

# Flash CS4 per a l'animació i el grafisme digital

Salvador Linares Mustarós  
Queralt Viladevall Valldeperas

PID\_00215869



# Índex

<b>Introducció.....</b>	<b>5</b>
<b>1. Animació en Flash i fotogrames.....</b>	<b>7</b>
1.1. Introducció .....	7
1.2. Instruccions per crear un programa en Flash estil fenaquistoscopi .....	8
1.3. Activitats .....	17
<b>2. Capes.....</b>	<b>19</b>
2.1. Introducció .....	19
2.2. Instruccions per crear un programa amb dues capes en Flash ....	19
2.3. Activitats .....	22
<b>3. Pel·lícules dins de pel·lícules.....</b>	<b>23</b>
3.1. Introducció .....	23
3.2. Instruccions per crear un clip de pel·lícula .....	23
3.3. Activitats .....	30
<b>4. Botons i camps de text.....</b>	<b>32</b>
4.1. Introducció .....	32
4.2. Un programa amb un botó i els tres tipus de capses de text .....	33
4.3. Activitats .....	38
<b>5. Action.....</b>	<b>39</b>
5.1. Introducció .....	39
5.2. Unes idees d'Action .....	39
5.3. Quan un programa "parla" .....	42
5.4. Activitats .....	43



## **Introducció**

Aquest mòdul va acompanyat d'un total de divuit programes realitzats en Flash.

La seva intenció principal és oferir un contingut teòric autosuficient amb uns primers exemples de treball.

Tenim la confiança que un cop eliminat el temor inicial al programa, haureu adquirit confiança per tal de generar múltiples idees que augmentaran la vostra creativitat fins a límits insospitats, tal com ens heu mostrat any rere any.



# 1. Animació en Flash i fotogrames

## 1.1. Introducció

El Flash està fonamentat en la capacitat humana per percebre moviment aparent a partir de l'observació de seqüències d'imatges estàtiques.

En el segle XIX, aquest efecte es coneixia com la teoria de persistència de la retina. Aquesta teoria, avui considerada un mite, explicava que una imatge projectada encara es visualitzava unes fraccions de segon després que hagués desaparegut la projecció. Aquesta idea permetia explicar per què el cervell podia enllaçar diverses imatges projectades i crear una sola imatge visual, mòbil i continua.

La teoria la va proposar el físic belga Joseph-Antoine Ferdinand Plateau el 1829. Per demostrar la seva teoria, Plateau inventà l'any 1832 el fenaquistoscopi, un dels precursors del cinematògraf.

L'aparell consisteix en una sèrie de dibuixos d'un mateix objecte lleugerament diferents i uniformement disposats en una placa circular. Quan la placa gira a una determinada velocitat, la visió de les imatges a través d'un mirall crea la il·lusió d'una imatge en moviment.

Poc després de la seva invenció, Plateau va estimar que el nombre d'imatges que s'havien de visualitzar per segon si es volia tenir una imatge de moviment òptima era setze. Aquest és el motiu pel qual les primeres pel·lícules van utilitzar setze fotogrames per segon.

Hi ha molts altres aparells que utilitzen l'efecte de percebre moviment aparent a partir de l'observació de seqüències d'imatges estàtiques, com ara el taumatrop, el zoòtrop, el praxinoscopi o el zoopraxiscopi. L'animació en Flash es basa també en aquest efecte, projectant fotogrames a una velocitat determinada, normalment acotada entre 12 i 24 fotogrames per segon.

En aquest primer aspartat aprendrem a crear un petit programa en el qual projectarem onze imatges d'un cavall fotografiat per Muybridge, per tal de produir la sensació de galop.

## 1.2. Instruccions per crear un programa en Flash estil fenaquistoscopi

1) **Primer.** Obtenim els dibuixos o imatges que formaran un cicle repetitiu en format jpg, tots de les mateixes dimensions de la imatge *Muybridge\_horse\_gallop.jpg* (la trobareu a la carpeta *programa1*) i els guardem amb noms que ens permetin conèixer l'ordre dins del cicle.

Figura 1. Imatge original Muybridge.jpg

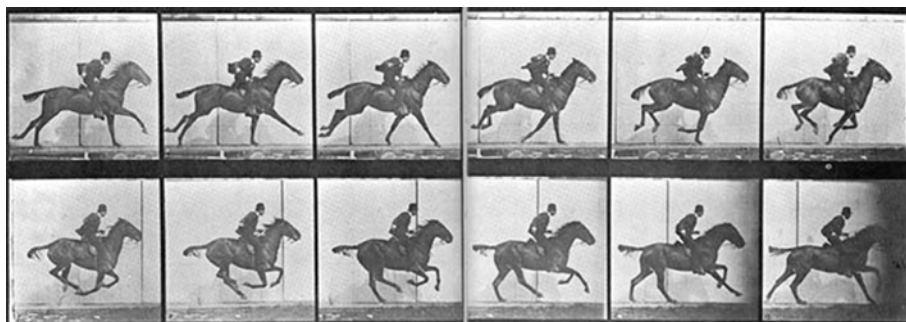
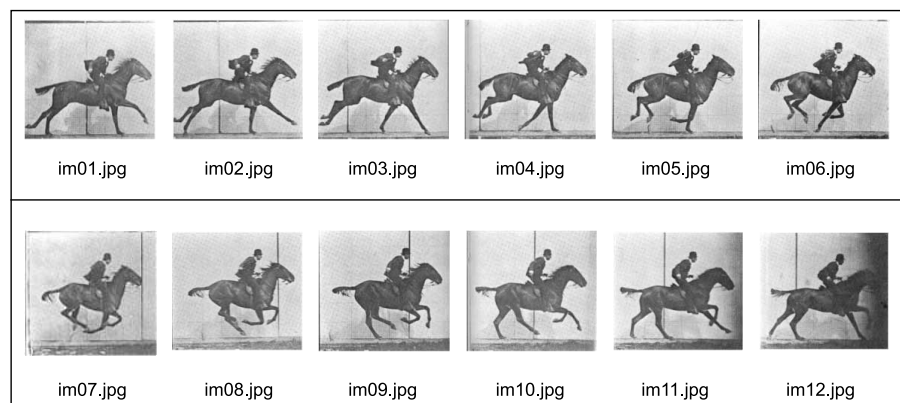


Figura 2. Visualització de les dotze imatges d'igual mida extretes de la imatge *Muybridge.jpg*



Nota: Donat el mal estat de la darrera imatge, en aquest exercici prescindirem de la imatge 12.

2) **Segon.** Obriu el Flash, aneu a *Arxiu-> Nou -> Arxiu de Flash(AS 3.0)*.

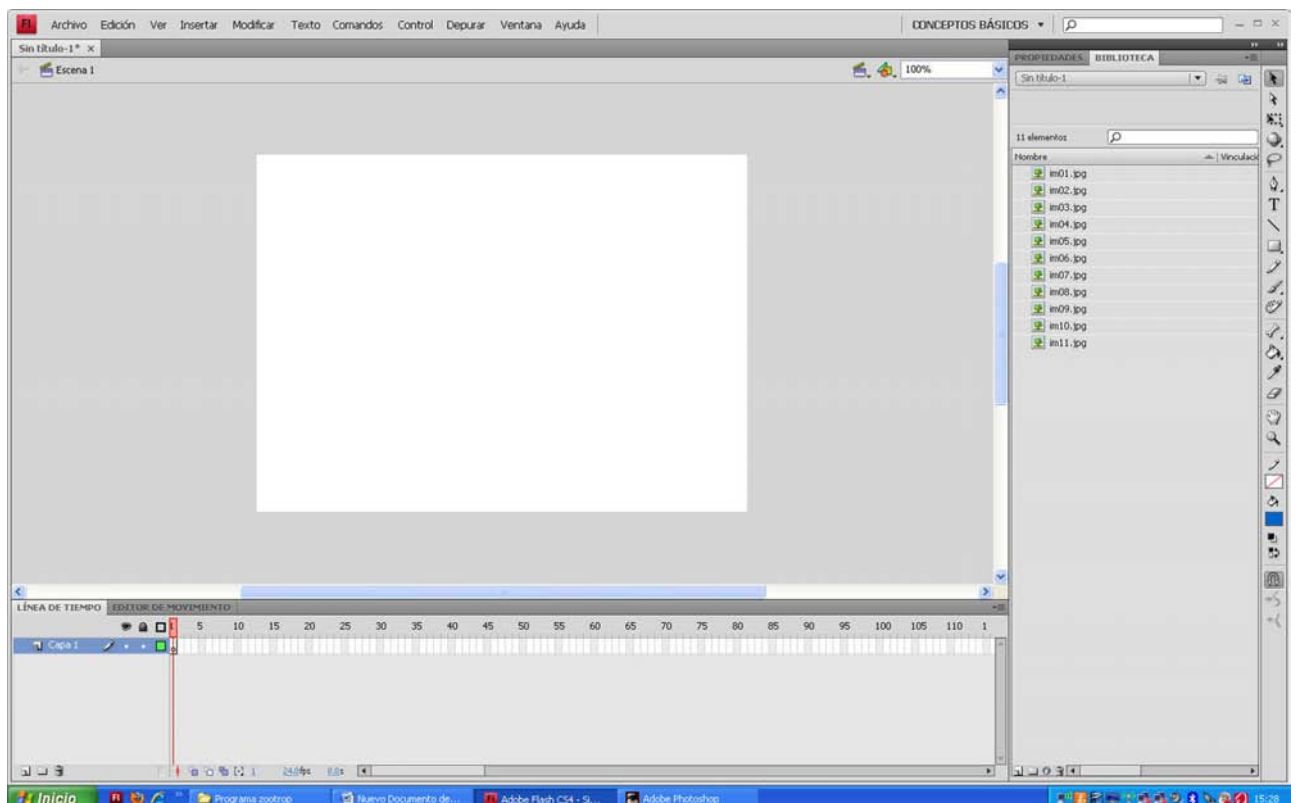
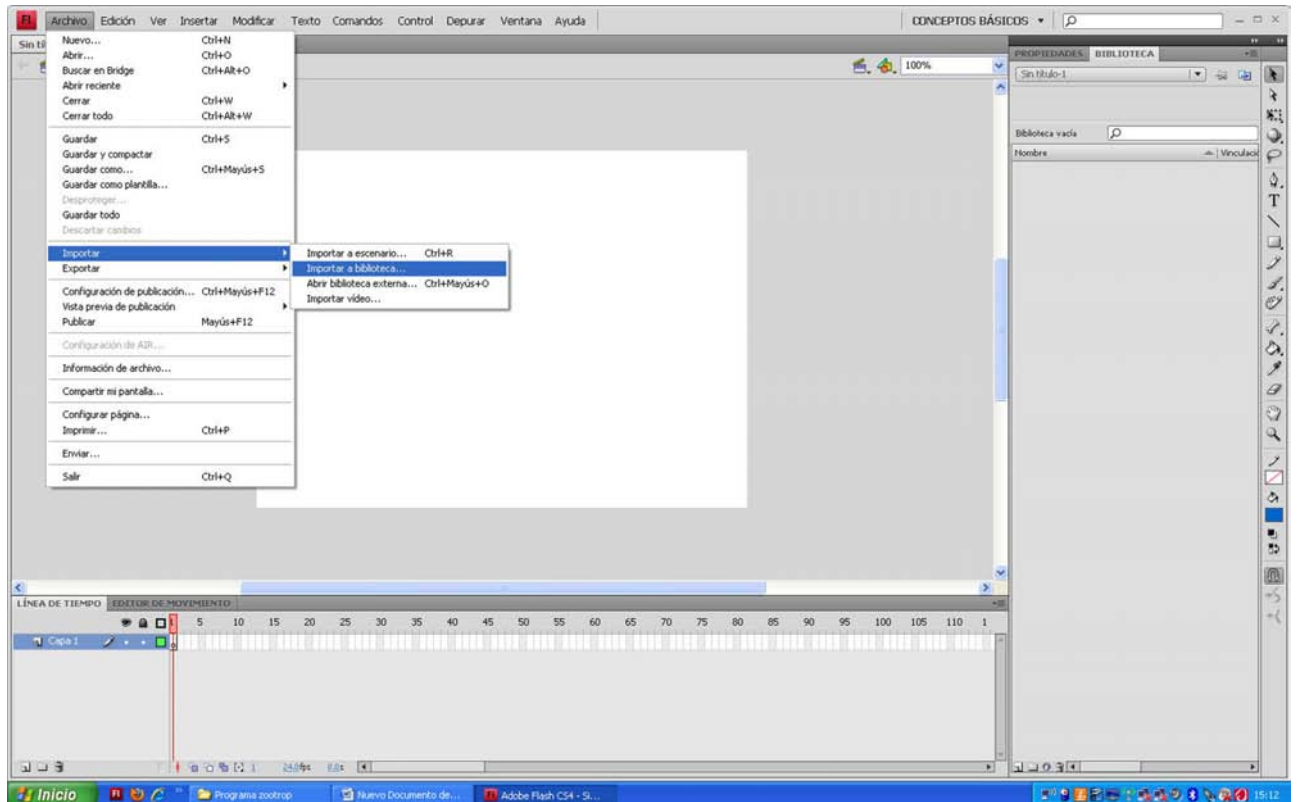
Recordeu de guardar de tant en tant amb el nom que vulgueu. Per exemple, *programa1* o *zootrop*).

3) **Tercer.** Importeu a la biblioteca totes les imatges creades

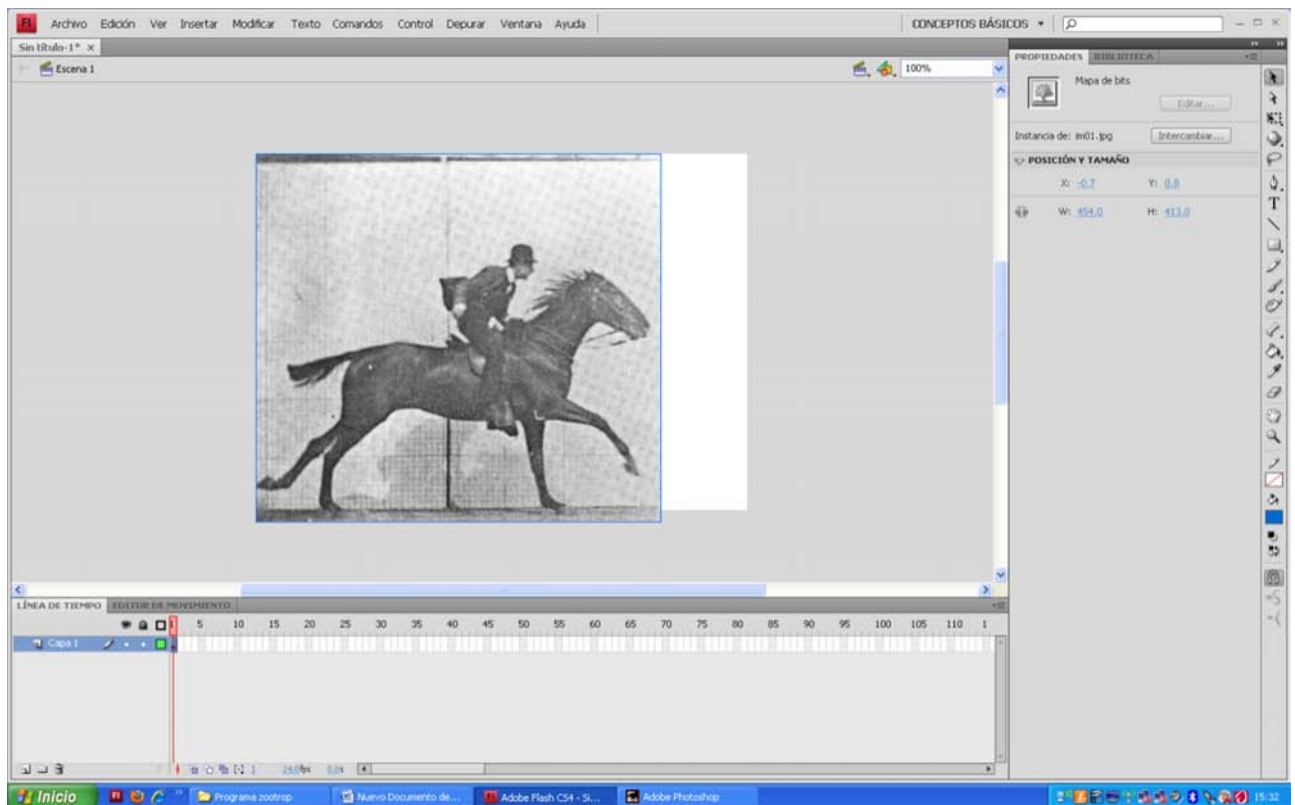
### Nota

Per tal d'obtenir les mateixes dimensions, primer s'ha creat un arxiu .psd en el qual s'han enganxat còpies dels fragments d'imatges i s'han anat movent fins a obtenir una posició semblant. (El .psd està dins la carpeta *programa1*.)



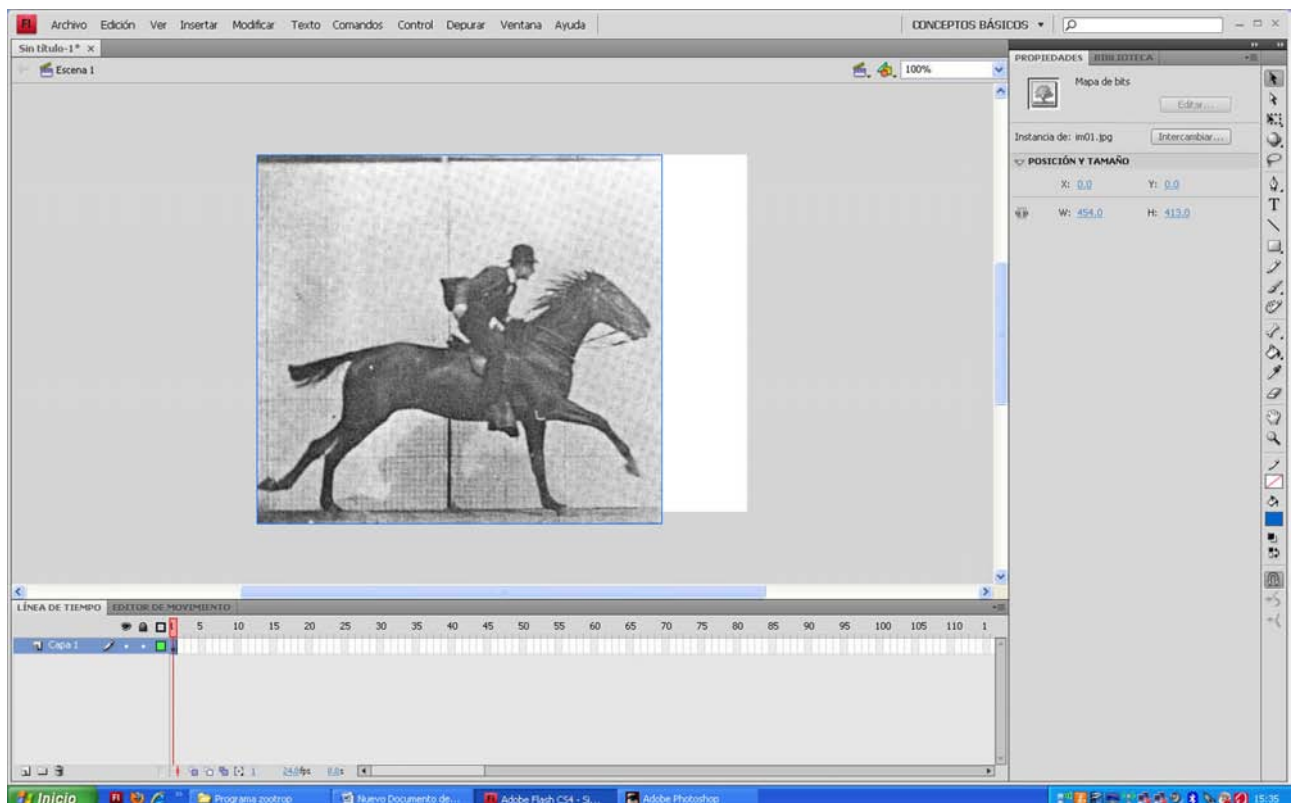
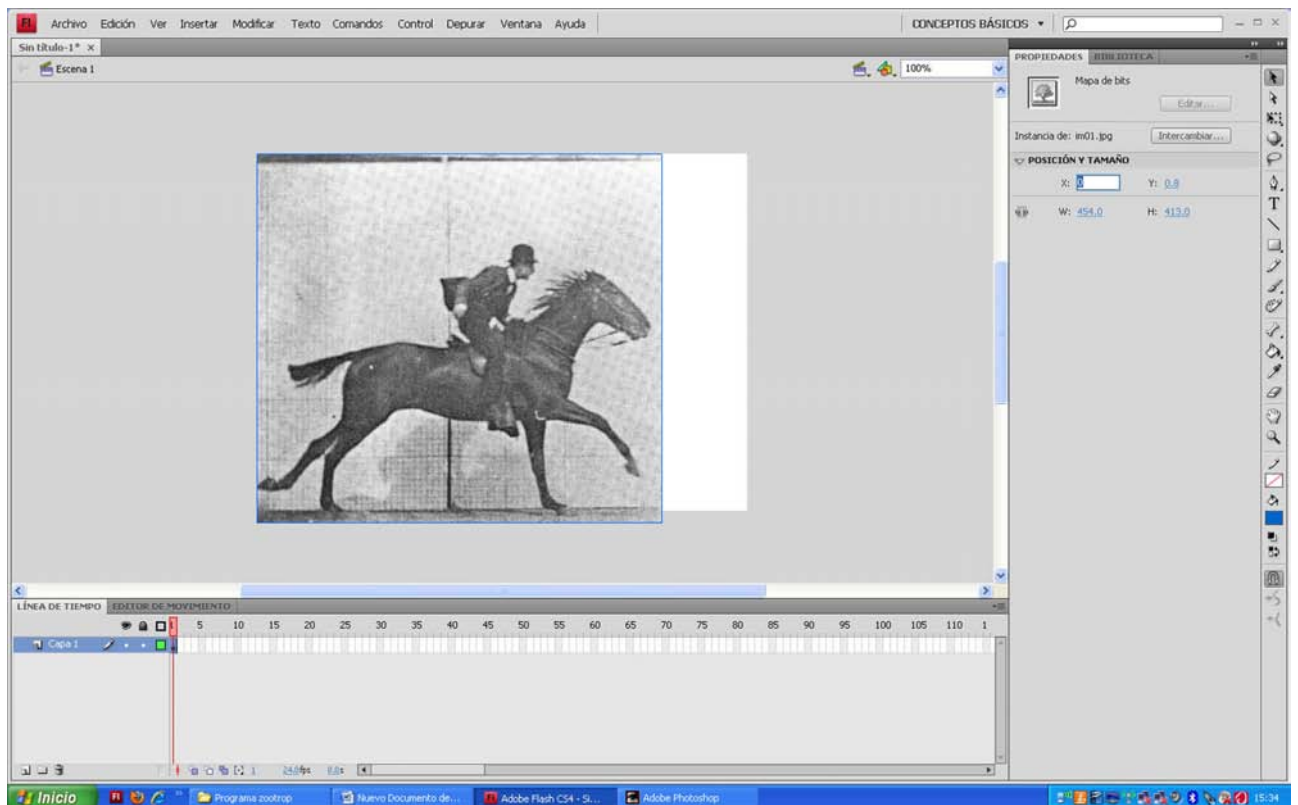


4) **Quart.** Arrossegeu im01.jpg de la biblioteca a l'escenari i feu clic a propietats. Observeu que la meua imatge im01.jpg té una mida de 450 d'amplada per 413 d'alçada.

