

Cuenca - Ecuador 2026

ISBN: 978-9942-7180-7-5

actionscript

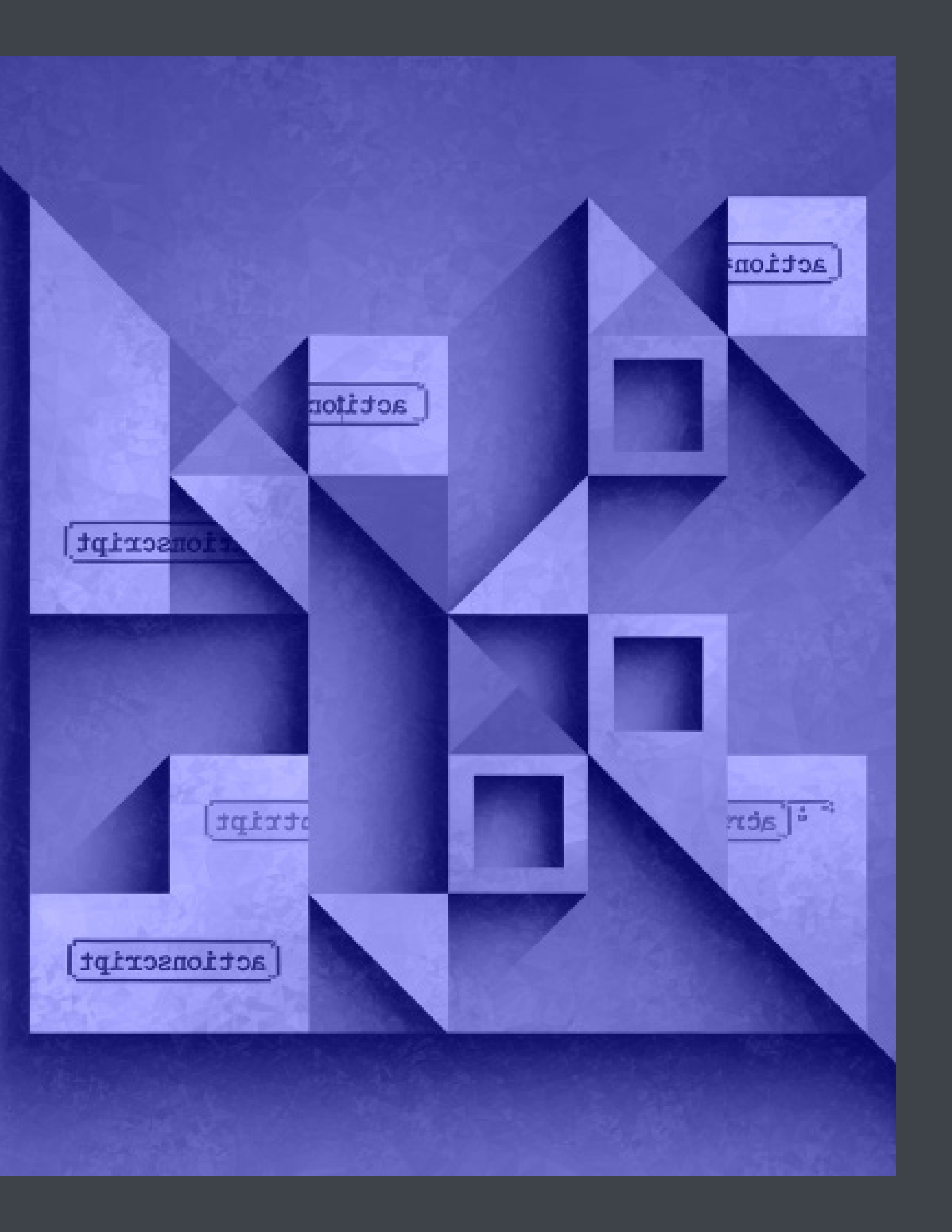
actionscript

actionscript

actionscript

actionscript

actionscript



INSTITUTO SUPERIOR TECNOLÓGICO PARTICULAR SUDAMERICANO

Rector Institucional

Dr. Carlos Gabriel Pérez Pérez

Vicerrectora Administrativa

Mgtr. Mónica Elizabeth Vargas Garcés

Vicerrector Académico

Mgs. Juan Marcelo Pérez

Coordinador de Investigación

Mgtr. Richard Martínez Villegas

Director del Departamento de Innovación y absorción

Mgtr. Nancy Maribel Eras Eras

Diseño y Diagramación:

Mg. Bruno Avila Nievecela

Autor:

Mg. Bruno Avila Nievecela

Directora del Proyecto de Investigación:

Mg. María Isabel Alvarado

Codirector del Proyecto de Investigación:

Mg. Bruno Avila Nievecela

Docentes Investigadores:

Mg. Marco Pazmiño Piedra

Lic. Amanda Espinoza Chiqui

Año de publicación: febrero 2026

EDITORIAL: Instituto Superior Tecnológico Sudamericano

Como citar:

Avila, B. (2025). Desarrollo de Aplicaciones en Adobe Animate AS3 (Ed. digital). Instituto Tecnológico Sudamericano.

Esta obra es fruto del proyecto de investigación "Desarrollo de aplicaciones en Adobe Animate AS3". Y fue revisada bajo la modalidad de doble par ciego.

Cuenca - Ecuador

Todos los derechos reservados.

1. Introducción

Las herramientas digitales han revolucionado la educación, facilitando el acceso a información actualizada y promoviendo un aprendizaje más dinámico e interactivo. Su integración en los entornos educativos permite el desarrollo de competencias, mejorando la autonomía, la colaboración y la toma de decisiones en contextos reales.

Las aplicaciones digitales ofrecen experiencias inmersivas mediante simulaciones, recursos interactivos y entornos de aprendizaje personalizados. Estas tecnologías no solo optimizan la enseñanza, sino que también potencian habilidades esenciales como la resolución de problemas, la comunicación y la empatía. Su implementación en la educación prepara a los estudiantes para los desafíos del mundo profesional, adaptándose a las demandas de una sociedad en constante evolución.

En la era digital, el desarrollo de aplicaciones interactivas se ha convertido en una habilidad esencial para diseñadores gráficos, desarrolladores y cualquier interesado en la creación de contenido dinámico. Esta guía está diseñada para ofrecer un enfoque sencillo y directo sobre cómo crear aplicaciones utilizando Adobe Animate y ActionScript 3.0.

El manual no requiere conocimientos avanzados de programación, ya que se enfoca en explicar de manera práctica, usando recursos que faciliten la inclusión de botones y programación, necesarias para construir una app funcional. Aprenderás a trabajar con movie clips, botones interactivos y código básico que controlará la navegación y la interactividad dentro de tu app. Para complementar la información, incluiremos indicaciones específicas sobre dónde colocar capturas de pantalla del programa, lo que te permitirá visualizar cada etapa del proceso y facilitará la comprensión de los conceptos técnicos.

Índice

01
pag. 05

Introducción

02
pag. 08

Requisitos Previos

03
pag. 14

Configuración del Proyecto en Adobe Animate

04
pag. 16

Organización de la Línea de Tiempo y Fotogramas

05
pag. 22

Creación de Movie Clips para Animación

06
pag. 27

Programación en ActionScript 3.0

07
pag. 38

Interactividad con Botones y Contenido

08
pag. 40

Enlazando Contenido a través de botones

09
pag. 41

Pruebas y Exportación de la App

10
pag. 48

Errores

11
pag. 50

Glosario

12
pag. 54

Bibliografía

Agradecimientos

THANK YOU

Este libro es el resultado de un camino de aprendizaje, esfuerzo y colaboración. Quiero expresar mi más sincero agradecimiento al Instituto Tecnológico Sudamericano, a sus docentes y estudiantes, quienes con su entusiasmo y compromiso hicieron posible el desarrollo de ERIA. Su apoyo y participación han sido fundamentales para dar vida a esta herramienta educativa.

A mis colegas y amigos, por sus ideas, su guía y su aliento constante. A mi familia, por ser mi mayor inspiración y por acompañarme en cada paso de este recorrido.

Finalmente, dedico este trabajo a todos aquellos que creen en la innovación y la educación como motores de cambio. Que este libro sea un aporte significativo para la enseñanza y el futuro de la educación superior en Ecuador.

2. Requisitos previos

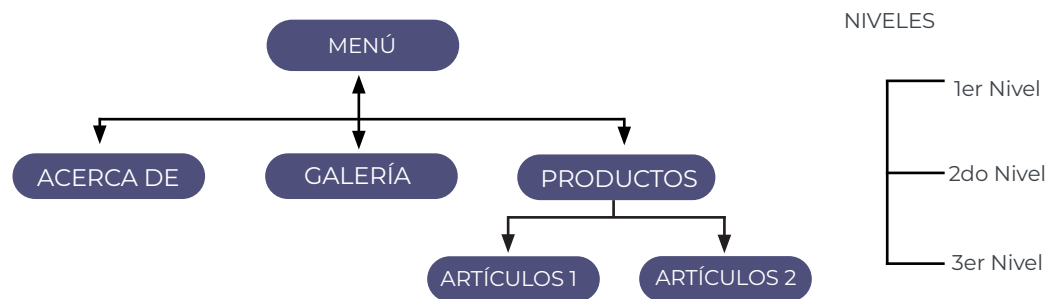
Crear una app puede ser un proceso complejo que si no se planifica correctamente. Para comenzar, es esencial definir el propósito de la aplicación: ¿qué necesitas que haga? Según los objetivos planteados, podrás identificar los recursos y herra-

mientas para cumplirlos. Las apps más sencillas son las informativas, las cuales se componen de contenido didáctico como textos, imágenes, videos, sonidos e infografías.



Es crucial ser claro y conciso con la información que incluirás en la app. Para lograrlo, delimita bien el alcance del proyecto y prioriza los elementos más importantes, manteniendo un enfoque ordenado y estructurado. Organiza los contenidos teniendo en cuenta cómo funcionan las apps en general: la mayoría cuenta con una pantalla de inicio y un menú que permite navegar entre las secciones. Una excelente herramienta para planificar esta estructura es un diagrama de flujo, que te ayudará a distribuir la información de manera lógica y eficiente.

DIAGRAMA APP



Teniendo en cuenta la cantidad de información, los recursos que utilizarás, el alcance que tendrá, la interfaz, que elementos multimedia se ocuparan, ya puedes empezar, esto tiene que estar correctamente planificado ya que los cambios que puedan surgir posteriormente pueden afectar de forma dramática el desarrollo de la app.

RECOMENDACIONES

Desarrollo de recursos visuales para tu aplicación

Al momento de desarrollar tu aplicación, es fundamental planificar y seleccionar cuidadosamente los recursos que utilizarás. Una herramienta clave para crear elementos visuales como fondos, botones y textos creativos es Adobe Illustrator. Este software te permite diseñar gráficos vectoriales de alta calidad, lo que ayuda a mantener una estética coherente y profesional en tu app.

Es importante tener en cuenta el formato y el peso de los recursos visuales, como imágenes o gráficos, ya que estos pueden afectar el rendimiento de la aplicación. Utiliza formatos optimizados (como PNG para imágenes con transparencia o JPEG para fotografías) y comprime los archivos para reducir su tamaño sin perder calidad. Esto evitará que la app se vuelva pesada y lenta, lo que podría perjudicar la experiencia del usuario.

