TEMA 4-A02: Analizador de código

Índice

1. Analizador de código.

1.1 Introdución

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas:

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

1.2 Actividade

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 de un 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

Tarefa 1. Procura de analizadores estáticos de código no mercado e comparativas.

Analizadores dinámicos

Os analizadores dinámicos miden en tempo de execución o comportamento dun programa co 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 que:

- Elíxase un grupo de datos de entrada que permita obter resultados fiables para a análise.

- Minimícese o efecto que poden ter na execución as ferramentas ou instrumentos utilizados na execución.

Tarefa 2. Procura de analizadores dinámicos de código no mercado e comparativas.

Analizar dinamicamente código java con netbeans

Introdución

NetBeans IDE posúe o módulo NetBeans profiler que provee a NetBeans dunha funcionalidade completa para o análise dinámico de código Java que inclúe análise da CPU, memoria e subprocesos así como monitorización JVM básica permitindo aos programadores ser máis produtivos na solución dos problemas de memoria ou no rendemento da aplicación. Pódense analizar proxectos Java SE, aplicacións Web, proxectos Java EE projects, proxectos Java Freeform e módulos NetBeans.

Máis información no seguinte enlace:

https://docs.oracle.com/netbeans/nb82/netbeans/NBDAG/test_profile_japps.htm#NBDAG673

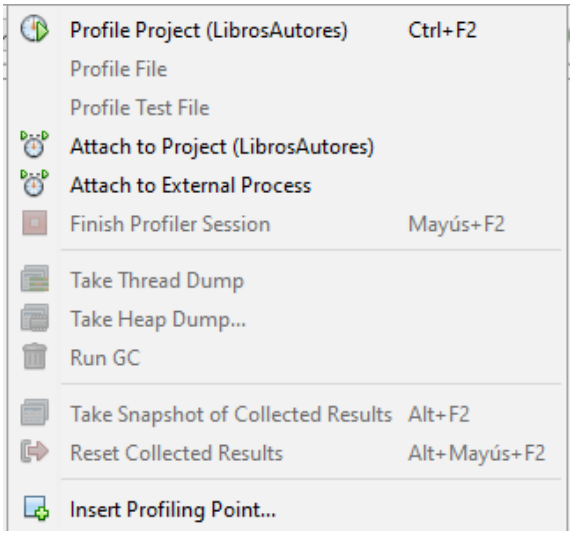
Esta análise permitirá identificar as instrucións do código que supoñen una maior carga (en tempo, memoria, recursos, etc.) para a aplicación, para que o programador optimice ese código e evite esa sobrecarga actuando en consecuencia.

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

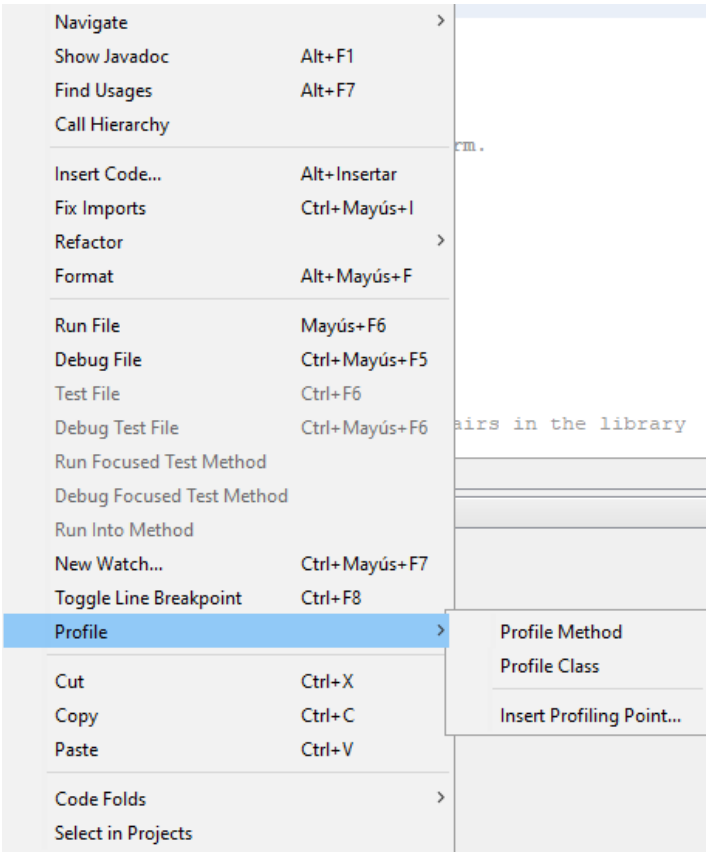
Os pasos para realizar unha análise son:

- 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.

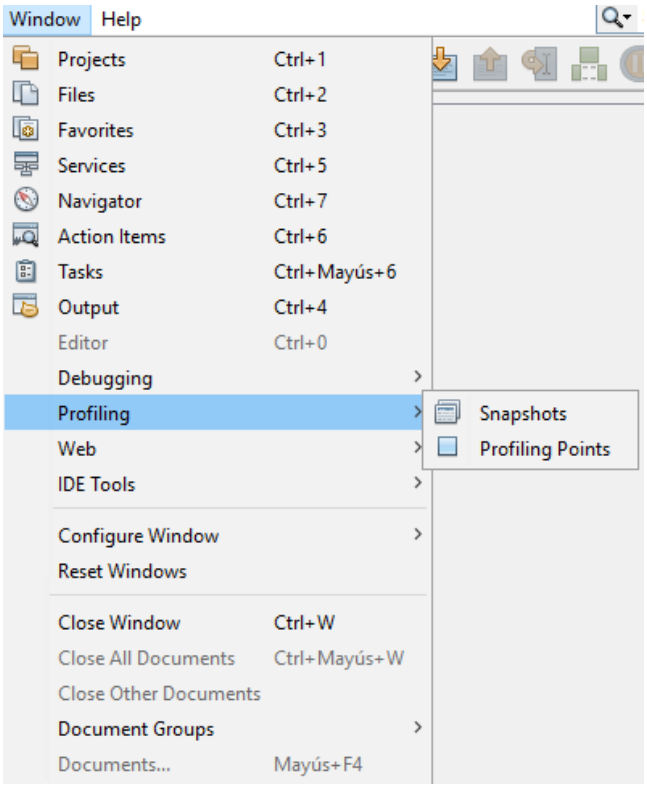
A opción *Profile* do menú principal permite xestionar a análise. A opción *Profile->Profile Main Project* do menú principal permite analizar o proxecto principal.



