



iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital

Módulo 3: Princípios de Desenvolvimento de Software

# Aula 11- Análise de Código (Lint), Documentação



# Lint? Linter? O que é?

- Lint é um programa que executa uma verificação de código.
- Tem como objetivo procurar erros de programação e de boas práticas (estilo).
- É usado para:
  - unificar o estilo de escrita num projeto.
  - obrigar o uso de boas práticas
  - melhorar a interpretação do código
  - aumentar a performance do projeto

# Lint? Linter? O que é?

- Exemplos de erros de programação:
  - Falta de “;” quando obrigatório
  - Variáveis declaradas e não usadas
  - Uso de variáveis antes de declaradas
  - Execução de código depois de declarado o retorno em funções
  - Uso de imports descontinuados ou não seguros
  - Loops infinitos

# Lint? Linter? O que é?

- Exemplos de erros de estilo:
  - Espaçamento indevido
  - Falta de indentação
  - Ultrapassagem de limites de caracteres por linha
  - Uso misto de tipo de aspas (“ e ‘)
  - Detecção de código grande

# SonarLint

- É um plugin que contém uma ferramenta de Lint
- É usada para identificar erros no código durante o desenvolvimento
- SonarLint é um projeto openSource desenvolvido pela SonarSource
- Está disponível para IDE's como o IntelliJ, Eclipse e VSCode

