

# Quebrando as Barreiras da Linguagem Java

Alessandro Binhara e Sandro Bihaiko

- ✓ Sun X Microsoft
- ✓ A Plataforma Java
- ✓ A Plataforma Mono
- ✓ IKVM
- ✓ Exemplo Prático
- ✓ Um Projeto Real



# Sun

## Uma Linguagem

- Padrão, Manutenção, Comunicação
- Foco na Empresa

# Microsoft

## Muitas Linguagens

- Facilidade, Aprendizado, Preferência
- Foco no Desenvolvedor

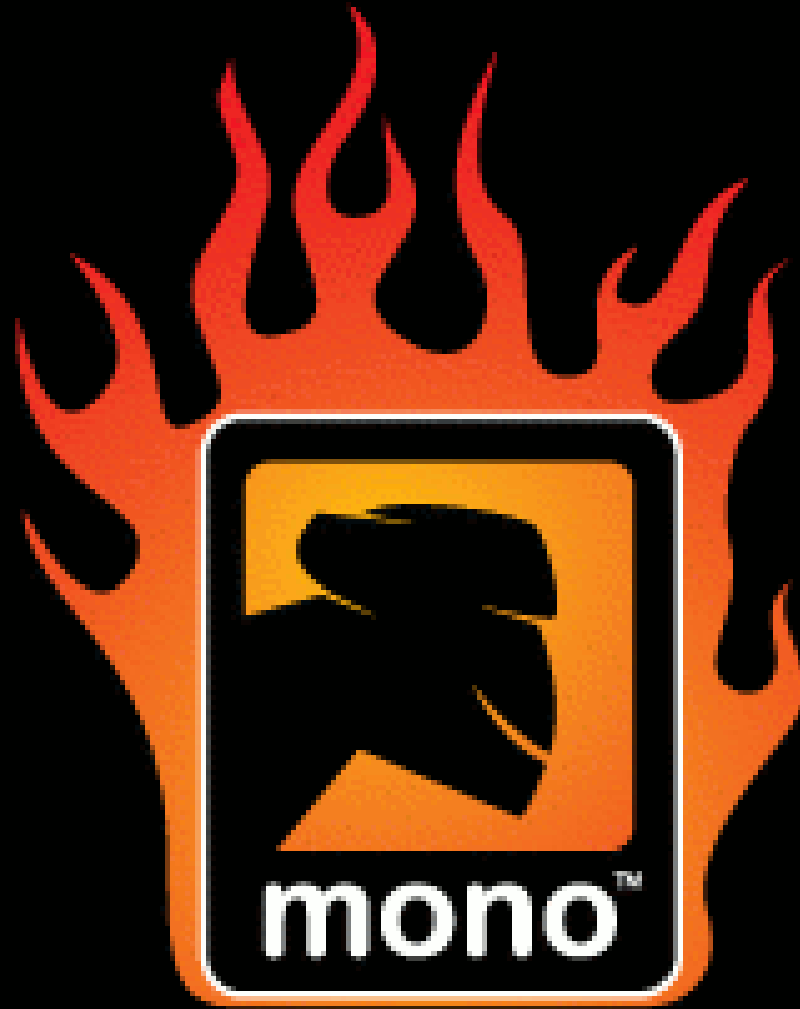


# Mono

## Muitas Linguagem e Muitas Plataformas

- Aproveita todas as características do mundo OpenSource
- Grande acesso a documentação JAVA e .NET
- Diversos Centros de capacitação já constituídos
- Comercialmente já aceito
  - *“Existe alguma grande empresa que de suporte?”*

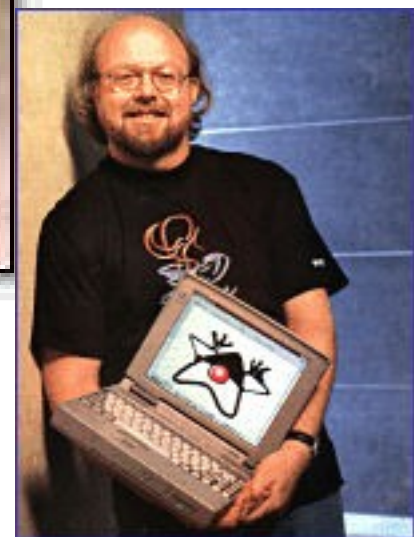




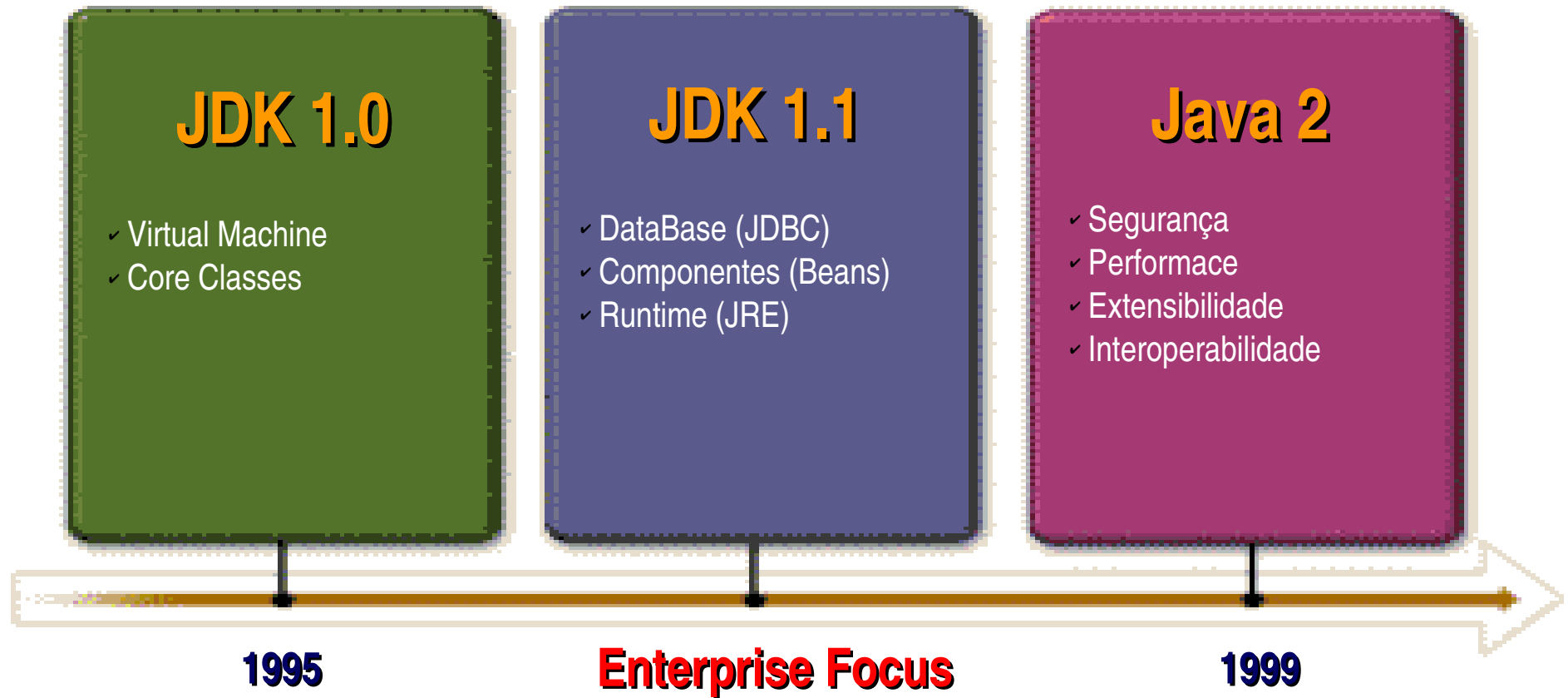
A Plataforma Java

- ✓ Green Oak (1991)
- ✓ \*7 (1992)
- ✓ Internet!!!!
- ✓ HotJava (1995)
- ✓ Netscape 2.0 Java enabled (1996)
- ✓ Primeiro Release Java 1.02 (1996)





