**TEST DE IMPRESIÓN**

ÍNDICE

***1. Introducción***  2

***2. Parte de HTML y JAVASCRIPT***  2

***3. Parte de PHP*** 2

***4. Parte de JAVA*** 3

1. Clase ‘*AppletImp()*’ y clase ‘*AbrirPuerto()*' 3
2. Clase '*Imprimir()*' 3
   * Clase ‘*CasosDefinidos()*’ 3
   * Clase ‘*ComandoDecimal()*’ 4
   * Clase ‘*ComandoHexadecimal()*’ 4
   * Clase ‘*CodigoBarras()*’ 4
     + EAN8 4
     + EAN13 5
   * Clase ‘*Code128()*’ 5
     + CODE128A / CODE128As 5
     + CODE128B. / CODE128Bs 5
     + CODE128C. / CODE128Cs 6

1. Finalización de la aplicación 6

***1. Introducción:***

Esta aplicación une los lenguajes de programación *HTML*, *JAVASCRIPT*, *PHP* y *JAVA*. Su función sería la de poder imprimir por una impresora de tickets conectada a un puerto serie del ordenador.

***2. Parte de HTML y JAVASCRIPT:***

La pantalla principal (el archivo ‘Test.php’) está diseñada en *HTML* y *PHP*, básicamente consta de:

* Un desplegable donde se escoge el puerto serie por el cual se quiere imprimir.
* \*Un campo de texto para introducir aquello que se va a imprimir.
* \*\*El botón ‘BORRAR’, que como su nombre indica se utiliza para dejar el campo de texto en blanco.
* El botón ‘Imprimir’ que finalmente, si se ha escogido el puerto y rellenado el campo, lo que hace es mandar la información para que sea procesada y enviada al puerto serie.
* Una pequeña pantalla al final de la página (sólo visible cuando se le introduce algo y se presiona el botón de imprimir) que muestra si se ha impreso lo introducido o ha habido algún error. Esta pantalla es un *APPLET* de *JAVA* por el cual se envían los datos (el puerto y el texto) a la aplicación *JAVA*.

\*Se le puede enviar texto, pero si se quiere cambiar el formato, el tipo de fuente, alineado, etc., se han de enviar comandos especiales (*ESC/POS*) que serán tratados en la parte de *JAVA* y enviados a la impresora para que los interprete. Estos comandos se deben enviar entre corchetes ‘[…]’ y hay varios tipos que se explicarán más adelante.

\*\*El botón ‘BORRAR’ ejecuta el archivo de *JAVASCRIPT* ‘Script.js’, que es el encargado de eliminar todo lo que haya en el campo de texto.

***3. Parte de PHP:***

La parte de PHP es para controlar, que al pulsar el botón de ‘Imprimir’, el campo de texto se haya rellenado y que se haya escogido un puerto serie, de lo contrario mostrará un mensaje de aviso en la pantalla indicando qué ha fallado y no permitirá que se ejecute el *APPLET* de *JAVA* ya que no hay datos para imprimir.

***4. Parte de JAVA:***

Esta parte es la encargada de gestionar los datos que se quieren enviar al puerto serie, de reconocer los comandos y controlar los errores que se presenten.

Consta de 8 clases distintas que se van ejecutando de esta manera:

1. La primera que se ejecuta es ‘*AppletImp()*’ que recibe los datos del puerto serie y el texto a imprimir. Entonces se envía el nombre del puerto a la clase ‘*AbrirPuerto()*’ para que compruebe si existe y si no está ocupado para poder utilizarlo.

La respuesta vuelve a ‘*AppletImp()*’ y si es negativa entonces informa por pantalla que el puerto no se puede utilizar, si por el contrario es positiva, la información es enviada a otra clase: ‘*Imprimir()*’.

1. La clase ‘*Imprimir()*’ recibe el nombre del puerto y el texto. Esta parte controla qué hay que enviar directamente por el puerto serie, y qué hay que gestionar de una manera especial (los comandos). Cualquier carácter que no se encuentre entre corchetes ‘[…]’ será enviado directamente para su impresión, pero en cuanto se detecta que hay algún comando se guarda en una cadena de texto y se envía por una serie de clases para determinar de qué tipo es y que se debe mandar exactamente al puerto.

