

## 1. Guía de programación con joPAS

### 1.1. ¿Qué es joPAS?

joPAS es una API que ha sido creada por el Grupo PAS de la Universidad de Deusto. Permite al programador utilizar la potencia de cálculo del lenguaje Octave desde Java, el usuario puede crear programas en Java con la facilidad de poder implementar toda la parte matemática en Octave. Además la sintaxis de Octave es compatible con Matlab con lo que todas las sentencias creadas para Matlab también pueden ser usadas en joPAS.

### 1.2. ¿Qué requisitos necesita joPAS?

Para poder desarrollar aplicaciones utilizando el API joPAS se necesita tener instalado el JDK de java versión 1.5 y tener instalado el interprete de OCTAVE.

### 1.3. Pasar variables de JAVA a OCTAVE

```
import org.jopas.*;

public class Example1 {

    public Example1() {

        Jopas jopas = new Jopas(); //joPAS inicialitation

        double a = 6;
        jopas.Load(a,"a");

        double b = 2;
        Matrix mb= new Matrix (b,"b");
        jopas.Load(mb);

        Matrix A = jopas.Save("a");
        System.out.println(A.getRealAt(0,0));

        Matrix B = jopas.Save("b");
        System.out.println(B.getRealAt(0,0));

        System.exit(0);
    }

    public static void main(String av[]) {

        Example1 example1 = new Example1();
    }
}
```

#### 1.4. Pasar vectores de JAVA a OCTAVE

```
import org.jopas.*;

public class Example2 {

    /**
     * The constructor of the class have to start joPAS to run Octave because
     * Octave program takes a long time to start
     */
    public Example2() {
        Jopas jopas = new Jopas(); //joPAS inicialitation

        double[] aA= {1,2,3,4};
        Matrix mA= new Matrix (aA,"vA");
        jopas.Load(mA);

        Matrix mA2 = jopas.Save("vA");

        String smA2 = mA2.toString();
        System.out.println(smA2);
        System.out.println("v(0)="+mA2.getRealAt(0,0));
        System.out.println("v(1)="+mA2.getRealAt(0,1));
        System.out.println("v(2)="+mA2.getRealAt(0,2));
        System.out.println("v(3)="+mA2.getRealAt(0,3));

        System.exit(0);
    }

    public static void main(String av[]) {

        Example2 example2 = new Example2();

    }
}
```

### 1.5. Ejecutando sentencias de OCTAVE

```
import org.jopas.*;

public class Example3 {

    /**
     * The constructor of the class have to start joPAS to run Octave because
     * Octave program takes a long time to start
     */
    public Example3() {
        Jopas jopas = new Jopas(); //joPAS inicialitation

        double[] aA= {1,2,3,4};
        Matrix mA= new Matrix (aA,"vA");
        jopas.Load(mA);

        double[] aB= {5,3,8,1};
        Matrix mB= new Matrix (aB,"vB");
        jopas.Load(mB);

        jopas.Execute("vY=vA+vB");

        Matrix mY = jopas.Save("vY");

        String smY = mY.toString();
        System.out.println(smY);

        System.exit(0);
    }

    public static void main(String av[]) {

        Example3 example3 = new Example3();

    }
}
```

## 1.6. Ejecutando funciones en OCTAVE

```
import org.jopas.*;

public class Example4 {

    /**
     * The constructor of the class have to start joPAS to run Octave because
     * Octave program takes a long time to start
     */
    public Example4() {
        Jopas jopas = new Jopas(); //joPAS inicialitation

        jopas.Load(5,"a");
        jopas.Load(7,"b");

        jopas.Execute("function y=suma(a,b)");
        jopas.Execute("y=a+b");
        jopas.Execute("endfunction");

        jopas.Execute("y=suma(a,b)");

        Matrix mY = jopas.Save("y");

        System.out.println(mY.getRealAt(0,0));

        System.exit(0);
    }

    public static void main(String av[]) {

        Example4 example4 = new Example4();

    }
}
```

## 1.7. Representaciones gráficas con joPAS

```
import org.jopas.*;

public class Example5 {

    /**
     * The constructor of the class have to start joPAS to run Octave because
     * Octave program takes a long time to start
     */
    public Example5() {
        Jopas jopas = new Jopas(); //joPAS inicialitation

        jopas.Execute("y=rand(1,50)");
        jopas.plot("y");
    }

    public static void main(String av[]) {

        Example5 example5 = new Example5();

    }
}
```

## 1.8. Primer GUI utilizando joPAS

```
import java.awt.*;
import org.jopas.*;

public class Example6 extends Frame {

    JopasLabel plot;
    Jopas jopas;
    TextField aleatorio;

    public Example6() {
        jopas = new Jopas ();
        aleatorio= new TextField();
        aleatorio.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                aleatorioActionPerformed(evt);
            }
        });
        addWindowListener( new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent evt) {
                System.exit(0);
            }
        });
        plot = new JopasLabel(jopas);
        add( "South",aleatorio );
        add( "Center",plot );
        jopas.Execute("y=rand(1,20)");
        plot.paintLabel("y","Example 6","Samples","Value");
        pack();
        setSize(300,300);
        setVisible(true);
    }

    private void aleatorioActionPerformed(java.awt.event.ActionEvent evt) {
        Matrix L = new Matrix(Double.parseDouble(aleatorio.getText()), "L");
        jopas.Load(L);
        jopas.Execute("y=rand(1,L)");
        plot.paintLabel("y","Example 6","Samples","Value");
    }

    public static void main( String args[] ) {
        Example6 example6 = new Example6();
    }
}
```