# Evolução da Plataforma







Java Technology  
Enabled Devices



Java Technology  
Enabled Desktops



Workgroup  
Server

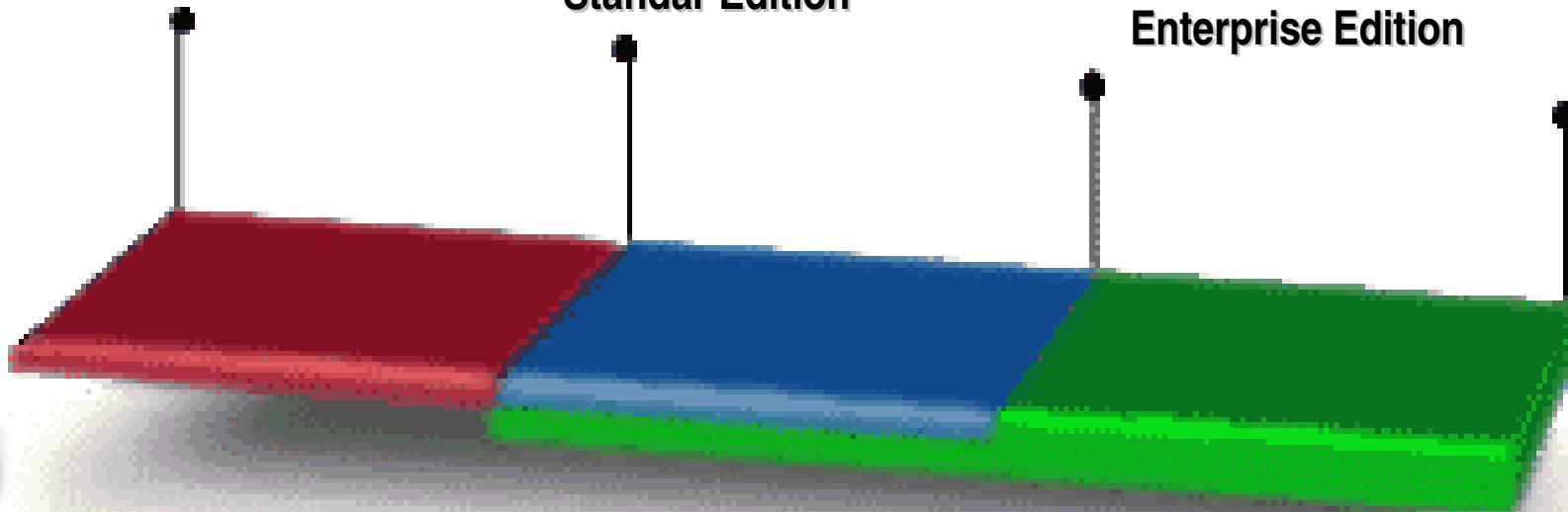


Hi-End  
Server

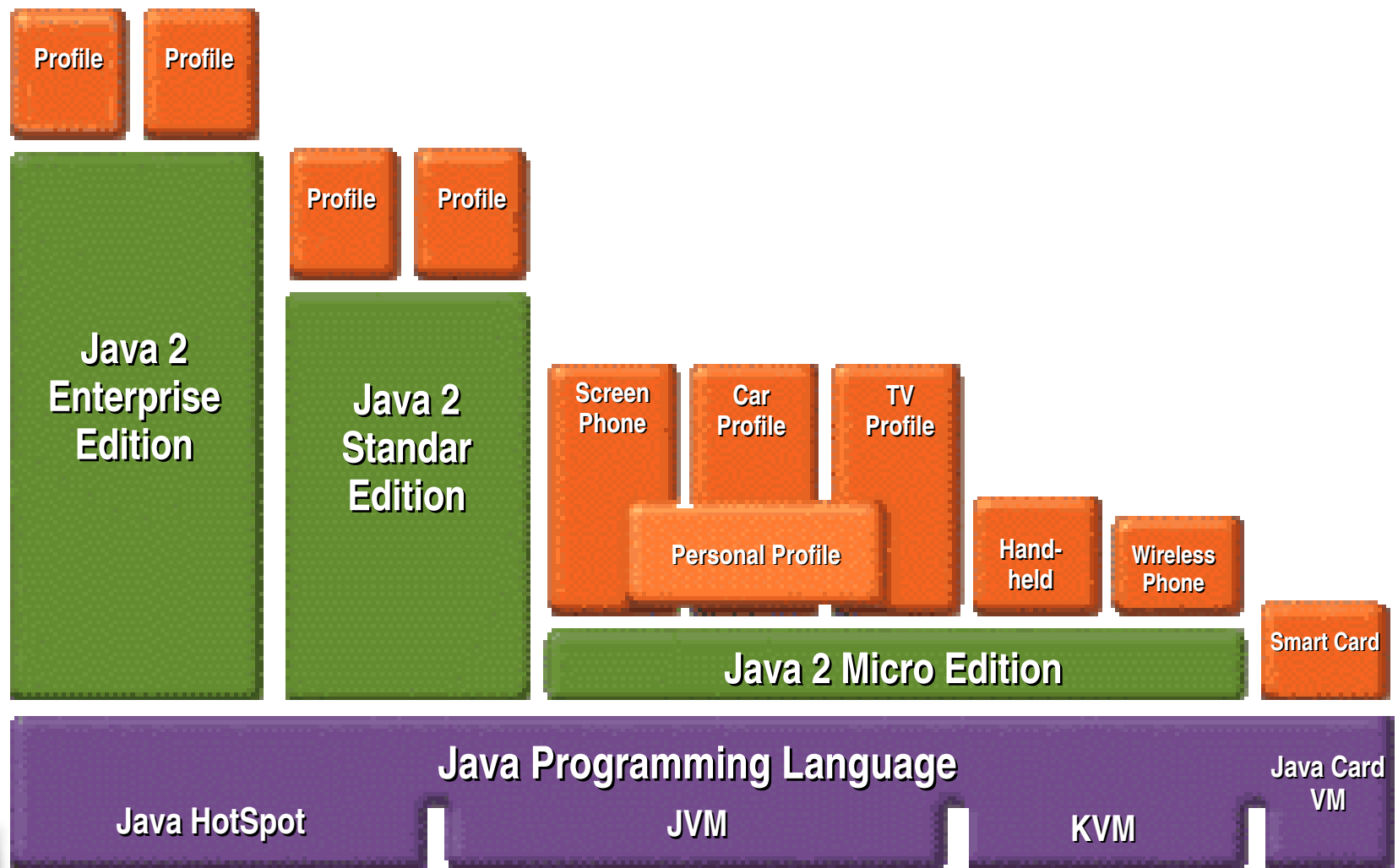
Micro Edition

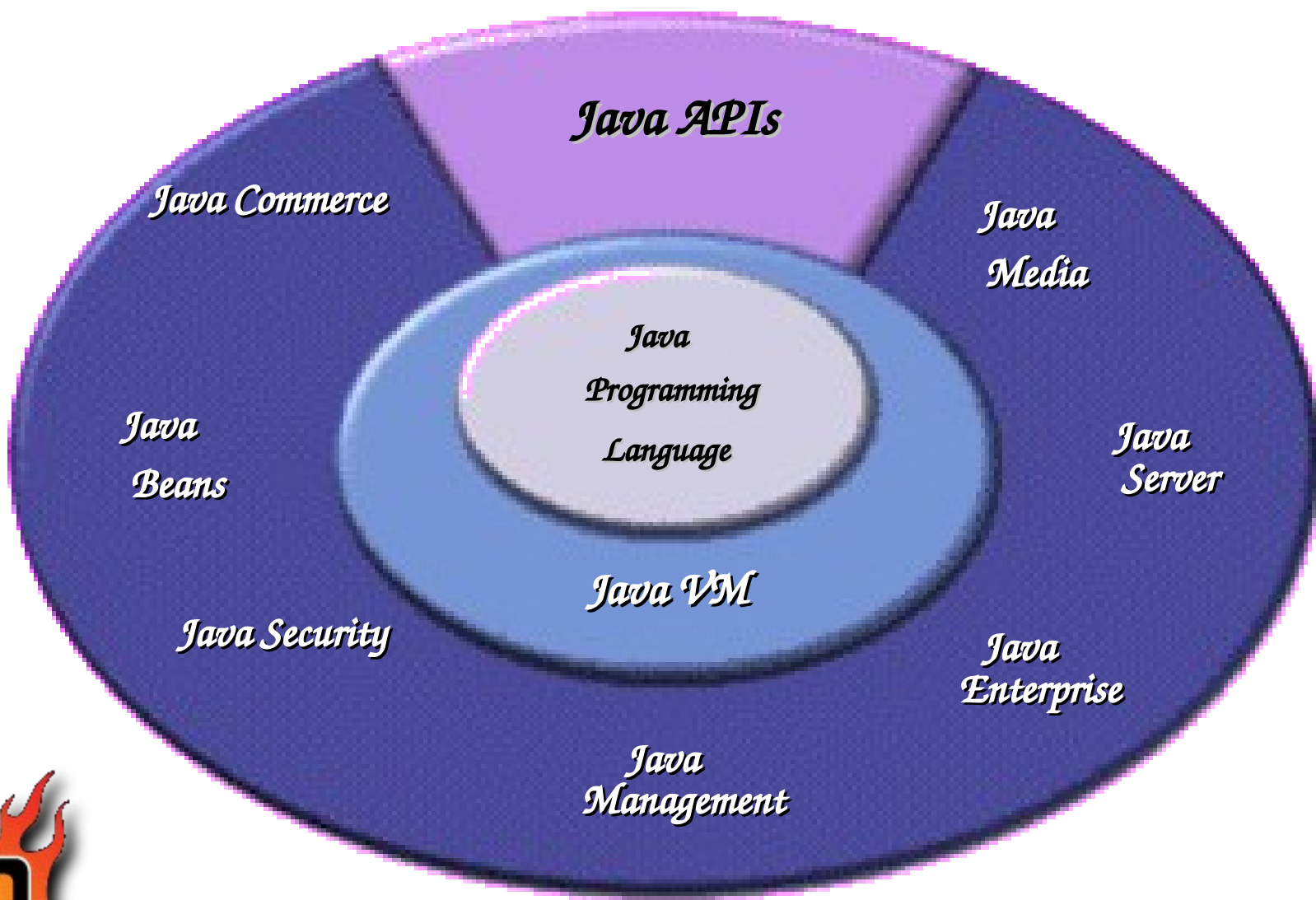
Standar Edition

Enterprise Edition



# A Plataforma Java





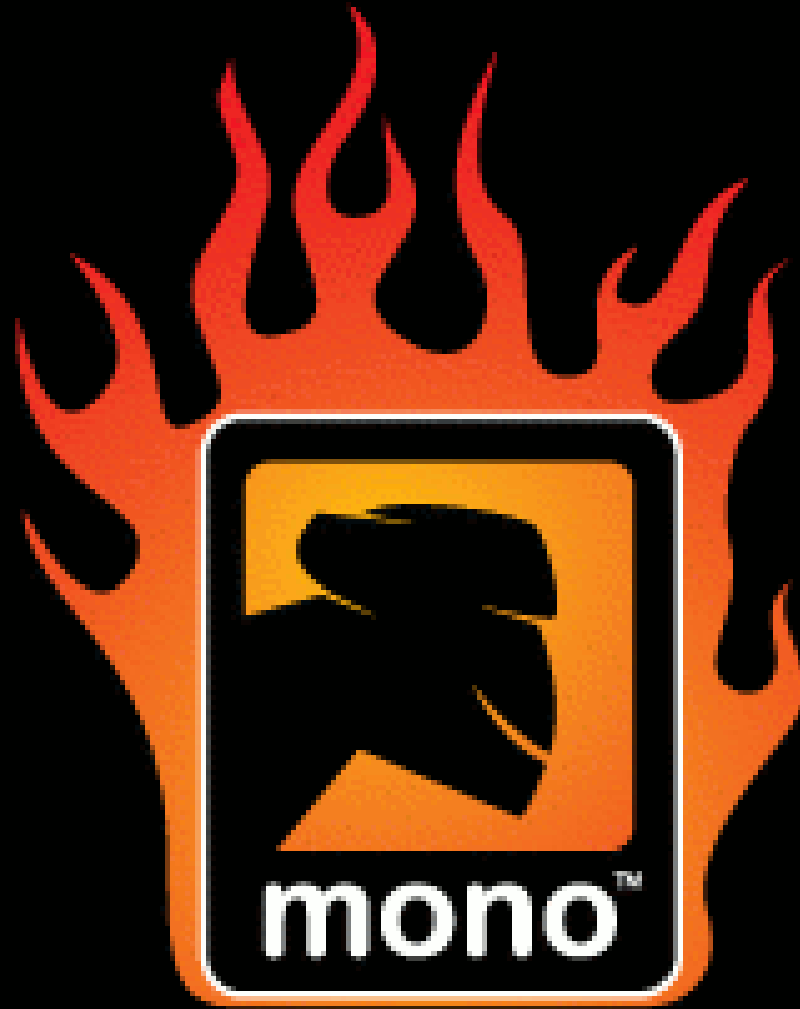
## ✓ Padrões

- ✓ Especificação da linguagem Java - [java.sun.com/docs/books/jls](http://java.sun.com/docs/books/jls)
- ✓ Especificação da JVM - [java.sun.com/docs/books/vmspec](http://java.sun.com/docs/books/vmspec)
- ✓ Especificações da plataforma J2EE - [java.sun.com/j2ee](http://java.sun.com/j2ee)

