1. Analizadores de código

1.1 Introdución

Nesta parte da unidade didáctica trataranse os seguintes conceptos:

- Identificar as posibilidades dun analizador de código.
- Revisar código fonte utilizando o analizador de código do contorno de desenvolvemento libre.

1.2 Os analizadores de código

Analizadores de código

Os analizadores de código permiten revisar o código e obter informes que axudarán ao programador a optimizalo. Poden ser estáticos ou dinámicos.

Os **analizadores estáticos** permiten realizar a análise sen executar o código. Están enfocados normalmente á validación e optimización do código.

Os **analizadores dinámicos** permiten realizar a análise executando o código nun procesador real ou virtual e obtendo un informe de rendemento.

Criterios de valoración dun analizador

- Tipo de **licenza** e custo.
- **Usabilidade**. Deberán de valorarse a axuda e a documentación, o tempo de instalación, o tempo de configuración do contorno, as actualizacións e a facilidade para interpretar os resultados.
- **Eficiencia**. Valoraranse o tempo de execución e o consumo de recursos.
- **Extensibilidade**. Valorarase a posibilidade de engadir regras facilmente cunha linguaxe sinxela (XPath, java, JML).
- **Precisión** dos resultados obtidos.

Analizadores estáticos

Os analizadores estáticos son ferramentas obxectivas que miden o estado do código e dan información sobre a **calidade** do mesmo para poder detectar e previr problemas derivados de código de baixa calidade: funcións duplicadas, métodos excesivamente complexos, estilos de codificación completamente diferentes para cada desenvolvedor...

A análise estática de código consigue aforros importantes nos custos ao poder anticipar problemas antes de que se fagan realidade. A organización pode así evitar o gasto que supón a corrección de defectos.

Idealmente debemos incorporar estas ferramentas ao comezo do desenvolvemento pero, de non ser así, farase en canto sexa posible.

Analizadores dinámicos

Os analizadores dinámicos miden en tempo de execución o comportamento dun programa coa fin de determinar o uso que fai dos recursos do sistema como CPU ou memoria ou os tempos de execución, obtendo un informe que permitirá ao programador optimizar o seu código.

Os analizadores dinámicos actúan durante a execución do código que se pretende analizar polo que é importante:

- Elixir un grupo de datos de entrada que permita obter resultados fiables para a análise.
- Minimizar o efecto que poden ter na execución as ferramentas ou instrumentos utilizados na medición.

Analizar código Java con netbeans

SonarQube é unha ferramenta de análise estática que avalía o código fonte. Trátase de software libre que emprega diversos plugins de análise estática de código fonte como Checkstyle, PMD ou Findbugs para obter métricas que podan axudar a mellor a calidade do código dun desenvolvemento software e entender que problemas temos no mesmo. SonarQube ten un servidor asociado.

SonarLint é unha extensión de **IDE** gratuíta e open source que identifica e axuda a solucionar problemas de **calidade** e **seguridade** mentres se escribe o código. SonarLint é como un corrector automático que proporciona comentarios en tempo real e unha guía de corrección para poder ter código limpo desde o principio.

En NetBeans podemos utilizar este plugin (SonarLint for NetBeans). O proxecto SonarLint para NetBeans está aloxado en GitHub en https://github.com/philippefichet/sonarlint4netbeans.

O IDE NetBeans posúe, ademais, unha ferramenta de **perfilado** (profiling) que fai que NetBeans teña a posibilidade de facer unha análise dinámica de código Java que inclúe **análise da CPU**, **memoria** e **subprocesos** así como **monitorización** básica da **JVM**, permitindo aos programadores ser máis produtivos na solución dos problemas de memoria ou no rendemento da aplicación.

Esta análise permitirá identificar as instrucións do código que supoñen una maior carga (en tempo, memoria, recursos, etc.) para a aplicación, para que o programador optimice ese código e evite esa sobrecarga actuando en consecuencia.

É recomendable que no momento de realizar a análise non se estea executando ningunha outra aplicación no equipo xa que entón os resultados da análise de rendemento poden non ser exactos.

Os pasos para realizar unha análise son:

- Definir o elemento do proxecto sobre o que se vai facer a análise, seleccionar o tipo de medición que se quere realizar e as condicións adicionais que se necesiten.
- Executar a análise.
- Emitir resultados. NetBeans informa sobre as partes do código que están consumindo máis recursos do sistema durante a execución, é dicir, informa sobre a sobrecarga (overhead) na execución do código.