

## Guia d'aprenentatge del Flash CS5

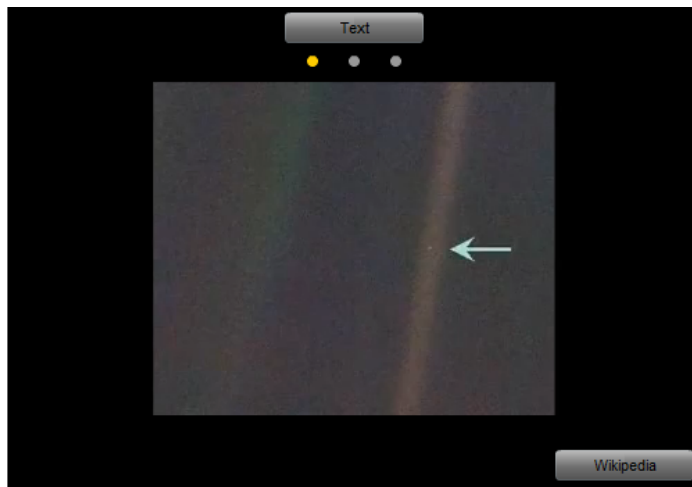
### Tutorial 12 - Vídeo i TLF

#### Pas 1 de 18

En aquest tutorial experimentarem amb les noves possibilitats que ofereix el treball amb vídeo en Flash i el nou motor de text anomenat Text Layout Framework.

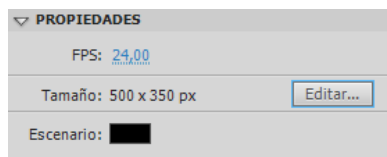
Continuant amb el tutorial anterior, utilitzarem components per crear botons i aprendrem a personalitzar-los.

Finalment, descobrirem com el nou panell *Fragmentos de código* pot facilitar-nos la programació de la nostra pel·lícula o aplicació.



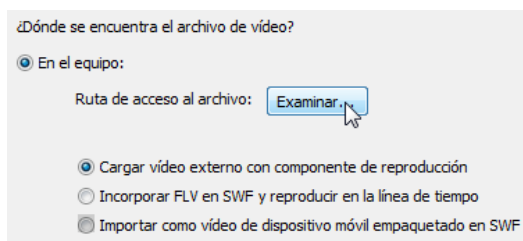
#### Pas 2 de 18

Per començar crearem un nou arxiu Flash amb fons negre i dimensions 500 x 350 px.



Començarem important el vídeo que utilitzarem en aquesta pel·lícula. El vídeo, així com altres arxius que necessitem per a la realització d'aquest tutorial, el trobarem en l'arxiu [recursosTutorial12.zip](#).

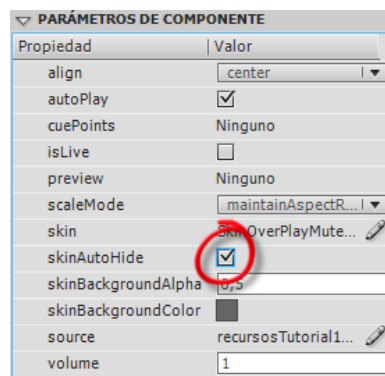
Seleccionem **Archivo > Importar > Importar vídeo**, i en la pantalla d'importació seleccionem **En el equipo**. Busquem la **Ruta de acceso al archivo** i seleccionem **Cargar vídeo externo con componente de reproducción**.



En la següent pantalla seleccionarem l'aspecte del component de vídeo que utilitzarem. Els noms dels skins ens indiquen si es mostraran a sota o a sobre el vídeo, i quins botons apareixeran.

En aquest cas utilitzarem **SkinOverPlayMute.swf**. Seleccionem com a color un gris amb una transparència del 50%.

Una vegada tenim el vídeo en l'escenari i seleccionat, marquem la casella **skinAutoHide** perquè el component s'amagui quan no tinguem el cursor sobre el vídeo.



