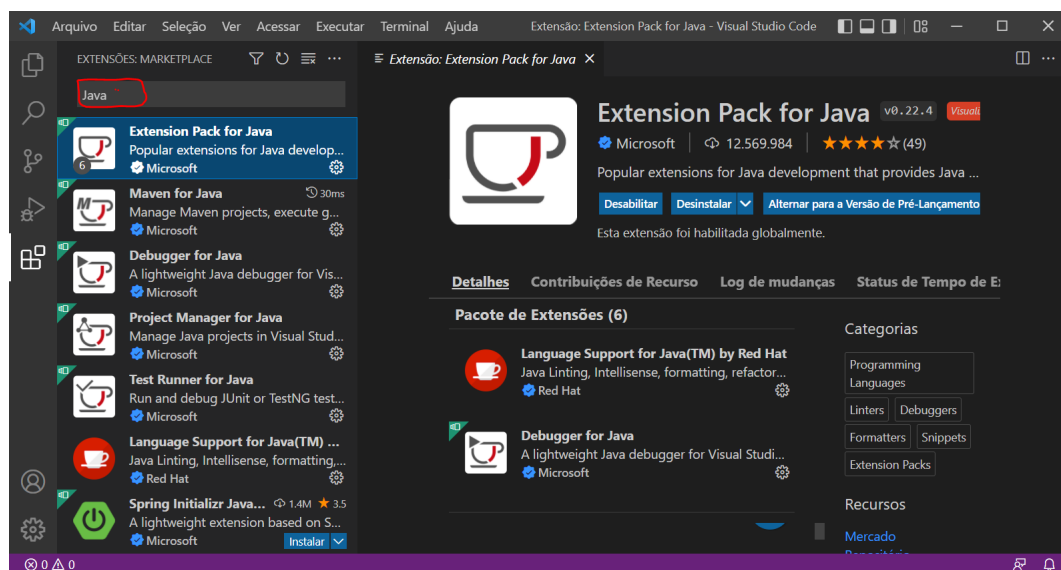


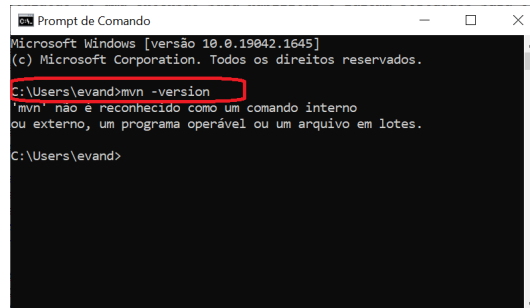
Figura 3 – Instalando a Extensão Java



1.3 Instalação e Configuração do Maven

Para podermos usar o Maven como gestor dos nossos projetos, precisamos ter ele instalado na nossa máquina. Você pode verificar se o Maven está instalado ou não na sua máquina, executando o comando `mvn -version` no terminal.

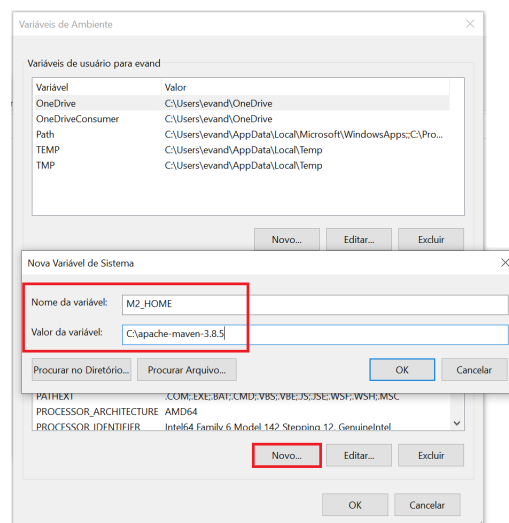
Figura 4 – Teste de Instalação do Maven



Caso não tenha o Maven instalado, faça o download pela URL <https://maven.apache.org/download.cgi>. Neste tutorial foi feito o download do arquivo `apache-maven-3.8.5-bin.zip`. Descompacte o arquivo e coloque em um local de sua preferência.

Depois de descompactar e definir o local de armazenamento do Maven na máquina, é necessário configurar a variável de ambiente **M2_HOME** e alterar a variável **Path** do ambiente. Veja nas imagens a seguir.

Figura 5 – Criando a variável de ambiente M2_HOME



O caminho para a variável **M2_HOME** é a pasta que descompactou o Maven.

O caminho para o novo item da variável **Path** é a subpasta `bin`, da pasta que descompactada do Maven.

