

# **ELECTRÓNICA DIGITAL Y ANALÓGICA EN INFORMÁTICA. ORIENTACIÓN A OBJETOS Y SIMULACIÓN DE CIRCUITOS.**

## **RESUMEN**

*Java es un lenguaje actualmente en boga. Su empleo en disciplinas científicas gracias a su sencillez, alta portabilidad e integración hacen que sea muy usado en entornos académicos. En este trabajo presentamos dos herramientas Jlógica y JAnalógica que son el resultado de una visión docente del lenguaje de programación Java. Con Jlógica se dispone de una herramienta homogénea, intuitiva y sin restricciones, capaz de facilitar el aprendizaje en asignaturas tales como Estructura de Computadores, Tecnología de Computadores y Laboratorio de Arquitectura. JAnalógica por otro lado pretende el acercamiento del alumno a los niveles más bajos de las arquitecturas, los circuitos analógicos.*

## **1. INTRODUCCIÓN**

Conseguir involucrar al alumno en la simulación de circuitos, ya sean analógicos o digitales, es una labor que en el caso de las ingenierías informáticas potenciamos con software de simulación basado en componentes y orientado a objetos. La simulación software permite al alumno disponer del código que simula el comportamiento del circuito y de cada uno de sus componentes. Inspeccionar el código permite al alumno involucrarse directamente con los conceptos teóricos que manejamos en clase. Forzar a que el alumno sea el constructor de sus conocimientos.

Con tal fin estamos desarrollando un conjunto de herramientas: JLógica y JAnalógica. JLógica es un simulador Java de circuitos digitales lógicos en el que el alumno puede, a partir de puertas AND, OR y NOT, crear puertas complejas, simular ALU's, etc. JLógica permite también crear código Java que simule el circuito para poder testear sus capacidades. JAnalógica es una suite, en desarrollo, que permite simular circuitos analógicos. El alumno dispone en su casa de un "entrenador" virtual y de todos los componentes de que dispondría en el laboratorio (resistencias, condensadores, etc). La gran ventaja es que el alumno puede "abrir" el código de una resistencia y ver la implementación de las leyes que rigen su comportamiento. Con JLógica y JAnalógica se dispone de herramientas homogéneas, intuitivas y sin restricciones, capaces de facilitar el aprendizaje en asignaturas tales como Estructura de Computadores, Tecnología de Computadores, Simulación de Circuitos y Laboratorio de Arquitectura. Jlógica es una herramienta Java pensada para homogeneizar el entorno de trabajo en las prácticas de varias asignaturas, de esta manera módulos realizados para una determinada asignatura podrían ser reutilizables a medida que el alumno vaya adquiriendo conocimientos de mayor nivel con la consiguiente ventaja del conocimiento y experiencia práctica obtenida con el entorno de trabajo. Una de las principales ventajas de la herramienta (y por supuesto también aplicable a JAnalógica) es la independencia que otorga Java con respecto a la plataforma, así como la capacidad de integración con la red. La primera es importante ya que no restringe el sistema operativo usado en los equipos de los laboratorios. La segunda ventaja es que se hace extremadamente sencilla su utilización ya que una vez instalado en un servidor Web, el acceso puede realizarse tanto desde el laboratorio como desde cualquier punto conectado a la red (por ejemplo sus domicilios).

Por otra parte, Jlógica dispone de un módulo generador de código Java a partir del diseño gráfico del circuito, el cual facilita el análisis del comportamiento del mismo mediante la modificación

de parámetros básicos. Las asignaturas para las que se ha desarrollado esta aplicación son Tecnología de Computadores y Estructura de Computadores.

JAnalógica, además de proporcionar las ventajas mencionadas por estar desarrollada en Java, que son, de manera resumida: integración completa con el entorno web y la red. Facilidad para la reutilización de componentes, etc. Además de estas ventajas nos ofrece un entorno muy intuitivo en el que, cómo si de un entrenador se tratara, podemos ir añadiendo componentes electrónicos. De esta forma “simulamos” el montaje de un circuito. Cada pieza que se añade es “configurable”. De esta forma podemos estudiar las propiedades del circuito. La facilidad más atractiva desde nuestro punto de vista es la posibilidad de ir más allá, la posibilidad de inspeccionar el código fuente que da lugar a los componentes que usamos en el laboratorio.