Com podem veure, des d'aquí podríem també modificar l'skin triat, la ruta del vídeo, si es reproduceix o no automàticament, etc.

## Pas 3 de 18

Mantenint el vídeo seleccionat en l'escenari, veurem que en l'inspector de *Propiedades* també apareix una àrea per marcar **Puntos de referencia** en el nostre vídeo.

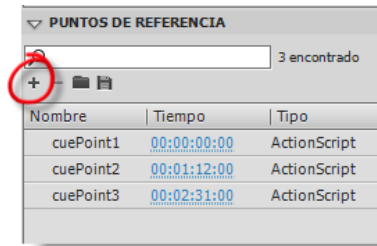
La utilització de punts de referència en un vídeo pot servir per navegar entre diferents parts d'ell, o bé perquè passi alguna cosa quan el vídeo arribi a un determinat punt. En aquest tutorial utilitzarem totes dues possibilitats.

Per afegir un punt de referència en un vídeo, podem reproduir-lo des de l'escenari i fer clic sobre el símbol + quan arribi el moment de reproducció que volem marcar. D'aquesta manera s'afegirà automàticament un punt de referència amb el codi de temps del moment en què hem premut.

Una altra manera d'afegir un punt de referència, la que utilitzarem en aquest cas, és fer clic directament sobre el símbol + i afegir manualment els codis de temps.

Afegim 3 punts de referència, amb els codis de temps 00:00:00:00 (l'inici del vídeo), 00:01:12:00 i 00:02:31:00.

Canviem els seus noms a *cuePoint1*, *cuePoint2* i *cuePoint3*, que són els noms que utilitzarem en el nostre codi.

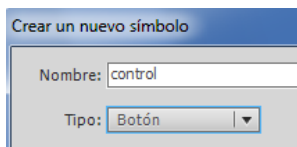


Donem al nostre vídeo el nom d'instància *sagan\_video*.

## Pas 4 de 18

Crearem tres petits botons per navegar entre els diferents punts de referència del vídeo.

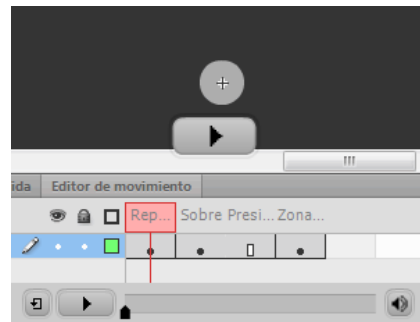
Seleccionem **Insertar > Nuevo símbolo**. Li donem el nom *control* i com a tipus seleccionem **Botón**.



En el fotograma *Reposo* dibuixem un cercle de 8 x 8 píxels, sense traç i amb farciment gris, i el centrem en el seu escenari.

Inserim un fotograma clau (**F6**) en el fotograma *Sobre*, i canviem el seu color a blanc.

Com el botó serà molt petit, podem augmentar l'àrea de clic creant un altre fotograma clau en el fotograma *Zona activa* i donant al cercle les dimensions de 20 x 20 píxels. No oblidem centrar aquest últim cercle en l'escenari.



## Pas 5 de 18

Situarem tres instàncies del botó *control*, en una nova capa que anomenarem *controls*. Situem els botons en les coordenades *x:220*, *x:250* i *x:280* i totes elles en *y:40*.

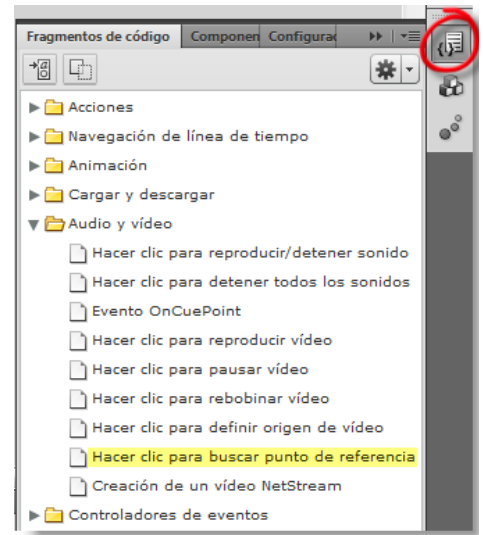
Donem als botons els noms d'instància *control1\_btn*, *control2\_btn* i *control3\_btn*.

