

Documenta tu Aplicación Java con Sphinx

David Dali Susanibar Arce

dsusanibara@uni.pe

Agenda

- Que es Sphinx?
- Demo Sphinx
- Demo Uso Java
- Demo Uso Java Maven

Que es Sphinx



- Sphinx es un software **generador de documentación** que convierte ficheros **reStructuredText** (lenguaje de marcas) en **sitios web HTML** y otros formatos, incluyendo **PDF**, **EPub** y **man**. Saca provecho de la naturaleza extensible de reStructuredText y sus extensiones (ej. para generar automáticamente documentación desde **código fuente**, escribir notación matemática o resaltar código).

Git - 0

- \$ clonar repositorio: <https://github.com/davisusanibar/jconf-sphinx>
- \$ git branch
 - main
 - sphinx-documentacion
 - sphinx-documentacion-texto
 - sphinx-documentacion-texto-codigo
 - sphinx-documentacion-texto-codigo-validacion
 - sphinx-documentacion-texto-codigo-validacion-extension-custom
 - sphinx-documentacion-texto-codigo-validacion-extension-custom-maven
 - sphinx-java-options
- \$ git checkout sphinx-documentacion

Sphinx en VM

- Instalar Sphinx
 - \$ pip install -U sphinx
- Para generar el esqueleto base de Sphinx debemos usar su interactive tool llamado sphinx-quickstart
- \$ sphinx-quickstart →

```
├── Makefile
├── make.bat
├── source
│   ├── _static
│   ├── _templates
│   ├── conf.py
│   └── index.rst
```

```
├── Makefile
├── _build
├── _static
├── _templates
├── conf.py
├── index.rst
└── make.bat
```

```
> sphinx-quickstart
```

Welcome to the Sphinx 7.2.6 quickstart utility.

Please enter values for the following settings (just press Enter to accept a default value, if one is given in brackets).

Selected root path: .

You have two options for placing the build directory for Sphinx output. Either, you use a directory "_build" within the root path, or you separate "source" and "build" directories within the root path.

```
> Separate source and build directories (y/n) [n]: y
```

The project name will occur in several places in the built documentation.

```
> Project name: JCONF Peru 2023
```

```
> Author name(s): JConf
```

```
> Project release []:
```

If the documents are to be written in a language other than English, you can select a language here by its language code. Sphinx will then translate text that it generates into that language.

For a list of supported codes, see

<https://www.sphinx-doc.org/en/master/usage/configuration.html#confval-language>.

```
> Project language [en]:
```

```
Creating file /Users/dsusanibar/Documents/testme/source/conf.py.
```

```
Creating file /Users/dsusanibar/Documents/testme/source/index.rst.
```

```
Creating file /Users/dsusanibar/Documents/testme/Makefile.
```

```
Creating file /Users/dsusanibar/Documents/testme/make.bat.
```

Finished: An initial directory structure has been created.


You should now populate your master file /Users/dsusanibar/Documents/testme/source/index.rst and create other documentation

source files. Use the Makefile to build the docs, like so:

```
make builder
```

where "builder" is one of the supported builders, e.g. html, latex or linkcheck.

Sphinx en Docker

- Crear directorio para el volumen docker que rapeara la estructura de archivos del proyecto Sphinx con el host:
 - `$ mkdir jconf`
- Generamos el esqueleto base de Sphinx con:
 - `$ docker run -it --rm -v ./jconf:/docs sphinxdoc/sphinx sphinx-quickstart`
 - `$ tree jconf` 

```
> tree jconf
jconf
├── Makefile
├── build
├── make.bat
├── source
│   ├── _static
│   ├── _templates
│   ├── conf.py
│   └── index.rst
5 directories, 4 files
```



```
> docker run -it --rm -v ./jconf:/docs sphinxdoc/sphinx sphinx-quickstart
```

Welcome to the Sphinx 7.1.2 quickstart utility.

Please enter values for the following settings (just press Enter to accept a default value, if one is given in brackets).

Selected root path: .

You have two options for placing the build directory for Sphinx output. Either, you use a directory "_build" within the root path, or you separate "source" and "build" directories within the root path.

```
> Separate source and build directories (y/n) [n]: y
```

The project name will occur in several places in the built documentation.

```
> Project name: JConf 2023
```

```
> Author name(s): JConf
```

```
> Project release []:
```

If the documents are to be written in a language other than English, you can select a language here by its language code. Sphinx will then translate text that it generates into that language.