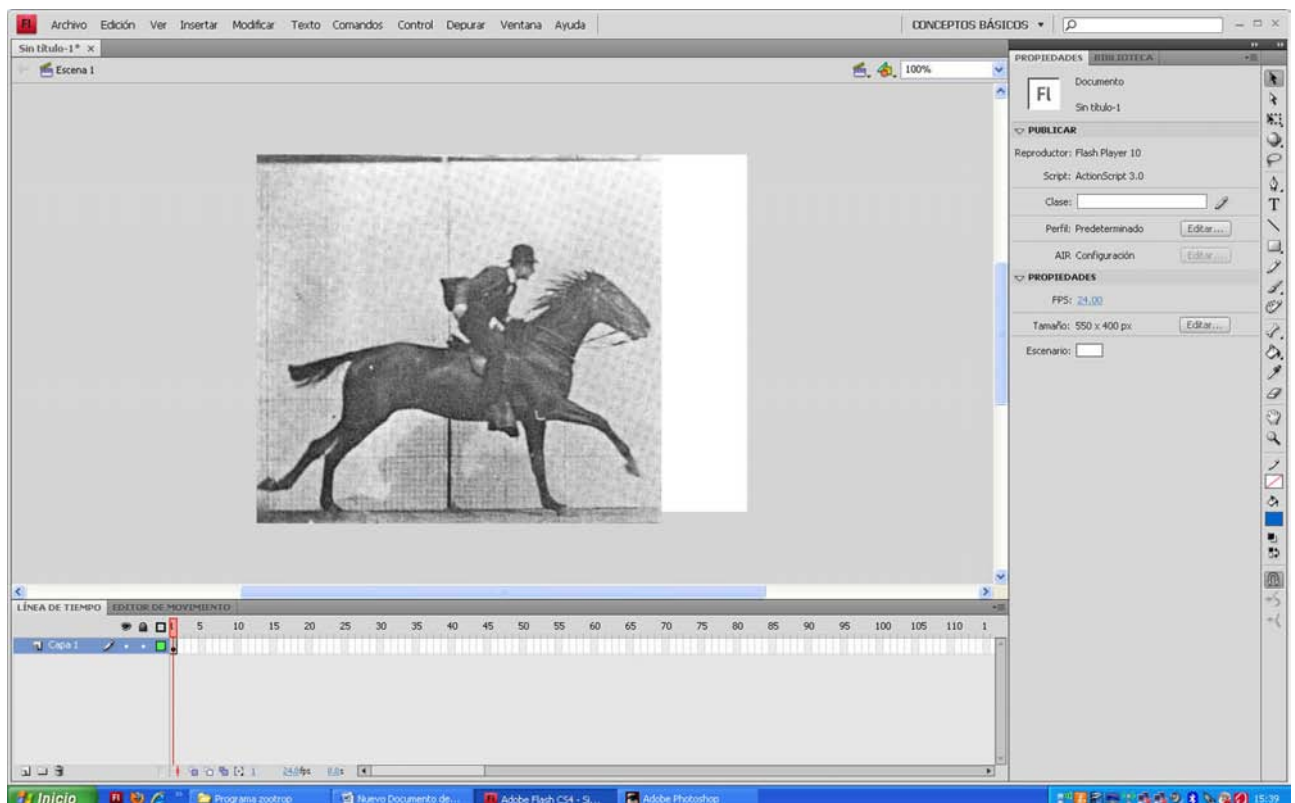
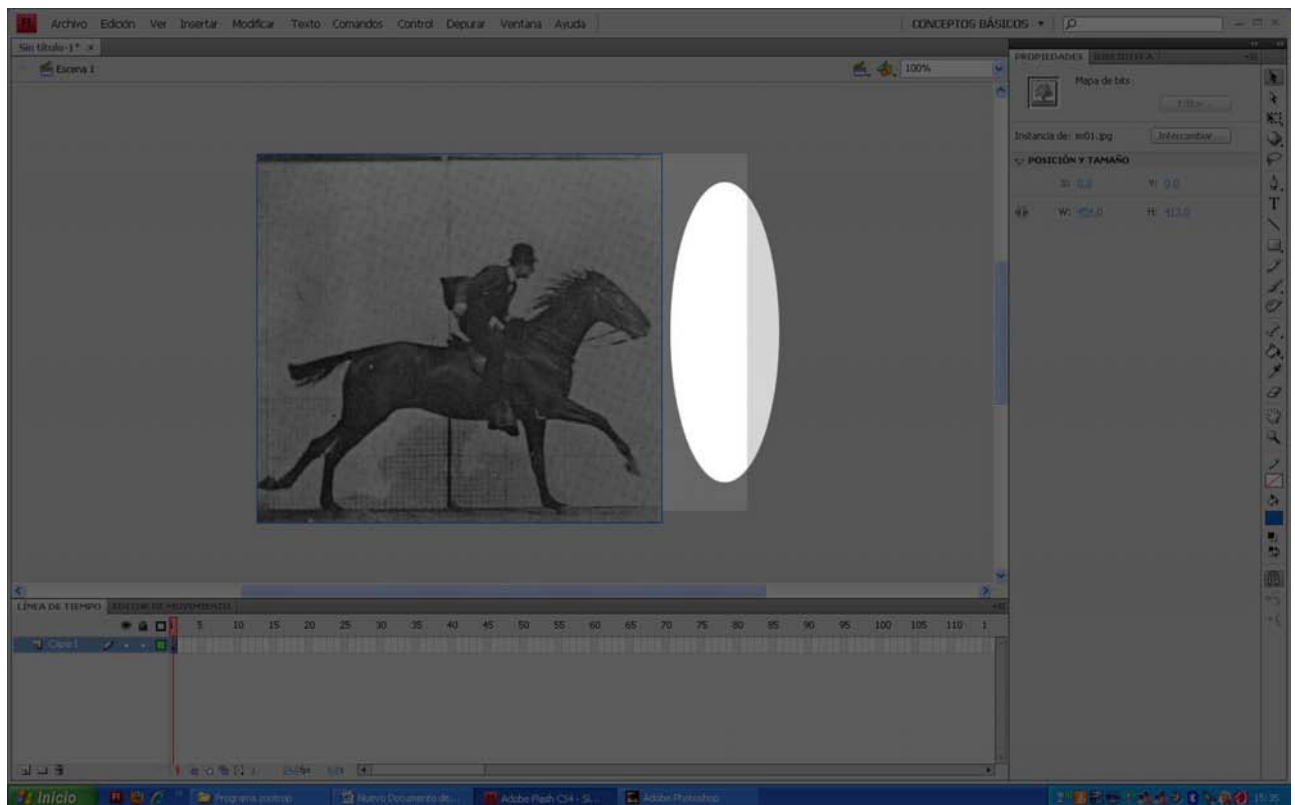


Recordeu que si voleu veure les propietats d'una imatge, heu de seleccionar la imatge fent clic a sobre.

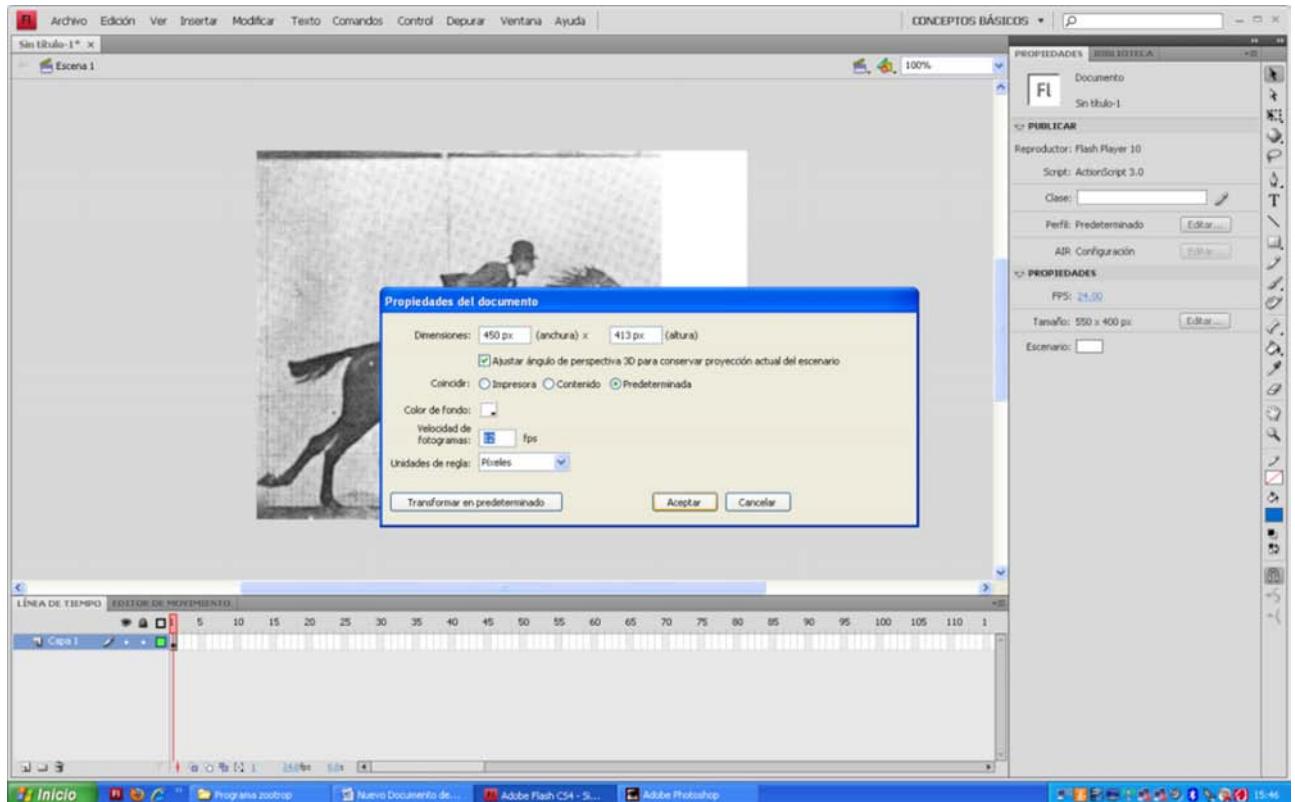
5) Cinquè. Feu clic on diu posició i mida  $x$  i poseu 0 a la  $x$  i 0 a la  $y$ . D'aquesta manera, la imatge estarà ben centrada respecte al nostre escenari. Els punts  $x = 0$  i  $y = 0$  corresponen a la cantonada de dalt a l'esquerra, que és el nostre centre de coordenades.



6) Sisè. Feu clic en qualsevol posició fora de la imatge per tal de veure les propietats de l'escenari.

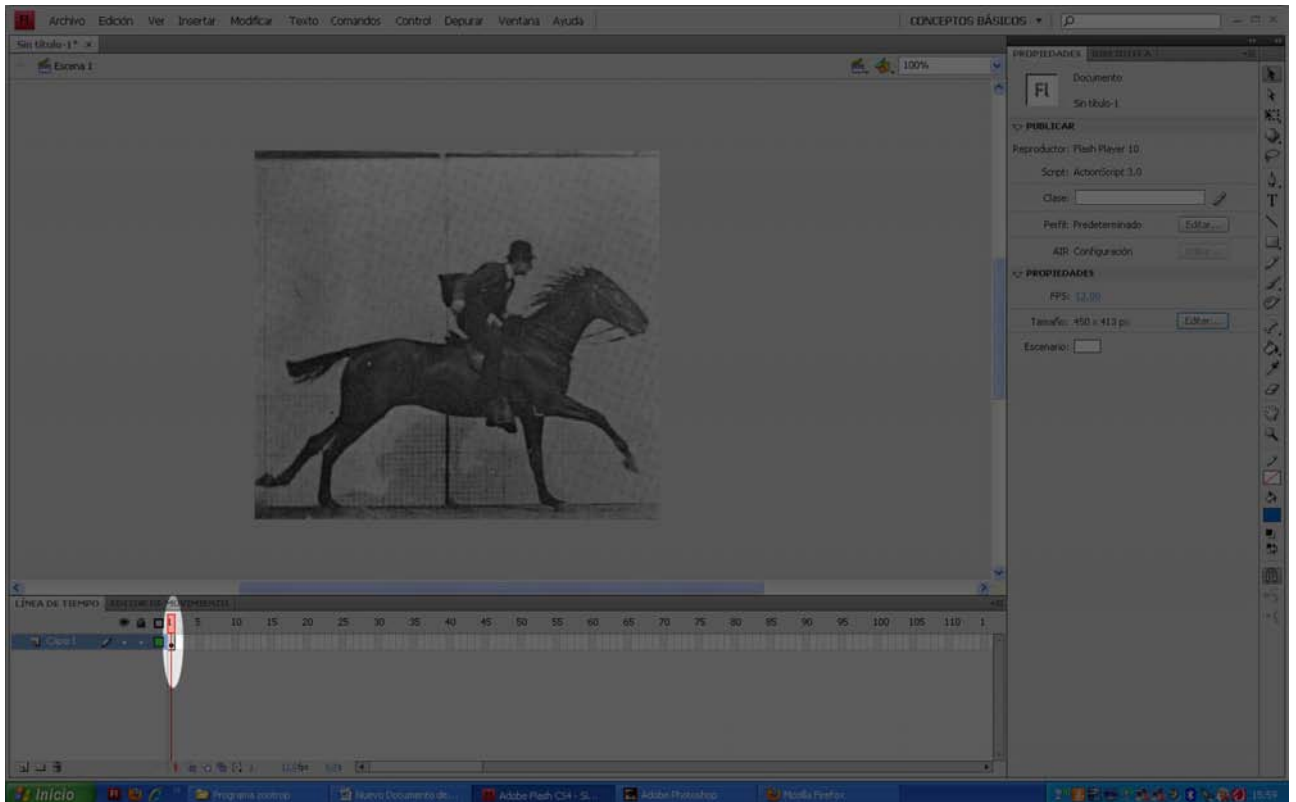


Observareu que la mida de l'escenari és 550 × 400. Si feu clic a *Editar* podeu canviar aquests valors per 450 i 413, i així fareu que l'escenari tingui les mateixes mides que les imatges.

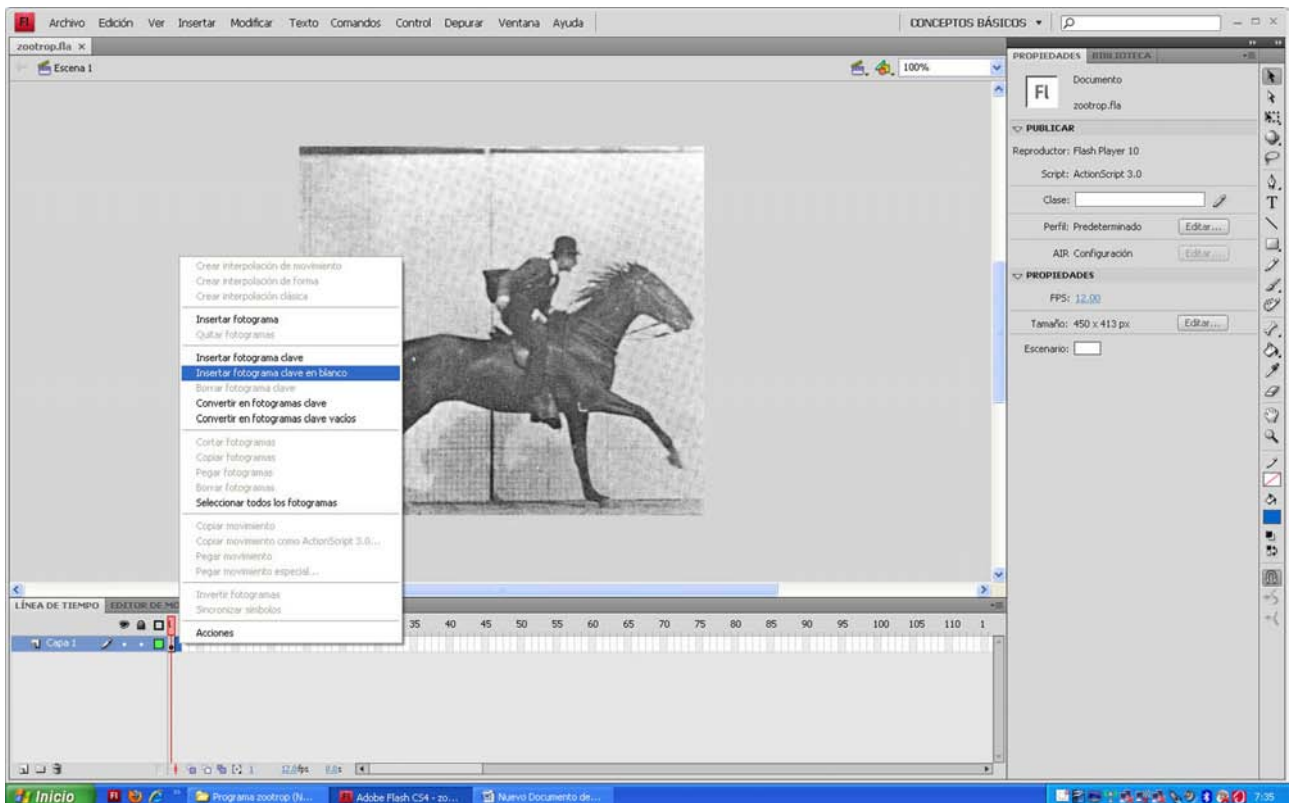


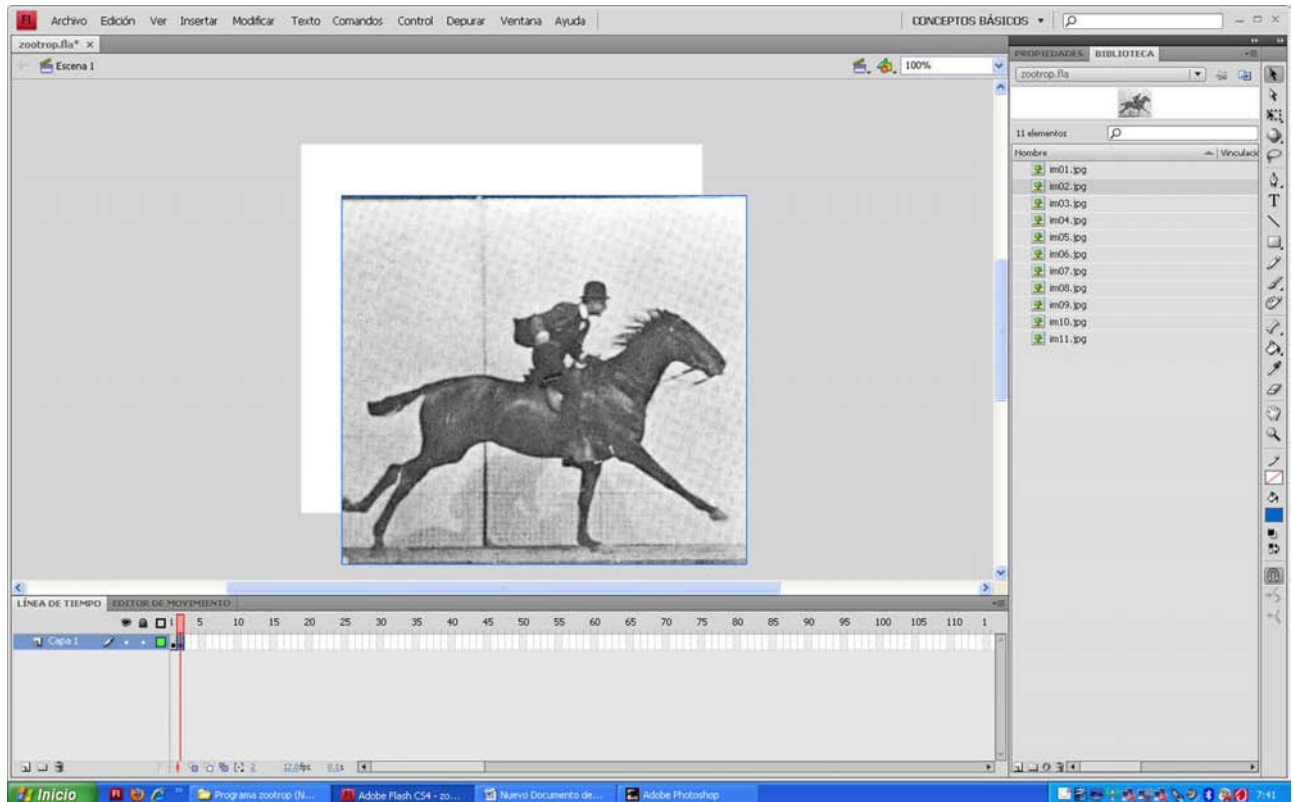
7) Setè. Poseu velocitat de fotogrames a dotze fotogrames per segon (habitualment en web se sol utilitzar 12 fps).

