



UNIVERSIDAD
POLITECNICA
DE VALENCIA

MetaCard: minitutoriales

Apellidos, nombre	Agustí i Melchor, Manuel (magusti@disca.upv.es).
Departamento	Departamento de Informática de Sistemas y Computadores
Centro	Universidad Politécnica de Valencia



1 Resumen de las ideas clave

MC [1] (en adelante MC) es un ejemplo de herramienta de autor multiplataforma para desarrollos multimedia. Esta aplicación ha sido el germen de otras dos: *RunRevolution* y *LiveCode* [2]. Aunque sus sucesoras han crecido en el número de asistentes para realizar las más actuales tareas, esta una aplicación muy didáctica que permitirá iniciarse en este tipo de herramientas que pertenecen a la metáfora de tarjetas y guiones.

Aquí se presenta el resultado de la recopilación de actividades de evaluación con el fin de ofrecer un nuevo tutorial con el que iniciarse con la herramienta MC.

2 Objetivos

El objetivo global es recopilar una serie de ejemplos que se han propuesto en evaluaciones reales y que la solución mostrada sea la que realiza un alumno para que sea más próxima en su forma de trabajar a la del lector esperado.

Se ha ilustrado los elementos de cada ejemplo que suelen ser más problemáticos y se quiere así también que el usuario tenga una pequeña curiosidad e implemente el resto.

3 Introducción

La aplicación realizada se estructura en base a la interfaz que el propio MC utiliza para agrupar los tutoriales que incorpora y que se muestra en la fig. 1. El código y el aprendizaje ya empieza aquí, para realizar este índice se ha creado una pila como la indicada y se han dispuesto tantos botones como ejemplos contiene. Cada botón ejecuta casi el mismo código, así que sólo se muestra el del primero en el listado 1.

```
on mouseUp
  go to stack "Pregunta 1"
end mouseUp

local wheat
on mouseEnter
  set the backColor of me to wheat
  set the backColor of me to orange
  set the visible of field "f1" to true
end mouseEnter

on mouseLeave
  set the backColor of me to wheat
  set the visible of field "f1" to false
end mouseLeave
```

Listado 1. Ejemplo de código de uno de los botones de la presentación de la aplicación.

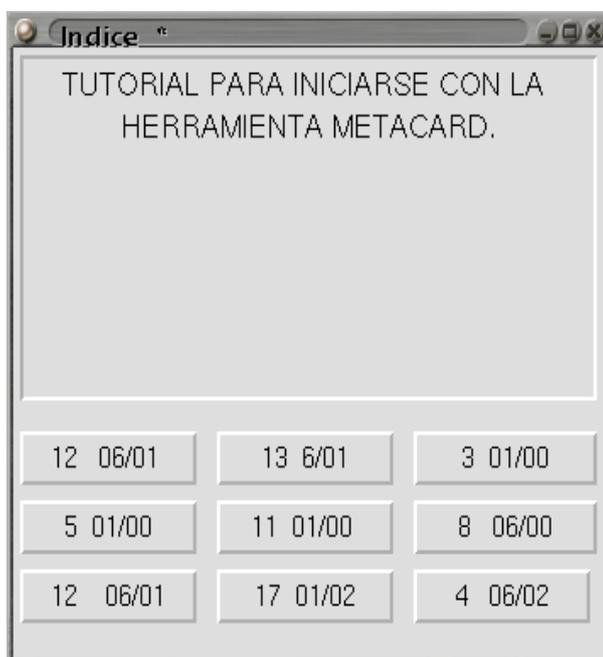


Figura 1: Índice de la aplicación recopilatoria de ejercicios de examen.

