

**Ingeniería en Tecnologías de la Información**

**Proyecto**

**Serpientes y escaleras**

**Unidad 1-2**

**Manual Tecnico**

**Catedrático: MSI Mario Humberto Rodríguez Chávez**

**Alumnos: Jair Alberto Martinez González**

**Juan Manuel Mota Hernández**

**Grupo: ITI-07116**

Enero-Abril 2018 Ciudad Victoria Tamaulipas

**Índice**

[**Introducción 3**](#_Toc508541948)

[**Pantalla Principal 4**](#_Toc508541949)

[**Registro se jugadores 5**](#_Toc508541950)

[**Reglas 6**](#_Toc508541951)

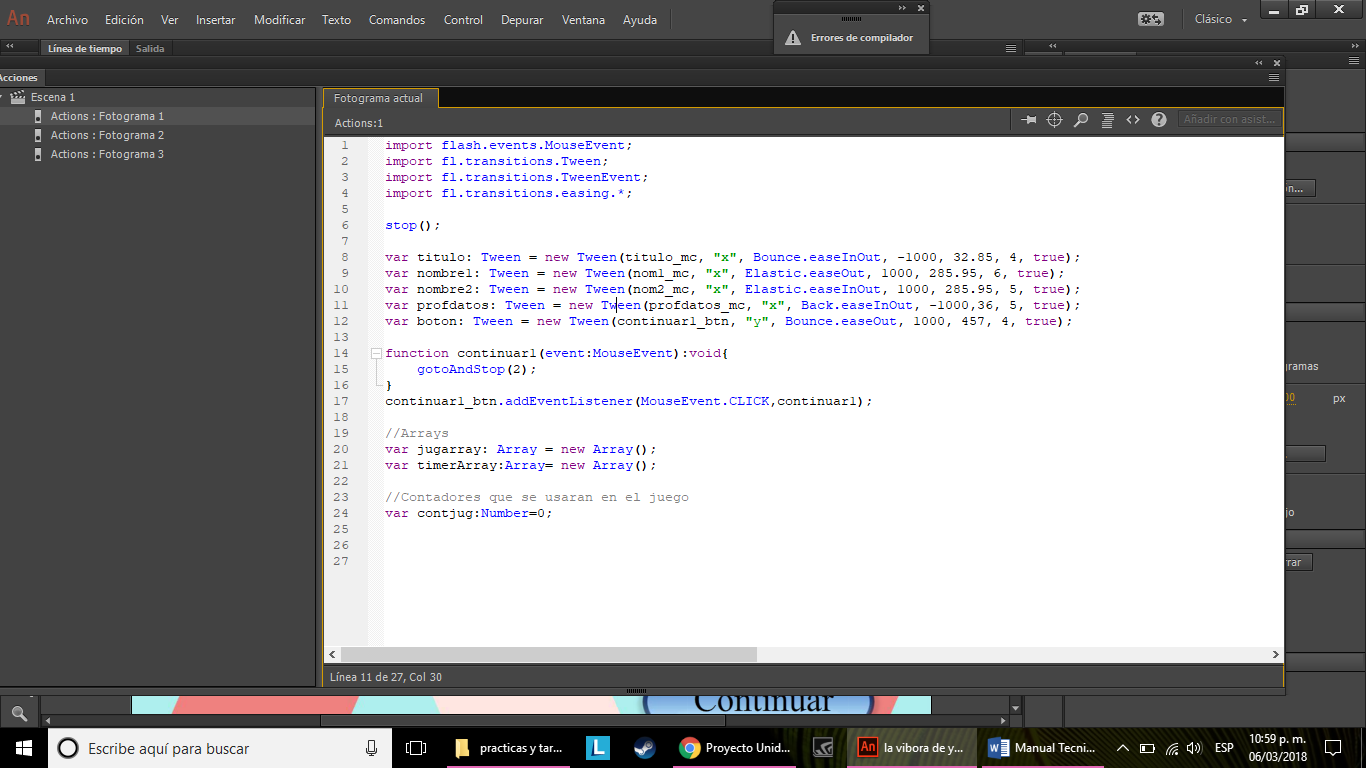
[**Juego 7**](#_Toc508541952)

[**Última pantalla 14**](#_Toc508541953)