Podreu observar com a la línia del temps el quadrat que s'anomena marc (*frame*) o fotograma ara és de color gris, el programa ens indica així que no està buit.

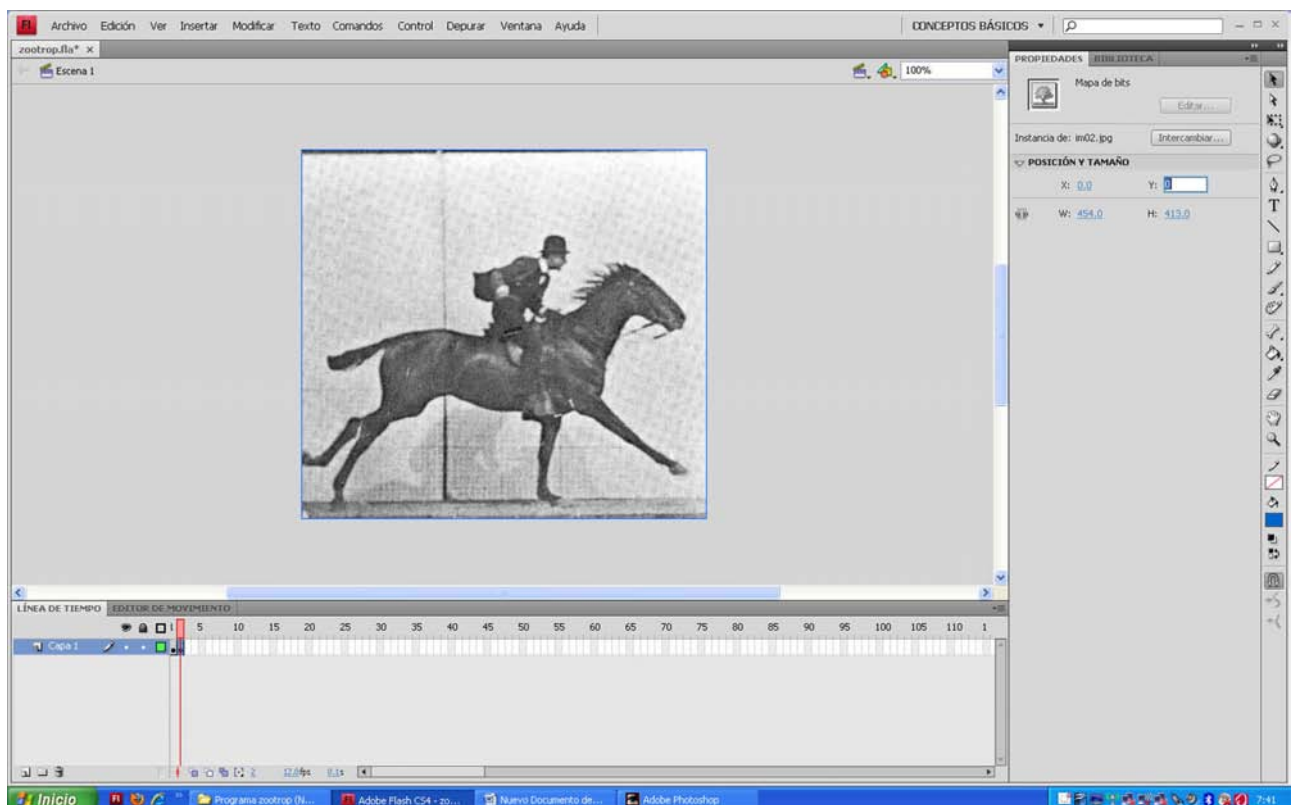


8) Vuitè. Si feu clic amb el botó dret del ratolí en el marc 2, s'obre un menú contextual. Seleccioneu l'opció de crear fotograma en blanc i veureu com apareix un marc nou i l'escenari queda buit. Arrossegueu la imatge de la biblioteca im02.jpg sobre l'escenari.

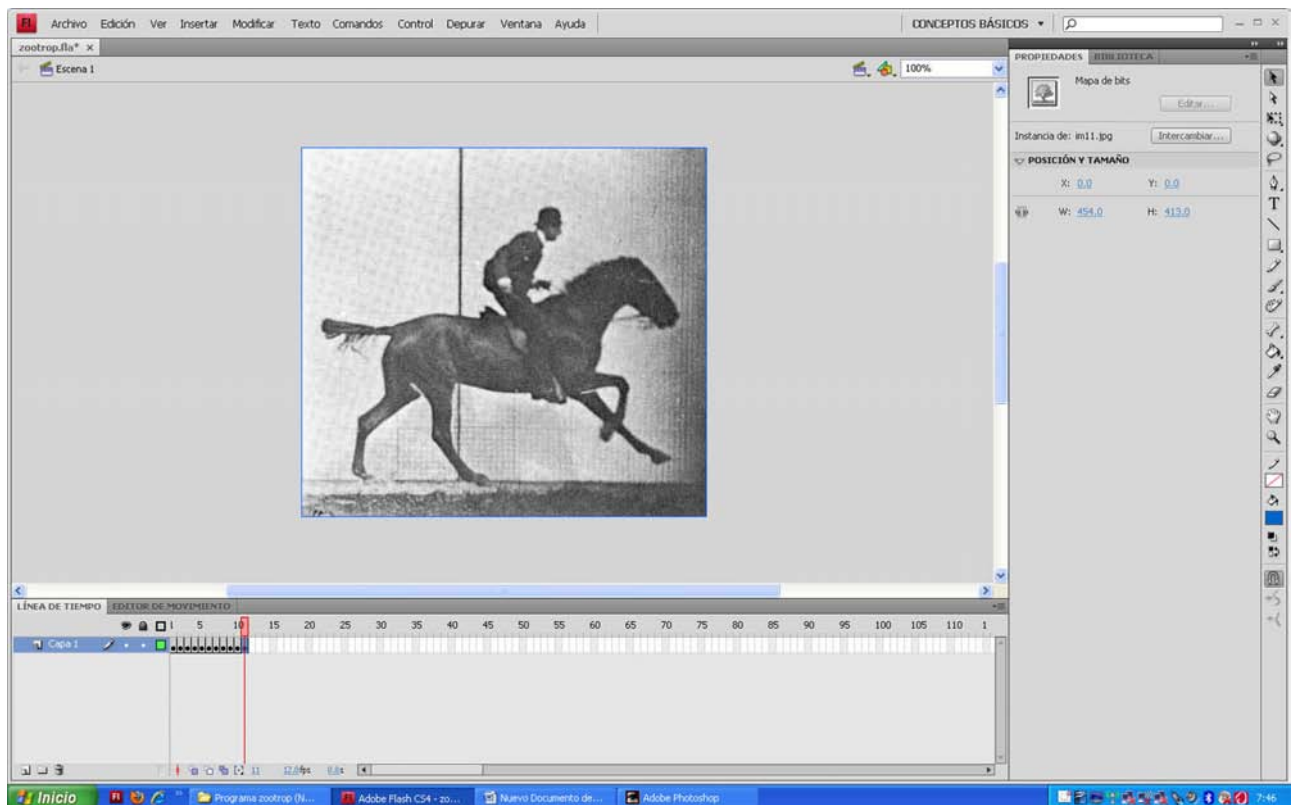




Torneu a posar la  $x$  i la  $y$  de propietats de la imatge a 0 per tenir-ho centrat respecte de la imatge anterior.



9) Novè. Aneu repetint el procés fins a inserir la imatge im11.jpg.



10) Desè. Finalment, només heu de generar el .swf, .exe o .html per veure el resultat.

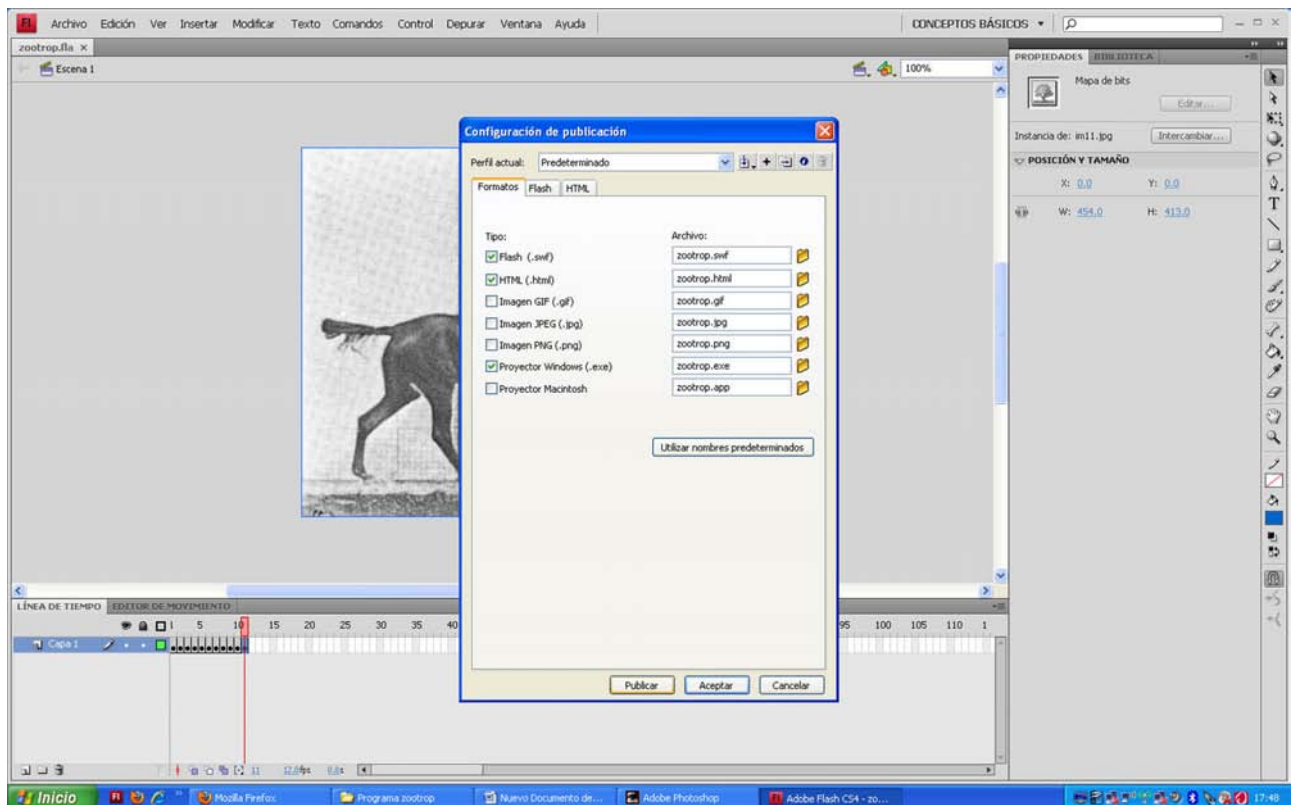
Aneu a *Arxiu* -> *Configuració de publicació* i seleccioneu les opcions .html, .swf i .exe.

Finalment seleccioneu *Publicar* i en el mateix lloc que tingueu l'arxiu .fla obtindreu de regal els tres arxius executables. Feu doble clic a sobre de cada un per veure'n el resultat.

**Nota**

Si teniu un mac trieu .app en comptes de .exe.





### 1.3. Activitats

#### Activitat 1

Cerqueu vídeos a la pàgina <http://www.youtube.com> relacionats amb els aparells presentats en aquest apartat 1. Personalment us recomanem les adreces següents:

<http://www.youtube.com/watch?v=Glk6xRT0Q3k>

<http://www.youtube.com/watch?v=RSF7n9SiTg>

<http://www.youtube.com/watch?v=6rmoAWgn7BI&feature=related>

<http://www.youtube.com/watch?v=V7-GCcJbHbg&feature=related>

#### Activitat 2

Proveu a modificar la velocitat d'fps per veure els efectes sobre la pel·lícula.

#### Activitat 3

Proveu a veure algun tros de la sèrie de dibuixos *Els Picapedra*. Per exemple, el fragment que podem trobar a l'adreça...

<http://www.youtube.com/watch?v=l5HXfp3-bzg&feature=related>

Aquest és un exemple clar del que es coneix com animació limitada. Podeu observar com els personatges parlen movent els ulls i la boca mentre el cos i el fons està estàtic. Aquesta idea permet realitzar molta producció de material en poc temps, amb la qual cosa es pot atendre molta demanda.

#### **Activitat 4**

Creeu un nou programa en Flash. Amb les eines de dibuix creeu en el primer marc una cara amb una boca somrient. Feu clic amb el botó dret del ratolí en el marc 2 i quan surti el menú contextual, seleccioneu l'opció de crear fotograma clau i veureu com apareix un nou marc amb una imatge exactament igual a la del marc anterior. Esborreu la boca i feu-la trista. Jugueu amb la velocitat de fotogrames per tal que quan executeu el programa aparegui un programa tipus taumàtrop.

## 2. Capes

### 2.1. Introducció

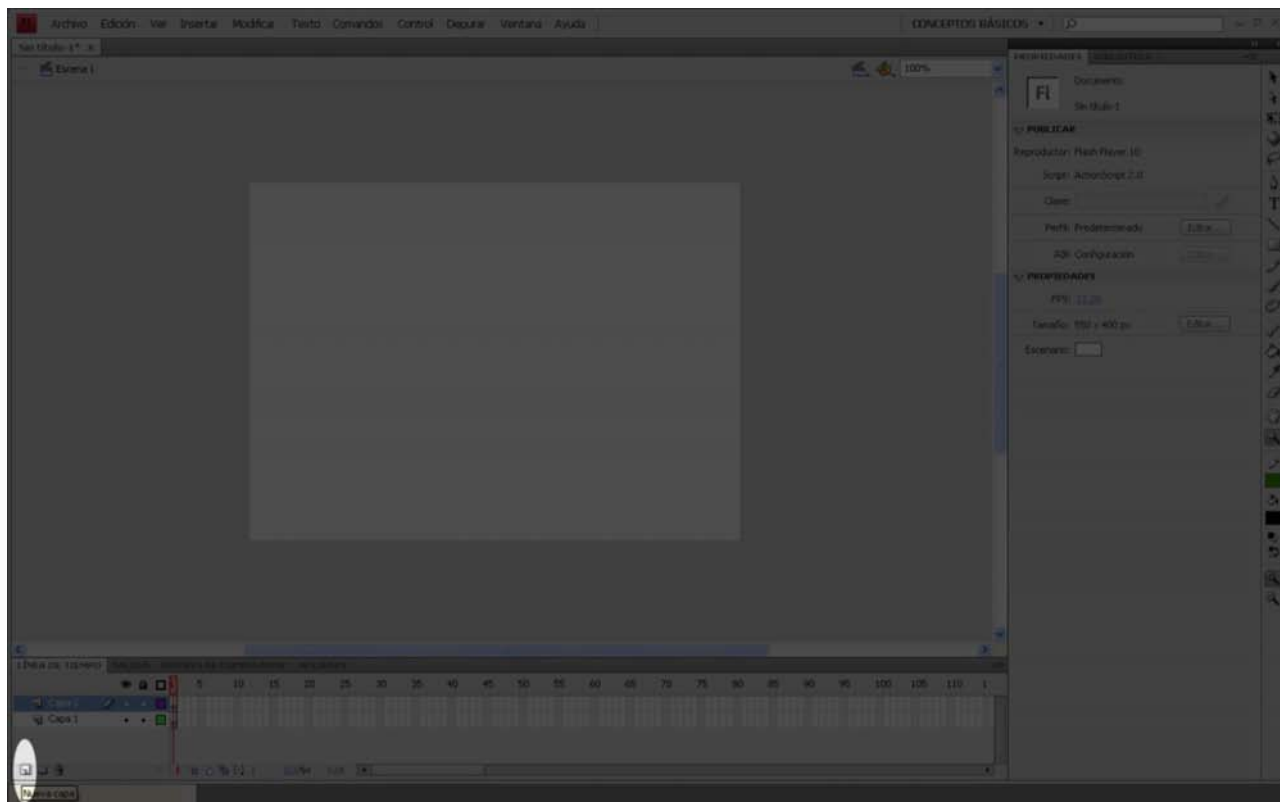
El Flash ofereix una opció molt interessant que consisteix a treballar per capes. Per a aquells que sou nous en el tema de les capes podeu entendre les capes com fulles d'acetat transparent situades una sobre de l'altra en què es poden treballar objectes de manera independent. Això ha facilitat molt el treball, ja que si, per exemple, voleu eliminar o moure de lloc un objecte i no hem treballat en capes, és possible que o no ho puguem fer o bé quedi un espai en blanc, fet que obliga a refer el fons o els objectes de sota, amb la qual cosa es perd temps en cada canvi.

### 2.2. Instruccions per crear un programa amb dues capes en Flash

1) **Primer.** Obriu el Flash, aneu a *Arxiu -> Nou -> Arxiu de Flash(AS 3.0)* i guardeu-lo de tant en tant amb el nom que vulgueu.