4 Desarrollo

Cada botón de la pila principal, fig. 1, lanza una subpila con un ejemplo. Veamos algunos de estos en este apartado. Cada ejemplo está compuesto por dos pilas siguiendo el formato de los tutoriales de MC. La primera es la documentación, que describe cómo se ha realizado cada ejemplo. En esta pila se utilizan una serie de botones para moverse entre los diferentes apartados de la explicación. En cada apartado otro conjunto de botones permite moverse entre los pasos de una explicación, fig. 2, obtener una versión HTML en disco del contenido de las tarjetas (el del logotipo del W3C) y otro para obtener una salida en texto plano (el del icono de un bloc de notas). Estos botones utilizan, respectivamente, el código del listado 2 que, puesto que se repite para todos los casos sólo lo mostramos aquí para el primero. Además, desde el botón “Ver ejemplo”, se puede lanzar la segunda pila de cada aparatado: el ejemplo terminado de lo que se quiere desarrollar para verlo en funcionamiento.

```
#Para el botón “botonHTML”
on mouseUp
ask file "Tria el fitxer on guardar el contingut del tutorial ..." with "texto1.html"
if it is not "Cancel" and it is not empty
then
put empty into url ("file:" & it)
repeat with i = 1 to the number of cards of this stack
put the htmlText of field "memoria" of card i & return after url ("file:" & it)
end repeat
end if
end mouseUp

#Código del botón “botonTXT”
on mouseUp
ask file "Tria el fitxer on guardar el contingut del tutorial ..." with "texto1.txt"
if it is not "Cancel" and it is not empty
then
```

```

put empty into url ("file:" & it)
repeat with i = 1 to the number of cards of this stack
  put the Text of field "memoria" of card i & return after url ("file:" & it)
end repeat
end if
end mouseUp

# botón "Ver ejemplo": en cada caso este código debe lanzar a la subpila que corresponda.
on mouseUp
  go to stack "ReproductorAudio"
end mouseUp

```

Listado 2. Ejemplo de código de uno de los botones que permiten exportar el contenido de cada presentación y lanzar la subpila asociada.

4.1 Reproducir un fichero de audio

Se propone desarrollar un *script* en MC que permita reproducir un fichero de audio cuyo nombre se haya introducido en un campo de texto. Esta opción se realizará al activar un botón con la etiqueta "Reproducir". Un ejemplo de lo que aparece en pantalla se ve en la fig. 2.

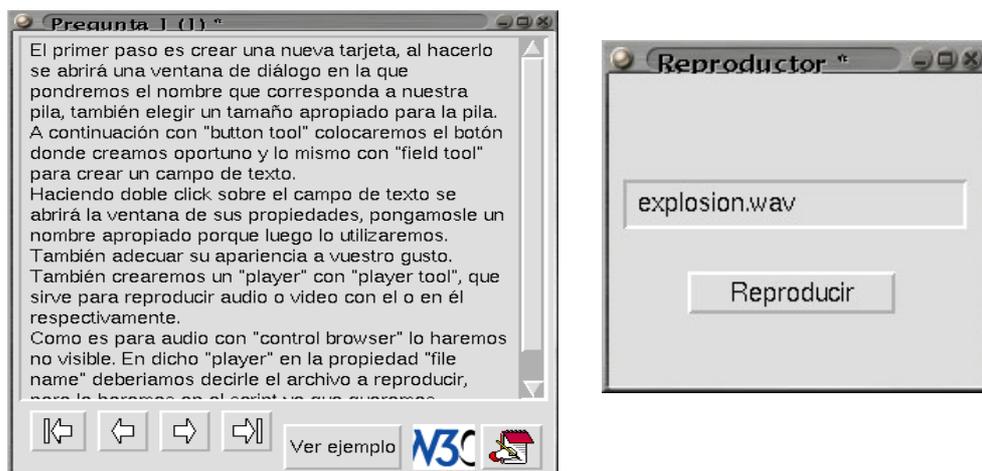


Figura 2: Descripción del ejercicio sobre cómo reproducir un fichero de audio (imagen de la izquierda) y una posible implementación de este ejercicio (derecha).

