Universidad Politécnica de Cartagena

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

Ingeniería Técnica de Telecomunicaciones, especialidad Telemática

PROYECTO FIN DE CARRERA

Nuevas tecnologías en docencia: Creación de actividades flash para la materia Telemática

AUTORA: Raquel Andreu Ballester
DIRECTORA: María Dolores Cano Baños

Cartagena, Diciembre 2011

Universidad Politécnica de Cartagena

INICIO

CAPITULO 1. INTRODUCCION

CAPITULO 2. PROGRAMANDO EN FLASH

CAPITULO 3. ANIMACIONES FLASH PARA LA ASIGNATURA TELEMATICA

CAPITULO 1. INTRODUCCION

1.1. OBJETIVOS

Por definición, el *e-Learning* es el suministro de programas educacionales y sistemas de aprendizaje a través de medios electrónicos. El *e-Learning* se basa en el uso de una computadora u otro dispositivo electrónico para proveer a las personas de material educativo.

El término de *e-Learning* o educación electrónica abarca un amplio paquete de aplicaciones y procesos, como el aprendizaje basado en Web, capacitación basada en computadoras, salones de clases virtuales y colaboración digital (trabajo en grupo).
Técnicamente, el *e-Learning* es la entrega de material educativo vía cualquier medio electrónico, incluyendo Internet, Intranets, Extranets, audio, vídeo, red satelital, televisión interactiva, CD y DVD, entre otros medios.

El *e-Learning* fue desarrollado a partir de mediados de los 90 en lo que se ha dado en llamar Plataformas de *e-Learning* o LMS (*Learning Management Systems*, Sistemas de Gestión del Aprendizaje), que agrupan funcionalidades de gestión y distribución de contenidos formativos, herramientas de comunicación y utilidades para el seguimiento en un entorno.

Mediante este formato, el estudiante o el trabajador tiene acceso a cursos interactivos y multimedia, que también pueden apoyarse con medios de comunicación que permitan la colaboración y discusión online de las materias estudiadas y debido a ello, se han ido superado las dificultades surgidas de las fronteras de espacio y tiempo, de tal manera que los alumnos pueden aprender lo que quieran, donde quieran y cuando quieran.

En la enseñanza presencial es el profesorado el que determina casi exclusivamente el ritmo de aprendizaje, pues decide la cantidad de materia que se explica cada vez. Si a esto se suma este tipo de aprendizaje electrónico, convierte al alumno en el centro de una formación independiente y flexible, ayudándole a completar la enseñanza presencial.

Son varias las ventajas que hacen que este tipo de aprendizaje se expanda con mayor rapidez:

- Su facilidad de uso, el hecho de que se puede acceder a todo tipo de documentos multimedia

- Costos muy bajos tanto para el alumno como para el profesor que genera contenidos. Además, la gran mayoría de centros de investigación y universidades permite acceder a la mayor biblioteca de publicaciones electrónicas jamás soñada con enormes posibilidades de interactividad mediante lenguajes tipo Java, o programas como Flash o Shockwave de Macromedia.

- Mayor productividad: Las soluciones de aprendizaje electrónico permiten a los alumnos estudiar desde su propio escritorio. La entrega directa de los cursos puede disminuir los tiempos muertos que implican una escasa productividad y ayuda a eliminar costos de viajes.

- Entrega oportuna: Durante la puesta en marcha de un nuevo producto o servicio, el *e-Learning* puede proveer entrenamiento simultáneo a muchos participantes acerca de los procesos y aplicaciones del nuevo producto. Un buen programa de *e-Learning* puede proveer la capacitación necesaria justo a tiempo para cumplir con una fecha específica de inicio de operaciones.

- Capacitación flexible: Un sistema *e-Learning* cuenta por lo general con un diseño modular. En algunos casos, los participantes pueden escoger su propia ruta de aprendizaje.

En el caso de los "inconvenientes", se presentan muy pocos. Por ejemplo:

- Requiere más inversión de tiempo por parte del profesor.

- Precisa unas mínimas competencias tecnológicas por parte del profesor y de los estudiantes.

- Requiere que los estudiantes tengan habilidades para el aprendizaje autónomo

En este proyecto se pretende combinar la enseñanza presencial con una ayuda de aprendizaje electrónico por medio de CD-ROM.
En este CD-ROM encontraremos innovadoras técnicas multimedia donde hemos desarrollado animaciones, ejercicios y test que permiten una mejor proyección del alumno y también con él se ayuda al profesor a complementar su enseñanza para obtener mejores resultados.

En concreto, el objetivo de este proyecto es diseñar e implementar con la herramienta Flash un CD interactivo que contenga actividades de *e-Learning* para facilitar a los alumnos un mejor entendimiento de la materia Telemática, por ejemplo: actividades drag&drop, cuestionarios tipo test, resolución interactiva de problemas, etc. De este modo, se pretende ayudar al estudio de la asignatura gracias al *e-Learning*, donde a través de las actividades multimedia se explicará con representaciones visuales aquellas definiciones más complejas y de difícil comprensión. En el caso de los problemas, la posibilidad de ver la resolución de los mismos paso a paso en cualquier momento (en casa, en la biblioteca, etc.) ayudará a mejorar el nivel de comprensión, por ejemplo evitando aquellas dudas en las cuales no se sabe seguir de un paso a otro sólo con los apuntes tomados en clase. Además, desde el punto de vista de los profesores, ayudará a completar su material de enseñanza y facilitar sus explicaciones.

1.2. ESTRUCTURA Y DESCRIPCION DEL PROYECTO

Este proyecto está formado por los siguientes capítulos:

- En el capítulo 2 se explica cómo realizar animaciones con Flash. Específicamente se verá, el entorno de trabajo, en el cual vamos a trabajar e iremos explicando la realización de las diferentes animaciones y ejercicios, así como el código ActionScript que se necesita en cada una de ellas.
- En el capítulo 3 se detallan todos los ejercicios y animaciones creadas en cada tema para su uso como herramienta de apoyo de la materia Telemática. Donde a través de capturas de pantalla, explicaremos todo lo realizado y su función.
- Por último, en el capítulo 4 explicaremos las conclusiones del trabajo realizado y los objetivos alcanzados.

CAPITULO 2. PROGRAMANDO EN FLASH

2.1. INTRODUCCION

Flash es una herramienta de edición con la que podemos crear presentaciones, aplicaciones, gif animados, formularios y otro tipo de contenido que permite la interacción del usuario.

Flash le proporciona todo lo necesario para crear y publicar complejas aplicaciones de grandes prestaciones y contenido Web. Tanto si diseña gráficos con movimiento como si crea aplicaciones gestionadas por datos, Flash tiene las herramientas precisas para producir excelentes resultados y ofrecer al usuario la posibilidad de utilizar los productos en distintas plataformas y dispositivos.

Esta aplicación permite llevar a cabo animaciones de poco peso, es decir, que tardan poco tiempo en ser cargadas por el navegador y además hace que estas creaciones sean rápidas y fáciles de realizar gracias a su mecanismo.

Para crear una aplicación en Flash, se crean gráficos con las herramientas de dibujo y se importan elementos multimedia adicionales al documento de Flash.

En los gráficos una imagen es representada a partir de líneas (o vectores) que poseen determinadas propiedades (color, grosor…). La calidad de este tipo de gráficos no depende del zoom o del tipo de resolución con el cual se esté mirando el grafico. Por mucho que nos acerquemos, el grafico no se pixela, ya que el ordenador traza automáticamente las líneas para ese nivel de acercamiento.

Flash permite controlar las interpolaciones, donde se pueden modificar las velocidades de los fotogramas así como la forma, color, movimiento de nuestras interpolaciones.

ActionScript es el lenguaje de programación de Macromedia Flash, le permite añadir interactividad a una película. ActionScript proporciona elementos, como acciones, operadores y objetos, que se combinan en scripts que indican a las películas qué deben hacer en cada momento; las películas se configuran de manera que determinados eventos, como pulsar un botón o presionar una tecla, activan tales scripts. Así, por ejemplo, puede utilizar ActionScript para crear botones de navegación en una película.

Flash incluye muchas funciones que la convierten en una herramienta con numerosas prestaciones sin perder por ello la facilidad de uso. Entre dichas funciones destacan la posibilidad de arrastrar y soltar componentes de la interfaz de usuario creados previamente, comportamientos integrados que permiten añadir fácilmente código ActionScript al documento y varios efectos especiales que pueden incorporarse a los objetos multimedia.

Flash almacena sus archivos con varias extensiones. La extensión ".fla" contiene el programa fuente mientras que los archivos ".swf" contienen el gráfico que será mostrada en la web.

Una vez que se ha terminado de editar el documento de Flash, se puede publicar a través del comando Archivo > Publicar. De este modo, se crea una versión comprimida del archivo con la extensión .swf (SWF). A continuación, se puede utilizar Flash Player para reproducir el archivo SWF en un navegador Web o como una aplicación independiente.

Los documentos de Flash se componen de varias partes que explicaremos a continuación.

2.2. ENTORNO DE TRABAJO

Los documentos de Flash8 se componen de un entorno o interfaz de trabajo como se puede ver en la *Figura 2.1*, las cuales explicaremos a continuación.

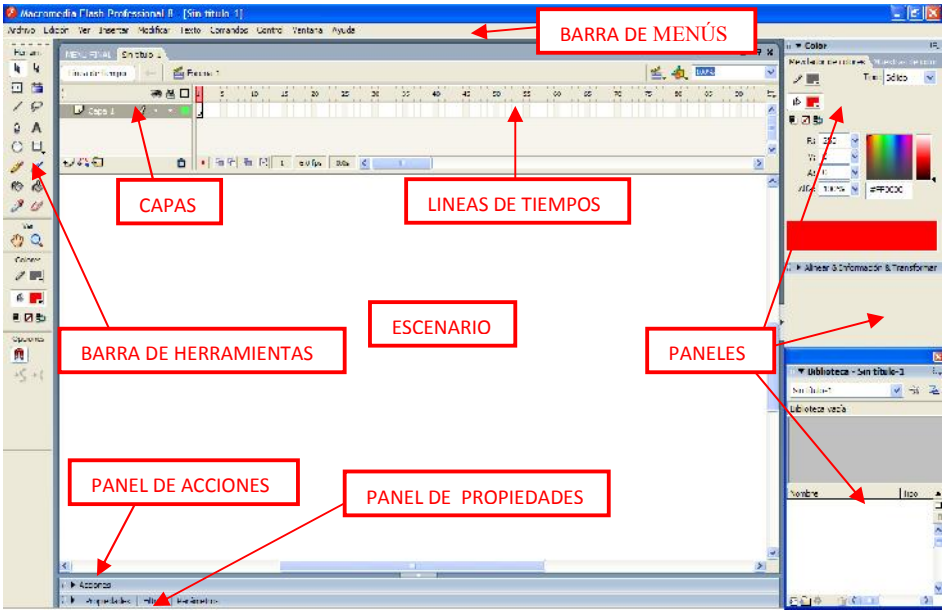


Figura 2.1

2.2.1 BARRA DE MENUS.

La barra de menús está situada en la parte superior de la ventana de aplicación Flash, muestra menús con comandos que sirven para controlar las funciones del programa.

Los menús son los siguientes, los cuales observamos en la *Figura 2.2:* Archivo- Edición- Ver- Insertar- Modificar- Texto- Comando- Control- Ventana- Ayuda.

Figura 2.2

2.2.2 BARRA DE HERRAMIENTAS

Las herramientas del panel de herramientas permiten dibujar, pintar, seleccionar y modificar ilustraciones, así como cambiar la visualización del escenario. El panel de herramientas se divide en cuatro secciones, lo podemos ver en la *Figura 2.3*:

- La sección de herramientas contienen las herramientas de dibujo, pintura y selección.
- La sección de visualización contiene las herramientas para ampliar, reducir así como para realizar recorridos de la ventana de aplicación.
- La sección de colores contiene modificadores de los colores de trazo y relleno.
- La sección de opciones muestra modificaciones de la herramienta actualmente seleccionada

Los modificadores afectan a las opciones de pintura o edición de la herramienta.

Figura 2.3

2.2.3 ESCENARIO

El escenario es el área rectangular donde se coloca el contenido grafico que incluye entre otros: gráficos vectoriales, cuadros de texto, botones o imágenes de mapas de bits importadas. El escenario del entorno de edición de Flash representa el espacio rectangular de Macromedia Flash Player o del navegador Web donde se muestra el documento de Flash durante la reproducción. Puede utilizar las funciones de acercar y alejar para ver el escenario cuando trabaja.

La cuadricula, guías y las reglas sirven para colocar con precisión el contenido en el escenario.

2.2.4 LINEA DE TIEMPO

La línea de tiempo organiza y controla el contenido de un documento a través del tiempo en capas y fotogramas. Al igual que en un largometraje los documentos de Flash dividen el tiempo en fotogramas. Las capas son como varias bandas de película apiladas unas sobre otra, cada una de las cuales contienen imágenes diferentes que aparecen en el escenario.

Los componentes principales de la línea de tiempo son las capas, fotogramas y la cabecera lectora. Estos componentes los podemos observar en la *Figura 2.4.*

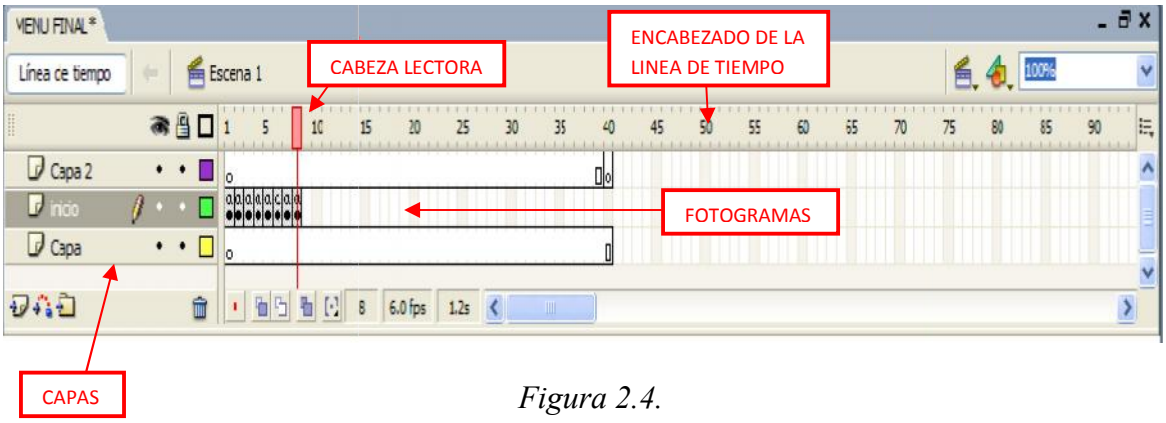


Figura 2.4.

Como hemos explicando antes , las capas son hojas apiladas en el escenario las cuales ayudan a organizar las ilustraciones de los documentos.
Los objetos de una capa pueden dibujarse y editarse sin que afecten a obletos de otras capas. Cuando una capa está vacía , las capas situadas debajo pueden verse a través de esta. Para dibujar una capa primero debe seleccionarse en la línea de tiempo para activarla. Solo se puede activar una capa en cada momento.

2.2.5 PANEL DE PROPIEDADES

El panel de propiedades simplifica la creacion de documentos facilitando el acceso a los atributos mas utilizados del elemento seleccionado, ya sea en el escenario o en la linea de tiempo. Puede modificar los atributos del objeto o documento en el panel de propiedades sin acceder a los menus o paneles que contienen estos atributos

En este panel se muestra información y la configuración del elemento que está seleccionado, como puede ser un texto,como se muestra en la *Figura 2.5.*, un objeto,un mapa de bits, un fotograma…

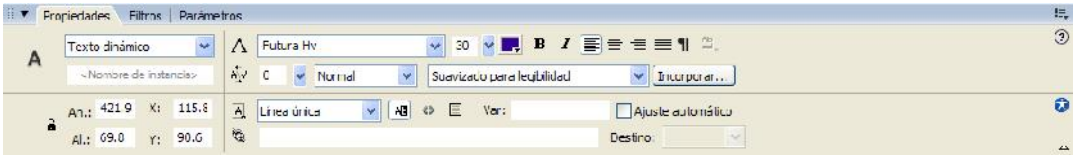


Figura 2.5.

2.2.6 PANEL DE ACCIONES

Este panel , *Figura 2.6*, permite crear y editar codigo ActionScrip para un objeto o fotograma. El panel acciones se activa cuando se selecciona una instancia de un fotograma , boton o clip de pelicula.

La utilización de los controles del panel Acciones en modo Normal permite insertar acciones sin tener que utilizar ActionScript; si es un usuario experto en ActionScript puede construir sus propios scripts. Las instrucciones pueden estar formadas por una sola acción, como solicitar la detención de la reproducción de una película, o bien por una serie de acciones, como primero evaluar una condición y, a continuación, realizar una acción. La configuración de muchas de las acciones requiere poca experiencia en programación. Para otras acciones, es necesaria cierta familiaridad con los lenguajes de programación y están diseñadas para un desarrollo avanzado.

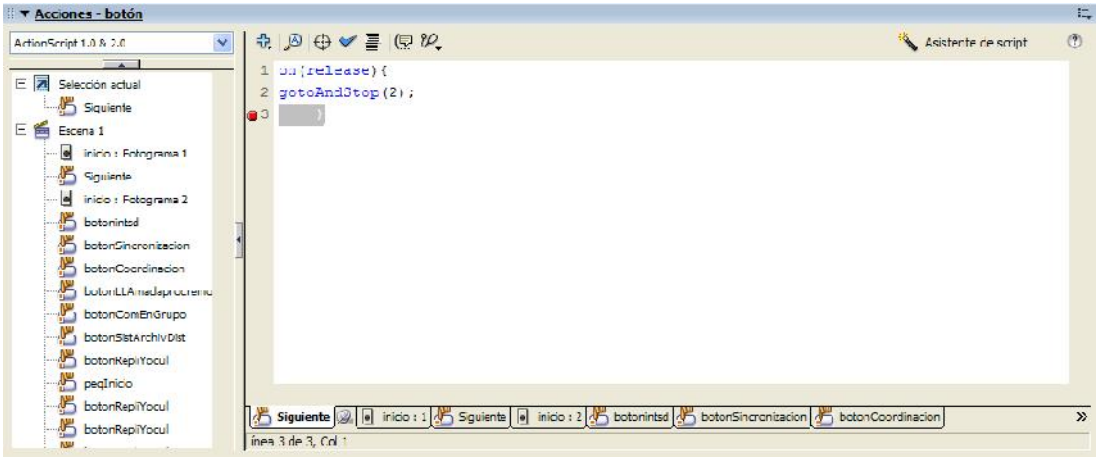


Figura 2.6

2.2.7 PANEL BIBLIOTECA

El panel Biblioteca es donde se guardan y organizan los simbolos creados en Flash, ademas de archivos importados tales como gráficos de imágenes de mapas de bits. En el panel de biblioteca se pueden organizar carpetas con los elementos y ver con qué frecuencia se utilizan y ordenarlas por tipos, *Figura 2.7*

Una **biblioteca** no es más que un almacén de objetos (gráficos o sonidos) que podrán ser utilizados en una misma animación en una o más ocasiones. Dependiendo del rango que presente esta biblioteca, ésta puede ser propia a la animación, compartida por varias animaciones, o bien permanente (empleada por la totalidad de animaciones).

Cada uno de los elementos que constituyen una biblioteca son denominados **símbolos**. Como

hemos dicho, estos elementos podrán ser utilizados en nuestra animación cuantas veces lo deseemos. No obstante, cada una de estas utilizaciones no es llamada símbolo, sino ocurrencia.

Por lo tanto, una **ocurrencia** es cada una de las ocasiones en las que un símbolo almacenado en nuestra biblioteca es utilizado en nuestra animación.

Cambiando las propiedades de un símbolo de la biblioteca, cambiamos todas cada una de las ocurrencias que aparecen en la animación. Contrariamente, la modificación de una ocurrencia no altera al símbolo de la biblioteca ni a las otras ocurrencias de la animación. Como podemos observar, el uso de las bibliotecas no solo nos ayuda a aligerar el archivo, sino que nos permite una creación, edición y borrado rápidos de cada una de las ocurrencias.

Más adelante abordaremos con más detalle la gestión de bibliotecas símbolos y ocurrencias. Pasaremos a continuación a la aplicación de lo aprendido a partir de la creación de una animación.

Figura 2.7

2.3 CREAR ANIMACIONES

Flash es una tecnología para crear animaciones gráficas vectoriales independientes del navegador y que necesitan poco ancho de banda para mostrarse en los sitios web. La animación en Flash se ve exactamente igual en todos los navegadores, un navegador sólo necesitan un plug-in para mostrar animaciones en Flash.

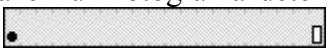
Por animación entenderemos que los objetos que aparecen en la pantalla cambien de posición, tamaño, aspecto, color, que giren, se deformen, etc...

En Flash existen tres tipos de animaciones:

- Animación fotograma a fotograma
- Animación interpolación de movimiento
- Animación interpolación de forma

Consideraciones:

Se llaman Fotogramas Clave a los que contienen cambios en la forma o posición del objeto

Para que un objeto se muestre en pantalla en un fotograma determinado, debe crearse un fotograma (no clave). Se mostrará así:  . El rectángulo blanco significa que no ha habido cambios de forma ni posición.

Podemos mover los fotogramas clave o los del rectángulo para cambiar los puntos de inicio o fin de la animación.

Podemos variar la velocidad de toda la animación.

Interpolar es una manera de decir "intercalar" y significa rellenar los fotogramas que hay entre dos fotogramas clave de manera que un gráfico que se muestra en el primer fotograma clave se convierta en el gráfico que se muestra en el segundo fotograma clave.

A continuación explicaremos los tres tipos de animaciones que hemos utilizado en este proyecto.

2.3.1 ANIMACION FOTOGRAMA A FOTOGRAMA

En la animación fotograma a fotograma, ejemplo *Figura 2.8*, debemos dibujar cada uno de los fotogramas de la animación uno por uno, al estilo de las clásicas películas de dibujos animados. Permite una gran flexibilidad, pero a costa de un elevado esfuerzo. Además, el archivo debe almacenar los cambios de cada fotograma.

La animación fotograma a fotograma, cambia el contenido del escenario en cada uno de ellos, siendo un fotograma clave cada uno de ellos.

En este ejemplo vemos que en cada fotograma clave van apareciendo diferentes imágenes o textos en el escenario.

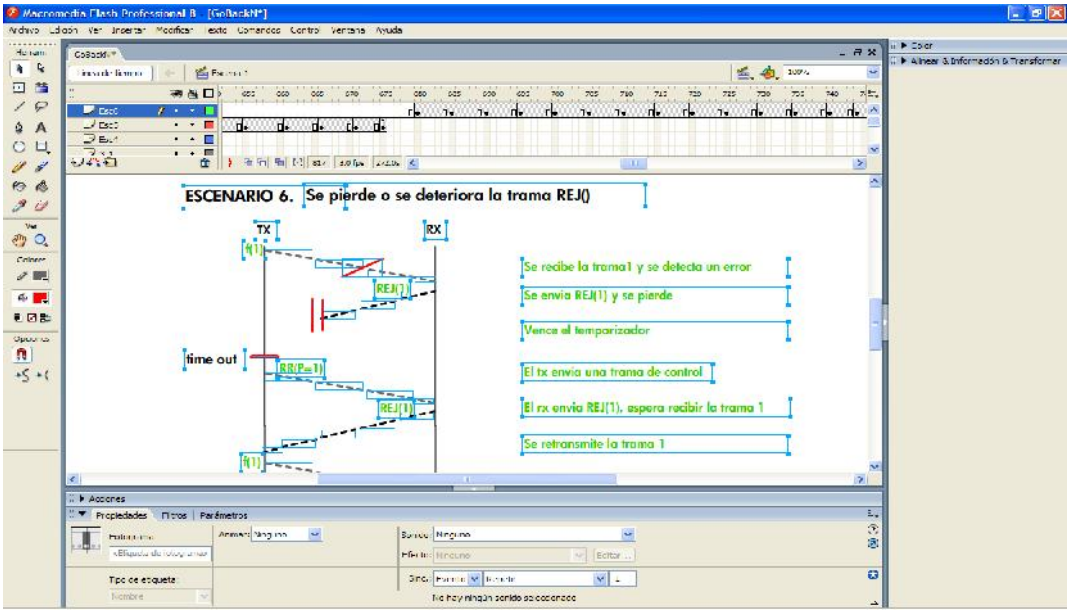


Figura 2.8

2.3.2 ANIMACION POR INTERPOLACION DE FORMA

Con la interpolación de forma, se especifican atributos para una forma en un fotograma clave y, a continuación, se modifica la forma o se dibuja otra forma en un fotograma clave posterior.

Por lo que la animación por interpolación de forma consiste en que gradúa el cambio de la forma del objeto. También permite cambiar el color, como podemos observar en el *Figura 2.9 y Figura 2.10,* en la que varía el color de los ceros en los datos transmitidos.

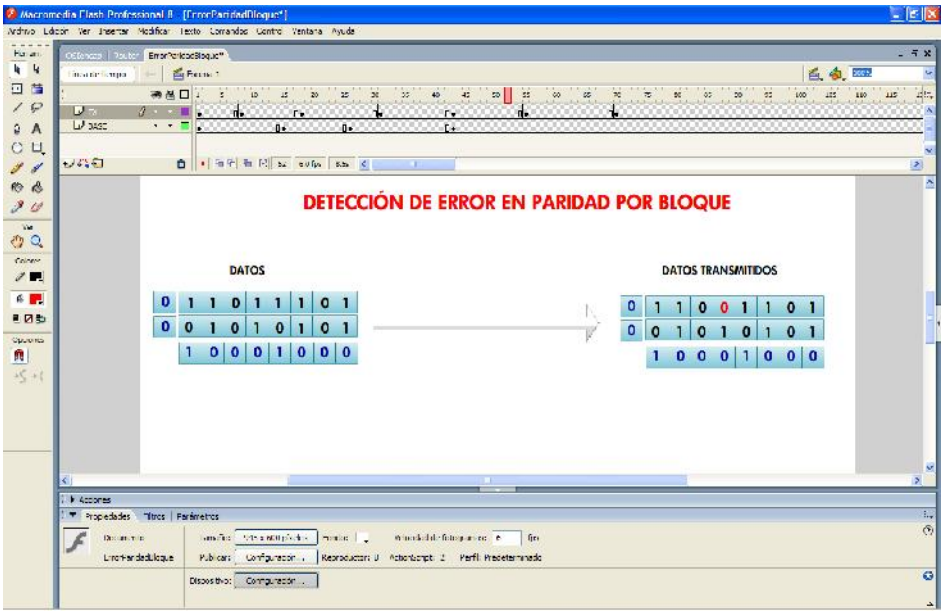


Figura 2.9

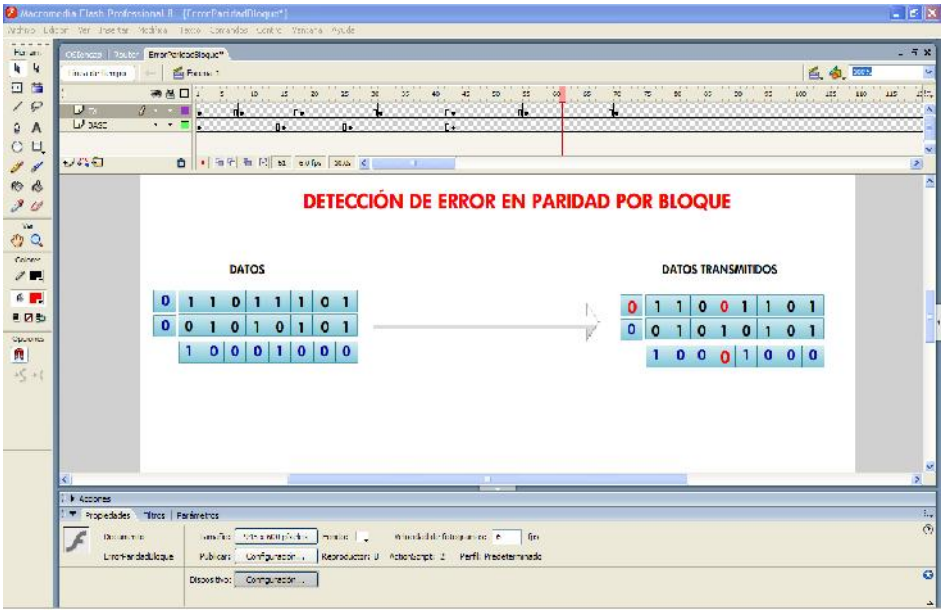


Figura 2.10

2.3.3 ANIMACION POR INTERPOLACION DE MOVIMIENTO

La animación por interpolación de movimiento, *Figura 2.11*, nos permite desplazar objetos de un sitio a otro del escenario, por lo que la usaremos cuando los objetos de los dos fotogramas claves solo se diferencien en su posición, escala o rotación.
Para realizar una interpolación de movimiento debemos crear un fotograma con un objeto, crear el siguiente fotograma moviendo ese objeto al lugar del escenario donde lo queramos poner y después hacer clic al botón derecho sobre dicho fotograma y seleccionar "crear interpolación de movimiento".

La velocidad en el movimiento de las películas la podemos cambiar, modificando su parámetro en la barra de tiempos. Mayor valor más velocidad, pero se debe poner siempre suficientes fotogramas para que se desarrolle la animación como queremos.

En esta imagen vemos como el paquete se va moviendo por el escenario a través de interpolación de movimiento.