## ✓ Implementações

- ✓ [www.java.sun.com](http://www.java.sun.com) - implementação dos padrões Java
- ✓ [www.ibm.com](http://www.ibm.com) - implementação dos padrões Java
- ✓ [www.gnu.org/software/classpath](http://www.gnu.org/software/classpath) - implementação livre do JavaCore
- ✓ [www.ikvm.net/](http://www.ikvm.net/) - implementação da JVM e suas classes em .NET e também ferramentas que permitem interoperabilidade entre Java e .NET





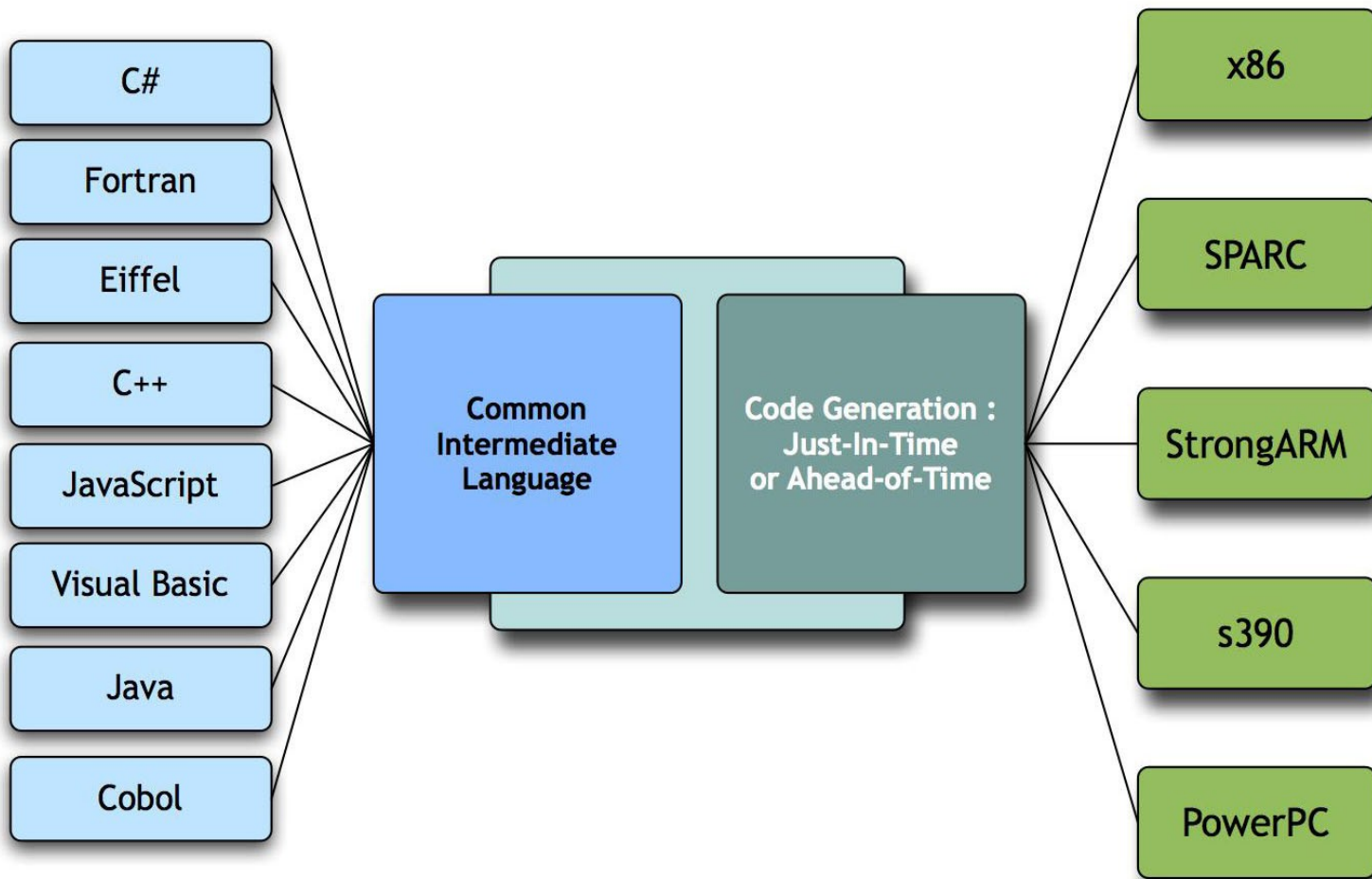
A Plataforma Mono

# O que é Projeto Mono

- Iniciador: Miguel de Icaza
- .Net é um Padrão ISO/ECMA
- Mono é uma Implementação do Padrão ECMA
- Hardware: Intel, PPC, S390, ARM, AMD64..



# MONO: Multi-Linguagem e Multi-Plataforma



# Características MONO

- Completamente escrito em licenças LGPL e BSD
- Integração completa com ambientes Linux e Windows
- Diversos toolkits gráficos: GTK, QT, Coca, Wxwindows, Windows.Forms
- Suporte completo a ASP.NET 1.0
- Suporte a Banco de Dados
- Todas as ferramentas da plataforma são escritas em C#
- Várias bibliotecas estendidas: Mozilla, OpenGL, Zip#, oogl#





# Linguagens Disponíveis

**Ada for .NET** – A# ([http://www.usafa.af.mil/df/dfcs/bios/mcc\\_html/a\\_sharp.cfm](http://www.usafa.af.mil/df/dfcs/bios/mcc_html/a_sharp.cfm))

**VisualBasic.NET** – mbas e bmcs (<http://www.mono-project.com>)

## **ANSI C**

- CSCC - <http://www.gnu.org/projects/dotgnu/pnet.html>
- ICC - <http://www.cs.princeton.edu/software/lcc/>

**C#** - mcs (mono-project) e csc (DotGNU)

**Eiffel** - Eiffel ENViSioN! (Eiffel Software)

**Fortran** - Lahey/Fujitsu Fortran for .NET (Lahey Computer Systems, Inc.)

## **Cobol**

- NetCOBOL - COBOL for .NET (Fujitsu -)
- Net Express (Micro Focus)
- KICKS for .NET™ (Intensity Software)

**Hasquel** - Hugs98 for .NET <http://galois.com/~sof/hugs98.net/>

## **Lisp**

- RDNZL - <http://www.weitz.de/rdnzl/>
- Foil - <http://foil.sourceforge.net/>
- L Sharp .NET - <http://www.lsharp.org/>



# Linguagens Disponíveis

## Logo

- TurtleTracks Logo - <http://www.mech.upatras.gr/~robgroup/logo/turtletracks/index.html>
- Mono Logo - <http://monologo.sourceforge.net/>

## Perl

- PerlSharp - <http://taubz.for.net/code/perlsharp/>

## LUA

- Lua.NET - <http://www.lua.inf.puc-rio.br/luanet/>

## Pascal

- TMT .NET Project
- RemObjects Chome - <http://www.chromesville.com/>

## JavaScript

- JScript .NET
- Jscript - (DotGNU)
- JANET - <http://janet-js.sourceforge.net/>

- **PHP**

- Phalanger
- IronPHP - <http://ironphp.sourceforge.net/>
- PHPSharp - [http://www.akbkhome.com/wiki.php/Projects/PHP\\_Sharp](http://www.akbkhome.com/wiki.php/Projects/PHP_Sharp)



# Linguagens Disponíveis

## Python

- IronPython - <http://www.ironpython.com/>
- Python for .NET compiler - <http://starship.python.net/crew/mhammond/dotnet/>
- Python for .NET - [http://www.zope.org/Members/Brian/PythonNet/index\\_html](http://www.zope.org/Members/Brian/PythonNet/index_html)

## Boo

- Liguagem Boo - <http://boo.codehaus.org/>

**Criada pelo Brasileiro  
Rodrigo Bamboo**

## Ruby

- Ruby/.NET - <http://www.saltypickle.com/rubydotnet/>
- NETRuby - <http://www.geocities.co.jp/SiliconValley-PaloAlto/9251/ruby/nrb.html>

## Smalltalk

- SmallScripts - <http://www.smallscript.org/>
- #Smalltalk
- LSWVST.Net - <http://www.lessor-software.com/en/content/products/lswvst/lswvst-net.htm>

## Prolog

- P # - <http://www.dcs.ed.ac.uk/home/jjc/psharp/psharp-1.1.3/dlpsharp.html>

## TCL/TK

- Mailframe - <http://www.mailframe.net/Products/TCL/>
- TickleSharp - <http://forge.novell.com/modules/xfmod/project/?ticklesharp>