El primer paso es crear una nueva pila, a la que pondremos por nombre "ReproductorAudio". En ella crearemos un campo de texto y un botón, con una apariencia como la que se muestra en la parte derecha de la fig. 2. El siguiente paso es realizar un *script* para el botón y modificar sus propiedades. Pongámosle el nombre "reproducir" y aprovechad para modificar la apariencia a vuestro gusto. En la pestaña *Behavior* hay un botón que tiene como etiqueta "script". Al pulsar dicho botón aparecerá un editor de texto en el cual aparecen el manejador por defecto. Esto es la secuencia de acciones asociada a un evento, que por defecto para un botón, es la "pulsación" sobre él (*mouseUp*). Dentro de él colocaremos las instrucciones necesarias para que el botón realice lo que nosotros queremos.

Para elegir el archivo a reproducir utilizaremos la orden **set**, que sirve para darle valores a las propiedades de los objetos, en este caso la propiedad a modificar del *player* es *fileName*. Otra propiedad a utilizar es *text* del campo de texto, con esto ya podéis

construir la orden para que el *player* "apunte" al archivo deseado. Para la puesta en marcha del reproductor debéis utilizar la orden **start**. El código de la pila del ejemplo es únicamente el del botón encargado de reproducir el fichero cuyo nombre haya sido tecleado y que se muestra en el listado 3.

```
# botón "Reproducir"
on mouseUp
  set the fileName of player "miplayer" to text of field "tema"
  start player "miplayer"
end mouseUp
```

Listado 3. Ejemplo de código del botón que se encarga de hacer sonar el fichero de audio indicado.

4.2 Buscar y localizar objetos

Durante la realización de un proyecto multimedia en MC se ha perdido un control: un objeto de tipo "button". Esto es, ya no es visible en la pila en pantalla. En este contexto: a) ¿Cómo se puede comprobar que existe tal objeto? b) En caso de no haya sido borrado, dar alguna explicación de la desaparición del control. c) Indicar alguna forma de recuperar el objeto "perdido".

Se puede construir un sencillo ejemplo que muestre estas acciones con dos botones "Desaparecer" y "Volver" que simulen el contexto y las respuestas. Para ello utilizaremos la orden *move*, mirar Metatalk Reference para que os sirva de ayuda para realizar este script. Podéis ver un ejemplo del intefaz en la fig. 3 y del código en el listado 4.

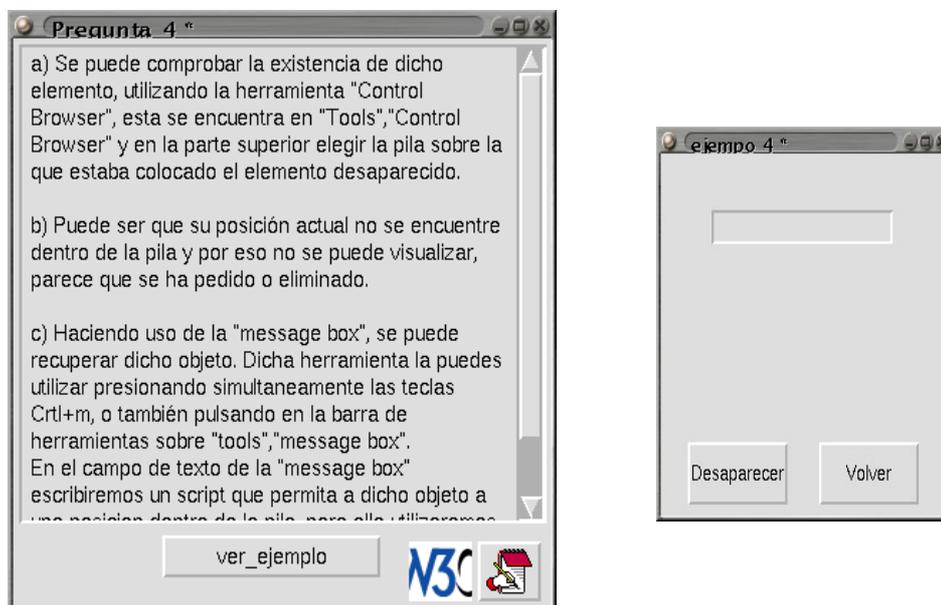


Figura 3: Texto y posible resultado del ejercicio sobre localización de objetos.