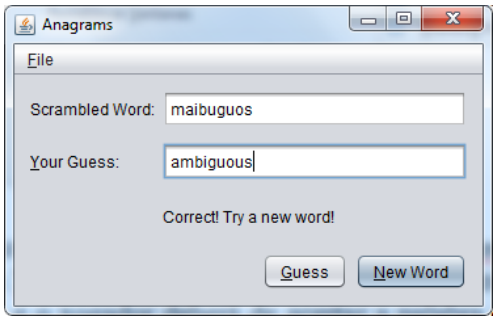
Nalgúns casos, poderanse realizar operación de análise directamente sobre un elemento do código facendo clic dereito sobre o elemento, elixindo *Analizando* e definindo ou seleccionado o elemento a analizar.



Ao longo do proceso de análise e consulta de resultados manéxanse distintas ventás que de non estar visibles poden activarse na opción *Ventana* do menú principal:



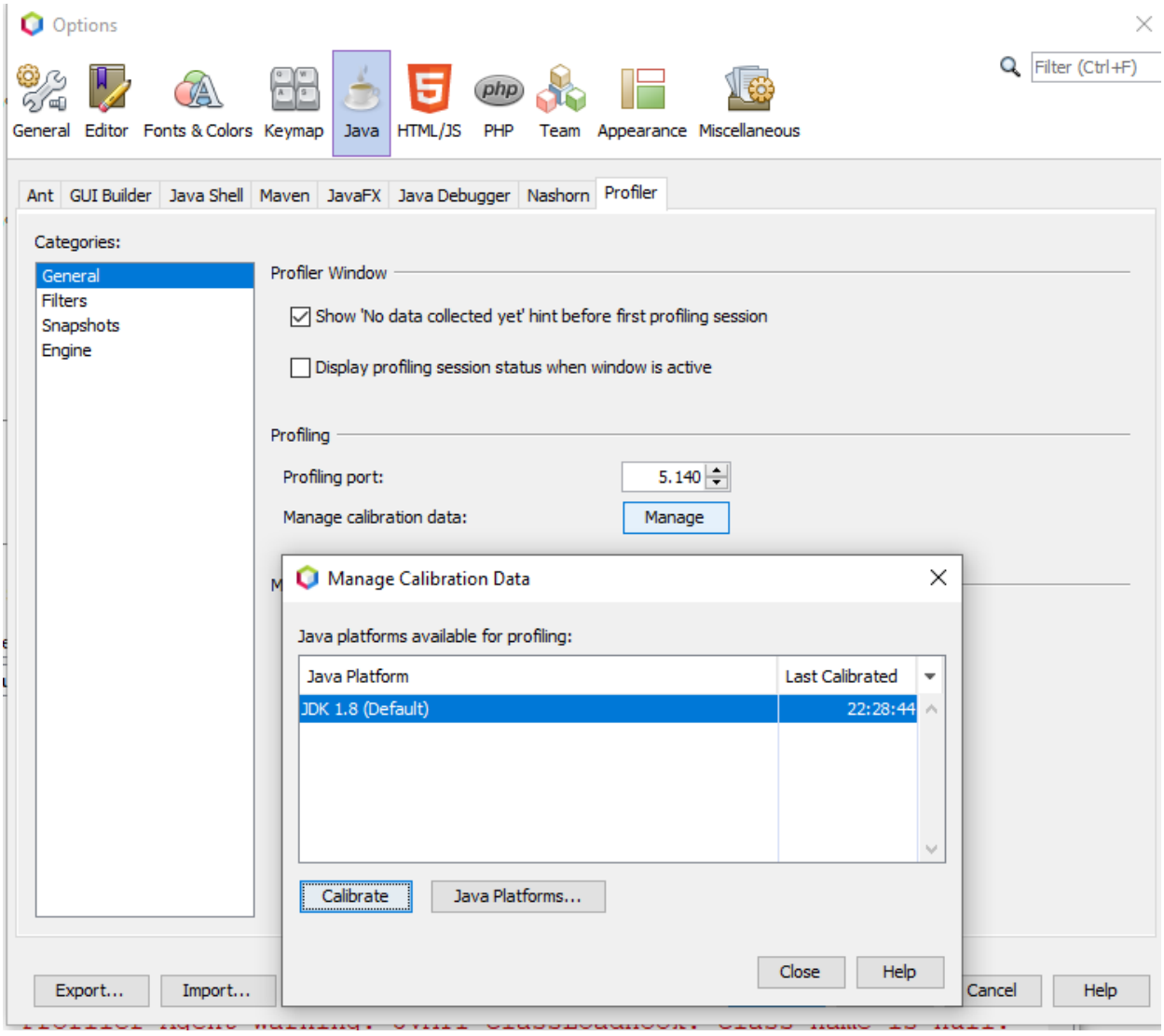
Neste documento utilizarase como proxecto *AnagramGame* de Java que se subministra como exemplo con NetBeans. Cando se executa este proxecto ábrese unha ventá cunhas letras descolocadas e o xogador deberá de acertar a palabra correcta formada con esas letras. Por exemplo, na seguinte execución acertaríase coa palabra *ambiguous*:



Calibrado inicial do equipo

A primeira vez que se fai unha análise de rendemento, o IDE avisa da necesidade de realizar a calibración inicial do equipo e a plataforma Java para que os resultados sexan precisos. Tools->Options_> Java-> Profiler-> General-> Manage calibration data e prmer no botón de Manage, onde sae unha ventá na que hai que escoller a versión de Java

Prémese en Calibrate para empezar a realizar a calibración:



Sobre a frecuencia dinámica da CPU haberá que consultar a información sobre a CPU do equipo. Por exemplo, na páxina de Intel:

http://www.intel.com/support/sp/processors/sb/CS-029908.htm aparece:

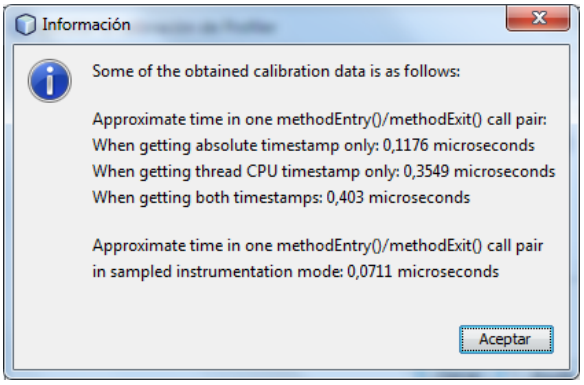
"¿Qué es frecuencia dinámica y cómo funciona?

Frecuencia dinámica funciona muy similar a la de la tecnología Intel Turbo Boost en forma dinámica que proporciona un impulso al desempeño de gráficos cuando se están realizando las tareas con uso intensivo de gráficos. Como la tecnología Intel Turbo Boost, debe haber margen de ampliación de energía y temperatura disponible en orden para que funcione.

¿Cómo puedo activar frecuencia dinámica?

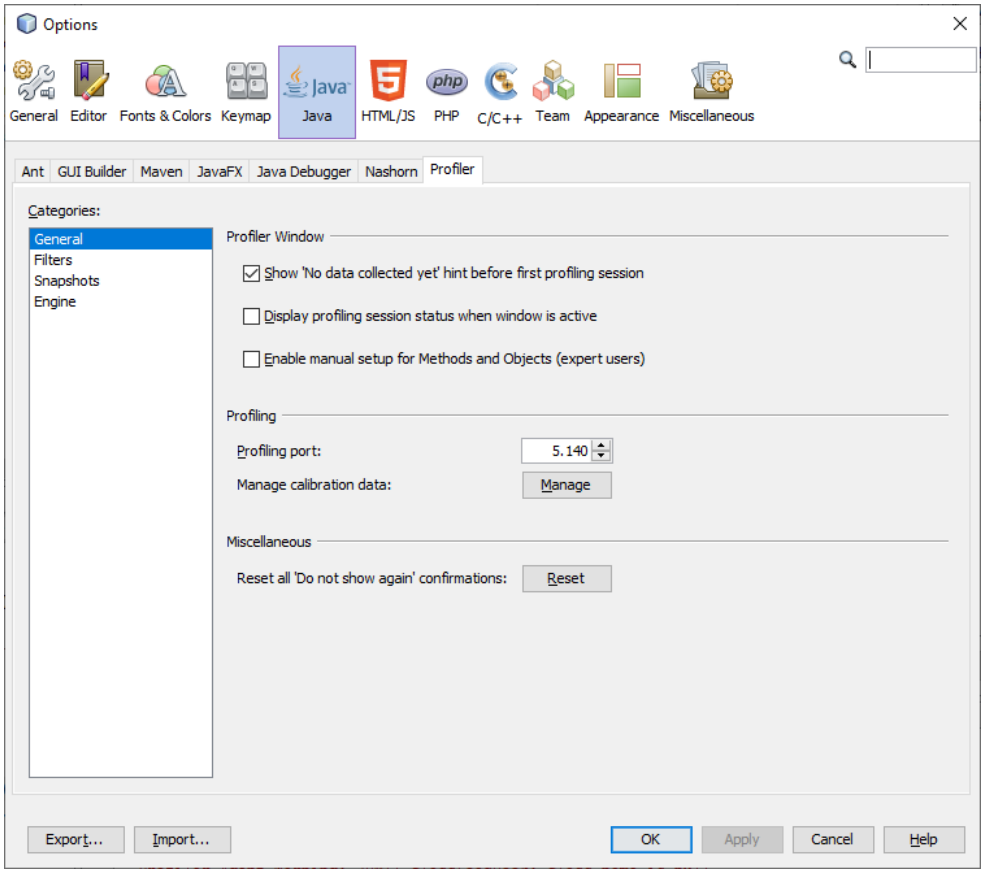
Frecuencia dinámica se activa automáticamente en la mayoría de los sistemas, por lo que no se requiere la intervención del usuario. "