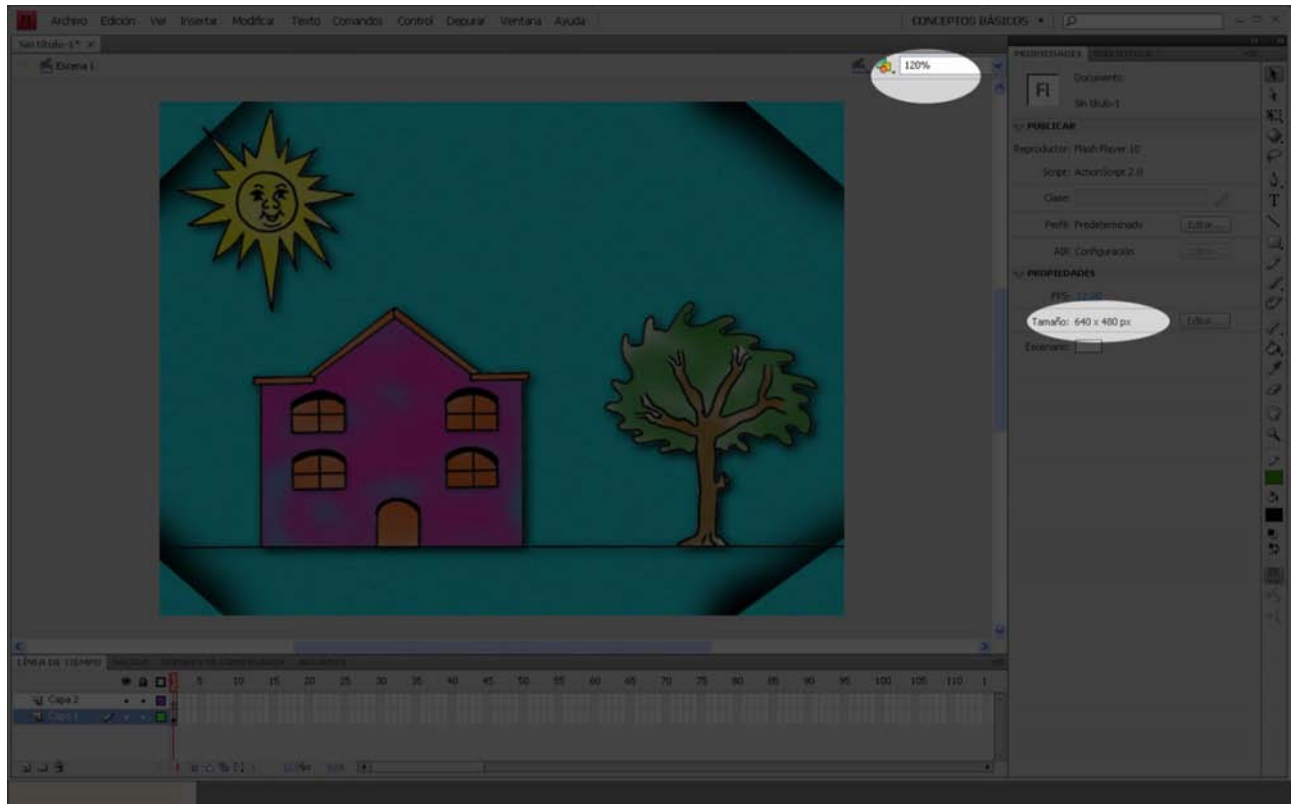
En aquest exemple programa2.

2) **Segon.** Feu clic a *Nova capa* i observeu com es crea una nova capa.



**3) Tercer.** Seleccioneu el primer fotograma de la capa 1 i dibuixeu o importeu algun objecte.

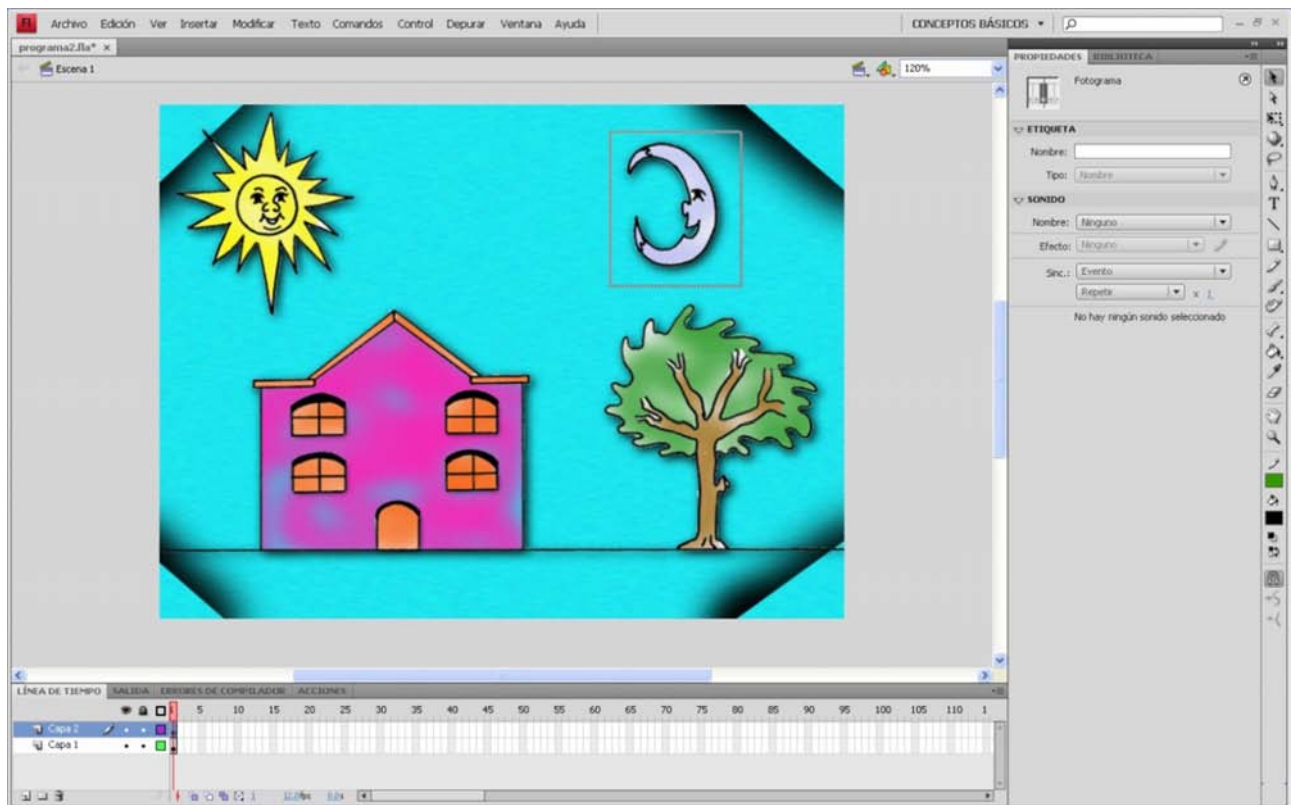
Per exemple, nosaltres importarem la imatge fons.jpg



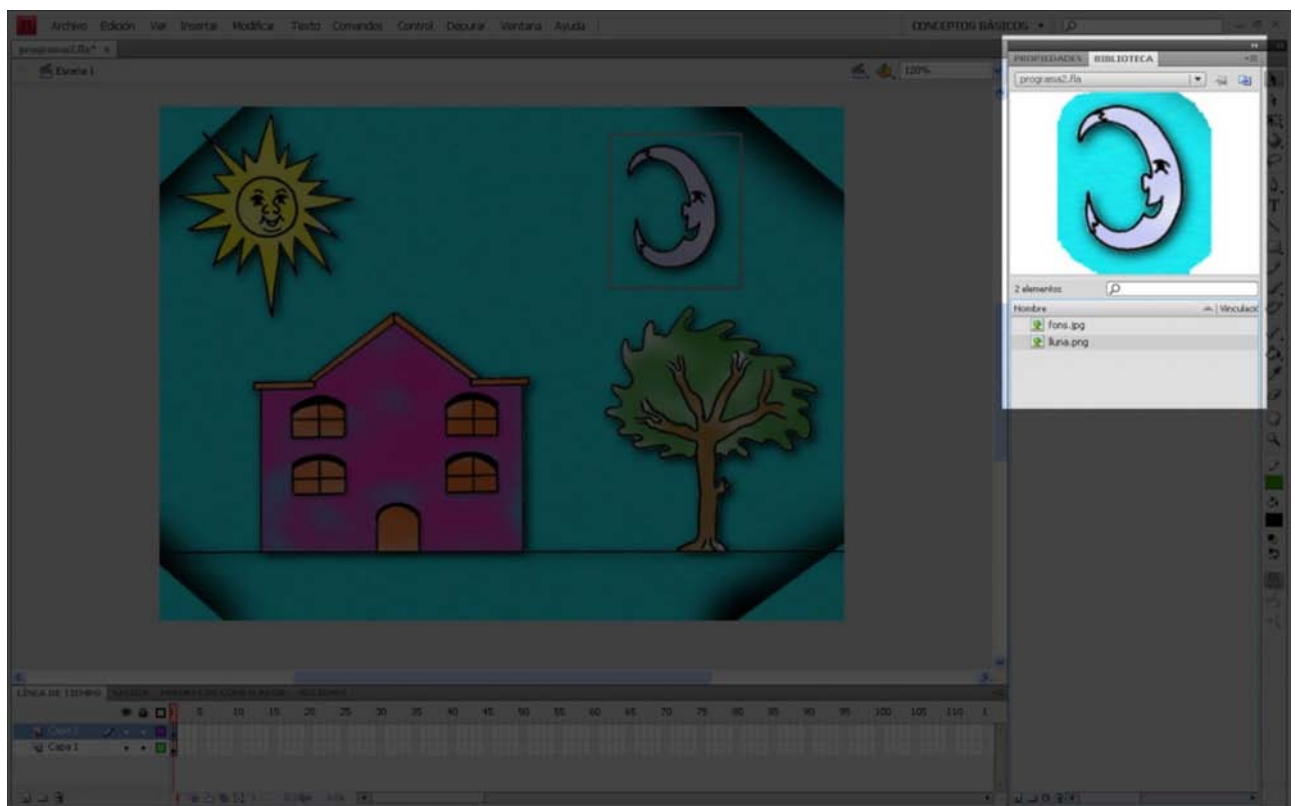
Ajusteu les mides de l'escenari a les mides de la imatge de fons i amplieu si és necessari l'escenari per tenir millor visió del conjunt.

**4) Quart.** Feu clic al fotograma 1 de la capa 2 i feu un segon dibuix o importeu una nova imatge.

Per exemple, nosaltres importarem la imatge lluna.png i la col·locarem dalt a la dreta de l'escenari.



Si feu clic a la pestanya *Biblioteca* veureu les imatges importades al programa.



És molt interessant observar que el Flash respecta les transparències dels formats png.

Finalment, només cal comentar que tot i que podríem haver integrat la imatge lluna en el fons, com ara la casa, el sol o l'arbre, en certes ocasions ens pot ser molt útil tenir-la separada.

Per exemple, si volem fer l'efecte que en passar el ratolí per damunt de la lluna, la lluna desapareix. Treballar en capes pot ser molt útil i us recomanem que useu aquesta tècnica.

### **2.3. Activitats**

#### **Activitat 1**

Proveu a importar a l'escenari una imatge modificada d'una ja importada a l'escenari. Per exemple, pinteu sobre la lluna i guardeu la imatge amb el mateix nom lluna.png i torneu a importar-la a l'escenari del programa 2. Observeu les dues opcions que ofereix el menú Flash quan intenteu fer aquesta opció. Sense guardar els canvis proveu cadascuna de les dues opcions per contrastar els resultats de triar cadascuna de les opcions.

#### **Activitat 2**

Creeu un taumàtrop de manera que hi hagi un fons fix en dos fotogrames i dues imatges lleugerament diferents, una en el fotograma 1 i l'altra en el fotograma 2. Modifiqueu la velocitat d'fps per tal d'obtenir la sensació de moviment de la pel·lícula òptima.

## 3. Pel·lícules dins de pel·lícules

### 3.1. Introducció

El Flash permet crear pel·lícules dins de pel·lícules. S'anomenen clips de pel·lícules. Això ens pot permetre reutilitzar molta feina feta anteriorment. Per exemple, imaginem que hem fet una pel·lícula on una abella (creada com un clip de pel·lícula on l'abella vola amunt i avall) vola sobre un camp de flors. Podem crear un nou programa d'una abella volant sobre un estany només canviant la imatge de fons. També podem copiar i enganxar més clips de l'abella de manera que obtindrem un munt d'abelles volant pel camp de flors o l'estany.

#### Nota

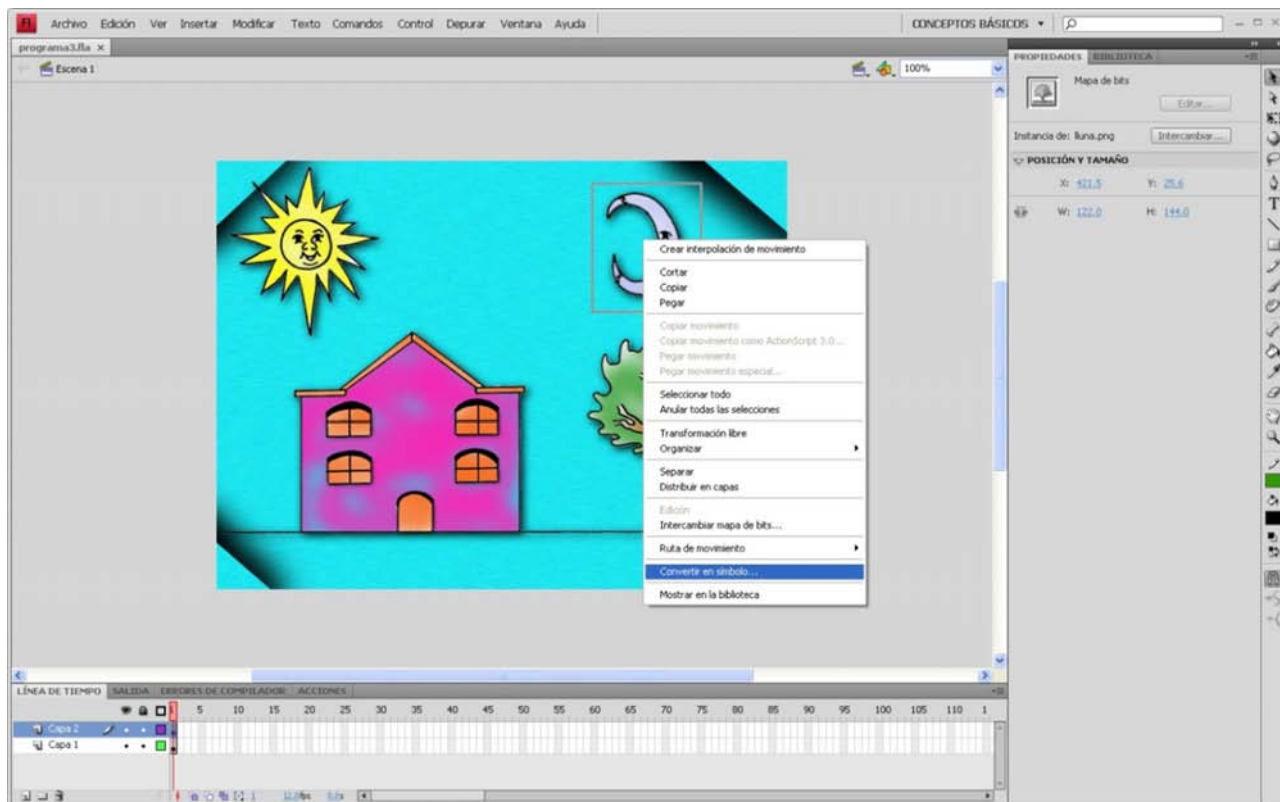
Canviar les mides dels nous clips aconseguirem més sensació de profunditat.

### 3.2. Instruccions per crear un clip de pel·lícula

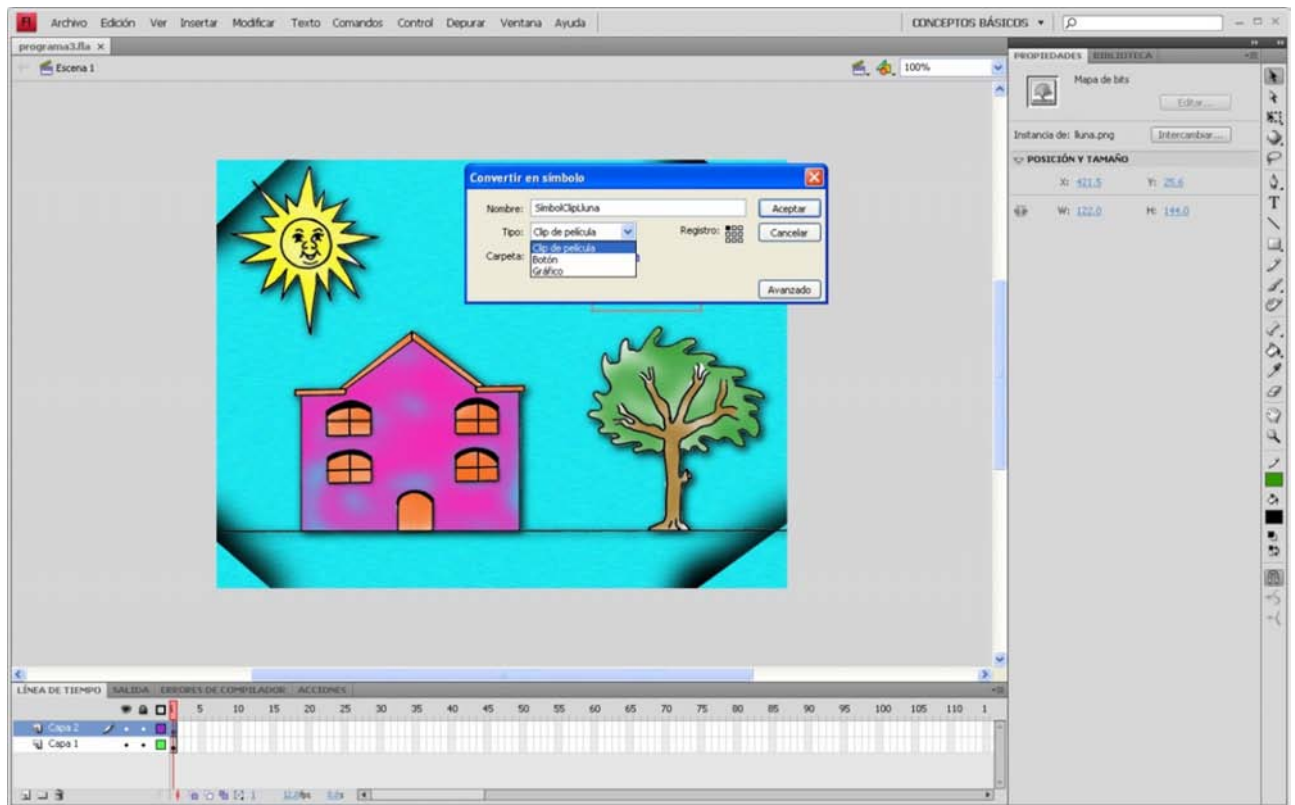
1) **Primer.** Obriu el programa2 i guardeu-lo amb el nom que vulgueu.

Per exemple programa3.

2) **Segon.** Situeu el ratolí sobre la lluna i seleccioneu el botó dret del ratolí. Observeu que apareix un menú contextual. Feu clic sobre l'apartat *Convertir en símbol* per convertir la imatge en símbol clip de pel·lícula.

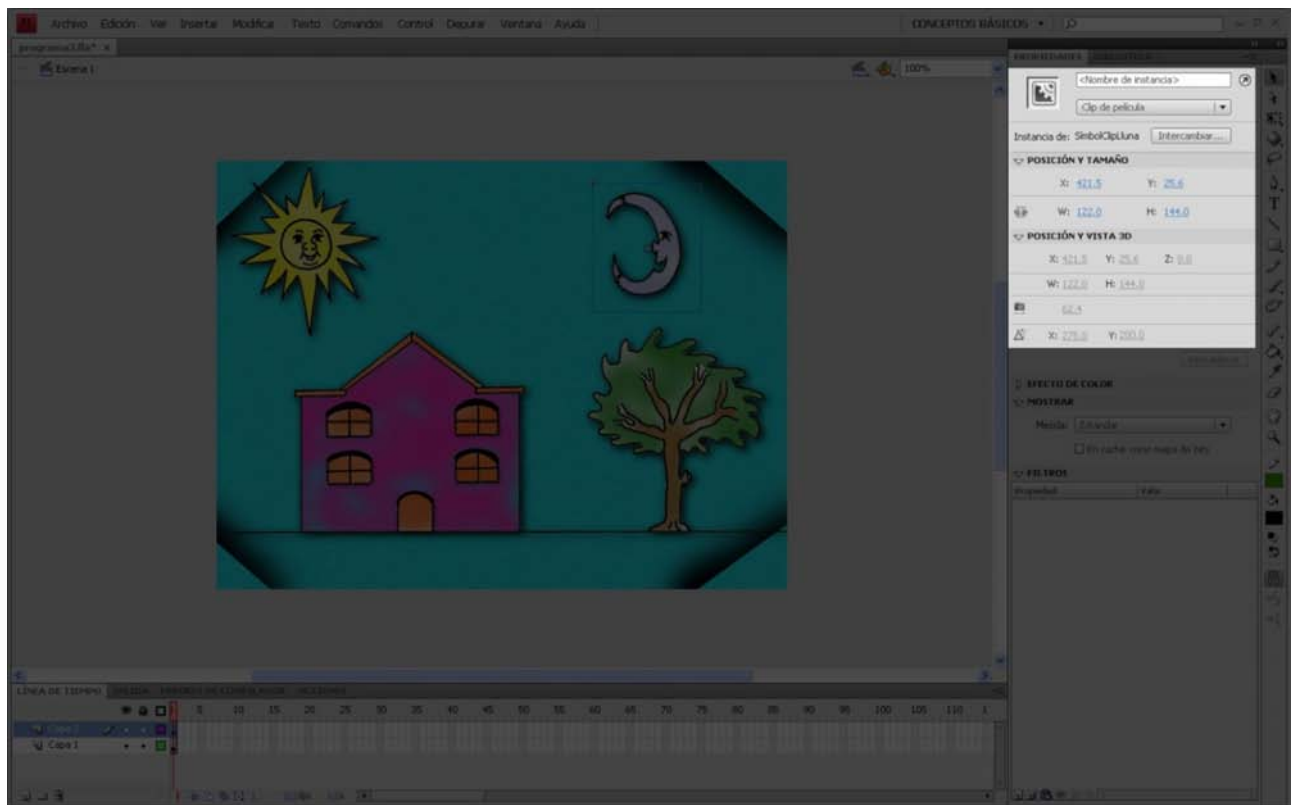


3) **Tercer.** Apareix una finestra de diàleg. Escriviu nom `simbolClipLluna` per tal de tenir un nom que identifiqui la pel·lícula.