Respondiendo al enunciado: a) Se puede comprobar la existencia de dicho elemento, utilizando la herramienta "Control Browser", esta se encuentra en "Tools", "Control Browser" y elegir la pila sobre la que estaba colocado el elemento supuestamente desaparecido. b) Puede ser que su posición actual no se encuentre dentro de la pila y por eso no se puede visualizar, parece que se ha pedido o eliminado. c) Haciendo uso de

la "Message Box", se puede recuperar dicho objeto. Dicha herramienta la puedes utilizar presionando simultáneamente las teclas **Ctrl+m**, o también pulsando en la barra de herramientas sobre Tools | Message Box.

```
#botón "Desaparecer"
on mouseUp
  move the field "texto"to 800,800
  move the button "boton"to 800,800
end mouseUp

# botón "Volver"
on mouseUp
  move the field "texto" to 125,52
  move the button "boton" to 120,96
end mouseUp
```

Listado 4. Ejemplo de código de los botones del ejemplo de mover objetos.

4.3 Contenido de una imagen asociado a un fichero

¿Como se podría incluir una imagen en una presentación multimedia desarrollada en MC? Para introducir una imagen en una pila se tiene que crear un objeto de tipo **image** con la herramienta *image tool* de la "Menu Bar". Al seleccionar esta herramienta, cambia la forma del cursor para identificar la opción seleccionada y se define, interactivamente, la posición y el tamaño que ocupará la imagen en la tarjeta.

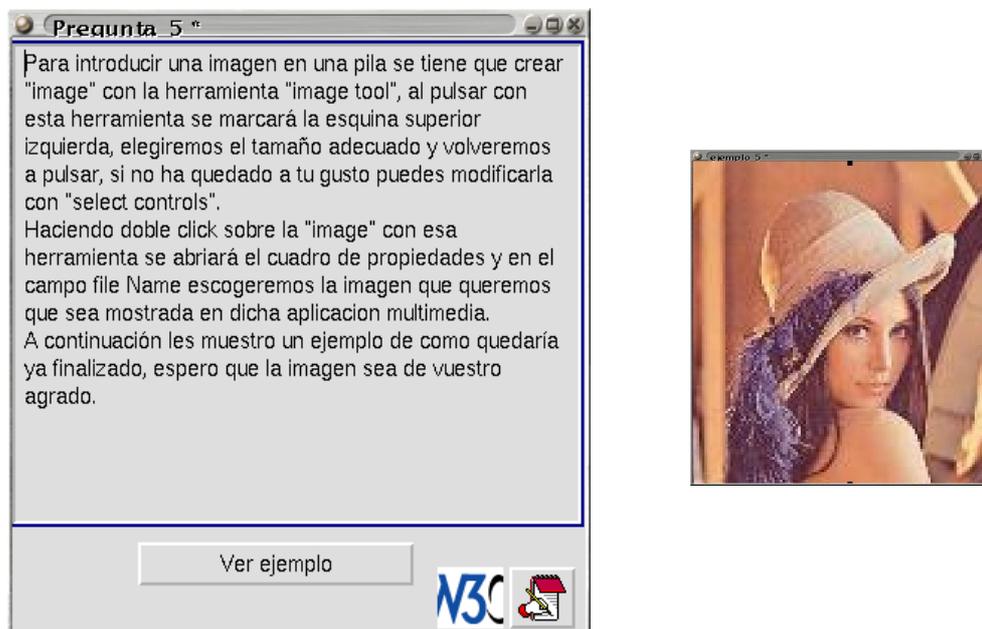


Figura 4: Texto del ejercicio sobre cómo incluir una imagen y un posible resultado.

Haciendo doble click sobre la imagen, se abrirá el cuadro de propiedades y en el campo *fileName* escogeremos la imagen que queremos que sea mostrada en dicha aplicación; reescribiendo la ruta al fichero que contiene la imagen en forma relativa. La fig. 4 muestra un ejemplo de cómo quedaría.