Depois de configurar as variáveis de ambiente, abra um novo terminal e teste novamente na linha de comando o Maven. Agora deve aparecer os dados como na figura a seguir.

Obs: neste tutorial foram configuradas variáveis para o Sistema, ou seja, vale para todos os usuários da máquina, porém é necessário ter a senha de administrador. Essas variáveis podem ser configuradas para um usuário específico, não exigindo a senha do administrador.

Figura 6 – Editando a variável de ambiente Path

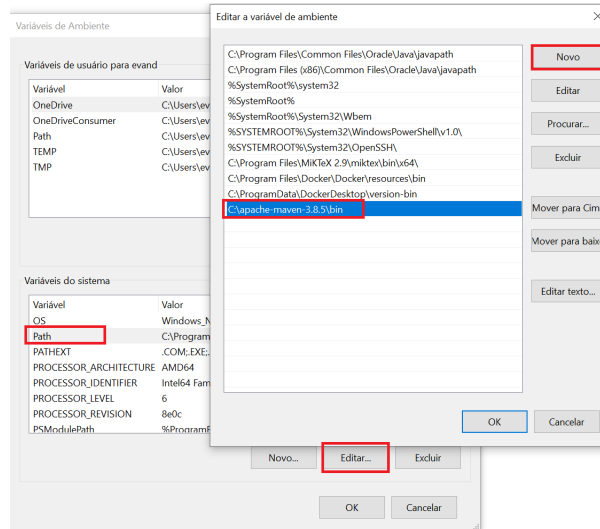
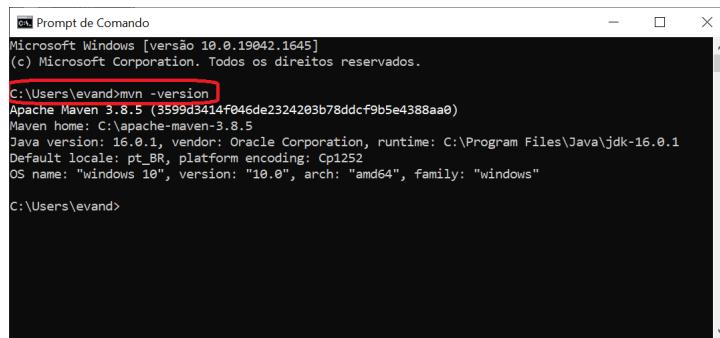


Figura 7 – Teste de Instalação do Maven



2 Criando um Projeto Maven Básico

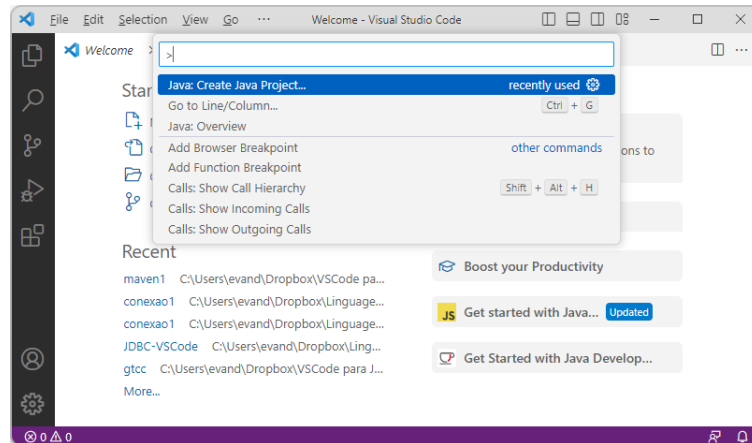
O Maven é uma ferramenta usada para gerenciar e construir as distribuições de um projeto Java. Além disso o Maven controla as dependências do projeto, quer seja com outros projetos, bibliotecas ou frameworks. Existe um repositório central onde são armazenadas as principais bibliotecas e frameworks Java disponíveis, além dos modelos de projetos Maven disponíveis. Os modelos de projetos são como Templates que são usados para criar a estrutura inicial de projetos Java (Aplicação Java, Aplicação Web, etc).

