



## ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO DE TELECOMUNICACIÓN,  
ESPECIALIDAD EN SONIDO E IMAGEN

Título del proyecto:

**“DESARROLLO DE UN SOFTWARE INTERACTIVO DE  
AYUDA AL APRENDIZAJE”**

Arancha León Muñoz

Tutor: Alfredo Pina

Asesora externa: Isabel Sánchez

Pamplona, 10 de Septiembre del 2012

# ÍNDICE

1. OBJETIVO.....	7
2. FASES DEL PROYECTO.....	7
3. ORGANIZACIÓN DEL PROYECTO .....	9
4. INTRODUCCIÓN .....	11
4.1. ¿QUÉ ES UNA APLICACIÓN MULTIMEDIA INTERACTIVA? .....	11
4.2. ¿POR QUÉ USAR UNA APLICACIÓN MULTIMEDIA INTERACTIVA? .....	13
4.3. PRINCIPIOS QUE DEBE SEGUIR UNA APLICACIÓN MULTIMEDIA EDUCATIVA.....	14
4.4. ESCUELA 2.0 .....	16
4.5. APRENDIZAJE INDIVIDUAL Y COOPERATIVO .....	18
4.6. TRASTORNO GENERAL DE LA COMUNICACIÓN .....	20
5. ESTADO DEL ARTE .....	22
5.1. RECURSOS ANALÓGICOS .....	22
5.2. RECURSOS DIGITALES .....	24
6. ENTORNO DE TRABAJO Y LENGUAJE DE PROGRAMACIÓN .....	28
6.1. ADOBE FLASH PROFESSIONAL .....	28
6.1.1. VERSIONES EMPLEADAS .....	29
6.1.2. INTERFAZ .....	29
6.1.2.1. Escenario .....	30
6.1.2.2. Barra de herramientas.....	31
6.1.2.3. Línea de tiempo .....	32
6.1.2.4. Otros paneles agrupados .....	33
6.1.2.4.1. Biblioteca .....	33
6.1.2.4.2. Propiedades .....	34
6.1.2.4.3. Acciones .....	35
6.1.3. TIPOS DE ARCHIVOS .....	35
6.2. ACTION SCRIPT .....	36
6.2.1. VERSIÓN EMPLEADA .....	37
6.2.2. INTRODUCCION A AS3.0 .....	37
6.2.2.1. TIPOS DE DOCUMENTOS .....	37

6.2.2.2.	OBJETOS .....	39
6.2.2.3.	VARIABLES .....	40
6.2.2.4.	MÉTODOS Y FUNCIONES .....	42
6.2.2.5.	EVENTOS .....	42
6.2.2.6.	ESTRUCTURAS “DE CONTROL” .....	43
6.2.2.6.1.	BUCLE FOR .....	43
6.2.2.6.2.	IF .....	44
6.2.2.6.3.	SWITCH .....	44
6.2.2.7.	CLASES .....	45
7.	APLICACIÓN .....	49
7.1.	TEMÁTICA ESCOGIDA .....	49
7.2.	PLATAFORMAS .....	50
7.3.	PRIMER BOCETO .....	51
7.4.	CONSULTA A LA EXPERTA .....	57
7.5.	APLICACIÓN “FINAL” .....	58
7.6.	ADECUACION A VARIOS FORMATOS .....	73
7.7.	PRUEBAS DE USUARIO .....	74
7.8.	MEJORA DE LA APLICACIÓN DEFINITIVA .....	83
8.	CONCLUSIONES Y LÍNEAS FUTURAS .....	87
9.	BIBLIOGRAFÍA .....	89
	ANEXO I: EXPORTAR ARCHIVO .APK DE ADOBE FLASH PROFESSIONAL CS5.5.....	90
	ANEXO II: IMPORTAR ARCHIVO APK A UN DISPOSITIVO ANDROID .....	98
	ANEXO III: MANUAL DE USUARIO.....	104
	MODO INDIVIDUAL .....	104
	MODO PAREJA .....	111

# TABLA DE FIGURAS

Fig. 1 Fases del proyecto.....	8
Fig. 2 Cuento con pictogramas.....	23
Fig. 3 Tarjetas .....	23
Fig. 4 Tarjeta Tabú.....	23
Fig. 5 Juego ¿Quién es quién?.....	24
Fig. 6 Captura de pantalla juego busca la palabra .....	25
Fig. 7 Interfaz ¿Quién es quién? 1.....	26
Fig. 8 Interfaz ¿Quién es quién? 2.....	26
Fig. 9 Captura de pantalla juego cohetes que riman .....	27
Fig. 10 Icono de Flash.....	28
Fig. 11 Parte de la interfaz Flash .....	29
Fig. 12 Botón, Mc y gráfico .....	30
Fig. 13 Herramientas de dibujo y selección .....	31
Fig. 14 Herramientas de visualización .....	31
Fig. 15 Herramientas de colores .....	31
Fig. 16 Herramientas de opciones .....	31
Fig. 17 Línea de tiempo .....	32
Fig. 18 Línea de tiempo 2 .....	32
Fig. 19 Biblioteca .....	34
Fig. 20 Propiedades.....	34
Fig. 21 Panel de acciones .....	35
Fig. 22 Configuración de la publicación .....	36
Fig. 23 Pantalla inicial Flash CS5 .....	38
Fig. 24 Crear objetos a partir de un símbolo .....	39
Fig. 25 Definición de un objeto .....	40
Fig. 26 Modificación de las propiedades de un objeto.....	40
Fig. 27 Definición variable.....	41
Fig. 28 Tipos de variables .....	41
Fig. 29 Método y función .....	42
Fig. 30 Eventos .....	43
Fig. 31 Bucle for.....	43
Fig. 32 If.....	44
Fig. 33 Switch .....	44
Fig. 34 Crear clase .....	45
Fig. 35 Estructura de una clase .....	46
Fig. 36 Llamar a clase 1 .....	47
Fig. 37 Llamar a clase 2 .....	48
Fig. 38 Componentes de las fichas.....	52
Fig. 39 Fichas .....	52
Fig. 40 Botones boceto .....	53
Fig. 41 Palabras boceto .....	53
Fig. 42 Interfaz boceto .....	54
Fig. 43 1º parte código del boceto.....	54
Fig. 44 2º parte código del boceto 1 .....	55
Fig. 45 2º parte código del boceto 2 .....	55

Fig. 46 detalle 3º parte código del boceto.....	56
Fig. 48 Botones de Photoshop .....	58
Fig. 47 Botones frases sin hacer.....	58
Fig. 49 Código de importar las clases.....	59
Fig. 50 Clase “cargarFichas” .....	60
Fig. 51 Clase “ElegirPersonaje” .....	61
Fig. 52 Código simplificado .....	61
Fig. 53 Pantalla de inicio .....	62
Fig. 54 Código pantalla de inicio .....	63
Fig. 55 Botones frases hechas.....	64
Fig. 56 Detalle animación de palabra.....	64
Fig. 57 Esquema de clases 1.....	65
Fig. 58 Esquema de clases 2.....	65
Fig. 59 Esquema de clases 3.....	66
Fig. 60 Pregunta final .....	69
Fig. 61 Respuestas finales .....	69
Fig. 62 Pantalla final.....	70
Fig. 63 Mensaje de no vidas.....	71
Fig. 64 Flecha volver a inicio .....	71
Fig. 65 Pantalla comprobación de selección personaje.....	72
Fig. 66 Pantalla selección de personaje .....	72
Fig. 67 Pantallas y bocadillos de ayuda.....	72
Fig. 68 Botones de responder .....	72
Fig. 69 Animación inicial .....	76
Fig. 70 Preguntas y respuestas finales .....	76
Fig. 71 Interfaz final Modo pareja Preguntas sin construir .....	76
Fig. 72 Interfaz final Modo pareja Preguntas construidas.....	76
Fig. 73 Pantalla final definitiva.....	76
Fig. 74 Pantalla inicial de Flash CS5.5 .....	76
Fig. 75 Pantalla propiedades.....	76
Fig. 76 Configuración Air Android .....	76
Fig. 77 Pantalla general de configuración air android .....	76
Fig. 78 Pantalla crear certificado digital .....	76
Fig. 79 Pantalla implementación de configuración air android.....	76
Fig. 80 Pantalla iconos de configuración air android.....	76
Fig. 81 Pantalla permisos de configuración air android .....	76
Fig. 82 captura pantalla de publicar .....	76
Fig. 83 Archivo creado .apk.....	76
Fig. 84 Página de descarga android SDK .....	76
Fig. 85 Pantalla instalador.....	76
Fig. 86 Pantalla ajustes Samsung Galaxy Tab .....	76
Fig. 87 Conexión física Tableta-Pc.....	76
Fig. 88 Directorio donde se encuentra adb.exe.....	76
Fig. 89 Comandos para cambiar de directorio.....	76
Fig. 90 Comando para reconocer tableta .....	76
Fig. 91 Comando instalar archivo .apk.....	76
Fig. 92 Portada, modo de juego individual .....	76
Fig. 93 Animación inicial .....	76
Fig. 94 Pantalla inicial.....	76

Fig. 95 Pantalla pregunta construida eliminación de fichas automática o manual.....	76
Fig. 96 Pantalla pregunta construida eliminación de fichas automática o manual.....	76
Fig. 97 Mensaje sin vidas .....	76
Fig. 98 Posibles resultados de la partida.....	76
Fig. 99 Pantalla fin del juego .....	76
Fig. 100 Portada, modo de juego pareja.....	76
Fig. 101 Animación inicial .....	76
Fig. 102 Pantalla inicial.....	76
Fig. 103 Jugador 1 .....	76
Fig. 104 Selección de personaje.....	76
Fig. 105 Asegurar personaje .....	76
Fig. 106 Jugador 1 y jugador 2 .....	76
Fig. 107 Nivel pregunta construida .....	76
Fig. 108 Nivel pregunta sin construir .....	76
Fig. 109 Responder a la cuestión .....	76

## 1. OBJETIVO

El objetivo general de este proyecto es el diseño, implementación y evaluación de una aplicación multimedia interactiva que ayude a la enseñanza, individual y cooperativo, de personas con problemas de aprendizaje. En concreto, destinada a personas con trastornos generales de la comunicación.

La implementación de dicha aplicación se desarrollará en el lenguaje de programación orientado a objetos ActionScript 3.0, por lo que se desarrollará en el entorno de trabajo de Adobe Flash Professional.

Por lo tanto, los objetivos específicos de este proyecto son:

- Adquisición de conocimientos sobre el software de desarrollo Adobe flash Professional.
- Aprendizaje del lenguaje de programación ActionScript 3.0.
- Diseño e implementación de una aplicación destinada a la enseñanza de personas con trastornos generales de la comunicación.
- Adecuación del software a varios soportes (PC y tableta)
- Evaluación del software con personas de características similares a los destinatarios del producto.

## 2. FASES DEL PROYECTO

El proyecto se ha llevado a cabo siguiendo diferentes fases o etapas estructuradas de la siguiente forma:

1º Fase: Familiarización con el entorno de trabajo Adobe Flash y lenguaje de programación ActionScript 3.0 para ver las posibilidades que estos ofrecen a la hora de la implementación de la aplicación.

2º Fase: Investigación de las aplicaciones existentes y del “estado del arte” para pensar en una aplicación novedosa y, a la vez, educativa.

3º Fase: Decisión sobre la temática del software a partir de las conclusiones obtenidas de las dos fases anteriores.

4º Fase: Implementación de un primer esbozo de la aplicación y consulta a una “experta” sobre su opinión acerca de la temática escogida y de la formas de implementación.

5º Fase: Desarrollo y programación de la aplicación final.

6º Fase: Adecuación de la aplicación a varias plataformas (PC y tableta)

7º Fase: Evaluación del software con personas con características parecidas a las destinatarias de la aplicación.

8º Fase: Mejora del software implementado a partir de las observaciones captadas en la fase anterior.

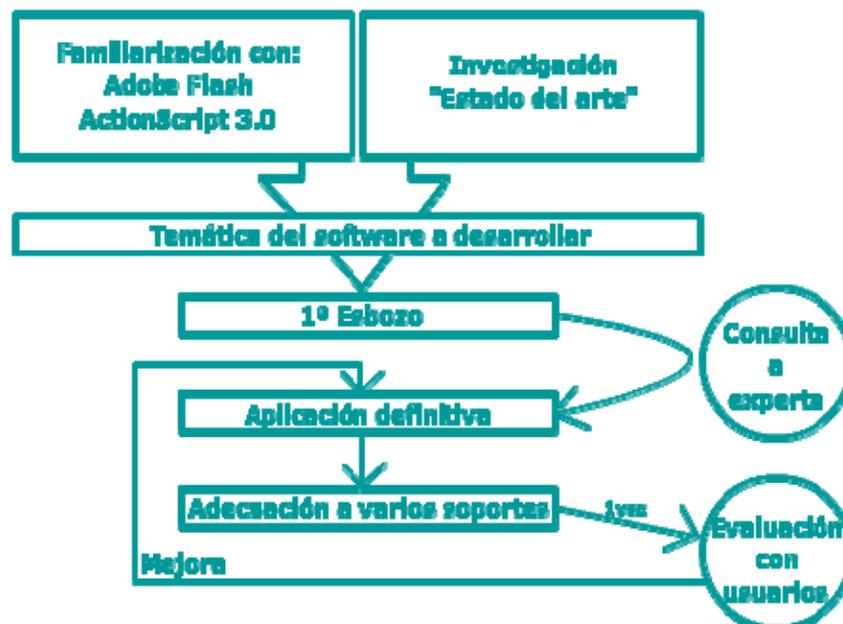


Fig. 1 Fases del proyecto

### 3. ORGANIZACIÓN DEL PROYECTO

A continuación se describe la organización de la presente memoria. Tanto sus apartados como una breve explicación de cada uno de ellos.

- Introducción:

En este primer apartado se concreta, define y justifica lo que se va a hacer. Es decir, se explican los conceptos que se van a desarrollar durante el proyecto; qué es lo que se va a hacer y cuál es la razón, o razones, por las que se ha decidido hacer eso y no otra cosa diferente.

- Estado del arte:

En este segundo apartado se estudian y observan diferentes métodos de aprendizaje ya existentes con la finalidad de obtener ideas novedosas para este proyecto.

