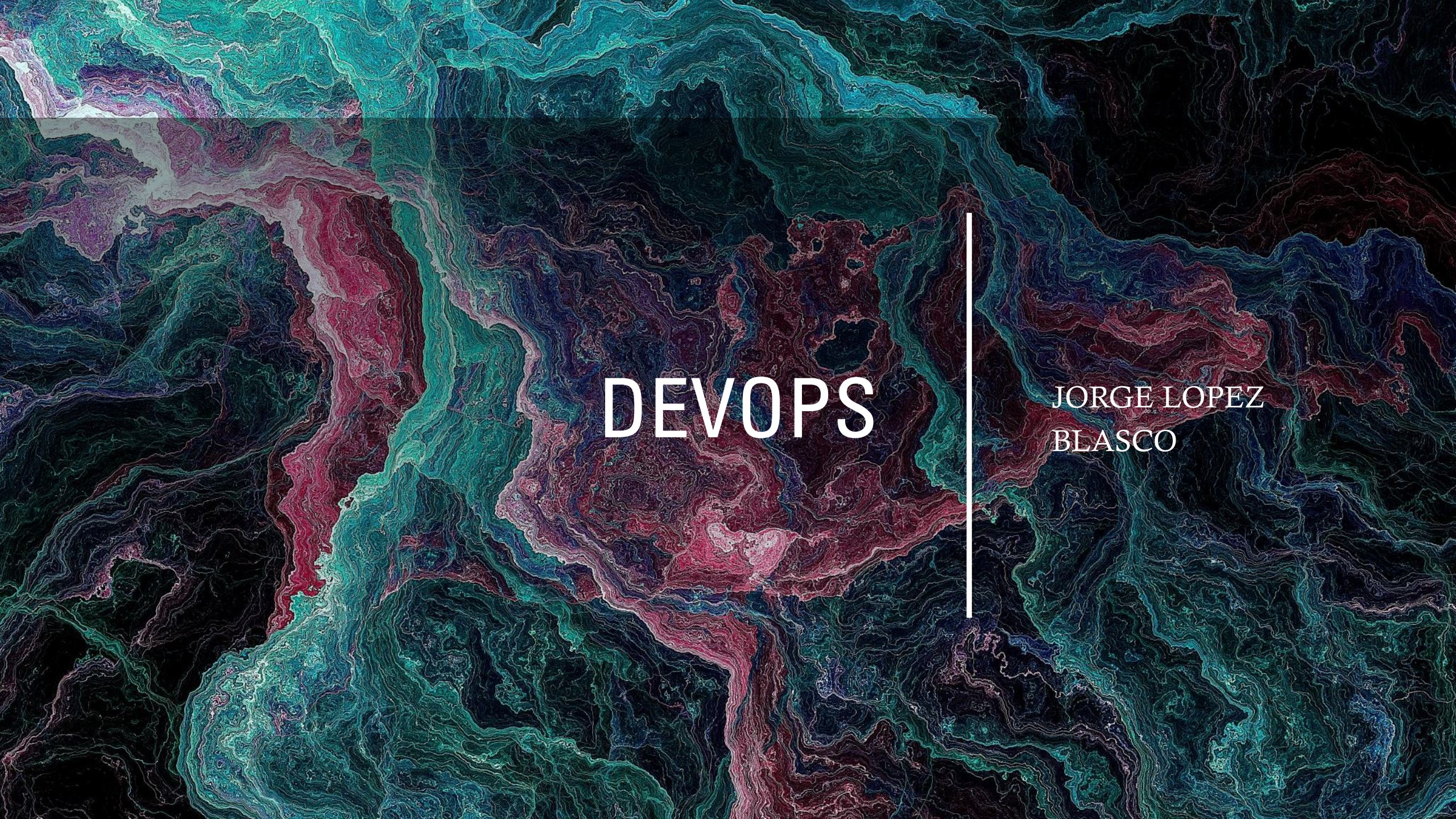
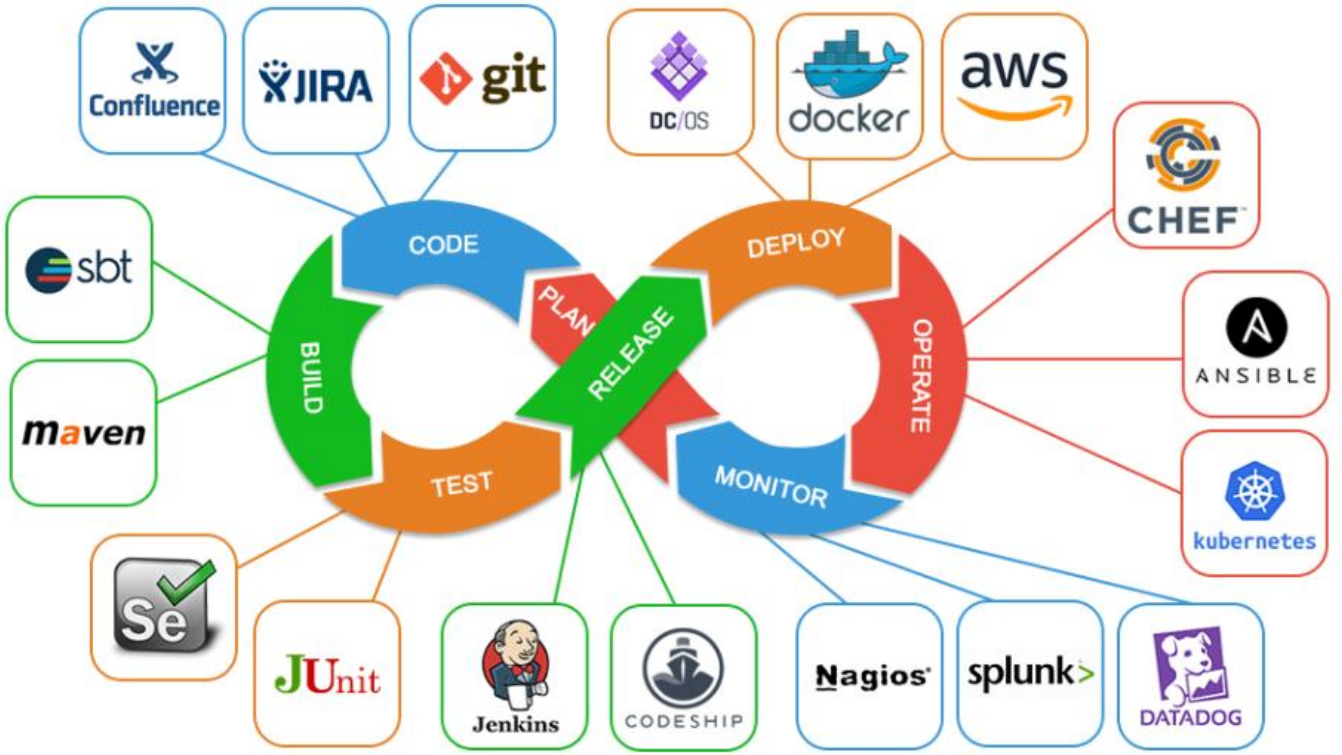

DEVOPS
JORGE LOPEZ
BLASCO

INTRODUCCIÓN A TESTING

- Introducción
- Conceptos
- Objetivos
- Principios
- Herramientas
- Versionado
- Calidad del Software

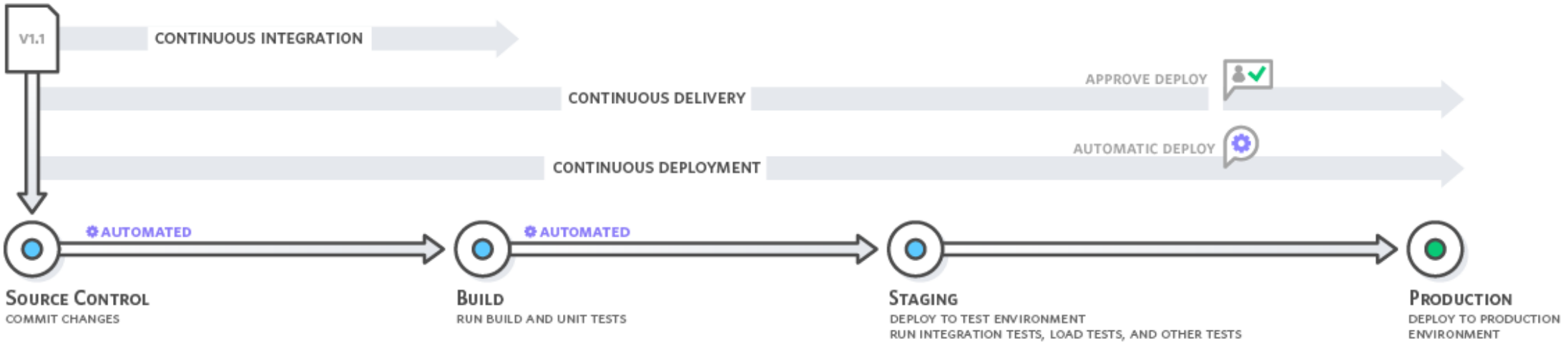
INTRODUCCIÓN



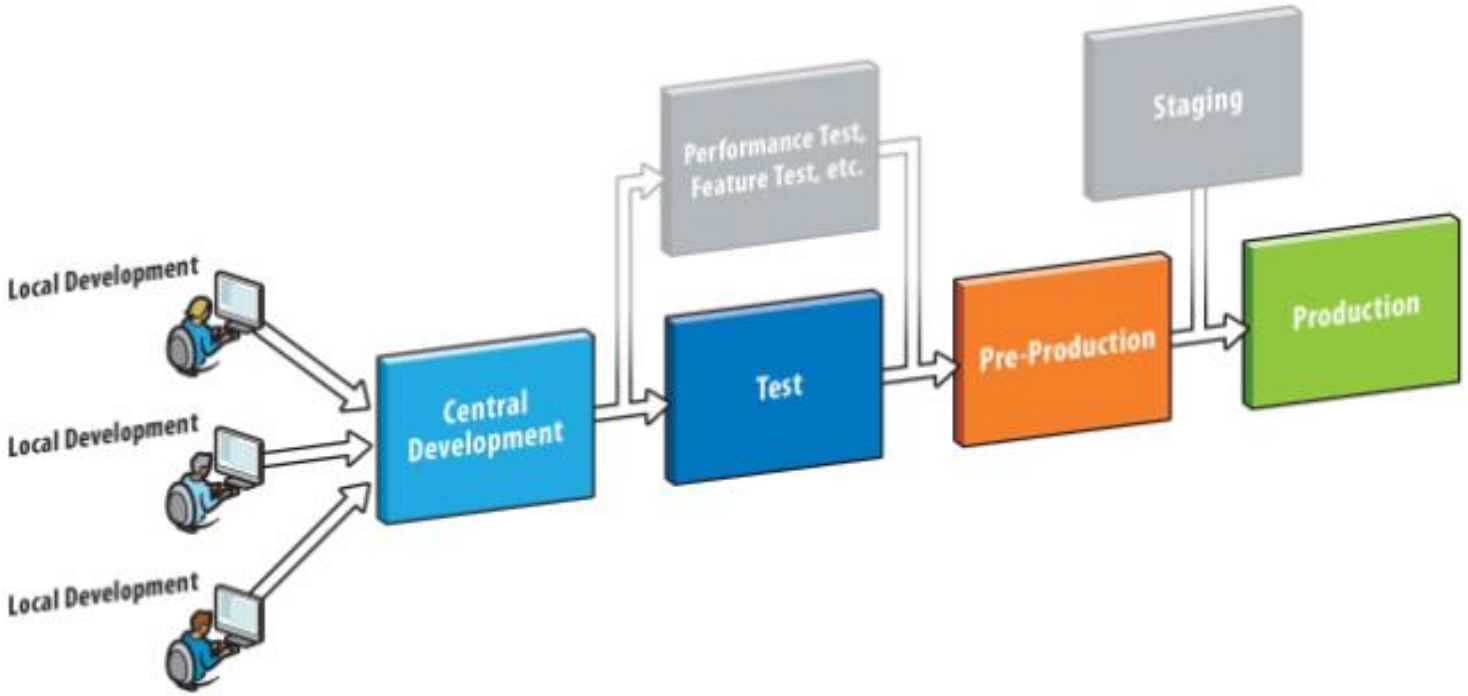
VENTAJAS DEVOPS

- Una mejor y más rápida entrega de productos
- Resolución de problemas en menos tiempo y con menor complejidad
- Mejor escalabilidad y disponibilidad
- Entornos de funcionamiento más estables
- Mejor utilización de los recursos
- Mayor automatización
- Mayor visibilidad de resultados del sistema
- Mayor innovación