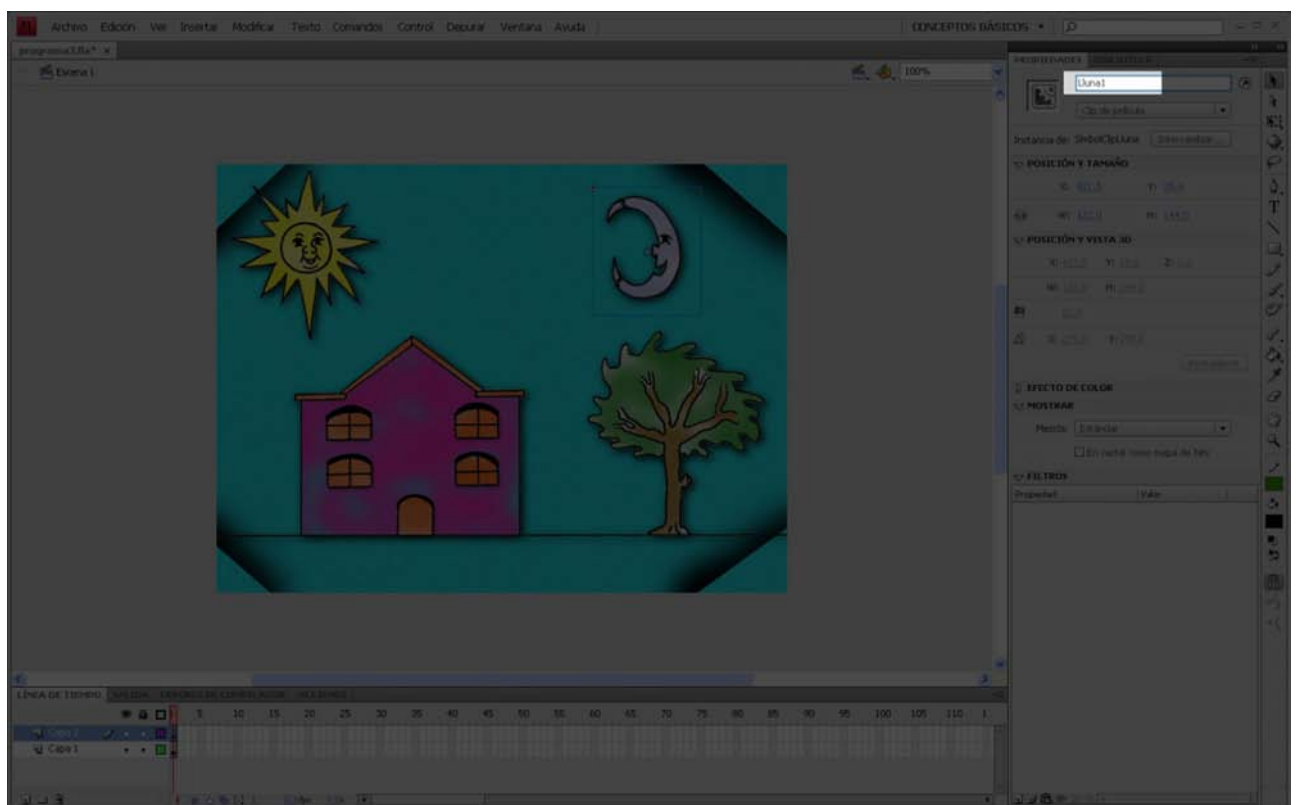
4) **Quart.** Observeu que un cop enllestit el diàleg apareixen a la dreta les propietats d'una pel·lícula *Lluna* que ha substituït la imatge de la lluna.





Observeu que H és l'amplada de la pel·lícula i H, l'alçada (de *width* i *height*).

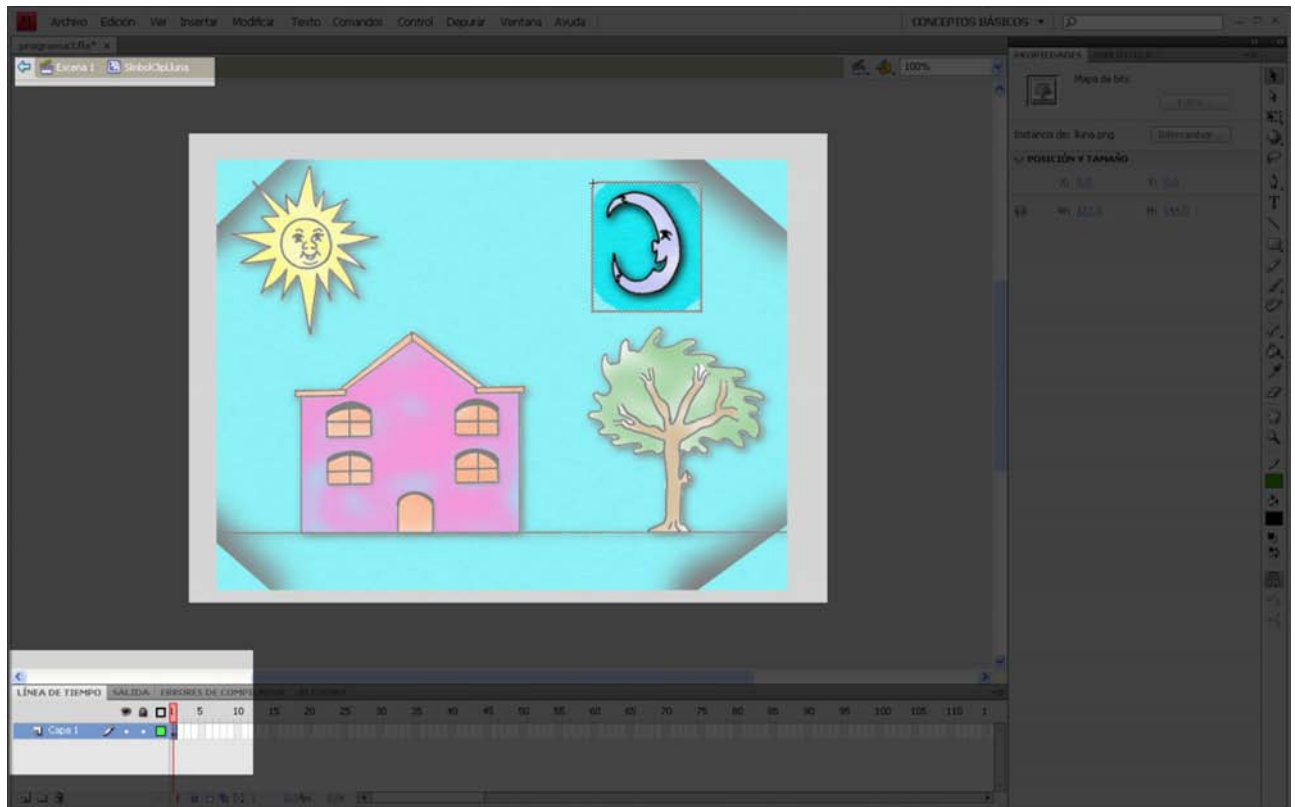
5) Cinquè. Poseu el nom lluna1 al clip de pel·lícula.



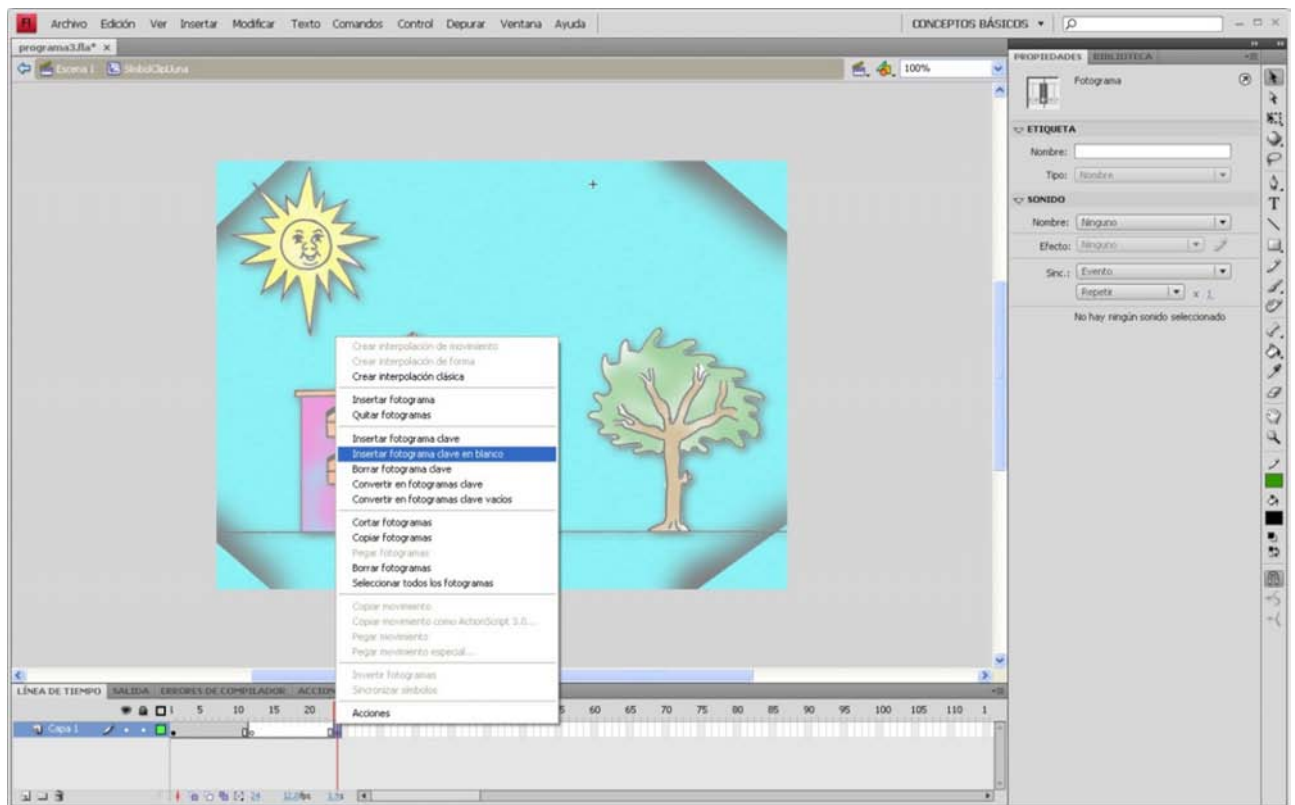
**Nota**

Si copiem i enganxem aquest clip de pel·lícula anem afegint còpies del símbolClipLluna a l'escenari. Recordeu d'anar canviant el nom a cada pel·lícula copiada per tal de tenir ben referenciats tots els objectes que apareixen en l'escenari.

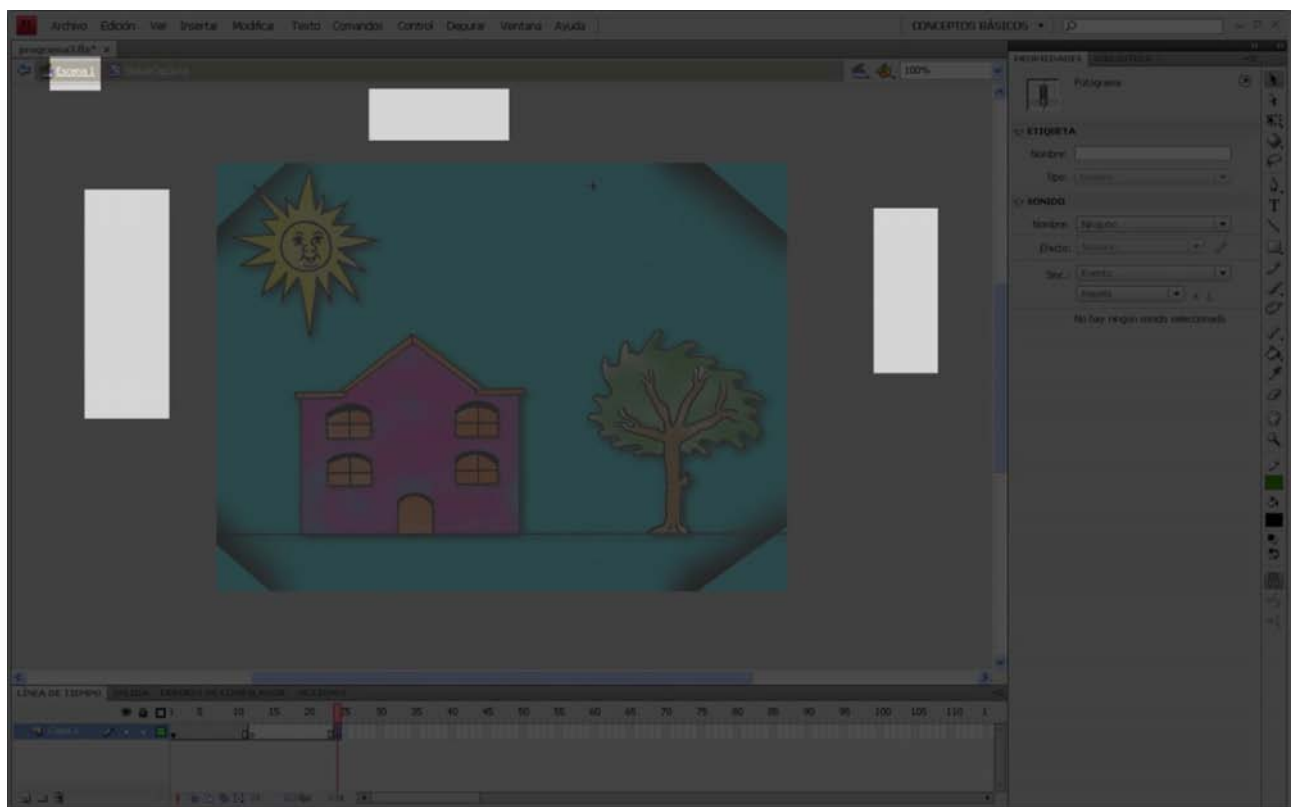
6) **Sisè.** Feu doble clic sobre *lluna1*. Observeu que ara sou a l'escenari símbol-ClipLluna. La part de l'escenari que no forma part de la lluna apareix amb una transparència per diferenciar què és del clip símbolClipLluna del que no ho és. Observeu també com la línia de temps és completament diferent de la pel·lícula original, ja que ara tenim només una capa i no dos.



7) **Setè.** Inserir en el fotograma 13 un fotograma clau en blanc i en el 24, un altre fotograma clau en blanc (recordeu... botó dret del ratolí).



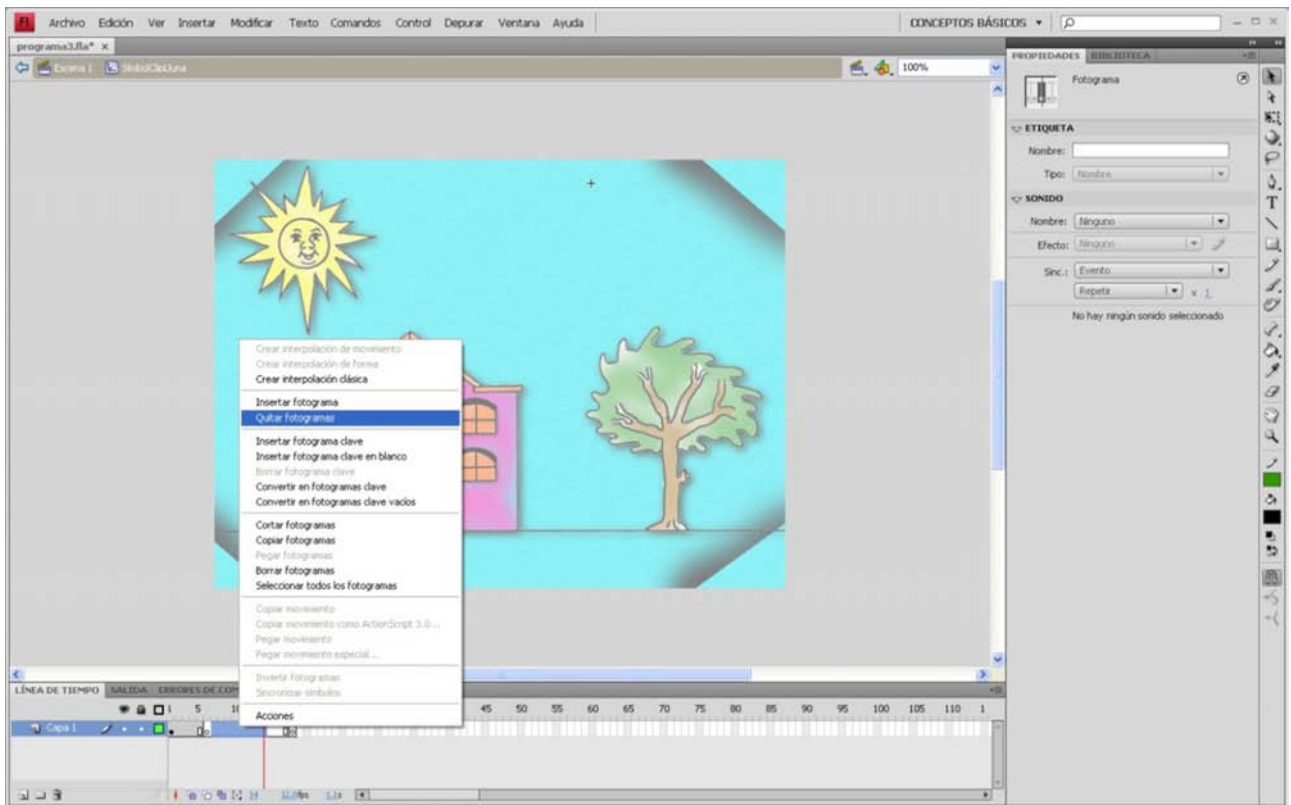
8) Vuitè. Feu clic sobre el text *Escena 1* o doble clic sobre alguna banda fora de l'escenari per tornar a la pel·lícula principal.



Observeu que la pel·lícula principal només té un sol marc, en comptes de la pel·lícula Lluna, que en té 24.