For a list of supported codes, see

<https://www.sphinx-doc.org/en/master/usage/configuration.html#confval-language>.

```
> Project language [en]:
```

```
Creating file /docs/source/conf.py.
```

```
Creating file /docs/source/index.rst.
```

```
Creating file /docs/Makefile.
```

```
Creating file /docs/make.bat.
```

Finished: An initial directory structure has been created.

You should now populate your master file /docs/source/index.rst and create 05 directories, 4 files source files. Use the Makefile to build the docs, like so:

```
make builder
```

where "builder" is one of the supported builders, e.g. html, latex or linkcheck.

```
> tree jconf
jconf
├── Makefile
├── build
├── make.bat
├── source
│   ├── _static
│   ├── _templates
│   ├── conf.py
│   └── index.rst
```


Lab - 0 - Lo que viene de caja

- VM
 - `$ pip install -U sphinx`
 - `$ mkdir jconf-lab0`
 - `$ cd jconf-lab0`
 - `$ sphinx-quickstart`
 - `$ make html`
- Docker
 - `$ mkdir jconf-lab0`
 - `$ docker run -it --rm -v ./jconf-lab0:/docs sphinxdoc/sphinx sphinx-quickstart`
 - `$ docker run -it --rm -v ./jconf-lab0:/docs sphinxdoc/sphinx make html`

Git - 01

- `$ git checkout sphinx-documentacion-texto`

Documentación Texto

- Abrimos el proyecto creado por el interactive tool en nuestro IDE favorito.
- Generamos nuestra documentación en el archivo “jconf/source/index.rst”
- Una vez terminada nuestra documentación lo compilamos al formato que necesitemos html, latex, ePub, otros.

Documentación Texto

Cabeceras / Títulos

#

*

=

-

^

“

Bloques de Código

```
.. code-block:: python
```

```
import ...
```

Links y Referencias

Mi Titulo

=====

Mayor detalle en `Guia de Sphinx`_
con ejemplos reales.

... _`Guia de Sphinx`: <https://demo.com>

<https://documentation-style-guide-sphinx.readthedocs.io/en/latest/style-guide.html>

https://sphinx-intro-tutorial.readthedocs.io/en/latest/rst_intro.html

Documentación Texto

#

*

=

-

^

“

```
6 Welcome to JConf 2023's documentation!
7 =====
8
9 .. toctree::
10    :maxdepth: 2
11    :caption: Contents:
12
13
14
15 Indices and tables
16 =====
17
18 * :ref:`genindex`
19 * :ref:`modindex`
20 * :ref:`search`
21
22 Expositores
23 -----
24
25 1. Carlos Zela
26 2. Victor Mandujano
27 3. David Susanibar
28
29 Carlos Zela
30 ^^^^^^^^^^^
31
32 - C++
33 - Python
34 - Java
35
36 Redes Sociales
37 ~~~~~~
38
39 - X
40 - Linkedin
```

JConf 2023

Navigation

Quick search

Go

Welcome to JConf 2023's documentation!

Indices and tables

- [Index](#)
- [Module Index](#)
- [Search Page](#)

Expositores

- Carlos Zela
- Victor Mandujano
- David Susanibar

Carlos Zela

- C++
- Python
- Java

Redes Sociales

- X
- Linkedin

\$ make html

Lab - 1 - Documentación Texto

- VM
 - `$ pip install -U sphinx`
 - `$ mkdir jconf-lab1`
 - `$ cd jconf-lab1`
 - `$ sphinx-quickstart`
 - Agregar texto al documento: `./jconf-lab1/source/index.rst`
 - `$ make html`
- Docker
 - `$ mkdir jconf-lab1`
 - `$ docker run -it --rm -v ./jconf-lab1:/docs sphinxdoc/sphinx sphinx-quickstart`
 - Agregar texto al documento: `./jconf-lab1/source/index.rst`
 - `$ docker run -it --rm -v ./jconf-lab1:/docs sphinxdoc/sphinx make html`

Git - 02

- `$ git checkout sphinx-documentacion-texto-codigo`

Documentación Código

```
29 Carlos Zela
30 ^^^^^^^^^^^^^
31
32 - Python
33
34 .. code-block:: python
35
36     print('Hi JConf 2023 by Python')
37
38
39 - Java
40
41 .. code-block:: java
42
43     System.out.println("Hi JConf 2023 by Java");
44
```

