



Código limpo inicia com a sua IDE

Novo Processo Customizados TaxOne.

Objetivo:

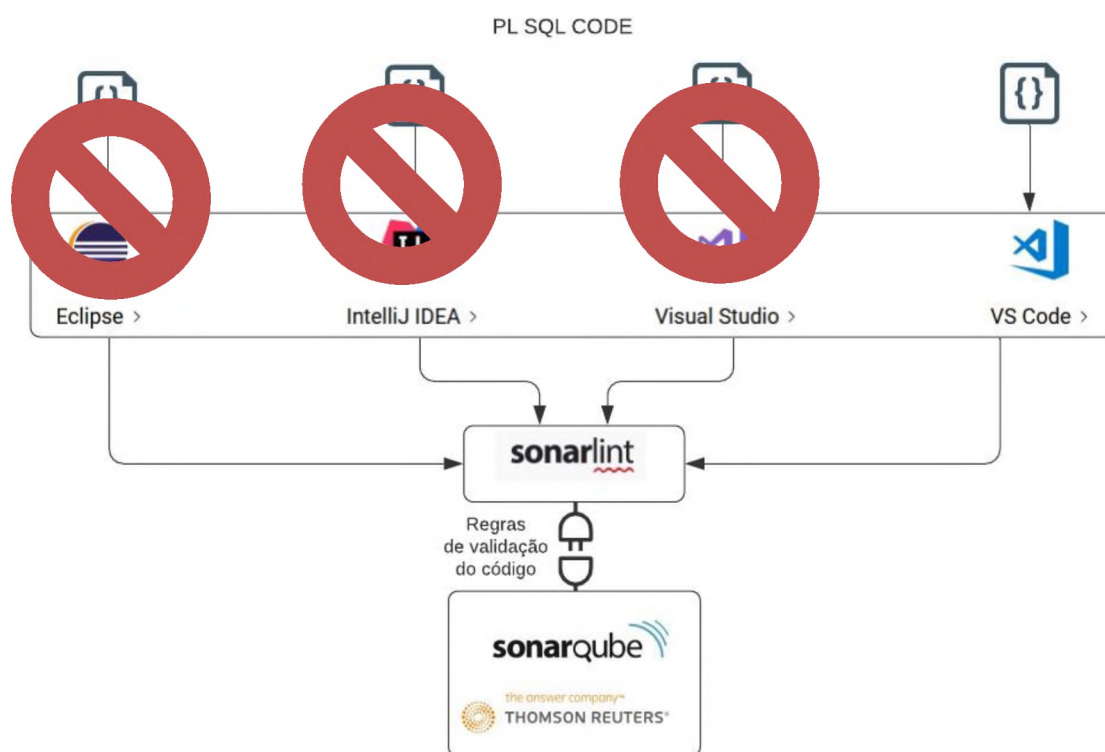
Simplificar o processo e validação dos customizados TaxOne com o uso do SonarLint

Como funciona:

SonarLint é uma extensão IDE (plugin) que ajuda a detectar e corrigir problemas de qualidade a medida que é implementado o código.

O funcionamento é muito semelhante a um corretor ortográfico, o mesmo rabisca as falhas para que possam ser corrigidas antes de confirmar o código. O plugin é conectado ao servidor do SonarQube e as validações feitas para o projeto a ser considerado são as mesmas.

Arquitetura:



IDE's compatíveis:

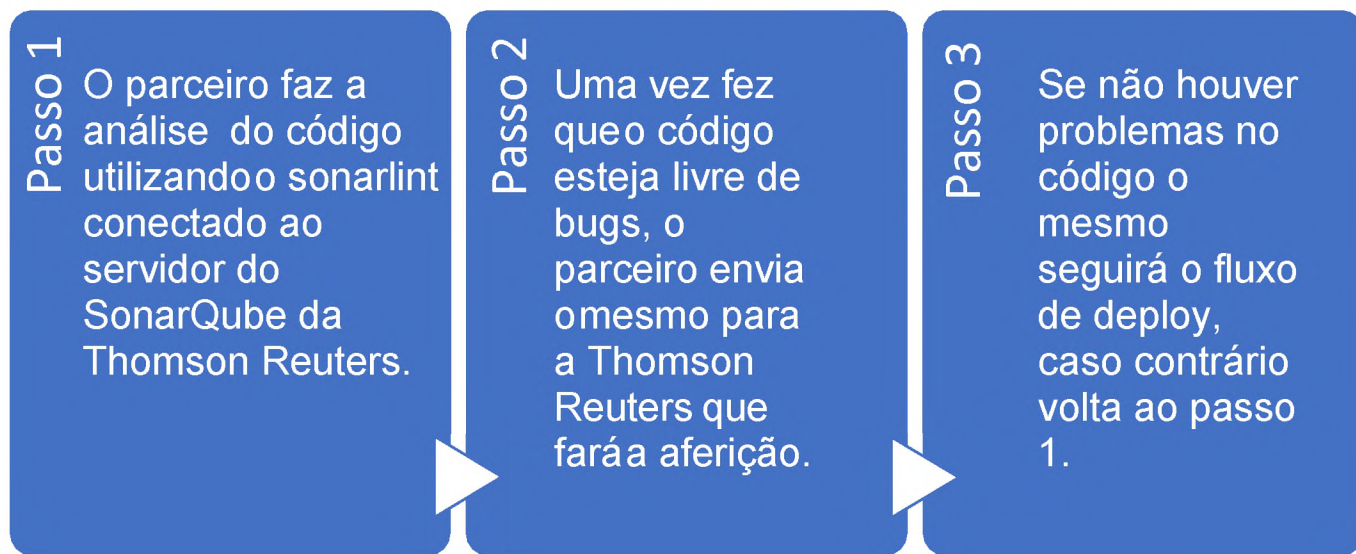
~~Eclipse, IntelliJ, Visual Studio (temporariamente desativados)~~ e VS Code.

Estamos priorizando o VS Code pela facilidade de análise dos indicadores.

Recomendamos para que consigam tirar o PRINT demonstrando zero Bugs utilizem o VS Code.

O Suporte irá aceitar apenas o VS Code.

Fluxo:



Instalação e parametrização do Sonarlint plugin.

VSCode

<https://www.sonarlint.org/vscode> << link para a instalação

Requisitos

O servidor de linguagem SonarLint precisa de um Java Runtime (JRE) 11+. Se um já estiver instalado em seu computador, o SonarLint deve encontrá-lo e usá-lo automaticamente.

Se um JRE adequado não puder ser encontrado nos locais habituais, o SonarLint solicitará sua permissão para baixar e gerenciar sua própria versão.

Por fim, você pode definir explicitamente o caminho em que o JRE está instalado usando a `sonarlint.ls.javaHome` variável nas configurações do VS Code. Por exemplo:

```
{
  "sonarlint.ls.javaHome": "C:\\Program Files\\Java\\jre-11.0.11"
}
```

Requisitos específicos de análise PL/SQL

O suporte para análise PL/SQL só está disponível em conjunto com o SonarQube Developer Edition ou o SonarCloud (consulte o modo conectado abaixo). Você também precisa da extensão [Oracle Developer Tools for VSCode](#).

Modo conectado

Você pode conectar o SonarLint ao SonarQube >= 7.9 ou SonarCloud e vincular suas pastas do espaço de trabalho a um projeto SonarQube/SonarCloud para se beneficiar das mesmas regras e configurações usadas para inspecionar seu projeto no servidor. O SonarLint oculta no VSCode os problemas marcados como **Não será corrigido** ou **Falso positivo**.

O modo conectado também permitirá desbloquear a análise desses idiomas:

- [Regras do Apex](#)
- [Regras PL/SQL](#).