- Entorno de trabajo y lenguaje de programación:

El entorno de trabajo en el que se va a desarrollar este proyecto es Adobe Flash Professional, más concretamente en sus versiones CS5 y CS5.5. Durante este apartado se van a explicar sus fundamentos básicos y su lenguaje de programación as3.0.

- Aplicación:

En este apartado se va a describir la aplicación que se ha desarrollado durante este proyecto una vez investigadas las características que debe cumplir para ser una buena aplicación multimedia educativa, observado el estado del arte y familiarizado con el entorno de trabajo.

- Conclusiones y líneas futuras:

Una vez desarrollado el proyecto, en el último apartado se hace una reflexión sobre si se han llevado a cabo los objetivos especificados al inicio del este y también se dan algunas ideas de cómo se puede seguir trabajando a partir de dicho proyecto en un futuro.

- Anexo I: Exportar archivo .apk de Adobe Flash Professional CS5.5

En el anexo I se explica paso a paso cómo exportar un archivo .apk de una aplicación desarrollada en el entorno de trabajo Adobe Flash Professional CS5.5. Gracias a este archivo .apk se pondrá instalar la aplicación en un dispositivo android.

- Anexo II: Importar archivo .apk a un dispositivo android

En el anexo II se describen los pasos a seguir de una de las maneras para pasar un archivo .apk desde un PC con sistema operativo Windows a un dispositivo android y así dejar la aplicación corriendo perfectamente en dicho dispositivo.

- Anexo III: Manual de usuario

En este último anexo se explica paso a paso cómo se juega a cada uno de los modos de juego.

## 4. INTRODUCCIÓN

En este apartado se va a explicar punto por punto el objetivo general del proyecto. Así mismo, se van a definir cada uno de los conceptos que se desarrollan en el trabajo y la justificación de por qué aparecen estos conceptos y no otros. Para ello, se desarrollan varios sub-apartados.

### 4.1. ¿QUÉ ES UNA APLICACIÓN MULTIMEDIA INTERACTIVA?

Para entender el objetivo del general del proyecto, cabe definir de manera individual cada uno de los términos de la expresión “aplicación multimedia interactiva”.

#### - APLICACIÓN:

Una aplicación informática es un tipo de software diseñado como herramienta para permitir al usuario realizar uno o diversos tipos de trabajos. Normalmente, están diseñadas para educar, entretener o ayudar en la vida diaria a sus usuarios.

En los últimos años, con la aparición de los Smartphones las aplicaciones móviles han alcanzado una gran importancia. Una aplicación móvil es una aplicación informática desarrollada para correr bajo Smartphones, también llamados teléfonos inteligentes.

Por lo tanto, las funcionalidades de las aplicaciones que antes solo estaban disponibles en PC ahora pueden ser ejecutadas en los dispositivos móviles, con un buen nivel de prestaciones.

- MULTIMEDIA:

El término multimedia se utiliza para referirse a cualquier objeto o sistema que utiliza más de un medio de expresión para presentar o comunicar información.

Los medios de expresión pueden ser muy variados. Desde texto, gráficos e imágenes hasta animación, sonido, video, etc.

- INTERACTIVA:

Para comprender el significado de interactivo hace falta conocer que se designa iteración a aquella acción que se ejerce de manera recíproca entre dos o más sujetos, objeto, agentes, fuerzas o funciones. De este modo, se define interactivo a todo aquello que provenga o proceda por interacción.

De esta manera, en la informática se designa a aquel programa a través del cual se permite una interacción, a modo de diálogo, entre un ordenador y un usuario.

Por lo tanto, se habla de **multimedia interactiva** cuando el usuario tiene libre control sobre la presentación de los contenidos, acerca de qué es lo que desea ver y cuando; a diferencia de una presentación lineal, en la que es forzado a visualizar contenido en un orden predeterminado.

Por todo esto, cuando se describe el objetivo del proyecto, se concibe que lo que se va a diseñar, implementar y evaluar es un software diseñado para correr en un PC, un móvil u otra plataforma cuya finalidad es educar, entretener o ayudar en la vida diaria de sus usuarios. Además éste integra texto, animación, sonidos, etc, es decir, emplea varios medios de expresión. Y además, dicho software permite un diálogo ente usuario y ordenador cediendo así a que el usuario tenga libre control sobre los contenidos.

#### 4.2. ¿POR QUÉ USAR UNA APLICACIÓN MULTIMEDIA INTERACTIVA?

Existen multitud de razones para desarrollar una aplicación multimedia interactiva frente a otros medios educativos tradicionales.

Por un lado, el que se implemente en un soporte digital conlleva numerosas ventajas. Por ejemplo, se puede utilizar en cualquier lugar y en cualquier momento del día como en un centro de aprendizaje, en el hogar, durante viajes, a través de intranets, Internet, etc. a cualquier hora.

También, se puede personalizar en función de las características y necesidades del usuario final. Cada tema o sección es independiente, por lo que los usuarios pueden profundizar en las áreas temáticas que necesitan. En muchos casos, las aplicaciones también incluyen la opción de personalizar el nivel de dificultad según crea conveniente el especialista.

Otra gran ventaja del soporte digital es que éste es fácilmente actualizable y, por ello, mucho más cómodo y barato que los medios tradicionales. También posee gran capacidad de almacenamiento por lo que es posible almacenar numerosos ejercicios al igual que los datos y resultados de cada uno de los usuarios.

Además ofrece muchas nuevas opciones como por ejemplo poder cambiar el idioma, añadir música, video, etc. todo ello con una gran calidad.

Por otro lado, el ser multimedia interactivo hace que el usuario tome menos tiempo de estudio, disfrute más y el aprendizaje aumente. Esto es debido a que la interacción que se produce entre usuario y aplicación facilita la atención, la comprensión y la retención de información de una forma intuitiva, espontánea y divertida, muy diferente a las metodologías tradicionales de enseñanza y aprendizaje.

Numerosos estudios han encontrado que la interactividad tiene un fuerte efecto positivo en el aprendizaje. Por ejemplo, Bosco (1986) revisaron 75 estudios de

aprendizaje y se encontró que los alumnos aprenden más rápido, y tienen mejores actitudes hacia la educación en el uso de multimedia interactiva.

Sin embargo, no hay que olvidar que a pesar de todas las ventajas mencionadas anteriormente este tipo de material didáctico requiere de una infraestructura adecuada y una formación didáctica tecnológica.

#### 4.3. PRINCIPIOS QUE DEBE SEGUIR UNA APLICACIÓN MULTIMEDIA EDUCATIVA

El psicólogo Richard E. Mayer, con un equipo de investigadores desarrollaron una teoría cognitiva de los aprendizajes multimedia teniendo presente que los humanos procesamos la información separadamente a través de dos canales diferentes: la información visual y la información sonora.

De dicho estudio obtuvieron una serie de principios que se han cumplido a la hora de diseñar e implementar la aplicación. Dichos principios se muestran a continuación:

- Principio multimedia:

El alumnado aprenderá mejor con imágenes y apoyo verbal que únicamente con el apoyo verbal.

- Principio de contigüidad espacial:

Los estudiantes aprenden mejor si el texto y las imágenes que lo ilustran están maquetadas preferentemente cerca las unas de las otras en la página o pantalla. Es importante por lo tanto no separar demasiado la imagen o animación de su texto alusivo y es fundamental no fragmentar en diferentes diapositivas la imagen del texto.

- Principio de contigüidad temporal:

Se aprende mejor cuando los recursos multimedia correspondientes son presentados simultáneamente preferentemente que sucesivamente.

- Principio de coherencia:  
Es mejor no incorporar en el documento sonidos, términos o imágenes prescindibles. A veces asistimos a presentaciones con sonidos innecesarios que no aportan ninguna clase de información, como fondos de pantalla absolutamente absorbentes.
- Principio de modalidad:  
Los estudiantes aprenden mejor de la combinación animación y narración que de la combinación animación y texto en pantalla.
- Principio de redundancia:  
Se aprende mejor (se enseña, por lo tanto mejor) con una animación y una narración que no solo con una animación o narración.
- Principio de segmentación:  
Se aprende mejor en presentaciones por segmentos que con discursos que recojan mucha información en una sola diapositiva.
- Principio de personalización:  
La comprensión se mejora si se usa un lenguaje coloquial más que uno excesivamente formal. No implica rehuir del lenguaje científico cuando acontezca imprescindible pero sí que hace falta personalizarlo a las características de las personas destinatarias.
- Principio de interactividad:  
Se aprende mejor cuando se puede hacer partícipe al usuario de las diversas secuencias de las que consta la presentación.
- Principio de señalización  
Se aprende mejor, si se optima toda presentación si va acompañada de los elementos icónicos necesarios.
- Principio de las diferencias individuales

Las estrategias mencionadas son más efectivas para novatos y aprendices visuales. Las presentaciones mejor estructuradas deben ser creadas para aquellos que más ayuda necesitan.

Por lo tanto, para que una aplicación sea una buena aplicación educativa, debe de cumplir el máximo posible de los principios explicados anteriormente.

#### 4.4. ESCUELA 2.0



En un apartado anterior se había mencionado que este tipo de material didáctico requiere de una infraestructura adecuada y una formación didáctica tecnológica. Sin embargo, estas infraestructuras se podrían conseguir gracias al programa Escuela 2.0 que aprobó el Consejo de Ministros en 2009.

El Programa Escuela 2.0 fue el último proyecto de integración de las Tecnologías de la Información y de la Comunicación (TIC) en los centros educativos. El objetivo era poner en marcha las aulas digitales del siglo XXI, aulas dotadas de infraestructura tecnológica y de conectividad.

El programa Escuela 2.0 se basaba en los siguientes ejes de intervención:

- Aulas digitales. Dotar de recursos TIC a los alumnos y alumnas y a los centros: ordenadores portátiles para alumnado y profesorado y aulas digitales con dotación eficaz estandarizada.
- Garantizar la conectividad a Internet y la interconectividad dentro del aula para todos los equipos. Posibilidad de acceso a Internet en los domicilios de los alumnos/as en horarios especiales.

- Promover la formación del profesorado tanto en los aspectos tecnológicos como en los aspectos metodológicos y sociales de la integración de estos recursos en su práctica docente cotidiana.
- Generar y facilitar el acceso a materiales digitales educativos ajustados a los diseños curriculares tanto para profesores y profesoras como para el alumnado y sus familias.
- Implicar a alumnos y alumnas y a las familias en la adquisición, custodia y uso de estos recursos.

Las actuaciones se centraron, en su primera fase, en el tercer ciclo de Educación Primaria, comenzando el curso 2009-10 con 5º de Primaria en todos los centros sostenidos con fondos públicos y, posteriormente, se extendería a 6º de Primaria y a los dos primeros cursos de la ESO.

La formación de todos los profesores y profesoras implicados en el Programa, así como de los técnicos que serán responsables del apoyo técnico, era fundamental para el éxito de la iniciativa.

La formación contemplaba entre sus contenidos los aspectos metodológicos y de gestión de un aula dotada tecnológicamente y asimismo los propios aspectos de funcionamiento del equipamiento del aula.

Además, la formación tenía en cuenta de forma destacada el conocimiento y la generación de los materiales digitales de carácter general y de carácter específico educativo que van a constituir las herramientas habituales del profesorado y del alumnado.

Para ello, el Ministerio, a través de este Instituto, amplió la oferta de sus cursos en Red sobre instrumentos TIC y sus aspectos metodológicos, de experimentación e innovación.

Sin embargo, este proyecto se ha paralizado en 2012 debido a la situación económica del país. El Ministerio de Educación, Cultura y Deporte suprimió dicho programa educativo, una vez presentados los Presupuestos Generales del Estado 2012.

Por lo tanto, aunque no hay infraestructuras en todas las aulas, sí hay algunas aulas dotadas con instalaciones que pueden usar material multimedia como el que se presenta en este proyecto. Y quién sabe, si cuando la situación económica mejore, este programa educativo se recuperará siendo posible que todos los alumnos puedan emplear este tipo de aplicaciones.

#### 4.5. APRENDIZAJE INDIVIDUAL Y COOPERATIVO

En la descripción del objetivo principal de este proyecto se menciona que la aplicación a desarrollar durante este trabajo debe de ayudar a la enseñanza individual pero también a la enseñanza cooperativa, por lo que es interesante conocer las estructuras de las actividades de ambas enseñanzas.

Sin embargo, antes de explicar los tipos de enseñanzas mencionadas anteriormente es preciso entender el concepto de aprendizaje como proceso a través del cual se adquieren o modifican habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio, la experiencia, la instrucción, el razonamiento y la observación.

De esta forma se entiende que la estructura de la actividad en el aprendizaje individualizado, como su propio nombre indica, es individual. Los alumnos realizan las actividades que se proponen y cada uno funciona como punto de referencia para sí mismo. Los criterios de progreso son personales y están basados en el rendimiento propio.

Aunque la forma más habitual de aprendizaje es el individual, en este proyecto también se ha considerado oportuno el empleo del aprendizaje cooperativo. Se habla de estructura de aprendizaje cooperativo cuando se organizan tareas en las que la

cooperación es la condición para realizarlas. Son tareas de aprendizaje que no se pueden realizar si no es colaborando entre los compañeros. No se puede tener éxito si los compañeros no lo tienen pues se liga el éxito propio al éxito del resto.

Para que se produzca aprendizaje en la cooperación los alumnos tienen que conocer la tarea a realizar y compartir entre todos el mismo objetivo. Asimismo, el éxito del trabajo del grupo debe descansar en que todos y cada uno de los alumnos y alumnas del grupo aprendan.

Las ventajas del aprendizaje cooperativo son por un lado, que los alumnos se ayudan y explican, los unos a los otros, los posibles métodos de hacer la tarea y las posibles soluciones que se pueden dar. Estas prácticas comunicativas y sociales desarrollan, a parte de las capacidades específicas de la tarea, facultades relativas a la comprensión, expresión oral y capacidades que tienen que ver con las relaciones de grupo.

También, es necesario que todos se hagan entender para que la actividad progrese y se pueda resolver. De esta forma se asegura que todos han aprendido, ya que la mejor forma para saber si alguien ha captado los conceptos explicados es cuando es capaz de explicarlos a los otros consiguiendo que estos los entiendan.