No existe código asociado a este ejemplo, únicamente la demostración del flexible funcionamiento de una ruta a un fichero externo introducida en modo relativo: la del

fichero imagen en este caso, que permite mover el ejemplo a otra localización o cambiar el fichero (reutilizando el nombre) y ver que sigue funcionando la aplicación.

4.4 Reduciendo el número de instrucciones

Para llevar a cabo una animación suele ser necesario utilizar un alto número de instrucciones. Este ejemplo muestra como se puede reducir el número de instrucciones necesarias cuando la repetitividad del comportamiento es patente utilizando bucles para nombrar (recorrer) los diferentes elementos animar.

Repartidas por una tarjeta. véase fig. 5, en una pila realizada con MC, se dispone de veinticinco objetos (de tipo "image" y de nombre "gota1" a "gota25") visibles. Se plantea simular que son tragadas, una a una para simplificar, por un objeto (también de tipo "image" y de nombre "desague") al recibir un evento de ratón sobre dicho objeto. Todo el código se debe realizar dentro del *script* del objeto "desague" y, al terminar el efecto, Sólo debe quedar a la vista el "desague". Entonces sonará un fichero de audio importado, de nombre "burp.wav", con 22 KHz, ocho bits y sonido monofónico.

El primer paso para realizar este ejercicio es crear dos imágenes con "image tool" y en ellas la propiedad "file name" asignarle la imagen deseada para la gota en una y, en la otra, el desagüe. La primera se clonará 24 veces para tener todas las gotas pedidas. Además se ha de crear un *player*, su propiedad "file name" apuntará al fichero "burp.wav" y la visibilidad del mismo a "no visible". Vamos a añadir un botón "Tragar" y otro "Colocar" para albergar el código necesario.

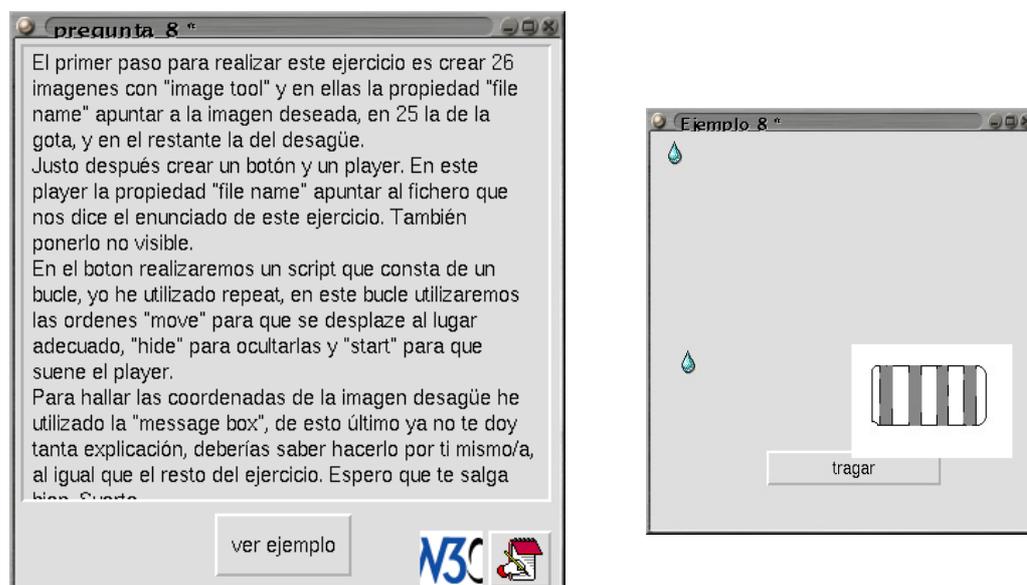


Figura 5: Ejercicio al respecto de la forma de nombrar objetos para facilitar la reducción de líneas de código necesarias para referenciarlos.