Integración entre Adobe Illustrator y Adobe Animate

Adobe Animate es una excelente herramienta para animar y dar interactividad a los elementos diseñados en Illustrator. Ambos programas trabajan de manera integrada, lo que facilita copiar y pegar elementos directamente de Illustrator a Animate. Sin embargo, si tus diseños incluyen efectos complejos como sombras, destellos o degradados, es recomendable eliminarlos en Illustrator y aplicarlos posteriormente en Animate.

Esto se debe a que estos efectos pueden aumentar considerablemente el peso de los vectores, afectando el rendimiento de la aplicación.

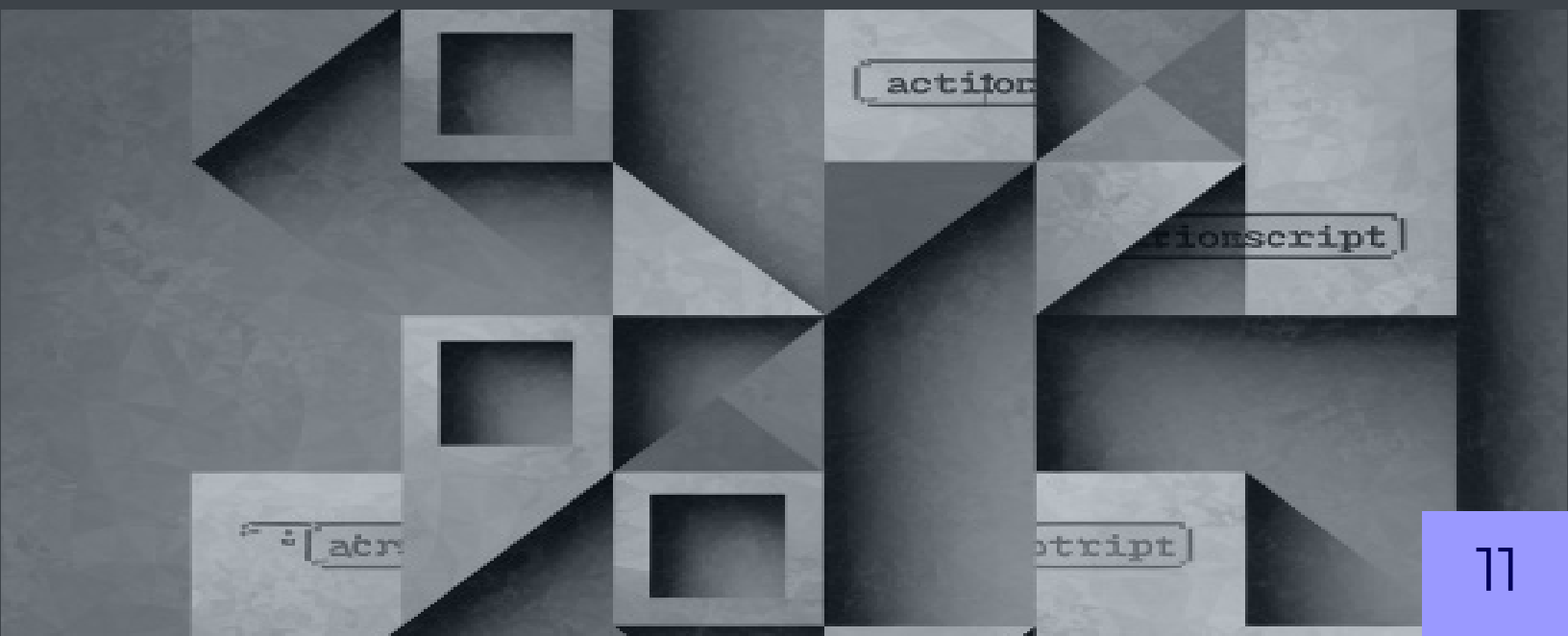
Consejos adicionales para optimizar recursos:

Simplifica los diseños: Evita el uso excesivo de efectos y detalles que no aporten valor al diseño final.

Organiza tus capas: Mantén una estructura ordenada de capas en Illustrator para facilitar la edición y exportación de elementos.

Prueba en diferentes dispositivos: Asegúrate de que los recursos visuales se vean bien y funcionen correctamente en distintos tamaños de pantalla y resoluciones. Utiliza herramientas de optimización: Emplea software o plugins para comprimir imágenes y reducir el peso de los archivos sin sacrificar calidad.

Siguiendo estas recomendaciones, podrás crear una aplicación visualmente atractiva, optimizada y con un rendimiento óptimo para los usuarios.



En el computador

Antes de comenzar con el desarrollo de tu app, asegúrate de contar con los siguientes requisitos:

1. Software Necesario:

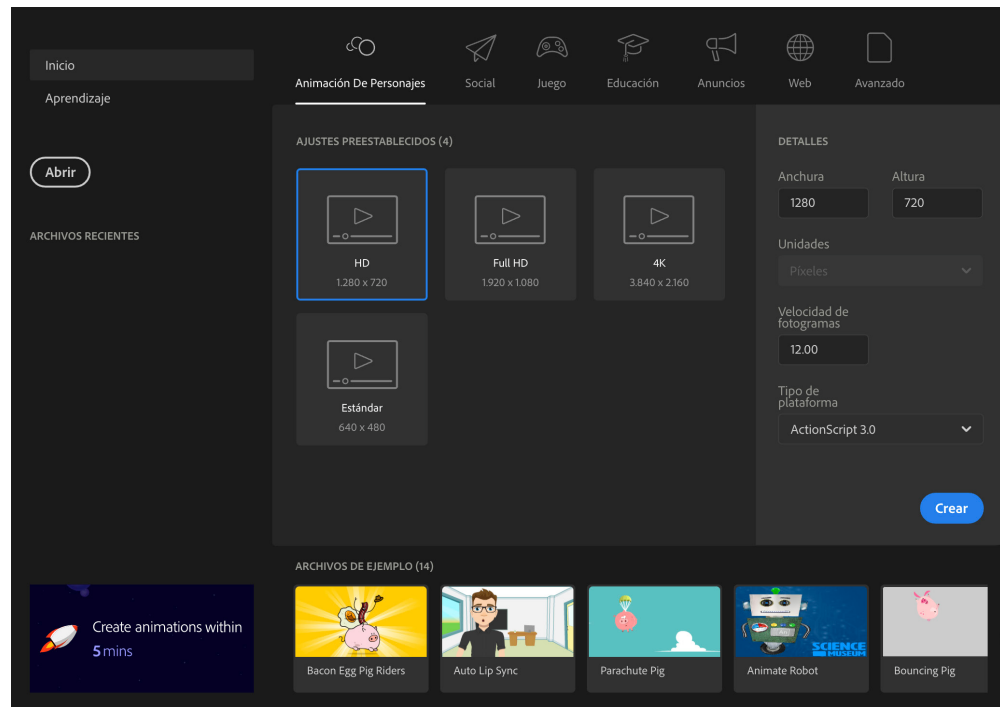
- Adobe Animate (versión actualizada): Puedes descargarlo desde la página oficial de Adobe.
- Adobe AIR SDK: Necesario para exportar aplicaciones móviles.
- Emulador de Android o un dispositivo Android físico para pruebas.

2. Conocimientos Básicos:

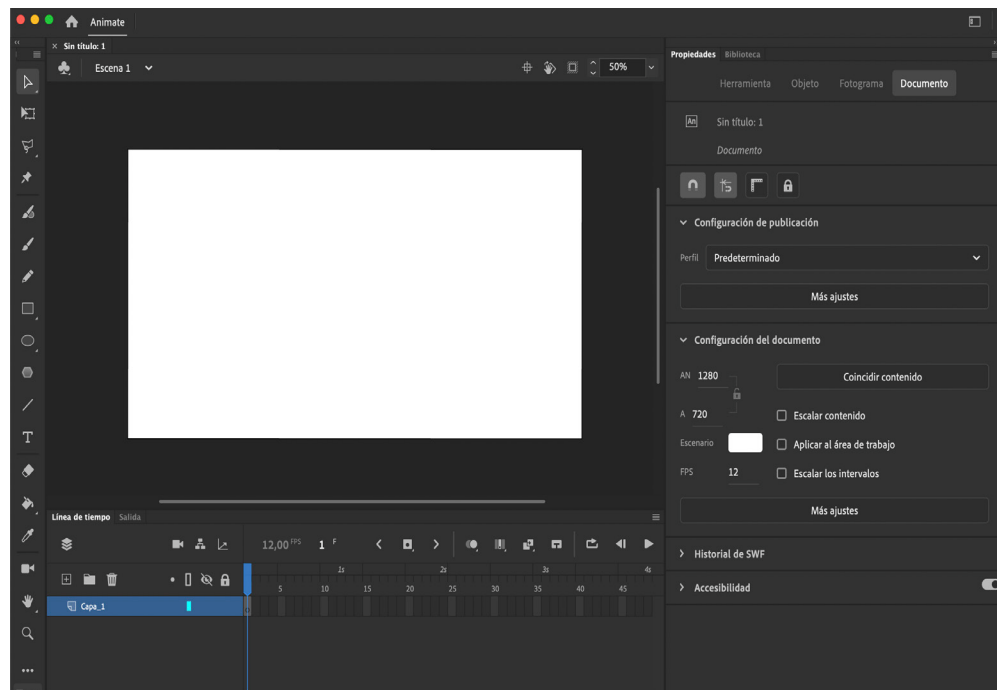
- Familiaridad con conceptos básicos de Diseño Gráfico.
- Conocimiento elemental sobre el manejo de la línea de tiempo y capas en Adobe Animate, recuerda que el manejo de los comandos y herramientas es muy similar al de Adobe Illustrator.
- Comprensión básica de programación (aunque explicaremos el uso de ActionScript 3.0 paso a paso).

3. Recursos Complementarios:

- Imágenes y gráficos vectoriales para las pruebas de animación.
- Iconos y botones que se utilizarán para la navegación.
- Sonidos, videos, enlaces, archivos



Creando un nuevo proyecto



Espacio de trabajo

4. Planificación de contenido:

Para el desarrollo de cualquier recurso multimedia o aplicación es necesaria una planificación que permita organizar los contenidos de acuerdo a una jerarquía muy similar utilizada en las webs, es decir estableciendo, por ejemplo una pantalla de inicio con un menú que permita moverse hacia los contenidos.

