

accenture

Cloud Platform Integration Governance Automation



Projeto de Informática - Mestrado em Engenharia Informática
Janeiro 2024

Contextualização

- Verificação dos artefatos criados na *Cloud Platform Integration* (CPI)
- Automatização da governança de fluxos de integração com a ferramenta CPILint
- Definição de um conjunto de regras de cobertura

Motivação

- Revisão automática de integrações deste gênero, não é novidade (e.g. Figaf)
- Soluções dispendiosas
- Pouco extensíveis



Objetivos

Componente principal



Catalogar e documentar melhores práticas

- Para o desenvolvimento de **fluxos de integração SAP CPI**
- Para o desenvolvimento de código em **Groovy**
- Verificar as regras cobertas pelo **CPILint** e propor novas regras



Revisão automática de integrações

- Criar uma aplicação para rever automaticamente a integração de interfaces desenvolvidas na plataforma **SAP CPI**
- Utilizar o CPILint como ferramenta base de **code review**



Estender as funcionalidades do CPILint

- O sistema de automação da governança deve estender as funcionalidades base do **CPILint**
- Criação/Integração de módulos para revisão automática de ficheiros **Groovy** e **JAR**

Metodologia e Gestão do projeto

Metodologia de desenvolvimento **SCRUM**

Planeamento de **Sprints**: Duração de 1 semana,
com reunião às quintas-feiras com os *Stakeholders*

Repositório de suporte ao desenvolvimento:
Github, utilizando **Issues** para definição das tarefas principais

Criação de um *product backlog*, *Roadmap* e cronograma de desenvolvimento

Estudo Inicial

1. Melhores práticas SAP CPI

Plataforma de integração em nuvem desenvolvida pela SAP



2. Melhores práticas Groovy

Linguagem orientada a objetos, incluída em scripts nos fluxos de integração do CPI

3. Regras do CPILint

Ferramenta de revisão automática, dado um conjunto de regras pré-definido, dos fluxos de integração SAP CPI



4. Potenciais regras a serem adicionadas

Revisão automática de scripts Groovy com auxílio do CodeNarc e deteção de vulnerabilidades em ficheiros JAR utilizando DependencyCheck



Segunda Componente

Objetivo principal:

1. Criação e utilização de um pipeline CI/CD para utilizar de forma remota o CPILint
2. Criar uma interface web



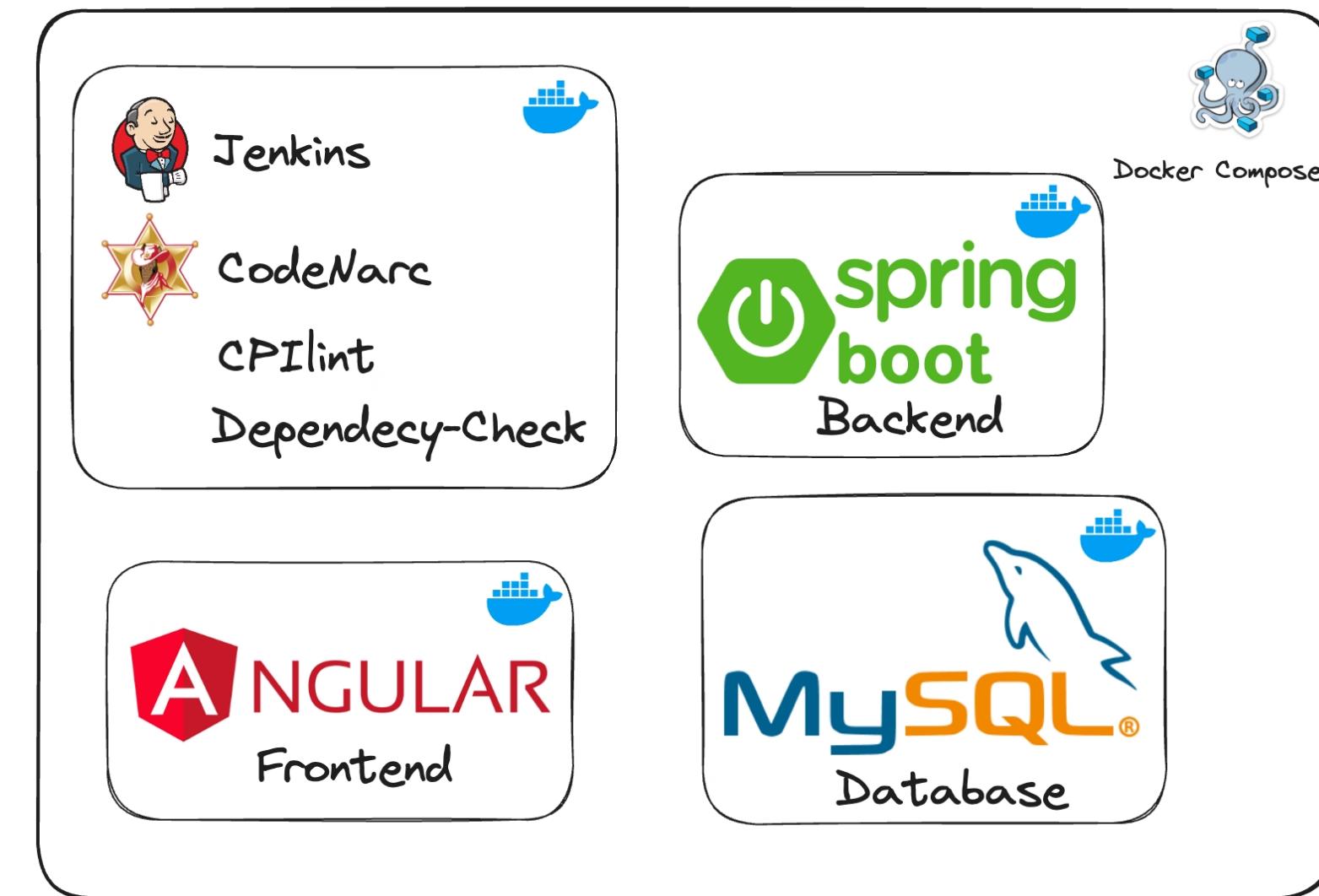
Tecnologias utilizadas:

1. Backend: Java, Spring Boot
2. Frontend: Angular
3. Base de dados: MySQL
4. Virtualização de containers: Docker
5. Servidor de automatização: Jenkins

Arquitetura da solução

Docker Compose com 4 containers:

- Backend
- Frontend
- Base de dados
- Jenkins



Revisão automática de Fluxos de Integração CPI

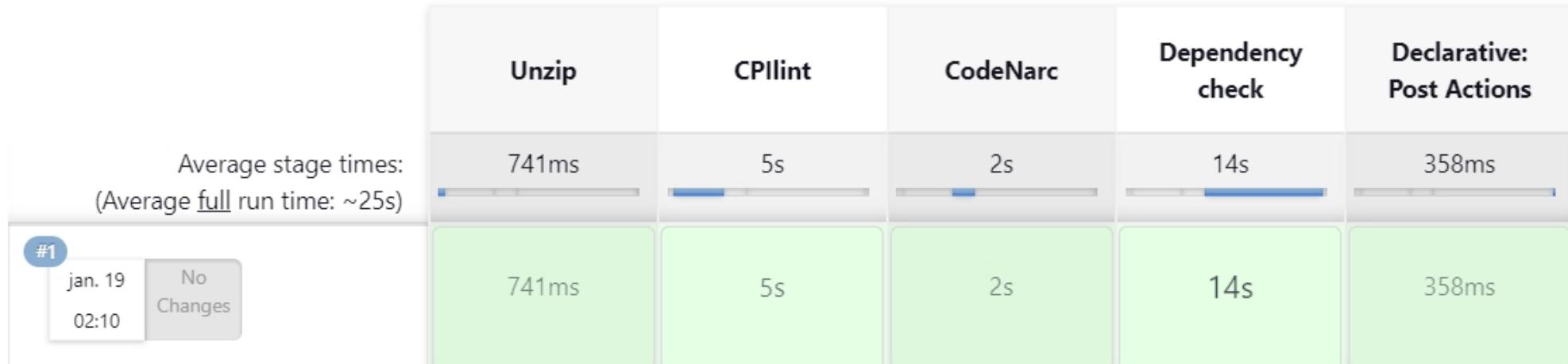
Como foi feita a automatização e extensão do processo de *code review* do CPILint, para incluir ficheiros Groovy e JAR?



Pipeline

Divisão do *pipeline* em 4 stages e uma ação pós construção:

- Stage Unzip
- Stage CPILint
- Stage CodeNarc
- Stage DepedencyCheck
- Limpeza da diretoria de trabalho



Relatório de execução da *code review*

Como são apresentados os resultados de forma intuitiva?