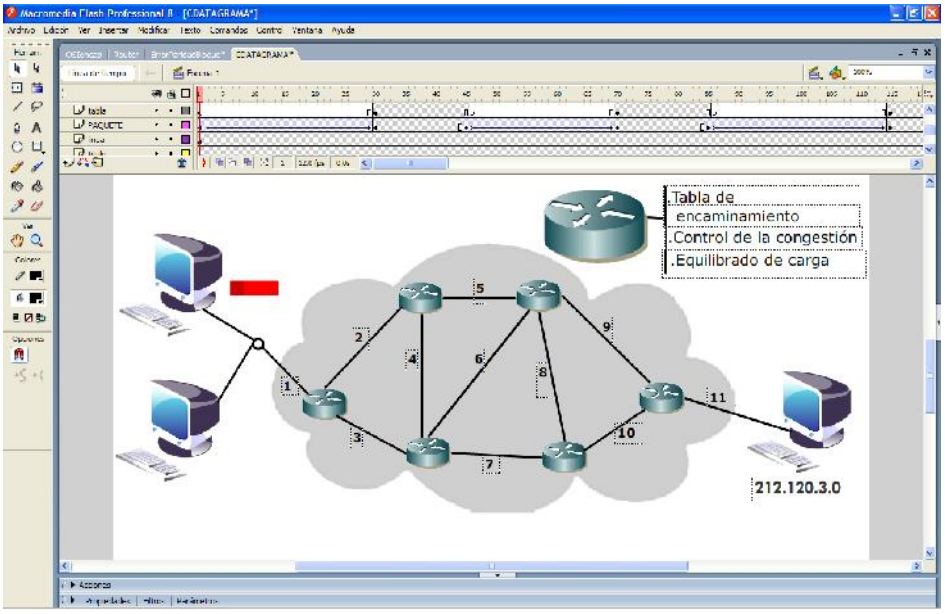


Figura 2.11

2.4 CREAR BOTONES

Los símbolos de tipo Botón son los que aportan la mayor parte de la interactividad a las películas Flash con aquel que las está visualizando.
Para convertir un objeto en botón, se debe seleccionar dicho objeto, hacer clic en el lado derecho del ratón y seleccionar *Convertir en símbolo*, de esta forma obtenemos una ventana como la de la *Figura 2.12*

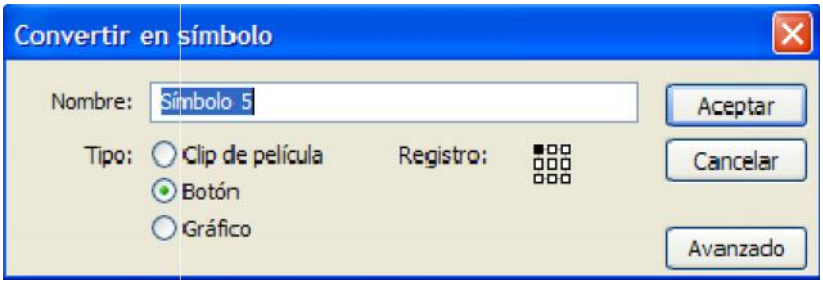


Figura 2.12

El usuario presiona estos botones y desencadena una serie de acciones. También es habitual ver como este tipo de elementos reaccionan cuando se les pasa el ratón por encima o cuando son pulsados.

Cada botón es clip de película interactivo de cuatro fotogramas, si pulsamos doble clic sobre el botón, obtenemos una Línea de tiempo de cuatro fotogramas, como observamos en la *Figura 2.13*, los tres primeros fotogramas muestran los tres posibles estados del botón (reposo, sobre y presionado) en los cuales se podría cambiar el color, forma… según el estado; el cuarto fotograma define el área activa del botón, es decir el área que responde al clic del ratón.

Figura 2.13

2.4.1 BOTON QUE SALTA A UN FOTOGRAMA

En esta animación, *Figura 2.14*, mostramos un fotograma que está parado hasta que no se pulse uno de los botones que observamos en el escenario. Eso ocurre gracias al código stop (); de Action Script escrito en el panel de Acciones.

Para ello seleccionamos el fotograma, pinchamos en el panel de Acciones y escribimos el código, en este caso stop().

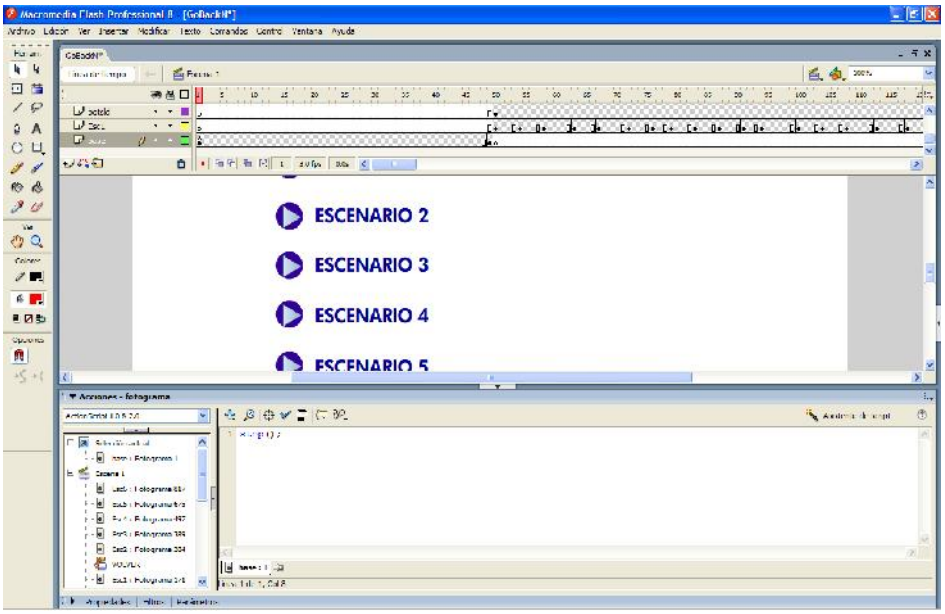


Figura 2.14

En la siguiente imagen, *Figura 2.15*, vemos los botones creados en esta animación al principio de cada texto ``ESCENARIO X´´.

Estos botones están programados para que cuando los seleccionemos salten a un fotograma indicado en el código escrito en el panel de Acciones.

```
on(release){
        gotoAndPlay(175);
}
```

La función ***gotoandPlay(fotograma),*** es probablemente, la más usada durante la realización de la película, para la ejecución de botones de menú. Dicha función saltará al fotograma marcado entre paréntesis en el código ActionScript.

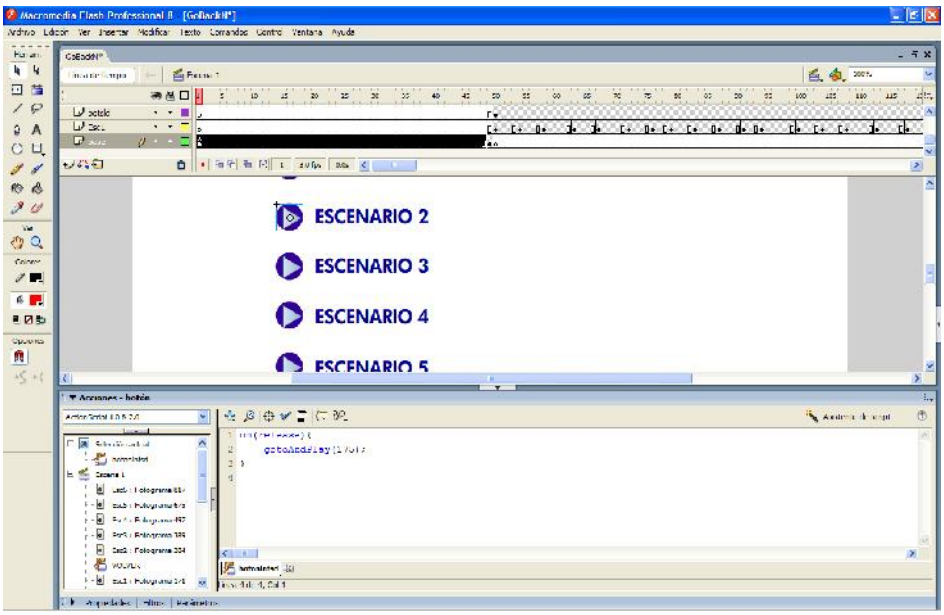


Figura 2.15

2.4.2 BOTON PARA ABRIR OTRA PELICULA FLASH

Podemos crear una película como principal, la cual tenga diferentes botones que al seleccionarlos nos lleven a diferentes películas.

En la imagen que vemos, *Figura 2.16*, observamos que pulsando el botón creado en un imagen se abriría el archivo "TEMA3/CCIRCUITOS.swf"-

Para ello utilizamos el siguiente código:

```
on(release){
        loadMovieNum("TEMA3/CCIRCUITOS.swf",1);
        gotoAndStop(40);
}
```

La función ***loadMovieNum(url, nivel/destino, variables)***, permite cargar nuevas películas Flash o imágenes de forma dinámica.

- url es la dirección donde está situada la película *swf* o la imagen *jpeg*, en este caso el botón carga un archivo *swf* , formato de la película de la animación
- Nivel/destino, nivel donde cargaremos la película, teniendo en cuenta que el nivel básico es el 0, luego el 1 y así sucesivamente. Cada nivel superior se sitúa delante del anterior y toma el control.

Dicho esto, cargaremos nuestro archivo en un nivel superior (1) tomando el control temporalmente, quedándose la otra película, en un nivel inferior, evitando así que se vean las dos películas en el mismo nivel (mezclándose entre sí) y por lo tanto en la misma ventana.

La función ***gotoAndStop(fotograma),*** cuyo objetivo es mandar la cabeza lectora al fotograma y detenerlo, en esta ocasión lo mandará al fotograma 40, en el cual se guarda temporalmente la película anterior.

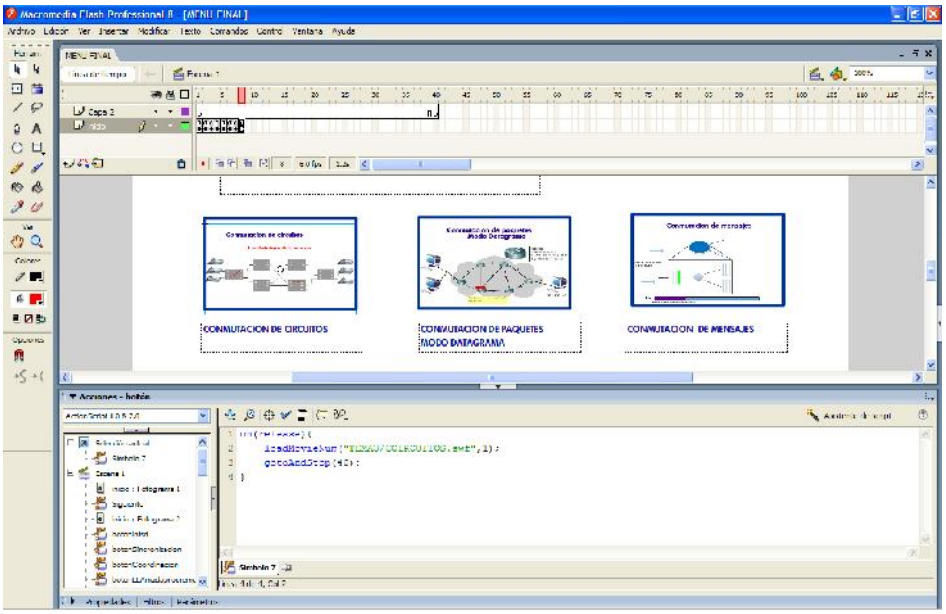


Figura 2.16

2.4.3 BOTON SALIR

En algunas películas hemos añadido un botón de Salir, como vemos en la *Figura 2.17*, vinculado con una ventana de otra película. En el panel de acciones se programaría el siguiente código.

```
on(release){
      _level0.gotoAndStop(4);
      this.unloadMovie();
}
```

- Al pulsar el botón Salir, el código asociado hace que nuestra película principal, pase al fotograma indicado en la otra película. Esto lo realiza la línea de código *_level0.gotoAndStop(fotograma).*
- Con la función *unloadMovie()* se elimina el contenido de una instancia de clip de película, lo que actuará cerrando la película, en concreto la que se reproduce gracias al *this.*

A continuación mostramos un ejemplo de una captación, animación en la que se observa el botón *Salir*, al que nos referíamos en el párrafo anterior.

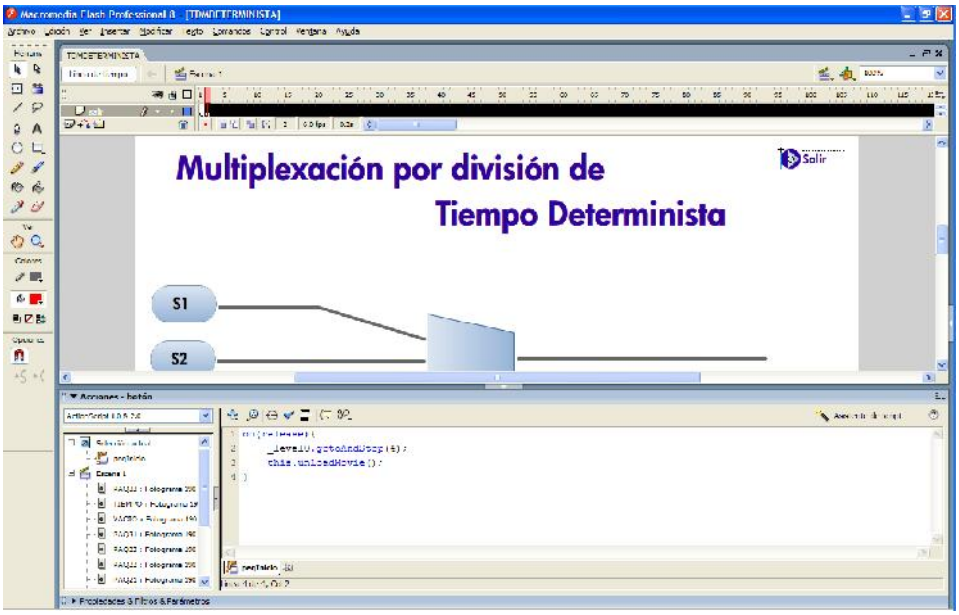


Figura 2.17

2.4.4 BOTON CORREGIR

En algunos ejercicios hemos añadido un botón de corregir, como vemos en la *Figura 2.18*.

Este botón se utiliza para ver si la respuestas seleccionadas en ejercicios tipo Drag&drop o tipo test han sido las correctas.

El código utilizado es el siguiente:

```
on (release) {
        mark();
}
```

Este código llama a la función mark() que está explicada en el apartado 2.5 CREAR UN EJERCICIO DRAG&DROP.

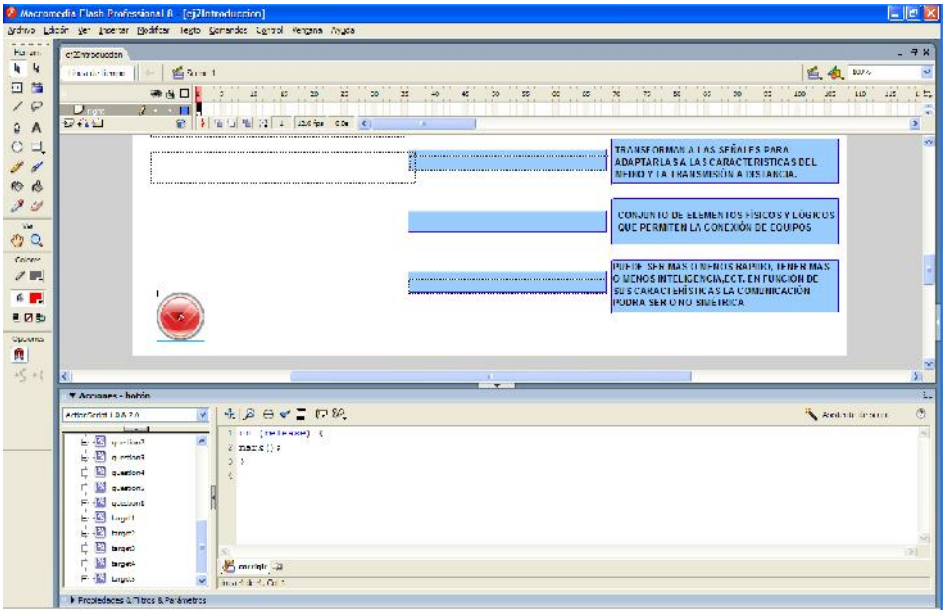


Figura 2.18

2.4.5 BOTON PARA CARGAR UN PDF

La función de este botón es abrir una página de navegador donde aparece el correspondiente *pdf* . *Figura 2.19*

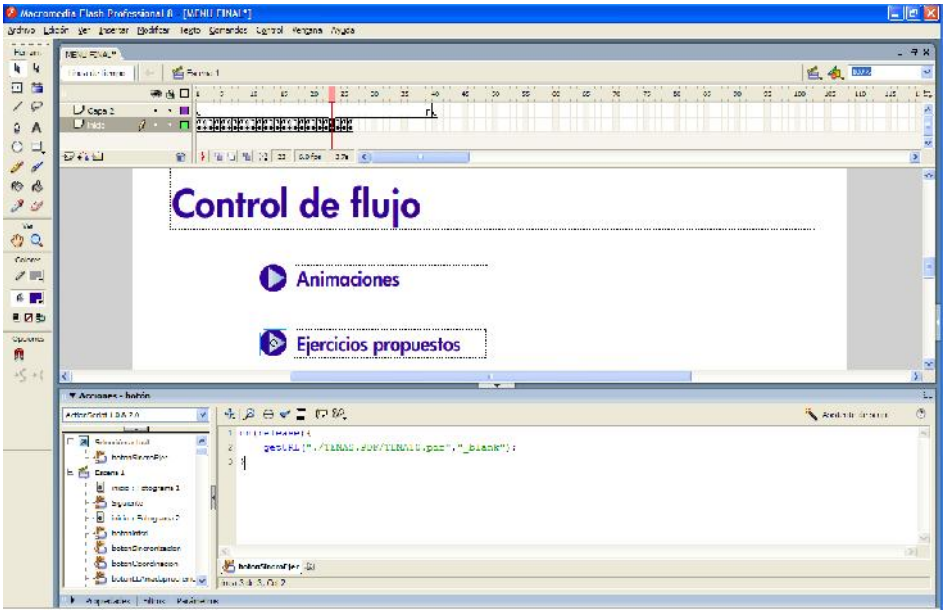


Figura 2.19

El código utilizado para programar dicha acción es el que se muestra a continuación:

```
on(release){
      getURL("./TEMAS.PDF/TEMA10.pdf","_blank");
```

```
}
```

La función *getURL(url, ventana, "variables"),* esta acción se emplea para abrir el navegador web y la página web que deseemos

• El comando *url* obtiene la dirección web a la que queremos acceder, en este caso abrirá la página del navegador con el archivo *pdf* del tema 1 (*T1*) incluido en la carpeta *TEMAS.PDF*, sin cerrar la ventana donde se encuentra el botón, quedándose éste en un segundo plano.

•El parámetro *ventana* es opcional, dependiendo de la forma que se quiera abrir, ya sea en la actual (_self) o en la nueva(_blank)).

•El tercer comando es opcional, si la página lo permite podemos enviarle variables.

2.4.6 FOTOGRAMA DE PRESENTACION

En las animaciones también se podría introducir un fotograma clave, para que aparezca como una presentación y un botón de entrada a la animación, botón *ENTRAR*, utilizando la función *gotoAndPlay(fotograma),* para pasar al siguiente fotograma, como se muestra en la *Figura 2.20*

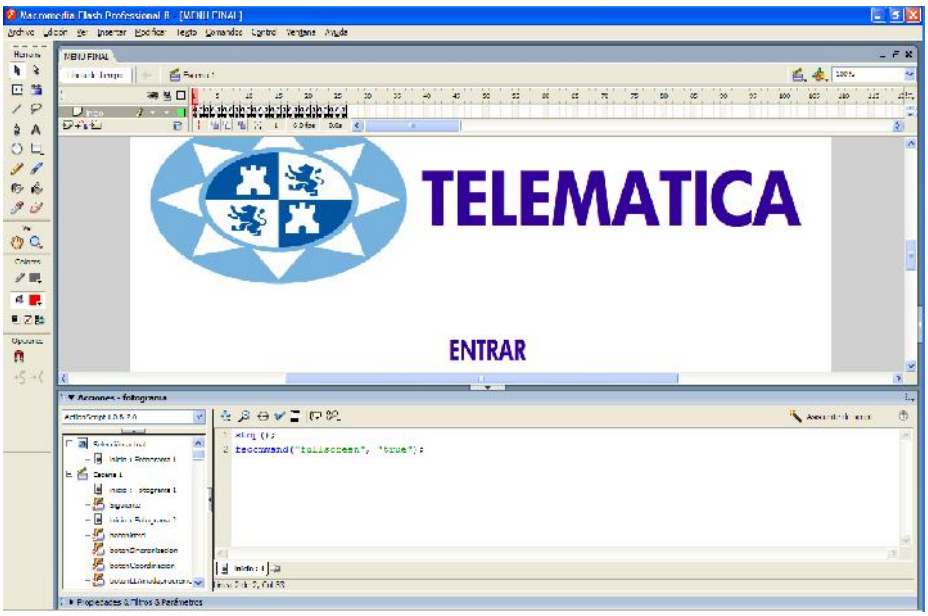


Figura 2.20

Las funciones predefinidas, para realizar lo comentado en el párrafo anterior,

```
stop();
fscommand("fullscreen", "true");
```

- La acción de control de película *stop()* no dispone de parámetros y detiene la reproducción de ésta, pudiendo usarla en un fotograma o en todos ellos, si queremos realizar la acción de detenernos.
- *Fscommand("comando", "true / false")* es capaz de ejecutar ciertos comandos muy potentes.

-El comando a ejecutar va a ser *fullscreen,* utilizado en esta ocasión para poner nuestra película a pantalla completa, muy útil en presentaciones en CD-Rom.
- El segundo comando será true o false según deseemos activar o desactivar la opción.

2.5 CREAR UN EJERCICIO DRAG&DROP

En la realización de un ejercicio tipo Drag and Drop al igual que en todos los proyectos que se realizan en Flash lo más adecuado es trabajar por capas para su fácil manejo.

En los ejemplos de este tipo de ejercicio, tenemos unas capas comunes para ciertos elementos indispensables para el correcto funcionamiento de dicho trabajo. En nuestro caso, uno de los ejercicios que hemos creado es de coger términos y arrastrarlos hasta sus casillas correspondientes en las que están las definiciones de dichos términos y realizar posteriormente su corrección, como muestra la *Figura 2.21*

Figura 2.21

Las capas en las que podemos dividir este ejercicio, Figura *2.22*, son:

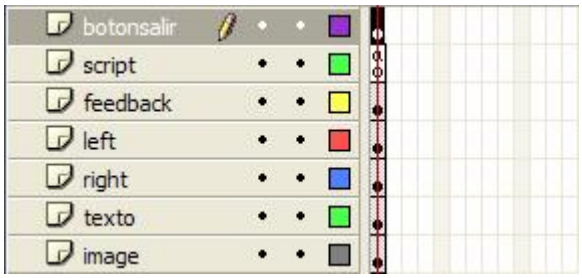


Figura 2.22

1. Capa script. Capa con solo código ActionScript en la que se declaran las variables y algunas funciones que luego harán referencia a botones situados dentro de nuestro escenario. También se declara un array con los términos que vamos a utilizar para enlazar con sus definiciones.
2. Capa left. Capa en la que tenemos *MovieClip* con texto dinámico, que a través de una función irá cogiendo los términos del array y se los va dando aleatoriamente a cada uno de estos *MovieClip*. Así aparecerán en

nuestro escenario de una forma desordenada en cada una de las ejecuciones de dicho ejercicio. En esta capa también se programa cada *MovieClip* para que se pueda seleccionar y arrastrar hacia cualquier parte de nuestro escenario y de esta forma al chocar con ciertos elementos se queden fijados en esa posición.

3. Capa right. Capa en la que tenemos *MovieClip* vacíos que utilizaremos como diana de los antes mencionados, los cuales se pueden seleccionar y arrastrar, así se dé la posibilidad de poder dejar en una ubicación determinada los términos y realiza posteriormente las comprobaciones correspondientes.

4. Capa feedback. Capa en la que según nuestro resultado, nos da un mensaje de correcto o no.

5. Capa texto e imagen. Y distintas capas que utilizaremos para introducir imágenes o texto que queramos tener impresa en nuestro escenario.

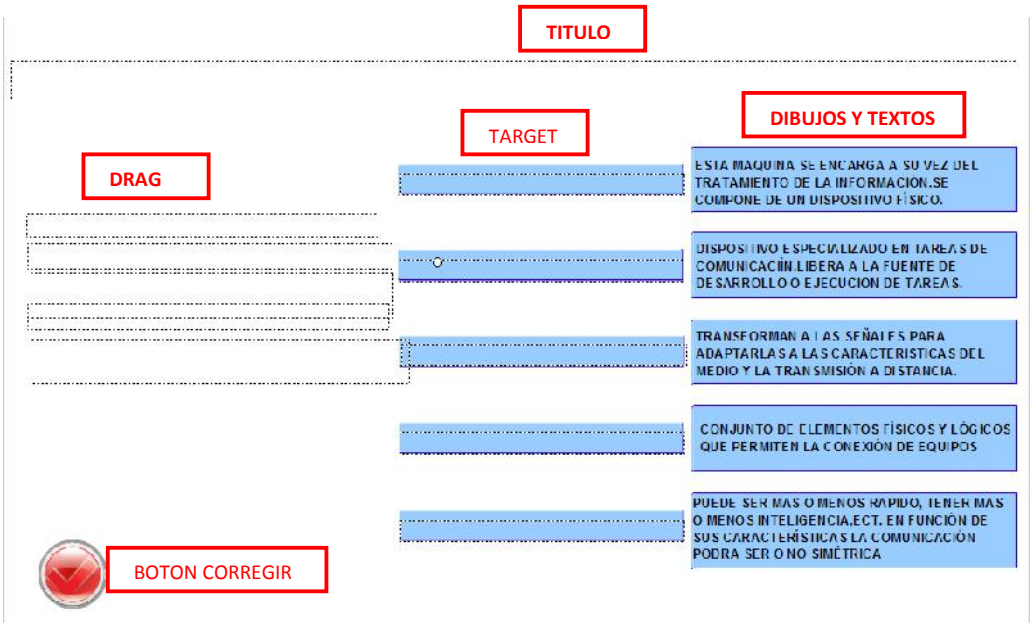


Figura 2.23

Como se puede observar en la *Figura 2.23,* se ha señalado con distintos las diferentes capas, para poder explicar más intensamente su contenido.

Existe una capa que no se puede apreciar en dicha figura, que solo contiene código ActionScript, es la capa script, la cual hemos mencionado anteriormente. De esta capa explicaremos ciertas partes del código, la cual nos hará entender posteriormente lo que realizan ciertos componentes de nuestra animación.

```
function reloader() {
        loadMovie ("TEMA1/ej2Introduccion.swf", 1);
}
```

PARTE 1

```
function start() {
quiz_title = "SELECCIONA Y ARRASTRA EL TÉRMINO A SU DEFINICIÓN CORRESPONDIENTE";
matchspace_x = 0
matchspace_y = 0

chosentarget_1 = 0;
chosentarget_2 = 0;
chosentarget_3 = 0;
chosentarget_4 = 0;
chosentarget_5 = 0;

drag1_choice = "";
drag2_choice = "";
drag3_choice = "";
drag4_choice = "";
drag5_choice = "";
}

question_1 = "";
question_2 = "";
question_3 = "";
question_4 = "";
question_5 = "";

answerArray = new Array ("FUENTE DE DATOS;CONTROLADOR DE COMUNICACIONES;CONVERTIDOR/ADAPTADOR;RED DE TELECOMUNICACION;TERMINAL REMOTO");
```

PARTE 2

```
answerArray[0] = answerArray[0].split(";");
answers = shuffleArray(answerArray[0].slice(0));

choice_1 = answers[0];
choice_2 = answers[1];
choice_3 = answers[2];
choice_4 = answers[3];
choice_5 = answers[4];

function shuffleArray (array1) {
        array2 = new Array();
        do {
                r = int(Math.random()*array1.length);
                array2.push(array1[r]);
                array1.splice(r, 1);
```

```
        } while (array1.length>0);
        return (array2);
        }
```

PARTE 3

```
function mark() {

        numbercorrect = 0


if (drag1_choice == answerArray[0][0])
                {numbercorrect ++}
if (drag2_choice == answerArray[0][1])
                {numbercorrect ++}
if (drag3_choice == answerArray[0][2])
                {numbercorrect ++}
if (drag4_choice == answerArray[0][3])
                {numbercorrect ++}
if (drag5_choice == answerArray[0][4])
                {numbercorrect ++}


feedback = "HAS TENIDO "+ numbercorrect + " CORRECTAS";

if (numbercorrect == 5) {_root.window.gotoAndStop(3);}
else {_root.window.gotoAndStop(2);}

}
stop ();
```

- En la Parte 1 del código, además de inicializar y declarar variables que utilizaremos en nuestro ejercicio hay dos declaraciones importantes a tener en cuenta:
  - El título de nuestro ejercicio, ya que se ha declarado con un texto dinámico y pondrá el valor que se introduzca en la variable *quiz_title.*
  - La declaración del array con los términos que usaremos para la realización del ejercicio.

- En la Parte 2 del código, a través de una función llamada *shuffleArray(),* iremos guardando término a término en variables para luego poder trabajar con ellos.
  - Lo que realiza la función *shuffleArray(),* es coger el array que se le pasa como parámetro y sacar los términos aleatoriamente de dicho array e introducirlos en otro que hemos creado, quedando en este nuevo los términos desordenados .
  - El seleccionar el término aleatorio se realiza gracias a la función *Math.random()* que nos da un número aleatorio entre el 0 y el 1, que multiplicado por la longitud de nuestro array, nos dará como máximo un

valor igual al tamaño de nuestro array, que indicaría en ese caso al último elemento de dicho array.
Posteriormente, se introduce el término que ocupa en dicha posición, en el nuevo array que se creó y se borra el elemento del array original para que no vuelva aparecer el mismo término en otra posición del nuevo array.

