



Dos viajes al centro de la JVM

CRaSH & YouDebug

{CODE}MOTION
MADRID · NOV 27-28 · 2015

José Miguel Rodríguez
@jmiguel

whoami --asPerson --verbose



José Miguel Rodríguez



@jmiguel

email: me@jmiguel.eu

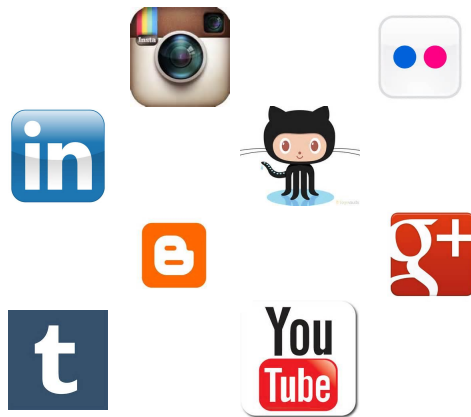
Blog Personal: <http://www.jmiguel.eu>

Toda la info social: <http://about.me/jmiguel.rodriguez>

Guitarrista de Jazz, Rock y Blues

Fotógrafo wannabe

Aerotrastornado (<http://www.aerotrastornados.com>) (Gracias M\$)



whoami --asDeveloper



Cofundador de **Virtual Software**

<http://www.virtualsw.com>

Programando desde 1992

GwBasic → Clipper → VisualBasic (todos tenemos un pasado) → Java → Groovy

BBS → Primeros ISP (¿alguien recuerda Infovía?) → Seguridad → AWS



Esta presentación...

<http://es.slideshare.net/jmiguelr/codemotion-2015-crash-y-youdebug>

en breve en <http://www.jmiguel.eu>

El video estará (¡espero!) disponible en el canal de MadridGUG

Habrà enlaces «sticky» por un tiempo en Twitter: **@jmiguel**



Dos Partes



Shell Interactivo



Debugger Automático

60%

40%

Shell Interactivo para la JVM

Creado por Julien Viet (@julienviet)
OpenSource

<http://crashub.org>

<https://github.com/crashub/crash>

```
git clone git@github.com:crashub/crash.git
```



Última versión 1.3.1

Requiere Java 6+ / Groovy 1.7+

Instalación:

Descarga de fuentes de github + *mvn package*

Binarios desde <http://crashub.org>

sdkman / gvm

Plugins para varios frameworks



¿Que significa «*conectar a una JVM*»?

«Entrar dentro de la misma mediante **telnet** o **ssh**, como si fuera otro ordenador, ver lo que está pasando y ejecutar comandos desde dentro. »

Es un interface de línea de comandos (**CLI**) para la JVM

Posibilidad de embeber CRaSH en nuestra aplicación, tanto escritorio como web (.jar o .war)

Tenemos comandos predefinidos pero podemos crear propios, en Java o en Groovy (probablemente más en el futuro)

Extrema facilidad para crear nuestros propios comandos

Por cierto....

!(Groovy)?



- ✓ Lenguaje en la JVM, parecido a Java (curva de aprendizaje muy suave)
- ✓ Hace la programación mucho más fácil, cómoda y legible
- ✓ Lenguaje dinámico, compilación en tiempo de ejecución (o no: @CompileStatic)
- ✓ Permite hacer scripts: no hay necesidad de crear una clase para todo.
- ✓ Closures: código como un tipo de datos (\pm Lambdas en Java 8)



- ✓ ¿Interesado?. Excelente introducción by Pablo Alba en Youtube - MadridGUG
- Busca «**Porque Groovy no es Java sin punto y coma**»
- ✓ ¿Más interesado?. Meetup de MadridGUG



Los comandos en CRaSH pueden cooperar siguiendo la filosofía de comandos Unix:

*Programas pequeños que hagan solo una cosa y la hagan bien.
Capaces de trabajar juntos usando texto como interfaz universal*
Ken Thompson

Diseño modular, es posible activar sólo lo que nos interese al embeberlo en nuestra aplicación.

- ✓ Conectores **ssh**, **telnet** , **crash.js** usando websockets

Plugins de autenticación

- ✓ ssh key
- ✓ simple (usuario / password)
- ✓ Extensible

Modos de operación

- ✓ **Standalone:** Ejecutamos CRaSH en solitario para crear comandos, por ejemplo. Utilidad limitada
- ✓ **Attach:** Nos conectamos a un PID de JVM local
- ✳ **Embedded** ← Nos conectamos por ssh/telnet a una JVM remota

Otras cosas que molan

- ✓ Autocompletado estilo bash de:
 - Comandos
 - Nombres de fichero, nombres de clases...
- ✓ Renders para tipos de datos
- ✓ Modificación de la salida de datos. Por ejemplo: colores, tablas...