Por otro lado, en la cooperación entre iguales el que explica o ayuda a otro a resolver un problema tiene más posibilidades de hacerse entender que el profesor. La razón es porque él ha pasado hace menos tiempo que el profesor por la misma dificultad que tiene el compañero y por eso puede entender mejor sus dificultades.

Además, en la cooperación que se crea para resolver el problema, cada alumno del grupo puede observar gran variedad de estrategias, procedimientos, habilidades y técnicas que los otros utilizan para intentar resolver dicho problema.

Por eso, cada alumno ha de intercambiar ideas, defender su punto de vista, argumentarlo, reconsiderarlo tras oír las razones dadas por los demás compañeros, ser capaz de exponer ideas con argumentos y de forma coherente, ser capaz de entender

las de los demás alumnos y rectificar las propias, llegar a acuerdos, contemplar varios puntos de vista, etc.,

Por último, en una organización cooperativa cada alumno se siente miembro del grupo, se da cuenta de que puede ayudar y puede ser ayudado, es consciente de que tiene que tomar en consideración las propuestas de los compañeros si quiere que la suya se tenga en cuenta y, así, avanzar en la resolución de la tarea. Además el éxito o fracaso depende de todos los miembros del grupo con lo cual es algo más manejable que en situación individual en la que el único responsable es uno mismo.

#### 4.6. TRASTORNO GENERAL DE LA COMUNICACIÓN

El lenguaje se considera como el instrumento básico de comunicación emocional y de relación en la especie humana. Por esta razón, se ha decidido que los destinatarios de la aplicación a desarrollar en este proyecto sean personas con trastorno general de la comunicación; para que dicha aplicación les ayude a progresar tanto en la comunicación emocional como en las relaciones con las demás personas.

Los trastornos de la comunicación son trastornos del discurso y del lenguaje que se refiere a problemas en la comunicación y áreas relacionadas. Los retrasos y trastornos pueden ir desde la sustitución de sonidos simples a la incapacidad de comprender o utilizar su lengua materna.

Sus déficits pueden ser causa de una gran variedad de desajustes psicológicos y dichos trastornos pueden ser reactivos a las dificultades de relación y comunicación del niño con los adultos de apego y con los de su entorno social en general, o bien, y dependiendo de la gravedad del déficit, a las alteraciones neurocognitivas que merman las habilidades de análisis, y por tanto de afrontamiento, frente a la gran variedad de situaciones a las que debe hacer frente el niño.

Dentro del DSM (Manual diagnóstico y estadístico de los trastornos mentales) clasifica este trastorno dentro del Eje I, Trastornos clínicos y otros problemas que pueden ser objeto de atención clínica. Y forma parte de Trastornos de inicio en la

infancia, la niñez o la adolescencia. Se diferencian cuatro trastornos relacionados con el habla, referentes a la producción de sonidos.

Los componentes de esta clasificación son:

- Trastorno del lengua expresivo.
- Trastorno mixto del lenguaje receptivo-expresivo.
- Trastorno fonológico.
- Tartamudeo.

En las interacciones sociales, la comunicación verbal y no verbal (mirada, gestos de la cara y las manos; cualidades vocales) incluye la evocación de palabras (acceso y recuperación de la información verbal), relaciones entre las palabras (semántica), formación de frases (sintaxis), dar y recibir feedback, seguir la estructura de la conversación y hablar con soltura, respuesta al contexto, adaptación a los significados y a los acontecimientos externos, respuesta al propio sentido interno de los acontecimientos y control de las propias producciones comunicativas (habilidades metalingüísticas). Todas estas habilidades se alcanzan normalmente a los cinco años. Entre los 2 y 3 años pueden observarse notables déficits en la articulación (producción de los sonidos del habla), en la expresión (producción y uso del lenguaje oral) y en la recepción (comprensión).

## 5. ESTADO DEL ARTE

Antes de pensar en una temática concreta del software a desarrollar, es interesante observar y estudiar aplicaciones y métodos de aprendizaje ya existentes. De esta forma se pueden sacar ideas novedosas y diferentes a lo ya existente.

### 5.1. RECURSOS ANALÓGICOS

La mayoría de recursos educativos hasta hace unos pocos años eran recursos analógicos: libros de texto, juegos, fichas educativas, etc. Por esta razón existen muchísimos procedimientos educativos, y en particular recursos para combatir los trastornos de comunicación, analógicos de los cuales, en este apartado, se va a describir algún ejemplo.

#### - Pictogramas:

Un pictograma es un dibujo convencionalizado que representa un objeto de manera simplificada y permite transmitir una información también convencionalizada.

Los pictogramas no representan ninguna palabra de ningún idioma y, mucho menos, estructuras sintácticas o morfológicas. Eso no impide, sin embargo, que cuando se ve un pictograma, se haga algo que es natural ante cualquier tipo de dibujo: verbalizar lo que se está viendo. De esta forma, se asocia un cierto dibujo con una cierta palabra y con su fonografía.

Por esta razón los pictogramas son utilizados en muchas técnicas para desarrollar un sistema de ayuda a los sujetos con problemas de comunicación. Además son un medio muy práctico, y de fácil construcción ya que se pueden hacer manualmente.

Existen poemas y cuentos con pictogramas como recurso para la lectura, escritura y otras habilidades comunicativas. Se tratan de cuentos o historias contadas con pictogramas, conforme se lee el cuento te vas encontrando con pictogramas que hacen referencia a algún objeto comentado.

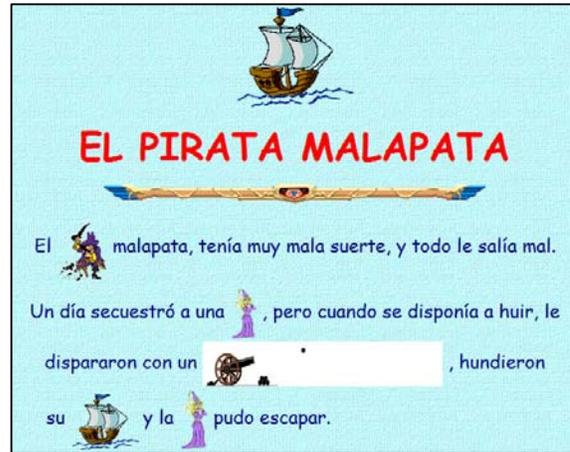


Fig. 2 Cuento con pictogramas

- Tarjetas de relación palabra/imagen



Fig. 3 Tarjetas

Estas tarjetas son una sencilla manera de que el niño o niña se inicie en el proceso de lectura y escritura a través de la relación entre la imagen y la palabra.

El objetivo de las tarjetas es simplemente que el niño encaje las partes del puzle correctamente uniendo un dibujo con su palabra correspondiente.

Con este ejercicio también se desarrolla la atención, observación, concentración y discriminación.

- Tabú

El tabú es un juego de mesa que se juega en equipos y consta de varias cartas, en cada una de las cuales hay una serie de palabras.



Fig. 4 Tarjeta Tabú

El objetivo del juego es conseguir el mayor número de cartas. Para ello, un miembro del equipo debe coger una carta y hacer que sus compañeros adivinen la primera palabra que aparece en la carta dando pistas pero sin decir las demás palabras antes de que se agoste el tiempo marcado.

Con este juego se promueve la expresión oral a la vez que se aprende nuevo vocabulario.

#### - ¿Quién es quién?

El quién es quién es un juego de los años 80 para dos jugadores. Cada jugador dispone de un tablero idéntico con dibujos de personajes identificados por sus nombres y una tarjeta, escogida al azar, con uno de los personajes del tablero.

El objetivo del juego es adivinar el personaje de la tarjeta del oponente. Para ello, los jugadores hacen preguntas acerca del aspecto del personaje cuya respuesta solo puede ser sí o no como por ejemplo: “¿Tu personaje lleva gorro?”. Cuando el oponente responde se tumban los personajes que no cumplen la característica hasta que en el tablero solo queda el personaje del oponente. Gana el jugador que primero consigue adivinar el personaje del oponente.



**Fig. 5** Juego ¿Quién es quién?

Gracias a este juego, los jugadores desarrollan la expresión oral, la comprensión y aprenden nuevo vocabulario.

## 5.2. RECURSOS DIGITALES

En los últimos años, los recursos digitales han tomado un fuerte papel en los procedimientos educativos. Muchos de los recursos analógicos ya existentes se han implementado en formato digital aparte de los nuevos recursos que se han desarrollado únicamente para este formato. Simplemente hace falta echar un vistazo

por la web para encontrar numerosos ejemplos. A continuación, se muestra algún ejemplo de estos recursos digitales.

- Busca palabras

Un ejemplo en formato digital de las tarjetas de relación palabra-imagen explicadas anteriormente es este juego encontrado en la página <http://childtopia.com>.



**Fig. 6** Captura de pantalla juego busca la palabra

El programa te indica un objeto y el jugador tiene que seleccionar el dibujo al que corresponde.

Cabe destacar, por un lado, que el aspecto del juego es mucho más atractivo que en el formato analógico y por otro lado, que en el formato digital se le ha añadido sonido, ayuda importante para el jugador.

- Quién es quién

Otro ejemplo de cómo se han implementado los recursos analógicos en formato digital es el juego anteriormente explicado “¿Quién es quién?”. De este juego se han encontrado varias versiones en la red.



Fig. 7 Interfaz ¿Quién es quién? 1

Las versiones digitales de este juego tienen aspectos muy diferentes las unas a las otras. Unas tienen un aspecto muy sencillo y una forma de jugar muy intuitiva mientras que otras son un poco más complejas. Resulta llamativo que las más complejas suelen coincidir con las versiones móviles.



Fig. 8 Interfaz ¿Quién es quién? 2

hora de expresarse.

Sin embargo, todas las versiones encontradas tienen en común que las preguntas están ya redactadas y para formularlas simplemente hay que seleccionar la pregunta. Otra cosa que tienen en común es que las preguntas realizadas sobre el personaje del jugador no pueden ser contestadas por este, sino que son contestadas automáticamente. Todo esto hace que el jugador tenga poca libertad a la

- Los cohetes que riman

Aunque es un juego en inglés es un buen ejemplo de recursos educativos desarrollado únicamente en formato digital.

Este juego consiste en disparar sobre los cohetes cuyo texto rime con una palabra que se encuentra debajo del avión.

Es un juego muy atractivo y divertido con el que el jugador trabaja la lectura y la pronunciación.



Fig. 9 Captura de pantalla juego cohetes que riman

## 6. ENTORNO DE TRABAJO Y LENGUAJE DE PROGRAMACIÓN

El entorno de trabajo en el que se va a desarrollar este proyecto es Adobe Flash Professional, más concretamente en sus versiones CS5 y CS5.5. Durante este apartado se van a explicar sus fundamentos básicos y su lenguaje de programación as3.0.

### 6.1. ADOBE FLASH PROFESSIONAL



**Fig. 10** Icono de Flash

Adobe Flash Professional es uno de los programas más populares de la casa Adobe, junto con sus programas hermanos Adobe Illustrator y Adobe Photoshop. Se trata de una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código mediante un lenguaje de scripting llamado ActionScript para crear principalmente elementos multimedia e interactivos para Internet.

Se ha escogido Adobe Flash Professional para el desarrollo de este proyecto porque es un estudio de animación que trabaja sobre fotogramas y está destinado a la producción y entrega de contenido multimedia interactivo para diferentes audiencias sin importar la plataforma. Justo lo que se está buscando para este proyecto.

Sin embargo, para que un archivo desarrollado con Flash se pueda ejecutar en una plataforma cualquiera esta debe tener instalado Adobe Flash Player. En sentido estricto, Flash es el entorno de desarrollo y Flash Player es el reproductor utilizado para visualizar los archivos generados con Flash. En otras palabras, Adobe Flash crea y edita las animaciones o archivos multimedia y Adobe Flash Player las reproduce.

Pero esto no es un problema ya que este reproductor ya lo tienen incorporado las distintas versiones de los sistemas operativos que se lanzan hoy por hoy. Como también los navegadores web, como Internet Explorer, Firefox, Netscape, Google Chrome, entre otros, que traen consigo los plugins necesarios para ejecutar un archivo Flash. Como consecuencia, ya sea un video o una animación, la mayoría de plataformas están preparadas para reproducir con fluidez este contenido que nos brinda Adobe Flash.

### 6.1.1. VERSIONES EMPLEADAS

En el desarrollo de este proyecto se han empleado dos versiones de Adobe Flash Professional; Las versiones CS5 y CS5.5.

Se comenzó empleando la versión CS5 porque era la única versión de Flash que soportaba el lenguaje de programación ActionScript 3.0 que, como ya se verá más adelante, presenta ventajas sobre las versiones anteriores de ActionScript.

Sin embargo, al poco tiempo, sacaron al mercado la siguiente versión de Flash, la versión CS5.5. Esta presentaba todas las características de la versión anterior pero, a demás, incluía la opción de exportar la aplicación en formato .apk. (Extensión de un paquete para el sistema operativo android). Con esta novedad también se podía implementar la aplicación para dispositivos android, por lo que se decidió desarrollar el proyecto en esta nueva versión.

### 6.1.2. INTERFAZ

Al abrir un nuevo documento de Flash la pantalla que aparece es similar a la mostrada en la figura 11 que se puede dividir en varias secciones.

