Quality Profile Samples

In the first section the sample macro shows quality profile name, enabled rules by rule type, with links to the rules.

In the second section the sample macro shows a breakdown with all enabled rules of the Quality Profile, with severity, rule name , language, rule type and tags.

SonarQube sample

Macro Perfil de CalidadMacro Perfil de CalidadEsta sección se focaliza en las reglas de un perfil de calidad

Nombre del perfil de calidad: excentia-css-profile

Reglas	Activas	Inactivas
Bugs	15	1
Vulnerabilidades	0	0
Code Smells	9	1
Security Hotspots	0	0
Total	**24**	**2**

Severidad	Regla	Lenguaje	Tipo	Etiquetas
	Multi-line comments should not be empty	CSS		
	Selectors should not be duplicated	CSS		
	Font declarations should contain at least one generic font family	CSS		
	Sections of code should not be commented out	CSS		unused
	Duplicate imports should be removed	CSS		unused
	Single line comment syntax should not be used	CSS		
	"at-rules" should be valid	CSS		
	CSS files should not be empty	CSS		
	Media features should be valid	CSS		
	Duplicated font names should be removed	CSS		
	Color definitions should be valid	CSS		

	Extra semicolons should be removed	CSS	reliability, unused
	Pseudo-element selectors should be valid	CSS	
	"!important" should not be used on "keyframes"	CSS	
	CSS properties should be valid	CSS	
	Shorthand properties that override related longhand properties should be avoided	CSS	
	Properties should not be duplicated	CSS	
	"calc" operands should be correctly spaced	CSS	
	Units should be valid	CSS	
	Strings should not contain new lines	CSS	
	Pseudo-class selectors should be valid	CSS	
	Empty blocks should be removed	CSS	
	Selectors should be known	CSS	
	"linear-gradient" directions should be valid	CSS	

SonarCloud sample

Macro Perfil de CalidadMacro Perfil de CalidadEsta sección se focaliza en las reglas de un perfil de calidad

Nombre del perfil de calidad: Sonar way
Organización: simgrid
Lenguaje: java

Reglas	Activas	Inactivas
Bugs	159	13
Vulnerabilidades	52	4
Code Smells	292	141
Security Hotspots	36	2
Total	**539**	**160**

Severidad	Regla	Lenguaje	Tipo	Etiquetas
	Bean names should adhere to the naming conventions	Java		spring
	Generic wildcard types should not be used in return types	Java		pitfall
	Raw types should not be used	Java		pitfall
	Credentials should not be hard-coded	Java		cwe
	Future keywords should not be used as names	Java		obsolete, pitfall
	NoSQL operations should not be vulnerable to injection attacks	Java		cwe
	".equals()" should not be used to test the values of "Atomic" classes	Java		multi-threading
	Regular expression quantifiers and character classes should be used concisely	Java		regex
	Nullable injected fields and parameters should provide a default value	Java		spring
	Use record pattern instead of explicit field access	Java		java21
	Virtual threads should not run tasks that include synchronized code	Java		java21, multi-threading
	"Math.clamp" should be used with correct ranges	Java		java21
	"String.indexOf" should be used with correct ranges	Java		java21
	Formatting SQL queries is security-sensitive	Java		bad-practice, cert,cwe, hibernate, spring,sql
	Secure random number generators should not output predictable values	Java		cert,cwe, pitfall
	Cipher Block Chaining IVs should be unpredictable	Java		cwe
	The "Object.finalize()" method should not be overridden	Java		cert, unpredictable

	Loggers should be named for their enclosing classes	Java	confusing ,logging
	Unused local variables should be removed	Java	unused
	Only static class initializers should be used	Java	pitfall
	Reverse iteration should utilize reversed view	Java	java21
	Reverse view should be used instead of reverse copy in read-only cases	Java	java21
	Unused "private" fields should be removed	Java	unused
	Encryption algorithms should be used with secure mode and padding scheme	Java	cwe, privacy
	Format strings should be used correctly	Java	cert, confusing
	Cryptographic keys should be robust	Java	cwe, privacy, rules
	Exceptions should be either logged or rethrown but not both	Java	error-handling, logging
	"setDaemon", "setPriority" and "getThreadGroup" should not be invoked on virtual threads	Java	java21
	Constant parameters in a "PreparedStatement" should not be set more than once	Java	sql, sustainability
	SQL queries should retrieve only necessary fields	Java	sql, sustainability
	Use batch Processing in JDBC	Java	sql, sustainability
	Use when instead of a single if inside a pattern match body	Java	java21
	Fields in a "Serializable" class should either be transient or serializable	Java	cwe, serialization
	XML signatures should be validated securely	Java	symbolic-execution
	All branches in a conditional structure should not have exactly the same implementation	Java	