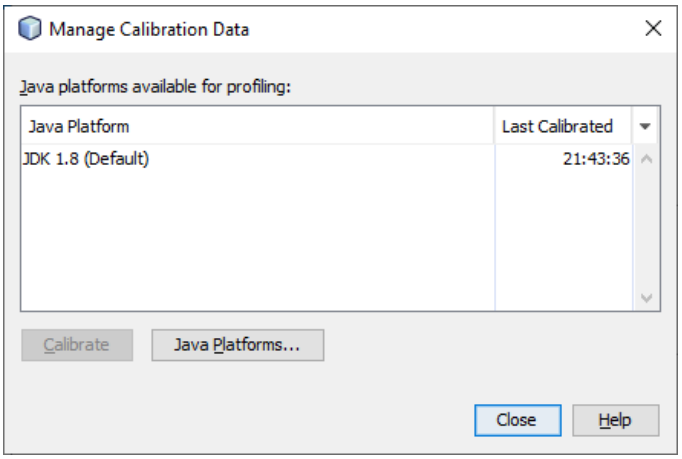
Pódense ver os resultados da calibración premendo en *Show Details*:



Normalmente a calibración só debe de realizarse unha vez. Sen embargo, deberá de volver a realizarse se se fan cambios substanciais na configuración da máquina que afecten ao rendemento da mesma. Para volver a realizar a calibración elíxese no menú principal:

Tools->Options->pestaña Profiler-> Categories: General-> Profiling-> Manage calibration data:



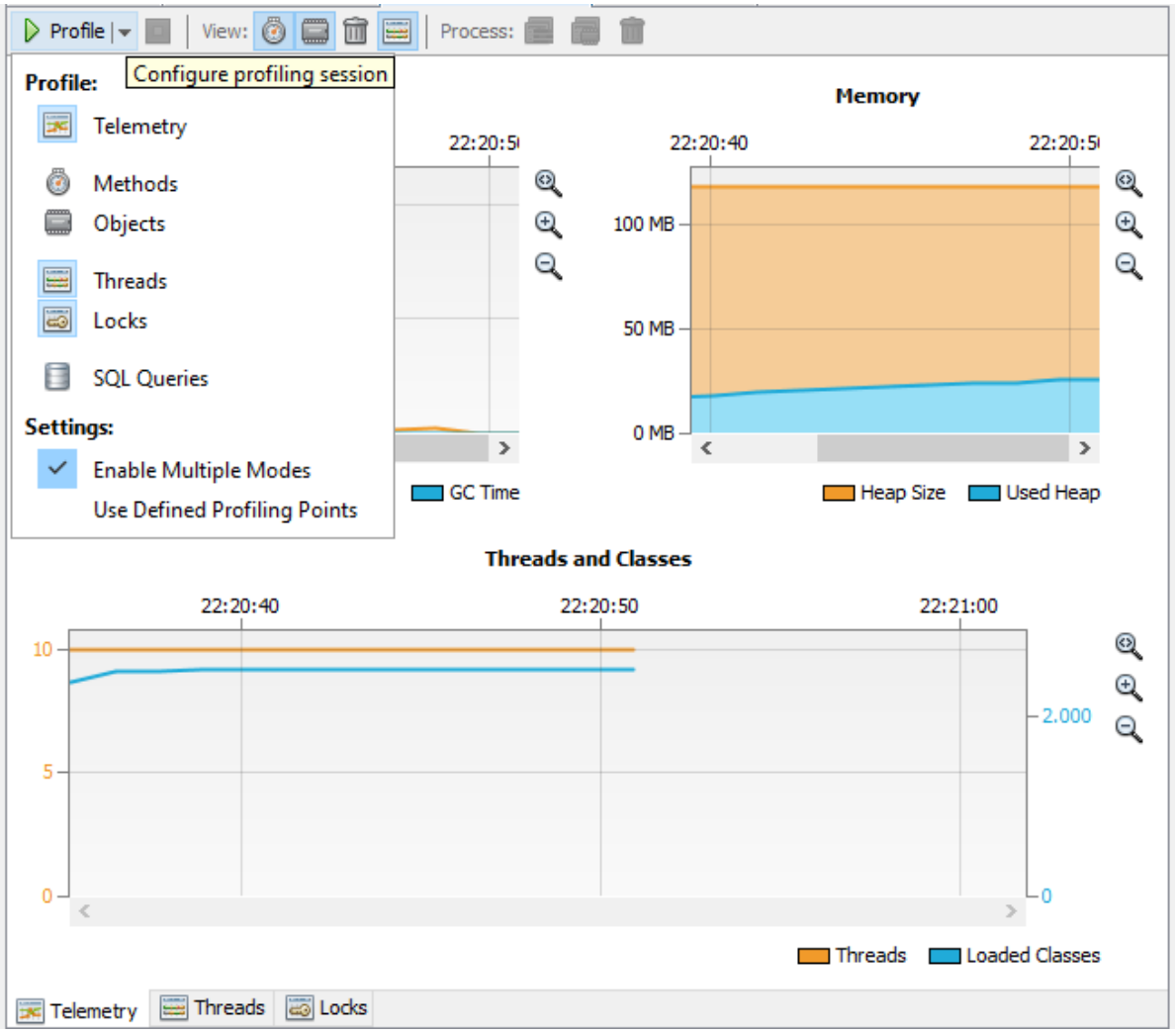


Selecciónase a plataforma Java a calibrar e prémese en *Calibrate*.

