

## Índex de tutorials

0. Introducció
1. Creació d'una il·lustració vectorial
2. Importació d'arxius externs
3. Creació d'un logotip textual
4. Animació del logotip
5. Cinemàtica inversa i eines 3D
6. Animació amb ActionScript 3.0
7. Control de la línia de temps
8. Afegir so a botons i a la línia de temps
9. Creació d'un joc (I)
10. Creació d'un joc (II)
11. Aplicacions dinàmiques
12. Publicació i exportació

## Introducció

El Flash CS4 és una eina molt potent que ens permet crear diferents tipus de continguts com, per exemple, gràfics vectorials, animacions, recursos interactius, aplicacions multimèdia, jocs, etc.

En aquesta guia començarem coneixent les eines d'edició gràfica, per després endinsar-nos en les noves possibilitats d'animació que ofereix el Flash CS4, com per exemple l'editor de moviment, l'animació d'osso i les eines 3D. També afegirem so a les nostres animacions.

Per afegir interactivitat, ens introduirem en la programació amb l'ActionScript. Aprendrem a controlar des de la reproducció d'una animació fins al cas més complex de la creació d'un joc. També aprendrem a treballar amb contingut extern (text, imatges i vídeos) per aconseguir aplicacions dinàmiques.

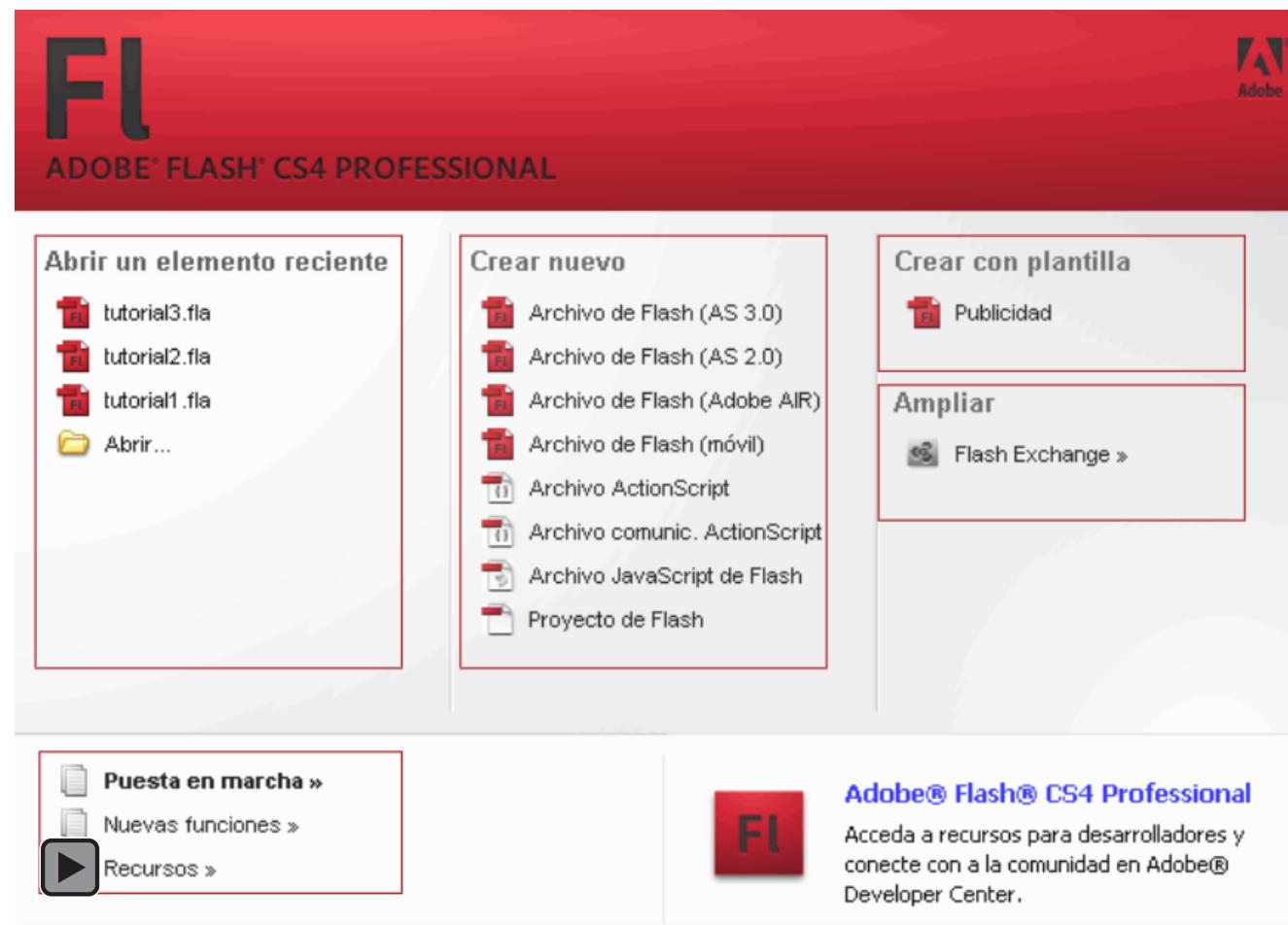
Finalment, coneixerem diferents maneres de publicar el contingut que hagim creat.

## Tutorial 1. Creació d'una il·lustració vectorial

### Pas 1 de 14

Aquesta és la pantalla de benvinguda que apareix en executar el Flash. En aquesta pantalla tenim cinc àrees:

- **Abrir un elemento reciente.** Accés directe a arxius recents i opció **Abrir** per a buscar altres arxius.
- **Crear nuevo.** Creació d'un document nou del Flash del tipus que seleccionem. Per començar seleccionarem **Crear nuevo > Archivo de Flash (AS 3.0)**.
- **Crear con plantilla.** Obre un nou arxiu a partir d'una plantilla.
- **Ampliar.** Vincle al Flash Exchange, un lloc d'intercanvi de recursos.
- **Puesta en marcha, Nuevas funciones i Recursos.** Vincles a llocs d'ajuda del Flash.



Desplaceu el cursor per sobre les diferents àrees per obtenir informació més detallada.

## Tutorial 1. Creació d'una il·lustració vectorial

### Pas 2 de 14

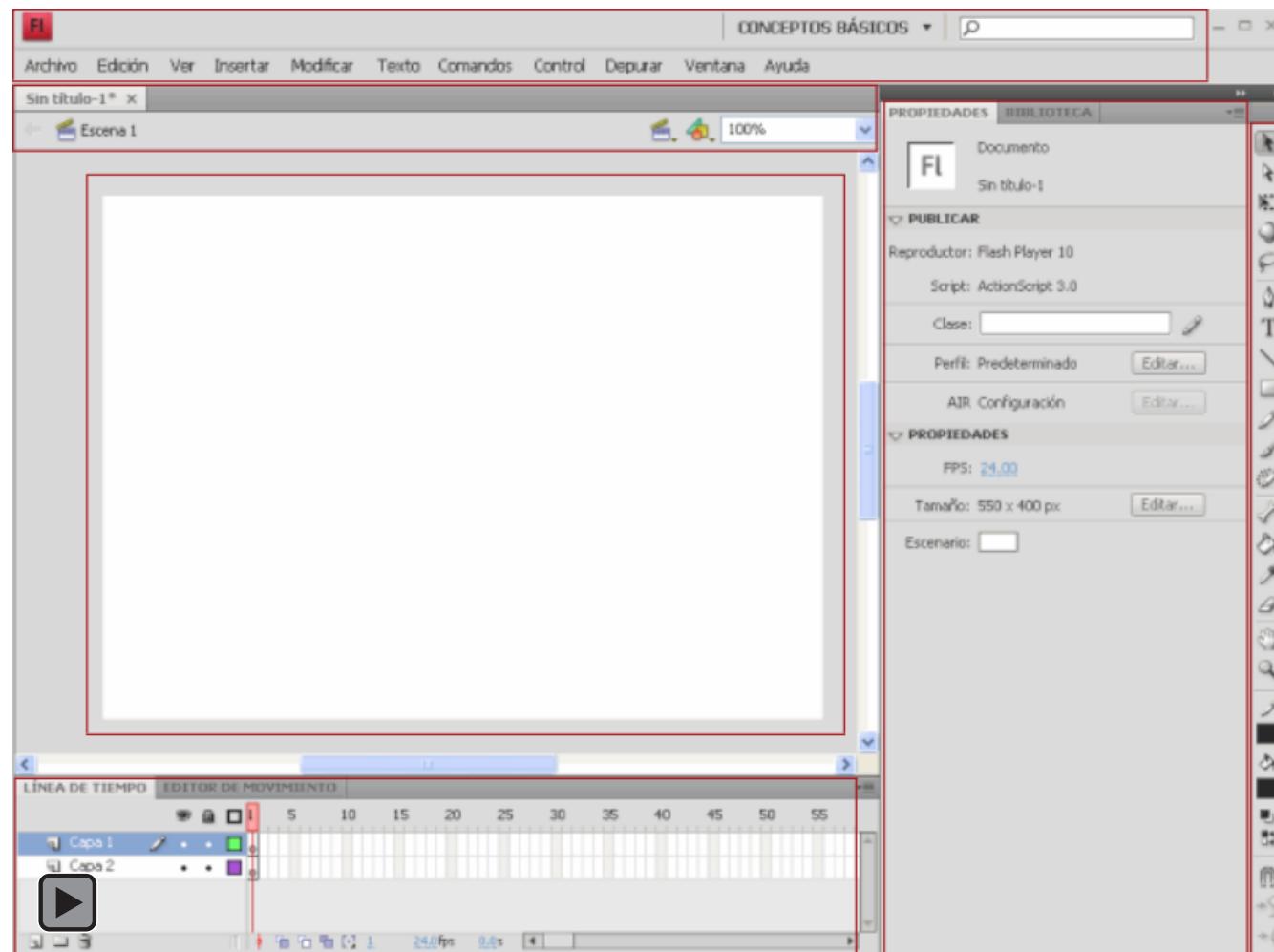
En el Flash CS4 podem triar entre diversos espais de treball, i també crear el nostre espai de treball personalitzat.

Cada espai de treball té una distribució diferent de les diverses àrees, i pot mostrar per defecte uns panells o uns altres.

Podem canviar a diferents espais de treball seleccionant en el menú desplegable, situat en la part superior al costat del cercador, l'espai de treball que vulguem.

En aquesta guia treballarem amb l'espai de treball anomenat **Conceptos básicos**.

Per veure una descripció de cada zona d'aquest espai, desplaceu el cursor per les diferents àrees.



## Tutorial 1. Creació d'una il·lustració vectorial

### Pas 3 de 14

Generalment, el primer pas que farem en treballar amb un arxiu Flash és configurar les característiques principals de la pel·lícula.

Podem veure les característiques de la nostra pel·lícula en l'**inspector de propietats**. Si no veiem aquest tauler, hem de prémer el menú **Ventana > Propiedades**.\*

Si en aquest tauler veiem altres dades que no corresponen a la imatge, seleccionarem l'eina **Selección** prement la fletxa superior de la barra d'eines o bé prement la tecla d'accés directe **V**. Aquest accés directe pot ser molt útil, ja que és una de les eines que més utilitzarem.

Amb l'eina **Selección**, premem un lloc buit de l'escenari per assegurar-nos que no hi hagi res seleccionat.

\*Quan un tauler no és visible, podem obrir-lo fent clic sobre el seu nom des del menú **Ventana**.

En la part superior d'aquest tauler veiem el títol del document. Després hi ha l'àrea **Publicar**, de la qual parlarem en altres tutorials. Finalment hi ha l'àrea de **Propiedades**.

L'fps mostra els fotogrames per segon en els quals hi ha configurada la pel·lícula. Com més gran és el nombre d'fps, més velocitat i fluïdesa té l'animació. Per a algunes animacions pot ser suficient un valor de 12 fps o 18 fps, però de moment deixem aquest valor a 24 fotogrames per segon.

Després tenim la **mida** de l'escenari. Si volem canviar aquest valor, hem de prémer el botó *Editar* que tenim al seu costat. Per a aquest tutorial no en modificarem la mida.

L'**Escenario** mostra el color de fons de la nostra pel·lícula. Canviarem el color de fons prement el quadre de color i seleccionant un color blavós, tal com es pot veure en aquest vídeo.



## Tutorial 1. Creació d'una il·lustració vectorial Pas 4 de 14

Per començar dibuixarem un núvol traçant diversos cercles superposats.

En primer lloc seleccionarem l'eina **Óvalo**, que s'agrupa amb l'eina **Rectángulo**. Totes les eines que tenen altres eines agrupades mostren en la barra d'eines una fletxa negra petita. Si mantenim premuda l'eina corresponent podem accedir a la resta d'eines que estan agrupades al seu costat.

Una vegada seleccionada l'eina **Óvalo**, l'inspector de propietats mostrarà atributs relacionats amb aquesta eina. L'única cosa que assegurarem de moment és que el **traç** sigui negre i el **farciment** blanc. Si no ho és, premerem el quadre de color corresponent per canviar-lo. Aquests valors també es poden canviar directament des del tauler *d'Herramientas*. El traç es refereix al color del perfil de l'objecte, i el farciment, al color que mostra en el seu interior.



Les **Opciones de óvalo** només poden modificar-se abans de dibuixar l'oval, llevat que haguéssim triat l'eina **Óvalo simple**. En aquest cas podríem modificar aquests valors després d'haver dibuixat l'objecte.

Tanmateix, amb **Óvalo simple** l'objecte roman agrupat i independent, per la qual cosa no es combina amb altres formes en superposar-se.

Entre **Rectángulo** i **Rectángulo simple** hi ha aquestes mateixes diferències.

En aquest cas superposarem i combinarem diversos cercles, motiu pel qual hem triat l'eina **Óvalo**.

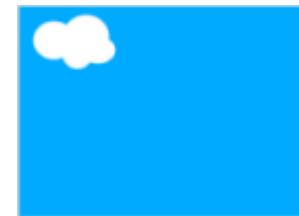
## Tutorial 1. Creació d'una il·lustració vectorial

### Pas 5 de 14

El **Dibujo de objeto**, un modificador que apareixerà en la part inferior de la barra d'*Herramientas*, haurà d'estar també desactivat, ja que altrament la forma creada també restarà agrupada i, per tant, no la podrem combinar amb altres formes.



Desactivarem també el modificador **Ajustar a objetos** per tenir més llibertat a l'hora de crear les formes ovalades i situar els cercles que dibuixem. Per exemple, si aquest modificador està activat, els ovals gairebé rodons es convertiran en cercles perfectes, la qual cosa en aquest cas no ens interessa.



Ara dibuixarem un núvol i el situarem en la cantonada superior esquerra de l'escenari.



Amb l'eina **Óvalo** seleccionada, dibuixem quatre ovals en l'escenari de manera similar a com es mostra en el vídeo.

Després, amb l'eina **Selección** (la fletxa superior de la barra d'eines, que també es pot activar prement la tecla **V**), fem clic sobre algun dels traços negres. Com podrem veure, quedarà seleccionat un tram del traç, ja que els ovals s'han combinat per a formar ara un sol dibuix. Eliminem el traç seleccionat prement la tecla **Supr**.

Podem eliminar la resta dels traços de la mateixa manera, és a dir, seleccionant cada tram i esborrant-lo amb la tecla **Supr**.

Tanmateix, com que tots els traços estan units per algun punt, una manera més ràpida d'esborrar-los és fent **doble clic** sobre qualsevol traç, i amb això quedaran seleccionats tots els traços que estiguin units. Així, en prémer la tecla **Supr** s'esboraran tots els traços seleccionats.

## Tutorial 1. Creació d'una il·lustració vectorial

### Pas 6 de 14

Si observem la **línia de temps**, podem veure que ara el primer fotograma de la *Capa 1* té un punt negre sobre un fons gris, la qual cosa indica que el nostre dibuix ha quedat inclòs en aquest fotograma.

Les capes, a més de definir l'ordre d'apilament dels diferents elements que tinguem en l'escenari, també són una bona manera d'organitzar i mantenir independents les diferents parts d'una il·lustració. És per tant molt recomanable crear una capa diferent per a cada element.

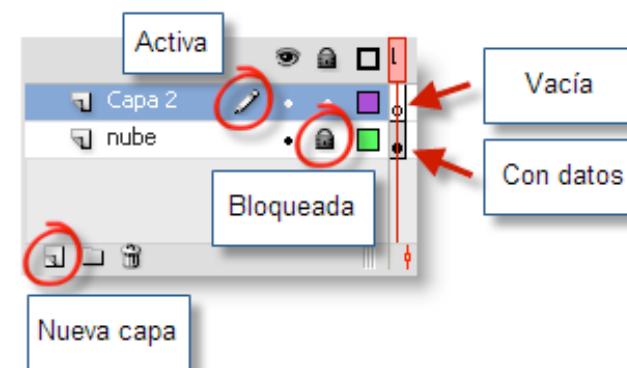
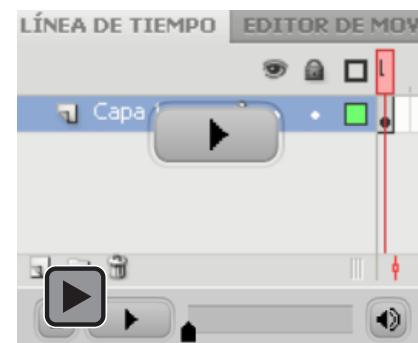
En primer lloc **reanomenem** la capa actual com a *nube* fent doble clic sobre el nom. Per evitar modificar-ne el contingut accidentalment, la **bloquegem** fent clic sobre el punt que hi ha a sota del cadenat.

Finalment, crearem una **nova capa**.

Si seleccionem aquesta nova capa, un llapis petit al costat del nom indica que està activa.

Abans de continuar, deseu l'arxiu amb el nom *tutorial1.fla* seleccionant **Archivo > Guardar**.

És convenient que ens acostumem a desar l'arxiu regularment a mesura que avancem amb el tutorial.



## Tutorial 1. Creació d'una il·lustració vectorial Pas 7 de 14

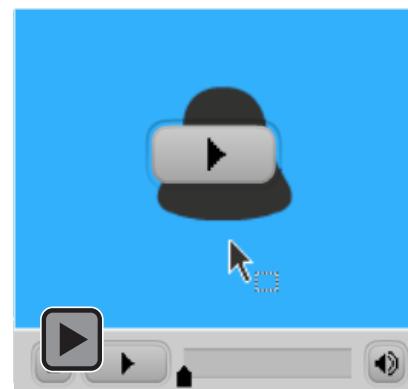
Dibuixarem un ocell amb les ales esteses, començant pel cos.

En primer lloc assignarem que no hi hagi **träç** (representat per un rectangle blanc amb una franja vermella) i un color de **farciment** negre.

Aquesta vegada canviarem els colors en la mateixa barra d'eines. El resultat és el mateix que si ho fem en l'inspector de propietats.



Una de les maneres més senzilles de començar a dibuixar és fent formes bàsiques com, per exemple, ovals o rectangles, i modificant després aquestes formes amb l'eina **Selección**. Dibuixarem l'ocell d'aquesta manera.



Per crear el cos de l'ocell, dibuixarem una forma ovalada amb l'eina **Óvalo**. Després modificarem l'oval amb l'eina **Selección (V)** fins a aconseguir una forma acampanada similar a la del vídeo.

Quan aprosem el punter a una forma amb l'eina **Selección**, el punter canvia per indicar el tipus de modificació que es pot fer en la línia o en el farciment: si apareix una cantonada podrem modificar un extrem, si apareix una corba podrem ajustar una corba, i si apareix una creu podrem desplaçar l'objecte.

Sempre que vulguem fer una d'aquestes modificacions haurem de clicar quan el punter adopti la forma que busquem i arrossegar-lo fins al lloc que volem.

En aquest cas, en situar-nos a la vora de l'oval, només apareixia la forma corbada, ja que en tractar-se d'una forma ovalada no hi ha cap cantonada.



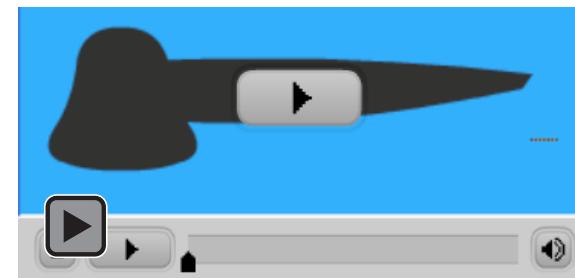
## Tutorial 1. Creació d'una il·lustració vectorial

### Pas 8 de 14

D'una manera similar a com ho hem fet en el pas 6, anomenarem *cuerpo* la capa actual, la bloquejarem, i crearem una nova capa anomenada *alas*.



Per a dibuixar l'ala utilitzarem l'eina **Rectángulo**, i posteriorment modificarem la forma creada amb l'eina **Selección**.



Com podem veure en el vídeo, en aquest cas sí que podem modificar tant cantonades com corbes. La forma del punter ajuda a trobar les diferents corbes i cantonades fins i tot en l'àrea en la qual se superposen totes dues formes negres (ales i cos).

Per ara ens interessa que l'ala es mostri totalment estesa encara que quedi menys realista. Més endavant, en els tutorials d'animació, la doblegarem i mourem convenientment.

Farem l'ala esquerra a partir d'una còpia de l'ala dreta.

Normalment, quan volem reutilitzar un objecte, el més adequat és convertir-lo prèviament en símbol. Tanmateix, per a aquest primer tutorial copiarem i enganxarem l'ala directament, sense convertir-la en símbol. Estudiarem l'ús dels símbols en el pròxim tutorial.

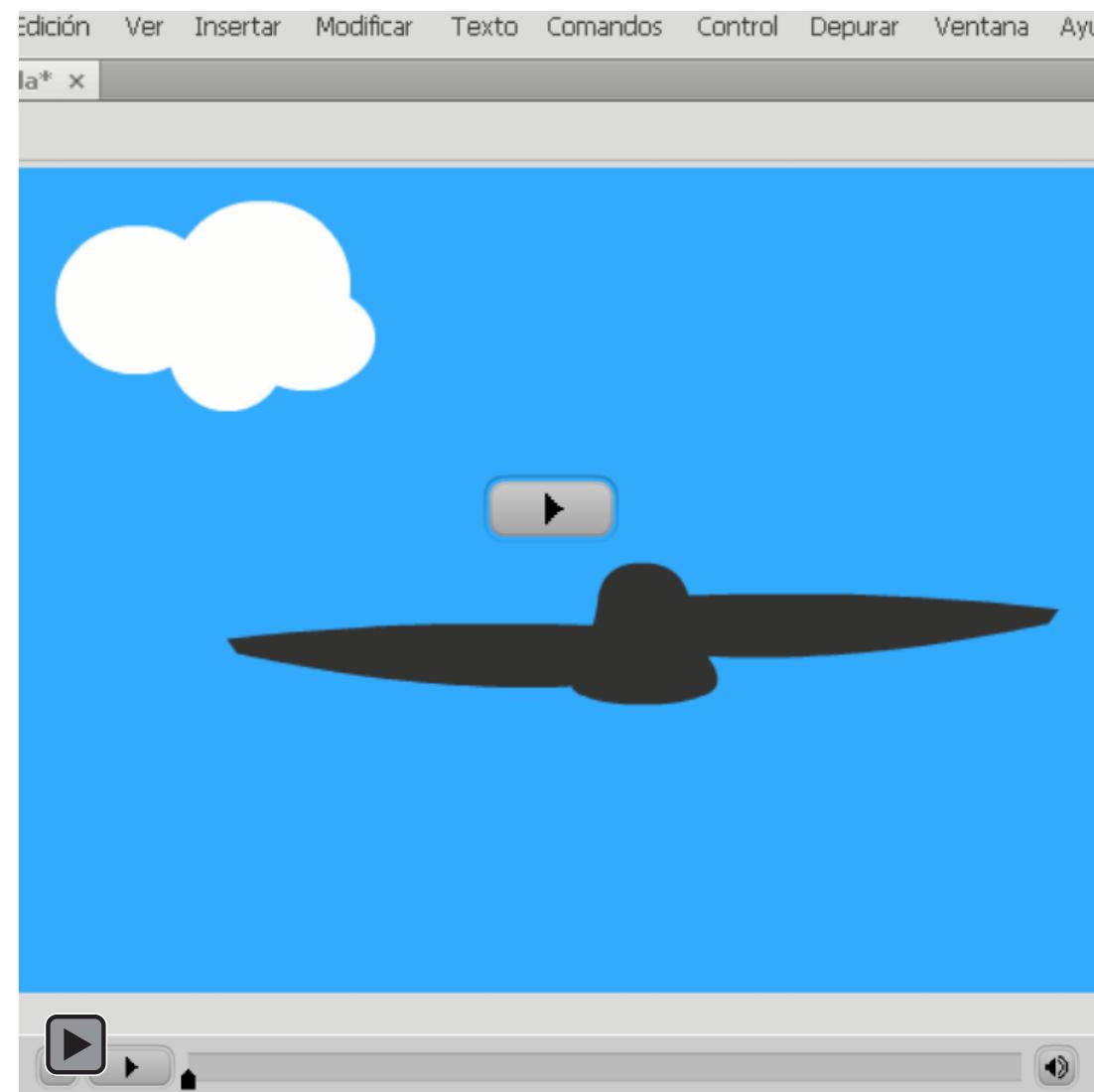
## Tutorial 1. Creació d'una il·lustració vectorial **Pas 9 de 14**

Fem clic sobre l'ala amb el **botó dret** del ratolí i seleccionem **Copiar** del menú contextual.

Després, sobre algun lloc buit de l'escenari, tornem a fer clic amb el **botó dret** del ratolí i seleccionem **Pegar**.

Amb la còpia de l'ala seleccionada, seleccionem **Modificar > Transformar > Voltear horizontalment**.

Fem clic sobre l'ala i l'arrosseguem fins a una posició aproximada a la seva ubicació final. En el pas següent, amb ajuda del **tauler Alinear**, la col·locarem en la posició definitiva.



## Tutorial 1. Creació d'una il·lustració vectorial Pas 10 de 14



Obrim el tauler *Alinear* seleccionant  
**Ventana > Alinear.**

En el Flash podem col·locar els taulers al lloc que ens resulti més còmode i podem personalitzar d'aquesta manera l'espai de treball segons les nostres necessitats.

En el vídeo es mostren algunes de les opcions de visualització que tenen tots els taulers, com ara desplaçar-se i minimitzar-se de diverses maneres.

També és possible agrupar els taulers amb altres àrees de l'espai de treball. Si en qualsevol moment volem tornar a l'espai de treball per defecte, seleccionarem **Restaurar** en el desplegable de l'espai de treball.



Si no disposem de gaire espai per a treballar, el més còmode serà obrir els taulers a mesura que els anem necessitant, resituar-los en un lloc que ens resulti còmode, i tancar-los després d'haver-los fet servir.

## Tutorial 1. Creació d'una il·lustració vectorial Pas 11 de 14

Tornant al nostre ocell, alinearem en primer lloc les ales amb ajuda del **tauler Alinear**.

El primer que hem de comprovar és que la casella *En escena* estigui desactivada, ja que altrament els elements no s'alineararan entre ells, sinó respecte a l'escenari.

Seleccionem les dues ales fent-hi clic a sobre amb l'eina **Selección**. Per a fer una selecció múltiple haurem de mantenir premuda la tecla **Mayúsculas**.

Amb totes dues ales seleccionades fem clic a **Alinear borde superior**.

Desbloquegem la capa del cos de l'ocell per poder-la seleccionar, i amb els dos elements seleccionats (el cos i les ales) fem clic a **Distribuir horizontalmente respecto al centro**.



Per a organitzar més bé els elements d'una pel·lícula, podem agrupar les capes en carpetes.

Crearem en la **Línia de temps** una carpeta nova anomenada *pájaro* que inclourà les capes de les ales i del cos. Per a això, després de crear la carpeta, arrossegarem les capes a l'interior.

En bloquejar la carpeta podem veure que es bloquegen totes les capes que conté.

Després crearem una nova capa per afegir més elements en l'escenari.

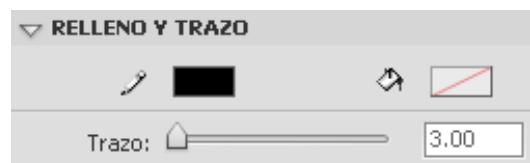


Recordeu-vos de desar regularment la vostra feina amb **Archivo > Guardar**.

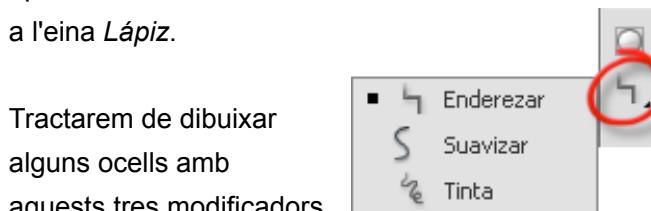
## Tutorial 1. Creació d'una il·lustració vectorial Pas 12 de 14

Farem una petita aproximació a dues eines vectorials més del Flash, començant per l'**eina Lápiz**.

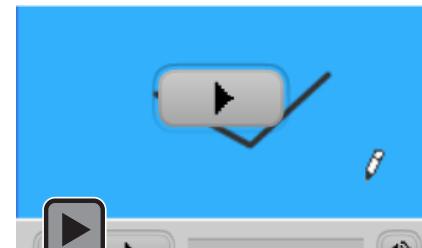
En activar el llapis, veurem que en l'**inspector de Propiedades** el farciment apareix automàticament amb una franja vermella que significa que no hi ha farciment, ja que el llapis només pot dibuixar traços. Seleccionarem un color negre per al traç, i un gruix de 3 pixels.



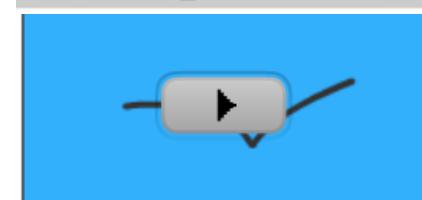
En la part inferior de la barra d'eines apareixeran els modificadors associats a l'eina **Lápiz**.



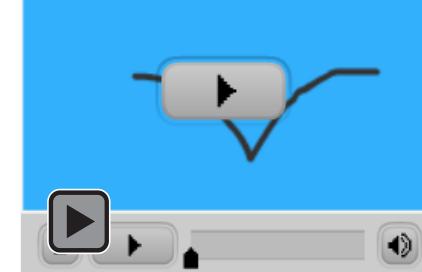
Tractarem de dibuixar alguns ocells amb aquests tres modificadors.



El modificador **Enderezar** converteix les línies en rectes o en figures geomètriques comunes.



El modificador **Suavizar** dibuixa corbes suaus.



El modificador **Tinta** dibuixa corbes a mà alçada, sense aplicar cap modificació.



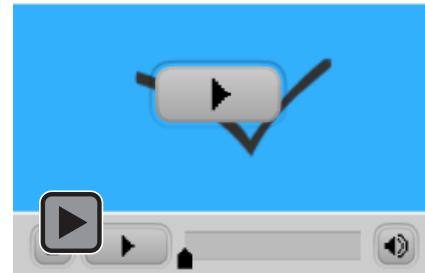
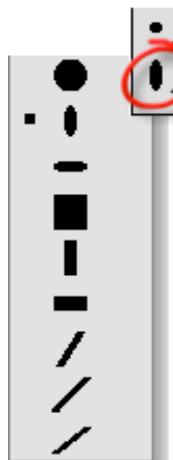
Una vegada dibuixada una forma, tant amb aquestes eines com amb d'altres, sempre es podrà suavitzar o redreçar fent clic repetidament en els modificadors **Suavizar** i **Enderezar** que apareixeran en seleccionar una forma en l'escenari.

## Tutorial 1. Creació d'una il·lustració vectorial Pas 13 de 14

Finalment utilitzarem l'eina **Pincel**.

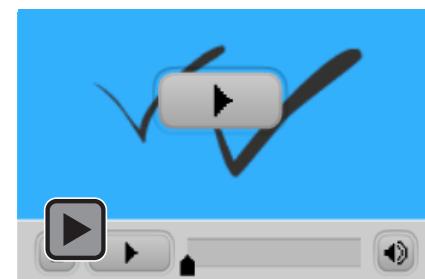
En la part inferior de la barra d'eines apareixeran els modificadors associats a l'eina *Pincel*.

Els dos modificadors inferiors es refereixen a la **mida** i la **forma** del pinzell. Seleccionarem com a **forma del pinzell** una forma ovalada vertical.



El dibuix fet amb el pinzell queda generalment suavitzat, per la qual cosa és una bona eina per a fer aquest tipus de dibuixos.

A diferència del llapis, el pinzell pinta farciments, i no traços. Per tant, el dibuix fet amb el pinzell es pot modificar com un farciment.



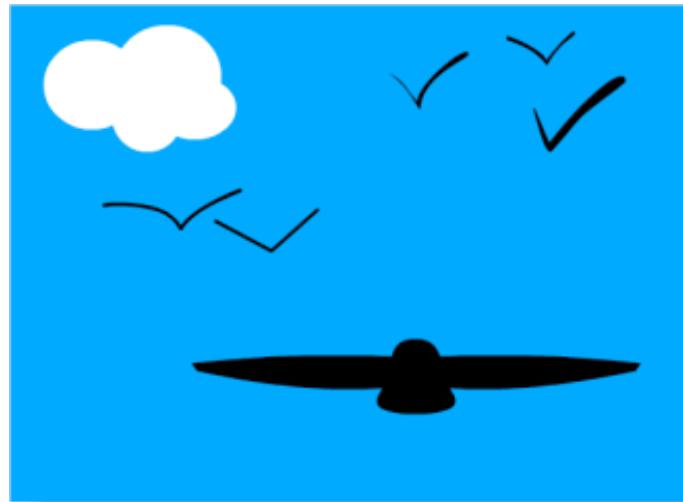
Si es té una tauleta sensible a la pressió apareixeran dos nous modificadors, el modificador **Presión** i el modificador **Inclinación**. Si s'activen aquests modificadors podrem variar l'amplada i la inclinació dels traços del pinzell dependent de la pressió i inclinació que exercim amb el nostre llapis.



Hem dibuixat els ocells d'aquests dos últims passos només per experimentar amb l'eina *Lápiz* i l'eina *Pincel*, però utilitzarem el primer ocell i el núvol en tutorials posteriors sobre animació, per la qual cosa és molt important que deseu el vostre treball. Per fer-ho, seleccioneu **Archivo > Guardar** o **Ctrl + S**.

Tutorial 1. Creació d'una il·lustració vectorial

**Pas 14 de 14**



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

1. Dibuixeu un altre núvol i després modifiqueu-ne la forma amb l'eina *Selección*.
2. Modifiqueu també la forma dels ocells que hem creat amb el llapis i amb el pinzell per comprendre millor les diferències entre aquestes dues eines.
3. Creeu una forma nova partint d'un oval o d'un rectangle.

## Tutorial 2. Importació d'arxius externs

### Pas 1 de 24

En aquest tutorial crearem un paisatge que utilitzarem en altres tutorials. Per això és important que desem el resultat.

Per crear aquest paisatge importarem i modificarem arxius externs. D'una banda, importarem un gràfic vectorial creat amb l'Adobe Illustrator que conté l'herba i els arbres. D'altra banda, treballarem amb una imatge importada de mapa de bits per crear la lluna.

Els gràfics vectorials representen la imatge mitjançant línies i corbes (vectors), mentre que els mapes de bits estan formats per punts (píxels). Per això la manera de treballar amb tots dos tipus de formats és diferent.

Per a completar el paisatge utilitzarem l'eina *Pluma* i crearem la muntanya, i treballarem amb els degradats de color tant per al cel com per a la muntanya.



## Tutorial 2. Importació d'arxius externs Pas 2 de 24

En primer lloc importarem l'arxiu vectorial *arbol.ai* que hi ha en la carpeta *tutorial2*.

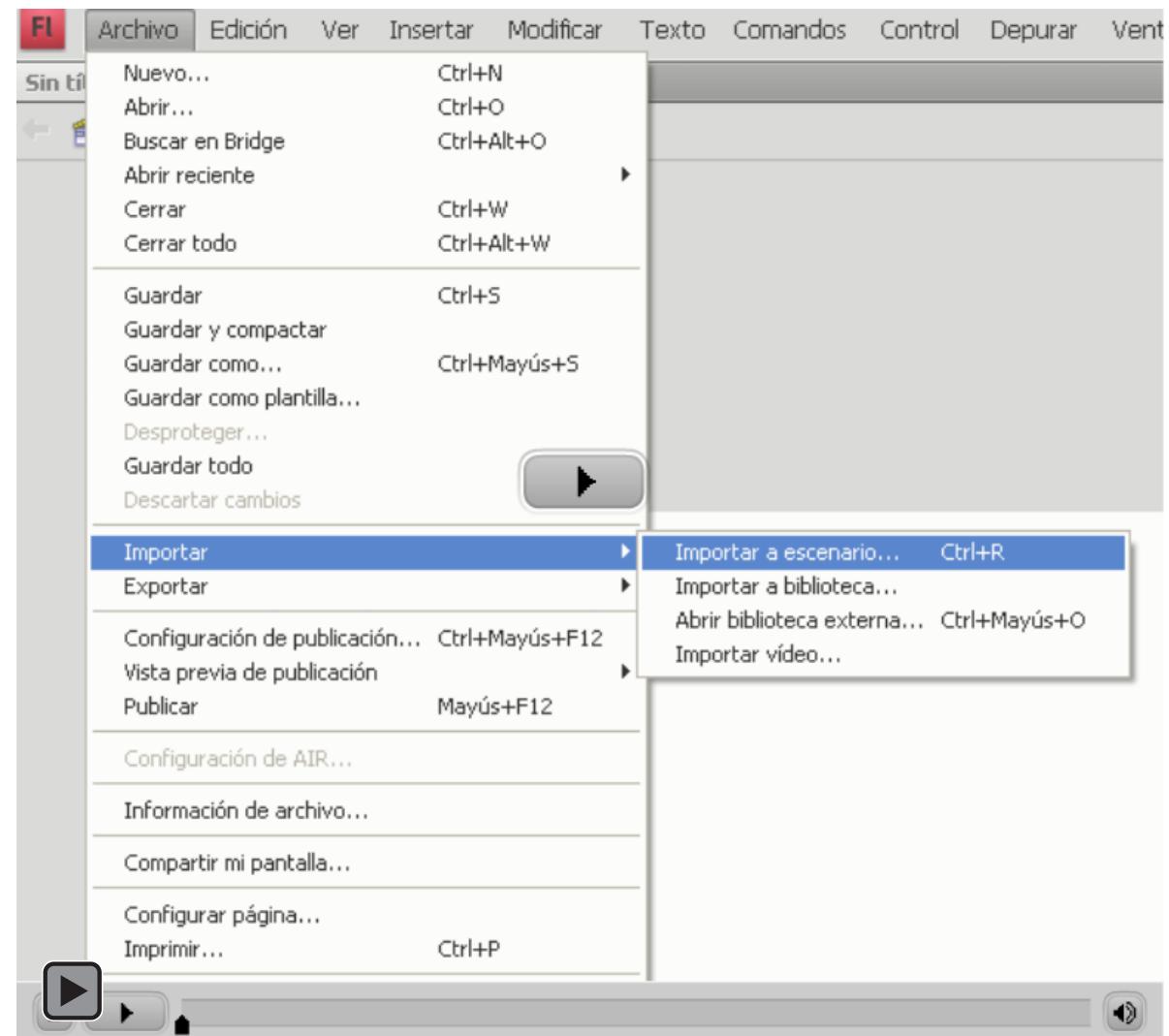
Per a això seleccionem

**Archivo > Importar > Importar a escenario.**

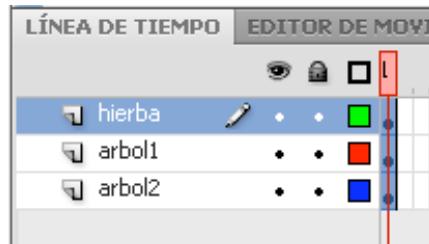
Durant la importació ens assegurem que l'opció de convertir capes en capes de Flash estigui seleccionada.

Una vegada importat, desem l'arxiu amb el nom *tutorial2.fla*. Per a això seleccionem **Archivo > Guardar (Ctrl + S)**.

És convenient que ens acostumem a desar l'arxiu regularment a mesura que avancem amb el tutorial.



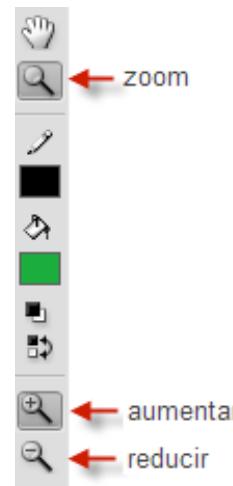
## Tutorial 2. Importació d'arxius externs Pas 3 de 24



Les capes que tenia originalment l'arxiu que hem importat apareixeran ara en la **línia de temps**.

**L'ordre de les capes** determina com se superposen els elements de l'animació. Les capes inferiors es mostraran més al fons en l'escenari. En aquest cas, l'herba es mostrarà en primer pla, i l'arbre 2 estarà per darrere de l'arbre 1.

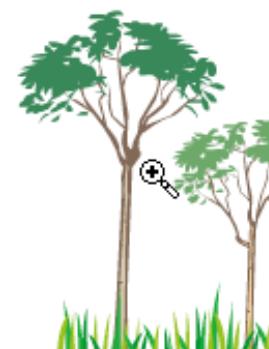
L'arxiu vectorial que hem importat es pot editar directament amb les eines del Flash, ja que manté la informació sobre traços i formes de l'arxiu original.



Modifarem una corba de l'arbre 1 (l'arbre que hi ha més a l'esquerra). Per facilitar-ne l'edició, ampliarem la zona que modificarem.

Seleccioneu l'eina **Zoom** de la barra d'eines. En la part inferior de la barra apareixeran ara els modificadors per a augmentar i reduir.

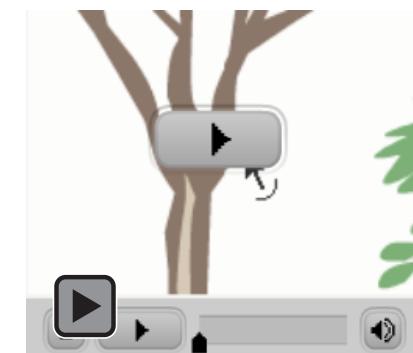
Amb **Aumentar** seleccionada, aproparem la lupa a la part superior del tronc de l'arbre 1. Hi fem clic dues vegades per ampliar prou aquesta zona.



Una vegada ampliada la zona, activarem l'**eina Selección** (la fletxa superior de la barra d'eines).



Amb la fletxa ens aprotem a la protuberància del tronc, fins que la fletxa mostra una petita corba que indica que podem fer una modificació de la corba de l'arbre. Hi fem clic i arrosseguem la fletxa fins a disminuir lleugerament aquesta protuberància.



## Tutorial 2. Importació d'arxius externs Pas 4 de 24

Fem clic amb l'eina **Zoom > Reducir** per visualitzar tot l'escenari. En passos posteriors cobrirem d'herba la part inferior de l'escenari i posicionarem els arbres en la ubicació definitiva. El primer que advertim és que necessitarem duplicar l'herba. Per a això la convertirem prèviament en **símbol**.

Els símbols són elements que s'emmagatzemen en la **Biblioteca** i que podem reutilitzar tantes vegades com calgui, i d'aquesta manera optimitzem la mida de l'arxiu. Podem crear tres tipus de símbols:

- **Gràfic.** S'utilitza principalment per a imatges estàtiques. També permet utilitzar efectes de color.
- **Botó.** Per als símbols que responguin a accions interactives. Un clip de pel·lícula també pot respondre a accions interactives, però la particularitat del botó és que té una línia de temps per a mostrar cada estat del botó (repòs, sobre, pressionat i zona activa).
- **Clip de pel·lícula.** És el tipus de símbol més adequat per a símbols que contenen animacions. També permet utilitzar eines 3D, filtres i mescles, i es pot controlar amb programació, per la qual cosa pot ser útil encara que no contingui animacions.

Per aquest paisatge crearem un símbol gràfic per a l'herba i també per a cada arbre, ja que, encara que no dupliquem els arbres, els donarem un efecte de color en els pròxims passos.

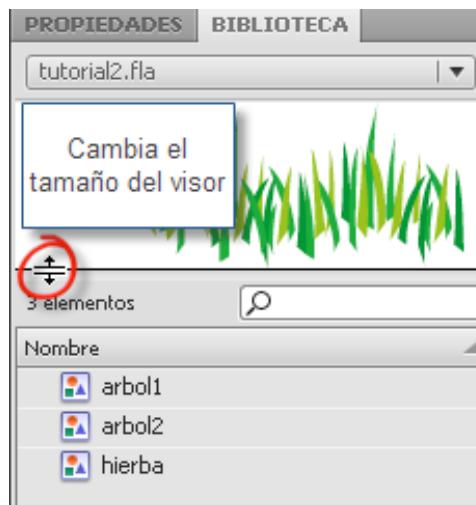
Per crear cada símbol gràfic, en primer lloc hem de seleccionar-ne tots els components. Una manera ràpida de fer aquesta selecció, aprofitant el fet que cada objecte és en una capa diferent, és clicar sobre el fotograma que conté cada element.

Per exemple, seleccionem el fotograma de la capa *arbol2*. Una vegada seleccionat, seleccionarem **Modificar > Convertir en símbolo** o premem directament **F8**. En la pantalla següent donarem el nom *arbol2* al símbol, i seleccionarem el tipus **Gràfico**. Repetim el mateix procés per a l'arbre 1 i per a l'herba.



## Tutorial 2. Importació d'arxius externs Pas 5 de 24

Ara tindrem emmagatzemats en la **Biblioteca** els tres símbols que hem creat en el pas anterior. Si no veiem aquest tauler, seleccionem **Ventana > Biblioteca**. Si premem el nom d'un dels símbols podrem veure en el visor una miniatura del seu contingut. Si és necessari, podem ampliar la zona del visor.

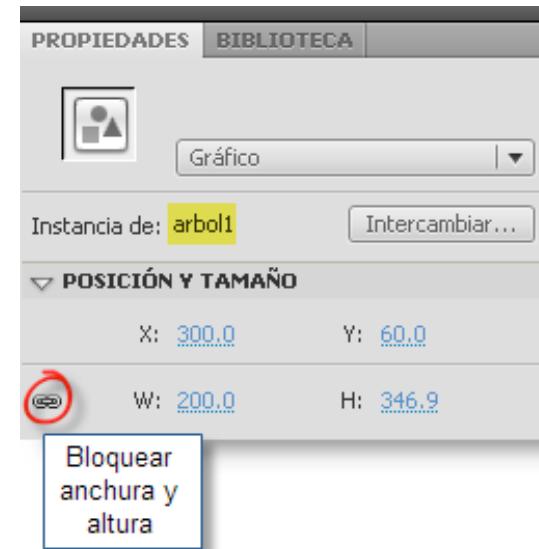


Des de l'inspector de **Propiedades** modificarem la posició i la mida de l'arbre 1 (el que està situat a l'esquerra). Si no veiem aquest tauler seleccionem **Ventana > Propiedades**.

En primer lloc, seleccionem l'arbre en l'escenari prement-hi a sobre amb l'eina **Selección** (la fletxa superior).

Quan l'hagim seleccionat, el tauler *Propiedades* indicarà que tenim seleccionada una instància del símbol gràfic *arbol1*. Les **modificacions** es faran en la **instància** de l'arbre que tenim en l'escenari, però el símbol en la biblioteca es mantindrà sense canvis.

A continuació introduïm els valors que es veuen en la imatge. Situem l'arbre en les coordenades X:300px i Y: 60px. Hi donem una amplada (*W*) de 200 px, i si tenim bloquejada l'amplada i l'altura, el valor de l'altura prendrà el valor apropiat perquè no es distorsioni la imatge.



## Tutorial 2. Importació d'arxius externs

### Pas 6 de 24

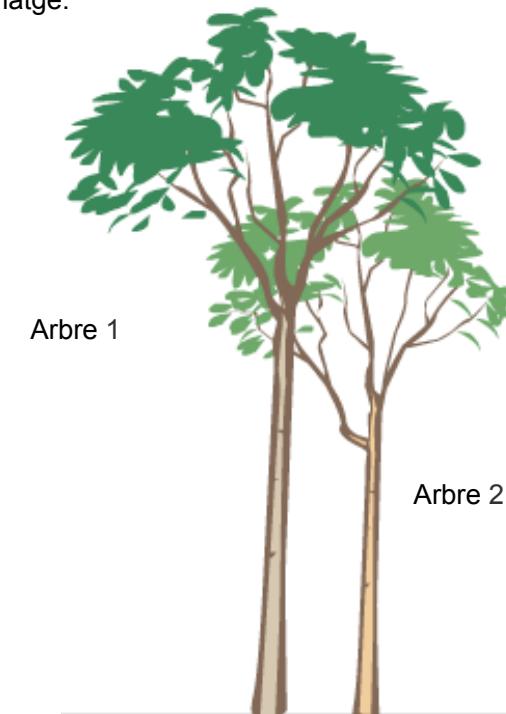
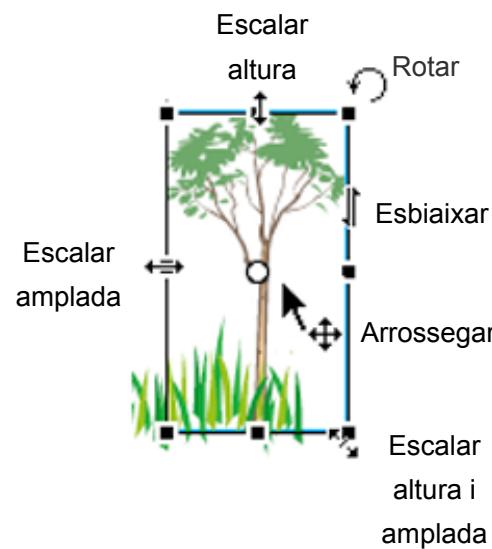
Una altra manera de fer modificacions és amb l'eina **Transformación libre**. Seleccionem aquesta eina i després premem **árbol 2** en l'escenari.

Segons on apropem el cursor, veurem que adquireix diferents formes que ens indiquen quin tipus de transformació s'aplicaria en cada cas.



Apropem el cursor a la cantonada inferior dreta de l'arbre per a **escalar l'altura i l'amplada** conjuntament. Mantenim premuda la tecla **Mayúsculas** mentre escalem per **mantenir la proporció** de l'arbre.

Una vegada escalat, l'arrosseguem fins a la posició final, de manera que quedi similar a la imatge.



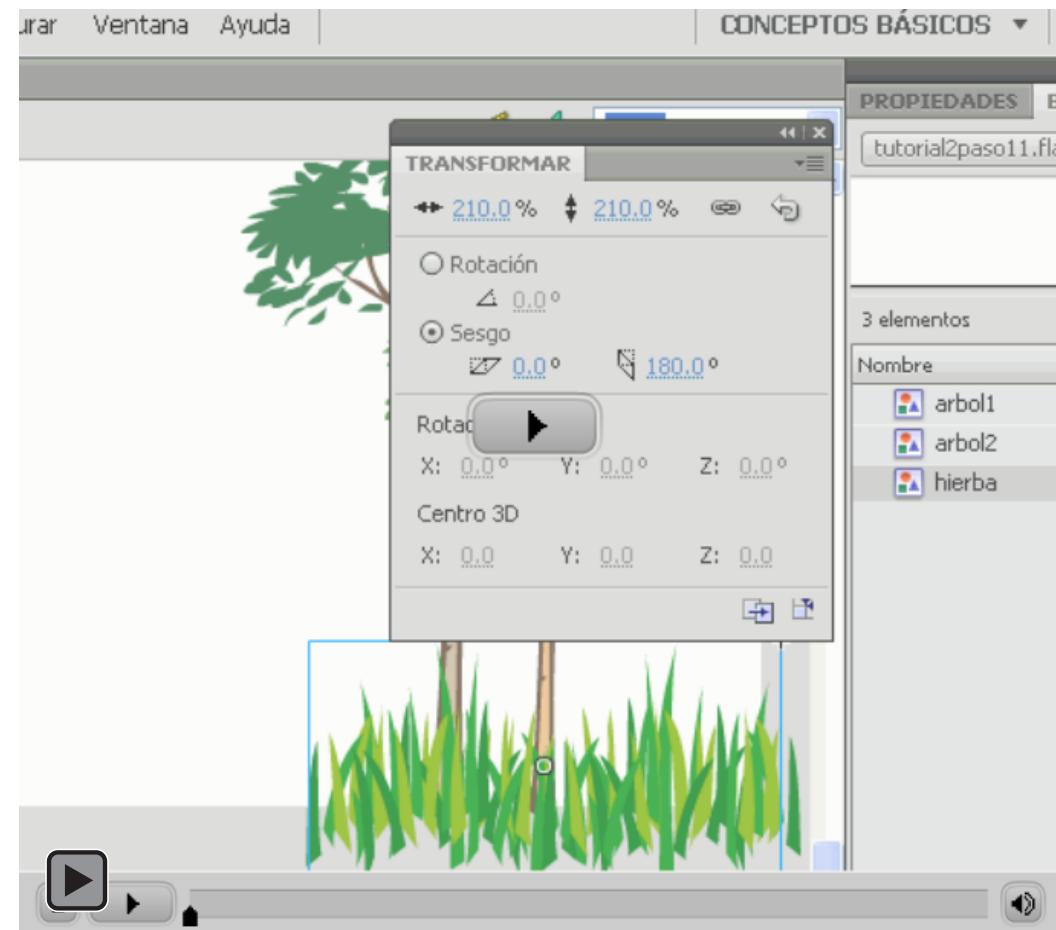
## Tutorial 2. Importació d'arxius externs

### Pas 7 de 24

Finalment, crearem sis instàncies del gràfic *herba* i les modificarem des del tauler *Transformar*, al qual accedirem des de *Ventana > Transformar*.

Donarem a cada instància valors d'escalat lleugerament diferents, i algunes de les instàncies les reflectirem horitzontalment perquè les còpies no es vegin exactament iguals.

Posicionarem les instàncies en la part inferior de l'escenari, depassant-ne lleugerament les vores.



## Tutorial 2. Importació d'arxius externs

### Pas 8 de 24

Modificarem el símbol gràfic *hierba*, del qual hi ha sis instàncies en l'escenari. En **modificar el símbol original**, veurem que **els canvis s'aplicaran a totes les instàncies del símbol**.

Una manera de modificar un símbol és fent doble clic sobre el seu nom en la **Biblioteca**. Això ens obriria el símbol des de la seva pròpia línia de temps.

Una altra manera d'editar un símbol, que és la que utilitzarem, és fent **doble clic** sobre qualsevol de les **instàncies** del símbol que hi ha en l'escenari. D'aquesta manera, podem editar el símbol sense perdre'n de vista el context. Els elements que no formen part del símbol apareixeran amb colors apagats i no es podran modificar.

Quan estiguem editant un símbol, el nom del símbol apareixerà en la barra d'edició, situada sobre l'escenari. Des d'aquesta mateixa barra podrem tornar a l'escena principal després de l'edició del símbol.



Fem doble clic sobre una instància del gràfic *hierba* per editar-ne el símbol original.

Baixem la cantonada superior de les herbes més altes. Veurem que els canvis s'aplicaran a la resta de les instàncies.

Tornem a l'escena principal fent clic a **Escena 1**.

## Tutorial 2. Importació d'arxius externs Pas 9 de 24



Creem una **nova capa** que anomenarem *montaña*.

Movem la capa cap a baix perquè quedi per darrere de l'herba i dels arbres.

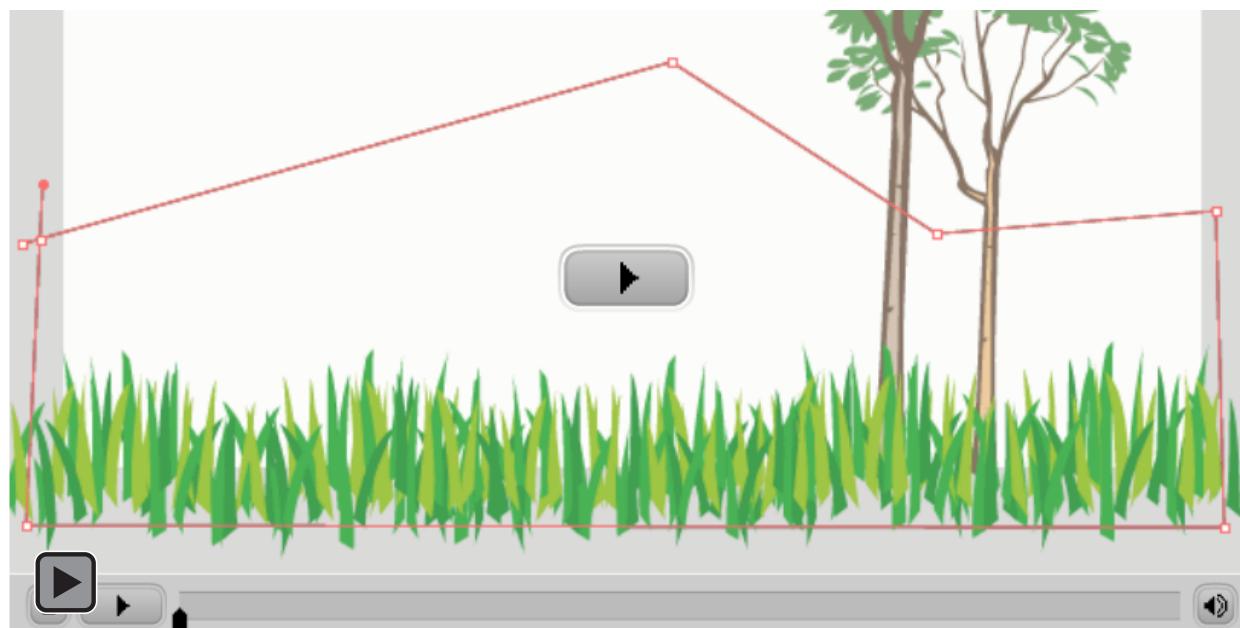
Bloquegem la resta de les capes amb el cadenat per evitar modificar-les accidentalment.

El llapis indica quina capa és la que està activa.



Dibuixarem la muntanya amb línies rectes. Amb l'eina **Pluma** fem clic per crear els punts d'àncora. Es crearan segments rectilinis connectats pels vèrtexs.

Dibuixem fora dels límits de l'escenari sense preocupar-nos, ja que el que estigui fora de l'escenari no apareixerà en l'arxiu final. Per tancar un traçat normalment ens aprosem al vèrtex inicial fins que aparegui un petit cercle i hi fem clic. En aquest cas, com que el lloc del tancament queda fora de l'escenari, creuarem l'últim segment per assegurar-nos que la forma queda perfectament tancada.