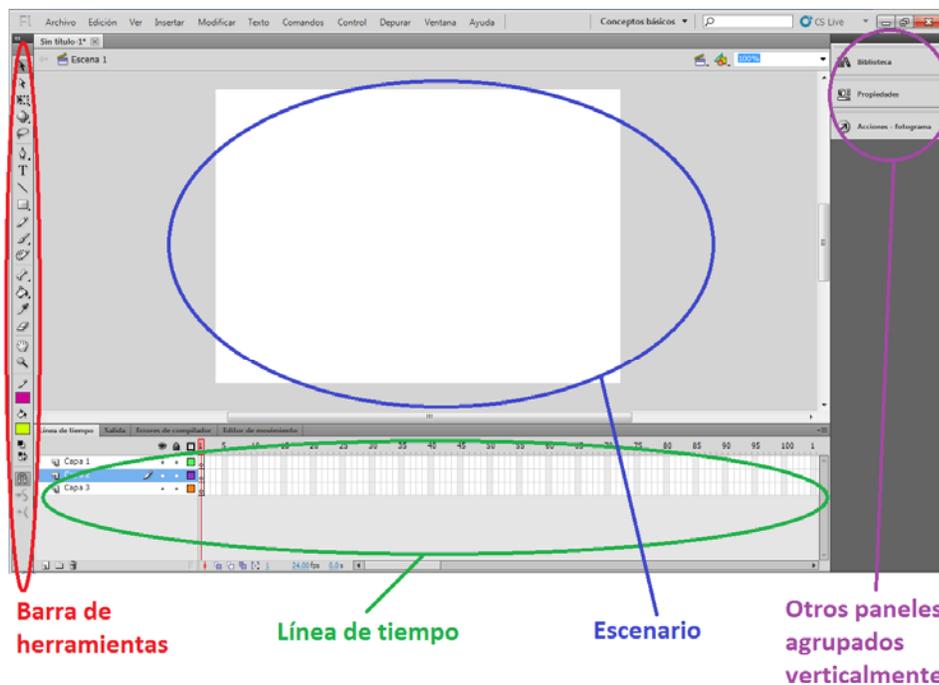


Fig. 11 Parte de la interfaz Flash

↪

### 6.1.2.1. Escenario

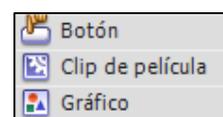
En la escena se sitúan los elementos que componen la película. Estos elementos se pueden importar o se crean mediante las herramientas de dibujo que se explican más adelante. Sin embargo, para reducir el tamaño de la película hay que convertir todos los elementos en símbolos.

Un símbolo es todo lo que queramos almacenar para su posterior uso y/o animación. A la representación de un símbolo en la escena se llama instancia. Además, si se hace un cambio en el símbolo se modificarán todas las instancias de ese símbolo.

No obstante, también se puede modificar una instancia de un símbolo sin modificar el original. Una vez que la instancia se encuentra en la escena, se puede escalar y rotar con la herramienta Transformación Libre y modificar el color y sus propiedades sin que el símbolo original se modifique.

Hay tres tipos de símbolos. Todos ellos tienen su propia línea de tiempo. Se distinguen por su comportamiento:

- Botón: Tiene su propia línea de tiempo, que contiene 4 fotogramas, cada uno de ellos con un comportamiento preestablecido por Flash.



**Fig. 12 Botón, Mc y gráfico**

- Clip de película (MovieClip): Su línea de tiempo no tiene un número fijo de fotogramas. Es el símbolo más utilizado, debido a su versatilidad.
- Gráfico: Tienen una duración de un fotograma. Recoge la mayor parte de los elementos de diseño, tales como líneas, formas, imágenes, etc. Sin embargo, estos símbolos no son compatibles con ActionScript 3.0 así que se recomienda no trabajar con ellos.

Para crear símbolos a partir de gráficos se selecciona lo dibujado y se pulsa F8. Al hacer esto se abre una ventana. En ella, se elige el tipo de símbolo, su nombre y, al pulsar sobre el botón aceptar, se creará el símbolo en la biblioteca dejando una instancia en la escena.

#### 6.1.2.2. Barra de herramientas

En la ventana de herramientas se encuentran todos los útiles necesarios para la creación de gráficos. Dichas herramientas se pueden dividir en tres grupos:

- Herramientas de dibujo y selección: Flecha, Subselección, Transformación libre, rotación 3D, Lazo, Pluma, Texto, Línea, Rectángulo, Lápiz, Pincel, Deco, Hueso, Bote de Tinta, Cuentagotas y Borrador.



**Fig. 13 Herramientas de dibujo y selección**

- Herramientas de visualización: Mano y Zoom.



**Fig. 14 Herramientas de visualización**

- Herramientas de colores: Color de trazo, Color de relleno, Blanco y Negro, Sin color e Intercambiar colores.



**Fig. 15 Herramientas de colores**

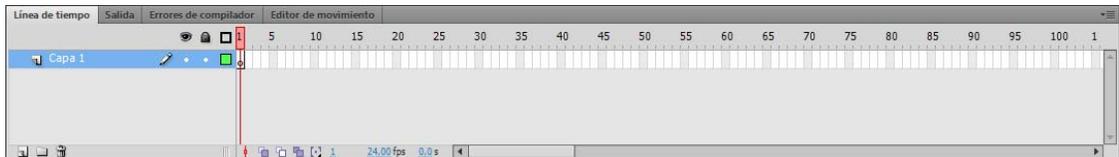
- Herramientas de opciones: Son modificadores especiales para cada herramienta, si es que los tiene.



**Fig. 16 Herramientas de opciones**

### 6.1.2.3. Línea de tiempo

El procedimiento que usa Flash para hacer animaciones es como en el cine, se basa en una sucesión de fotogramas que se agrupan en la línea de tiempo. Así pues, la línea de tiempo es la encargada de organizar y controlar el contenido de una película a través del tiempo.

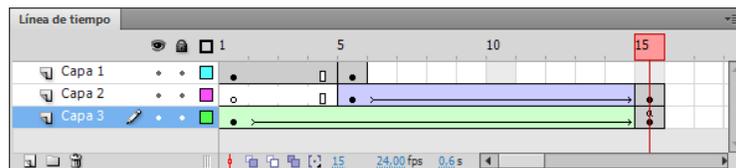


**Fig. 17 Línea de tiempo**

Para navegar por la línea de tiempo, basta con mover el marcador, compuesto por el cuadro rojo de la parte superior y la línea roja que atraviesa todas las capas.

Tipos de fotogramas:

- Fotograma clave (F6): Es un fotograma que tiene contenido. Tienen un punto negro en la línea de tiempo. En la Fig.13 el fotograma 1 de la capa 1.
- Fotograma clave vacío (F7): Carece de contenido. En la línea de tiempo aparecen con un círculo blanco. En la Fig.13 el fotograma 1 de la capa 2.



**Fig. 18 Línea de tiempo 2**

- Fotograma normal (F5): Estos fotogramas tienen el mismo contenido que el fotograma clave que les precede. Su misión es prolongar el contenido del fotograma clave. El último de los fotogramas normales de una

secuencia presenta un cuadrado gris. En la Fig.13 el fotograma 4 de la capa 1.

- Fotograma de animación de movimiento: Son fotogramas que representan animaciones vectoriales de movimiento o cambios de color (interpolación clásica). Aparecen unidos por una flecha, y son de color azul. En la Fig.13 el fotograma 10 de la capa 2.
- Fotograma de animación de forma: Son fotogramas que representan cambios de forma vectorial en los dibujos. Aparecen unidos por una flecha, y son de color verde. En la Fig.13 el fotograma 10 de la capa 3.
- Fotogramas con acciones: Este tipo de fotogramas contienen acciones asociadas que se ejecutarán cuando la cabeza lectora pase por ellos. Se distinguen porque tienen una “a” en la línea de tiempo. En la Fig.13 el fotograma 15 de la capa 3.

#### 6.1.2.4. Otros paneles agrupados

En esta agrupación, puede haber una gran variedad de paneles pero se van a comentar los tres paneles más importantes y que por lo tanto, más se emplean.

##### 6.1.2.4.1. Biblioteca

Desde la Biblioteca se gestionan todos los símbolos creados e importados. También se puede visualizar una vista previa de cada símbolo, así como seleccionar uno y editarlo. Es recomendable tener los símbolos agrupados en carpetas para una mejor organización.

En el momento en que creamos un símbolo en nuestra escena, éste pasa a formar parte de la biblioteca. Sin embargo, no se permite nombrar a dos símbolos con el mismo nombre.

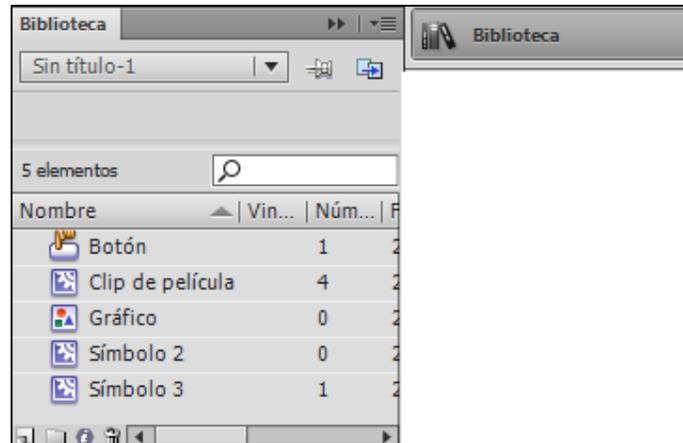


Fig. 19 Biblioteca

#### 6.1.2.4.2. Propiedades

En la ventana Propiedades se muestran todos los campos que se pueden modificar de las herramientas. Su contenido pues, varía en función del elemento que se tenga seleccionado.

Si no hay ningún elemento seleccionado, se muestran las propiedades de la escena como se muestra en la figura 20.

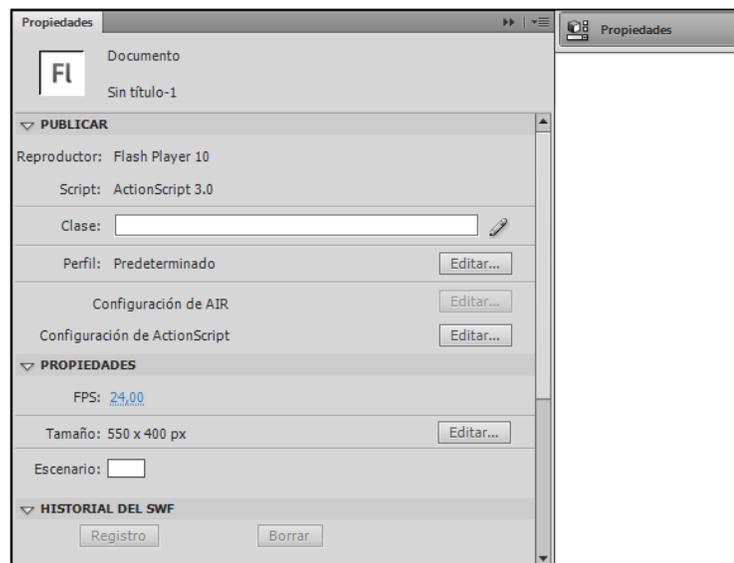


Fig. 20 Propiedades

#### 6.1.2.4.3. Acciones

En esta ventana es donde se escribe el código de programación, que es el que permitirá añadir interactividad al documento.

Esta venta también contiene varios botones de ayuda como por ejemplo, Revisar sintaxis, Formato automático, Aplicar comentario, etc.

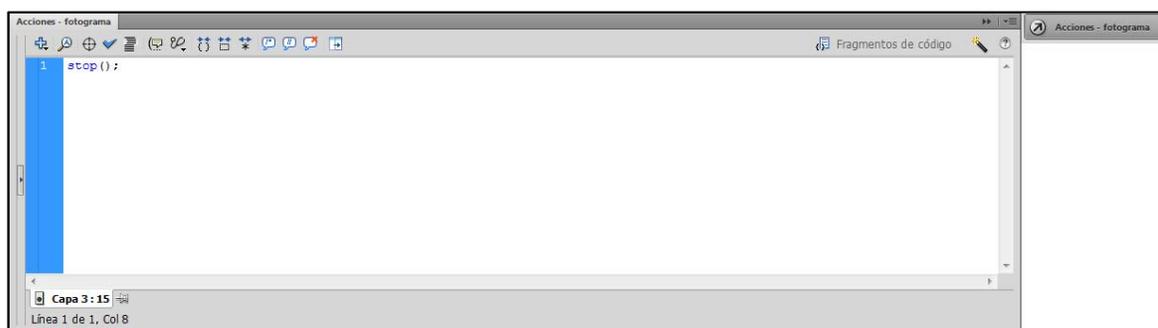


Fig. 21 Panel de acciones

#### 6.1.3. TIPOS DE ARCHIVOS

El archivo con extensión **.fla** es el que contiene la película sin compilar, es decir, el archivo sobre el que se trabaja en Flash y que permite tanto editar, como visualizar cómo está hecha la película.

Pulsando las teclas CTRL + ENTER, además de reproducir la película, se genera un archivo con el mismo nombre y extensión **swf**. Este archivo contiene la película compilada y necesita el reproductor de Flash para su reproducción.

Flash también permite la creación automática de otros archivos cada vez que se pulse las teclas MAYUS + F12 o se elija la opción menú Archivo->Publicar. Pero antes se deben seleccionar los tipos de archivos que se desean que se generen. Esto se hace en la ventana que se abre al pulsar sobre "Archivo"->"Configuración de la Publicación" que se muestra en la figura 22.

Cada vez que se publique la película se obtendrán todos los tipos de archivos que se han seleccionado en esta ventana.

Existen las siguientes opciones: Flash (.swf), HTML (.html), GIF (.gif), JPEG (.jpg), PNG (.png), Windows (.exe), y Macintosh (.app).

Cada casilla activada genera su propia pestaña que se podrá configurar con varias opciones.

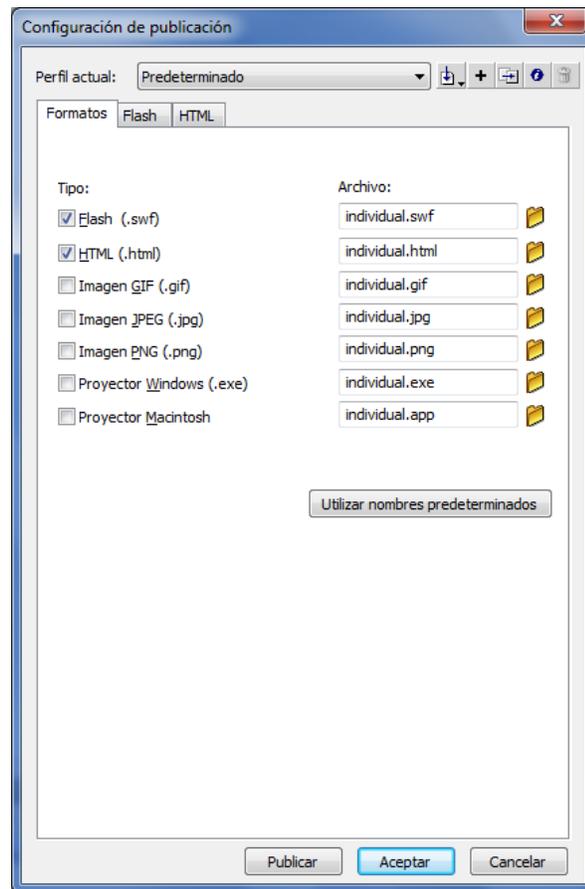


Fig. 22 Configuración de la publicación

## 6.2. ACTION SCRIPT

ActionScript es el lenguaje de programación de la Plataforma Adobe Flash desarrollado como una forma para que los desarrolladores programen de forma más interactiva. La programación con ActionScript permite mucha más eficiencia en las aplicaciones de la plataforma Flash para construir animaciones de todo tipo, desde simples a complejas, ricas en datos e interfaces interactivas.

