

# Tutorial

Cesar Hernandez

Version 1.0-SNAPSHOT, 2019-04-08 23:02:51 UTC

# Tabla de Contenidos

1. Introducción .....	1
2. Prerequisitos .....	1
3. Crear la estructura base del proyecto .....	1
3.1. Crear un nuevop proyecto con Maven .....	1
3.2. Agregar Asciidoctor al proyecto .....	2
3.3. Creación de tu primer documento AsciiDoc. ....	3
3.4. Generación de tu primer documento en formatos HTML y PDF .....	4

# 1. Introducción

Este tutorial muestra cómo iniciar un proyecto Java Maven desde cero y construyó toda la estructura necesaria para tener capacidades AsciiDoctor junto con la revisión ortográfica de diferentes idiomas como parte de las pruebas del proyecto.

## 2. Prerequisitos

- Java Development Kit >= 7
- Maven >= 3.3.9

## 3. Crear la estructura base del proyecto

### 3.1. Crear un nuevop proyecto con Maven

En una terminal ejecutar el siguiente comando:

```
mvn archetype:generate -DgroupId=asciidoctor.doc.demo -DartifactId=book-demo
-DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4
-DinteractiveMode=false
```

En una terminal `cd` en el directorio generado en el paso anterior:

```
cd book-demo/
```

La estructura de directorios generados debe ser igual a esta:

```
├── pom.xml
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── asciidoctor
│   │   │   │   ├── doc
│   │   │   │   │   ├── demo
│   │   │   │   │   │   App.java
│   │   └── test
│   │       ├── java
│   │       │   ├── asciidoctor
│   │       │   │   ├── doc
│   │       │   │   │   ├── demo
│   │       │   │   │   │   AppTest.java
```

El archivo `pom.xml` contiene la información básica del proyecto recién creado.

## 3.2. Agregar Asciidoctor al proyecto

En un editor de texto, abrir el archivo `pom.xml` y realizar las siguiente modificaciones:

- Actualizar la sección `<properties>` con el siguiente contenido:

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
  <asciidoctor.maven.plugin.version>1.5.7.1</asciidoctor.maven.plugin.version>
  <asciidoctorj.pdf.version>1.5.0-alpha.16</asciidoctorj.pdf.version>
  <asciidoctorj.version>1.5.8.1</asciidoctorj.version>
  <maven.build.timestamp.format>yyyy-MM-dd HH:mm:ss
z</maven.build.timestamp.format>
</properties>
```

- En la sección `<build>`, agregar el siguiente contenido despues de la etiqueta: `</pluginManagement>`

```
<plugins>
  <plugin>
    <groupId>org.asciidoctor</groupId> ①
    <artifactId>asciidoctor-maven-plugin</artifactId>
    <version>${asciidoctor.maven.plugin.version}</version>

    <dependencies> ②
      <dependency>
        <groupId>org.asciidoctor</groupId>
        <artifactId>asciidoctorj-pdf</artifactId> ③
        <version>${asciidoctorj.pdf.version}</version>
      </dependency>
      <dependency>
        <groupId>org.asciidoctor</groupId>
        <artifactId>asciidoctorj</artifactId>
        <version>${asciidoctorj.version}</version>
      </dependency>
    </dependencies>

    <executions>
      <execution>
        <id>output-html</id> ④
        <phase>generate-resources</phase>
        <goals>
          <goal>process-asciidoc</goal>
        </goals>
        <configuration>
          <backend>html5</backend>
          <sourceHighlighter>highlight.js</sourceHighlighter>
          <preserveDirectories>true</preserveDirectories>
```

```

    <attributes>
      <toc>left</toc>
      <sectanchors>>true</sectanchors>
    </attributes>
  </configuration>
</execution>
<execution>
  <id>generate-pdf-doc</id> ⑤
  <phase>generate-resources</phase>
  <goals>
    <goal>process-asciidoc</goal>
  </goals>
  <configuration>
    <backend>pdf</backend>
    <sourceHighlighter>highlight.js</sourceHighlighter>
    <preserveDirectories>>true</preserveDirectories>
    <attributes>
      <pagenums/>
      <toc/>
      <idprefix/>
      <idseparator>-</idseparator>
    </attributes>
  </configuration>
</execution>
</executions>

<configuration> ⑥
  <attributes>
    <project-version>${project.version}</project-version>
    <revdate>${maven.build.timestamp}</revdate>
    <sectnums>>true</sectnums>
    <icons>font</icons>
  </attributes>
</configuration>
</plugin>
</plugins>

```

- ① Referencia al plugin `asciidoctor-maven-plugin`.
- ② Dependencias utilizadas por el plugin `asciidoctor-maven-plugin`.
- ③ Provee PDF export al proyecto.
- ④ Configuración para reportes en formato HTML.
- ⑤ Configuración para reportes en formato PDF.
- ⑥ Configuración global para ambos executions.

### 3.3. Creación de tu primer documento AsciiDoc

- Crea el directorio por defecto de donde almacenaras tus archivos AsciiDoc (.adoc):

```
mkdir src/main/asciidoc
```

- Dentro del directorio `src/main/asciidoc` crea un archivo con nombre `index_en.adoc` y agrega el siguiente contenido:

```
= Tutorial ①
:nofooter: ②
:toc-title: Tabla de Contenidos ③
Autor ④
{project-version}, {revdate} ⑤

== Capítulo 1 ⑥
Este es mi primer capítulo.
```

- ① Define el encabezado y nombre principal del documento.
- ② Remueve del documento el pie de página que se genera por defecto.
- ③ Sobre escribe el valor por defecto (Table content) del título de la tabla de contenidos.
- ④ Nombre del autor del documento.
- ⑤ Mapeo de la versión y fecha de revisión del documento obtenido a partir del archivo `pom.xml`.
- ⑥ Indica el primer sub encabezado del documento.



Debes respetar que antes del header principal no haya texto, esto ayuda a que durante el proceso de renderización los parámetros del documento (líneas 2 a la 5) sean efectivos.

## 3.4. Generación de tu primer documento en formatos HTML y PDF

- En la terminal ejecuta el siguiente comando:

```
mvn clean generate-resources
```

- Los archivos para cada formato se generan en el folder: `target/generated-docs/`
  - `index_en.html`
  - `index_en.pdf`

La version html generada es similar a la siguiente imagen:

Tabla de Contenidos

1. Capítulo 1

# Tutorial

Autor – Version 1.0-SNAPSHOT, 2019-04-07 18:35:27 UTC

---

## 1. Capítulo 1

Este es mi primer capítulo.