- Por último, en la Parte 3 del código, comentar una última función a la cual se hará referencia desde el botón situado en nuestro escenario utilizado para la corrección del ejercicio. La función es *Mark(),* el cual hemos nombrado anteriormente en el apartado 2.4.4 Botón de corregir, que compara si los términos se encuentran colocados en sus posiciones correctas o no. Una vez realizadas estas comprobaciones se crea un mensaje mostrando el número de aciertos que se ha tenido, apareciendo otro mensaje en pantalla si se ha tenido algún error, o por lo contrario, un mensaje de que se ha realizado correctamente, siendo esto la función de las dos últimas líneas. Estos mensajes veremos a continuación que se han introducido en una capa distinta.

La capa que indica si ha sido correcta o no la realización de nuestro ejercicio, consta dentro de ésta, de tres fotogramas como podemos observar en la *Figura 2.24.*

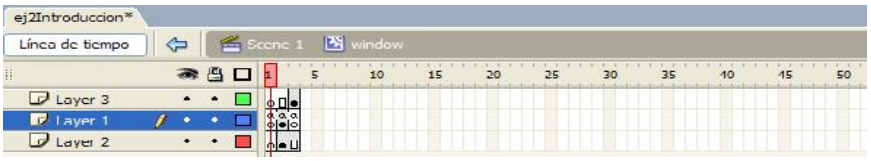


Figura 2.24

El fotograma 1 está vacío y no realiza ninguna función, pero en el fotograma 2 nos da el mensaje de erróneo y nos indica el número de aciertos que se han tenido gracias a que en ese fotograma se ha introducido un cuadro de texto dinámico que está asociado a la función *Mark()*, anteriormente vista. De lo contrario si todo se ha realizado correctamente se muestra un mensaje de correcto. Ambos mensajes cuando aparecen en la pantalla llevan un elemento que le hemos añadido para que parezca que lo que hay por debajo del mensaje esta como en un segundo plano ya que aparece difuminado. Estos mensajes se pueden observar en la *Figura 2.25*

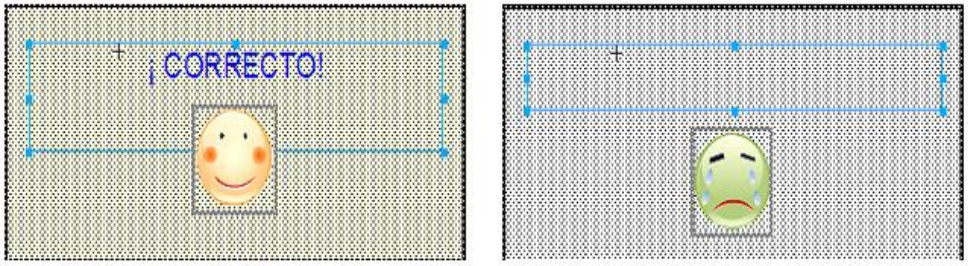


Figura 2.25

Las capas de texto y los recuadros que hemos dibujado solo contienen esos elementos y no contienen código que tengamos que explicar. La única observación que hay que tener en cuenta de estas capas, es que los cuadros de texto utilizados aquí son de texto

estático ya que sobre dicho texto no se va a realizar ninguna modificación a lo largo de la ejecución de nuestra película. Esta es la diferencia entre este tipo de cuadros de texto y el introducido en el mensaje de error comentado anteriormente, de tipo dinámico, pues según lo que ocurría en la animación salía un mensaje diferente, es decir, era un mensaje modificable según lo ocurrido en la ejecución de nuestro ejercicio.

A continuación explicaremos las capas más importantes en este tipo de ejercicios que son las que realizan la acción de coger y arrastrar. Como podemos ver en la *Figura 2.23,* estas capas son las que hemos nombrado como *Target y Drag.*

En la capa de los target, nombrada capa right, hemos creado unos *MovieClip* que se usaran como diana u objetivo de los términos que hay que coger y arrastrar. Estos *MovieClip* contienen unos cuadros de texto dinámicos que recogerán los términos una vez que se suelten encima de ellos.

En la capa de Drag se situarán los términos de manera desordenada, de forma que la persona que realice el ejercicio coja y arrastre los términos a su definición correcta. En esta capa hemos creado un *Movieclip* para cada uno de los términos que deseamos que aparezca en nuestro escenario. Cada uno de ellos se encuentra programado para poder realizar la acción de coger y arrastrar.

Cada *MovieClip* tiene en código ActionScript, recogida las coordenadas de la posición inicial de dicho objeto, como se observa en el código de uno de los *MovieClip*

```
onClipEvent (load) {
        ../:initial_drag1X = _x;
        ../:initial_drag1Y = _y;
        }
```

Dentro de estos *MoviClip*, se ha introducido un botón mediante el cual nuestro MovieClip podrá ser arrastrado gracias a la programación que tiene asignado dicho botón.

El código es el siguiente:

PARTE 1

```
on (press) {
        startDrag ("../drag_1");
        drag1X = _x;
        drag1Y = _y;
}
on (release) {
        stopDrag ();
```

PARTE 2

```
if ((_root.drag_1, hitTest(_root.target_1)) && (_root.target_1._currentframe == 1))
        {
                setProperty ("../drag_1", _x, getProperty("../target_1", _x));
                setProperty ("../drag_1", _y, getProperty("../target_1", _y));
        _x = (_x+_root.matchspace_x);
        _y = (_y+_root.matchspace_y);

gotoAndStop (2);
_root.target_1.gotoAndStop (2);
_root.chosentarget_1 = 1
_root.drag1_choice = _root.choice_1;


} else if ((_root.drag_1, hitTest(_root.target_2)) && (_root.target_2._currentframe ==
1))      {
                setProperty ("../drag_1", _x, getProperty("../target_2", _x));
                setProperty ("../drag_1", _y, getProperty("../target_2", _y));
                        _x = (_x+_root.matchspace_x);
        _y = (_y+_root.matchspace_y);

gotoAndStop (2);
        _root.target_2.gotoAndStop (2);
        _root.chosentarget_1 = 2
        _root.drag2_choice = _root.choice_1;

} else if ((_root.drag_1, hitTest(_root.target_3)) && (_root.target_3._currentframe ==
1)) {
                setProperty ("../drag_1", _x, getProperty("../target_3", _x));
                setProperty ("../drag_1", _y, getProperty("../target_3", _y));
                        _x = (_x+_root.matchspace_x);
        _y = (_y+_root.matchspace_y);
gotoAndStop (2);
        _root.target_3.gotoAndStop (2);
        _root.chosentarget_1 = 3
        _root.drag3_choice = _root.choice_1;
                //
} else if ((_root.drag_1, hitTest(_root.target_4)) && (_root.target_4._currentframe ==
1))      {
                setProperty ("../drag_1", _x, getProperty("../target_4", _x));
                setProperty ("../drag_1", _y, getProperty("../target_4", _y));
                        _x = (_x+_root.matchspace_x);
        _y = (_y+_root.matchspace_y);
```

```
gotoAndStop (2);
        _root.target_4.gotoAndStop (2);
        _root.chosentarget_1 = 4
        _root.drag4_choice = _root.choice_1;
                //
} else if ((_root.drag_1, hitTest(_root.target_5)) && (_root.target_5._currentframe ==
1))      {
                setProperty ("../drag_1", _x, getProperty("../target_5", _x));
                setProperty ("../drag_1", _y, getProperty("../target_5", _y));
                        _x = (_x+_root.matchspace_x);
        _y = (_y+_root.matchspace_y);
                gotoAndStop (2);
        _root.target_5.gotoAndStop (2);
        _root.chosentarget_1 = 5
        _root.drag5_choice = _root.choice_1;

}else {
                setProperty ("../drag_1", _x, drag1X);
                setProperty ("../drag_1", _y, drag1Y);
        }
}
```

- PARTE 1. Para coger y arrastrar se utiliza la función *startDrag()* para comenzar a arrastrar cuando se presiona sobre el MoviClip que contiene el botón, y la *función stopDrag ()* parar dejar de arrastrarlo una vez se cese de pinchar sobre dicho *MovieClip*.

- PARTE 2. Lo que realiza esta parte del código es comprobar si el Drag ha colisionado con el Target que se indica y si es así, quedará colocado el Drag sobre el Target y si no es así vuelve a su posición inicial.

  Si el Drag ya está colocado sobre un Target, al volver a pulsar sobre él, éste volverá a su posición inicial.
  En esta parte del código las comprobaciones se han realizado sólo para el caso de colisión con el Target_1, pero en realidad cada Drag tiene la función de comprobar cada uno de los Target de nuestro ejercicio.

Por último explicar que en la animación existen 2 botones. Un botón lo utilizaremos para corregir el ejercicio que hará referencia a la función *Mark()* de la capa donde solo lleva código ActionScript. Y un último botón *Salir*, cuya función es volver al menú del que procedía y cierra la película del ejercicio Drag and Drop. Esto botones los podemos observar en la *Figura 2.26*

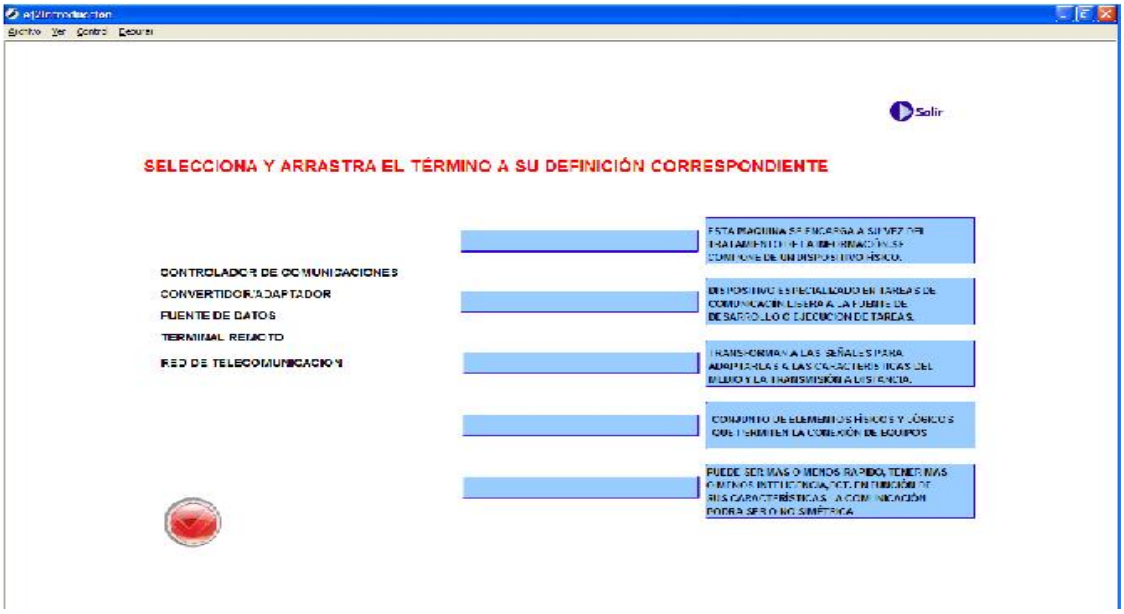


Figura 2.26

2.6 CREAR UN TEST

Para la realización de un test en flash se ha introducido en cada fotograma una pregunta y sus respuestas correspondientes. Dichas respuestas llevan un botón asociado para seleccionar la verdadera, la cual quedará marcada cuando se presione sobre ella. Al final de las preguntas tipo test nos mostrará el resultado de nuestro ejercicio.

Para explicar detenidamente este tipo de ejercicio iremos comentando cada una de sus partes y los códigos asociados a cada una de ellas, si lo contienen.

En la *Figura 2.27*, podemos observar el aspecto que va a tener cada uno de las preguntas que contiene nuestro test.

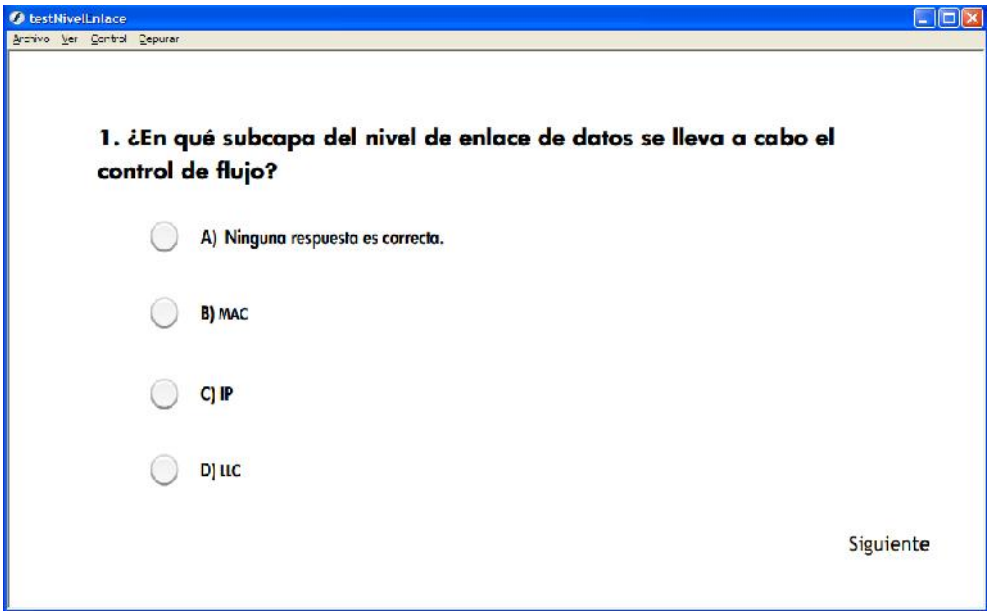


Figura 2.27

En cada fotograma en las que existan preguntas tipo test, existirá una capa en la que están escritas las preguntas con un cuadro de texto estático, y otra capa para las respuestas, escritas de igual forma en cuadros de texto estático. En la capa de las respuestas, al lado de cada solución existe un componente creado de forma que al pinchar sobre él se quede marcado y al pinchar sobre otro posteriormente, se quede marcado el segundo y se desmarque el primero. Estos componentes se configuran en parámetros como se muestra en la *Figura 2.28*

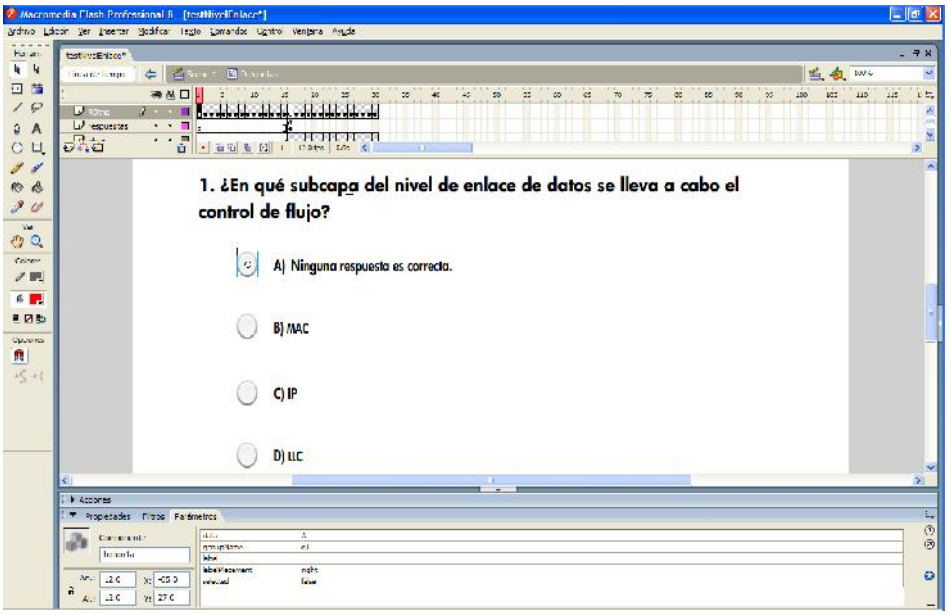


Figura 2.28

En esos parámetros se asigna a *DATA,* el valor que se quiera, en este caso al ser el componente de la respuesta A, le asignamos ese dato especifico. También existe un parámetro que lo llama nombre de grupo. Ese valor se utiliza para meter en un mismo grupo varios componentes iguales pero con distinto nombre y así poder seleccionar unos u otros. Otro parámetro es la posición de etiqueta que lo ponemos en correcto. EL parámetro de selección se pone en false, para que no se seleccione al ejecutarse la película.

Existe una capa, capa receptor, así la hemos llamado, en la que se escribe el código necesario para guardar en una variable el valor del componente que se pulsa. El código es el siguiente:

```
g1Btns = new Object();
g1Btns.click = function(evento) {
_root.respuesta1=evento.target.selection.data;
};
```

Lo único que hace es crear un objeto para el grupo al que se refiere y al hacer clic sobre uno de los componentes del grupo guarde en la variable respuesta, el valor Data de dicho componte.

El que un componente cambie a estar marcado se hace en el código que hay en otra capa, el cual compara la variable respuesta con A, B, C y D, y pone al componente que coincida con dicho valor su campo *selected* a true, como se puede observar en el siguiente código escrito en la capa que hemos llamado resp.sel.

```
if(_root.respuesta1=="A"){
                boton1a.selected= true;}
if(_root.respuesta1=="B"){
                boton1b.selected= true;}
if(_root.respuesta1=="C"){
                boton1c.selected= true;}
if(_root.respuesta1=="D"){
                boton1d.selected= true;}
```

En estos mismos fotogramas tenemos dos botones *Siguiente* y *Anterior,* como vemos en la *Figura 2.29,* que lo único que tienen programado es pasar al siguiente fotograma o volver al anterior. Eso tanto uno como otro, lo realizan a través de la *función gotoAndStop()* como muestra las líneas de ActionScript de uno de ellos en el siguiente código.

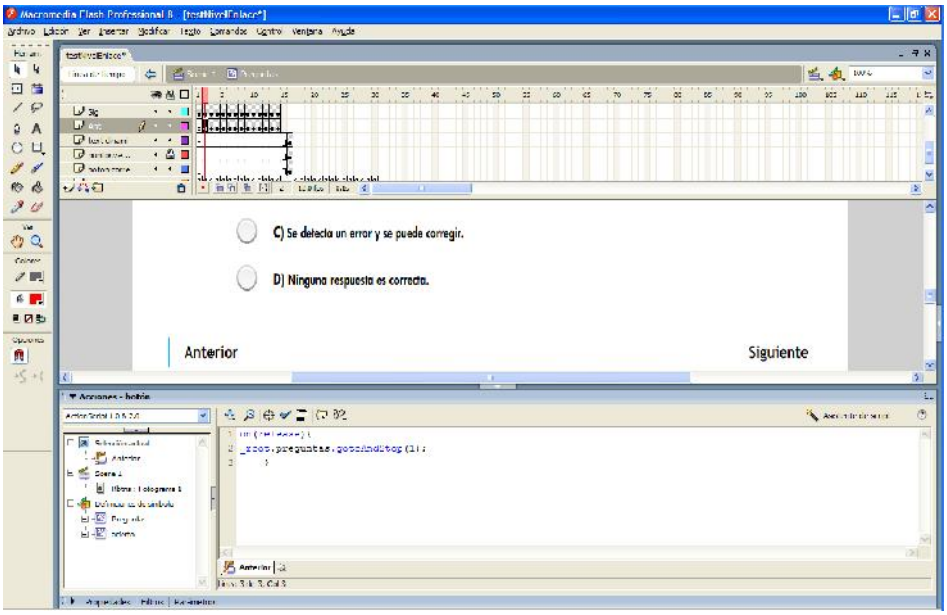


Figura 2.29

```
on(release){
_root.preguntas.gotoAndStop(1);
        }
```

Al terminar de responder todas las preguntas, aparece una pregunta pidiéndote si quieres finalizar el tipo test, como se observa en la *Figura 2.30*. Si contestas que *NO* te manda a la primera pregunta del test para que continúes con él y si contestas que *SI* te manda a un fotograma en el que te sale un resumen de las contestaciones verdaderas y las que tú has contestado. En este resumen debajo de las respuestas que tú has ido seleccionando en cada pregunta aparecerá un círculo verde si es acertada o rojo en caso contrario. Y también aportará un porcentaje de aciertos.

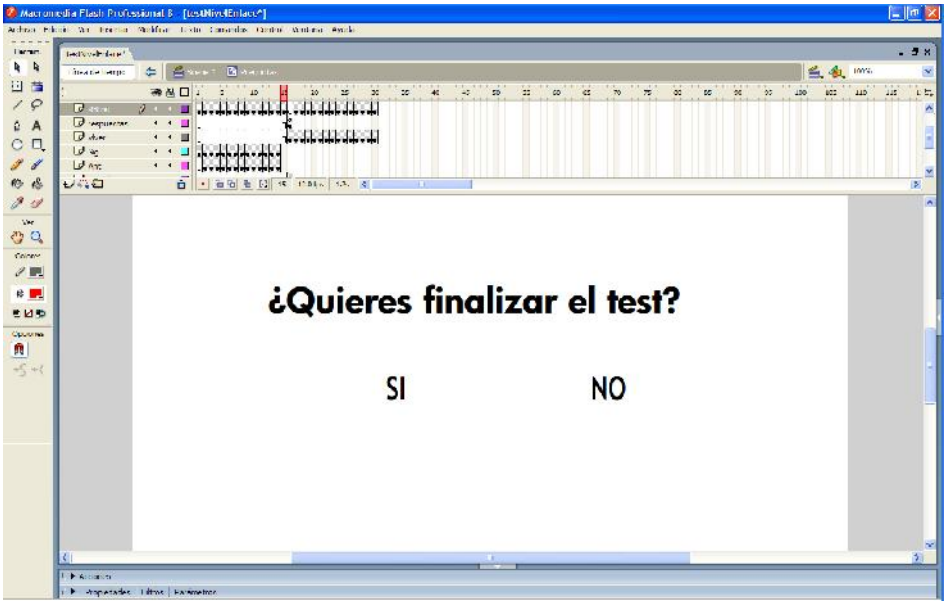


Figura 2.30

Este resumen, del tipo test lo podemos observar en la *Figura 2.31*, y a raíz de esta iremos explicando los objetos que lo componen y por supuesto el código que alguno de esos objetos llevan asociado.

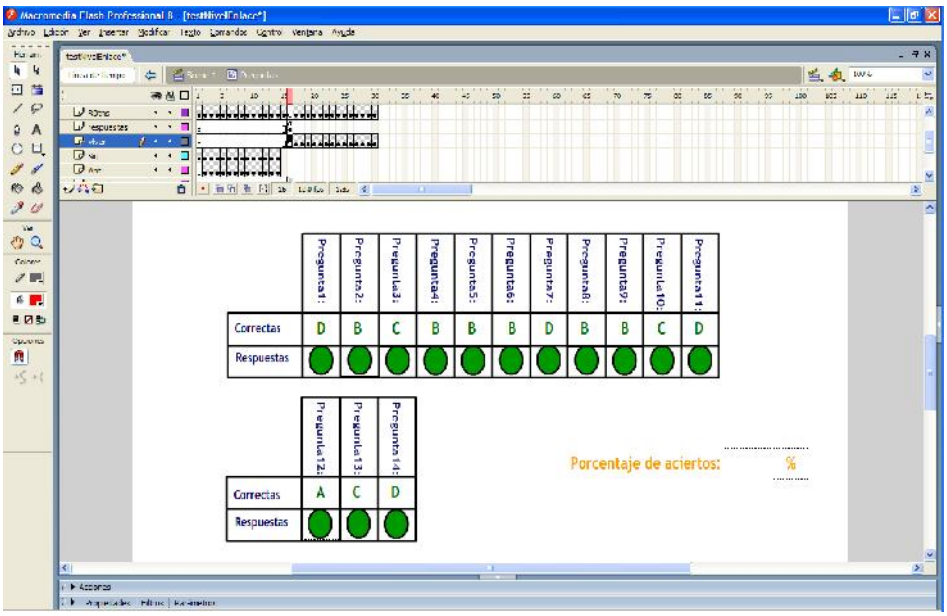


Figura 2.31

En la *Figura 2.31*, podemos observar lo anteriormente dicho, que dependiendo de si la respuesta es correcta o no, aparece un círculo en verde o rojo. Esa capa en la que se encuentran– estos símbolos simplemente consta de dos fotogramas en los que están dichos dibujos, y dependiendo del resultado, se referencia a un fotograma u otro.

Un detalle que tenemos que tener en cuenta, es que si en la selección de respuestas de nuestro tipo test, existe alguna pregunta sin seleccionar ninguna opción, en el resumen aparece también con un círculo rojo y un guión que nos indica que no se ha seleccionado ninguna respuesta, y de lo contrario si se ha seleccionado una de las respuestas se le asocia al cuadro de texto dinámico que hay dentro de los recuadros para nuestras respuestas, la del contenido de la variable que obtuvo su valor. Estas comprobaciones las podemos observar en el siguiente código, escrito en la capa respuestas como vemos en la *Figura 2.32*

```
if(_root.respuesta1==null){res1.text="-";}
else{
        res1.text=_root.respuesta1;
}
if(_root.respuesta2==null){res2.text="-";}
else{
        res2.text=_root.respuesta2;
}
if(_root.respuesta3==null){res3.text="-";}
```

```
else{
        res3.text=_root.respuesta3;
}
```

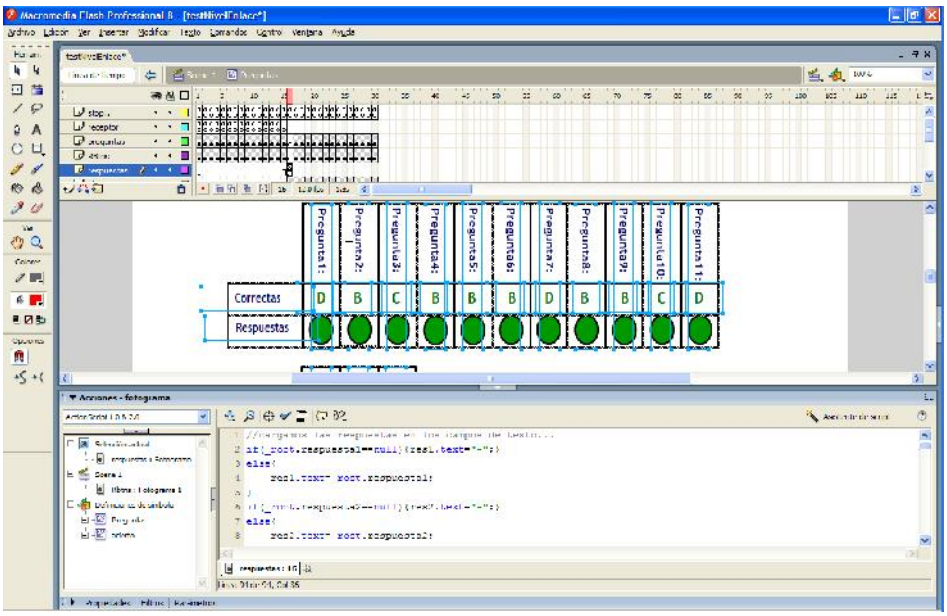


Figura 2.32

El código que realiza la aparición del círculo verde o el rojo, y como se calcula el porcentaje de aciertos se muestra en el siguiente código escrito en la capa respuestas

```
_root.total=14;
if(_root.respuesta1 ¡= “D”){aci1.gotoAndStop(2);
                _root.total=_root.total-1;}
if(_root.respuesta2 ¡= “B”){aci2.gotoAndStop(2);
                _root.total=_root.total-1;}
if(_root.respuesta3 ¡= “C”){aci3.gotoAndStop(2);
                _root.total=_root.total-1;}
//Aquí se hacen las comparaciones para el resto de preguntas

//de la misma forma que las 3  primeras

restotal.text=(_root.total/14)*100;
```

De este fotograma nos queda explicar, que sobre nuestras respuestas elegidas, hay situado un botón sin imagen, cuya función es que al presionar encima, nuestro test nos envíe a un nuevo fotograma en el que se encuentra la pregunta que queremos comprobar, la cual era incorrecta, estando señalada nuestra respuesta y en el texto color verde la respuesta correcta. La *Figura 2.33* nos muestra un ejemplo de dicho fotograma.

En este fotograma existe un botón *Volver* que nos devuelve al resumen de respuestas, por si queremos seguir mirando otras preguntas.