# Projeto IKVM

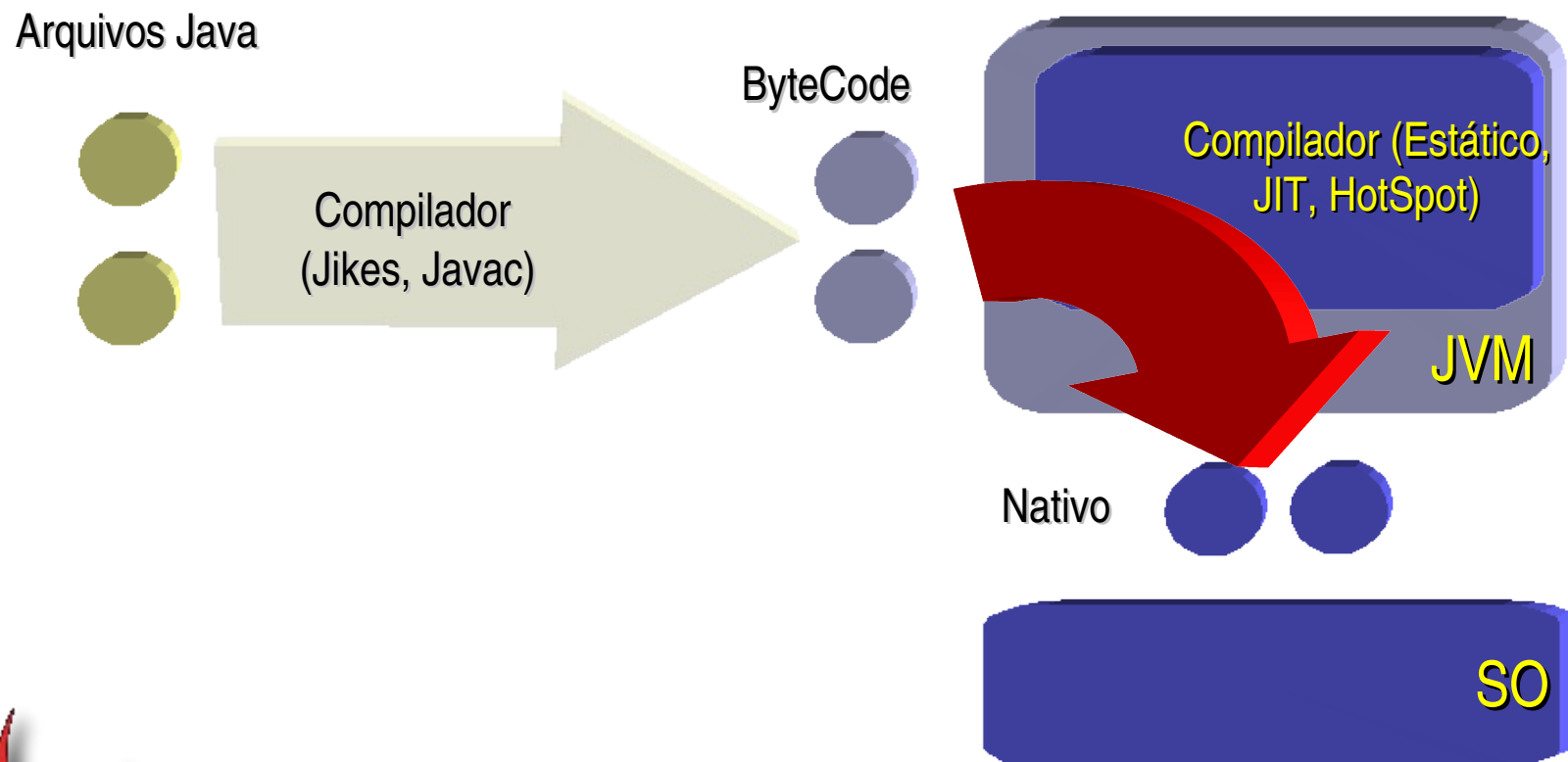
Integração Java com Mono

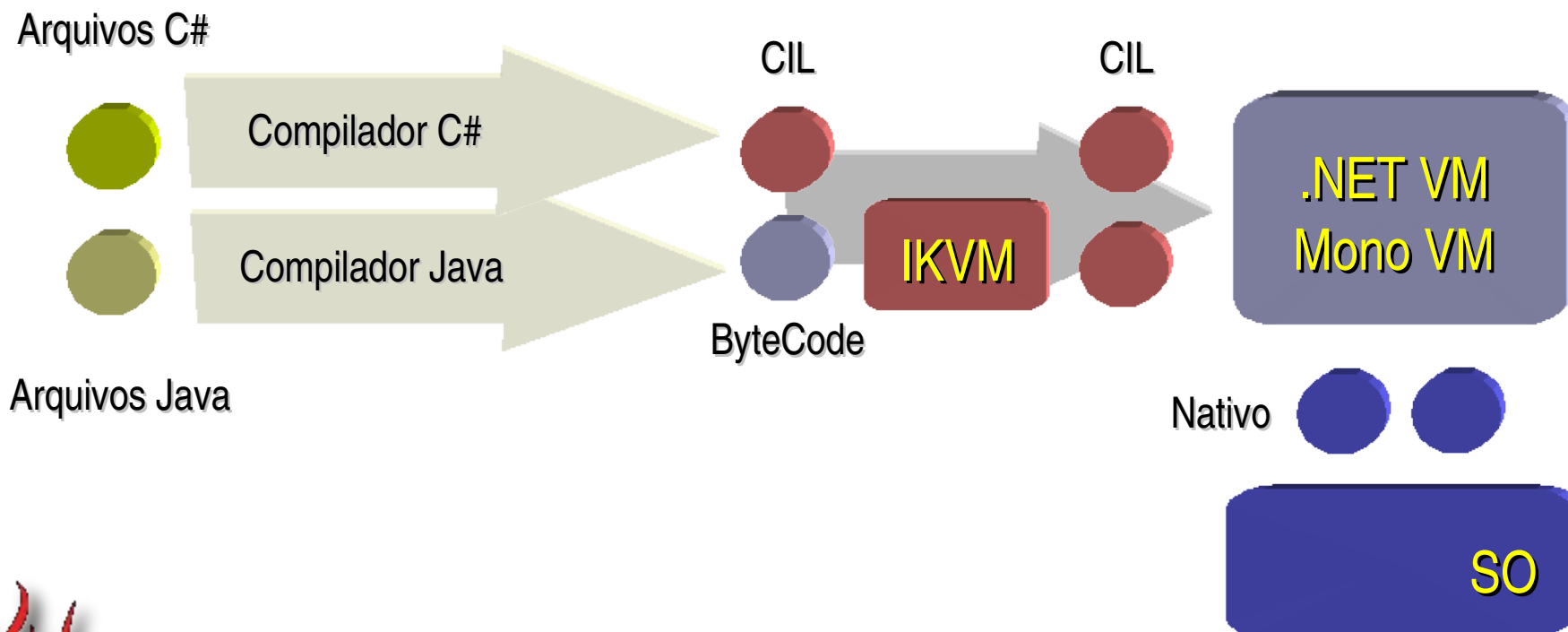
O **IKVM** é uma **VM JAVA** escrita em **C#** que possibilita executar byte code Java no Mono além de um conjunto de ferramentas para conversão bytecode (java) -> IL (mono) e IL (mono)-> bytecode (java)

✓ <http://www.ikvm.net>



# Java Virtual Machine





Programas Java

Programas .NET

ByteCode  
Adaptador  
CIL

ByteCode

CIL

ByteCode  
compilado para  
CIL














# Instalando a IKVM

Downloads - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.mono-project.com/downloads/

What is Mono? Mono Installation Downloads

	Red Hat 9.0/x86	- <a href="#">Packages</a>
	Fedora Core 1/x86	- <a href="#">Packages</a> For YUM repository, please use: <a href="http://www.go-mono.com/archive/1.0.2/fedora-1-1386/">http://www.go-mono.com/archive/1.0.2/fedora-1-1386/</a>
	Fedora Core 2/x86	
	SUSE LINUX Enterprise Server 8 for x86	
	SUSE LINUX Enterprise Server 9 for x86	
	SUSE LINUX 9.0 for x86	
	SUSE LINUX 9.1 for x86	
	Windows Installer Windows 2000 and above	
	Mac OS X	

Done

Mono 1.0.2 Release - Mozilla Firefox

File Edit View Go Bookmarks Tools Help


http://www.go-mono.com/archive/1.0.2/suse-91-i586/

What is Mono? Mono Installation Mono 1.0.2 Release

## Mono 1.0.2 Release: SuSE Linux 9.1, x86

### Single Package

Everything as a single Zip file:

 [mono-all.zip](#)

Unzip the above file, and install all of the resulting RPMs on your system.

### Individual Components

Optionally, for more control, you can pick and choose:

Mono Runtime and C# compiler

- [mono-core-1.0.2-1.ximian.9.1.i586.rpm](#)
- [mono-core-devel-1.0.2-1.ximian.9.1.i586.rpm](#)
- [mono-runtime-devel-1.0.2-1.ximian.9.1.i586.rpm](#)

Dependencies:

- [icu-2.6.2-1.ximian.9.0.i586.rpm](#)
- [libicu-devel-2.6.2-1.ximian.9.0.i586.rpm](#)
- [libicu26-2.6.2-1.ximian.9.0.i586.rpm](#)