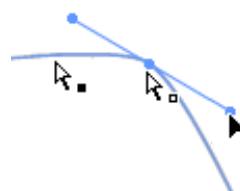
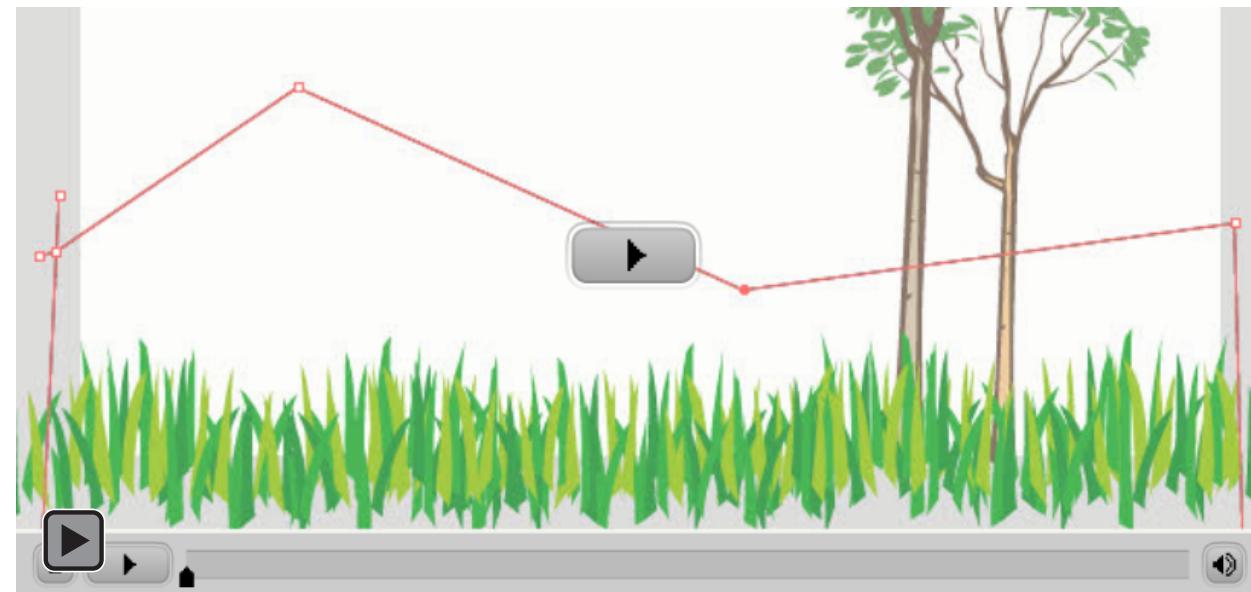
## Tutorial 2. Importació d'arxius externs

### Pas 10 de 24



Activem l'eina

**Subselección.** Quan amb aquesta eina premem un traçat, ens apareixeran els **punts d'àncora** i els **tiradors** d'aquest traçat. Els tiradors permeten determinar l'angle de la corba compresa entre dos punts d'àncora.



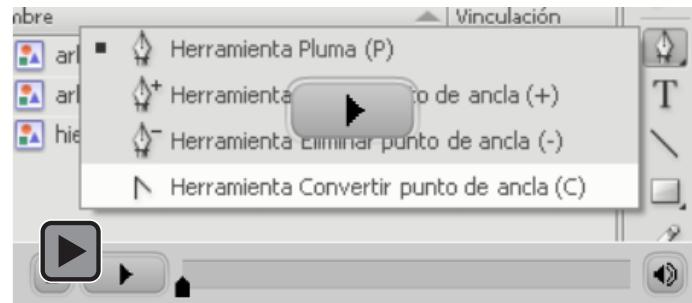
L'eina **Subselección** adoptarà diferents formes segons apropiem el cursor a diferents punts del traçat. Si la fletxa mostra un quadrat blanc petit estarem davant d'un punt d'àncora que podrem desplaçar. Si apareix un quadrat negre podrem desplaçar tot el traçat. Si el punter es converteix en una fletxa negra podrem modificar els tiradors per a ajustar la corba.

En el nostre cas la muntanya està formada, per ara, per trams rectes, per la qual cosa només apareixen els punts d'àncora, però no els tiradors. Amb l'eina **Subselección** desplaçarem els punts d'àncora que havíem creat en el pas anterior, i finalment desplaçarem lleugerament tot el traçat.

## Tutorial 2. Importació d'arxius externs

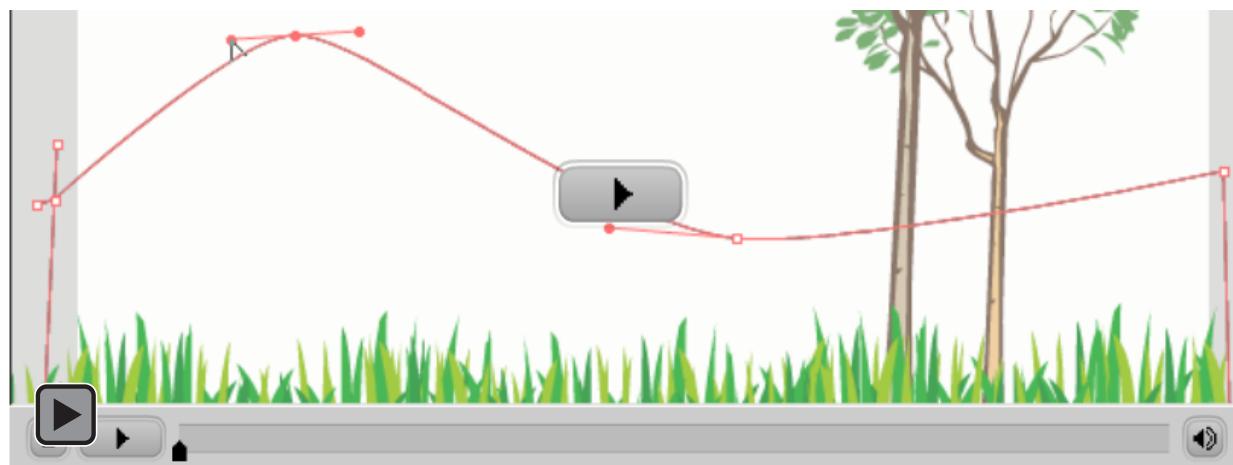
### Pas 11 de 24

Ara suavitzarem la forma de la muntanya amb l'ajuda de l'eina **Convertir punto de ancla**. Aquesta eina és en el menú emergent de l'eina **Pluma**.



L'eina **Convertir punto de ancla** converteix un punt de cantonada sense tiradors en un punt de línia corba amb tiradors.

Per a això hem de prémer els punts d'àncora creats anteriorment i arrossegar-los fins a aconseguir el tipus de corba que volem.



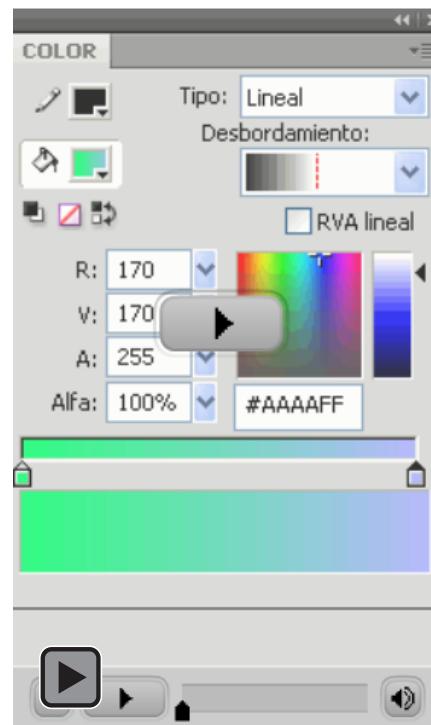
## Tutorial 2. Importació d'arxius externs Pas 12 de 24

Farcirem la muntanya amb un color **degradat**. Un degradat és un farciment multicolor, en el qual hi ha un canvi gradual entre diversos colors.

Obrim el tauler *Color* seleccionant **Ventana > Color**. Recordem que podem arrossegar el tauler prement la barra gris fosc superior del tauler.

Seleccionem un color de farciment, **Tipo: Lineal**. En la barra inferior apareixeran els colors extrems del degradat.

Fem clic dues vegades sobre un dels extrems per obrir la paleta de mostres, i seleccionem el color #00FF66. L'altre extrem el canviarem donant directament nous valors RVA (vermell, verd, blau) directament en el tauler: R:170, V:170 i A:255.

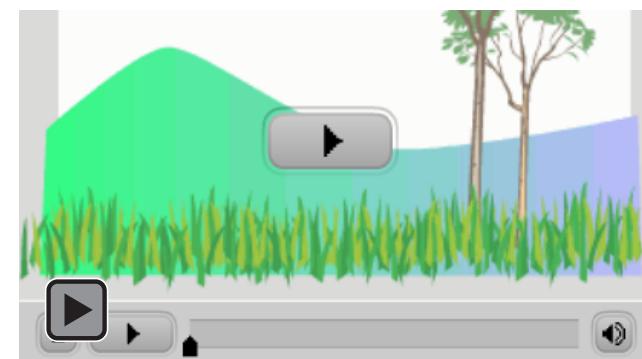


Activem l'eina **Cubo de pintura**. Fem clic sobre l'interior de la muntanya per farcir-la amb el degradat que acabem de crear.



Eliminarem el traçat de la muntanya i deixarem només el farciment. Per a això activem l'eina **Selección** des de la barra d'eines o premem la lletra **V**.

Fem doble clic sobre el traçat de la muntanya per seleccionar-lo per complet, i premem **Supr** per eliminar-ne el traçat.

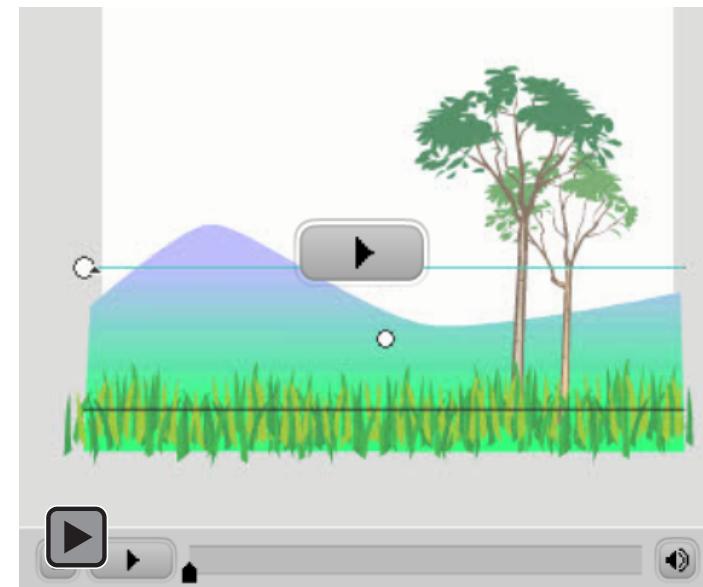
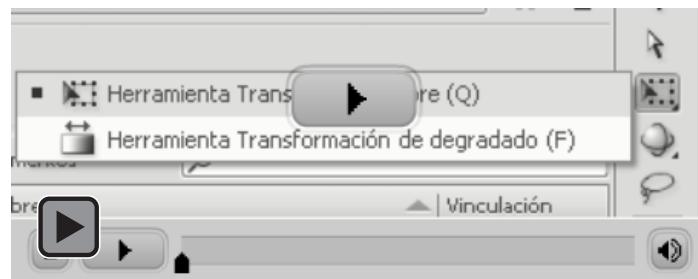


## Tutorial 2. Importació d'arxius externs

### Pas 13 de 24

Ara utilitzarem l'eina **Transformación de degradado** per a reorientar el degradat de la muntanya, de tal manera que el color blavós quedi en la part superior i el verd, en la inferior.

Aquesta eina està agrupada amb l'eina **Transformación libre**, per la qual cosa hem de mantenir premuda l'eina de **Transformación libre**, i seleccionar **Transformación de degradado** en el menú emergent.



Una vegada seleccionada aquesta eina, premerem el farciment degradat de la muntanya. El primer pas serà girar el degradat, de tal manera que el color més blavós quedi en la part superior.

Finalment estrenyerem l'àrea en què s'aplica el degradat, de tal manera que les dues línies paral·leles de delimitació quedin dins de la muntanya. Aquestes línies són les que limiten l'àrea en què es produeix la transició entre un color i un altre. La part del farciment que queda fora d'aquestes dues línies paral·leles tindrà el color pur de l'extrem que els correspongui.

## Tutorial 2. Importació d'arxius externs Pas 14 de 24

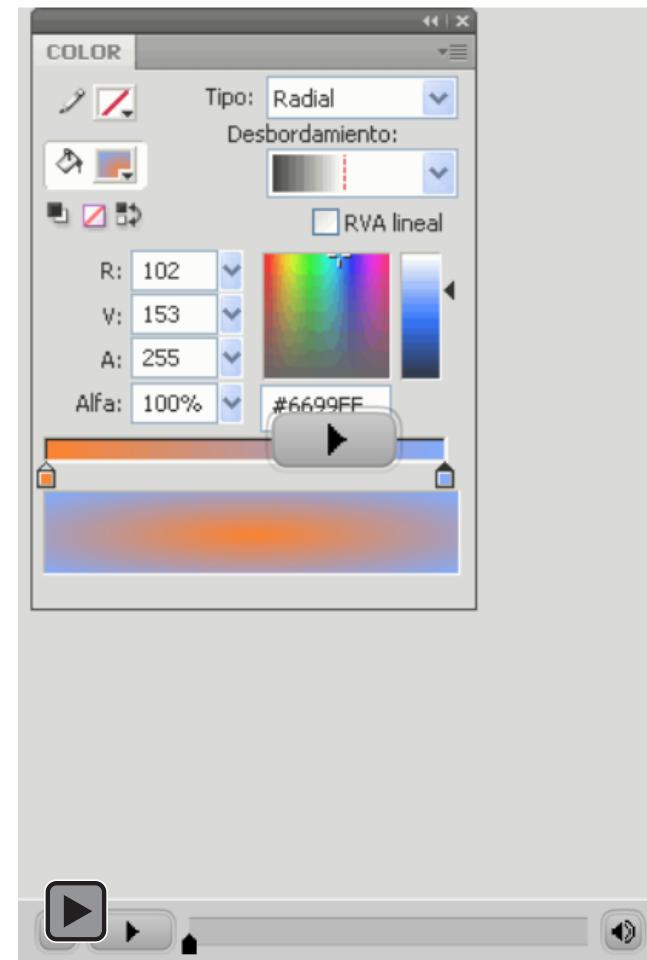


Creem una nova capa que anomenarem *cielo* de la mateixa manera que hem creat la capa per a la muntanya en el pas 9 d'aquest tutorial. Arrosseguem la capa a la part inferior, ja que ha de quedar per darrere de tots els altres elements del paisatge. Bloquegem la resta de les capes i la seleccionem perquè sigui la capa activa.

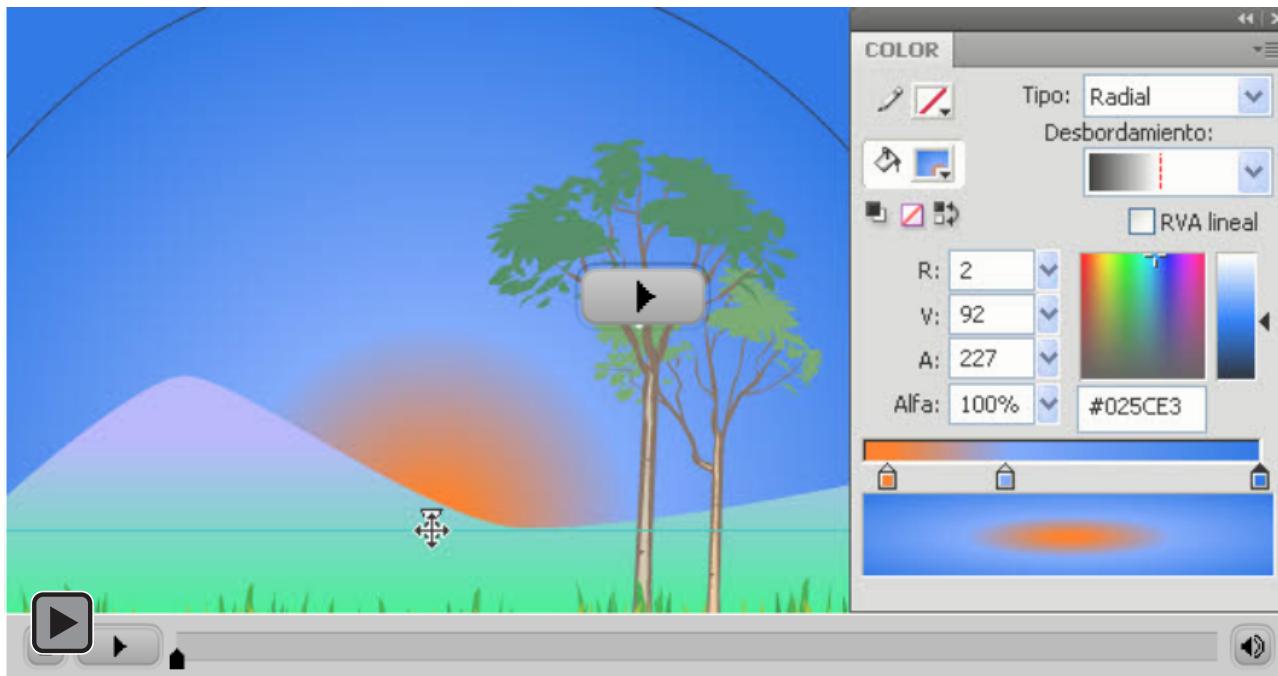
Per al cel dibuixarem un rectangle sense línia de traç (sense perfil) i amb un farciment degradat, encara que en aquest cas serà un degradat radial. Per a això tornarem a utilitzar el tauler *Color* (*Ventana > Color*). Per al traç seleccionarem **Tipo: Ninguno** o bé la franja vermella que indica que no hi ha color. Per al farciment seleccionem **Tipo: Radial**, i donarem a un extrem el valor #FF6600 i a l'altre, el valor #6699FF.



Activem l'eina **Rectángulo** i traçarem un rectangle que serà el cel, traspassant de nou els límits de l'escenari per dalt i per tots dos costats, i de tal manera que la part inferior quedi per darrere de les muntanyes, sense veure'n el límit.



Tutorial 2. Importació d'arxius externs  
**Pas 15 de 24**



Farem canvis en el degradat del cel. Amb l'eina **Selección (V)** seleccionem el cel per poder visualitzar els canvis a mesura que els fem. Desplaçem el cursor blau cap al taronja per fer més petit el cercle que representarà el sol. També desplaçem lleugerament el color taronja cap al centre. Fem clic en la barra de degradat per crear un nou color. Ens apareixerà el mateix blau que teníem, i l'enfosquirem lleugerament. Finalment, amb l'eina **Transformación de degradado** desplaçarem el centre del degradat fins a situar-lo darrere de la muntanya.

## Tutorial 2. Importació d'arxius externs Pas 16 de 24

En contrast amb el cel, veiem que la resta dels elements del paisatge tenen colors massa lluminosos, per la qual cosa els enfosqurem.

Primer, igual que hem fet en el pas 4 d'aquest tutorial, convertirem la muntanya en un símbol gràfic.  
Desbloquegem la capa *montaña* i bloquegem la resta de capes.  
Seleccionem la muntanya en l'escenari o bé el fotograma en la línia de temps.  
Premem **F8** o **Modificar > Convertir en símbolo**, seleccionem **Tipo: Gráfico** i l'anomenem *montaña*.

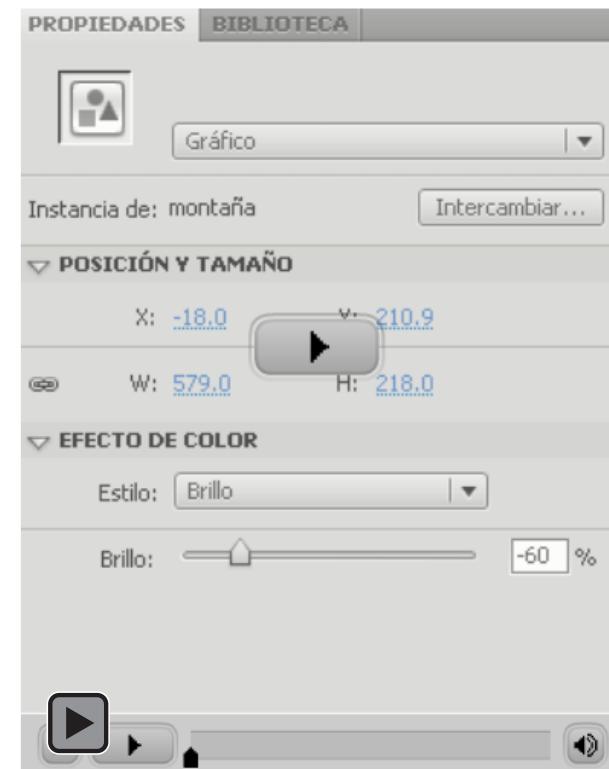


Amb la instància de la muntanya seleccionada en l'escenari, en l'inspector de **Propiedades** hi assignem un efecte de color d'estil **Brillo**, amb un valor de -60. D'aquesta manera enfosqurem el símbol sense necessitat de variar els colors originals. Podem provar diversos valors de brillantor fins que trobem un color que ens agradi.

Repetim el mateix procés amb els arbres i amb l'herba. Per a això haurem d'anar desbloquejant cada capa, seleccionant totes les instàncies, i assignant-los la mateixa brillantor negativa pel que fa als efectes de color.

També podem desbloquejar alhora les capes dels arbres i de l'herba, i incloure tots els elements amb l'eina *Selección*, i després aplicar l'efecte de *Brillo*, que afectaria tots els elements seleccionats.

Això no funcionaria si també tenim seleccionat el cel, ja que com que no és un símbol no admet aquest efecte. Per això és important mantenir bloquejada la capa que conté el cel.



## Tutorial 2. Importació d'arxius externs Pas 17 de 24

Abans d'afegir la lluna, que serà l'últim element del paisatge, farem un marc de referència per saber on són els límits de l'escenari, ja que ara apareixen coberts pel cel i les muntanyes.

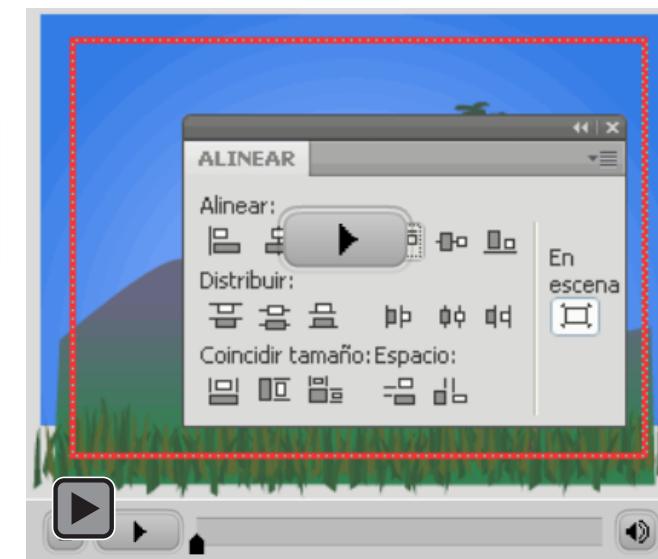
Igual que hem vist en altres passos, creem una capa nova anomenada *marco*, i aquesta vegada la situem per sobre de totes les altres. Bloquegem la resta de capes i ens assegurem que la capa *marco* estigui activa (mostra un llapis petit).

Després convertim aquesta capa en guia, la qual cosa significa que ens pot ajudar com a guia visual, però no apareixerà en l'arxiu que es publicui. Per fer-ho, fem clic amb el botó dret del ratolí sobre el nom de la capa, i seleccionem **Guía**. A l'esquerra del nom de la capa apareixerà una icona petita per indicar que aquesta capa ara és una guia.



Seleccionem l'eina **Rectángulo**, traç vermell i sense color de farciment, i tracem un rectangle en l'escenari, sense preocupar-nos de la mida o de la ubicació. Una vegada dibuixat, el seleccionem i en l'inspector de **Propiedades** hi donem els valors següents: X:0 – Y:0., perquè es posicioni en l'extrem superior esquerre de l'escenari, i W:550 -H:400, que són les mides de l'escenari.

També es podria arribar al mateix resultat amb el tauler **Alinear** (**Ventana > Alinear**), alineant-lo a l'escena tal com es veu en aquest vídeo.

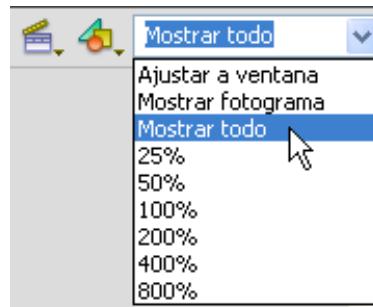


## Tutorial 2. Importació d'arxius externs Pas 18 de 24

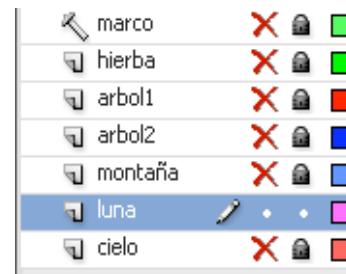


Creem una nova capa anomenada *luna*, que situarem entre la muntanya i el cel.

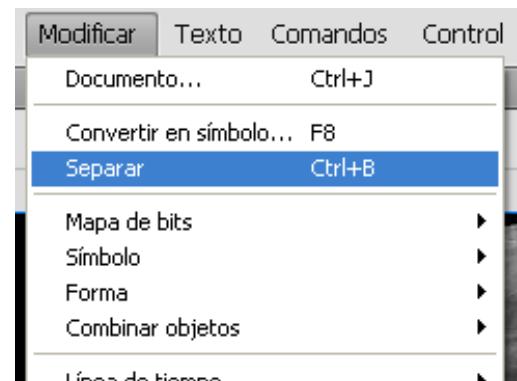
Importem l'arxiu de mapa de bits *luna\_llena.jpg* que hi ha en la carpeta *tutorial 2*. Per fer-ho, seleccionem **Archivo > Importar > Importar a escenario**.



Seleccionem **Mostrar todo** en la barra d'edició, per mostrar la lluna completament i ocupant tota l'àrea disponible.



Ocultem totes les capes excepte la de la lluna marcant els punts que hi ha al costat dels cadenats.

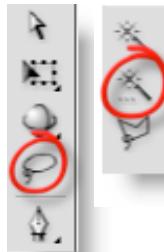


Amb la imatge de la lluna seleccionada, triem **Modificar > Separar**, o bé la premem amb el botó dret del ratolí i marquem **Separar** en el menú contextual.

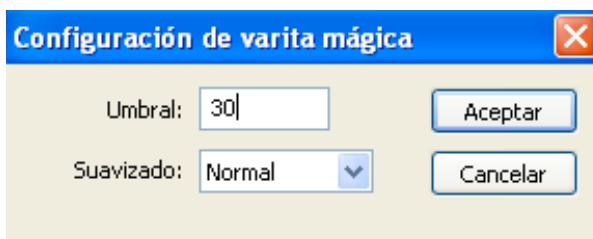
Premem qualsevol lloc de l'àrea de l'escenari però fora de la imatge per desseleccionar-la.

## Tutorial 2. Importació d'arxius externs Pas 19 de 24

Quan separem un mapa de bits, els píxels de la imatge es converteixen en àrees que es poden seleccionar i modificar separadament. Per tant, ara podem editar la imatge de la lluna.



Seleccionem l'eina **Lazo**, i en la part inferior de la barra d'eines apareixeran els modificadors associats a aquesta eina. Seleccionem en primer lloc **Propiedades de varita mágica**.



L'**umbral** es refereix a la diferència de color entre píxels adjacents, i el **suavizado** es refereix al nivell de detall de les vores de la selecció. Deixem un valor de llindar de 30 i suavitzat normal i premem **Aceptar**.



Seleccionem l'eina **Varita mágica**, i amb aquesta anem marcant les quatre grans àrees negres de la imatge. Després de cada selecció, premem **Supr** per esborrar aquestes àrees.



## Tutorial 2. Importació d'arxius externs

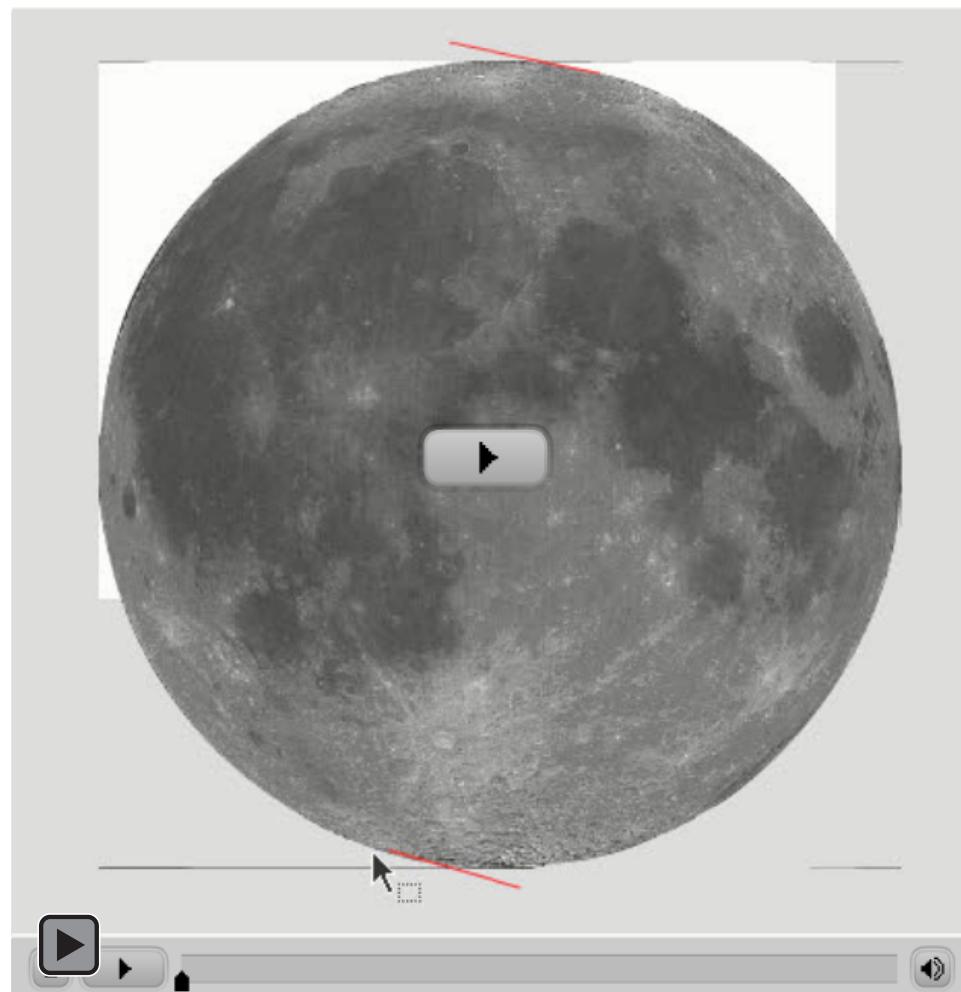
### Pas 20 de 24

Encara que ja gairebé tenim la lluna sense el fons, veiem que ens han quedat petites línies que hem d'eliminar. Per fer-ho, dividirem la imatge amb l'ajuda de l'eina **Línea**.

Dibuixem dues línies de tall per separar les petites franges negres que han quedat al costat de la lluna.

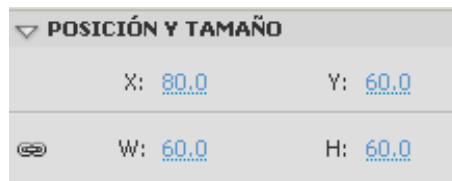
Després, amb l'eina **Selección**, anem marcant totes les àrees externes a la lluna i les esborrem amb la tecla **Supr**.

Finalment, esborrem també les línies que hem utilitzat per a efectuar els talls en la imatge. La manera més ràpida per a seleccionar completament un traç que està dividit en segments és fent doble clic sobre el traç. En aquest cas, les línies vermelles s'havien dividit en segments en entrar en contacte amb part del dibuix de la lluna, per això hi fem doble clic per seleccionar-les.

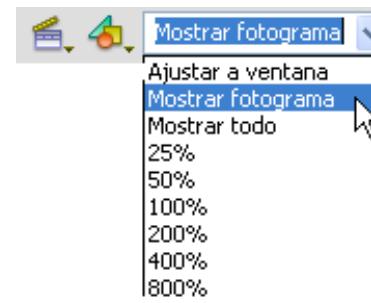


## Tutorial 2. Importació d'arxius externs Pas 21 de 24

Ara que ja tenim la lluna totalment separada del fons, la seleccionem i en el tauler **Propiedades**, en posició i mida, li donem els valors X: 80, Y: 60, W: 60 i H: 60.



Fem visibles les capes que teníem ocultes.

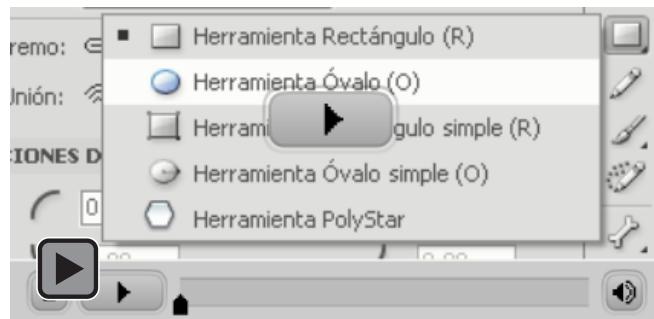


Canviem el grau d'augment de l'escenari seleccionant **Mostrar fotograma** en la barra d'edició.

Ara tindrem un paisatge similar al d'aquesta imatge. En els pròxims passos modificarem la lluna plena deixant només una petita porció de lluna creixent, ja que és l'aspecte que mostra quan es veu a prop de l'horitzó cap al tard. Finalment li donarem un toc de llum utilitzant un efecte de color avançat.

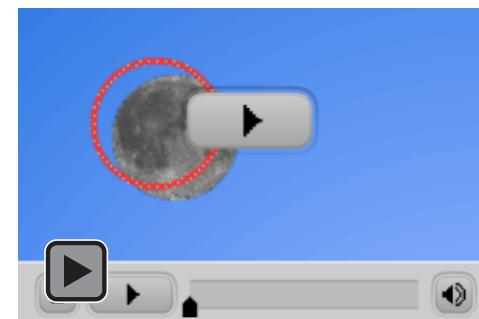


## Tutorial 2. Importació d'arxius externs Pas 22 de 24



En la mateixa capa que la lluna, a prop d'ella dibuixarem un petit cercle amb l'eina **Óvalo**, amb traç vermell i sense farciment. Perquè la forma sigui circular i no ovalada, mantindrem premuda la tecla *Mayúsculas* mentre el dibuixem.

Seleccionem aquest cercle, i en l'inspector de *Propiedades* hi donem un valor d'altura i amplada de 60, que és la mateixa mida que té la lluna.



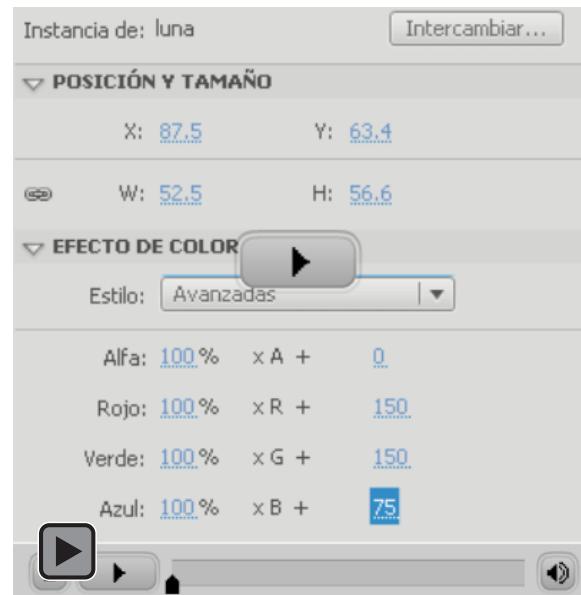
Arrosseguem el cercle fins a creuar-lo amb la imatge de la lluna, de tal manera que a la dreta de la imatge quedi el petit grill que volem mostrar.

Una vegada posicionat el cercle, seleccionem i esborrem amb **Supr** la part de la lluna que volem eliminar.

Finalment eliminem el cercle que hem utilitzat per a tallar la imatge.

## Tutorial 2. Importació d'arxius externs Pas 23 de 24

Finalment aplicarem efectes de color a la lluna, amb la qual cosa, com hem fet en passos anteriors, convertim la imatge de la lluna en un **símbol gràfic**.



Per il·luminar la imatge, en l'inspector de **Propiedades** seleccionem un efecte de color **Estilo: Avanzadas**.

Assignem un valor de desplaçament del vermell de 150, un desplaçament del verd de 150, i 75 en el desplaçament del blau.

**No oblideu desar el vostre treball!**

Tutorial 2. Importació d'arxius externs  
**Pas 24 de 24**



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

1. Col·loqueu una altra instància d'un dels arbres en l'escenari,  
i canieu-ne la mida i la posició.
2. Modifiqueu la forma de la muntanya amb l'eina *Subselección*.
3. Experimenteu amb diferents degradats per al cel i la muntanya,  
i també amb l'efecte de *Brillo*, per simular diferents hores del dia.

## Tutorial 3. Creació d'un logotip textual Pas 1 de 13

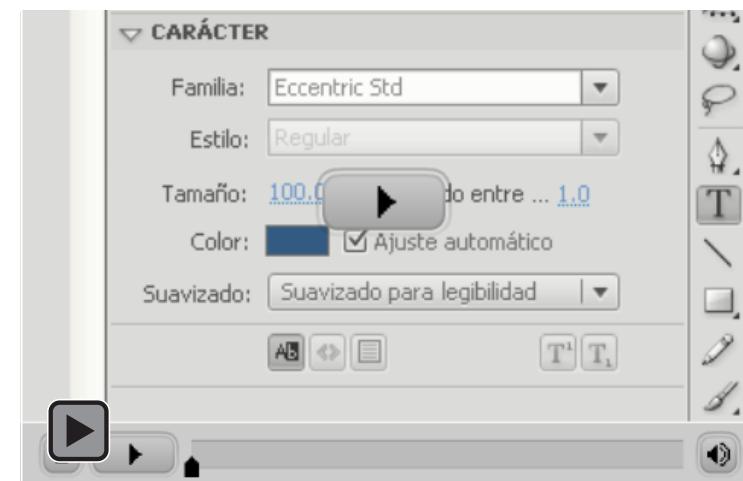


En aquest tutorial crearem un logotip basat en un text al qual aplicarem alguns efectes utilitzant filtres i mescles.

Per començar seleccionarem l'eina **Texto**. L'inspector de **Propiedades** mostrarà els diferents atributs d'aquesta eina. Modifarem alguns atributs del bloc **Carácter**.

En el primer desplegable que apareix en l'inspector de *Propiedades* seleccionem **Texto estático**. Els altres tipus de text (text dinàmic i introducció de text) els veurem en els tutorials de programació.

En el camp **Familia** seleccionarem **Eccentric Std**. Per a això, podem buscar el tipus de lletra en el desplegable o escriure directament les primeres lletres del nom. Si no disposem d'aquesta lletra en podem triar qualsevol altra. Després hi assignarem una **mida** de **100 pt** i un **color** blau fosc (**#003366**).



## Tutorial 3. Creació d'un logotip textual Pas 2 de 13

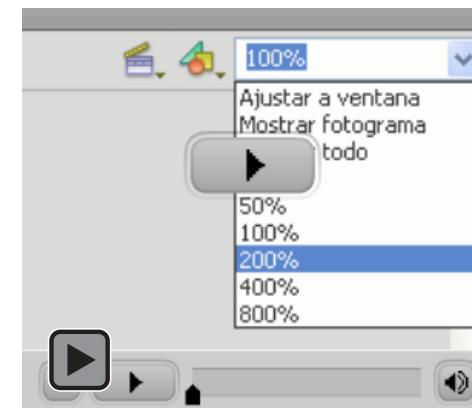
Amb l'eina **Texto** configurada, fem clic amb ella en l'escenari, i escrivim la paraula *logo*.



La paraula es mostrarà amb els atributs que hem seleccionat en l'inspector de *Propiedades*.

També seria possible canviar aquests atributs després d'haver creat el text. Per a això, seleccionem el camp de text que hi ha en l'escenari i fem les modificacions que vulguem en l'inspector de *Propiedades*.

Per poder editar el logo amb més comoditat, aproparem l'escenari seleccionant un **zoom** de **200%**.



Si l'escenari no ha quedat ben posicionat per a treballar després de fer el **zoom**, podem desplaçar-lo amb l'eina **Mano**.



## Tutorial 3. Creació d'un logotip textual

### Pas 3 de 13

El pas següent serà modificar la lletra *g*. Per a això hem de convertir prèviament el text en un gràfic editable.

Seleccionem el camp de text en l'escenari i seleccionem **Modificar > Separar**, o bé fem clic sobre el text amb el botó dret del ratolí i seleccionem **Separar** en el menú contextual.

Quan separem un text es crea un camp de text per cada caràcter. En aquest punt podríem canviar, per exemple, una lletra per una altra, triar un tipus de lletra diferent per a alguna lletra individual, posicionar cada lletra en diferents llocs, animar caràcters individuals, etc. Cada caràcter es trobarà, per tant, en un camp de text independent.

Si el que volem és manipular la forma d'una lletra, haurem de tornar a separar aquest camp de text. En aquest cas, com que només manipularem la forma de la lletra *g*, només separarem de nou aquesta lletra, mantenint la resta com a caràcters individuals.

Per fer-ho, seleccionem la lletra *g* i després **Modificar > Separar**.



## Tutorial 3. Creació d'un logotip textual Pas 4 de 13

Canviarem un petit detall de la lletra *g*. Amb l'eina **Selección** arrosseguem una cantonada per modificar-ne la posició, tal com es mostra en el vídeo.



El pas següent serà duplicar els camps de text que contenen la lletra *o*. Perquè la distància entre les lletres duplicates sigui la mateixa que entre les lletres originals, copiarem i enganxarem tots dos camps alhora.

Per a això, en seleccionem un, i amb la tecla **Mayúsculas** premuda, seleccionem l'altre. Fem clic amb el **botó dret** del ratolí i seleccionem **Copiar** del menú contextual. Després, en qualsevol punt de l'escenari, tornem a fer clic amb el botó dret del ratolí i seleccionem **Pegar**.

Ara tindrem seleccionats en l'escenari els dos camps que hem enganxat.

Tamaño: **50.0 pt**

En l'**inspector de Propiedades** assignem una **mida** de **50 pt** a aquests camps de text. Això no hauria estat possible amb la lletra *g*, ja que en tornar-la a separar ja no és un camp de text.

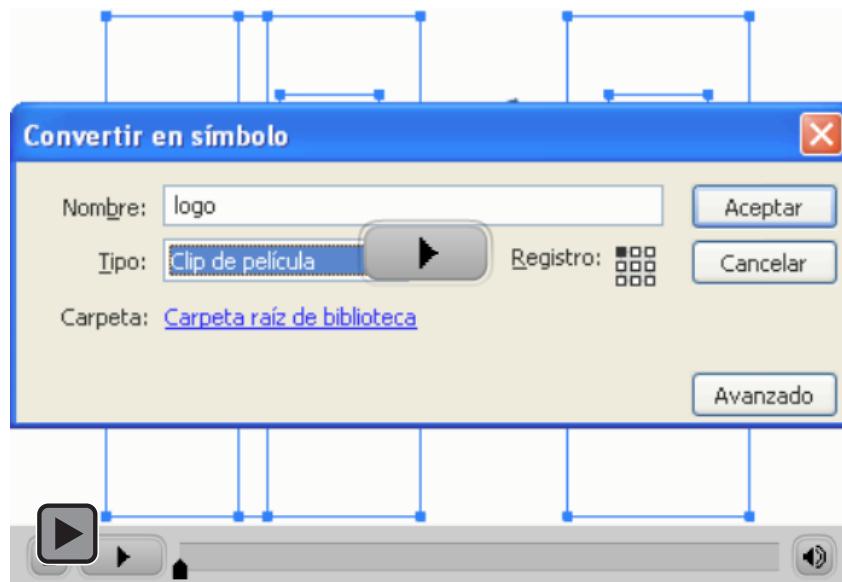
Finalment, amb tots dos camps encara seleccionats, arrosseguem aquestes còpies fins a situar-les dins de les lletres o originals. Per afinar millor la posició podem fer servir les fletxes del teclat.



## Tutorial 3. Creació d'un logotip textual Pas 5 de 13

Ara que ja hem fet la part superior del logo, ens queda fer el reflex de la part inferior. Per a crear aquest reflex utilitzarem una còpia del logo que ja hem creat, i a aquesta còpia hi afegirem efectes especials utilitzant **filtres i mescles**.

Per a poder afegir aquests efectes necessitem crear un **clip de pel·lícula**. L'ús d'un clip de pel·lícula també tindrà l'avantatge que, si decidim canviar el logo original, el seu reflex també es modificarà, ja que serà una instància del mateix clip de pel·lícula.



Seleccionem el logo abraçant-lo amb l'**eina Selección** o bé clicant sobre el fotograma que el conté en la línia de temps.

Després seleccionem **F8** o **Modificar > Convertir en símbolo**, i l'anomenem *logo*. Com a *Tipo* seleccionem **Clip de pel·lícula**.

Anomenem *superior* la capa actual i creem una **nova capa** que anomenem *reflejo*. Situem aquesta nova capa per sota de la capa superior i la seleccionem perquè sigui la capa activa.



## Tutorial 3. Creació d'un logotip textual

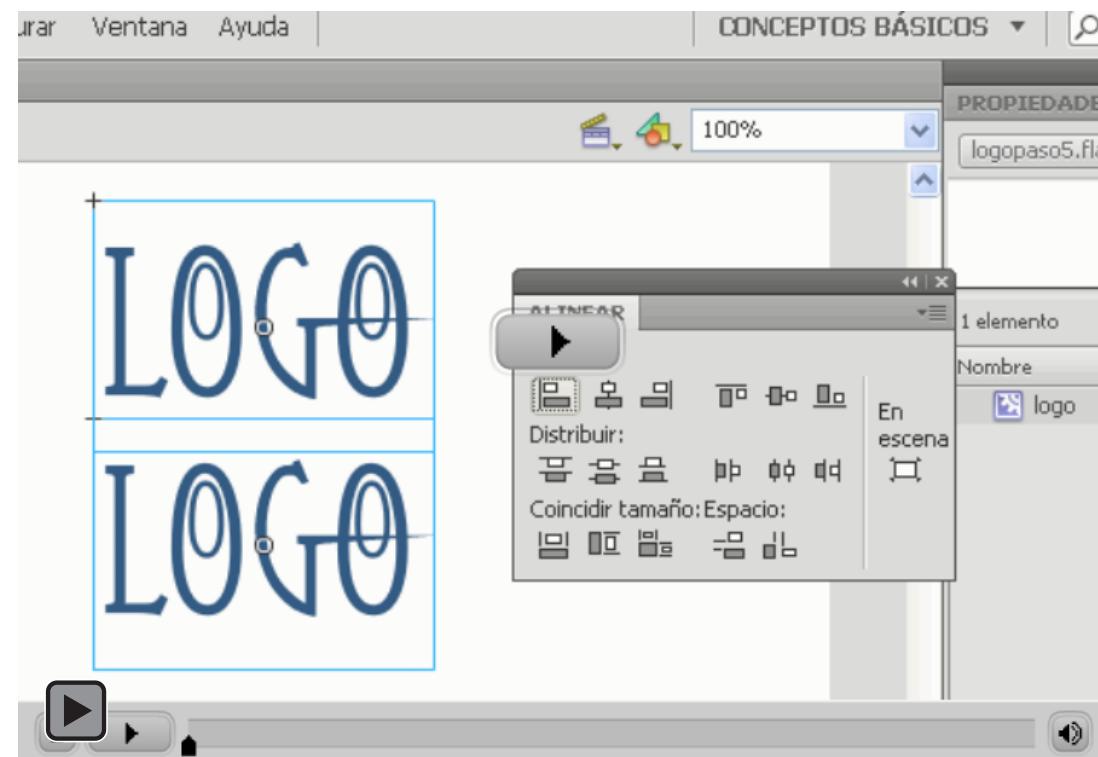
### Pas 6 de 13

Amb la capa **reflejo** activa, arrosseguem una **instància** del clip de pel·lícula *logo* des de la biblioteca a l'escenari.

Seleccionem **Ventana > Alinear**. Ens assegurem que el botó **En escena** no estigui seleccionat.

Seleccionem la instància del logo que hi ha en la capa superior. Recordem que per a fer una selecció múltiple hem de tenir la tecla **Mayúsculas** premuda, i les capes on hi ha els elements que seleccionarem no han d'estar bloquejades.

En el tauler **Alinear** seleccionem **Alinear borde izquierdo**.



## Tutorial 3. Creació d'un logotip textual

### Pas 7 de 13

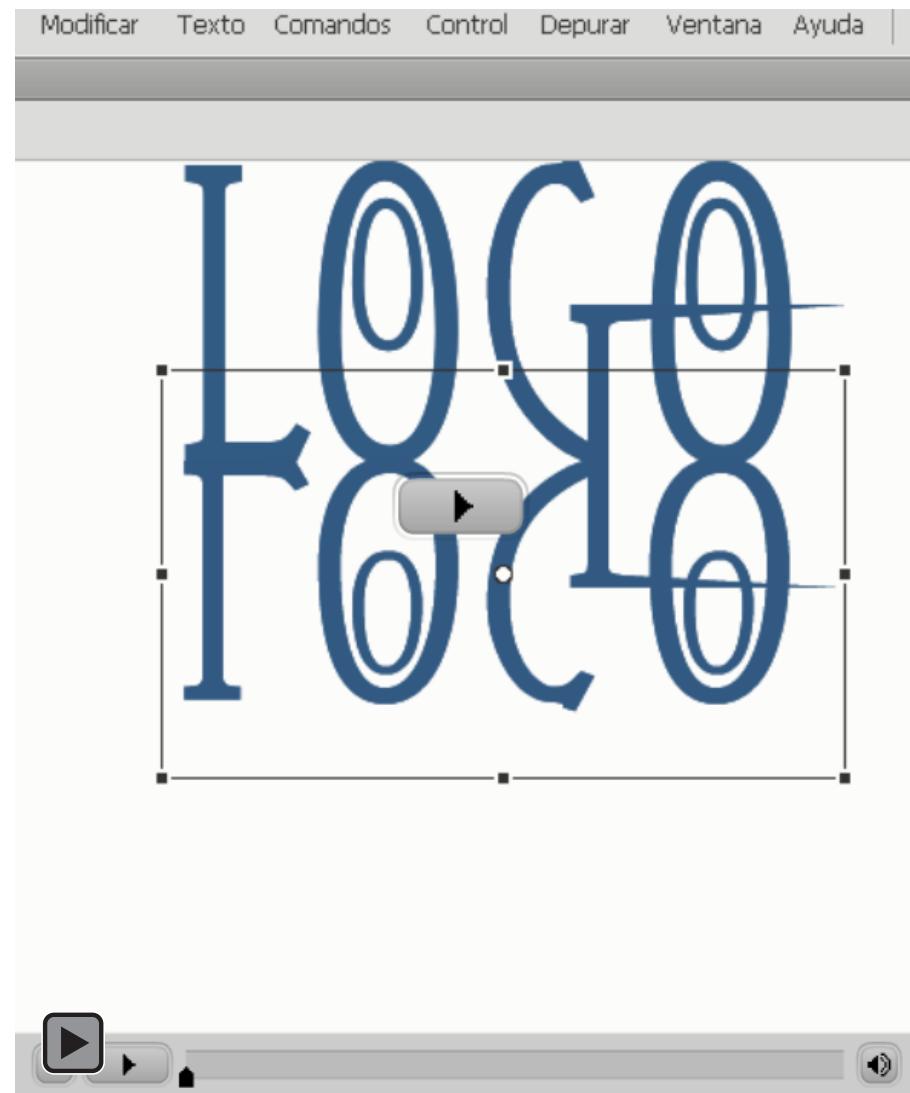


Seleccionem la instància del logo corresponent al reflex amb l'eina **Transformación libre** i en reduïm lleugerament l'altura.

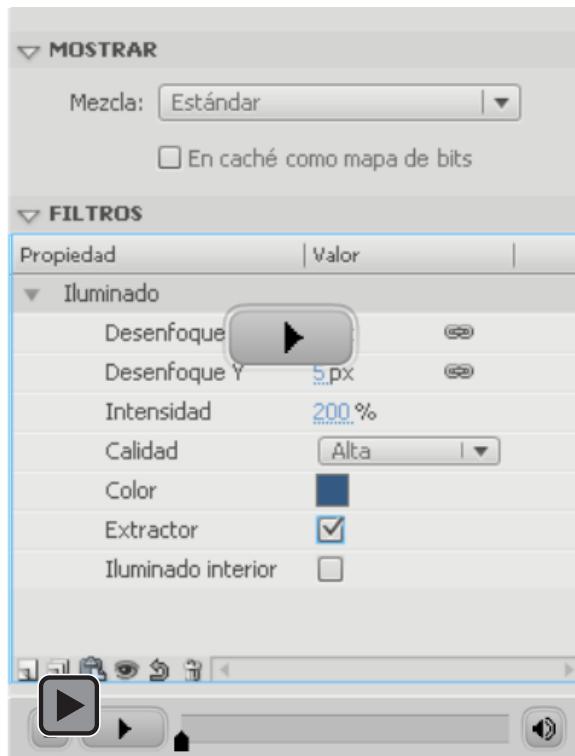
Seleccionem **Modificar > Transformar > Voltear verticalmente**.

Ajustem la posició vertical del reflex amb l'ajuda de les **fletxes amunt i avall del teclat**. Podem mantenir una fletxa premuda per a avançar ràpidament, i després podem ajustar-ne la posició amb petites pulsacions.

Pugem la instància del reflex fins a eliminar l'espai entre la base de la lletra / i el seu reflex.



## Tutorial 3. Creació d'un logotip textual Pas 8 de 13



Afegeirem un **filtre** a la instància del reflex. Els filtres es poden afegir a camps de text, botons i clips de pel·lícula. Si no haguéssim separat el text, també podríem haver afegit un filtre sense necessitat de convertir-lo prèviament en clip de pel·lícula.

Seleccionem la instància del reflex en l'escenari.

Seleccionem **Añadir filtro** en l'àrea de **Filtros** de l'**inspector de Propiedades**. S'obrirà un menú desplegable que mostrarà els filtres disponibles. Afegirem un filtre de tipus **Iluminado**.

Efectuarem alguns canvis en els atributs d'aquest filtre. Per començar assignem una **intensitat del 200%**. Seleccionem **qualitat alta**. Modifiquem el color seleccionant el mateix color que tenia el logo (#003366). Finalment, marquem la casella **Extractor** per mostrar només l'il·luminat.

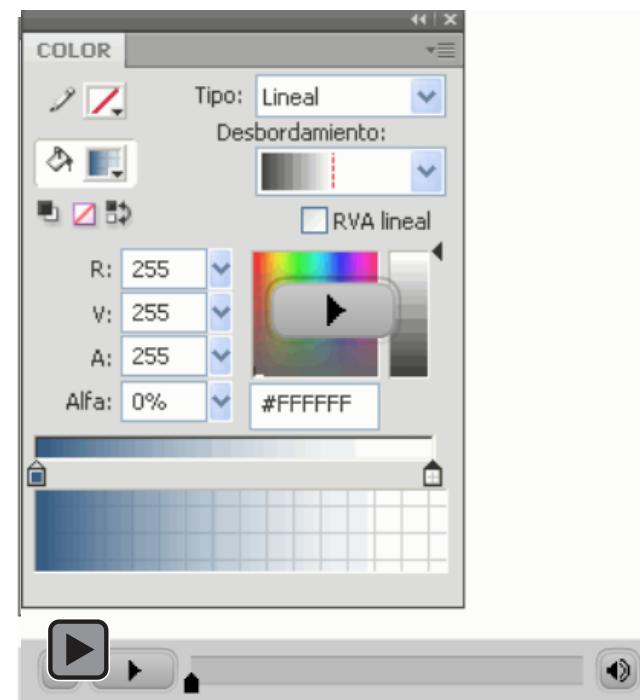
## Tutorial 3. Creació d'un logotip textual

### Pas 9 de 13

Per a dibuixar el rectangle on sembla que recolza el logo creem una **nova capa**, que anomenarem *base*, i que situarem per sota de les altres dues capes.



Obrim el tauler **Color** (**Ventana > Color**). Eliminem el traç i per al farciment seleccionem un **degradat lineal** amb un extrem amb el color blau que estem utilitzant en aquest tutorial (#003366) i amb un altre extrem transparent (**alfa 0**). D'aquesta manera el color blau es tornarà transparent progressivament.



### Tutorial 3. Creació d'un logotip textual

#### Pas 10 de 13

En la capa *base* dibuixem un petit rectangle amb l'eina **Rectángulo**, que apareixerà amb les opcions de color que havíem seleccionat en el pas anterior.

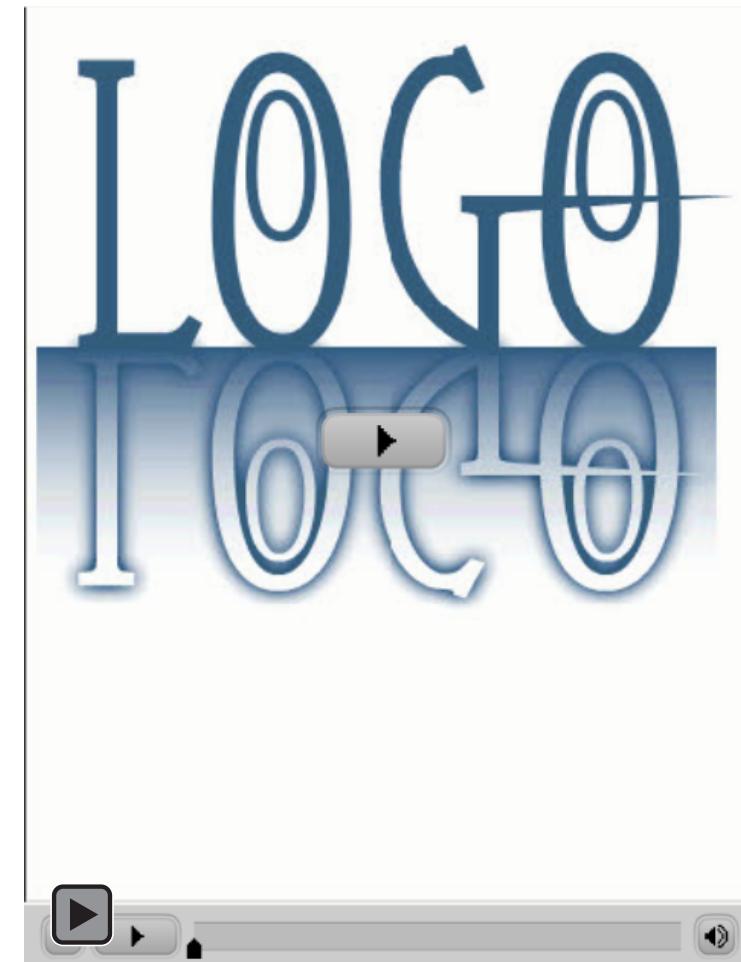


Amb l'eina **Transformación de degradado** modifiquem el degradat lineal, de tal manera que el color blau quedi a dalt i el color transparent a baix. També canviarem l'altura del degradat perquè coincideixi amb la del rectangle.