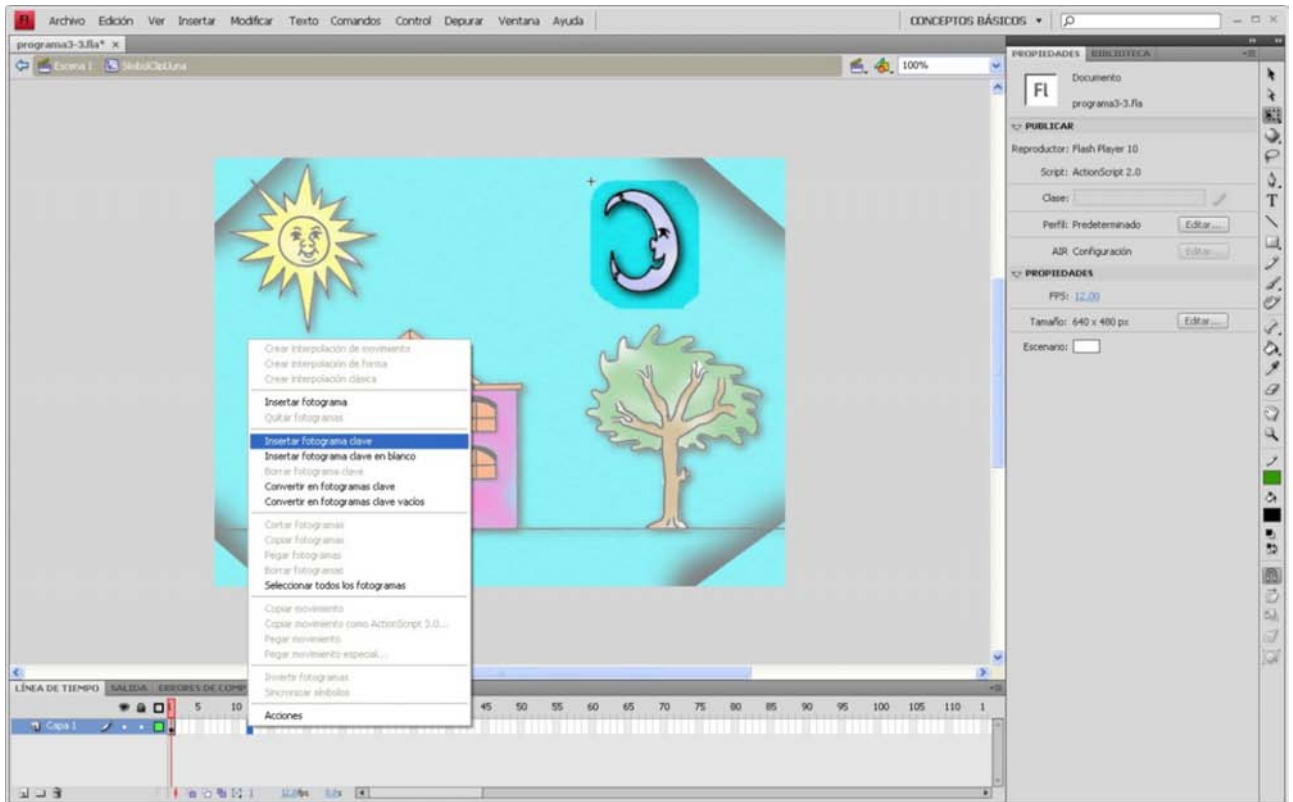
9) Novè. Si executeu el programa veureu un efecte de pampalluga en la Lluna.

Si trobeu que és massa lent, podeu eliminar fotogrames de la pel·lícula SimbolClipLluna seleccionant els fotogrames i seleccionant l'opció *Treure fotogrames* del menú del botó dret del ratolí.

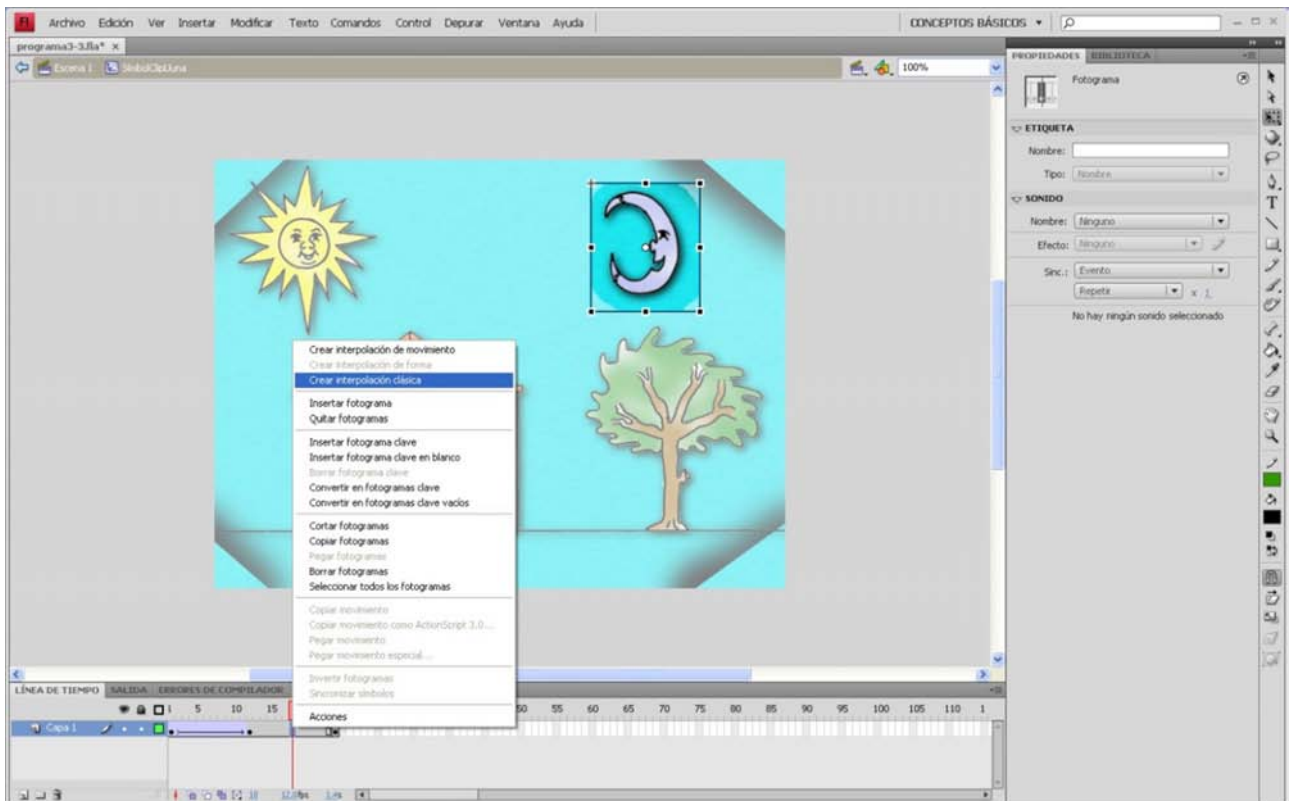


Personalment, el pampallugueig és un efecte molt molest. Tot i que compleix perfectament l'objectiu de centrar l'atenció en l'objecte que fa pampallugues, hi ha altres formes de centrar l'atenció en els objectes sense ofendre tant la percepció. A tall d'exemple, per veure altres efectes per centrar l'atenció en un objecte a continuació us mostrem l'efecte d'engrandir o disminuir l'objecte cíclicament.

Torneu a obrir el clip simbolClipLluna i elimineu tots els fotogrames menys el primer. Seleccioneu el fotograma 12 i seleccioneu l'opció *Insereix fotograma clau*.

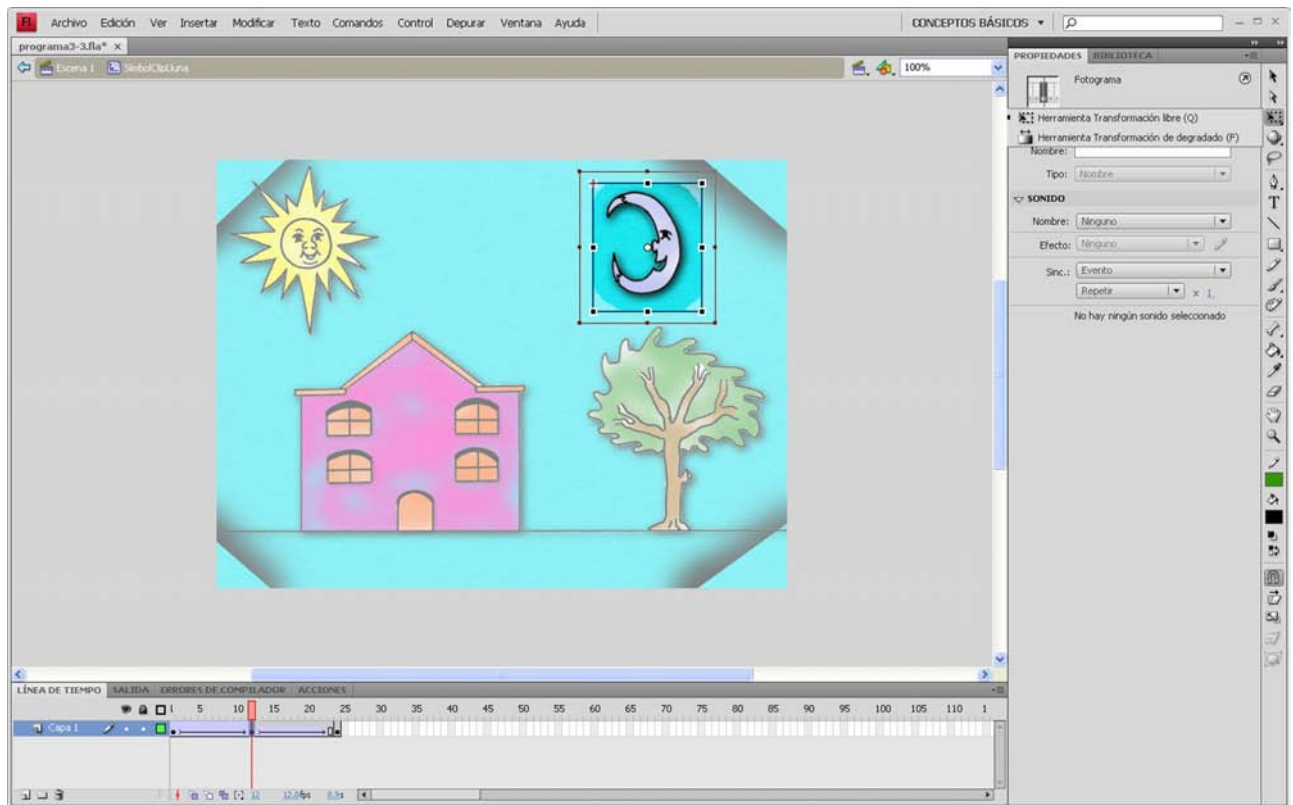


Torneu a seleccionar el fotograma 24 i torneu a seleccionar l'opció *Insertar fotograma clau*. Poseu-vos en un el fotograma 5 o 6 i seleccioneu l'opció *Crea interpolació clàssica*. Poseu-vos en el fotograma 15 o 16 i torneu a seleccionar l'opció *Crea interpolació clàssica*.



Observeu que apareixen unes fletxes als fotogrames que ens indiquen que hi ha una interpolació.

Situeu-vos al fotograma 13 i amb l'eina de transformació lliure disminuïu una mica la mida de la lluna.



Executeu el programa per veure l'efecte creat.

Segurament us heu adonat, amb aquest darrer exemple, de la importància que tenen les transparències completes al voltant dels objectes, ja que sinó el fons de l'objecte pot provocar efectes que enlletgeixen la pel·lícula.

### 3.3. Activitats

#### Activitat 1

Copieu i enganxeu a l'escenari del programa 3 el clip de nom Lluna1. Canvieu el nom del clip copiat pel de Lluna2 i observeu l'efecte de pampallugueig tot a l'hora.

Feu doble clic al clip de pel·lícula Lluna1 per poder modificar el clip. Feu algun canvi en el clip (per exemple, disminuir el temps de pampallugueig). Observeu que els canvis s'apliquen a totes les llunes i no sols a la lluna que heu modificat. Això es deu al fet que en realitat vosaltres heu modificat el clipSimbolLluna i no el clip amb nom Lluna1. Sortiu sense guardar els canvis.

## Activitat 2

Copieu i enganxeu a l'escenari del programa 3 el clip amb nom Lluna1. Canvieu el nom del clip copiat pel de Lluna 2 i observeu l'efecte de pampallugueig tots a l'hora.

Feu un sol clic en el clip de pel·lícula Lluna1 de l'escanari per poder modificar les propietats del clip (per exemple, canvieu les propietats H i W). Observeu com el canvi s'aplica només al clip de pel·lícula Lluna1. Això es deu al fet que ara vosaltres heu modificat el clip Lluna1 i no el clip SimbolLluna, per tant els canvis només s'apliquen al clip de pel·lícula seleccionat.

## Activitat 3

Creeu un camp de flors amb tot de flors que es mouen amb el vent seguint les idees d'aquest apartat. Cada flor ha de ser una instància d'un clip de pel·lícula simbolClipFlor. Modifiqueu les mides de les flors.

## Activitat 4

Sobre el camp de flors anterior dibuixeu un sol en una capa nova. Convertiu-lo en un símbol clip de pel·lícula. Amb la interpolació de moviment mireu que faci un recorregut pel cel.

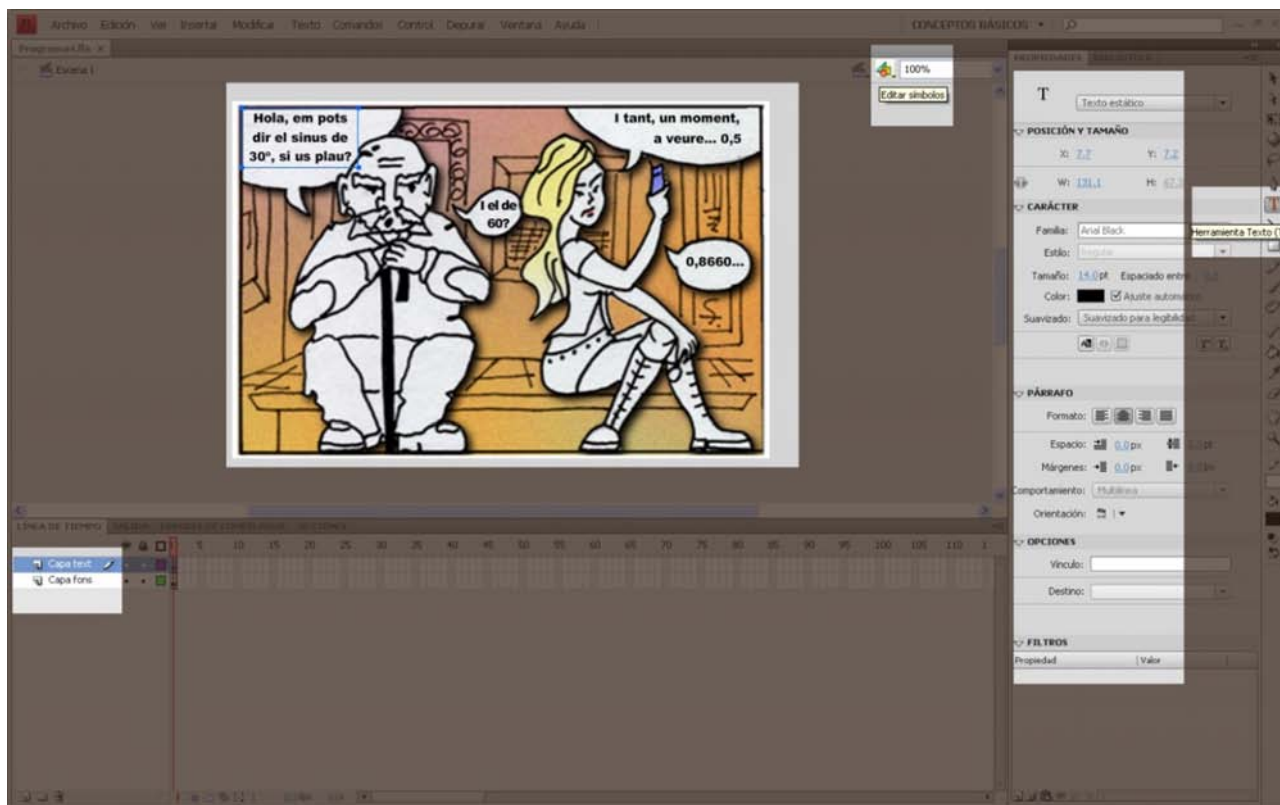
## 4. Botons i camps de text

### 4.1. Introducció

El Flash permet inserir textos dins del programa amb l'eina text (T).

En el programa 4 teniu un exemple que podeu obrir per veure com s'hi treballa.

Cal comentar, però, que els textos estàtics no són l'únic tipus de text amb què el Flash treballa. A part ofereix la possibilitat d'introduir text dinàmic i text d'introducció. Aquest fet dota el Flash de la capacitat per comunicar-nos amb l'usuari.



#### Nota

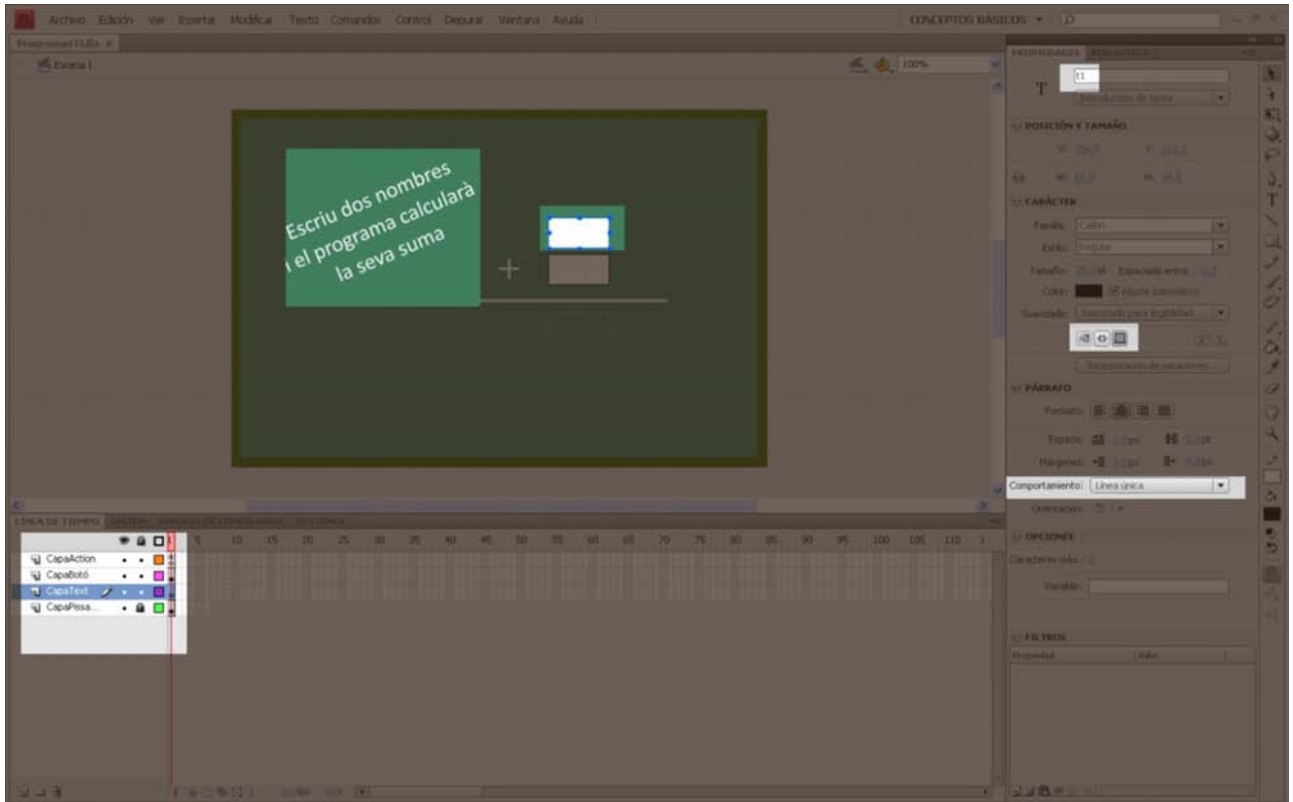
Aprofitem aquest exemple per dir que es poden canviar els noms de les capes fent doble clic a sobre.

També aprofitem per explicar una nova manera d'editar els símbols a banda de fer-hi doble clic. La idea és fer clic sobre la pestanyeta *Editar símbols*, destacada a la part superior dreta de la imatge.



## 4.2. Un programa amb un botó i els tres tipus de capses de text

Imaginem que volem crear un programa que calculi la suma de dos nombres introduïts per un usuari.