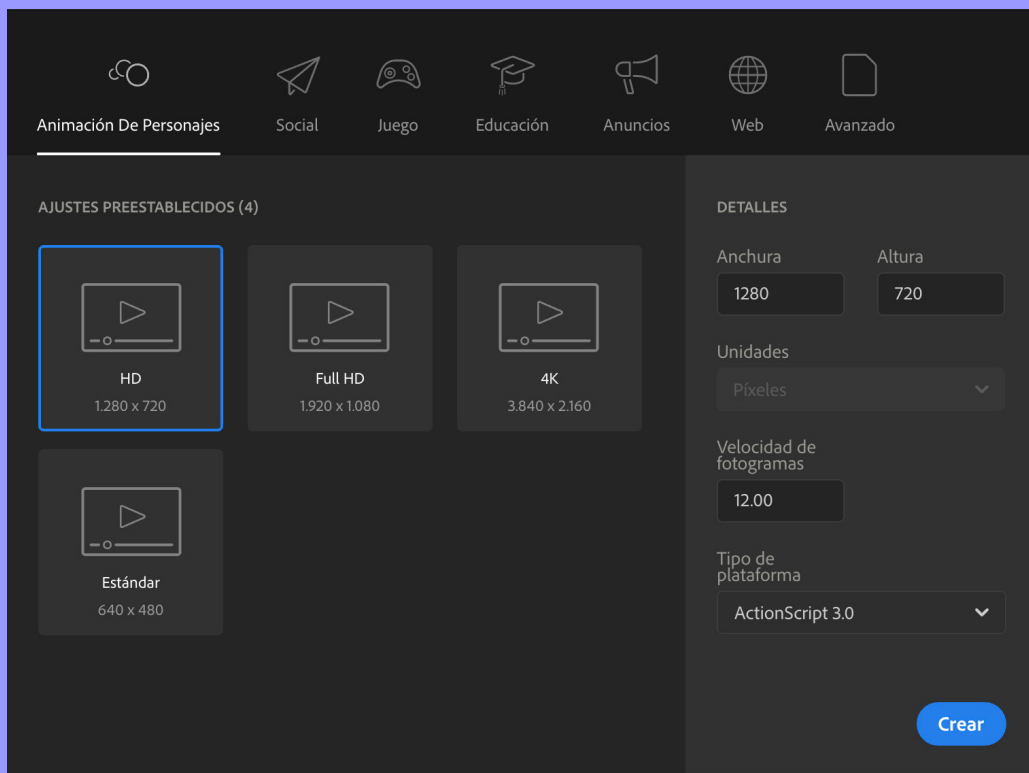
Recordar también la importancia de delimitar la longitud de los contenidos, esto ya que a mayor cantidad de recursos visuales o sonoros, esto aumentará el tamaño de la aplicación y esto podría afectar al rendimiento de la misma.

3. Configuración del Proyecto en Adobe Animate

Para comenzar a crear tu aplicación, es fundamental realizar una organización previa y recopilar la información necesaria. A continuación, se presentan los pasos que debes seguir para desarrollar tu app de forma ordenada. Esto te permitirá mantener un control claro del proceso, saber el tamaño, formato, fotogramas, y destino final de tu app.

1. Creación de un Nuevo Proyecto

- Abre Adobe Animate.
- Haz clic en Archivo > Nuevo.
- Selecciona la opción ActionScript 3.0.
- Configura el tamaño del escenario: 1080 x 1920 píxeles para un formato vertical.
- Establece la velocidad de fotogramas en 24 fps.



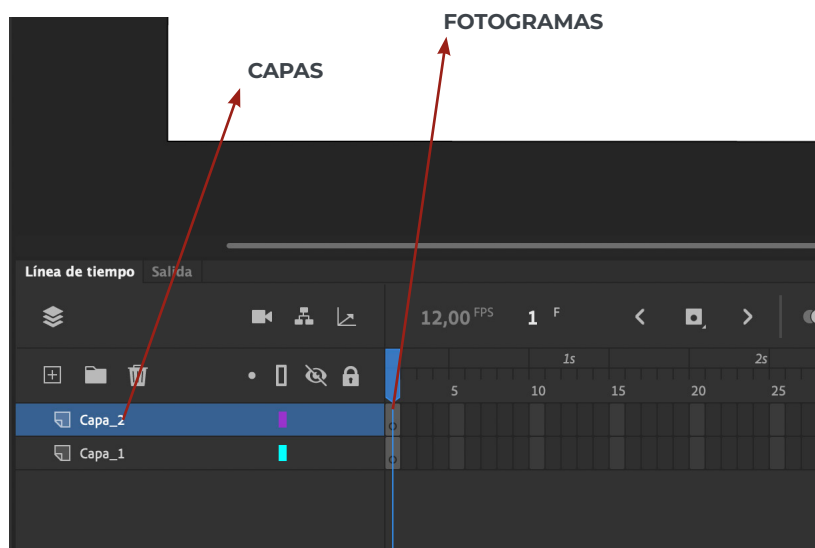
Determinando el espacio de trabajo para la App

4. Organización de la línea de tiempo y fotogramas

La línea de tiempo es una herramienta fundamental en Adobe Animate, ya que te permite organizar y controlar la secuencia de animaciones e interacciones dentro de tu app.

1. Creación de un Nuevo Proyecto

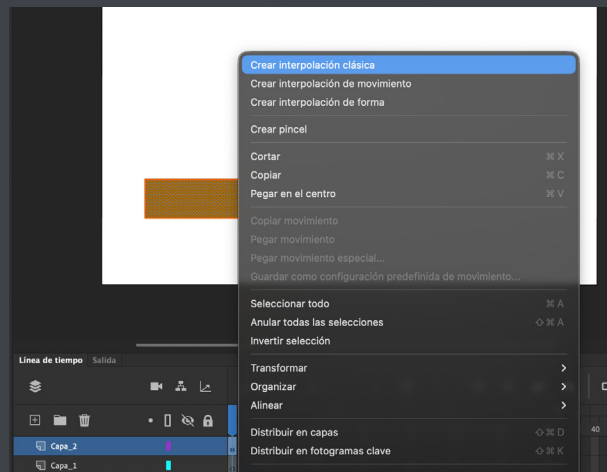
- La línea de tiempo está compuesta por capas y fotogramas.
- Las capas permiten organizar diferentes elementos de la animación (imágenes, texto, botones, etc.).
- Los fotogramas representan el paso del tiempo y permiten definir cuándo aparece o desaparece un elemento.



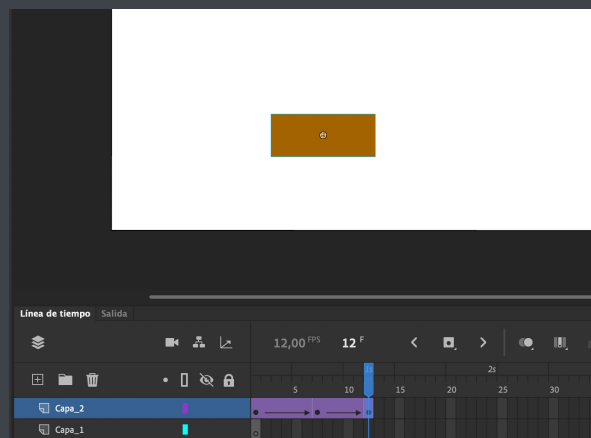
Ubicación de capas y fotogramas

2. Añadiendo Capas y Fotogramas Clave

- Haz clic en el ícono de “Nueva Capa” para agregar una capa.
- Para insertar un fotograma clave, selecciona un fotograma en la línea de tiempo, haz clic derecho y elige “Insertar fotograma clave”.
- Los fotogramas clave permiten definir cambios importantes, como el inicio o fin de una animación.



Creando en una interpolación clásica



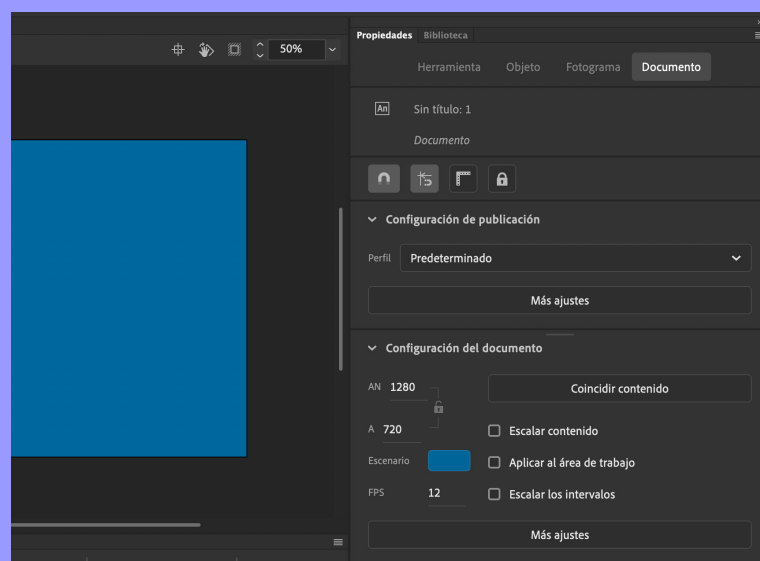
Usando la interpolación clásica para animar un botón por escala

3. Organización de Contenido en Fotogramas

- Asigna un nombre descriptivo a cada capa (ej. “Fondo”, “Botones”, “Animación”).
- Distribuye los elementos de la app en diferentes capas para mantener un flujo de trabajo ordenado.
- Utiliza el panel de propiedades para ajustar la duración de los fotogramas según la velocidad de la animación.

4. Configuración del Escenario

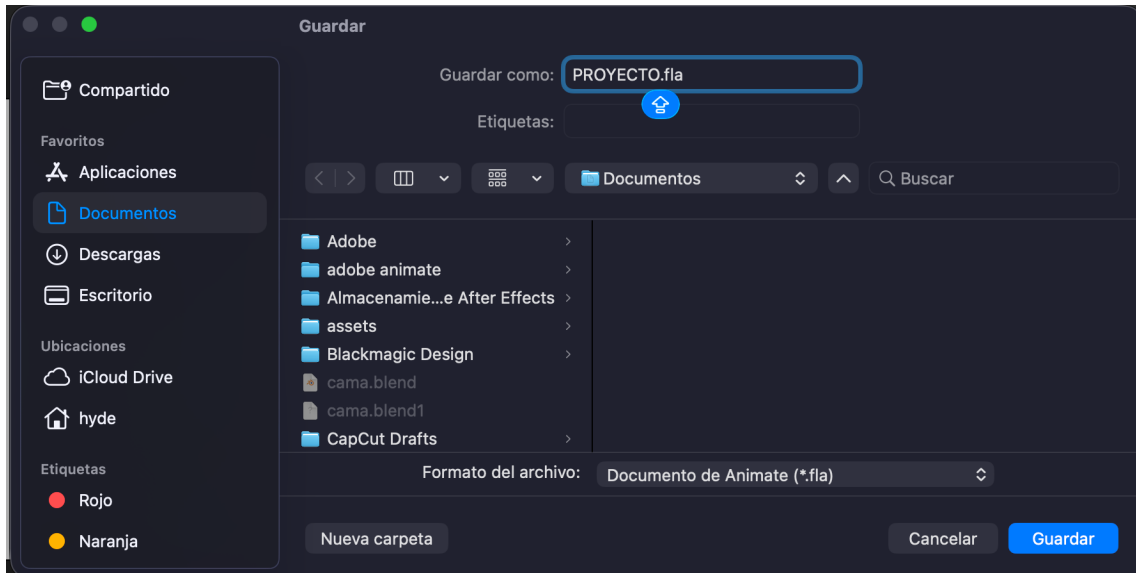
- Ajusta el color de fondo desde el panel de Propiedades.
- Asegúrate de que la opción de escala de contenido esté desactivada para mantener el tamaño fijo en dispositivos móviles.



Panel de propiedades donde puedes ajustar tamaño y color de fondo

5. Guardar el Proyecto

- Guarda tu progreso desde del menu: Archivo > Guardar Como.
- Nómbralo como “MiApp_Proyecto” para mantener un orden en tus archivos.