# **Introducción**

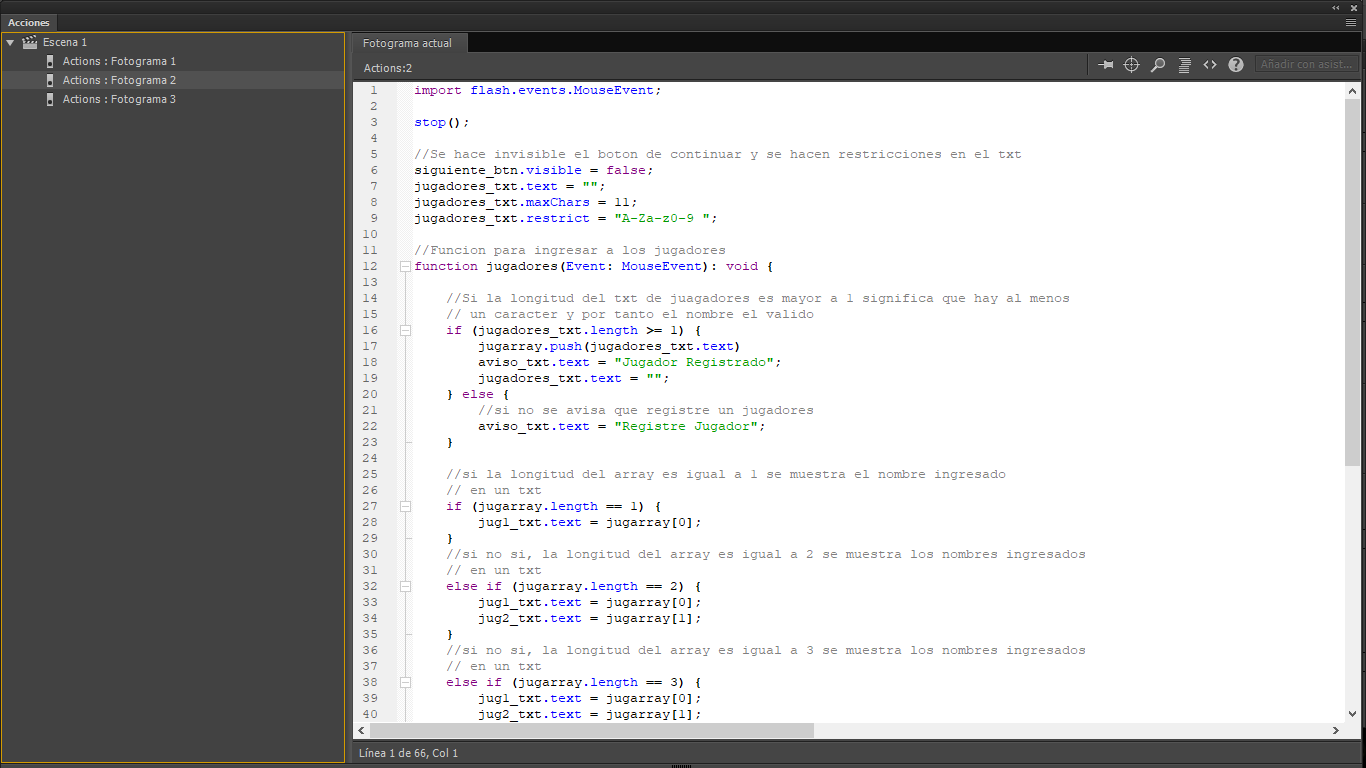
En este proyecto de la unidad 1-2 hicimos el juego de serpientes y escaleras, para poder realizar este juego se tuvo que investigar cosas nuevas y es por eso de que estamos seguros que la realización de este juego es una gran experiencia que nos deja muchas enseñanzas sobre cómo aplicar más la lógica de programación haciendo juegos, esto siendo muy importante ya que en la universidad necesitaremos mucha lógica para poder programar.

# **Pantalla Principal**



En código del primer frame tenemos las declaración de las variables tipo Tween para hacer las animación de los elementos de la portada, es decir: el nombre de los alumnos, el nombre del profesor, el nombre del juego y el botón de continuar, además tenemos la función que no sirve para continuar al frame 2, esta función es llamada por el botón continuar del frame 1.