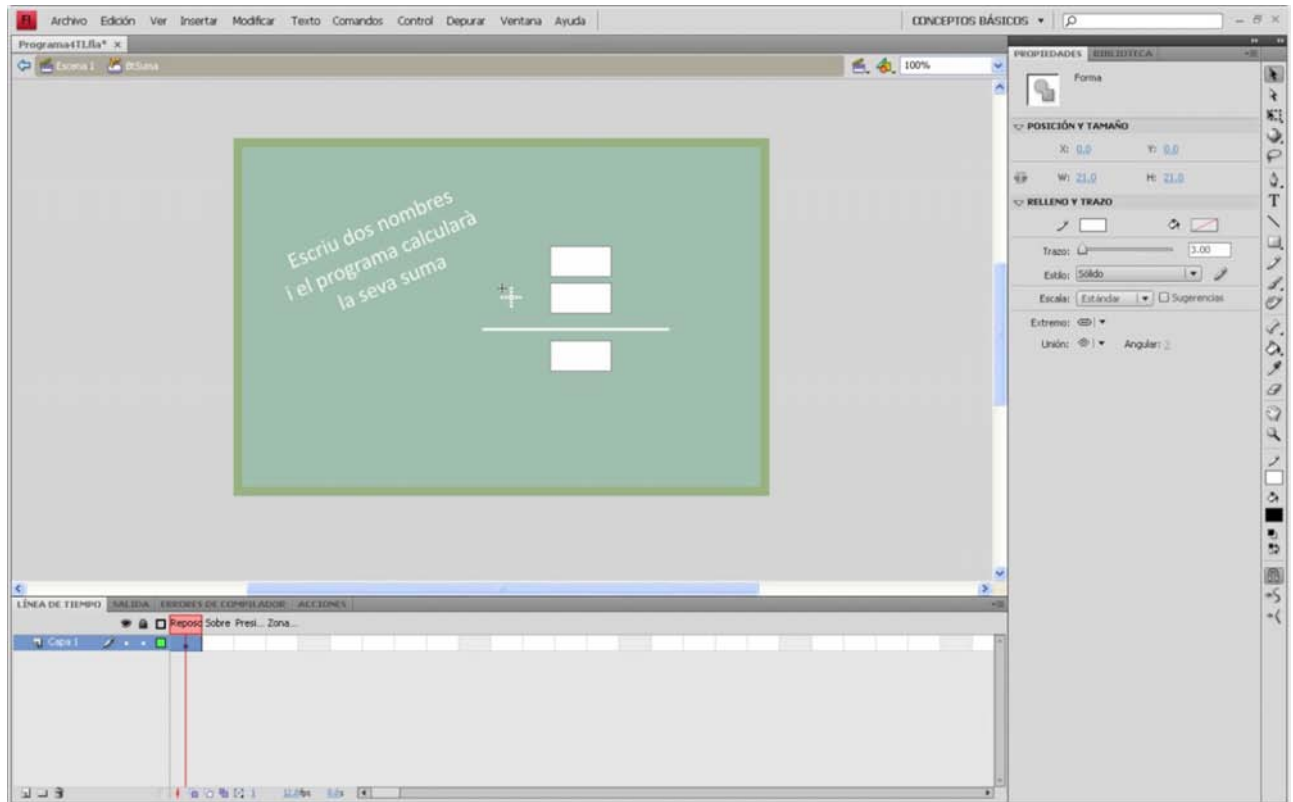


En el programa4TI trobareu un exemple d'interfície format pels elements següents:

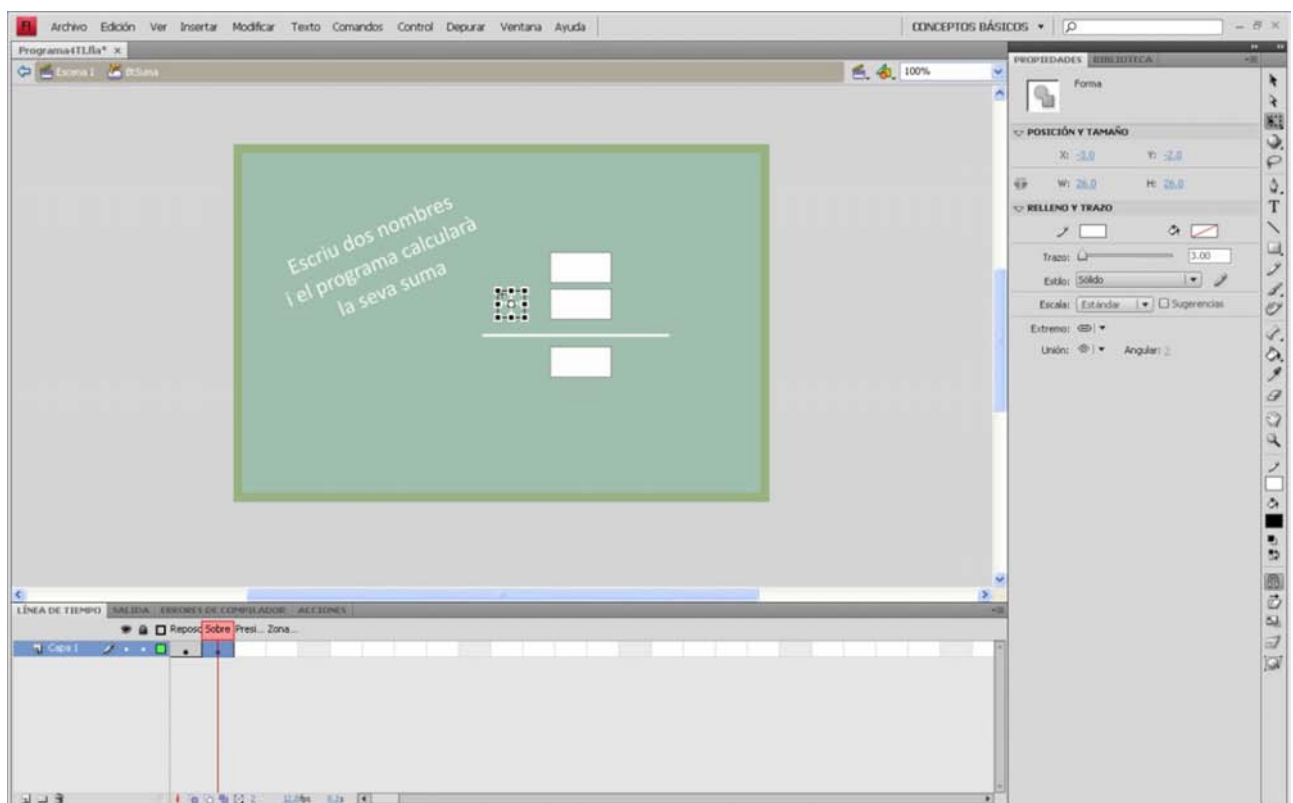
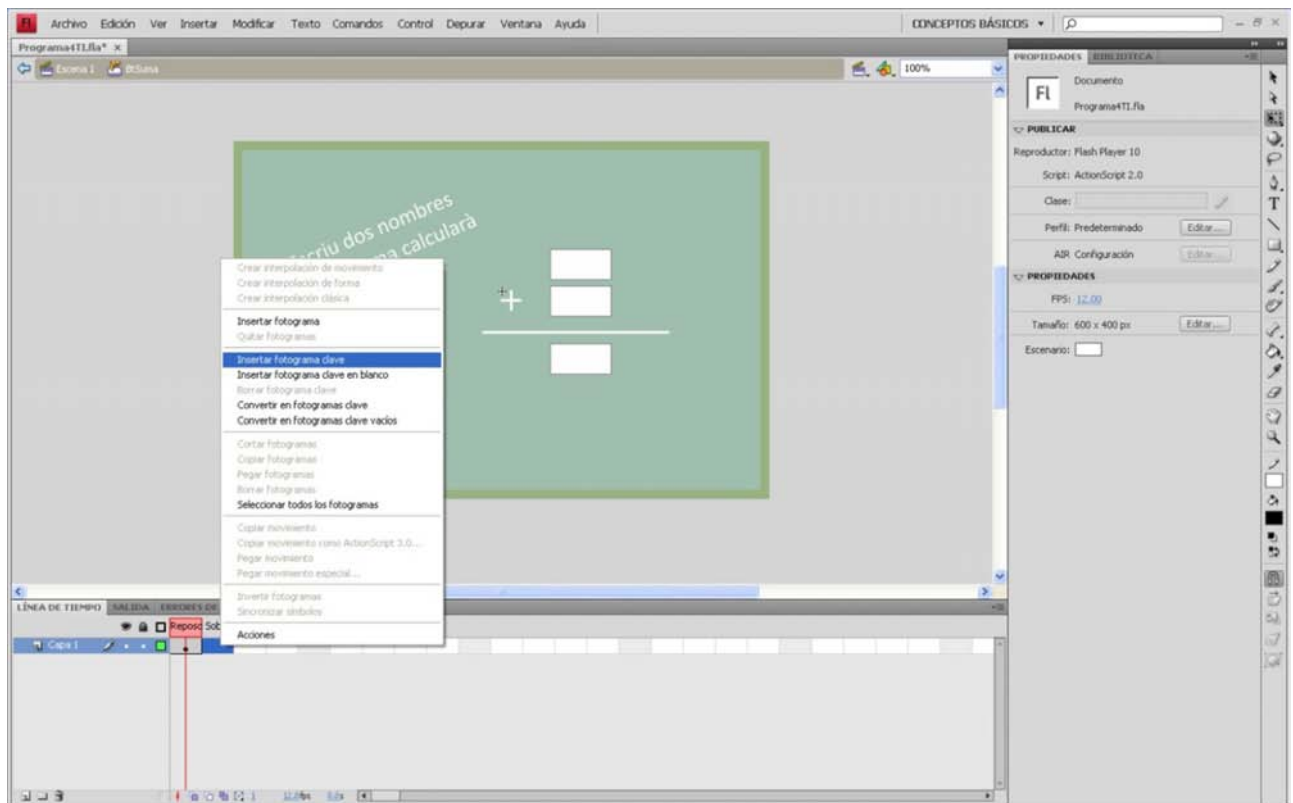
- Una pissarra com a fons creada en una capa anomenada capa Pissarra.
- Un text estàtic en què podem llegir "Escriu dos nombres i el programa en calcularà la suma".
- Dos textos d'introducció (compte a triar línia única i no multilínia, així com a seleccionar el botó perquè es visualitzi un marc al voltant seu i no seleccionar el botó de generar text com a html) amb el nom t1 i t2.
- Un text<sup>1</sup> Dinàmic, amb el nom t3, en què escriurem el resultat dels dos nombres introduïts per l'usuari.
- Un símbol<sup>2</sup> tipus botó amb aspecte de + que hem anomenat btCalcular. Per crear-lo, primer hem dibuixat, amb l'eina línia, dues línies que formen una suma. Aleshores seleccionant tot el dibuix l'hem convertit en un símbol botó. Fent doble clic a sobre, podem editar el botó tal com veiem en la imatge següent.

<sup>(1)</sup>Tots els textos estan en la capa de nom CapaText.

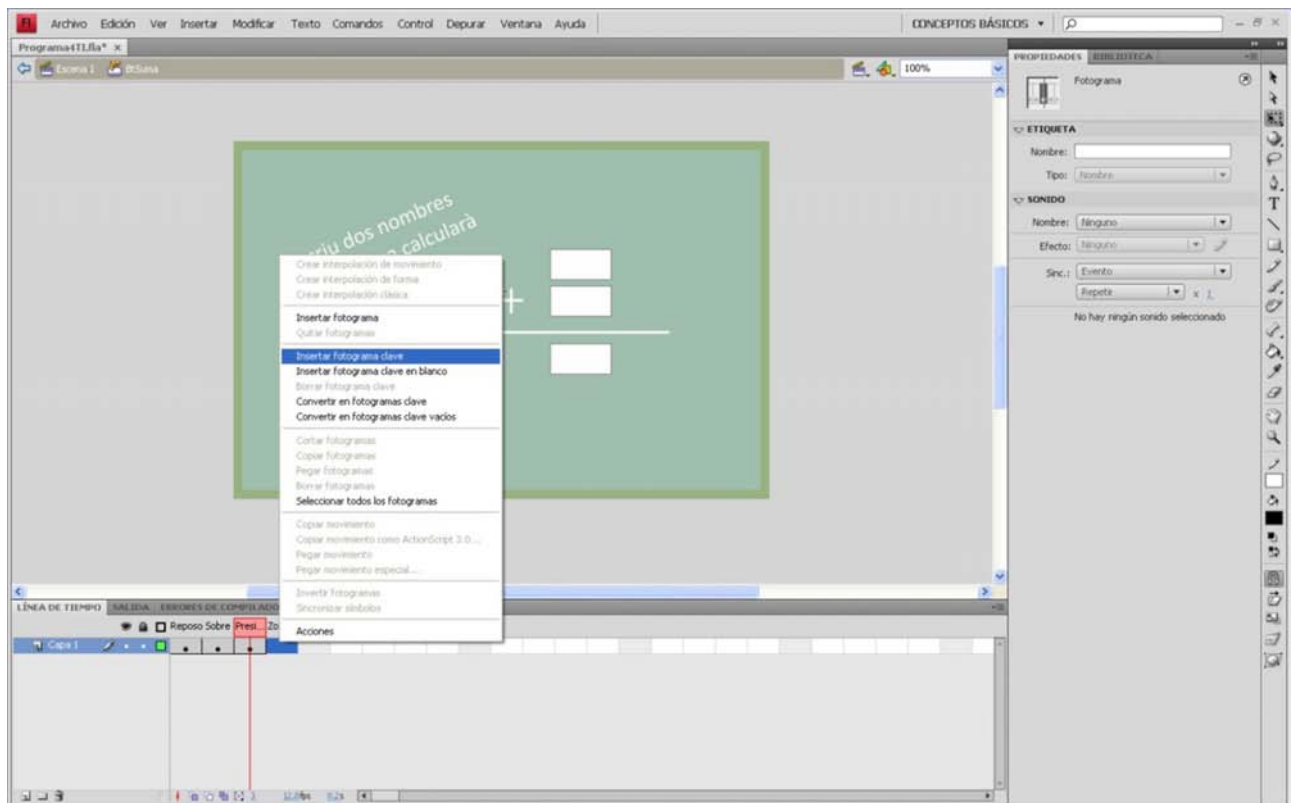
<sup>(2)</sup>En la capa de nom CapaBotó.



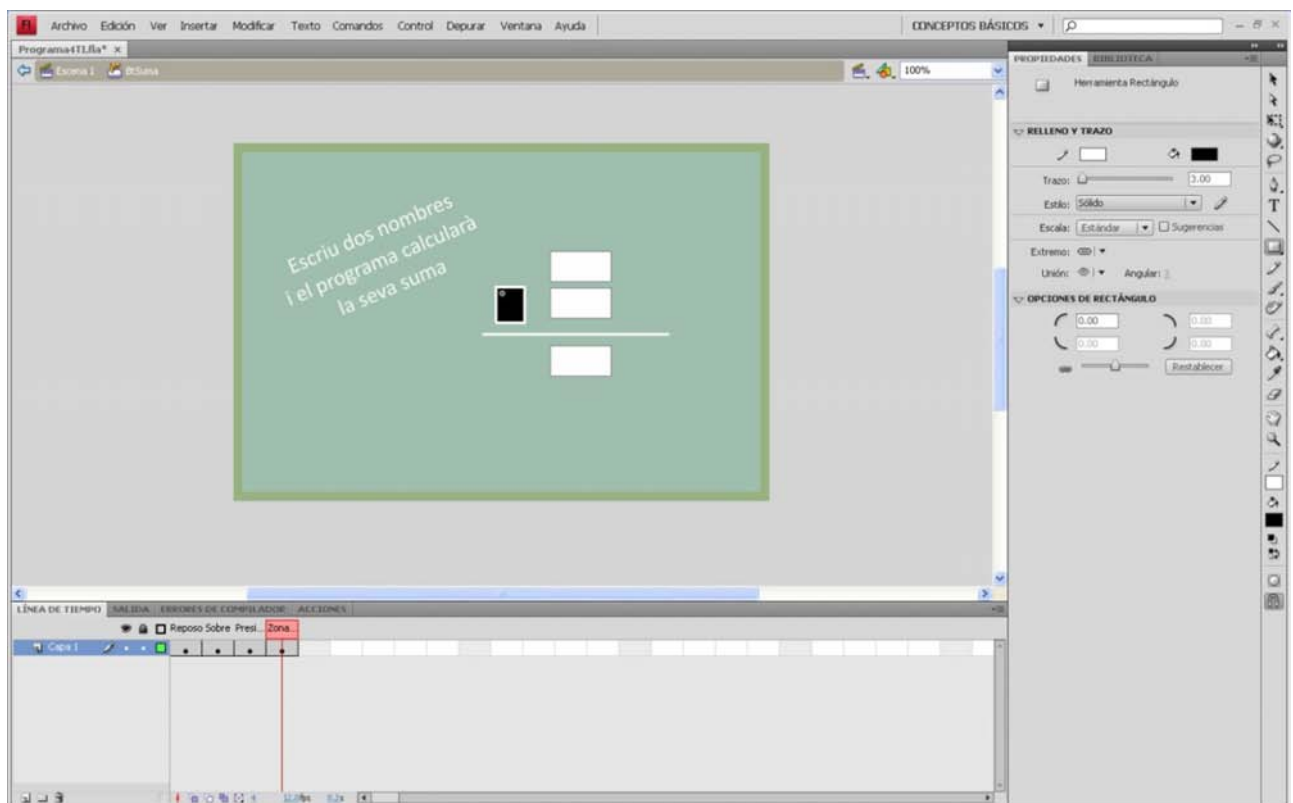
Observem que el programa utilitza la imatge inicial com a fons del botó en l'estat de repòs. Si fem clic amb el botó dret del ratolí en el marc "Sobre" i seleccionem del menú l'opció *Inserir fotograma clau*, obtindrem una còpia de la imatge inicial en aquest marc que podrem modificar per tenir la imatge del botó quan som a Sobre. Nosaltres hem escalat una mica la imatge per tenir un efecte d'augment en passar per damunt.



Tornant a repetir el procés podem crear la imatge que veurà l'usuari en clicar el botó. En aquest cas no hem creat cap efecte.



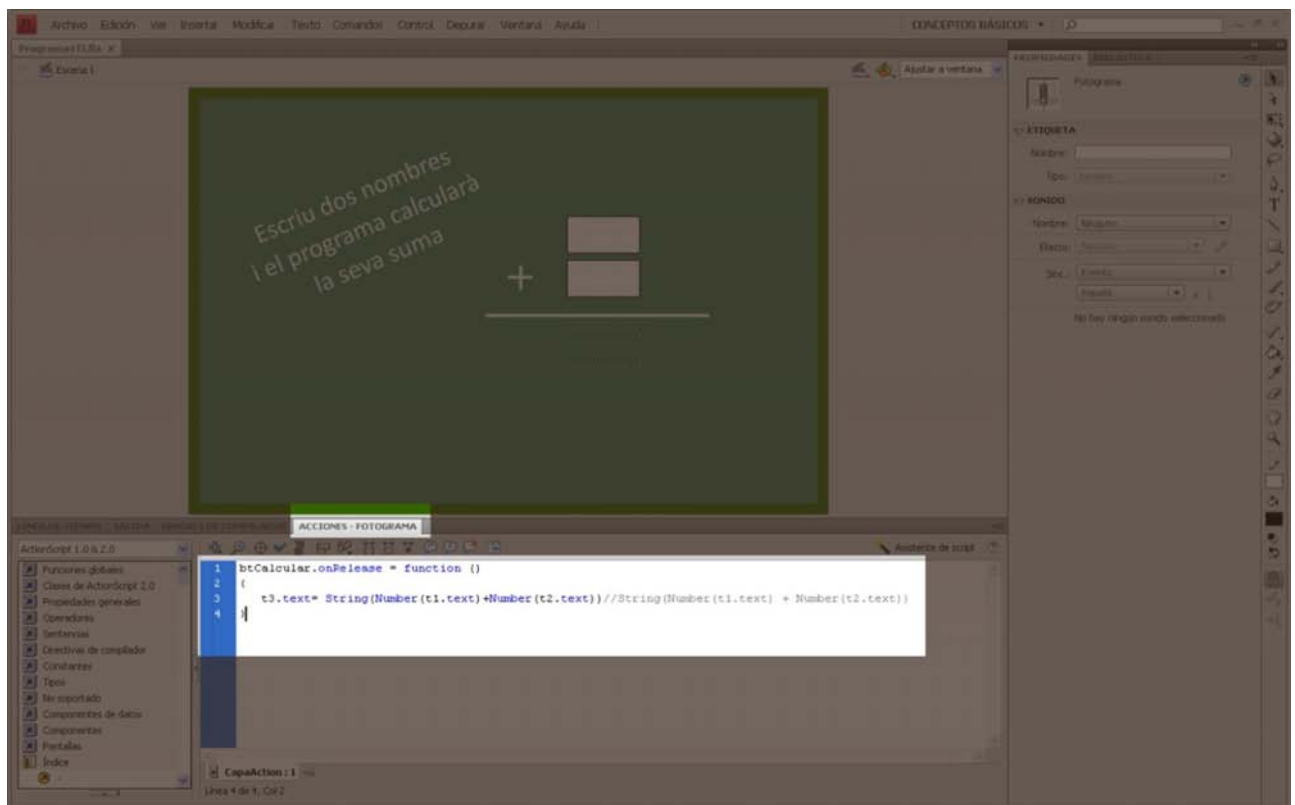
Finalment cal assenyalar l'àrea al voltant del dibuix inicial que serà sensible al ratolí. Per això creem un rectangle que tapi la + en el marc Zona.