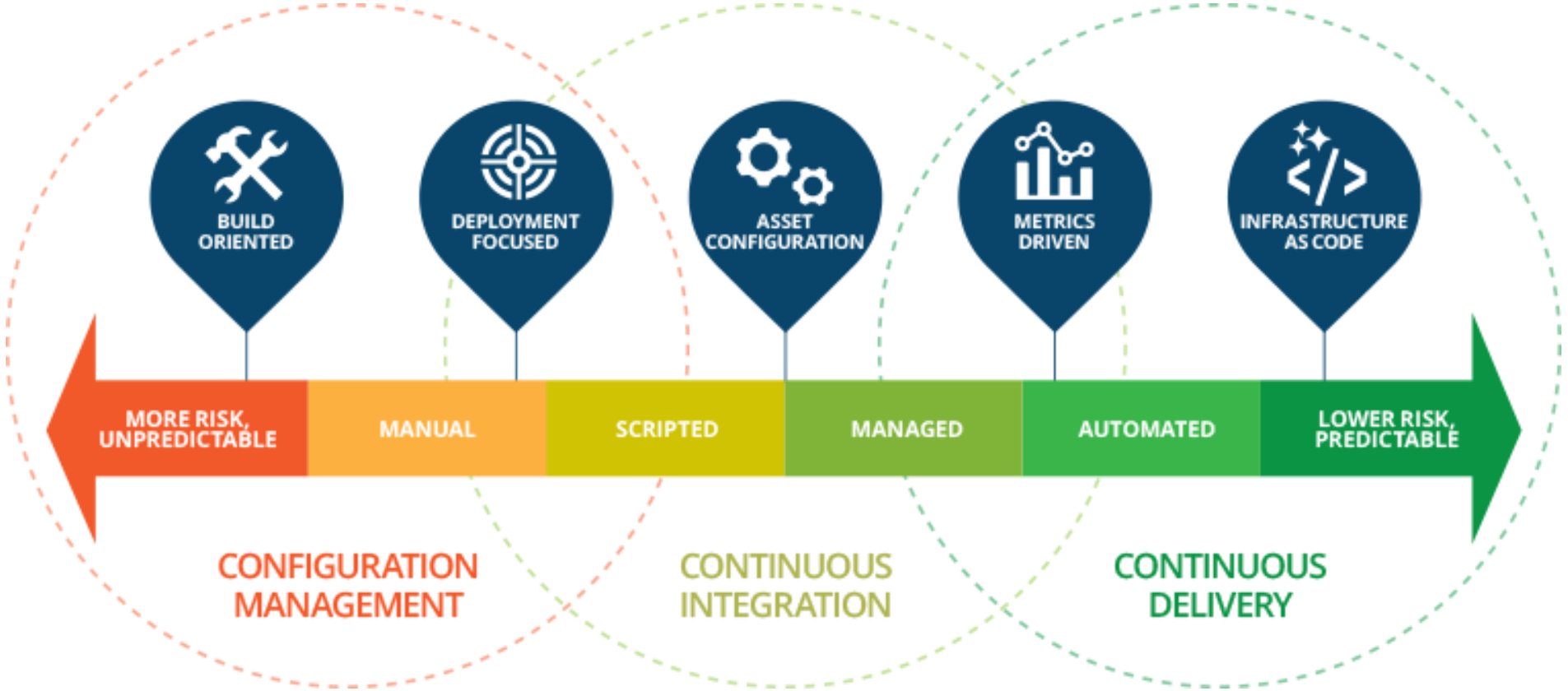
CONCEPTOS: INTEGRACIÓN CONTINUA



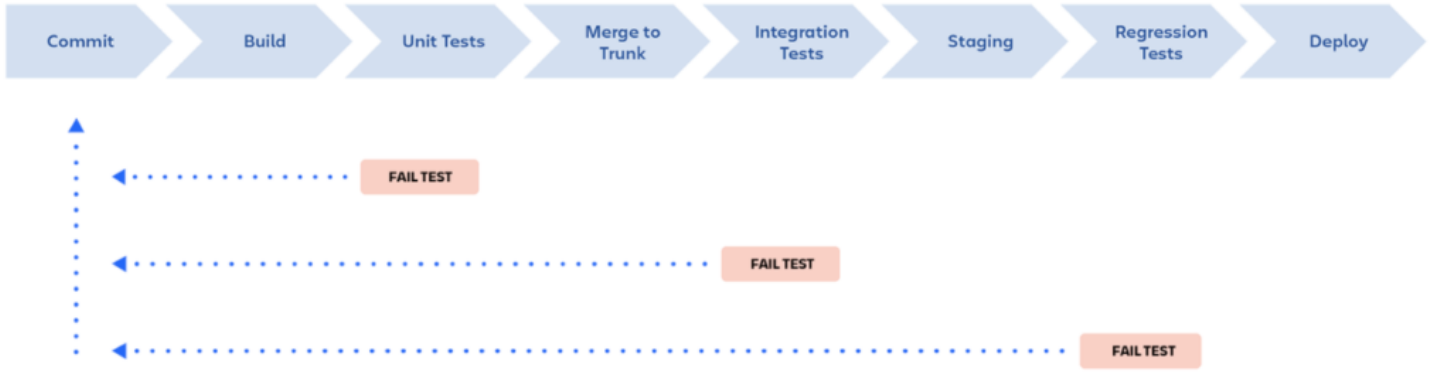
CONCEPTOS: ENTORNOS



CONCEPTOS: ENTREGA CONTINUA

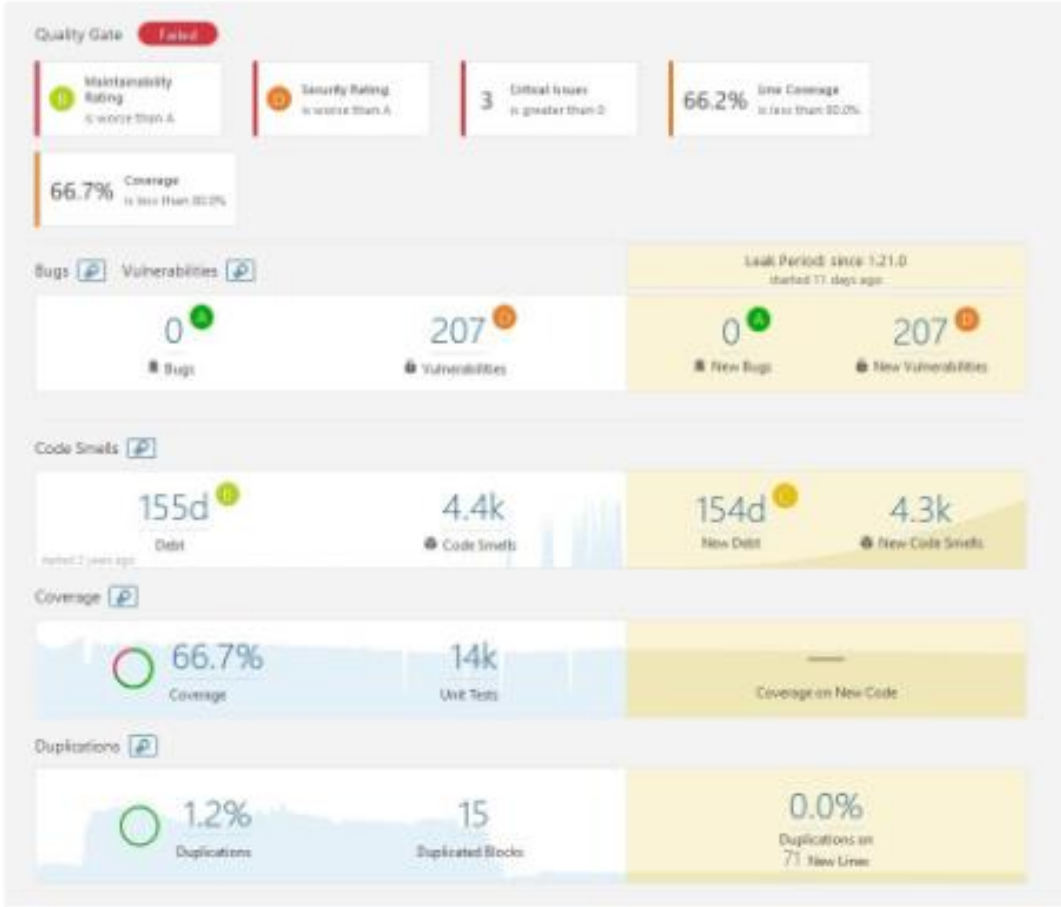


CONCEPTOS: PIPELINE



HERRAMIENTAS

CALIDAD DE SOFTWARE

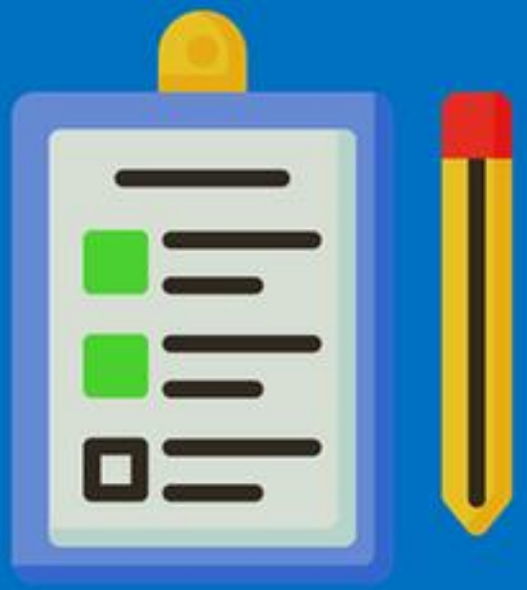


PRUEBAS

- Tipos de pruebas
- Pruebas de unidad
- Pruebas de integración
- Pruebas de aceptación
- Pruebas de rendimiento

INTRODUCCIÓN A PRUEBAS

vs.