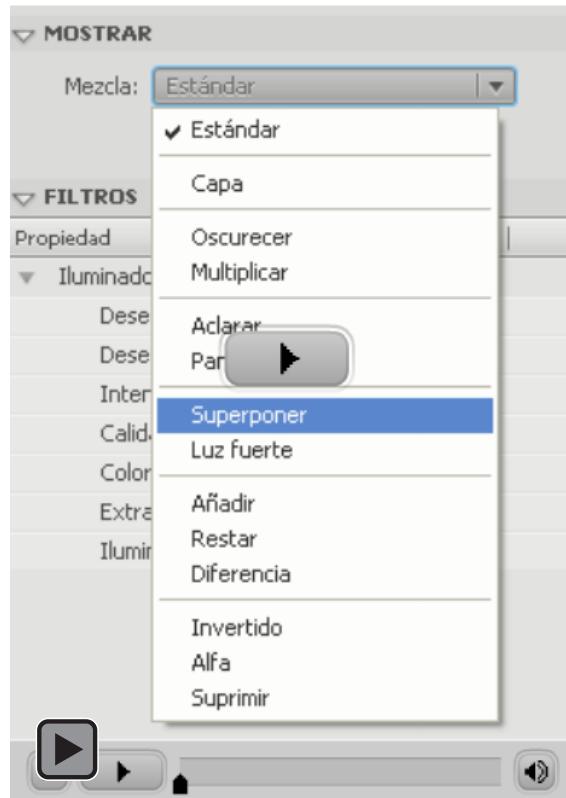


Després, amb l'eina **Transformación libre**, situem el rectangle i hi donem les proporcions adequades.

Recordeu que totes dues eines s'agrupen en el mateix lloc de la barra d'eines. Si teniu algun dubte sobre el seu ús repasseu el tutorial anterior.



## Tutorial 3. Creació d'un logotip textual Pas 11 de 13



Per acabar el logo, crearem una imatge composta entre el reflex i el rectangle de la base.

Seleccionem en l'escenari la imatge reflectida del logo. Podem bloquejar les altres capes per a assegurar-nos que estem seleccionant el reflex.

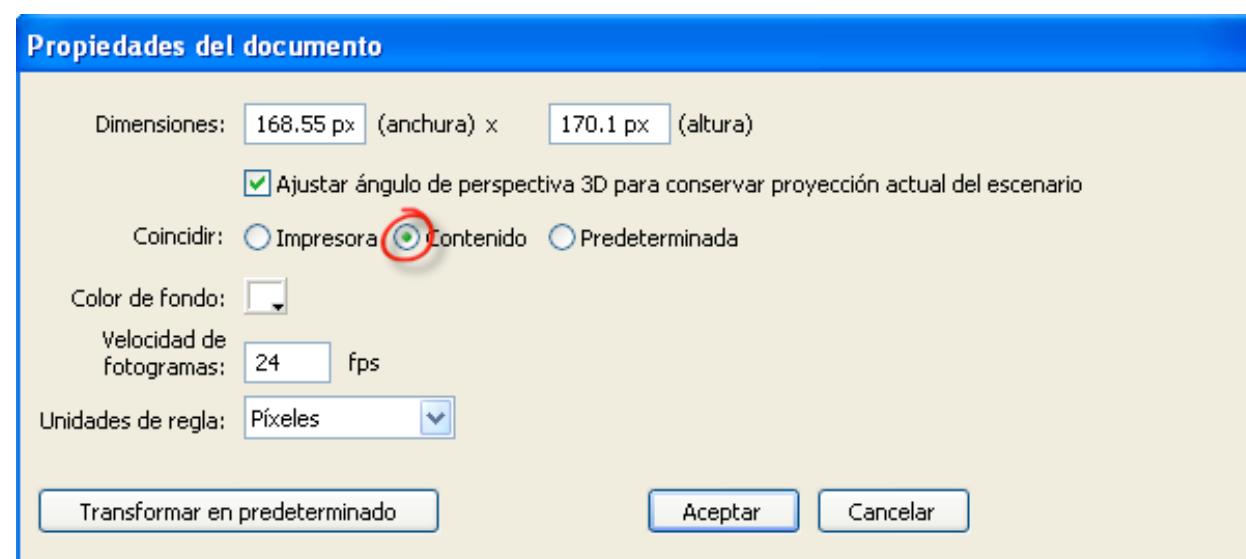
En l'**inspector de Propiedades**, en l'àrea **Mostrar**, seleccionem **Mezcla > Superponer**.

## Tutorial 3. Creació d'un logotip textual Pas 12 de 13



Ara canviarem la mida de l'escenari, ja que ens ha quedat massa gran en comparació amb el logotip. Seleccionem un lloc buit de l'escenari perquè l'**inspector de Propiedades** ens mostri la informació sobre la nostra pel·lícula. Premem el botó **Editar** que hi ha al costat de la mida de l'arxiu.

En el tauler de **Propiedades del documento** seleccionem **Coincidir: Contenido**. D'aquesta manera el document adaptarà automàticament la mida a la del nostre logo.



Deseu l'arxiu com a *tutorial3.fla*. Segons l'ús que volguéssim donar al logotip, podríem desar, per exemple, la imatge seleccionant **Archivo > Exportar > Exportar imagen**, i en **Tipo** seleccionar **PNG**.

Tutorial 3. Creació d'un logotip textual

**Pas 13 de 13**



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

1. Feu alguna altra modificació en les lletres del símbol original del logotip, i observeu com aquest canvi s'aplica tant al logotip com al seu reflex.
2. Experimenteu amb diferents tipus de filtres i de valors.
3. Proveu els efectes dels diferents tipus de mescles.

## Tutorial 4. Animació del logotip

### Pas 1 de 13

Per introduir-nos en el camp de les animacions, comencem fent una animació senzilla partint del logotip que hem creat.



Per a això, en primer lloc obrirem l'arxiu *tutorial3.fla*, que havíem desat en el tutorial anterior, i el desem com a *tutorial4.fla*. D'aquesta manera mantindrem intacte l'arxiu del tutorial 3 per si necessitem tornar-hi en algun moment.

En el Flash hi ha diferents maneres de crear animacions, i cada una ofereix diferents possibilitats:

- **Interpolacions de moviment.** Si en diferents fotogrames assignem diferents valors a una o diverses propietats d'un objecte com, per exemple, la posició, l'escala, la transparència, etc., el Flash calcularà els valors intermedis entre tots dos fotogrames, de tal manera que el canvi sigui gradual. És el tipus d'interpolació que crearem en aquest tutorial.

- **Interpolacions clàssiques.** Són la manera en què es creaven interpolacions en les versions anteriors del Flash. Són similars en diversos aspectes a les interpolacions de moviment, encara que són més difícils de crear i manipular.

- **Poses de cinemàtica inversa.** Són animacions mitjançant ossos que tractarem en el tutorial següent.

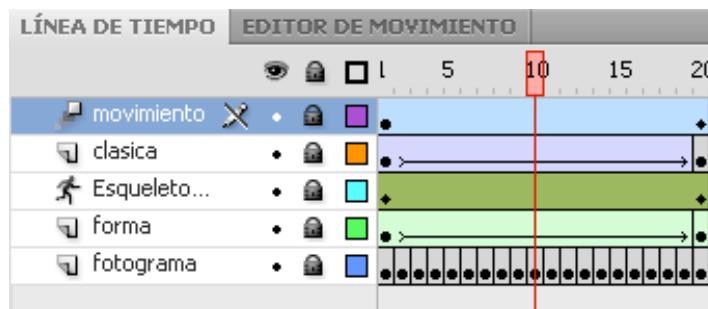
- **Interpolacions de forma.** Són animacions en les quals passem d'una forma a una altra de diferent. Igual com passa en un altre tipus d'animacions, el Flash crea els passos intermedis entre dos fotogrames, en aquest cas entre formes diferents.

- **Fotograma a fotograma.** El Flash mostra simplement cada fotograma en un moment en el temps, per la qual cosa, com en el cinema, es pot crear la sensació d'animació, ja que cada fotograma mostra un contingut diferent. És la manera més laboriosa de fer animacions.

## Tutorial 4. Animació del logotip

### Pas 2 de 13

Aquesta és la manera en què el Flash mostra en la **línia de temps** els diferents tipus d'animacions:



Llevat del cas de l'animació fotograma a fotograma, en la resta només hem indicat el primer i l'últim fotograma, i és el Flash el que ha creat automàticament els passos intermedis (interpolacions).

En aquest cas podem veure que totes les animacions tenen una durada de 20 fotogrames. La velocitat de les animacions variarà segons a quants fotogrames per segon tinguem configurat el nostre document. Si el tinguéssim configurat a 20 fps, aquestes animacions durarien exactament 1 segon.

També podem veure que, en aquest moment, el capçal lector és en el fotograma 10.

Començant ja amb la nostra animació, seleccionem la instància del **clip de pel·lícula logo** que hi ha en la part superior, fem clic amb el **botó dret del ratolí**, i seleccionem **Crear interpolación de movimiento**.



Altres maneres de crear aquesta interpolació és seleccionant **Insertar >Interpolación de movimiento** o bé seleccionant el fotograma en la línia de temps, i també fent clic amb el **botó dret del ratolí sobre el fotograma** en el qual hi ha la instància i seleccionant **Crear interpolación de movimiento**.

Per a poder fer aquest tipus d'interpolacions hem de tenir cada objecte que vulguem animar en una capa different. En aquest cas ja teníem el logo distribuït en capes, per la qual cosa no tindrem cap problema.

## Tutorial 4. Animació del logotip

### Pas 3 de 13

Després de crear la interpolació, veiem que s'han creat 24 fotogrames en la capa en la qual teníem la part superior del logo. El nombre de fotogrames que s'afegeixen automàticament depèn dels fotogrames per segon a què tinguem configurat el document (24 fps per defecte).

El cap lector s'ha desplaçat també al fotograma 24. En l'escenari només veurem la part superior del logo, ja que les altres dues capes només es mostren en el primer fotograma.

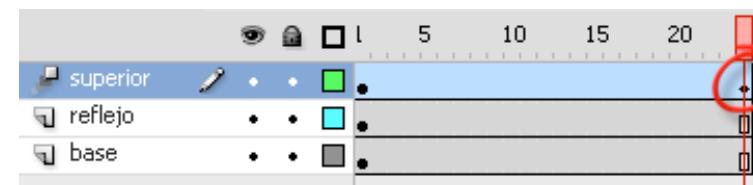
Per a poder veure la resta de les capes, hem d'inserir fotogrames perquè es mantinguin visibles fins al fotograma 24. Per fer-ho, ens situem en el fotograma 24 de cada capa i seleccionem **Insertar > Línea de tiempo > Fotograma**, o bé premem **F5**.



Tornem a seleccionar la capa superior. Tenint el cap lector en el fotograma 24, si ara canviéssim el logo de posició es crearia automàticament una interpolació de moviment des de la posició d'inici a la posició nova.

Tanmateix, en aquest cas volem que l'animació acabi amb el logo en la posició actual, per la qual cosa el procés consistirà a crear, en primer lloc, un **fotograma clau** per a mantenir la posició actual en el fotograma 24.

Per fer-ho, seleccionem **Insertar > Línea de tiempo > Fotograma clave**. Apareixerà un rombe petit en el fotograma 24 de la primera capa que indica que s'ha creat un **fotograma clau de propietat**, és a dir, que en aquest fotograma s'emmagatzemen els valors actuals de les diferents propietats de la instància del logo.



## Tutorial 4. Animació del logotip

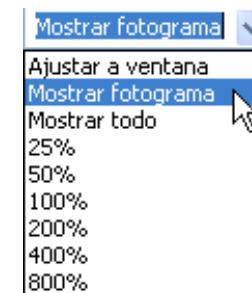
### Pas 4 de 13



**Desplaçem el cap lector al primer fotograma**, i baixem la posició del logo fins que quedi per sota de la base. Per a desplaçar-lo amb facilitat i assegurar-nos que només variem l'altura, utilitzarem la **fletxa avall del teclat**.

De moment, el logo no s'ocultarà després de la base, sinó que es veurà per davant d'aquesta, però això ho solucionarem més endavant.

Per a poder definir la posició amb més detall, podem ampliar la mida de l'escenari seleccionant **Mostrar fotograma**.

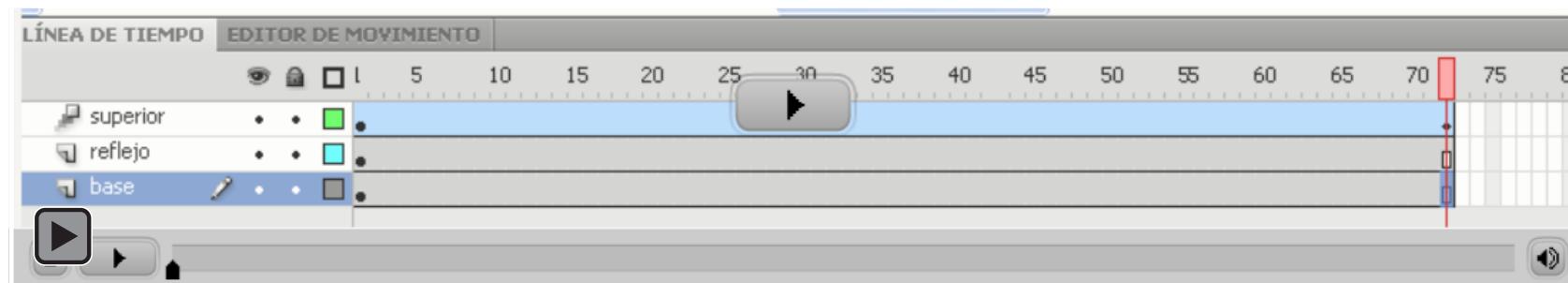


A mesura que desplaçem la posició vertical del logo, veurem com en l'escenari apareix una línia amb punts en el seu recorregut. Els punts inicial i final de la línia mostren la posició del logo en els fotogrames clau d'inici i final de l'animació, i els punts intermedis representen les posicions que tindrà el logo en cada fotograma de l'animació.

Si desplaçem el cap lector podrem veure el recorregut que realitza el logo al llarg dels fotogrames. Si volem veure l'animació en temps real, ens posicionem al fotograma 1 i premem la tecla **Intro**.

## Tutorial 4. Animació del logotip

### Pas 5 de 13



En visualitzar l'animació en temps real, veiem que va massa ràpida, per la qual cosa n'allargarem la durada a 72 fotogrames (3 segons). Per fer-ho, apropiem el cursor a l'últim fotograma de l'animació i simplement premem i arrosseguem el fotograma fins a la nova posició.

Veurem que apareixen més punts en la línia que mostra la interpolació, ja que ara l'animació transcorre al llarg de més fotogrames.

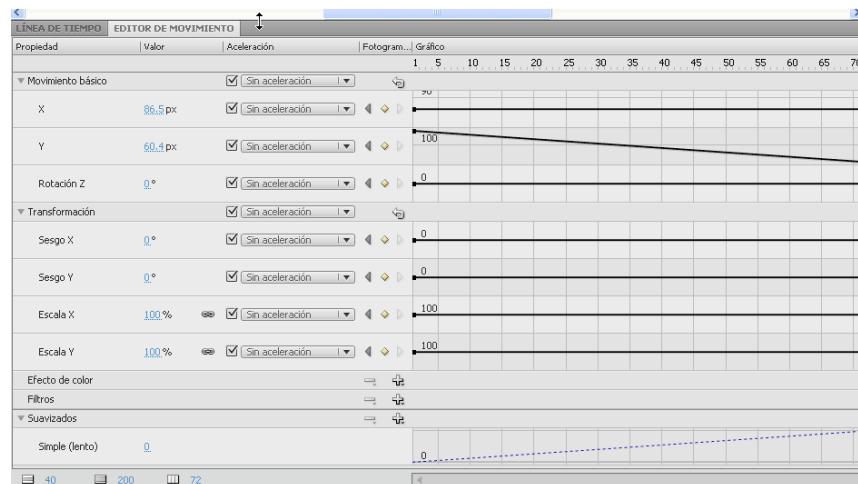
Afegim fotogrames a les altres dues capes tal com hem fet en el pas 3, és a dir, seleccionant el fotograma buit corresponent i prement **F5**, ja que, com que no és una interpolació de moviment, no podem estendre directament la durada de la capa.

De moment, la part superior del logo es desplaça amb velocitat uniforme des de la posició inicial, sota la línia superior de la base, fins a la posició final en la part superior, i triga 3 segons a fer aquest recorregut.

En els passos següents, amb ajuda de l'editor de moviment, transformarem aquest moviment uniforme en un moviment que simularà un petit rebot en la part superior.

Amb el logo superior o la línia verda seleccionada, fem clic sobre la pestanya **Editor de movimento**, que hi ha al costat de la pestanya **Línea de tiempo**.

## Tutorial 4. Animació del logotip Pas 6 de 13



Per a poder visualitzar una àrea més àmplia d'aquest tauler, podem arrosseggar la part superior (la part superior de la barra gris fosc).

En la part inferior del tauler podem seleccionar diferents valors per a visualitzar els diferents elements del tauler. Per a poder veure tota l'evolució de l'animació al llarg de la línia de temps, a **tamaño de gráfico** hi donarem un valor de 20, i a **fotogramas visibles** un valor de 72.

En la columna **Propiedad** podem veure les diferents propietats que podem modificar en una interpolació de moviment (*X*, *Y*, *Rotació Z*, *Sesgo Y*, etc.), agrupades en diferents categories.

En la columna **Valor** veiem els valors de les diferents propietats en el fotograma on hi ha el cap lector.

La columna **Aceleración** mostra si la interpolació de la propietat corresponent té assignada algun tipus d'acceleració.

En la columna **Fotogramas** podem afegir o eliminar fotogrames clau, i també desplaçar-nos entre ells.

L'àrea principal, la corresponent a **Gráfico**, mostra l'evolució dels valors de cada propietat al llarg del temps. En aquest cas podem veure que tot són línies rectes excepte la propietat *Y*, que mostra una línia descendente. Això significa que la propietat *Y* (altura) del logo comença amb un valor alt, que va disminuint segons avança l'animació (els valors més baixos de *Y* corresponen a posicions més altes en l'escenari). Les altres propietats no presenten cap modificació.

## Tutorial 4. Animació del logotip

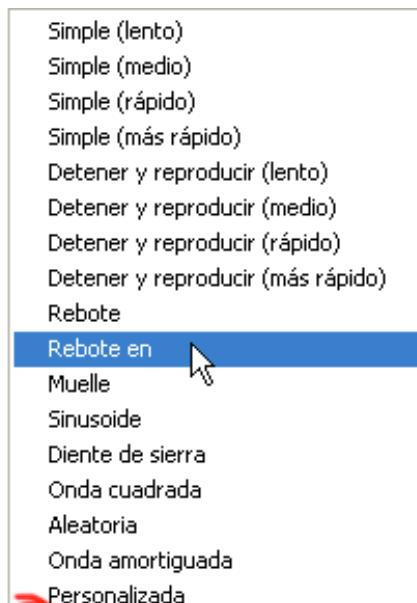
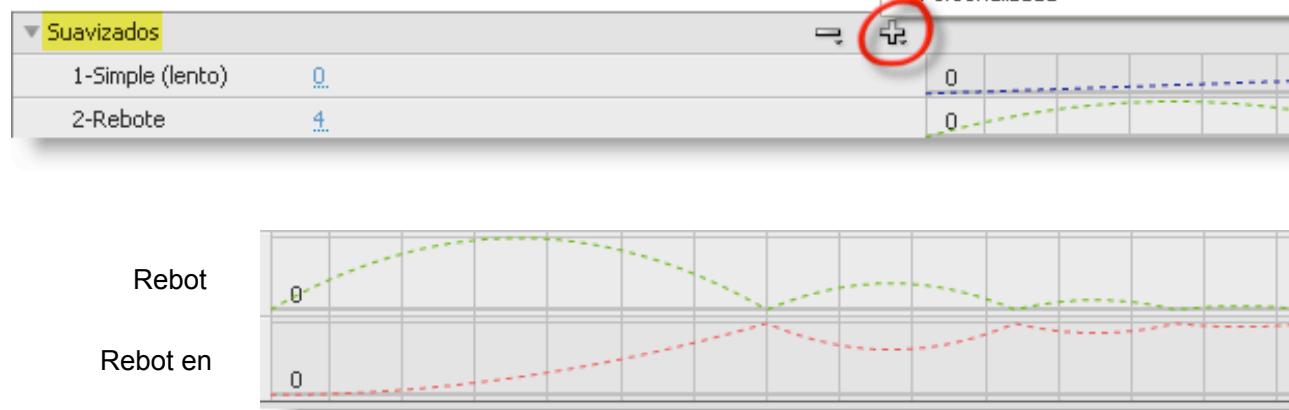
### Pas 7 de 13

El pas següent serà afegir una acceleració en la propietat Y.

Per fer-ho, primer hem d'afegir en l'àrea de **Suavizados** l'acceleració que utilitzarem.

Disposem de diverses corbes d'acceleracions ja creades, i a més podem afegir les nostres pròpies acceleracions personalitzades.

En aquest cas afegim, en primer lloc, una acceleració de tipus **Rebote**, i després una altra de tipus **Rebote en** per examinar-ne les diferències i veure quina ens interessa utilitzar. Podem premer el nom de les corbes que hem afegit per visualitzar-les amb més detall.



En l'àrea de **Suavizados**, el valor més baix de cada corba (0) indica el valor de la propietat en el fotograma inicial de l'animació, i el valor més alt (100) indica el valor de la propietat en el fotograma final.

En el cas de la corba **Rebote**, veiem que l'animació passa del valor del primer fotograma a l'últim per tornar al valor inicial, i l'animació acaba, després d'uns quants rebots, en el valor que tenia la propietat en el primer fotograma.

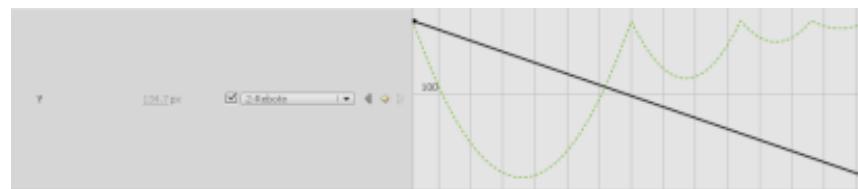
En el cas de **Rebote en**, l'animació rebota en el valor de l'últim fotograma, i és en aquest on acaba. Per tant, és aquest tipus de corba el que ens interessa per a la nostra animació.

## Tutorial 4. Animació del logotip

### Pas 8 de 13

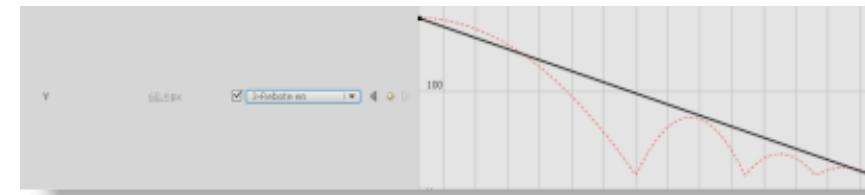
Per a entendre millor el significat de les corbes d'acceleració, aplicarem tots dos tipus de corbes en la nostra animació.

En la fila corresponent a la **proprietat Y**, seleccionarem **Rebote** en el desplegable de l'acceleració. Veiem que en el desplegable només apareixen les acceleracions que hem afegit en l'àrea de suavitzats.



Per veure la corba amb més detall premem el nom de la propietat Y (podem tornar a premer el nom per a tornar-la a visualitzar en una mida reduïda).

En aquest cas veiem que l'animació comença amb el valor inicial, després baixa al valor final, després puja al valor inicial, baixa lleugerament, etc., fins a acabar en el valor inicial. Premem la tecla **Intro** per veure el resultat d'aquesta animació. El resultat és que el logo puja per finalment tornar a caure.



Ara seleccionem **Rebote en** en el desplegable de l'acceleració de la propietat Y. Aquí podem veure com l'animació rebota i acaba en la posició final. Premem de nou la tecla **Intro** per veure l'animació resultant i veiem que ja s'assembla al resultat que volem aconseguir.

Si volem canviar la velocitat del rebot, podem assignar diferents valors a l'acceleració en l'àrea de suavitzat. Per exemple, assignarem un valor de 10 a **Rebote en** en l'àrea de **Suavizados**. Proveu amb diferents valors per veure les diferències en la corba i, per tant, en l'animació.



Si necessitem ajustar la posició inicial o final del logo, podem fer-ho directament des de l'àrea **Gràfico** de la propietat Y, arrossegant directament la posició dels petits quadrats negres que representen els fotogrames clau.

## Tutorial 4. Animació del logotip

### Pas 9 de 13

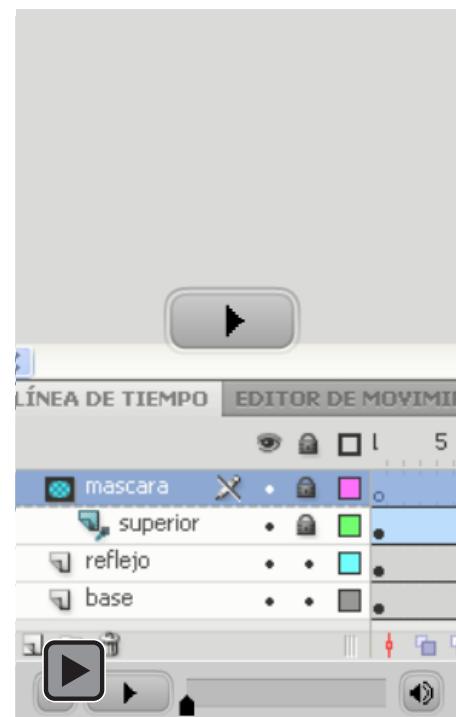
En aquest pas ocultarem el logo després de la base, de tal manera que només es vegi la part del logo superior que quedí per sobre de la base, i per a això utilitzarem una **màscara**.

Una màscara és una capa que determina quines àrees de les capes que té associades per sota són visibles. Les zones farcides

de la màscara corresponen a les zones visibles de les capes associades; allà on la màscara és buida, el contingut de les capes associades no és visible.

En primer lloc, creem una nova capa que anomenem *màscara* (pot ser un altre nom).

Fem clic amb el **botó dret** del ratolí sobre el nom de la nova capa i seleccionem **Màscara** en el menú contextual.



Automàticament la icona de la capa canviarà per mostrar que es tracta d'una capa tipus *Màscara*, i s'hi associarà la capa immediatament inferior. Podríem arrosseggar més capes sota la màscara perquè també s'hi associessin, encara que en aquest cas no ho farem.

Desbloquegem la capa *màscara* fent clic sobre la icona del cadenat. En aquesta nova capa dibuixem un rectangle sense traç i amb farciment vermell que ocupa exactament la part superior, és a dir, la part en la qual volem que es mostri la capa superior, que és la que conté l'animació que hem fet fins ara.



Per a veure l'efecte de la màscara que hem creat, hem de bloquejar de nou tant la màscara com les capes que hi estiguin associades.

Si premem **Intro**, veiem que només es mostra el logo superior quan és en l'àrea que havíem definit amb el rectangle vermell.

## Tutorial 4. Animació del logotip

### Pas 10 de 13

Ara farem l'animació del reflex de manera similar a com hem animat la part superior del logo.

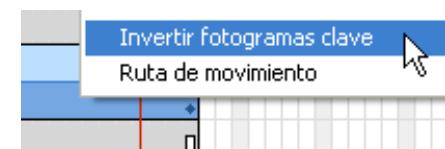
Bloquegem totes les capes excepte la capa *reflejo*, i fem clic sobre el reflex en l'escenari amb el **botó dret** del ratolí. Seleccionem **Crear interpolación de movimiento** en el menú contextual.



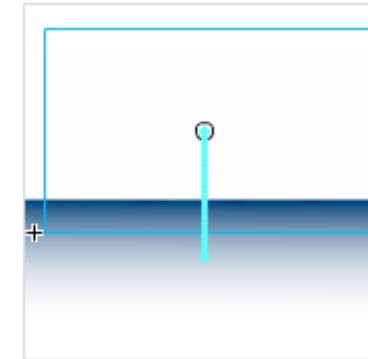
En aquest cas veiem que l'animació ocupa automàticament els 72 fotogrames que ocupava aquesta capa. Per tant, si una capa disposa d'un nombre de fotogrames més gran que 1, les animacions que creem en aquesta capa ocuparan per defecte el nombre de fotogrames disponibles per a aquesta capa, i no el nombre d'fps del document.

En el fotograma 72 pugem el reflex en l'escenari amb ajuda de la **fletxa amunt** del teclat, fins que desaparegui completament per sobre de la base. Automàticament es crearà un **fotograma clau de propietat**.

Després seleccionem la interpolació en la línia de temps i, prement el **botó dret** del ratolí, seleccionem **Invertir fotogramas clave**. Ara el reflex començarà invisible a dalt i acabarà a baix. Aquesta és una altra manera d'aconseguir que el primer fotograma passi a ser l'últim de l'animació.



En aquest cas el reflex desapareix sense utilitzar una màscara, ja que, en el tutorial anterior, per a aquesta capa havíem usat la mescla **Superponer**, i això fa que no sigui visible en un fons blanc.



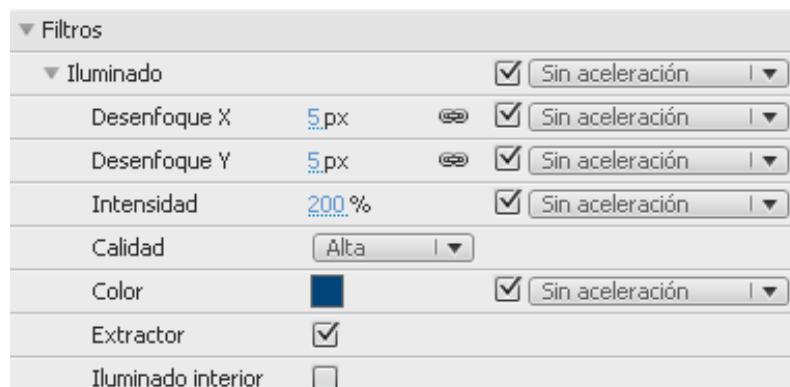
## Tutorial 4. Animació del logotip

### Pas 11 de 13

Si premem la tecla **Intro** per provar l'animació, veurem que el reflex va de dalt a baix en l'escenari de manera uniforme.

Per afegir el mateix efecte que en la part superior, tornarem a fer els mateixos passos que hem fet per animar la part superior.

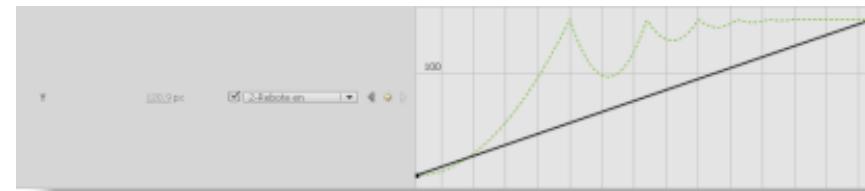
Seleccionem el reflex en l'escenari i obrim **l'Editor de movimiento**. En observar aquest tauler podem veure que hi ha alguns canvis respecte a l'animació anterior. Per exemple, ara apareix un valor en negatiu en la propietat **Escala Y** que indica que hem invertit el logotip. També apareixerà el filtre **Iluminado** que havíem aplicat, i que també podríem animar.



Afegim **Rebote en** primer en l'àrea de **Suavizados**. Hi donem un valor de 10 (o el valor que hagim decidit donar-hi en el pas 8 en la part superior del logo). En el desplegable **Aceleración** de la propietat **Y** seleccionem **Rebote en**.

En aquest cas veiem que la forma que adopta la corba en la propietat **Y** és diferent, ja que comença amb un valor de **Y** baix, i acaba amb un valor alt.

Per a més precisió, recordem que podem ajustar les posicions de principi o de final de l'animació desplaçant els quadres negres que representen els fotogrames clau d'inici i final de l'animació.



## Tutorial 4. Animació del logotip

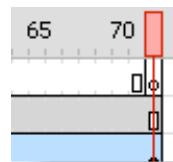
### Pas 12 de 13

Seleccionem **Control > Probar película** per provar la nostra animació. Aquesta opció genera automàticament un arxiu **swf** en la mateixa carpeta en la qual tinguem el nostre arxiu **fla**.

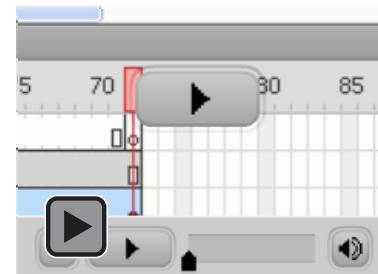
En l'arxiu **swf** que s'ha generat podem veure que l'animació es repeteix indefinidament. Per a mostrar l'animació una sola vegada, haurem d'aturar la línia de temps en l'últim fotograma de l'animació.



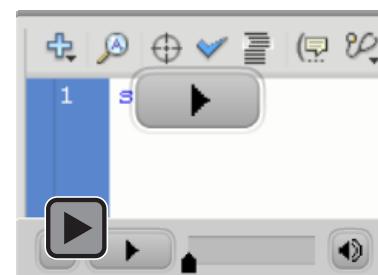
Creem una nova capa que anomenarem **as** (per ActionScript), i en la qual escriurem el codi necessari per a aturar la pel·lícula.



En el fotograma 72 d'aquesta capa, premem **F6** per inserir-hi un **fotograma clau**. Es mostrerà un cercle buit petit, ja que de moment aquest fotograma no té contingut.



Amb el fotograma seleccionat, fem clic amb el **botó dret** del ratolí i seleccionem **Acciones**, o bé seleccionem **Ventana > Acciones** per obrir el tauler **Acciones**.



Escrivim **stop();** i tanquem el tauler. El fotograma mostrarà una a petita que indica que aquest fotograma té assignada una acció de fotograma.

Tornarem a seleccionar **Control > Probar película** per provar la nostra animació. Ara l'animació s'atura en arribar a l'últim fotograma, ja que és el que té l'acció **stop()** assignada.

Ara ja tindrem un arxiu **swf** que conté la nostra animació del logotip. En tutorials més avançats veurem amb detall les diferents opcions per a publicar les nostres pel·lícules.

**Tutorial 4. Animació del logotip**  
**Pas 13 de 13**



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

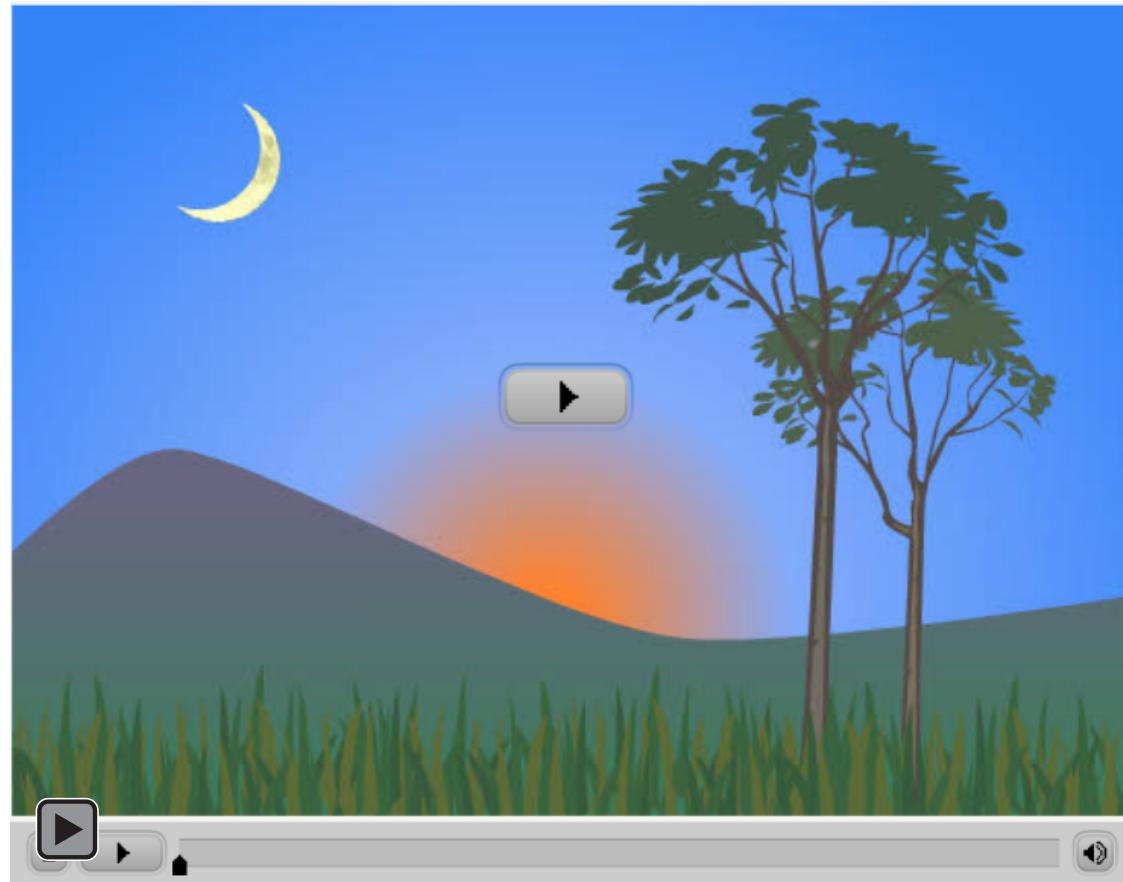
1. Proveu els diferents tipus de suavitzats disponibles per a explorar-ne el funcionament.
2. Creeu una animació per al filtre il·luminat del reflex modificant els valors del fotograma 1 en l'editor de moviment.
3. Creeu des de l'editor de moviment un fotograma clau en un punt intermedi de la línia de temps en el logo superior, i varieu-ne la posició per veure'n els efectes en l'animació.

## Tutorial 5. Cinemàtica inversa i eines 3D **Pas 1 de 21**

En aquest tutorial coneixerem algunes eines que poden resultar molt útils per a fer animacions, com l'**eina hueso** o les **eines de rotació i translació 3D**.

També veurem els avantatges d'utilitzar clips de pel·lícula imbricats per a fer alguns tipus d'animació.

Per a fer aquest tutorial utilitzarem l'ocell que hem creat en el tutorial 1 i el paisatge que hem creat en el tutorial 2.



## Tutorial 5. Cinemàtica inversa i eines 3D

### Pas 2 de 21

En molts casos en què necessitem fer una animació, serà convenient crear una estructura jeràrquica de clips de pel·lícula amb la pròpia línia de temps per a cada un.

Imaginem, per exemple, el cas de l'animació d'un cotxe. En primer lloc, tindrem un clip de pel·lícula anomenat *coche* que es desplaçarà d'un costat a l'altre de l'escenari. El clip de pel·lícula estarà format possiblement de diverses capes amb els seus diferents elements (finestres, rodes, etc.). En desplaçar el clip *coche* per l'escenari, es desplaçaran a l'uníson totes les capes que conté, la qual cosa facilitarà l'animació.

També necessitarem que les rodes girin mentre el cotxe es desplaça. Podem convertir les rodes en un clip de pel·lícula, i en la seva línia de temps fer que girin sobre si mateixes. És suficient amb una volta completa, ja que els clips de pel·lícula es reproduiran per defecte com un bucle.

Ara tindrem en l'escenari una línia de temps principal en la qual un clip anomenat *coche* es desplaça per l'escenari, i al seu torn el clip *coche* estarà format per diversos objectes, entre els quals hi ha els clips de les rodes, que al seu torn tenen una línia de temps pròpia en la qual giren.

D'aquesta manera, vagi on vagi el cotxe en la línia de temps principal, veurem com giren les rodes, ja que és una línia de temps imbricada dins del clip *coche*.



Els clips imbricats poden ser útils en molts tipus d'animacions, per exemple, el moviment o parpelleig dels ulls estaria dins del clip d'un cap que, al seu torn, estaria amb altres parts del cos dins d'un clip d'un cos humà.

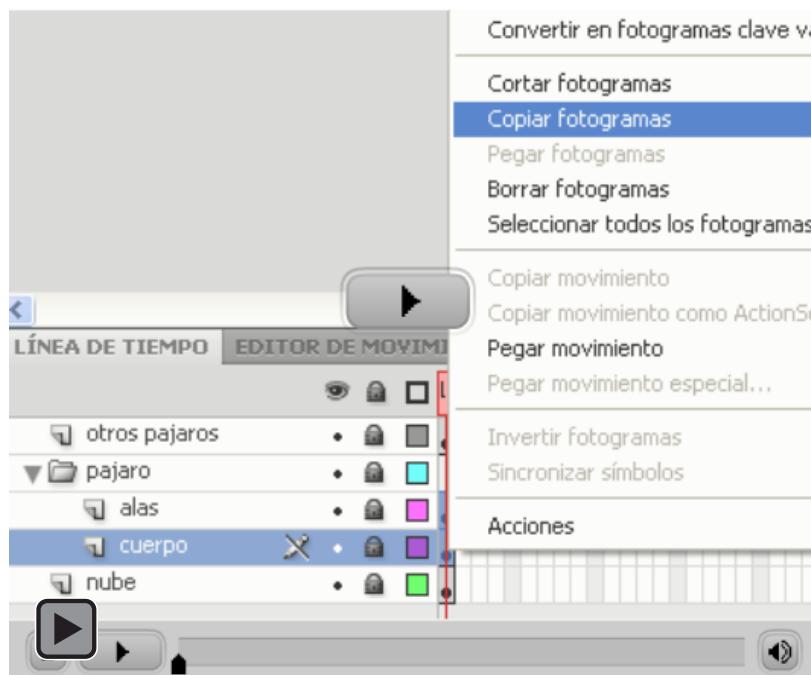
En el nostre cas, en l'escenari desplaçarem un clip anomenat *pájaro*, que al seu torn estarà format pel cos i les ales, que al seu torn tindran la pròpia línia de temps en la qual crearem el moviment de batre les ales. D'aquesta manera, en desplaçar l'ocell per l'escenari principal, sempre veurem batre les ales, ja que seran un clip de pel·lícula imbricat dins del clip *pájaro*.

## Tutorial 5. Cinemàtica inversa i eines 3D

### Pas 3 de 21

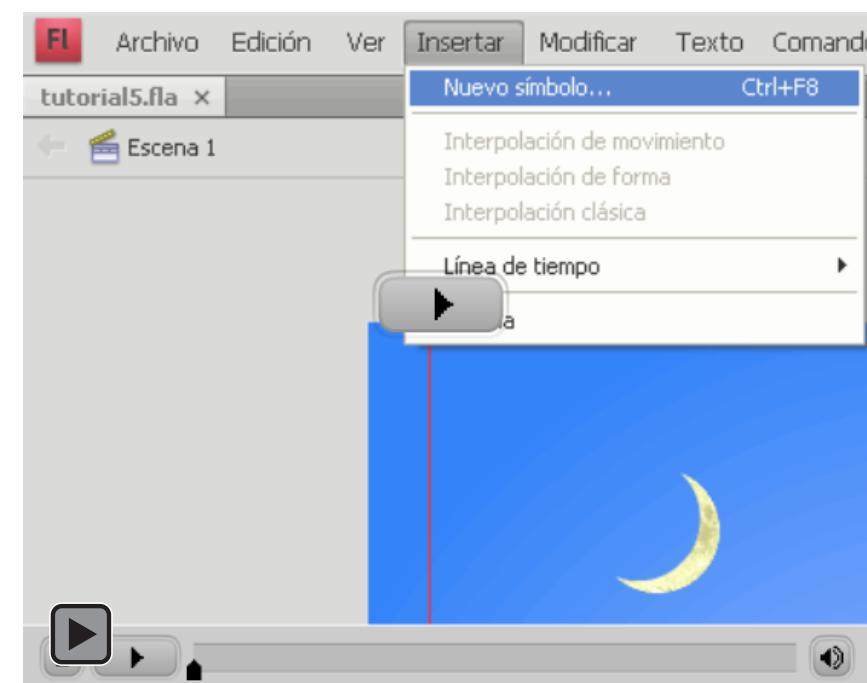
Obrim el document *tutorial1.fla* i despleguem la carpeta que conté l'ocell principal del document. Seleccionem els fotogrames de les dues capes fent-hi clic amb la tecla **Ctrl** premuda.

Després fem clic amb el **botó dret** del ratolí i seleccionem **Copiar fotogramas** del menú contextual.

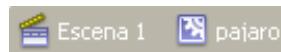


Ara ja podem tancar el *tutorial1.fla* sense necessitat de desar els canvis. Obrim el *tutorial2.fla*, que conté el paisatge, i el desem com a *tutorial5.fla*, que serà el document en el qual farem l'animació.

Creem un nou clip de pel·lícula buit seleccionant **Insertar > Nuevo Símbolo**, i l'anomenem *pájaro*.



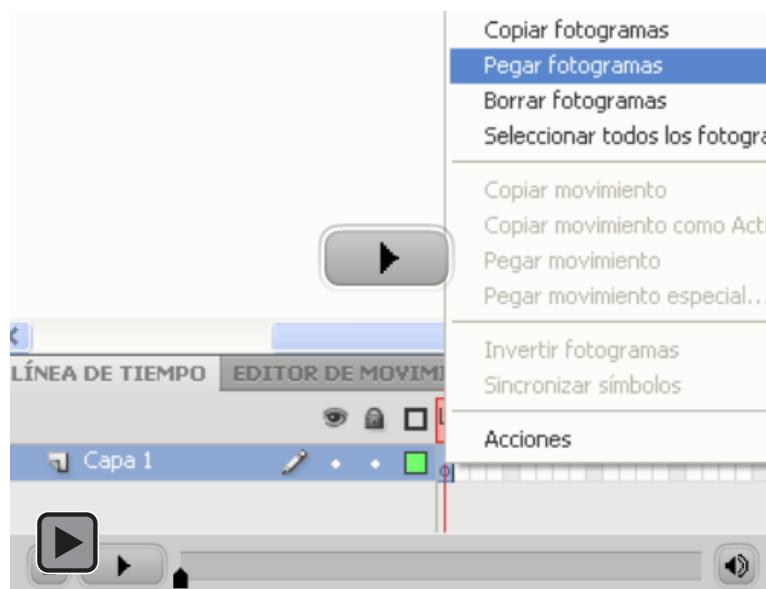
## Tutorial 5. Cinemàtica inversa i eines 3D Pas 4 de 21



Ara som dins del clip de pel·lícula que hem creat.

En ser dins d'un clip de pel·lícula, la línia de temps mostrarà el contingut d'aquest clip de pel·lícula. En aquest cas veurem que de moment només hi ha una capa i no té contingut.

Seleccionem el fotograma buit amb el **botó dret** del ratolí, i seleccionem **Pegar fotogramas**.



Els dos fotogrames que havíem copiat del *tutorial1.fla* apareixeran automàticament en aquest clip de pel·lícula. Els noms de les capes també es copiaran de l'arxiu d'origen.

Encara dins del clip *pájaro*, seleccionem el cos i les ales en l'escenari i situem el conjunt seleccionat en la posició X:0 i Y:0. Amb l'amplada i l'altura bloquejades, assignem una amplada total de 200.



El motiu de fer aquest pas és que en el tutorial 1 vam poder fer ocells amb mides molt diverses, i d'aquesta manera tots els ocells tindran una amplada de 200 amb les ales esteses, per la qual cosa podrem seguir el tutorial partint dels mateixos valors.

## Tutorial 5. Cinemàtica inversa i eines 3D Pas 5 de 21

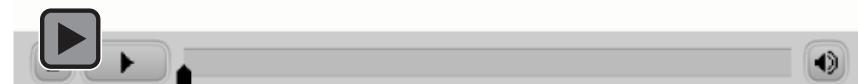
Esborrem l'ala esquerra, seleccionem l'ala dreta i la convertim en un clip de pel·lícula anomenat *a/a*.



En aquest moment tindrem un clip de pel·lícula anomenat *pájaro*, que al seu torn està compost del cos i de dues instàncies del clip de pel·lícula *a/a*.

Fem doble clic sobre l'ala dreta per poder editar el clip de pel·lícula sense perdre de vista la resta de l'ocell. La barra d'edició mostrarà que estem editant el clip *a/a*, dins del context del clip *pájaro*.

Aquesta barra també mostrarà una referència a l'escena principal per a poder tornar-hi, tot i que encara no hagim inclòs el clip en l'escena principal.



L'ala esquerra serà una instància del mateix clip de pel·lícula *a/a* que hem creat a partir de l'ala dreta. Per fer-ho arrosseguem una instància del clip de pel·lícula *a/a* de la biblioteca a l'escenari.

Girem aquesta instància seleccionant **Modificar > Transformar > Voltear horizontalmente**, i finalment l'alineem amb ajuda del tauler **Alinear**, tal com hem fet en el pas 9 del tutorial 1.



## Tutorial 5. Cinemàtica inversa i eines 3D Pas 6 de 21

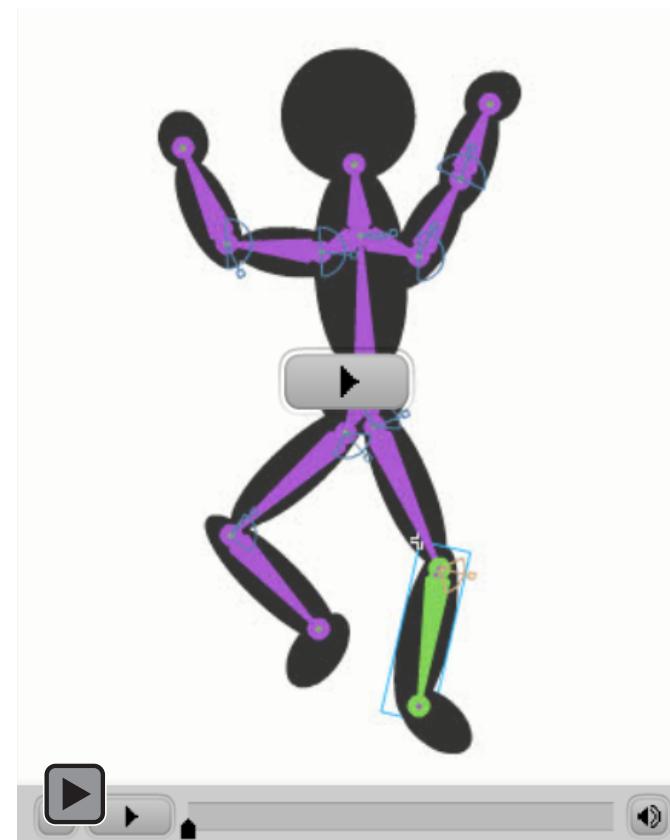
Animarem l'ala utilitzant la **cinemàtica inversa**, que és un mètode d'animació amb una estructura articulada d'osso

Els ossos poden connectar diferents instàncies de símbols, com es pot veure en el vídeo d'exemple, o bé es poden afegir a l'interior d'una forma, com farem en aquest tutorial.

Amb l'**eina Hueso** podem generar un **esquelet**, és a dir, una cadena d'osso vinculats entre ells. Després podem definir diferents poses en diferents fotogrames, i el Flash interpolarà les posicions dels fotogrames intermedis.

Amb la cinemàtica inversa podem crear animacions més complexes i naturals amb més facilitat. Per a més control, podem afegir restriccions de moviment limitant, per exemple, l'angle en què permetem que una articulació pugui doblar-se.

En aquest tutorial tan sols farem una petita aproximació a aquesta eina, per la qual cosa recomanem experimentar amb totes les possibilitats que ofereix.



## Tutorial 5. Cinemàtica inversa i eines 3D Pas 7 de 21

Tornant al nostre clip *a/la*, en primer lloc fem un zoom prou gros per a treballar amb comoditat i amb prou detall.

Seleccionem l'**eina Hueso** en la barra d'eines.

Fem clic en el centre de l'extrem interior de l'ala, i l'arrosseguem fins a aproximadament el punt central de l'ala, que apareixerà marcat amb una creu petita.

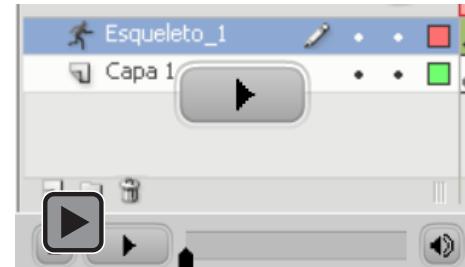
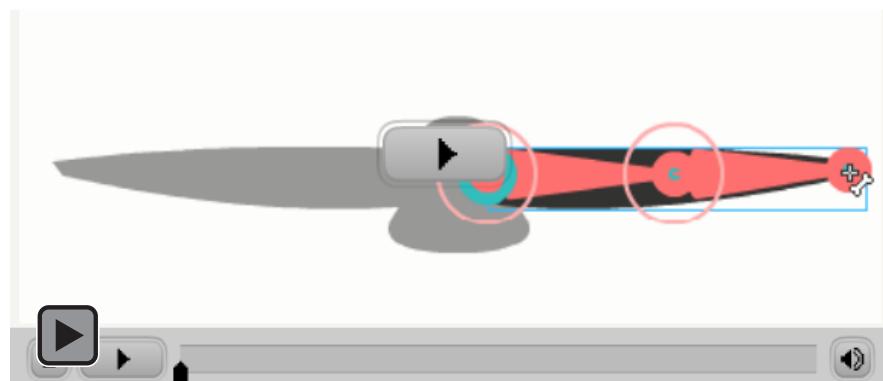
Després fem clic sobre el mateix punt en el qual hem acabat l'os anterior, i l'arrosseguem fins a l'extrem exterior de l'ala.



Tots els elements als quals s'afegeix un os passen automàticament a una nova **capa de posa**. Aquesta capa tindrà una icona amb una figura humana, i els fotogrames en la línia de temps tindran un fons verd.

En aquest cas, com que hem afegit ossos a l'únic element que hi havia, la capa original queda buida i tot el seu contingut passa automàticament a la nova capa de posa. Esborrem la capa que ha quedat buida.

Els ossos mostraran el color aleatori que s'hagi assignat al contorn de la nova capa. Si volem podem canviar-lo prement la icona de color de la capa.

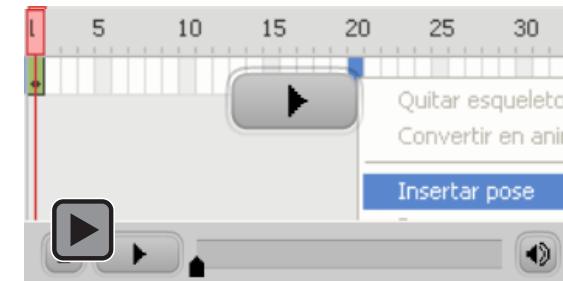
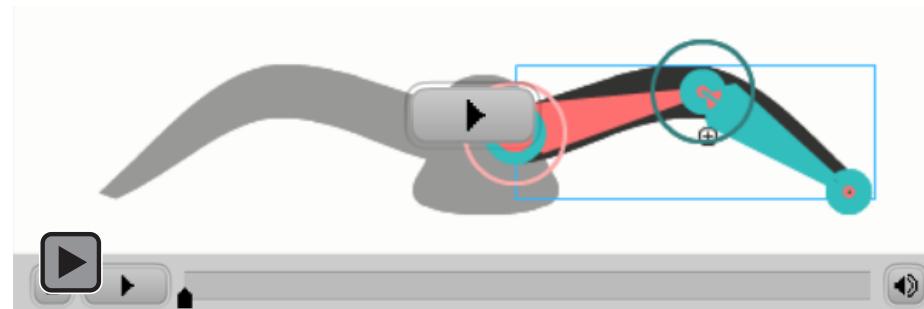


## Tutorial 5. Cinemàtica inversa i eines 3D Pas 8 de 21

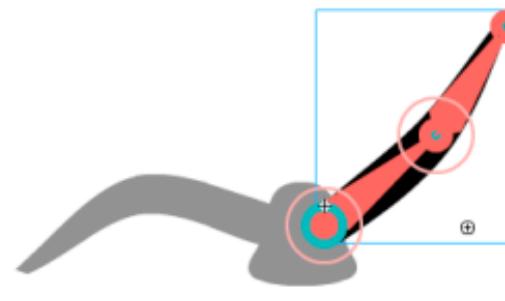
Començarem l'animació amb una posició lleugerament doblegada de l'ala.

Amb l'eina **Selección (V)** fem clic i arrosseguem la posició de l'ala fins a aconseguir una posició similar a la que es mostra en el vídeo.

Podem veure que les dues ales adopten la mateixa posició, ja que es tracta d'instàncies del mateix clip de pel·lícula.



Fem clic amb el **botó dret** del ratolí sobre el fotograma 20 i seleccionem **Insertar pose**. En aquest fotograma modifiquem la posició fins a aconseguir que sigui semblant a la de la imatge.

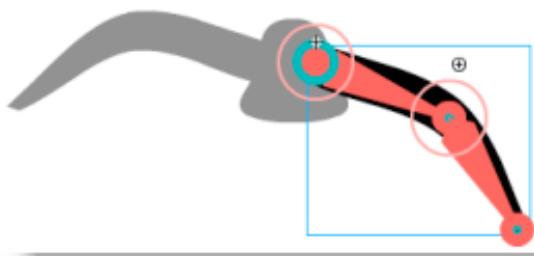


En aquest cas l'ala esquerra no mostrarà el canvi de posició, ja que quan editem un clip dins d'un context la resta dels elements resten en el fotograma en el qual estaven de la seva pròpia línia de temps (en aquest cas, el primer fotograma).

## Tutorial 5. Cinemàtica inversa i eines 3D

### Pas 9 de 21

De la mateixa manera, inserim una nova posa en el fotograma 40, i baixem l'ala fins a una posició com la de la imatge.



En el fotograma 50 repetirem la posa que havíem creat en el fotograma 1. Per fer-ho, en primer lloc inserim els fotogrames necessaris prement **F5** en el fotograma 50.

Per seleccionar un fotograma d'una posa, fem clic sobre el fotograma mentre mantenim premuda la tecla **Ctrl**. D'aquesta manera seleccionem el fotograma 1, després fem clic amb el **botó dret** del ratolí i seleccionem **Copiar pose**.

Seleccionem el fotograma 50 de la mateixa manera, premem el **botó dret** del ratolí i seleccionem **Pegar pose**.