Passos a serem executados:

1. Escolha a opção: Create a Maven Project
2. Escolha a arquitetura/template: maven-archetype-quickstart
3. Escolha a versão mais recente: neste tutorial 1.4
4. Forneça o nome do group ID: ifmt.cba.<identificação do projeto>
5. Forneça o nome do artifact ID: <identificação do projeto>
6. Selecione a pasta onde o projeto será criado

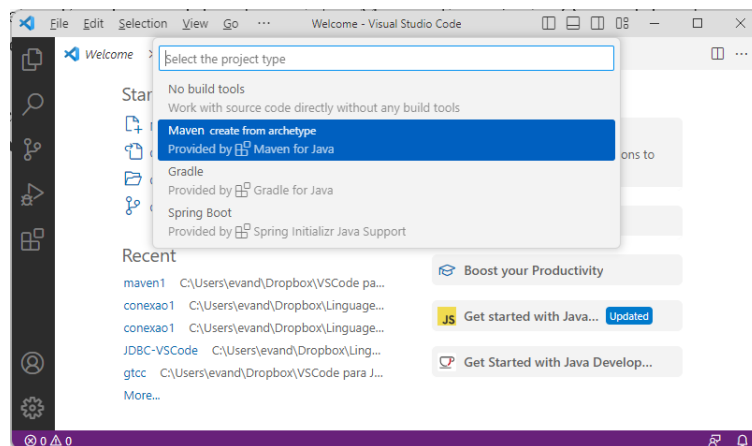
Essas informações serão preenchidas conforme as ilustrações a seguir. Para criar um projeto Java Maven, acesse a opção com a combinação de **Ctrl+Shift+P** e em seguida a opção: **Java: Create Java Project**. A Figura 8 ilustra essa ação.

Figura 8 – Criando um Projeto Java



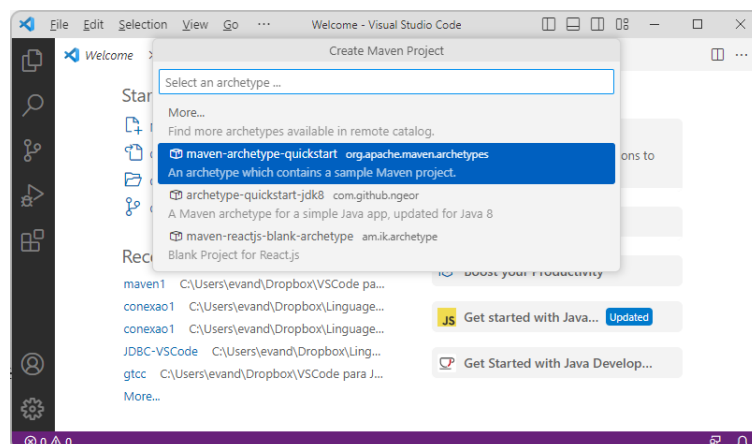
Após escolher a opção para criar um Projeto Java, temos que escolher a opção **Maven create from archetype**, ilustrado na Figura 9.

Figura 9 – Escolhendo a categoria de Projetos Maven



Em seguida temos que selecionar qual o template do Maven que será usado, neste exemplo será usado o template **maven-archetype-quickstart**, ilustrado nas Figuras 10 e 11.

Figura 10 – Definindo o Template Maven a ser usado



Próxima definição é usada para configurar o id do grupo de artefatos. Normalmente isso define uma hierarquia de pacotes e subpacotes, para agrupar conjuntos de artefatos. Nesse exemplo usaremos o grupo id: ifmt.cba, ilustrado na Figura 12.