Ens ajudarem del nou panell **Fragmentos de código** per programar el funcionament d'aquests botons. Aquest nou panell pot facilitar la programació d'algunes de les tasques més comunes de les pel·lícules de Flash. També podem crear i desar fragments de codi personalitzat.

Seleccionem el primer botó en l'escenari. En el panell **Fragmentos de código** despleguem la carpeta **Audio y vídeo** i seleccionem **Hacer clic para buscar punto de referencia**, ja que volem que, al fer clic sobre el botó, el vídeo es desplaci al punt de referència corresponent.

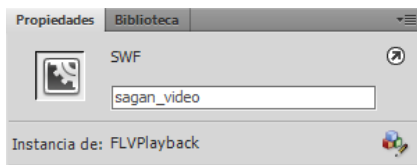
Es crearà automàticament una capa anomenada *Actions*, en la qual apareixerà un codi i les instruccions a seguir pel seu correcte funcionament.

En el següent pas modificarem aquest codi perquè s'adapti a les nostres necessitats.



## Pas 6 de 18

Les instruccions del codi que s'ha generat ens indiquen que necessitem un component *FLVPlayback*. Si seleccionem el vídeo, en l'inspector de *Propiedades* podrem comprovar que efectivament estem utilitzant aquest component.



En la primera línia de codi després dels comentaris s'ha afegit un listener al primer botó, que és el que teníem seleccionat quan vam afegir el fragment de codi.

Com necessitem afegir els mateixos listeners al altres botons, copiem aquesta línia i canviem els noms de les instàncies perquè els tres botons també cridin a la mateixa funció.

```
control1_btn.addEventListener(MouseEvent.CLICK,
fl_ClickToSeekToCuePoint);
control2_btn.addEventListener(MouseEvent.CLICK,
fl_ClickToSeekToCuePoint);
control3_btn.addEventListener(MouseEvent.CLICK,
fl_ClickToSeekToCuePoint);
```

Dins de la funció *fl\_ClickToSeekToCuePoint*, substituïm *video\_instance\_name* per *sagan\_video* en les dues línies, tal com indiquen les instruccions.

Depenent del botó premut, ens interessarà anar a un punt de referència o a un altre, així que utilitzarem el nom de la instància que ha cridat a la funció per saber el nom del punt de referència al que hem d'anar.

Hauré de substituir "Punto de referencia 1" pel nom del punt de referència corresponent, que serà "cuePoint" + `event.target.name.substr(7,1)`, ja que el caràcter 7 del nom d'instància dels botons correspon amb el nombre que hem posat en el nom del punt de referència.

Si tens dubtes sobre aquest pas pots consultar el [pas 10 del tutorial 11](#).

La funció *fl\_ClickToSeekToCuePoint* quedarà com s'indica a continuació:

```
function fl_ClickToSeekToCuePoint(event:MouseEvent):void
{
    var cuePointInstance:Object =
sagan_video.findCuePoint("cuePoint" + event.target.name.substr(7,1));
    sagan_video.seek(cuePointInstance.time);
}
```

Si ara provem la nostra pel·lícula, podem comprovar que els botons ja ens traslladen als punts de referència que havíem marcat en el vídeo.

## Pas 7 de 18

Per millorar la usabilitat podem utilitzar un marcador que ens indiqui en quin punt ens trobem. Per exemple, quan premem un botó podria veure's un petit cercle groc en comptes del botó gris, i així indicar visualment que és aquí on ens trobem.