# **Registro se jugadores**

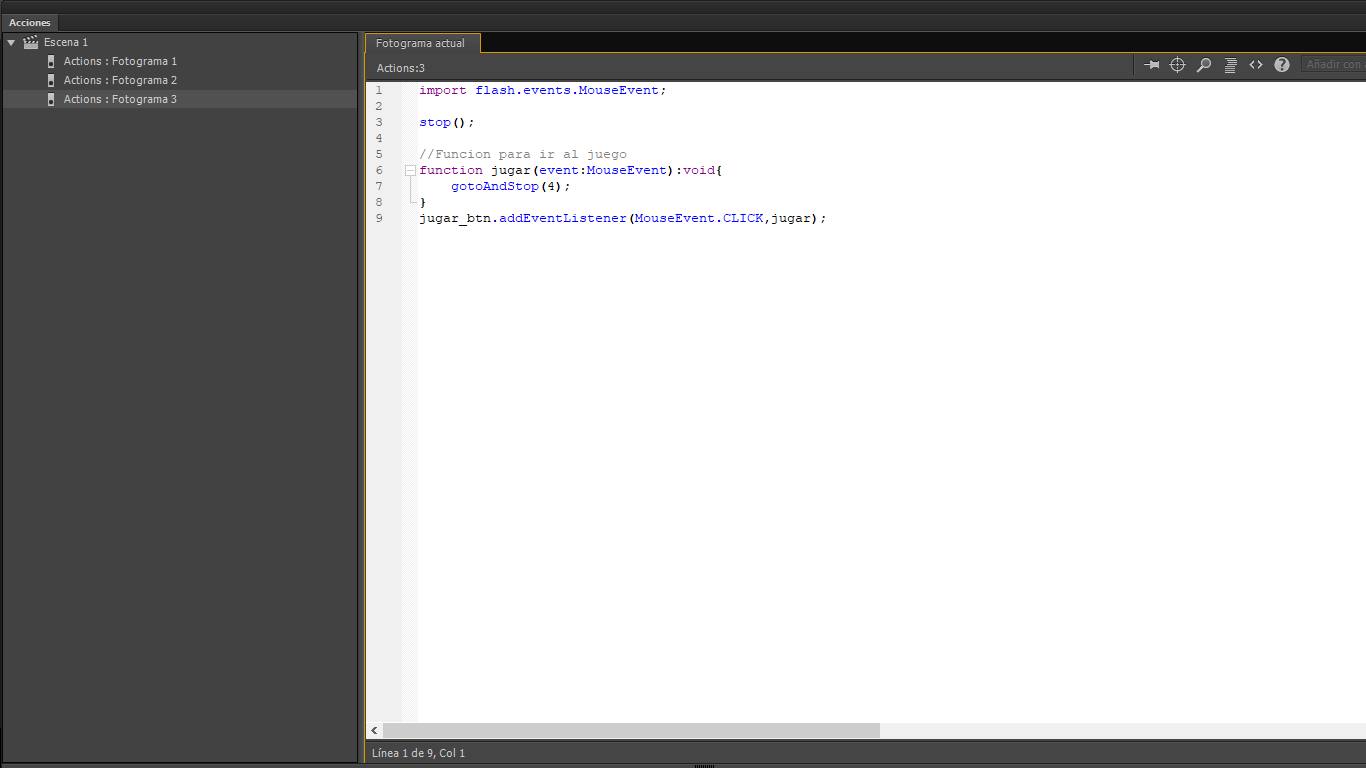


En el segundo frame se tiene el registro de jugadores, se hacen restricciones para que solo se puedan ingresar letras y numero, además de que el número máximo de caracteres que se pueden poner se restringe a 11.

En la función para registrar a los jugadores se valida que este ingresado un nombre utilizando la longitud del txt donde se registran los jugadores, si mayor o igual a 1 significa que hay algún carácter valido y por ende el nombre es válido.

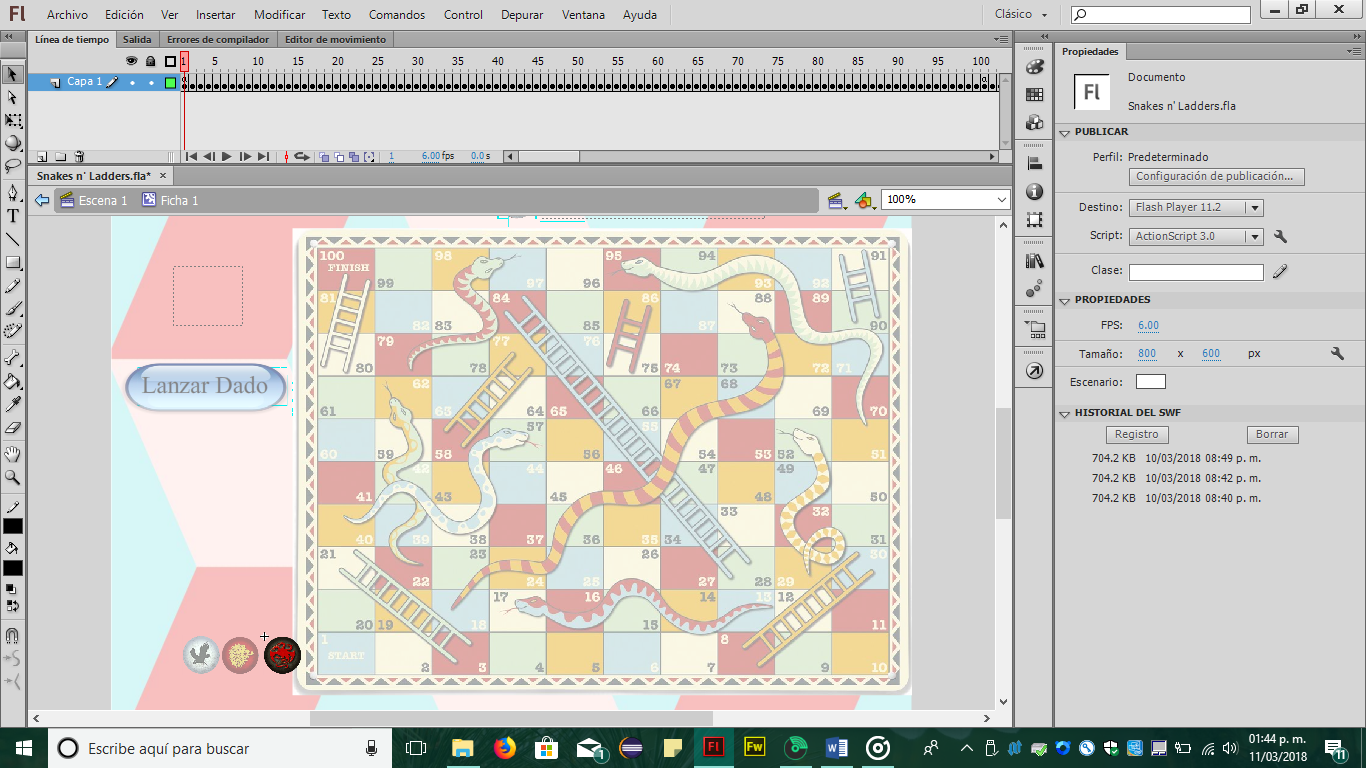
Hay 3 txt donde se mostraran los nombres ingresados, eso se hace utilizando la longitud del Array, por ejemplo, si la longitud del array es igual a 1 significa que se ingreso un nombre, entonces el primer txt tomara el valor de 0 del array ya que es el primer dato que esta en el array.

