

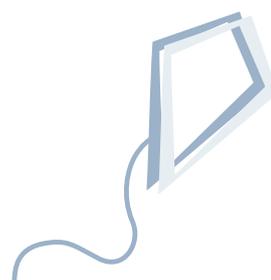


GOBIERNO
DE ESPAÑA

MINISTERIO
DE EDUCACIÓN

ACTION SCRIPT 2

PROGRAMACIÓN



 AULA
MENTOR

educacion.es



NIPO: 820-10-107-5

Autoría:

Violeta Menéndez

Fernando Posada

Edición y maquetación:

Sol Emilia Jiménez Alonso

Diseño gráfico de portada:

Lorena Gordo López

ActionScript 2

Presentación	5
1. Introducción	6
1.1 ¿Qué es ActionScript?	7
1.2 ¿Para qué ActionScript?	7
1.3 Panel de Acciones	8
1.4 Panel de Comportamientos	11
1.5 Actividad de aplicación 1: botón Google	11
1.6 Acciones en fotografías y objetos	13
1.7 Actividad de aplicación 2: vuelo de ovni	15
1.8 Panel de Salida	19
1.9 Programación Orientada a Objetos	20
2. Datos y variables	22
2.1 Introducción	23
2.2 Comentarios	23
2.3 Declaración de variables	23
2.4 Strings	25
2.5 Actividad de aplicación 3: saludo personalizado	27
2.6 Tipo de dato numérico	29
2.7 Actividad de aplicación 4: conversor de divisas	32
2.8 Otros tipos de datos	33
2.9 Uso de strings	35
2.10 Búsqueda de caracteres en un string	37
2.11 Actividad de aplicación 5: validación de formularios	38
2.12 Extracción de caracteres de un string	40
2.13 Actividad de aplicación 6: formulario de login	42
2.14 Uso de variables	45
2.15 Incremento de variables numéricas	47
2.16 Actividad de aplicación 7: animación por AS	49
2.17 Variables vacías	50
2.18 Comparación de variables	50
2.19 Actividad de aplicación 8: el número escondido	51
2.20 Conversión de variables	53
2.21 El alcance de una variable	55
2.22 Actividad de aplicación 9: el molinillo de viento	62
3. Condicionales	67
3.1 ¿Qué son los condicionales?	68
3.2 Operadores de comparación	69
3.3 El operador lógico not (!)	71
3.4 Condicionales if anidados	71
3.5 El operador and (&&)	71
3.6 El operador or ()	72
3.7 El condicional ?	73
3.8 La sentencia else	73
3.9 La sentencia else if	74
3.10 La sentencia switch	74
3.11 Actividad de aplicación 10: conducir un coche	76

4. Bucles	80
4.1 El bucle while	81
4.2 El bucle do while	82
4.3 El bucle for	83
4.4 Bucle con el evento onEnterFrame	84
4.5 Bucle con la línea de tiempo	84
4.6 Activ. de aplicación 11: el acuario	85
5. Arrays	90
5.1 ¿Qué es un array?	91
5.2 Estructura de un array	91
5.3 Declaración de un array	91
5.4 Recuperando información de un array	93
5.5 Añadir elementos a un array	94
5.6 Activ. de aplicación 12: juego de memoria musical	99
5.7 Arrays asociativos	108
5.8 Eliminación de elementos de un array	109
5.9 Modificar elementos de un array	112
5.10 Arrays anidados	112
5.11 Otros métodos de array	114
5.12 Ordenación de arrays	115
5.13 Activ. de aplicación 13: grabación del ratón	119
5.14 Activ. de aplicación 14: mastermind	121
6. Funciones	128
6.1 ¿Qué es una función?	129
6.2 Parámetros de una función	129
6.3 Funciones que devuelven valores	131
6.4 Anidamiento de funciones	132
6.5 Alcance de una función	133
6.6 Variables y funciones	137
6.7 La clase arguments	138
6.8 Activ. de aplicación 15: calculadora	139
7. Objetos y clips de película	143
7.1 Programación orientada a objetos	144
7.2 Objetos personalizados	146
7.3 La sentencia with	147
7.4 La sentencia for ... in	149
7.5 Referencia a objetos	150
7.6 Activ. de aplicación 16: el cuestionario	152
7.7 ¿Qué es un clip de película?	159
7.8 Crear clips de película	160
7.9 Propiedades de un clip de película	166
7.10 Activ. de aplicación 17: el vuelo de las moscas	167
7.11 Eliminación de clips de película	172
7.12 Profundidad y orden de apilamiento	172
7.13 Modificar el color de un clip de película	175
7.14 Filtros	177
7.15 Máscaras	170
7.16 Activ. de aplicación 18: memory card	180
7.17 Dibujar con ActionScript	185
7.18 Activ. de aplicación 19: dibujo de formas	189

8. Interactividad	194
8.1 Captura de eventos	195
8.2 Eventos de botón	200
8.3 Activ. de aplicación 20: el juego de las diferencias	201
8.4 Eventos de Clip de Película	206
8.5 Activ. de aplicación 21: el juego del conejo.....	210
8.6 Eventos de Mouse.....	213
8.7 Eventos de Teclado	215
8.8 Activ. de aplicación 22: adivina la frase.....	217
8.9 Eventos de Cuadro de Texto	223
8.10 Puntero de ratón personalizado	224
8.11 Posición del puntero del ratón.....	226
8.12 Activ. de aplicación 23: conducir con teclado	227
8.13 Activ. de aplicación 24: control de audio	229
8.14 Detección de colisiones	231
8.15 Activ. de aplicación 25: laberinto	233
9. Componentes	236
9.1 ¿Qué es un componente?	237
9.2 Activ. de aplicación 26: conversor de unidades	237
9.3 Personalizar el aspecto de un componente	248
9.4 Interacciones de aprendizaje.....	250
9.5 Activ. de aplicación 27: arrastrar y colocar.....	251
9.6 Activ. de aplicación 28: cuestionario	258
9.7 Plantilla de cuestionario	264
10. Cargadores	268
10.1 Carga de imágenes externas	269
10.2 El componente Loader	272
10.3 Carga de archivos SWF	273
10.4 Cargador de una película	277
10.5 Crear un cargador visual	280
10.6 El componentes ProgressBar	283
10.7 Activ. de aplicación 29: galería de fotos.....	285
10.8 Activ. de aplicación 30: visor de películas	288

Presentación

Adobe Flash se ha convertido con el tiempo en una herramienta muy utilizada para el diseño de espacios web. Algunas de las razones que justifican este éxito son:

- **Economía de tamaño.** El uso de gráficos vectoriales y la arquitectura símbolo-instancia producen importantes desarrollos gráficos en archivos SWF de peso reducido. Flash es un programa que está especialmente pensado para crear contenidos en internet. La reducción de tamaño de los archivos a descargar facilita la navegación del usuario.
- **Streaming.** Se puede comenzar a visualizar una animación sin que se haya descargado por completo. La reproducción se inicia y la descarga se completa en segundo plano sin que el usuario perciba este proceso.
- **Caché.** Cuando una película Flash se descarga, queda almacenada en la caché del navegador, no siendo necesario repetir la operación en posteriores accesos.
- **Compatibilidad.** La reproducción de un objeto Flash se realiza gracias a un plugin o extensión que se instala en el navegador. Esto asegura que se accederá a él y se visualizará de igual forma en cualquier sistema operativo, plataforma o dispositivo.
- **Impacto multimedia.** Con Flash se puede implementar audio y movimiento de objetos de una forma fácil y con una gran calidad final. Por este motivo Flash se ha convertido en el estándar preferido por diseñadores para la creación de espacios web de alto impacto visual y sonoro.
- **Interactividad.** Gracias a Flash se pueden programar aplicaciones donde el usuario interactúa usando el ratón y el teclado: introducir datos a través del teclado, imprimir información, ir a distintos puntos, arrastrar y colocar objetos, etc. Por ello Flash es una herramienta muy utilizada por muchos programadores para la creación de aplicaciones funcionales en la web.

Los desarrollos multimedia e interactivos más complejos con Flash se consiguen utilizando un lenguaje de script propio que incorpora esta herramienta. Se llama **ActionScript**. Se pueden crear animaciones Flash que no utilizan código pero su empleo permite múltiples opciones:

- **Interacción con el usuario.** Permite programar qué acciones tendrán lugar cuando el usuario hace clic en un botón, pulsa una tecla, arrastra un objeto sobre otro, modifica el contenido de un cuadro de texto, etc.
- **Eventos.** También es posible planificar comportamientos de la aplicación en función de ciertos eventos: cada cierto tiempo, en determinado momento o fecha, etc.
- **Control de los clips de película.** Se puede controlar mediante código y en tiempo de ejecución las propiedades de cualquier clip de película sobre el escenario. De esta forma el propio usuario puede tener el control sobre estos objetos durante la ejecución de la animación.
- **Animación mediante código.** Gracias al ActionScript se pueden animar los clips de película con mayor precisión y sin utilizar la línea de tiempo. Esta técnica reduce de una forma considerable el tamaño final del archivo SWF.
- **Transferencia de datos.** Una aplicación Flash puede recibir o enviar datos de otras tecnologías: XML, WebServices, PHP, ASP, etc.

El curso "ActionScript 2" propone una iniciación al conocimiento y uso de este lenguaje para la creación de contenidos multimedia interactivos destinados a la web. Como en el aprendizaje de otros lenguajes de programación se comienza exponiendo los detalles de la sintaxis específica relacionada con la declaración de variables, arrays, condicionales, bucles, etc. Sin embargo siguiendo un enfoque práctico del curso y aprovechando la dimensión multimedia de la herramienta se incluyen ejemplos de aplicaciones completas.

Para el seguimiento de este curso se recomienda utilizar **Adobe Flash 8, CS 3 Pro** o superior. Estas versiones funcionan en un equipo Windows XP SP2, Windows Vista o Mac.

1. Introducción

1.1 ¿Qué es ActionScript?

ActionScript es el lenguaje de programación de Flash. Está basado en el modelo ECMA para lenguajes de script (ECMA=European Computer Manufacturers' Association). Java o Javascript también se basan en este modelo.

ActionScript aparece en las primeras versiones de Flash pero no adquiere importancia hasta la versión Flash 5. Con Flash MX se convierte en un auténtico lenguaje de programación orientado a objetos. Flash MX 2004 y Flash 8 introducen ActionScript 2 con más objetos, una sintaxis más robusta y un modelo de construcción de clases más potente. Con Flash CS3 Professional se presenta ActionScript 3, una versión orientada al mundo profesional con un desarrollo más amplio y complejo orientado a objetos y clases.

En este curso de iniciación se propone el estudio de la versión ActionScript 2 que se desarrollará utilizando tanto Flash 8 como Flash CS3 Professional.

1.2 ¿Para qué ActionScript?

Cuando surgió Flash se utilizaba sobre todo como una herramienta de dibujo y de animación. Sin embargo, a medida que la comunidad de diseñadores de páginas web fueron utilizándola para crear sitios web, Macromedia fue poniendo en el mercado versiones donde el ActionScript era cada vez más potente. Por ello Flash pasó de ser un programa de animación a convertirse en una potente herramienta para el desarrollo de aplicaciones completas y complejas.

Al visitar un sitio web no sólo esperamos animaciones multimedia impactantes (con una excelente calidad de imagen o de audio) sino también una importante dosis de interactividad. La interacción del usuario se puede lograr con otras tecnologías como Java, Javascript, etc. Sin embargo Flash permite el diseño de contenidos interactivos con mayor facilidad y elegancia. Por otra parte el contenido Flash se ha convertido en un estándar al poder utilizar el reproductor en todo tipo de navegadores web, sistemas operativos y dispositivos. Todo ello ha producido la rápida expansión de Flash entre la comunidad de desarrolladores de aplicaciones y sitios web.

Actualmente, si se pretende desarrollar un sitio web en Flash, el diseño de dibujos y animaciones sobre la línea de tiempo no es suficiente. El uso de ActionScript abre un mundo de posibilidades:

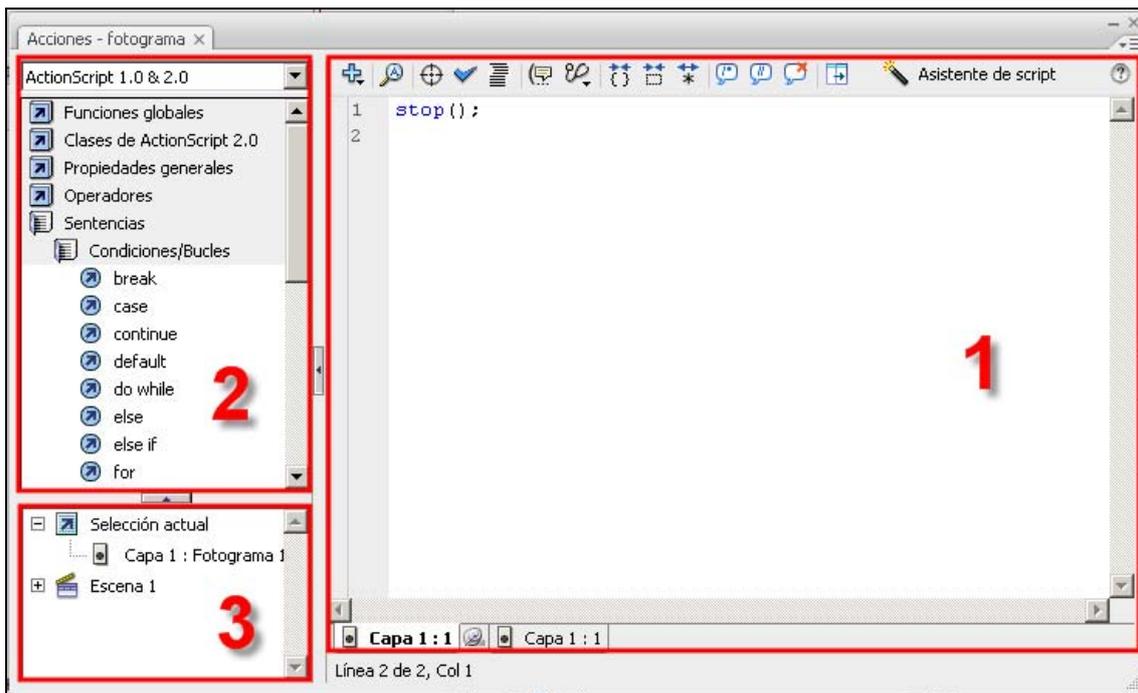
- Proporcionar una experiencia de usuario personalizada: programar qué acciones tendrán lugar si el usuario hace clic en un botón, modifica el contenido de un cuadro de texto, pulsa en una tecla, arrastra un objeto sobre otro, etc. Por otra parte permite la programación de eventos temporales, la configuración del interfaz de acuerdo con la elección del usuario, etc.
- Disponer de un mayor control sobre los clips de película y sus propiedades en tiempo de ejecución de tal forma que el propio usuario puede decidir qué parámetros le interesa modificar.
- Animar objetos sobre el escenario mediante programación, es decir, sin utilizar la línea de tiempo. Esto permite crear animaciones más complejas y sobre todo conseguir un tamaño de archivo más reducido.
- Manejar la información que se envía o recibe desde una película Flash: formularios, carga de contenidos, bases de datos, comunicación con otras tecnologías (XML, WebServices, PHP, ASP, etc).
- Controlar de forma dinámica y en tiempo de ejecución el volumen o balance de la reproducción del audio, vídeo, etc.
- Etc, etc, etc.

El uso de ActionScript no sólo permite mejorar la experiencia web del usuario sino que también facilita la creación de aplicaciones autoejecutables o miniprogramas que funcionan en un

equipo con independencia del navegador web. Esta posibilidad es utilizada cada vez por más programadores para crear juegos, aplicaciones de aprendizaje, etc. Para hacer todo esto, es necesario tener ciertos conocimientos de ActionScript.

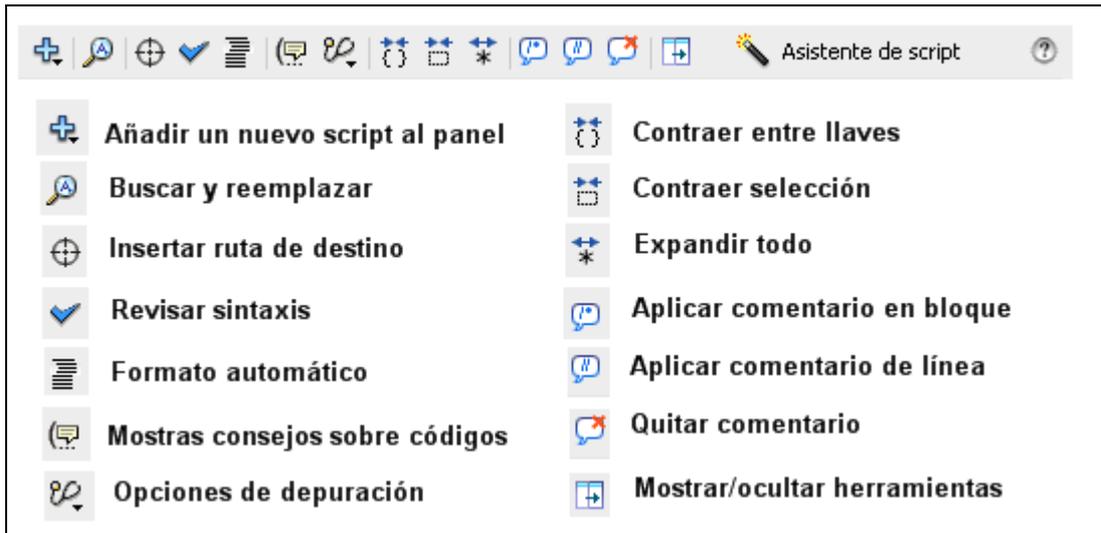
1.3 Panel de Acciones

El código ActionScript se introduce en una película Flash a través del panel **Acciones**. Para acceder a este panel selecciona **Ventana > Acciones**. Otra posibilidad más rápida es pulsar la tecla de función <F9>



Este panel consta de 3 áreas:

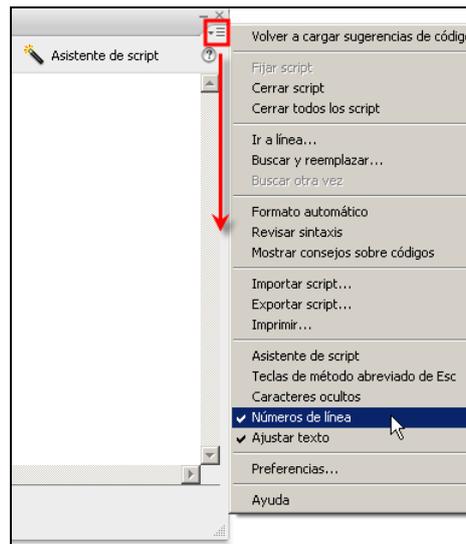
1. **Panel de scripts.** Es donde se teclea el código.
2. **Caja de herramientas de acciones.** Contiene toda la sintaxis de ActionScript 2 organizada en categorías y alfabéticamente. Para insertar un elemento en el panel de scripts, búscalo en esta caja de herramientas y haz doble clic sobre él o bien arrástralo al panel de Scripts.
3. **Navegador de scripts.** Es un pequeño explorador que permite manejar todas las acciones de la película. Al hacer clic sobre un elemento de este navegador, el script asociado se mostrará en el panel de scripts y la cabeza lectora se desplazará a esa posición de la línea de tiempo.