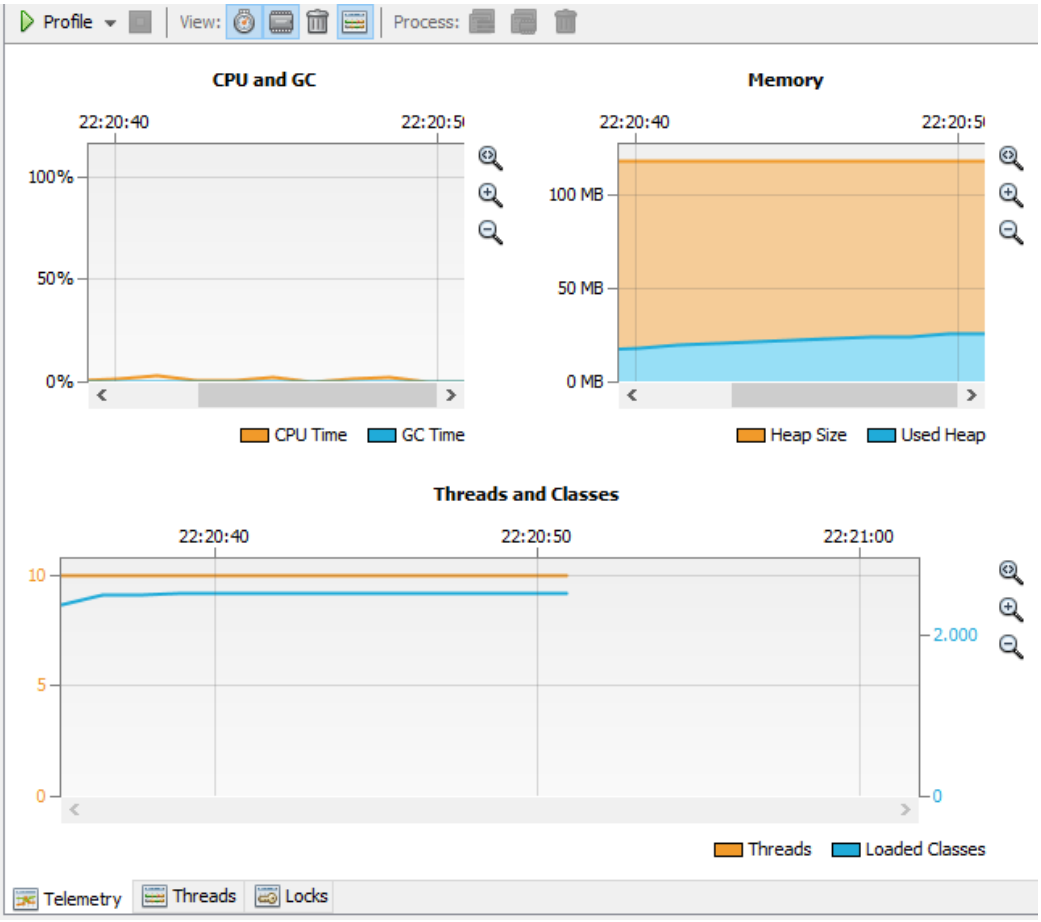
Sesión de análise

Para realizar unha análise dinámica de código Java con NetBeans hai que seguir os seguintes pasos:

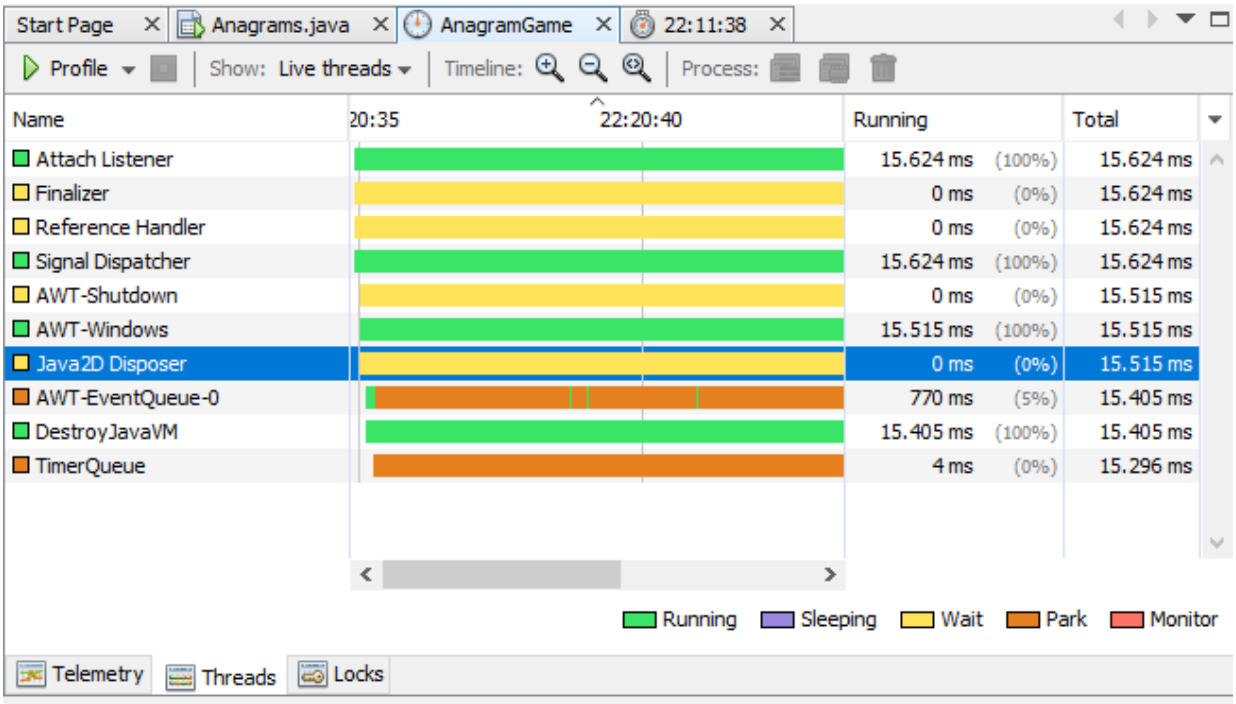
- Iniciar sesión de análise elixindo a opción *Profile->Profile Project* do menú principal ou facendo clic dereito sobre o nome do proxecto na ventá de proxectos e elixindo *Profile*.