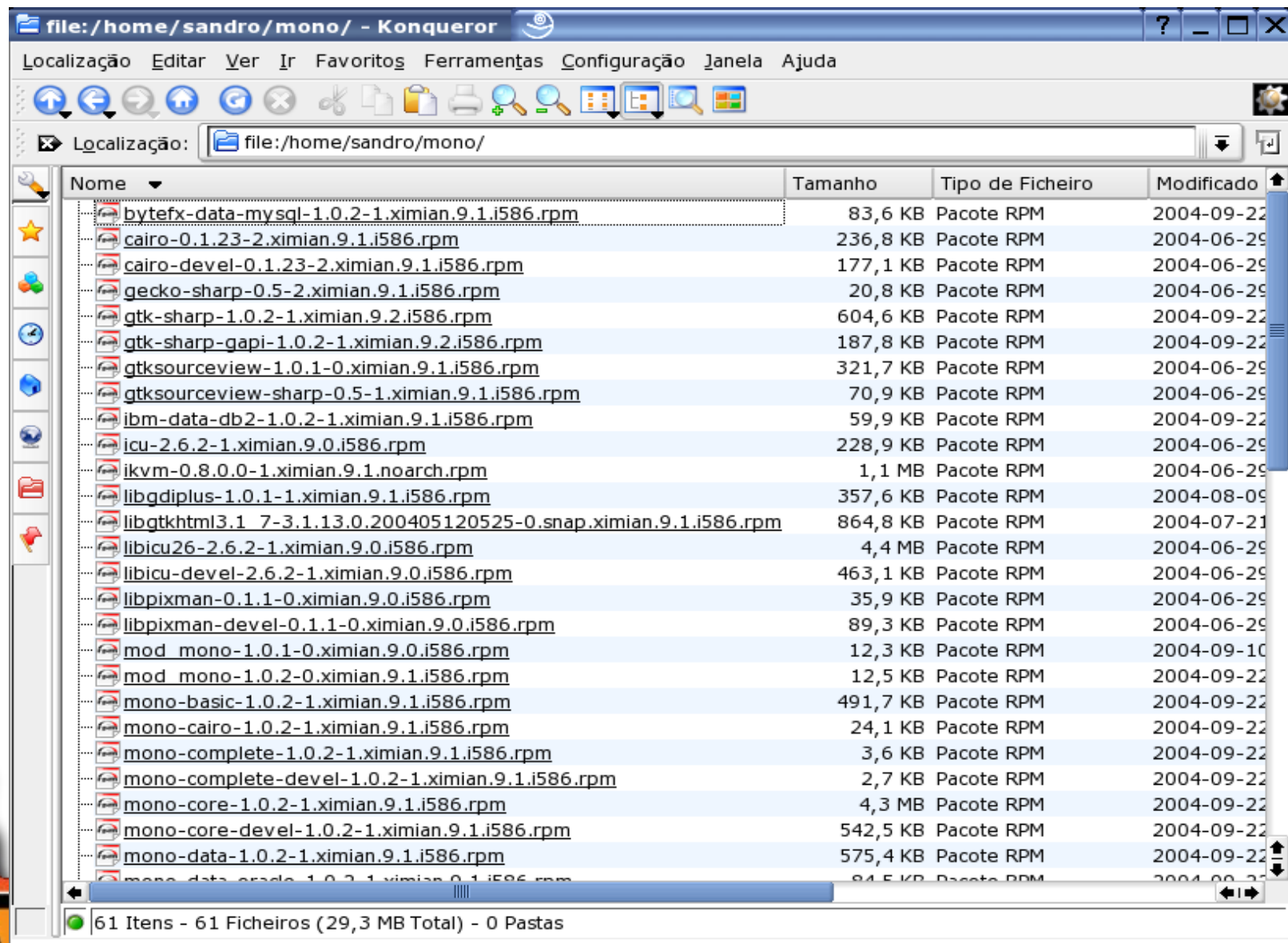
If you are installing mono-drawing or mono-cairo, you need to install Cairo and its dependencies:

- [cairo-0.1.23-2.ximian.9.1.i586.rpm](#)
- [cairo-devel-0.1.23-2.ximian.9.1.i586.rpm](#)
- [render-0.8-1.ximian.9.1.i586.rpm](#)
- [xrender-0.8.3-1.ximian.9.2.i586.rpm](#)
- [libpixmap-0.1.1-0.ximian.9.0.i586.rpm](#)
- [libpixmap-devel-0.1.1-0.ximian.9.0.i586.rpm](#)

Optional locales:



# Instalando a IKVM



- ✓ Mono CORE
  - ✓ mono-core-1.0.2-1.ximian.9.1.i586.rpm
  - ✓ mono-core-devel-1.0.2-1.ximian.9.1.i586.rpm
  - ✓ mono-runtime-devel-1.0.2-1.ximian.9.1.i586.rpm
  - ✓ mono-peapi-1.0.2-1.ximian.9.1.i586.rpm (!!)
- ✓ Dependencies
  - ✓ icu-2.6.2-1.ximian.9.0.i586.rpm
  - ✓ libicu-devel-2.6.2-1.ximian.9.0.i586.rpm
  - ✓ libicu26-2.6.2-1.ximian.9.0.i586.rpm



## ✓ IKVM

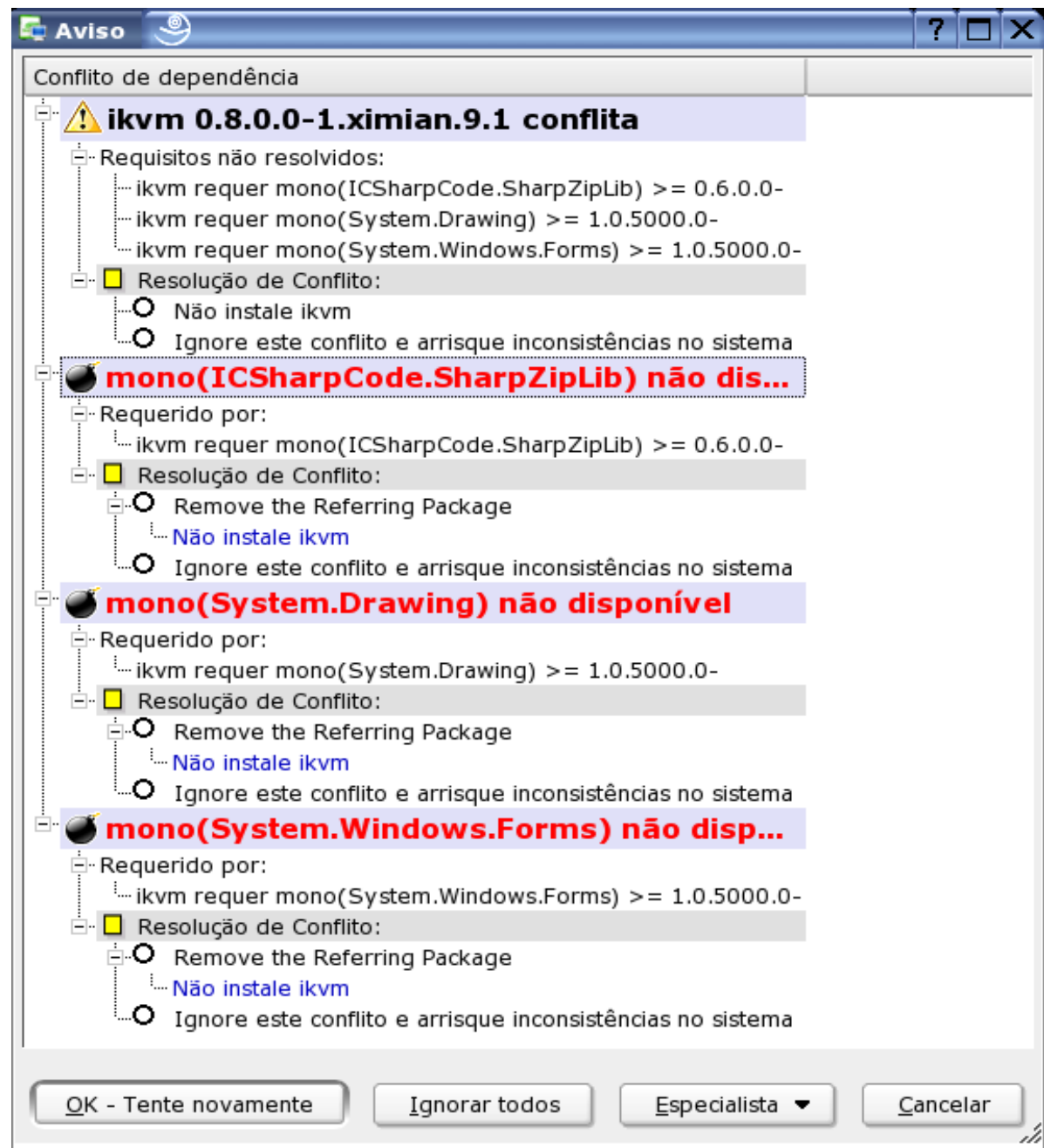
- ✓ mono-ikvm-1.0.2-1.ximian.9.1.i586.rpm
- ✓ ikvm-0.8.0.0-1.ximian.9.1.noarch.rpm

## ✓ Dependencies

- ✓ mono-drawing-1.0.2-1.ximian.9.1.i586.rpm
- ✓ mono-winforms-1.0.2-1.ximian.9.1.i586.rpm
- ✓ mono-data-1.0.2-1.ximian.9.1.i586.rpm
- ✓ mono-ms-enterprise-1.0.2-1.ximian.9.1.i586.rpm
- ✓ mono-ziplib-1.0.2-1.ximian.9.1.i586.rpm
- ✓ mono-remoting-1.0.2-1.ximian.9.1.i586.rpm
- ✓ mono-web-forms-1.0.2-1.ximian.9.1.i586.rpm



# Yast!!



## ##Mono Debian Project

```
deb http://debian.meebey.net/ ./
```

## ## Officials Backports

```
deb http://archive.ubuntu.com/ubuntu breezy-backports main universe multiverse restricted
```

## ## Backports

```
deb http://ubuntu-backports.mirrormax.net/ breezy-backports main universe multiverse restricted  
deb http://ubuntu-backports.mirrormax.net/ breezy-backports-staging main universe multiverse restricted  
deb http://ubuntu-backports.mirrormax.net/ breezy-extras-staging main universe multiverse restricted
```