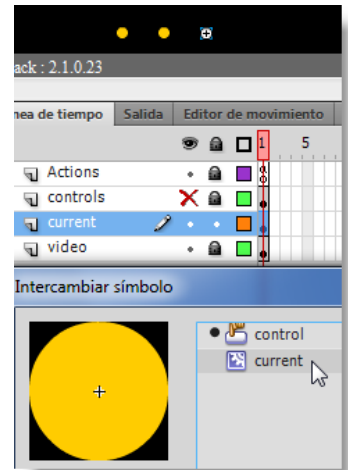
Seleccionem **Insertar > Nuevo símbolo**, l'anomenem *current* i indiquem que sigui **Clip de película**. En aquest cas no és necessari que sigui un botó, ja que no necessitem els diferents estats de la línia de temps dels botons.

En aquest clip dibuixarem un cercle groc (#FFCC00, per exemple) de 8 x 8 píxels i centrat en el seu escenari.

Creem una nova capa anomenada *current* per sota de la capa *controls*. Copiem el fotograma de la capa *controls* i peguem el fotograma en la capa *current*. D'aquesta manera tindrem els tres botons en el mateix lloc en totes dues capes.

Bloquegem i ocultem la capa *controls*. Seleccionem cadascun dels botons que hem copiat en la capa *current* amb el **botó dret** del ratolí i marquem **Intercambiar símbolo** en el menú contextual.

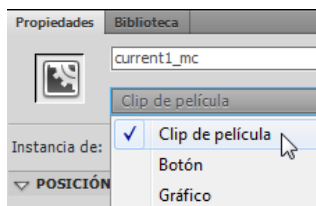
Seleccionem *current* en comptes de *control*. Així tindrem els clips en el mateix lloc en el que es trobaven els botons, ja que hem intercanviat uns símbols per uns altres.



## Pas 8 de 18

Donem els noms d'instància *current1\_mc*, *current2\_mc* i *current3\_mc* a aquests clips.

Seleccionem en l'inspector de *Propiedades* un comportament de **Clip de película**, ja que en haver aparegut en l'escenari com a substituïts d'uns botons, per defecte adoptaran el mateix tipus que el dels símbols als quals substitueixen.



Volem que quan es premi un botó aquest desaparegui, d'aquesta manera veurem el marcador groc que es troba per sota. La resta de botons hauran de romandre visibles per poder ser premuts.

Una manera de solucionar aquest problema és fer visible tots els botons (no sabem si hi haurà algun invisible per haver estat premut prèviament), i després fer invisible el botó que acabem de prémer.

Per fer visibles tots els botons crearem una funció que anomenarem *controlVisible*:

```
function controlVisible():void
{
    control1_btn.visible = true;
    control2_btn.visible = true;
    control3_btn.visible = true;
}
```

Afegim dins de la funció *f1\_ClickToSeekToCuePoint* les línies següents:

```
controlVisible();
event.target.visible=false;
```

D'aquesta manera primer farem visibles tots els botons, i després ocultarem el que hem premut.

A l'inici del vídeo el primer botó ha d'estar invisible, i per a fer-ho escriurem fora de la funció la línia següent:

```
control1_btn.visible=false;
```

## Pas 9 de 18

A causa de la gran distància temporal entre els punts de referència que hem marcat en el vídeo, seria convenient que els petits cercles grocs també poguessin ser premuts per si l'usuari volgués tornar al punt de referència corresponent.

Per a això afegim a les tres instàncies del clip *current* els mateixos listeners que tenen les instàncies de *control* i d'aquesta manera executaran la mateixa funció en ser premudes.

```
current1_mc.addEventListener(MouseEvent.CLICK,
    fl_ClickToSeekToCuePoint);
current2_mc.addEventListener(MouseEvent.CLICK,
    fl_ClickToSeekToCuePoint);
current3_mc.addEventListener(MouseEvent.CLICK,
    fl_ClickToSeekToCuePoint);
```

Dins de la funció `fl_ClickToSeekToCuePoint`, ara no ens interessa que el símbol que hagi cridat a la funció sigui invisible, ja que pot haver estat un clip *current* i no un *control*.