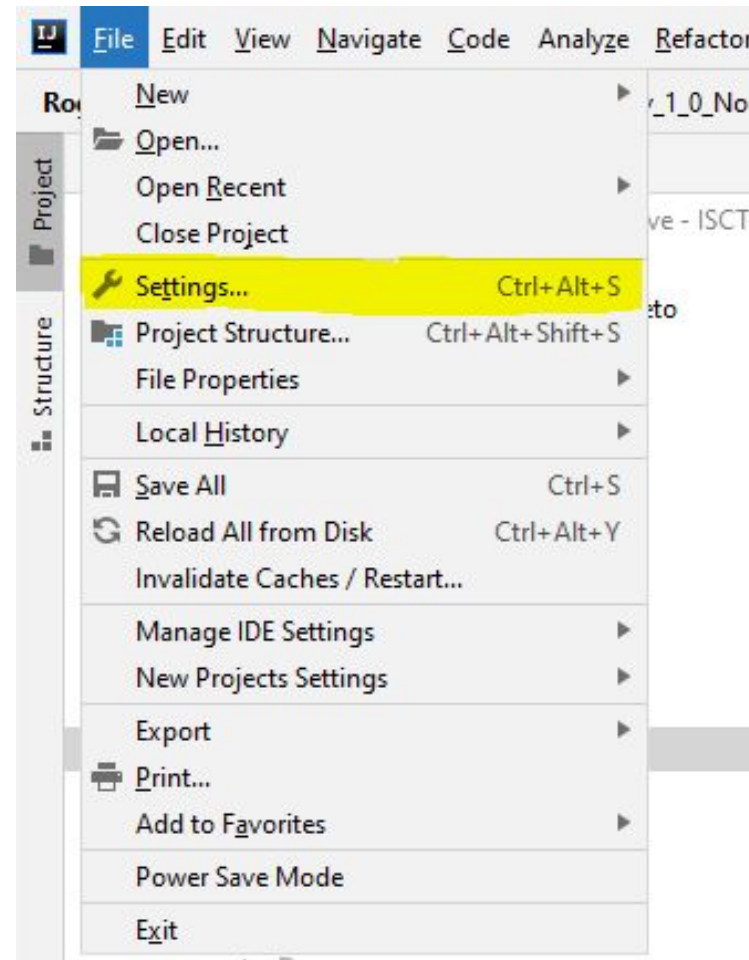


**Como usar? Como instalar?  
É só seguir o próximo tutorial!**

**sonarlint**  

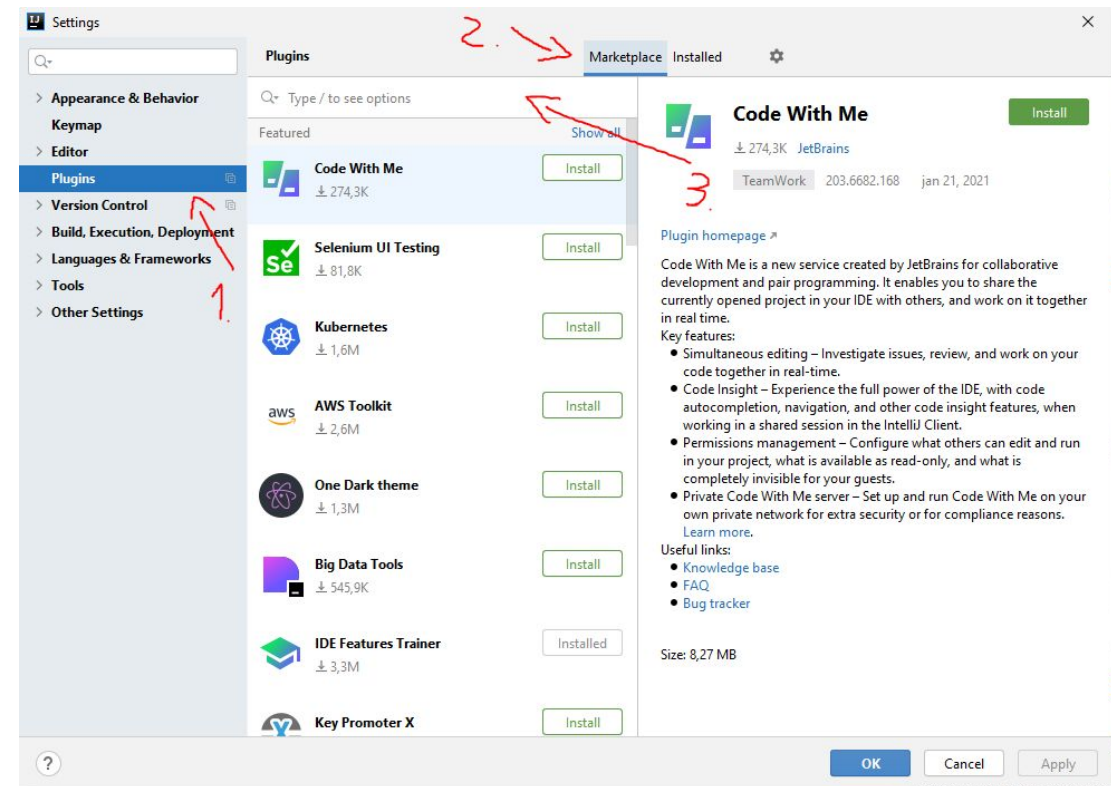

# Instalar o SonarLint no IntelliJ?

1. Abrir o IntelliJ
2. File > Settings



# Instalar o SonarLint no IntelliJ?

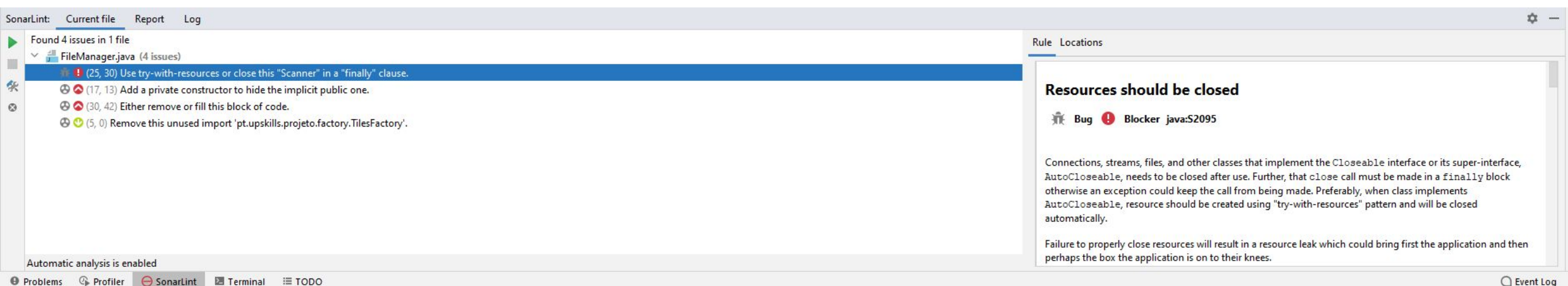
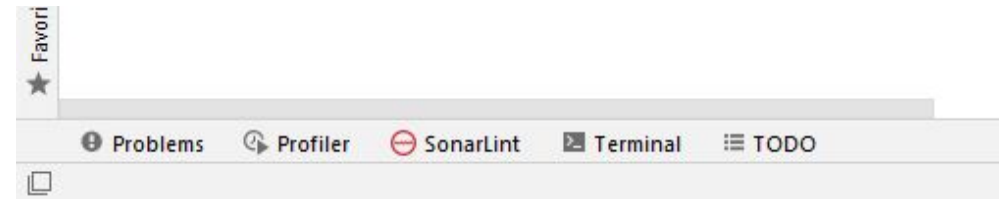
1. Selecionar Plugins
2. Selecionar Marketplace
3. Pesquisar por “SonarLint”
4. Instalar o SonarLint
5. Reiniciar o IntelliJ





# Como usar o SonarLint no IntelliJ?

1. Abrir o ficheiro a analisar
2. Clicar em SonarLint, na aba em baixo
3. Um relatório detalhado da análise irá aparecer.