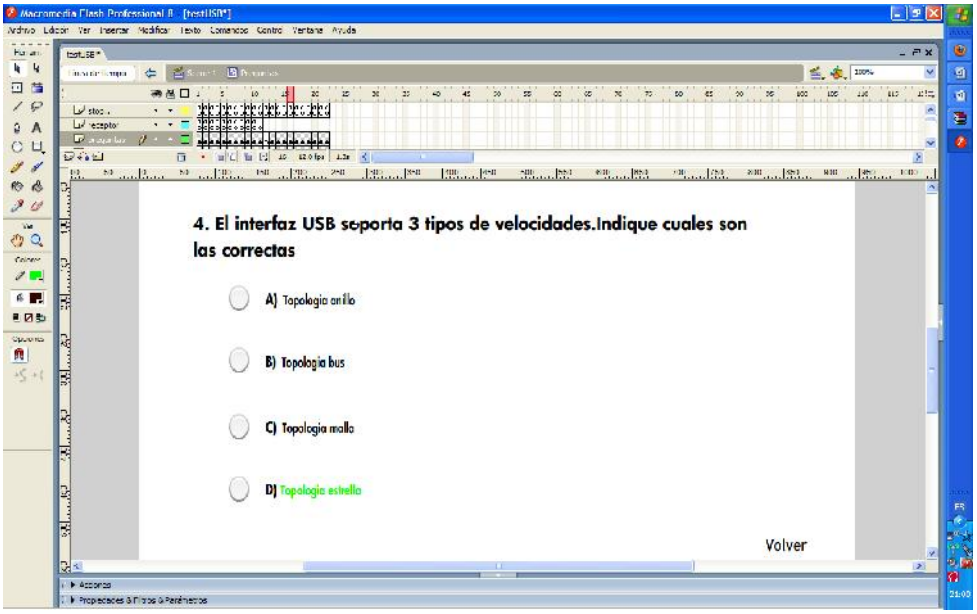


Figura 2.33

Para concluir, indicar que también existe en el resumen de respuesta un botón llamado *Finalizar*, se muestra en la *Figura 2.34*, que manda a la película principal a la parte del menú de donde salto a dicho test, y cierra el archivo .swf del test que se está ejecutando. Esto lo podemos observar siguiente código.

```
on(release){
        _level0.gotoAndStop(20);
        this.unloadMovie();
}
```

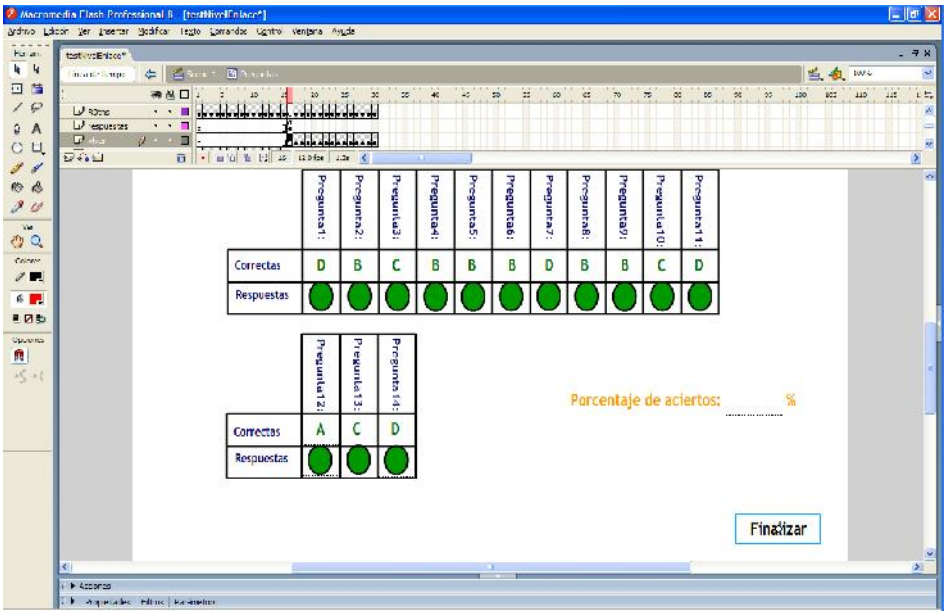

Pregunta1:
Pregunta2:
Pregunta3:
Pregunta4:
Pregunta5:
Pregunta6:
Pregunta7:
Pregunta8:
Pregunta9:
Pregunta10:
Pregunta11:
Correctas
D B C B B B D B B C D
Respuestas
Pregunta12:
Pregunta13:
Pregunta14:
Correctas
A C D
Respuestas
Porcentaje de aciertos: %
Finalizar

Figura 2.34

CAPITULO 3. ANIMACIONES FLASH PARA LA ASIGNATURA TELEMATICA

3.1. INTRODUCCION

En este apartado mostraremos las partes utilizadas para construir un CD-Rom interactivo, el cual se utilizará en la asignatura Telemática.
A continuación se muestra una descripción del esquema que hemos seguido.

3.2 PORTADA Y MENU INICIAL

Ejecutamos el CD-Rom, y la primera ventana que nos encontramos es la siguiente *Figura 3.1*:

Figura 3.1

Pulsando el botón *ENTRAR* aparecerá el menú principal, *Figura 3.2*, el cual consta de 10 botones, estos botones representan cada uno de los temas de los que consta la asignatura de Telemática.

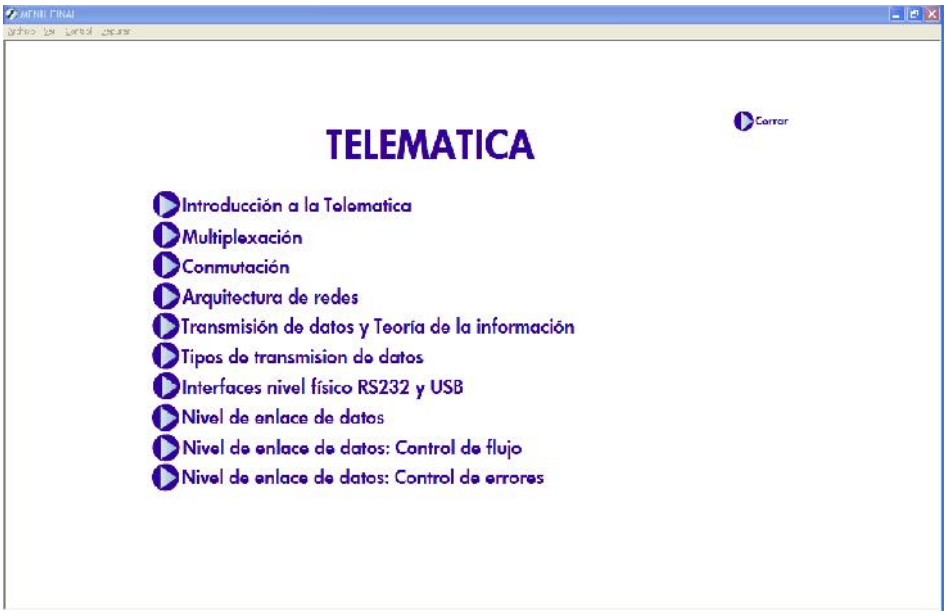


Figura 3.2

El botón de *Cerrar*, facilitará al usuario, concluir la ventana de ejecución cuando lo desee.

3.3 INTRODUCCION A LA TELEMATICA

Con sólo presionar el botón de dicho tema, se accede a éste, *Figura 3.3*. En el menú representado aparece el nombre de dicho tema en letra de mayor tamaño y un submenú, cuyo contenido consta solo de una parte: ejercicios.

Figura 3.3

Existe un pequeño botón, en la esquina superior derecha, llamado *Inicio*, el cual ayuda al usuario a volver al menú inicial y así poder elegir otro tema o salir de la ejecución.

3.3.1 Botón de Ejercicios

El botón de ejercicios, abre una escena, en la cual visualizamos una serie de gráficos con sus títulos correspondientes, cuyo conjunto se ha programado con botones, los cuales dan acceso a los diferentes ejercicios de este tema (*Figura 3.4*).

Figura 3.4

En la *Figura 3.4*, se puede observar dos pequeños botones en la esquina superior derecha, llamados *Inicio e Introducción a la telemática*. Pulsando *Inicio* se vuelve al menú inicial y el otro botón vuelve al menú de dicho tema. Estos botones se incluirán en todas los menús de los temas con sus nomenclaturas correspondientes.

A continuación visualizaremos las animaciones, a las que se han hecho referencia en la *Figura 3.4*.

- DEFINICIONES SOBRE SISTEMAS DE TELECOMUNICACIONES

Este ejercicio, *Figura 3.5*, es un ejercicio Drag&Drop (explicado en el capítulo 2) exponiendo los términos de un sistema de telecomunicaciones a un lado y sus definiciones a otro lado.

Figuro 3.5

- DEFINICIONES DE UN SISTEMA TELEMATICO

Este ejercicio, *Figura 3.6*, es un ejercicio Drag&Drop donde tenemos a un lado los términos de las partes de un sistema telemático y al otro lado sus definiciones.

Figura 3.6

- ESQUEMA BASICO SOBRE UN SISTEMA TELEMATICO

Este ejercicio, *Figura 3.7*, es un ejercicio Drag&Drop donde tenemos el esquema de un sistema telemático y por otro lado los términos de las parte que lo forman, se deben ir colocando cada termino en el sitio donde corresponda.

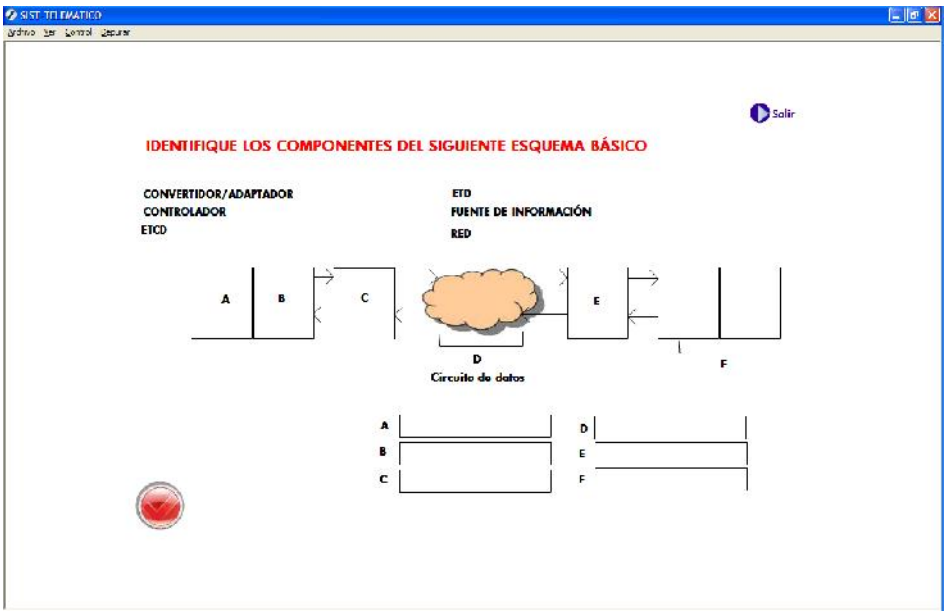


Figura 3.7

3.3.2 Botón de Ejercicios Propuestos

La finalidad de este botón, es abrir una página de navegador, *Figura 3.8*, en la cual aparezca el correspondiente *pdf* del tema, donde se han recopilado preguntas tipo test, cuestiones y problemas de todos los exámenes anuales del tema concreto.

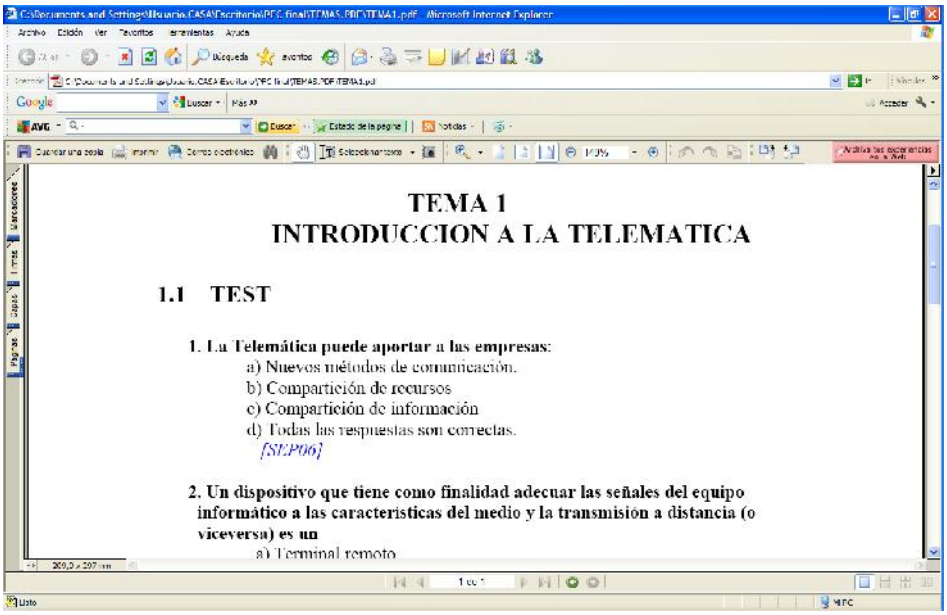


Figura 3.8

3.4 MULTIPLEXACION

Cuando presionamos este botón saltamos a la pantalla que observamos en la *Figura 3.9*, en la cual vemos los apartados que forman este tema.

Figura 3.9

3.4.1 Botón de Animaciones

Presentaremos las distintas animaciones, que aparecen en la escena, *Figura 3.10*, al hacer clic en dicho botón.

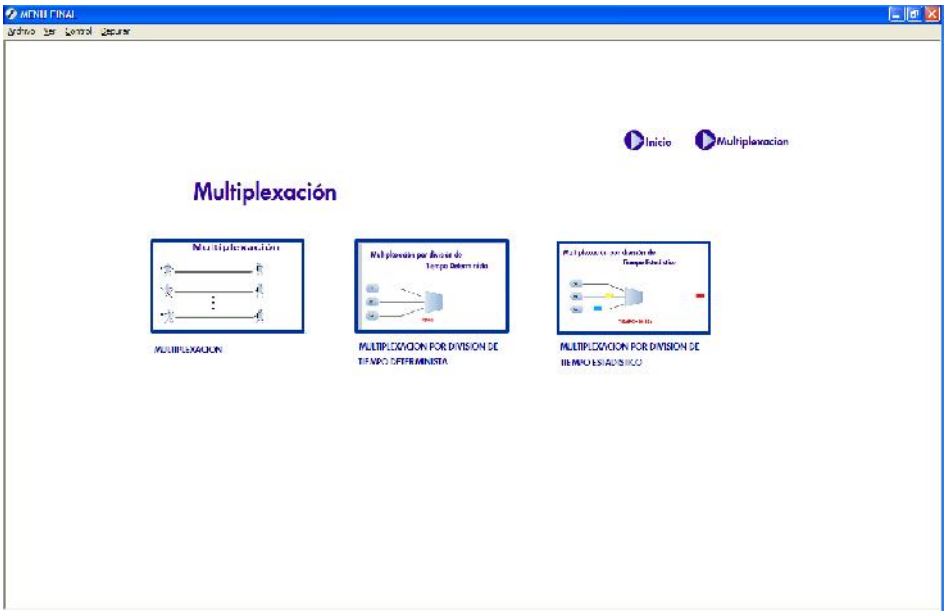


Figura 3.10

- MULTIPLEXACION

En esta animación, *Figura 3.11*, vemos como varias personas se quieren comunicar a la vez y para ello se utiliza un mismo enlace con varios canales.

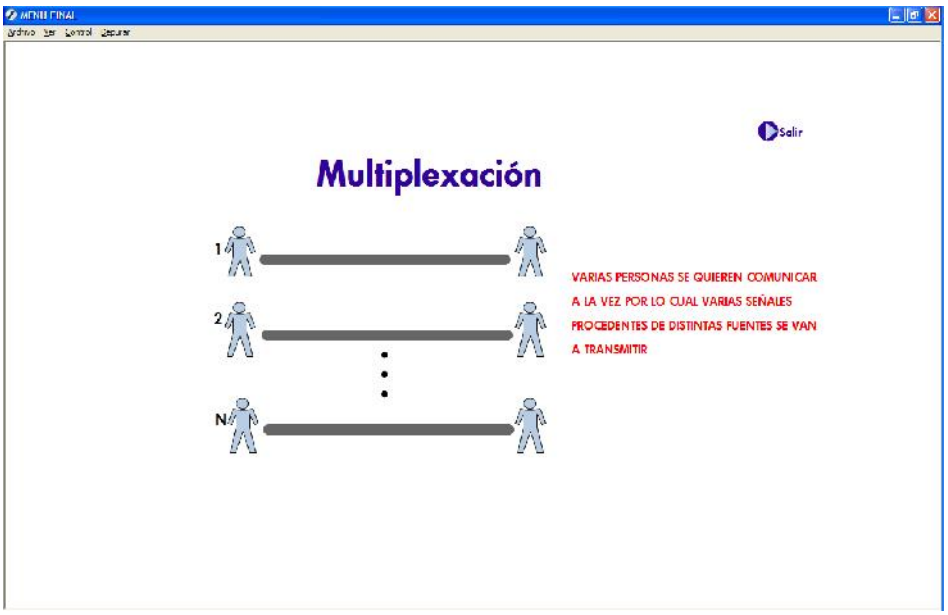


Figura 3.11

- MULTIPLEXACION POR DIVISION DE TIEMPO DETERMINISTA

En esta animación vemos cómo funciona este tipo de multiplexación, *Figura 3.12*

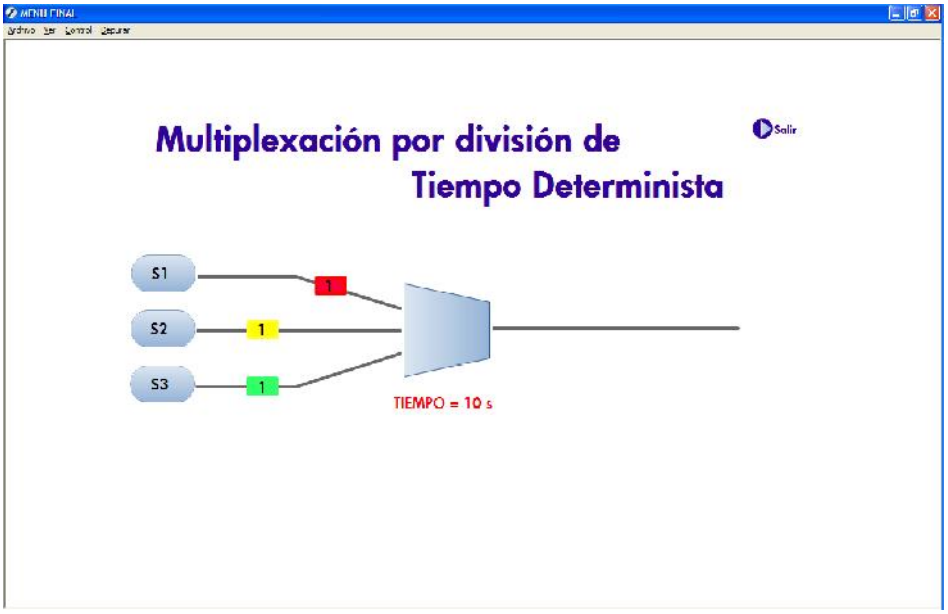


Figura 3.12

- MULTIPLEXACION POR DIVISION DE TIEMPO ESTADISTICO

En esta animación vemos el funcionamiento de la multiplexación por división de tiempo estadístico, *Figura 3.13*

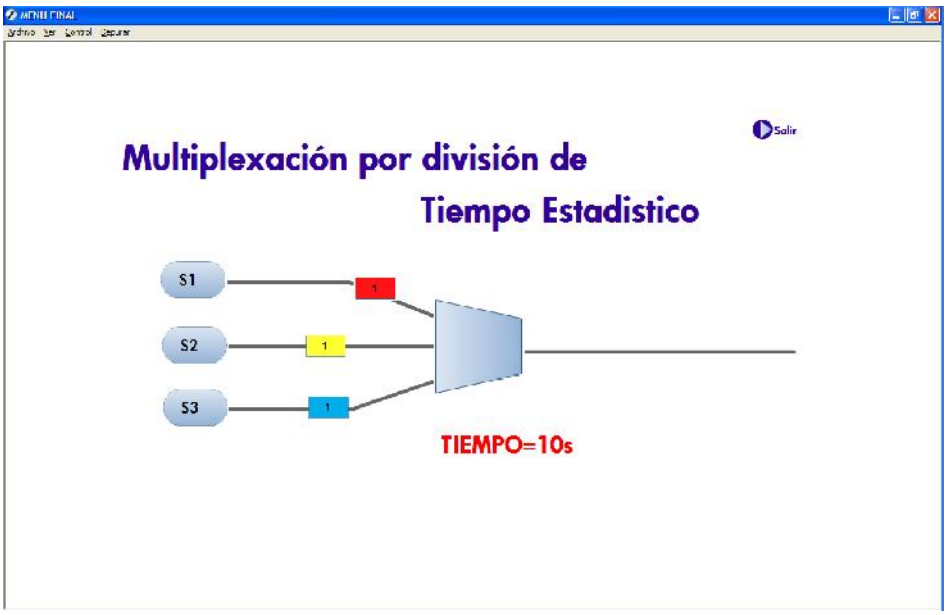


Figura 3.13

3.4.2 Botón de Ejercicios Propuestos

Similar al *Apartado 3.3.2* el cual hace referencia a la función del botón de ejercicios propuestos.

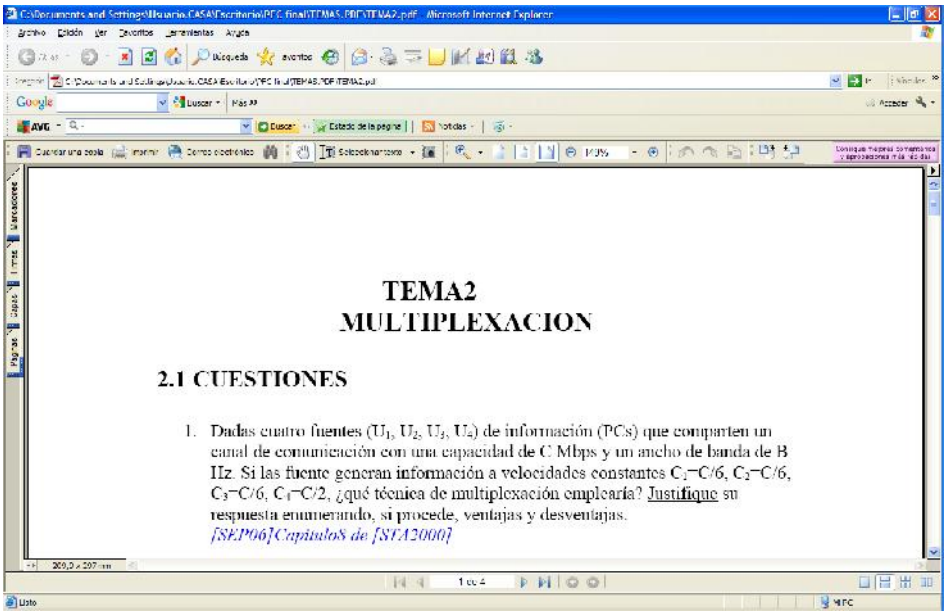


Figura3.14

3.5 CONMUTACION

Conmutación es otro de los temas de la asignatura de Telemática, al cual se puede acceder desde el menú principal.

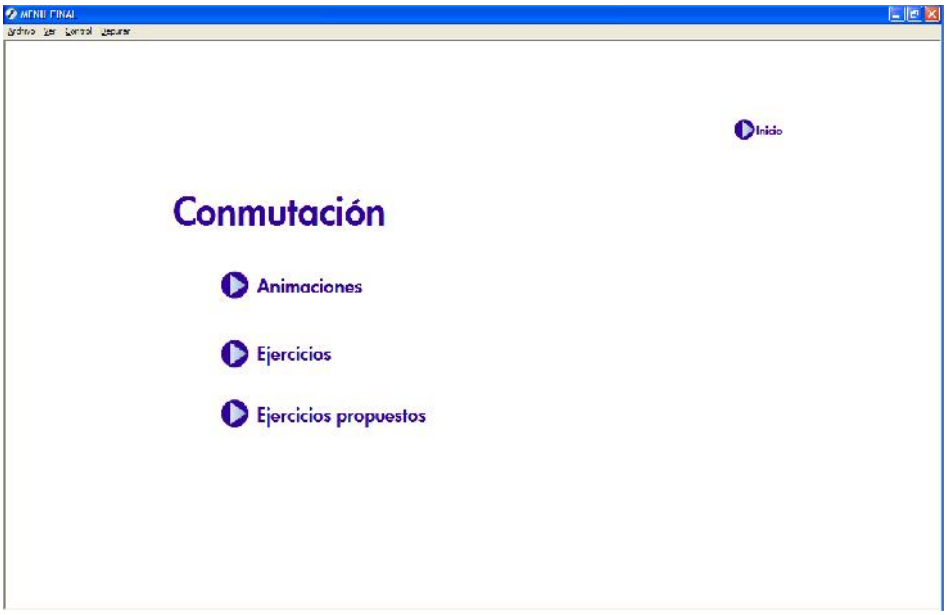


Figura 3.15

3.5.1 Botón Animaciones

Presentaremos las distintas animaciones, que aparecen en la escena, al hacer clic en dicho botón.

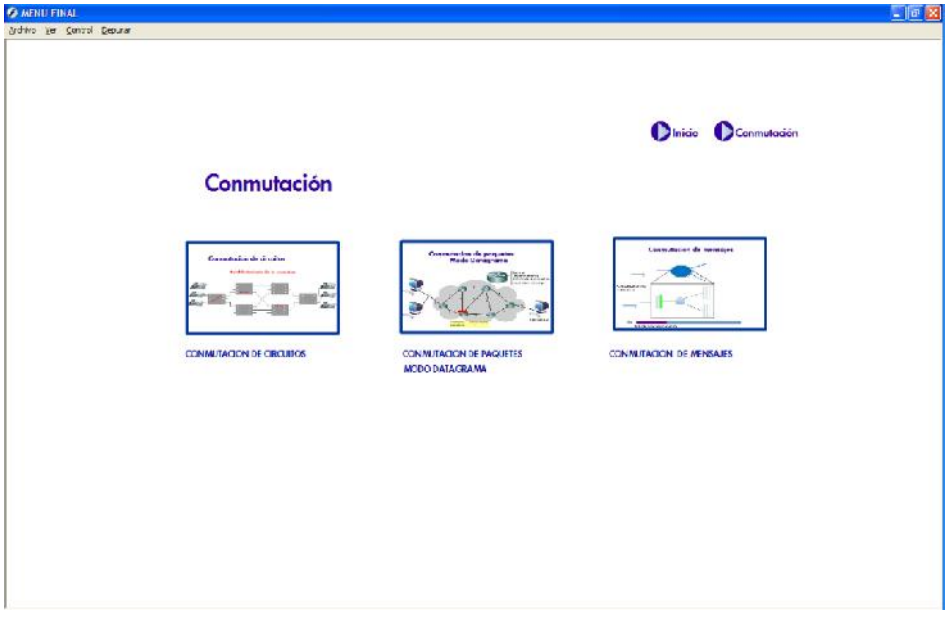


Figura 3.16

- CONMUTACION DE CIRCUITOS

En esta animación observamos el establecimiento del circuito, la transferencia de información y la desconexión.

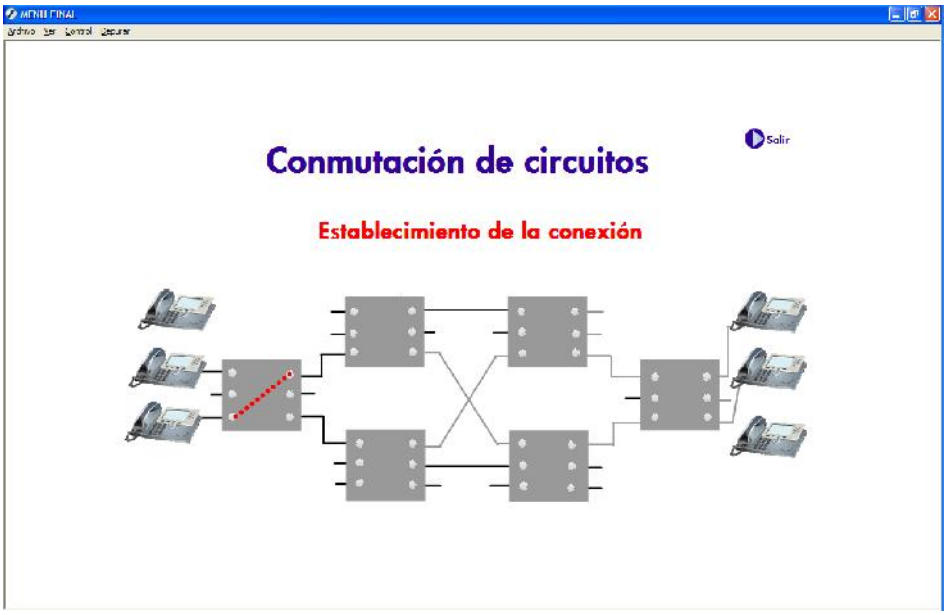


Figura 3.16

- CONMUTACION DE PAQUETES MODO DATAGRAMA

Representa el funcionamiento de una red de conmutación con una función del mecanismo de encaminamiento elegido para los paquetes en modo datagrama.

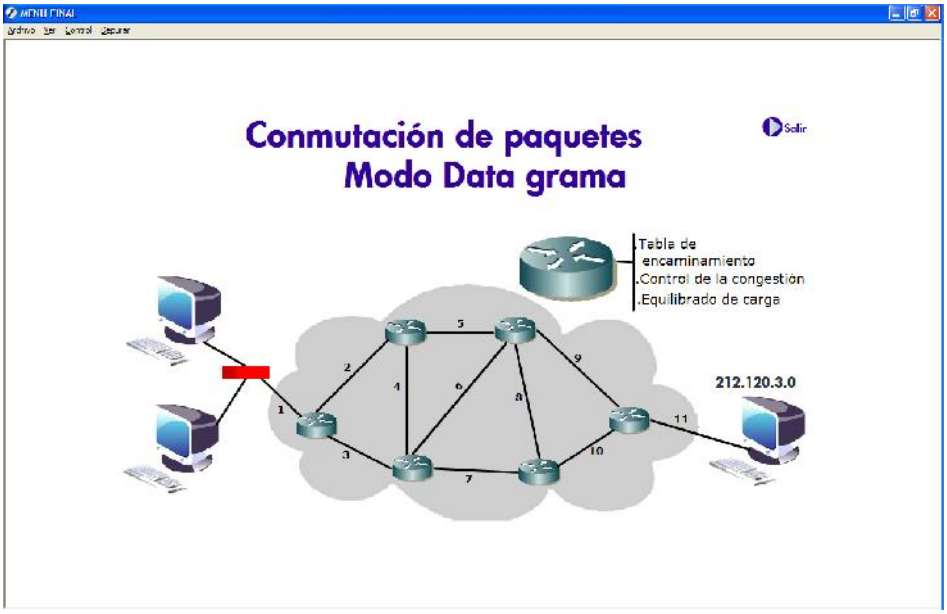


Figura 3.17

- CONMUTACION DE MENSAJES

Representa el funcionamiento de una red de conmutación de mensajes.