	Passwords should not be stored in plaintext or with a fast hashing algorithm	Java	cwe, spring
	Local variable and method parameter names should comply with a naming convention	Java	convention
	Proper Sensor Resource Management	Java	android, leak, sustainability
	Use built-in "Math.clamp" methods	Java	java21
	Use switch instead of if-else chain to compare a variable against multiple cases	Java	java21
	Exception types should not be tested using "instanceof" in catch blocks	Java	cert, clumsy, error-handling
	Virtual threads should be used for tasks that include heavy blocking operations	Java	java21, multi-threading
	String literals should not be duplicated	Java	design
	Password hashing functions should use an unpredictable salt	Java	cwe
	Equals method should be overridden in records containing array fields	Java	java16
	Endpoints should not be vulnerable to reflected cross-site scripting (XSS) attacks	Java	cwe
	"toString()" and "clone()" methods should not return null	Java	cert,cwe
	Boolean expressions should not be gratuitous	Java	cwe, redundant, suspicious, symbolic-execution
	Track uses of "TODO" tags	Java	cwe
	Similar tests should be grouped in a single Parameterized test	Java	bad-practice, clumsy, tests
	Track uses of "FIXME" tags	Java	cwe

	Locks should be released on all paths	Java	cwe,multi-threading,symbolic-execution
	Tests should be stable	Java	design,tests,unpredictable
	Using slow regular expressions is security-sensitive	Java	cwe,regex
	Conditionally executed code should be reachable	Java	cert,cwe,pitfall,suspicious,symbolic-execution,unused
	"@Deprecated" code marked for removal should never be used	Java	cert,cwe,obsolete
	A secure password should be used when connecting to a database	Java	cwe
	Jump statements should not occur in "finally" blocks	Java	cert,cwe,error-handling
	"Random" objects should be reused	Java	
	"@Bean" methods for Singleton should not be invoked in "@Configuration" when proxyBeanMethods is false	Java	spring
	Optional value should only be accessed after calling isPresent()	Java	cwe,symbolic-execution
	Blocks should be synchronized on "private final" fields	Java	cert,cwe,multi-threading
	Cognitive Complexity of methods should not be too high	Java	brain-overload
	Disabling CSRF protections is security-sensitive	Java	cwe,spring
	Non-serializable objects should not be stored in "javax.servlet.http.HttpSession" instances	Java	cwe
	The "Object.finalize()" method should not be called	Java	cert,cwe

	Multiline blocks should be enclosed in curly braces	Java	cert,cwe
	Delivering code in production with debug features activated is security-sensitive	Java	cwe, debug, error-handling, spring, user-experience
	"null" should not be returned from a "Boolean" method	Java	cert,cwe, pitfall
	Setting JavaBean properties is security-sensitive	Java	cert,cwe
	Boolean literals should not be redundant	Java	clumsy
	Assignments should not be made from within sub-expressions	Java	cert,cwe, suspicious
	Mutable fields should not be "public static"	Java	cert,cwe, unpredictable
	Beans in "@Configuration" class should have different names	Java	spring
	"InterruptedException" and "ThreadDeath" should not be ignored	Java	cwe,error-handling, multi-threading
	"Object.finalize()" should remain protected (versus public) when overriding	Java	cert,cwe
	JWT should be signed and verified with strong cipher algorithms	Java	cwe, privacy
	Cipher algorithms should be robust	Java	cwe, privacy
	Classes that override "clone" should be "Cloneable" and call "super.clone()"	Java	cert, convention,cwe
	Throwable and Error should not be caught	Java	bad-practice, cert,cwe, error-handling
	A new session should be created during user authentication	Java	cwe, spring

	Weak SSL/TLS protocols should not be used	Java	cwe, privacy
	Using unsafe Jackson deserialization configuration is security-sensitive	Java	cwe
	Using pseudorandom number generators (PRNGs) is security-sensitive	Java	cwe
	"Integer.toHexString" should not be used to build hexadecimal strings	Java	cwe
	Creating cookies without the "HttpOnly" flag is security-sensitive	Java	cwe, privacy
	Null pointers should not be dereferenced	Java	cert,cwe, symbolic-execution
	Allowing deserialization of LDAP objects is security-sensitive	Java	cwe
	"@PathVariable" annotation should be present if a path variable is used	Java	spring
	SpEL expression should have a valid syntax	Java	spring
	Using non-standard cryptographic algorithms is security-sensitive	Java	cwe
	"HttpServletRequest. getRequestedSessionId()" should not be used	Java	cwe
	LDAP connections should be authenticated	Java	cwe
	Server hostnames should be verified during SSL/TLS connections	Java	cwe, privacy, ssl
	Using weak hashing algorithms is security-sensitive	Java	cwe
	"@NonNull" values should not be set to null	Java	cert,cwe, symbolic-execution
	XML parsers should not be vulnerable to XXE attacks	Java	cwe, symbolic-execution
	"@Deprecated" code should not be used	Java	cert,cwe, obsolete
	Classes should not be compared by name	Java	cert,cwe

	Setting loose POSIX file permissions is security-sensitive	Java	cert,cwe