Figura 11 – Definindo a versão do Template Maven a ser usado

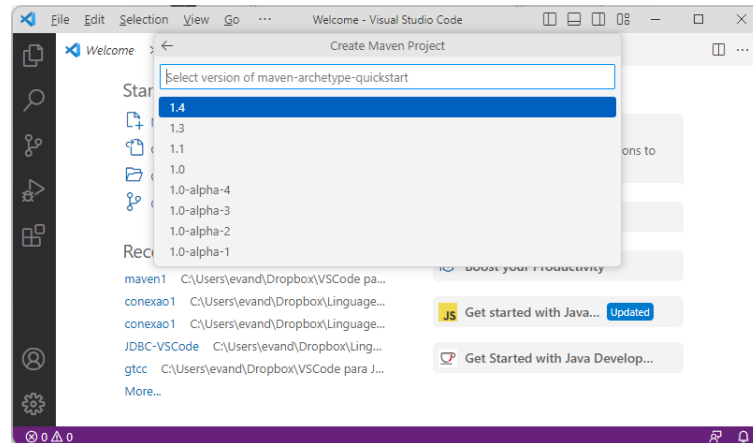
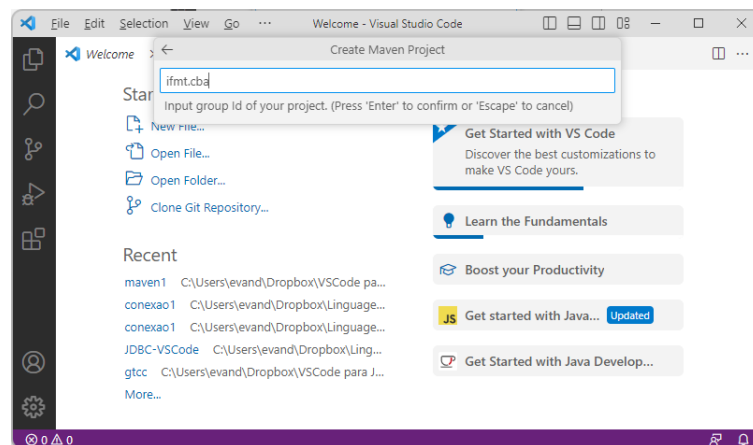
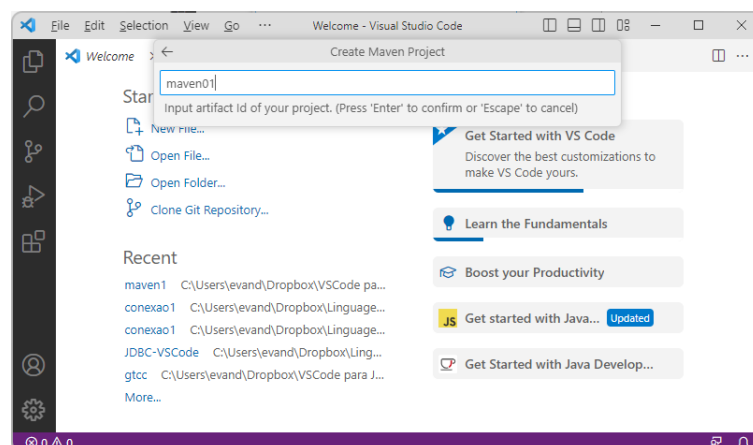


Figura 12 – Definindo o identificador de grupo do artefato



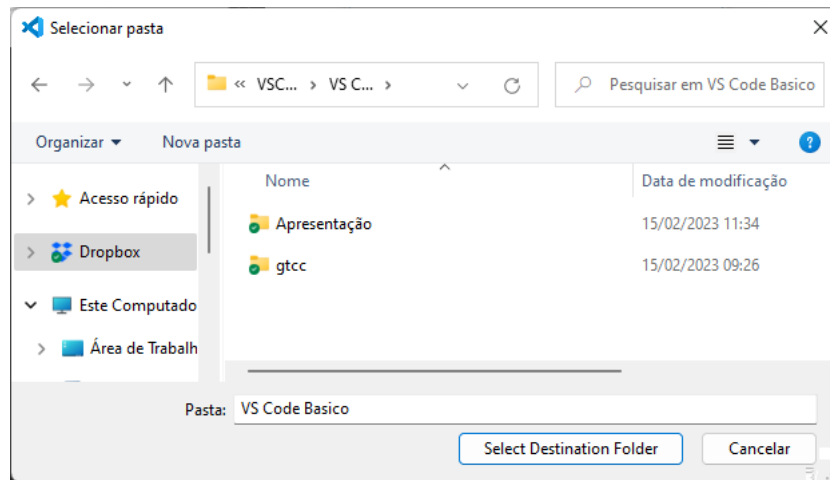
Em seguida precisamos fornecer o nome do artefato que estamos criando, no caso o nome que identifica o projeto em si. Isso foi ilustrado na Figura 13.

Figura 13 – Definindo o ID do artefato que será implementado



Próxima ação é informar a pasta que servirá de base para o Maven criar a pasta do projeto, isso está ilustrado na Figura 14.

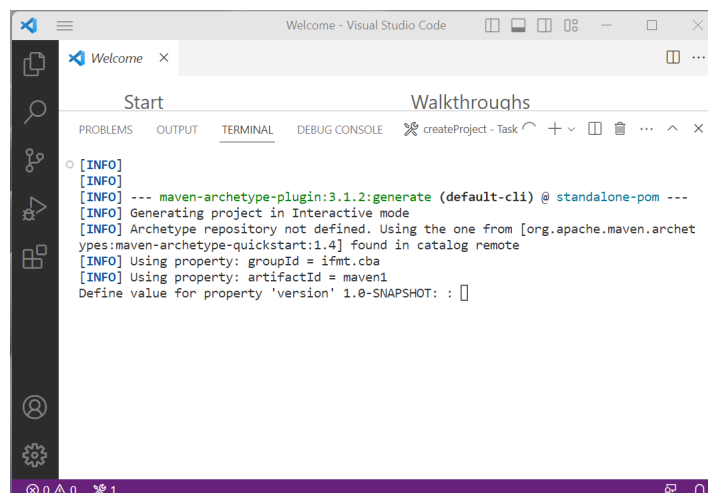
Figura 14 – Definindo a pasta onde o projeto será criado



O VS Code irá gerar o projeto a partir desse passo. Isso pode levar alguns minutos caso nunca tenha usado o Maven na máquina em questão, visto que o Maven irá baixar arquivos para um repositório local.