En el botón "Tragar" hay que asignar un *script* que consta de un bucle, p. ej., **repeat**. En este bucle utilizaremos las órdenes **move** para que se desplace al lugar adecuado, **hide** para ocultarlas y **start** para que suene el *player*. Para hallar las coordenadas de la imagen del desagüe he utilizado la propiedad *location* del objeto imagen. El botón "Colocar" las pondrá todas en la misma posición para que sólo se vea una.

El listado 5 muestra cómo es posible hacer uso de las propiedades del lenguaje para realizar un alto número de acciones repetidas sobre un número de objetos. En realidad el enunciado pide que los objetos (las imágenes) se llamen "gota1" a "gota25". Dejo de parte del lector interesado modificar el código de Victor para que realmente haga uso de este esquema de nombre y el operador de concatenación: **&**.



```
# botón "Tragar"
local i
on mouseUp
  repeat with i = 25 down to 1
    move image i to 36,202
    set the filename of player "miplayer" to popupblock.wav
    start player "miplayer"
    move image ("gota"&i) to the location of image "desague"
    hide image i
  end repeat
  send mouseUp to button colocar
end mouseUp

# botón "colocar"
local i
on mouseUp
  repeat with i = 25 down to 1
    move image ("gota" &i) to 22,17
    show image i
  end repeat
end mouseUp
```

Listado 5. Ejemplo de código de uno de los botones del ejemplo de reducir líneas de código.

4.5 Cuestionario

La tarea de este apartado es permitir a un alumno contestar a una serie de preguntas y que permita generar un fichero con las respuestas para poder enviar el resultado por correo, por ejemplo, para su evaluación.

La fig. 6 muestra ejemplos de uso de esta aplicación: las cuatro instantáneas corresponden a la presentación del cuestionario, dos de las preguntas de este y, al final, se cierra con la opción de guardar las respuestas.

Para recoger las respuestas al cuestionario, el listado 6 muestra una posible solución basada en la exportación del contenido de los tutoriales en los ejemplos precedentes, ese es el código que corresponde a los botones de la captura inferior derecha de la fig. 6.

```
# Boton W3C
on mouseUp
  ask file "Tria el fitxer on guardar el contingut del tutorial ..." with "respuestas.html"
  if it is not "Cancel" and it is not empty
  then
    put empty into url ("file:" & it)
    repeat with i = 1 to the number of cards of this stack
      put the htmlText of field "respuesta" of card i & return after url ("file:" & it)
    end repeat
  end if
end mouseUp
```

```
# Boton "bloc de notas"
on mouseUp
ask file "Tria el fitxer on guardar el contingut del tutorial ..." with "respuestas.txt"
if it is not "Cancel" and it is not empty
then
  put empty into url ("file:" & it)
  repeat with i = 1 to the number of cards of this stack
    put the Text of field "respuesta" of card i & return after url ("file:" & it)
  end repeat
end if
end mouseUp
```

Listado 6. Ejemplo de código de uno de los botones de la presentación de la aplicación.