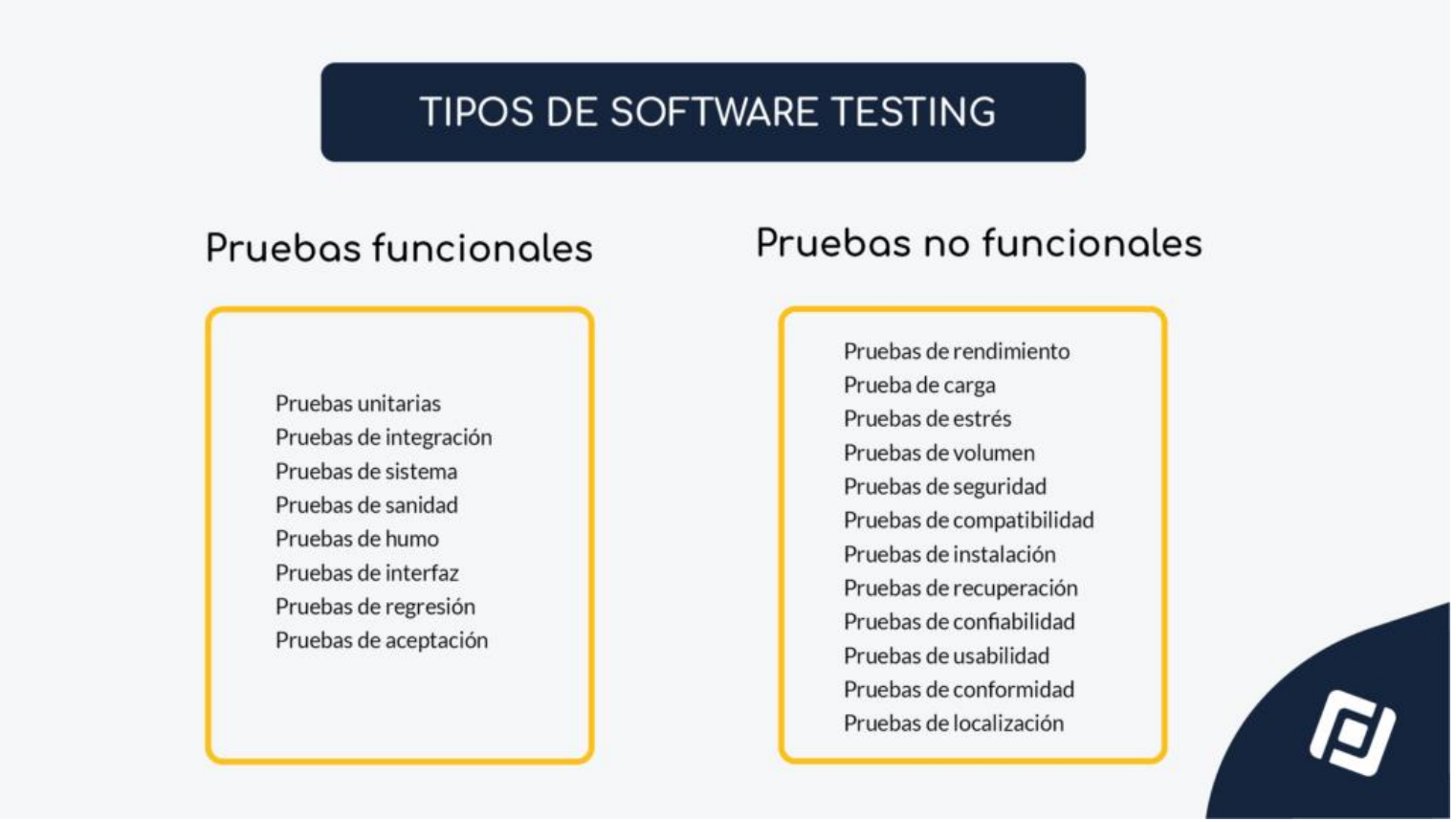


Pruebas dinámicas

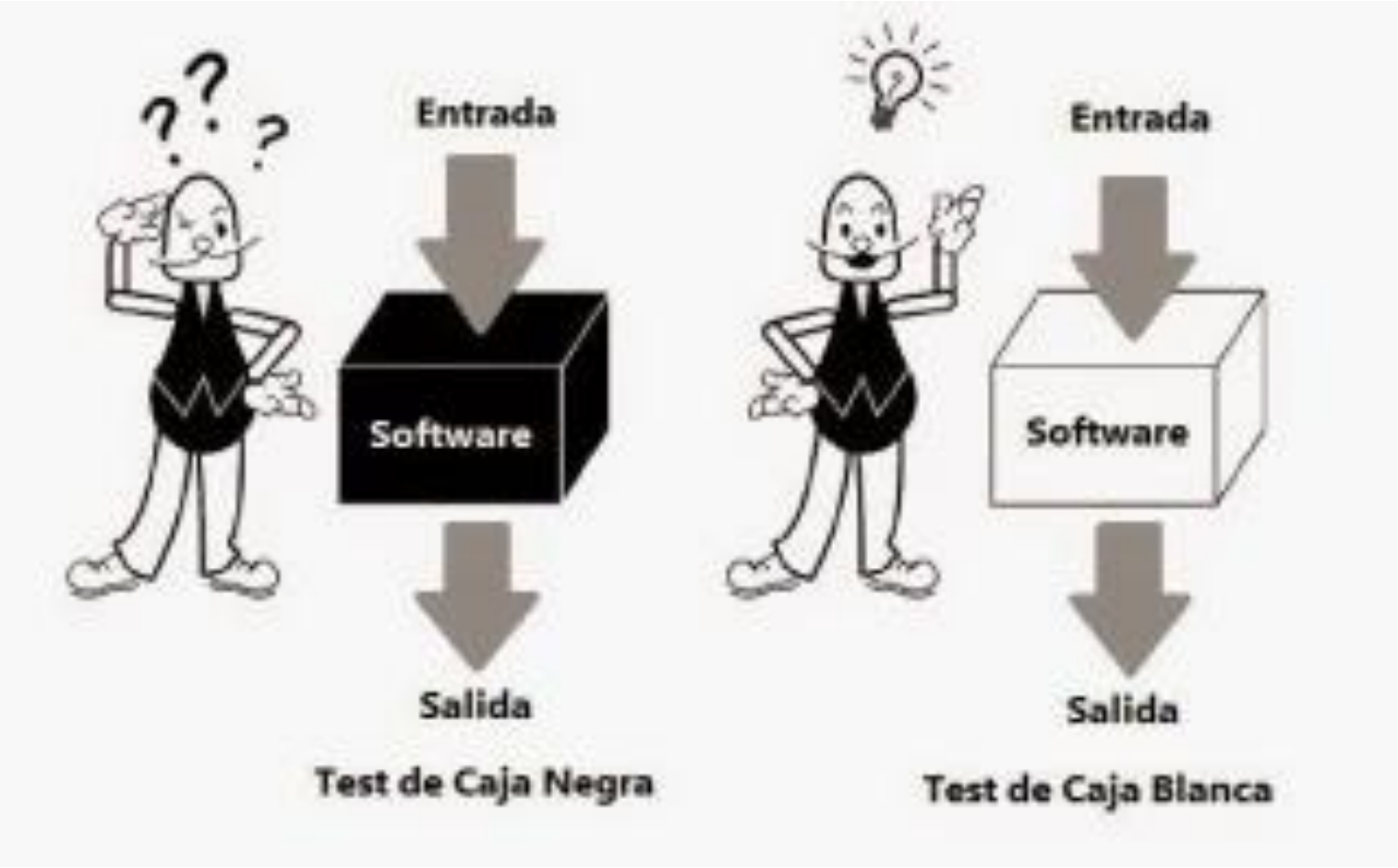
Pruebas estáticas

TIPOS DE PRUEBAS

TIPOS DE PRUEBAS



TIPOS DE PRUEBAS



ESTRATEGIA DE PRUEBAS

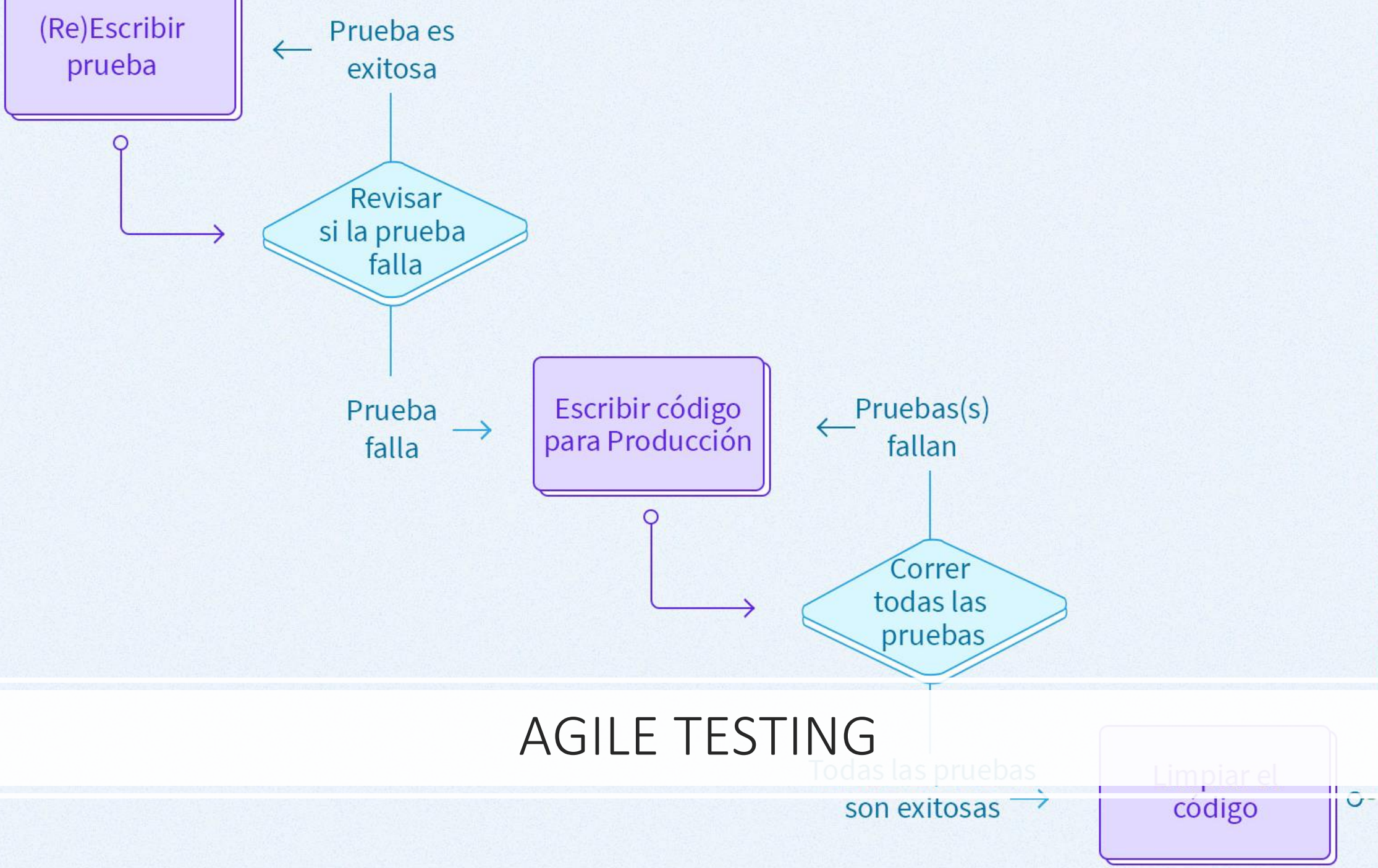
- Requisitos de análisis: recogida de requisitos iniciales
- Planificación de Pruebas
- Desarrollo de Pruebas
- Ejecución de Pruebas
- Informes de Pruebas
- Análisis de los Informes de Pruebas
- Reprueba de Fallos: Cuando se detecta un fallo se prueba esa funcionalidad
- Test de Regresión: paso de pruebas que puedan afectar a la corrección
- Cierre de las Pruebas: Finalización del proyecto