\$ make html



Carlos Zela

- Python

```
print('Hi JConf 2023 by Python')
```

- Java

```
System.out.println("Hi JConf 2023 by Java");
```

Lab - 2 - Documentación Código

- VM
 - `$ pip install -U sphinx`
 - `$ mkdir jconf-lab2`
 - `$ cd jconf-lab2`
 - `$ sphinx-quickstart`
 - Agregar código al documento (usando la directiva: code-block): `./jconf-lab2/source/index.rst`
 - `$ make html`
- Docker
 - `$ mkdir jconf-lab2`
 - `$ docker run -it --rm -v ./jconf-lab2:/docs sphinxdoc/sphinx sphinx-quickstart`
 - Agregar código al documento (usando la directiva: code-block): `./jconf-lab2/source/index.rst`
 - `$ docker run -it --rm -v ./jconf-lab2:/docs sphinxdoc/sphinx make html`

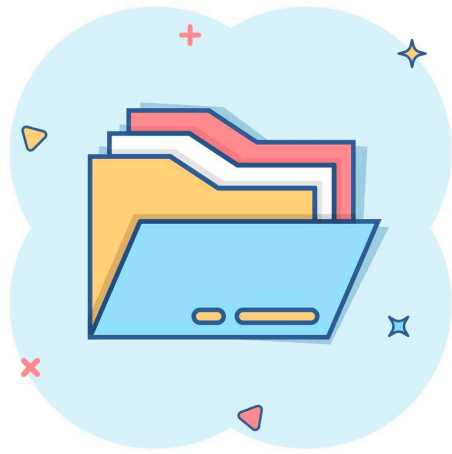
Git - 03

- `$ git checkout sphinx-documentacion-texto-codigo-validacion`

Documentación Validación Código

- Sphinx cuenta con una variedad de “extensiones” para poder extender la funcionalidad de la aplicación.
- Estas extensiones se registran en el archivo “conf.py”

Documentación Validación Código



conf.py

```
extensions = [  
    'sphinx.ext.doctest',  
]
```

Este es un ejemplo que declara su código a ejecutar usando la directiva `testcode` y valida la salida del programa con lo que se declara en la directiva `testoutput`.

```
.. testcode::
```

```
print('JConf Python')
```

```
.. testoutput::
```

```
JConf Python
```

Resultado Validación

\$ make doctest

```
Doctest summary  
=====  
1 test  
0 failures in tests  
0 failures in setup code  
0 failures in cleanup code  
build succeeded.
```

\$ make html

Este es un ejemplo que declara su código a ejecutar usando la directiva `testcode` y valida la salida del programa con lo que se declara en la directiva `testoutput`.

```
print('JConf Python')
```

```
JConf Python
```


Lab - 3 - Documentación Validación Código

- VM
 - `$ pip install -U sphinx`
 - `$ mkdir jconf-lab3`
 - `$ cd jconf-lab3`
 - `$ sphinx-quickstart`
 - Agregar código al documento (usando la directiva: `testcode`): `./jconf-lab3/source/index.rst`
 - Agregar extension al archivo de configuración `./jconf-lab3/source/conf.py`: `sphinx.ext.doctest`
 - `$ make html`
 - `$ make doctest`
- Docker
 - `$ mkdir jconf-lab3`
 - `$ docker run -it --rm -v ./jconf-lab3:/docs sphinxdoc/sphinx sphinx-quickstart`
 - Agregar código al documento (usando la directiva: `testcode`): `./jconf-lab3/source/index.rst`
 - Agregar extension al archivo de configuración `./jconf-lab3/source/conf.py`: `sphinx.ext.doctest`
 - `$ docker run -it --rm -v ./jconf-lab3:/docs sphinxdoc/sphinx make doctest`
 - `$ docker run -it --rm -v ./jconf-lab3:/docs sphinxdoc/sphinx make html`