En este artículo se detallan las características de utilización de estas herramientas. En el apartado 2 se describe la interfaz de Jlógica y su utilización como entorno de simulación de circuitos además se detalla la utilización de esta herramienta como generador de código a partir del diseño gráfico de un circuito. En el apartado 3 describiremos igualmente a Janalógica, a su interfaz y su modo de empleo en asignaturas como Simulación de Circuitos. La sección 4 recoge las experiencias recogidas al implantar estos sistemas en la docencia de estas asignaturas, siempre experimentalmente.

Además de haber optado por Java para la implementación por las razones antes aducidas: portabilidad e integración en la red, se pueden argumentar razones secundarias tales como que Java dispone de librerías de clases (Swing [4] y Awt [2]) que facilitan el desarrollo de la interfaz además de sobrados componentes visuales libres (Libre ha de entenderse que el propietario del software cede el componente para su uso de manera desinteresada) y reutilizables [1].

## 2. J-LOGICA. SIMULACIÓN DE CIRCUITOS DIGITALES.

Jlógica cuenta como punto de entrada con una interfaz muy similar a la interfaz ofrecida por Outlook para relacionar la aplicación con las asignaturas que harán uso de ella y separar adecuadamente la funcionalidad de la aplicación.

El panel de la izquierda dispone las áreas:

- Tecnología Computadores.
- Estructura de Computadores.
- Laboratorio de Arquitectura.

Dentro de cada una de estas áreas se encuentra la funcionalidad requerida para las asignaturas, sea el caso mostrado en Fig.1 en el que para la asignatura de Tecnología se dispone de opciones como *Simulador* que inicia la aplicación, *Enviar* para mandar al profesor resultados, etcétera. Pero la parte importante de la aplicación no es la interfaz que la relaciona con las asignaturas sino el propio simulador Fig.2.