DISEÑO DE PRUEBAS

- Conocimiento de los Requisitos
- Reducción de la funcionalidad a probar
- Definición de Casos de Prueba
- Relacionar con los Casos de Usuario (historias)
- Pensar bien los costes de las pruebas
- Enfocarse en las funcionalidades más críticas
- Cada prueba requiere su herramienta

HERRAMIENTAS

- Pruebas Unitarias: Junit
- Pruebas de Integración: Mockito
- Pruebas de Aceptación/Funcionalidad Web: Selenium
- Bases de Datos/Web Rendimiento: Jmeter/Blazemeter/Gatling
- Servidor de Integración Contínua: Jenkins/TravisCI/CircleCI
- Herramienta de Seguimiento de Proyectos e Incidencias: Jira/Mantis
- Automatizador de la Compilación, Despliegue y Ejecución de proyectos: Maven/Gradle/NPM
- Despliegue de aplicaciones: Docker/Kubernetes



AGILE TESTING

AUTOMATIZACIÓN DE LAS PRUEBAS

LABORATORIO 1: PRUEBAS UNITARIAS EN PYTHON

EJERCICIO 1

DEMO: PRUEBAS DE INTEGRACIÓN EN PYTHON

EJERCICIO 2

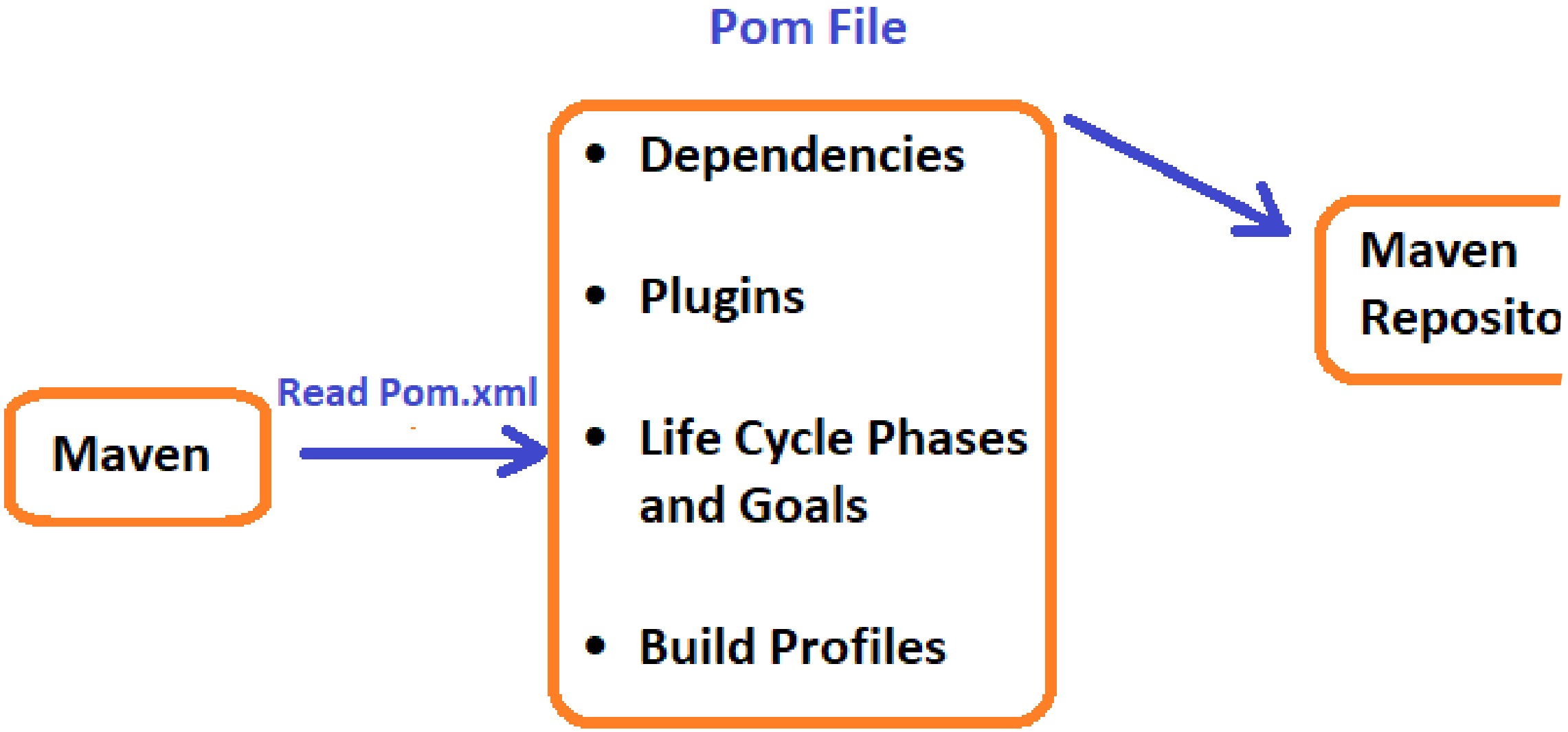
LABORATORIO 2: PRUEBAS DE RENDIMIENTO EN PYTHON

EJERCICIO 3

MAVEN

- Introducción
- Estructura
- Fichero pom.xml
- Compilación del proyecto
- Dependencias
- Repositorios
- Objetivos
- Plugins
- Despliegue
- Integración con IDE’s


Maven™


Pom File
Dependencies
Plugins
Life Cycle Phases and Goals
Build Profiles
Maven
Read Pom.xml
Maven
Reposito

LABORATORIO 1: INSTALAR MAVEN

```
|-- pom.xml : fichero de configuración de maven
-- src : carpeta principal de código
  |-- main: código de la aplicación
    | `-- java : código Java
      | `-- com
        | `-- mycompany
          | `-- app : Paquete principal
            | `-- App.java : Código de la aplicación
  `-- test : Código de pruebas
    `-- java : Código Java
      `-- com
        `-- mycompany
          `-- app : Paquete principal
```

Creación de un proyecto Mave

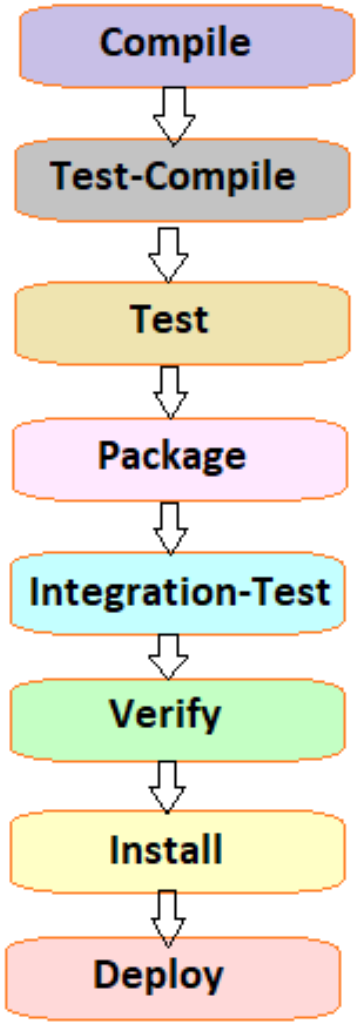
```
.B archetype:generate \
:hetypeGroupId=org.apache.maven.arch
upId=com.mycompany.app \
ifactId=my-app
```