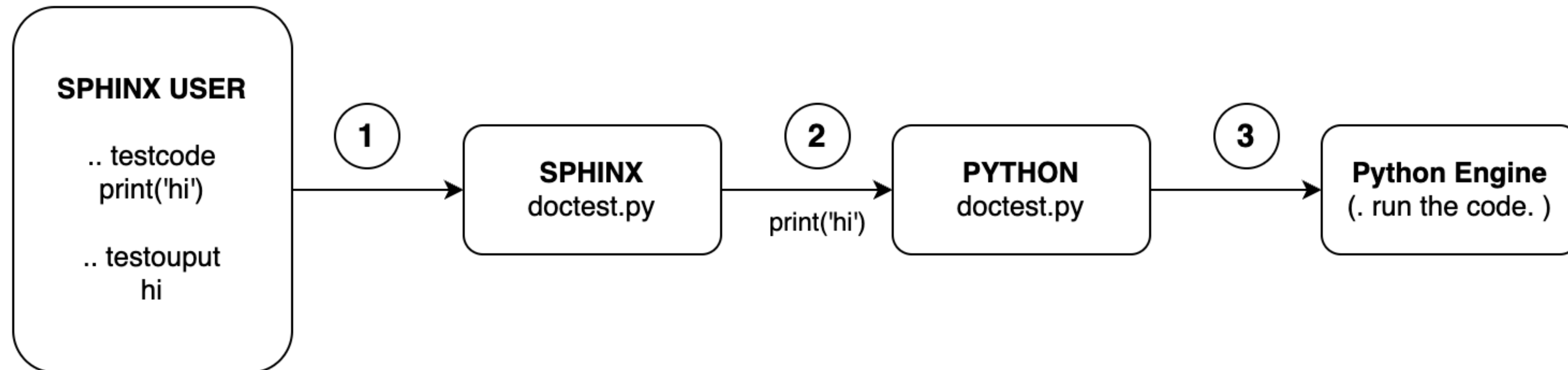
Git - 04

- `$ git checkout sphinx-documentacion-texto-codigo-validacion-extension-custom`

Documentación Validación Código con Extension Custom

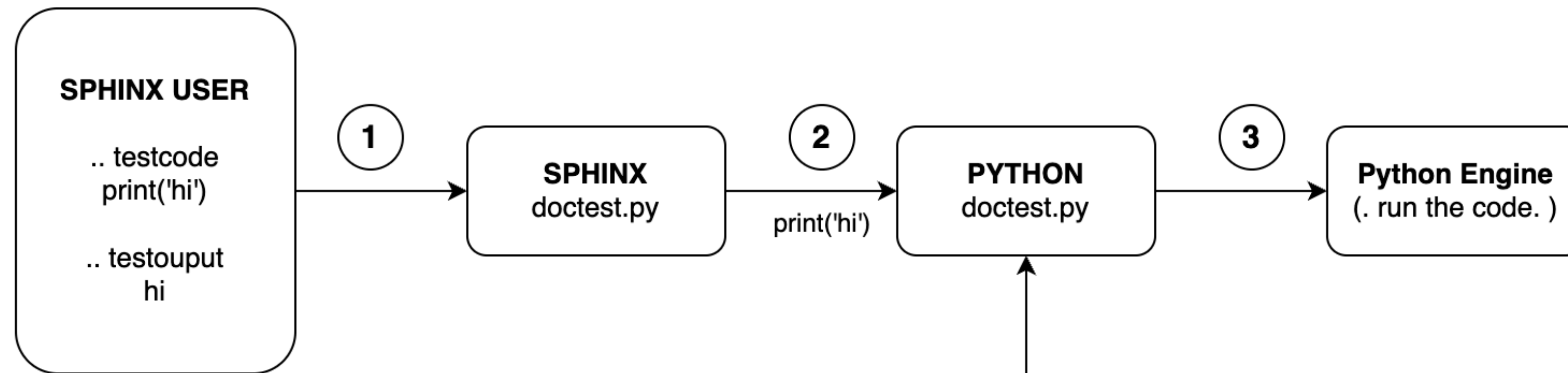
- Sphinx ofrece configuraciones para prender cosas que podrías necesitar y apagar otras no necesarias.
- En el siguiente ejemplo vamos a ver como cargar una extension que nos permita documentar y validar nuestro código Java usando Sphinx.