Per provar l'animació de l'ala, premem la tecla **Intro**. D'aquesta manera l'animació de l'ala es mostrerà una sola vegada.

Per mostrar l'animació amb un **bucle**, que és com es reproduirà en la pel·lícula final, seleccionem **Control > Reproducir indefinidamente**, i tornem a premer la tecla **Intro** tant per iniciar com per aturar la reproducció.

Si és necessari, modificarem alguna de les poses fins que ens sembli que tenim un moviment natural.

## Tutorial 5. Cinemàtica inversa i eines 3D Pas 10 de 21

En visualitzar el bucle, ens adonem que l'animació sembla aturar-se un instant en la posa inicial.

Es deu al fet que el fotograma 50 i el fotograma 1 tenen la mateixa posa, i després del fotograma 50 es torna a reproduir el fotograma 1, que té el mateix contingut.



Per solucionar aquest problema, inserirem una nova posa en el fotograma 49 seleccionant **Insertar pose**, que mostrarà per defecte la posa calculada pel Flash per al fotograma immediatament anterior al de la posa inicial (la mateixa del fotograma 50).

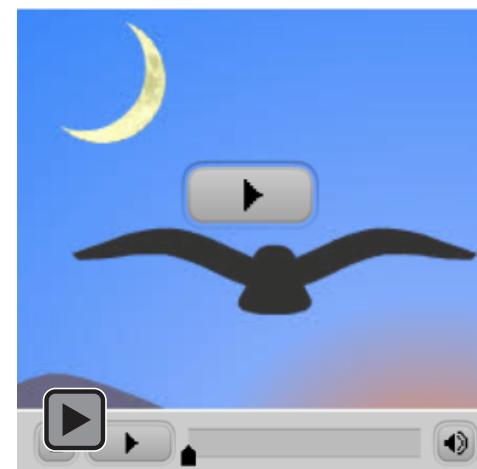
Després ja podem eliminar el fotograma 50 seleccionant **Quitar fotogramas** en el menú contextual.

Ara el bucle mostrerà el pas del fotograma 49 al fotograma 1, sense que hi hagi cap fotograma repetit, i amb l'última posa del fotograma 49 prèvia al fotograma 1 perfectament calculada.



Ara que ja tenim creat el bucle amb el batec de les ales, tornem a l'escena principal prement **Escena 1** en la barra d'edició.

Creem una nova capa anomenada *pajaro1* entre la capa *montaña* i la capa *arbol2*. Amb aquesta capa seleccionada, arrosseguem una instància del clip de pel·lícula *pájaro* des de la biblioteca a l'escenari.



Si ara premem la tecla **Intro** no veurem cap animació, ja que la línia de temps actual només té un fotograma.

Si seleccionem **Control > Probar película** podrem veure totes les línies de temps imbricades de la pel·lícula, incloent-hi les ales.

## Tutorial 5. Cinemàtica inversa i eines 3D Pas 11 de 21

Veiem que el moviment de l'ala és una mica lent, així que accelerarem lleugerament la velocitat del bucle.

Podem editar directament el clip *a/la* fent doble clic sobre el seu nom en la **biblioteca**, ja que per a augmentar la velocitat no necessitem tenir una referència visual de la resta de l'ocell.

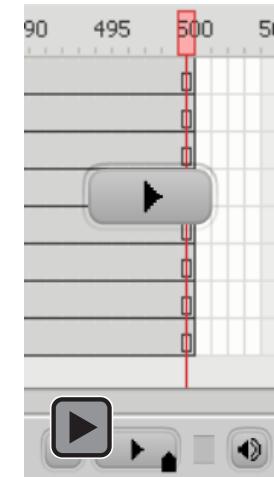
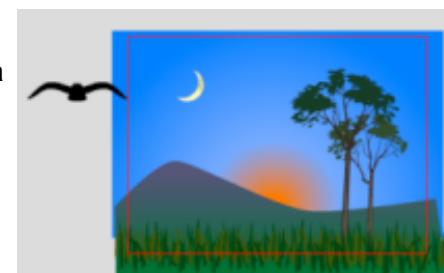
Reduirem la durada del bucle a 10 fotogrames. Per fer-ho, fem clic sobre l'extrem de l'animació i l'arrosseguem fins al fotograma 39. Veiem que la posició de la resta dels fotogrames s'ha reajustat proporcionalment a la nova durada.



Si tornem a provar la pel·lícula veurem que ara les ales es mouen més ràpidament. Com hem pogut veure, en qualsevol moment podem variar alguna de les poses d'un clip o la seva durada, i aquests canvis es reflectiran en les instàncies que estiguem utilitzant.

El pas següent serà crear l'animació amb el desplaçament de l'ocell en la línia de temps principal, és a dir, en la línia de temps que conté el paisatge.

Com a punt de partida situem fora de l'escenari, a l'altura de la lluna, la instància de l'ocell que tenim en la capa *pajaro1*.

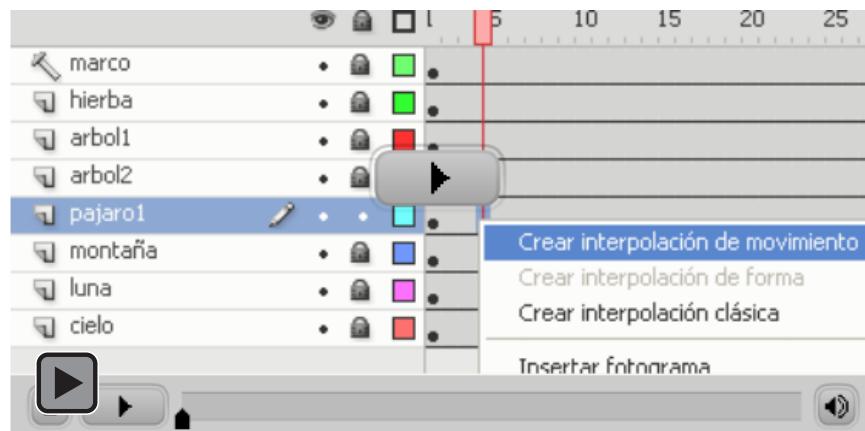


Inserirem 500 fotogrames en totes les capes de l'escena, ja que és la durada prevista per a la nostra animació.

La manera més ràpida de fer-ho és situar-nos en el fotograma 500 amb ajuda de la barra inferior de desplaçament, prémer el fotograma 500 d'una de les capes i, sense deixar-lo anar, arrossegar-lo per a seleccionar el fotograma 500 de la resta de les capes. Finalment, premem **F5** per afegir fotogrames a totes les capes seleccionades.

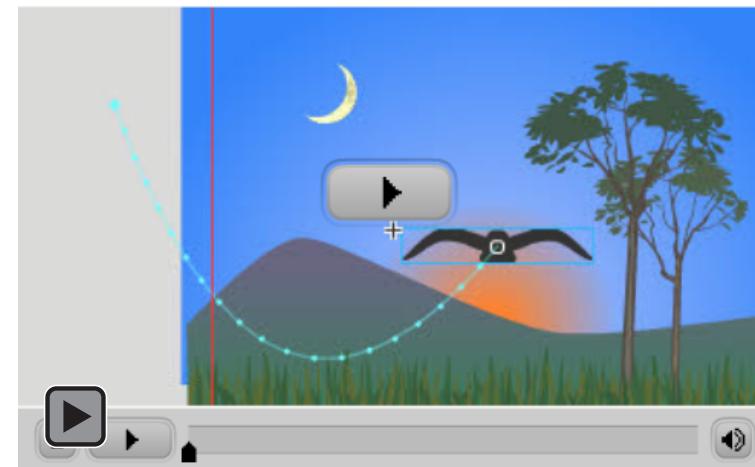
## Tutorial 5. Cinemàtica inversa i eines 3D Pas 12 de 21

Amb el **botó dret** del ratolí seleccionem qualsevol punt de la capa *pajaro1* i seleccionem ***Crear interpolación de movimiento***.



Situem el cap lector en el fotograma 500 i, amb l'eina **Selección**, desplaçem la instància del clip pájaro a un punt més o menys central de l'escenari.

Apareixerà una línia d'interpolació recta amb diversos punts intermedis equidistants. A diferència del tutorial anterior, en aquest cas no apareix un punt per cada fotograma, ja que en tenim un nombre massa elevat.



Corbarem la trajectòria de la interpolació prement i arrossegant directament la línia d'interpolació. Amb això definim un recorregut en el qual es desplaça cap a la dreta, primer baixant i finalment pujant.

Com en qualsevol altra corba, podríem afinar-ne la forma fent clic sobre els seus punts extrems amb l'eina **Subselección**, encara que en aquest cas no serà necessari.

Si introduíssim nous fotogrames clau de propietat en punts intermedis de la interpolació, podríem definir corbes complexes amb facilitat.

## Tutorial 5. Cinemàtica inversa i eines 3D Pas 13 de 21

En el fotograma 500, seleccionem la instància de l'ocell en l'escenari.

En l'**inspector de Propiedades**, en l'àrea de **Posición y vista 3D**, assignarem un valor de **profunditat (Z)** de 5.000. Com més gran sigui aquest valor, més semblarà que el clip s'allunya.

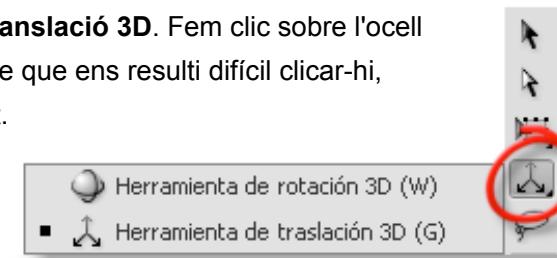


La forma de la corba s'haurà suavitzat considerablement, ja que ara el recorregut de la corba el segueix mentre s'allunya.

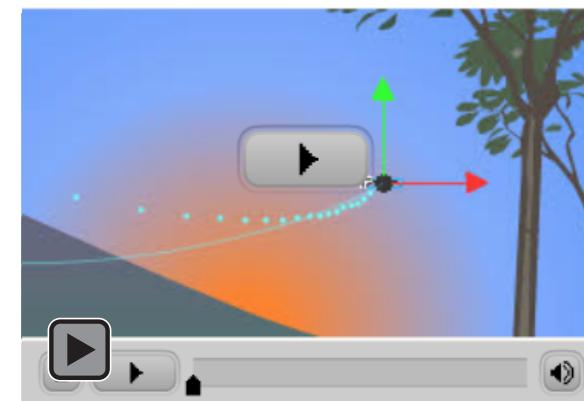
Amb aquest nou valor, veurem que els punts de la corba d'interpolació s'han redistribuït i que al principi de la corba apareixen més distanciats que al final. Això significa que, al principi de l'animació, el clip de pel·lícula avançarà més ràpid respecte a la corba i que s'anirà alentint al final.

Aquesta desacceleració simularà perfectament l'allunyament del clip, ja que un objecte proper varia més la seva posició respecte a l'observador que un objecte llunyà. Per exemple, un avió a la llunyania sembla que es desplaci amb més lentitud que si ens passa pel costat.

Seleccionem l'eina de **translació 3D**. Fem clic sobre l'ocell en l'escenari. És possible que ens resulti difícil clicar-hi, ja que està molt allunyat.



L'eix Z apareixerà com un punt negre central, però no el desplaçarem, ja que ho hem definit manualment en l'inspector de propietats. Premem i arrosseguem les puntes de les fletxes per desplaçar el clip en els **eixos X i Y** fins a la posició que vulguem. Haurem d'arrossegar trams amplis perquè el clip es desplaci, ja que és en un punt molt llunyà de l'eix Z.

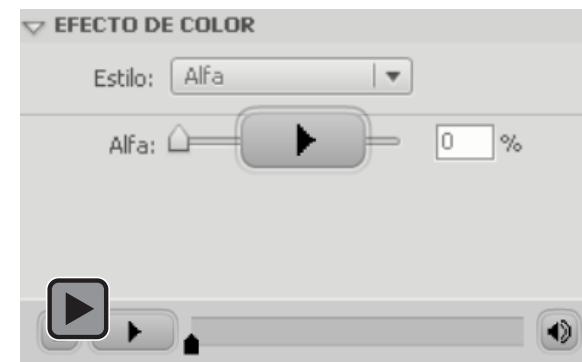


## Tutorial 5. Cinemàtica inversa i eines 3D

### Pas 14 de 21

El fet de desplaçar un objecte amb l'eina **Selección** pot semblar similar a desplaçar-lo amb l'eina de **Translación 3D** en els eixos horitzontal (X) i vertical (Y). Tanmateix, si l'objecte per traslladar tingués una rotació 3D, podríem observar clarament la diferència que representa utilitzar aquesta eina.

En el vídeo d'exemple podem veure que en traslladar un objecte en aquests eixos amb l'eina de *traslación 3D*, l'objecte canvia la seva perspectiva. Si això ho haguéssim fet amb l'eina **Selección**, la perspectiva que ens mostra el clip no hauria variat.



En l'**inspector de Propiedades** assignarem un valor d'**alfa** de 0 a la instància del clip *pájaro* en el fotograma 500.

Els valors possibles de la propietat alfa (transparència) oscil·len entre 100 (totalment opac) i 0 (transparent).

En assignar un alfa de 0 a l'últim fotograma de l'animació de la instància de l'ocell, aconseguim que, a mesura que avança l'animació, és a dir, segons es va allunyant l'ocell, l'animació es faci cada vegada més transparent i arribi a desaparèixer completament en l'últim fotograma.

## Tutorial 5. Cinemàtica inversa i eines 3D Pas 15 de 21

Si provem la pel·lícula veurem que l'animació no resulta gaire natural en els primers fotrames, ja que sembla que l'ocell voli de costat.

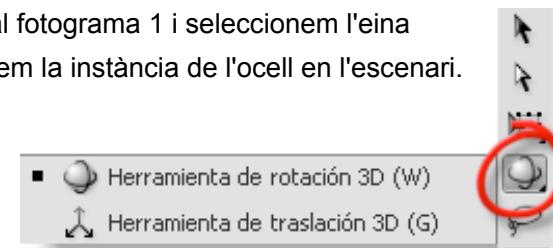
Per solucionar aquest problema, modificarem la rotació 3D de la instància en el primer fotograma. Abans de fer-ho, i perquè la posició final en el fotograma 500 no es modifiqui, desarem les propietats actuals de rotació en l'últim fotograma.

Per fer-ho, en l'**'Editor de movimento'**, afegirem **fotrames clau de propietat** per als tres tipus de **rotació 3D** en el fotograma 500.

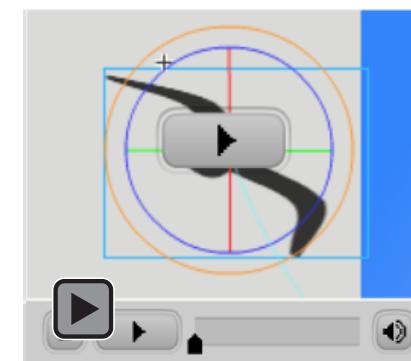
D'aquesta manera, l'animació de la instància acabarà amb una rotació amb valor de 0 en els tres eixos, independentment dels canvis que posteriorment fem en el fotograma 1.



Desplaçem el cap lector al fotograma 1 i seleccionem l'eina **de rotació 3D**. Seleccionem la instància de l'ocell en l'escenari.



Modifiquem la rotació en els eixos **X** (vermell), **Y** (verd) i **Z** (blau).



|            |    |
|------------|----|
| Rotación X | 0° |
| Rotación Y | 0° |
| Rotación Z | 0° |

També podríem haver assignat els valors directament en l'**'Editor de movimento'**.

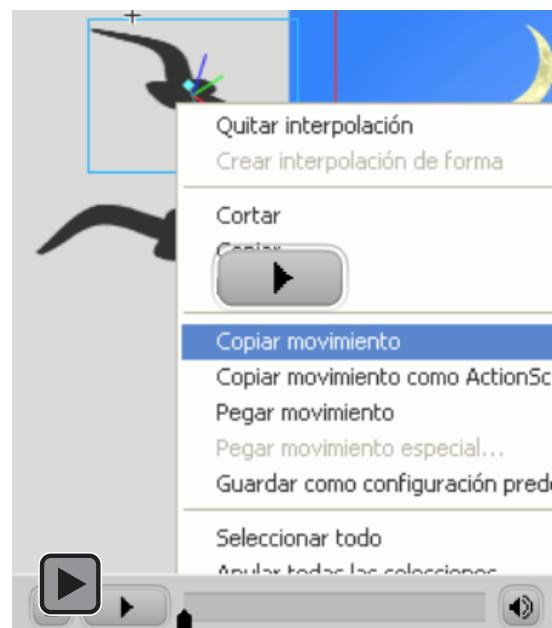
## Tutorial 5. Cinemàtica inversa i eines 3D Pas 16 de 21

Per fer l'animació d'un segon ocell, en primer lloc creem una nova capa anomenada *pajaro2* per sobre de la capa *pajaro1*.

Arrosseguem una altra instància del clip ocell des de la biblioteca a l'escenari en aquesta nova capa, i el situem sota el primer ocell, fora dels límits de l'escenari.

Fem clic sobre l'ocell de la capa 1 amb el **botó dret** del ratolí, i seleccionem **Copiar movimiento** en el menú contextual.

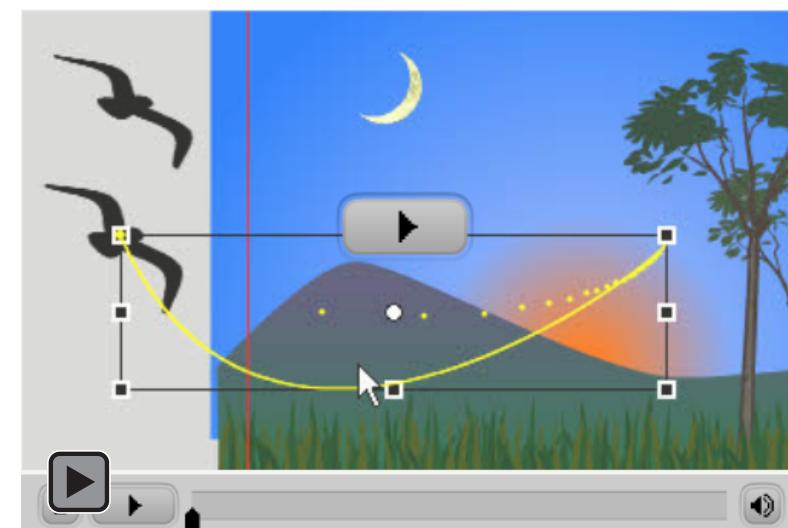
Després, fem clic sobre el nou ocell amb el **botó dret** del ratolí i seleccionem **Pegar movimiento** en el menú contextual.



Podem modificar les interpolacions de moviment com si fossin qualsevol objecte gràfic.

En aquest cas girarem lleugerament la interpolació del segon ocell, seleccionant-la en l'escenari amb l'eina **Transformación libre**.

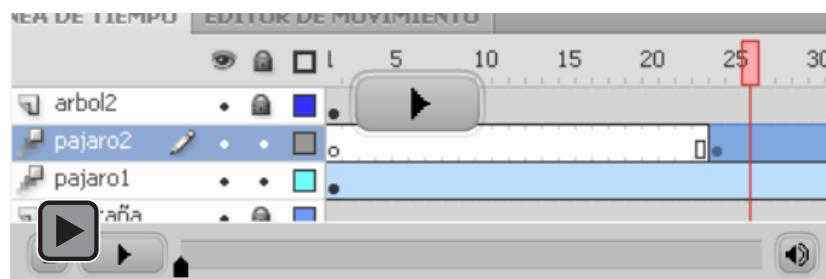
En haver girat la interpolació, el segon ocell començarà el vol per sota del primer ocell, però acabarà en una posició superior.



## Tutorial 5. Cinemàtica inversa i eines 3D Pas 17 de 21

Perquè el segon ocell comenci l'animació més tard que el primer, fem clic en la interpolació en la línia de temps i l'arrosseguem a una nova posició.

Si arrosseguem la interpolació perquè comenci en el fotograma 24 i tenim configurat el document a 24 fps, l'animació d'aquest ocell començarà un segon més tard que l'anterior.



En desplaçar l'animació, ara la pel·lícula s'estendrà més de 500 fotogrames (524 en aquest cas). Per tant, hem de tornar a afegir fotogrames (**F5**) en la resta de les capes.

Afegirem fotogrames per a totes les capes excepte les dels ocells, i estendrem la durada fins a 600 fotogrames.

D'aquesta manera, després que desapareguin els dos ocells, el paisatge es continuarà veient una estona.

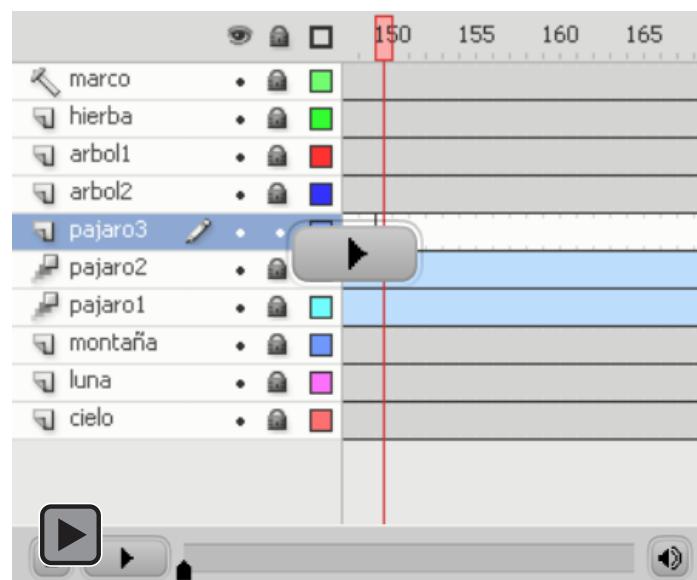


Recordem que per a afegir fotogrames sense interpolació de moviment, hem de fer clic sobre el fotograma que vulguem que hi hagi al final de cada capa, i premer **F5** o bé, si premem el fotograma amb el **botó dret** del ratolí, seleccionar **Insertar fotograma** en el menú contextual.

## Tutorial 5. Cinemàtica inversa i eines 3D Pas 18 de 21

Farem l'animació d'un últim ocell. Crearem una nova capa que anomenarem *pajaro3*.

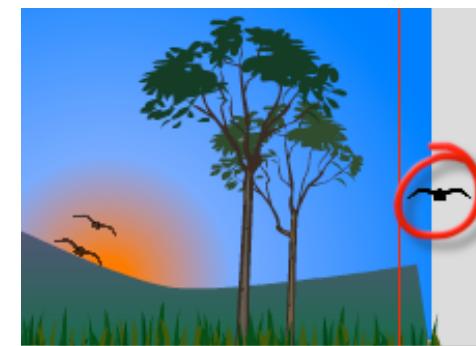
Aquesta vegada l'animació començarà en el fotograma 150. Una manera de controlar el punt d'inici d'una animació és inserint en primer lloc un fotograma clau en el fotograma en el qual començarà l'animació. En aquest cas fem clic amb el **botó dret** del ratolí sobre el fotograma 150, que es convertirà en el primer de l'animació, i seleccionem **Insertar fotograma clave**.



Com que en aquesta capa no hi ha contingut previ, es crearà un fotograma clau buit.

Arrosseguem una instància del clip *pájaro* de la biblioteca a l'escenari, i veurem que es posiciona en el fotograma clau que hem creat, és a dir, en el fotograma 150. El contingut sempre es posicionarà en l'últim fotograma clau que trobi per a aquesta capa i, si no n'hi ha cap, es posicionarà en el fotograma 1.

En l'**inspector de Propiedades**, en l'àrea de **Posición y vista 3D**, hi assignem un valor de **profunditat (Z)** de 1.000. Després, amb ajuda de l'eina de **translació 3D**, el posicionem en la part dreta fora dels límits de l'escenari, en una altura inferior a la copa dels arbres.

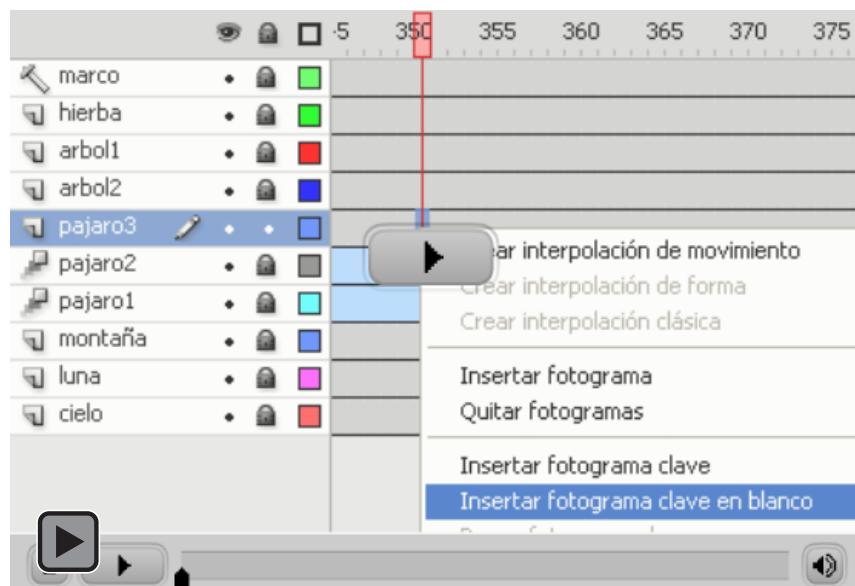


## Tutorial 5. Cinemàtica inversa i eines 3D

### Pas 19 de 21

Si ara féssim una interpolació de moviment, aquesta s'estendria fins al fotograma 600. En el nostre cas farem que l'animació acabi en el fotograma 350.

Per a això, abans de crear la interpolació fem clic amb el **botó dret** del ratolí en el fotograma 351, i seleccionem **Insertar fotograma clave en blanco**. Després fem clic amb el **botó dret** sobre qualsevol punt de la capa on hi ha el clip (en gris), i seleccionem **Crear interpolación de movimiento**.



Tornant al fotograma 150, és a dir, a l'inici de l'animació d'aquest ocell, assignem una **rotació Z** de  $-50^{\circ}$  en l'**editor de moviment**.

|            |               |
|------------|---------------|
| Rotación X | $0^{\circ}$   |
| Rotación Y | $0^{\circ}$   |
| Rotación Z | $-50^{\circ}$ |

Finalment, de nou en el fotograma 350, assignem en l'**inspector de Propiedades**, en l'àrea de **Posición y vista 3D**, els valors de **X: $-100$ , Y: $100$  i Z: $-100$** .



D'aquesta manera l'ocell volarà apropiant-se fins a desaparèixer per la cantonada superior esquerra. Com podem veure, els valors negatius de **Z** fan que un clip aparegui més a prop.

Si observem l'editor de moviment, veiem que la línia de la propietat de rotació Z apareix puntejada, cosa que significa que no té interpolació, i que es manté la mateixa rotació en tota l'animació.



## Tutorial 5. Cinemàtica inversa i eines 3D Pas 20 de 21

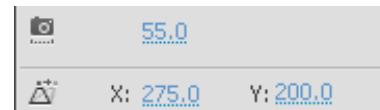
Hi ha diverses opcions interessants relacionades amb les interpolacions que no hem vist en aquest tutorial, però que són interessants de conèixer. Farem un breu resum sobre algunes de les opcions amb les quals recomanem experimentar.

Per començar, en l'àrea de **Posición y vista 3D** que apareix en l'**inspector de Propiedades** quan seleccionem un clip de pel·lícula, també hi ha altres àrees que no hem utilitzat.

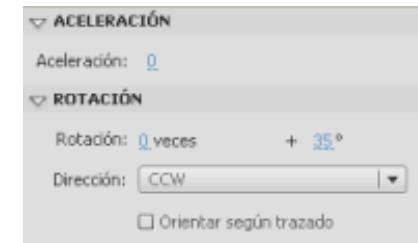
En primer lloc, tenim l'**Ángulo de perspectiva**. Aquesta propietat defineix l'angle de visió de manera similar a com podria fer-ho un zoom fotogràfic. El valor de l'angle pot oscil·lar entre 1 i 180. Els valors més alts produeixen un escenari amb més profunditat.

També tenim la posició X i Y del **Punto de desvanecimiento**. Aquesta posició es refereix a l'orientació de l'eix Z, que per defecte és el centre de l'escenari.

Si modifiquem l'angle de perspectiva o el punt d'esvaïment, aquests canvis afectaran tots els clips 3D de la nostra pel·lícula.



D'altra banda, quan seleccionem una interpolació, podem modificar-ne l'**acceleració** i la **rotació** en l'**inspector de Propiedades**.



Una altra opció molt interessant és el nou tauler que es troba a **Ventana > Configuración predefinida de movimiento**.

Amb aquest tauler podem assignar fàcilment un moviment predefinit a un clip, i després podem modificar-ne la interpolació creada de la mateixa manera que si l'haguéssim creat nosaltres.



També permet desar les nostres pròpies interpolacions per a poder reutilitzar-les amb altres clips.

Tutorial 5. Cinemàtica inversa i eines 3D

**Pas 21 de 21**



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

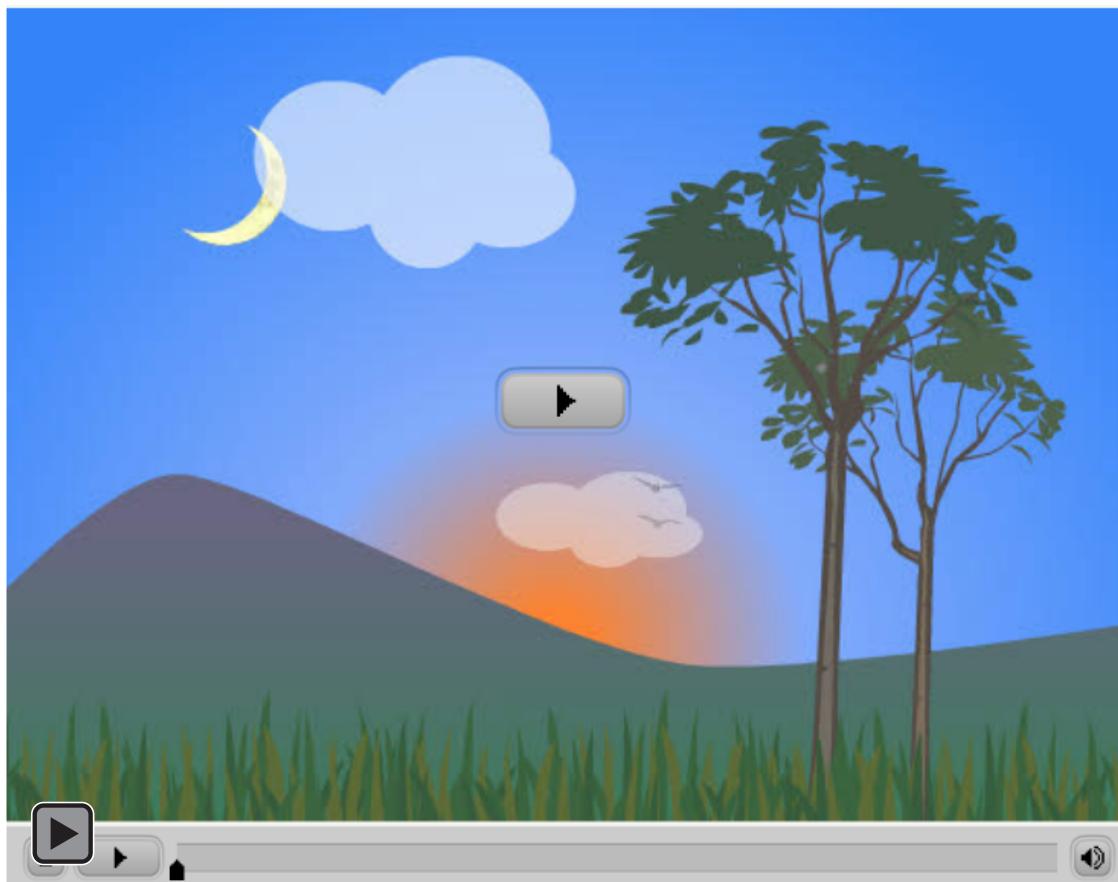
1. Canvieu el punt d'esvaïment i l'angle de perspectiva.
2. Afegiu un fotograma clau de propietat en un punt intermedi d'una interpolació, i modifiqueu-ne les propietats de rotació en aquest punt. Per a això podeu utilitzar l'editor de moviment.
3. Afegiu noves poses en el clip *ala*, afegint-hi un nou cicle del batec de les ales que no sigui exactament igual al primer.

## Tutorial 6. Animació amb ActionScript 3.0 **Pas 1 de 19**

En aquest tutorial farem una primera aproximació a la programació amb ActionScript 3.0, l'última versió del llenguatge de programació del Flash.

Amb aquest llenguatge de programació podem afegir interactivitat a les nostres pel·lícules, controlar l'aspecte o moviment d'objectes en l'escenari, crear aplicacions complexes, etc.

En primer lloc, aprendrem a controlar els clips que hi ha en l'escenari mitjançant programació. En aquest cas controlarem un clip fet a partir del núvol que hem creat en el tutorial 1.



## Tutorial 6. Animació amb ActionScript 3.0 Pas 2 de 19

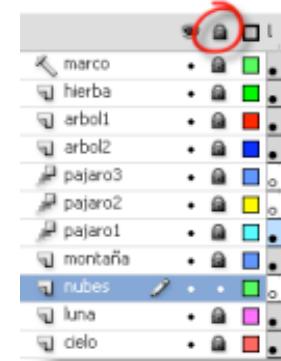
Obrim el document *tutorial1.fla*. Fem clic sobre el núvol amb el **botó dret** del ratolí i seleccionem **Copiar**. Si no podem seleccionar-lo possiblement és perquè tenim bloquejada la capa *nube* en la línia de temps, així que primer l'haurem de desbloquejar.



Una vegada copiat, ja podem tancar el document sense necessitat de desar els canvis. Obrim l'arxiu *tutorial5.fla*, que conté l'animació dels ocells, i el desem com a *tutorial6.fla*.

Creem una nova capa per darrere de la muntanya i l'anomenem *nubes*.

Bloquegem la resta de les capes per evitar possibles modificacions accidentals. La manera més ràpida és clicar sobre el cadenat que hi ha sobre totes les capes, i després desbloquejar la capa que ens interessa.



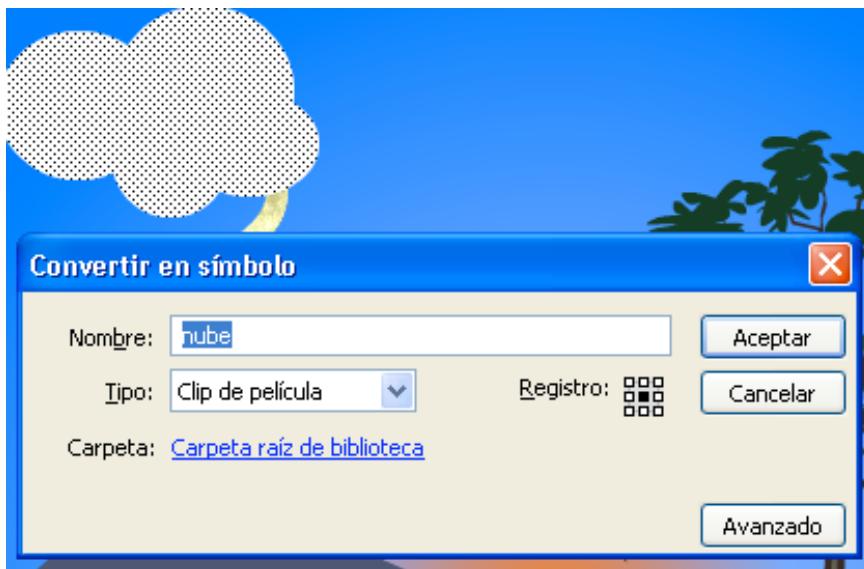
Amb la capa *nubes* activa, seleccionem qualsevol punt de l'escenari amb el **botó dret** del ratolí i seleccionem **Pegar**.

Situem el núvol en la cantonada superior esquerra de l'escenari.



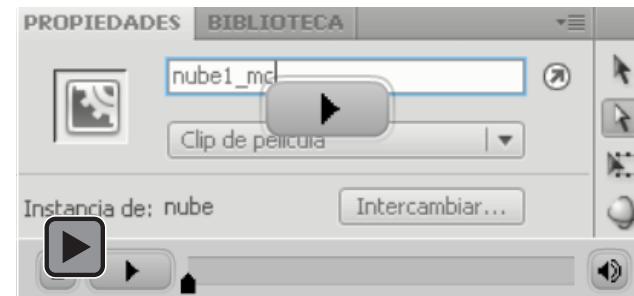
## Tutorial 6. Animació amb ActionScript 3.0 Pas 3 de 19

Amb el núvol seleccionat, premem **F8** o **Modificar > Convertir en símbolo**. Li donem el nom de *nube* i com a tipus seleccionem **Clip de película**.



Ara tindrem un clip de pel·lícula anomenat *nube* en la biblioteca, mentre que en l'escenari tindrem una instància d'aquest clip de pel·lícula, tal com ens indicarà la part superior de l'**inspector de Propiedades**.

Per a poder controlar una instància amb ActionScript, el primer que hem de fer és donar-li un nom d'instància en l'inspector de *Propiedades*. Li donarem el nom *nube1\_mc*.



És convenient que els noms d'instància sempre comencin per una lletra minúscula i que acabin de la manera següent:

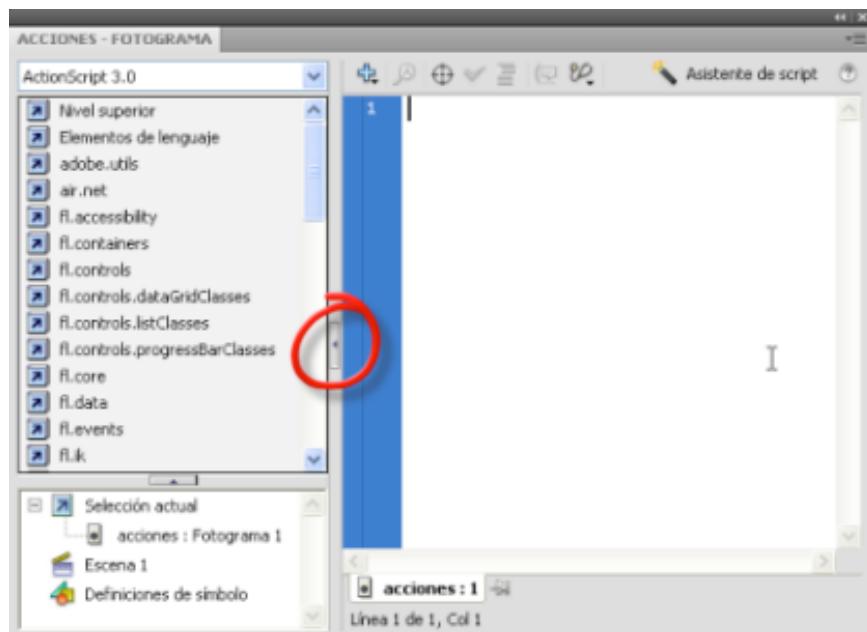
- *\_mc* per a clips de pel·lícula (*movie clip*)
- *\_btn* per a botons
- *\_txt* per a camps de text

La raó per a això és que quan escriguem codi sobre una instància amb un d'aquests acabaments, el Flash detectarà el tipus d'objecte de què es tracta, i ajudarà amb els suggeriments de codi adequats a aquest tipus d'objecte.

## Tutorial 6. Animació amb ActionScript 3.0 Pas 4 de 19

Escriurem les accions en una **nova capa** que situarem per damunt de totes les altres, i que anomenarem *acciones*. És convenient tenir totes les accions separades en una capa per a facilitar l'organització del document.

Obrim el **tauler Acciones** prement **F9** o **Ventana > Acciones**.



En la columna esquerra d'aquest tauler tenim en primer lloc un accés exhaustiu a les diferents classes i funcions disponibles. En el nostre cas no utilitzarem aquesta manera d'introduir codi.

En la part inferior de la mateixa columna tenim informació sobre la selecció actual (en el nostre cas, el fotograma 1 de la capa *acciones*). Quan tinguem accions distribuïdes en més fotogrames, des d'aquí tindrem un accés còmode i directe que ens permetrà accedir a la programació dels diferents fotogrames.

En la part central tenim l'editor, que és l'àrea sobre la qual escriurem el codi. Sobre aquesta àrea tenim algunes eines per a comprovar el codi, inserir comentaris, accedir a l'ajuda, etc. Recomanem tenir desactivat l'assistent de *script*.

Per a disposar de més espai per a escriure el codi, recomanem reduir la columna esquerra clicant sobre la fletxa petita que separa les dues columnes. Podrem expandir-la de nou quan ho necessitem.

## Tutorial 6. Animació amb ActionScript 3.0 Pas 5 de 19

Comencem assignant un valor de .6 a la propietat *alpha*. Aquest valor significa una opacitat del 60%. Aquest valor pot oscil·lar entre 0 (transparent) i 1 (opac).

Comencem escrivint el nom de la instància seguit per un punt. En escriure el punt, apareixerà automàticament un desplegable amb diferents mètodes i propietats que podem assignar a un clip de pel·lícula. Aquesta ajuda no apareixerà si no haguéssim posat *\_mc* com a final del nom de la instància.

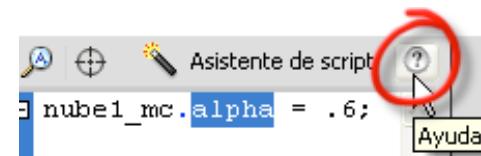
Després del punt s'escriu el mètode que volem executar o la propietat que volem assignar. En aquest cas hem seleccionat la propietat *alpha*. Les paraules clau del llenguatge apareixen en color blau.



Tot seguit, després d'un signe =, escrivim el valor per a aquesta propietat. En aquest cas hem escrit .6 (també podríem haver escrit 0.6). Els espais abans i després del signe = no són necessaris, però milloren la llegibilitat del codi. Acabem cada sentència amb un punt i coma (;).

Si tenim problemes amb la mida del text del codi, podem obrir les **preferències (Ctrl+U)** i, en la categoria **ActionScript**, canviar el tipus de text, la mida de la lletra o els colors. És preferible mantenir els colors del codi, ja que ens ajuda a distingir paraules clau, comentaris, etc.

Una manera ràpida d'accedir a l'ajuda sobre alguna paraula clau és seleccionar-la directament en l'editor i clicar sobre la icona de l'ajuda. En la pàgina d'ajuda podrem veure definicions i exemples d'ús de cada paraula clau.



## Tutorial 6. Animació amb ActionScript 3.0

### Pas 6 de 19

Si reduïm el **tauler Acciones** prement la barra grisa superior del tauler per a veure l'escenari, no veiem cap canvi en el núvol. Això es deu al fet que la programació s'executa quan provem o publiquem la pel·lícula (temps d'execució), i no directament en l'escenari d'edició (temps d'edició).

Per a comprovar com ha canviat l'alfa del núvol haurem de provar la pel·lícula amb **Control > Probar película (Ctrl+Intro)**.

Si hem assignat un valor per a una propietat en el codi, aquest valor prevaldrà sobre el que hagim definit en el temps d'edició. Per exemple, en el cas de l'alfa, podem assignar diferents valors en l'inspector de propietats, però quan provem la pel·lícula sempre veurem el núvol amb el 60% de l'alfa, ja que és el valor que hem introduït en el codi.

De la mateixa manera, si assignem en el codi valors per a altres propietats com, per exemple, les posicions X i Y del clip, aquest apareixerà en les coordenades que especificuem en la programació, i no en les coordenades en les quals l'hagim situat en l'escenari.

Per a afegir **interactivitat** a una pel·lícula, el més probable és que necessitem **detectar** quan es produeix un determinat **esdeveniment** (que s'hagi fet clic sobre un botó, que s'hagi premut una tecla, etc.), i després executar una **acció** sobre la base d'aquest esdeveniment.

Per a això hem d'agregar un **listener**, és a dir, afegir a un objecte la capacitat de detectar si ha passat alguna cosa, i que cridi una funció quan això passi.

La manera general de crear un *listener* és la següent:

```
nombreClip.addEventListener(nombreEvento,nombreFuncion);
```

En el nostre cas farem que el núvol s'arrossegi quan el premem.

Per a això afegirem la línia següent al codi:

```
nube1_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);
```

Amb aquest codi hem afegit, a la nostra instància *nube1\_mc*, un detector per a un esdeveniment del ratolí (*MouseEvent*). En concret, l'esdeveniment del ratolí que volem que detecti és si l'usuari l'ha premut (*MOUSE\_DOWN*). Si és així, executarà una funció que hem anomenat *arrastrar*, i que definirem més endavant.

## Tutorial 6. Animació amb ActionScript 3.0 Pas 7 de 19

Un esdeveniment de ratolí similar és *MouseEvent.CLICK*. La diferència és que amb *CLICK* es detecta si s'ha premut i s'ha deixat anar el ratolí, és a dir, si s'ha fet clic. Més endavant veurem que en el nostre cas executarem accions diferents per a les accions de premer i deixar anar el ratolí, i és per això que hem triat l'esdeveniment *MOUSE\_DOWN* (i després utilitzarem *MOUSE\_UP*).

Ara passarem a definir la nostra funció *arrastrar*. Una funció és un bloc d'instruccions agrupades, i que s'executa quan es crida des d'una altra funció o mètode. En escriure el codi, les funcions i mètodes es diferencien de les propietats perquè després del nom dels primers hi ha un parèntesi. El parèntesi pot ser buit, o bé pot rebre un o diversos paràmetres separats per comes. Per exemple, *alpha* és una propietat, mentre que *addEventListener* és un mètode amb dos paràmetres (esdeveniment i funció).

Per a crear una funció hem de començar amb la paraula clau *function*. A continuació d'aquesta escrivim el nom que volem donar a la nostra funció. En aquest cas li donem el nom *arrastrar*. Després del nom, escrivim els parèntesis:

```
function arrastrar()
```

Les funcions que s'hagin de cridar des d'un *addEventListener*, com és el nostre cas, reben un paràmetre. Aquest paràmetre és un objecte del tipus d'esdeveniment que ha desencadenat la funció, en aquest cas *MouseEvent*. Entre els parèntesis escriurem el nom de l'objecte (el nom que vulguem), dos punts (:) i el tipus d'objecte.

Per tant, de moment, la nostra funció començarà així:

```
function arrastrar(e:MouseEvent)
```

Hem donat al paràmetre el nom *e* per *esdeveniment*, però n'hi podíem haver donat un altre. Generalment, a aquest paràmetre se sol donar el nom *e* o *event*.

Després del nom de la funció és convenient especificar el tipus de dades que tornarà. En el nostre cas, la funció no tornarà cap resultat, sinó que simplement executarà una acció. En aquests casos, com a tipus de dades s'assigna *void* (buit).

La nostra funció queda com s'indica a continuació:

```
function arrastrar(e:MouseEvent) :void
```

## Tutorial 6. Animació amb ActionScript 3.0 Pas 8 de 19

Ara ens queda especificar les accions que volem que executi aquesta funció. Tot el contingut d'una funció s'escriu entre claus, així que per a assegurar-nos que no se'ns oblidin, és convenient escriure primer les claus, i després passar a escriure entre elles les diferents instruccions.

```
function arrastrar(e:MouseEvent):void
{
    //aquí irá nuestro código
}
```

Les dues barres que hem afegit en el codi anterior abans de la frase "aquí irá nuestro código" indiquen que és un comentari d'una sola línia. Podem afegir els comentaris que vulguem en el nostre codi, ja que no s'executarán. En comentaris de diverses línies utilitzarem /\* i \*/ per a delimitar el text que volem que sigui un comentari.

Els comentaris poden ser molt útils no solament per a afegir informació, sinó també per a fer que algunes línies del codi no s'executin temporalment, la qual cosa ens ajudarà a comprovar el funcionament del nostre codi.

Provem d'escriure el codi següent:

```
function arrastrar(e:MouseEvent):void
{
    trace("Estoy pulsando sobre la nube");
}
```

La funció *trace* mostrarà en un tauler anomenat **Salida** el que hi hagi dins del parèntesi. Si escrivim entre cometes el contingut del *trace*, en el tauler **Salida** es mostrarà exactament la frase que hagim escrit. Les cometes permeten escriure cadenes de caràcters, i les veurem de color verd en el tauler **Acciones**.

Les dades que apareguin en aquest tauler no les veurà l'usuari final, però sí quan provem la pel·lícula. Si apareix aquesta frase, vol dir que aquesta línia de codi s'està executant correctament.

Recapitem: hem afegit al núvol un detector de l'esdeveniment de premer el núvol amb el ratolí, i quan aquest esdeveniment es detecti, s'executarà la funció *arrastrar*, que té la instrucció de mostrar en el tauler **Salida** la frase *Estoy pulsando sobre la nube*. Provem la pel·lícula amb **Control > Probar película** i fem clic sobre el núvol per a comprovar el funcionament del nostre codi.

## Tutorial 6. Animació amb ActionScript 3.0

### Pas 9 de 19

Ara que hem comprovat que el nostre codi funciona correctament, substituïm la funció *trace* per l'acció que realment volem que tingui lloc, que és que el núvol s'arrosseguí.

```
function arrastrar(e:MouseEvent):void
{
    nube1_mc.startDrag();
}
```

Si tornem a provar la nostra pel·lícula, podrem veure que en premer el núvol aquest comença a arrossegar-se juntament amb el ratolí, i no deixa d'arrossegar-se en cap moment.

Perquè l'arrossegament s'aturi, haurem d'introduir una funció nova que permeti que, quan es deixi anar el botó del ratolí, el núvol deixi d'arrossegar-se.

El funcionament serà el mateix que l'anterior. Haurem de crear un altre detector per a l'esdeveniment de deixar anar el ratolí, i assignar-hi una funció que aturi l'arrossegament del núvol.

Per llegibilitat del codi podem ajuntar, d'una banda, les línies amb la creació dels *listeners* i, d'altra banda, les funcions.

La manera més ràpida per a escriure aquest codi nou és copiar i enganxar el que ja havíem creat, ja que en molts aspectes és similar. Substituirem el *MOUSE\_DOWN* per *MOUSE\_UP*, i el nom de la funció nova serà *soltar* en comptes d'*arrastrar*. Finalment, la funció que atura un arrossegament actiu és *stopDrag()*.

El codi que hem creat fins ara en el primer fotograma de la capa *acciones* és el següent:

```
nube1_mc.alpha = .6;

nube1_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);
nube1_mc.addEventListener(MouseEvent.MOUSE_UP,soltar);

function arrastrar(e:MouseEvent):void
{
    nube1_mc.startDrag();
}

function soltar(e:MouseEvent):void
{
    nube1_mc.stopDrag();
}
```

## Tutorial 6. Animació amb ActionScript 3.0

### Pas 10 de 19

A més de l'arrossegament interactiu del núvol, afegirem un desplaçament continu del núvol per l'escenari mitjançant una funció que anomenarem *desplazar*.

Hi ha un tipus d'esdeveniment anomenat *ENTER\_FRAME*, que pertany a una categoria genèrica d'esdeveniments anomenada *Event*, que s'executa cada vegada que el cap lector es desplaça un fotograma. Si el cap lector està aturat, però l'objecte a què s'associa és a l'escenari (tant si és visible com no), també s'executarà amb la mateixa freqüència que els fotogrames per segon que tinguem definits.

En el cas de la nostra pel·lícula, una funció que s'associi a un *ENTER\_FRAME* s'executarà cada vegada que el cap lector avanci, és a dir, 24 vegades per segon. Canviarem lleugerament la posició del núvol cada 1/24 de segon, per la qual cosa mostrarà una animació fluida.

Afegim aquest *listener* nou sota els altres dos que havíem creat. Per a això escrivim aquest codi:

```
nubel_mc.addEventListener(Event.ENTER_FRAME,desplazar);
```

A falta de definir les instruccions que executarà la funció *desplazar*, la seva definició quedarà com s'indica a continuació:

```
function desplazar(e:Event):void
{
}
```

Com podem veure, l'esdeveniment *ENTER\_FRAME* pertany a una categoria anomenada *Event*, i no a esdeveniments de ratolí (*MouseEvent*). Aquest *event* apareixerà tant en la creació del *listener* com en el paràmetre de la funció.

Perquè el núvol avanci cap a la dreta haurem de variar-ne la posició x a poc a poc dins de la funció *desplazar*. Per exemple, podem especificar que la posició x del núvol augmenti d'un en un el seu valor, que traduït a codi seria:

```
nubel_mc.x = nubel_mc.x + 1;
```

o bé, més senzill:

```
nubel_mc.x += 1;
```

que significa sumar-ne el valor més el nombre que hi ha després de =.

## Tutorial 6. Animació amb ActionScript 3.0

### Pas 11 de 19

De moment tindrem la funció *desplazar* com s'indica a continuació:

```
function desplazar(e:Event):void
{
    nubel_mc.x += 1;
}
```

Provem la pel·lícula per comprovar com es desplaça el núvol cap a la dreta, mentre encara es pot arrossegar. Podem provar amb diferents valors fins a trobar una velocitat que ens sembli adequada (per exemple, 0.4).

Si mantenim premut el núvol per arrossegar-lo però no deixem anar el ratolí, el núvol continuarà avançant cap a la dreta fora del ratolí.

Encara que després deixem anar el ratolí, l'esdeveniment

*MOUSE\_UP* ja no es produirà, ja que significa aixecar el botó del ratolí damunt del núvol (no fora d'aquest). D'aquesta manera ja no podrem deixar anar l'arrossegament del núvol.

Per a solucionar-ho podem associar a la funció *soltar* un altre esdeveniment de ratolí anomenat *ROLL\_OUT*, que significa prèmer el núvol però que el ratolí s'arrossegi fora d'ell. Podem crear el nou esdeveniment sota l'esdeveniment *MOUSE\_UP*.

Per ara els *listeners* que hem creat seran els següents:

```
nubel_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);
;
nubel_mc.addEventListener(MouseEvent.MOUSE_UP,soltar);
nubel_mc.addEventListener(MouseEvent.ROLL_OUT,soltar);
nubel_mc.addEventListener(Event.ENTER_FRAME,desplazar);
```

En aquest cas no cal que creem una funció nova, ja que aquest *listener* cridarà la mateixa funció *soltar* que havíem creat abans.

Si provem la pel·lícula veurem que si en algun moment deixem d'arrossegar el núvol, aquest acabarà per desaparèixer per la part dreta de l'escenari. Encara que deixi de visualitzar-se, la instància continuarà executant el codi.

El següent que programarem serà que quan hagi desaparegut per la part dreta, torni a aparèixer per la part esquerra de l'escenari.

Per fer-ho, introduirem una sentència condicional dins de la funció *desplazar*, de tal manera que, si es compleixen uns determinats requisits, el núvol es posicioni en la part esquerra de l'escenari.

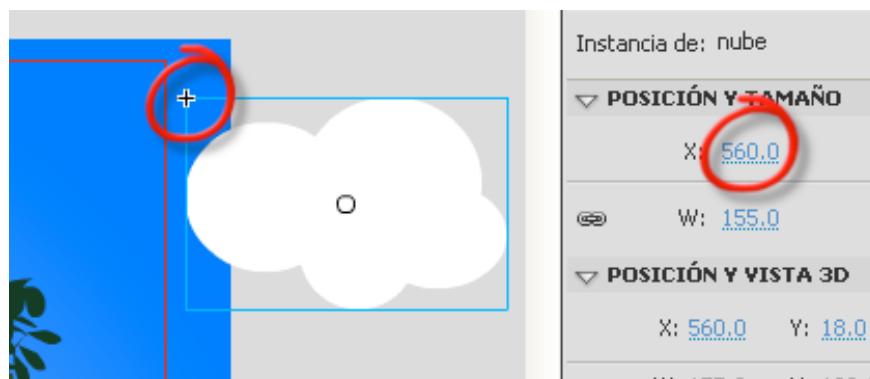
## Tutorial 6. Animació amb ActionScript 3.0

### Pas 12 de 19

En el nostre cas, definirem que si la posició x del núvol assoleix un valor determinat (quan hagi desaparegut per la dreta), que torni a la part esquerra de l'escenari.

Una manera ràpida de fer el càlcul de les posicions d'inici i de final és col·locar el núvol en un extrem i un altre de l'escenari, i anotar el valor d'x que apareix en l'inspector de *Propiedades*.

La posició del clip la marca una creu petita. Aquesta posició depèn del punt de registre que hagim seleccionat en crear el clip (per defecte és la part superior esquerra del clip).

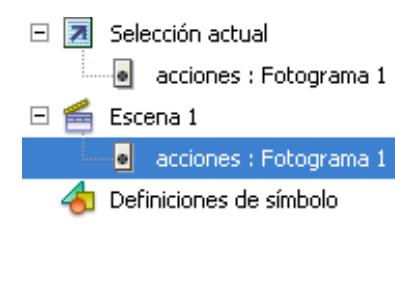


En aquest cas, seleccionarem com a extrem dret 560 (una mica més que l'amplada de l'escenari), i com a extrem esquerre -160 (una posició en la qual encara no es veu el núvol). Aquests valors dependran de la mida del vostre núvol.

Podríem fer aquests càlculs utilitzant com a dades l'amplada del clip i l'amplada de l'escenari. El motiu de no fer-ho d'aquesta manera és que en els pròxims passos afegirem un altre núvol al qual programarem profunditat en l'eix z, i llavors la posició x de l'objecte ja no serà equivalent als píxels de l'escenari, ja que es mourà en un escenari que simularà més profunditat i, per tant, més amplada que els 550 píxels del nostre escenari.

Col·loquem el núvol en ell lloc on vulguem que comenci l'animació i tornem al **tauler Acciones (F9)**. Recordem que podem navegar per les diferents accions de la nostra pel·lícula prement la línia corresponent en la part inferior esquerra del tauler.

En aquest cas, si no tingüéssim seleccionat el fotograma 1 de la capa *acciones* en la línia de temps, hi podríem anar prement *acciones*: fotograma 1



## Tutorial 6. Animació amb ActionScript 3.0

### Pas 13 de 19

La sentència *if*, que és la que utilitzarem, avalua si es compleix una condició, i en cas de complir-se executa les instruccions que indiquem entre les claus. L'estructura és la següent:

```
if (condició)
{
    //instruccions
}
```

Per tant, en la sentència, haurem d'especificar que si la posició x actual del núvol és més gran que 560 (el signe > significa més gran que), llavors, la posició hauria de ser -160.

La funció *desplazar* quedarà d'aquesta manera:

```
function desplazar(e:Event):void
{
    nube1_mc.x += .4;
    if (nube1_mc.x > 560)
    {
        nube1_mc.x = -160;
    }
}
```

Veiem que la funció té les seves pròpies claus, i que dins d'aquesta també hi ha un condicional amb les pròpies claus d'inici i de fi.