### 6.2.1. VERSIÓN EMPLEADA

Como se ha mencionado en apartados anteriores, la versión empleada de ActionScript es la versión 3.0. Actualmente esta es la versión más extendida debido a que significó una mejora en el manejo de programación orientada a objetos al ajustarse mejor al estándar ECMA-262 (estándar sobre las especificaciones que debe cumplir el lenguaje de programación orientado a objetos).

ActionScript 3.0 ofrece, por tanto, un modelo de programación robusto que resultará familiar a los desarrolladores con conocimientos básicos sobre programación orientada a objetos.

Una ventaja que presenta sobre versiones anteriores es que se ha diseñado para facilitar la creación de aplicaciones muy complejas con conjuntos de datos voluminosos y bases de código reutilizables y orientadas a objetos. Aunque no se requiere para el contenido que se ejecuta en Adobe Flash Player, ActionScript 3.0 permite introducir unas mejoras de rendimiento que sólo están disponibles con AVM2, la nueva máquina virtual. El código ActionScript 3.0 puede ejecutarse con una velocidad diez veces mayor que el código ActionScript heredado.

### 6.2.2. INTRODUCCION A AS3.0

#### 6.2.2.1. TIPOS DE DOCUMENTOS

Al abrir el programa Adobe Flash Professional lo primero que aparece es una pantalla como la que se muestra en la figura 23. Normalmente se trabaja con los documentos “ActionScript 3.0” o “AIR for Android” que tienen la interfaz explicada en el apartado anterior 6.1.2. Sin embargo, hay otros documentos de código de ActionScript.



Fig. 23 Pantalla inicial Flash CS5

- “Archivo ActionScript”: Es un documento de ActionScript vacío cuya interfaz es solamente el panel de acciones. En él se puede escribir tanto código suelto que luego se incluye en el archivo de Flash como clases de ActionScript e interfaces.
- “Clase de ActionScript 3.0”: Este documento, al igual que el anterior, solo tiene como interfaz el panel de acciones. Sin embargo este panel no se encuentra vacío. Contiene la estructura común a todas las clases de ActionScript 3.0. Este documento es útil para no empezar a escribir una clase desde cero.
- “Interfaz de ActionScript 3.0”: Al igual que los otros dos documentos, este documento también tiene como interfaz únicamente el panel de acciones. Pero este documento posee la estructura común a todas las interfaces de ActionScript 3.0. Para el desarrollo de este proyecto no ha sido necesario el empleo de este tipo de documento.

Como se puede observar en la figura 22, existen otros tipos de documentos de código. Sin embargo, al emplear estos documentos códigos diferentes al que se va a emplear en este proyecto, no se van a explicar.

También es interesante destacar que los archivos generados por estos documentos son archivos con extensión .as. Así los documentos de Flash crean archivos .fla mientras que los de ActionScript crean archivos .as.

#### 6.2.2.2. OBJETOS

Dado que ActionScript es un lenguaje de programación orientado a objetos, los objetos son necesarios para programar. Estos se pueden crear básicamente de dos formas.

La primera manera de crear un objeto es a partir de un símbolo creado como se ha explicado en el apartado 6.1.2.1. Cuando se tiene una instancia de un símbolo, seleccionar esta y en el panel de propiedades rellenar el campo “nombre de instancia” como se muestra en la figura 24.

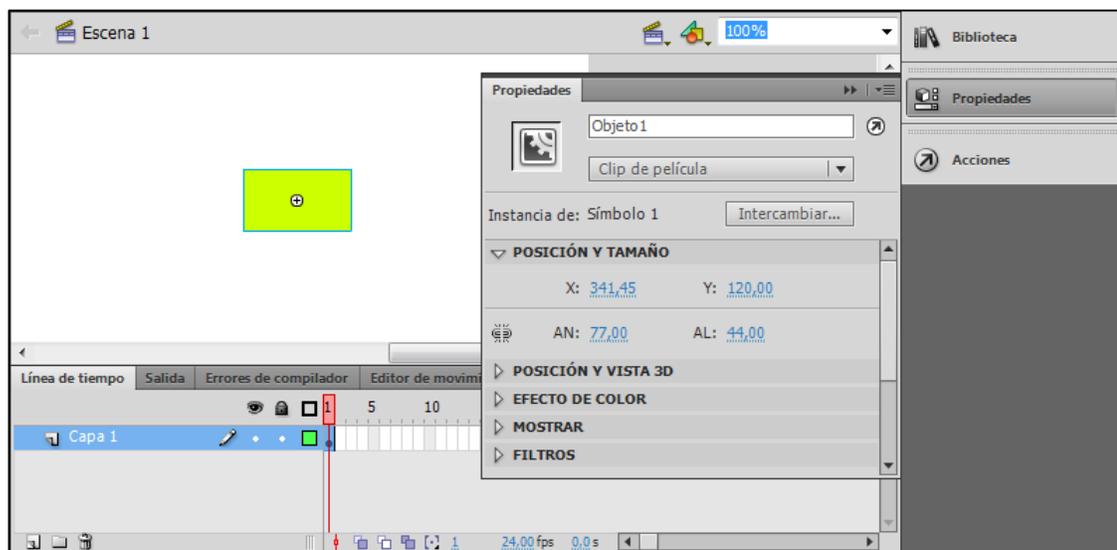


Fig. 24 Crear objetos a partir de un símbolo

Otra manera de crear un objeto es mediante código como se muestra en la figura 25.

```
1 var objeto1:Object = new Object();
```

Fig. 25 Definición de un objeto

Una vez creado un objeto, de cualquiera de las dos formas mencionadas anteriormente, se le pueden modificar sus propiedades también mediante código. Así se le pueden modificar las propiedades de posición, de tamaño, de escala, de transparencia, etc. como se ve en la figura 26.

```
1 //Posicion:
2 Objeto1.x = 10; //posición eje x
3 Objeto1.y = 100; //posición eje y
4
5 //Tamaño:
6 Objeto1.width = 65; //ancho
7 Objeto1.height = 120; //alto
8
9 //Escala:
10 Objeto1.scaleX = 0,6; //empequeñece
11 Objeto1.scaleY = 10; //agranda
12
13 //Transparencia:
14 Objeto1.alpha = 0,7; //70% de transparencia
```

Fig. 26 Modificación de las propiedades de un objeto

### 6.2.2.3. VARIABLES

Las variables son datos que se guardan en la memoria del equipo para ser utilizados. Pueden ser números, letras, valores de verdadero o falso (Booleanos), etc. En ActionScript 3.0, una variable se compone realmente de tres partes distintas:

- El nombre de la variable.
- El tipo de datos que puede almacenarse en la variable.
- El valor real almacenado en la memoria del equipo.

La forma de definir una variable es mediante una sentencia como la que se muestra en la figura 27.

```
1 var NombreVariable:TipoVariable = valorInicial;
```

Fig. 27 Definición variable

Como es lógico, existen varios tipos de variables dependiendo del tipo de datos que almacenen. A continuación se explican unos ejemplos.

- Variables numéricas: son variables que guardan valores numéricos. Incluyen tres tipos de datos:
  - int: almacena un número entero, sin coma.
  - Number: almacena cualquier número.
  - uint: almacena un número entero sin signo, es decir, que no puede ser negativo.
- Variables booleanas: almacena un valor verdadero (true) o falso (false).
- Variables string: almacena un valor de texto.
- Variables MovieClips: almacena un movieClip.

```
1 //variables numericas
2 var numero1:int = 10;
3 var numero2:Number = 10;
4 var numero3:uint = 10;
5
6 //variable booleana
7 var booleano:Boolean = true;
8
9 //variable string
10 var numero3:String = "a";
11
12 //variable MovieClip
13 var Movieclip1:MovieClip;
```

Fig. 28 Tipos de variables

#### 6.2.2.4. MÉTODOS Y FUNCIONES

Un método es la forma de llamar a una función predefinida en flash o escrita por el programador.

Una función es un grupo de instrucciones que realizan una tarea específica. La estructura de una función en ActionScript 3.0 es la que se muestra en la figura 29.

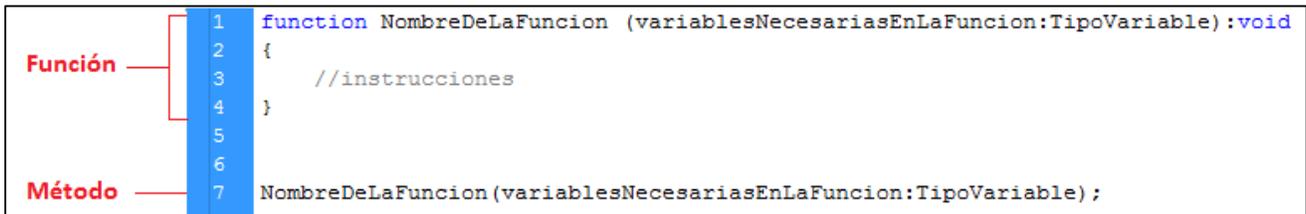


Fig. 29 Método y función

#### 6.2.2.5. EVENTOS

Algunos programas informáticos sencillos no son más que unos cuantos pasos que el ordenador ejecuta, tras los cuales finaliza el programa. Sin embargo, los programas de ActionScript se han diseñado para continuar ejecutándose, esperando los datos introducidos por el usuario u otras acciones.

Los eventos son las acciones que ActionScript espera y que se relacionan con la interacción del usuario (hacer clic en un botón, presionar una tecla del teclado, etc.) pero también con otro tipo de eventos como por ejemplo, cuando se finaliza la carga de una imagen externa.

A la hora de definir un evento se tienen que tener en cuenta tres elementos importantes:

- El origen del evento: ¿en qué objeto va a repercutir el evento?
- El evento: ¿qué va a suceder, a qué se va a responder?

- La respuesta: ¿qué pasos hay que llevar a cabo cuando ocurra el evento?  
Normalmente es una función.

Para definirlo también hay que emplear el método `addEventListener`. Este método utiliza dos parámetros. En primer lugar, el nombre del evento específico al que se desea responder y en segundo lugar, el nombre de la función de respuesta al evento.

```
Evento ----- 1 OrigenEvento.addEventListener(TipoEvento.EVENT_NAME, Respuesta);
                2
Respuesta ----- 3 function Respuesta(ObjetoEvento:TipoEvento):void
                4 {
                5     // Acciones
                6 }
```

Fig. 30 Eventos

#### 6.2.2.6. ESTRUCTURAS “DE CONTROL”

##### 6.2.2.6.1. BUCLE FOR

El bucle `for` permite repetir una variable para un rango de valores específico. Se deben proporcionar tres expresiones en una sentencia `for`: una variable que se establece con un valor inicial, una sentencia condicional que determina cuándo termina la reproducción en bucle y una expresión que cambia el valor de la variable con cada bucle.

```
1 for (i = 0; i < 5; i++)
2 {
3     trace(i);
4 }
```

Fig. 31 Bucle for

#### 6.2.2.6.2. IF

La sentencia condicional if..else permite comprobar una condición y ejecutar un bloque de código si dicha condición se cumple, o ejecutar un bloque de código alternativo si la condición no se cumple.

```
1  if (x > 20)
2  {
3      trace("x is > 20");
4  }
5  else
6  {
7      trace("x is <= 20");
8  }
```

Fig. 32 If

#### 6.2.2.6.3. SWITCH

La sentencia switch evalúa una expresión, en lugar de probar una condición para un valor booleano, y utiliza el resultado para determinar el bloque de código que debe ejecutarse. Los bloques de código empiezan por una sentencia case y terminan con una sentencia break.

```
1  switch (numHoras)
2  {
3      case 0 :
4          trace("Ninguna hora");
5          break;
6      case 1 :
7          trace("Una hora");
8          break;
9      case 2 :
10         trace("Dos horas");
11         break;
12     case 3 :
13         trace("Tres horas");
14         break;
15 }
```

Fig. 33 Switch

### 6.2.2.7. CLASES

Una clase es simplemente la definición de un tipo de datos. Todos los objetos de una clase poseen las mismas características. Es pues, como una especie de plantilla para todos los objetos de un mismo tipo de datos. El uso de clases permite la reutilización con mayor facilidad del código.

Una manera de crear una clase es al crear un símbolo. En la ventana que se muestra en la figura 34, que aparece a la hora de crear un símbolo, existe la posibilidad de crear una clase que contenga las características de ese símbolo simplemente seleccionando la opción “Exportar para ActionScript”. La clase tendrá el mismo nombre que el símbolo y estará construida cuando se presione la opción “aceptar” de dicha pantalla.