Arquitectura Sphinx

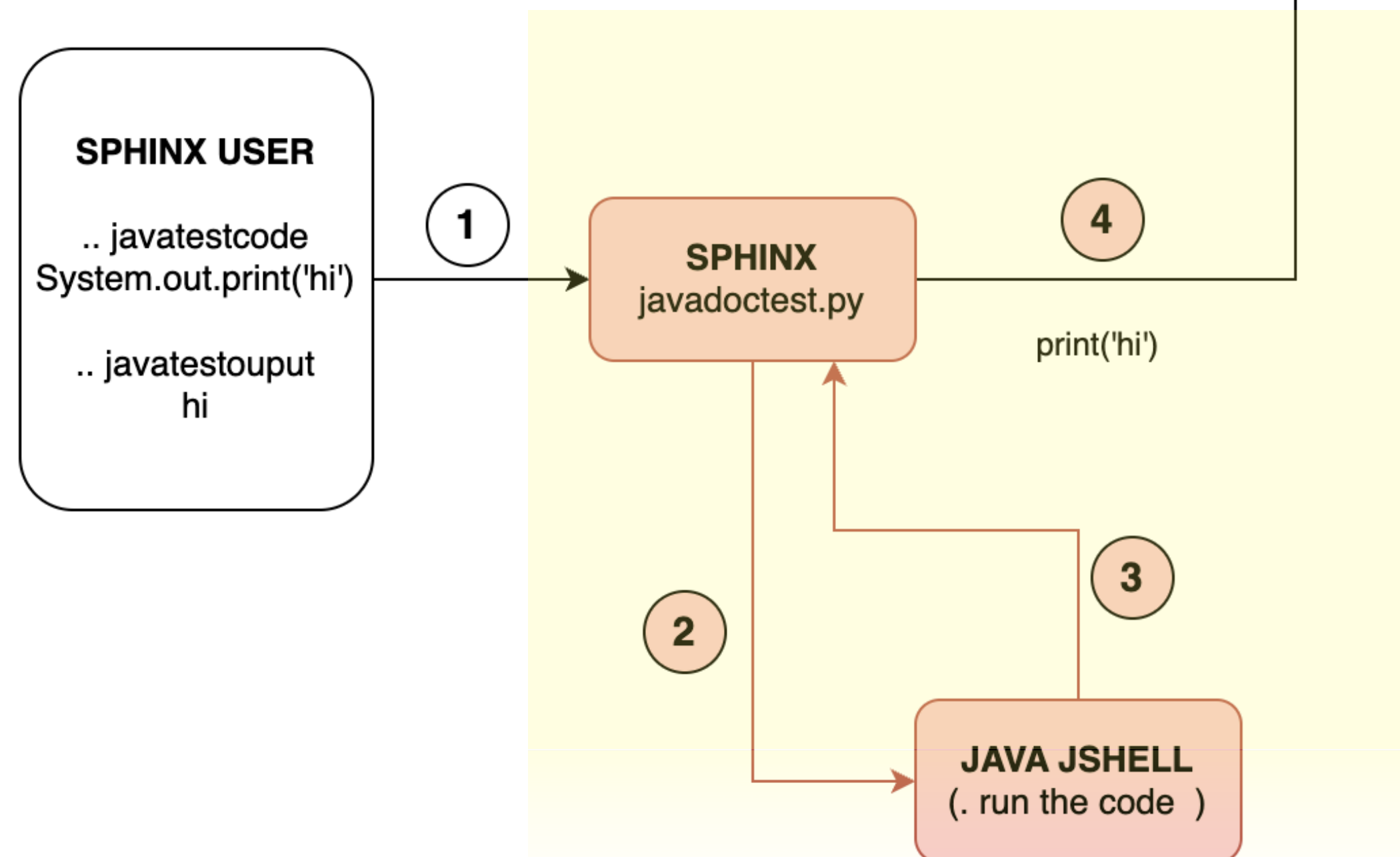


https://docs.google.com/document/d/1VXaqgFdZR07UeCJH-R7kztX4WE1VK_DFr5mZz_lu-vM/edit?usp=sharing

Arquitectura **Propuesta** Sphinx para Soportar **Java**



<https://gist.github.com/davisusanibar/1ad47ec8a12d91e311558ab01f912f26>



<https://groups.google.com/g/sphinx-dev/c/0H1JWxUYMNM>

Documentación Validación Código con Extension Custom



conf.py

```
import os
import sys
```

```
sys.path.append(
    os.path.abspath("../ext"))
```

```
extensions = [
    "javadoctest"
]
```

Resultado Validación

\$ make javadoctest

```
Doctest summary
=====
1 test
0 failures in tests
0 failures in setup code
0 failures in cleanup code
build succeeded.
```

\$ make html

Este es un ejemplo que declara su código a ejecutar usando la directiva `javatestcode` y valida la salida del programa con lo que se declara en la directiva `javatestoutput`.