En el **Panel de scripts** se sitúan algunos botones para realizar las tareas más habituales cuando se trabaja con actionscript:

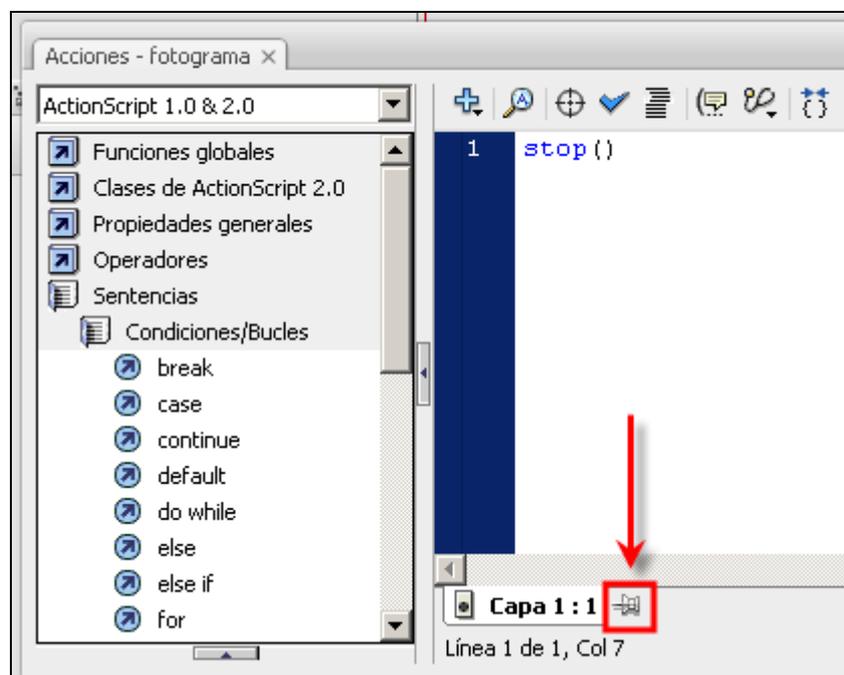
- **Añadir un nuevo script al panel.** Este botón, cuando se pulsa, muestra un menú desplegable con las mismas categorías que la caja de herramientas de acciones. Cuando se localiza una acción, se pulsa sobre ella y pasa al panel de scripts.
- **Buscar y reemplazar.** Se utiliza para localizar y sustituir fragmentos de código en el panel de scripts.
- **Insertar ruta de destino.** Abre el cuadro de diálogo **Insertar ruta de destino** permitiendo que desde AS se pueda apuntar a objetos individuales del escenario.
- **Revisar sintaxis.** Comprueba rápidamente los errores de sintaxis existentes en el script actual. Estos errores se muestran en el panel **Salida**.
- **Formato automático.** Aplica formato al código logrando una sintaxis correcta y aumentando la legibilidad.
- **Mostrar consejos sobre códigos.**
- **Opciones de depuración.** Se utiliza para depurar el código permitiendo fijar y quitar puntos de parada donde al depurar se puede avanzar línea a línea.
- **Contraer entre llaves.** Contrae el código que aparece entre llaves o paréntesis donde se encuentra el punto de inserción.
- **Contraer selección.** Contrae el bloque de código seleccionado actualmente.
- **Expandir todo.** Expande el código contraído en el script actual.
- **Aplicar comentario en bloque.** Añade marcadores de comentario al principio y al final del bloque de código seleccionado.
- **Aplicar comentario de línea.** Añade un marcador de comentario de la línea actual o de todas las líneas del bloque de código seleccionado.
- **Quitar comentario.** Elimina todas las marcas de comentario de la línea actual o de todas las líneas del bloque de código seleccionado.
- **Mostrar/ocultar caja de herramientas.**
- **Asistente de scripts.** Muestra un panel que pretende facilitar la introducción de los elementos necesarios para crear scripts. No se aconseja su utilización en este curso.
- **Ayuda.** Muestra la información de referencia para el elemento de ActionScript seleccionado en el panel de scripts.

En la esquina superior derecha del panel de **Acciones** se encuentra también el botón de acceso al **Menú Panel** desde donde se puede acceder a distintas tareas relacionadas con el panel de Acciones: **Cerrar script**, **Cerrar todos los scripts**, **Ir a línea**, **mostrar/ocultar Números de línea**, **Imprimir**, etc.



Otra utilidad interesante del panel de Acciones es la posibilidad de fijar varios paneles de script. Esto es especialmente útil cuando se desea acceder simultáneamente a varios fotogramas u objetos que contienen ActionScript. Tras fijar las distintas ventanas de los scripts correspondientes se puede mover entre ellas simplemente pulsando en la correspondiente pestaña. Para fijar una ventana de script sigue los siguientes pasos

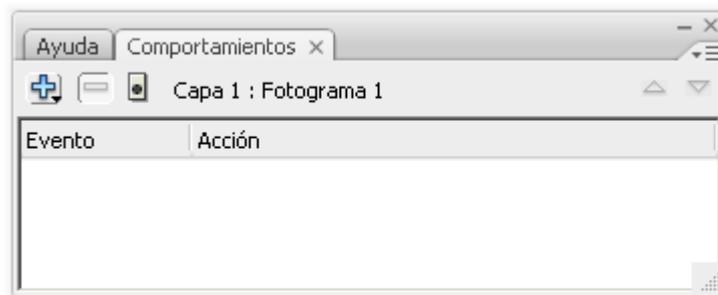
1. Selecciona el fotograma u objeto que contiene ActionScript para que el código se muestre en el panel **Acciones**.
2. Pulsa sobre el icono de un alfiler que aparece en la esquina inferior izquierda del panel acciones o bien despliega el menú de opciones de panel y selecciona **Fijar script**.
3. Para acceder a la acciones de un fotograma u objeto específico haz clic sobre la correspondiente pestaña.



1.4 Panel de comportamientos

Un comportamiento es un fragmento de código AS que se puede asociar a un botón, clip de película o fotograma. Cuando se selecciona un comportamiento se muestra un interface donde se pueden definir distintos parámetros de ese comportamiento bien por selección o por teclado directo. Flash ofrece un cierto número de comportamientos predefinidos. Al asociar un comportamiento a un objeto realmente lo que estamos haciendo es creando código AS que se genera automáticamente tomando los valores que hemos definido en los parámetros del comportamiento y que se asocia a ese objeto.

Para abrir el panel **Comportamientos** selecciona: **Ventana > Comportamientos** o bien pulsa la combinación de teclas: **<Mayús>+<F3>**



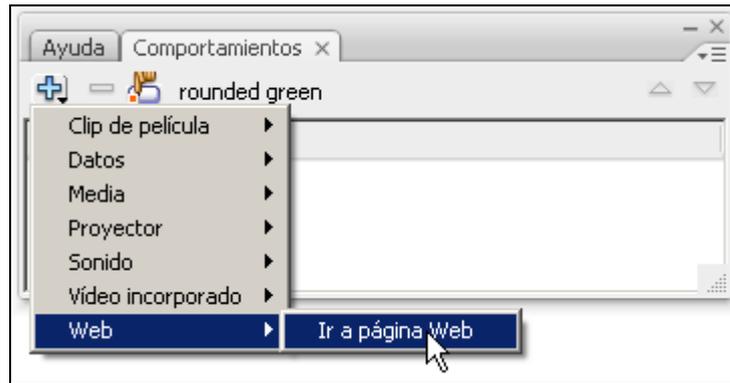
1.5 Actividad de aplicación 1: Botón Google

En la siguiente actividad vamos a utilizar comportamientos para crear un botón interactivo que abra la ventana del navegador y muestre la página de Google.