Per assegurar-nos que és el botó i no el clip el que es fa invisible, i aprofitant que el nombre del nom d'instància es troba en la mateixa posició en els clips *current* que en els botons *control*, substituïrem la línia:

```
event.target.visible=false;
```

per la línia:

```
this["control" + event.target.name.substr(7,1) + "_btn"].visible =
false;
```

Fins ara hem programat que, en fer clic sobre unes instàncies, el cap lector del vídeo es desplaci fins a un punt de referència. També ens interessa que, encara que no fem clic, quan el vídeo estigui reproduint-se i arribi a un punt de referència, es mostri el *current* corresponent.

## Pas 10 de 18

Com que aquesta vegada serà el mateix vídeo el que detecti l'esdeveniment d' haver arribat a un punt determinat, ho seleccionem en l'escenari, obrim el panell **Fragmentos de código** i seleccionem **Audio y vídeo > Evento onCuePoint**.

Si no modifiquem el nou codi generat i provem la nostra pel·lícula, veurem que en el panell *Salida* apareixen els noms dels nostres punts de referència quan el capçal passa per ells.

Per tant, `event.info.name` retorna el nom dels nostres punts de referència.

Canviarem la funció generada per una altra en la qual, en primer lloc, es facin visibles tots els botons. Després, amb un `switch`, analitzarem en quin punt ens trobem, i a partir d'això farem invisible un botó o un altre.

Crearem un últim punt de referència al que anomenarem *cuePoint4* en la posició *00:03:31:00*, que és la que correspon al final del vídeo. Quan el cap lector arribi a aquesta posició, deixarem marcat l'inici del vídeo. Així, si l'usuari torna a reproduir-lo trobarà marcat el primer punt.

Si tens dubtes sobre la utilització de la sentència `switch`, pots consultar el [pas 10 del tutorial 9](#).

```
function fl_CuePointHandler(event:MetadataEvent):void
{
    controlVisible();
    switch (event.info.name)
    {
        case "cuePoint1" :
            control1_btn.visible = false;
            break;
        case "cuePoint2" :
            control2_btn.visible = false;
            break;
        case "cuePoint3" :
            control3_btn.visible = false;
            break;
        case "cuePoint4" :
            control1_btn.visible = false;
            break;
        default :
            trace("Error: " + event.info.name);
    }
}
```

## Pas 11 de 18

En els passos següents crearem un clip que contingui el text amb la transcripció del vídeo, i ho repartirem en tres fotogrames. En cadascun dels fotogrames mostrarem el text corresponent a cada punt de referència. L'usuari podrà triar si veure el vídeo o llegir el text mentre escolta l'àudio del vídeo.

La transcripció del vídeo es troba en l'arxiu *Pale-Blue-Dot.txt* que podem trobar en el mateix arxiu de recursos on es trobava el vídeo.

Creem una nova capa anomenada *text*. Bloquegem y ocultem la capa que conté el vídeo per a treballar amb més comoditat en aquesta capa.

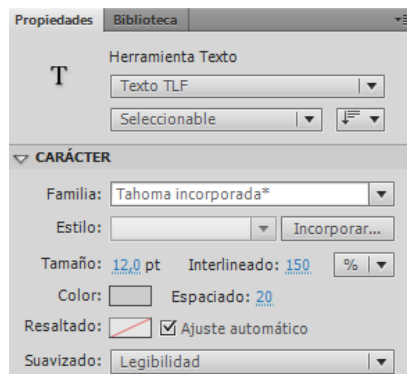
Seleccionem **Archivo > Importar > Importar a escenario** i seleccionem la imatge *Pale\_Blue\_Dot.png*. Posicionem la imatge en X:310 i Y:51.

Per al text utilitzarem el nou motor de text anomenat **Text Layout Framework (TLF)**. Utilitzarem algunes de les noves característiques d'aquest motor de text, però per aprofundir podem consultar la [informació d'Adobe sobre el TLF](#).