O processo de criação é pausado, sendo necessário a confirmação ou alteração no nome da versão do artefato/projeto que está sendo gerado. Caso queira manter a sugestão do Maven (1.0-SNAPSHOT), basta pressionar a tecla ENTER, caso queira, pode fornecer um novo nome para a versão do projeto. Isso está ilustrado na Figura 15.

Figura 15 – Criando um Projeto Maven



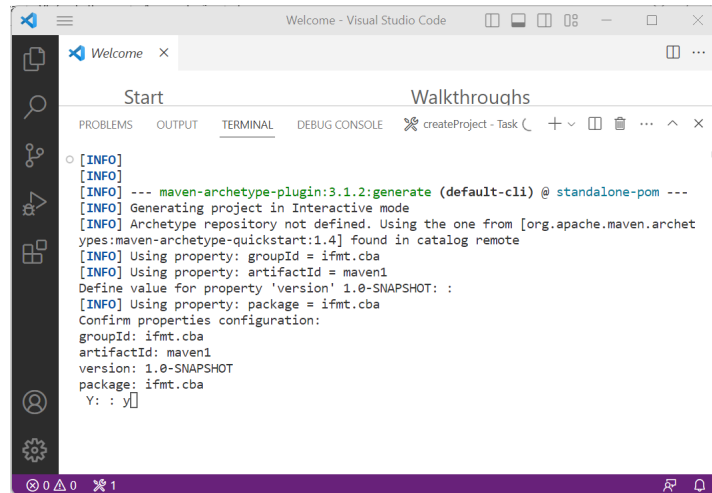
Por último o Maven irá pedir a confirmação das informações para que o processo de criação do projeto seja concluído. Caso tudo esteja correto, forneça o valor Y (Yes) para finalizar o processo, caso contrário forneça o valor N (No) para retomar as duas últimas ações. Isso está ilustrado na Figura 16.

A Figura 17 apresenta o resultado do processo de criação do projeto Maven.

A Figura 18 apresenta a visualização do projeto no ambiente VS Code. Nessa configuração estão sendo exibidas três visualizações: de Pasta, Java Projects e Maven. Nessa figura foi assinalado o local onde temos uma reticência (...), que é um botão que pode ser usado para trocar o tipo de visualização.

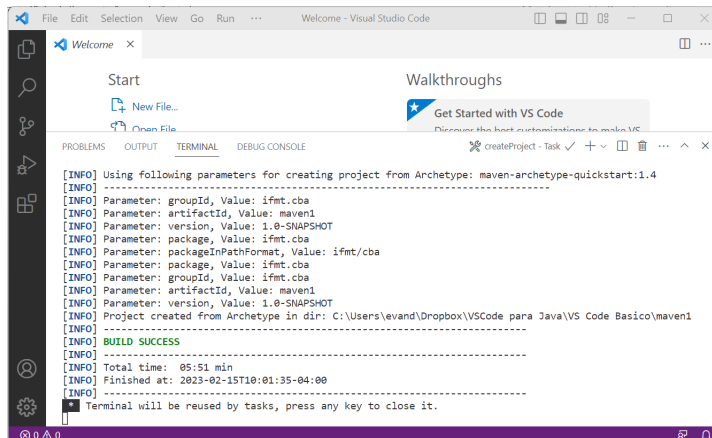
No lado do editor, está sendo exibido o arquivo pom.xml, onde pode observar os valores que foram fornecidos durante o processo de criação do projeto, como o groupId, artifactId e version. Observe

Figura 16 – Criando um Projeto Maven



```
Welcome - Visual Studio Code
Welcome x
Start Walkthroughs
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE createProject - Task
[INFO]
[INFO] --- maven-archetype-plugin:3.1.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archet
ypes:maven-archetype-quickstart:1.4] found in catalog remote
[INFO] Using property: groupId = ifmt.cba
[INFO] Using property: artifactId = maven1
Define value for property 'version' 1.0-SNAPSHOT: :
[INFO] Using property: package = ifmt.cba
Confirm properties configuration:
groupId: ifmt.cba
artifactId: maven1
version: 1.0-SNAPSHOT
package: ifmt.cba
Y: : y
```