1. Descarga y descomprime a una carpeta de tu equipo el archivo **botonGoogle.zip**. En su interior se encuentra el archivo **botonGoogle fla**.
2. Abre Flash y selecciona **Archivo > Abrir** para abrir el archivo FLA anterior.
3. Elige **Ventana > Biblioteca**
4. Arrastra el botón **rounded green** desde la Biblioteca hasta el Escenario.



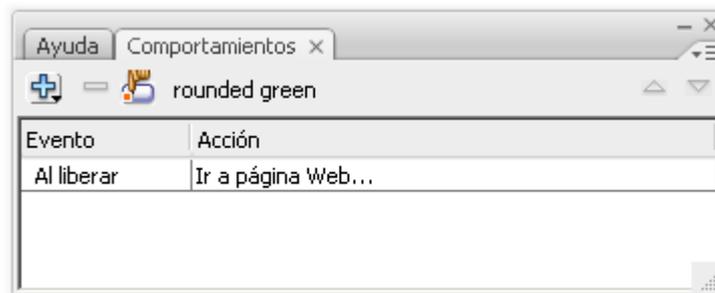
5. Haz clic sobre el botón situado en el escenario para seleccionarlo y a continuación abre el panel de Comportamientos mediante **Ventana > Comportamientos**.
6. Asegúrate de que en este panel aparece referenciado la instancia del botón "rounded green". En este panel pulsa en el botón "+" y elige **Web > Ir a página Web**.



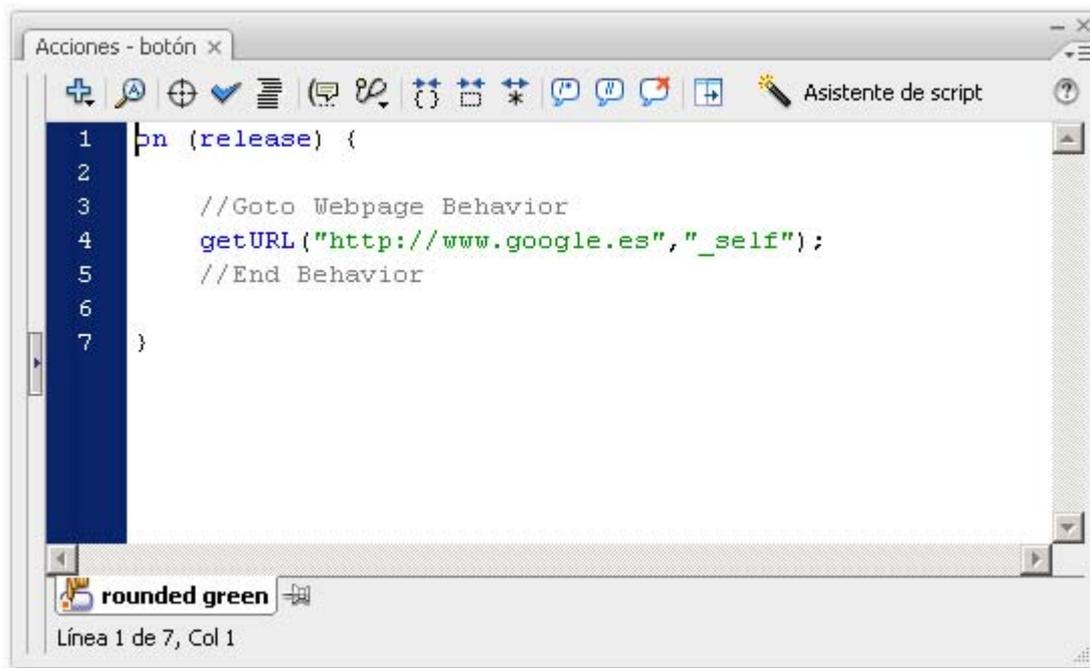
7. En el cuadro de diálogo **Ir a URL** introduce la URL: <http://www.google.es> y elige el marco de destino, por ejemplo, `_self` (el mismo marco). Pulsa en el botón **Aceptar**.



8. Ahora el panel de Comportamientos mostrará el comportamiento que se ha situado en el botón. Observa que en la columna **Evento** se muestra **Al liberar**. Esto indica que se irá a esa dirección cuando el usuario suelte el botón izquierdo del ratón después de haber hecho clic sobre este botón. Si despliegas el evento puedes elegir otro distinto. Se estudiarán con detalle más adelante.



9. Si seleccionas el botón sobre el escenario y pulsas **<F9>**, se mostrará el panel **Acciones-Botón** donde podrás leer el código que se ha generado de forma automática tomando los datos que has introducido al crear el comportamiento.



10. Para probar el funcionamiento de esta película elige **Control > Probar película** o bien pulsa la combinación de teclas **<Ctrl>+<Enter>**.

Los comportamientos son un método fácil y rápido de añadir interactividad a un archivo Flash.

1.6 Acciones en fotogramas y en objetos

Flash permite asignar acciones a fotogramas y también a objetos como botones o clips de película.

1.6.1 Acciones - Fotograma

Si asignamos código AS a un fotograma, éste se ejecutará cuando la cabeza lectora alcance ese fotograma durante la reproducción de la película Flash.

Ejemplo:

1. Abre un nuevo documento de Flash
2. Clic en el primer fotograma de la **Capa 1** y pulsa **<F9>** para visualizar el panel **Acciones-Fotograma**.
3. En este panel escribe el siguiente código:

```
getURL("http://www.google.es");
```

4. Elige **Control > Probar película**. Al reproducir la película, la cabeza lectora leerá la orden situada en el primer fotograma y automáticamente abrirá una ventana del navegador web y mostrará la página de Google.

1.6.2 Acciones – Botón

Flash no sólo permite vincular código a los fotogramas sino también a objetos situados en el escenario como botones o clips de película. Cuando asociamos código a un botón o clip de película siempre tenemos que indicar el evento que deberá ocurrir para que se ejecute el mencionado código.

Ejemplo:

1. Abre un nuevo documento de Flash.
2. Selecciona **Ventana > Bibliotecas Comunes > Botones**. En la ventana que se muestra abre una carpeta y arrastra hasta el escenario un botón.
3. Sobre el escenario haz clic en el botón que has colocado y a continuación pulsa en **<F9>** para visualizar el panel **Acciones – Botón**.
4. En este panel escribe el siguiente código:

```
on(release){
  getURL("http://www.google.es")
}
```

5. Elige **Control > Probar película**. En este caso se abrirá la ventana del navegador web y se mostrará la página Google cuando el usuario libere el botón izquierdo del ratón después de haberlo pulsando sobre el botón.