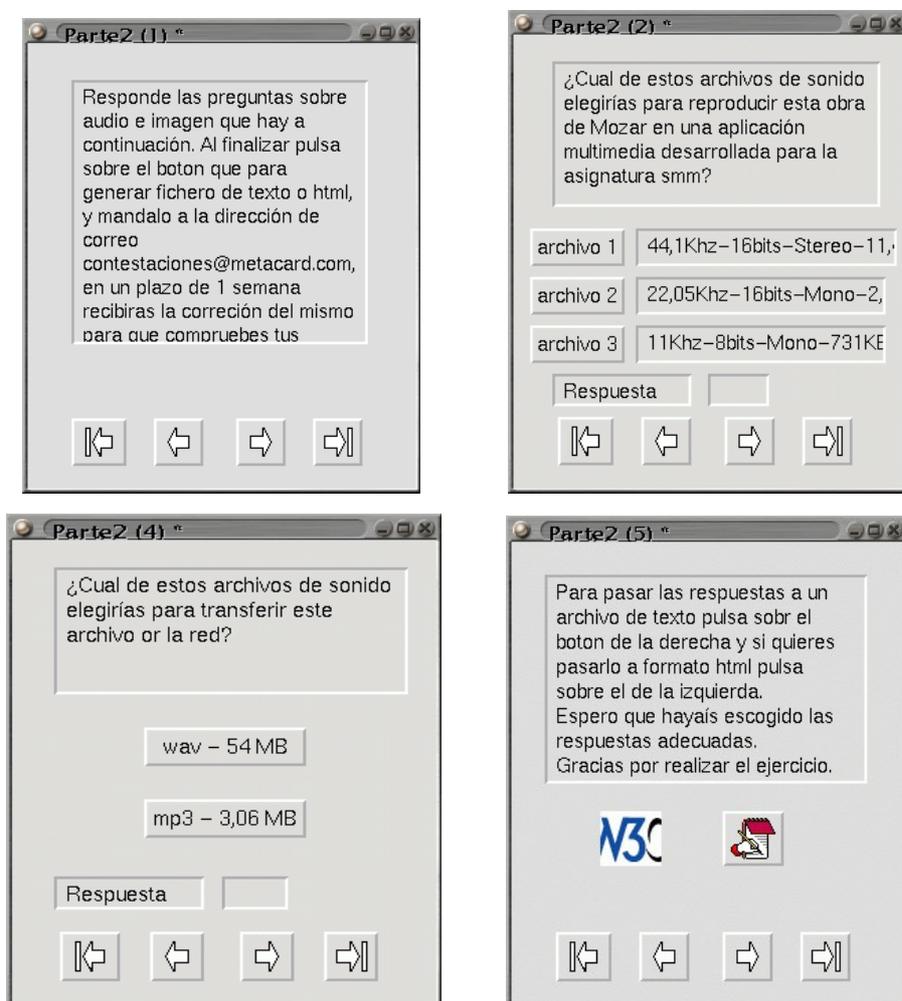


Figura 6: Ejemplo de apariencia de las tarjetas del cuestionario, de arriba a la izquierda hacía abajo a la derecha: presentación; preguntas del cuestionario y guardado de las respuestas.



5 Conclusión y trabajos futuros

En este trabajo se han mostrado los resultados obtenidos por las respuestas de alumnos en prácticas y exámenes, comprobados y ampliados, pero manteniendo básicamente la forma de explicación de aquellos para que el lector pueda encontrar la explicación con las palabras, la forma de emplear los controles y el código sea más cercana, posiblemente, al esquema mental que se está formando en su mente.

Admito que el código es mejorable, pero sobre todo lo es la interfaz. En cualquier caso no nos critique demasiado, amigo lector, la intención es ofrecer una solución más dinámica a los problemas que aparecieron en exámenes y que sean accesibles de una forma práctica.

Hay algunas cosas que se nos han quedado por el camino:

- Tener una contrapartida para los botones que generan la versión HTML (o en texto) del tutorial, para que pueda también existir la opción de asignar el texto a partir del contenido de un fichero.
- Las cajas de diálogo debería sustituir a la escritura de un nombre de fichero en un campo de texto, para facilitar la tarea del usuario.

Por cuestiones de espacio se ha quedado fuera ejemplos de modificar el tamaño del texto, mover y ocultar objetos y un “castillo de fuegos artificiales”. Si se anima el lector ... Tanto si lo hace como si no, puede enviar un correo a la dirección de correo electrónico que figura en la portada de este documento y se los presentaremos.

No quiero acabar sin antes agradecer a V. J. Mazcuñan su colaboración en la elaboración de parte del material gráfico y de parte del código que acompaña a este artículo como resultado de su trabajo en la asignatura de Integración de Medios Digitales.

6 Bibliografía

[1] MC. Disponible en <<http://www.MC.com/>>

[2] Run Revolution Ltd. Disponible en <<http://www.runrev.com/>>.