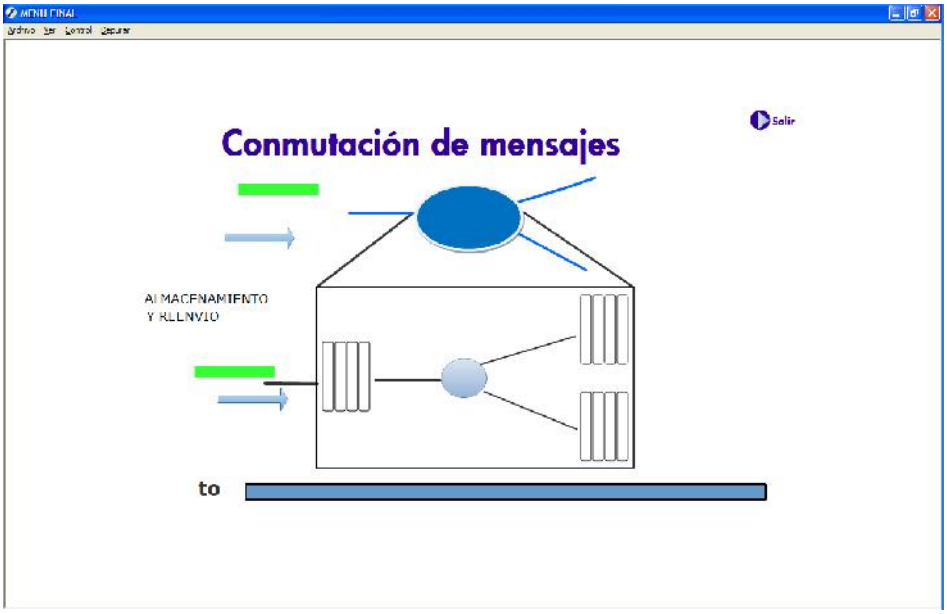


Figura 3.18

3.5.2 Botón Ejercicios.

Obtenemos la siguiente escena pulsando el botón de Ejercicios.

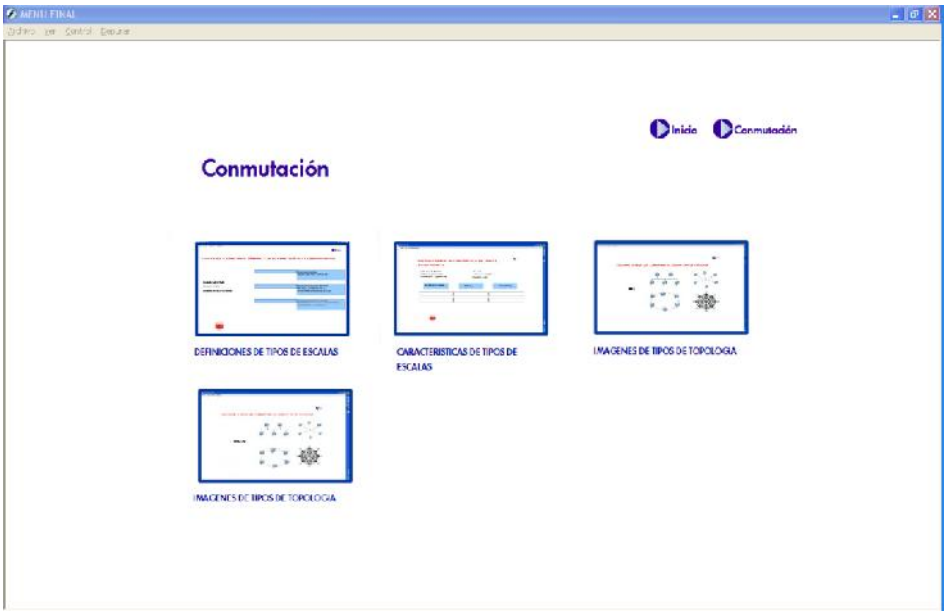


Figura 3.19

- DEFINICIONES DE TIPOS DE ESCALA

Este ejercicio será realizado en forma de Drag&Drop (explicado en el capítulo 2) exponiendo los tipos de escala a un lado y las definiciones de dichos términos en otro.

Figura 3.20

- CARACTERISTICAS DE TIPOS DE ESCALA

De igual forma que en el apartado anterior, se exponen las características de los distintos tipos de escala arriba y los tipos de escala abajo para ir arrastrando cada característica a su tipo de escala correspondiente.

Figura 3.2

- IMÁGENES DE TIPOS DE TOPOLOGIA

A continuación tenemos dos ejercicios similares, en los cuales tenemos imágenes de los tipos de topología y tenemos que pulsar la que corresponde con el tipo de topología indicado a la izquierda.

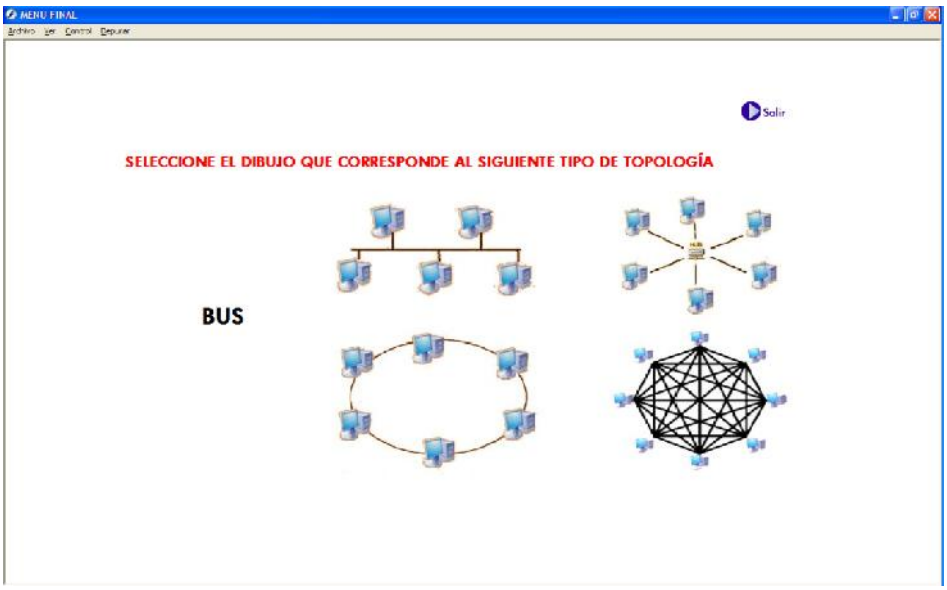


Figura 3.22

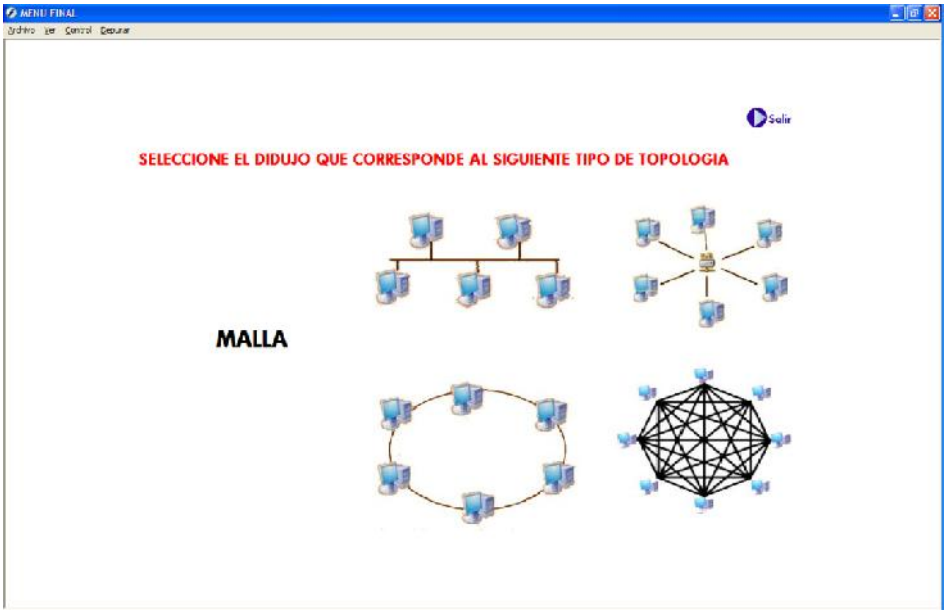


Figura 3.23

3.5.3 Botón Ejercicios Propuestos

Similar al *Apartado 3.3.2* el cual hace referencia a la función del botón de ejercicios propuestos.

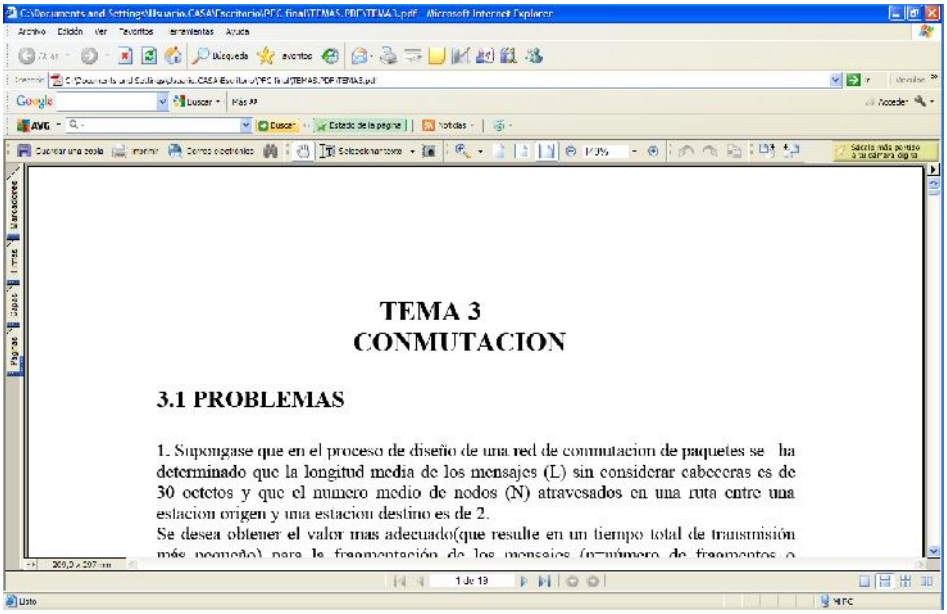


Figura 3.24

3.6 ARQUITECTURA DE REDES

Figura 3.25

3.6.1 Botón animaciones

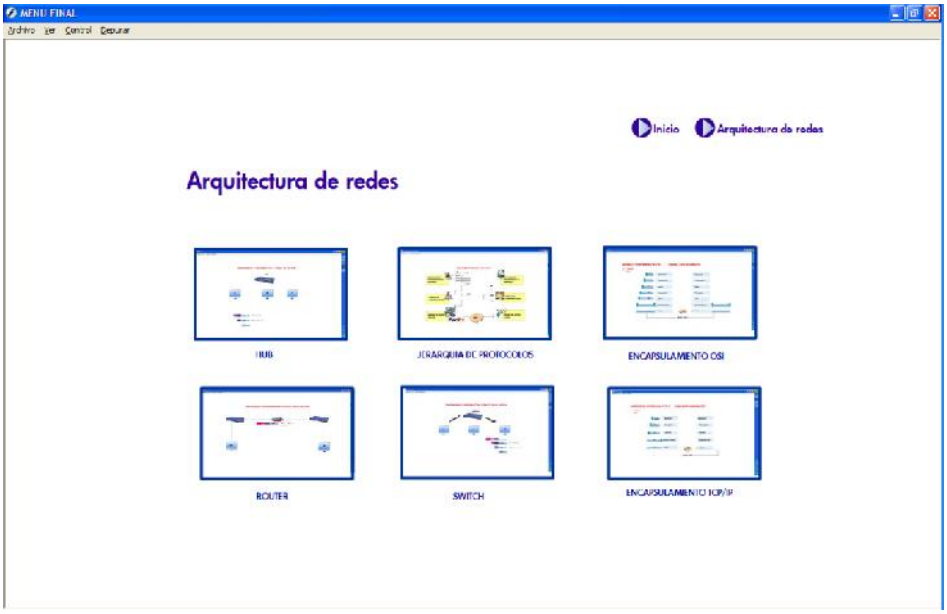


Figura 3.26

- HUB

En esta animación observamos el funcionamiento de un Hub en la transmisión de información.

Figura 3.27

- JERARQUIA DE PROTOCOLOS

En esta animación observamos como diferentes entidades pertenecientes a sistemas distintos se pueden comunicar.

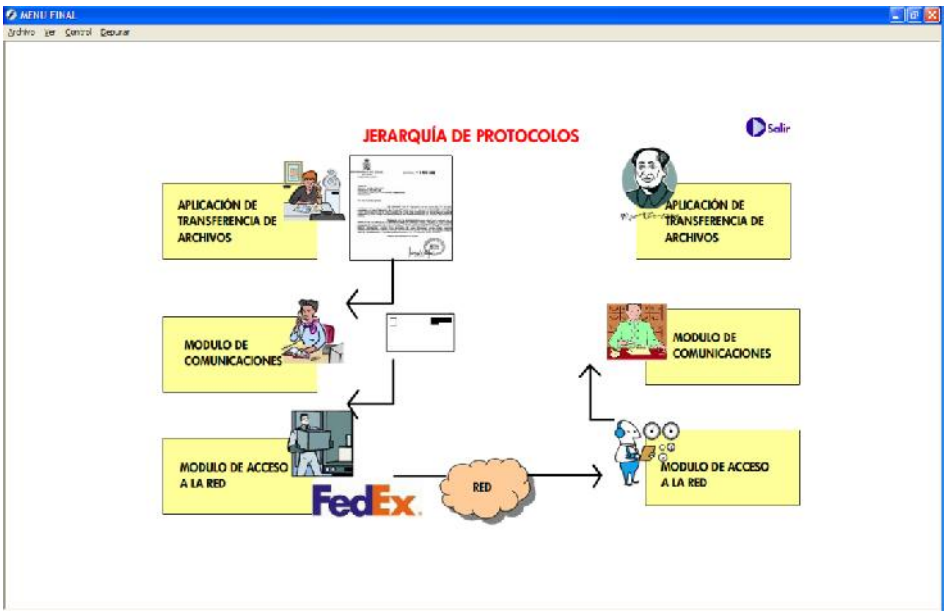


Figura 3.28

- ENCAPSULAMIENTO OSI

En esta animación observamos el encapsulamiento, transferencia y des-encapsulamiento de datos en un sistema OSI.

Figura 3.29

- ROUTER

En esta animación observamos el funcionamiento de un Router en la transmisión de información.

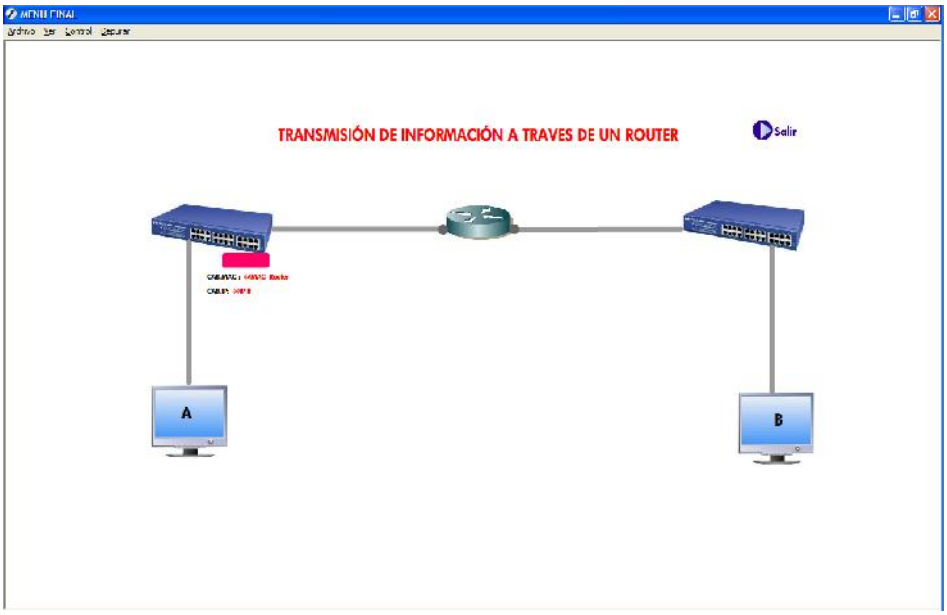


Figura 3.30

- SWITCH

En esta animación observamos el funcionamiento de un switch en la transmisión de información.

Figura 3.31

- ENCAPSULAMIENTO TCP/IP

En esta animación observamos el encapsulamiento, transferencia y des-encapsulamiento de datos en un sistema TCP/IP.

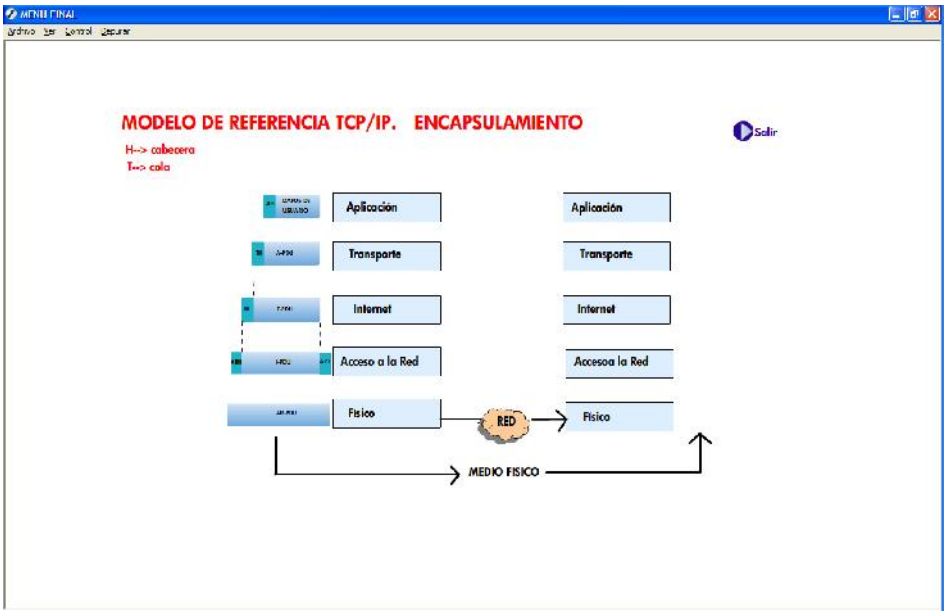


Figura 3.32

3.6.2 Botón Ejercicios

Pulsando a este botón obtenemos la siguiente escena.

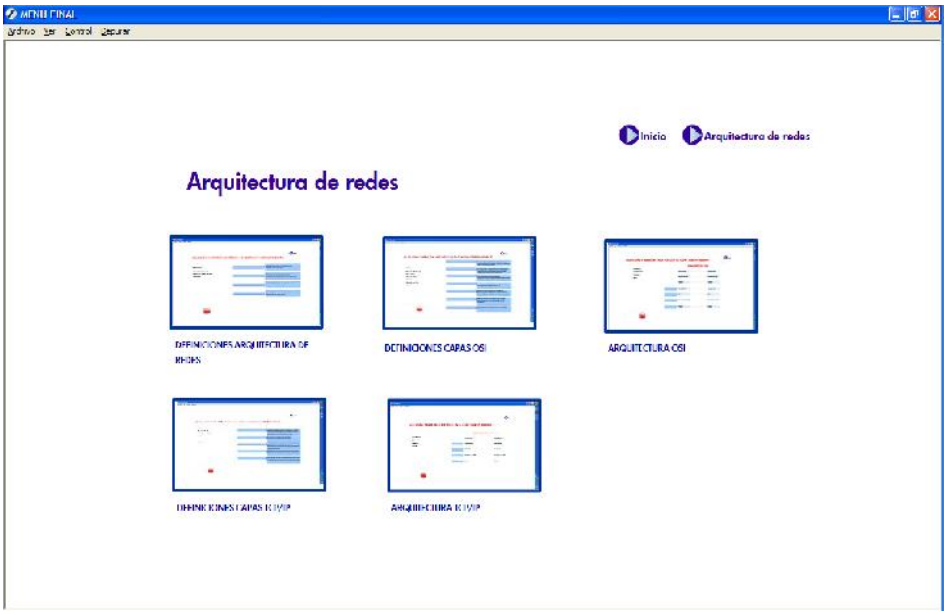


Figura 3.33

- DEFINICIONES DE ARQUITECTURA DE REDES

A través de la programación de ejercicios Drag&Drop, visto en el capítulo 2, se muestran una serie de términos en un lado y en el otro lado sus definiciones, pudiendo unir unas con otras.

Figura 3.34

- DEFINICIONES CAPA OSI

En este ejemplo, por medio de Drag&Drop, se realizará un ejercicio con las definiciones de las distintas capas de un sistema OSI.

Figura 3.35

- ARQUITECTURA OSI

Ejercicio Drag&Drop donde debemos colocar cada PDU con su capa correspondiente.

Figura 3.36

- DEFINICIONES CAPAS TCP/IP

En este ejemplo, por medio de Drag&Drop, se realizará un ejercicio con las definiciones de las distintas capas de un sistema TCP/IP.

Figura 3.37

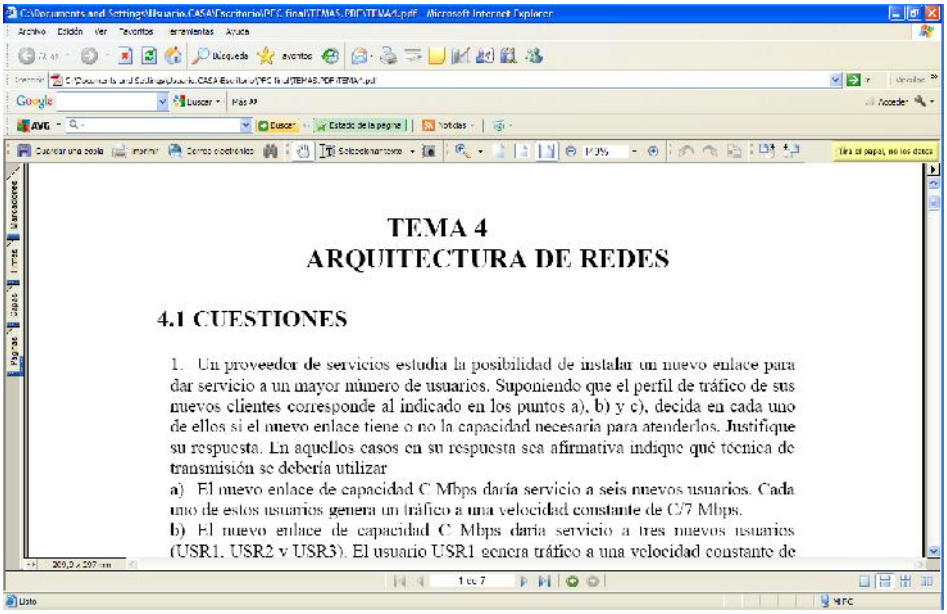
- ARQUITECTURA TCP/IP

Ejercicio Drag&Drop donde debemos colocar cada PDU con su capa correspondiente.

Figura 3.38

3.6.3 Botón Ejercicios Propuestos

Similar al *Apartado 3.3.2* el cual hace referencia a la función del botón de ejercicios propuestos.


TEMA 4
ARQUITECTURA DE REDES

4.1 CUESTIONES

1. Un proveedor de servicios estudia la posibilidad de instalar un nuevo enlace para dar servicio a un mayor número de usuarios. Suponiendo que el perfil de tráfico de sus nuevos clientes corresponde al indicado en los puntos a), b) y c), decida en cada uno de ellos si el nuevo enlace tiene o no la capacidad necesaria para atenderlos. Justifique su respuesta. En aquellos casos en su respuesta sea afirmativa indique qué técnica de transmisión se debería utilizar
a) El nuevo enlace de capacidad C Mbps daría servicio a seis nuevos usuarios. Cada uno de estos usuarios genera un tráfico a una velocidad constante de C/7 Mbps.
b) El nuevo enlace de capacidad C Mbps daría servicio a tres nuevos usuarios (USR1, USR2 y USR3). El usuario USR1 genera tráfico a una velocidad constante de

Figura 3.39

3.7 TRANSMISION DE DATOS Y TEORIA DE LA INFORMACION

Pulsando este botón nos saltará a la siguiente pantalla.

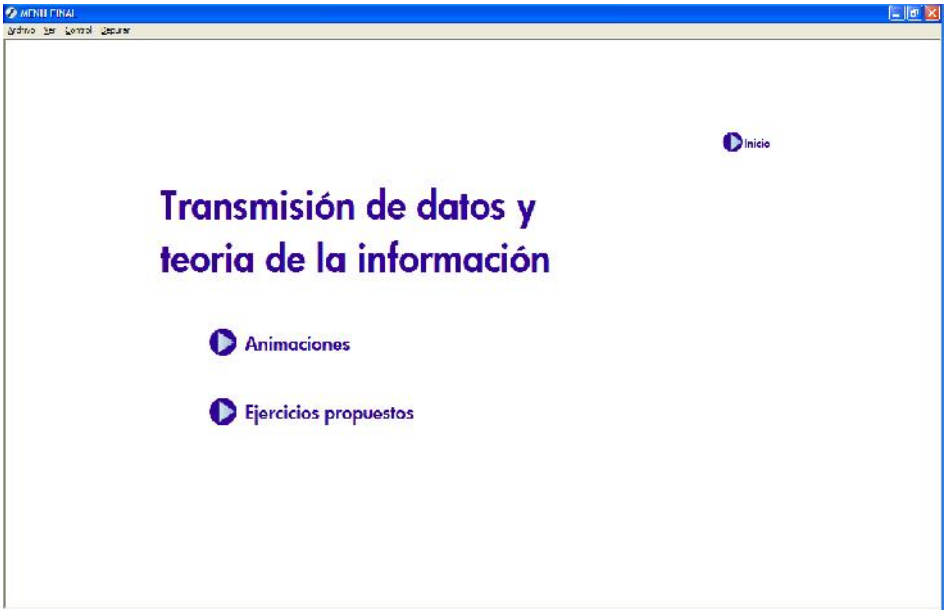


Figura 3.40

3.7.1 Botón Animaciones

En estas animaciones vamos a ver los distintos tipos de errores que se producen en la transmisión de datos según el tipo de paridad.
La detección y corrección de errores es una importante práctica para el mantenimiento e integridad de los datos a través de canales ruidosos y medios de almacenamiento poco confiable.

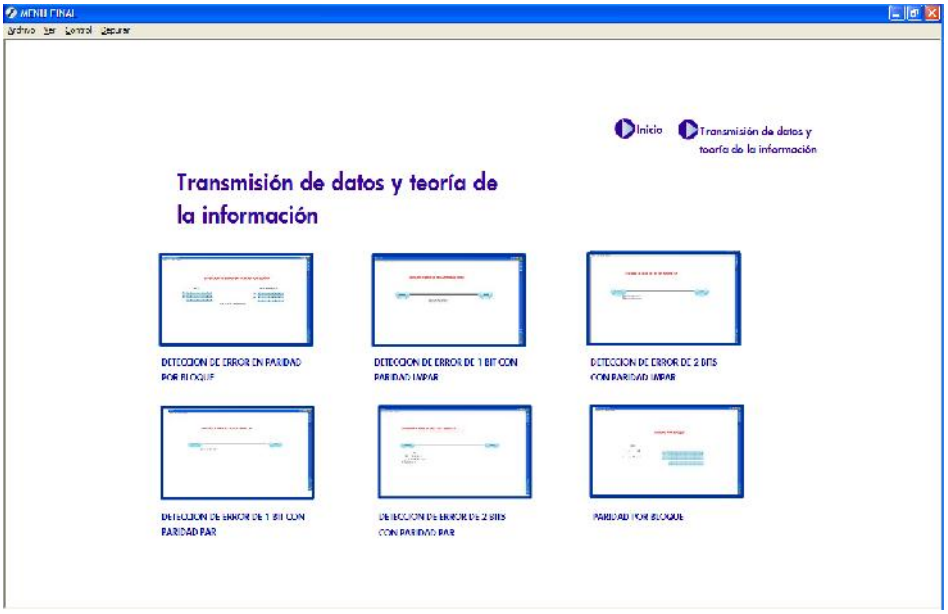


Figura 3.41

- DETECCION DE ERROR EN PARIDAD POR BLOQUE

En esta animación se explica cómo se detecta un error cuando se transmiten datos en bloque gracias a los bits de paridad.