1.6.3 Acciones – Fotograma versus Acciones - Objetos

Como hemos visto Flash permite asignar acciones tanto a fotogramas como a objetos. Sin embargo con ActionScript 2 se recomienda que la asignación de acciones a un objeto no se realice directamente sobre el objeto sino sobre el fotograma que contiene a ese objeto. El principal motivo que justifica esta recomendación es facilitar la búsqueda y edición del código.

Ejemplo:

1. Abre un nuevo documento de Flash.
2. Selecciona **Ventana > Bibliotecas Comunes > Botones**. En la ventana que se muestra abre una carpeta y arrastra hasta el escenario un botón.
3. Sobre el escenario haz clic en el botón y en el **Inspector de propiedades** introduce un nombre de instancia: **google_btn**
4. A continuación haz clic en el fotograma 1 y pulsa en **<F9>** para visualizar el panel **Acciones – Fotograma**.
5. En este panel escribe el siguiente código:

```
google_btn.onRelease = function() {
  getURL("http://www.google.es");
}
```

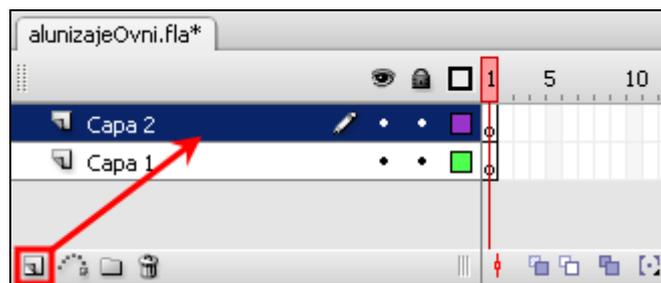
6. Elige **Control > Probar película**. En este caso también se abrirá la ventana del navegador web y se mostrará la página Google cuando el usuario libere el botón izquierdo del ratón después de haberlo pulsando sobre el botón.

1.7 Actividad de aplicación 2: El vuelo del ovni

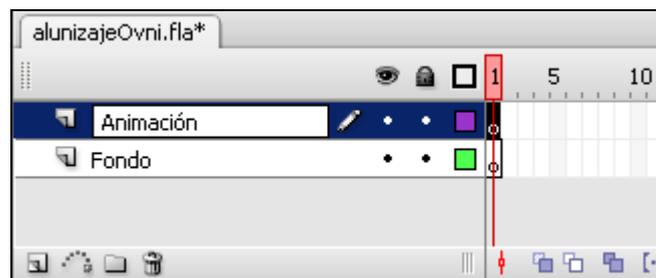
En este apartado se expone cómo crear una animación utilizando botones play/stop para controlar la reproducción.

1.7.1 Animación simple

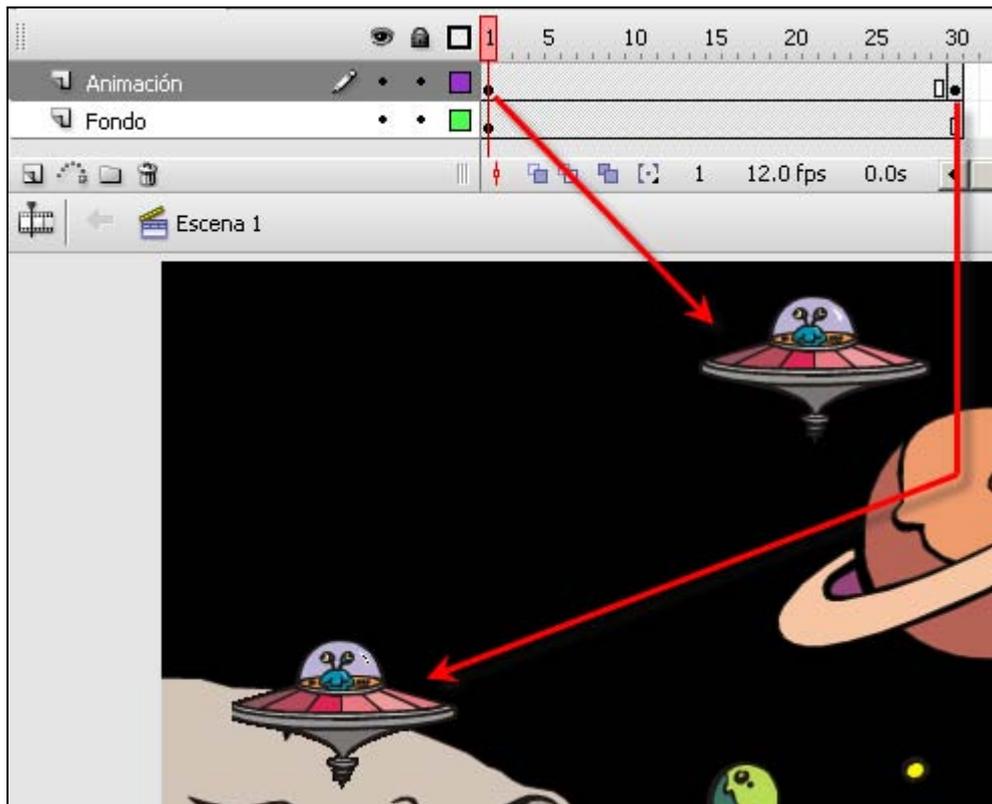
1. Descarga y descomprime a una carpeta de tu equipo el contenido del archivo **alunizajeOvni1.zip**.
2. Doble clic sobre el archivo **alunizajeOvni1 fla** para abrirlo con Flash.
3. Selecciona **Ventana > Biblioteca** para abrir la biblioteca de este documento.
4. Clic en el botón **Insertar capa** para añadir una segunda capa encima de la actual.



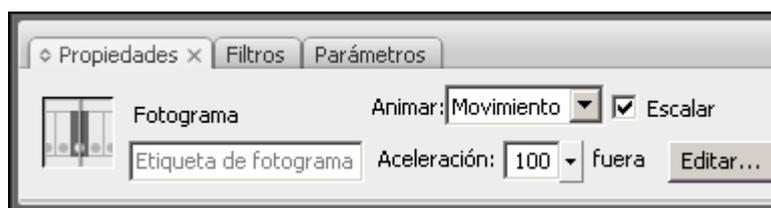
5. Doble clic sobre el nombre "Capa 1" para introducir **Fondo** como nombre de capa. También sobre "Capa 2" para introducir **Animación**.