Ventana para guardado de proyecto

Es recomendable guardar tus proyectos en Adobe Animate por versiones, ya que trabajar con múltiples elementos visuales y código puede generar errores o hacer difícil revertir cambios. Para evitar pérdida de trabajo, guarda copias incrementales con nombres como Proyecto_V1.fla, Proyecto_V2.fla, permitiéndote regresar a una versión anterior si los cambios recientes no funcionan como esperabas.

Este método facilita el control del proceso, ayuda a identificar errores y monitorear cambios sin comprometer el archivo principal. También es recomendable complementar esta práctica con respaldos en la nube o en unidades externas para mayor seguridad.

Uso del Código 'stop' para Controlar la Navegación

El comando `stop();` en ActionScript 3.0 es esencial para evitar que la animación avance automáticamente de un fotograma a otro, permitiéndote controlar la navegación de manera más precisa.

1. Aplicación del Código 'stop'

- Selecciona el fotograma clave donde deseas detener la animación.
- Abre el panel de Acciones (F9).
- Escribe el siguiente código: `stop();`

Este simple código detendrá la reproducción en el fotograma actual hasta que se active una acción (por ejemplo, al hacer clic en un botón).

2. Ejemplo Práctico

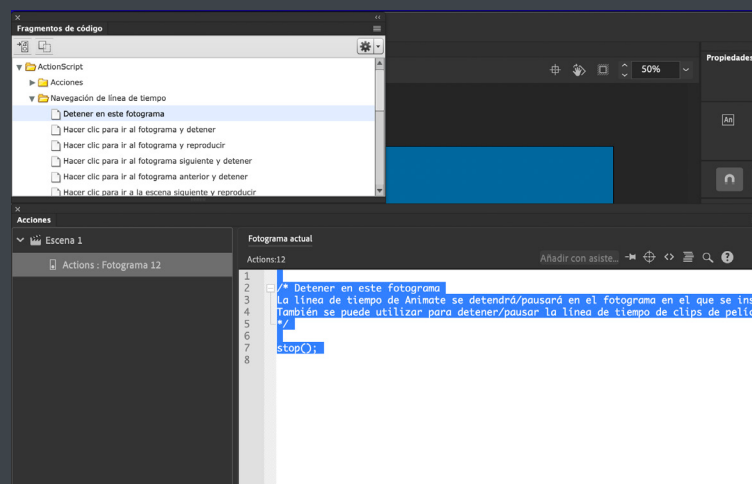
Si tienes una app con varias pantallas:

- En cada fotograma clave principal, inserta el código `stop()`;
- Luego, crea botones para navegar entre los fotogramas, utilizando el siguiente código para el botón:

```
boton_siguiente.addEventListener(MouseEvent.CLICK, irASiguiente);
```

```
function irASiguiente(event:MouseEvent):void {  
    gotoAndStop(2);  
}
```

Este código indica que al hacer clic en el botón llamado “boton_siguiente”, la animación saltará al fotograma 2 y se detendrá allí.



Panel de Acciones con el código `stop()` y el código de navegación básica

5. Creación de Movie Clips para Animación

1. ¿Qué es un Movie Clip?

Los movie clips son símbolos en Adobe Animate que permiten crear animaciones internas y reutilizables dentro de tu proyecto. Son fundamentales para el desarrollo de apps interactivas, ya que facilitan la organización de los elementos gráficos y su programación.

Un movie clip es un tipo de símbolo que tiene su propia línea de tiempo independiente. Esto permite que puedas animar elementos de forma aislada sin afectar la línea de tiempo principal de tu proyecto.

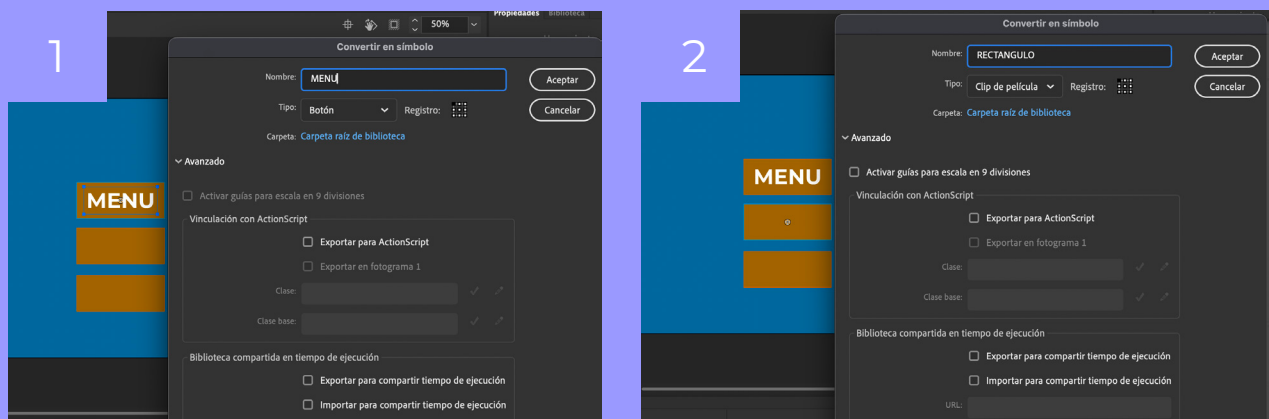
Los movie clips son ideales para:

- Animaciones repetitivas o en bucle.
- Efectos visuales como destellos, parpadeos o movimientos continuos.
- Elementos interactivos que requieren programación específica (botones animados, banners, iconos).

2. Creación de un Movie Clip

Para crear un movie clip en Adobe Animate, sigue estos pasos:

1. Dibuja o importa un gráfico en el escenario que quieras convertir en movie clip.
2. Selecciona el gráfico con la herramienta de selección (V).
3. Haz clic derecho sobre el gráfico y elige Convertir en símbolo (o presiona F8).
4. En la ventana emergente:
Asigna un nombre descriptivo (ej. "AnimacionBoton").
Selecciona la opción Movie Clip.
Haz clic en Aceptar.



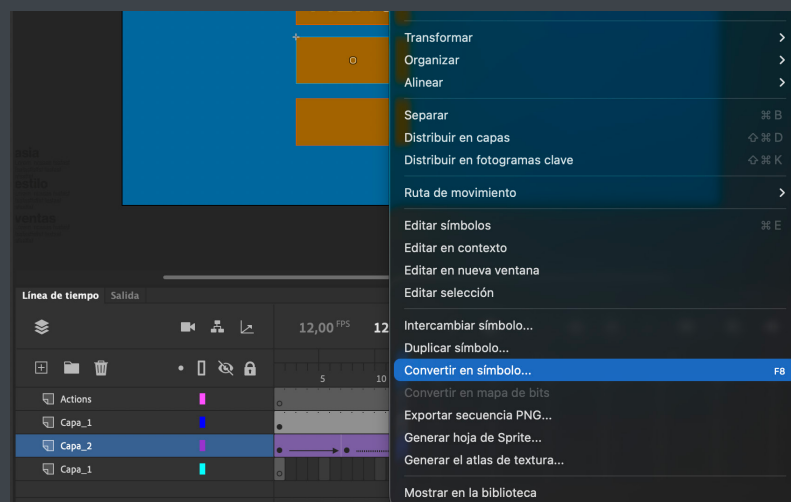
Ventana de creación de símbolo (Clip de película o Botón)

Ahora verás que el gráfico se ha convertido en un símbolo, con un contorno azul que indica que es un movie clip.

3. Edición de un Movie Clip

Una vez creado el movie clip, puedes agregarle animaciones internas:

1. Haz doble clic sobre el movie clip para entrar en su línea de tiempo interna.
2. Verás una nueva línea de tiempo específica para el movie clip, separada de la línea de tiempo principal.
3. Agrega fotogramas clave, interpolaciones de movimiento o cualquier otro tipo de animación.
4. Para salir del modo de edición, haz clic en el botón de navegación ubicado en la parte superior izquierda (breadcrumb).

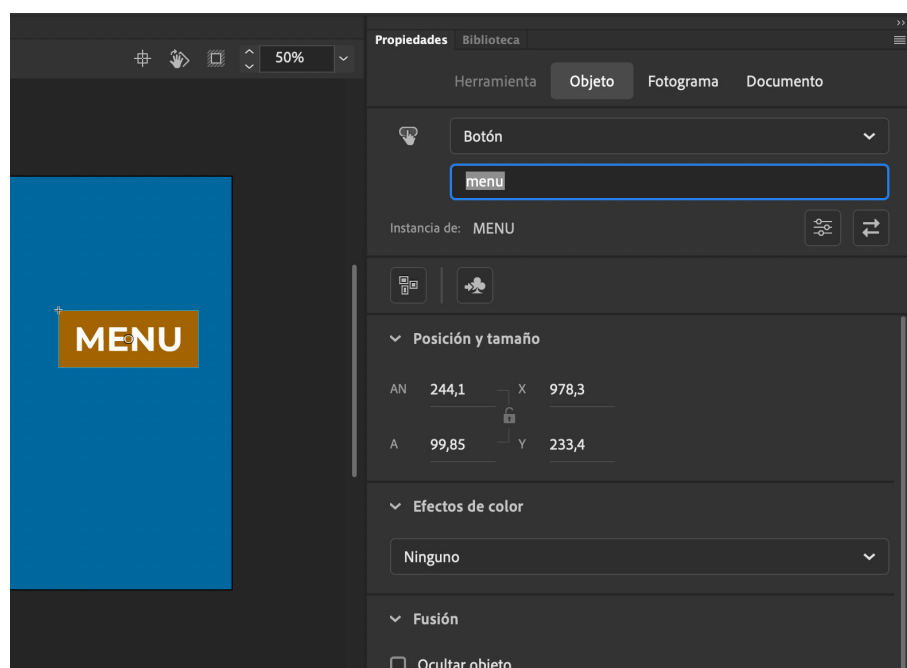


Convirtiendo en símbolo

4. Propiedades de los Movie Clips

Selecciona el movie clip y abre el panel de Propiedades (Ctrl + F3) para ver y modificar:

- Nombre de instancia: Importante para programar con ActionScript.
- Transformaciones: Cambiar tamaño, rotación, opacidad.
- Filtros: Aplicar efectos como sombras, resplandores o desenfoques.