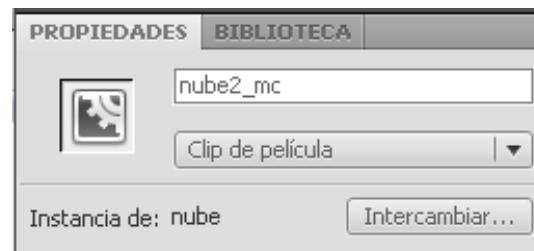
Provem ara la pel·lícula amb **Control > Probar película**. Veiem que, al cap de poc de desaparèixer per l'extrem dret, torna a aparèixer per l'extrem esquerre.

A diferència de la línia de temps principal, que té una extensió de 600 fotogrames, veiem que l'animació del moviment del núvol és independent d'aquesta extensió. En cada repetició del vol dels ocells, el núvol pot ser en un punt diferent.

Per a animar mitjançant programació n'hi ha prou amb un sol fotograma en la línia de temps. Si haguéssim fet aquesta animació en la línia de temps, hauríem necessitat estendre molt més el nombre de fotogrames.

## Tutorial 6. Animació amb ActionScript 3.0 Pas 14 de 19

Arrosseguem una altra instància del clip *nube* a l'escenari en la mateixa capa *nubes* que el núvol anterior, i en l'**inspector de Propiedades** li donem el nom d'instància *nube2\_mc*.



Al principi de la programació, després de la línia en la qual havíem definit l'alfa de la primera instància, afegim algunes propietats per a la nova instància.

```
nube1_mc.alpha = .6;  
nube2_mc.alpha = .4; //més transparent que el primer núvol  
nube2_mc.scaleY = .7; //per a disminuir verticalment (70%)  
nube2_mc.z = 300; //per a afegir profunditat
```

Després afegim els mateixos *listener* per a aquesta segona instància. Per a això n'hi ha prou de copiar i enganxar els *listeners* ja creats i substituir *nube1\_mc* per *nube2\_mc*:

```
nube1_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);  
nube1_mc.addEventListener(MouseEvent.MOUSE_UP,soltar);  
nube1_mc.addEventListener(MouseEvent.ROLL_OUT,soltar);  
nube1_mc.addEventListener(Event.ENTER_FRAME,desplazar);  
  
nube2_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);  
nube2_mc.addEventListener(MouseEvent.MOUSE_UP,soltar);  
nube2_mc.addEventListener(MouseEvent.ROLL_OUT,soltar);  
nube2_mc.addEventListener(Event.ENTER_FRAME,desplazar);
```

Dins de cada funció també fèiem referència a la instància *nube1\_mc*. En comptes de repetir les línies amb la programació per a *nube2\_mc* dins de les funcions, el que farem és que la programació faci referència a l'objecte que va iniciar l'esdeveniment.

És a dir, si un *listener* de la *nube1\_mc* ha cridat l'acció arrossegar, arrossegarem aquest núvol, però si l'acció ha estat iniciada per *nube2\_mc* serà aquest núvol el que arrosseguem.

## Tutorial 6. Animació amb ActionScript 3.0 Pas 15 de 19

Per a això hem d'escriure en primer lloc el nom del paràmetre de l'esdeveniment que hem posat en la funció (*e* en el nostre cas). Després afegim la propietat *target*. Amb això es farà referència a l'objecte que va disparar l'acció en rebre l'esdeveniment.

Per tant, si substituïm dins de les funcions el nom de la instància per *e.target*, aconseguirem tenir una referència directa als objectes que han enviat la funció.

Per a comprovar-ho en la funció *arrastrar*, a més de substituir *nube1\_mc* per *e.target*, afegirem un *trace* que ens mostri el nom (*name*) de la instància que desencadena l'esdeveniment.

```
function arrastrar(e:MouseEvent):void
{
    e.target.startDrag();
    trace(e.target.name);
}
```

Com que el contingut del *trace* no està entre cometes, la funció no escriu el valor literal que hem escrit entre parèntesis, sinó el seu valor (en aquest cas és un nom).

D'aquesta manera, veurem que en prémer cada núvol el tauler de *salida* ens mostra el seu nom.

Afegim *e.target* a totes les funcions. Ara tenim un problema amb les posicions que havíem determinat en el condicional *if*, ja que en tenir el segon núvol més profunditat hauríem de distanciar els dos extrems.

Una manera de conèixer aproximadament aquests valors és canviar el *trace* de la funció *arrastrar* perquè ens mostri la posició *x* del clip sobre el qual fem clic:

```
trace(e.target.x);
```

Després desactivem temporalment l'*if* de la funció *desplazar* convertint-lo en un comentari afegint /\* abans de l'*if* i \*/ després de la clau de tancament de l'*if*.

```
/*if (e.target.x > 560)
{
    e.target.x = -160;
}*/
```

## Tutorial 6. Animació amb ActionScript 3.0

### Pas 16 de 19

Ara provem la pel·lícula i tractem de prémer el segon núvol en posicions molt properes al límit de l'escenari per tots dos costats per fer-nos una idea dels valors d'*x* adequats.

El **tauler Salida** mostrarà el valor *x* per a aquest núvol, que com podem veure ha variat considerablement en estar programat per a mostrar-se en un pla més profund.

De fet, aquesta diferència entre la propietat *x* dependent de la profunditat també la podem veure en l'animació, ja que encara que tots dos núvols tenen assignat el mateix increment en el valor *x*, el més llunyà sembla avançar més a poc a poc que el més proper.

En el nostre cas, com a valor mínim seleccionarem -315 i com a valor màxim 715, però aquests valors dependran de la mida del vostre núvol.

Ara el primer núvol trigarà més a aparèixer. Podríem afegir un condicional i, dependent de si el *target* és el primer núvol o el segon, prendre uns límits o uns altres. En aquest cas no té importància que un núvol tardí més en aparèixer de nou, així que deixem com a límits els del núvol més llunyà.

Després d'esborrar el *trace*, ja que ja no el necessitem, les funcions quedaràn així:

```
function arrastrar(e:MouseEvent):void
{
    e.target.startDrag();
}

function soltar(e:MouseEvent):void
{
    e.target.stopDrag();
}

function desplazar(e:Event):void
{
    e.target.x += .4;
    if (e.target.x > 715)
    {
        e.target.x = -315;
    }
}
```

## Tutorial 6. Animació amb ActionScript 3.0

### Pas 17 de 19

Com a últim detall, farem que es mostri un cursor amb forma de mà quan estiguem sobre el núvol, que sembli un botó i que permeti intuir que el núvol es pot arrossegar.

Per fer-ho, abans dels *listeners* afegim les propietats següents:

```
nube1_mc.buttonMode = true;  
nube2_mc.buttonMode = true;
```

Amb això ja veurem que el cursor adopta la forma d'una mà quan estem a sobre d'un núvol, ja que tracta visualment els núvols com si fossin un *botó*.

Analitzem què passa amb la nostra pel·lícula. Comença en el fotograma 1 i el Flash llegeix tota la programació que hem escrit i la va executant. Després la línia de temps contínua durant 600 fotogrames, i torna al fotograma 1, on torna a llegir tota la programació.

Per a no sobrecarregar el Flash amb informació repetitiva, podem fer que el bucle de la línia de temps principal vagi del fotograma 600 al fotograma 2, sense tornar a passar pel fotograma 1, que és on hem escrit tota la nostra programació.

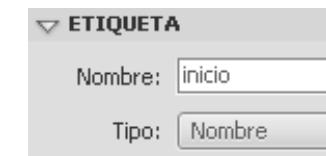
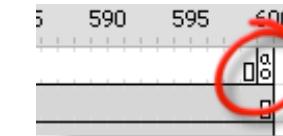
Per a això creem un fotograma clau en el fotograma 600 de la capa *acciones*, i escrivim la programació següent:

```
gotoAndPlay(2);
```

Això significa que quan el cap lector arribi al fotograma 600, on hi ha escrita la programació, anirà al fotograma 2 i continuará amb la reproducció.

Si haguéssim creat un fotograma clau i, en l'inspector de propietats, hi haguéssim assignat el nom *inicio*, també podríem haver anat a aquest fotograma escrivint:

```
gotoAndPlay("inicio");
```



Aquest pas no és necessari en aquest cas, però pot resultar útil conèixer aquesta opció per a crear fàcilment menús de navegació.

## Tutorial 6. Animació amb ActionScript 3.0

### Pas 18 de 19

Aquest és l'aspecte complet que tindrà la programació completa d'aquest tutorial:

```
nube1_mc.alpha = .6;
nube2_mc.alpha = .4;
nube2_mc.scaleY = .7;
nube2_mc.z = 300;
nube1_mc.buttonMode = true;
nube2_mc.buttonMode = true;

nube1_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);
;
nube1_mc.addEventListener(MouseEvent.MOUSE_UP,soltar);
nube1_mc.addEventListener(MouseEvent.ROLL_OUT,soltar);
nube1_mc.addEventListener(Event.ENTER_FRAME,desplazar);

nube2_mc.addEventListener(MouseEvent.MOUSE_DOWN,arrastrar);
;
nube2_mc.addEventListener(MouseEvent.MOUSE_UP,soltar);
nube2_mc.addEventListener(MouseEvent.ROLL_OUT,soltar);
nube2_mc.addEventListener(Event.ENTER_FRAME,desplazar);
```

```
function arrastrar(e:MouseEvent):void
{
    e.target.startDrag();
}

function soltar(e:MouseEvent):void
{
    e.target.stopDrag();
}

function desplazar(e:Event):void
{
    e.target.x += .4;
    if (e.target.x > 715)
    {
        e.target.x = -315;
    }
}
```

Tutorial 6. Animació amb ActionScript 3.0

**Pas 19 de 19**



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

1. Canviu la programació perquè els núvols es desplacin de dreta a esquerra.
2. Afegiu un núvol nou a l'escenari que també es desplaci, però que no es pugui arrosseggar.
3. Feu que els núvols es tornin més transparents en desplaçar-se, però que en aparèixer de nou per l'altre costat de l'escenari recuperin el valor alfa original.

## Tutorial 7. Control de la línia de temps

### Pas 1 de 11

En aquest tutorial afegirem dos botons per a reproduir i reprendre la nostra animació.

Aprendrem a controlar tant la línia de temps principal d'una pel·lícula com les línies de temps del clips imbricades en ella.

També aprendrem a controlar la reproducció d'animacions creades amb programació.

Per fer aquest tutorial, obrim l'arxiu creat en el tutorial anterior (*tutorial6.fla*) i el desem com a *tutorial7.fla*.



## Tutorial 7. Control de la línia de temps

### Pas 2 de 11

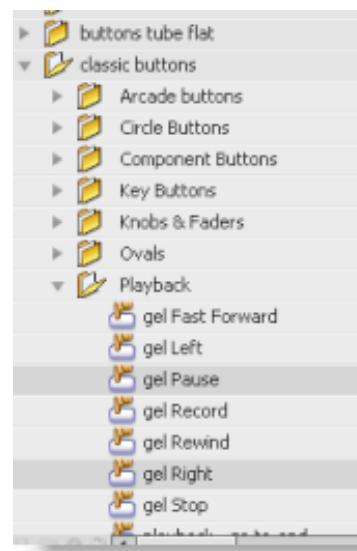
En primer lloc crearem una nova capa per sobre de l'herba, on col·locarem els botons amb els qual controlarem la línia de temps.



Per a aquests botons utilitzarem botons ja creats que podem trobar seleccionant **Ventana > Bibliotecas comunes > Botones**. En un altre tutorial aprendrem a modificar i crear els nostres propis botons, i n'explicarem en detall el funcionament.

Arrosseguem a l'escenari els botons *gel Pause* i *gel Right* que hi ha en la carpeta **classic buttons> Playback**.

Ara aquests botons apareixeran en la biblioteca del nostre document. Per tant, ja podem tancar la biblioteca dels botons.



Seleccionem la instància de *gel Pause* en l'escenari i li donem el nom d'instància *pausa\_btn* en l'**inspector de Propiedades**.

A la instància del botó *gel Right*, li donem el nom *play\_btn*.



Quan la línia de temps s'estigui reproduint volem que el botó per a interrompre la pel·lícula sigui visible, i quan la pel·lícula estigui en pausa necessitarem tenir visible el botó per a reprendre-la.

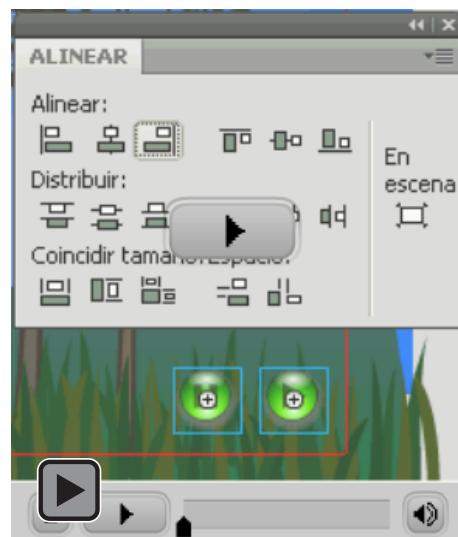
És a dir, farem que tots dos botons se situïn al mateix lloc, i que es mostri l'un o l'un altre dependent de l'estat de la reproducció, com si fos un commutador.

## Tutorial 7. Control de la línia de temps

### Pas 3 de 11

Seleccionem els botons amb la **tecla Mayúsculas** premuda per poder-los seleccionar tots dos alhora. Els situem a prop de la cantonada inferior dreta de l'escenari. Recordem que, en aquest cas, els límits de l'escenari estan definits pel marc vermell.

Amb l'ajuda del **tauler Alinear**, col·loquem tots dos botons al mateix lloc de l'escenari, un a sobre de l'altre.



Obrim el **tauler Acciones (F9)**. Escriurem després de l'última línia que hem programat en el tutorial anterior, en el fotograma 1 de la capa *acciones*.

Per diferenciar els dos tutorials i millorar la llegibilitat del codi, introduirem, a més d'alguna línia en blanc, un comentari que separi les dues parts del codi i que ens indiqui en quina zona de la programació som. Per exemple, podem escriure el comentari següent:

```
//botons de control de la línia de temps
```

Quan provem la pel·lícula, aquesta es reproduceix automàticament, per la qual cosa el botó que volem que sigui visible en un primer moment és el botó per a interrompre l'animació. Per tant, haurem de fer invisible el botó *play\_btn*. Per a això escriurem aquest codi:

```
play_btn.visible = false;
```

La propietat *visible* admet els valors *true* (visible) i *false* (invisible).

## Tutorial 7. Control de la línia de temps

### Pas 4 de 11

Un botó invisible no solament no es veurà, sinó que a més no estarà actiu. Tanmateix, un botó amb un valor 0 d'alfa es podrà prémer encara que no sigui visible. És per això que, quan volem desactivar un objecte a més de fer-lo invisible, sigui preferible utilitzar la propietat *visible* en lloc de la propietat *alpha*.

El pas següent és afegir detectors de l'esdeveniment de ratolí CLICK a tots dos botons, que s'anomenen botó *play\_btn* per a una funció que anomenarem *reproducir*, i botó *pausa\_btn* per a una funció anomenada *detener*:

```
pausa_btn.addEventListener(MouseEvent.CLICK,detener);
play_btn.addEventListener(MouseEvent.CLICK,reproducir);
```

La funció *detener* pararà la pel·lícula. Quan la pel·lícula estigui aturada, el botó per a reprendre la reproducció haurà de ser visible, mentre que ja no caldrà mostrar el botó per a interrompre-la.

En el cas de la funció *reproducir*, en primer lloc reprendrà la reproducció de la pel·lícula. Quan la pel·lícula es reproduueixi de nou, el botó per a interrompre-la haurà de ser visible, però ja no caldrà que el botó per a reproduir-la sigui visible.

Les funcions *stop()* i *play()* aturen i reproduueixen respectivament la línia principal de temps.

Per tant, de moment les funcions d'aturar i reproduir quedaran de la manera següent:

```
function detener(e:MouseEvent):void
{
    stop();
    play_btn.visible = true;
    pausa_btn.visible = false;
}

function reproducir(e:MouseEvent):void
{
    play();
    pausa_btn.visible = true;
    play_btn.visible = false;
}
```

Si hi ha cap dubte amb aquesta part d'aquest codi, recomanem repassar el tutorial anterior.

## Tutorial 7. Control de la línia de temps

### Pas 5 de 11

La línia `play_btn.visible = false;` que havíem col·locat abans de les funcions (pas 3) s'executarà directament en provar la pel·lícula quan el cap lector sigui en el primer fotograma. Tanmateix, la resta d'instruccions sobre la propietat visible, en ser dins d'una funció, només s'executarà quan es cridi la funció que les conté.

Seleccionem **Control > Probar pel·lícula (Ctrl+Intro)**.

Podem comprovar com s'alterna la visibilitat dels botons *play\_btn* i *pausa\_btn* quan els premem.

Veiem que prement *pausa\_btn* s'atura la línia de temps principal, és a dir, l'avanç dels ocells. Tanmateix, les ales continuen en moviment, ja que la funció *stop()* fa referència a la línia de temps principal, però no a les línies de temps imbricades dins d'ella (com en aquest cas les ales).

D'altra banda, també continua l'avanç dels núvols per l'escenari, ja que el seu moviment depèn de l'esdeveniment *enter frame*, i aquest esdeveniment és independent del desplaçament del cap lector en la línia de temps principal.

Per a solucionar el problema del moviment de les ales, primer haurem de saber com accedir-hi des de la programació en la línia de temps principal.

Per a accedir a una línia de temps imbricada dins d'una altra, en la programació escriurem el nom de la instància principal, seguida d'un punt, i seguida de l'element que està contingut en ella.

Per exemple, suposem que volem reproduir la línia de temps de la mà dreta que hi ha dins del clip del braç dret que al seu torn pertany a un cos. Per a això hauríem d'escriure el següent:

```
cuerpo.brazo_derecho.mano_derecha.play();
```

D'aquesta manera la funció *play()* faria referència a aquesta mà.

Per tant, per a poder accedir a les ales de cada ocell, en primer lloc haurem de donar un nom d'instància a cada ocell, i després a les ales.

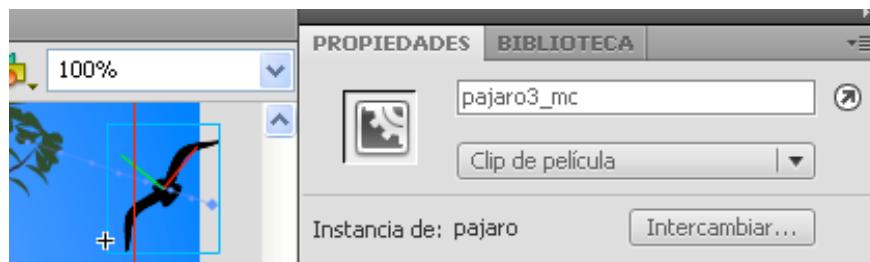
## Tutorial 7. Control de la línia de temps

### Pas 6 de 11

Per a poder donar un nom d'instància als ocells, en primer lloc desbloquegem les capes que els contenen.

Seleccionem cada ocell en l'escenari i els assignem els noms d'instància *pajaro1\_mc*, *pajaro2\_mc* i *pajaro3\_mc*, seguint la numeració dels noms de les capes en les quals es troben.

Per a seleccionar els ocells 2 i 3 haurem de desplaçar el cap lector, ja que no apareixen en l'escenari fins als fotogrames 24 i 150 respectivament.



Si fem clic dues vegades sobre qualsevol dels ocells de l'escenari, entrarem en l'edició del clip *pájaro* original.

A l'interior, aquest clip de pel·lícula principal té dues instàncies del clip de pel·lícula *ala*. Seleccionem l'ala esquerra i li donem el nom d'instància *ala1\_mc*. A l'ala dreta li donem el nom *ala2\_mc*.



Tornem a l'escena principal prement *Escena 1* en la barra d'edició, o bé seleccionant-la en el desplegable.



## Tutorial 7. Control de la línia de temps

### Pas 7 de 11

Si premem dues vegades qualsevol dels altres ocells, veurem que ens porta al mateix clip *pájaro* original, al qual ja hem donat els noms d'instància *ala1\_mc* i *ala2\_mc*.

Provem d'aturar una ala d'un ocell. Per a això haurem d'afegir la línia de codi següent dins de la funció *detener*,

```
pajaro1_mc.alal_mc.stop();
```

de tal manera que la funció *detener* quedi com s'indica a continuació:

```
function detener(e:MouseEvent):void
{
    stop();
    play_btn.visible = true;
    pausa_btn.visible = false;
    pajaro1_mc.alal_mc.stop();
}
```

Provem la pel·lícula i premem el botó *pausa\_btn*. L'ala esquerra del primer ocell s'aturarà alhora que la línia de temps principal. Encara que reprenguem la pel·lícula, aquesta ala no tornarà a reproduir-se, ja que no hem escrit instruccions per a això.

Abans de continuar, provem de substituir en la programació *pajaro1\_mc* per *pajaro3\_mc*, amb la qual cosa, la línia queda com s'indica a continuació:

```
pajaro3_mc.alal_mc.stop();
```

Si tornem a provar la pel·lícula, i premem el botó *pausa\_btn* abans que l'ocell 3 aparegui en escena, hi haurà l'error següent en el **tauler Salida**:

```
TypeError: Error #1009: No se puede acceder a una propiedad o a un método de una referencia a un objeto nulo.
```

```
at tutorial7_fla::MainTimeline/detener()
```

Això vol dir que, en prémer el botó per a aturar la pel·lícula i executar-se la funció *detener*, es fa referència a un objecte que no existeix en aquell moment (*pajaro3\_mc* en aquest cas), i per tant no pot executar la instrucció d'aturar l'ala d'aquest ocell.

Malgrat que això no impedeix el funcionament correcte de la nostra pel·lícula, per evitar aquest error comprovarem quins ocells són en l'escenari en el moment de prémer els botons, i només executarem les instruccions per aturar les ales d'aquests ocells.

## Tutorial 7. Control de la línia de temps

### Pas 8 de 11

Per a saber quins ocells hi ha en l'escenari en el moment de premer un botó cal saber en primer lloc en quin fotograma som en aquest moment.

Per a saber en quin fotograma és el cap lector en un moment donat, utilitzarem la propietat *currentFrame*.

Sabem que l'ocell 1 hi és des del primer fotograma i que desapareix en el fotograma 500. Per tant, si la propietat *currentFrame* torna un nombre més petit que 501, l'ocell 1 estarà en escena.

L'ocell 2 apareix en el fotograma 24 i desapareix en el 524, així que, si el cap lector és entre aquests dos valors, també podrem assegurar que l'ocell 2 és en l'escenari. De la mateixa manera, l'ocell 3 és present entre els fotrames 150 i 350.

Per considerar que un ocell està en escena no és necessari que sigui dins del marc visible de l'escenari, sinó que el que es té en compte és que estigui present en la línia de temps.

Si tenim en compte que < significa 'més petit que', > significa 'més gran que', i **&&** equival a l'*AND* lògic, la funció *detener* quedarà com segueix:

```
function detener(e:MouseEvent):void
{
    stop();
    play_btn.visible = true;
    pausa_btn.visible = false;

    if (currentFrame < 501)
    {
        pajaro1_mc.alal_mc.stop();
        pajaro1_mc.alal2_mc.stop();
    }
    if (currentFrame > 23 && currentFrame < 525)
    {
        pajaro2_mc.alal1_mc.stop();
        pajaro2_mc.alal2_mc.stop();
    }
    if (currentFrame > 149 && currentFrame < 351)
    {
        pajaro3_mc.alal1_mc.stop();
        pajaro3_mc.alal2_mc.stop();
    }
}
```

## Tutorial 7. Control de la línia de temps

### Pas 9 de 11

Explicarem un dels condicionals per a comprendre millor el codi.

Per exemple, amb la condició

```
if (currentFrame > 23 && currentFrame < 525)
```

comprovem si el fotograma actual és més gran que el fotograma 23 i, a més, si és més petit que el fotograma 525.

Si es compleixen tots dos requisits, llavors la condició completa es compleix, i s'executaràn les instruccions que hi hagi entre les claus d'aquest *if*. Si alguna de les parts de la condició no es compleix, no s'executaràn les instruccions corresponents.

Segons en quin fotograma siguem, pot ser que es compleixin les condicions que hem escrit per a tots els ocells (per exemple, el fotograma 200), només per a algun (per exemple, el fotograma 15), o per a cap dels tres (per exemple, el fotograma 580).

Si provem la pel·lícula, veurem que ara es paren correctament les ales dels tres ocells, i no se'n informa de cap error.

Ara ens falta tornar a activar el moviment de les ales quan tornem a reproduir la pel·lícula.

Per tornar a activar el moviment de les ales, copiem en la funció *reproducir* les sentències condicionals que hem utilitzat en la funció *detener*. Després, simplement substituïm el mètode *stop()* pel mètode *play()*, per exemple:

```
if (currentFrame < 501)
{
    pajaro1_mc.ala1_mc.play();
    pajaro1_mc.ala2_mc.play();
}
```

Tornem a provar la pel·lícula. Ara amb els botons *play\_mc* i *pausa\_mc* podem aturar i reprendre tant l'avanç dels ocells per l'escenari, com el moviment dels ales.

A més de parar amb *stop()* i reprendre amb *play()*, hi ha altres mètodes interessants a l'hora de controlar una línia de temps com, per exemple:

- *nextFrame()* i *prevFrame()*, per a avançar i retrocedir un fotograma
- *gotoAndPlay()* i *gotoAndStop()*, per a anar a un fotograma concret, i una vegada allà continuar o aturar-nos, i escriure dins dels parèntesis el número de fotograma al qual volem anar o el seu nom (etiqueta).

## Tutorial 7. Control de la línia de temps

### Pas 10 de 11

Fins aquí hem vist com podem controlar el cap lector tant de la línia de temps principal com de les línies de temps imbricades.

Ara ens queda aturar i reprendre el moviment dels núvols, ja que aquest moviment és independent del desplaçament del cap lector en la línia de temps principal.

El moviment de cada núvol l'havíem creat en el tutorial anterior amb el codi següent:

```
nube1_mc.addEventListener(Event.ENTER_FRAME,desplazar);  
nube2_mc.addEventListener(Event.ENTER_FRAME,desplazar);
```

Perquè els núvols s'aturin, n'hi haurà prou d'eliminar els detectors de l'esdeveniment *enter frame*. Per a eliminar-los n'hi ha prou de copiar el mateix codi que es va usar per a crear-los, i substituir *addEventListener* per *removeEventListener*.

```
nube1_mc.removeEventListener(Event.ENTER_FRAME,desplazar);  
nube2_mc.removeEventListener(Event.ENTER_FRAME,desplazar);
```

Per tant, afegirem les dues sentències amb *removeEventListener* a la funció *detener* abans de l'última clau que marca el final de la funció. Perquè el moviment es reprengui després, tornarem a afegir en la funció *reproducir* les dues sentències amb *addEventListener*.

Si, a més d'aturar el moviment, volguéssim que els núvols deixessin de ser arrossegables, hauríem de seguir el mateix procediment, és a dir, copiar la mateixa sentència que utilitzem per a agregar el *listener* i substituir *add* per *remove*. En aquest cas permetrem que els núvols puguin arrossegar-se encara que la línia de temps estigui aturada,

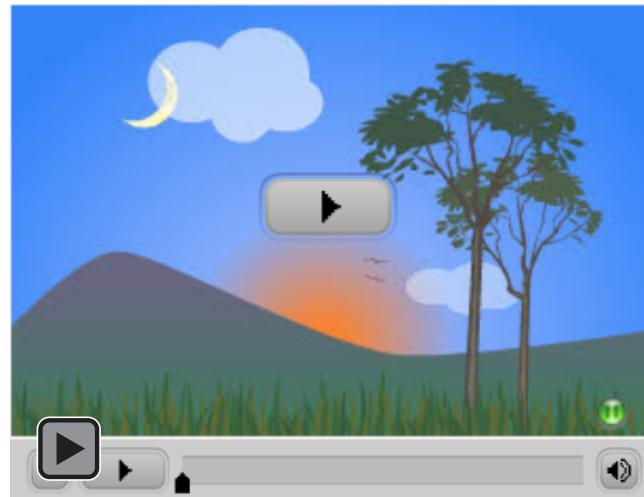
així que no eliminarem aquests *listeners*.

El mètode *removeEventListener* pot ser molt útil en diverses circumstàncies, per exemple, per a eliminar la interactivitat d'un botó o un clip sense necessitat d'amagar-lo ni eliminar-lo de l'escenari.

Seleccionem **Control > Probar pel·ícula**. Ara podem aturar i reprendre totes les animacions (ocells, ales i núvols).

Tutorial 7. Control de la línia de temps

**Pas 11 de 11**



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

1. Afegiu botons per a avançar i retrocedir un fotograma.
2. Afegiu un botó que vagi al principi de la pel·lícula.
3. Feu que els núvols es puguin arrosseggar només quan  
la pel·lícula estigui aturada.

## Tutorial 8. Afegir so a botons i a la línia de temps **Pas 1 de 14**

En aquest tutorial aprendrem a afegir un so de fons a la nostra pel·lícula. També afegirem so als nostres botons.

Crearem nous botons per a activar i desactivar el so general de la pel·lícula. Aquests nous botons els crearem duplicant els botons que ja teníem en la nostra biblioteca.

En primer lloc, obrirem l'arxiu *tutorial7.fla*, que hem creat en el tutorial anterior, i el desarem com a *tutorial8.fla*.

Els sons que utilitzarem són en la carpeta *tutorial8*. Per importar-los seleccionem **Archivo > Importar > Importar a biblioteca**, i seleccionem els arxius *fondoMusical.mp3* i *Mouse.mp3*.

Els sons passaran a formar part de la nostra biblioteca automàticament.



## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 2 de 14

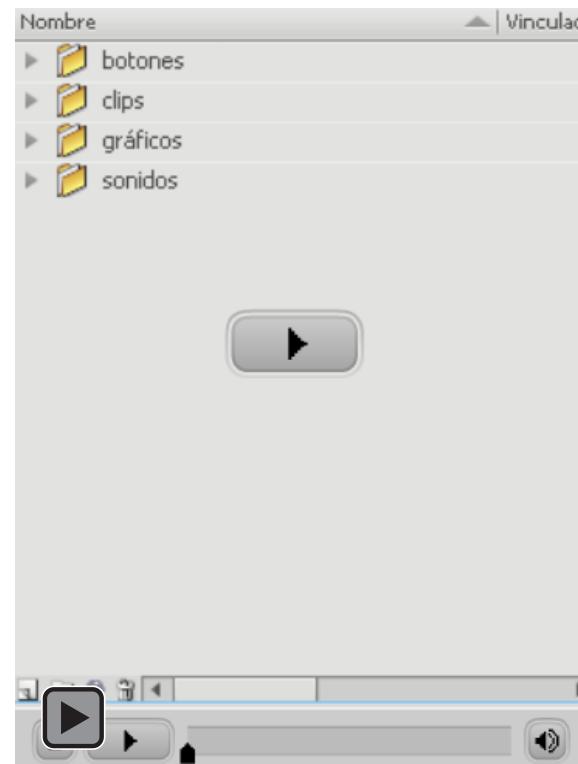
Organitzarem el contingut de la biblioteca en carpetes.

Per crear una carpeta premerem la icona inferior **Nueva carpeta**.

Crearem quatre carpetes, una per a gràfics, una altra per a clips, una altra per a botons i una altra per a sons.

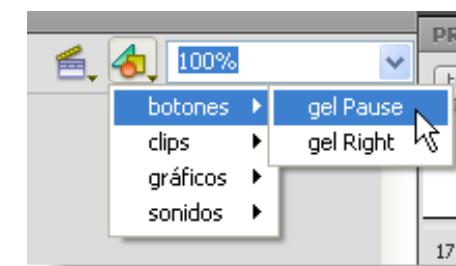
Seleccionem els diferents objectes que volem incloure en cada carpeta prement-hi a sobre amb la tecla **Ctrl** premuda per a fer una selecció múltiple.

Després arrossegarem els objectes seleccionats a la carpeta en què volem incloure'ls. El mapa de bits de la lluna també l'incluirem a la carpeta de gràfics.



Ara accedirem a la línia de temps dels botons per afegir-hi el so del clic del ratolí. Per a accedir a aquestes línies de temps tenim diversos camins possibles com, per exemple, buscar el símbol corresponent a la biblioteca i fer doble clic sobre el nom.

Una altra manera d'accedir a cada botó és seleccionant-ne el nom des de la icona **Editar símbolos** de la barra d'edició. Aquí apareixerà la mateixa estructura de carpetes que tinguem en la biblioteca.



També podem fer doble clic sobre l'objecte en l'escenari. Per poder accedir a un objecte que és sota un altre en la mateixa capa, premem l'objecte que és en primer pla amb el **botó dret** del ratolí, i seleccionem **Organizar > Enviar al fondo**.

## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 3 de 14

Accedim a la línia de temps del botó *gel Pause* utilitzant qualsevol de les formes descrites abans.

Fem zoom sobre l'escenari per veure amb més detall l'aspecte del botó.

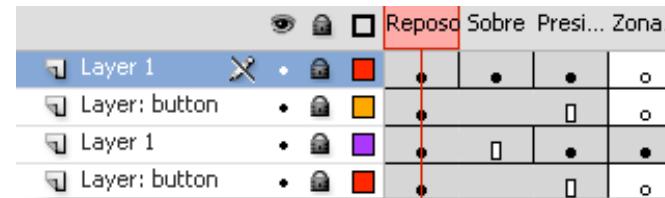
Les línies de temps dels botons tenen quatre fotogrames que representen diferents estats d'un botó:

· **Reposo.** Aspecte inicial del botó. Representa el moment en què el punter no és damunt del botó.

· **Sobre.** Aspecte del botó quan hi situem el punter.

· **Presionado.** Aspecte del botó quan hi fem clic.

· **Zona activa.** Defineix l'àrea que respon al clic del ratolí. Aquesta zona no serà visible per a l'usuari final. Permet crear àrees de clic amb una forma o mida diferents de l'aspecte del botó. També permet que en zones aparentment buides es pugui clicar.



Els botons, igual com altres línies de temps, poden tenir el nombre de capes que vulguem.

Els aspectes del botó poden variar d'un fotograma a l'altre o restar sense canvis. Aquí podem veure que entre el fotograma *Reposto* i el fotograma *Sobre* només hi ha canvis en la capa superior, en què un punt negre indica que hi ha un fotograma clau; mentre que en l'estat *Presionado* hi ha diferències en dues de les capes.

Tot i que podem afegir interactivitat tant als clips de pel·lícula com als botons, és la diferència en les característiques de la línia de temps el que fa que pugui ser més adequat utilitzar un tipus de símbol o un altre dependent de les nostres necessitats.

Si volem que, de manera senzilla i sense programació, el símbol mostri diferents aspectes dependent del seu estat (*reposto*, *sobre*, *presionado*), llavors els botons són una bona elecció.

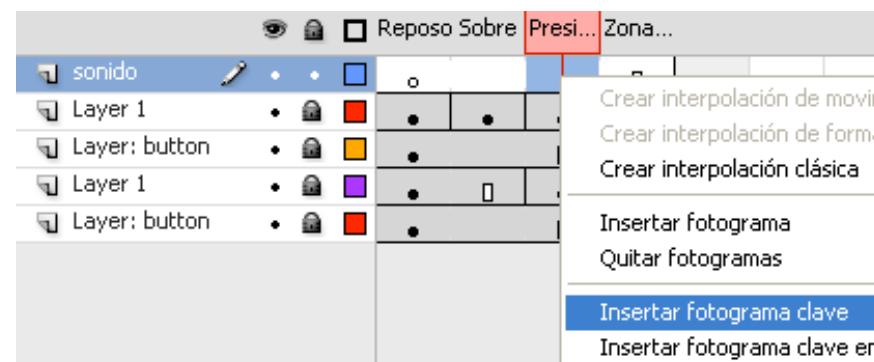
## Tutorial 8. Afegir so a botons i a la línia de temps Pas 4 de 14

Afegirem un petit so de clic en el moment de pressionar el botó.

En primer lloc, afegim una **nova capa** que anomenarem *sonido*.

Com que volem que el so se senti quan fem clic sobre el botó, inserirem un fotograma clau per a l'estat *Presionado*.

Per a això, fem clic amb el **botó dret** del ratolí en el fotograma *Presionado* de la capa *sonido*, i seleccionem **Insertar fotograma clave** en el menú contextual.



Amb aquest fotograma seleccionat, anem a l'àrea **Sonido** de l'**inspector de Propiedades**. Seleccionem en el desplegable **Nombre** l'arxiu Mouse.mp3 que ja tenim en la nostra biblioteca.

Com a **sincronització** triem **Evento**, que significa que el so es reproduirà completament quan el cap lector arribi al fotograma en el qual hi ha el so (en aquest cas, quan pressionem el botó).

Més endavant explicarem les diferents opcions de sincronització.

## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 5 de 14

La capa *sonido* mostrarà, començant en el fotograma *Presionado*, una línia blava petita que representa l'ona del so seleccionat per a aquest fotograma.

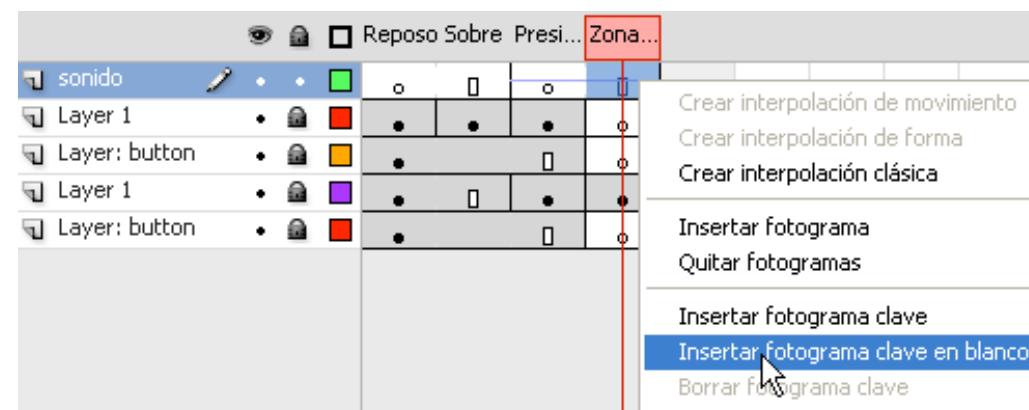
Podem observar que el so s'estén al fotograma de la *Zona activa*. En aquest cas, això no tindrà repercussió sobre el so, però per a més claredat inserirem un fotograma clau buit al fotograma de la *Zona activa* de la capa *sonido*.

Per fer-ho, seleccionem el fotograma amb el **botó dret** del ratolí, i seleccionem **Insertar fotograma clave en blanco**.

Repetim el mateix procés per afegir so al botó *gel Right* i provem la pel·lícula (**Control > Probar película**).

Ara se sentirà un petit clic quan premem qualsevol dels botons.

Tornem a la línia de temps principal de la pel·lícula, i afegim una nova capa per al so. Anomenem aquesta capa *sonido fondo*.



En l'inspector de propietats li assignem el so *fondoMusical.mp3*. Veurem com la línia blava de l'ona del so s'estén pels 600 fotogrames de la nostra pel·lícula.

## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 6 de 14

Analitzem les diferents opcions de sincronització que podem assignar a un so:

• **Esdeveniment.** El so començarà quan el cap lector arribi al fotograma que conté l'inici del so. Una vegada hagi començat, el so es reproduirà en la seva totalitat, independentment del cap lector. Per tant, n'hi ha prou que el so es trobi en un sol fotograma perquè es reproduixi en tota la seva extensió. Si el cap lector torna a llegir el fotograma que conté el so abans que aquest hagi acabat, començarà una nova reproducció del so que es barrejarà amb l'anterior.

• **Inici.** És similar a Esdeveniment, amb la diferència que si el so s'està reproduint prèviament, no s'iniciarà una nova reproducció.

• **Aturar.** Atura el so especificat.

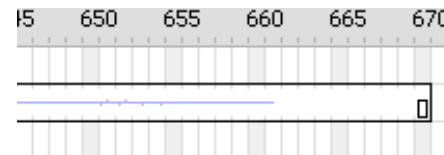
• **Flux.** El so estarà sincronitzat amb la línia de temps. Per tant, si la línia de temps està aturada, el so també s'aturarà. En aquest cas els fotogrames han d'estendre's durant tot el temps que volem que se senti el so. D'aquesta manera podem sincronitzar el so amb les imatges.

En l'**inspector de Propiedades** seleccionem una sincronització de tipus **Flujo** per al so *fondoMusical.mp3* i provem la pel·lícula.

Podem comprovar que la música s'atura i es reprèn juntament amb la resta de la línia de temps.

La durada de la pista de música és més gran que els 600 fotogrames disponibles, per la qual cosa no hi ha cap moment de silenci durant la reproducció.

Si la línia de temps tingués més fotogrames, podríem veure que la música s'estén fins al fotograma 661 aproximadament, i si l'animació continués més enllà d'aquests fotogrames, la música pararia.



La nostra línia de temps arriba al fotograma 600, i aquí el codi l'envia de nou al fotograma 2 amb la instrucció `gotoAndPlay(2)`. Amb la resta de la línia de temps, el so també torna a aquest fotograma. En aquest cas no encaixa musicalment el final del fotograma 600 amb l'inici al fotograma 2, per la qual cosa se sent un salt en la música. Per tant, en els casos de bucles de fons que no necessitin estar sincronitzats amb l'animació, l'opció de sincronització com a flux no és la més adequada.

## Tutorial 8. Afegir so a botons i a la línia de temps Pas 7 de 14

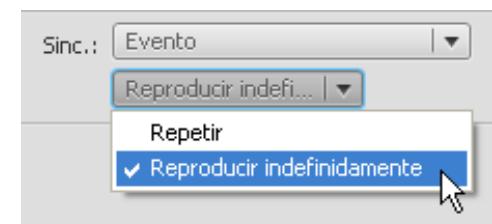
Canviem el tipus de sincronització de la pista a una sincronització de tipus **Evento** amb una repetició.



Tornem a provar la pel·lícula (**Ctrl+Intro**). Ara la reproducció del so és independent de la reproducció de la línia de temps. És per això que podem inserir un fotograma clau en blanc en el segon fotograma de la capa *sonido fondo*.



Perquè el so es repeteixi de manera contínua en un bucle, seleccionem el fotograma que el conté el so, i seleccionem **Reproducir indefinidamente**. Ara el so es reproduirà de manera contínua.



Si en el fotograma 600 no tinguéssim la instrucció *gotoAndPlay(2)*, la línia de temps tornaria al fotograma 1, fotograma en el qual hi ha l'inici del so de fons. En tenir seleccionada una sincronització tipus *Evento*, el so començaria de nou, sense que això vulgui dir parar la reproducció en curs, sonant de manera simultània. En aquest cas la sincronització més adequada seria *Inicio*.

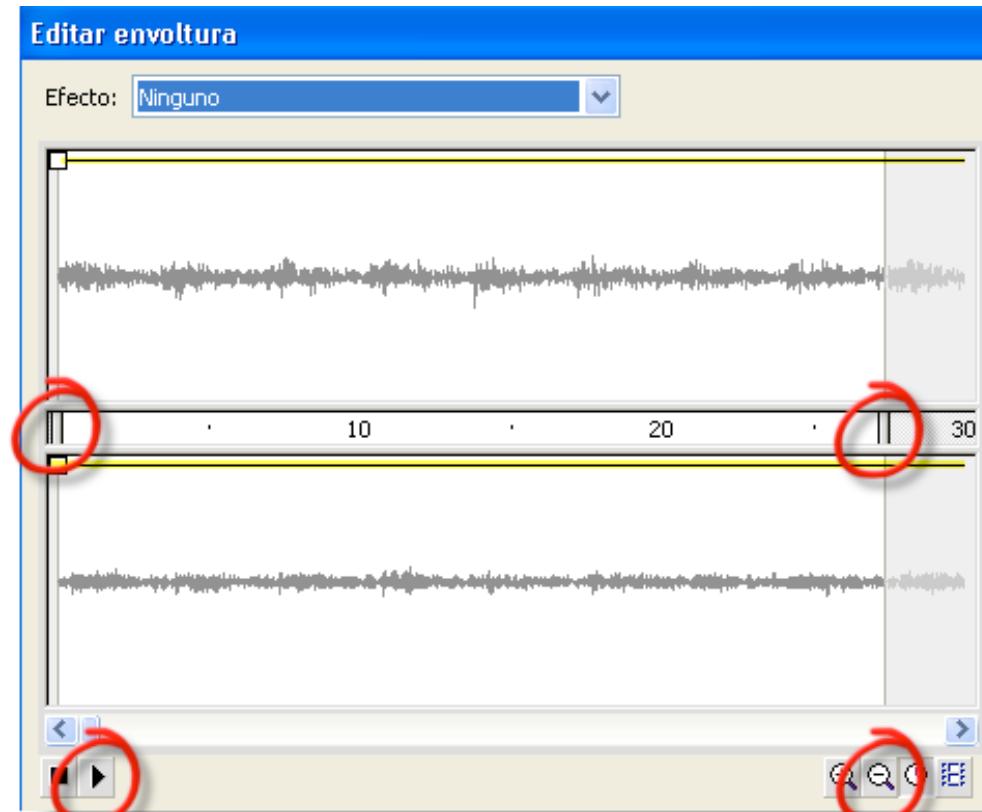
Com que en la nostra pel·lícula el cap lector només passa pel primer fotograma una vegada, podem mantenir la sincronització com a tipus *Evento*.

## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 8 de 14

En cada repetició del so, que dura uns 27 segons, podem escoltar una lleugera desincronització.

Podem arreglar aquest petit desajust amb l'ajuda de la pantalla **Editar envoltura**. Per obrir aquesta pantalla premerem la icona del llapis que hi ha al costat del desplegable **Efecto**.



En aquesta nova pantalla, podem ajustar el punt inicial i final del so. Podem reduir el zoom per a visualitzar tota l'ona de so.

Després del final de la pista d'àudio (segon 27), podrem veure en to gris el començament de la repetició següent. Provem amb el botó *play* els nostres ajusts. Quan ens agradi el resultat, farem clic a **Aceptar**. En aquesta pantalla també podríem haver ajustat els volums de tots dos canals seleccionant algun efecte predeterminat, o bé fent clic en la línia superior de cada canal per a crear nous punts de control que podem arrossegar per a controlar el volum.

## Tutorial 8. Afegir so a botons i a la línia de temps Pas 9 de 14

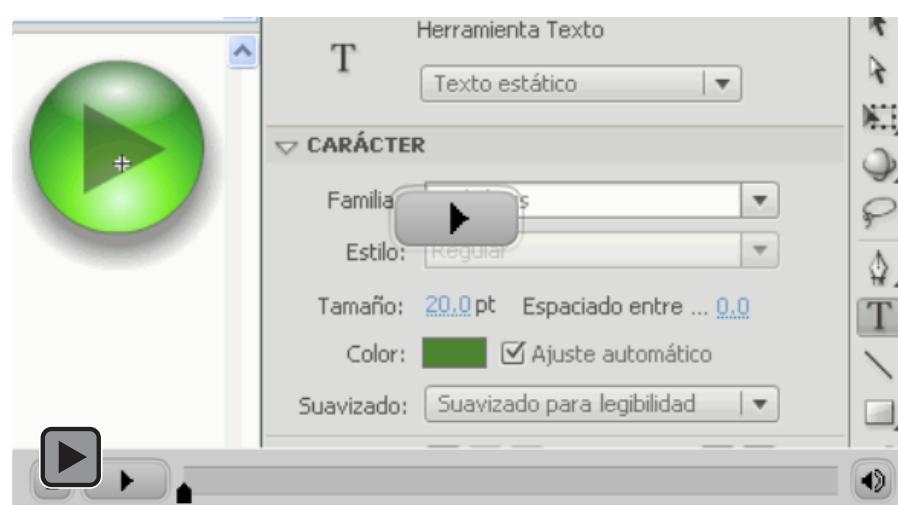
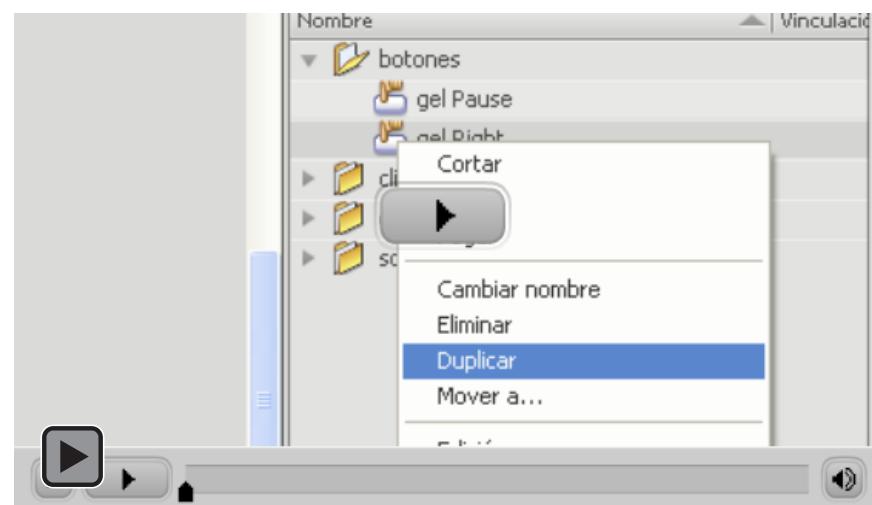
De vegades, el so de fons pot resultar molest, així que afegirem dos nous botons per a activar i desactivar el so de la pel·lícula.

Perquè siguin similars als botons que ja tenim, el més ràpid és duplicar un dels botons, i després transformar-lo perquè s'adapti a les nostres necessitats.

En la biblioteca, seleccionem el botó *gel Right* amb el **botó dret** del ratolí, i seleccionem **Duplicar** en el menú contextual. Al nou botó li donem el nom de *gel Volumen*.

Per a editar aquest símbol fem **doble clic** sobre el nom (*gel Volumen*) en la biblioteca. Substituirem la icona de *play* en el fotograma *Reposo* per la icona d'un altaveu. Activem la capa *Layer4*, que és la que conté la icona de *play*.

Per dibuixar aquest altaveu, seleccionem l'**eina Texto, família Webdings** i **mida 20 pt**. Per seleccionar el color, punxem la casella de **Color**, i després el comptagotes sobre la icona perquè desi el color. Finalment, fem clic en l'escenari per a crear un quadre de text, i premem la **tecla X majúscula**, que és la icona de l'altaveu.



## Tutorial 8. Afegir so a botons i a la línia de temps Pas 10 de 14

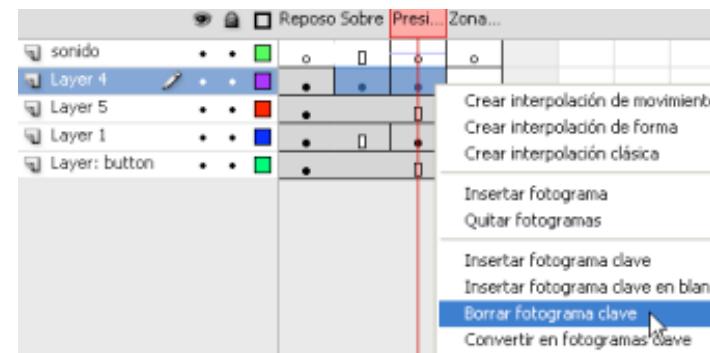
Amb el botó dret del ratolí seleccionem la icona de l'altaveu que hem creat, i seleccionem **Separar** per convertir-lo en una forma.

Després esborrem la icona del *play*, i situem la icona de l'altaveu en el centre del botó.



Si observem els fotogrames *Sobre* i *Presionado*, veurem que la icona *play* apareixia en color blanc en aquests fotogrames.

Seleccionem els fotogrames *Sobre* i *Presionado* on encara hi ha la icona de *play* i, fent clic amb el **botó dret** del ratolí, seleccionem **Borrar fotograma clave**.



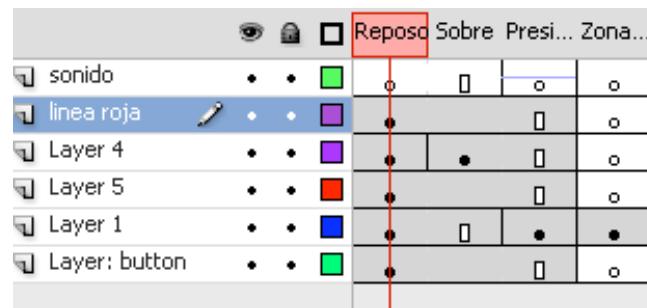
Ara tindrem la icona de l'altaveu en el fotograma *Reposo*, que es mostrarà en els tres fotogrames. Fem clic sobre el fotograma *Sobre* amb el botó dret del ratolí i seleccionem **Insertar fotograma clave** per crear un fotograma clau amb el mateix contingut que el fotograma *Reposo*.

Seleccionem l'altaveu en el fotograma *Sobre* i li canviem el color a blanc en l'**inspector de Propiedades**. Aquest color es mantindrà sense canvis en el fotograma *Presionado*.

## Tutorial 8. Afegir so a botons i a la línia de temps Pas 11 de 14

Per a crear el botó que mostri que l'àudio està desactivat, **duplicarem** el botó *gel Volumen* que acabem de crear, i anomenarem el botó nou *gel Volumen Off*.

Després afegim una **nova capa** per sobre de la capa *Layer4*, i dibuixem una línia vermella transversal que es mantingui durant els fotogrames *Reposo*, *Sobre* i *Presionado*.



Arrosseguem instàncies dels dos botons en l'escenari, en la capa *botones*.

A l'altaveu amb la línia vermella li donarem el **nom d'instància** *encender\_btn*, ja que quan es mostri estarem sense àudio i, per tant, si el premem serà per a tornar a encendre l'àudio. A l'altre botó li donem el nom d'instància *apagar\_btn*.

Amb l'ajuda del tauler **Alinear**, col·loquem un altaveu sobre l'altre perquè faci l'efecte de commutador, i tots dos al costat de l'altre botó.



## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 12 de 14

Afegim la programació al fotograma 1 de la capa *acciones*, sota la programació que hem creat en tutorials anteriors, que serà com s'indica a continuació:

```
encender_btn.visible = false;
encender_btn.addEventListener(MouseEvent.CLICK, encender);
apagar_btn.addEventListener(MouseEvent.CLICK, apagar);

function encender(e:MouseEvent):void
{
    apagar_btn.visible = true;
    encender_btn.visible = false;
    SoundMixer.soundTransform = new SoundTransform(1);
}

function apagar(e:MouseEvent):void
{
    encender_btn.visible = true;
    apagar_btn.visible = false;
    SoundMixer.soundTransform = new SoundTransform(0);
}
```

Com podem veure, la part de la programació que es refereix a mostrar o ocultar cada botó és similar a la dels botons *play\_btn* i *pausa\_btn*, que hem fet en el tutorial anterior.

Les úniques línies noves estan relacionades amb una classe anomenada *SoundTransform*:

```
SoundMixer.soundTransform = new SoundTransform(1);
SoundMixer.soundTransform = new SoundTransform(0);
```

La utilització de les classes excedeix els límits d'aquest tutorial. Tanmateix, és útil saber que simplement amb aquesta línia de codi podem modificar el volum general de tots els sons d'una pel·lícula. El volum varia entre 0 (apagat) i 1 (volum màxim). Aquests nombres els col·locarem en el parèntesi del final de la línia.

Provem la pel·lícula i comprovem que si premem el botó per a apagar l'àudio, deixarà de sonar no solament la pista de la música de fons, sinó també el clic dels botons.

## Tutorial 8. Afegir so a botons i a la línia de temps

### Pas 13 de 14

Per a optimitzar el pes final de la pel·lícula, podem variar la qualitat de compressió dels arxius d'àudio que utilitzem.

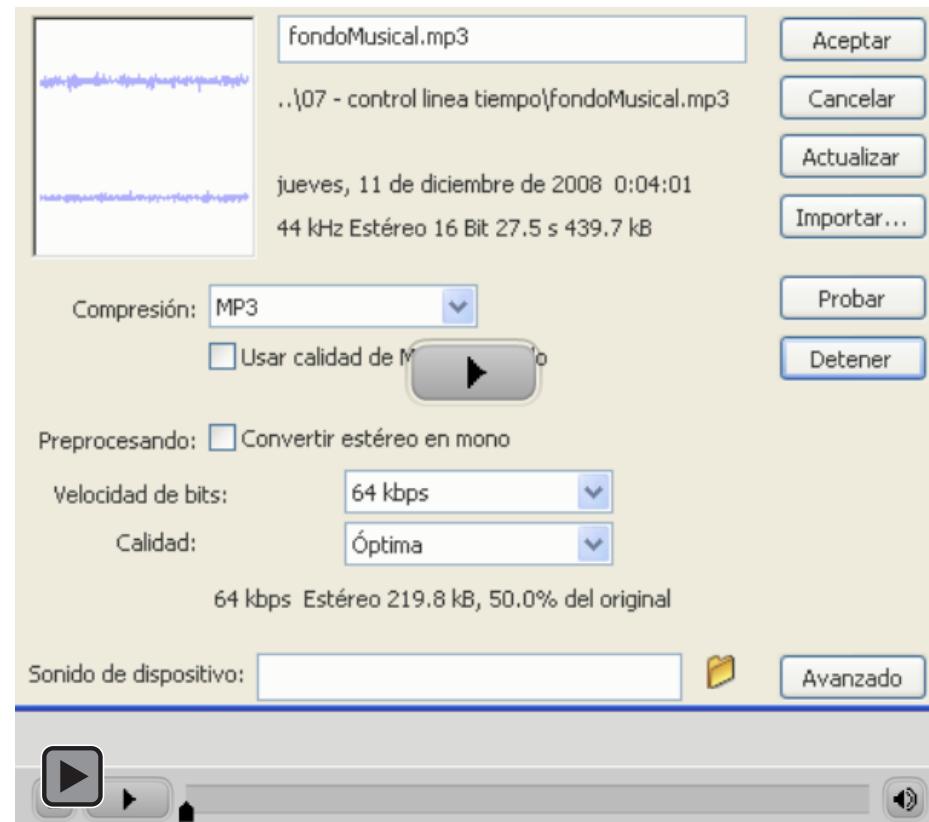
Una manera de fer-ho és anant a la finestra **Propiedades de sonido**, a la qual s'accedeix prement el **botó dret** del ratolí sobre el nom del so a la **biblioteca** (*fondoMusical.mp3* en aquest cas), i seleccionant **Propiedades** en el menú contextual.

En primer lloc, **desactivem** la casella **Usar calidad de MP3 importado**. És possible que ens aparegui una velocitat de 16 kbps i qualitat ràpida. Seleccionem **Probar** i escoltem el principi del so. Després premem **Detener**. Farem diverses proves amb diferents valors fins a trobar una qualitat acceptable sense gaire pes.

La qualitat es refereix a la velocitat amb què es comprimeix l'arxiu, com més velocitat, menys qualitat en la compressió. És per això que és recomanable utilitzar una qualitat òptima.