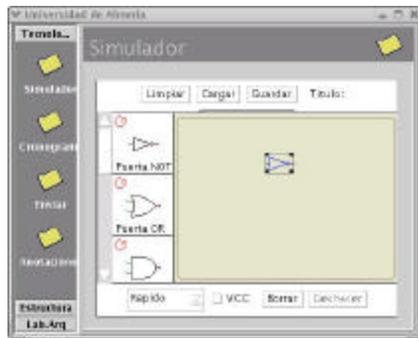


Figura1. Aspecto de la interfaz Jlógica

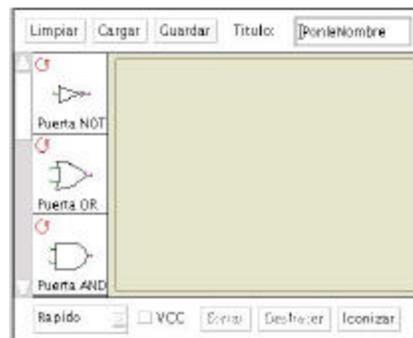


Figura2. Aspecto del núcleo Jlógica

El simulador dispone de un juego de puertas lógicas básicas (AND, NOT, AND) y de una serie de piezas de utilidad como conectores de entrada/ salida y de empalme (Fig. 3), las cuales están dispuestas en la barra de puertas a la izquierda del área de trabajo Jlógica. Con estas puertas y piezas de utilidad el alumno puede comenzar a montar circuitos. Además, Jlógica consta de otras dos barras la barra superior que actúa sobre el panel del circuito como un todo en la que se pueden ver las opciones *Limpiar*, *Cargar*, *Guardar* (El circuito se copia en un fichero binario al que se le añade un identificador para que el simulador sólo abra los ficheros que él ha generado y no cualquier otro tipo de fichero) y un área para nombrar el circuito y poder guardarlo en un fichero o encapsularlo en una pseudo-puerta de funcionalidad más compleja, etcétera; y la barra de la parte inferior en la que constan opciones aplicables a subconjuntos de puertas del circuito o bien al circuito completo, como *Borrar* que elimina la puerta seleccionada del circuito (así como sus conexiones), *Deshacer* que tras una modificación regresa a la situación anterior, *Iconizar* que “almacena” el circuito que se haya diseñado en una puerta que a partir de ese mismo momento aparecerá en la barra de la izquierda para poder ser reutilizada como componente de futuros circuitos. Además de estas opciones aparecen otras dos que tienen que ver con la simulación en sí, estas son el desplegable en el que se puede elegir la velocidad con la que se verán la transición entre estados de una tabla de verdad y el checkbox (opción que se activa o desactiva marcando con un tick en el lugar habilitado) *VCC* que es el que inicia la simulación haciendo las veces de fuente de alimentación.

## 2.1. Áreas de aplicación

Se entiende por áreas de aplicación a las asignaturas que pueden, en principio, hacer uso del simulador para adaptar parte de sus lecciones prácticas. Entre ellas se encuentran las que a continuación se enumeran.

### 2.1.1. Tecnología de computadores

En esta asignatura Jlógica le permite al alumno comprobar de forma fácil cómo los resultados obtenidos tras una minimización correcta de funciones coinciden con las versiones canónicas de los mismos, bien expresada en su forma de tabla lógica o circuito combinacional.

La herramienta simula al panel-entrenador que con carácter general es utilizado por los alumnos en este tipo de prácticas, disponiendo de los elementos fundamentales integrados en el panel (interruptores, leds, conectores, etc.) La disponibilidad de puertas lógicas, flip-flops u otro tipo de circuito lógico se obtiene a partir de la posible combinación de los elementos básicos (inversor, puerta AND y puerta OR) que se hayan integrados (Fig.2) con el desarrollo de la herramienta. De

esta forma la tarea del alumno, ante el testeo de cualquier circuito propuesto, consiste en el ensamblaje apropiado de estos tres elementos hasta conseguir la función necesitada. Toda función diseñada anteriormente puede encapsularse como elemento de circuito y reutilizarse posteriormente en diseños más complejos, confeccionando de esta forma una librería tan extensa como se necesite. A continuación se describe un ejemplo.

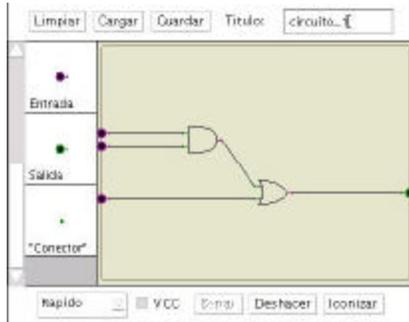


Figura3. Diseño de circuito.

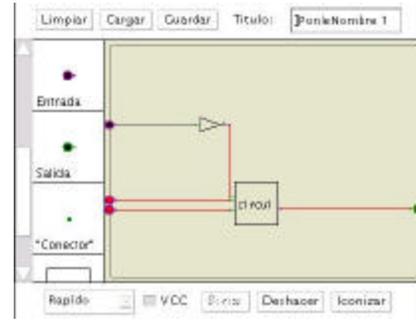


Figura4. Encapsulación y reutilización

La Fig.3 muestra cómo construir un circuito para implementar una determinada función. En Fig.4 se ve un ejemplo de cómo se encapsula para construir una puerta adicional llamada "circuito\_1" con funcionalidad ampliada.