Seleccionem l'eina **Texto**, **Texto TLF** i **Seleccionable**.

En l'àrea **Carácter** triem la font *Tahoma Regular*, amb una mida de 12 pt, color *gris clar*, i un interlineat del 150 %.

Seleccionem com **Suavizado: Legibilidad** i incorporem la font, que podem anomenar *Tahoma incorporada*, marcant com rangs de caràcters les majúscules, minúscules i puntuació. De nou en l'inspector de **Propiedades**, seleccionem com **Familia** la font que hem incorporat.



Si dubtem sobre la incorporació de fonts podem consultar el [pas 17 del tutorial 10](#).

## Pas 12 de 18

Amb l'eina *Texto* fem clic sobre l'escenari i arrosseguem per crear un camp de text en la part esquerra de la imatge. Repetim el mateix procés per crear un altre camp de text en la part inferior de l'escenari.

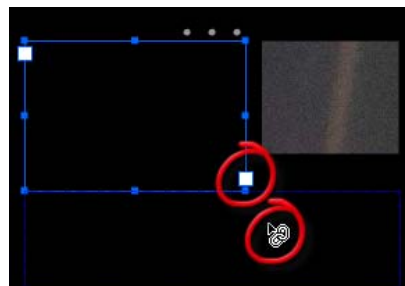
Seleccionem el primer camp de text. L'inspector de *Propiedades* mostrarà unes opcions o unes altres depenent si hem seleccionat l'eina *Texto*, un bloc de text en l'escenari, o bé si estem editant el camp de text.

Donem al primer camp els valors X:25, Y:50, AN:265 i AL:180 en l'inspector de *Propiedades*. A l'altre camp de text li donem els valors X:25, Y:230, AN:450 i AL:118.

Una de les novetats que suposa l'ús de TLF és que podem unir dos camps de text de tal manera que el text flueixi d'un camp a l'altre.

Per unir els dos blocs fem clic sobre el primer quadre, després fem clic sobre el port de sortida del bloc (un requadre que apareix en la part inferior dreta), i després fem clic sobre el bloc de text inferior.

D'aquesta manera tots dos camps de text quedaran vinculats i el text fluirà entre ells.



Copiem els dos primers paràgrafs del text del txt adjunt (els que es troben sota el número 1) Peguem el text en qualsevol dels dos blocs, i veurem com flueix d'un bloc a un altre.

En aquest cas hem utilitzat la versió en anglès, però podem utilitzar la versió en castellà o en català.

## Pas 13 de 18

Seleccionem tot el contingut de la capa *text* (la imatge i els blocs de text), i premem **F8** per convertir-ho en un símbol de tipus **Clip de pel·lícula**. Situem el punt de registre en l'extrem superior esquerre. Li donem el nom *text-img*.

Fem doble clic sobre aquest nou símbol en l'escenari per afegir els altres dos fotogrames que contindrà sense perdre la perspectiva de la seva posició i mida respecte a l'escenari.

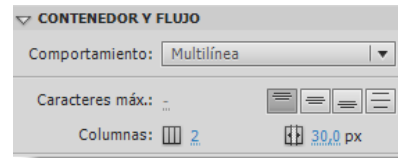
Afegirem fotogrames en la línia de temps d'aquest clip. Per a això seleccionem el segon fotograma amb el **botó dret** del ratolí i seleccionem **Insertar fotograma clave en blanco**. Repetim el mateix procés per afegir un tercer fotograma.

Ens situem en el segon fotograma i creem un bloc de text de les mateixes característiques que els del primer fotograma (mateixa font, mida, etc.).

Posicionem el bloc de text en el punt X:0 i Y:0 de la seva línia de temps, i li donem la mida AN:450 i AL:270.

Copiem dins del text els cinc paràgrafs que es troben sota el número 2 en el document de text adjunt.