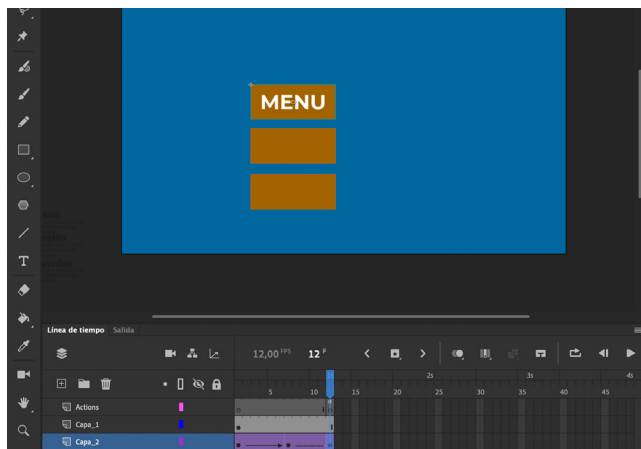


Convirtiendo en botón y agregando su descripción

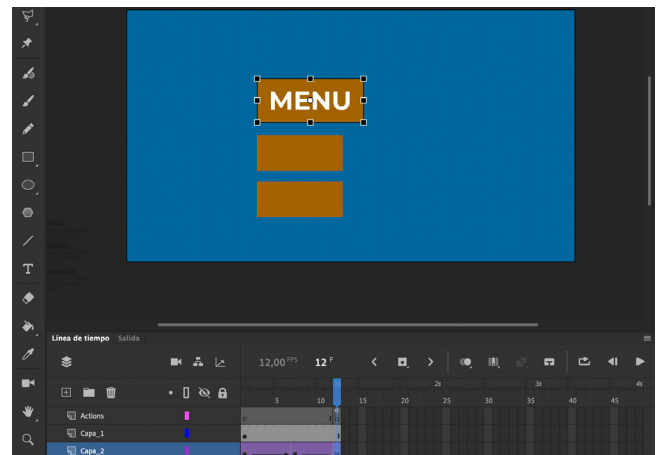
5. Ejemplo Práctico: Animación de un Botón

Vamos a crear una animación simple para un botón interactivo:

1. Dibuja un botón simple en el escenario.
2. Conviértelo en un movie clip (F8).
3. Dale un nombre de instancia en el panel de propiedades, por ejemplo: botonAnimado.
4. Haz doble clic para entrar en la línea de tiempo del movie clip.
5. Inserta varios fotogramas clave (F6) y crea una interpolación clásica para simular un efecto de “latido” (aumentar y disminuir el tamaño).
6. Vuelve a la línea de tiempo principal.



Fotograma con el tamaño normal del botón



Fotograma con el botón escalado

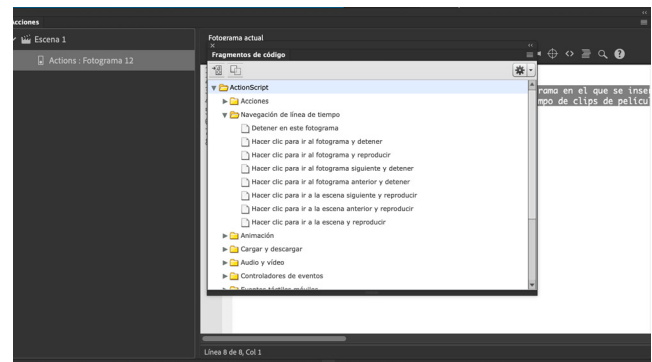
Proceso de animación del botón dentro del movie clip (Botón)
Ahora el botón tiene una animación interna que se repetirá automáticamente cada vez que la app esté en ejecución.

6. Programación en ActionScript 3.0

1. Introducción a ActionScript 3.0

AS3 está basado en la programación orientada a objetos (POO), lo que facilita la reutilización de código y la creación de estructuras más organizadas.

ActionScript 3.0 (AS3) es el lenguaje de programación utilizado en Adobe Animate para agregar interactividad a las aplicaciones. Permite controlar animaciones, gestionar eventos de usuario, y crear aplicaciones más dinámicas.



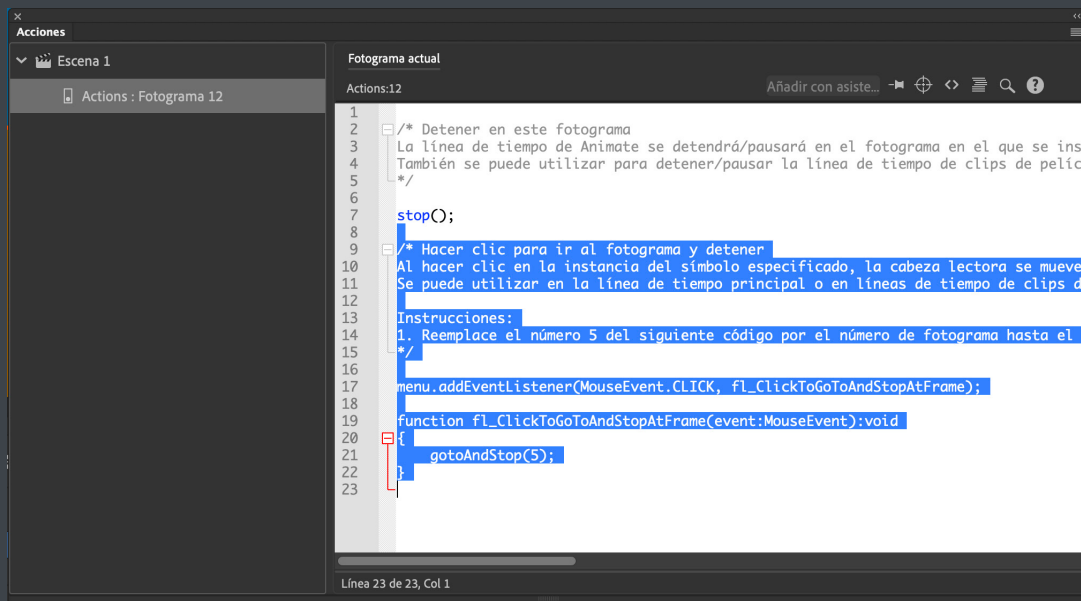
Panel de acciones (Comando rápido "F9")

Conceptos básicos:

- **Eventos:** detectan acciones del usuario (clics, movimientos, teclas presionadas).
- **Funciones:** bloques de código que realizan tareas específicas.
- **Instancias:** objetos en el escenario que se pueden controlar mediante programación.

2. Configuración del Entorno de Programación

1. Selecciona un fotograma en la línea de tiempo principal.
2. Abre el panel de Acciones (F9).
3. Aquí es donde escribirás el código AS3.



Pantalla del panel "Acciones"

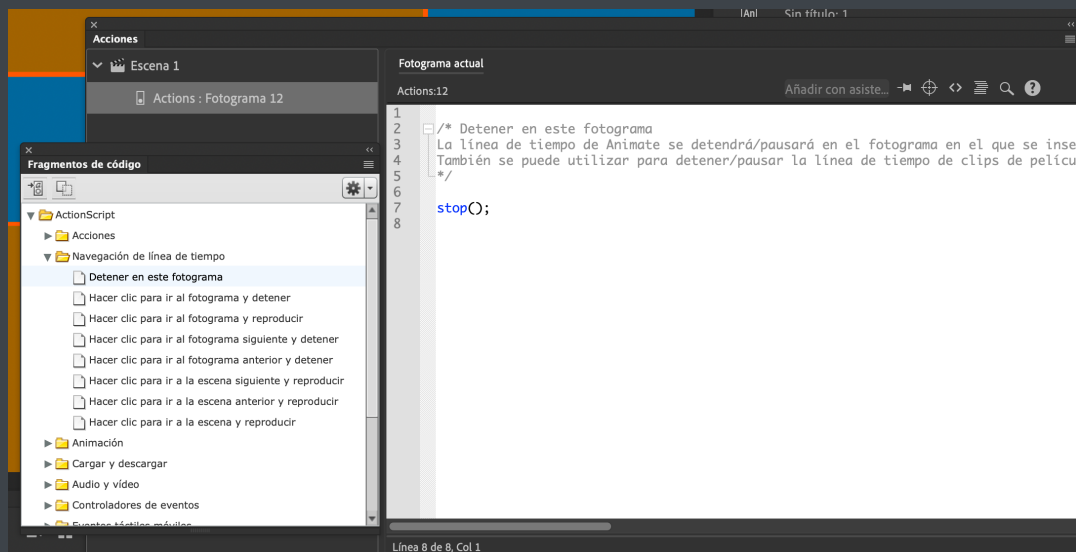
3. Uso del Código 'stop()' para Controlar la Reproducción

Para evitar que la animación se reproduzca infinitamente:

`stop();`

Este código detiene la reproducción en el fotograma donde se coloca.

1. Selecciona el primer fotograma de la línea de tiempo.
2. Abre el panel de Acciones (F9).
3. Escribe `stop();`.
4. Guarda el archivo (Ctrl + S).



Incorporando el código `stop()` en el panel de acciones.

4. Programación de Botones con ActionScript 3.0

Para enlazar fotogramas mediante botones:

1. Crea un botón y dale un nombre de instancia en el panel de propiedades, por ejemplo: botonIr.
2. Selecciona el fotograma donde quieres agregar el código y abre el panel de acciones (F9).

Código para Navegar entre Fotogramas:

```
botonIr.addEventListener(MouseEvent.CLICK, irAFotograma);
```

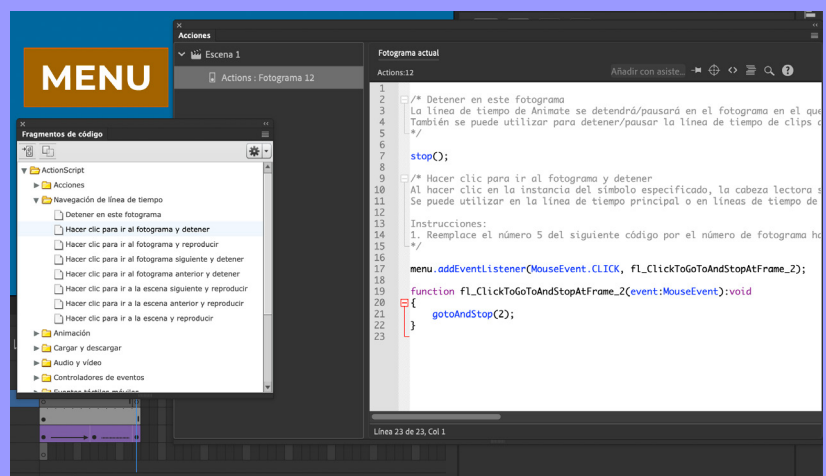
```
function irAFotograma(event:MouseEvent):void {
```

```
gotoAndStop(5); // Cambia el 5 por el número de fotograma deseado }
```

Explicación del código:

- `addEventListener`: Detecta el clic del usuario.
- `MouseEvent.CLICK`: Evento que se dispara al hacer clic.
- `gotoAndStop(2)`: Navega al fotograma número 2 y se detiene ahí.

Pantalla del botón en el escenario y el código en el panel de acciones



5. Ejemplo Práctico: App Interactiva Básica

Estructura de la app:

- Pantalla de inicio (fotograma 1): Botón para iniciar.
- Pantalla de contenido (fotograma 2): Información o animación.
- Pantalla final (fotograma 3): Botón para regresar al inicio.

Pasos para Programar la Navegación:

1. Crea tres fotogramas clave (F6) en la línea de tiempo.
2. En el primer fotograma, agrega un botón btnInicio.
3. En el segundo, muestra el contenido.
4. En el tercero, agrega un botón btnRegresar.

Código para la Navegación:

```
// Detener la reproducción automática  
stop();  
  
// Botón para ir al contenido  
btnInicio.addEventListener(MouseEvent.CLICK, irAlContenido);  
function irAlContenido(event:MouseEvent):void{gotoAndStop(2); }
```

```
// Botón para regresar al inicio
btnRegresar.addEventListener(MouseEvent.CLICK,
regresarInicio);
function regresarInicio(event:MouseEvent):void {
    gotoAndStop(1);
}
```

Botón 1



Botón Volver



Los códigos de AS3 deben ser colocados correctamente, de lo contrario, generarán errores.

Al insertar el fragmento de código puedes modificar lo necesario en este caso cambiando únicamente el número de fotograma.