En circuitos complejos donde el número de puertas sea elevado, existe la posibilidad de perder el diseño del circuito encapsulado que se está usando. La confusión es mayor si en lugar de un encapsulado se usan varios. Por tanto, Jlógica permite inspeccionar el contenido de los circuitos encapsulados sin más que seleccionar la pieza encapsulada y pulsar sobre el botón "Expandir" (La opción expandir sólo se hace visible cuando existen componentes compuestos en el circuito), tal y como se puede ver en la Fig. 5

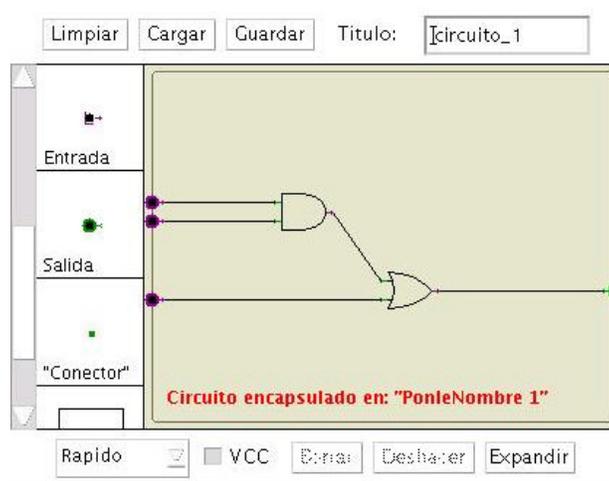


Figura5. Contenido del encapsulado utilizado en el montaje

### 2.1.2. Estructura de computadores.

El enfoque que se le otorga a esta asignatura en nuestro actual plan de estudios es el de iniciar al alumno en el conocimiento de los bloques fundamentales constitutivos de un computador.

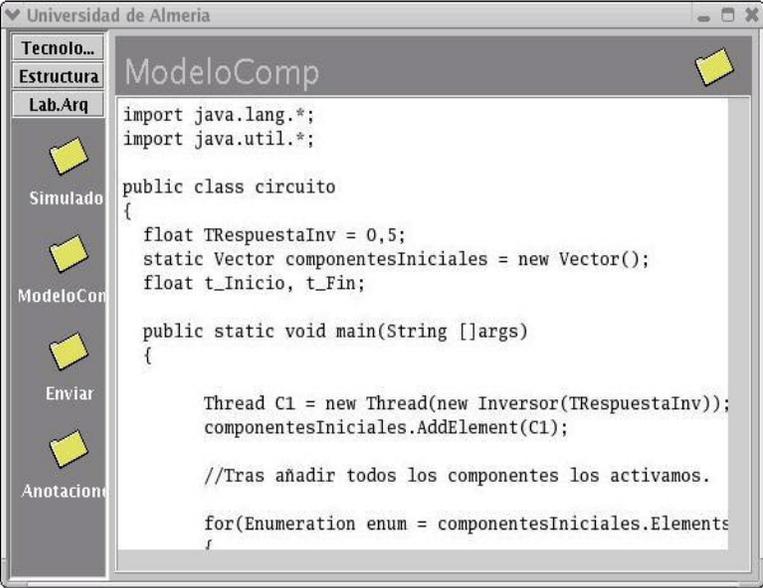
El alumno al finalizar la asignatura ha debido adquirir el conocimiento de la metodología de diseño a nivel de registros así como la estructura y organización de una unidad central de proceso. Así pues, el aprendizaje realizado con módulos lógicos elementales en la asignatura Tecnología de Computadores son aplicados en esta asignatura para constituir bloques más complejos como ALU's o el diseño completo de una unidad central de proceso sencilla.

Las características de Jlógica explicadas anteriormente facilitan la consecución de estos objetivos sin más que sustituir el conjunto de piezas básicas usadas en la asignatura Tecnología de Computadores por un nuevo conjunto de piezas necesarias para esta asignatura como pueden ser una ALU, bancos de memoria, bancos de registros, multiplexores, etcétera .

### 2.2. Generación de código.

Uno de los aspectos prácticos tratados en Laboratorio de Arquitectura de Computadores es la simulación computacional de circuitos lógicos [3], mediante la que el alumno experimenta el testeo de bloques computacionales constitutivos de una arquitectura analizando el comportamiento ante distintos patrones de entrada y modificación de parámetros característicos.