Figura 17 – Criando um Projeto Maven

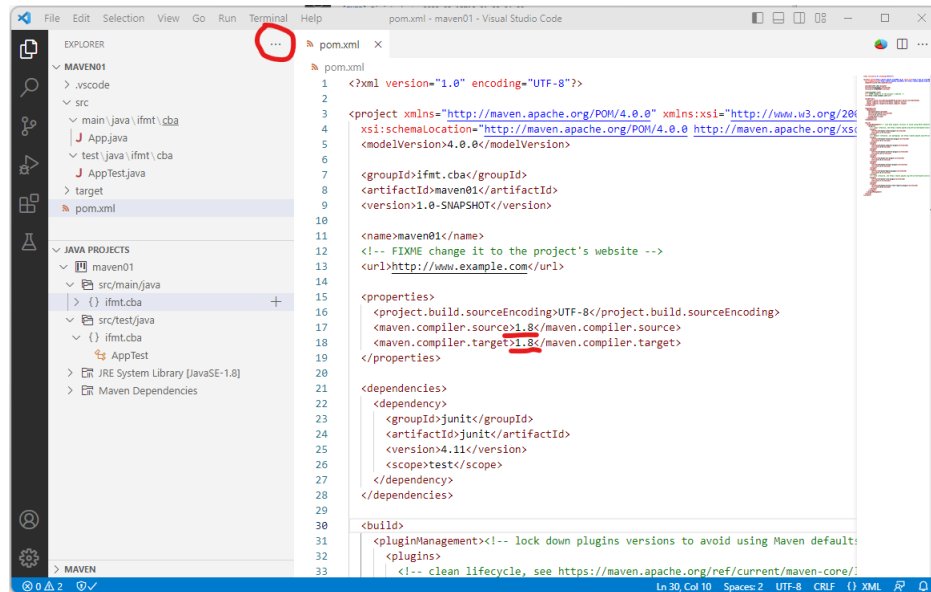


```
File Edit Selection View Go Run ... Welcome - Visual Studio Code
Welcome x
Start Walkthroughs
New File...
Open File
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE createProject - Task ✓
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: ifmt.cba
[INFO] Parameter: artifactId, Value: maven1
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: ifmt.cba
[INFO] Parameter: packageInPathFormat, Value: ifmt/cba
[INFO] Parameter: package, Value: ifmt.cba
[INFO] Parameter: groupId, Value: ifmt.cba
[INFO] Parameter: artifactId, Value: maven1
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\evand\Dropbox\VSCode para Java\VS Code Basico\maven1
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 05:51 min
[INFO] Finished at: 2023-02-15T10:01:35-04:00
[INFO] -----
Terminal will be reused by tasks, press any key to close it.
```

que o valor das propriedades source e target foram alteradas para a 1.8, pois é a versão mais apropriada para a compilação e geração de bytes code para o tipo de projeto básico que iremos trabalhar.

Nesse template usado, o Maven gera também uma configuração de dependência com JUnit, que é um framework de implementação e execução de testes automatizado do Java.

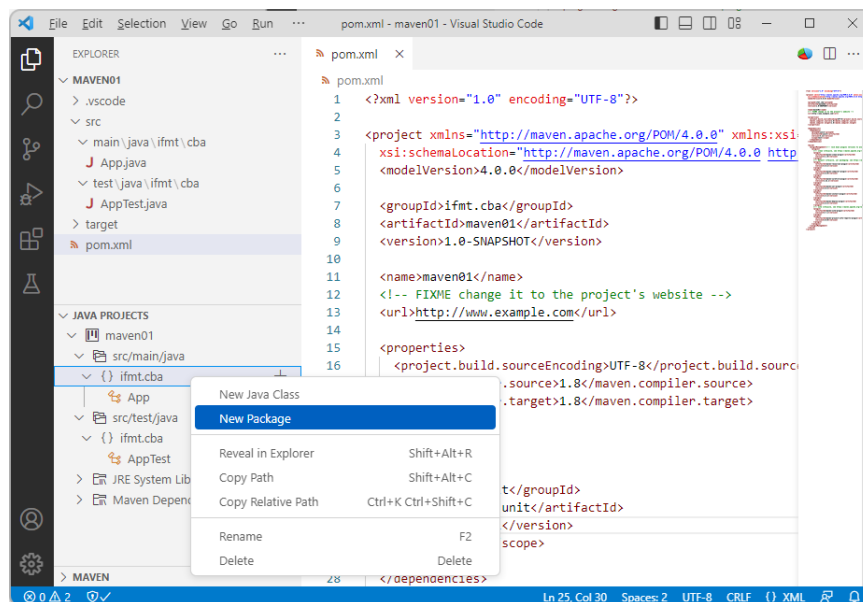
Figura 18 – Estrutura do Projeto Criado



2.1 Criando a Estrutura de Pacotes

Após criar o projeto, deve-se criar a estrutura de pacotes para abrigar os códigos fontes a serem implementados. Para isso, use a aba de Java Projects do VSCode, e no pacote raiz de código fonte `src/main/java/ifmt/cba`, crie os seguintes subpacotes: `negocio` e `execucao`. Veja Figura 19 ilustra esse procedimento.