A primeira etapa é configurar os detalhes da conexão (token do usuário, URL do servidor SonarQube ou organização SonarCloud). Por motivos de segurança, o token não deve ser armazenado no SCM com configurações de espaço de trabalho. É por isso que sugerimos configurá-los nas configurações do usuário do VSCode.

Exemplo para SonarQube:

```
{
  "sonarlint.connectedMode.connections.sonarqube": [
    { "serverUrl": "https://sonar.qa.thomsonreuters.com",
      "token": "76ee9bcfb3653d92a232b9b1b4665ca70cb923ef" }
  ]
}
```




```
}
```

A segunda etapa é configurar a vinculação do projeto, no nível do espaço de trabalho ou em todas as pastas do espaço de trabalho. Exemplo:

```
{  
  "sonarlint.connectedMode.project": {  
    "projectKey": "customizados"  
  }  
}
```

Eclipse:

<https://www.sonarlint.org/eclipse> << link para a instalação

<https://github.com/SonarSource/sonarlint-eclipse/wiki/Connected-Mode> << link para a configuração e conexão

<https://www.sonarlint.org/visualstudio> << link para a instalação

<https://github.com/SonarSource/sonarlint-visualstudio> << link para a configuração e conexão

<https://www.sonarlint.org/intellij> << link para a instalação

<https://github.com/SonarSource/sonarlint-intellij> << link para a configuração e conexão


Dados para conexão com o servidor SonarQube:

url: <https://sonar.qa.thomsonreuters.com>

token: 76ee9bcfb3653d92a232b9b1b4665ca70cb923ef

projeto: customizados

Legenda issues apresentadas pelo Sonarlint:

 Bugs

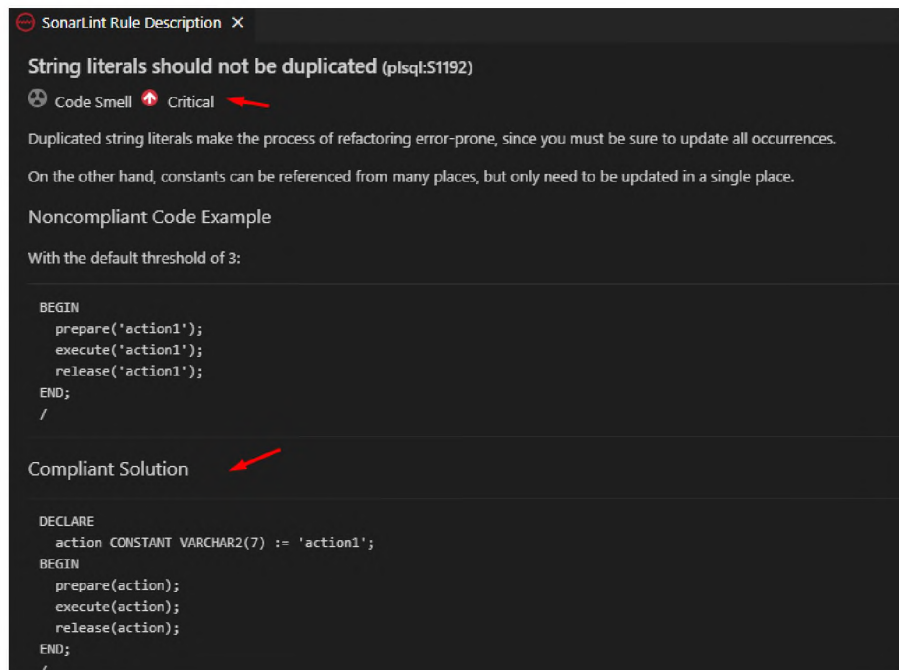
 Vulnerabilities

 Code Smells

 Hotspots

Observação:

Para entrar no detalhe da regra e verificar a sua classificação, basta clicar com o botão direito do mouse na mesma que será apresentado uma tela com tais informações e com uma sugestão para a resolução.