6. Clic sobre el fotograma 1 de la capa **Fondo** y a continuación arrastra hasta el escenario una instancia del símbolo gráfico **universo** situado en la **Biblioteca**
7. Clic en el fotograma 1 de la capa **Animación** para seleccionarlo y a continuación arrastra hasta el escenario una instancia del símbolo gráfico **ovni**
8. Clic derecho sobre el fotograma 30 de la capa **Fondo** y selecciona **Insertar fotograma**. No es necesario insertar un fotograma clave puesto que el contenido no variará
9. Clic derecho sobre el fotograma 30 de la capa **Animación** y selecciona **Insertar fotograma clave**.
10. Haz clic sobre el fotograma 1 de la capa **Animación** y una vez seleccionado arrastra el ovni para situarlo en el espacio sideral. Ésta será la posición inicial de la animación. Clic en el fotograma 30 de la capa **Animación** y arrastra el ovni para situarlo sobre la luna. Ésta será la posición final de la animación.



11. Haz clic sobre el fotograma 1 de la capa **Animación** y en el **Inspector de Propiedades** (**Ventana > Propiedades > Propiedades**) despliega la lista **Animar** y elige la opción **Movimiento**. Arrastra el deslizador **Aceleración** para definir un valor de **100**. Con ello el movimiento será rápido al principio y lento al final como corresponde a un alunizaje sin complicaciones.

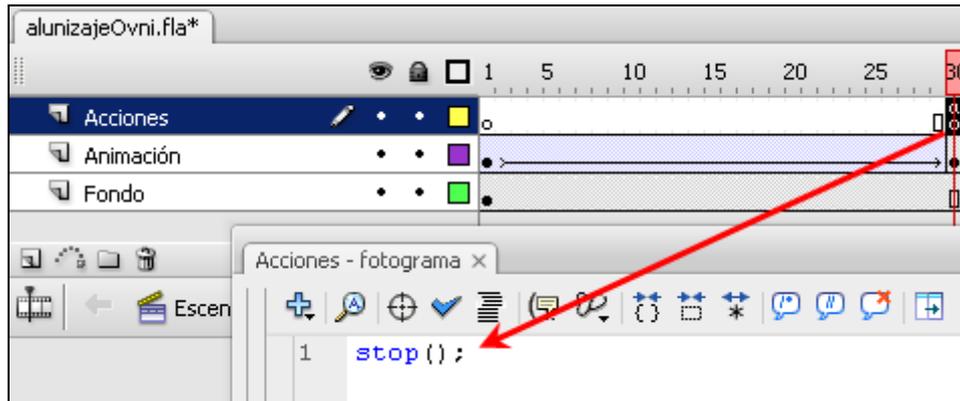


12. Para comprobar el funcionamiento de la película elige **Control > Probar película**.

1.7.2 Animación simple con parada final

Si dejamos la animación como está, cuando la cabeza lectora llega al fotograma nº30 salta al fotograma nº1 y sigue reproduciéndose en un bucle infinito. Para conseguir que la nave se detenga sobre la luna es necesario asociar una acción al fotograma final. Para ello:

- 1) Crea una nueva capa y asígnale **Acciones** como nombre.
- 2) Inserta un fotograma clave en el fotograma nº30 de esta nueva capa: **clic derecho** y selecciona **Insertar fotograma clave**.
- 3) Clic sobre este fotograma clave para seleccionarlo
- 4) Pulsa **<F9>** para mostrar el panel **Acciones – fotograma**.
- 5) En este panel escribe: `stop();`



Cuando la cabeza lectora alcance el fotograma nº30 leerá el código ActionScript asociado a ese fotograma de la capa **Acciones** y la animación se detendrá (stop());

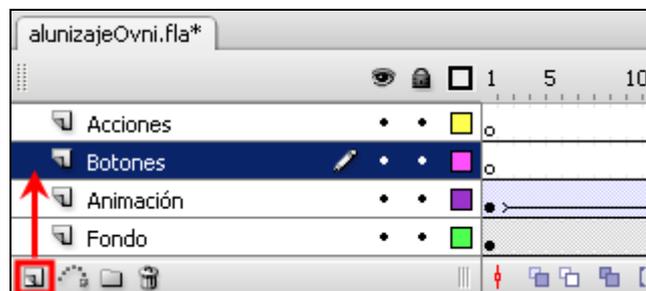
Nota:

- Es una práctica muy aconsejable y elegante crear en la parte más superior una capa independiente con el nombre **Acciones** para situar en ella el código ActionScript. De esta forma se facilitará su lectura y edición.

1.7.3 Animación con control de reproducción

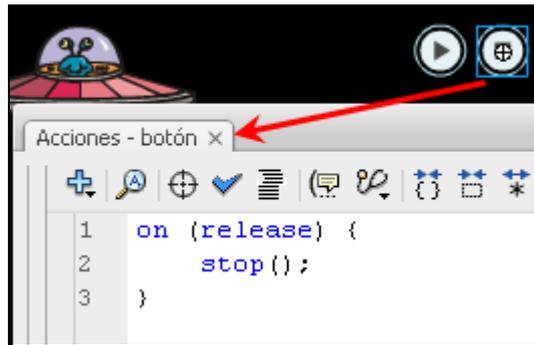
Vamos a seguir con el FLA anterior. Ahora añadiremos una capa con dos botones para controlar la reproducción.

1. Clic sobre la capa **Animación** y a continuación pulsa en el botón **Insertar capa**. De esta forma se creará una nueva capa encima de la anterior entre ésta y la capa **Acciones**. Asígnale como nombre: **Botones**.



2. Clic en el fotograma 1 de la capa **Botones** para seleccionarla.
3. Arrastra desde la Biblioteca sendas instancias de los botones **play** y **stop** para situarlos en la esquina superior derecha del escenario.
4. A continuación haz clic sobre la instancia del botón **play** situado en el escenario y pulsa **<F9>** para mostrar el panel **Acciones**.
5. Se muestra el panel **Acciones – Botón**. Escribe:

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



6. Clic sobre el botón **stop** situado en el escenario y en el panel **Acciones – Botón** introduce el siguiente código:

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

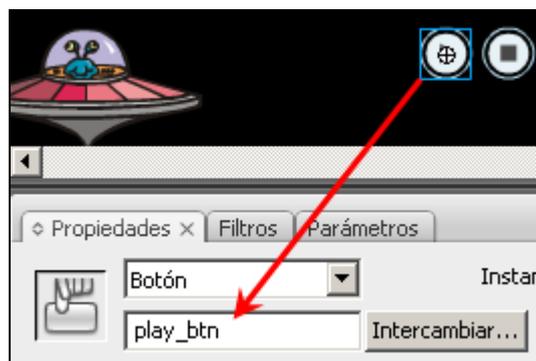
En ambos casos el evento es **release** que se produce cuando el usuario libera el botón izquierdo del ratón después de haberlo pulsado sobre este objeto del escenario.