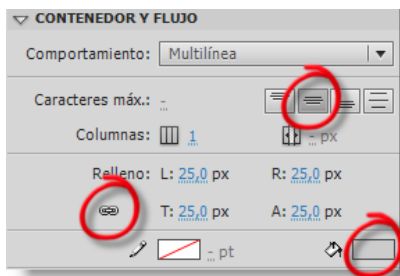
En l'àrea de **Contenedor y flujo** de l'inspector de *Propiedades*, seleccionem 2 columnes amb una separació entre elles de 30 px.



## Pas 14 de 18

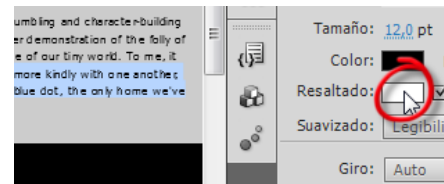
Crearem un altre camp de text en el tercer fotograma. En l'inspector de *Propiedades* li donem els valors X:0, Y:0, AN:450 i AL:250. Copiem els dos últims paràgrafs de l'arxiu de text.

En l'àrea **Contenedor y flujo**, alineem el text verticalment amb el centre del contenidor, seleccionem un padding per als quatre costats de 25 px i seleccionem un farciment *gris*.



En l'àrea **Carácter** seleccionem un color *negre* perquè el text es llegeixi amb més claredat.

Seleccionem una part de l'últim paràgraf, des de *our responsibility* fins al final, i seleccionem un **resaltado** blanc.



Amb això ja tindrem els nostres tres fotogrames amb el text del vídeo, i hem explorat part de les noves possibilitats del Text Layout Framework.

Tornem a l'escena principal i donem al nostre clip el nom d'instància *tlf\_mc*.

## Pas 15 de 18

Programarem que el text vagi sincronitzat amb el vídeo. Després crearem el botó que utilitzarem per alternar les vistes entre vídeo i text.

Quan la pel·lícula s'iniciï, en primer lloc haurem de detenir el clip que conté el text en el primer fotograma, així com fer-ho invisible. Per fer-ho escriurem en la capa *Actions* el codi següent:

```
tlf_mc.gotoAndStop(1);
tlf_mc.visible = false;
```

Perquè el text vagi al fotograma que coincideixi amb el nom del botó premut, dins de la funció `fl_ClickToSeekToCuePoint` afegirem la línia:

```
tlf_mc.gotoAndStop(event.target.name.substr(7,1));
```

Dins de la funció `fl_CuePointHandler` haurem de fer que el clip amb el text vagi al fotograma el nombre del qual coincideixi amb el nombre del nom del botó que hem fet invisible, per la qual cosa haurem d'afegir línies en la sentència `switch`:

```
case "cuePoint1" :
    controll_btn.visible = false;
    tlf_mc.gotoAndStop(1);
    break;
```

Haurem d'anar al fotograma 2 en el cas *cuePoint2*, al fotograma 3 en el cas *cuePoint3*, i de nou al 1 en el cas *cuePoint4*.

Ara el vídeo i el text funcionaran de la mateixa manera respecte als punts de referència.

El pas següent serà afegir un botó per alternar entre totes dues vistes.

## Pas 16 de 18

Afegim una capa anomenada *switch*, on col·locarem el botó que servirà de commutador.

Seleccionem **Componentes > User interface > Button**. Posicionem el botó en X:200 i Y:5. L'anomenem instància *switch\_btn*.

Afegim aquest codi, que seguidament explicarem:

```
switch_btn.label = "Text";
var videoVisible:Boolean = true;
switch_btn.addEventListener(MouseEvent.CLICK, view);

function view(event:MouseEvent)
{
    if (videoVisible)
    {
        switch_btn.label = "Video";
        tlf_mc.visible = true;
        sagan_video.visible = false;
        videoVisible = false;
    }
    else
    {
        switch_btn.label = "Text";
        tlf_mc.visible = false;
        sagan_video.visible = true;
        videoVisible = true;
    }
}
```

La propietat `label` d'un component de tipus *Button* assigna el text que apareixerà en el botó. Començarem assignant a `label` el valor `Text`, ja que a l'inici és el vídeo el que és visible, i el botó ha de servir per a veure el text.