# apt-get install IKVM



# **Download do Executável do Mono**

## **Download dos Executáveis da IKVM**

**Muito Fácil!!**

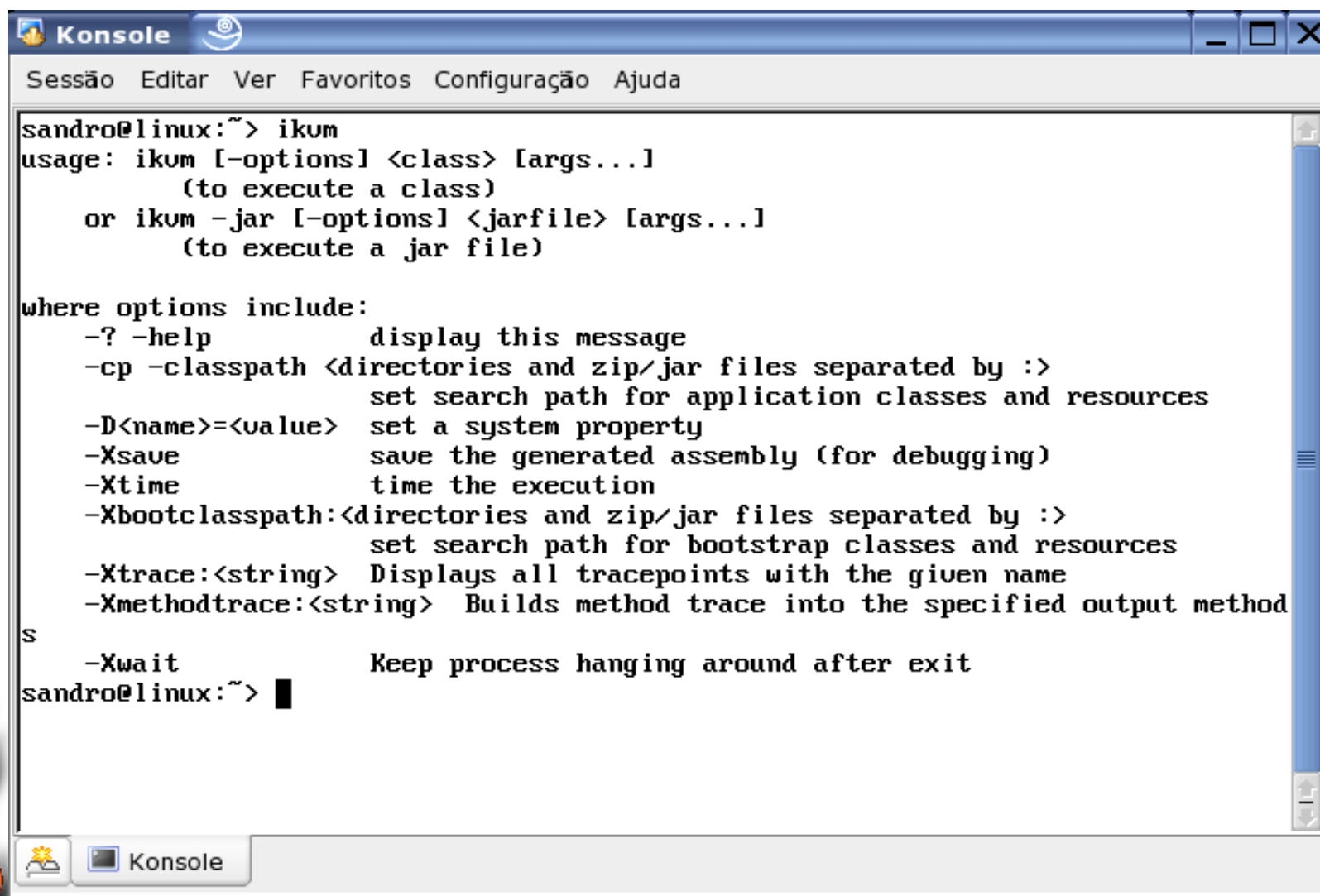




2 horas depois...

**Dica!!**

**- IKVM\_EXPERIMENTAL\_JDK\_5\_0=TRUE**



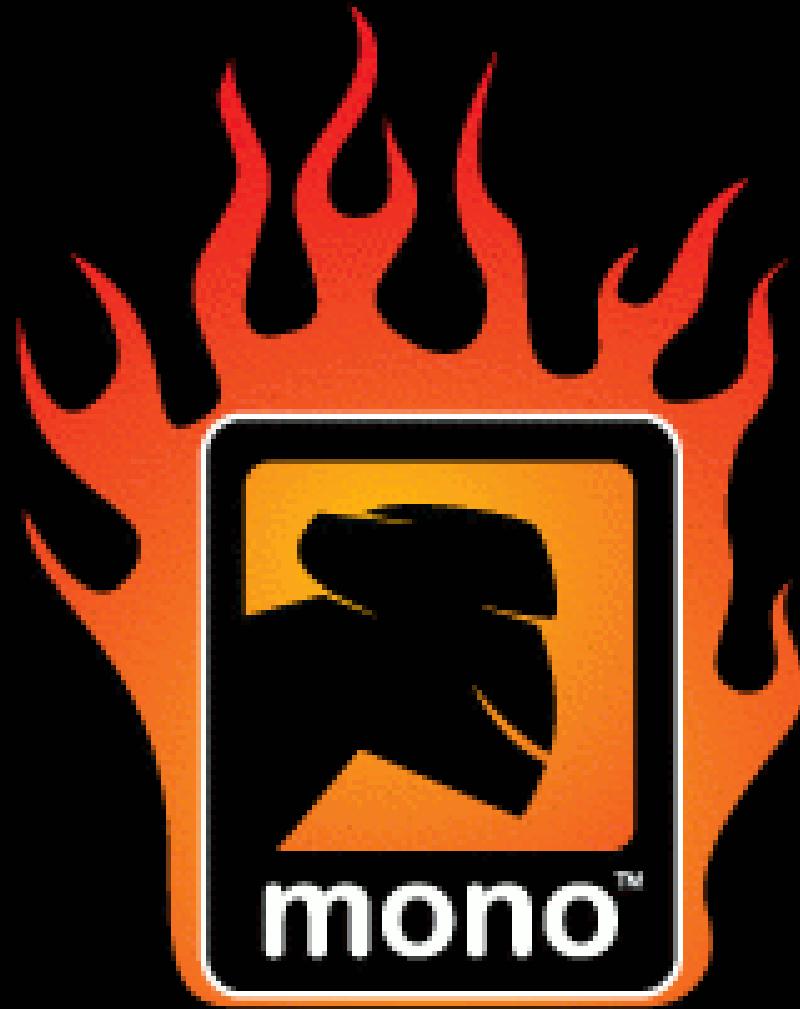
```
Konsole
Sessão Editar Ver Favoritos Configuração Ajuda

sandro@linux:~> ikvm
usage: ikvm [-options] <class> [args...]
        (to execute a class)
or ikvm -jar [-options] <jarfile> [args...]
        (to execute a jar file)

where options include:
  -? -help           display this message
  -cp -classpath <directories and zip/jar files separated by :>
                    set search path for application classes and resources
  -D<name>=<value>   set a system property
  -Xsave             save the generated assembly (for debugging)
  -Xtime             time the execution
  -Xbootclasspath:<directories and zip/jar files separated by :>
                    set search path for bootstrap classes and resources
  -Xtrace:<string>   Displays all tracepoints with the given name
  -Xmethodtrace:<string> Builds method trace into the specified output method
s
  -Xwait             Keep process hanging around after exit
sandro@linux:~> █
```







Exemplos

- Rodando uma Classe Java (dinamicamente)
- Compilando um ByteCode para CLI (.jar->.dll)
- Compilando um CLI para ByteCode (.dll -> .jar)
- Usar uma classe Java no C#
- Usar uma classe C# no Java



## Rodando uma aplicação java no Mono

```
import java.io.*;

public class Hello {

    public static void main(String[] args) throws Exception {
        BufferedReader rd = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("What's your name?");
        String name = rd.readLine();
        System.out.println("Hello, " + name);
    }
}
```

### Comandos

- > javac \*.java
- > jar cfm hello.jar manifest.mf Hello.class
- > ikvmc -reference:/usr/lib/ikvm/IKVM.GNU.Classpath.dll hello.jar
- > mono hello.exe



## Usando uma Classe Java dentro do C# (jar -> dll)

-> mcs -reference:IKVM.GNU.ClassPath hodgepodg.cs

-> mono hodgepodge.exe

```
using System;
```

```
using java.util;
```

Class JAVA

```
public class hodgepodge {  
    public static void Main() {  
        // Cria a instancia de uma string .NET  
        string str = "abc|def|ghi|jkl";
```

```
        // Tokenize é uma instancia da classe java.util.StringTokenizer  
        StringTokenizer st = new StringTokenizer( str, "|" );
```

```
        // Cria uma instancia da classe java.util.HashSet  
        Set s = new HashSet();
```

```
        // Faz a interação e pega os tokens  
        while( st.hasMoreTokens() ) {  
            s.add( st.nextToken() );  
        }
```

```
        // Passa a instancia como parametro para a saída .NET  
        // mecanismo -- pega toString() apropriados para o metodo  
        Console.WriteLine( "Set: {0}", s );  
    }  
}
```

JAVA  
CODE



## Usando as bibliotecas do Mono dentro do Java (dll -> jar)

### Passos

- 1) Gerar os stubs de .NET para Java, isto irá resultar em um arquivo mscorlib.jar
- 2) Quando compilar as classes java irão fazer uso das bibliotecas .NET bastando incluir o mscorlib.jar no classpath
- 3) Podemos ter 2 resultados:
  - criação dos binários Java para executá-los no IKVM
  - ou a conversão desses binários Java para .NET

### Comandos

- > ikvmstub mscorlib.dll
- > javac -classpath mscorlib.jar \*.java
- > ikvmc -reference:IKVM.GNU.Classpath.dll CreateFile.class
- > ikvmc -reference:IKVM.GNU.Classpath.dll ShowDir.class



# CreateFile.java

Biblioteca **MONO**

```
import cli.System.IO.*;
public class CreateFile {
    public static void main(String[] args) {
        String filename;
        if (args.length > 0) {
            filename = args[0];
        } else {
            filename = "MyFile.txt";
        }

        if (File.Exists(filename)) {
            System.out.println(filename + " already exists." );
            return;
        }

        StreamWriter sr = File.CreateText(filename);
        sr.WriteLine ("Some data.");
        sr.WriteLine ("I can write ints {0} or floats {1}, and so on.",
            new Object[] { new Integer(51), new Double(-4.2) });
        sr.Close();
        System.out.println(filename + " created.");
    }
}
```



# ShowFile.java

Biblioteca **MONO**

```
import cli.System.Environment;
import cli.System.IO.*;

public class ShowDir {
    public static void main(String[] args) {

        DirectoryInfo dirinfo = new DirectoryInfo(Environment.get_CurrentDirectory());

        FileInfo[] files = dirinfo.GetFiles();
        System.out.println("The following files exist in the current directory:");

        FileInfo fi;
        for (int i = 0; i < files.length; i++) {
            FileInfo info = files[i];
            System.out.println(info.get_Name() + "\t" + info.get_Length() + "\t" +
                (((info.get_Attributes().Value & FileAttributes.ReadOnly) != 0) ? "(read only)" :
                ""));
        }
    }
}
```



Estendendo uma classe Java em C#

## Passos

- 1) Criar a classe Java : “**Subject.java**”
- 2) Extender essa classe dentro do C#: “**Exntesion.cs**”
- 3) Compilar a classe java e gerar o .JAR
- 4) Converter o .JAR para um .DLL com o IKVM
- 5) Compilar a classe extension.cs com mono

## Comandos

- > javac \*.java
- > jar cvf subject.jar \*.class
- > ikvmc -reference:IKVM.GNU.Classpath.dll subject.jar
- > mcs -reference:IKVM.GNU.Classpath.dll -reference:subject.dll extension.cs -out:extension.exe