Sota de la casella de qualitat podrem veure el pes de l'arxiu amb les condicions actuals de compressió.

En aquest cas, una opció adequada podria ser un arxiu estèreo, amb una velocitat de bits de 64 kbps i qualitat òptima.



Tutorial 8. Afegir so a botons i a la línia de temps

**Pas 14 de 14**



Per a complementar els conceptes desenvolupats en aquest tutorial,  
es recomana fer l'activitat següent:

1. Creeu en l'escenari un clip que contingui un text amb informació sobre la pel·lícula, i feu-lo invisible a l'inici de la programació.
2. Afegiu-hi botons per mostrar i ocultar aquesta informació.

## Tutorial 9. Creació d'un joc (I)

### Pas 1 de 26

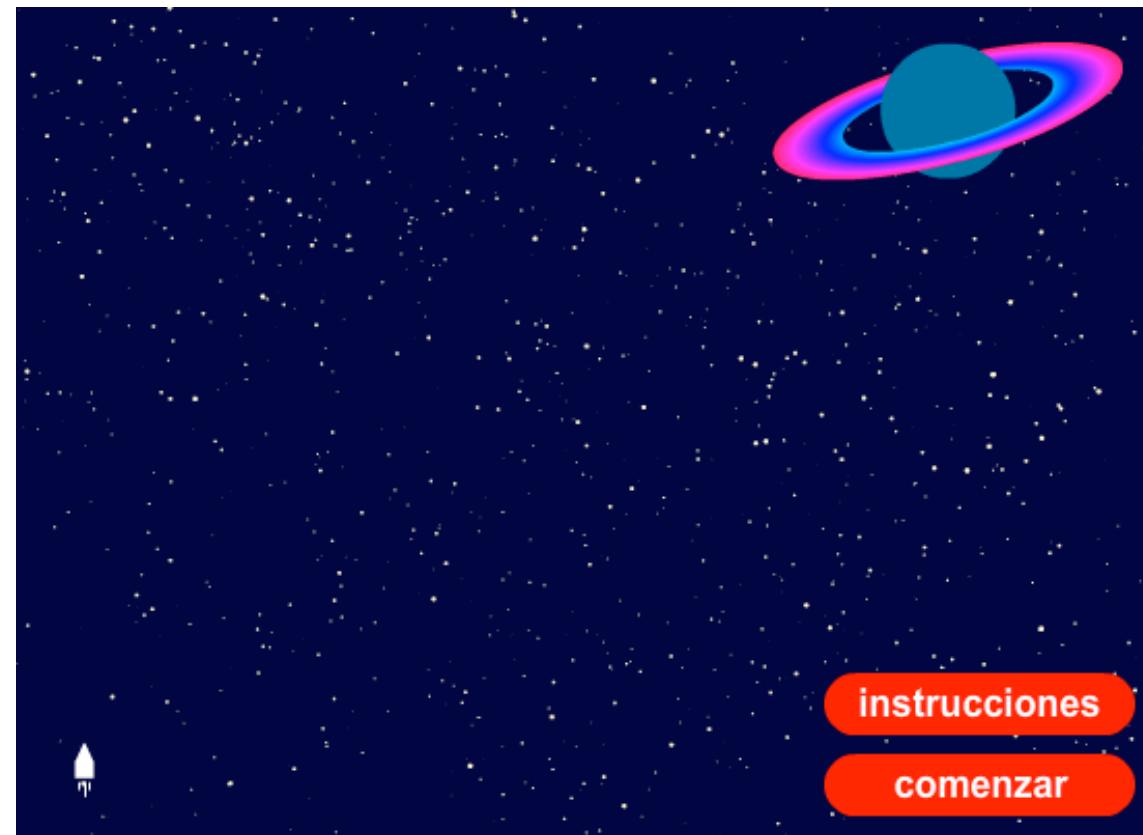
En aquest tutorial crearem la primera part d'un joc en el qual hem de portar una nau fins a un planeta, amb l'ajuda de les fletxes del teclat.

Començarem creant els elements gràfics necessaris per al joc, utilitzant eines com l'oval i el rectangle simple, i el pinzell ruixador.

Referent a la programació, aprendrem a controlar un objecte amb el teclat, i també a detectar-ne la posició o saber si xoca amb algun altre objecte.

També aprendrem a crear botons que ens mostrin informació quan hi situem el punter a sobre.

En el tutorial següent afegirem complexitat al nostre joc afegint obstacles amb moviments aleatoris.



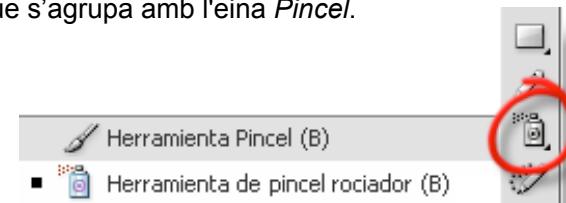
## Tutorial 9. Creació d'un joc (I) Pas 2 de 26

Creem un arxiu nou del **Flash (AS 3.0)** des de l'àrea **Crear nuevo** de la pantalla de benvinguda o bé seleccionant **Archivo > Nuevo**. Desem l'arxiu com a *tutorial9.fla*.

Canviem el **color de l'escenari** en l'inspector de propietats. Hi assignem un color blau fosc (#000033).

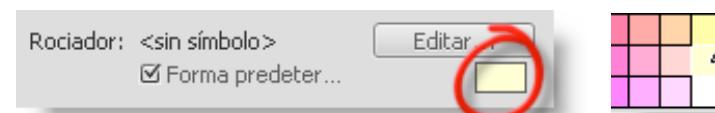


En primer lloc, crearem els elements gràfics del nostre joc, començant pel fons d'estrelles. Per a crear aquest fons utilitzarem l'eina **Pincel rociador** que s'agrupa amb l'eina **Pincel**.



Aquesta eina emet de manera predeterminada un esprai de punts de partícules amb el color que seleccionem.

Amb l'eina **Pincel rociador** seleccionada, en l'**inspector de Propiedades** canviem el color de la partícula a un groc clar (#FFFFCD).



Amb el botó *Editar* podríem seleccionar com a partícula emesa un símbol que tinguéssim en la biblioteca. En aquest cas mantindrem la forma predeterminada.

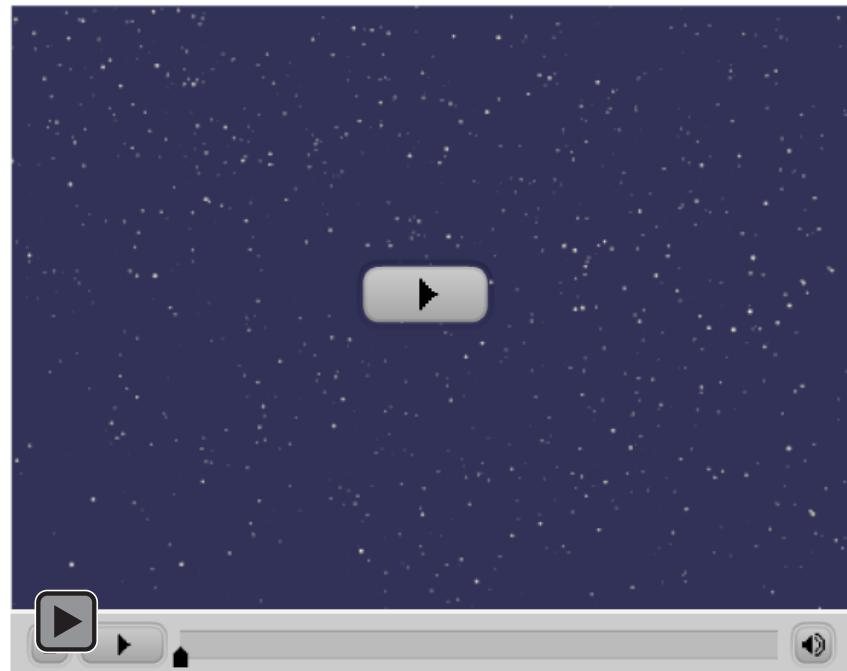
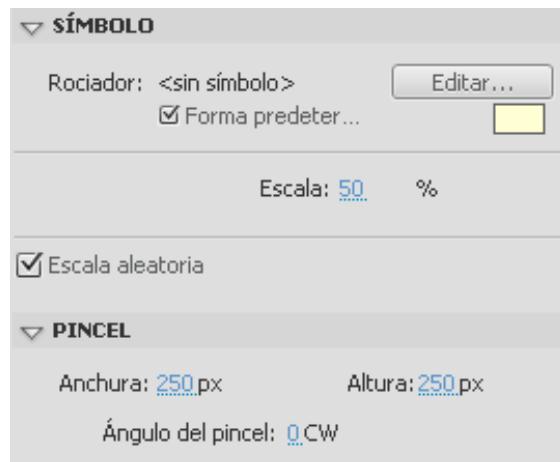
## Tutorial 9. Creació d'un joc (I)

### Pas 3 de 26

Canviem l'**Escala** a 50%. L'escala es refereix a la mida de les partícules emeses.

Seleccionem la casella **Escala aleatoria**. Això fa que les partícules siguin de diferents mides, encara que la mida màxima serà la que hem indicat prèviament en la casella *Escala*.

Per acabar, en l'àrea *Pincel*, seleccionem una **amplada** i una **altura** de 250 px. Això farà que les partícules estiguin més distanciades entre elles.

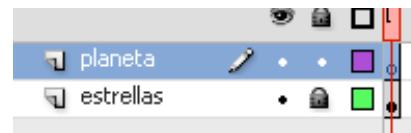


**Premem** el pinzell ruixador i l'arrosseguem sobre l'escenari fins a aconseguir un fons estrellat que ens agradi.

Anomenem aquesta capa **estrelles**. Després bloquegem la capa per evitar canvis accidentals.

## Tutorial 9. Creació d'un joc (I) Pas 4 de 26

Creem una nova capa que anomenarem *planeta*.

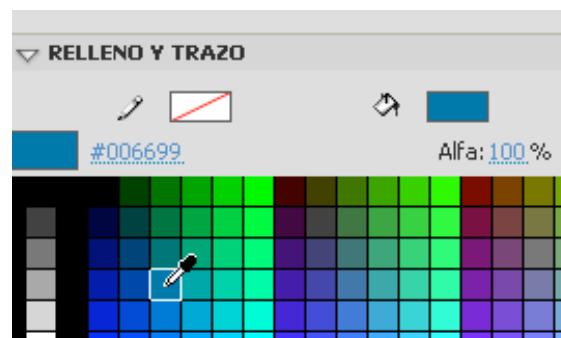


El nostre planeta estarà format per la part central i els anells, és a dir, contindrà més d'una capa. Perquè totes les capes formin part del mateix objecte, el més còmode és crear-les directament dins d'un mateix símbol.

Seleccionem **Insertar > Nuevo símbolo**, li donem el nom *planeta* i com a **Tipo** seleccionem **Clip de película**.

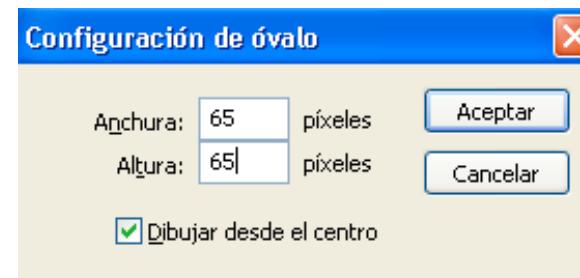
Dins del símbol planeta, dibuixem en primer lloc la part central del planeta, i per a això seleccionem l'eina **Óvalo**.

En l'inspector de **Propiedades** eliminem el color del traç i seleccionem com a color de farciment el color #006699.

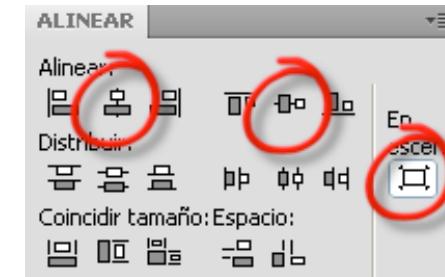


Amb l'eina **Óvalo** seleccionada, premem la tecla **Alt** i, sense deixar-la anar, fem clic sobre l'escenari.

S'obrirà la pantalla **Configuración de óvalo**, que permet definir directament la mida d'un oval abans de dibuixar-lo. En aquest cas li assignarem una amplada i una altura de 65 px.

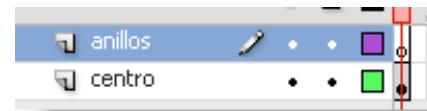


Seleccionem l'oval que hem creat i el centrem dins del seu propi escenari amb ajuda del taulell **Alinear**.



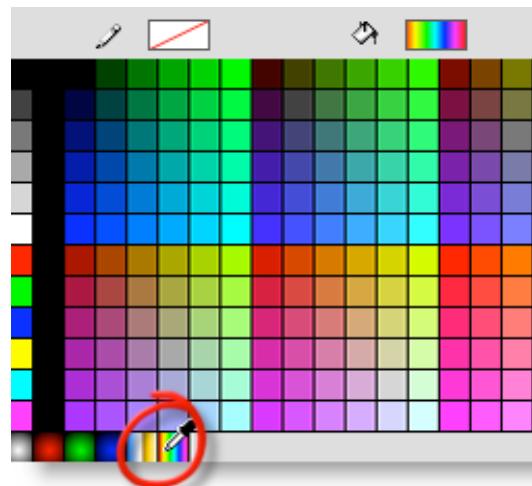
## Tutorial 9. Creació d'un joc (I) Pas 5 de 26

Anomenem aquesta capa *centro*, i afegim una nova capa que anomenarem *anillos*.



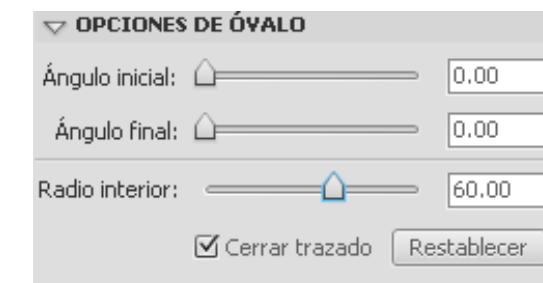
Aquesta vegada seleccionem l'eina **Óvalo simple** per poder modificar algunes característiques de l'oval després de dibuixar-lo.

Aquesta vegada tampoc no seleccionarem un traç en l'**inspector de Propiedades**. Com a color de farciment seleccionarem el degradat multicolor que apareix en la part inferior de la paleta.



Com en el pas anterior, premem la tecla **Alt** i, sense deixar-la anar, fem clic sobre qualsevol punt de l'escenari. De moment no és necessari que l'oval dels anells estigui centrat.

En la pantalla **Configuración de óvalo** assignem una amplada i una altura de 150 px. Acceptem per crear l'oval.



A **Opciones de óvalo** establim un **Radio interior** de 60. Recomanem provar les opcions d'angle inicial i final per a conèixer les possibilitats que ofereix, encara que finalment deixarem els angles a 0.

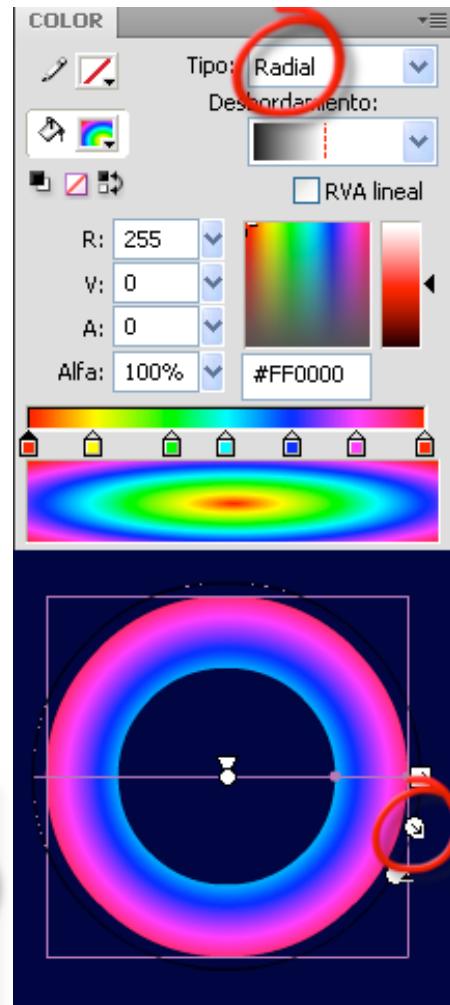
## Tutorial 9. Creació d'un joc (I)

### Pas 6 de 26

El farciment multicolor que hem utilitzat és per defecte un degradat lineal.

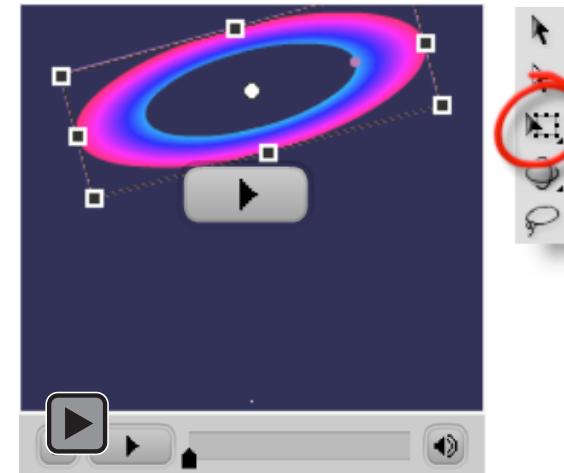
Per convertir el degradat en radial, obrim el **tauler Color** (*Ventana > Color*) i, amb l'oval dels anells seleccionat, canviem el **Tipo** a **Radial**.

Si volem canviar els colors dels anells, podem provar amb l'eina **Transformación de degradado**, i canviar l'amplitud del degradat perquè es mostrin diferents colors.

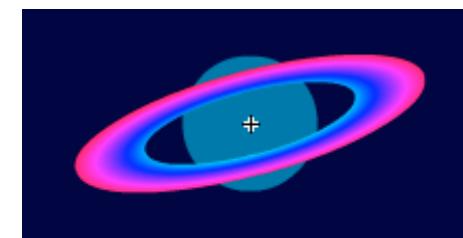


Amb l'eina

**Transformación libre**  
modifiquem la forma  
dels anells, tal com es  
mostra en el vídeo.



Centrem els anells en l'escenari amb ajuda del **tauler Alinear**. La part central del planeta haurà de sortir lleugerament tant per sobre com per sota dels anells. Si no és així, farem les rectificacions pertinentes en els anells amb l'eina *transformación libre*.



## Tutorial 9. Creació d'un joc (I) Pas 7 de 26

La part central del planeta queda completament per darrere dels anells. Tanmateix, perquè sembli que aquesta part està dins dels anells, una part del planeta s'hauria de veure per davant dels anells, i una altra part per darrere.

Per a aconseguir aquest efecte, duplicarem el centre del planeta en una nova capa superior, que podem anomenar *centro*.

Copiem l'oval del centre del planeta fent-hi clic a sobre amb el

**botó dret** del ratolí  
i seleccionant **Copiar**.



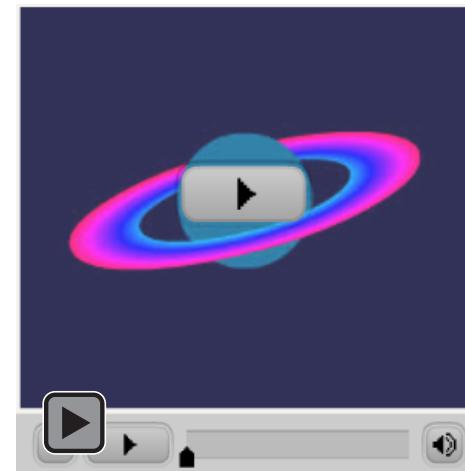
Bloquegem les dues capes inferiors i, amb la capa superior activa, fem clic amb el botó dret sobre l'escenari i seleccionem **Pegar in situ**.

Aquesta funció copiarà el dibuix del centre del planeta en la mateixa posició que tenia en la capa inferior.

Si ara esborrem de la capa superior una part del planeta, podrem veure el que hi ha sota aquesta part, és a dir, els anells en primer lloc i, al darrere, el centre del planeta en la capa inferior.

Provem d'eliminar la part superior o inferior del centre del planeta en la capa superior, segons la perspectiva que vulguem donar al nostre planeta (vist des de dalt o vist des de baix).

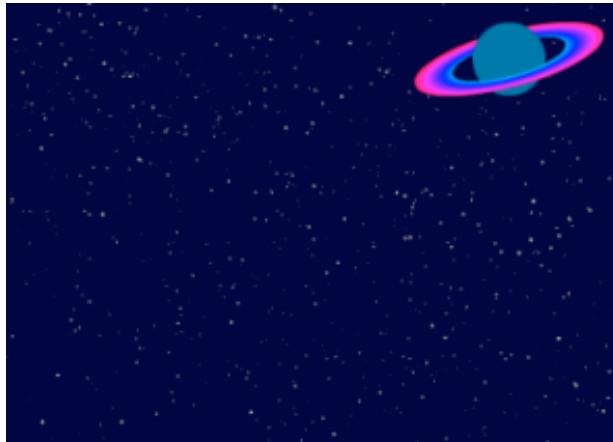
Observeu el vídeo per entendre millor el que passa quan s'esborra una zona del planeta en la capa superior. Trieu la perspectiva que més us agradi.



## Tutorial 9. Creació d'un joc (I) Pas 8 de 26

Tornant a l'escena principal, i amb la capa *planeta* activa, arrosseguem des de la biblioteca una instància del clip *planeta*.

Situem el clip en la part superior dreta de l'escenari, i li donem el nom d'instància *planeta\_mc*.



Creem una nova capa anomenada *nave* en la qual situarem la petita nau que controlarà l'usuari.

Aquesta nau tindrà una sola capa, així que podem crear-la a la capa *nave* de l'escena principal.

Dibuixem una nau d'aspecte similar al d'aquesta imatge. Per a fer-ho podem utilitzar eines com el rectangle i la línia.



La imatge està molt ampliada. L'altura total de la nau ha de ser aproximadament de 25 px, i l'amplada d'uns 10 px.

Una vegada dibuixada, seleccionem **Modificar > Convertir en símbolo** o premem **F8**. Li donem al clip el nom *nave*, i com a **Registro** li assignem el punt central. Això farà que el clip quedi centrat en el seu propi escenari..



Situem la nau en la part inferior esquerra de l'escenari. En l'**inspector de Propiedades** li assignem el nom *nave\_mc*.

Amb aquests dos clips ja podem començar la programació del nostre joc. Creem una nova capa anomenada *acciones* i, amb el primer fotograma d'aquesta capa seleccionat, premem **Ventana > Acciones** o **F9**.

## Tutorial 9. Creació d'un joc (I)

### Pas 9 de 26

Volem que la nau giri i es desplaci quan premem les fletxes del teclat, així que el primer pas serà convertir la nau en un detector d'esdeveniments del teclat.

Programem que, quan es detecti una pulsació de tecla, s'executi una funció anomenada *flechas*. La manera d'afegir un esdeveniment de teclat és similar al que hem vist fins ara.

```
nave_mc.addEventListener(KeyboardEvent.KEY_DOWN, flechas);  
  
function flechas(e:KeyboardEvent):void  
{  
    trace ("He pulsado una tecla");  
}
```

Amb aquesta programació diem a la nau que, quan detecti que s'ha premut una tecla, executi una funció anomenada *flechas*. Aquesta funció, que rep un esdeveniment de teclat, té al seu torn l'ordre d'escriure en el tauler *Salida* la frase *He pulsado una tecla*.

No obstant això, si provem la pel·lícula i premem les fletxes del teclat no passarà res.

Això es deu al fet que, perquè funcioni un detector d'esdeveniments de teclat, l'objecte encarregat de detectar-los ha de rebre el focus de la pel·lícula, és a dir, ha d'estar seleccionat. Per a seleccionar-lo, a l'inici de la programació afegim el codi següent:

```
stage.focus = nave_mc;
```

Ara que la nau té el focus, detectarà les pulsacions del nostre teclat, i després de cada pulsació escriurà la frase que havíem posat en el *trace*. Quan estiguem en fase de proves, aquesta detecció no funcionarà amb les tecles que tinguem com a mètodes abreujats de teclat en el Flash. És per això que en la fase de proves també ens limitarem a les fletxes del teclat.

Com que ara la nau té el focus, podem veure que apareix un rectangle al voltant de la nau per a indicar que té el focus. Perquè no aparegui aquest rectangle, abans de donar el focus a la nau afegirem una sentència que especificui que, encara que la nostra nau tingui el focus, no mostri un rectangle. Les dues primeres línies de la nostra programació quedaran així:

```
nave_mc.focusRect = false;  
stage.focus = nave_mc;
```

## Tutorial 9. Creació d'un joc (I)

### Pas 10 de 26

Ara que hem provat que el *trace* funciona, i que la nau no mostra el rectangle del focus, procedirem a substituir el *trace* per les sentències que volem executar.

Volem avaluar quina fletxa s'ha premut, i executar accions diferents segons si s'ha premut una fletxa o una altra. Per a aquest tipus de problemes és molt útil una sentència anomenada *switch*.

Suposem el cas d'un test amb respostes *a*, *b* i *c*:

```
switch (respuesta)
{
    case a :
        trace("el sujeto ha elegido a");
        break;
    case b :
        trace("el sujeto ha elegido b");
        break;
    case c :
        trace("el sujeto ha elegido c");
        break;
    default :
        trace("ha ocurrido un error");
}
```

El que la sentència *switch* faria en aquest cas és avaluar la resposta (variable que hi ha entre parèntesis).

En cas que la resposta sigui *b*, s'analitzaran els casos fins a trobar-ne un que sigui correcte (*case b*), i llavors s'executaràn les sentències que hi hagi dins d'aquest *case* (en aquest cas un *trace* que informa que s'ha respost *a*), i ja no es revisaran els altres casos (*break* o trencament de servei).

Cada *case* ha d'acabar sempre amb una sentència *break* perquè s'ometi la resta de la sentència *switch* quan trobem un *case* correcte. Així evitem que comprovi la resta dels casos.

Si no hi ha cap *case* correcte, llavors s'executarà el que s'indiqui a *default*. Aquí les úniques respostes possibles són *a*, *b* o *c*, per la qual cosa si *respuesta* és qualsevol altra cosa, hem programat un *trace* que indiqui que hi ha hagut un error. Aquí no faria falta un *break*.

Sempre cal incloure *default*, encara que no prevegem que hagi de passar res fora dels *case*.

És important no oblidar que aquesta sentència també té les seves pròpies claus a l'inici i al final.

## Tutorial 9. Creació d'un joc (I)

### Pas 11 de 26

Avaluarem quina tecla ha detectat el *listener* que s'ha premut (*e.keyCode*). Segons la tecla premuda, farem girar el clip cap a un costat o cap a un altre. Si s'ha premut qualsevol altra tecla, no farem res, llevat sortir de la sentència. La sentència, que és dins de la funció *flechas* (substituint *trace*), quedarà com s'indica a continuació:

```
switch (e.keyCode)
{
    case Keyboard.RIGHT :
        e.target.rotation = 90;
        break;
    case Keyboard.LEFT :
        e.target.rotation = -90;
        break;
    case Keyboard.UP :
        e.target.rotation = 0;
        break;
    case Keyboard.DOWN :
        e.target.rotation = 180;
        break;
    default :
        break;
}
```

La rotació es mesura en graus (360 graus seria un gir complet). El clip girarà sobre el seu punt de registre, que en aquest cas és en el centre del clip.

Provem la pel·lícula (**Ctrl+Intro**). La nau ja s'orienta cap a un costat o cap a un altre depenent de la fletxa que hagim premut.

El pas següent serà afegir sentències perquè la nau, a més de girar, es desplaci. Per a aconseguir-ho podem utilitzar sentències similars a les que hem utilitzat per a desplaçar el núvol en el tutorial 6.

Per exemple, podem afegir les sentències següents en cada *case*, segons correspongui:

```
e.target.x += 3; //para que vaya 3px hacia la derecha
e.target.x -= 3; //para que vaya 3px hacia la izquierda
e.target.y -= 3; //para que suba 3px
e.target.y += 3; //para que baje 3px
```

El primer *case* quedaria, per tant, de la manera següent:

```
case Keyboard.RIGHT :
    e.target.rotation = 90;
    e.target.x += 3;
    break;
```

## Tutorial 9. Creació d'un joc (I) Pas 12 de 26

Completem els quatre case amb les sentències corresponents i tornem a provar la pel·lícula. És convenient provar la pel·lícula amb cada petit pas que fem, per a poder comprovar què fa exactament cada part del codi que anem afegint.

La nau ja es mou cap als quatre costats, depenent de la fletxa que premem. Tanmateix, el moviment no és gaire fluid.

Si volem que, en prémer una fletxa, el moviment cap a aquest costat es mantingui de manera contínua fins que en premem una altra, llavors necessitarem afegir un *enter frame*.

Afegim a la nau un detector de l'esdeveniment *enter frame*, que cridi una funció que anomenarem *navegar*:

```
nave_mc.addEventListener(Event.ENTER_FRAME, navegar);
```

La funció *navegar* ha de detectar l'últim moviment que s'ha activat, i continuar en la mateixa direcció. Per exemple, en cas que el moviment sigui cap a la dreta, llavors ha de continuar avançant cap a la dreta.

Crearem una variable anomenada *movimiento*, que emmagatzemi un text amb la direcció del moviment.

Les variables es creen amb l'estruatura següent:

```
var nombreVariable:tipoVariable = valor inicial;
```

El valor inicial és opcional. Podria posar simplement un punt i coma després de la declaració de la variable. Per a la llegibilitat del codi, és convenient declarar les variables que utilitzarem en la part superior del codi.

Per tant, afegirem aquesta primera línia al nostre codi:

```
var movimiento:String = "";
```

El tipus de variable *String* representa cadenes de caràcters. Les cadenes de caràcters sempre van entre cometes. Com a valor inicial hem creat una cadena buida (sense cap contingut).

## Tutorial 9. Creació d'un joc (I)

### Pas 13 de 26

Com que ara volem que els desplaçaments de la nau els faci una altra funció que rep un *enter frame* (la funció *navegar*), en la funció *flecha* només desarem la dada sobre cap a on s'ha de dirigir el moviment.

Per tant, substituïm les sentències *case* del desplaçament en la funció *flechas* de la manera següent:

```
case Keyboard.RIGHT :  
    e.target.rotation = 90;  
    movimiento = "derecha";  
    break;
```

En els altres casos, assignem a la variable *movimiento* els valors "izquierda", "arriba" i "abajo" respectivament.

La funció *navegar*, que crearem a continuació, mourà de manera contínua el clip cap a un costat o un altre dependent de l'últim valor assignat a la variable *movimiento*, ja que s'executarà contínuament quan la cridi un esdeveniment *enter frame*.

La funció *navegar* avaluarà el contingut de la variable *movimiento*, i a partir d'això mourà la nau.

```
function navegar(e:Event):void  
{  
    switch (movimiento)  
    {  
        case "derecha" :  
            e.target.x += 3;  
            break;  
        case "izquierda" :  
            e.target.x -= 3;  
            break;  
        case "arriba" :  
            e.target.y -= 3;  
            break;  
        case "abajo" :  
            e.target.y += 3;  
            break;  
        default :  
            break;  
    }  
}
```

El valor 3 és la quantitat de píxels que es desplaça la nau en cada *enter frame*. Per tant, és la velocitat de la nau.

## Tutorial 9. Creació d'un joc (I) Pas 14 de 26

Per tant, podríem crear una variable anomenada *velocidad*, que especifiqui el nombre de píxels en els quals varia la posició de la nau en cada moment.

Podem crear aquesta variable en la part superior de la programació, després de la declaració de la variable *movimiento*:

```
var movimiento:String = "";
var velocidad:Number = 3;
```

Dins de la funció *navegar*, substituïm el nombre de píxels per la variable *velocidadt* en els quatre *case*, que prendrà el valor que li hem assignat quan creem la variable:

```
case "derecha" :
    e.target.x += velocidad;
    break;
```

D'aquesta manera, si canviem el valor de la variable *velocidad*, canviarà la velocitat dels desplaçaments en les quatre direccions. Podem provar diferents valors fins que ens sembli una velocitat adequada.

El pas següent serà detectar si la nau arriba al planeta, que serà la meta del joc. Per a fer-ho, mentre la nau es desplaça hem d'anar comprovant si xoca amb el planeta.

Dins de la funció *navegar*, per sota de la clau de tancament de la sentència *switch*, afegirem aquest codi:

```
if (e.target.hitTestObject(planeta_mc))
{
    trace ("He llegado al planeta. He ganado.");
}
```

És un condicional que comprova si la nau (*e.target*) xoca (*hitTestObject*) amb el planeta (*planeta\_mc*). Si xoca, llavors es mostrarà la frase "*He llegado al planeta*" en el tauler *Salida*.

Afegirem un altre condicional, aquesta vegada per detectar si la nau surt fora dels límits de l'escenari, que mesura 550 x 400. Per tant, haurem de comprovar si la propietat *x* (posició horitzontal) de la nau té un valor inferior a 0 o superior a 550, i si la propietat *y* (posició vertical) té un valor per sota de 0 o per sobre de 400. En qualsevol dels quatre casos, la nau estarà posicionada fora de l'escenari.

## Tutorial 9. Creació d'un joc (I) Pas 15 de 26

Aquest nou condicional, escrit també dins de la funció *navegar*, quedarà de la manera següent (les barres verticals ||, que s'escriuen amb *Alt Gr+1*, equivalen a l'*OR* lògic):

```
if (e.target.x < 0 || e.target.x > 550 || e.target.y < 0  
|| e.target.y>400)  
{  
    trace("Estoy fuera del escenario. He perdido.");  
}
```

Provem la pel·lícula per comprovar si funciona correctament. La nau es desplaçarà per l'escenari depenent de les fletxes que premem en el teclat. Si arribem al planeta, apareixerà la frase que hem guanyat en el tauler *Salida*, i si som fora dels límits de l'escenari, apareixerà la frase que hem perdut.

Ara el joc comença directament, però seria adequat disposar d'una pantalla prèvia amb les instruccions del joc, i que en arribar al planeta (guanyar) o sortir de l'escenari (perdre) es mostrés una pantalla diferent i el joc acabés, donant opció a tornar a jugar.

Per tant, a continuació crearem quatre fotogrames diferents:

- Un fotograma inicial amb dos botons, un amb les instruccions del joc i un altre per a començar a jugar.
- Un segon fotograma en el qual es desenvoluparà el joc.
- Un tercer fotograma per a indicar que hem guanyat.
- Un últim fotograma per a indicar que hem perdut.

Aquests dos últims fotogrames tindran un botó per a tornar jugar.

Inserim una nova capa amb el nom *etiquetas*. Inserim **fotogrames clau** en els fotogrames 2, 3 i 4. Podem crear aquests fotogrames clau fàcilment si fem clic sobre cada fotograma i premem la tecla **F6**.



## Tutorial 9. Creació d'un joc (I)

### Pas 16 de 26

Premem sobre cada fotograma de la capa *etiquetas*, i assignem a cada un un nom d'etiqueta en l'**inspector de Propiedades**.

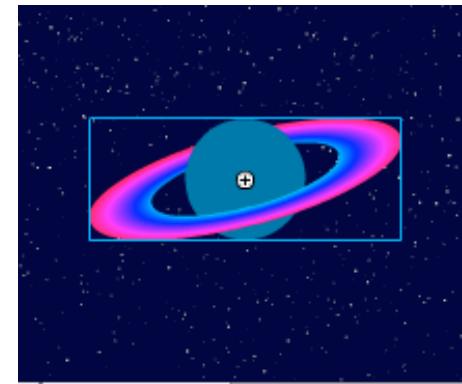
El primer l'anomenem *inicio*, el 2, *juego*, el 3, *ganar*, i el 4, *perder*.



Els fotogrames que tinguin una etiqueta mostraran una bandera vermella petita en la línia de temps.

La nau i el planeta hauran de mantenir-se visibles i sense canvis en els fotogrames *inicio* i *juego*, mentre que les estrelles es veuran com a fons en tots 4 fotogrames. En els fotogrames *ganar* i *perder* utilitzarem de nou una imatge del planeta, però aquesta vegada el situarem a una mida més gran i en el centre de la pantalla.

Per tant, inserim els fotogrames corresponents prement **F5**, i després inserirem un fotograma clau amb **F6** per fer un canvi en la mida del planeta que es mantindrà en els fotogrames 3 i 4.



## Tutorial 9. Creació d'un joc (I) Pas 17 de 26

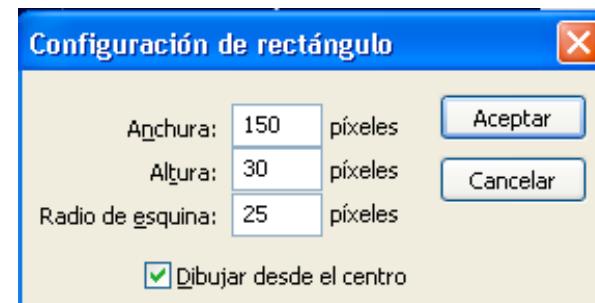


Canviem la posició i mida del planeta en el fotograma 3, i el posicionem en el centre de l'escenari i amb una mida més gran.

Creem una **nova capa** anomenada *texto*. En el fotograma 3 d'aquesta nova capa creem un **fotograma clau** (F6). Amb l'eina **Texto** creem en aquest fotograma un **camp de text estàtic** amb el text *has ganado!!!* En el fotograma 4 de la capa *texto* creem un nou fotograma clau (F6), i canviem el text a *has perdido!!!*

Creem una **nova capa** anomenada *botones*. Per crear un rectangle que després serà el fons del botó, seleccionem l'eina **Rectángulo**, sense traç i amb color de farciment vermell. Amb la tecla **Alt** premuda fem clic sobre l'escenari.

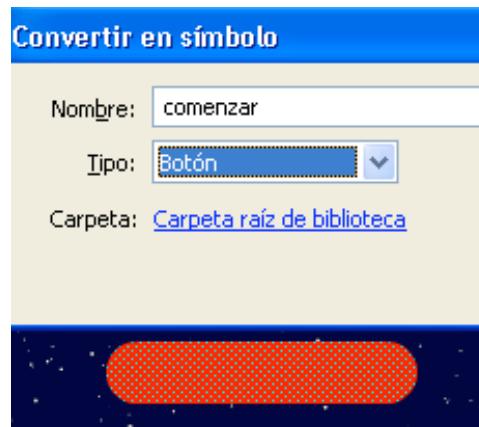
En la configuració del rectangle, establim una **amplada** de 150 px, una **altura** de 30 px, i un **radi de cantonada** de 25 px, perquè apareguin les cantonades arrodonides.



## Tutorial 9. Creació d'un joc (I)

### Pas 18 de 26

Seleccionem el rectangle vermell que hem creat, i premem **F8** per **convertir-lo en símbol**. Li donem el nom *comenzar*, i seleccionem que sigui del tipus **Botón**.



Fem doble clic sobre el botó en l'escenari per editar-lo.

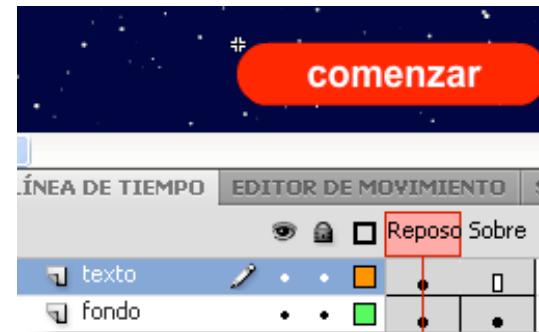
Inserim un fotograma clau prement **F6** en l'estat *Sobre*, i canviem el color del rectangle a blau en l'**inspector de Propiedades**.



Afegim una nova capa sobre el fons del botó, en la qual situarem el text del botó.

Amb l'eina **Texto** creem un camp de text estàtic, i escrivim *comenzar* en color blanc. Centrem el camp de text en el rectangle de fons.

Aquest text es mantindrà tant en el fotograma *Reposo* com en el fotograma *Sobre*.



Tornem a l'escena principal, i situem el botó en la part inferior dreta de l'escenari. En l'**inspector de Propiedades** li assignem el nom d'instància *comenzar\_btn*.

## Tutorial 9. Creació d'un joc (I) Pas 19 de 26

Creem un botó nou anomenat *instrucciones* amb les mateixes características que l'anterior, però aquesta vegada amb el text *instrucciones*. El situem en l'escenari sobre el botó *comenzar*.

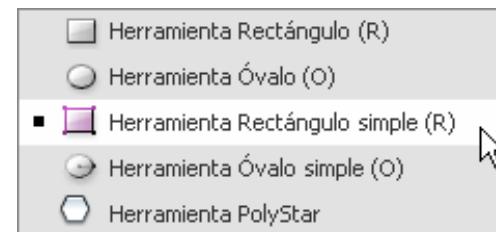


Farem que quan situem el punter sobre el botó *instrucciones* apareguin directament les instruccions del joc. La manera més senzilla d'aconseguir aquest efecte és afegint el contingut que volem mostrar al fotograma *Sobre* del botó.

Fem doble clic sobre el botó *instrucciones* per afegir alguns canvis al seu fotograma *Sobre*.

Afegim una nova capa en la línia de temps del botó, i inserim un **fotograma clau** en l'estat *Sobre*.

Seleccionem l'**eina Rectángulo simple**, per poder efectuar modificacions en les característiques del rectangle després de dibuixar-lo.



En el fotograma *Sobre* d'aquesta nova capa que hem creat, dibuixem un rectangle amb el mateix color blau que havíem utilitzat per al botó.

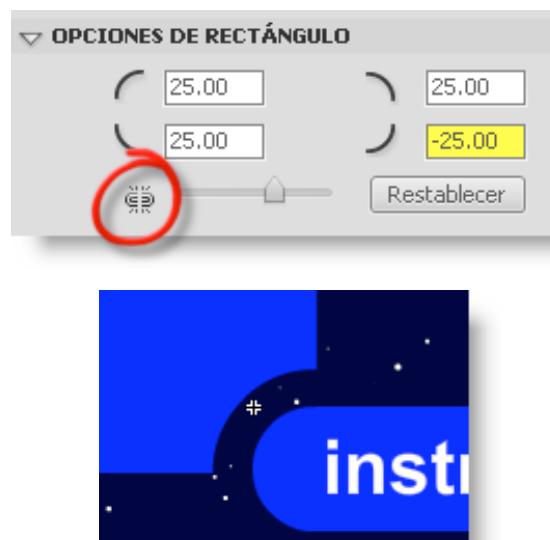


## Tutorial 9. Creació d'un joc (I) Pas 20 de 26

Amb el rectangle acabat de crear seleccionat en l'escenari, canviem el valor de les cantonades a l'**inspector de Propiedades**.

Per a poder assignar un valor diferent a una cantonada, en primer lloc hem de desactivar el bloqueig del radi de les cantonades.

Després assignem un valor de 25 a totes les cantonades excepte a la cantonada inferior dreta, a la qual assignarem un valor de -25. D'aquesta manera aconseguirem una cantonada amb el radi invers.



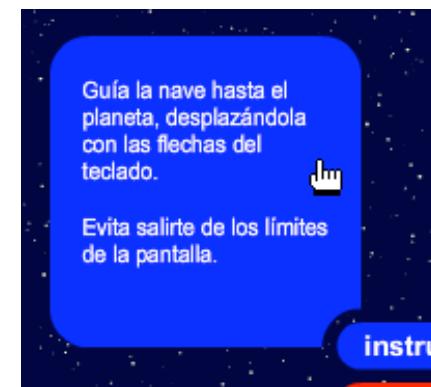
Creem una altra capa per sobre d'aquest rectangle per a escriure-hi el text de les instruccions. Inserim un fotograma clau en l'estat Sobre d'aquesta capa.

El text, en color blanc per ressaltar sobre el requadre blau, podria ser similar al següent: *Guía la nave hasta el planeta desplazándola con las flechas del teclado. Evita salirte de los límites de la pantalla.*

Tornant a l'escena principal, seleccionem **Control > Habilitar botones simples** per poder comprovar l'efecte de situar el punter sobre un botó sense necessitat de provar la pel·lícula.

Podem comprovar que si situem el punter sobre l'àrea en què es mostren les instruccions, el botó també s'activa.

Tanmateix, volem que només es mostrin les instruccions si ens situem sobre el rectangle vermell del botó *instrucciones* en estat de repòs.



## Tutorial 9. Creació d'un joc (I)

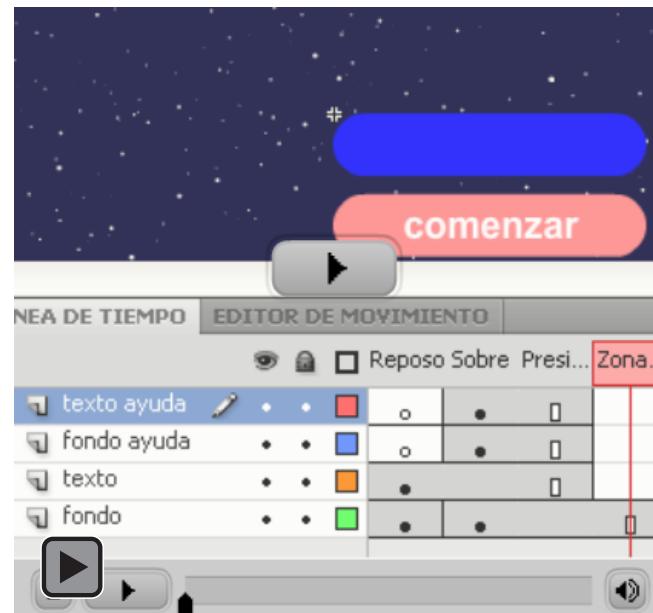
### Pas 21 de 26

Seleccionem de nou **Control > Habilitar botones simples**

per desactivar els botons, i així poder fer doble clic sobre el botó *instrucciones* per a editar-lo.

Inserim fotogrames de tal manera que en la zona activa es mostri el fons del botó, és a dir, el rectangle arrodonit que servia de base al botó.

Inserim fotogrames en la resta de les capes perquè també es mostrin en l'estat *Presionado*, i així evitar que en aquest estat només es mostri el fotograma del fons del botó.



Si ara tornem a habilitar els botons simples per fer la prova, veurem que el botó es comporta de la manera esperada.

Aquesta manera d'utilitzar l'estat *Sobre de un botón*, també anomenat *rollover*, pot resultar molt útil per a crear aplicacions educatives, ja que podem mostrar diferents tipus d'informació en situar-nos sobre una àrea concreta.

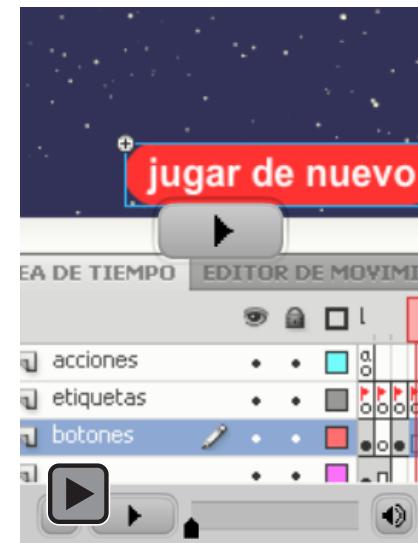
## Tutorial 9. Creació d'un joc (I) Pas 22 de 26

Tornem a l'escena principal. Els botons *instrucciones* i *comenzar* només s'han de veure en el primer fotograma, ja que durant el joc no han de ser visibles.

Creem el botó que ens falta, que anomenarem *repetir*. Serà un botó igual que *comenzar*, llevat que en el text escriurem *jugar de nuevo*. Aquest botó haurà de ser visible en els fotogrames *ganar* i *perder* (3 i 4).

Col·loquem aquest tercer botó en el tercer fotograma de la capa *botones*, en la mateixa posició que tenia el botó *comenzar* en el primer fotograma i li donem el nom d'instància *repetir\_btn*. Al botó *instrucciones* no li vam donar nom d'instància perquè no portarà associada cap programació.

Inserim un fotograma en blanc en el segon fotograma (en el qual te lloc el joc) perquè en aquest fotograma no es mostri cap botó.



Repassant el contingut que hem creat, en el primer fotograma (*inicio*) es veuran, a més de la nau i el planeta, els botons *instrucciones* i *comenzar*. En el segon fotograma (*juego*) només vearem la nau i el planeta. En el tercer fotograma (*ganar*) i en el quart fotograma (*perder*), vearem el botó *repetir*, i també una imatge ampliada del planeta en el centre de l'escenari, amb diferents textos segons es guanya o es perd. Les estrelles seran el fons comú dels quatre fotogrames.

## Tutorial 9. Creació d'un joc (I)

### Pas 23 de 26

Ara que tenim els botons creats, necessitem programar-ne el funcionament. Tornem al fotograma 1 de la capa *acciones* i premem **F9** per obrir el **tauler Acciones**.

En primer lloc, en iniciar la nostra pel·lícula volem que s'aturi al primer fotograma, motiu pel qual inclourem la instrucció *stop()*; Podem col·locar aquesta instrucció, per exemple, en la primera línia de la nostra programació.

Ara el moviment de la nau no s'ha de permetre directament, sinó que s'ha d'activar quan premem el botó *comenzar\_btn*. D'una altra banda, el botó *comenzar\_btn* també ens ha de portar al fotograma 2, etiquetat com a *juego*, en què no es mostra cap botó.

Per tant, afegim un *listener* al botó *comenzar\_btn*, de tal manera que executi una funció que anomenarem *jugar* quan hi fem clic.

```
comenzar_btn.addEventListener(MouseEvent.CLICK, jugar);
```

Les sentències que havíem creat per seleccionar (posar el focus) i afegir els *listeners* a la nau han de ser ara dins de la funció *jugar*.

La funció *jugar* quedará, per tant, de la manera següent:

```
function jugar(e:MouseEvent):void
{
    gotoAndStop("juego");
    nave_mc.focusRect = false;
    stage.focus = nave_mc;
    nave_mc.addEventListener(KeyboardEvent.KEY_DOWN,flechas);
    nave_mc.addEventListener(Event.ENTER_FRAME,navegar);
}
```

Per al botó *instrucciones* no hi ha programació, ja que la seva funció és mostrar l'ajuda quan situem el punter sobre el botó, i això ho hem aconseguit mostrant l'ajuda en l'estat *Sobre* del botó.

Ara canviem les sentències *trace* de la funció *navegar* per les sentències següents:

```
gotoAndStop("ganar"); //si llegamos al planeta
gotoAndStop("perder"); //si nos salimos del escenario
```

Si no haguéssim posat etiquetes en els fotogrames, les instruccions serien *gotoAndStop(3)* en un cas, i *gotoAndStop(4)* en l'altre cas.

## Tutorial 9. Creació d'un joc (I) Pas 24 de 26

Tot i que encara que ens falta per programar el botó per tornar a jugar, si ara provem la pel·lícula i arribem al planeta, comprovarem que després de mostrar el fotograma que indica que hem guanyat, al cap de poca estona es mostrerà el fotograma que hem perdut.

Això es deu al fet que, malgrat que no veiem la nau en els fotogrames 3 i 4, no hem eliminat els *listeners*, per la qual cosa es continuen calculant posicions per a la nau. És a dir, que després d'arribar al planeta, la posició de la nau sortirà per un extrem de l'escenari, i això farà que vegem el fotograma que indica que hem perdut.

Per a solucionar aquest problema, abans del *gotoAndStop* que envia el cap lector als fotogrames *ganar* o *perder*, hem d'eliminar els *listeners* que havíem creat per a la nau.

```
nave_mc.removeEventListener(KeyboardEvent.KEY_DOWN,flechas);  
nave_mc.removeEventListener(Event.ENTER_FRAME,navegar);
```

Després de la sentència *gotoAndStop*, que enviarà el cap lector als fotogrames del final, podem afegir el *listener* per habilitar el botó *repetir*, ja que és llavors quan el botó apareixerà en escena:

```
repetir_btn.addEventListener(MouseEvent.CLICK,repetir);
```

Per tant, els passos que es faran, tant si guanyem com si perdem, seran en primer lloc esborrar els *listeners* de la nau, després anar al fotograma *ganar* o *perder*, segons el cas, i finalment afegir un *listener* al botó *repetir*.

Llevat del nom del fotograma al qual ens dirigim, les instruccions que donem són les mateixes en tots dos casos (guanyar o perdre). Per a no repetir la programació en dos llocs diferents (en els dos condicionals *if* dins de *navegar*), el més adequat en aquests casos és crear una funció independent que contingui aquestes sentències.

Per a solucionar el fet que calgui anar en un fotograma diferent en cada cas, podem afegir un paràmetre a la funció, de tal manera que rebi el nom del fotograma on ha d'anar.

Així podríem incloure en la crida a la funció, que anomenarem *juegoAcabado*, el nom del fotograma de la manera següent (el mateix però amb *perder* en l'altre cas):

```
if (e.target.hitTestObject(planeta_mc))  
{  
    juegoAcabado ("ganar");  
}
```

## Tutorial 9. Creació d'un joc (I)

### Pas 25 de 26

La funció, que rep un paràmetre amb el nom del fotograma, tindrà aquesta definició:

```
function juegoAcabado(nombreFotograma:String):void
{
    nave_mc.removeEventListener(KeyboardEvent.KEY_DOWN,flechas);
    nave_mc.removeEventListener(Event.ENTER_FRAME,navegar);
    gotoAndStop(nombreFotograma);
    repetir_btn.addEventListener(MouseEvent.CLICK,repetir);
}
```

Como veiem, el paràmetre que rep la funció l'hem anomenat *nombreFotograma*, i a la sentència *gotoAndStop* li indiquem que vagi a un fotograma amb el valor del paràmetre rebut (en el nostre cas, els valors rebuts són "ganar" o "perder").

La funció *repetir*, que es cridarà quan premem el botó *repetir\_btn*, només indicarà que tornem al fotograma inicial:

```
function repetir(e:MouseEvent):void
{
    gotoAndStop("INICI");
}
```

Resumint la programació creada en la primera part del joc, tenim en primer lloc un *stop* i la creació de les variables *movimiento* i *velocidad*.

Després afegim un *listener* al botó *comenzar\_btn*.

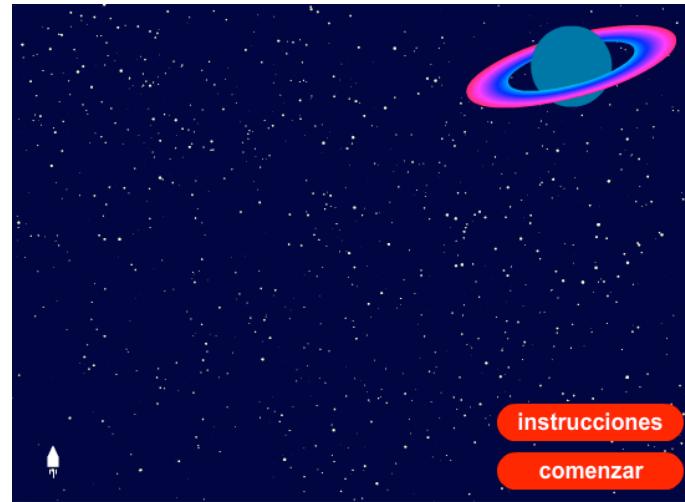
Per acabar, tenim creades cinc funcions, que es cridaran en diferents moments. Les funcions que hem creat són:

- *jugar* (activa el moviment de la nau).
- *flechas* (fa girar la nau i indica la direcció del moviment).
- *navegar* (desplaça la nau i evalua si ha arribat al planeta o ha sortit fora de l'escenari).
- *juegoAcabado* (esborra els *listeners*, va a la pantalla final i prepara el botó *repetir*).
- *repetir* (va al fotograma inicial del joc).

En el tutorial següent afegirem complexitat al joc afegint obstacles que es mouran de manera aleatòria per l'escenari.

Tutorial 9. Creació d'un joc (I)

**Pas 26 de 26**



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

1. Creeu en un nou document un dibuix d'un cos humà utilitzant botons, de tal manera que, quan situem el punter sobre cada àrea del cos, l'àrea ressalti i es mostri un quadre amb informació sobre aquesta.
2. Creeu un joc nou amb diferents nivells, en el qual puguem moure un objecte amb el teclat i en arribar a un objectiu passem a una altra pantalla. En cada nova pantalla augmentem la velocitat.

## Tutorial 10. Creació d'un joc (II)

### Pas 1 de 22

Continuem amb el joc que hem començat en el tutorial anterior. Ara en completarem el desenvolupament afegint-hi obstacles que la nau ha d'evitar.

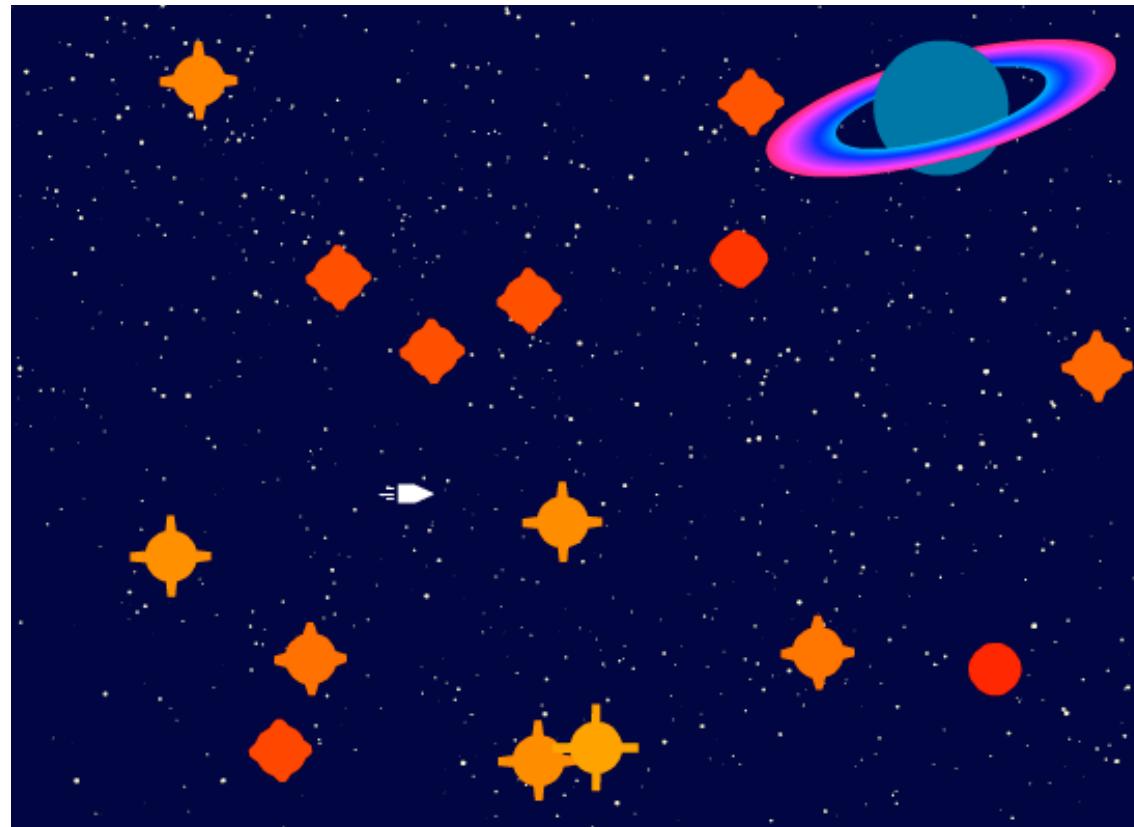
En primer lloc, per als obstacles crearem una animació utilitzant una interpolació de forma.