Creem una variable anomenada `videoVisible` de tipus booleà (pot tenir el valor `true` o `false`). Com a l'inici el vídeo és visible li donem el valor inicial `true`.

Afegim un listener al botó `switch_btn`, perquè en ser premut executi la funció `view`.

La funció `view` avalua en primer lloc si el vídeo és visible. Si hi és, canvia l'etiqueta del botó a `Video`, mostra el clip de text, amaga el vídeo, i desa la dada que el vídeo ja no és visible.

Si en prémer el botó el vídeo no era visible s'executarà el que es troba entre les claus de l'`else`, és a dir, canviarà l'etiqueta del botó a `Text`, el clip amb el text s'amagarà, es mostrarà el vídeo, i s'emmagatzemarà la dada que el vídeo és visible.

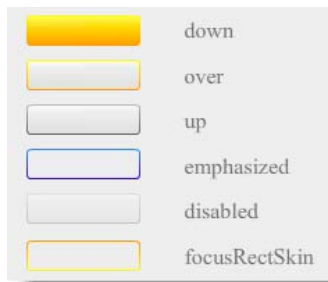
Podem provar la pel·lícula per comprovar si el nostre botó commutador funciona correctament.



## Pas 17 de 18

Fent doble clic sobre el botó en l'escenari podem personalitzar el seu aspecte. D'aquesta manera entrarem en la línia del temps del component, i podem canviar els gràfics.

Per exemple, perquè l'estil sigui més similar amb les marques d'on ens trobem, podríem seleccionar colors grocs o ataronjats per a les vores o el farciment del botó depenent del estat.



Afegim una nova capa anomenada *web*, on posarem un botó amb un enllaç a una web. Com en el cas anterior, crearem un botó partint també d'un component *Button*. El posicionarem en X:395 i Y:320.

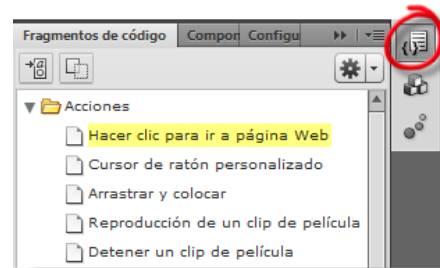
En haver-hi ja un botó en la biblioteca i no reemplaçar-lo, tindrà les mateixes característiques de personalització que hàgim posat a l'altre botó.

Com no canviarem la propietat *label* de manera dinàmica, podem assignar el valor del *label* en l'àrea **Parámetros del componente** de l'inspector de *Propiedades*. Assignem *Wikipedia* com a valor del *label*.

PARÁMETROS DE COMPONENTE	
Propiedad	Valor
emphasized	<input type="checkbox"/>
enabled	<input checked="" type="checkbox"/>
label	Wikipedia
labelPlacement	right
selected	<input type="checkbox"/>
toggle	<input type="checkbox"/>
visible	<input checked="" type="checkbox"/>

Anomenem a aquest botó instància *web\_btn*.

Amb el botó seleccionat, seleccionem **Fragmentos de código > Acciones > Hacer clic para ir a página web**.



En el codi generat, substituïm <http://www.adobe.com> per [http://en.wikipedia.org/wiki/Pale\\_Blue\\_Dot](http://en.wikipedia.org/wiki/Pale_Blue_Dot) mantenint les cometes, com indica l'ajuda.

## Pas 18 de 18

Per complementar els conceptes desenvolupats en aquest tutorial, es recomana fer les activitats següents:

1. Fer servir els coneixements d'aquest tutorial i del anterior, fes una minigaleria de vídeos (sense punts de referència), utilitzant un component de reproducció de vídeo amb més opcions, com per exemple la barra de desplaçament.
2. Crea una nova pel·lícula que contingui un vídeo i la imatge de fons de la qual canviï segons el lloc en què es trobi el vídeo.