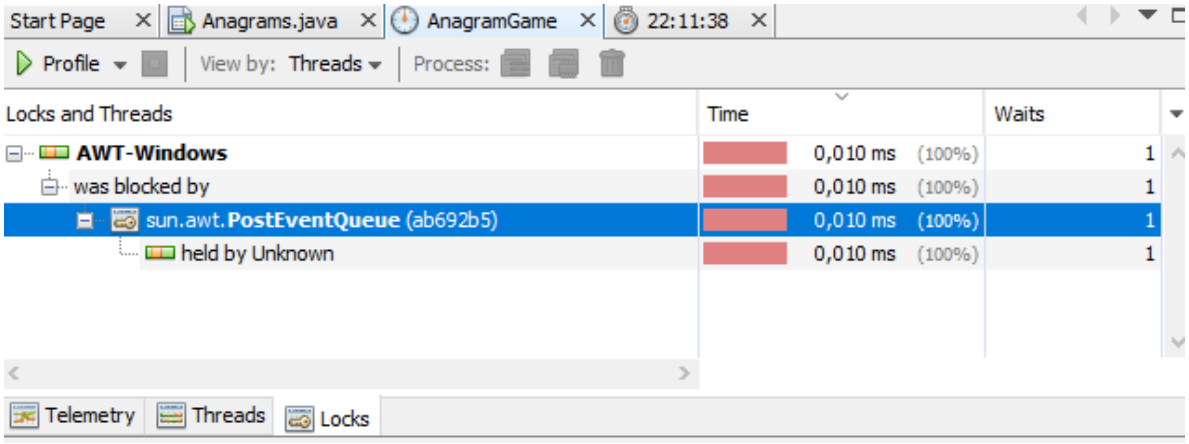
- Elixir a tarefa de análise a realizar:
  - ✓ Telemetría: NetBeans subministra unha serie de tarefas predeterminadas que normalmente son suficientes e son: análise do rendemento da CPU e o colector de basura (*CPU and GC*), análise da memoria (*Memory*) e os fíos e as clases (Threads and Classes)



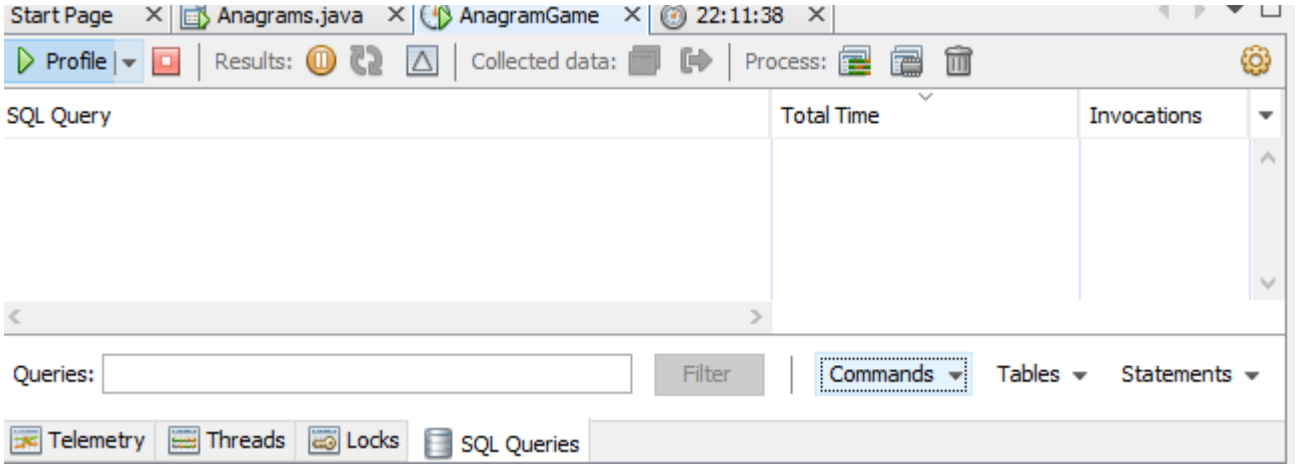
✓ Fíos: Mostra a actividade dos fíos (Un fío é un anaco de código do noso programa que pode ser executado ó mesmo tempo que outro..)



✓ Bloqueos: amosa os fíos bloqueados e os monitores de bloqueo.