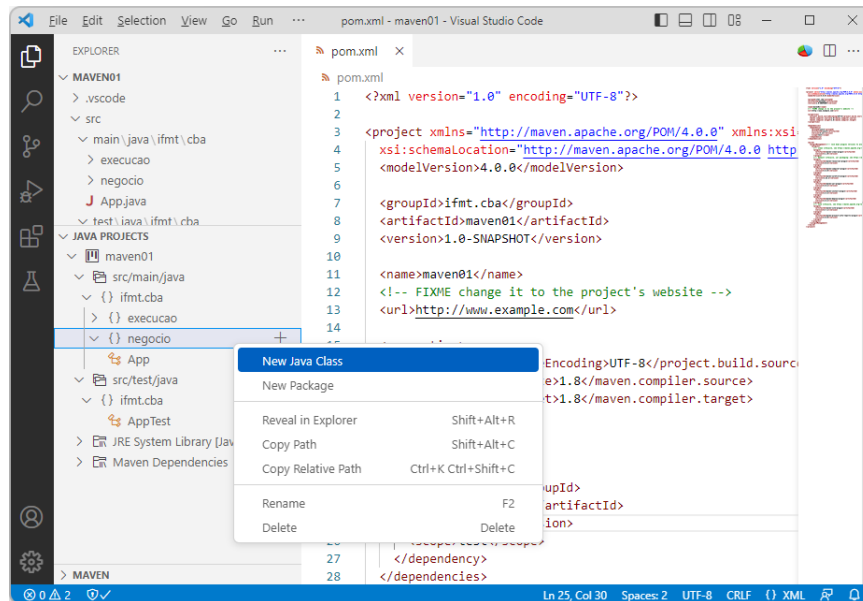
Figura 19 – Criando Pacotes



2.2 Codificando os Elementos de Código

Na sequência, vamos criar uma classe denominada Calculadora, no pacote negocio. Veja a Figura 20.

Figura 20 – Criando Classes



Veja o código implementado na classe Calculadora.

```
1 package ifmt.cba.negocio;
2
3 public class Calculadora {
4     public double somar(double valor1, double valor2) {
5         return valor1 + valor2;
6     }
7
8     public double subtrair(double valor1, double valor2) {
9         return valor1 - valor2;
10    }
11
12    public double multiplicar(double valor1, double valor2) {
13        return valor1 * valor2;
14    }
15
16    public double dividir(double valor1, double valor2) throws ArithmeticException {
17        if (valor2 != 0) {
18            return valor1 / valor2;
19        } else {
20            throw new ArithmeticException("Divisao por zero");
21        }
22    }
23 }
24 }
```

Código 1 – Classe Calculadora

Veja o código implementado na classe AppCalculadora.

```
1 package ifmt.cba.execucao;
2
3 import ifmt.cba.negocio.Calculadora;
4
5 public class AppCalculadora {
6     public static void main(String[] args) {
7         Calculadora calc = new Calculadora();
8         System.out.println("_____");
9         System.out.println("Soma de 10+30= "+calc.somar(10, 30));
10        System.out.println("Subtracao de 30-15= "+calc.subtrair(30, 15));
11        System.out.println("Multiplicacao de 20*30= "+calc.multiplicar(20, 30));
12        System.out.println("Divisao de 50/2= "+calc.dividir(50, 2));
13        System.out.println("_____");
14    }
15 }
```