7. Para comprobar el funcionamiento de la película selecciona **Control > Probar película**.

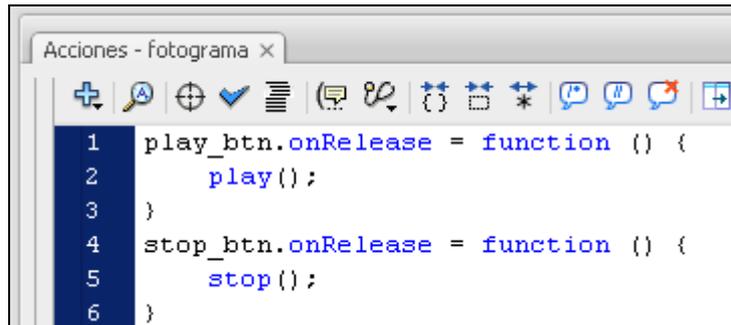
1.7.4 Animación con código en fotogramas

Vamos a retomar el FLA del apartado anterior.

1. Clic sobre el botón **play** situado en el escenario y pulsa **<F9>** para mostrar el panel **Acciones – Botón**. Borra el código que has introducido en el ejercicio anterior. Repite estos pasos para el botón **stop**.
2. Antes de asignarles el código a los botones desde un fotograma es necesario nombrar esas instancias de botón. Para ello pulsa sobre el botón **play** del escenario para seleccionarlo y asígnale **play_btn** como nombre de instancia en el **Inspector de propiedades**.



3. Selecciona el botón **stop** y asígnale **stop_btn** como nombre de instancia.
4. Clic en el fotograma 1 de la Capa **Acciones** y pulsa **<F9>** para que se muestre el panel **Acciones - Fotograma**.
5. Introduce el siguiente código:



```
play_btn.onRelease = function () {  
    play();  
}  
  
stop_btn.onRelease = function () {  
    stop();  
}
```

Observa que se utiliza una sintaxis de punto para indicar que la instancia "stop_btn", cuando ocurre el evento "onRelease" (liberar el ratón) ejecuta todas las órdenes incluidas entre las llaves. En este caso detener la reproducción de la película: stop().

A partir de este momento utilizaremos este estilo más estructurado de programación para asignar acciones a las instancias de botones y clips de películas situados en el escenario.

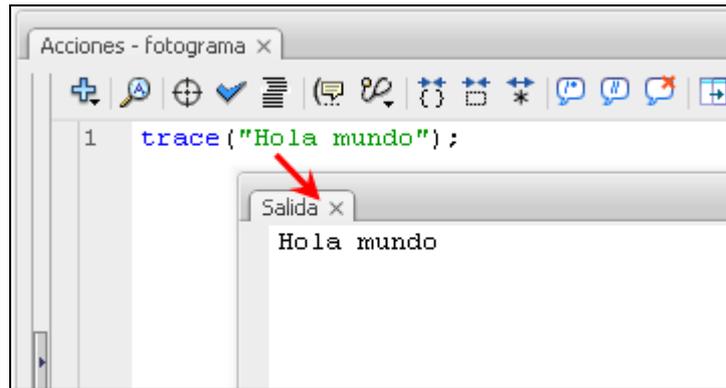
1.8 Panel de Salida

Como ayuda a la programación en ActionScript se pueden incluir líneas en el código que impriman mensajes o el valor de una variable en el panel de salida.

1. Abre un documento nuevo en Flash
2. Clic sobre el primer fotograma de la línea de tiempo.
3. Pulsa la tecla <F9> para visualizar el panel **Acciones – Fotograma**
4. En esta ventana escribe:

```
trace("Hola mundo");
```

5. A continuación selecciona **Control > Probar película.**
6. El mensaje "Hola mundo" se muestra en el panel de salida



Nota:

- Este mensaje sólo se muestra cuando ejecutamos la película desde el entorno de edición de Flash. Nunca se mostrará si se visualiza a través del navegador web o del reproductor de Flash.

1.9 Programación Orientada a Objetos

ActionScript es un lenguaje de Programación Orientado a Objetos (POO). Esto significa que se programa en base a objetos.

1.9.1 ¿Qué es un objeto?

En programación, un objeto es un fragmento de código autónomo que contiene 3 elementos importantes: propiedades, métodos y eventos.

1.9.2 Propiedad de un objeto

Una propiedad de un objeto es información del mismo. Básicamente cada propiedad de un objeto soporta la información específica de una instancia de ese objeto. Algunas propiedades se pueden leer y modificar en tiempo de ejecución pero otras son sólo de lectura.

Ejemplo de propiedades en objetos integrados de Flash:

movieClip._y

La posición vertical de un objeto movie clip. Se puede leer y escribir.

textField.text

Una propiedad de lectura/escritura para obtener y situar el texto en un objeto cuadro de texto.

Math.PI

Una propiedad constante del objeto Math. Es de sólo lectura y su valor es: 3.14159265358979

1.9.3 Métodos de un objeto

Mientras que las propiedades son los **adjetivos** de los objetos, los métodos son los **verbos**. El método hace que algo ocurra. Es una función que se suele asociar a un objeto específico.

Existen varios métodos predefinidos en objetos propios de Flash:

movieClip.stop()

Este método detiene la reproducción en la línea de tiempo propia de ese clip de película.

Math.random()

Este método devuelve un número aleatorio entre 0 y 1.

Un objeto puede tener cientos de métodos.

Se puede utilizar parámetros en las llamadas a un método. Con estos parámetros se pasa información al método.

Ejemplos:

movieClip.gotoAndStop(2)

situará la cabeza lectora en el fotograma 2 de este clip de película y se detendrá.

movieClip.gotoAndStop("final")

situará la cabeza lectora en el fotograma etiquetado como "final" dentro de este clip de película y se detendrá.

1.9.4 Eventos de un objeto

Las propiedades son informaciones sobre objetos. Los métodos hacen algo con los objetos. Los **eventos** comunican a Flash que se ha producido algo con el objeto.

Los eventos suele tener asociado código de programación que se ejecuta cuando se disparan. Esto facilita la ejecución de módulos de código en determinados puntos del programa.

A continuación se citan algunos eventos predefinidos de objetos Flash:

movieClip.onEnterFrame

Este evento se caracteriza por dispararse a una velocidad similar a la velocidad de fotogramas con que se reproduce el clip de película (movieClip), es decir, si la velocidad en fotogramas por segundo es 12, este evento ocurre aproximadamente 12 veces por segundo. Esto significa que si escribimos código dentro de este evento, este código se ejecutará 12 veces en cada segundo.

textField.onChanged

Este evento se dispara cuando un usuario cambia el contenido de un cuadro de texto específico (textField). En este caso puede resultar útil comprobar el contenido del cuadro de texto y mostrar una ventana de aviso si se introducen caracteres no permitidos, por ejemplo: signos de puntuación.

Button.onRelease

Ocurre cuando un usuario presiona el botón izquierdo del ratón sobre el objeto Button (botón).

XML.onLoad

Este evento se dispara cuando un documento XML haya sido cargado por completo dentro de una película Flash.