Hay cuatro tipos de comandos, y una clase distinta para gestionar cada uno:

* + La clase ‘*CasosDefinidos()*’, recibe la cadena y compara si está entre una serie de comandos definidos por el programador. Si coincide con alguno se envían los parámetros necesarios internamente. Son los siguientes:
    - SALTOLINEA – Lo que se encuentre a continuación de este comando se imprimirá en una nueva línea.
    - CORTARPAPEL – La impresora cortará el rollo de papel.
    - FUENTEA / FUENTEB / FUENTEC / FUENTED – Son cuatro tipos de fuentes para el texto. La FUENTEA es la estándar (‘fuente A’), la FUENTEB es más pequeña (‘fuenteB’), la FUENTEC (‘**fuente C**’) y la FUENTED (‘**fuente D**’).
    - IZQUIERDA / CENTRO / DERECHA – Este comando es para alinear el texto a la izquierda, centro o derecha.
    - SUBRAYADO1 / SUBRAYADO2 / NOSUBRAYADO – Es para subrayar el texto que va a continuación del comando. SUBRAYADO1 hace un subrayado flojo, SUBRAYADO2 hace un subrayado fuerte y NOSUBRAYADO lo desactiva en caso de que se hubiera insertado antes.
    - INVERTIR – Invertirá los colores de blanco y negro, cuando se vuelva a introducir el mismo comando se volverá a invertir, quedando el texto como al principio.
    - NORMAL – La impresora configura los parámetros por defecto, si había algún comando activado (centro, subrayado…) lo desactiva.
  + Si el comando no pertenece a la clase anterior (‘*CasosDefinidos()*’) entonces pasa a la siguiente: ‘*ComandoDecimal()*’. Esta parte del código comprueba si lo que se quiere enviar entre corchetes ‘[…]’ son números en decimal que se correspondan con los del código *ASCII* (rango del 0 al 127). Si es así lo manda al puerto serie para que la impresora lo interprete, sino vuelve atrás para revisar otras clases.
  + Si el comando tampoco pertenece a la clase anterior, es decir, no es un número decimal, entonces pasa a ‘*ComandoHexadecimal()* ‘ para verificar si es un número en hexadecimal, que como antes, ha de corresponderse con los del código *ASCII* (rango del 0 al 7F, que en decimal es 127).

Para que la clase detecte la cadena como número hexadecimal, ha de escribirse ‘0x’ y el número a continuación. Por ejemplo: ‘[0x1B]’ o ‘[0x7H]’. Si no se introduce de esta manera no lo reconocerá y será enviado a la próxima clase.

* + Si la cadena no se corresponde a ninguna de las tres clases anteriores, pasa a la última comprobación: ‘*CodigoBarras()*’. En este apartado se verifica si lo que se quiere mandar es un código de barras, pero para que lo detecte como tal ha de ser uno de estos tres tipos: *EAN8*, *EAN13* o *CODE128* (que a su vez puede ser del tipo *A*, *B* o *C*). Cada tipo se debe introducir de una forma específica para que llegue con éxito al puerto serie:
    - *EAN8* – Es un código que consta de 8 números, el último es el dígito de control y si no se le envía la impresora lo calcula automáticamente. Se puede mandar de dos maneras distintas:
      * Escribiendo ‘EAN8s’ haciendo referencia a la versión estándar, con lo cual los parámetros del código están preestablecidos por el programador y sólo se han de introducir los números correspondientes. Por ejemplo: ‘[EAN8s:8745746]’.
      * Introduciendo ‘EAN8’, para así poder configurar dos parámetros: el primero hace referencia a la posición que cogerán los números al imprimir el código de barras (arriba, abajo…), y el segundo a la altura que se le quiera dar.

Para la posición de los números se ha de introducir un número del 0 al 3 (0 – No imprimir, 1 – Arriba, 2 – Abajo, 3 – Arriba y abajo).

Para la altura se ha de introducir una cifra del 000 al 255, pero siempre manteniendo los 3 dígitos.

Por ejemplo: ‘[EAN8:20504758962]’ (El primer ‘2’ es la posición de los números, el ‘050’ es la altura, y el resto son los propios números).