✓ Consultas SQL: en caso de habelas, habería que indicalas

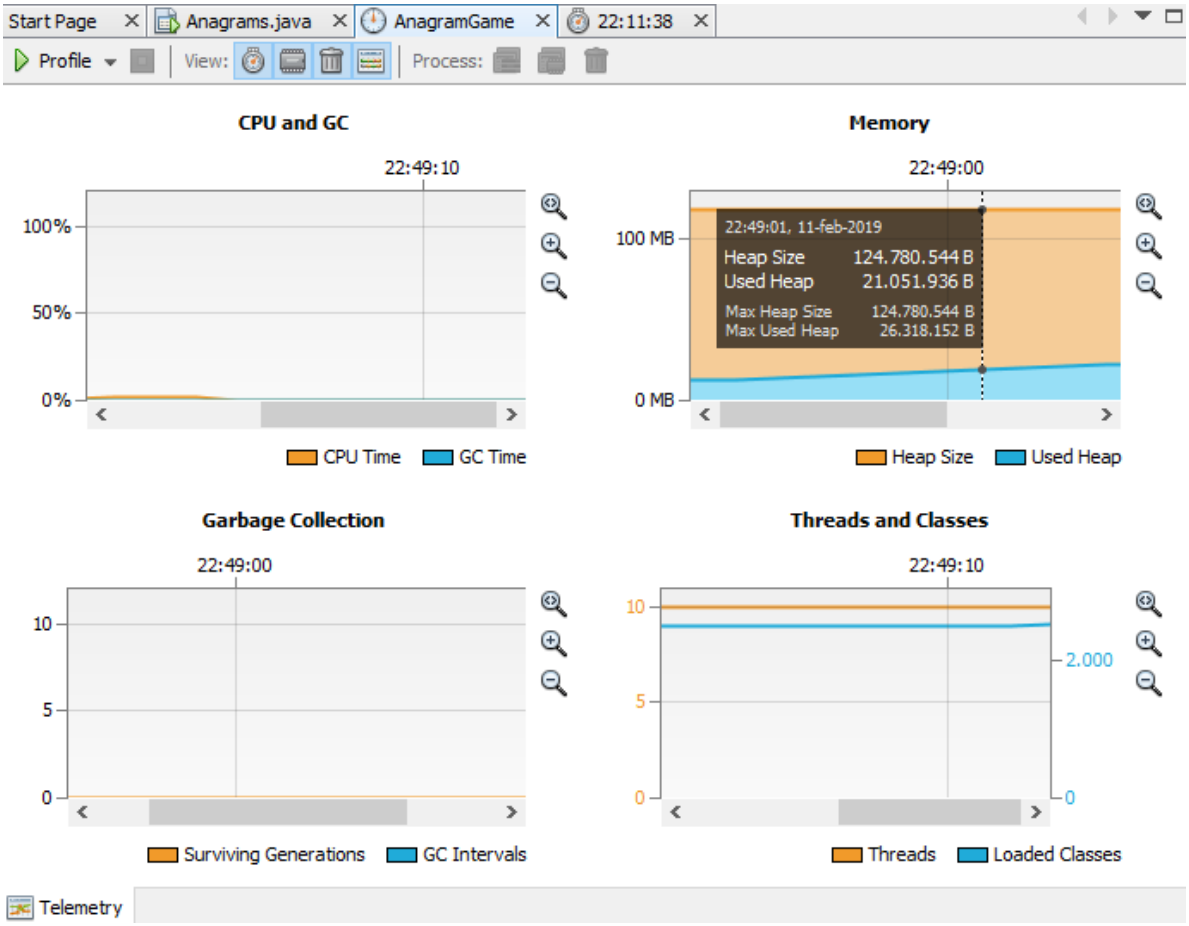


- Executar a tarefa de análise. Isto ten como consecuencia a execución do código Java, a apertura da ventá *Analizador* e a xeración de resultados
- Finalizar a sesión de análise porque finaliza a execución do código Java ou forzar a finalización da análise premendo na icona  .

Ventá de Telemetry

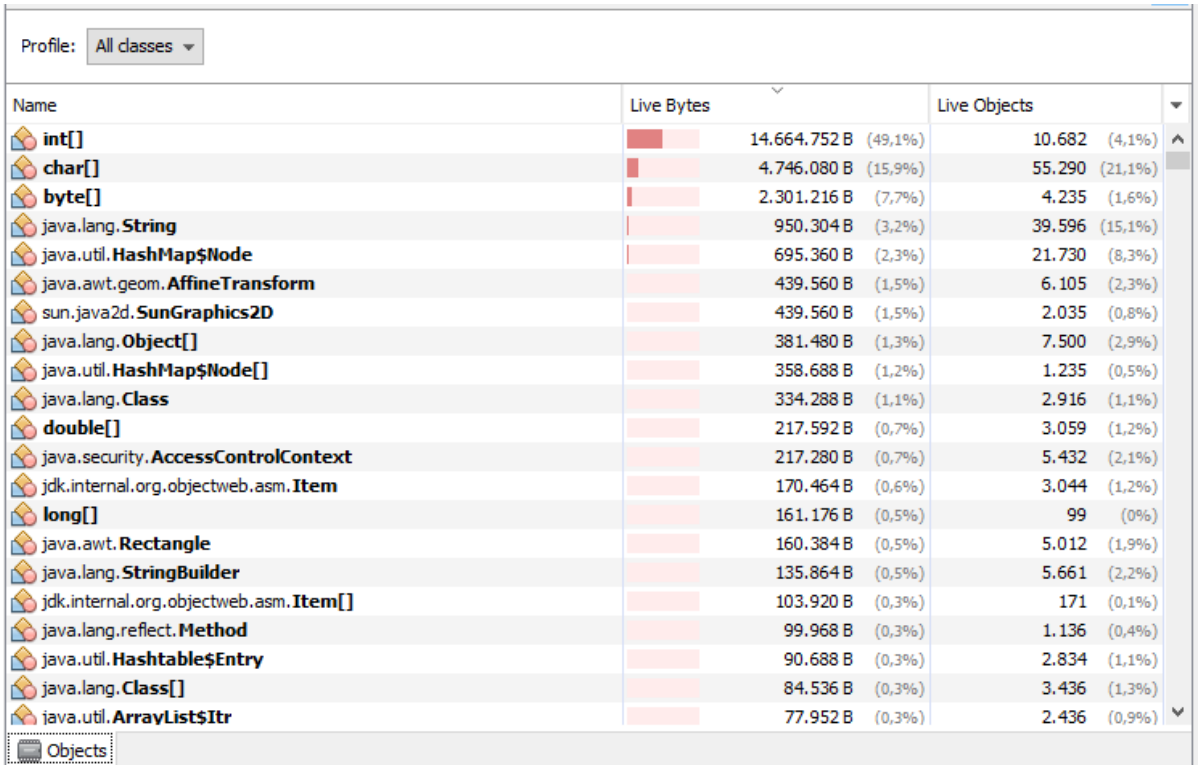
Esta ventá de Telemetry pode ser vista en calquera das diferentes tarefas de análise e permite ter unha vista dos datos de monitorización en tempo real: utilización da CPU e memoria, xeracións vivas (para detectar posibles perdas de memoria) e tempo de uso do GC, clases cargadas e fíos.