6. Organización del Contenido en Fotogramas Enlazados

Para estructurar el contenido de forma eficiente en Adobe Animate:

1. Crea múltiples capas en la línea de tiempo para separar elementos (botones, texto, animaciones).
2. Añade fotogramas clave (F6) para dividir las secciones de la app.
3. Nombrar los fotogramas clave usando etiquetas (clic derecho en el fotograma → Insertar etiqueta).

Código para Navegar Usando Etiquetas:

```
btnMenu.addEventListener(MouseEvent.CLICK, irAMenu);
```

```
function irAMenu(event:MouseEvent):void {
```

```
    gotoAndStop("menu"); // "menu"  
    es la etiqueta del fotograma  
}
```

```
btnInfo.addEventListener(MouseEvent.CLICK, irAInfo);
```

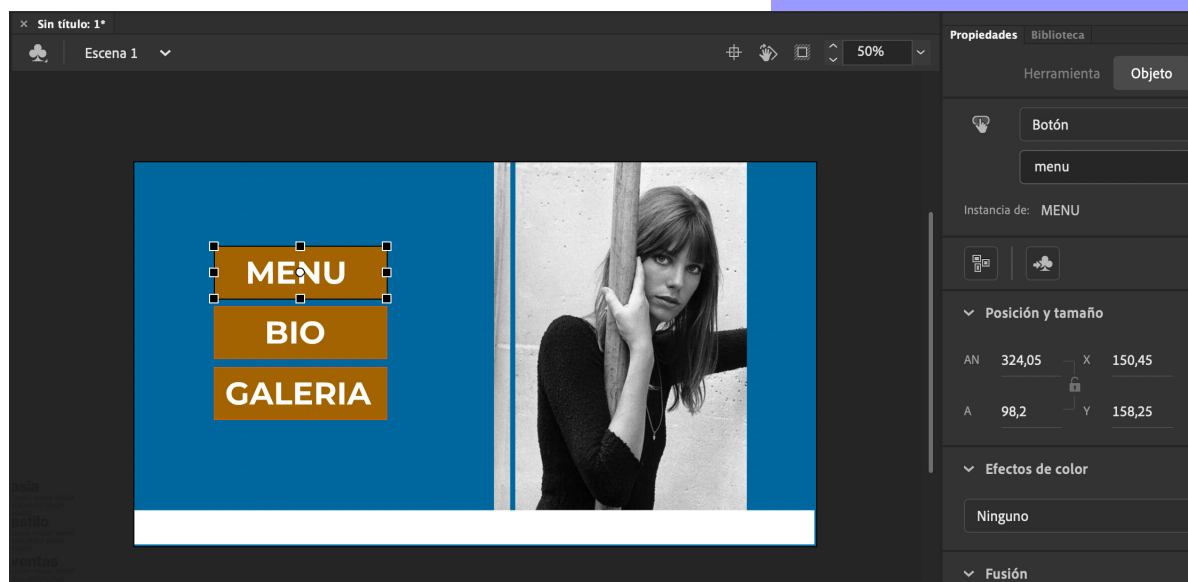
```
function irAInfo(event:MouseEvent):void {
```

```
    gotoAndStop("informacion"); // "in-  
    formacion"  
    es la etiqueta del fotograma  
}
```

Consejos:

- Usar etiquetas facilita la navegación sin depender de números de fotogramas.
- Organiza los elementos visuales por capas para mayor claridad.

Las etiquetas facilitan la identificación de los elementos, ayudan a identificar el tipo de símbolo que se está utilizando como por ejemplo: un clip de película, un botón o un gráfico.



Etiquetas en la línea de tiempo y el código

Esta información se encuentra ubicada en el panel de propiedades, además de etiquetas también puedes encontrar información como la ubicación o efectos que puedes utilizar.

7. Manipulación de Movie Clips para Animaciones Interactivas

Los Movie Clips permiten crear animaciones reutilizables e interactivas dentro de la app.

Cómo Crear un Movie Clip:

1. Dibuja un objeto en el escenario.
2. Haz clic derecho y selecciona Convertir en símbolo (F8).
3. Elige el tipo “Movie Clip” y asígnale un nombre de instancia, por ejemplo: animacionMC.

Código para Controlar un Movie Clip:

// Reproducir la animación al hacer clic

```
animacionMC.addEventListener(MouseEvent.CLICK,  
reproducirAnimacion);
```

```
function reproducirAnimacion(event:MouseEvent):void {  
    animacionMC.play();  
}
```

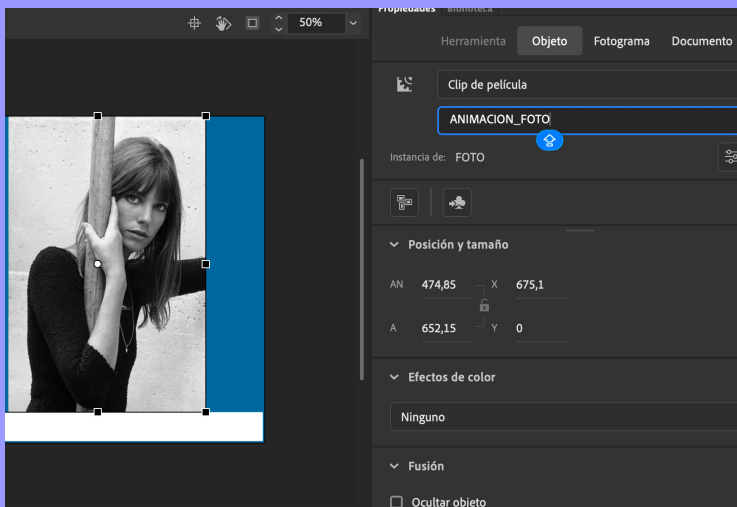
// Detener la animación

```
animacionMC.addEventListener(MouseEvent.MOUSE_OVER,  
detenerAnimacion);
```

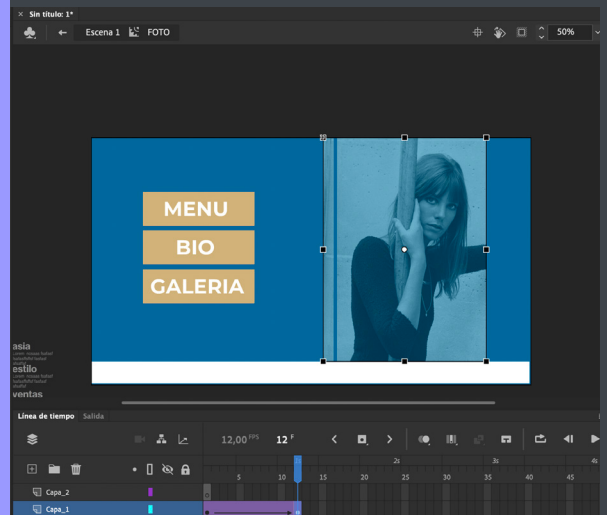
```
function detenerAnimacion(event:MouseEvent):void {  
    animacionMC.stop();  
}
```

Consejos:

- Los Movie Clips pueden contener su propia línea de tiempo.
- Puedes anidar Movie Clips para crear efectos complejos.



Colocando un nombre a la instancia



Es posible utilizar varias animaciones en un movie clip, por lo tanto es posible crear varias a modo de capas, una sobre otra.

8. Optimización del Código y Buenas Prácticas

Para mejorar el rendimiento y la organización de la app:

- Usa nombres de instancia claros y descriptivos.
- Agrupa funciones relacionadas en bloques para facilitar la lectura.
- Elimina sonidos innecesarios o silenciar todos los sonidos cuando ya no se usen:
- Minimiza el uso excesivo de animaciones automáticas. Usa stop() y controla la reproducción con eventos.
- Documenta tu código con comentarios para entenderlo fácilmente:

```
function fl_ClickToStopASounds(e-  
vent:MouseEvent):void
```

```
// Este botón silencia todos los sonidos
```

```
button_1.addEventListener(MouseEvent.CLICK, fl_ClickToStopAllSounds);
```

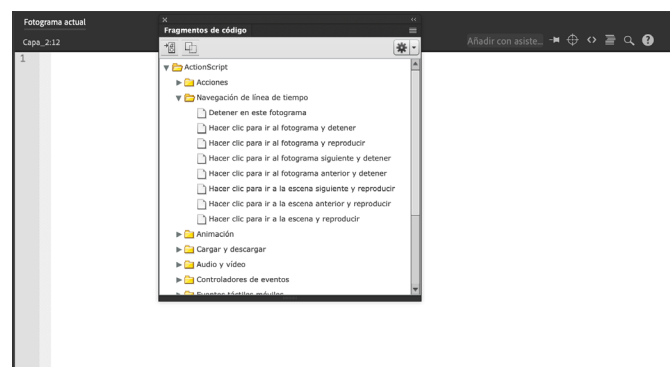
```
function fl_ClickToStopAllSounds(event:MouseEvent):void{
```

```
SoundMixer.stopAll();}
```

```
function detenerAnimacion(event:MouseEvent):void {
```

```
animacionMC.stop();
```

```
}
```



Usando los fragmentos de código

7. Interactividad entre botones y contenido

Para garantizar una estructura sólida en tu aplicación, es fundamental organizar el contenido en fotogramas de manera eficiente. Esto no solo facilitará una navegación fluida, sino que también mejorará la experiencia del usuario, haciéndola más intuitiva y agradable.

Los botones serán clave para permitir la navegación entre los fotogramas, donde cada uno contendrá información o funcionalidades específicas. Por ello, es crucial distribuir y organizar el contenido de manera lógica y ordenada en los fotogramas.

En caso de que existan subcontenidos (como animaciones o elementos interactivos), estos deben colocarse dentro de movieclips. Un movieclip es un contenedor que tiene su propia línea de tiempo con fotogramas, lo que te permite agregar contenido adicional, como animaciones, botones o más información, sin afectar la estructura principal de la aplicación.

Consejos adicionales:

- 1. Planifica la estructura:** Antes de comenzar, define cómo se organizará el contenido en los fotogramas y movieclips.
- 2. Mantén la coherencia:** Asegúrate de que la navegación sea consistente en toda la aplicación.
- 3. Optimiza el uso de movieclips:** Utilízalos para agrupar contenido relacionado y simplificar la gestión de animaciones o interacciones complejas.

Uso de Capas y Etiquetas

1. Crea capas separadas para cada tipo de contenido (botones, texto, imágenes, código, etc.).
2. Etiqueta los fotogramas clave para identificarlos fácilmente en el código.
3. Agrupa elementos relacionados en una misma capa para mantener la organización.

Ejemplo de etiquetas en la línea de tiempo:

- Inicio (fotograma 1)
- Contenido (fotograma 2)
- Final (fotograma 3)

Esto facilita la navegación usando `gotoAndStop("Inicio");` en lugar de números de fotograma.

```
function detenerAnimacion(event:MouseEvent):void {  
    animacionMC.stop();  
}
```

8. Enlazando Contenido a Través de Botones

Para mejorar la experiencia del usuario, se pueden enlazar diferentes secciones de la app a través de botones utilizando ActionScript 3.0. Es recomendable utilizar nombres de instancia descriptivos para facilitar su manejo en el código.