* *EAN13* – Es un código de 13 cifras, y como ocurre con *EAN8*, el último es el dígito de control que si no se envía lo calcula la impresora. También se puede mandar de dos formas:
  + La versión estándar introduciendo ‘EAN13s’, de esta manera sólo se tienen que escribir los números del código. Por ejemplo: ‘[EAN13s:741258369753]’.
  + Escribiendo ‘EAN13’, así se podrán cambiar dos parámetros del código: donde posicionar los números (0-3) y la altura (000-255). Se deben introducir igual que en *EAN8*.

Por ejemplo: ‘[EAN13:1070456987123357]’ (El primer ‘1’ es la posición de los números, el ‘070’ es la altura, y el resto son los propios números).

* + Si no es ni *EAN8* ni *EAN13*, pasa a la clase ‘*Code128()*’, que se realiza la llamada desde la clase ‘*CodigoBarras()*’. Hay tres clases de *CODE128*: el tipo *A*, el tipo *B* y el tipo *C*.
* *CODE128A* – Es un tipo de código que no sólo puede imprimir números, sino también símbolos y letras que se correspondan con los números del 0 al 95 del código *ASCII*, y a diferencia de los anteriores, no tiene límite de símbolos introducidos. Hay dos métodos para imprimir este código:
  + Introduciendo ‘CODE128As’, que es la versión estándar y no hay que configurar nada, tan sólo los dígitos o símbolos a imprimir. Por ejemplo: ‘[CODE128As:17A5T2]’.
  + Escribiendo ‘CODE128A’, donde se configurarán unos parámetros: donde posicionar los números (0-3) y la altura (000-255). Se deben introducir igual que en *EAN8* y *EAN13*.

Por ejemplo: ‘[CODE128A:302547M2C]’ (El primer ‘3’ es la posición de los números, el ‘025’ es la altura, y el resto son los propios números o símbolos).

* *CODE128B* – Como el tipo *A* no sólo puede imprimir números, también símbolos y letras pero esta vez deben coincidir con los números del 32 al 127 del código *ASCII*, y como el tipo *A* no tiene límite de símbolos introducidos. Hay dos métodos para imprimir este código:
  + Introduciendo ‘CODE128Bs’, que es la versión estándar y no hay que configurar nada, sólo los números o símbolos para imprimir. Por ejemplo: ‘[CODE128Bs:2d7Hf]’.
  + Escribiendo ‘CODE128B’, donde se configurarán los siguientes parámetros: donde posicionar los números (0-3) y la altura (000-255). Se deben introducir igual que en los códigos anteriores.

Por ejemplo: ‘[CODE128B:203087g5F]’ (El primer ‘2’ es la posición de los números, el ‘030’ es la altura, y el resto son los propios números o símbolos).

* *CODE128C* – El tipo *C*, a diferencia del *A* y el *B* sólo puede imprimir números, y no tiene límite de cifras introducidas. Se deben introducir parejas de números del 00 al 99, siempre manteniendo los dos dígitos. Hay dos métodos para imprimir este código:
  + Introduciendo ‘CODE128Cs’, que es la versión estándar y no hay que configurar ningún parámetro, tan sólo escribir los números. Por ejemplo: ‘[CODE128Cs:457802]’.
  + Escribiendo ‘CODE128C’, donde se configurarán los parámetros de: donde posicionar los números (0-3) y la altura (000-255). Se deben introducir igual que en todos los códigos anteriores.

Por ejemplo: ‘[CODE128C:204098721702]’ (El primer ‘2’ es la posición de los números, el ‘040’ es la altura, y el resto son los propios números o símbolos).

1. Finalmente vuelve a la clase ‘*Imprimir()*’, si el comando no ha entrado en ninguna de las clases anteriores se activará un error para luego mostrarlo al usuario y seguirá leyendo el texto introducido para volver a realizar la misma operación.

Cuando no hay nada más que procesar vuelve a la primera clase (‘*AppletImp()*’) donde para acabar, mostrará por pantalla si se ha impreso correctamente, si ha habido errores o si no se ha impreso nada.

*Viernes, 21 de enero de 2011, Abel Martín.*