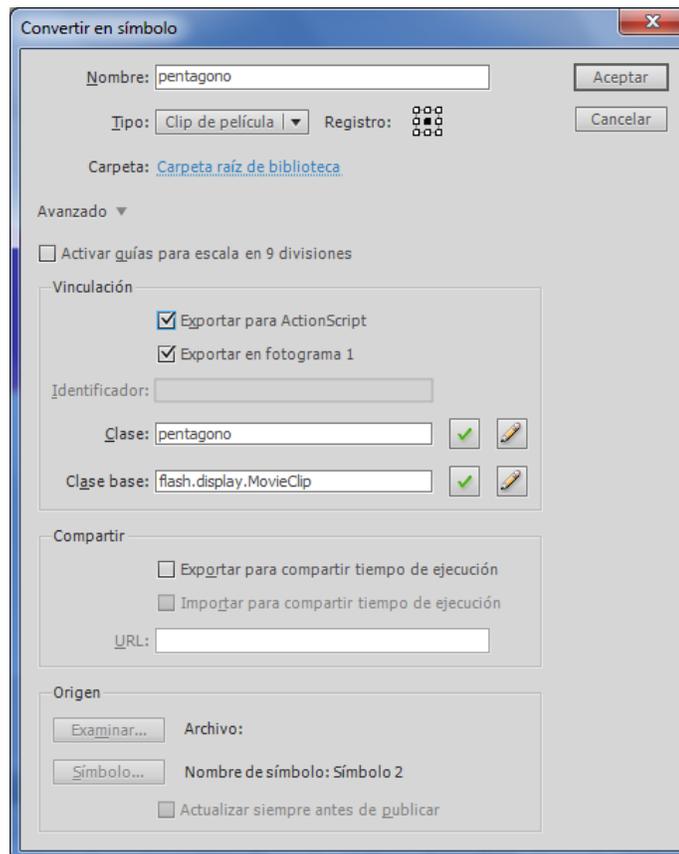


Fig. 34 Crear clase

La otra manera de crear una clase es mediante código. Para ello, se abre un tipo de documento “Clase de ActionScript 3.0” en el cual aparece el código común a todas las clases de ActionScript 3.0, exceptuando el nombre de la clase. En el caso de la figura 35, el nombre de la clase es “clase1”.

En ActionScript 3, todas las clases tienen que tener 3 componentes, Una definición de paquete (package), una declaración de la clase, y un cuerpo de la clase con un método constructor.

```

1 package {
2
3     public class clase1 {
4
5         public function clase1() {
6             // código constructor
7         }
8
9     }
10
11 }

```

**Fig. 35 Estructura de una clase**

- Definición de paquete: La primera línea de una clase dice al compilador a que paquete o grupo de clases es al que pertenece. Es el nombre que sigue a la palabra “package”.
- Declaración de clase: En la tercera línea la palabra clave class dice al compilador que una clase está por ser definida. Esta, está precedida por una palabra clave de atributo y sigue con el nombre de la clase.

Atributo	Definición
dynamic	Permite que se adhieran propiedades a las instancias en tiempo de ejecución.
final	No puede ser extendido por otra clase.
internal (default)	Visible a las referencias dentro del paquete actual.
- C public	Visible a referencias en cualquier lugar.

**Tabla 1**

uerpo de la clase con el método constructor: Todo el código que está contenido en las llaves que están después de la declaración de la clase es considerado el cuerpo de la clase.

Cada clase tiene un método constructor. El método constructor es un método que se llama automáticamente en el momento en el que se genera la instancia. Cuando usas la palabra clave new para crear una instancia de una clase, lo que realmente estás haciendo es llamando al método constructor de la clase. Cualquier código incluido en el método constructor es ejecutado cada vez que la clase es instanciada.

Una vez comentadas la función y la estructura de una clase, es interesante mencionar cómo se llama a una clase desde el documento Flash principal.

Una de las formas es vinculando la clase al documento Flash rellenando el campo "Clase" del panel "Propiedades" con el nombre de la clase.

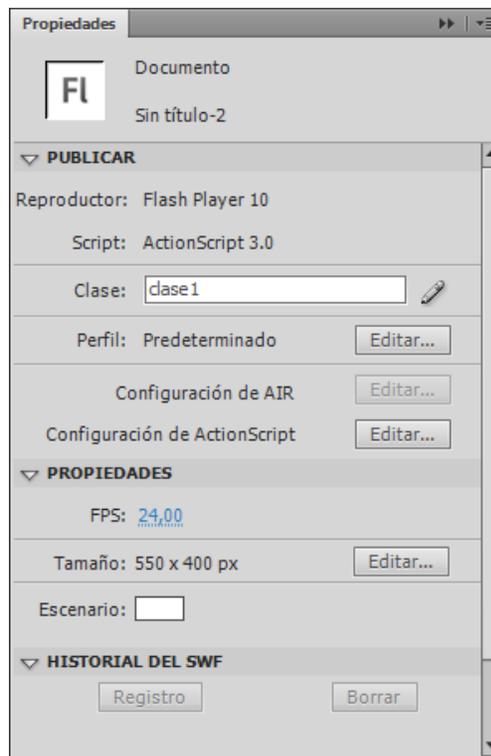


Fig. 36 Llamar a clase 1

Sin embargo, la manera más interesante de emplear las clases desarrolladas es creando variables del tipo de la clase programada. Estas variables se definen de la misma manera que se ha explicado en el apartado 6.2.2.3 y poseen todas las características definidas en la clase.

```
1 var objeto1:clase1 = new clase1();
```

**Fig. 37 Llamar a clase 2**

Para que no de error ninguna de las maneras de emplear las clases en el documento Flash principal, es necesario que ambos documentos, .fla y .as, estén guardados en el mismo lugar y que el nombre del documento .as sea el mismo que el nombre de la clase.

## 7. APLICACIÓN

En este apartado se va a describir la aplicación que se ha desarrollado durante este proyecto una vez investigadas las características que debe cumplir para ser una buena aplicación multimedia educativa, observado el estado del arte y familiarizado con el entorno de trabajo.

Para explicar dicha aplicación se siguen las fases que se han llevado a cabo para el desarrollo del proyecto y que se han descrito en el apartado 2.

### 7.1. TEMÁTICA ESCOGIDA

Teniendo en cuenta las aplicaciones ya existentes que se han mencionado en el estado del arte y los principios que debe cumplir una aplicación para que sea una buena aplicación educativa se ha decidido que la aplicación se va a basar en el juego ¿Quién es quién?

Se ha escogido este tema porque, a pesar de que existen varias versiones en la web, todas se basan en el juego analógico y no aportan nada nuevo respecto al aspecto educativo. Pues, como se ha mencionado en el apartado 5.2, las preguntas en todas las ocasiones están construidas y el juego no deja responder a las preguntas que se hacen sobre el personaje del jugador. Por lo tanto, los jugadores en todos los casos tienen poca libertad para expresarse.

También, este es un buen juego para la expresión y la comprensión. Por el lado de la expresión, el jugador tiene que formar preguntas y relacionar lo que quiere preguntar con la forma de preguntarlo. Pues tiene que fijarse en las fichas de los personajes que le quedan sin eliminar para elegir la pregunta que debe formular.

Por el lado de la comprensión, lógicamente el jugador tiene que comprender la pregunta hecha por el otro jugador para responder correctamente. Pero lo más

interesante de este juego es que el jugador tiene que comprender la respuesta dada a la pregunta realizada por él para eliminar correctamente las fichas de los personajes.

Aunque en su versión analógica este es un juego competitivo en el que dos jugadores compiten por ser el primero en adivinar el personaje del otro, en la versión que se va a implementar en este proyecto esto se va a modificar.

Como se ha indicado en el apartado de introducción, se va a llevar a cabo un aprendizaje individual y colaborativo. Lógicamente, ambos no pueden aparecer conjuntamente por lo que se ha decidido hacer dos aplicaciones. Una en la que el jugador haga preguntas a la máquina y esta “elija” al personaje y contesta. Y otra aplicación donde un jugador haga preguntas a su compañero y este otro elija al personaje y conteste.

Con la primera aplicación se consigue que los jugadores aprendan la dinámica del juego, el vocabulario, etc. Mientras que con la segunda aplicación ambos jugadores pueden aprovechar las ventajas de un aprendizaje colaborativo como las que se han explicado en el apartado 4.5.

## 7.2. PLATAFORMAS

Este proyecto se va a implementar para dos plataformas, PC y tableta, por lo que hay que tener en cuenta varios aspectos.

Para que la aplicación funcione en ambas plataformas correctamente ambas tienen que tener instalado Adobe Flash Player. En el PC, como se ha mencionado anteriormente, esto no es problema ya que normalmente viene instalado en las distintas versiones de los sistemas operativos y de los navegadores.

Por otro lado, hay que tener en cuenta la resolución de las pantallas de ambas plataformas a la hora de diseñar los elementos de la aplicación como por ejemplo el tamaño de los botones, de las letras, etc.

Para la aplicación se ha escogido una resolución de 1280x800 píxeles. Esta coincide con la resolución de la tableta y, además, es una buena resolución para un PC de sobremesa o portátil, que son los que se van a emplear, ya que actualmente suelen tener esta resolución o mayor.

### 7.3. PRIMER BOCETO

Inicialmente se ha hecho un boceto de la aplicación que da una idea de lo que se pretende con ella y de la interfaz que esta tendrá.

Este boceto sólo se va a desarrollar para el caso individual, en el que el usuario pregunta a la máquina. Una vez se tenga claro cómo debe de ser el juego individual el juego colaborativo es muy “similar”. Basta con cambiar las decisiones que toma la máquina por opciones para el otro jugador.

El funcionamiento que se ha pensado en un principio para el juego consiste en que el usuario tenga botones con palabras unidas a pictogramas y que al pulsar estos botones se vayan construyendo frases. Estas frases serán frases afirmativas si el personaje posee la característica que se menciona en la frase, o frases negativas si no es así. Por ejemplo: si el jugador pulsa sobre los botones “lleva” y “gorro” se formará la frase “El personaje lleva gorro” si efectivamente el personaje lleva gorro mientras que si esto no es así se escribirá la frase “el personaje no lleva gorro”.

Para eliminar las fichas en el boceto se ha decidido que estas se eliminen automáticamente según las características del personaje aunque esto de poca libertad al jugador.

A la hora de diseñar la interfaz del boceto, lo primero que se han diseñado han sido los elementos gráficos. Se ha empezado con los más importantes, las fichas de los personajes. Para realizarlas se ha decidido emplear el mismo entorno de trabajo ya que con las herramientas de dibujo explicadas en el apartado 6.1.2.2 es suficiente.

Primeramente, se ha diseñado cada componente de la cara por separado y cada uno de ellos se ha convertido en un Clip de película.

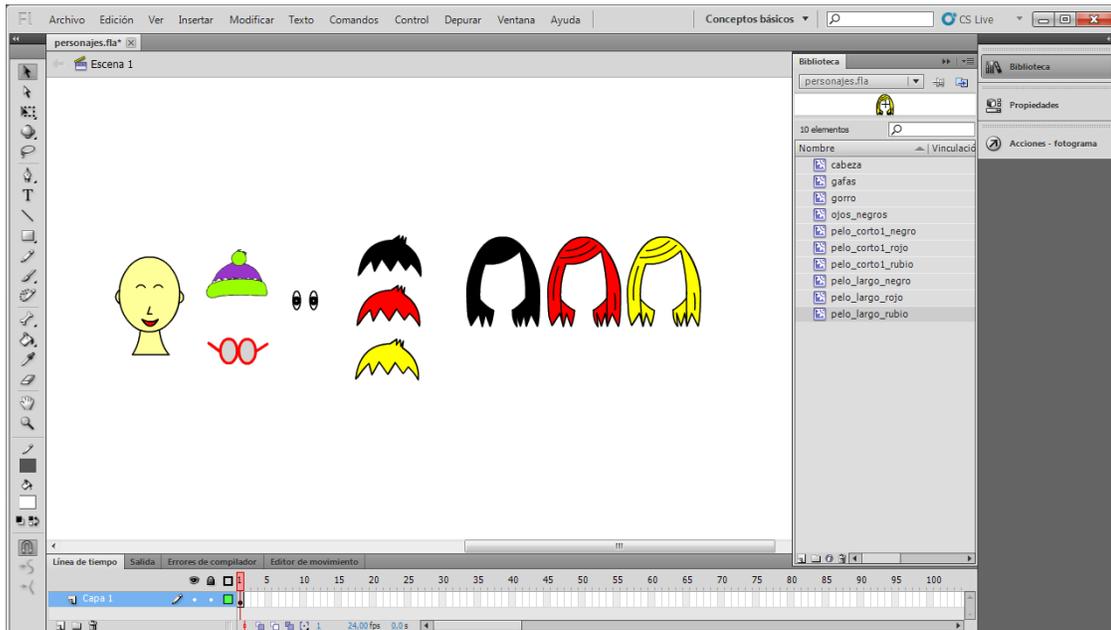


Fig. 38 Componentes de las fichas

A continuación se han combinado todos los elementos formando varios personajes diferentes entre sí. A cada uno de ellos se le ha añadido un nombre y un óvalo de un color aleatorio alrededor para dar aspecto de ficha y cada ficha se ha convertido en un Clip de película. Como resultado se han obtenido las fichas que se muestra en la figura 39.



Fig. 39 Fichas

Los siguientes elementos que se han diseñado han sido los botones que debe de pulsar el usuario. Como se trata de un boceto, se ha decidido elaborar unos botones simples que consisten simplemente en rectángulos con texto y un pictograma dentro. Los botones se han coloreado de diferente color según las palabras que contienen, verbo o complemento de la oración.



Fig. 40 Botones boceto

Por último, se han diseñado las palabras de las frases. Estas están formadas por letras grandes, coloreadas siguiendo la misma regla que los botones y con pictogramas encima.

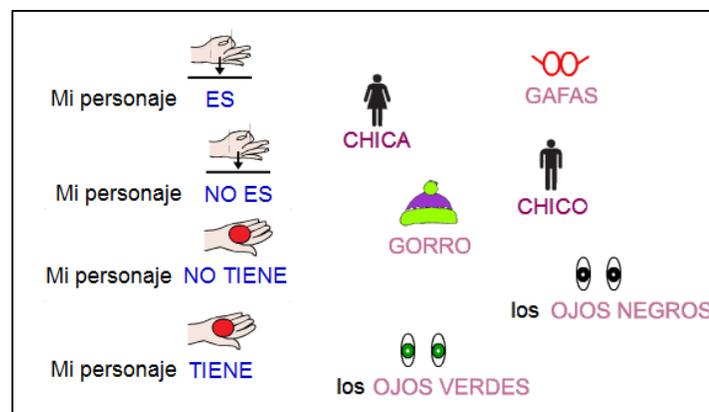


Fig. 41 Palabras boceto

El aspecto general del juego queda como se muestra en la figura 42.