ARQUETIPOS

POM.XML

```
<?xml version="1.0"?>
- <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://maven.apache.org/POM/4.0.0">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany.app</groupId>
    <artifactId>my-app</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>my-app</name>
    <url>http://maven.apache.org</url>
  - <dependencies>
      - <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

CICLO DE VIDA MAVEN

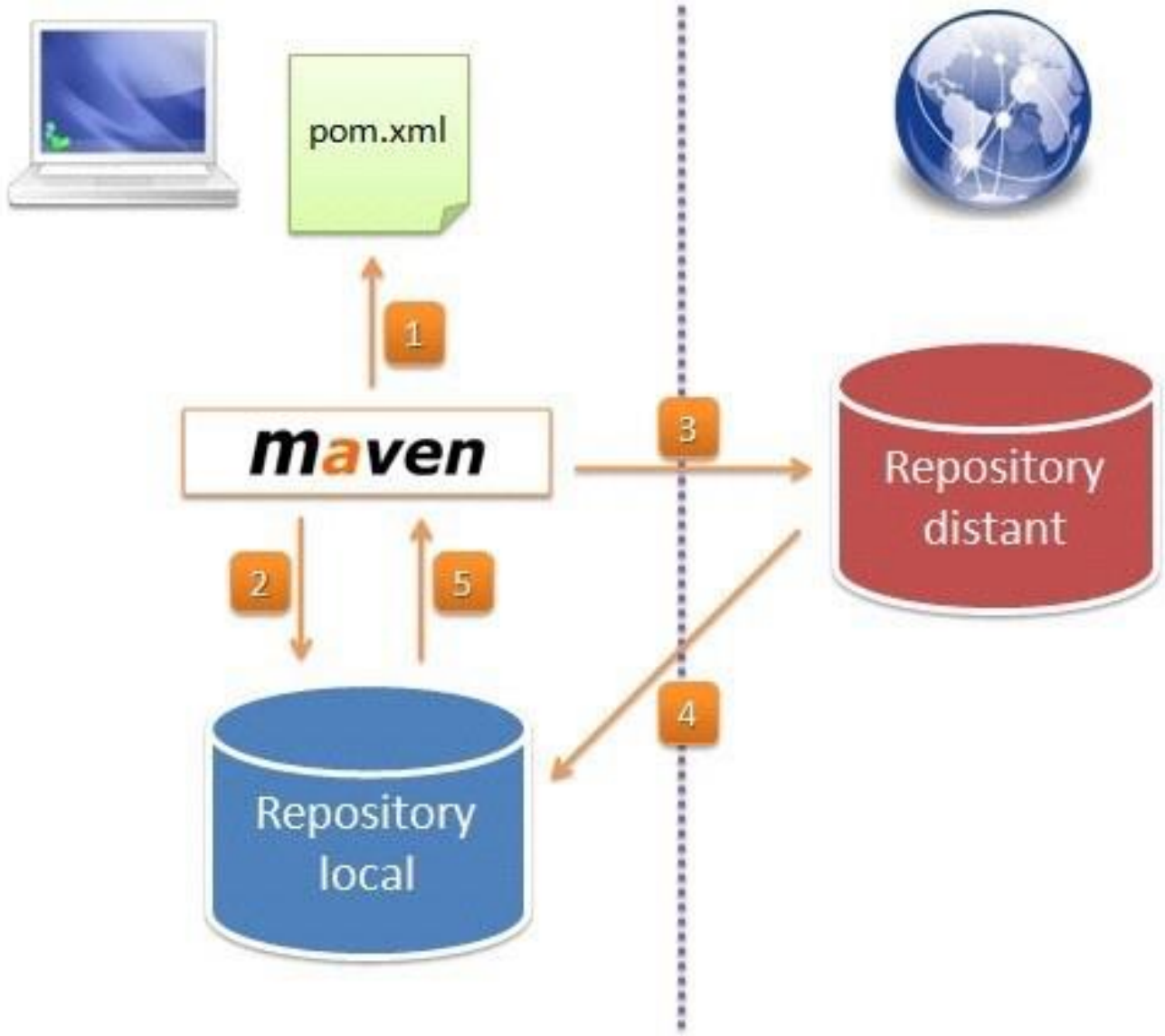


DEPENDENCIAS

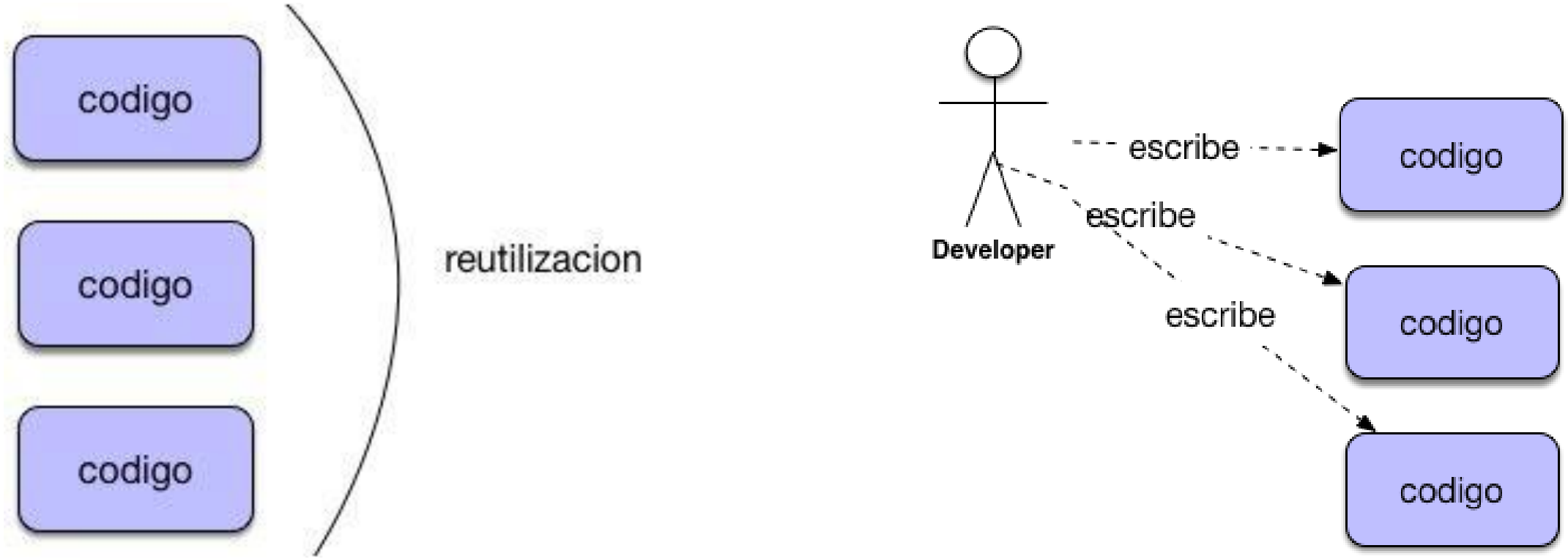
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/
4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
<...>
<dependencies>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>3.8.1</version>
<scope>test</scope>
</dependency>
</dependencies>
</project>
```

LABORATORIO 2

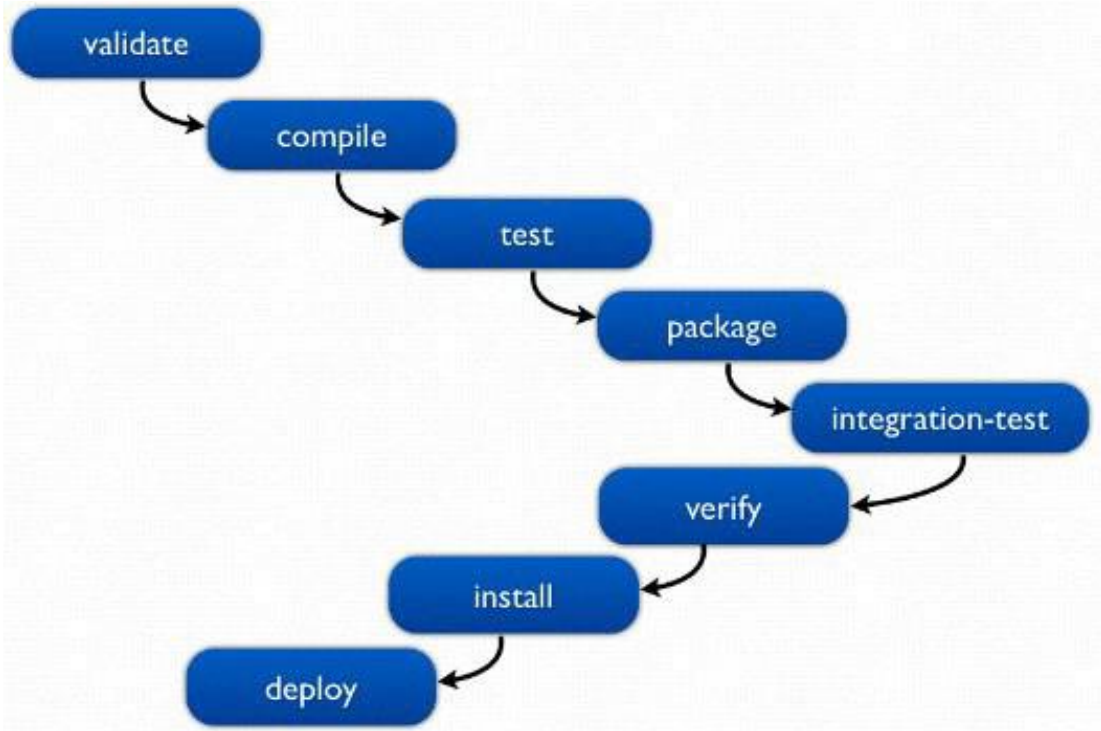
REPOSITORIOS



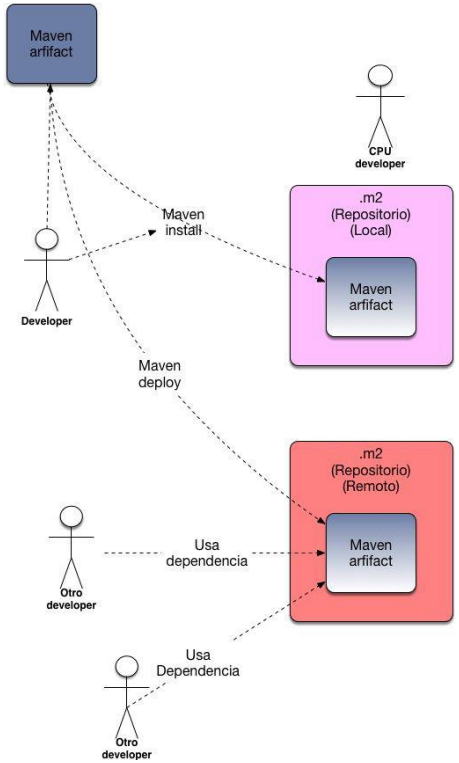
ARTEFACTOS

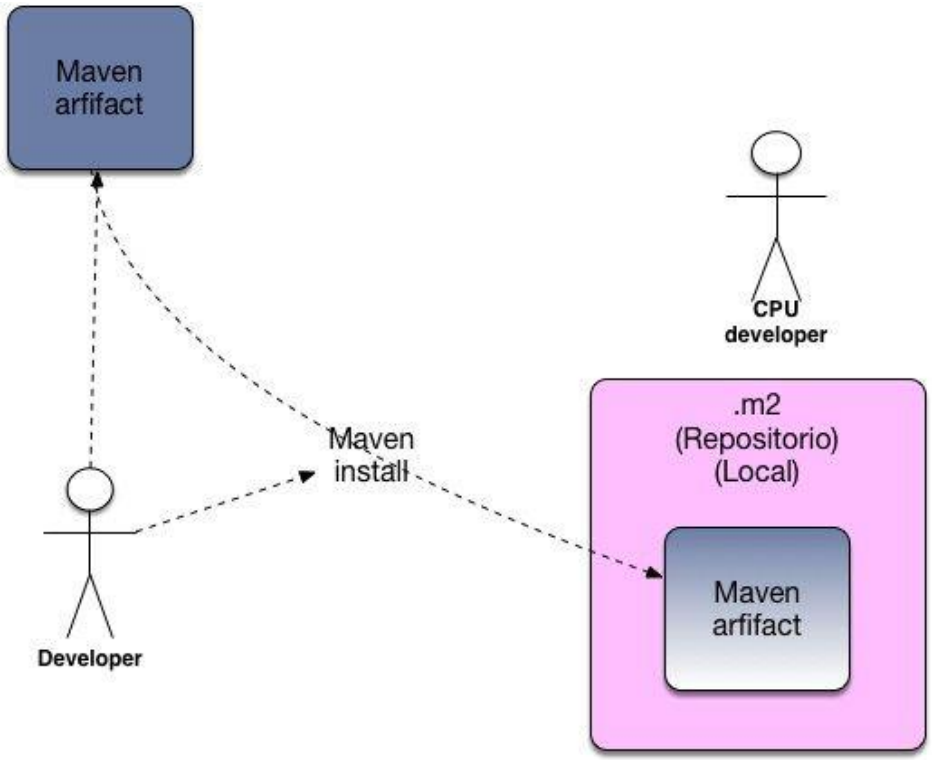

codigo
codigo
codigo
reutilizacion
Developer
escribe
escribe
escribe
codigo
codigo
codigo