# **Reglas**

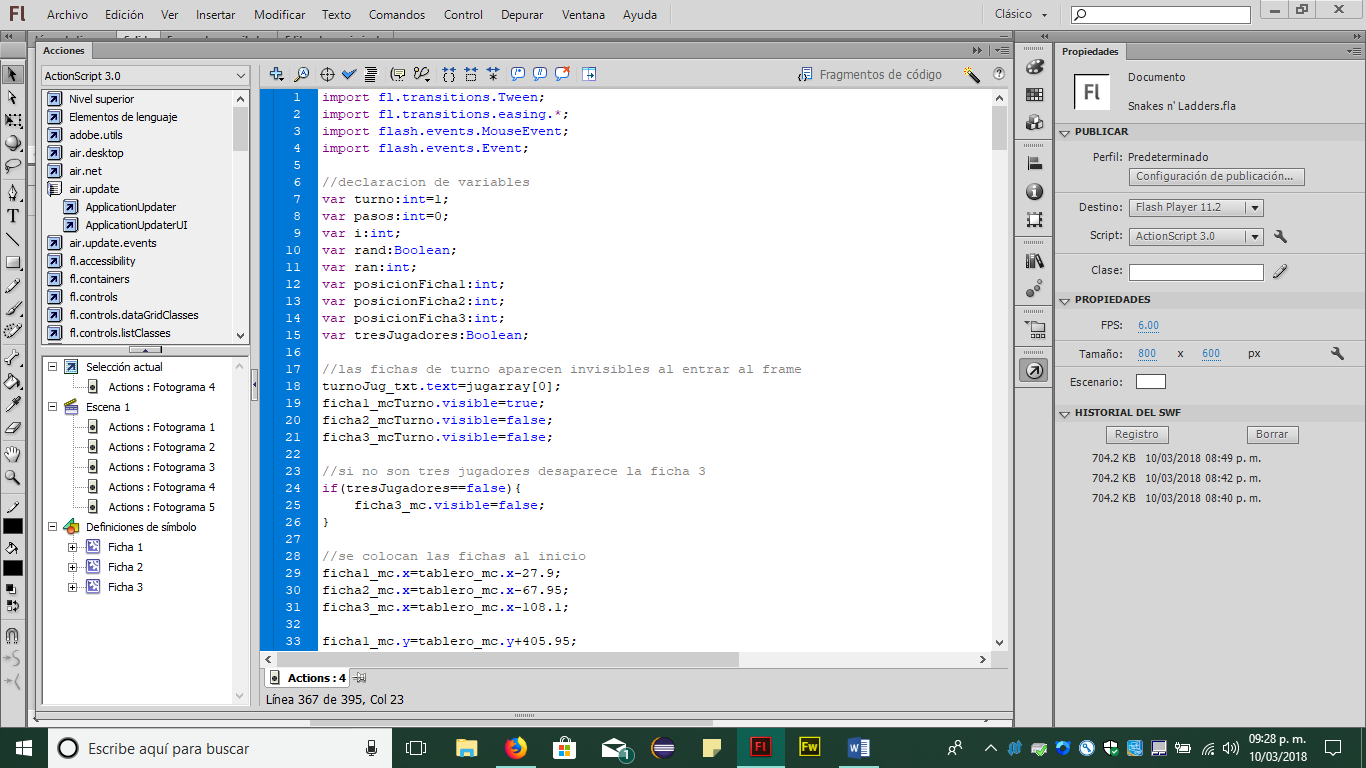


En el tercer frame que es el de las reglas se hace una función para que nos lleve al juego (frame 4).