## Herança!!! - subject.java

```
import java.util.*;
// A Classe Observable é totalmente escrita em Java
public class Subject extends Observable
{
    private int foo;

    public Subject() {
        // Adiciona o seu Observer
        addObserver( new InnerObserver() );
    }

    public void setFoo( int foo ) {
        // Captura o mudando o estado e notificandoi qualquer Observers
        this.foo = foo;
        setChanged();
        notifyObservers( new Integer( foo ) );
    }

    public class InnerObserver implements Observer {
        public void update( Observable o, Object value ) {
            // Só queremos imprimir o indicador quando é notificado
            System.out.println("Java: I was told by: " + o + " about: " + value );
        }
    }
}
```



## extension.cs

## Herança!!!

```
using System;
using java.util;

// Esta class implementa a interface java.util.Observer
// Nota para progmdadores Java developers: use o ':', e não 'implements'!
public class Extension : Observer
{
    public void update( Observable o, Object value ) {
        // So quer imprimir alguma coisa quando for notificado
        Console.WriteLine(".NET: I was told by {0} about: {1}", o, value );
    }

    public static void Main() {
        // Cria a instancia desta classe
        Extension e = new Extension();

        // Cria a instancia da classe Java subject
        Subject s = new Subject();

        // Adiciona a esta classe como um Observer para a classe Java subject
        s.addObserver( e );

        // Muda o estado no Subject
        s.setFoo( 3 );
        Console.WriteLine("Done.");
    }
}
```



# Mono Movies Flash

## Aplicação Multi-linguagem

- Criar classe MyClass.java (desenha um gráfico na tela)
- A classe java usa as bibliotecas Mono (GTK#) para desenhar na tela
- Criar uma aplicação Boo para executar a classe Java
- Passos: compilar o Java -> converte para DLL -> compilar a aplicação Boo

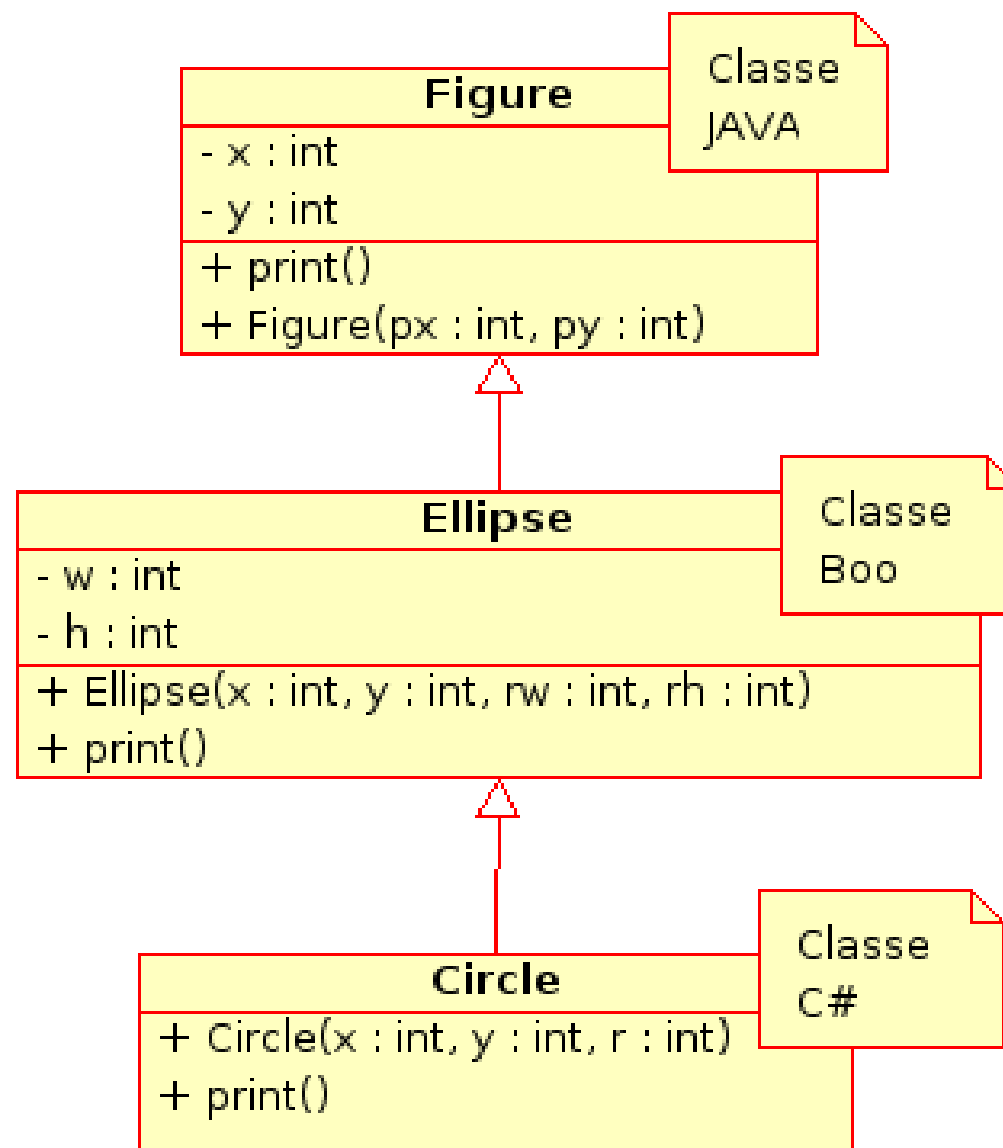


# Mono Movies Flash

## Debug Multi-linguagem

- Criação de 3 classes: Figure.java, Ellipse.boo e Circle.cs
- Circle (herda) Ellipse (herda) Figure
- Uso de chamadas a super classe
- Sobrecarga de Método Construtor
- Reescrita de métodos





## Figure.java

```
public class Figure
{
    int x;
    int y;

    public Figure(int px, int py)
    {
        x = px;
        y = py;
    }

    public void print()
    {
        System.out.println("Figure [" + x + " , " + y + "] ");
    }
}
```



## Ellipse.boo

```
import system
```

```
class Ellipse (Figure) ←
```

Herança

```
protected w
```

```
protected h
```

```
def construct (x,y,rw,rh)
```

```
super(x,y)
```

```
w = rw
```

```
h = rh
```

```
override def print ():
```

```
Console.Write("Ellipse [w:" + w + " h:" + h "]" )
```

```
super() ←
```

Chamada  
Super Classe



## Circle.cs

```
public class Circle : Ellipse
{
    public Circle(int x, int y, int r ) : base (x, y, r, r)
    {
    }

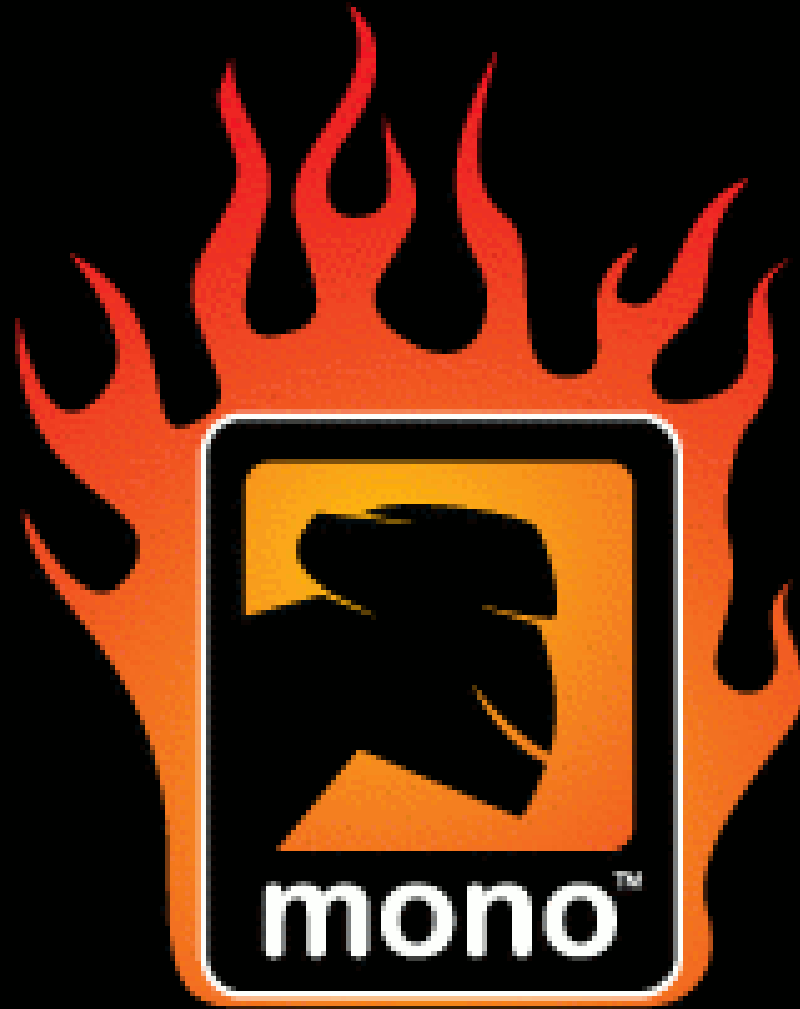
    public override void print()
    {
        Console.WriteLine("Circle (r: " + w + ") ");
        base.print();
    }
}
```