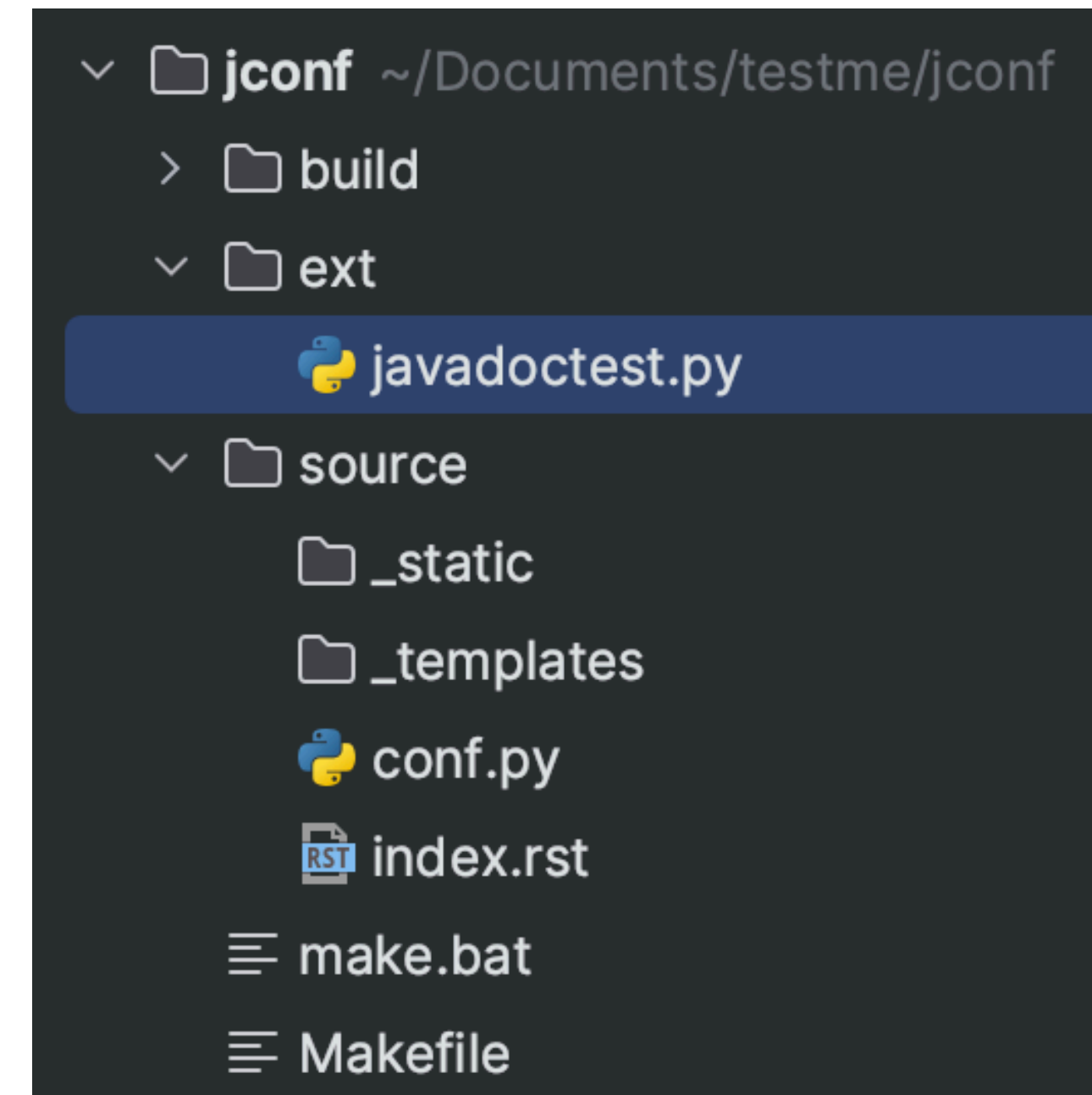
```
System.out.println("JConf Java");
```

```
JConf Java
```

ext/javadoctest.py

```
app.add_directive('javatestsetup', JavaTestsetupDirective)
app.add_directive('javatestcleanup', JavaTestcleanupDirective)
app.add_directive('javadoctest', JavaDoctestDirective)
app.add_directive('javatestcode', JavaTestcodeDirective)
app.add_directive('javatestoutput', JavaTestoutputDirective)
```

(Nueva Directivas Para Java)



Lab - 4 - Documentación **Validación Código con** **Extension Custom**

- Docker

- \$ mkdir jconf-lab4

- \$ cd jconf-lab4

- \$ cat Dockerfile

```
# Sphinx + Java/Maven Support
```

```
FROM sphinxdoc/sphinx
```

```
WORKDIR /docs
```

```
RUN apt-get update && apt-get -y install maven openjdk-17-jdk
```