OBJETIVOS


validate
compile
test
package
integration-test
verify
install
deploy

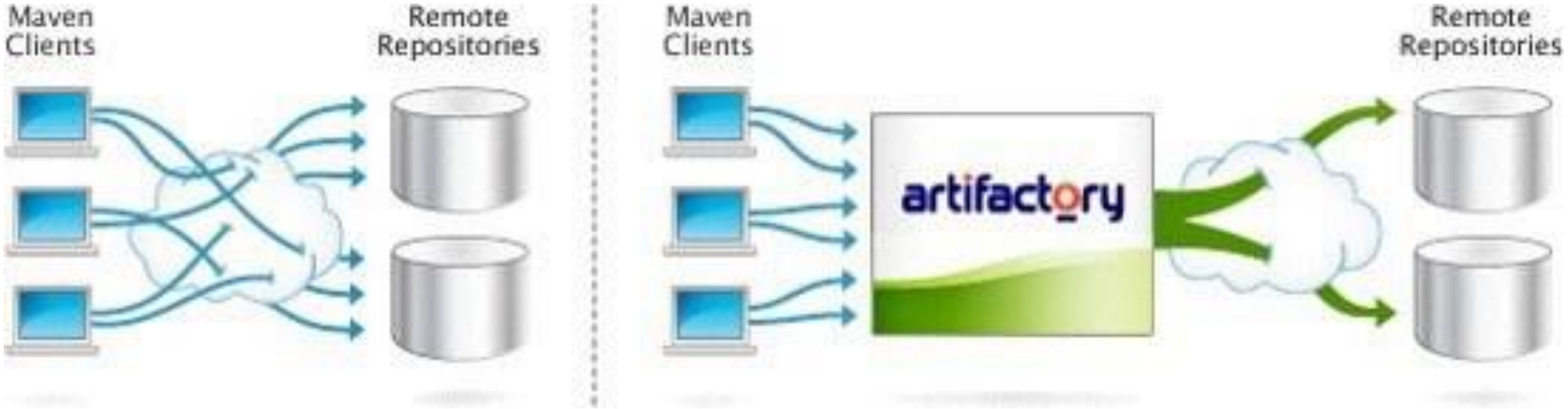
INSTALL Y DEPLOY





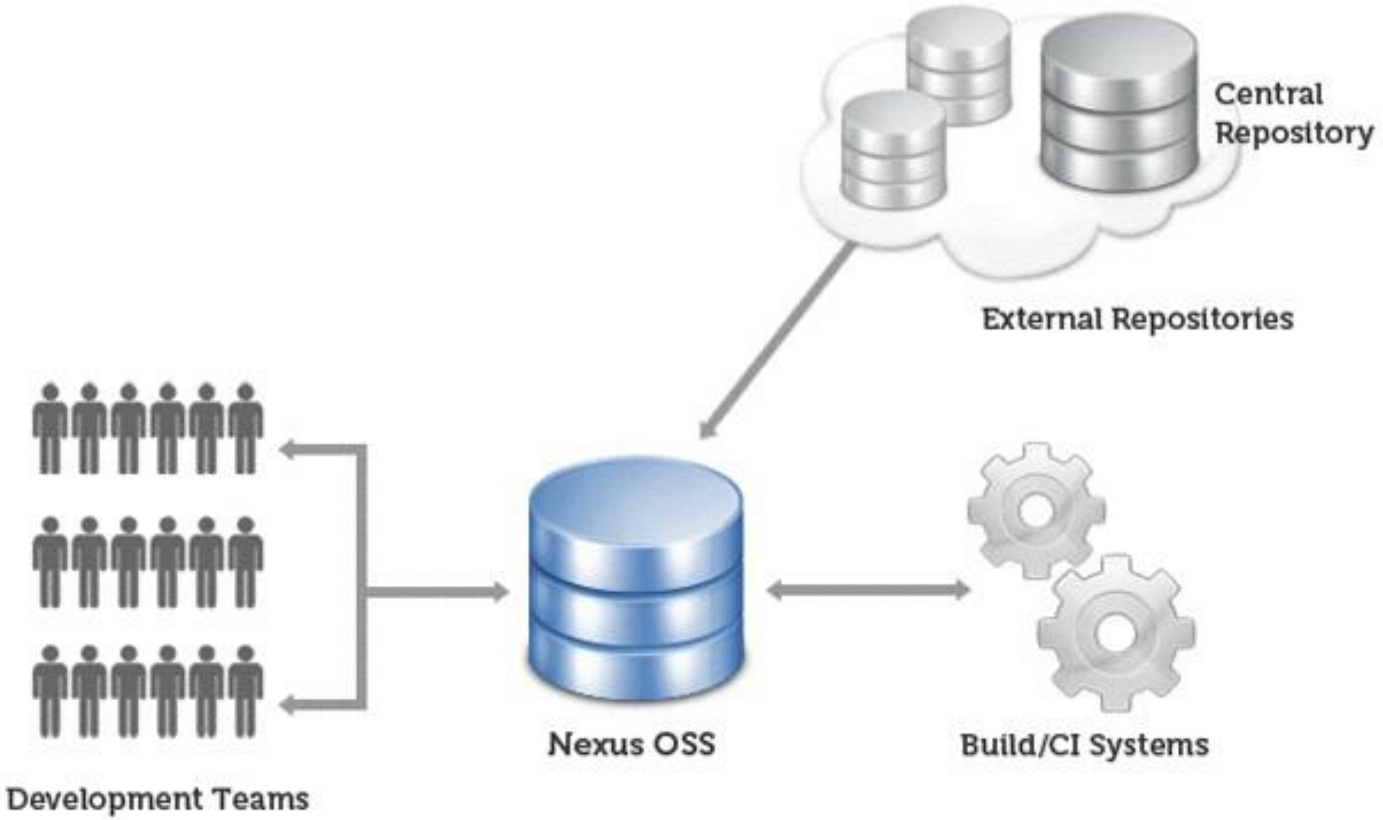
LABORATORIO 3

ARTIFACTORY



LABORATORIO 4

NEXUS

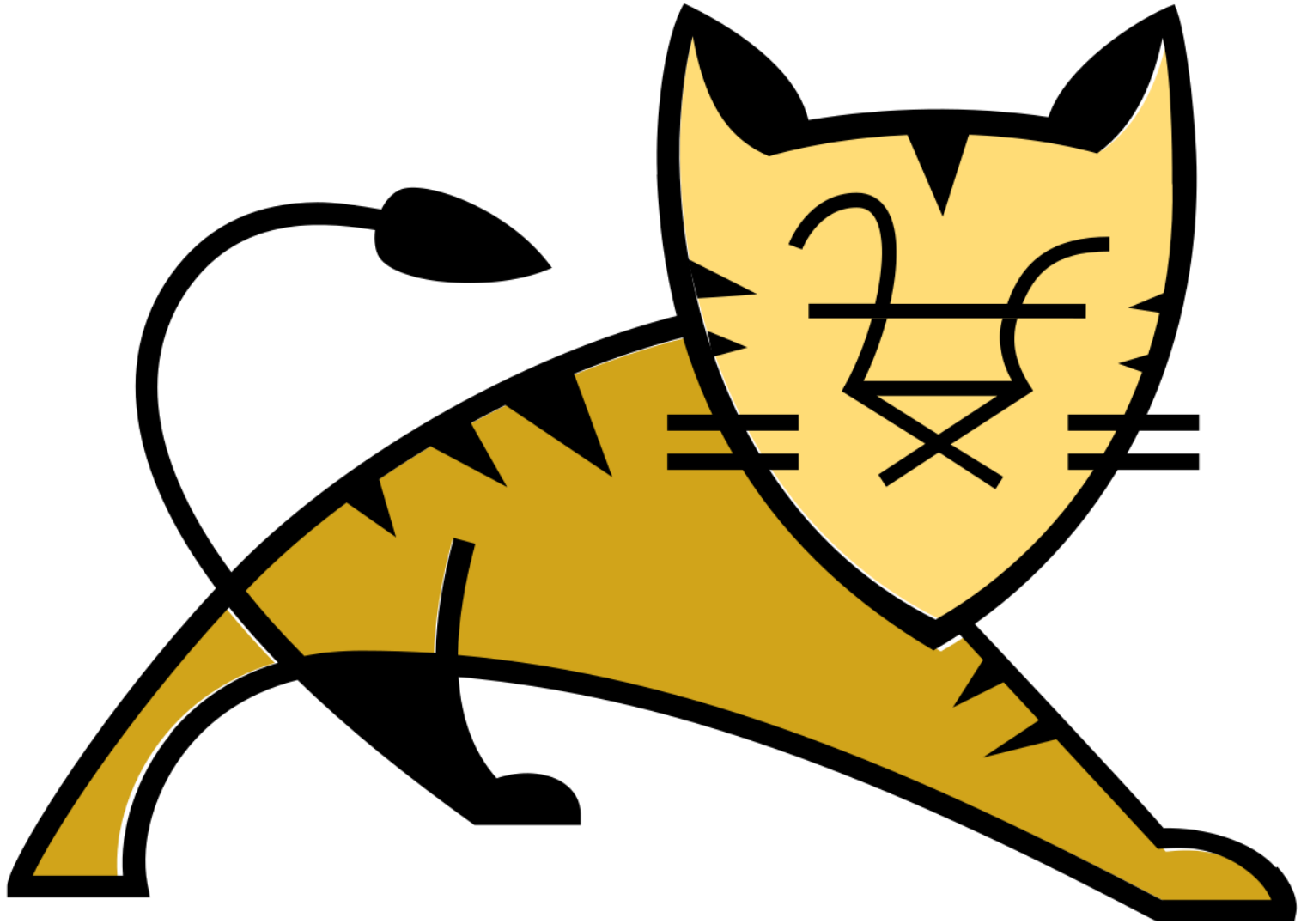

Central
Repository
External Repositories
Development Teams
Nexus OSS
Build/CI Systems

LABORATORIO 5

PLUGINS

```shell
<project>
    ...
    <build>
        ...
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <source>1.5</source>
                    <target>1.5</target>
                </configuration>
            </plugin>
        </plugins>
        ...
    </build>
    ...
</project>
```