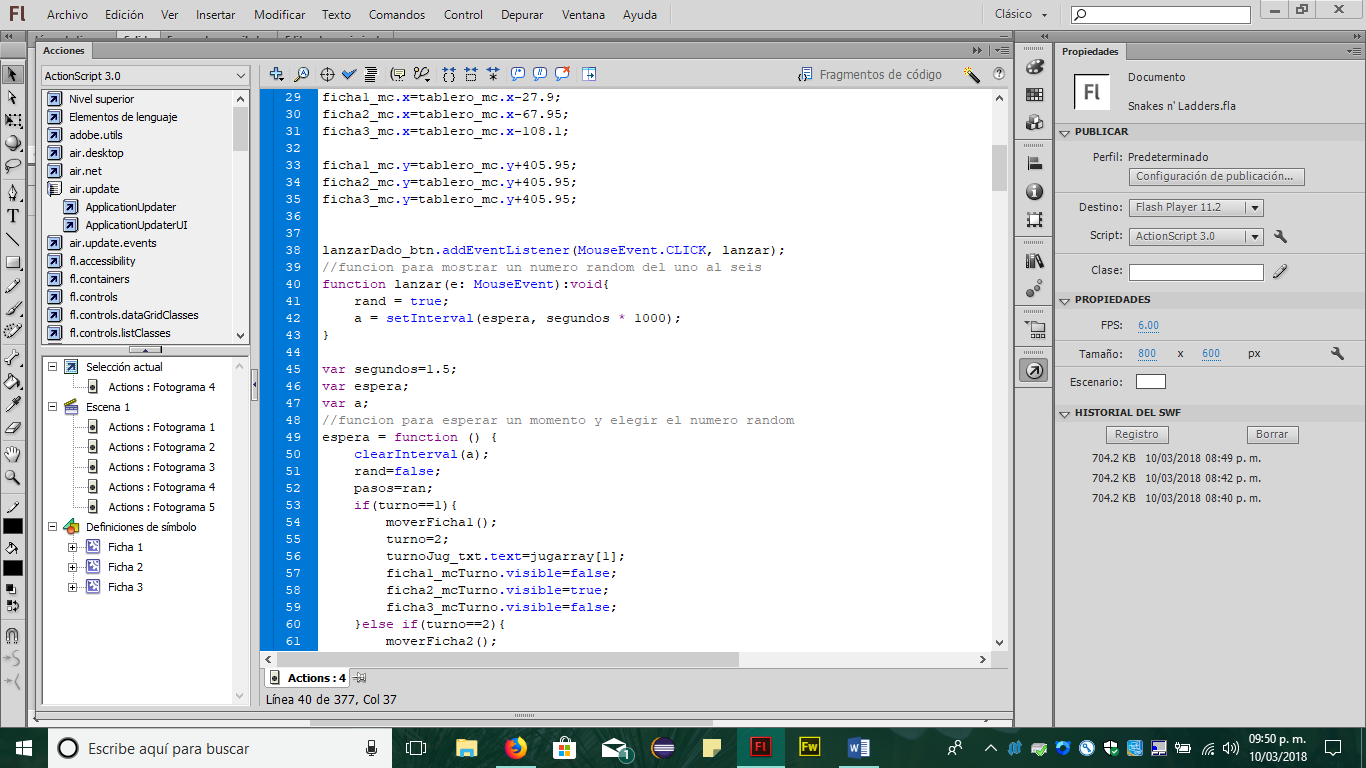
# **Juego**



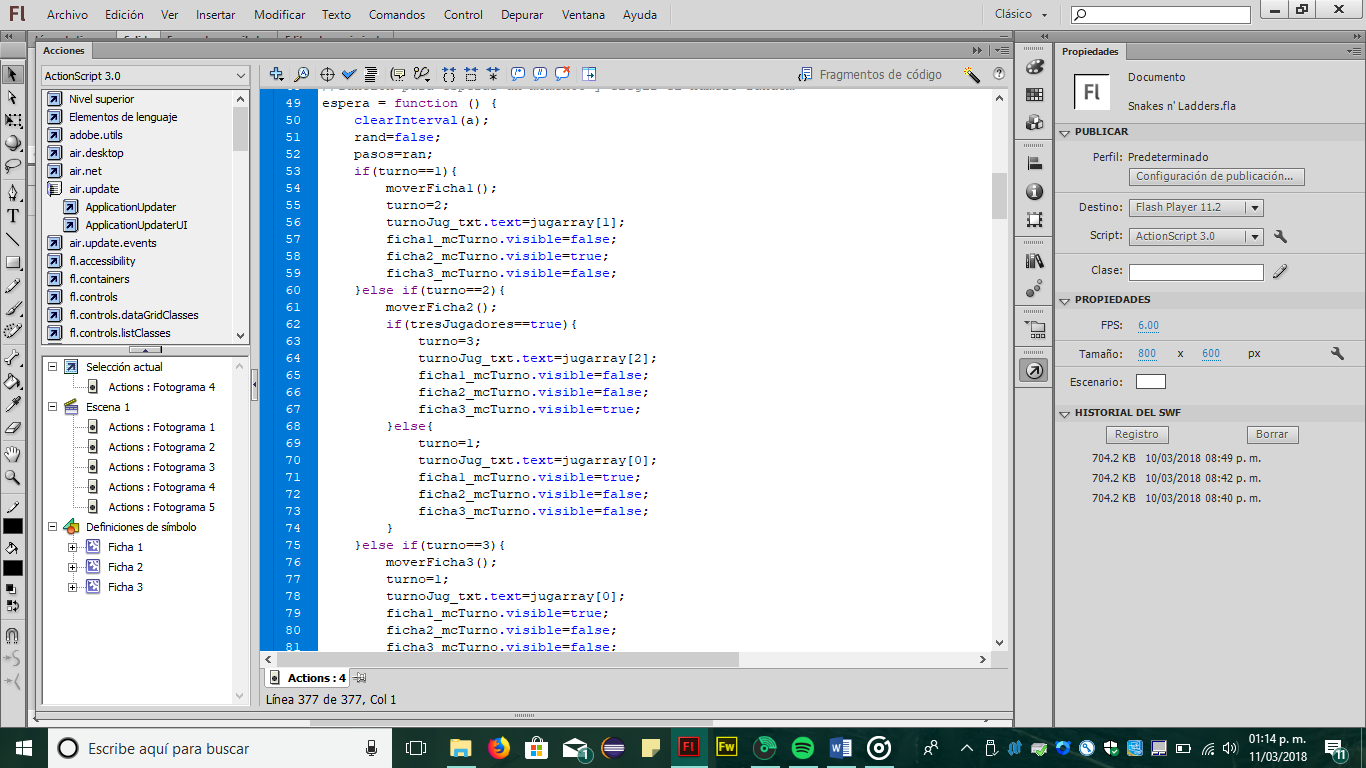
Para mover las fichas hicimos que cada frame de la ficha fuera una posición en el tablero y la secuencia de las escaleras y de las serpientes.