Per tal de poder escriure el resultat de la suma d'introducció dels dos nombres hem creat una tercera capa que hem anomenat capaAction.

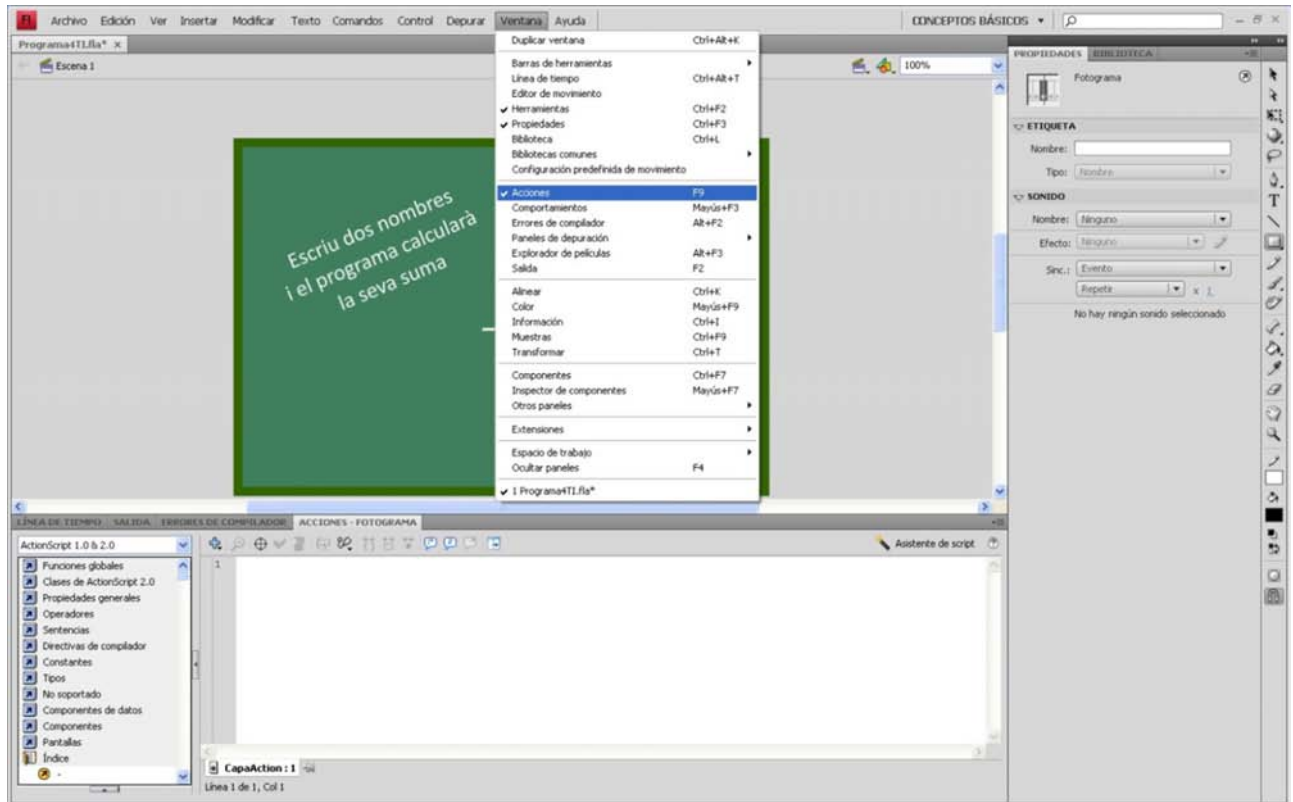
Selecciónant el marc 1 d'aquesta capa s'ha obert la finestra *Accions* i s'ha afegit el codi següent:

```
btCalcular.onRelease = function ()
{
    t3.text= String(Number(t1.text)+Number(t2.text))
    //String(Number(t1.text) + Number(t2.text))
}
```



### Nota

Si no teniu la finestra *Accions* obriu el menú de la pestanya *Finestres* i feu clic a *Accions*.



Un cop publicat el programa, si l'usuari introdueix dos nombres i fa clic sobre la +, el programa mostra el valor d'aquesta suma en el camp de text amb el nom t3.

#### Nota molt important

Cal tenir present que és possible que si triem fonts pròpies i l'usuari no les té instal·lades al seu ordinador, pot succeir que el programa no acabi de funcionar bé. Si us passa alguna cosa d'aquest tipus, podeu seleccionar per als textos fonts del dispositiu en comptes de suavitzat per llegibilitat tot canviant la família a una d'estàndard del sistema, com ara Arial. Altres problemes poden sorgir en triar en la configuració de la publicació un Flash Player "molt" actual. Penseu que els usuaris no solen anar a l'última. Per aquest motiu no solem recomanar triar la darrera versió del Flash Player a l'hora de publicar el programa.

### 4.3. Activitats

#### Activitat 1

Al llarg dels apunts hem vist tres tipus diferents de símbols: els clips de pel·lícula, els botons i els gràfics. Utilitzeu el Google per trobar les diferències entre ells i quan és recomanable utilitzar-los.

## 5. Action

### 5.1. Introducció

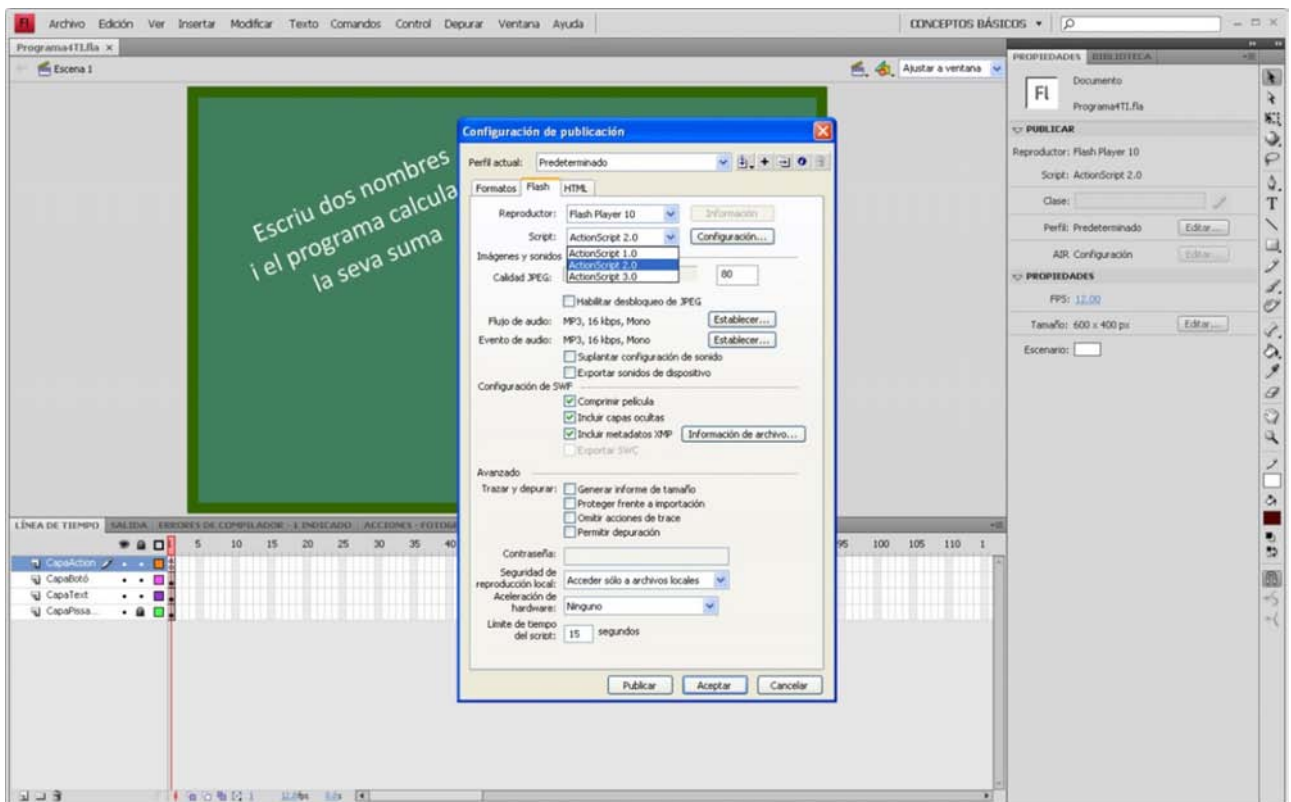
ActionScript és el llenguatge de programació de la plataforma Adobe Flash.

Actualment la majoria de programadors ja treballen en la versió 3, però no està de més aprendre una mica de la 2 per si ens toca modificar algun programa antic.

Només hi ha una opció per aprendre programació i és practicant i practicant. La nostra intenció per a aquest apartat és presentar unes primeres nocions amb totes les funcions que es poden utilitzar en aquest primer curs de Matemàtiques.

### 5.2. Unes idees d'Action

Si obriu el programa4TI i aneu a *Arxiu* -> *Configuració de publicació* i feu clic sobre el menú a la finestra Flash, veureu que el programa ha estat fet en actionScript 2.



En el codi accions del botó del programa4TI hi ha les expressions `Number()` i `String()`. Podeu observar en el resum de codi següent que `Number()` és una funció amb la capacitat de convertir un text que conté un nombre en una variable que l'ordinador ja entén que és un nombre amb què podrà treballar matemàticament. Si no el convertim hi ha l'opció que el programa en comtes de sumar concateni les expressions pensant que són paraules.

### Resum de propietats i mètodes de variables de text i numèriques

Imaginem que tenim tres variables *a*, *vText* i *b*, on *a* i *vText* són de tipus text i *b* de tipus numèric.

```
var a:String
var vText:String
var b:Number
```

### Nota

Observeu que les variables poden tenir nom de lletres o paraules combinant majúscules i minúscules.

### Conversió de nombres en frases i de frases en nombres

Si tenim  $b = 234$  i volem que *a* sigui la frase  $a = 234$  podem fer:

```
a=String(b)
```

A partir d'ara en *a* tenim la frase 234 i podem fer mètodes relacionats amb cadenes de text.

Si sabem que  $a = 123$ , és a dir, és una frase que es pot interpretar com un nombre, podem fer:

```
b=Number(a)
```

Aleshores en *b* tenim un nombre i podem fer mètodes relacionats amb nombres.

### Mètodes de les variables numèriques

Generalment, per treballar amb nombres i variables numèriques utilitzarem una construcció anomenada la classe `Math`. Aquesta construcció conté mètodes i constants que representen funcions i valors matemàtics comuns. Podem imaginar aquesta construcció com un sac on hi ha guardats nombres especials i funcions que fan coses. Per utilitzar-les només hem de dir que agafem el sac i després traiem de dins el nombre o la funció que necessitem per al nostre treball. Per exemple:

```
b=Math.min(4,5) //En b guardem el mínim de 4 o 5
b=Math.max(4,5) //En b guardem el màxim de 4 o 5
b=Math.pow(2,3) //En b guardem 2 elevat a 3 = 8 (= 2^3)
b=Math.floor(2.345) //En b guardem el valor arrodonit a la baixa que és 2
b=Math.random() //En b guardem un nombre entre 0 i 1 aleatori
b=Math.abs(-3) //En b guardem el valor absolut del nombre
b=4+2*2-4/2 //En b guardem el valor 4+4-2=6
b=5 %2 //En b guardem la resta de la divisió de 5 per 2, que és 1
// (relacionat amb mòdul)
b=Math.Pi //En b guardem el valor de Pi. Bé en realitat guardem una
// aproximació, ja que Pi és un nombre amb infinits decimals
// impossible de ser determinat totalment.
```

### Nota per a quan hàgim de treballar amb decimals

En Flash els nombres decimals s'han de posar amb punts i no amb comes.



## Mètodes de les variables cadenes de text

En les variables de tipus text, podem aplicar directament funcions especials desenvolupades expressament pels creadors del Flash per operar amb aquest tipus de classe. Per exemple,  $b = a.length$  guarda en  $b$  la longitud o nombre de caràcters de la frase guardada en  $a$ . Per exemple, si  $a$  conté la paraula *vaca*, en  $b$  obtindrem el valor numèric 4.

Vegem altres tipus de funcions especials:

```

a=vText.substr(0,1) //En a guardem la primera lletra de la frase guardada en text
a=vText.substr(1,1) //En a guardem la segona lletra de la frase guardada en text
a=vText.substr(2,1) //En a guardem la tercer lletra de la frase guardada en text
a=vText.substr(3,1) //En a guardem la quarta lletra de la frase guardada en text
a=vText.substr(4,1) //En a guardem la cinquena lletra de la frase guardada en text
...
a=vText.substr(0,2) //En a guardem la primera i segona lletra de la frase guardada
//en text
a=vText.substr(0,3) //En a guardem la primera, segona i tercera lletra de la frase
//guardada en text
...
a=vText.substr(1,2) //En a guardem la segona i tercera lletra de la frase guardada
//en text
a=vText.substr(2,2) //En a guardem la tercera i quarta lletra de la frase guardada
//en text
a=vText.charAt(0) //En a guardem la primera lletra de la frase guardada en text
a=vText.charAt(1) //En a guardem la segona lletra de la frase guardada en text
a=vText.charAt(2) //En a guardem la tercera lletra de la frase guardada en text
a=vText.charAt(3) //En a guardem la quarta lletra de la frase guardada en text
a=vText.charCodeAt(0) //En a guardem el codi en decimal ascii de la primera lletra
//de la frase guardada en text
a=vText.charCodeAt(1) //En a guardem el codi en decimal ascii de la segona lletra
//de la frase
a=vText.charCodeAt(2) //En a guardem el codi en decimal ascii de la tercera lletra
//de la frase
a=vText.charCodeAt(3) //En a guardem el codi en decimal ascii de la quarta lletra
//de la frase
a=text.toUpperCase() //En a guardem la frase guardada en text en majúscules
a=vText.toLowerCase() //En a guardem la frase guardada en text en minúscules

```

## Consideracions de sumar o concatenar variables

### a) Imaginem tres variables de text $a$ , $b$ , $c$

Si en  $a$  tenim Pep i en  $b$  tenim Anna

En  $c = a + b$  (ó també  $a \& b$ ) tindrem la frase PepAnna

En  $c = a + " " + b$  tindrem la frase Pep Anna

En  $c = a + "estima l'" + b$  tindrem la frase Pep estima l'Anna

Si en  $a$  tenim 5 i en  $b$  tenim 4

En  $c = a + b$  tindrem 54

### b) Imaginem tres variables numèriques $a$ , $b$ , $c$

Si en  $a$  tenim 5 i en  $b$  tenim 4

En  $c = a + b$  tindrem el nombre 9

Observeu que la suma de nombres si estan en text no és el mateix que la suma de nombres si estan en variables numèriques. Això obliga a pensar en quin tipus de variables estem treballant. Per això sempre recomanem declarar les variables a l'inici del programa, tot i que no és obligatori.

És convenient saber que si no declarem les variables i en mig del codi de programació fem l'assignació:  $a = 10$ , automàticament el Flash entén que  $a$  és una variable numèrica i que té guardat el nombre 10.

L'assignació  $a = "10"$  permet al Flash entendre que  $a$  és una variable de text i que té per valor l'expressió o frase 10. Observeu que en una variable de text les cometes desapareixen i, per tant, només serveixen per indicar que  $a$  és una variable de text, en comptes de numèrica. Per exemple, si fem  $a = "La llum està encesa"$ , en  $a$  tenim guardada la frase *La llum està encesa* que comença per *L*. És a dir  $a.Text.charCodeAt(0)$  és *L* i no la cometa ".

Recordem, doncs, que per definir una variable de text o introduir una paraula o frase en una variable de text s'han de posar les cometes obligatòriament a l'expressió, encara que tal com hem explicat la variable no les guarda i, per tant, no les podem recuperar.

### 5.3. Quan un programa "parla"

Per tal que un programa mostri a l'usuari un valor final calculat mitjançant programació sol haver generalment<sup>3</sup>, un botó que indicarà al programa que executi certes instruccions i una línia d'aquest codi que demani al programa mostrar una informació en alguna caixa de text que es visualitza en la pantalla.

<sup>(3)</sup>Generalment però no sempre, ja que per exemple, podem fer que un programa mostri una acció en passar per sobre d'un objecte, en arrossegar una imatge, en mantenir pressionat el botó dret del ratolí, etc.

En el codi accions del botó del programa<sup>4</sup>TI, l'expressió següent: `btCalcular.onRelease = function ()` amb `onRelease` darrere de `btCalcular` indica que realitzarem les accions del parèntesi quan l'usuari faci clic sobre el botó amb el nom `btCalcular`. Amb l'expressió: `btCalcular.onMouseUP = function ()` les accions s'executarien quan passéssim el punter del ratolí per sobre del botó `btCalcular`.

Amb l'expressió: `t3.text = ...`, recordeu que `t3` era el nom d'una caixa de text, en `t3` apareix exactament el text que hem volgut que el programa calculés.

#### Nota

Podem també canviar el color de la caixeta de text `t3` modificant una mica el codi de la finestra accions:

```
btCalcular.onRelease = function ()
{
    t3.text= String(Number(t1.text)+Number(t2.text))
    //String(Number(t1.text)+Number(t2.text))
    t3.textColor = 0x009900;
}
```

Tot i el bon funcionament del programa, si un usuari introduís una lletra en comptes d'un nombre en una de les capsetes d'introducció, el programa no hauria d'executar les instruccions de la suma, sinó que hauria d'avisar l'usuari que no ha introduït correctament la informació sol·licitada. Protegir un programa de la "inconsciència" de l'usuari és un dels punts més difícils que un programador ha d'intentar. I diem "intentar" ja que us assegurem que per molt que ens esforcem, mai no estarem a l'alçada del singular enginy d'algun usuari.

## 5.4. Activitats

### Activitat 1

Cerqueu acudits d'informàtics per la xarxa i presteu especial atenció a les lleis de Murphy per a la informàtica.

### Activitat 2

Creeu un nou arxiu en Flash(AS 3.0) format per un text estàtic que demani d'introduir un nombre, un text d'introducció anomenat txtEntrada, un text dinàmic anomenat txtResultat i un botó anomenat botoCalcular.

Afegiu el codi següent a l'acció de clicar a sobre el botó:

```
btCalcular.addEventListener(MouseEvent.CLICK, funcioEscriureSegüent);
function funcioEscriureSegüent(tipusEvent:MouseEvent):void
{
    var a:Number
    a = Number(_root.txtEntrada.text);
    if (isNaN(a))
    {
        _root.txtResultat.text="Malament, no has introduït un nombre";
    }else {
        _root.txtResultat.text= String(a+1);
    }
}
```

Cerqueu a la web que fa l'expressió `isNaN()` (per exemple introduint a Google les paraules *isNaN Flash* o directament a la pàgina d'Adobe ([http://help.adobe.com/es\\_ES/AS2LCR/Flash\\_10.0/help.html?content=00000576.html](http://help.adobe.com/es_ES/AS2LCR/Flash_10.0/help.html?content=00000576.html))).

Què penseu que fa el programa?

Podeu observar que si l'heu creat AS 3.0, esteu publicant en ActionScript 3.

Observeu les diferències a l'hora de fer funcionar un botó entre AS 2.0 i AS 3.0.

### Activitat 3

Obriu la carpeta **Exemples if, while i for**. Hi trobareu tot d'exemples de programes que demanen d'introduir nombres i ofereixen un resultat. Llegiu el codi en Action i cerqueu les diferències d'un tipus de programa a d'altres.

### Activitat 4

Obriu la carpeta **Paisatge marítim**. Hi trobareu un programa que fa un dibuix d'un paisatge mariner sense utilitzar cap altra cosa que codi `actionScript`. Observeu la construcció de les funcions que ajuden a fer més reutilitzable el codi.

### Activitat 5

Obriu la carpeta **Cel Nocturn**. Hi trobareu un programa que duplica una pel·lícula anomenada Estrella i la disposa en l'escenari aleatòriament. Observeu el codi i cerqueu l'expressió: `random()`. Trobeu com podeu utilitzar aquesta funció.

### Activitat 6

Obriu la carpeta **Bombolles**. Hi trobareu un programa que escala unes bombolles i les va elevant onduladament. Observeu el codi i cerqueu l'expressió `gotoAndPlay()`. Cerqueu en els fòrums d'internet què fa i com s'utilitza aquesta funció. Cerqueu també les expressions `gotoAndStop()`, `Stop()` i `Play()`.

### Activitat 7

Esbrineu l'ús de les expressions `_root` i `_parent`.

### Activitat 8

Canvieu les configuracions de publicació de tots els programes del tutorial en arxius amb script `ActionScript 3.0`. Modifiqueu, si és necessari, la programació per tal que els programes funcionin correctament en `AS3`.