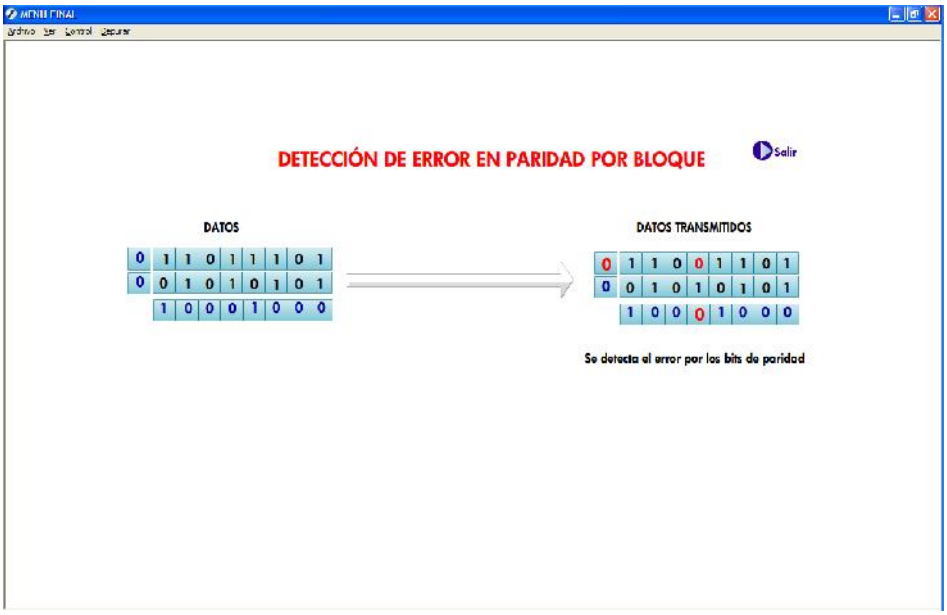


Figura 3.42

- DETECCION DE ERROR DE 1 BIT CON PARIDAD IMPAR

En esta animación se explica cómo se detecta un error cuando se transmiten datos con paridad impar y en la transmisión cambia un bit.

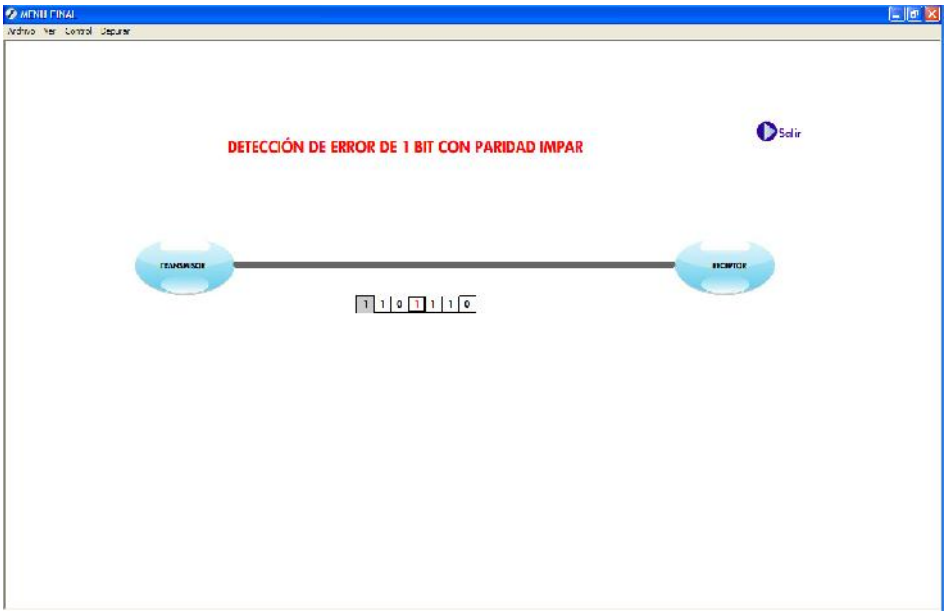


Figura 3.43

- DETECCION DE ERROR DE 2BITS CON PARIDAD IMPAR

En esta animación se explica cómo NO se detecta un error cuando se transmiten datos con paridad impar y en la transmisión cambian un par de bits.

Figura 3.44

- DETECCION DE ERROR DE 1BIT CON PARIDAD PAR

En esta animación se explica cómo se detecta un error cuando se transmiten datos con paridad par y en la transmisión cambia un bit.

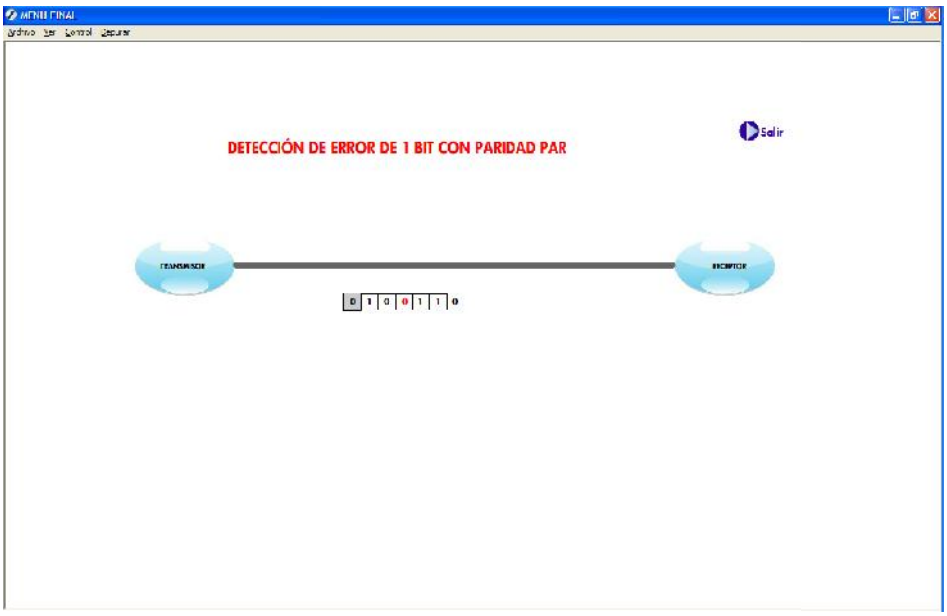


Figura 3.45

- DETECCION DE ERROR DE 2BITS CON PARIDAD PAR

En esta animación se explica cómo NO se detecta un error cuando se transmiten datos con paridad par y en la transmisión cambian un par de bits.

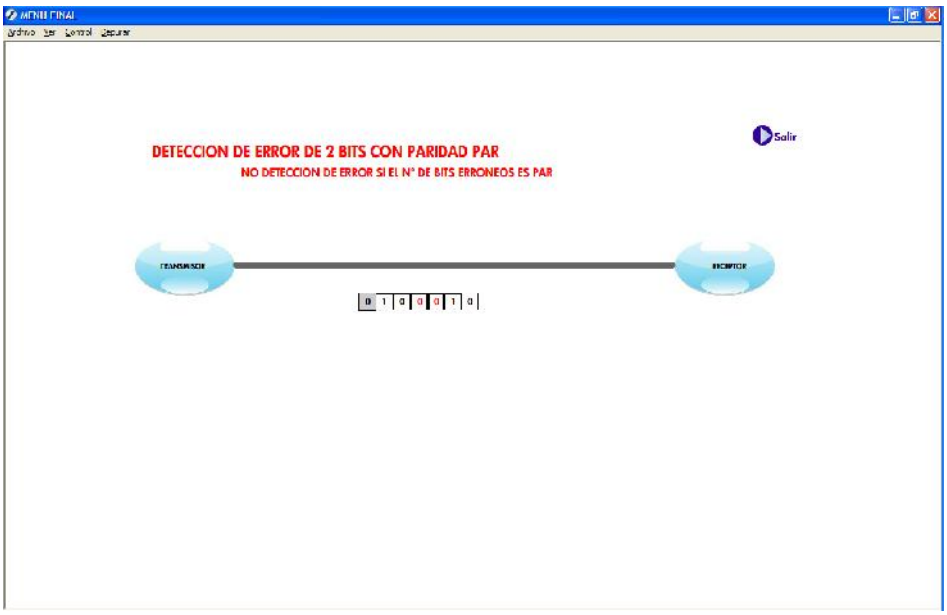


Figura 3.46

- PARIDAD POR BLOQUE

En esta animación observamos el funcionamiento de la paridad por bloque

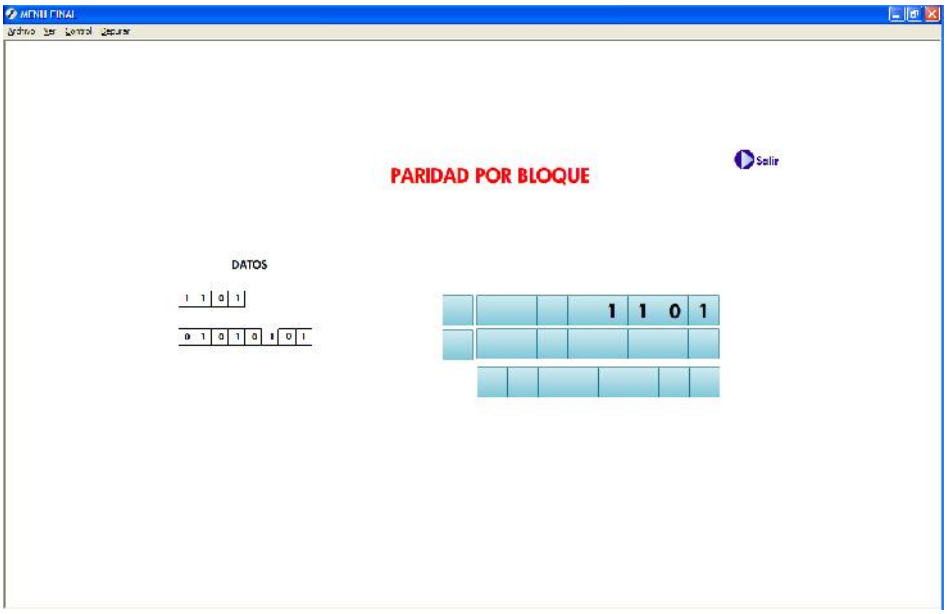


Figura 3.47

3.7.2 Botón Ejercicios Propuestos

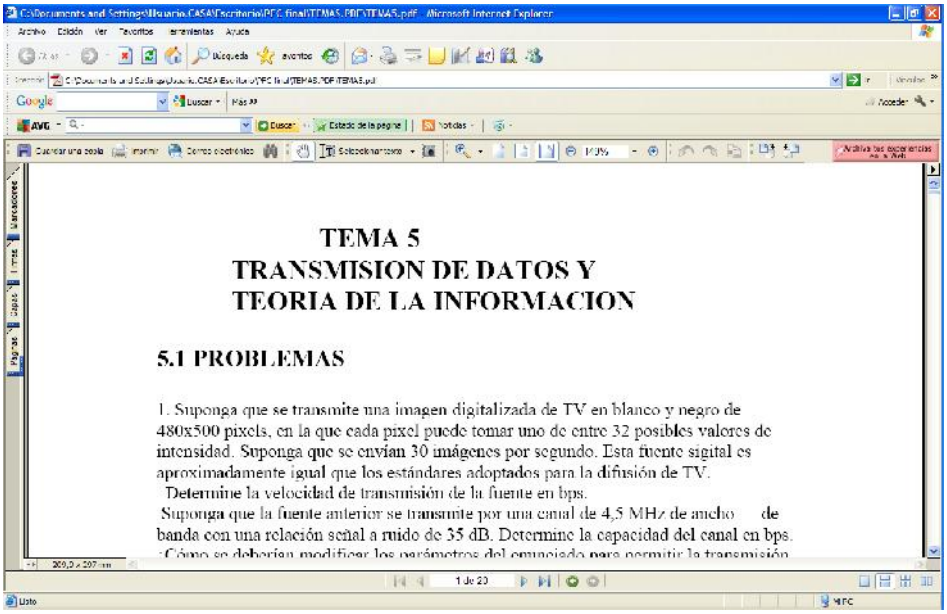


Figura 3.48

3.8 TIPOS DE TRANSMISION DE DATOS

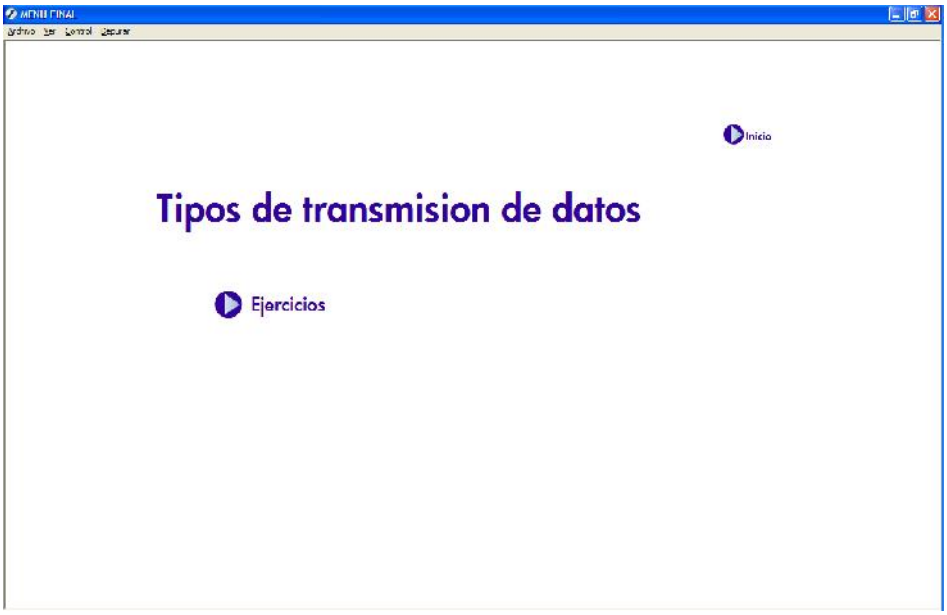


Figura 3.49

3.8.1 Botón Ejercicios.

Tenemos dos tipos de ejercicios Drag&Drop: Modulación y Perturbación.

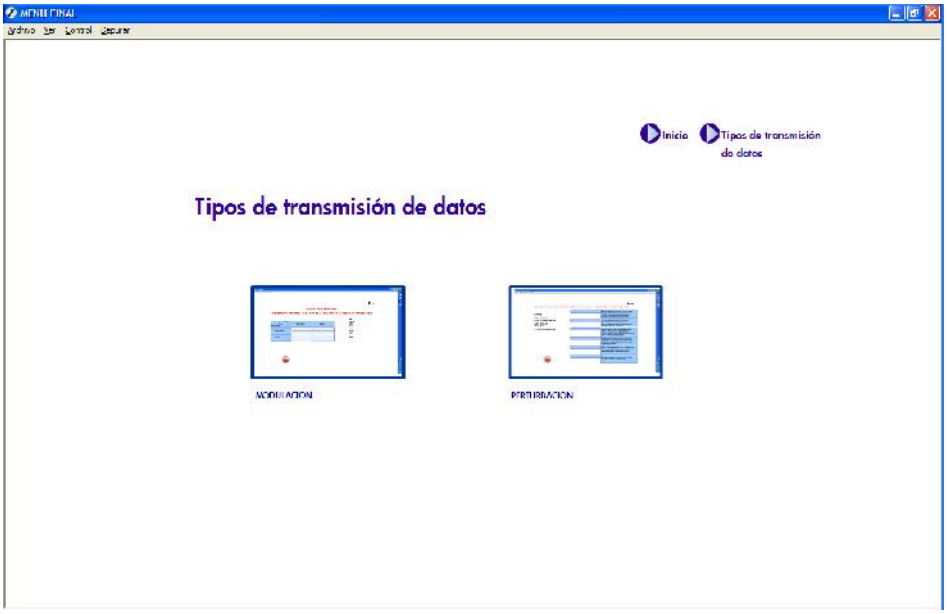


Figura 3.50

- MODULACION.

A través de la programación de ejercicios Drag&Drop, visto en el capítulo 2, se muestran los tipos de modulación y en el otro lado las distintas técnicas.

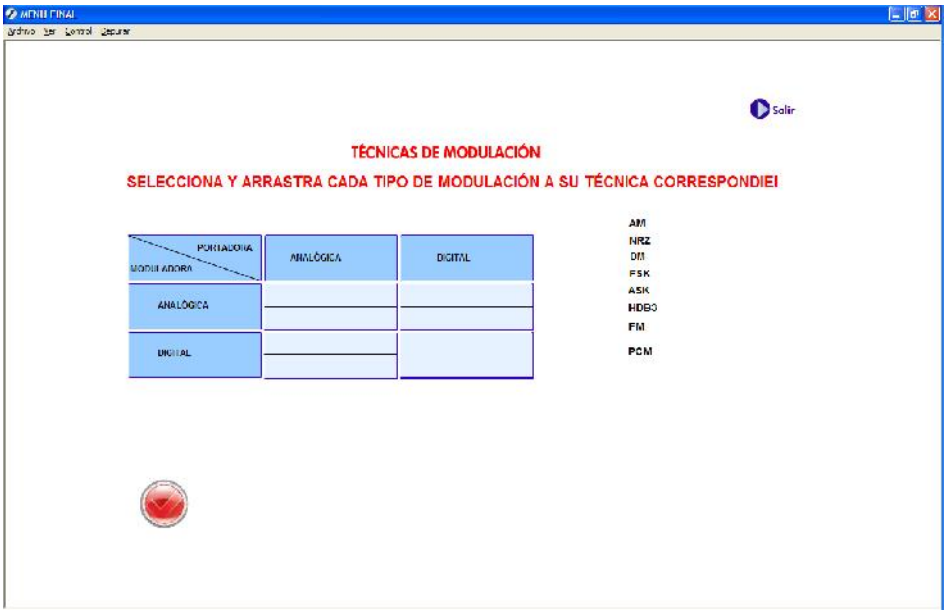


Figura 3.51

- PERTURBACION.

A través de la programación de ejercicios Drag&Drop se muestran los tipos de perturbaciones y en el otro lado sus definiciones, pudiendo unir unas con otras.

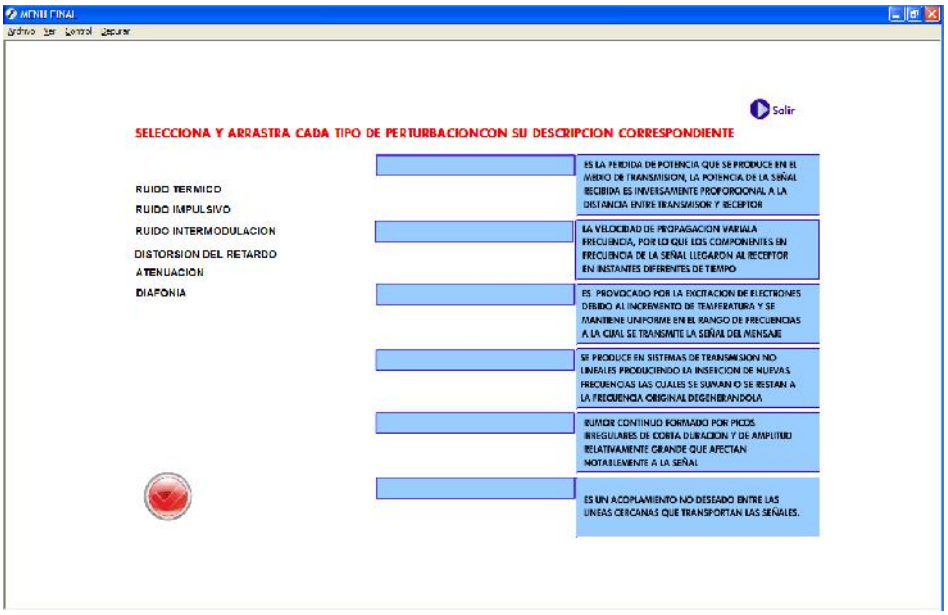


Figura 3.52

3.9 INTERFACES NIVEL FISICO RS232 y USB

Figura 3.53

3.9.1 Botón Animaciones.

Figura 3.52

- COMUNICACIÓN SOBRE LINEA TELEFONICA CONVENCIONAL. RS-232

En esta animación observamos el funcionamiento de una conexión RS-232

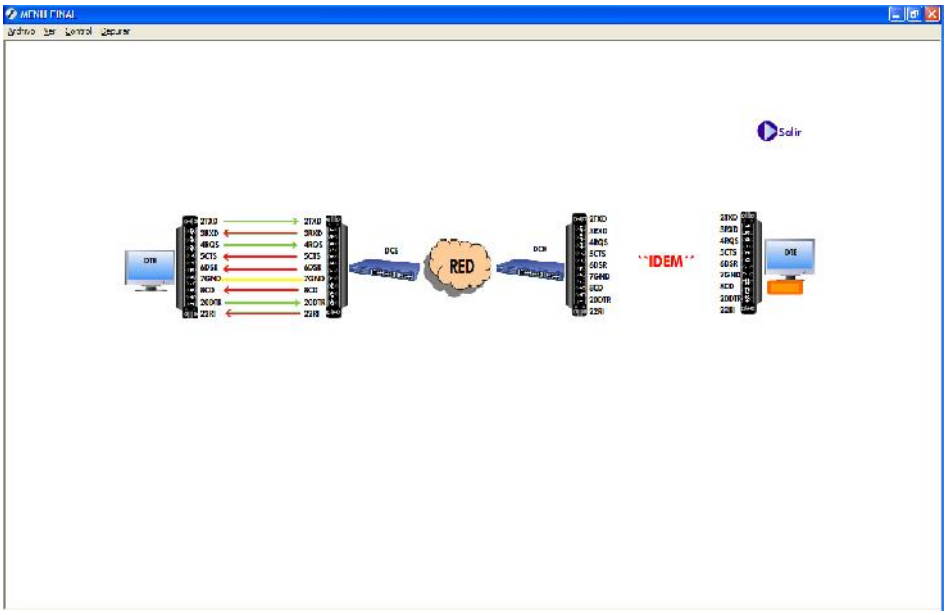


Figura 3.53

3.9.2 Botón Test

Este botón salta a la siguiente pantalla que vemos a continuación donde podremos acceder a dos tipos de test: RS-232 y USB.

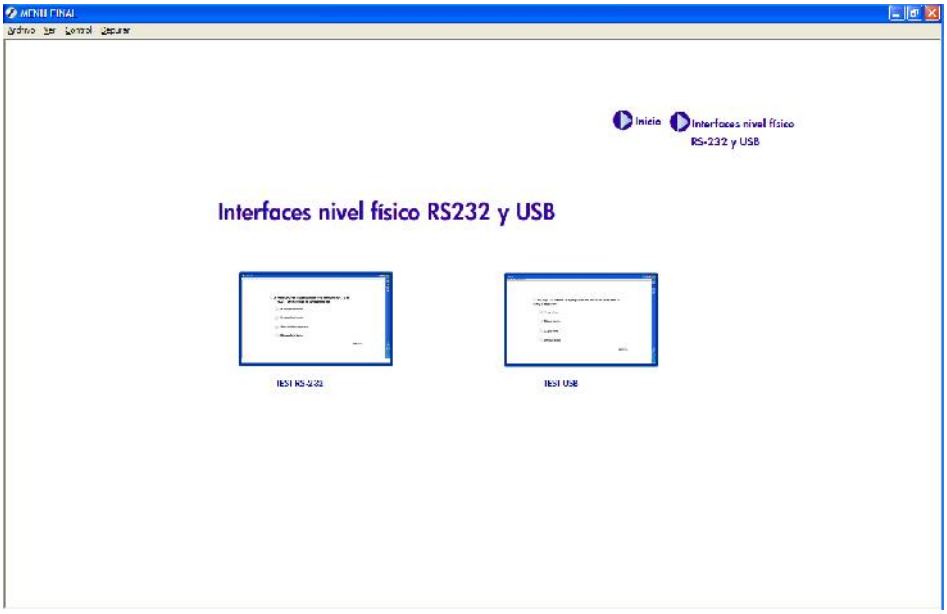


Figura 3.54

- TEST RS-232

El botón indicado abre una película que contiene un tipo test, el cual consta de 10 a 15 preguntas. Recopilación de cuestiones, algunas de exámenes, configurado de la forma explicada en el capítulo 2.

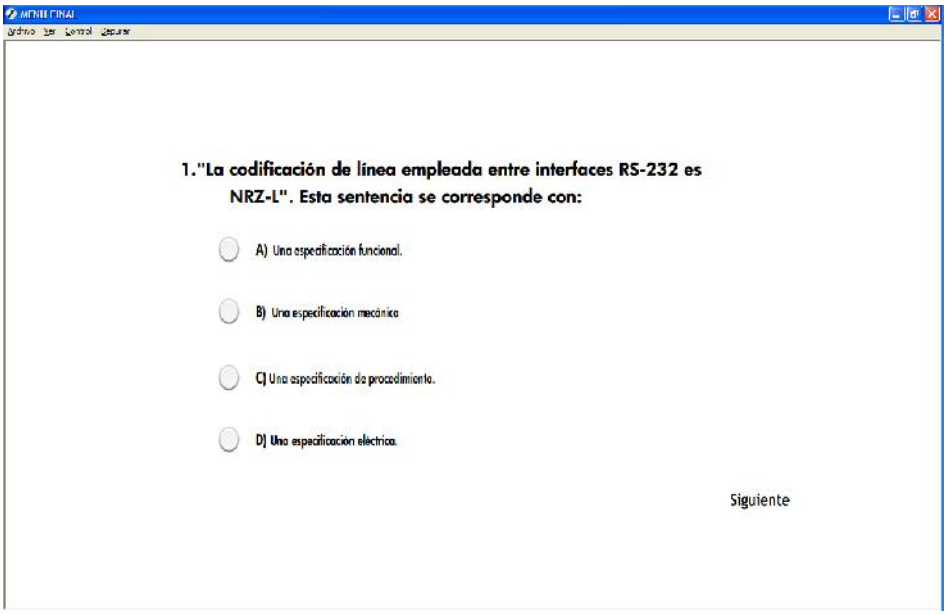


Figura 3.55

- TEST USB

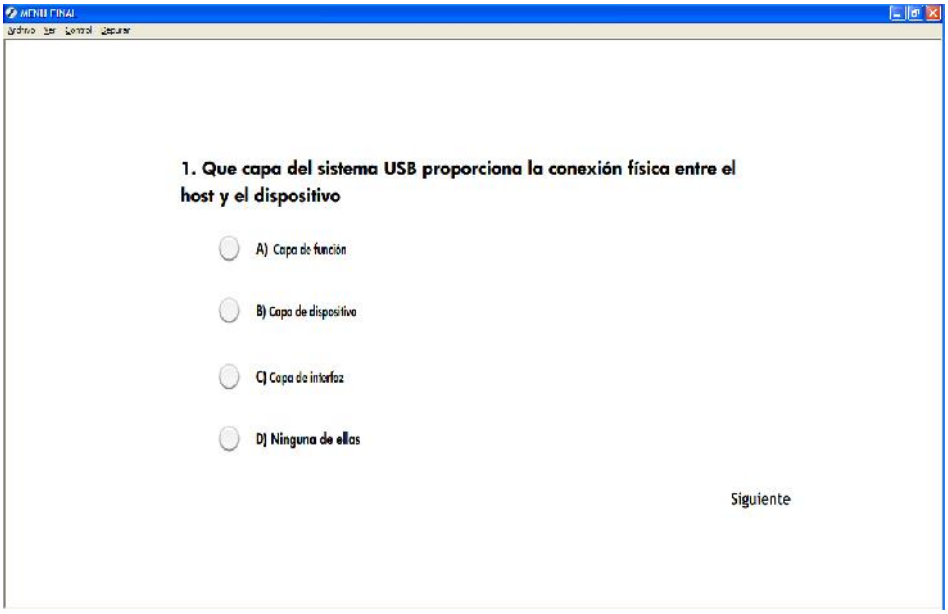


Figura 3.56

3.9.3 Botón Ejercicios Propuestos

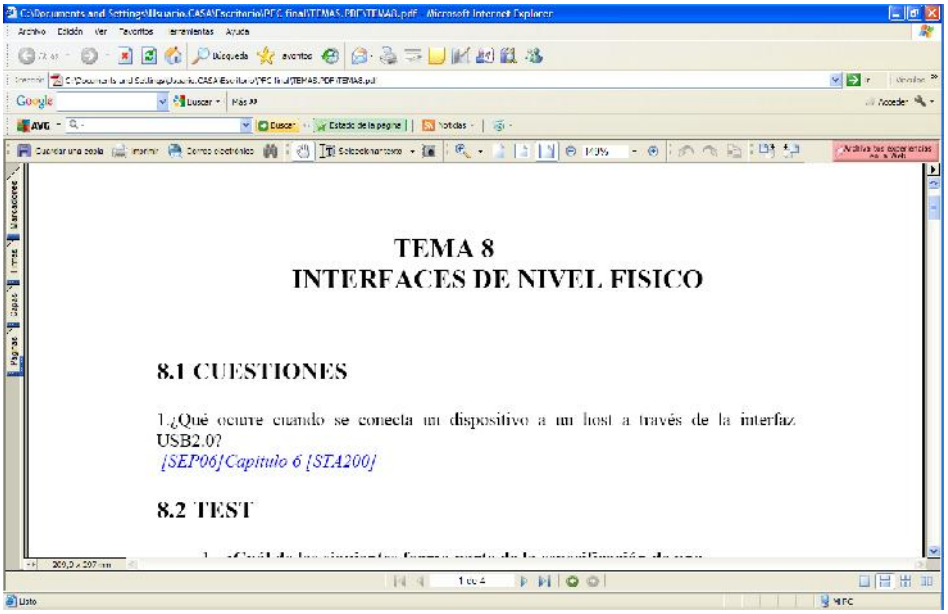


Figura 3.57

3.10 NIVEL DE ENLACE DE DATOS

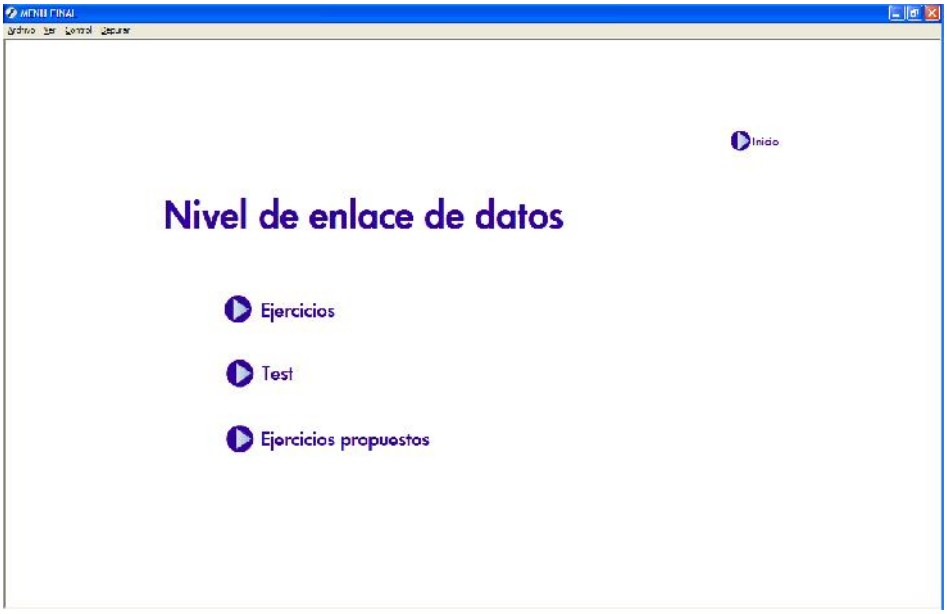


Figura 3.58

3.10.1 Botón Ejercicios

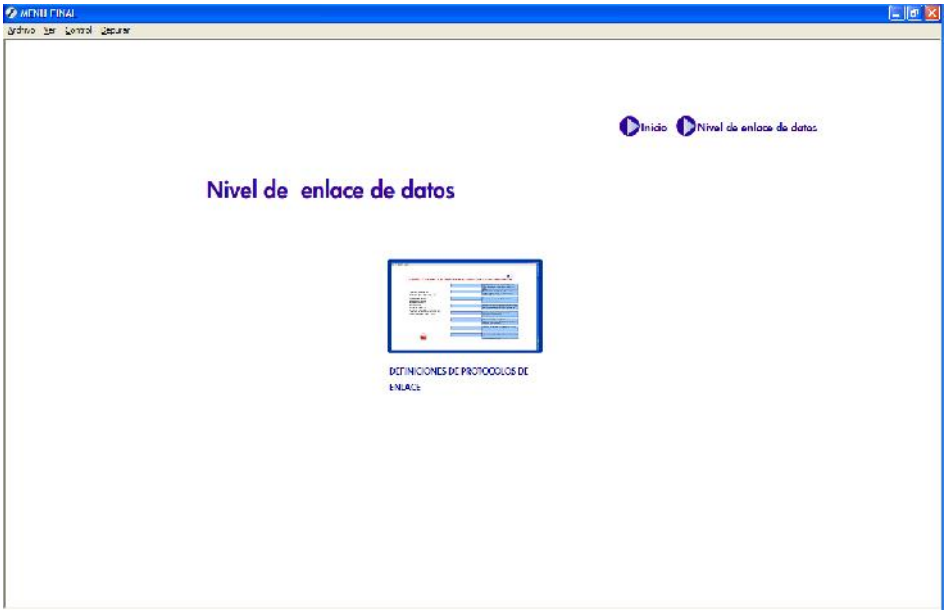


Figura 3.59

- DEFINICIONES DE LAS FUNCIONES DE PROTOCOLOS DE ENLACE

Ejercicio Drag&Drop donde tenemos en un lado las distintas funciones de los protocolos de enlace y en el otro las definiciones.