Aprendrem a afegir objectes dinàmicament des de la biblioteca, a més de posicionar-los a l'escenari aleatoriament.

També aprendrem a generar moviment aparentment erràtic.

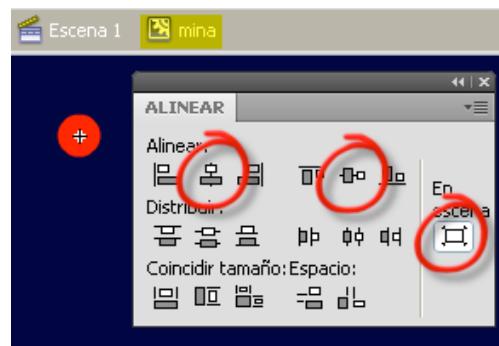
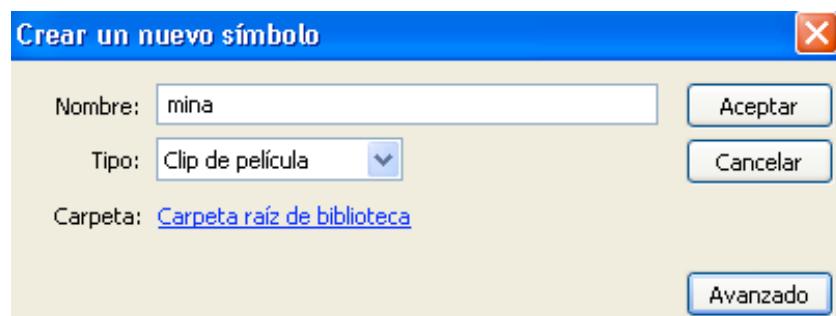
Farem servir sons de la biblioteca sense necessitat d'afegir-los a l'escenari.

Finalment, permetrem al jugador triar el nombre d'obstacles que apareixeran en el joc utilitzant un camp d'introducció de text.



## Tutorial 10. Creació d'un joc (II) Pas 2 de 22

En primer lloc, crearem els obstacles que haurà d'esquivar la nau. Seleccionem **Insertar > Nuevo símbolo**. Donem al símbol el nom *mina* i com a tipus seleccionem **Clip de película**.



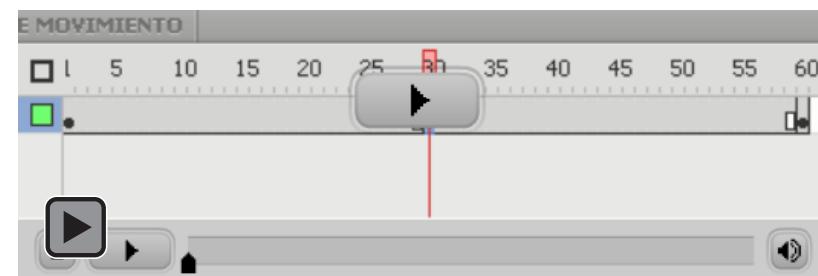
En la línia de temps d'aquest nou clip, dibuixarem un cercle vermell petit de 25 x 25 px, sense traç i amb farciment vermell.

Amb el **tauler Alinear**, centrem aquest cercle al seu escenari.

Aquest clip tindrà una animació senzilla de la qual canviarem la forma i el color, per després tornar-la a la forma original.

L'animació durarà 60 fotogrames. Inserim un **fotograma clau** amb la tecla **F6** en el fotograma 60. D'aquesta manera es copiarà el contingut del fotograma 1 en el fotograma 60.

Inserim un altre **fotograma clau** en el fotograma 30. També es copiarà el contingut del fotograma 1, però en aquest cas la intenció és modificar en aquest fotograma la forma del cercle original.



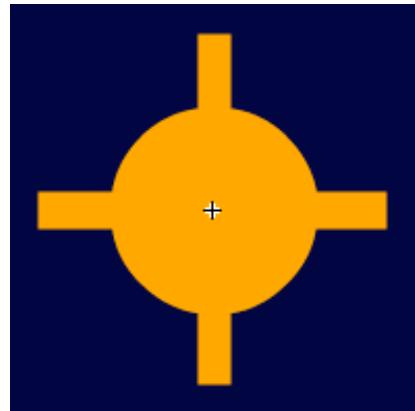
## Tutorial 10. Creació d'un joc (II)

### Pas 3 de 22

Amb el fotograma clau 30 seleccionat, dibuixem dos rectangles, un de vertical i un altre d'horitzontal, que centrarem a mesura que els dibuixem. Podem treballar amb zoom per a més comoditat.

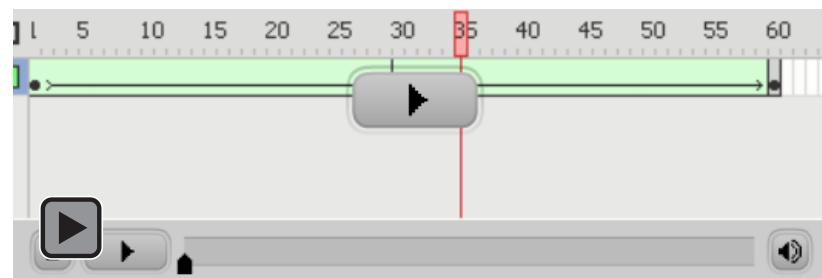
Aquests rectangles tampoc tindran traç i tindran com a farciment el mateix color vermell que el cercle. D'aquesta manera, els rectangles quedarán units al cercle en una sola forma.

Per acabar, canviem el color de la forma composta a un to ataronjat.



Per a crear una transició des de la forma del cercle original al cercle amb els rectangles, i una altra transició fins a tornar a la forma i color originals, hem de crear interpolacions de forma.

Per a això fem clic amb el **botó dret** del ratolí sobre l'àrea grisa que hi ha entre dos fotogrames clau, i seleccionem el menú contextual **Crear interpolación de forma**. Repetim el procés per a la transició compresa entre els altres dos fotogrames clau.

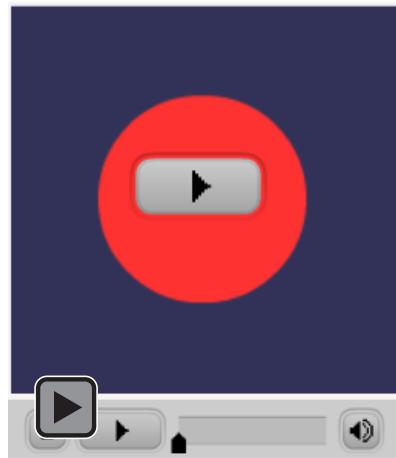


Podem crear interpolacions de forma entre dues formes qualssevol. Per a controlar canvis de forma complexos, podem utilitzar els consells de forma, encara que en el nostre cas no caldrà. Consulteu l'ajuda del Flash referent a això si heu de crear interpolacions de forma complexes.

## Tutorial 10. Creació d'un joc (II)

### Pas 4 de 22

Si premem **Intro** o movem el cap lector, veurem que la forma canvia del cercle vermell inicial, a la forma composta ataronjada, per a tornar posteriorment al cercle vermell.



Podeu experimentar amb diferents formes o colors fins a crear una interpolació de la manera que us agradi més.

Amb això donem per finalitzada la creació de la mina, per la qual cosa tornem a l'escena principal.



A continuació afegirem còpies a l'escenari del clip *mina* utilitzant ActionScript.

Fins ara, per a poder fer referència a un objecte utilitzàvem en la programació el seu nom d'instància. Per a assignar un nom d'instància a un clip, seleccionàvem el clip en l'escenari i escrivíem el seu nom d'instància en l'inspector de *Propiedades*.

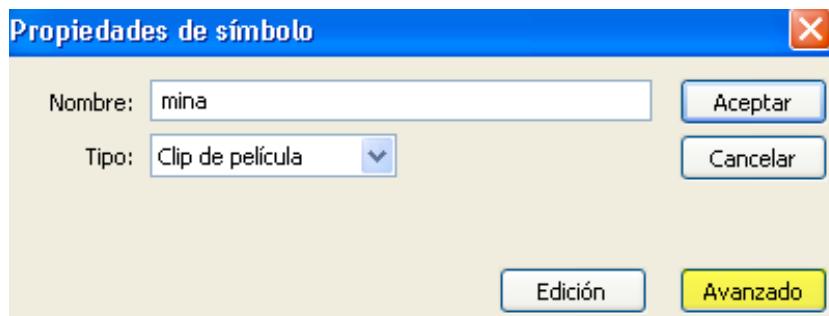
En cas que l'objecte no sigui en l'escenari, l'hem d'exportar per a ActionScript des de la biblioteca.

Fem clic amb el **botó dret** del ratolí sobre el nom del clip (*mina* en aquest cas), i seleccionem **Propiedades** en el menú contextual.

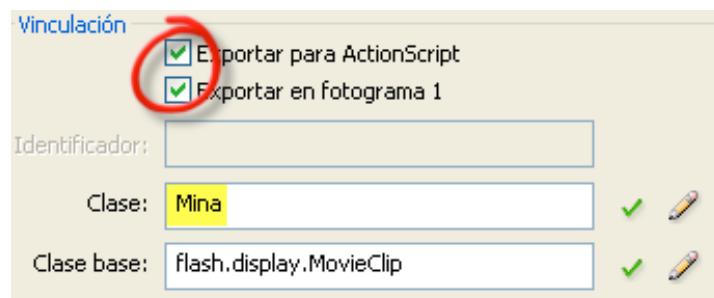


## Tutorial 10. Creació d'un joc (II) Pas 5 de 22

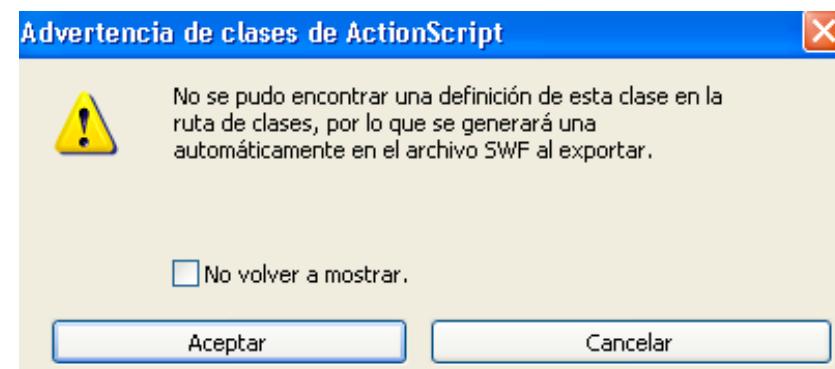
En la finestra **Propiedades de símbolo**, premem el botó **Avanzado**.



En l'àrea **Vinculación** marquem les caselles **Exportar para ActionScript** i **Exportar en fotograma 1**. Escrivim el nom de classe **Mina** (els noms de les classes comencen amb majúscula).



Quan premem **Aceptar** ens apareixerà una finestra d'avertiment. Tornem a prémer **Aceptar** perquè es generi de manera automàtica la definició de la classe *Mina*.



Després d'haver exportat el clip en una classe anomenada *Mina*, ja podem fer referència a aquesta classe en la programació.

## Tutorial 10. Creació d'un joc (II)

### Pas 6 de 22

Definirem una funció nova que anomenarem *crearMinas*.

```
function crearMinas():void
{
    var mina:Mina;
    mina = new Mina();
    mina.x = 275;
    mina.y = 200;
    stage.addChild(mina);
}
```

En les primeres línies d'aquesta funció, creem un objecte nou de nom *mina* que pertany a la classe *Mina*, i que serà una còpia nova del clip que tenim en la biblioteca. Després d'aquesta definició, ja podem fer referència al nou objecte creat.

En les línies següents, definim que la posició de l'objecte sigui el centre de l'escenari (x:275 i y:200). Finalment, afegim aquest objecte a l'escenari (*stage*) amb el mètode *addChild*.

Perquè aquesta funció s'executi, en algun moment l'hem de cridar amb la instrucció *crearMinas () ;*. Escriurem aquesta instrucció dins de la funció *jugar*, ja que serà el moment en què volem que s'afegeixin les mines a l'escenari.

Si provem la pel·lícula, comprovarem que en prémer el botó *comenzar* (que crida la funció *jugar*, que al seu torn crida la funció *crearMinas*) apareix una mina en el centre de l'escenari.

Perquè la posició en la qual apareix la mina sigui aleatòria, utilitzarem el mètode *random* de la classe *Math*. *Math.random()* torna un nombre aleatori comprès entre 0 i 1. Si multipliquem aquest valor aleatori per 550 (l'amplada de l'escenari), el nombre que tornarà estarà comprès entre 0 i 550.

Per tant, per a crear una posició aleatòria de la mina dins de l'escenari, que mesura 550 x 400, les línies que determinen la posició del clip dins de la funció *crearMinas* quedaràn com s'indica a continuació:

```
mina.x = Math.random() * 550;
mina.y = Math.random() * 400;
```

La posició *x* del clip serà un nombre comprès entre 0 i 550, mentre que la posició *y* serà un nombre comprès entre 0 i 400. Cada vegada que s'executa la funció *Math.random()*, el nombre tornat serà diferent.

## Tutorial 10. Creació d'un joc (II) Pas 7 de 22

Per ara només s'afegeix una còpia del clip a l'escenari. Per a afegir diverses còpies, hem de fer un bucle que creï diverses mines i les afegeixi a l'escenari.

Juntament amb les variables *movimiento* i *velocidad*, definirem una nova variable anomenada *numMinas*, el valor del qual serà el nombre de mines que volem crear (per exemple, 10).

```
var numMinas:Number = 10;
```

Ara farem servir un bucle *for* per a crear aquestes 10 còpies de la mina.

Els bucles *for* tenen l'estruatura següent:

```
for (valor inicial; condición; cambio en el valor)
{
    //sentencias a ejecutar
}
```

Per exemple, en el nostre cas:

```
for (var i:Number = 0; i < numMinas; i++)
{
    //sentencias a ejecutar
}
```

Aquest bucle funcionaria de la manera següent:

- Creem una variable anomenada *i* amb un valor inicial de 0.
- Comprovem si es compleix la condició, que en aquest cas és que el valor d'*i* sigui més petit que el valor de *numMinas*.
- En complir-se la condició que *i* < *numMinas*, executarem les sentències que hi hagi entre les claus del bucle *for*.
- Augmentem el valor d'*i* en una unitat (*i*+ significa *i* = *i* + 1).
- Tornem a comprovar la condició. Ara *i* val 1, que continua essent més petit que 10 (valor de *numMinas*).
- Atès que la condició es continua complint, tornem a executar les sentències, i sumem una altra unitat a *i*, que ara valdrà 2.
- Quan *i* valgui 10, moment en què no es complirà que *i* sigui més petit que *numMinas*, ja no s'executarà el bucle. En començar amb un valor de *i*=0, el bucle s'haurà executat un total de 10 vegades.

## Tutorial 10. Creació d'un joc (II)

### Pas 8 de 22

Per tant, per crear la quantitat de mines que hàgim indicat en la variable *numMinas*, la funció *crearMinas* quedarà com s'indica a continuació:

```
function crearMinas():void
{
    var mina:Mina;
    for (var i:Number = 0; i < numMinas; i++)
    {
        mina = new Mina ;
        mina.x = Math.random() * 550;
        mina.y = Math.random() * 400;
        stage.addChild(mina);
    }
}
```

Si ara provem la pel·lícula, podem comprovar que apareixen 10 mines en l'escenari.

Totes les mines fan la seva animació alhora. Podem fer que cada mina comenci l'animació en un fotograma diferent, i que porti el seu cap lector a un fotograma aleatori entre l'1 i el 60, que és el nombre de fotrames que té l'animació.

Per tant, aquesta vegada necessitem generar un nombre aleatori entre 1 i 60. En aquest cas el nombre tornat ha de ser sencer.

Si utilitzem *Math.random()* \* 60 obtindrem nombres decimals entre 0 i 60. Per a assegurar-nos que el nombre resultant sigui un enter comprès entre 1 i 60 podem utilitzar el mètode *ceil*, que arrodoneix a l'alça un nombre decimal.

Per tant, dins del bucle *for*, després d'haver determinat una posició aleatòria per a cada mina i abans d'afegir la mina a l'escenari, escriurem la instrucció:

```
mina.gotoAndPlay(Math.ceil(Math.random() * 60));
```

Tornem a provar la pel·lícula. Ara, en començar el joc, es creen 10 còpies de la mina, i cada una comença l'animació en moments diferents.

## Tutorial 10. Creació d'un joc (II) Pas 9 de 22

Per convertir les mines en obstacles contra els quals hem d'evitar xocar, afegirem un detector a cada mina, perquè avaluï en cada moment si està xocant amb la nau.

Per això afegim en primer lloc un *listener* per a cada mina en el mateix bucle que utilitzem per a crear-les, dins, per tant, de la funció *crearMinas* i dins del bucle *for*, i abans d'afegir les mines a l'escenari amb *addChild*.

```
mina.addEventListener(Event.ENTER_FRAME, enemigos);
```

La nova funció, que hem anomenat *enemigos*, comprovarà si cada mina a la qual hem afegit el *listener* xoca amb la nau en algun moment. Si hi xoca, llavors executarem la funció *juegoAcabado*, amb el paràmetre "perder".

```
function enemigos(e:Event):void
{
    if (e.target.hitTestObject(nave_mc))
    {
        juegoAcabado("perder");
    }
}
```

Si ara provem el nostre joc, tant si guanyem com si perdem, ens apareixerà un error perquè, malgrat que no hi ha cap nau, la funció *enemigos* continuarà comprovant si cada mina xoca amb la nau. En no trobar cap nau en l'escenari, es mostrarà l'error.

Per tant, el primer que hem de fer en la funció *juegoAcabado*, abans de l'ordre d'anar a un altre fotograma en el qual no hi hagi la nau, és eliminar els *listeners* que hem afegit a les mines. Alhora que eliminem cada *listener*, aprofitarem per eliminar les mines de l'escenari. En el pas següent mostrarem com fer-ho.

L'escenari (stage) funciona com un contenidor que conté en primer lloc la línia de temps, i que també conté cada mina que li hem anat afegint amb *addChild*. Després de crear totes les mines, l'*stage* tindrà 11 elements secundaris. El primer, en el nivell inferior de visualització (nivell 0), serà la línia de temps. Després hi haurà totes les mines, cada una en un nivell superior a l'anterior.

La quantitat de mines creades depèn del valor que hagim donat a *numMinas*. Per tant, l'última mina creada es mostrarà en un nivell de visualització que coincideix amb *numMinas*.

## Tutorial 10. Creació d'un joc (II)

### Pas 10 de 22

Al començament de la funció *juegoAcabado*, abans de les sentències en les quals eliminàvem els *listeners* de la nau, podem eliminar el *listener* de cada mina, i després eliminar cada mina, amb el codi següent:

```
for (var i:Number = numMinas; i > 0; i--)  
{  
    stage.getChildAt(i).removeEventListener(Event.ENTER_FRAME,enemigos);  
    stage.removeChildAt(i);  
}
```

Analitzem el que fa aquest bucle. El bucle començarà amb un valor d'*i* igual a *numMinas*, que en el nostre cas és 10. Com que la variable *i* és més gran que 0, s'executaran les sentències que després explicarem. Després reduirem en una unitat el valor de *i* (*i--* significa *i = i - 1*). Per tant, *i* valdrà 10 en la primera execució del bucle, 9 en la volta següent, etc., fins a l'últim valor que compleix la condició, és a dir, fins que *i* valgui 1 (compleix *i > 0*).

En les sentències que hi ha dins del bucle, en primer lloc eliminem el *listener* de l'element que hi ha dins de l'*stage*, en el nivell de visualització *i* (10, 9, 8, 7..., 1), que en el nostre és cada mina que hem creat. D'aquesta manera ja no tractaran de detectar possibles col·lisions amb la nau. Amb la línia següent eliminem la mina a la qual acabem d'eliminar el seu *listener*.

Ara el nostre joc serà funcional i no tindrà errors, però introduirem millors en els passos següents:

- En primer lloc, ens assegurarem que la posició inicial de les mines no sigui gaire a prop de la nau, perquè ens doni temps a començar a jugar.
- Per fer el joc més complex, afegirem un moviment aleatori de les mines per l'escenari.
- Canviarem la imatge del planeta per la imatge d'una mina en el fotograma *perder*.
- Després afegirem un so per a quan guanyem i un altre per a quan perdem.
- Finalment, permetrem a l'usuari decidir el nombre de mines que vol que apareguin en el joc.

## Tutorial 10. Creació d'un joc (II) Pas 11 de 22

Comencem per la primera millora.

A causa de l'aleatorietat de la posició de les mines, és possible que fins i tot abans de moure la nau ja hi hagi alguna mina que hi xoqui.

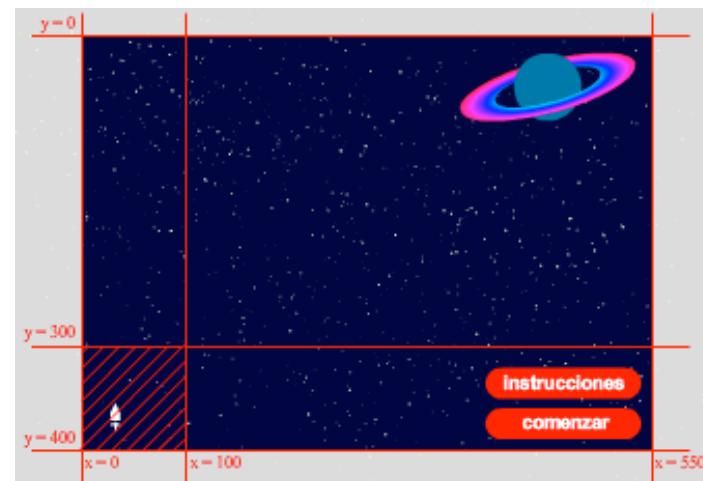
Per solucionar aquest problema, substituirem el codi en el qual creàvem la posició x i y de cada mina, dins de la funció *crearMinas*, per aquest codi:

```
do
{
    mina.x = Math.random() * 550;
    mina.y = Math.random() * 400;
}
while (mina.x < 100 && mina.y > 300);
```

La sentència *do...while* el que fa és executar en primer lloc el que hi hagi entre claus. Mentre es compleixi la condició que hi ha entre els parèntesis de *while*, es tornarà a executar el que hi hagi entre les claus del *do*.

En aquest cas, creem una posició a l'atzar per a la mina. Si la posició és a prop de la cantonada inferior esquerra (si la posició horitzontal de la mina és més petita que 100, i a més la seva posició vertical és més gran que 300), llavors tornarà a generar-se una posició aleatòria per a la mina. Aquest procés es repetirà fins que la condició no es compleixi, cosa que significarà que la posició generada per a la mina ja no és a prop de la nau.

D'una manera més gràfica, si la posició aleatòria de la mina és en la zona del quadrat ratllat, llavors canviarem la posició de la mina de manera aleatòria, fins a trobar una posició vàlida.



## Tutorial 10. Creació d'un joc (II)

### Pas 12 de 22

Afegirem el moviment de vibració de les mines en l'escenari.

Creem una variable nova a l'inici de la programació, juntament amb les variables *movimiento*, *velocidad* i *numMinas*, que anomenarem *vibracion*. Hi assignem un valor de 5.

```
var vibracion:Number = 5;
```

Programarem que la posició de la mina varii, des de la posició en la qual es troba en cada moment, a una nova posició aleatòria que sigui a una distància màxima de 5 píxels de la posició actual, que és la quantitat que hem assignat a la variable *vibracion*.

És a dir, si la posició actual d'una mina és  $x=200$  i  $y=300$ , a l'instant següent el valor d' $x$  estaria entre 195 i 205, mentre que el valor d' $y$  podria estar entre 295 i 305. D'aquesta manera, la mina vibrarà un màxim de 5 píxels en cada sentit (horitzontal i vertical).

Tal com hem après anteriorment, la sentència

```
Math.random() * vibracion
```

tornarà un valor entre 0 i 5 (que és el valor que hem donat a la variable *vibracion*).

Per tant, la sentència

```
Math.random() * vibracion - Math.random() * vibracion
```

tornarà un nombre entre -5 (0-5) i 5 (5-0).

Dins de la funció *enemigos* (que s'executa cridada per un *enter frame* de cada mina), però fora del condicional que avalua si hi ha xoc entre la mina i la nau, escriurem el codi següent:

```
e.target.x+=Math.random()*vibracion-
Math.random()*vibracion;
e.target.y+=Math.random()*vibracion-
Math.random()*vibracion;
```

Per evitar que amb aquesta vibració aleatòria la mina pugui acabar fora de l'escenari, dins de la mateixa funció afegirem unes sentències condicionals que avaluïn si la mina és fora dels límits, i si hi és la tornarem a col·locar en el límit de l'escenari.

```
if (e.target.x < 0)
{
    e.target.x = 0;
}
```

Hem d'afegir-hi altres condicionals similars (si  $x>550$  llavors que  $x=550$ , si  $y<0$  llavors  $y=0$ , i si  $y>400$ , llavors  $y=400$ ).

## Tutorial 10. Creació d'un joc (II)

### Pas 13 de 22

Provem la pel·lícula per comprovar si s'efectua la vibració de les mines. Veurem que encara que les mines mai no desapareixen completament de l'escenari, pot ser que en alguns moments vegem mitja forma de la mina fora de l'escenari, perquè el punt de registre de la mina és en el seu centre.

Si volem, podem modificar els límits de moviment de la mina canviant els valors dels condicionals del pas anterior. Per exemple, podem reduir l'espai en el qual poden moure's les mines en 25 píxels en cada banda.

La millora següent va a ser eliminar el planeta en el quart fotograma (fotograma *perder*), i mostrar com a substitució una mina en el centre de l'escenari.

En primer lloc, afegim una nova capa que anomenarem *mina*. Situem aquesta capa per sobre de la capa *planeta*.

Eliminem el quart fotograma de la capa *planeta* prement-lo amb el **botó dret** del ratolí, i seleccionant **Quitar fotogramas**.

Després premem amb el **botó dret** del ratolí el quart fotograma de la capa *mina*, i seleccionem **Insertar fotograma clave**.



En aquest nou fotograma clau, arrosseguem una instància del clip *mina* des de la biblioteca fins al centre de l'escenari, i n'augmentem la mida fins, per exemple, *W*: 100 i *H*:100.



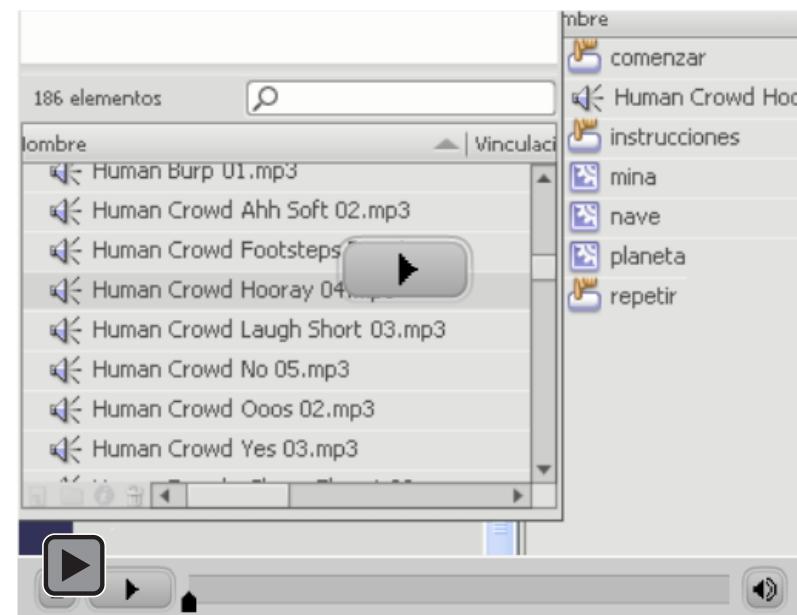
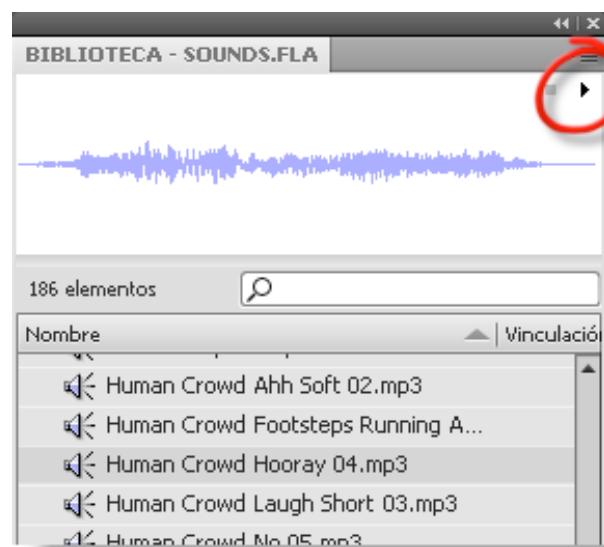
## Tutorial 10. Creació d'un joc (II) Pas 14 de 22

Buscarem sons que associarem al moment de guanyar o de perdre la partida.

A **Ventana > Bibliotecas comunes > Sonidos** podem trobar alguns exemples de sons que podem utilitzar en els nostres projectes.

Per escoltar els diferents sons, seleccionem el so en la biblioteca i premem el botó petit per reproduir el que hi ha en la finestra on es mostra l'ona del so.

Utilitzarem els sons *Human Crowd Hooray 04.mp3* per al moment de guanyar, i *Multimèdia Internet CD-Rom Flash Blast 06.mp3* per al de perdre. Si voleu podeu utilitzar altres sons. Podem ampliar la mida de la biblioteca de sons per a llegir els noms dels sons amb més comoditat. Per utilitzar aquests sons, els arrossegarem des de la biblioteca de sons fins a la biblioteca del nostre joc. Una vegada incorporats en la nostra biblioteca, podem tancar la biblioteca de sons.



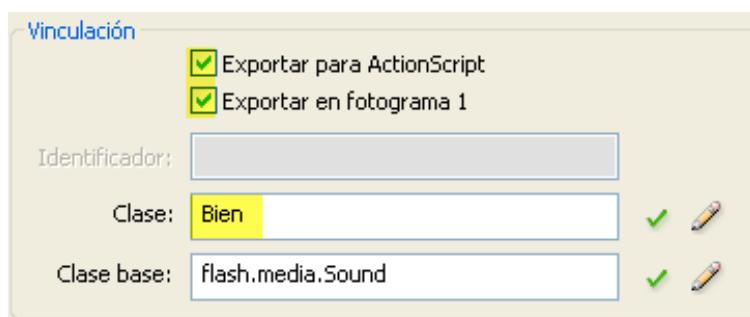
## Tutorial 10. Creació d'un joc (II) Pas 15 de 22

Igual com vam fer amb el clip de la mina, exportarem tots dos sons per a l'ActionScript.

Per fer-ho primer premem el nom del so *Human Crowd Hooray 04.mp3* en la biblioteca amb el **botó dret** del ratolí i seleccionem **Propiedades**.

En l'àrea **Vinculación** de la finestra de **Propiedades de sonido** marquem **Exportar para ActionScript** i **Exportar en fotograma 1**. Si no veiem aquesta àrea, premem el botó **Avanzado**.

Com a nom de classe escrivim simplement *Bien*. Acceptem la pantalla d'avertiment final. Repetim el procés per a l'altre so, i com a nom de classe escrivim *Mal*.



En la biblioteca podrem veure els noms de les classes que hem donat als nostres sons, i també el nom que vam donar al clip *mina*.

Per utilitzar aquests sons afegirem, juntament amb la definició d'altres variables a l'inici de la programació, les línies següents:

```
var sonidoGanar:Bien = new Bien();  
var sonidoPerder:Mal = new Mal();
```

| Nombre                    | Vinculación    |
|---------------------------|----------------|
| comenzar                  |                |
| Human Crowd Hooray 04.mp3 | Exportar: Bien |
| instrucciones             |                |
| mina                      | Exportar: Mina |
| Multimedia Internet ...   | Exportar: Mal  |
| nave                      |                |
| planeta                   |                |
| repetir                   |                |

La variable *sonidoGanar* pertanyerà a la classe *Bien*, i serà una instància nova d'aquesta classe, la classe base de la qual és *Sound*, com hem pogut veure en la finestra de vinculació. De la mateixa manera, *sonidoPerder* serà un objecte de la classe *Mal*.

## Tutorial 10. Creació d'un joc (II)

### Pas 16 de 22

Dins de la funció *juegoAcabado* reproduirem un so o un altre dependent de si hem guanyat o hem perdut.

En aquesta funció, la variable que ens indica si hem guanyat o perdut és *nombreFotograma*. Per tant, el que farem és comprovar si *nombreFotograma* té el valor "ganar". Si el té, reproduirem *sonidoGanar*. Si no el té, reproduirem *sonidoPerder*.

Per tant, dins de la funció *juegoAcabado* incloureml aquest codi:

```
if (nombreFotograma == "ganar")  
{  
    sonidoGanar.play();  
} else {  
    sonidoPerder.play();  
}
```

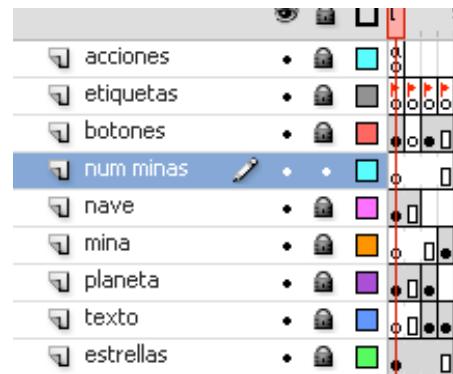
El signe d"igual doble (==) compara si *nombreFotograma* és "ganar". Aquest signe és per a comparar, mentre que un sol signe d"igual (=) és per a assignar.

Si *nombreFotograma* és "ganar", es reproduirà *sonidoGanar*. Si no ho és (else), es reproduirà *sonidoPerder*.

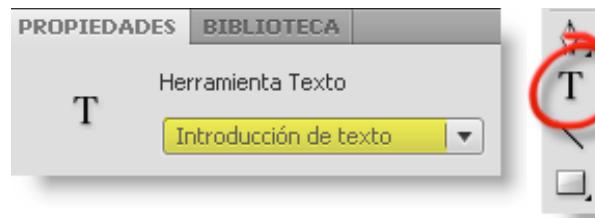
Ja tenim el joc pràcticament acabat. Podem augmentar els valors a les variables *velocidad*, *numMinas* i *vibracion* per a augmentar la dificultat del joc. Per exemple, una vibració de 15 complicaria considerablement el joc, ja que el moviment de les mines seria menys predictable.

Podem fer que el mateix jugador modifiqui aquests valors amb un camp d'introducció de text. Com a exemple, permetrem que el jugador decideixi el nombre de mines amb què vol jugar.

Per fer-ho, en primer lloc creem una nova capa anomenada *num minas* i la situem sota la capa *botones*.



## Tutorial 10. Creació d'un joc (II) Pas 17 de 22



Seleccionem l'eina **Texto**, i en l'inspector de **Propiedades** seleccionem **Introducción de texto**.

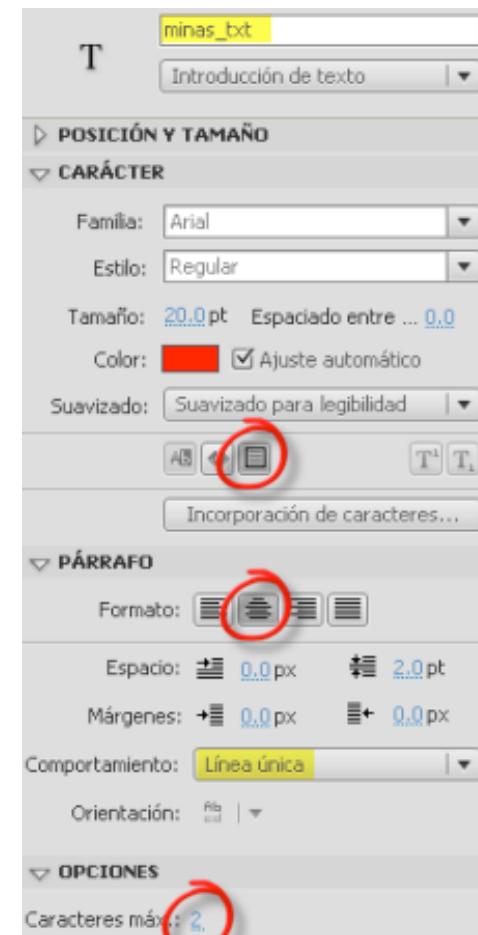
Fem clic sobre qualsevol punt en l'escenari per crear un camp d'introducció de text en la capa *num minas*.

Seleccionem aquest camp de text acabat de crear. En l'**inspector de Propiedades**, li assignem el nom *minas\_txt*.

En l'àrea **Carácter** seleccionem el tipus de lletra i la mida que vulguem, el color **vermell** a fi que ressalti sobre el fons, i marquem la casella **Mostrar borde alrededor del texto**. Això farà que aparegui un requadre blanc amb un marc negre. El marc negre amb prou feines el veurem, ja que el nostre joc té un fons molt fosc, però el color blanc del requadre ens mostrerà clarament on hi ha el nostre camp de text.

En l'àrea **Párrafo**, alineem el text en el centre i triem un **comportamiento de Línea única**.

Finalment, en l'àrea **Opciones** seleccionem que es pugui escriure un màxim de 2 caràcters.



## Tutorial 10. Creació d'un joc (II)

### Pas 18 de 22

Perquè el bloc de text no ens quedi massa gran, podem escriure-hi dos nombres de prova, i ajustar llavors la mida del camp.

Una vegada adaptada la mida, esborrem els nombres que havíem escrit com a guia i centrem el camp de text en l'escenari.

En la mateixa capa, creem un camp de **text estàtic** sota el camp d'introducció de text, i escrivim el text *número de minas en color blanc*.



En la mateixa capa també podem incloure una instància de la mina, perquè es mostri sobre el camp de text. Per fer-ho, l'arrosseguem directament des de la biblioteca.



Perquè aquests elements només es mostrin en el primer fotograma, seleccionem la resta de fotogrames d'aquesta capa, fem clic amb el **botó dret** del ratolí, i seleccionem **Quitar fotogramas** del menú contextual.



## Tutorial 10. Creació d'un joc (II)

### Pas 19 de 22

Tornant a la programació, en primer lloc podem indicar que, abans de premer cap botó, el focus de l'escenari hauria d'estar en el camp d'introducció de text, que havíem anomenat *minas\_txt*.

D'aquesta manera podrem escriure directament en el camp de text, sense necessitat de clicar-hi prèviament.

Per fer-ho, afegirem, al costat de la definició inicial de les variables del nostre joc, la instrucció següent:

```
stage.focus = minas_txt;
```

Després, quan premem el botó *comenzar*, el focus passarà a la nau, ja que és el que havíem indicat en la funció *jugar*. És per això que no hi ha cap problema a establir el focus inicial del joc en el camp de text.

Si ara provem el joc comprovarem que, si escrivim un nombre amb el teclat, aquest apareixerà en el camp de text sense necessitat d'haver-lo seleccionat prèviament.

Canviem el valor inicial de la variable *numMinas* a 0 en comptes de 10:

```
var numMinas:Number = 0;
```

Una vegada que es premi el botó *comenzar*, el primer que s'ha de fer és assignar a la variable *numMinas* el valor que hagim escrit en el camp *minas\_txt*.

Per tant, la primera instrucció dins de la funció *jugar*, serà:

```
numMinas = Number(minas_txt.text);
```

Per a accedir al contingut d'un camp de text, hem d'accedir a la propietat *text* del camp. El contingut d'un camp de text sempre és del tipus *String*. Tanmateix, la variable *numMinas* és de tipus numèric. Per això, és necessari convertir el contingut textual del camp de text *minas\_txt* en un nombre utilitzant *Number*.

Tornem a provar el nostre joc. El nombre de mines es correspondrà al que indiquem en el camp *minas\_txt*. Tanmateix, el joc també començarà si deixem el camp en blanc o si hi introduïm lletres. El pas següent serà controlar aquestes possibles circumstàncies.

## Tutorial 10. Creació d'un joc (II)

### Pas 20 de 22

En la funció *jugar* podem indicar que, una vegada que tinguem el *numMinas* a partir del text introduït en el camp de text, s'avaluï el nombre de mines. Si s'introduceix un nombre de mines entre 1 (que hi hagi almenys una mina) i 50 (nombre amb el qual és molt difícil arribar al planeta), llavors que s'executin totes les altres instruccions de la funció *jugar*.

Per tant, en la funció *jugarprimero* primer hi haurà la línia

```
numMinas = Number(minas_txt.text)
```

Després hi haurà un condicional que avaluarà si *numMinas* és dins del marge que considerem vàlid, i, si és així, s'executarà la resta de la funció i el joc començarà.

```
if (numMinas > 0 && numMinas < 51)
{
    //posem aquí la resta de les sentències que teníem
    //en la funció jugar, i el joc començarà
} else
{
    //què fem si numMinas és fora dels límits?
    //farem que aparegui un missatge d'error
}
```

Per al missatge d'error, podem dibuixar un rectangle vermell similar al dels botons, i escriure-hi un text d'avís (introduceix un nombre de mines comprès entre 1 i 50).



Seleccionem el rectangle i el text d'avís, i premem **F8** per convertir-lo en un **clip de pel·lícula** anomenat *aviso*. Li donem el nom d'instància *aviso\_mc*.

A l'inici de la pel·lícula, aquest clip ha de ser invisible, per la qual cosa, juntament amb la definició de variables a l'inici de la programació, escriurem

```
aviso_mc.visible = false;
```

Si *numMinas* no està dins dels límits, volem que es mostri l'avís. A més, podem afegir instruccions per a esborrar el nombre erroni que hem introduït, i també per a tornar el focus al camp de text, ja que s'haurà percutit en prémer el botó *comenzar*.

Per a tot això escriurem entre les claus de l'*else*

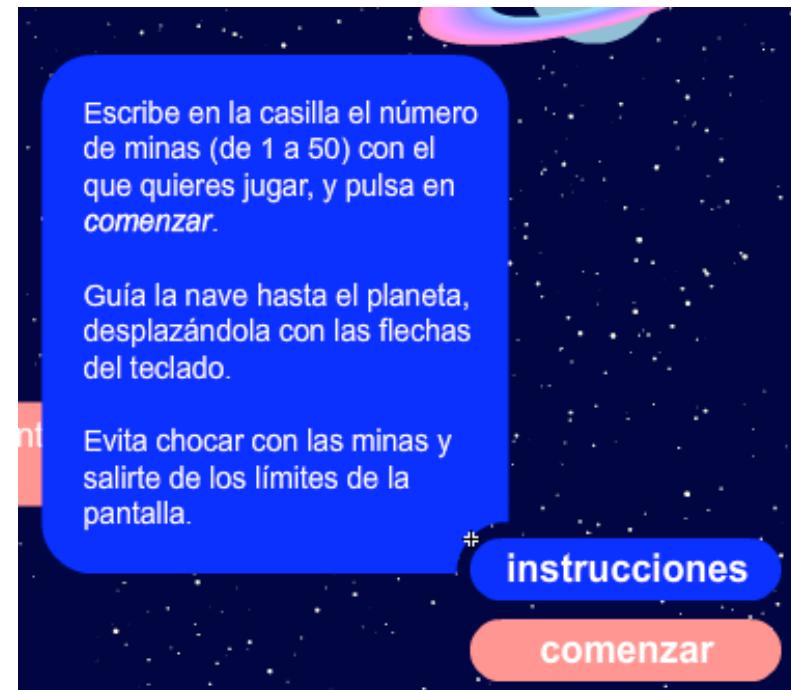
```
aviso_mc.visible = true;
minas_txt.text = "";
stage.focus = minas_txt;
```

## Tutorial 10. Creació d'un joc (II) Pas 21 de 22

En cas que *numMinas* estigui dins del marge vàlid, la pel·lícula es desenvoluparà en el fotograma *juego* (fotograma 2), en què ja no hi ha l'avís; per això en aquest cas no és necessari indicar que *numMinas* no està visible.

Per acabar, ens queda modificar la informació que mostra el botó *instrucciones* perquè també hi aparegui informació sobre les mines. Possiblement haurem d'augmentar la mida del rectangle sobre el qual apareixen les instruccions.

Amb això ja haurem completat la programació del nostre joc, amb el qual hem après a programar moltes tasques habituals en els jocs, com ara controlar objectes amb el teclat, detectar col·lisions, afegir elements des de la biblioteca, generar atzar, etc.



Tutorial 10. Creació d'un joc (II)

**Pas 22 de 22**



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

1. Permeteu al jugador seleccionar la velocitat de la nau i la vibració de les mines.
2. Creeu nivells per al joc, de tal manera que en arribar al planeta es comenci un joc nou amb més mines, més velocitat i més vibració.
3. Afegiu un nou control per al moviment de la nau, i feu que la barra espaiadora n'aturi el moviment.

## Tutorial 11. Aplicacions dinàmiques Pas 1 de 23

En aquest tutorial farem una petita aplicació que mostrerà imatges, textos i vídeos externs.

Aquest tipus d'aplicació es pot actualitzar fàcilment, ja que el contingut de la pel·lícula canvia quan es canvien els arxius externs. Per tant, és una bona opció utilitzar càrregues de dades externes en aplicacions que s'hagin d'actualitzar amb freqüència.

També aprendrem a incloure enllaços en les nostres pel·lícules.

Per començar, obrim un nou arxiu del Flash (AS 3.0). Canviem les dimensions de l'escenari a 500 x 425 px. Desem el document com a *tutorial11.fla*.

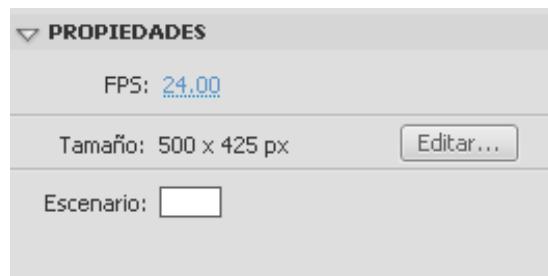


imagen  
vídeo  
enlaces



### Bar y habitación 3D

Imágenes realizadas con la herramienta 3D Studio.

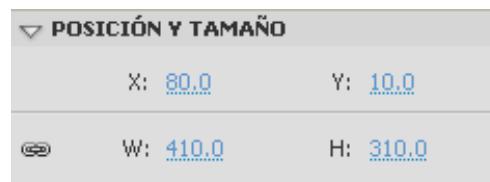
## Tutorial 11. Aplicacions dinàmiques Pas 2 de 23

En aquesta pel·lícula carregarem imatges amb una mida de 400 x 300 px.

Perquè les imatges tinguin un marc negre petit al voltant, dibuixarem un rectangle negre lleugerament més gran que aquesta mida, i el deixarem com a capa de fons.

Seleccionem l'eina **Rectángulo**, i dibuixem un rectangle negre de 410 x 310. Com ja hem vist en el tutorial 9, recordem que, si sabem amb exactitud la mida d'una forma abans de dibuixar-la, podem premer la tecla **Alt** i fer clic en l'escenari per a introduir manualment la mesura de la figura.

Situem aquest rectangle en la posició X:80 i Y:10.



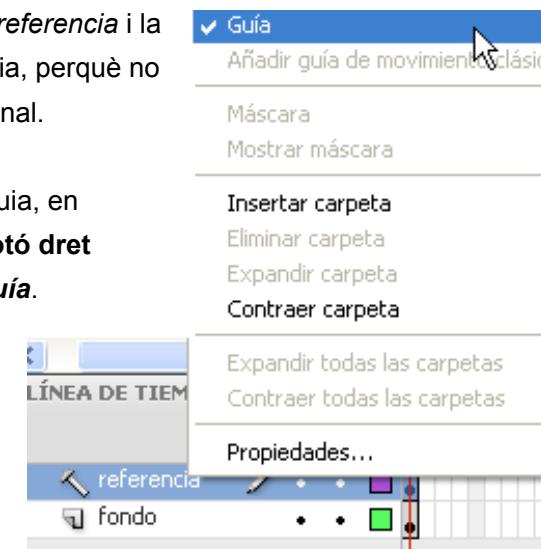
Anomenem *fondo* aquesta capa, i la mantindrem sempre per sota de totes les altres.

Creem una nova capa que ens serveixi de guia per a la posició on aniran les imatges i els vídeos. Dibuixem un **rectangle** vermell amb la mida que tindran les imatges, això és, 400 x 300.

Centrem aquest rectangle dins del fons negre. La seva posició final serà X:85 i Y:15.

Anomenen aquesta capa *referencia* i la convertim en una capa guia, perquè no es mostri en la pel·lícula final.

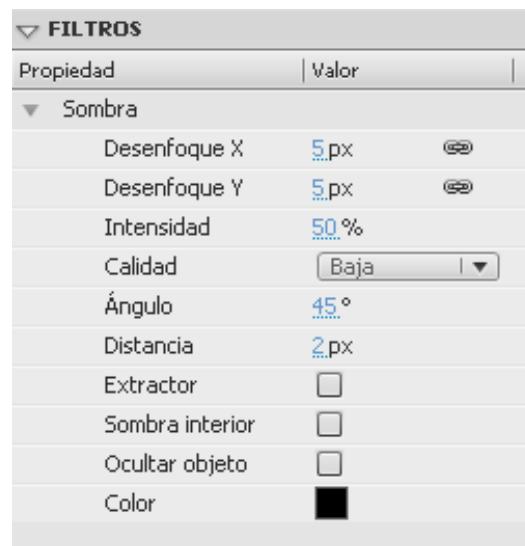
Per convertir la capa en guia, en premem el nom amb el **botó dret** del ratolí i seleccionem **Guía**.



## Tutorial 11. Aplicacions dinàmiques Pas 3 de 23

Afegim una nova capa que anomenarem *botones*, en la qual poscionarem els tres botons per a anar a les seccions d'imatge, vídeo i enllaços respectivament.

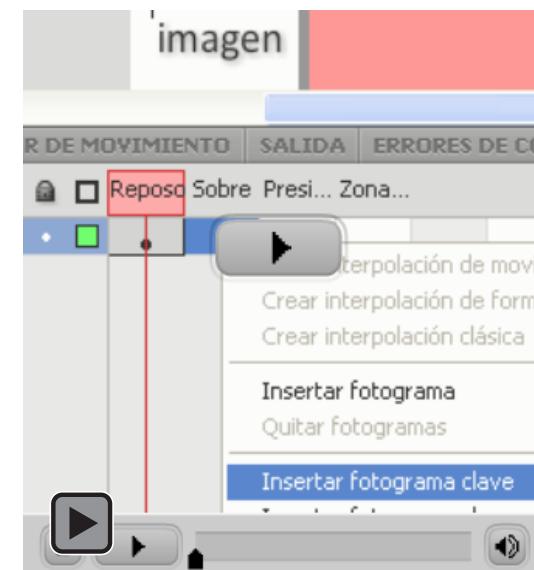
Començarem per crear el botó *imagen*. Escrivim la paraula *imagen* amb el tipus de lletra Calibri (podeu triar el tipus de lletra que preferiu), amb una mida de 20 pt. Després afegim un filtre d'ombra amb el 50% d'intensitat i a una distància de 2 px.



Seleccionem el camp de text creat i premem F8 per convertir-lo en un **símbol de tipus Botón**, que anomenarem *imagen*.

Dins de la línia de temps del botó, **inserim un fotograma clau** per a l'estat **Sobre**, i li canviem el color a #006699.

Quan tornem a l'escena principal, donem a aquesta instància el nom *imagen\_btn*.



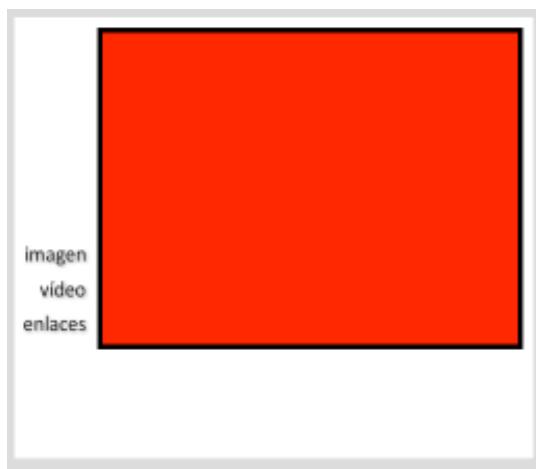
## Tutorial 11. Aplicacions dinàmiques Pas 4 de 23

Creem els altres dos botons (*vídeo* i *enlaces*) de la mateixa manera.

Una manera ràpida de crear-los pot ser duplicant el botó *imagen*, tal com hem fet en el tutorial 8. D'aquesta manera només hauríem de substituir el text *imagen* per *vídeo* i *enlaces* respectivament.

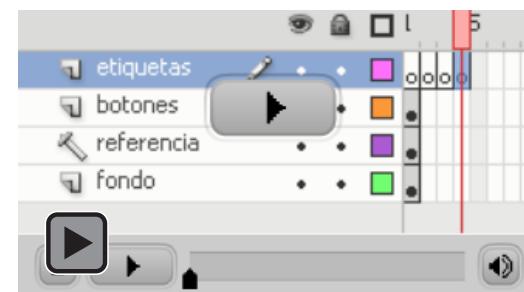
En l'**inspector de Propiedades**, assignem a aquests dos botons nous els noms d'instància *video\_btn* i *enlaces\_btn*.

Els situem en la part inferior esquerra del lloc reservat per a les imatges i els vídeos.



Cada botó ha d'anar a un fotograma diferent que mostrarà la secció corresponent. Crearem aquests fotogrames i els afegirem una etiqueta amb el seu nom de secció, per a facilitar d'aquesta manera la programació de la navegació.

Inserim una nova capa que anomenem **etiquetas**. Inserim **fotogrames clau** en els fotogrames 2, 3 i 4. Podem crear aquests fotogrames clau fent clic sobre cada fotograma i prement la tecla **F6**.



## Tutorial 11. Aplicacions dinàmiques Pas 5 de 23

Premem els fotogrames clau 2, 3 i 4 que hem creat, i els assignem un nom d'etiqueta en l'**inspector de Propiedades**. El número 2 l'anomenem *imagen*, el 3 *vídeo*, i el 4 *enlaces*.



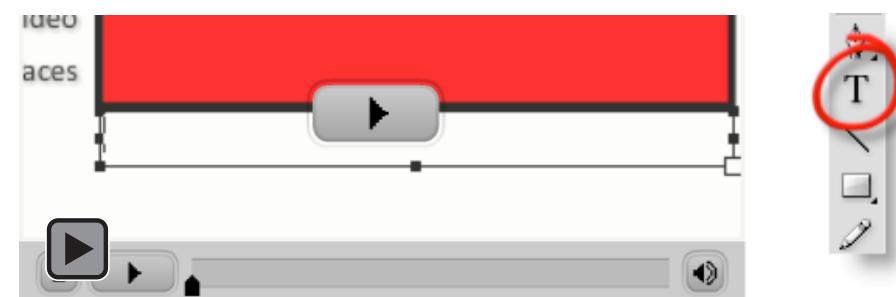
Inserim **fotogrames** en les altres tres capes amb la tecla **F5**, perquè totes les capes creades fins aleshores es mostrin en els quatre fotogrames.



Afegim una nova capa amb el nom *texto* per sobre de la capa *botones*. En aquesta capa afegirem dos quadres de text dinàmic. Un d'aquests quadres mostrarà el títol de la imatge o el vídeo, i l'altre inclourà una petita descripció.

Seleccionem l'**eina Texto**, i en el menú desplegable superior de l'**inspector de Propiedades** seleccionem **Texto dinàmico**.

Fem clic amb l'**eina Texto** sobre l'escenari, en la part esquerra sota la zona en què aniran les imatges i els vídeos. Després arrosseguem la cantonada per a canviar la mida del camp de text.



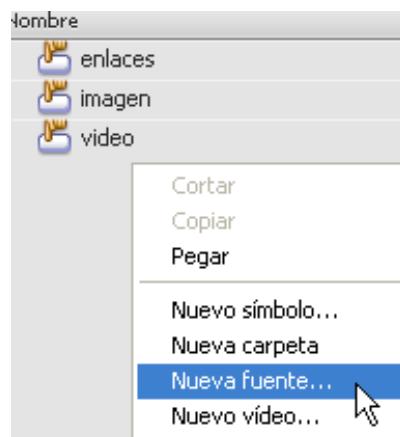
Una vegada creat, podem ajustar-ne la posició arrossegant-lo amb l'eina **Selección**.

## Tutorial 11. Aplicacions dinàmiques Pas 6 de 23

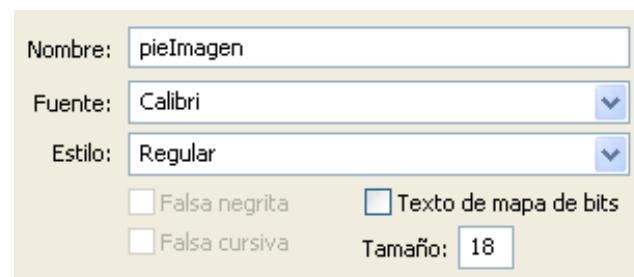
Per a garantir una bona llegibilitat, i perquè el text es visualitzi amb el tipus de lletra triada encara que l'usuari final no la tingui, incorporarem el tipus de lletra en la nostra pel·lícula.

Cal tenir en compte que això augmenta la mida de l'arxiu final. Per als textos estàtics no és necessari incorporar els tipus de lletra.

En primer lloc, fem clic amb el **botó dret** del ratolí sobre un espai buit de la **biblioteca**, i seleccionem **Nueva fuente** en el menú contextual.



Com a nom de la lletra escrivim *pieImagen* com a nom, tipus de lletra Calibri i estil regular (la mateixa que hem utilitzat en els botons).



Si tornem a seleccionar el nostre camp de text en l'escenari, seleccionem com a família el tipus de lletra *pieImagen*, que ara apareixerà inclòs en la part superior del menú desplegable dels tipus de lletra.