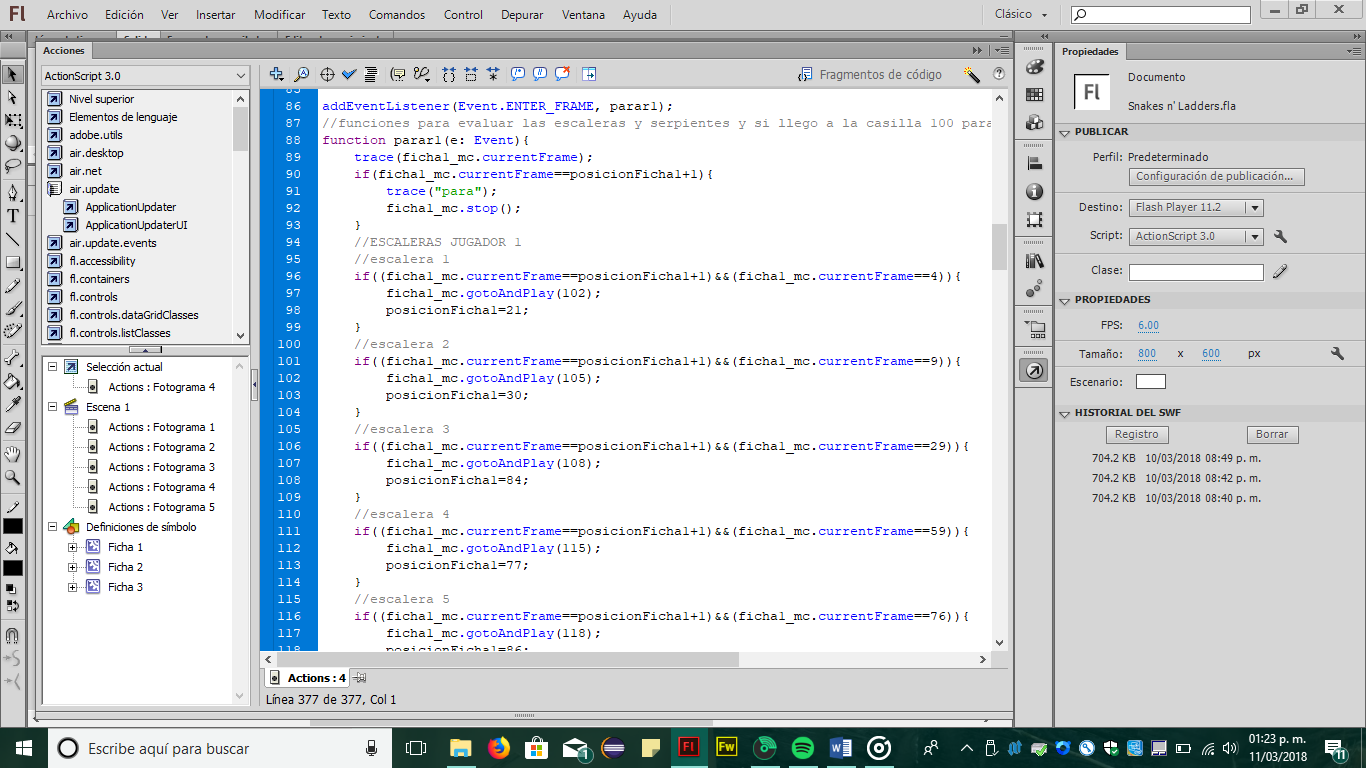
En el cuarto frame, donde se tiene el juego, es donde está lo más “complejo” del proyecto. Empieza la parte de código con las importaciones necesarias para la utilización de algunos métodos, en seguida para mostrar al jugador en turno (que siempre será el primero que se registró) el txt: turnoJug\_txt toma el valor del array de los jugadores en la primera posición (que será el jugador 1). Ahora se comprobará si son dos o tres personas las que jugarán y si no son tres, la tercera ficha se pondrá invisible.



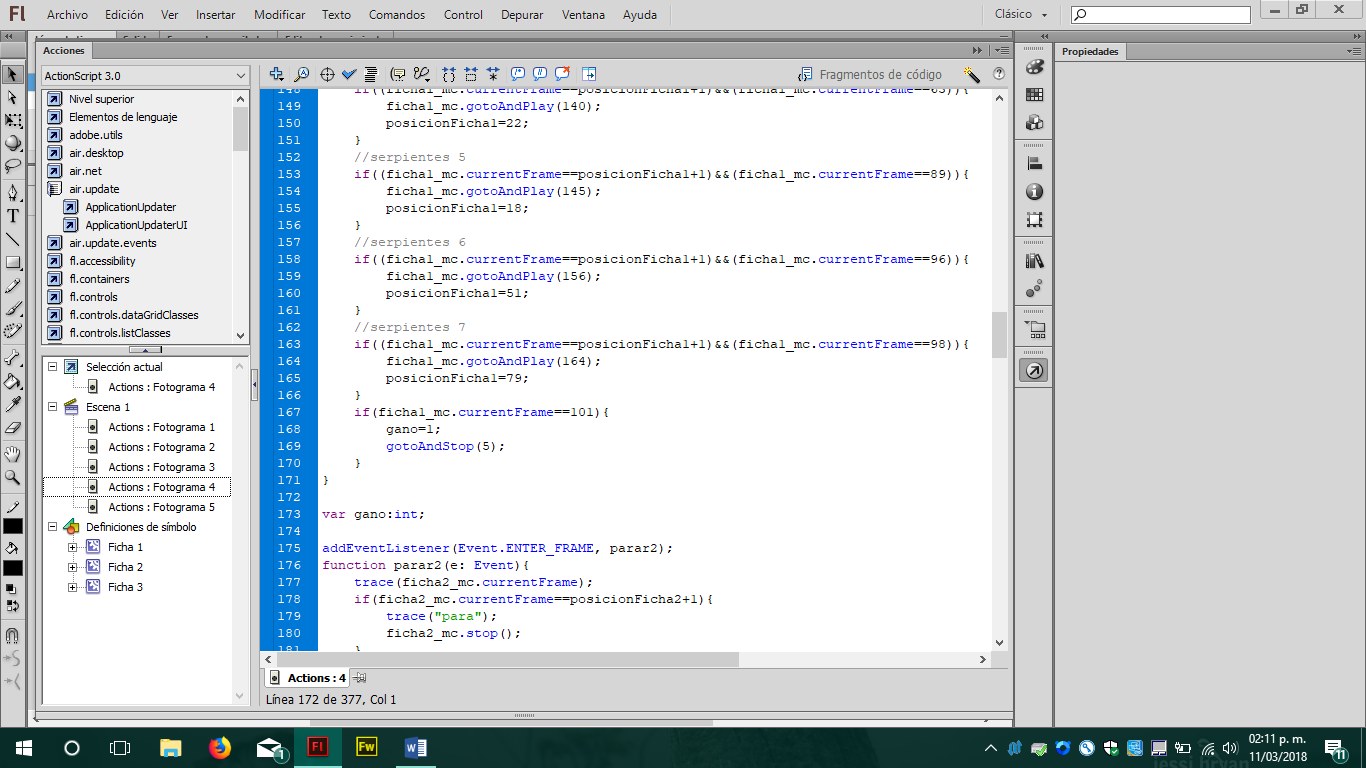
Ahora se situarán las fichas en una posición fija para que partan de ahí. La función: lanzar, pondrá la variable rand en true, para que se ejecute algo que está en una función tipo “enter frame” y hasta que pase el tiempo de “a”, se ejecutarán las instrucciones de la función “espera”.