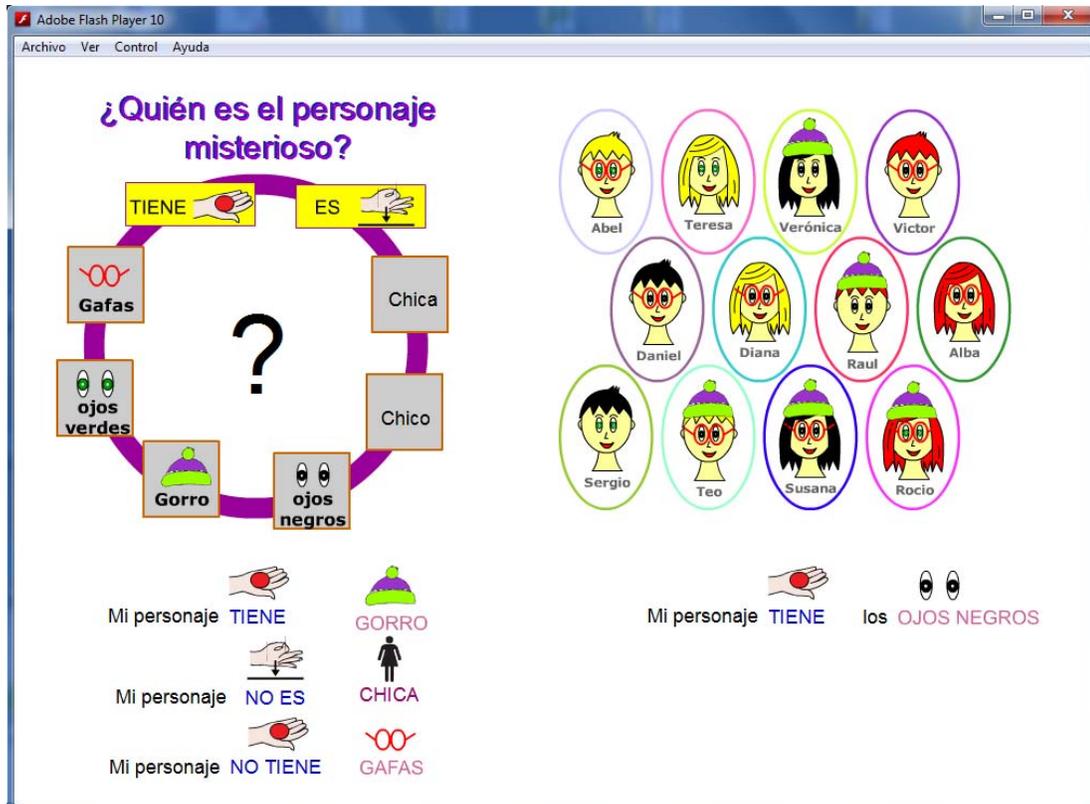


Fig. 42 Interfaz boceto

Respecto a la programación, también es bastante simple. No se han empleado clases, por lo que todo el código se encuentra en el documento Flash.

El código se puede clasificar en tres partes. En una primera parte, se cargan las fichas de los personajes empleando variables. Cada variable pertenece a un tipo de ficha a la que se le cambian las propiedades de posición para que se posicionen en la parte del stage deseada.

```

1
2 ////////////////////////////////////////////////// CARGAR FICHAS //////////////////////////////////////
3
4
5 var FiAbel:FichaAbel= new FichaAbel();           FiAbel.x = 550;           FiAbel.y = 100;           addChild(FiAbel);
6 var FiTeresa:FichaTeresa = new FichaTeresa();       FiTeresa.x = 650;         FiTeresa.y = 100;         addChild(FiTeresa);
7 var FiVeronica:FichaVeronica = new FichaVeronica(); FiVeronica.x = 750;      FiVeronica.y = 100;      addChild(FiVeronica);
8 var FiVictor:FichaVictor = new FichaVictor();     FiVictor.x = 850;        FiVictor.y = 100;        addChild(FiVictor);
9
10 var FiDaniel:FichaDaniel = new FichaDaniel();     FiDaniel.x = 500;        FiDaniel.y = 225;        addChild(FiDaniel);
11 var FiDiana:FichaDiana = new FichaDiana();        FiDiana.x = 600;         FiDiana.y = 225;         addChild(FiDiana);
12 var FiRaul:FichaRaul = new FichaRaul();           FiRaul.x = 700;          FiRaul.y = 225;          addChild(FiRaul);
13 var FiAlba:FichaAlba = new FichaAlba();           FiAlba.x = 800;          FiAlba.y = 225;          addChild(FiAlba);
14
15 var FiSergio:FichaSergio = new FichaSergio();     FiSergio.x = 550;        FiSergio.y = 350;        addChild(FiSergio);
16 var FiTeo:FichaTeo = new FichaTeo();              FiTeo.x = 650;           FiTeo.y = 350;           addChild(FiTeo);
17 var FiSusana:FichaSusana = new FichaSusana();     FiSusana.x = 750;        FiSusana.y = 350;        addChild(FiSusana);
18 var FiRocio:FichaRocio = new FichaRocio();        FiRocio.x = 850;         FiRocio.y = 350;         addChild(FiRocio);
19

```

Fig. 43 1º parte código del boceto

En la segunda parte del código se elige aleatoriamente el personaje misterioso y se cargan sus características. Cada personaje va asociado a un número por lo que primero se elige aleatoriamente un número comprendido entre el 0 y el 11 mediante operaciones matemáticas y el método random de Flash. Así, por ejemplo, si sale el número aleatorio 9 el personaje misterioso es Sergio.

```
22 ////////////////////////////////////////////////// ELEGIR PERSONAJE MISTERIOSO ////////////////////////////////////////
23 var numeroAleatorio:Number;
24 numeroAleatorio= Math.floor(Math.random() * 12);
```

Fig. 44 2º parte código del boceto 1

Una vez se tiene el número del personaje misterioso se cargan las propiedades del personaje. Esto se hace con variables booleanas y con una estructura de control switch. Así según el número que sea las distintas variables toman un valor true o false.

```
////////////////////////////////////// CARGAR PROPIEDADES ////////////////////////////////////////
var gorro:Boolean = new Boolean;
var gafas:Boolean = new Boolean;
var ojosVerdes:Boolean = new Boolean;
var ojosNegros:Boolean = new Boolean;
var chico:Boolean = new Boolean;
var chica:Boolean = new Boolean;

switch(numeroAleatorio){
  case 0: gorro = false; gafas = true; ojosVerdes = true; ojosNegros = false; chico = true; chica = false; break;
  case 1: gorro = false; gafas = true; ojosVerdes = false; ojosNegros = true; chico = false; chica = true; break;
  case 2: gorro = true; gafas = true; ojosVerdes = true; ojosNegros = false; chico = true; chica = false; break;
  case 3: gorro = true; gafas = false; ojosVerdes = true; ojosNegros = false; chico = false; chica = true; break;
  case 4: gorro = true; gafas = true; ojosVerdes = false; ojosNegros = true; chico = true; chica = false; break;
  case 5: gorro = true; gafas = true; ojosVerdes = true; ojosNegros = false; chico = false; chica = true; break;
  case 6: gorro = false; gafas = true; ojosVerdes = true; ojosNegros = false; chico = true; chica = false; break;
  case 7: gorro = true; gafas = true; ojosVerdes = false; ojosNegros = true; chico = false; chica = true; break;
  case 8: gorro = true; gafas = false; ojosVerdes = false; ojosNegros = true; chico = true; chica = false; break;
  case 9: gorro = true; gafas = false; ojosVerdes = true; ojosNegros = false; chico = false; chica = true; break;
  case 10: gorro = false; gafas = true; ojosVerdes = false; ojosNegros = true; chico = false; chica = true; break;
  case 11: gorro = true; gafas = false; ojosVerdes = false; ojosNegros = true; chico = true; chica = false; break;
}
```

Fig. 45 2º parte código del boceto 2

La tercera parte del código corresponde con la parte interactiva del programa. En ella se desarrollan las funciones que se ejecutan al pulsar cada uno de los botones. Por lo tanto, esta parte del código contiene eventos y funciones.

Estas funciones, al igual que el resto del código son muy sencillas. Al pulsar un botón se escribe su palabra correspondiente y se comprueba si la propiedad del

personaje para esa característica es true o false y, dependiendo de esto, se cambia la propiedad alpha de unas fichas u otras.

```
53 //////////////// ESCRIBIR FRASE Y ELIMINAR FICHAS //////////////////////
54
55 btn_gorro.addEventListener(MouseEvent.CLICK,funcion1);
56 btn_gafas.addEventListener(MouseEvent.CLICK,funcion2);
57 btn_ojosVerdes.addEventListener(MouseEvent.CLICK,funcion3);
58 btn_ojosNegros.addEventListener(MouseEvent.CLICK,funcion4);
59 btn_chico.addEventListener(MouseEvent.CLICK,funcion5);
60 btn_chica.addEventListener(MouseEvent.CLICK,funcion6);
61
62 btn_tiene.addEventListener(MouseEvent.CLICK,funTiene);
63 btn_es.addEventListener(MouseEvent.CLICK,funEs);

96 function funcion1(event:MouseEvent) {
97
98     var Go:go = new go();
99     Go.x = posicionX2 + horiz;
100    Go.y = posicionY + vertical;
101    addChild(Go);
102
103    switch(gorro){
104
105        case false:
106
107            var NoTiene:noTiene = new noTiene();
108            NoTiene.x = posicionX1 + horiz;
109            NoTiene.y = posicionY + vertical;
110            addChild(NoTiene);
111
112            FiDaniel.alpha = .15;
113            FiDiana.alpha = .15;
114            FiRaul.alpha = .15;
115            FiRocio.alpha = .15;
116            FiSusana.alpha = .15;
117            FiTeo.alpha = .15;
118            FiTeresa.alpha = .15;
119            FiVictor.alpha = .15;
120            break;
121
122        case true:
123
124            FiAbel.alpha = .15;
125            FiAlba.alpha = .15;
126            FiSergio.alpha = .15;
127            FiVeronica.alpha = .15;
128            break;
129    }
```

Fig. 46 detalle 3º parte código del boceto

#### 7.4. CONSULTA A LA EXPERTA

Una vez finalizado el boceto descrito anteriormente, se ha citado a la asesora externa Isabel Sánchez para que proporcione su opinión acerca de este e indique las posibles mejoras que se le pueden añadir.

Por un lado Isabel menciona que le gusta la temática elegida, así como el diseño y el uso de pictogramas con las palabras. Sin embargo, lo interesante de su intervención son los aspectos que se pueden mejorar de la aplicación.

Isabel indica que los botones que había que pulsar para escribir la frase deberían de estar ordenados. Es decir, cada uno de los botones con verbos debería de estar de alguna manera agrupado con los botones de los complementos con los que se puede juntar para formar correctamente la frase.

Otro aspecto que menciona es que sería interesante cambiar la forma de escribir las frases. Indica que sería mejor escribir primero la pregunta formulada y a continuación la respuesta “Sí” o “No” en vez de directamente la frase positiva o negativa para que quedara claramente señalada la pregunta formulada y no se entremezclara con la respuesta.

También, expone que ayudaría a trabajar la comprensión que existiera la opción de poder eliminar las fichas de los personajes manualmente en vez de que las eliminara automáticamente la máquina. Ya que, de esta nueva forma, para eliminar correctamente las fichas el jugador debe de entender tanto la pregunta como la respuesta dada.

Por último, sugiere que sería interesante tener un nivel de juego más sencillo en el que las preguntas ya estuviesen construidas. Este podría ser un primer nivel de contacto entre el jugador y juego para que al jugador le quedara clara la dinámica de juego.

## 7.5. APLICACIÓN “FINAL”

Teniendo en cuenta todo lo que ha mencionado Isabel, se procede a desarrollar la aplicación final. Para ello se va modificando poco a poco el boceto descrito.

Primero se procede al diseño de los nuevos botones. Para que posean aspecto de botones, se han dibujado dos botones sencillos en Photoshop, idénticos pero de diferente color, y estos se han importado a Flash. Una vez en Flash, se hacen copias y se le añade a cada uno un pictograma y su respectivo texto. Cada botón se convierte en Clip de película y por último se agrupan los botones como sugirió Isabel.

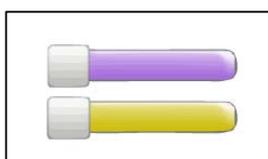


Fig. 47 Botones de Photoshop



Fig. 48 Botones frases sin hacer

A continuación se procede a mejorar el código escrito hasta el momento, para ello se crean dos clases. Una clase corresponde a la parte del código en la que se cargan las fichas en el escenario y la otra clase a la parte del código en el que se eligen al personaje misterioso y se cargan sus características en variables.

Ambas clases se crean siguiendo el segundo procedimiento descrito en el apartado 6.2.2.7, es decir, creando un tipo de archivo “Clase de ActionScript 3.0” y escribiendo en él el código oportuno. Estos documentos se guardan en una carpeta llamada “clases” destinada para guardar en ella todas las clases empleadas durante el proyecto y dicha carpeta se guarda dentro de la carpeta del proyecto. De esta manera se tienen todos los documentos ordenados.

Sin embargo, no hay que olvidar que al guardar las clases en un directorio diferente al que se encuentra el documento Flash, hay que cargar en este el directorio donde se encuentran las clases. Esto se hace con el método “import” como se muestra en la figura 49.

```
2 import clases.*;
```

**Fig. 49 Código de importar las clases**

También, como definición de paquete ambas clases deben de hacer referencia a la carpeta “clases” que es a la que pertenecen. De esta manera se le indica al compilador que dichos documentos pertenecen al grupo de clases al que pertenecen.

El código de la primera clase, sigue la estructura general de todas las clases: definición de paquete, que ya se ha comentado; declaración de clase pública, que hace que sea visible a referencias en cualquier lugar; y cuerpo, cuyo código es idéntico al código de la primera parte del boceto.