Ejemplo:

```
botonSeccion1.addEventListener(MouseEvent.CLICK, irASeccion1);  
function irASeccion1(event:MouseEvent):void {  
    gotoAndStop("Seccion1");  
}
```

Uso de Movie Clips para Animaciones y Elementos Interactivos

Cómo Crear y Controlar un Movie Clip

1. Crea un nuevo movie clip (Insertar > Símbolo > Tipo: Movie Clip).
2. Asígnale un nombre de instancia, por ejemplo, animacion_mc.
3. Añade una animación dentro del movie clip.
4. Controla la animación con ActionScript 3.0:
animacion_mc.stop(); // Detiene la animación al inicio

```
botonPlay.addEventListener(MouseEvent.CLICK, iniciarAnimacion);  
  
function iniciarAnimacion(event:MouseEvent):void {  
    animacion_mc.play(); // Reproduce la animación  
}  
  
botonStop.addEventListener(MouseEvent.CLICK, detenerAnimacion);  
  
function detenerAnimacion(event:MouseEvent):void {  
    animacion_mc.stop(); // Detiene la animación  
}
```

9. Pruebas y Exportación de la app

Antes de finalizar, es importante probar la aplicación para detectar errores y verificar su funcionamiento. Para exportarla:

1. Ir a Archivo > Configuración de publicación.
2. Elegir el formato deseado (HTML5, SWF, APK, etc.).
3. Ajustar las configuraciones según la plataforma de destino.
4. Exportar y probar en diferentes dispositivos.

Productos que puedes exportar desde Adobe Animate y sus extensiones.

AIR

1. Archivo SWF (.swf)

- Descripción: Formato nativo de Adobe Animate, utilizado para animaciones interactivas y contenido multimedia en la web.

Usos:

- Creación de banners animados para sitios web.
- Desarrollo de juegos simples o interactivos.
- Integración en páginas web mediante reproductores compatibles con Flash (aunque este formato ha perdido popularidad debido al declive de Flash Player).
- Aplicaciones: Sitios web, presentaciones interactivas, publicidad digital.

2. Archivo HTML5 Canvas (.html)

- Descripción: Exporta la animación como un archivo HTML5, compatible con navegadores modernos sin necesidad de plugins adicionales.

Usos:

- Animaciones y contenido interactivo para páginas web.
- Juegos simples o aplicaciones web ligeras.
- Presentaciones multimedia en línea.
- Aplicaciones: Desarrollo web, e-learning, publicidad digital.

3. Archivo WebGL (.html)

Descripción: formato basado en tecnología WebGL, ideal para animaciones 2D y 3D con alto rendimiento en navegadores.

Usos:

- Animaciones complejas y gráficos 3D interactivos.
- Juegos en línea con gráficos avanzados.
- Visualizaciones interactivas para proyectos creativos.
- Aplicaciones: juegos web, visualización de datos, proyectos artísticos.

4. Archivo EXE (.exe)

Descripción: Exporta la animación como un archivo ejecutable para Windows.

Usos:

- Creación de aplicaciones independientes para Windows.
- Presentaciones offline o demostraciones interactivas.
- Juegos o aplicaciones que no requieren instalación.
- Aplicaciones: presentaciones empresariales, aplicaciones educativas, prototipos de software.

5. Archivo APP (.app)

Descripción: Exporta la animación como una aplicación independiente para macOS.

Usos:

- Desarrollo de aplicaciones nativas para Mac.
- Presentaciones o proyectos interactivos para usuarios de Apple.
- Aplicaciones: proyectos creativos audiovisuales, aplicaciones educativas, demostraciones offline.

6. Archivo APK (.apk)

Descripción: Exporta la animación como una aplicación Android instalable.

Usos:

- Creación de aplicaciones móviles simples para Android.
- Juegos o aplicaciones interactivas para dispositivos móviles.
- Aplicaciones: desarrollo de apps móviles, juegos para Android, herramientas educativas.

7. Archivo GIF (.gif)

Descripción: Exporta la animación como un archivo GIF animado.

Usos:

- Creación de animaciones cortas para redes sociales o sitios web.
- Banners publicitarios o elementos visuales ligeros.
- Aplicaciones: redes sociales, marketing digital, blogs.

8. Archivo de Video (.mp4, .mov)

Descripción: Exporta la animación como un archivo de video en formatos como MP4 o MOV.

Usos:

- Creación de videos promocionales o explicativos.
- Exportación de animaciones para plataformas de video (YouTube, Vimeo).
- Integración en presentaciones o proyectos multimedia.
- Aplicaciones: marketing digital, e-learning, redes sociales.

9. Archivo AIR (.air)

Descripción: Formato para crear aplicaciones multiplataforma utilizando Adobe AIR.

Usos:

- Desarrollo de aplicaciones para Windows, macOS, Android e iOS.
- Creación de aplicaciones ricas en multimedia e interactividad.
- Aplicaciones: empresariales, juegos, herramientas educativas.

10. Archivo OAM (.oam)

Descripción: formato de paquete utilizado para exportar animaciones y contenido interactivo creado en Adobe Animate, especialmente diseñado para su uso en plataformas como Adobe Captivate o Dreamweaver.

Usos:

- Integración de animaciones interactivas en proyectos de e-learning.
- Contenido multimedia en sitios web creados con Dreamweaver.
- Exportación de widgets o componentes reutilizables para otros proyectos.

11. Archivo IPA (.ipa)

Descripción: formato de paquete utilizado para crear aplicaciones instalables en dispositivos iOS (iPhone, iPad). Este formato se genera a través de Adobe AIR o herramientas adicionales como Xcode.c

Usos:

- Desarrollo de aplicaciones nativas para iOS.
- Creación de juegos o aplicaciones interactivas para iPhone y iPad.
- Publicación en la App Store de Apple.
- Cursos interactivos en Adobe Captivate.

Desarrollo web: uso en Dreamweaver para enriquecer páginas web con animaciones.

Prototipado: exportación de elementos interactivos para pruebas o demostraciones.

Aplicaciones:

Aplicaciones móviles: Juegos, herramientas educativas, aplicaciones empresariales.

Prototipos: Pruebas de aplicaciones en dispositivos iOS.

Proceso para exportar a iPhone (iOS):

Usar Adobe AIR:

Configura el proyecto en Adobe Animate para exportar como AIR para iOS.

Genera un archivo .ipa que puede ser probado en dispositivos iOS o subido a la App Store.

Integración con Xcode:

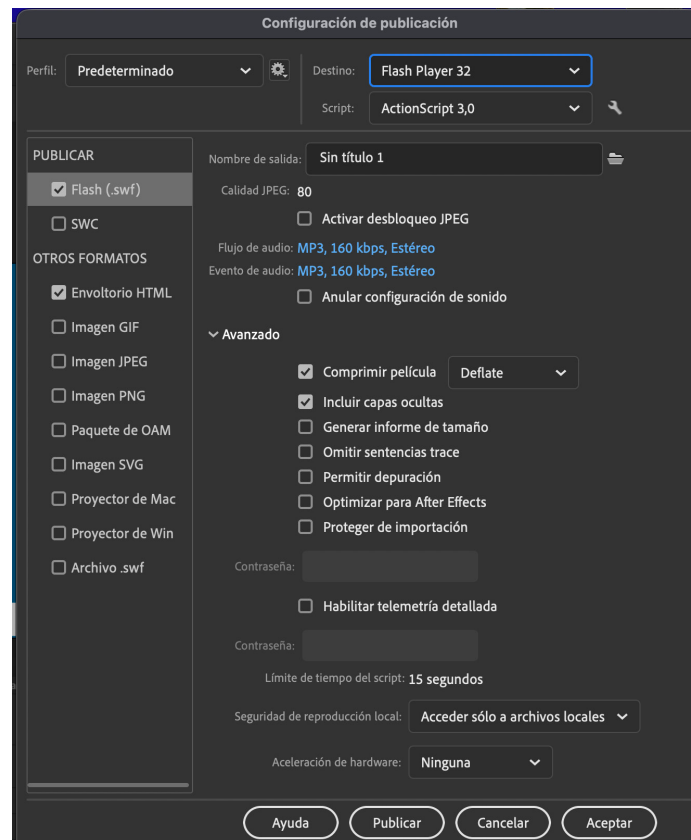
Exporta el proyecto desde Adobe Animate y ábrelo en Xcode (herramienta de desarrollo de Apple).

Ajusta los detalles de la aplicación (iconos, permisos, etc.) y compila el archivo .ipa final.

Pruebas en dispositivos reales:

Utiliza un certificado de desarrollador de Apple para probar la aplicación en dispositivos iOS antes de su publicación.

Recuerda que para hacer pruebas en un dispositivo Apple requerirá de la compra de una licencia. Para mas informacion: <https://developer.apple.com>



Panel de exportación de archivos

Para mejorar la experiencia del usuario, se pueden enlazar diferentes secciones de la app a través de botones utilizando ActionScript 3.0. Es recomendable utilizar nombres de instancia descriptivos para facilitar su manejo en el código.

Ejemplo:

```

botonSeccion1.addEventListener(MouseEvent.CLICK, irASeccion1);
function irASeccion1(event:MouseEvent):void {
    gotoAndStop("Seccion1");
}

```

10. Errores

Adobe Animate es una herramienta poderosa para la creación de contenido interactivo, pero un mal uso de su código o configuración puede generar errores que afectan el rendimiento y la funcionalidad de una aplicación. A continuación, se describen algunos errores frecuentes, sus causas y sus efectos en el desarrollo.

1. Errores en ActionScript 3.0 por Código Incorrecto

Uno de los errores más comunes al programar en ActionScript 3.0 es escribir mal el código o usar una sintaxis incorrecta. Por ejemplo, olvidar un punto y coma (;), escribir mal el nombre de una variable o usar una función inexistente. Esto provoca que el script no se ejecute, generando mensajes de error como “1120: Access of undefined property”.

Ejemplo de error:

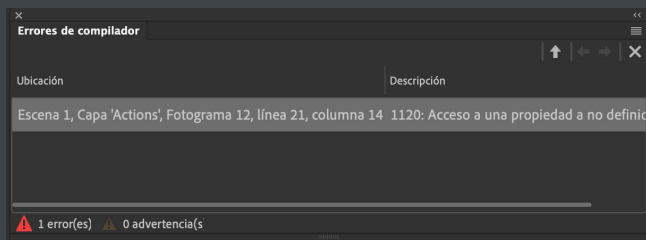
Si se escribe `gotoandSto(5);` en lugar de `gotoAndStop(5);`, Adobe Animate mostrará un error porque la función está mal escrita.

2. Bucles Infinitos y Consumo Excesivo de Memoria RAM

Otro problema crítico ocurre cuando un elemento dentro de la animación entra en un bucle infinito, repitiéndose sin control. Esto puede suceder cuando no se usa correctamente el código `stop();` o cuando una animación de MovieClip no está programada para detenerse en un punto específico.