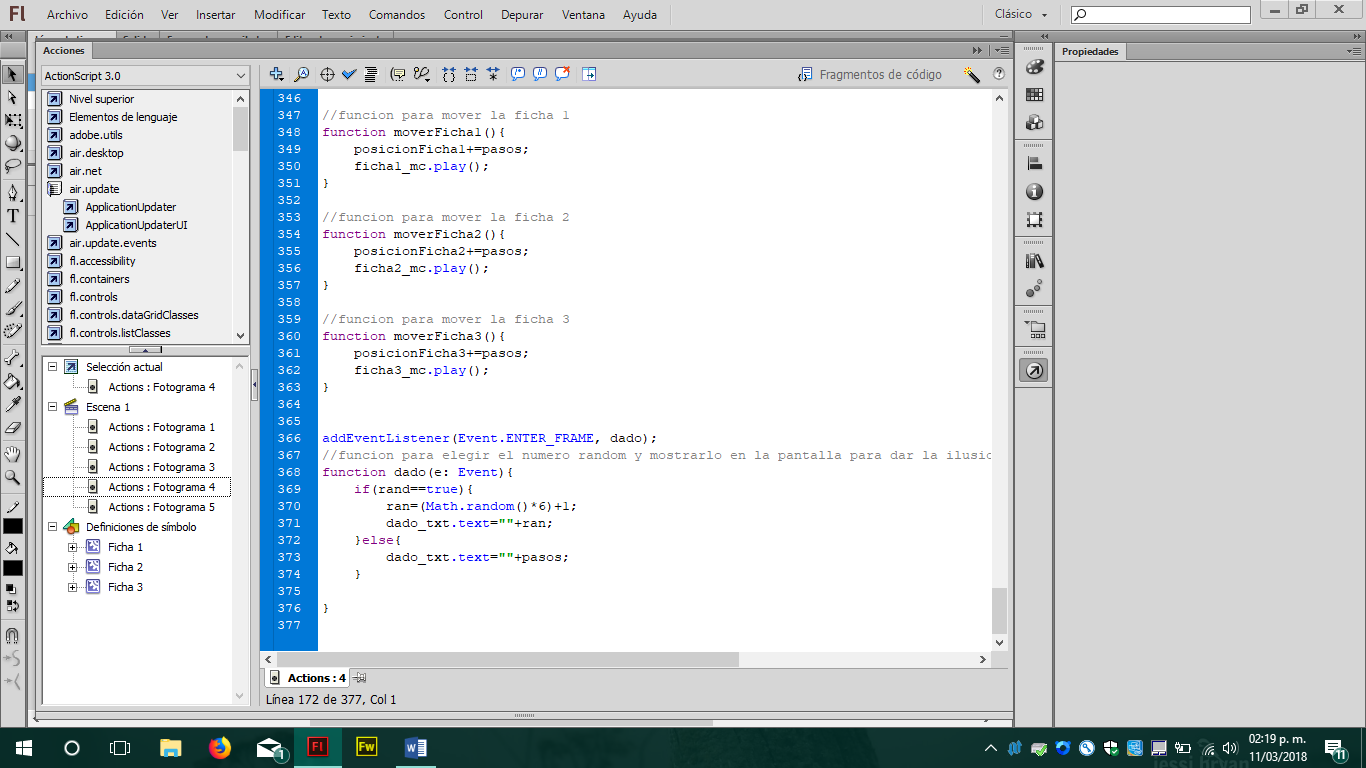
La función “espera”, básicamente lo que hace es esperar a que pasen los 1.5 segundos para que la variable rand tome el valor de falso y así “detenga” una acción que se ejecuta en una función enter frame, después pasos toma el valor que haya tenido la variable ran, después de que pasaran los 1.5 segundo, ahora se comprueba el turno del jugador, cabe mencionar que la variable turno se inicializó con el valor de 1, ya que siempre será primero el jugador 1, se evalúa si tiene el valor de 1, 2 o 3, en cualquier caso, dependiendo el turno, el txt: “turnoJug\_txt” toma el nombre del jugador en turno y los demás Movie Clips se vuelven invisibles excepto el del jugador actual.