Código 2 – Classe AppCalculadora

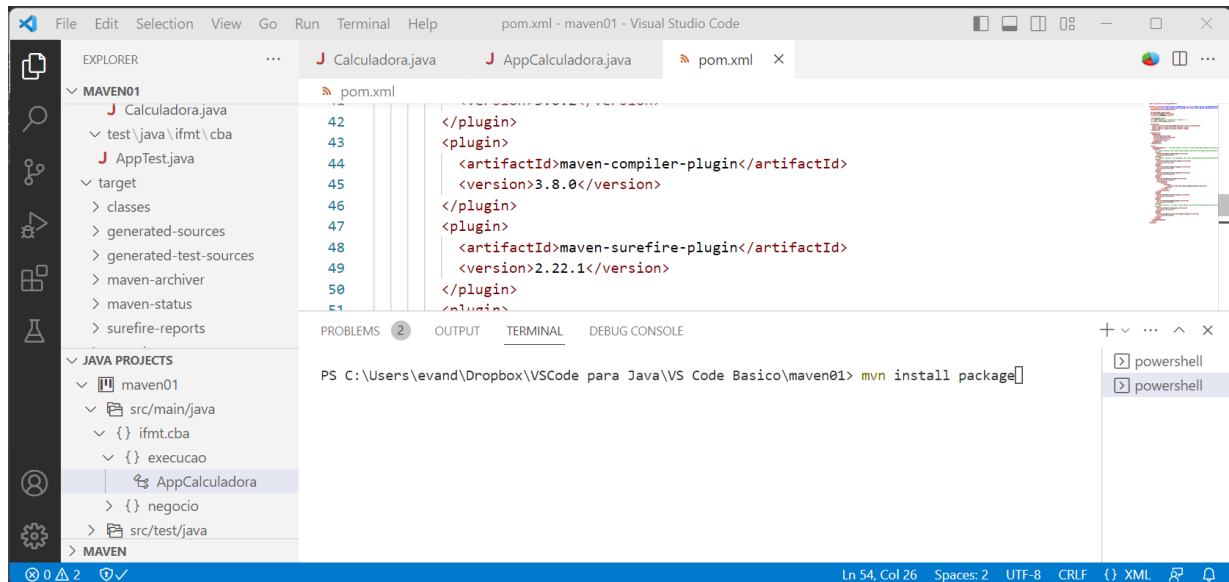
O código a seguir apresenta o arquivo de configurações do Maven (pom.xml) resultante.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
5   >
6   <modelVersion>4.0.0</modelVersion>
7
8   <groupId>ifmt.cba</groupId>
9   <artifactId>maven01</artifactId>
10  <version>1.0-SNAPSHOT</version>
11
12  <name>maven01</name>
13  <!-- FIXME change it to the project's website -->
14  <url>http://www.example.com</url>
15
16  <properties>
17    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18    <maven.compiler.source>1.8</maven.compiler.source>
19    <maven.compiler.target>1.8</maven.compiler.target>
20  </properties>
21
22  <dependencies>
23    <dependency>
24      <groupId>junit</groupId>
25      <artifactId>junit</artifactId>
26      <version>4.11</version>
27      <scope>test</scope>
28    </dependency>
29  </dependencies>
30
31  <build>
32    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to
   parent pom) -->
33    <plugins>
34      <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#
   clean_Lifecycle -->
35      <plugin>
36        <artifactId>maven-clean-plugin</artifactId>
37        <version>3.1.0</version>
38      </plugin>
39      <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/
   default-bindings.html#Plugin_bindings_for_jar_packaging -->
40      <plugin>
41        <artifactId>maven-resources-plugin</artifactId>
42        <version>3.0.2</version>
43      </plugin>
44      <plugin>
45        <artifactId>maven-compiler-plugin</artifactId>
46        <version>3.8.0</version>
47      </plugin>
48      <plugin>
49        <artifactId>maven-surefire-plugin</artifactId>
50        <version>2.22.1</version>
51      </plugin>
52      <plugin>
53        <artifactId>maven-jar-plugin</artifactId>
54        <version>3.0.2</version>
55        <configuration>
56          <archive>
57            <manifest>
58              <mainClass>ifmt.cba.execucao.AppCalculadora</mainClass>
59            </manifest>
60          </archive>
61        </configuration>
62      </plugin>
63      <plugin>
64        <artifactId>maven-install-plugin</artifactId>
65        <version>2.5.2</version>
66      </plugin>
67      <plugin>
68        <artifactId>maven-deploy-plugin</artifactId>
69        <version>2.8.2</version>
70      </plugin>
71      <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#
   site_Lifecycle -->
72      <plugin>
73        <artifactId>maven-site-plugin</artifactId>
74        <version>3.7.1</version>
75      </plugin>
76      <plugin>
77        <artifactId>maven-project-info-reports-plugin</artifactId>
78        <version>3.0.0</version>
79      </plugin>
80    </plugins>
81  </pluginManagement>
82</build>
</project>
```

Código 3 – Arquivo pom.xml

Gerando o Jar executável do projeto a partir de uma tarefa do Maven (mvn install package)

Figura 21 – Gerando Jar Executável



Executando o Jar do projeto (java -jar <nome-jar>)

Figura 22 – Executando o Jar do projeto