TOMCAT

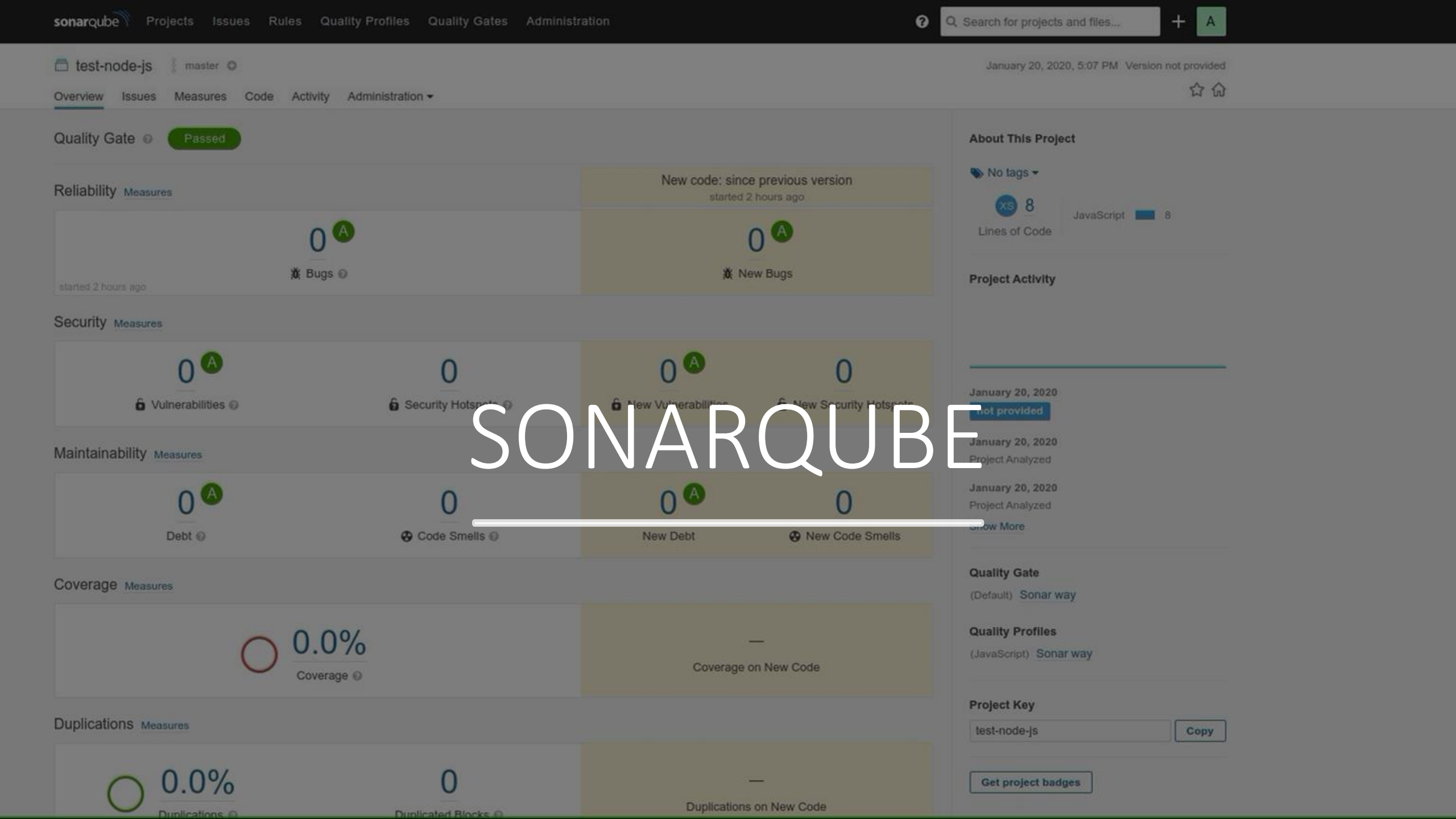


LABORATORIO 6

SECURITY TESTING

- Introducción
- SonarQube
- Reglas
- Cobertura de código
- Smells
- Bugs
- Fortify
- Análisis
- Reporting

INTRODUCCIÓN


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects and files...
A
test-node-js
master
January 20, 2020, 5:07 PM Version not provided
Overview
Issues
Measures
Code
Activity
Administration
Quality Gate
Passed
Reliability Measures
New code: since previous version
started 2 hours ago
0 A
Bugs
started 2 hours ago
0 A
New Bugs
Security Measures
0 A
Vulnerabilities
0
Security Hotspots
0 A
New Vulnerabilities
0
New Security Hotspots
Maintainability Measures
0 A
Debt
0
Code Smells
0 A
New Debt
0
New Code Smells
Coverage Measures
0.0%
Coverage
—
Coverage on New Code
Duplications Measures
0.0%
Duplications
0
Duplicated Blocks
—
Duplications on New Code
About This Project
No tags
XS 8
Lines of Code
JavaScript 8
Project Activity
January 20, 2020
Not provided
January 20, 2020
Project Analyzed
January 20, 2020
Project Analyzed
Show More
Quality Gate
(Default) Sonar way
Quality Profiles
(JavaScript) Sonar way
Project Key
test-node-js
Copy
Get project badges

SONARQUBE











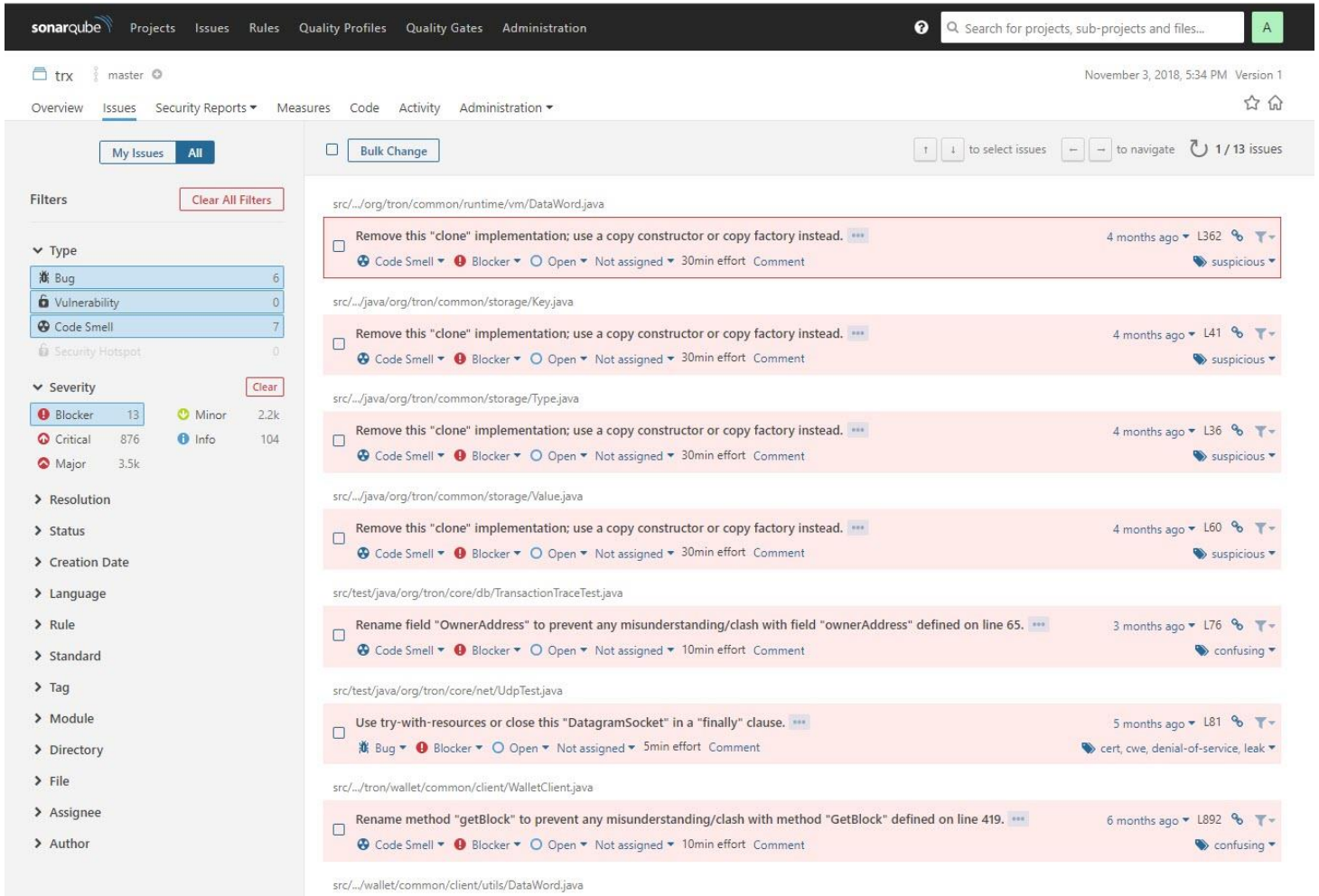

LENGUAJES

LABORATORIO 1

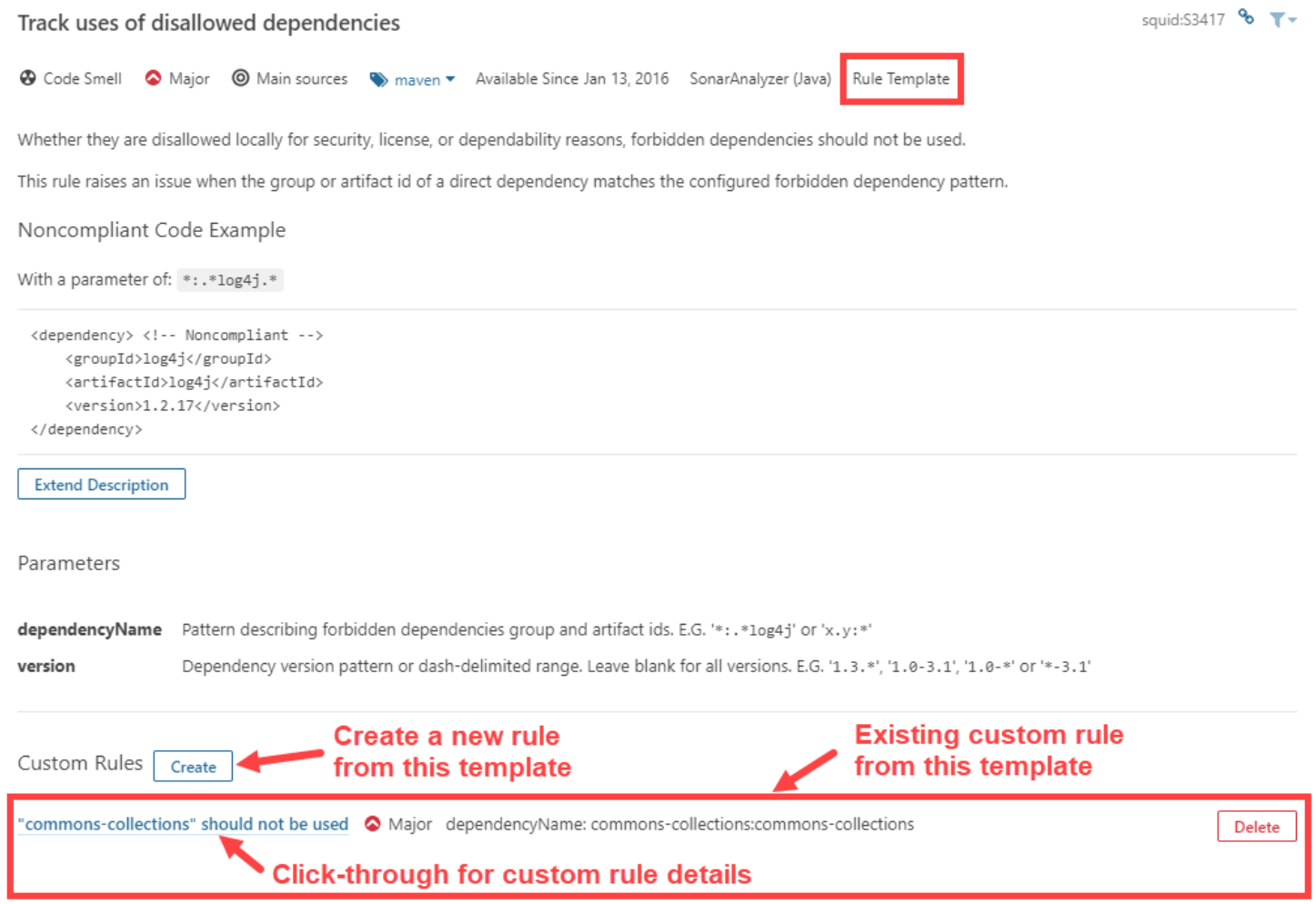
REGLAS

- SonarQube ejecuta reglas sobre el código fuente para generar problemas. Hay cuatro tipos de reglas:
- Smells
- Bug
- Vulnerabilidad
- Punto de acceso de seguridad