En la siguiente función tipo enter frame, se evaluará cada frame, el frame en que se encuentre el Movie Clip del jugador en turno, y cuando éste se encuentre en una escalera o serpiente (después de acabado el turno) irá al frame del Movie Clip donde se encuentre la “animación” para que suba o que baje dependiendo donde caiga.

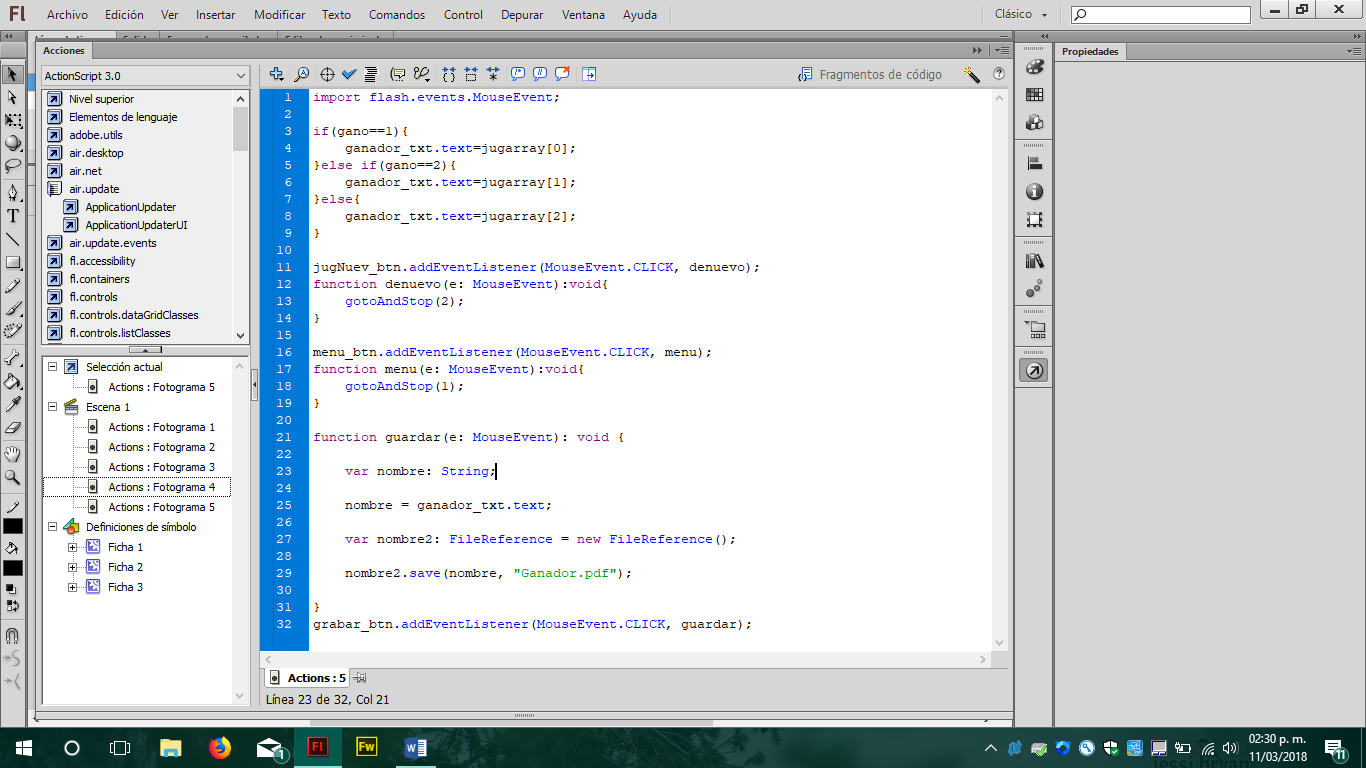


Al final de la función se comprueba si el frame actual de la ficha es 101 (que es la última casilla), si es así, la película irá al frame 5 donde se mostrará el ganador y otras cosas.



En la parte final del frame 4 se encuentran 4 funciones: “moverFicha1”, “moverFicha2”, “moverFicha3” y ”dado”. Las funciones moverFichax le sumará a la variable posicionFichax lo que contiene la variable pasos y reproducirá la película del Movie Clip hasta que se detenga dependiendo de los pasos que salió del lanzamiento del dado. La función dado, que es una función tipo enter frame, mientras que la variable rand sea verdadera, en cada frame, el txt: “dado\_txt” mostrará un número random del 1 al 6, cuando ya no sea verdadera, el txt mostrará el número de pasos que dará el jugador.

# **Última pantalla**



En el último frame, en la capa de Actions, se importa el paquete necesario para usar un evento tipo “MouseEvent”, después se comprueba quien fue el que ganó por medio de checar el valor de la variable gano, si es igual a 1, se mostrará el nombre del jugador uno, si es igual a 2, el del dos y sino el tercero. Ahora la función denuevo corresponderá al botón: jugNuev\_btn que lo que hará será ir al frame donde se registran a los jugadores. La función menu irá corresponderá al botón: menu\_btn que irá al frame 1 donde está la portada. Por último la función guardar lo que hará será guardar el nombre del jugador que ganó en un archivo PDF con el nombre de “Ganador.pdf”.