A screenshot of a Jenkins report titled "Jenkins report". The report has a sidebar with icons for home, file, key, and refresh. The main content shows three tabs: "Flow issues", "Groovy issues", and "JAR vulnerabilities", with "JAR vulnerabilities" being the active tab. A table below the tabs displays the following data:

JAR name	Vulnerability	Severity
gson-2.8.4.jar	CVE-2022-25647	HIGH

Resultados da *pipeline*

São gerados 3 ficheiros com os resultados do processo de *code review*:

- CPILint, em formato log
- CodeNarc, em formato JSON
- DependencyCheck, em formato JSON

Interfaces Finais

The screenshot shows the CPI-Lint application interface. On the left, there is a dark sidebar with various icons and links: Home, Packages, Jenkins report, How to, Credentials (which is highlighted), Repositories, Rules, and Logout. Below the sidebar are two small flags: UK and Portugal.

The main content area has a header "Home" and "CPILint". Below the header, there is a section titled "Credentials" with three tabs: SAP CPI (selected), GitHub, and Jenkins.

The SAP CPI tab displays a form for creating a new credential:

- Name***: cpi_credentials
- Base Url***: https://45438691trial.it-cpitrial05.cfapps.us10-001.hana.ondemand.com
- Token Url***: https://45438691trial.authentication.us10.hana.ondemand.com/oauth/token
- Client Id***: sb-4717a4ba-e9a1-4524-917e-7f3392a3be8a!b227937|it!b26655
- Client Secret***: (redacted)

At the bottom of the form are "Update" and "Delete" buttons. The footer of the main content area also includes the UK and Portugal flags.

Interfaces Finais

The screenshot displays the 'Rules' section of the CPI Governance Automation interface. The left sidebar includes icons for Repositories, Home, Folders, Keys, Databases, and a central 'Rules' icon. The main content area has a dark header bar with a 'c' icon and the word 'Rules'. Below this are three buttons: 'Upload CPILint Rule' (document icon), 'Upload Codenarc File' (file icon), and 'All Rules' (list icon). The 'All Rules' button is underlined, indicating it is active. The 'CPILint Rules' section shows one entry: ID 3, Filename test.xml, with a 'Delete' button next to it. The 'Codenarc Rules' section shows one entry: ID 2, Filename rules.groovy, with a 'Delete' button next to it.

ID	Filename	Actions
3	test.xml	<button>Delete</button>

ID	Filename	Actions
2	rules.groovy	<button>Delete</button>

Interfaces Finais

The screenshot displays two main windows from a software application:

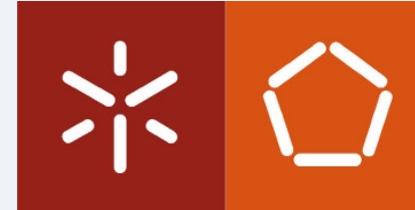
- Package Details: teste**: This window shows detailed information about a package named "teste".
 - Name: teste
 - Created By: rodrigorodrigues56@hotmail.com
 - Creation Date: 1702726112596
 - Description: <p></p>
 - ID: teste
 - Mode: EDIT_ALLOWED
 - Modified By: rodrigorodrigues56@hotmail.com
 - Modified Date: 1702726112607
 - Partner Content: No
 - Resource ID: e47ec5b363e64241b2a9d45b8a1075b8
 - Short Text: teste_pi
 - Supported Platform: Cloud Integration
 - Update Available: No
 - Vendor:
 - Version: </>
- Enable Jenkins Pipeline**: This window is a modal dialog for configuring a Jenkins pipeline.
 - It contains fields for selecting a CPILint Rule (test.xml) and a Codemarc File (rules.groovy).
 - A "Submit" button is at the bottom.

A blue arrow points from the "Actions" menu in the Package Details window towards the "Enable Jenkins Pipeline" dialog, indicating a workflow or integration point between the two features.

Trabalho futuro

Próximos passos

- ✓ Registar várias credenciais para as diversas tecnologias
- ✓ Manter, no repositório, as diferentes versões dos ficheiros de *code review*
- ✓ Possibilidade de realizar download dos ficheiros de *report*, diretamente da interface web
- ✓ Formulário para a escolha de quais as ferramentas usar para fazer *code review*
- ✓ Utilização de IA para realizar correções e sugestões de forma automática



accenture

Cloud Platform Integration Governance Automation



Projeto de Informática - Mestrado em Engenharia Informática
Janeiro 2024