# Documentação

Fundamental para:

- Manter e melhorar o software existente
- Permitir gestão de equipas de desenvolvimento
- Partilhar código
- Grandes projetos
- Grandes equipas de desenvolvimento
- Desenvolvimento distribuído

# E que documentação usamos em Java?

- Essencialmente Javadoc.
- Javadoc é uma ferramenta que permite a geração automática de documentação.
- Gera páginas HTML que descrevem classes, interfaces, construtores, métodos, atributos, etc...
- Os comentários Javadoc começam com `/**` e aparecem sempre antes do elemento que documentam
- Alternativa ao Javadoc: Doxygen

# Javadoc: Exemplo

```
/**
 * Classe representativa do veículo autocarro.
 * @author Jorge Rafael Santos
 */
public class Autocarro extends VeiculoTerrestre {

    /**
     * Construtor da classe autocarro
     * @param capacidade do veículo
     */
    public Autocarro(int capacidade) { super(capacidade); }

    /**
     * Devolve o consumo do autocarro. O consumo depende da lotação do veículo: lotação
     * superior a 50% devolve o dobro do consumo normal.
     * @return int consumo do veículo
     */
    @Override
    public int getConsumo(){
        if((double)(getLotacao()/getCapacidade()) > 0.5){
            return getConsumoBase()*2;
        }else {
            return super.getConsumo();
        }
    }
}
```

# Javadoc: principais anotações

- @author: Nome do autor da classe
- @param: Descreve o parâmetro de um método
- @return: Descreve o tipo de valor devolvido por um método
- @throws: Descreve a exceção que pode ser lançada por um método
- @exception: Descreve uma exceção
- @see: Documentação a associação a uma outra classe

# Javadoc: como gerar no IntelliJ?

1. Tools > Generate Javadoc.
2. Selecionar o projeto/ficheiros a gerar.
3. Selecionar a pasta de destino.
4. Clicar em OK.

# Javadoc

← → ↻ ⓘ Ficheiro | C:/Users/N580V/Desktop/index.html

All Classes

Autocarro

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

## Package pt.iscte.formacao.java.aula9

**Class Summary**

Class	Description
Autocarro	Classe representativa do veículo autocarro.

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

## Class Autocarro

java.lang.Object

Veículo

pt.iscte.formacao.java.aula9.VeiculoTerrestre

pt.iscte.formacao.java.aula9.Autocarro

```
public class Autocarro
extends pt.iscte.formacao.java.aula9.VeiculoTerrestre
```

Classe representativa do veículo autocarro.

Author:

Jorge Rafael Santos

### Constructor Summary

#### Constructors

##### Constructor and Description

Autocarro(int capacidade)

Construtor da classe autocarro

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

int

##### Method and Description

getConsumo()

Devolve o consumo do autocarro.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### Autocarro

public Autocarro(int capacidade)

Construtor da classe autocarro

# Javadoc



# Dúvidas?

**Digam agora ou... perguntem depois.**

# Exercícios

- Corrigir o projeto 1 (Rogue) aplicando o Sonarlint.
- Documentar as classes desse projeto.

# Bibliografia

[https://medium.com/@emerson\\_pereira/lint-o-que-%C3%A9-isso-afinal-83b3dc0dec59](https://medium.com/@emerson_pereira/lint-o-que-%C3%A9-isso-afinal-83b3dc0dec59)

<https://www.testim.io/blog/what-is-a-linter-heres-a-definition-and-quick-start-guide/>

<https://medium.com/@rafaelvicio/mantenha-seu-c%C3%B3digo-java-limpo-com-sonarlint-951a8f375ece>

<https://www.sonarlint.org/>

<https://www.oracle.com/pt/technical-resources/articles/java/javadoc-tool.html>

# O futuro profissional começa aqui

iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital



UPskill