Resumo apresentado no último treinamento 15/Set/22 do SonarLint:

Customizados – SonarLint substituiu o Sonar

Objetivo: Simplificar o processo e validação dos customizados

SonarLint é uma extensão IDE (plugin) que ajuda a detectar e corrigir problemas no código IDE's compatíveis: Eclipse, IntelliJ, Visual Studio e VS Code.

Exemplo com o IDE VS Code:

Fazer instalação do IDE da sua preferência, neste exemplo VC Code.

Instalação e parametrização do plugin do SonarLint neste link. <https://www.sonarlint.org/vscode>

Dentro dos settings do IDE colar os parâmetros abaixo:

```
{
  "sonarlint.connectedMode.connections.sonarqube": {
    { "serverUrl": "https://sonar.qa.thomsonreuters.com", "token": "76ee9bcfb3653d92a232b9b1b4665ca70cb923ef" } }
  {
    "sonarlint.connectedMode.project": { "projectKey": "customizados" } }
}
```

A partir de agora o IDE conectará com o SonarQube e terá acesso as regras de validação dos customizados da TR.

Customizados – SonarLint substituiu o Sonar

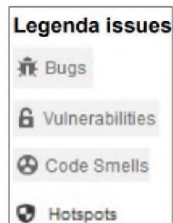
Exemplo com o IDE VS Code (continuação):

Caso não tenha, o IDE precisará instalar um plug in (requisitos específicos de análise PL/SQL)

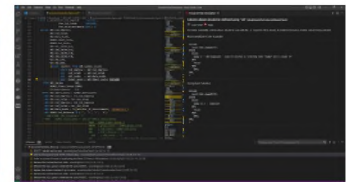
A partir disso, basta jogar o código do Customizado dentro do IDE para análise e correção (1º pode demorar pois o IDE faz toda baixa de regras)
Após a finalização da análise, corrigir o código até ter o padrão abaixo.

Análise e correção do Código:

- ZERO Bugs
- ZERO Vulnerabilidades
- ZERO Code Smell: Blocker, High e Security
- Code Smell restante não é necessário
- Hotspots não é necessário tb corrigir
- No print deve aparecer ZERO Erros e ZERO Warnings.



Enviar print da tela para Time Migração



15

Customizados – SonarLint

Demonstração

Issues list:

- ⚠ Add the AS keyword prior to the column alias. sonarlint(plsql:UseAsForColumnAliases)
- ⚠ Define a constant instead of duplicating this literal 51 times. [+50 location]
- ⚠ Remove this commented out code. sonarlint(plsql:S125) [Ln 53, Col 1]
- ⚠ Refactor this SQL query to eliminate the use of EXISTS. sonarlint(plsql:S125) [Ln 99, Col 1]
- ⚠ Replace the column number 5 by its name. sonarlint(plsql:OrderByColumnNumber)
- ⚠ Remove this commented out code. sonarlint(plsql:S125) [Ln 99, Col 1]
- ⚠ Refactor this SQL query to eliminate the use of EXISTS. sonarlint(plsql:S125) [Ln 99, Col 1]
- ⚠ Refactor this SQL query to eliminate the use of EXISTS. sonarlint(plsql:S125) [Ln 99, Col 1]

PRINT
Local do IDE onde vem número ERROS, WARNINGS que tem que ZERAR.
INFOS não precisam corrigir.

16

[Olhar todos os requisitos no manual](#)

Requisitos

O servidor de linguagem SonarLint precisa de um Java Runtime (JRE) 11+. Se um já estiver instalado em seu computador, o SonarLint deve encontrá-lo e usá-lo automaticamente.

Se um JRE adequado não puder ser encontrado nos locais habituais, o SonarLint solicitará sua permissão para baixar e gerenciar sua própria versão.

Requisitos específicos de análise PL/SQL

O suporte para análise PL/SQL só está disponível em conjunto com o SonarQube Developer Edition ou o SonarCloud (consulte o modo conectado abaixo). Você também precisa da extensão [Oracle Developer Tools for VSCode](#).