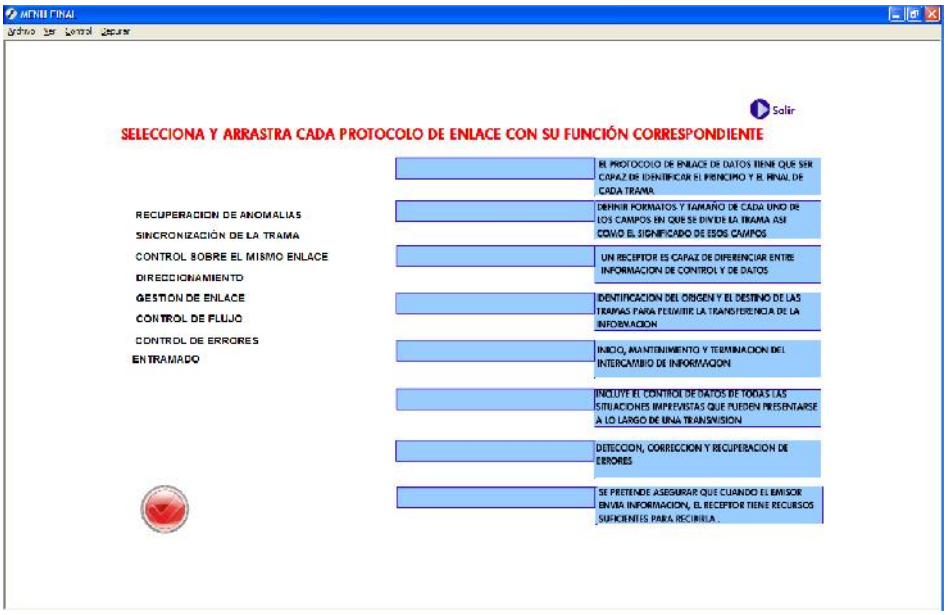


Figura 3.60

3.10.2 Botón Test

Este botón salta a la siguiente pantalla que vemos a continuación, donde podremos acceder a un test acerca de los tipos de enlace.

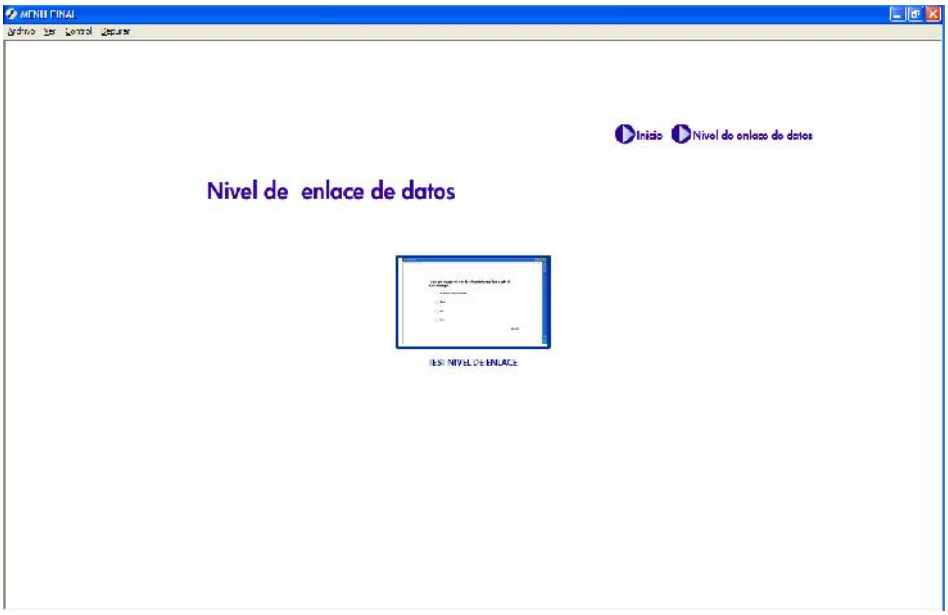


Figura 3.61

- TEST NIVEL DE ENLACE

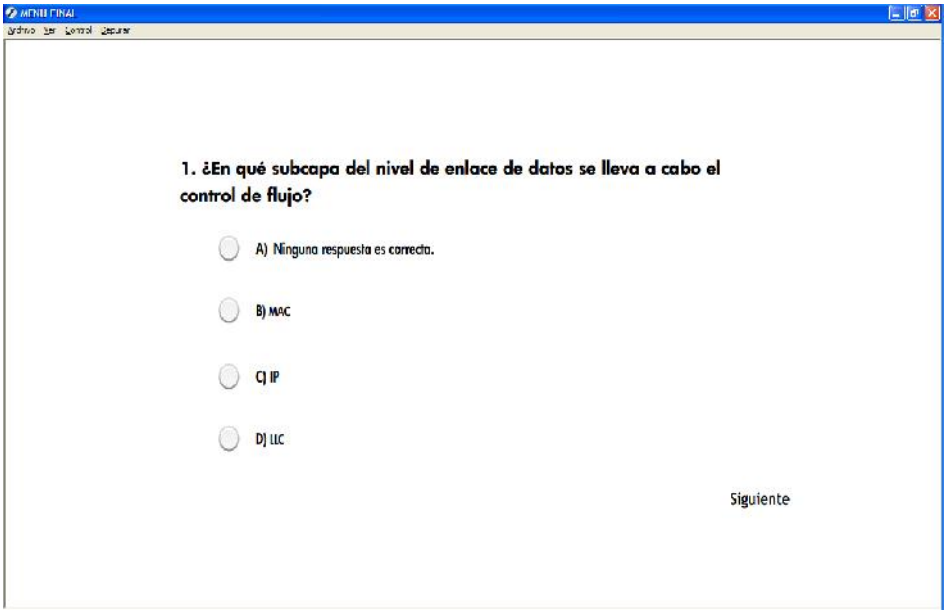


Figura 3.62

3.10.3 Botón Ejercicios Propuestos

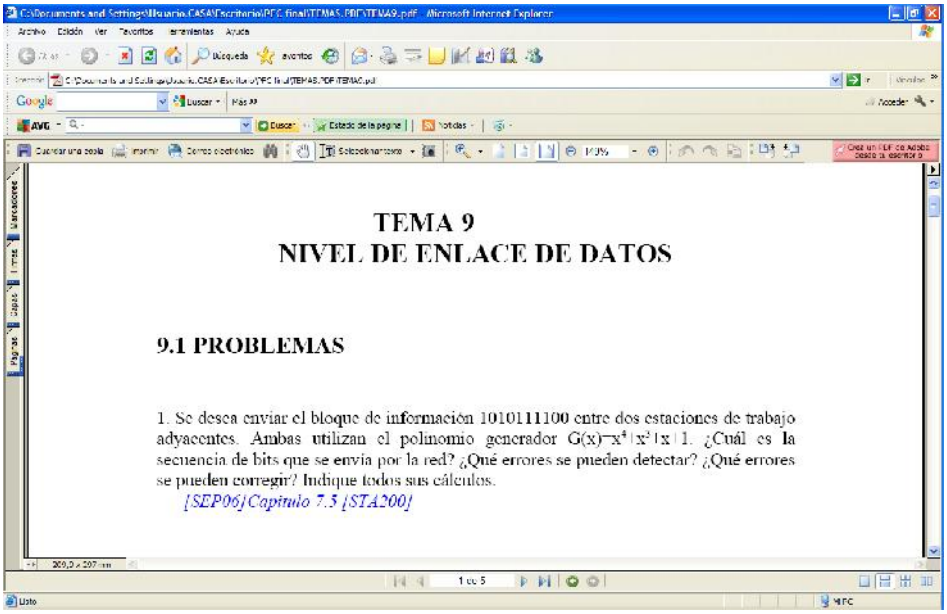


Figura 3.63

3.11 NIVEL DE ENLACE DE DATOS: CONTROL DE FLUJO

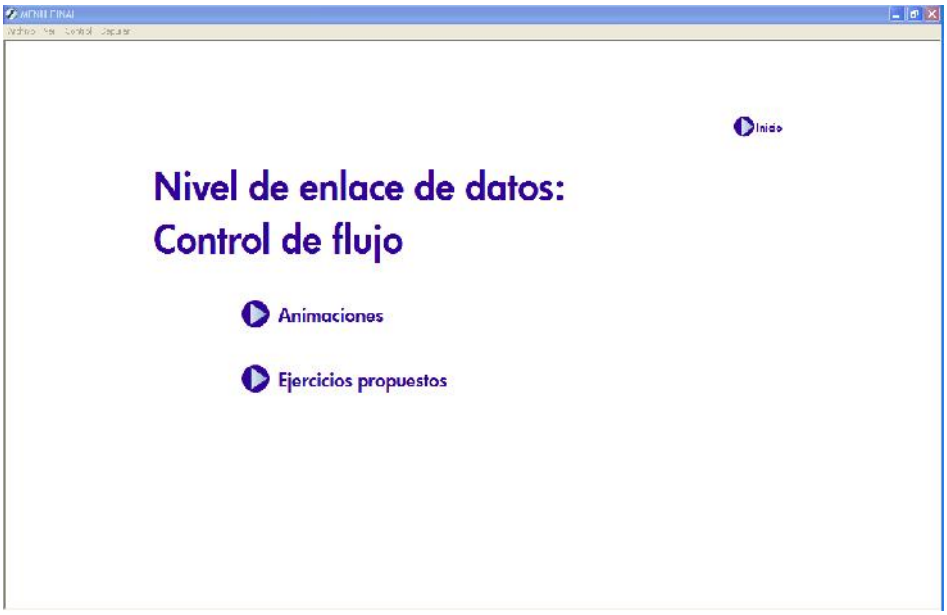


Figura 3.64

3.11.1 Botón Animaciones

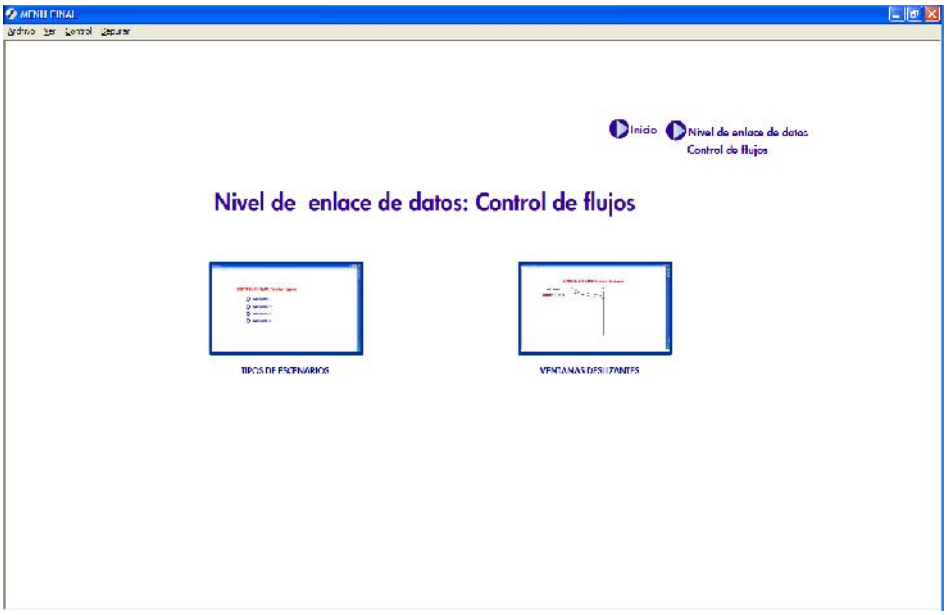


Figura 3.64

- TIPOS DE ESCENARIOS

En esta animación tenemos varios botones donde presionando sobre cada uno de ellos nos lleva a ver los diferentes escenarios del control de flujo de parada y espera.

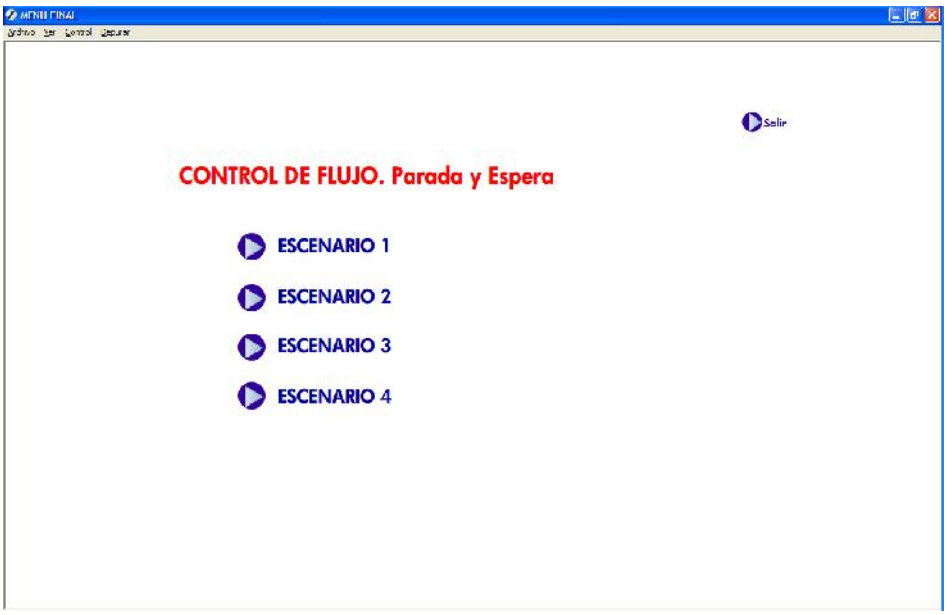


Figura 3.65

ESCENARIO1. La trama se recibe correctamente y el reconocimiento llega a tiempo.

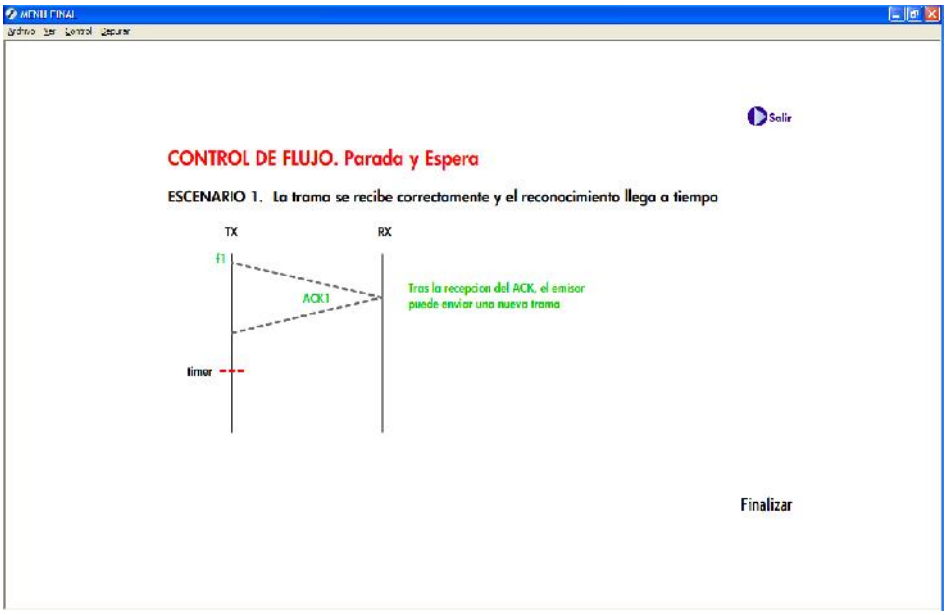


Figura 3.66

ESCENARI0 2. La trama se pierde.

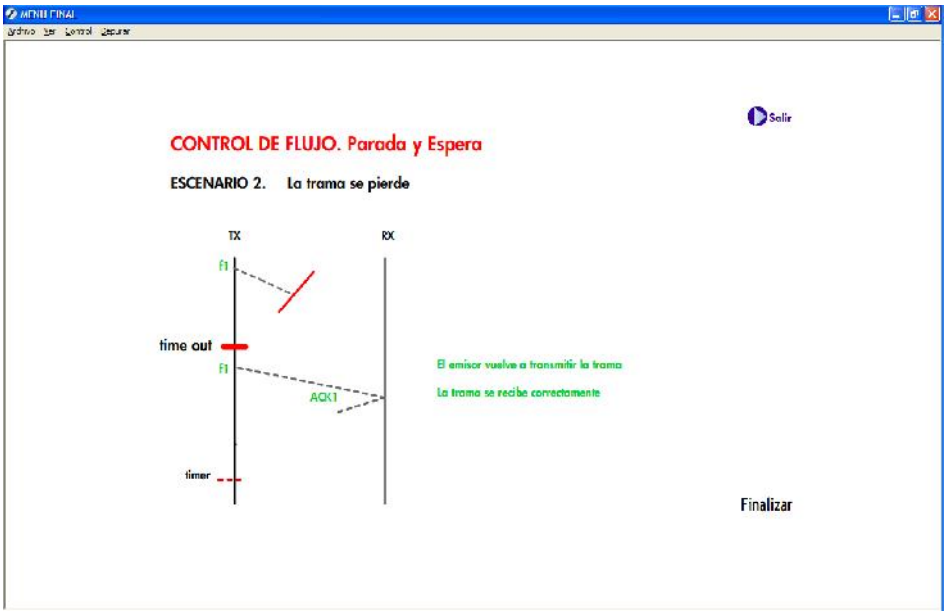


Figura 3.67

ESCENARI0 3. La trama se recibe correctamente pero el reconocimiento se pierde

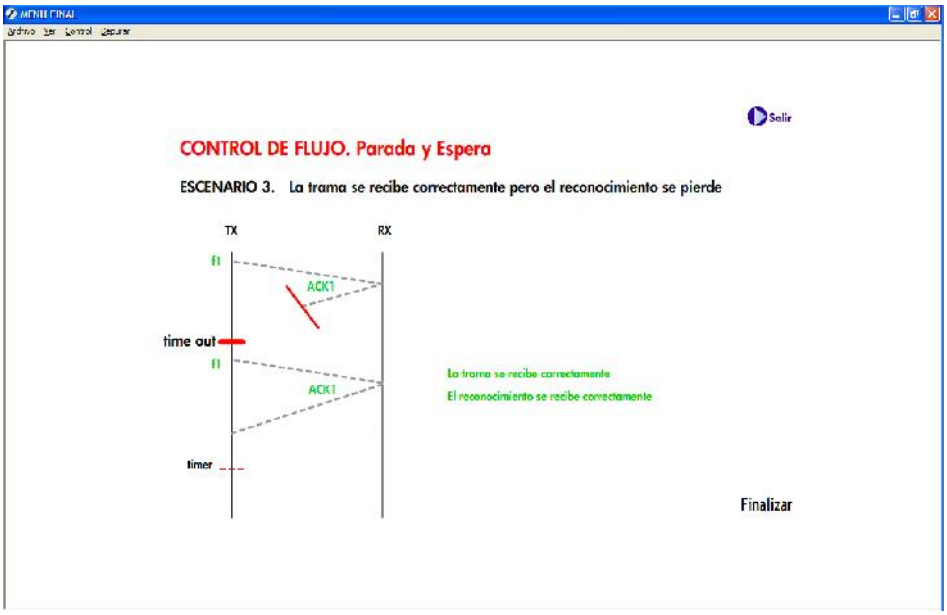


Figura 3.68

ESCENARI0 4.El temporizador vence antes de que llegue el reconocimiento

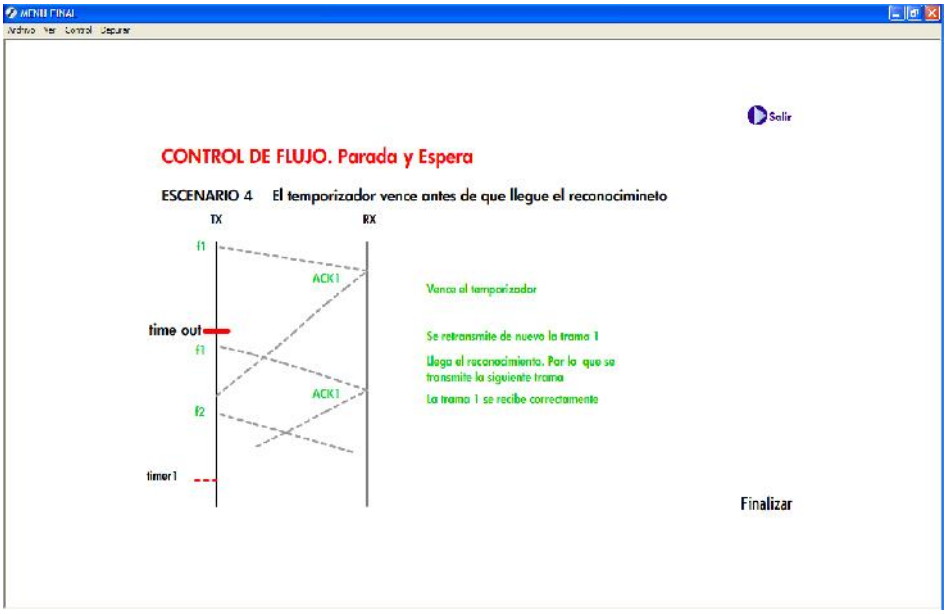


Figura.3.69

- VENTANA DESLIZANTE

En esta animación observamos el funcionamiento de la ventana deslizante durante la transmisión de datos.

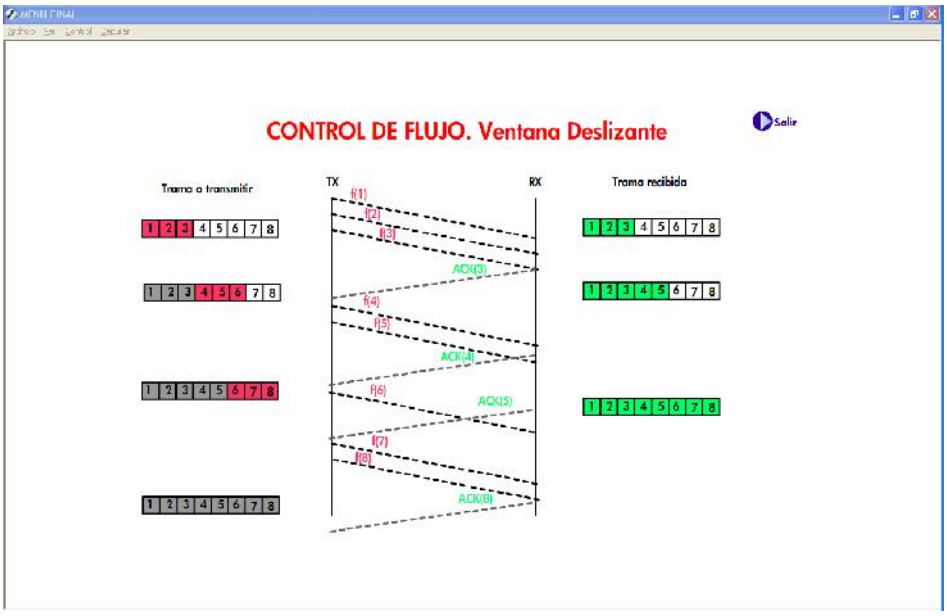


Figura 3.70

3.11.2 Botón Ejercicios Propuestos

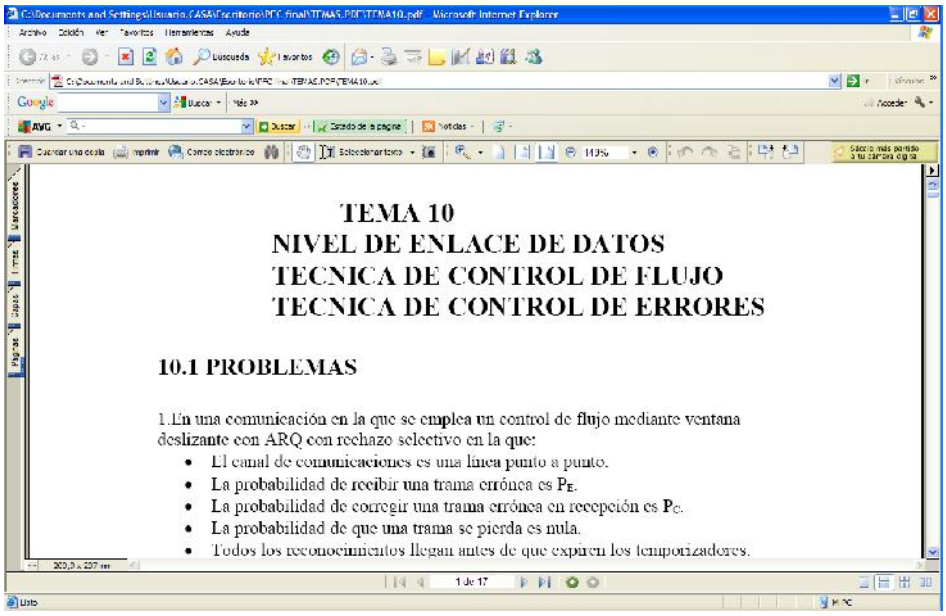


Figura 3.71

3.12 NIVEL DE ENLACE DE DATOS: CONTROL DE ERRORES

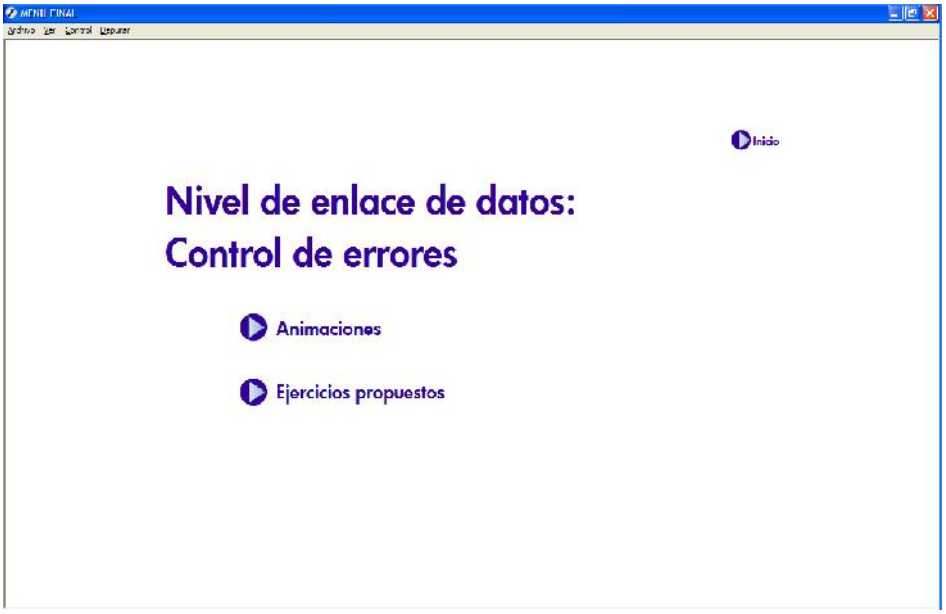


Figura 3.72

3.12.1 Botón Animaciones

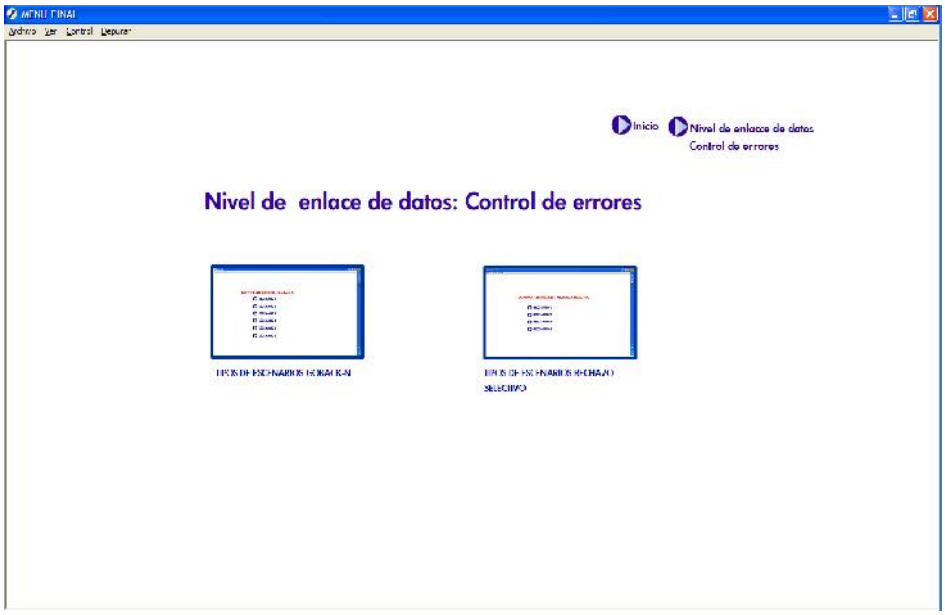


Figura 3.73

- TIPOS DE ESCENARIOS GO-BACK-N

En esta animación tenemos varios botones donde presionando sobre cada uno de ellos nos lleva a ver los diferentes escenarios del control de errores Go-Back-N.