Jlógica facilita el análisis anterior mediante la generación de código (Fig.6) a partir del diseño gráfico del bloque a estudiar. El modelo en esta versión se genera en código Java, si bien se pretende ampliar a otros lenguajes tal capacidad.



The image shows a software window titled 'Universidad de Almería' with a sub-window 'ModeloComp'. On the left is a vertical menu with options: 'Tecnolo...', 'Estructura', 'Lab.Arq', 'Simulado', 'ModeloCon', 'Enviar', and 'Anotacion'. The main area displays Java code for a class named 'circuito'. The code includes imports for 'java.lang.\*' and 'java.util.\*', a class definition with fields 'TRespuestaInv', 'componentesIniciales', and 't\_Inicio', and a 'main' method that creates a thread and iterates over components.

```
import java.lang.*;
import java.util.*;

public class circuito
{
    float TRespuestaInv = 0,5;
    static Vector componentesIniciales = new Vector();
    float t_Inicio, t_Fin;

    public static void main(String []args)
    {

        Thread C1 = new Thread(new Inversor(TRespuestaInv));
        componentesIniciales.AddElement(C1);

        //Tras añadir todos los componentes los activamos.

        for(Enumeration enum = componentesIniciales.Elements
        {
```

Figura6. Ejemplo de generación de código para un inversor.

### 3. J-ANALOGICA. SIMULACIÓN DE CIRCUITOS ANALÓGICOS.

Esta herramienta está siendo usada en la asignatura de Simulación de Circuitos. JAnalógica cuenta como punto de entrada a la aplicación con el applet que se muestra en la Fig.7.

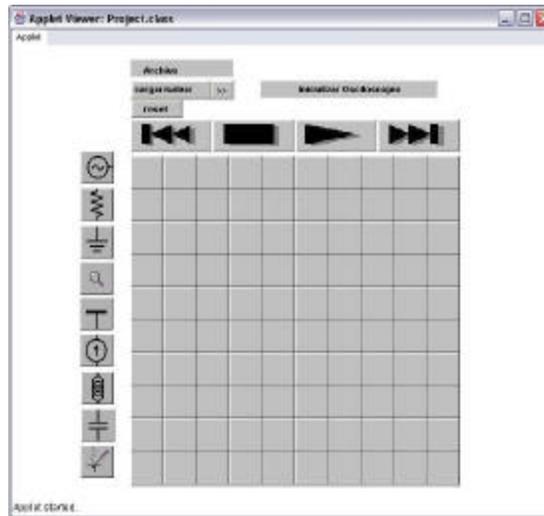


Figura7. Aspecto de la interfaz de JAnalógica.

Como se puede observar se presenta una cuadrícula que hace las veces de entrenador o placa de montaje. A la izquierda se disponen los elementos disponibles por ahora. De arriba abajo disponemos de una fuente de corriente alterna configurable (Amplitud de la señal, frecuencia ...). Resistencia con el parámetro de resistividad configurable. Enlace a tierra. Sniffer (icono marcado con la lente de aumento), es un elemento que se destina a “capturar” el estado de la señal que lo atraviesa para mostrarla en el osciloscopio si está activo (parámetro configurable : color). Tras el sniffer encontramos a la fuente de corriente continua, su único parámetro configurable es el voltaje. Tras esta y caracterizado con el icono de la flecha encerrada en el círculo se haya la configuración de la fuente actual (parámetros configurables son los amperios que genera y la tensión máxima). A continuación se haya la bobina (parámetro configurable: Inductancia). Tras la bobina se haya el condensador (parámetro configurable : capacitancia). Por último encontramos a la pieza que sirve para unir a los elementos emplazados en cada una de las casillas de la malla.

La forma de trabajar con esta herramienta es sencilla e intuitiva. Se seleccionan los elementos a incorporar en el circuito y se arrastran desde su posición original a la casilla elegida de la malla. Una vez dispuestos todos los elementos necesarios en la malla, los uniremos entre si con el “soldador” (último componente de los que hemos citado anteriormente). Una vez esté montado el circuito sólo resta pulsar el botón “play” para iniciar la simulación. Si el circuito está mal conectado o faltan elementos es en este momento cuando se detecta. Ver Fig 8.



