

## Práctica Global. Junio 2003

**Fecha de publicación:** 10 de Febrero de 2003

**Conceptos:** Entrada-salida, herencia, árboles, estructuras de datos

**Soluciones:**

**Pruebas:**

**Enunciado comprimido:**

**Profesores:**

- ▣ [Introducción](#)
- ▣ [Visión global del programa](#)
- ▣ [¿Qué se pide?](#)
- ▣ [¿Cómo se evalúa la práctica?](#)
- ▣ [¿Cómo documentar el código?](#)
- ▣ [Plazos de entrega](#)
- ▣ [Pruebas de programa](#)

### INTRODUCCIÓN

El objetivo de la práctica es representar circuitos digitales combinacionales formados por puertas lógicas. Los circuitos se caracterizan por tener varias entradas (entre  $x_n$  y  $x_0$ ), siendo  $x_0$  el bit menos significativo y  $x_n$  el bit más significativo, y una sola salida ( $F$ ). Los circuitos pueden representarse de tres formas distintas:

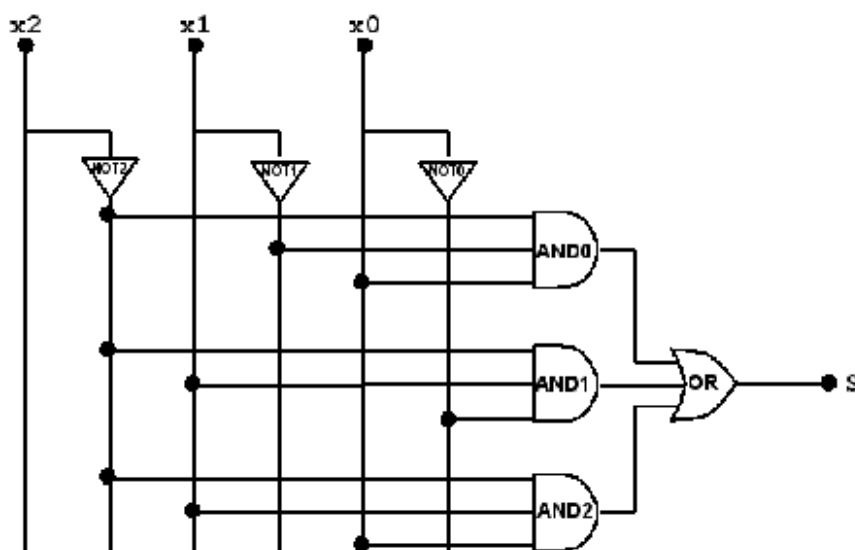
- mediante su función lógica en primera forma canónica (suma de productos)  

$$F = !x_2 * !x_1 * x_0 + !x_2 * x_1 * !x_0 + x_2 * x_1 * x_0$$

- mediante su tablas de verdad

#	$x_2$	$x_1$	$x_0$	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

- mediante la estructura de conexión de sus subcircuitos.



Se puede conectar la salida de un circuito a una entrada de otro circuito para componer circuitos más complejos. Se puede conectar una misma salida a distintas entradas de circuito. Sin embargo, no se pueden conectar varias salidas juntas (lógicamente, porque en ese caso no se sabría qué circuito está determinando el valor de la salida). El objetivo final será evaluar la salida de un circuito.

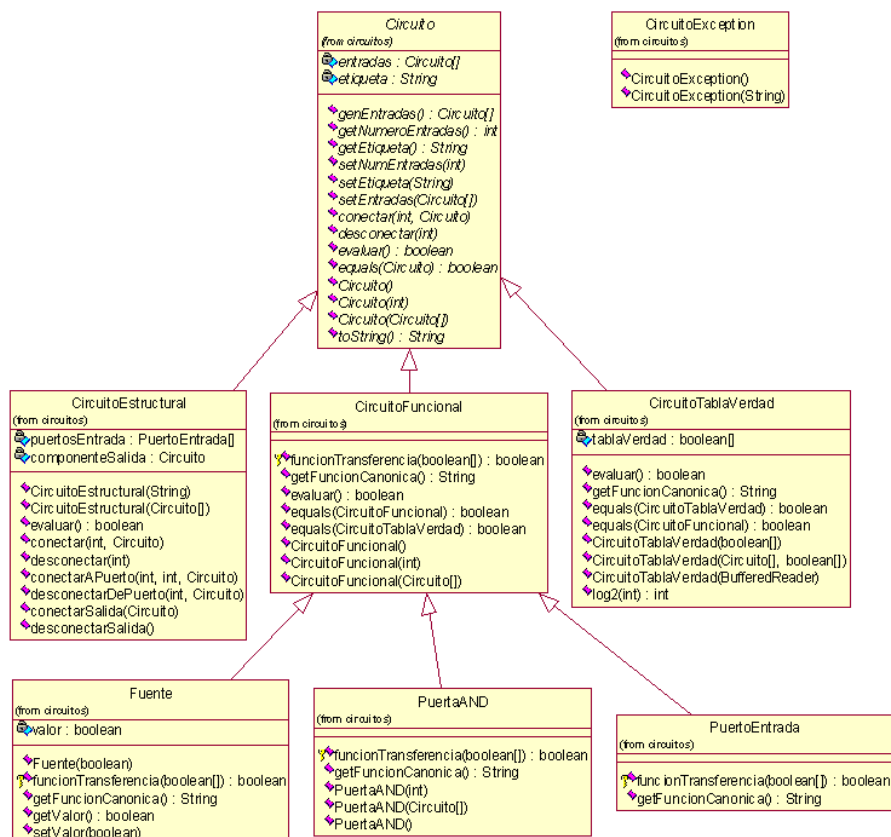
Esta práctica profundiza en tres aspectos de la programación en Java:

- herencia
- manejo de pilas
- creación y recorrido de árboles



## VISIÓN GLOBAL DEL PROGRAMA

La práctica se basa en una jerarquía de clases para representar los circuitos. A continuación se muestra el diagrama de clases global de la práctica, representado en UML. En él se pueden apreciar las dependencias entre clases, y los atributos y métodos de cada clase.