- ✓ Útil para primera toma de contacto
- ✓ Limitado a que ejecutar todo en la misma máquina física.
- ✓ En modo Attach los comandos se ejecutan en la JVM controlada
- ✓ Conectamos desde CRaSH a otra JVM dentro de nuestra misma **máquina local**.
- ✓ No tenemos que **tocar nada** en nuestro programa. PID
- ✓ Podemos crear nuevos script y ejecutarlos ¡sin compilar!.
(*Groovy Power!*)

- ✓ Vamos a ver la estructura de carpetas
- ✓ Creación de nuevos scripts
- ✓ Ejemplo 1: Hola mundo
- ✓ Ejemplo 2: Comando de ejemplo cmotionCommands

<demo mode="on">

Modo Embebido

- ✓ Nos conectamos a una máquina remota por **telnet** o **ssh**
- ✓ Tenemos que haber embebido CRaSH previamente
- ✓ Aplicaciones de escritorio: **jar** provisto
- ✓ Aplicaciones web: **jar** o **war**

<demo mode="on">

Modo Embebido - Aplicación Web

- ✓ Es igual que el modo embebido, pero ahora tenemos un interface desde el navegador
- ✓ Sólomente tenemos que desplegar el war provisto -o construirlo- y desplegarlo en nuestro servidor
- ✓ Es posible incluir nuestros comandos creando un war a medida e incluyendo en él los jar que necesitamos
- ✓ El contexto de despliegue también es configurable en web.xml

Modo Embebido - Aplicación Web

- ✓ El contexto de despliegue también es configurable en web.xml
- ✓ O bien lo embebemos como listener en nuestra webapp

```
<web-app>  
  <listener>  
    <listener-class>org.crsh.plugin.WebPluginLifecycle</listener-class>  
  </listener>  
</web-app>
```

<mini-micro-demo mode="on">

>CRaSH_

</CRaSH>

YouDebug

Debugger no interactivo
(un proyecto tan modesto que ni siquiera tiene logotipo)



Kohsuke Kawaguchi

YouDebug es un depurador **no interactivo** escrito en Groovy que nos puede ayudar en ciertas situaciones.

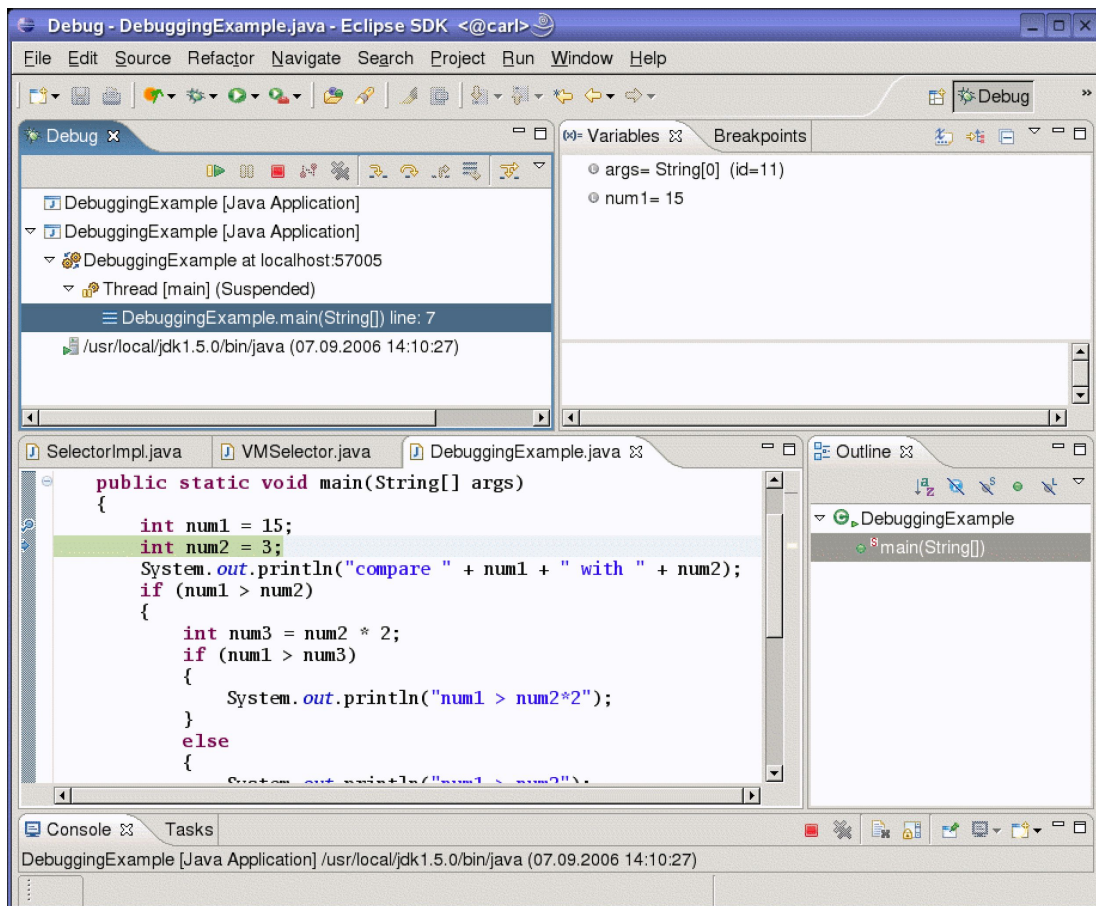
- ✓ Sistema desplegado en producción
- ✓ Encontrar bugs de difícil reproducción en nuestro entorno
- ✓ Usa el **Java Debugger Interface** para ejecutarse junto con la aplicación destino
- ✓ <https://youdebug.kenai.com/>

¿Como hacemos debug?

println() nadie, ¿verdad?