Figura8. Detalle de circuito mal dispuesto.

Una vez que el circuito a montar esté correctamente ensamblado y conectado podemos ejecutar la simulación. Para estudiar el comportamiento del circuito en simulación hemos de añadir en los puntos deseados, piezas sniffers para así extraer la información deseada. Una vez hecho esto podremos iniciar el osciloscopio. El “osciloscopio” es un elemento en el que se vierte de manera gráfica la información capturada por cada uno de los sniffers. Ver Fig 9.

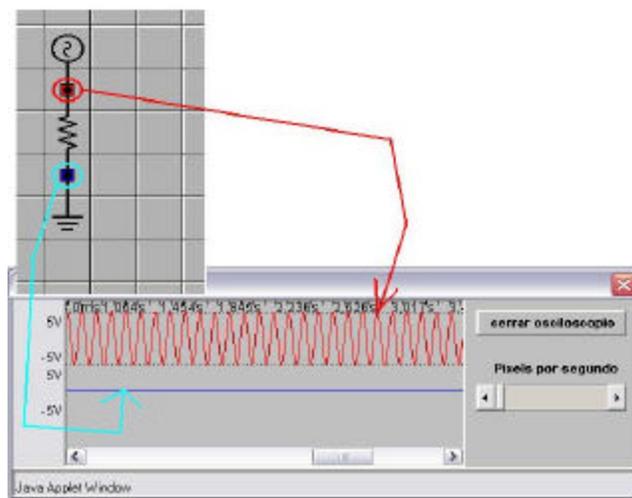


Figura9. Osciloscopio

JAnalógica cuenta también con la posibilidad de llevar a/cargar desde disco (grabar en un archivo) el circuito sobre el que se está trabajando así como sniffers, etc. Característica esta que lo hace extremadamente útil para la asignatura de Simulación de Circuitos.

### 3. CONCLUSIONES.

Proporcionarle al alumno la posibilidad de “investigar” cómo están hechas las cosas supone un reactivo importante para su aprendizaje. Con estas herramientas Jlógica y JAnalógica, el alumno dispone de un conjunto de código reutilizable y orientado a objetos. Características estas que lo hacen idóneo para el aprendizaje ya que el código se encuentra estructurado al máximo y con total claridad.

Por otro lado, el emplear Java les dota de gran flexibilidad y queda demostrado por experiencia propia que la curva de aprendizaje es mucho menor pues con Java están familiarizados prácticamente desde primero.

Jlógica y JAnalógica gozan de las capacidades propias del lenguaje Java como son independencia de la plataforma lo que le confiere gran portabilidad y las hacen herramientas muy flexibles y poco restrictivas, integrable en Internet. Ambas se integran en páginas Web sin dificultad (applets). Otra ventaja de estas herramientas es el carácter interdisciplinario que adquieren al poder utilizarse en diferentes asignaturas. El desarrollo de Jlógica es el fundamento de una herramienta con la que se pretende en un futuro inmediato alcanzar niveles de complejidad de una arquitectura completa, abordando tanto el diseño del camino de datos como el de la unidad de control. El desarrollo de JAnalógica pretende acercar al alumno de ingenierías informáticas a los niveles más bajos de las arquitecturas (más bajo que el nivel de puertas digitales) los transistores.

### 5. BIBLIOGRAFÍA

- [1] Duguay, Claude. Revista electrónica JavaPro. <http://www.fawcette.com/archives/magazines/javapro>
- [2] Knudsen, Jonathan. Java 2D Graphics. First Edition. ISBN 1.56592-484-3. Ed. O'Reilly.
- [3] Ruiz, V.G. *Diseño de Simuladores Digitales usando SDLC++*. IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE) . Barcelona, Spain. Septiembre. 2000.
- [4] Walrath, Kathy y Campione, Mary. The JFC Swing Tutorial, a guide to construct GUIs. ISBN 0-201-43321-4. Ed. Addison-Wesley.