Lab - 4 - Documentación **Validación Código con Extension Custom**

- Docker (cont)
 - `$ docker build . -t sphinx-jconf:v1`
 - Agregar código al documento (usando la directiva: `javatestcode`): `./jconf-lab4/source/index.rst`
 - Crear el archivo `./jconf-lab4/ext/javadoctest.py` con las directivas creadas en <https://gist.github.com/davisusanibar/1ad47ec8a12d91e311558ab01f912f26>
 - Agregar la nueva extension al archivo de configuración `./jconf-lab4/source/conf.py`: `javadoctest`
 - `$ cd ..`
 - `$ docker run -it --rm -v ./jconf-lab4:/docs sphinx-jconf:v1 make javadoctest`
 - `$ docker run -it --rm -v ./jconf-lab4:/docs sphinx-jconf:v1 make html`

Git - 05

- `$ git checkout sphinx-documentacion-texto-codigo-validacion-extension-custom-maven`

Documentación Validación Código con Extension Custom Maven

- La nueva extension customizada javadoctest.py también ofrece soporte para proyectos que manejan dependencias de tercero via un gestor de dependencias por ejemplo maven.
- En el siguiente ejemplo vamos a ver como cargar una extension que nos permita documentar y validar nuestro código Java usando Sphinx usando un gestor de dependencias Maven.

Documentación Validación Código con Extension Custom Maven



conf.py

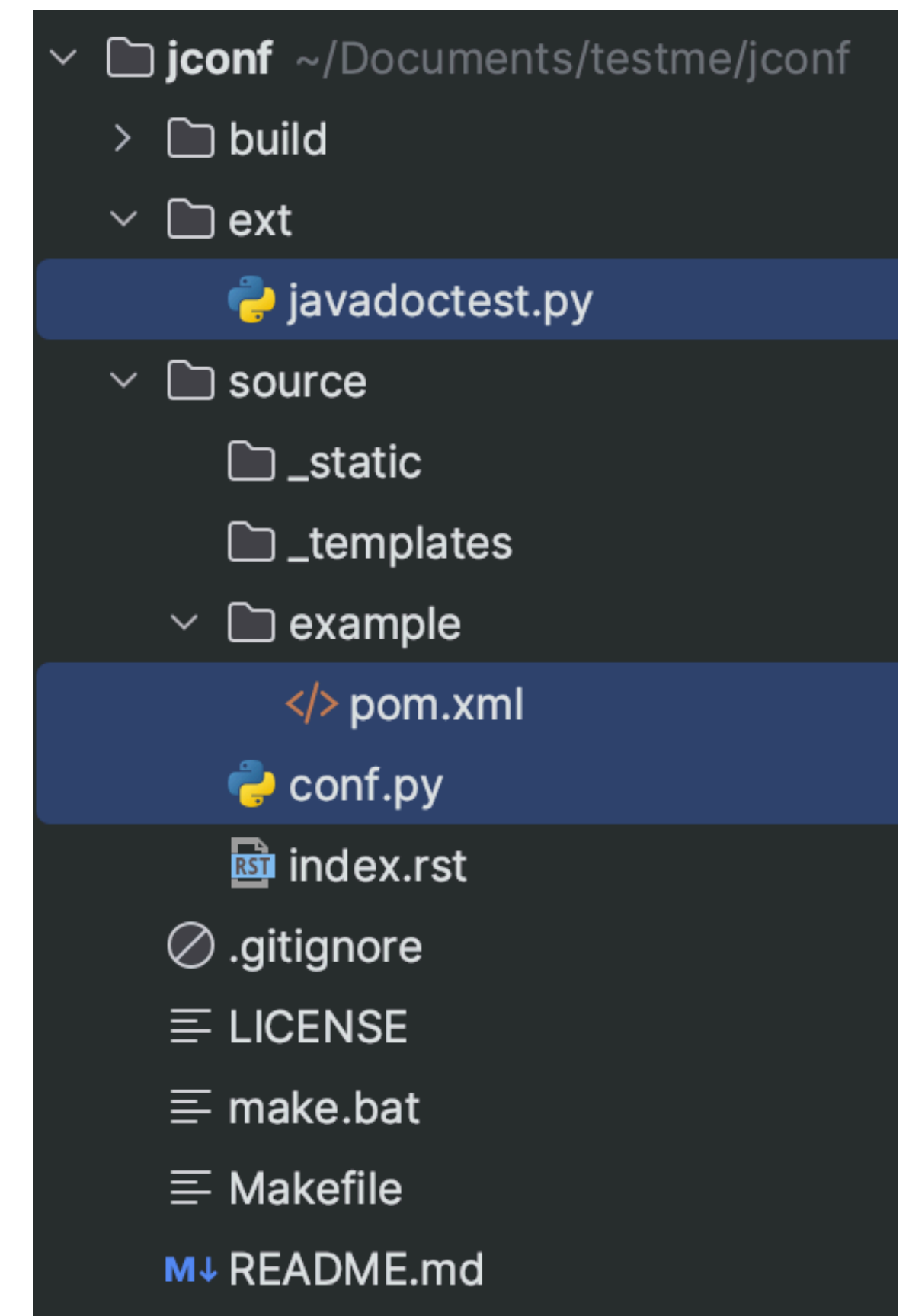
```
import os
import sys
import pathlib

sys.path.append(
    os.path.abspath("../ext"))
extensions = [
    "javadoctest"
]
javadoctest_config = {
    'flavor': 'java_with_maven',
    'path': pathlib.Path(__file__).parent / 'example',
}
```