Para ver la información en detalle, se recomienda [navegar por la documentación](#) generada por javadoc.



## ¿QUÉ SE PIDE?

La práctica está dividida en tres módulos, cada uno de ellos con un enunciado aparte. Cada módulo será una práctica en si misma, es decir, tendrá un plazo de entrega y será evaluada como tal. Sin embargo, cada módulo dependerá del código escrito para el módulo anterior.

Para cada módulo se pide que el programa compile y ejecute correctamente. Cada módulo se probará con una batería de pruebas.

Además de pasar las pruebas, el código ha de estar **minuciosamente comentado**. Cada decisión de diseño, cada clase, cada método ha de tener toda su funcionalidad comentada. Se valorará la claridad tanto de la explicación como del código.



## ¿CÓMO SE EVALÚA LA PRÁCTICA?

Cada módulo recibirá una puntuación entre 0 y 10. La nota final de la práctica se obtendrá haciendo la media ponderada entre los tres módulos siempre y cuando se hayan entregado **todos los módulos**. El primer y tercer módulo tienen un valor de 3 puntos sobre 10 y el segundo tiene un valor de 4 sobre 10. Es necesario entregar los tres módulos para aprobar, si no se entrega uno de los módulos, la nota de prácticas será directamente 0.

La evaluación de la práctica se basará en los siguientes criterios:

1. El programa debe compilar correctamente.
2. El programa debe pasar **toda** la batería de pruebas proporcionada.
3. Calidad y claridad del código escrito (nombres de variables, márgenes, etc).
4. Calidad y claridad de la documentación del código. Se exige la documentación del código en formato "javadoc" (ver siguiente sección).
5. Calidad y claridad de la memoria.



## CÓMO DOCUMENTAR EL CÓDIGO

Para la labor de documentación del código, se utilizará el programa javadoc de extracción automática de documentación del código que viene incluido en la distribución del jdk. Este programa lee el fichero en el que está escrito el programa, extrae los comentarios escritos por el programador, y genera un conjunto de ficheros HTML de aspecto similar a la documentación del [API](#) de java cuyo contenido es el incluido en los comentarios y que se puede visualizar con cualquier navegador.

Para que este documento se genere adecuadamente es preciso poner los comentarios inmediatamente antes de la definición de una clase, interface, método, constructor o campo. Cada comentario se ha de comenzar por la secuencia de caracteres "/\*" y terminarla con la secuencia "\*/". He aquí un ejemplo de dónde se debe incluir la documentación (el texto correspondiente a la documentación del código está indicado en **negrita**):

```
/**
 * Descripción de la funcionalidad de la clase. Lo que representa y lo que hace.
 * @author (Nombre del autor o autores de la clase)
 * @version (Indica el número de versión si hay más de una)
 * @see (cuando se quiere referenciar alguna clase, variable o método relacionado)
 */
public class nombreClase {

    /** Descripción de un campo de la clase
     * @see (cuando se quiere referenciar alguna clase, variable o método relacionado)
     */
    public String nombreAtributo;

    /** Descripción del método con su funcionalidad, parámetros de entrada, valor de retorno,
     * excepciones si las hay, etc.
     * @param (para describir los parámetros)
     * @return (para describir el valor de retorno)
     * @exception (para describir las excepciones si las hay)
     * @see (cuando se quiere referenciar alguna clase, variable o método relacionado)
     */
    public static void nombreMetodo() {
        //código del método
    }
} // End of class
```

Una vez terminado de escribir el código con su documentación, la herramienta "javadoc" se encarga de confeccionar el conjunto de documentos HTML que engloban los comentarios. Se pide que la documentación de la práctica se produzca ateniéndose al formato que se acaba de describir.



## PLAZOS DE ENTREGA

Módulo	Enunciado	Fecha de Entrega
<a href="#">Módulo 1</a>	<a href="#">enunciado_M1.zip</a>	14 de marzo de 2003
Módulo 2		25 de abril de 2003
Módulo 3		23 de mayo de 2003

- Las prácticas se entregarán en formato electrónico (ya se indicará el mecanismo) hasta las 8:00 de la tarde del

día indicado para cada uno de los módulos.

- La práctica entregada deberá contener todo el código fuente convenientemente documentado (comentarios de código + comentarios en javadoc)
- En caso de que se reciban varias entregas por parte del mismo equipo antes de la fecha indicada ÚNICAMENTE se tendrá en cuenta la última entrega realizada. ¡¡Tened cuidado con el control de versiones!!



## PRUEBAS DE PROGRAMA

El programa se va a probar con una batería de pruebas que se determinará más adelante.



*Última actualización: 03/24/2016 11:34:59*

[Localización](#) | [Personal](#) | [Docencia](#) | [Investigación](#) | [Novedades](#) | [Intranet](#)  
[inicio](#) | [mapa del web](#) | [contacta](#)