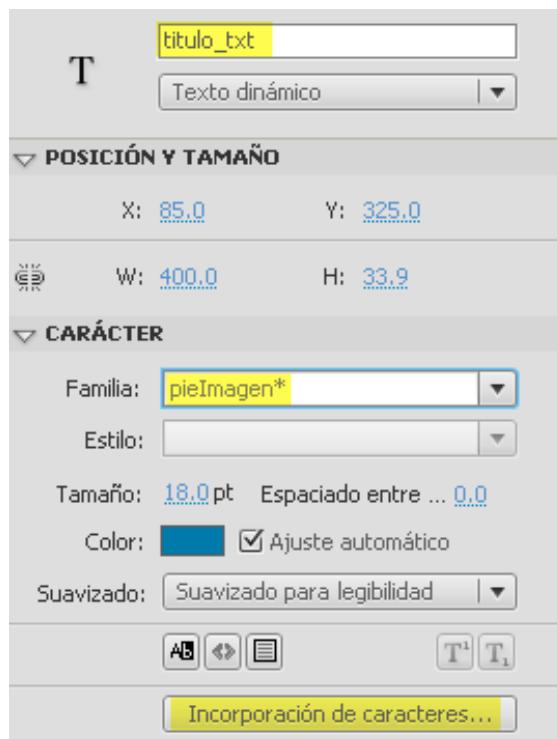
El nom del tipus de lletra té un asterisc per a indicar que és un tipus de lletra inclòs en la pel·lícula.



## Tutorial 11. Aplicacions dinàmiques Pas 7 de 23

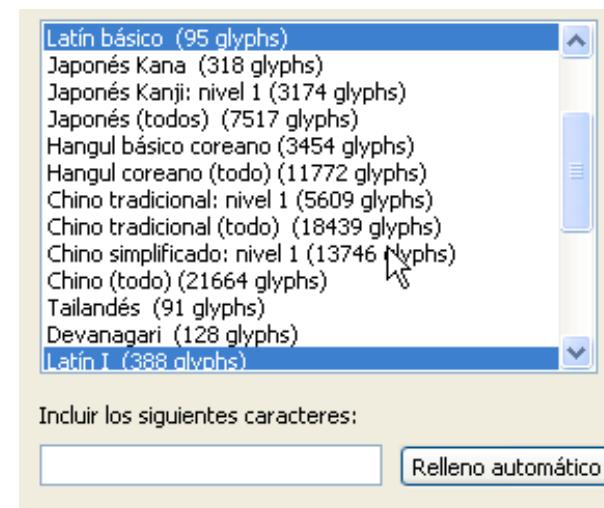
Triem també una mida de 18 pt i el color #06699. Li assignem el nom d'instància *título\_txt*. Li afegim també el mateix efecte d'ombra que incloem en els botons (pas 3).

Ara ens queda incorporar els caràcters que necessitarem d'aquest tipus de lletra. Per a això premem el botó **Incorporación de caracteres**.



Si sabéssim els caràcters exactes que necessitem, els podríem escriure en la casella *Incluir los siguientes caracteres*, i d'aquesta manera disminuiria la mida de l'arxiu final.

Atès que aquests camps no tindran un text fix, inclourem els jocs de caràcters *Latín básico* i *Latín 1* (aquest últim per a accENTS, ñ, ç, etc.).



Aquí podem veure la diferència entre un camp de text que no té caràcters incorporats, i el mateix text amb els caràcters incorporats.

título

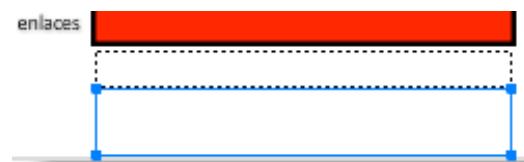
título

## Tutorial 11. Aplicacions dinàmiques Pas 8 de 23

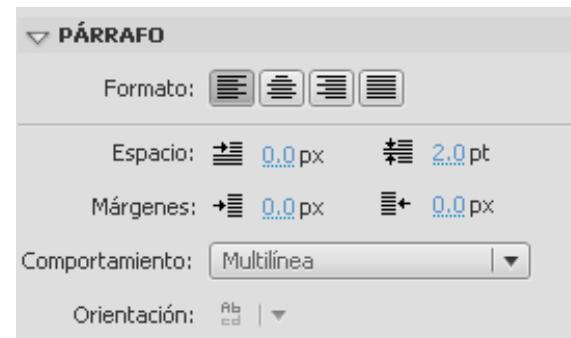
Seguint el mateix procés, creem un altre camp de text al qual donarem el nom d'instància *descripcion\_txt*.

Seleccionem el mateix tipus de lletra incorporat *pielImagen* per a mostrar-lo. En canviem el color pel negre, i la mida, a 16 pt. Aquesta vegada no inclourem cap filtre com lesombres.

Situem aquest camp alineat amb el camp del títol, i el fem més ample perquè hi pugui haver diverses línies.



Perquè puguin mostrar-se unes quantes línies, també hem de seleccionar **Comportamiento Multilínea** en l'inspector de *Propiedades*.



És interessant conèixer la resta d'opcions de paràgraf, com ara l'alineació, la sagnia, l'espai interlineal o els marges, encara que en aquest cas no farem canvis en aquestes propietats.

Ara que ja tenim els camps de text preparats, amb el bloc de notes crearem un arxiu de text extern que anomenarem *informacion.txt*.

Aquest arxiu de text contindrà els títols i les descripcions de les tres seccions de la nostra pel·lícula.

## Tutorial 11. Aplicacions dinàmiques Pas 9 de 23

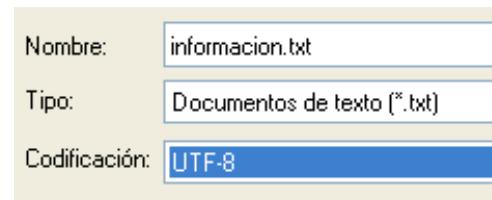
El contingut de l'arxiu de text han de ser parells de variable-valor, és a dir, escriurem el nom d'una variable, després el signe igual, i després el valor per a aquesta variable.

Excepte en la primera variable, les altres han d'anar precedides pel signe &, que és la manera d'indicar que comença una nova variable.

```
tituloImagen=Aquí escrivím el títol de la imatge
&descripcionImagen=La descripció de la imatge.
&tituloVideo=Aquí posarem el títol del vídeo
&descripcionVideo=La descripció del vídeo.
&tituloEnlaces=Aquest és el títol de la secció enllaços
&descripcionEnlaces=I això és la descripció de la secció
enllaços. Provem d'escriure un text una mica més llarg per
comprovar si escriu unes quantes línies.
```

Com podem veure, hem anomenat les variables *tituloImagen*, *descripcionImagen*, *tituloVideo*, etc.

Aquests seran els noms de les variables a què fem referència en la nostra programació des del Flash.



Anomenem l'arxiu *informacion.txt* i seleccionem la codificació UTF-8. El desem en la mateixa carpeta on tinguem el document *tutorial11.fla*.

Ara ja passem a la programació del nostre document del Flash.



Afegim una nova capa per sobre de les altres, que anomenarem **acciones**.

Amb el primer fotograma d'aquesta capa seleccionat, premem F9 per accedir a la finestra Acciones.

Escrivim `gotoAndStop("imagen")`; perquè, en provar la pel·lícula, aquesta vagi a la secció amb l'etiqueta *imagen* i s'aturi. Sense un `stop()` o un `gotoAndStop()` la pel·lícula es reproduiria en un bucle continu.

## Tutorial 11. Aplicacions dinàmiques Pas 10 de 23

Explicarem el procés que cal seguir per a carregar un text extern. En primer lloc, hem de crear un carregador que estigui preparat per a rebre dades de tipus text. Anomenarem aquest carregador *cargadorTexto* (incloem *var* la primera vegada que el definim):

```
var cargadorTexto:URLLoader = new URLRequest();
```

Després li diem que les dades que processarà tenen el format de parells variable-valor, és a dir, que són variables:

```
cargadorTexto.dataFormat = URLLoaderDataFormat.VARIABLES;
```

Li indiquem que, quan hagi carregat les dades per complet, executi una funció que hem anomenat *variablesLeidas*:

```
cargadorTexto.addEventListener(Event.COMPLETE, textoCargado);
```

Finalment li diem que carregui el nostre arxiu *informacion.txt* (si l'arxiu el tinguéssim en una altra carpeta, aquí hauríem d'indicar la ruta relativa a l'arxiu):

```
cargadorTexto.load(new URLRequest("informacion.txt"));
```

Creem la funció *textoCargado*, que s'executarà quan es generi l'esdeveniment *complete*, és a dir, quan tot l'arxiu de text s'hagi carregat.

Amb aquesta funció, li direm que en el text del camp de text *titulo\_txt* escrigui el valor de la variable *tituloImagen* de l'arxiu carregat. Al seu torn, donem la instrucció que en el camp *descripcion\_txt* s'escrigui el valor de la variable *descripcionImagen*.

Ara aquestes dades les té emmagatzemades el carregador del text que havíem creat, per la qual cosa la funció quedarà com s'indica a continuació:

```
function textoCargado(e:Event):void
{
    titulo_txt.text=cargadorTexto.data.tituloImagen;
    descripcion_txt.text=cargadorTexto.data.descripcionImagen;
}
```

Si ara provem la pel·lícula, veurem que apareixen els textos en els camps corresponents. A continuació programarem que aquests textos canviïn en funció del botó premut.

## Tutorial 11. Aplicacions dinàmiques Pas 11 de 23

Afegim els *listeners* per a cada botó de manera similar a com ho hem fet en tutorials anteriors.

Les funcions a les quals criden aquests *listeners* reben el nom de *seccionImagen*, *seccionVideo* i *seccionEnlaces* respectivament.

```
imagen_btn.addEventListener(MouseEvent.CLICK, seccionImagen);  
video_btn.addEventListener(MouseEvent.CLICK, seccionVideo);  
enlaces_btn.addEventListener(MouseEvent.CLICK, seccionEnlaces);
```

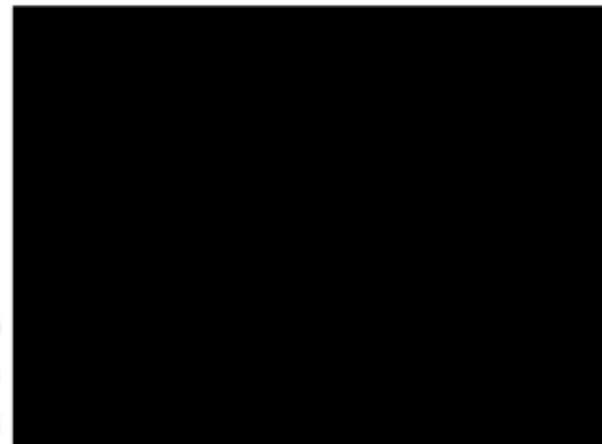
A continuació crearem les tres funcions. D'una banda, s'encarregaran de portar el cap lector al fotograma de la secció corresponent (tot i que per ara tots els fotogrames són iguals). D'altra banda, canviaran el valor dels camps de text del títol i de la descripció.

Aquest seria l'exemple per a la funció *seccionVideo*:

```
function seccionVideo(e:MouseEvent):void  
{  
    gotoAndStop("video"); //nom del fotograma  
    titulo_txt.text = cargadorTexto.data.tituloVideo;  
    descripcion_txt.text = cargadorTexto.data.descripcionV  
}
```

Completem la programació de les altres dues funcions (*seccionImagen* i *seccionEnlaces*) i tornem a provar la pel·lícula.

Els camps de text mostraran en primer lloc el que havíem indicat en la funció *textoCargado* del pas anterior. Després canviaran depenent del botó que premem. Això es produeix malgrat canviar de fotograma, ja que els camps de text es mantenen visibles durant els fotrames de totes les seccions.



Esto es el título de la sección enlaces

Y esto es la descripción de la sección enlaces. Probamos a escribir algo más largo para comprobar si escribe varias líneas.

## Tutorial 11. Aplicacions dinàmiques Pas 12 de 23

Carregarem la imatge que apareixerà en la secció *imagen*, i que també serà la que es carregui per defecte en obrir el document.

En primer lloc, creem un carregador per a la imatge, que en aquest cas és un objecte *Loader*, en comptes d'un objecte *URLLoader*, que era l'utilitzat per a carregar el text. L'objecte *Loader* es fa servir quan hem de carregar de manera externa arxius jpg, png, gif i swf.

```
var cargadorImagen:Loader = new Loader();
```

Com veurem a continuació, en aquest cas, els *listeners* no s'afegeixen al carregador, sinó al contingut, per la qual cosa, hem d'afegir *contentLoaderInfo* darrere del nom del carregador. Afegirem dos *listeners*, un que ens informi del progrés de la càrrega de la imatge executant una funció anomenada *cargandoImagen*, i un altre que iniciï la funció *imagenCargada* quan el contingut s'hagi carregat per complet.

```
cargadorImagen.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS,cargandoImagen);  
cargadorImagen.contentLoaderInfo.addEventListener(Event.COMPLETE,imagenCargada);
```

Per acabar, carreguem el contingut, que en aquest cas és una imatge. Aquesta imatge és a la carpeta *tutorial11*. Heu de traslladar-la, juntament amb el vídeo *videodeldia.flv* que també trobareu en la mateixa carpeta, al mateix directori on tingueu el document *tutorial11.fla*.

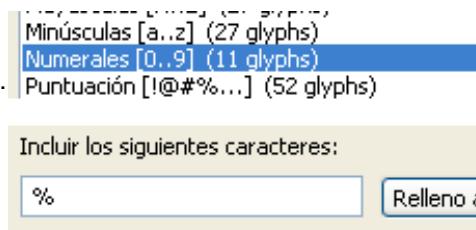
```
cargadorImagen.load(new URLRequest("imagendeldia.jpg"));
```

Si voleu, podeu provar aquest procés amb qualsevol altra imatge, sempre que tingui una mida de 400 x 300 px. En aquest cas caldrà posar el nom de la imatge que vulgueu carregar al seu lloc, o bé anomenar la vostra imatge *nombreimagendeldia.jpg*. Si la imatge la tinguéssiu dins d'una carpeta anomenada *img*, la manera de fer-hi referència seria *img/imagendeldia.jpg*.

## Tutorial 11. Aplicacions dinàmiques Pas 13 de 23

Volem que la funció *cargandoImagen* ens mostri el tant per cent carregat. Per mostrar aquest text, en la capa *texto* creem un altre camp de text.

Per a aquest camp de text n'hi hauria prou d'incloure els caràcters numerales i el caràcter %.



Alineem el paràgraf en el centre, i situem el camp de text en un lloc central de la zona en què es carregarà la imatge. Canviem el color a blanc, ja que aquest text es veurà sobre fons negre. Finalment hi donem el nom d'instància *porcentaje\_txt*.

Calculem el percentatge de la càrrega, que serà la quantitat de bytes carregats dividida per la quantitat total de bytes de l'objecte que s'està carregant (l'objecte que ha cridat aquesta funció), multiplicada per 100. Perquè no aparegui un nombre decimal, arrodonim a la baixa amb *Math.floor* la xifra obtinguda. Després mostrem el resultat en el camp de text que havíem creat.

```
function cargandoImagen(e:ProgressEvent):void
{
    var porcentaje:Number = Math.floor(e.bytesLoaded/e.bytesTotal *
100);
    porcentaje_txt.text = porcentaje + "%";
}
```

Com veiem, primer hem creat la variable *porcentaje*, que és un nombre. Després, en el camp de text del percentatge, mostrem el percentatge seguit del símbol %.

Quan sumem un nombre i una cadena de text, el resultat és un text, i per tant es pot mostrar directament en un camp de text.

Si no haguéssim afegit el símbol %, hauríem d'haver convertit prèviament el nombre del percentatge en un text amb *String* perquè no donés error:

```
porcentaje_txt.text = String(porcentaje);
```

## Tutorial 11. Aplicacions dinàmiques Pas 14 de 23

Ara programarem la funció *imagenCargada*.

En primer lloc, volem que quan la imatge s'hagi carregat ja no es mostri el text del percentatge de la càrrega. Una manera senzilla de fer-ho és assignar al text un valor buit, i això s'aconsegueix obrint i tancant cometes sense res a l'interior.

Podíem haver dit que el camp de text fos invisible, però això tindria el desavantatge que ens hauríem de recordar de tornar-lo a fer visible quan una altra imatge estigués a punt de carregar-se.

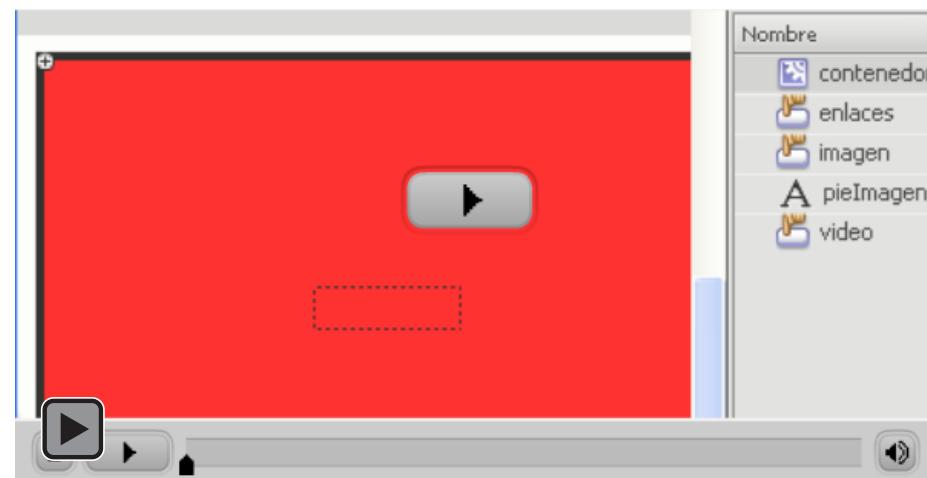
Per acabar, hem d'afegir a la pel·lícula la imatge que s'ha carregat. Podríem afegir-la directament a l'escena principal. En aquest cas es posicionaria en la part superior esquerra de l'escenari. Després podríem donar-li les coordenades de la posició correcta.

Una manera més pràctica de posicionar i controlar un objecte que s'afegeix a una pel·lícula és crear un clip contenidor, i carregar l'objecte dins del contenidor. D'aquesta manera, podem posicionar el contenidor on vulguem, i automàticament l'objecte contingut es posicionarà al mateix lloc.

Seleccionem **Insertar > Nuevo Símbolo** i creem un símbol de tipus **Clip de película** amb el nom *contenedor*.

Creem una capa que també anomenarem *contenedor*, i que situarem per sobre de la capa *texto*.

Tornem a l'escena principal. Encara que el clip *contenedor* no té contingut, es pot arrosseggar igualment de la biblioteca a l'escenari per a crear una instància. Una creu petita ens indicarà la posició. Situem el clip en la cantonada superior esquerra del rectangle de referència.



## Tutorial 11. Aplicacions dinàmiques Pas 15 de 23

Donem al clip el nom d'instància *contenedor\_mc* en l'inspector de *Propiedades*, i de pas ens assegurem que la posició del clip sigui X: 85 i Y:15.



Ara ja podem crear la funció *imagenCargada* i afegir la imatge que s'ha carregat dins del clip *contenedor\_mc*.

```
function imagenCargada(e:Event):void
{
    porcentaje_txt.text = "";
    contenedor_mc.addChild(cargadorImagen);
}
```

La imatge carregada mostrarà la seva cantonada superior esquerra en la posició de registre del clip *contenedor*.

Provem la pel·lícula, i veiem que la imatge es carrega directament, sense veure el text amb el percentatge. Això es produeix perquè quan es tenen els arxius de manera local, la càrrega és instantània.

Des del menú del *player* en què es mostra l'swf, quan provem la pel·lícula, podem simular una descàrrega per a comprovar si el nostre carregador funciona correctament.

En primer lloc, en el *player* seleccionem **Ver > Configuración de descarga > DSL** (o 56 K o el valor que vulguem). Després seleccionem **Ver > Visor de anchos de banda**. Finalment, seleccionem **Simular descarga**.

En la part superior esquerra de la pantalla es mostraran uns quants detalls de la pel·lícula, com la mida, fps, en quin fotograma és, com va la càrrega, etc.

Veurem que la pantalla resta una estona en blanc, que és el temps en què s'estan carregant els elements gràfics previs, i després podrem veure el percentatge de la càrrega de la nostra imatge.

## Tutorial 11. Aplicacions dinàmiques Pas 16 de 23

La imatge de la nostra pel·lícula només pesa 20 kB, per la qual cosa, el percentatge de càrrega és molt ràpid. Podem fer proves amb fotografies de més pes per a comprovar el funcionament del carregador.

De moment, el nostre document pesarà uns 90 kB, que en realitat es carregaria més ràpidament del que mostra el simulador de descàrrega. De tota manera és una eina molt útil per a comprovar el funcionament de carregadors o fer-nos una idea clara de quines parts de la nostra pel·lícula triguen més a carregar-se.

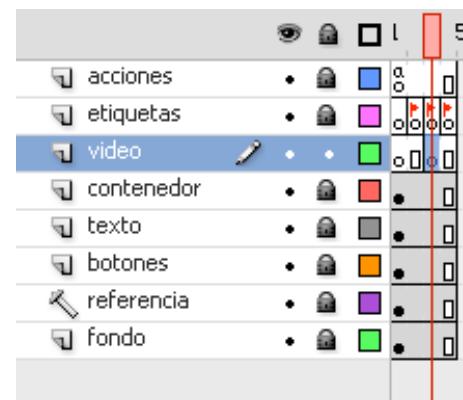
Continuem amb la programació de la nostra pel·lícula. La imatge s'ha carregat però es mostra en els fotogrames de totes les seccions. Aquesta vegada l'ocultarem assignant un codi `visible = false` quan es premi en la secció de vídeo o d'enllaços, i `visible = true` quan es premi en la secció d'imatges.

També podríem eliminar el fotograma que conté el contingut en les seccions de vídeo i enllaços, però llavors hauríem de tornar a carregar la imatge cada vegada que tornéssim a l'apartat d'imatge.

Per tant, afegim el codi `contenedor_mc.visible = false;` dins de les funcions `seccionVideo` i `seccionEnlaces`, i `contenedor_mc.visible = true;` en la funció `seccionImagen`.

Ara que tenim la secció d'imatge acabada, passarem a la secció de vídeo.

Afegim una nova capa, i aquesta vegada inserim un fotograma clau en el fotograma 3, que és el fotograma al qual correspon l'etiqueta `video`.

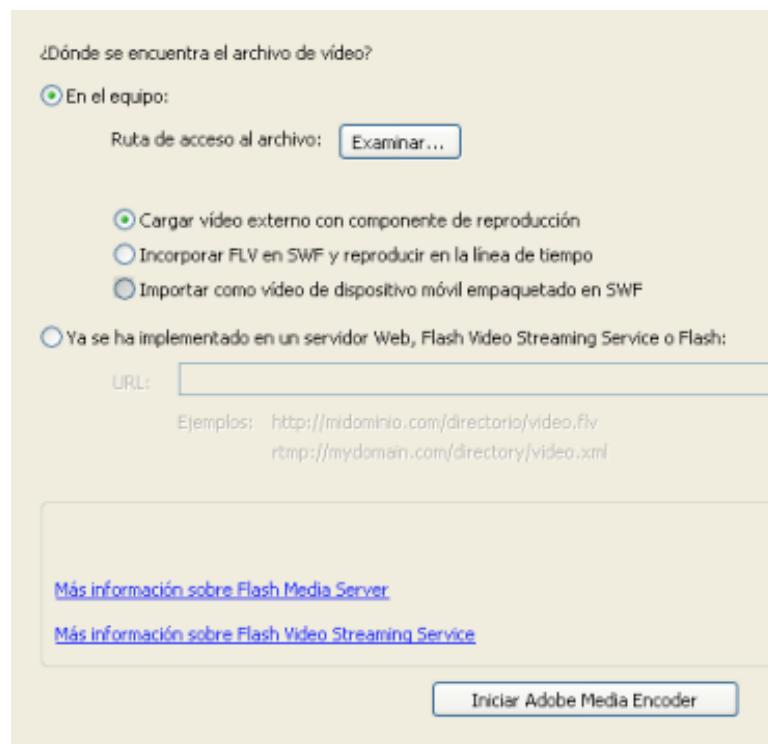


En aquest fotograma, seleccionem **Archivo > Importar > Importar vídeo**. Se'n obrirà una finestra nova amb les opcions per a importar vídeo.

## Tutorial 11. Aplicacions dinàmiques Pas 17 de 23

Si el vídeo estigués implementat en un servidor, podríem indicar directament la URL on és.

En el nostre cas, seleccionem **En el equipo**, i fem clic a **Examinar** per veure la ruta d'accés.



Seleccionem l'arxiu *videodeldia.flv*. Si tenim algun altre arxiu FLV també el podem utilitzar.

L'opció d'inserir FLV en SWF i reproduir en la línia de temps només seria recomanable per a vídeos molt curts (uns quants segons) i sense àudio, ja que hi pot haver problemes de sincronització.

L'opció d'importar com a vídeo de dispositiu mòbil és per a quan s'està preparant contingut per a distribuir-lo en telèfons mòbils i altres dispositius electrònics.

Per tant, seleccionarem **Cargar vídeo externo con componente de reproducción** i farem clic a **Siguiente**.

Si el vídeo no fos FLV, apareixeria un avís perquè iniciéssim l'Adobe Media Encoder per convertir el vídeo en un arxiu FLV compatible.

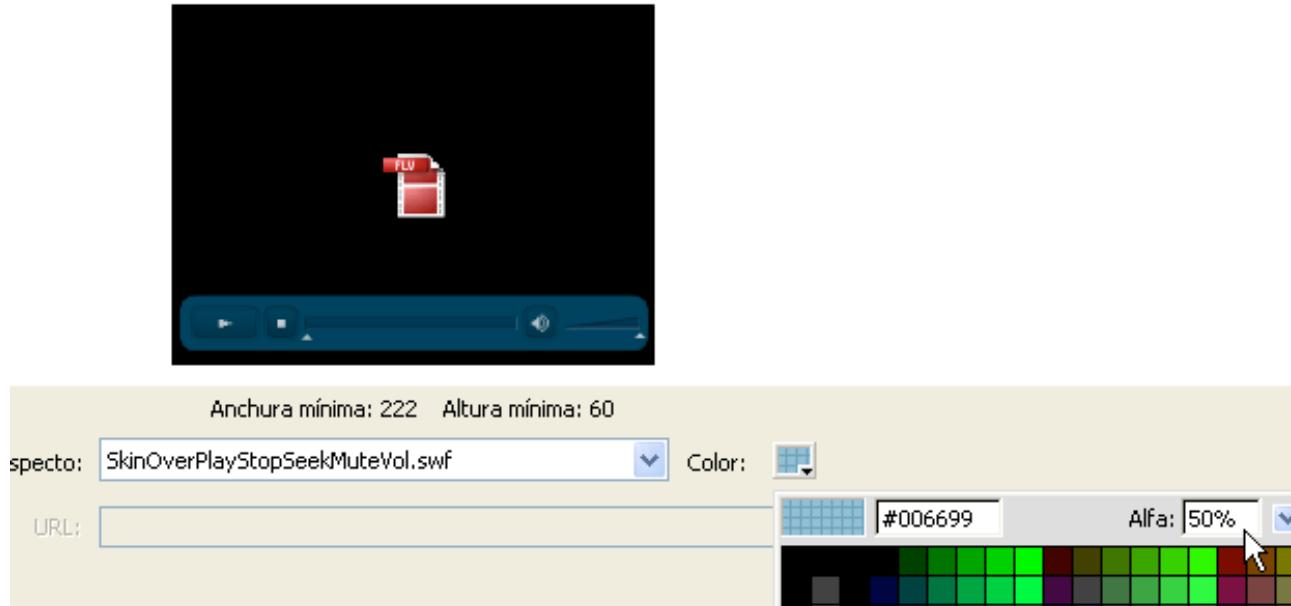
## Tutorial 11. Aplicacions dinàmiques

### Pas 18 de 23

En la pantalla següent podem triar un reproductor amb més o menys controls, i que es mostri per sota o per sobre del vídeo. També podrem personalitzar el color.

Per exemple, podem seleccionar el color #006699 amb una transparència del 50%, i que es mostri sobre el vídeo (*SkinOver...*).

Fem clic a **Siguiente** i després a **Finalizar** sense necessitat de marcar la casella de veure l'ajuda del vídeo.

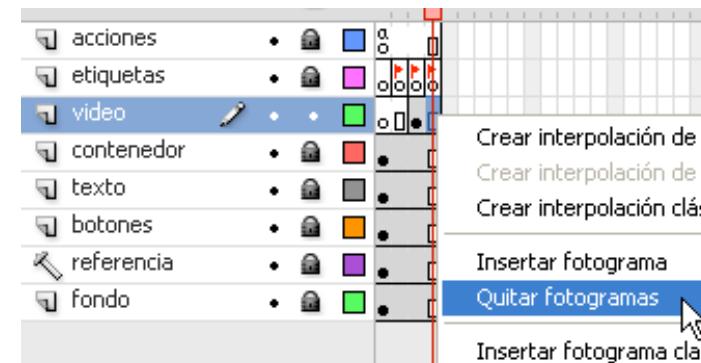
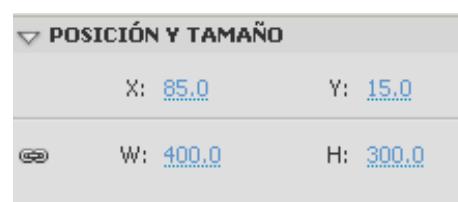


## Tutorial 11. Aplicacions dinàmiques Pas 19 de 23

En l'escenari ara tindrem una instància del component FLVPlayback. No cal donar un nom a aquesta instància.

La posicionem en el mateix lloc que el contingut de les imatges, és a dir, a X:85 i Y:15.

També podem ampliar-ne la mida a W: 400 i H:300. No és recomanable canviar la mida dels vídeos, perquè pot ser que es vegin pixel·lats, però en aquest cas canviarem la mida perquè coincideixi amb l'àrea de la imatge.



Traiem l'últim fotograma per eliminar el vídeo en la secció enllaços, és a dir, en el fotograma 4.

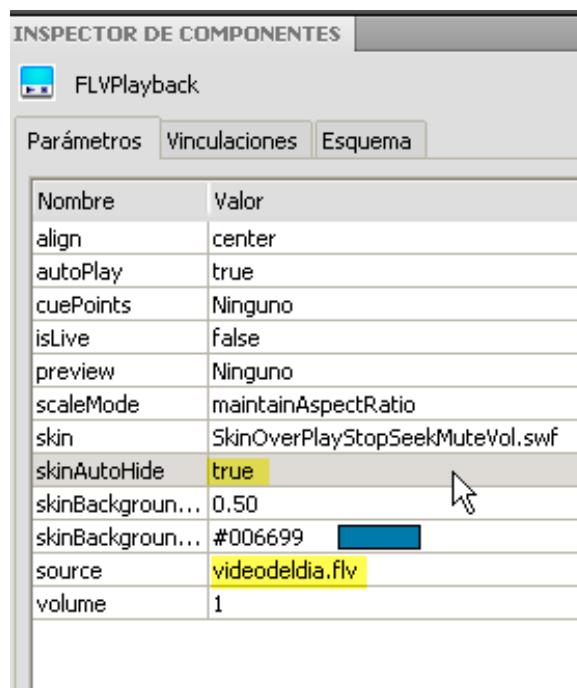
Per a això fem clic amb el **botó dret** del ratolí sobre el quart fotograma de la capa vídeo, i seleccionem **Quitar fotogramas**. També podríem haver seleccionat *Insertar fotograma clave en blanco*.

Provem la pel·lícula per comprovar que el vídeo es reproduceix correctament.

## Tutorial 11. Aplicacions dinàmiques Pas 20 de 23

En el cas dels vídeos, a diferència de les imatges o el text, el contingut no es carrega en la pel·lícula, sinó que es llegeix directament de manera externa.

Amb el component del vídeo seleccionat en l'escenari, fem clic en el símbol  en l'**inspector de Propiedades** o seleccionem **Ventana > Inspector de componentes**.



Des d'aquesta finestra podem configurar diversos aspectes del nostre vídeo.

Per exemple, ara està seleccionat **autoPlay** com a **true**, la qual cosa significa que el vídeo comença a reproduir-se automàticament. El mantindrem com a **true**.

Des d'aquí també podem canviar el reproductor (**skin**) utilitzat, i també la seva visualització. Canviarem la propietat **skinAutoHide** a **true** perquè els controls només es mostrin si posicionem el punter sobre el vídeo.

Des d'aquí també podem canviar el vídeo que es reproduirà. Així que en el cas dels vídeos, si volguéssim actualitzar el vídeo que es mostra, podríem o bé canviar des d'aquesta pantalla el vídeo a què fa referència o bé, de manera externa, podríem seleccionar un altre vídeo i donar-hi aquest mateix nom.

Si provem la pel·lícula, podem comprovar que malgrat que el vídeo desapareix en les seccions *imagen* i *enlaces*, el seu àudio continua sonant. Per a apagar el so inclourem la instrucció `SoundMixer.stopAll();` en les funcions *secciónImagen* i *secciónEnlaces*.

## Tutorial 11. Aplicacions dinàmiques Pas 21 de 23

Per a l'última secció, creem una nova capa anomenada *enlaces*, i creem un fotograma clau en el fotograma 4, que és el corresponent a la secció *enlaces*.

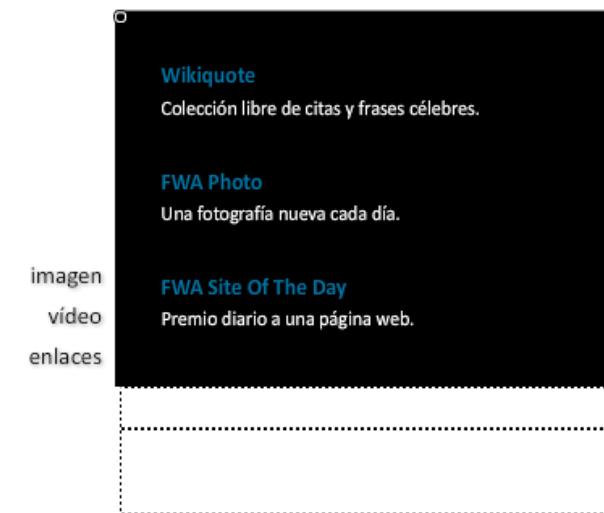
Amaguem la capa *referencia*, que ens havia ajudat per posicionar el contingut i el vídeo, per poder veure així el fons negre real sobre el qual es mostraran els textos dels enllaços.



Aquesta vegada crearem els camps de text com a text estàtic. Per tant, aquest apartat no serà dinàmic.

Escrivim els noms dels webs que vulguem enllaçar i, a sota, una petita descripció del lloc.

Seleccionem tipus de text diferents per al títol del web, que es convertirà en l'enllaç, i la descripció. En aquest cas hem utilitzat el mateix tipus de lletra que en la resta de la pel·lícula. Hem assignat als títols l'estil *Bold* (negreta) i una mida de lletra més gran.



## Tutorial 11. Aplicacions dinàmiques Pas 22 de 23

La manera més ràpida de crear un enllaç és seleccionar una part d'un text, o un camp de text al complet i, en l'**inspector de Propiedades**, en l'apartat **Vínculo**, escriure l'adreça web (URL).



Quan creem un vincle d'aquesta manera, el text que s'ha convertit en un vincle apareix subratllat. Tanmateix, quan provem la pel·lícula, aquest subratllat no es mostrerà.

Com a destinacions més habituals podem triar entre **\_blank**, que obre el vincle en una finestra nova **\_self**, que obre el vincle en la finestra actual

Per defecte, si no seleccionem una destinació, el contingut s'obrirà en una nova finestra.

Si volem que el vincle sigui una adreça electrònica, escriurem el vincle amb el format <mailto:nombre@web.com>.

Si preferim afegir el vincle a un botó en comptes d'afegir-lo en un text, haurem d'afegir el codi següent a la funció corresponent a aquest botó:

```
navigateToURL(new URLRequest("http://www.direccion.com"));
```

Els vincles funcionaran quan provem la pel·lícula, però tanmateix, per un problema de seguretat, no funcionaran si obrim directament l'swf.

Perquè un swf amb vincles sigui totalment funcional, s'ha d'haver pujat a un servidor d'Internet.

En el cas de la nostra aplicació, perquè tot funcioni hauríem de pujar els arxius extens que hem utilitzat (text, imatges i vídeo) amb l'swf generat, i també l'swf de l'skin que hem seleccionat per al vídeo, que podem veure que és en la mateixa carpeta que el nostre *tutorial11.swf*.

**Tutorial 11. Aplicacions dinàmiques**  
**Pas 23 de 23**

imagen  
vídeo  
enlaces



**Bar y habitación 3D**

Imágenes realizadas con la herramienta 3D Studio.

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

1. Modifiqueu els títols i les descripcions de cada secció en l'arxiu extern de text.
2. Afegiu al menú principal, sota el botó *enlaces*, un botó anomenat *contacto* que sigui un vincle al vostre correu electrònic.
3. Sense modificar el document fla, canvieu la imatge i el vídeo que es mostren en la pel·lícula

## Tutorial 12. Publicació i exportació Pas 1 de 10

Una vegada hagim creat una pel·lícula Flash, tant si és una animació, com un joc, un recurs educatiu, etc., l'últim pas és publicar-la o exportar-la a un altre format, segons la seva finalitat i canal de distribució.

Una opció habitual és que la nostra pel·lícula es visualitzi en una pàgina web. La manera més ràpida de generar un arxiu HTML en el qual es pugui visualitzar el nostre arxiu SWF és amb les opcions de publicació incorporades en el Flash.

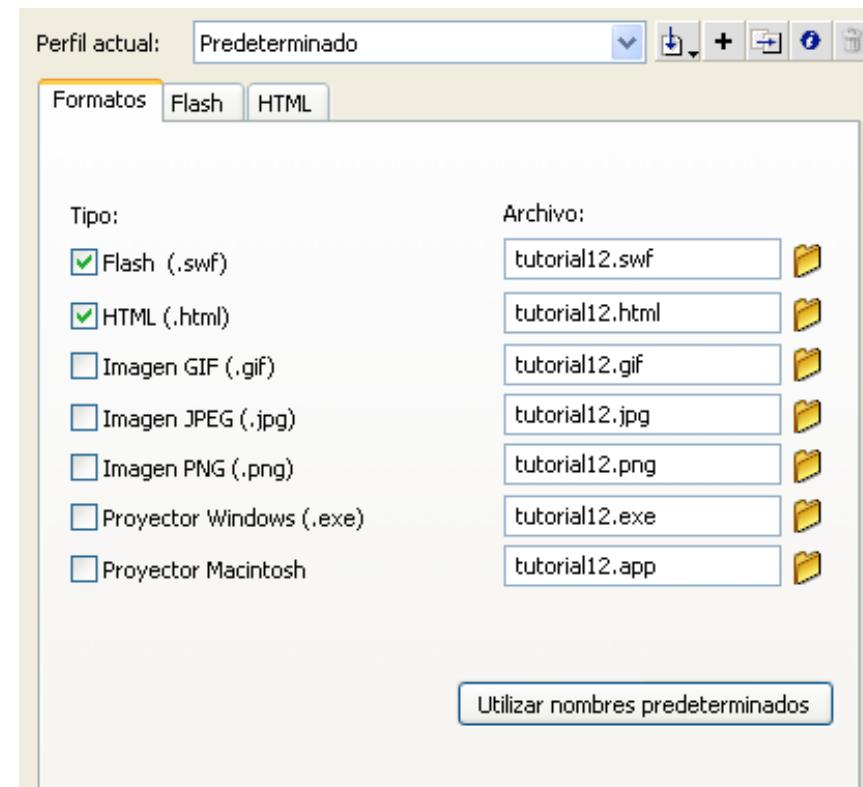
Obrim un document fla amb el contingut que vulguem publicar.

Seleccionem **Archivo > Opciones de configuración**.

De manera predeterminada, apareixeran seleccionades les caselles **Flash** i **HTML**, que són les que calen per a crear una pàgina html que contingui el document SWF.

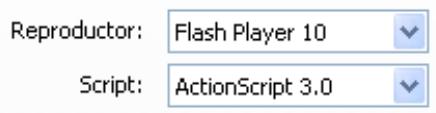
Els noms dels arxius que es crearan tindran el nom del nostre arxiu fla. Podem canviar aquests noms sense problemes.

Per cada tipus de format que seleccionem, excepte per als projectors, es mostrarà una pestanya en la qual podrem configurar les característiques de l'arxiu que volem publicar.



## Tutorial 12. Publicació i exportació Pas 2 de 10

Començarem per descriure les opcions més importants que cal tenir en compte de la pestanya **Flash**.



En l'opció **Reproductor** seleccionarem la versió de Player necessària per a veure la nostra pel·lícula. Llevat de si hem de crear un projector, que porta incorporat el propi *player*, els usuaris hauran de tenir una versió igual o superior del connector Flash Player que la que seleccionem per veure correctament el contingut de la pel·lícula.

Per això és convenient, si fos possible, seleccionar una versió del reproductor el més baixa possible. Tanmateix, pot ser que moltes característiques que hagim utilitzat en la nostra pel·lícula no siguin compatibles amb les versions anteriors del Player.

Si seleccionem una versió del reproductor que no sigui compatible amb algunes de les funcions que hem utilitzat en la nostra pel·lícula, una finestra d'alerta ens informarà d'aquesta incompatibilitat.

En l'opció **Script** seleccionarem la versió del llenguatge de *script* que hem utilitzat en la nostra pel·lícula. En aquesta guia hem utilitzat ActionScript 3.0. En algunes pel·lícules, encara que no hagim escrit codi, pot ser que també calgui exportar amb la versió 3.0.

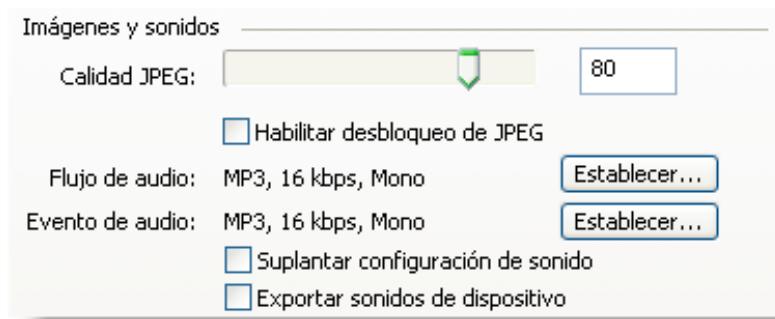
Si seleccionem una versió del reproductor inferior a Flash Player 9, automàticament la versió de l'*script* canviarà a 2.0 o 1.0, segons la versió del reproductor seleccionada.

Si hem seleccionat una versió de l'*script* que sigui incompatible amb el nostre codi o amb algun element de la pel·lícula, en publicar la pel·lícula es mostrarà un avís en el tauler *Salida*, o apareixeran errors al tauler *Erros de compilador*.

Algunes de les funcions del Flash que necessiten l'última versió tant del reproductor com de l'*script* són les interpolacions 3D o l'eina *huesos*.

## Tutorial 12. Publicació i exportació Pas 3 de 10

En la part superior de l'àrea d'*Imágenes y sonidos* tenim en primer lloc **Calidad JPEG**. Aquí podem seleccionar la qualitat dels mapes de bits utilitzats en la nostra pel·lícula. Una imatge amb una qualitat més baixa s'ha de comprimir més, per tant, s'obté un arxiu amb un pes més petit.



Respecte a l'àudio, en principi, la configuració de la qualitat de so serà la que estigui configurada des de les propietats de cada so en la biblioteca en el cas dels esdeveniments d'àudio, tal com hem vist en el tutorial 8.

Tanmateix, podem seleccionar la casella **Suplantar configuración de sonido** per a establir des d'aquí la qualitat dels sons en l'arxiu final.

En cas que en la nostra pel·lícula hagim utilitzat sons a com flux d'àudio, és necessari establir des d'aquí la qualitat de la compressió del so, fins i tot encara que la casella de suplantar la configuració del so no estigui seleccionada.

Quan la nostra pel·lícula estigui publicada, una de les diferències que podrem trobar entre els sons de flux i els d'esdeveniment, a més de les explicades en el tutorial 8, és que un so de **flux** començarà a reproduir-se juntament amb la resta de la pel·lícula quan s'hagin descarregat els primers fotogrames, i podrà aturar-se en algun punt si la reproducció avança a més velocitat que la descàrrega.

Tanmateix, un so de tipus **esdeveniment** s'haurà d'haver descarregat completament abans de començar la reproducció, per la qual cosa trigarà més a iniciar-se, encara que després no s'aturarà fins que no ho indiquem de manera explícita.

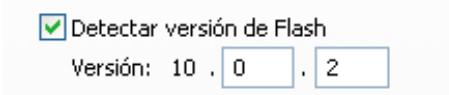
## Tutorial 12. Publicació i exportació Pas 4 de 10

Si passem a la pestanya **HTML**, una de les primeres opcions que ens podem trobar és si volem que es detecti de manera automàtica la versió del Flash del navegador.

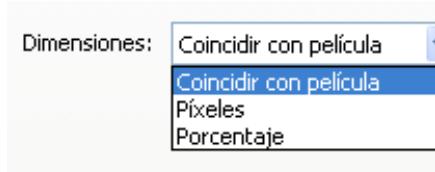
Si seleccionem la casella **Detectar versión de Flash** i l'usuari té una versió inferior del Flash, se l'enviarà automàticament a una pàgina per a actualitzar el reproductor.

Si la casella no està seleccionada, l'arxiu es reproduirà encara que l'usuari no disposi de la versió adequada, però les funcions incompatibles amb la versió de l'usuari es mostraran amb errors.

Per això, sempre resulta convenient assegurar-se que l'usuari visualitza la pel·lícula amb la versió adequada del *player*.



En l'apartat *Dimensiones*, podem seleccionar si la pel·lícula ha de mostrar-se amb la mateixa mida que el nostre document, o bé si ha d'ocupar un nombre concret de píxels o un percentatge de la finestra del navegador.

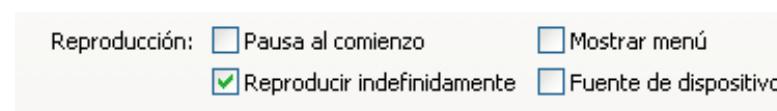


Si seleccionem píxels o percentatge, és possible que l'usuari visualitzi parts de la pel·lícula que eren fora dels límits del nostre document.

En pel·lícules que incloguin mapes de bits o vídeos, no és convenient redimensionar la pel·lícula, ja que les imatges i vídeos poden mostrar-se pixel·lats.

## Tutorial 12. Publicació i exportació Pas 5 de 10

En l'àrea de *Reproducción*, podem seleccionar, entre altres options, si la pel·lícula s'ha de **reproducir indefinidament** (com un bucle). D'aquesta manera, només s'aturarà quan ho explicitem amb un *stop*. En cas que la casella no estigui seleccionada, la pel·lícula es reproduirà al complet, i s'aturarà en arribar a l'últim fotograma.



Si la casella **Mostrar menú** està activada, es mostrarà un menú contextual quan iniciem la pel·lícula amb el botó dret del ratolí. L'usuari podrà avançar i retrocedir la pel·lícula, canviar la visualització, etc.

Per desactivar aquest menú contextual, i que només aparegui informació sobre el Flash, desactivem la casella *Mostrar menú*.

Consulteu l'ajuda del Flash per conèixer les característiques de la resta d'opcions de publicació d'HTML.

Seleccionem **Archivo > Vista previa de publicación > HTML** per veure el resultat de la nostra pel·lícula amb la configuració actual. Podem fer diverses proves amb les options d'HTML fins a aconseguir que la pel·lícula es mostri com volem.

Per publicar la pel·lícula seleccionem **Archivo > Publicar**, o bé premem el botó **Publicar** de la finestra de *Configuración de publicación*.

Per a poder visualitzar la pel·lícula a Internet, haurem de pujar tant l'arxiu swf com l'arxiu html que hem publicat.

Si observem el codi font de l'arxiu html que el Flash ha generat automàticament, veurem que hi ha moltes línies de codi.

Si volem incloure la nostra pel·lícula com a part d'una pàgina web amb un altre contingut, l'arxiu HTML que creï el Flash pot resultar confús i difícil de compatibilitzar.

Per això recomanem publicar des del Flash només el nostre arxiu swf, i per a integrar el contingut en un arxiu HTML utilitzar una biblioteca JavaScript que detallarem a continuació.

## Tutorial 12. Publicació i exportació Pas 6 de 10

Per a la publicació de contingut Flash incrustat en una pàgina web, hi ha una biblioteca JavaScript anomenada **swfobject.js**, que permet incloure la nostra pel·lícula en un arxiu html respectant els estàndards web.

Els arxius necessaris i la documentació sobre aquesta biblioteca la podem trobar a <http://code.google.com/p/swfobject/>.

En la mateixa pàgina podem trobar un generador del codi necessari per a integrar el nostre SWF amb aquesta biblioteca, la qual cosa facilita la tasca de crear el codi.

En la part superior del generador del codi hem d'incloure, en primer lloc, la ubicació on hi ha el nostre arxiu swfobject.js.

SWFObject configuration [ [\\_](#) ]

|                         |   |
|-------------------------|---|
| SWFObject (.js): *      | <input type="text" value="swfobject.js"/>   |
| Publishing method: *    | <input type="text" value="Static publishing"/> <a href="#">what is this?</a>                      |
| Detect Flash version: * | <input type="text" value="10"/> . <input type="text" value="0"/> . <input type="text" value="0"/> |
| Adobe Express Install:  | <input checked="" type="checkbox"/> <input type="text" value="expressInstall.swf"/>               |

També hem de determinar la versió del Flash Player necessària per a visualitzar la nostra pel·lícula.

Podem incloure l'Adobe Express Install, un petit instal·lador que apareix en cas de no disposar de la versió del connector necessària, i que permet la instal·lació d'una versió actual del Player sense sortir de la nostra pàgina.

En l'àrea de definició de l'SWF, hem d'indicar el nom i la ubicació del nostre arxiu respecte al document HTML. També hem d'especificar-ne les dimensions (en pixels o en percentatge). En l'àrea *more* podrem definir altres paràmetres com, per exemple, si volem que es mostri o no el menú contextual.

SWF definition [ [\\_](#) ]

|                      |   |
|----------------------|---|
| Flash (.swf): *      | <input type="text" value="tutorial12.swf"/>   |
| Dimensions: *        | <input type="text" value="550"/> x <input type="text" value="400"/> <input type="text" value="pixels"/> |
| Flash content id *   | <input type="text" value="mipelicula"/>   |
| <a href="#">more</a> |   |

## Tutorial 12. Publicació i exportació Pas 7 de 10

En l'àrea de l'HTML podem definir el contingut alternatiu que es mostra als usuaris que no disposen de la versió del Flash Player necessària.



Si, per exemple, es tractés d'un bàner animat que no fos imprescindible per al funcionament de la pàgina, podríem mostrar simplement una imatge fixa en substitució de l'animació swf. En altres casos en què la pel·lícula tingui més importància, podríem incloure una descripció del contingut, i indicar que per a visualitzar correctament l'aplicació cal tenir una versió més actual del Flash Player.

Finalment premem **Generate** per generar el codi html que cal per a visualitzar la nostra pel·lícula.

Si volem que la nostra pel·lícula formi part d'un altre document html ja creat, n'hi ha prou de copiar les línies de *script* que hi ha en la capçalera del codi creat.

```
<script type="text/javascript" src="swfobject.js"></script>
<script type="text/javascript">
    swfobject.registerObject("mipelicula", "10.0.0");
</script>
```

També hem d'enganxar tot el contingut que hi ha entre les etiquetes *<div> </div>* (incloses) dins del cos de la nostra pàgina.

D'aquesta manera tindrem tot el contingut de la nostra pel·lícula dins d'una etiqueta div. Per a comprendre els avantatges que això representa, cal que tinguem coneixements previs d'HTML i CSS.

No hem d'oblidar incloure l'arxiu swfobject.js amb la publicació de la nostra pàgina, i també expressInstall.swf si l'haguéssim seleccionat.

## Tutorial 12. Publicació i exportació Pas 8 de 10

Tornant a la finestra **Configuración de publicación**, també podem publicar arxius GIF, JPEG i PNG per a mostrar una imatge del nostre document.

Les últimes caselles són per a crear **projectors**, és a dir, arxius executables que contenen la nostra pel·lícula i que inclouen un reproductor propi independent. D'aquesta manera no cal que l'usuari final disposi de la versió necessària del Flash Player per a visualitzar el contingut.

Aquesta forma de publicació és la més habitual per a crear CD interactius amb contingut fet en Flash.

Per a un executable en Windows n'hi ha prou amb l'arxiu EXE que genera el projector, per la qual cosa no cal generar un arxiu SWF.

Quan es genera un executable per a un CD, és habitual incloure un arxiu anomenat *autorun.inf* que indiqui que el nostre executable s'ha de reproduir automàticament. Per a això tindrà el contingut següent:

```
[autorun]
open=nombreArchivo.exe
```

Si hem de crear un projector per al Linux, podem descarregar-ne un a <http://www.adobe.com/support/flashplayer/downloads.html>, obrir amb ell el nostre swf, i crear un projector nou des del Linux amb el nostre contingut.

En l'arxiu que hem creat amb el Flash i que està pensat per a publicar-se en un CD, és freqüent incloure algunes instruccions perquè el contingut es visualitzi a pantalla completa:

```
stage.displayState = StageDisplayState.FULL_SCREEN;
```

Dins de la pantalla completa podem indicar si el contingut s'ha d'ampliar o no s'ha d'escalar, amb una de les línies següents:

```
stage.scaleMode = StageScaleMode.SHOW_ALL;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

## Tutorial 12. Publicació i exportació Pas 9 de 10

A més d'aquestes opcions de publicació que hem vist, també podem exportar el contingut de la nostra pel·lícula a diversos formats.

Seleccionem **Archivo > Exportar > Imagen** per crear una imatge amb el fotograma actual. Podem triar entre diversos formats a més dels que ja apareixen en les opcions de publicació com, per exemple, *Mapa de bits* o *Adobe Illustrator*.

Si seleccionem **Archivo > Exportar > Película** també podem triar entre diversos formats d'exportació.

Per exemple, podem exportar la nostra pel·lícula com una seqüència d'imatges que després podrem importar i modificar amb programes d'edició de vídeo. També podem crear directament un arxiu de vídeo avi o mov (aquest últim només funcionarà correctament si la publicació és per a versions del Flash Player 5 o inferior).

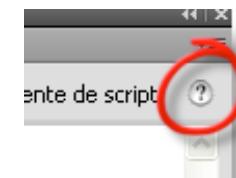
Cal tenir en compte que, llevat del format SWF, cap de les altres opcions d'exportació de pel·lícula no executarà la programació d'ActionScript. Per tant, si tenim alguna animació creada amb codi, aquesta no es visualitzarà en l'arxiu exportat. Tampoc no es podran visualitzar les línies de temps imbricades.

Davant de qualsevol dubte sobre els temes vists en aquest tutorial o en els anteriors, recomanem consultar l'ajuda del Flash.

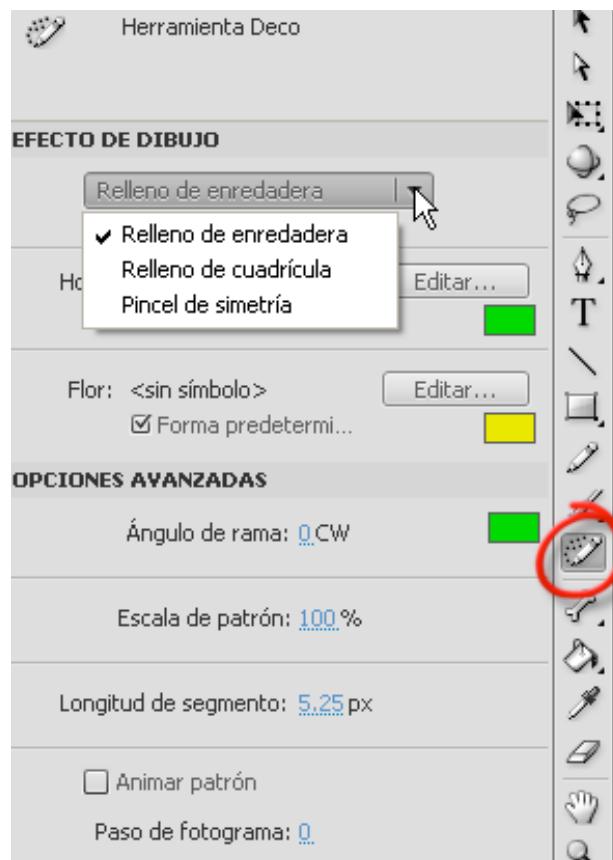
Recordeu que podem accedir a l'ajuda general del Flash des del cercador de la barra de menú si es tracta de dubtes generals sobre l'entorn d'edició



o des de la icona d'ajuda del tauler *Acciones* si es tracta de dubtes sobre programació.



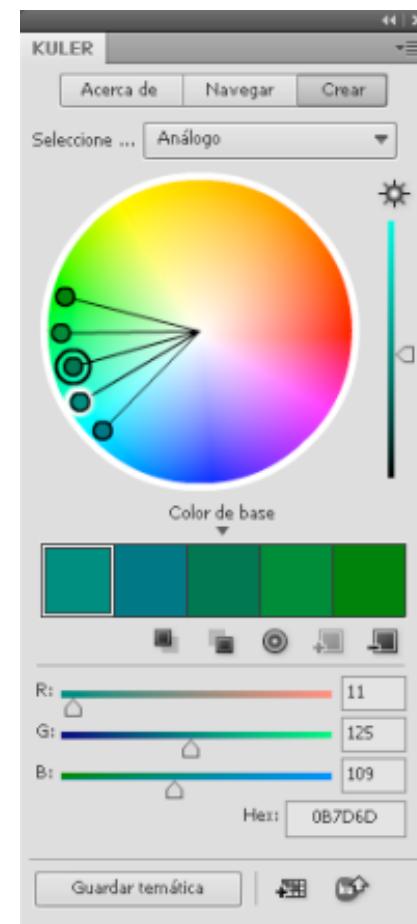
## Tutorial 12. Publicació i exportació Pas 10 de 10



Abans de finalitzar, cal comentar que el Flash disposa d'algunes eines molt útils que no hem vist al llarg d'aquests tutorials i que recomanem provar com, per exemple:

· **L'eina Deco**, que ens permet crear fàcilment farciments complexos a partir de formes predeterminades, o utilitzant símbols que tinguem en la nostra biblioteca.

· **El tauler de kuler**, que podem obrir seleccionant **Ventana > Extensiones > Kuler**, i amb el qual podem accedir a grups de colors o temes creats per una comunitat en línia de dissenyadors. També ens permet crear i desar els nostres propis temes de colors per a incloure'l's fàcilment en els nostres projectes.



## CRÈDITS

**Grau i màster de Multimèdia (GMMD)**

<http://multimedia.uoc.edu/guias>

**Coordinació**

Ferran Giménez Prado

**Autoria**

Raquel García Cabañas

**Data de publicació**

01.09.2009

PID\_00154426