ISSUE


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects, sub-projects and files...
A
trx
master
November 3, 2018, 5:34 PM Version 1
Overview
Issues
Security Reports
Measures
Code
Activity
Administration
My Issues
All
Bulk Change
to select issues
to navigate
1 / 13 issues
Filters
Clear All Filters
Type
Bug 6
Vulnerability 0
Code Smell 7
Security Hotspot 0
Severity
Clear
Blocker 13
Critical 876
Major 3.5k
Minor 2.2k
Info 104
Resolution
Status
Creation Date
Language
Rule
Standard
Tag
Module
Directory
File
Assignee
Author
src/.../org/tron/common/runtime/vm/DataWord.java
Remove this "clone" implementation; use a copy constructor or copy factory instead.
4 months ago L362
Code Smell Blocker Open Not assigned 30min effort Comment
suspicious
src/.../java/org/tron/common/storage/Key.java
Remove this "clone" implementation; use a copy constructor or copy factory instead.
4 months ago L41
Code Smell Blocker Open Not assigned 30min effort Comment
suspicious
src/.../java/org/tron/common/storage/Type.java
Remove this "clone" implementation; use a copy constructor or copy factory instead.
4 months ago L36
Code Smell Blocker Open Not assigned 30min effort Comment
suspicious
src/.../java/org/tron/common/storage/Value.java
Remove this "clone" implementation; use a copy constructor or copy factory instead.
4 months ago L60
Code Smell Blocker Open Not assigned 30min effort Comment
suspicious
src/test/java/org/tron/core/db/TransactionTraceTest.java
Rename field "OwnerAddress" to prevent any misunderstanding/clash with field "ownerAddress" defined on line 65.
3 months ago L76
Code Smell Blocker Open Not assigned 10min effort Comment
confusing
src/test/java/org/tron/core/net/UdpTest.java
Use try-with-resources or close this "DatagramSocket" in a "finally" clause.
5 months ago L81
Bug Blocker Open Not assigned 5min effort Comment
cert, cwe, denial-of-service, leak
src/.../tron/wallet/common/client/WalletClient.java
Rename method "getBlock" to prevent any misunderstanding/clash with method "GetBlock" defined on line 419.
6 months ago L892
Code Smell Blocker Open Not assigned 10min effort Comment
confusing
src/.../wallet/common/client/utils/DataWord.java

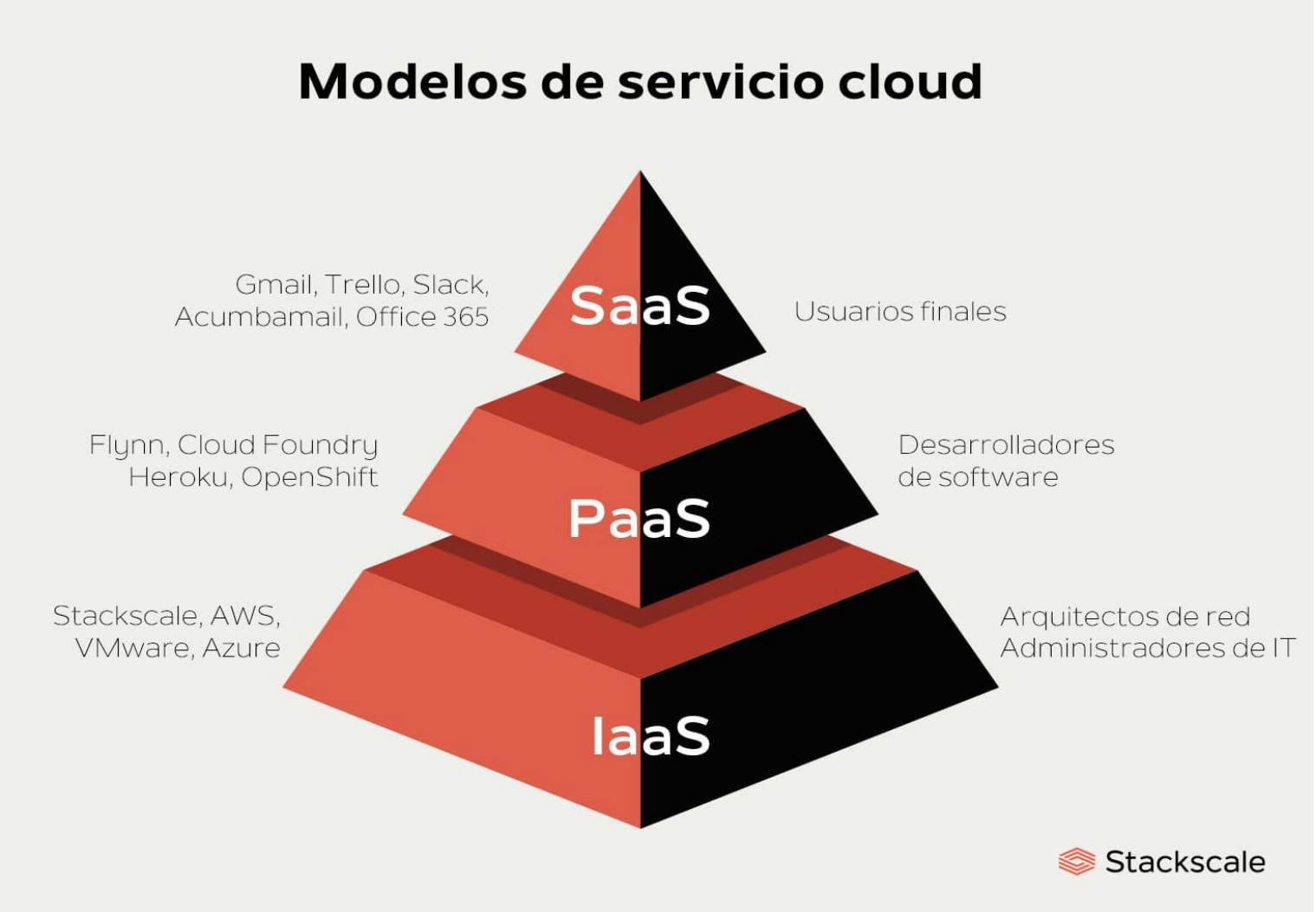
REGLAS PERSONALIZADAS



LABORATORIO 2

SONARCLOUD

MODELOS DE SERVICIO



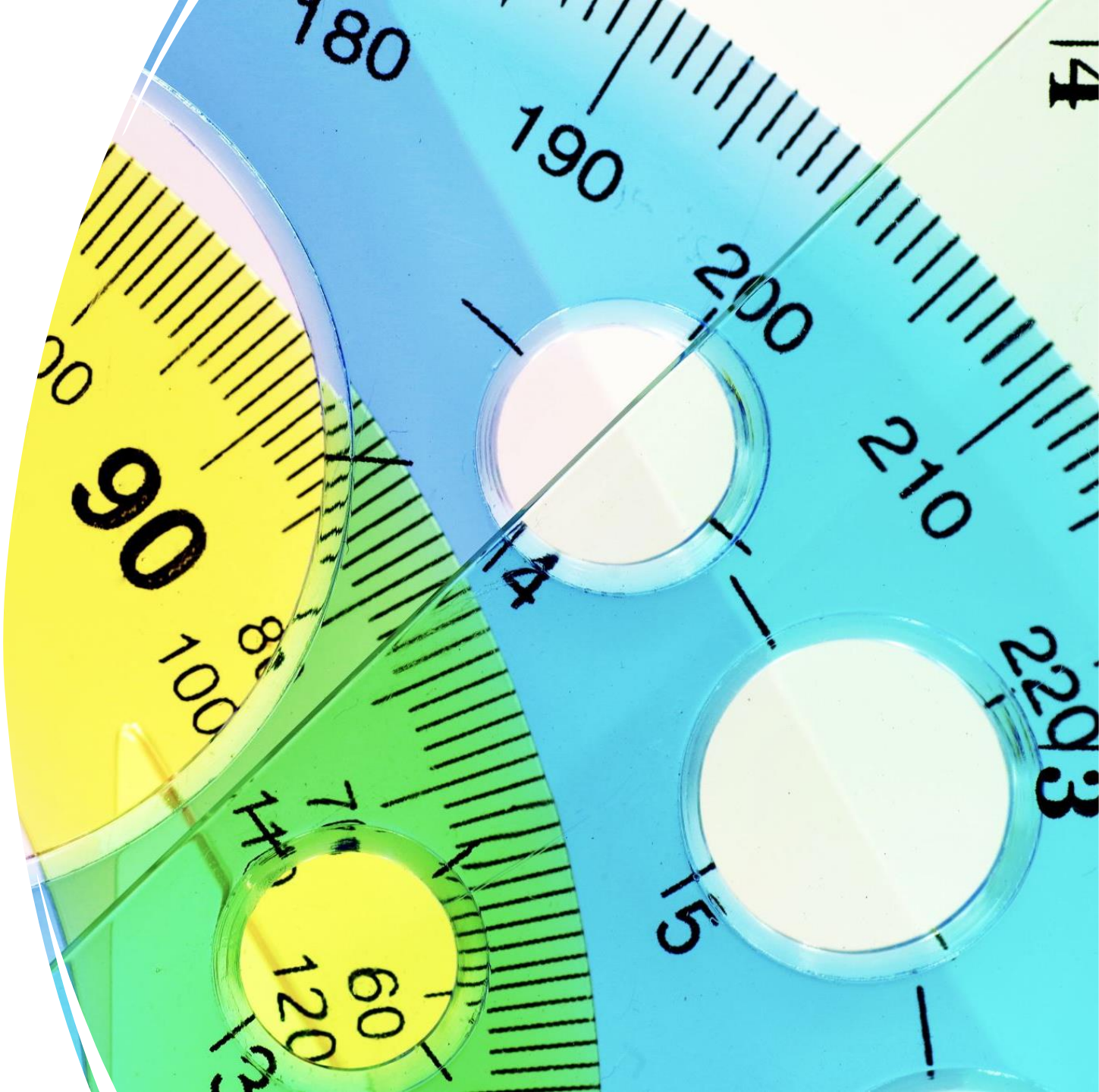
LABORATORIO 3

LABORATORIO 4

FORTIFY


CyberRes
Fortify
Interfaces & Integrations
Scanning Technologies
Enterprise AppSec Program Mgmt.
Supporting Offerings
Software Security Research

SONAR VS FORTIFY



LABORATORIO 5

EJERCICIO 4