ext/javadoctest.py

```
app.add_directive('javatestsetup', JavaTestsetupDirective)
app.add_directive('javatestcleanup', JavaTestcleanupDirective)
app.add_directive('javadoctest', JavaDoctestDirective)
app.add_directive('javatestcode', JavaTestcodeDirective)
app.add_directive('javatestoutput', JavaTestoutputDirective)
```

(Nueva Directivas Para Java)



Resultado Validación

\$ make javadoctest

```
Doctest summary
=====
1 test
0 failures in tests
0 failures in setup code
0 failures in cleanup code
build succeeded.
```

\$ make html

Java Guava Example

```
import java.util.List;
import com.google.common.collect.Lists;
import com.google.common.primitives.Ints;

List<Integer> theList = Ints.asList(1, 2, 3, 4, 5);
List<Integer> countDown = Lists.reverse(theList);
System.out.println(countDown);
```

[5, 4, 3, 2, 1]

Lab - 5 - Documentación **Validación Código con** **Extension Custom Maven**

- Docker
 - \$ mkdir jconf-lab5
 - \$ cd jconf-lab5
 - \$ cat Dockerfile

```
# Sphinx + Java/Maven Support
```

```
FROM sphinxdoc/sphinx
```

```
WORKDIR /docs
```

```
RUN apt-get update && apt-get -y install maven openjdk-17-jdk
```


Lab - 5 - Documentación **Validación Código con Extension Custom**

- Docker (cont)
 - `$ docker build . -t sphinx-jconf:v1`
 - Agregar código al documento (usando la directiva: `javatestcode`): `./jconf-lab5/source/index.rst`
 - Crear el archivo `./jconf-lab5/ext/javadoctest.py` con las directivas creadas en <https://gist.github.com/davisusanibar/1ad47ec8a12d91e311558ab01f912f26>
 - Agregar la nueva extension al archivo de configuración `./jconf-lab5/source/conf.py`: `javadoctest`
 - Definir que se usara un proyecto Maven y especificarle la ruta del proyecto `example/pom.xml`:

```
javadoctest_config = {  
    'flavor': 'java_with_maven',  
    'path': pathlib.Path(__file__).parent / 'example',  
}
```
 - `$ cd ..`
 - `$ docker run -it --rm -v ./jconf-lab5:/docs sphinx-jconf:v1 make javadoctest`
 - `$ docker run -it --rm -v ./jconf-lab5:/docs sphinx-jconf:v1 make html`

Git - 05

- `$ git checkout sphinx-java-options`

Git - 05

- Ejemplos disponibles en: <https://raw.githubusercontent.com/davisusanibar/jconf-sphinx/sphinx-java-options/source/index.rst>

Dudas y/o Preguntas

Referencias

- Sphinx project - <https://github.com/sphinx-doc/sphinx>
- Como añadir tu propia extension de Sphinx - https://sphinx-intro-tutorial.readthedocs.io/en/latest/sphinx_extensions.html
- Nuevas Directivas Java Sphinx - <https://gist.github.com/davisusanibar/1ad47ec8a12d91e311558ab01f912f26>
- Ejemplos Java usando la nueva Directiva - <https://github.com/sphinx-doc/sphinx/blob/1a3af3ca4d2e509039bf164f15d98f0f4992f60b/tests/roots/test-ext-javadoctest/javadoctest.rst>
- Sphinx PR relacionado para agregar nueva Java directiva al Core de Sphinx - <https://github.com/sphinx-doc/sphinx/pull/11305>
- GH Project Demo - <https://github.com/davisusanibar/jconf-sphinx> (checkout a ramas de acuerdo al nivel a revisar)