Ejemplo de error:

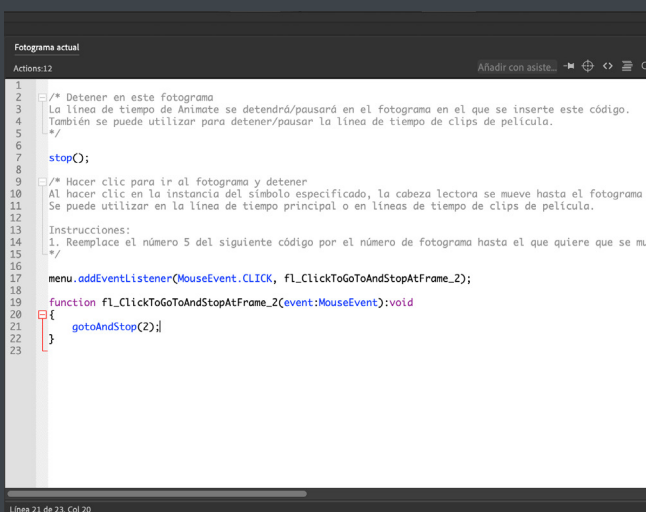
Si un clip de película contiene una animación en bucle y no se le agrega `stop();`, continuará ejecutándose indefinidamente, lo que puede sobrecargar la memoria RAM y ralentizar



Alerta de error



Con error



Corregido

la aplicación. En dispositivos con menos recursos, esto puede hacer que la aplicación se cierre inesperadamente.

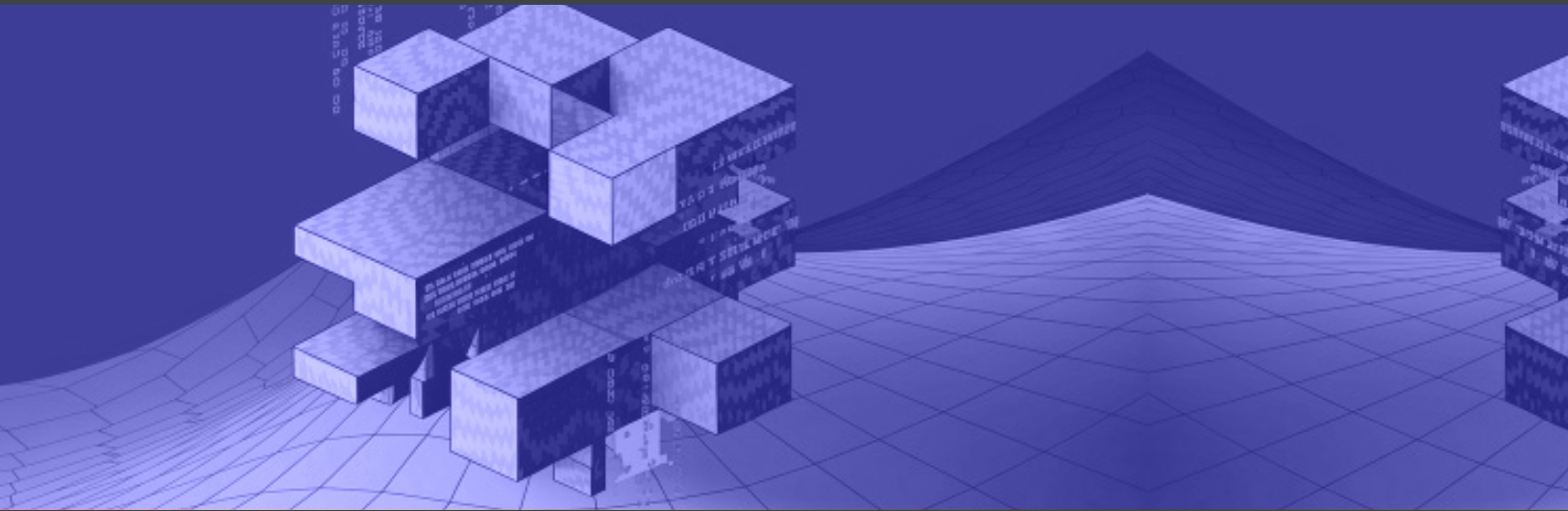
3. Carga Incorrecta de Archivos Externos

Cuando se trabaja con imágenes, videos o sonidos externos, un error frecuente es no verificar la existencia del archivo antes de intentar cargarlo. Si la ruta del archivo es incorrecta o el archivo no existe, la aplicación puede fallar al ejecutarse.

Ejemplo de error:

Si se usa `var imagen:Loader = new Loader(); imagen.load(new URLRequest("imagen.png"));` sin comprobar que `imagen.png` realmente existe, la aplicación generará un error en tiempo de ejecución.

11. Glosario



1. Adobe Animate: Software de animación y diseño interactivo para crear contenidos multimedia, juegos, series de TV y publicidad web. Permite dibujar, animar y agregar interactividad desde una misma plataforma

2. ActionScript 3.0 (AS3): Lenguaje de script basado en ECMAScript usado en Animate para añadir interactividad, control de reproducción y mostrar datos en tiempo de ejecución.

3. Escenario (Stage): Área principal donde se colocan y visualizan elementos gráficos y animaciones; sirve como lienzo de salida del proyecto.

4. Línea de Tiempo (Timeline): Herramienta visual que organiza capas y fotogramas para controlar el flujo temporal de la animación.

5. Fotograma Clave (Keyframe): Punto en el timeline donde comienza una animación o se aplica un cambio visual; permite definir estados iniciales o finales.

6. Símbolo: Objeto reutilizable (gráfico, clip o botón) creado desde el Stage que se almacena en la biblioteca y puede instanciarse múltiples veces.

7. Movie Clip: Tipo de símbolo con línea de tiempo propia y controlable vía ActionScript durante la ejecución.



8. Botón (Button Symbol): símbolo interactivo con cuatro estados (Up, Over, Down, Hit) que responde a clics o toques.

9. Código Stop(): función en AS3 que detiene la reproducción del timeline en el fotograma donde se coloca.

10. Eventos (Events): acciones del usuario o del sistema (como clics o teclas) que el programa detecta para disparar respuestas.

11. Listeners (Escuchadores de Eventos): Código que escucha eventos específicos usando `addEventListener()`, ejecutando una función al dispararse el evento.

12. GotoAndStop(): método en AS3 que posiciona el playhead en un fotograma o etiqueta especificada y detiene la reproducción.

13. GotoAndPlay(): método en AS3 que mueve el playhead a un fotograma o etiqueta indicada y continúa la reproducción automáticamente.

14. Navegación entre fotogramas: técnica que genera automáticamente fotogramas intermedios entre dos estados usando propiedades como posición, escala o rotación.

15. Interpolación de Movimiento (Motion Tween): técnica para animar la posición, escala o rotación de un objeto en un rango de fotogramas.

16. Interpolación de Forma (Shape Tween): proceso que genera transformaciones suaves de formas vectoriales entre dos fotogramas clave.

17. Punto de Registro (Registration Point): marco de referencia interno (coordenada 0,0) usado para rotación y escala, definido al crear el símbolo.

18. Propiedades (Properties

Panel): panel donde se ajustan atributos como tamaño, posición y color de los elementos.

19. Biblioteca (Library): almacén de símbolos, gráficos y elementos reutilizables dentro del proyecto.

20. Publicación (Publishing): proceso de exportar la aplicación en formato SWF, HTML5 o APK para dispositivos móviles.

21. Código de Navegación: fragmentos de AS3 que permiten moverse entre pantallas o secciones, usando botones y eventos.

22. Responsividad: adaptación de la app a diferentes tamaños de pantalla en móviles, tablets y aplicaciones antes de su publicación.

23. Variables: elementos que almacenan datos y pueden ser modificados en el código, como `var nombre:String = "Usuario";`.

24. Funciones (Functions): bloques de código reutilizables que ejecutan tareas específicas, como cambiar de pantalla o reproducir un sonido.

25. Carga de Imágenes Externas: uso de ActionScript para importar imágenes externas en tiempo de ejecución.

26. Sonido en Animate: inserción y control de audio con ActionScript (`sound.play();`).

27. Optimización de Recursos: técnicas para reducir el tamaño del archivo final y mejorar el rendimiento de la app.

28. Depuración (Debugging): proceso de prueba y corrección de errores en la aplicación antes de su publicación.



Códigos más utilizados

1. **Detener la reproducción en un fotograma:**
`stop();`
2. **Ir a un fotograma específico y reproducir:**
`gotoAndPlay(10);` o `gotoAndPlay("inicio");`
3. **Ir a un fotograma específico y detener:**
`gotoAndStop(5);` o `gotoAndStop("menu");`
4. **Capturar eventos (ejemplo de clic en un botón):**
`boton.addEventListener(MouseEvent.CLICK, irAPantalla);` function
`irAPantalla(event:MouseEvent):void { gotoAndStop("pantalla2"); }`
5. **Controlar un MovieClip (reproducir, detener, ir a fotograma):**
`clip_mc.play(); clip_mc.stop(); clip_mc.gotoAndPlay(5);`
`clip_mc.gotoAndStop("final");`
6. **Detener todos los sonidos en la aplicación:**
`import flash.media.SoundMixer; SoundMixer.stopAll();`

12. Bibliografía

Adobe Inc. (2023). How to create buttons with Animate. <https://helpx.adobe.com/animate/using/creating-buttons.html>
Citado en: p. 50 del manual

Adobe Inc. (2023). How to use ActionScript with Animate. <https://helpx.adobe.com/animate/using/actionscript.html>
Citado en: p. 50 del manual

Adobe Inc. (2023). Motion tween and ActionScript 3.0. <https://helpx.adobe.com/animate/using/animation-motion-tween.html>
Citado en: p. 50 del manual

Adobe Inc. (2023). Shape tweening in Animate. <https://helpx.adobe.com/animate/using/shape-tweening.html>
Citado en: p. 50 del manual

Adobe Inc. (2023). Symbols and ActionScript. <https://helpx.adobe.com/animate/using/symbols-actionscript.html>
Citado en: p. 50 del manual

Adobe Inc. (2025). Adobe Animate: Diseña animaciones interactivas para juegos, series de televisión y la web. <https://www.adobe.com/products/animate.html>
Citado en: p. 50 del manual

BMCC OpenLab. (2019). Adobe Animate: Introduction. <https://openlab.bmcc.cuny.edu/mmp260-1301-f2019/adobe-animate-introduction/>
Citado en: p. 50 del manual

Zuuví. (2025). Adobe Animate Glossary: Keyframe. <https://zuuvi.com/blog/adobe-animate-glossary/>
Citado en: p. 50 del manual

Zuuvi. (2025). Adobe Animate Glossary: Registration Point. <https://zuuvi.com/blog/adobe-animate-glossary/>

Citado en: p. 51 del manual

Zuuvi. (2025). Adobe Animate Glossary: Timeline. <https://zuuvi.com/blog/adobe-animate-glossary/>

Citado en: p. 52 del manual

Un ejercicio bastante útil al desarrollar un recurso multimedia consiste en analizar productos similares, apps, catálogos, o Interfaces, este análisis previo te dará nociones de organización y de métodos de trabajo. En el contenido de la bibliografía se puede encontrar varios ejemplos prácticos de aplicación de código.

actionscript

actionscript

actionscript

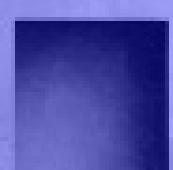
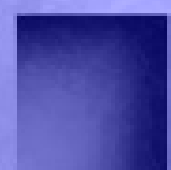
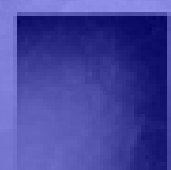
actionscript

actionscript

actionscript

action

action



action

action

action

action



Contactos:

(593-7) 283 8323 / 284 3619 / 0996976449 | info@sudamericano.edu.ec | Simón Bolívar y Manuel Vega Esq. Cuenca EC

ISBN: 978-9942-7180-7-5