Pódese colocar o cursor sobre unha das gráficas, para ter máis información en forma de texto e pódese facer dobre clic sobre unha das gráficas, e abrirase na pestana Telemetry a mesma gráfica pero con máis detalle no tempo.



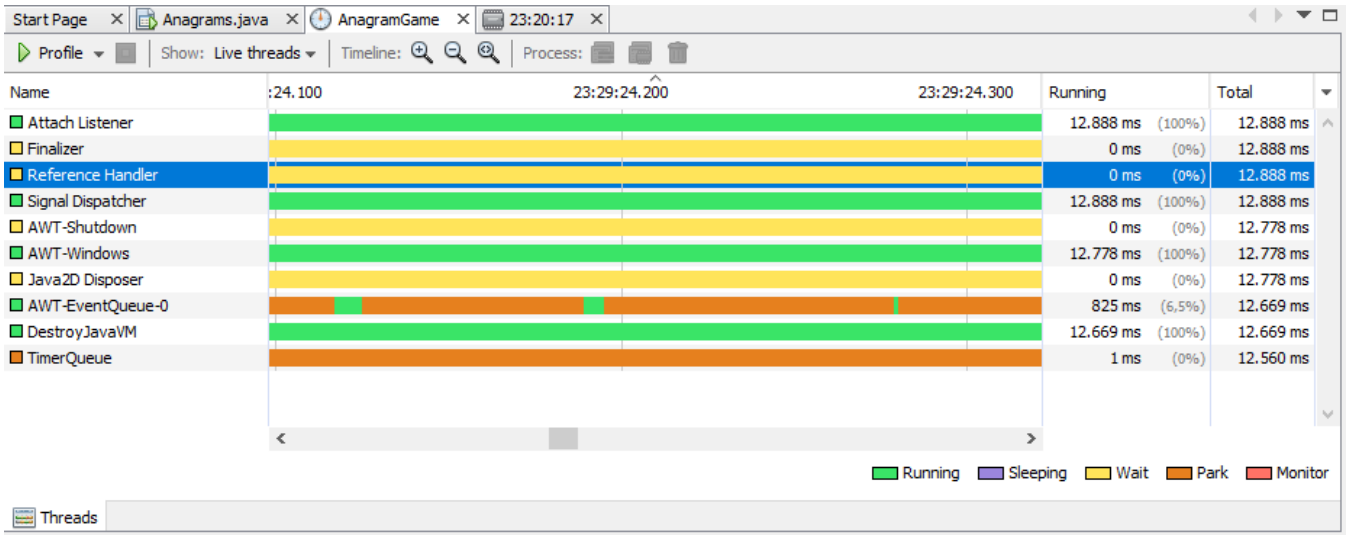
Obxectos

Serve para definir o tamaño e o número de obxectos asignados, incluídas as rutas de asignación.



Threads o subprocesos

Permítenos ver os fíos vivos, os terminados, uns escollidos polo usuario ou todos. Se se amplía o período de observación, pódese ver os diferentes estados polos que pasa (running, sleeping, wait, ...)



 Tarefa 3. Exercicios para practicar co análise de código Java en NetBeans

1.3 Tarefas

As tarefas propostas son as seguintes.

- **Tarefa 1**. *Procura de analizadores estáticos de código no mercado e comparativas.*
- **Tarefa 2**. *Procura de analizadores dinámicos de código no mercado e comparativas.*
- **Tarefa 3**. *Exercicios para practicar co análise de código Java en NetBeans*.

1.3.1 Tarefa 1. Procura de analizadores estáticos de código no mercado e comparativas

1.3.2 Tarefa 2. Procura de analizadores dinámicos de código no mercado e comparativas

1.3.3 Tarefa 3. Exercicios para practicar co análise de código Java en NetBeans

- Abre o proxecto Java de exemplo do NetBeans AnagramGame
- Executa a opción *Profile/Profile Project (AnagramGame)* coas opcións de análise por defecto
  - Toma unha *instantánea* do rendemento e fai unha copia da pantalla, pega nun documento de Open Office.

  - Garda esta instantánea. Aparece a lista de instantáneas que se tomaron do rendemento do programa en distintos momentos.
  - Fai unha análise dos métodos e mostra os resultados en vivo (Hot spots)
- Executa a opción Profile/Profile Project (AnagramGame) con monitoreo de subprocesos. Fai unha copia da pantalla.
- Executa a opción *Profile/Profile Project (AnagramGame)* analizando parte da aplicación, fixando os métodos raíz (do arquivo StaticWordLibrary.java):
  - public int getSize()
  - public boolean isCorrect(int idx, String userGuess)

  ¿Aparecen resultados de rendemento? ¿Por qué? ¿Qué hai que facer para que aparezan?

  Cando consigas que aparezan resultados do rendemento fai unha copia da pantalla do análise en vivo gárdaa no documento de OpenOffice xunto ca explicación as preguntas anteriores
- Análise de memoria
  - Seleccionar Profile - > Profile Main Project do menú principal.
  - Seleccionar Object.
  - Garda dúas instantáneas deste análise de memoria.
  - Fai a comparación entre as dúas instantáneas e cando se mostre o resultado fai unha copia da pantalla.