Herança

Chamada  
Super Classe



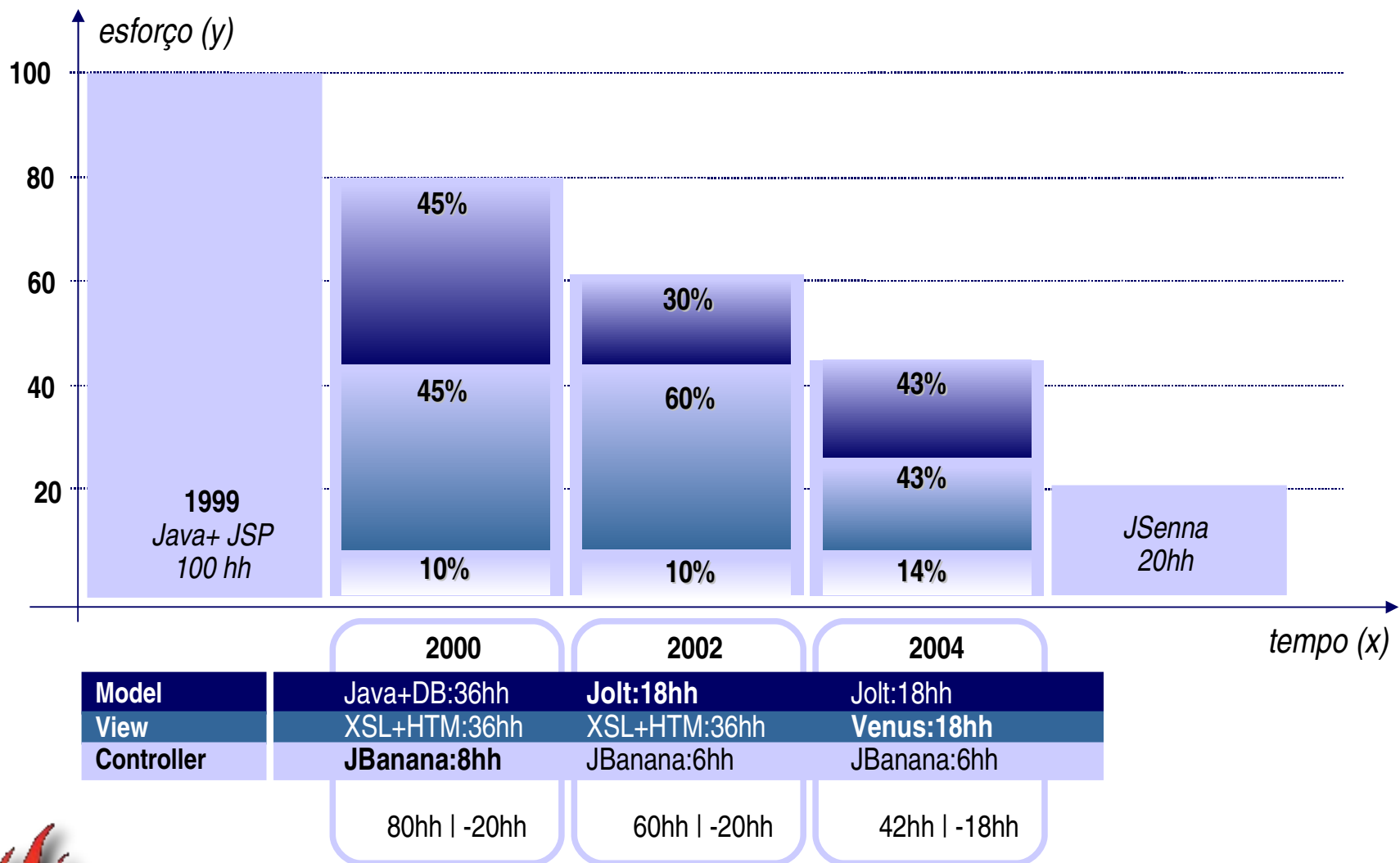




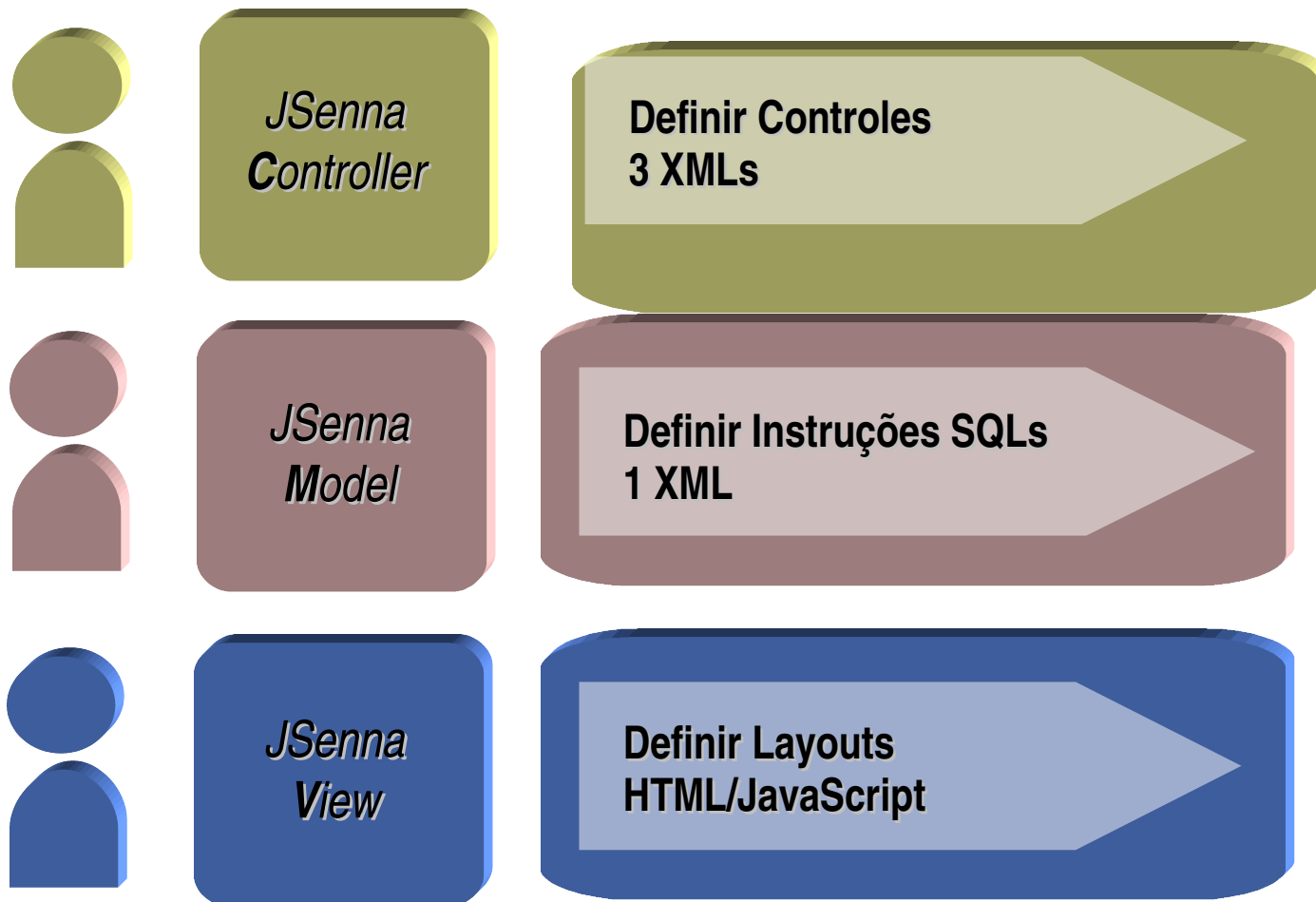
Projeto Rhodes

Uma Aplicação Web Real!

# CASE: Esforço x Tempo (Projeto+Implementação)



# CASE: Solução Completa



# CASE: UFPR / JBANANA + Jolt + Venus

**Dados Maternos - Mozilla**

File Edit View Go Bookmarks Tools Window Help

http://www.jbanana.org/jbanana/sala\_parto/dados\_maternos/index.html

Go Home Bookmarks mozilla.org Labs Build

Back Forward Reload Stop

Search

FireFox

Guida de Dados Maternos

Sala de Parto

Reg.	Nome	1.º Est. Gest.	Altura(m)	Escolaridade	Cor	IS	RH	DT.Últ. Parto	G.	P.	C.	A.	N.	I.
4-12	Maria da Silva Vale Almi	Cesária	1,85	1º Grau - Completo	Branca	A	+	10/10/2000	3	2	1	0	0	3

DL.Cadastr.	Trab. Parto	P. Espul.	B. Rota	Tempo B.R.(s)	DL.Mensl.	Corleza.DL	Ciclo Ok	D.P.P.	Anl. Con.	Gêmeos
10/10/2000	Sim	Não	Sim	EC	2/2/2000	Sim	Sim	Sim	Não	0
10/10/2000	Sim	Não	Sim	H	20/2/2000	Sim	Sim	Sim	Não	0

História da Gestação

- Úbitos
- Hábitos
- Pré Natal
- Medicações

Medicação	Uso (dias)	Trimestre de Uso
Medicação 1	10	Primeiro
Medicação 2	5	Primeiro
Medicação 3	0	Segundo

Sorologias

Diagnóstico Obstétrico

**Inserção/Atualização - Mozilla**

Detalhamento História Gestacional e Obstétrica - Medicações

Cód.Mãe: 1

Código: 1

Medicação: Medicação 1

Tempo de Uso: 10

Trimestre de Uso: Primeiro

Salvar



# CASE: Nova Arquitetura JSenna

Criação de uma camada de abstração do Application Server

Refatoração em todo o Código do JSenna

Regras de Negócio - XML

Framework JSenna

Abstração Servidor Aplicações

Implementação em  
ASPX MONO

Implementação em  
J2EE Java



## Porte do Projeto JSenna para Mono

- Refactoring JSenna, JBanana, Jolt
- Frameworks Portados
  - JBanana
  - Jolt
  - Venus
  - Log4J
  - Jasper
  - JDBC
    - PostgreSQL
    - MySQL
- Esforço Aproximado 800hh

É possível hoje  
conectar no MySQL  
usando JDBC no Mono!!!



## Projeto Rhodes

- ✓ Projeto Militar
- ✓ Requisitos:
  - ✓ Desenvolvimento Rápido
  - ✓ Prototipagem rápida
  - ✓ Baixo consumo de Banda
  - ✓ Várias interfaces com usuário
  - ✓ Redundância de Plataforma (Java, Mono e .NET)
  - ✓



- ✓ A IKVM é uma ponte para as Arquiteturas .NET e Java.
- ✓ A IKVM é a Virtual Machine da Virtual Machine.
- ✓ A IKVM permite o uso transparente das bibliotecas Java nas Plataformas .NET e MONO, e também de bibliotecas Mono no Java
- ✓ É muito possível no futuro sejam construídos outros projetos como a IKVM.





# Conclusão

A plataforma Mono apresenta um novo horizonte no desenvolvimento de software quebrando as barreiras entre diversas tecnologias possibilitando ao desenvolvedor uma liberdade que não era possível antes.

***<http://monobrasil.softwarelivre.org>***

## Perguntas?

Alessandro Binhara

[binhara@psl-pr.softwarelivre.org](mailto:binhara@psl-pr.softwarelivre.org)

Sandro Bihaiko

[sandro@jbanana.org](mailto:sandro@jbanana.org)