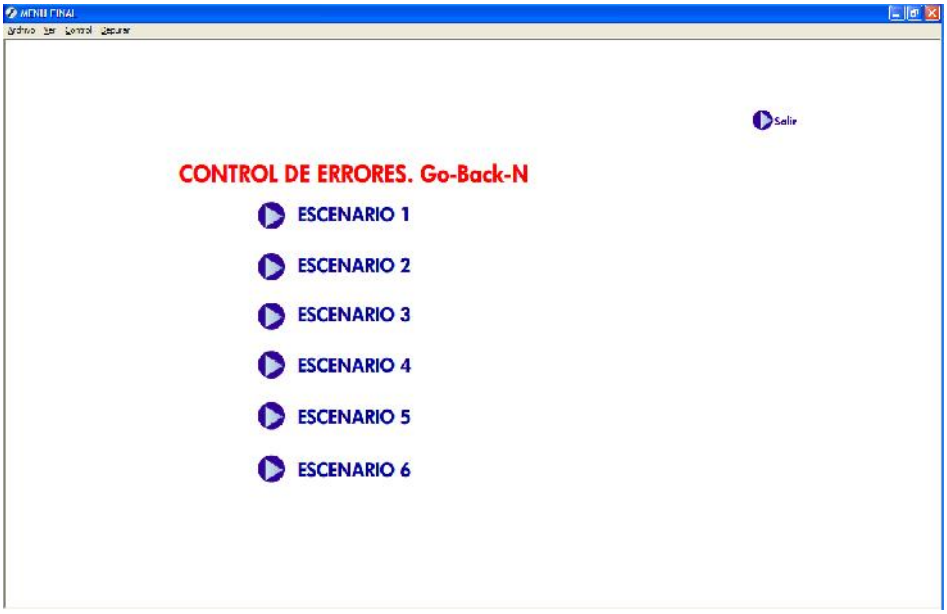


Figura 3.74

ESCENARIO1. Se deteriora la trama de información.

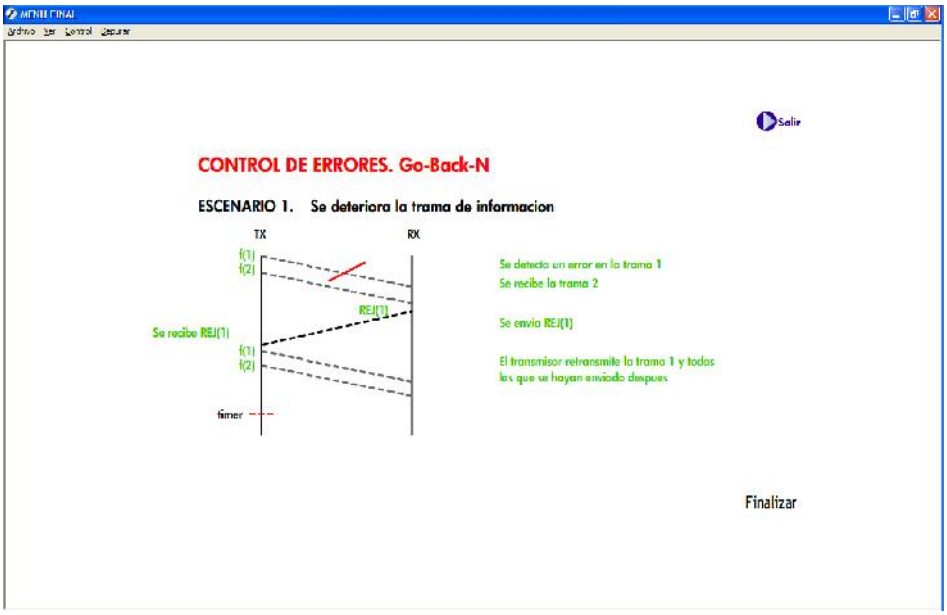


Figura 3.75

ESCENARIO2. Se pierde la trama de información (I).

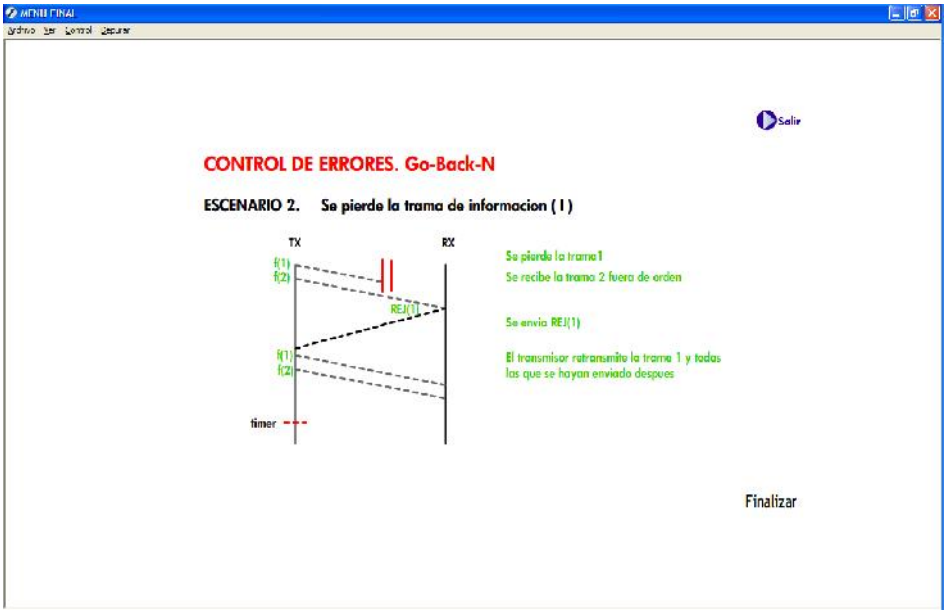


Figura 3.76

ESCENARIO3. Se pierde la trama de información (II).

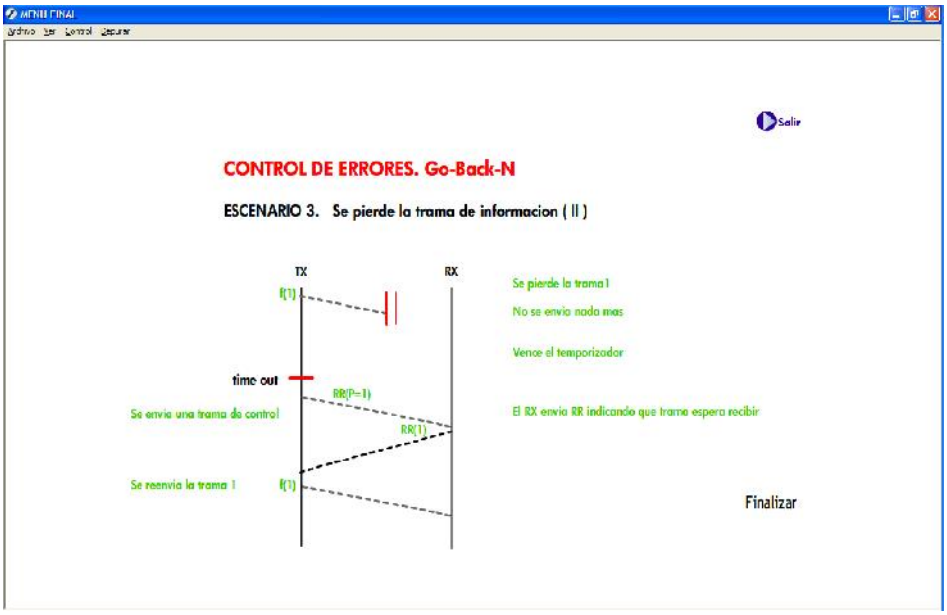


Figura 3.77

ESCENARIO4. Se pierde o se deteriora RR (P=1).

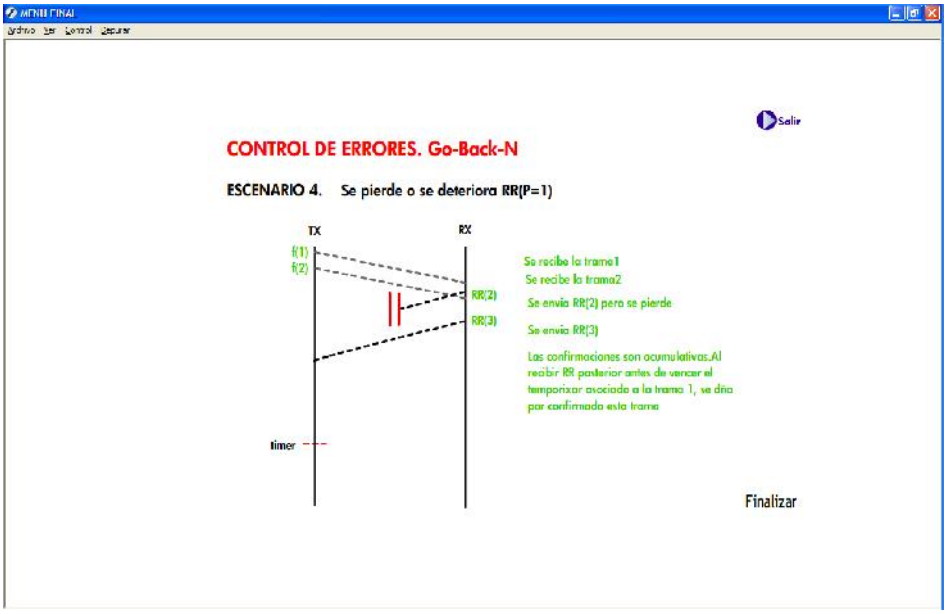


Figura 3.78

ESCENARIO5. Se pierde o se deteriora RR (P=1) II

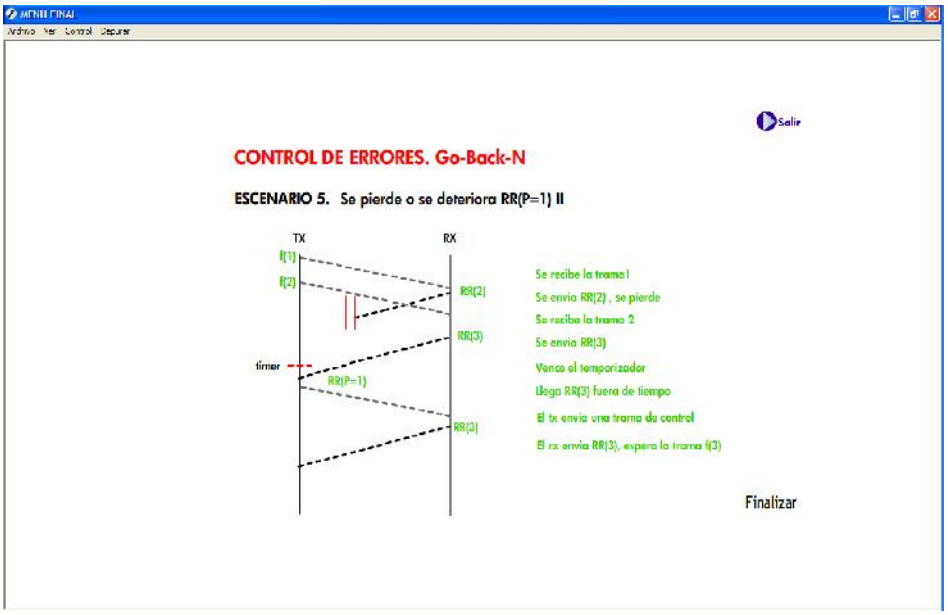


Figura 3.79

ESCENARIO6. Se pierde o se deteriora la trama REJ().

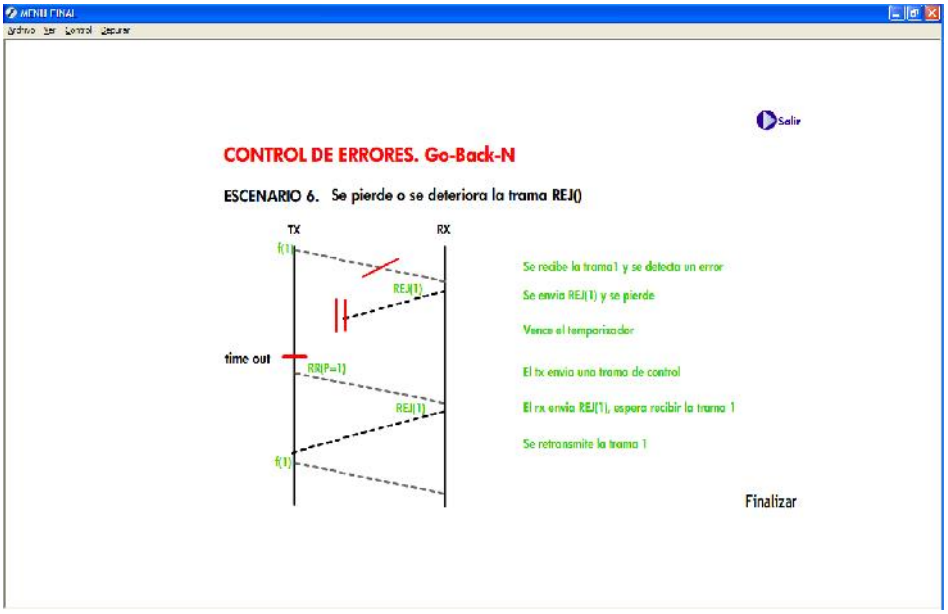


Figura 3.80

- TIPOS DE ESCENARIOS RECHAZO SELECTIVO

En esta animación tenemos varios botones donde presionando sobre cada uno de ellos nos lleva a ver los diferentes escenarios del control de errores Rechazo Selectivo.

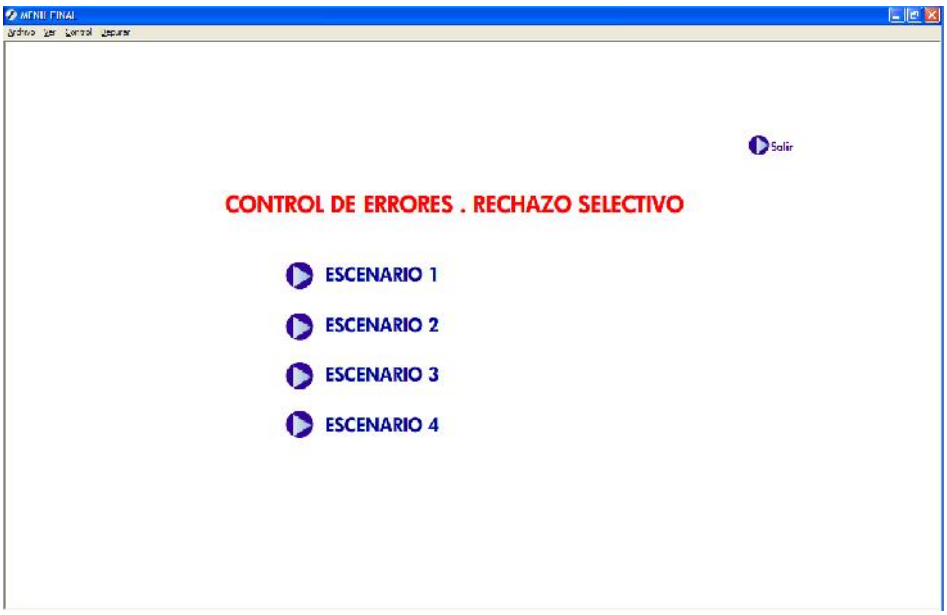


Figura 3.81

ESCENARIO 1. Se pierde la trama de información.

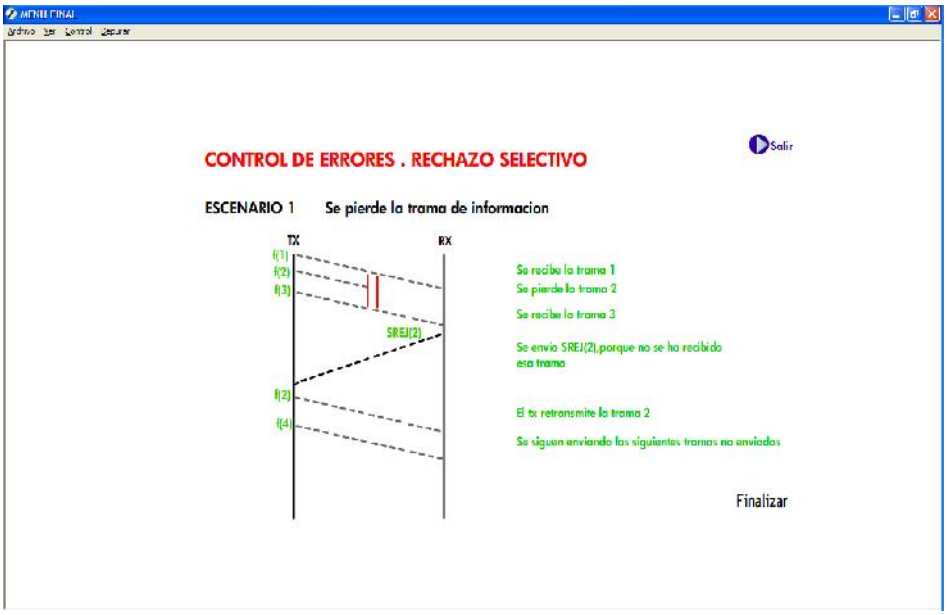


Figura 3.82

ESCENARIO 2. La trama de información llega errónea.

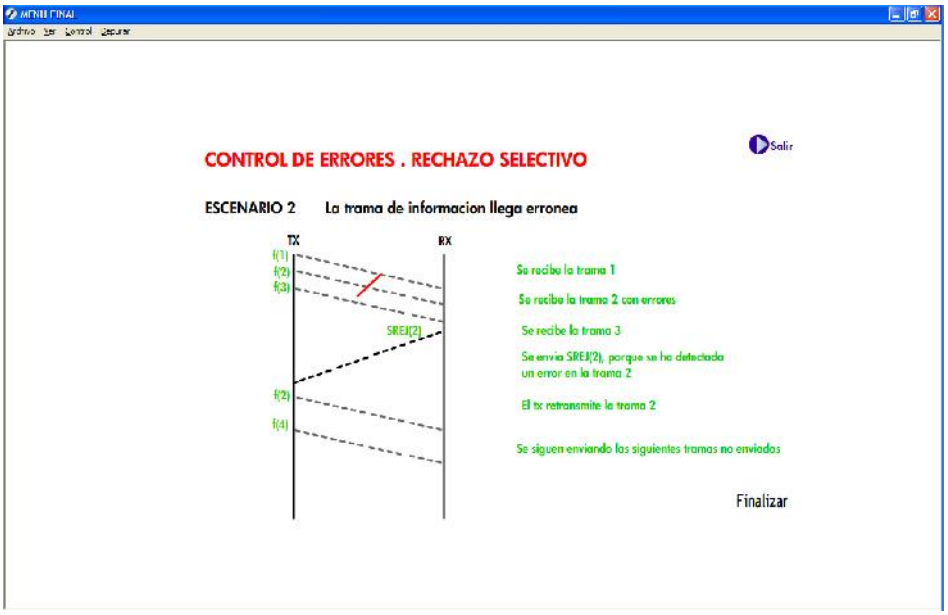


Figura 3.83

ESCENARIO 3. No llega el RR

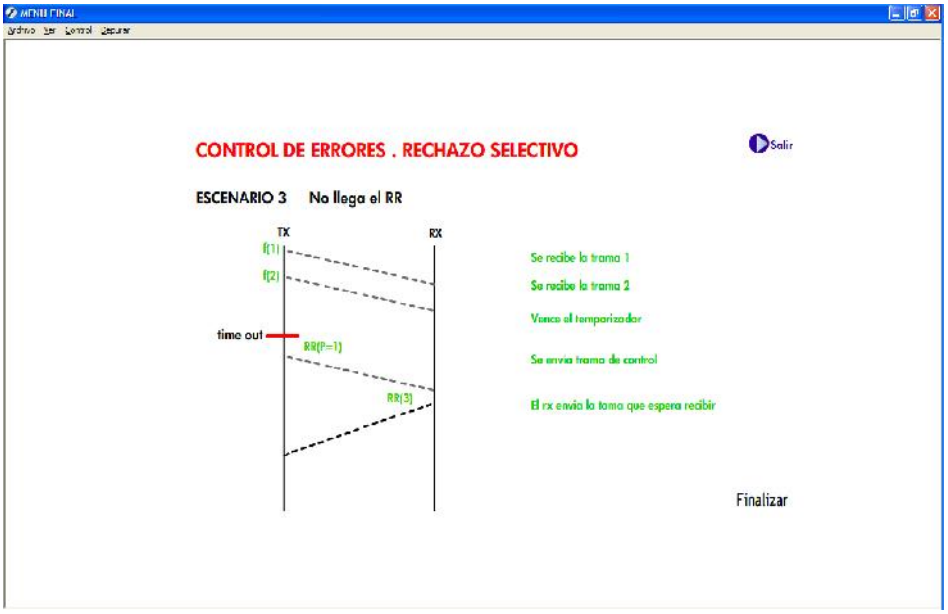


Figura 3.84

ESCENARIO 4. No hay errores ni perdidas.

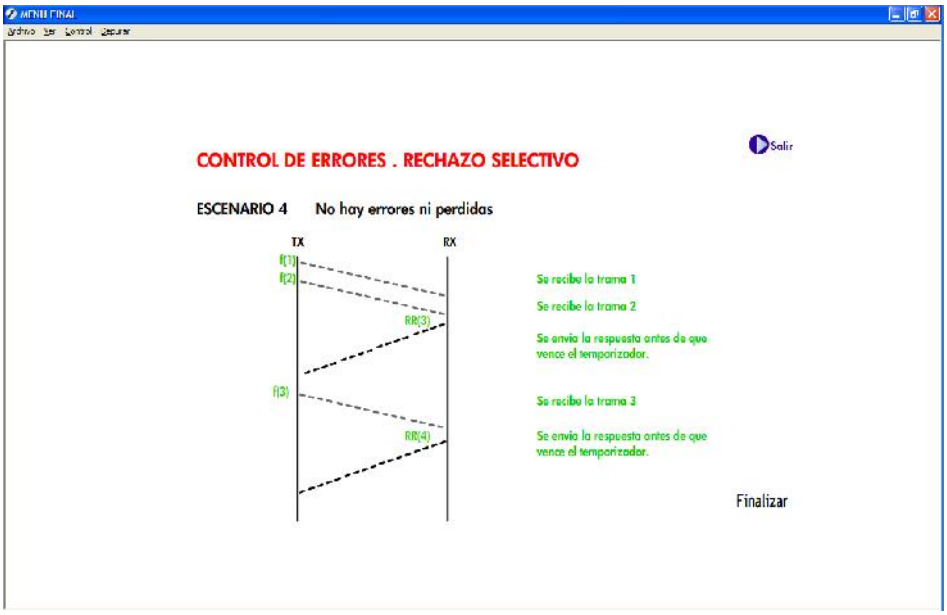


Figura 3.85

3.12.2 Botón Ejercicios Propuestos

Desde este botón se puede acceder al mismo test del tema anterior, ya que en este tema hay ejercicios relacionados con el control de flujo y el control de errores.

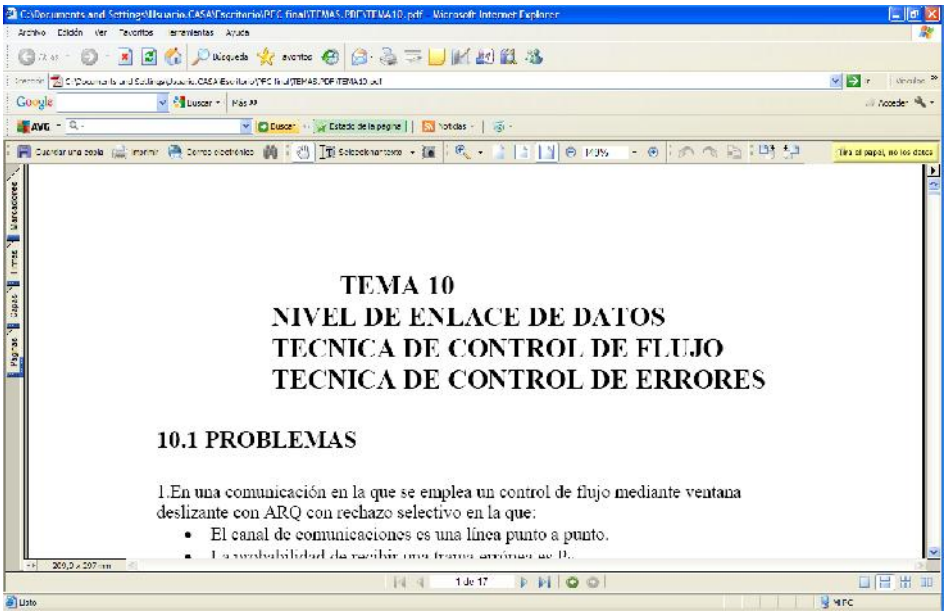


Figura 3.86

CAPITULO 4. CONCLUSIONES Y TRABAJOS FUTUROS

4.1 CONCLUSIONES

Con este proyecto se ha conseguido ofrecer al alumno y al profesor una herramienta, la cual pueda ayudar a ambos al estudio y la ampliación de material de docencia para que se entienda con más facilidad la asignatura.

Gracias al uso de Flash como herramienta potente, permite crear dinamismo, con animaciones de todo tipo, permitiendo al usuario verlo como algo más divertido y sucesos más visibles.

Se ha creado un CD-Rom, que complementa la enseñanza presencial, el cual contiene una serie de animaciones, para todas las definiciones más importante y complejas de entender, todo tipo de documentación de exámenes necesario para ayudar al alumno en su estructura y varios tipos de ejercicios, como test, unir términos con sus definiciones, seleccionar la respuesta correcta…, facilitando una aclaración a su conocimiento.

Para el desarrollo tuvo que hacerse un estudio exhaustivo de todo el contenido, después buscando los objetos más adecuados para su desarrollo y cómo implementarlos. Todo el desarrollo del código en ActionScript se hizo a través del panel de acciones, herramienta de Macromedia Flash.

Como resultado se ha obtenido una herramienta de estudio muy útil para el alumno y a su vez se ha iniciado una nueva idea que complementa el material de enseñanza, ayudando a facilitar el estudio de una forma más entretenida.

4.2 TRABAJOS FUTUROS

Como principales trabajos futuros se podrían realizar los siguientes:

- Ampliación de la herramienta utilizada en un entorno web, donde cada profesor disponga de uno, en el cual expliquen de forma interactiva su temario y poder realizar ejercicios. También incorporaría una sección de dudas online, en el cual cada alumno pueda realizar sus preguntas, tanto instantáneamente como un chat a ciertas horas o por mensajes.

- Implementación de cursos online a distancia o semipresenciales por la universidad, utilizando esta herramienta para su página web principal.

CAPITULO 5. BIBLIOGRAFIA

5.1 Bibliografía

1. Tutoriales de Flash 8, Sorenson Spark. 2005

2. Apuntes, diapositivas y exámenes de la asignatura Telemática.

3. Ingeniería en micro-controladores. Protocolo USB, Eric López Pérez (www.i-micro.com).

4. Aprendizaje electrónico, Wikipedia.

5. E-Learning, Jorge A. Mendoza (www.informaticamilenium.com)

6. (www.videotutoriales.com)

7. (www.macromedia.com)

8. (www.livedocs.adobe.com/flash)

9. (www.programacion.com/tutoriales/flas)

10. DragandDrop flash exercises_ Macromedia flash

11. (www.reciseict.co.uk/flash)