Depurando «en casa»

- ✓ IDE
- ✓ Fuentes disponibles
- ✓ Breakpoints
- ✓ Evaluación de variables
- ✓ Evaluación de condiciones
- ✓ Asignación de variables



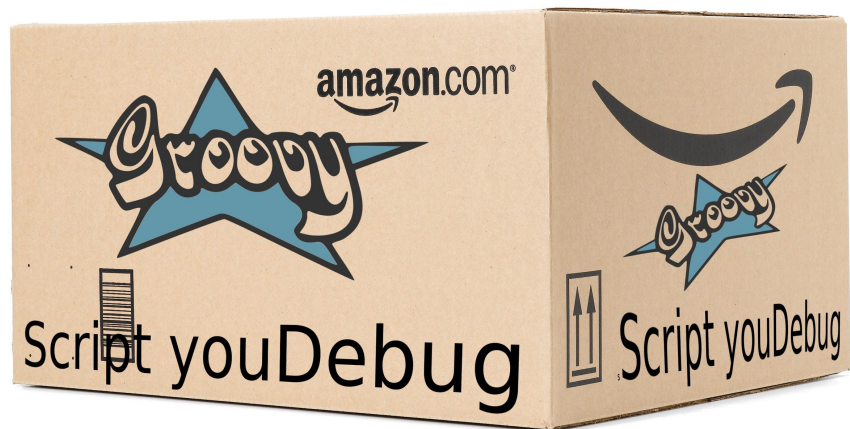
¿En producción?

- ✓ Llama al cliente. ¡Tenemos un bug!
- ✓ El cliente está lejos. O en un búnker
- ✓ No conseguimos reproducir el error en local. Los datos del cliente son privados. No nos los deja para probar.
- ✓ No le podemos mandar los fuentes (NDAs) y decirle que depure el.



Solución

Enviar un script para que haga nuestro trabajo



Entorno del cliente

- Base de datos (y datos)
- Sistema Operativo
- Versión de JDK
- Memoria

Receta de uso

1 - Ejecutar la aplicación en modo debug

```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5005
```

2 - Ejecutar nuestro script de YouDebug

```
java -jar youdebug.jar -socket 5005 nuestroScript.ydb
```

¿Qué podemos hacer con YouDebug?

- ✓ Definir **breakpoints**
- ✓ Evaluar expresion
- ✓ Obtener valores y modificar **variables**
- ✓ Listar **threads** y obtener stack **traces** de los mismos
- ✓ Obtener un heap dump
- ✓ Monkey Patching (no recomendable... pero divertido)
- ✓ No tener que esperar que pase algo por 1m, 1d, 1M ...

¿Cómo definimos un BreakPoint?

- ✓ Cuando llegamos a una determinada **línea de código**
- ✓ Cuando salta una **exception**
- ✓ Cuando se hace **referencia** a una variable o cambia su valor
- ✓ Cuando una clase es **cargada / descargada**
- ✓ Cuando se **entra o sale** de un **método**
- ✓ Cuando un **thread** es **creado o destruido**

Heisenbug

En jerga de **programación**, un **heisenbug** es un tipo de **bug** que parece desaparecer o comportarse de otro modo al intentar ser observado en detalle.¹ El término es un **juego de palabras** a partir del nombre de **Werner Heisenberg**, el **físico** que dedujo el efecto de observación de la **mecánica cuántica**, según el cual el mero hecho de observar un sistema de una manera determinada altera el estado de este.

... a veces **System.out.println()** puede ser la mejor opción. Pero la siguiente es sin duda YouDebug: introduce poca variación.

Scripts mínimos

```
vm.methodEntryBreakPoint( "eu.jmiguel.myClass", "main",  
    { method -> println ("Entramos en main")})
```

```
vm.breakpoint( "eu.jmiguel.myClass", 12) {  
    println ("Estamos en la linea 12 de myClass")  
}
```

A tener en cuenta

- ✓ La ejecución de consultas a variables desde nuestro script son, en general, contra un **proxy** de las variables. Se puede forzar a que sea contra las variables de la otra JVM
- ✓ Si nuestro script lanza una exception podemos dejar **inestable** la JVM remota

<demo mode="on">

</demo mode>

(Agradecimientos a **Wolfgang Schell** @jetztgradnet)



That's all Folks!



¿Preguntas?

(las que dé tiempo)

YouDebug

¿Quieres más información?

«La única pregunta estúpida es la que **no** se hace»

Twitter: @jmiguel

email : me@jmiguel.eu

Cuestionario (4 preguntas, 15 segundos en contestar)

Please! <https://goo.gl/2wqoze>