A esta clase también se le añade la Indicación de extensión de MovieClips. Que sea una extensión de MovieClip indica que esta clase posee todas las propiedades de los MovieClip más las añadidas en esta clase.

```

1 package clases
2 {
3
4     import flash.display.*;
5
6     import abel;
7     import teresa;
8     import veronica;
9     import victor;
10    import daniel;
11    import diana;
12    import raul;
13    import alba;
14    import sergio;
15    import teo;
16    import susana;
17    import rocio;
18
19
20    public class cargarFichas extends MovieClip
21    {
22
23        public var Abel:abel;
24        public var Teresa:teresa;
25        public var Veronica:veronica;
26        public var Victor:victor;
27        public var Daniel:daniel;
28        public var Diana:diana;
29        public var Raul:raul;
30        public var Alba:alba;
31        public var Sergio:sergio;
32        public var Teo:teo;
33        public var Susana:susana;
34        public var Rocio:rocio;
35
36
37        public function cargarFichas()
38        {
39
40            var pos1y:Number = 100;
41            var pos2y:Number = pos1y + 125;
42            var pos3y:Number = pos2y + 125;
43
44            var pos1x:Number = 550;
45            var pos2x:Number = pos1x + 100;
46            var pos3x:Number = pos2x + 100;
47            var pos4x:Number = pos3x + 100;
48
49
50            Abel = new abel();           Abel.x = pos1x;           Abel.y = pos1y;           addChild(Abel);
51            Teresa = new teresa();       Teresa.x = pos2x;       Teresa.y = pos1y;       addChild(Teresa);
52            Veronica = new veronica();   Veronica.x = pos3x;     Veronica.y = pos1y;     addChild(Veronica);
53            Victor = new victor();       Victor.x = pos4x;       Victor.y = pos1y;       addChild(Victor);
54            Daniel = new daniel();       Daniel.x = pos1x + 50;  Daniel.y = pos2y;       addChild(Daniel);
55            Diana = new diana();         Diana.x = pos2x + 50;   Diana.y = pos2y;       addChild(Diana);
56            Raul = new raul();           Raul.x = pos3x + 50;   Raul.y = pos2y;       addChild(Raul);
57            Alba = new alba();           Alba.x = pos4x + 50;   Alba.y = pos2y;       addChild(Alba);
58            Sergio = new sergio();       Sergio.x = pos1x;       Sergio.y = pos3y;       addChild(Sergio);
59            Teo = new teo();             Teo.x = pos2x;         Teo.y = pos3y;         addChild(Teo);
60            Susana = new susana();       Susana.x = pos3x;       Susana.y = pos3y;       addChild(Susana);
61            Rocio = new rocio();         Rocio.x = pos4x;       Rocio.y = pos3y;       addChild(Rocio);
62        }
63    }
64 }

```

Declaración de paquete

Declaración de clase pública

Cuerpo

Fig. 50 Clase “cargarFichas”

La clase perteneciente al código en el que se selecciona al personaje misterioso también posee las mismas partes que las indicadas en la anterior clase. Sin embargo, el código del cuerpo es el mismo que la segunda parte del boceto.

```

1 package clases
2 {
3
4     public class elegirPersonaje
5     {
6
7         public var gorro:Boolean = new Boolean ;
8         public var gafas:Boolean = new Boolean ;
9         public var rubio:Boolean = new Boolean ;
10        public var moreno:Boolean = new Boolean ;
11        public var pelirrojo:Boolean = new Boolean ;
12        public var chico:Boolean = new Boolean ;
13        public var chica:Boolean = new Boolean ;
14        public var numeroAleatorio:Number;
15
16
17        public function elegirPersonaje()
18        {
19            numeroAleatorio = Math.floor(Math.random() * 12);
20
21            switch (numeroAleatorio)
22            {
23
24
25                case 0 : gorro = false; gafas = true; rubio = true; moreno = false; pelirrojo = false; chico = true; chica = false; break;
26                case 1 : gorro = false; gafas = true; rubio = false; moreno = false; pelirrojo = true; chico = false; chica = true; break;
27                case 2 : gorro = false; gafas = true; rubio = false; moreno = true; pelirrojo = false; chico = true; chica = false; break;
28                case 3 : gorro = false; gafas = true; rubio = true; moreno = false; pelirrojo = false; chico = false; chica = true; break;
29                case 4 : gorro = true; gafas = false; rubio = false; moreno = false; pelirrojo = true; chico = true; chica = false; break;
30                case 5 : gorro = true; gafas = true; rubio = false; moreno = false; pelirrojo = true; chico = false; chica = true; break;
31                case 6 : gorro = false; gafas = false; rubio = false; moreno = true; pelirrojo = false; chico = true; chica = false; break;
32                case 7 : gorro = true; gafas = true; rubio = false; moreno = true; pelirrojo = false; chico = false; chica = true; break;
33                case 8 : gorro = true; gafas = true; rubio = true; moreno = false; pelirrojo = false; chico = true; chica = false; break;
34                case 9 : gorro = false; gafas = false; rubio = true; moreno = false; pelirrojo = false; chico = false; chica = true; break;
35                case 10 : gorro = true; gafas = false; rubio = false; moreno = true; pelirrojo = false; chico = false; chica = true; break;
36                case 11 : gorro = false; gafas = true; rubio = false; moreno = false; pelirrojo = true; chico = true; chica = false; break;
37
38            }
39        }
40    }

```

Declaración de paquete

Declaración de clase pública

Cuerpo

Fig. 51 Clase “ElegirPersonaje”

Creando estas clases el código Flash queda mucho más simple que el código del boceto pues en el documento Flash simplemente hay que crear una variable de cada tipo de clase como se puede observar en la figura 52.

```

2 import clases.*;
3
4 ////////////////////////////////////////////////// CARGAR FICHAS //////////////////////////////////////
5
6 var Fichas:cargarFichas = new cargarFichas();
7 addChild(Fichas);
8
9
10 //////////////ELEGIR PERSONAJE Y SUS PROPIEDADES //////////////////////////////////
11
12 var propiedades:elegirPersonaje = new elegirPersonaje ();
13

```

Fig. 52 Código simplificado

Una vez simplificadas las dos partes del boceto, se procede a eliminar el resto del código puesto que el funcionamiento del programa va a cambiar.

A continuación se decide unir las ideas que expuso Isabel acerca de tener un nivel inicial en el que las preguntas ya estuviesen hechas y de que el usuario pudiese eliminar personalmente las fichas. Para ello, se añade una pantalla al comienzo del juego en la que el profesional que acompaña al jugador pueda seleccionar el nivel o dificultad de juego.

Dicha pantalla posee dos apartados. En el primero se selecciona si se desea que la pregunta realizada por el jugador esté ya construida o por el contrario se tenga que construir. En el segundo apartado se elige si la eliminación de las fichas es manual o automática. De este modo, se consiguen cuatro niveles de juego. El más sencillo es el que tiene las preguntas construidas y las fichas se eliminan automáticamente mientras que el de mayor dificultad es en el que las preguntas están sin construir y las fichas se eliminan manualmente.

La implementación de dicha pantalla inicial se ha llevado a cabo en una diapositiva anterior a la diapositiva en la que se estaba trabajando y se han empleado componentes de los que vienen por defecto en Flash. Más concretamente componentes del tipo RadioButton. Estos se disponen en grupos en los que solo se puede seleccionar uno de los botones pues cuando el usuario selecciona un botón, cualquier botón seleccionado anteriormente en el mismo grupo se convierte en no seleccionado. El aspecto de la pantalla inicial es el que aparece en la figura 53.

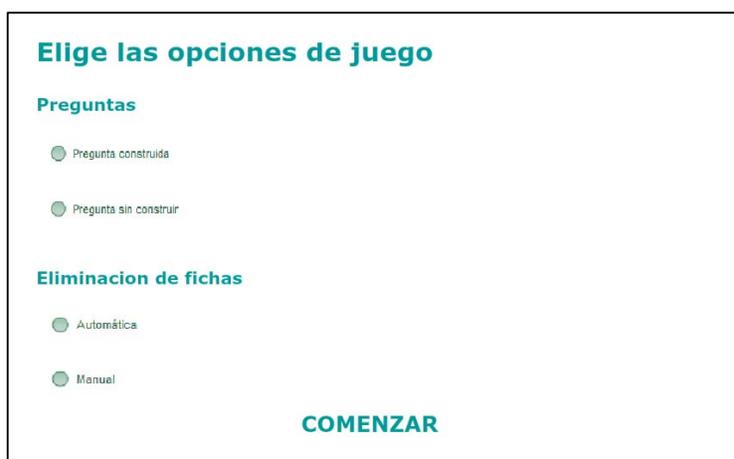


Fig. 53 Pantalla de inicio

El código de la pantalla inicial se ha escrito directamente en el panel de acciones de Flash debido a que es un código bastante sencillo y, al estar en otro fotograma, no se mezcla con el resto de código.

Como en cada grupo de botones solo hay dos opciones se ha decidido crear dos variables booleanas, una por cada grupo de botones, y en función de sus valores se conoce cuál de las opciones se ha elegido pues su valor se modifica al pulsar sobre un botón u otro del grupo.

```
1  stop();
2
3  import flash.display.*;
4  import flash.events.MouseEvent;
5
6
7  /// elegir opciones
8
9  var preg:Boolean = false;
10 var resp:Boolean = false;
11
12
13 opc1.addEventListener(MouseEvent.CLICK,pul1);
14 function pul1(event:MouseEvent) {
15     preg = true;
16 }
17
18 opc2.addEventListener(MouseEvent.CLICK,pul2);
19 function pul2(event:MouseEvent) {
20     preg = false;
21 }
22
23 opc3.addEventListener(MouseEvent.CLICK,pul3);
24 function pul3(event:MouseEvent) {
25     resp = true;
26 }
27
28 opc4.addEventListener(MouseEvent.CLICK,pul4);
29 function pul4(event:MouseEvent) {
30     resp = false;
31 }
32 }
33
34 /// boton comenzar
35
36 eee.addEventListener(MouseEvent.CLICK,comenzar);
37 function comenzar(event:MouseEvent) {
38     gotoAndStop(2);
39 }
```

Fig. 54 Código pantalla de inicio

Una vez se tienen claros los niveles de juego, se dispone a diseñar el resto de elementos gráficos necesarios en la aplicación. Los botones con las preguntas ya construidas se crean de la misma manera que los botones con solo palabras descritos anteriormente.



Fig. 55 Botones frases hechas

Las preguntas que se escribirán al pulsar sobre los botones y las palabras que se escriben como respuesta a las citadas preguntas se diseñan con la Herramienta texto de Flash y a continuación se convierten en MovieClips. Todas las palabras tendrán las letras grandes e irán acompañadas de pictogramas. Además las respuestas “Sí”, “No” y “Frase mal construida” estarán coloreadas de verde, rojo y azul respectivamente. También, para crear la sensación de que las palabras se están escribiendo, se ha añadido una pequeña animación a cada MovieClip.

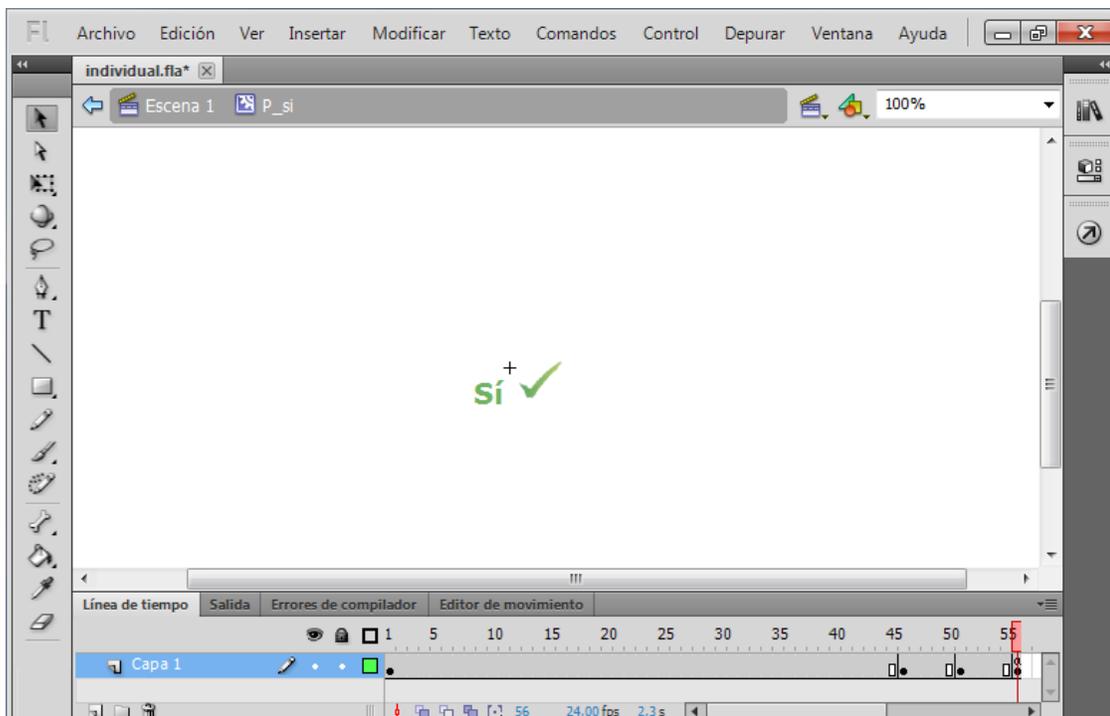


Fig. 56 Detalle animación de palabra

A continuación se procede a implementar el resto del código de la aplicación. Para ello se piensa en la creación de varias clases aparte de las ya construidas ya que es una manera muy cómoda y clara de trabajar.

Dependiendo de las opciones escogidas en la pantalla inicial, la interfaz del juego será de una manera u otra. Por ejemplo, si se pulsa sobre la opción “Pregunta construida” los botones no serán los mismos que si se elige la opción “Pregunta sin construir” por lo que es interesante tener la carga de los botones en dos clases diferentes y según se seleccione una opción u otra crear un tipo de variable de la clase oportuna.

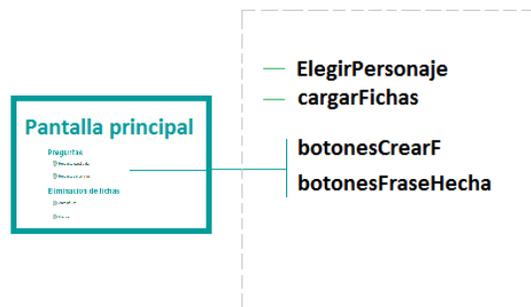


Fig. 57 Esquema de clases 1

Para escribir la pregunta que selecciona el usuario tienen que existir eventos de botón y estos tienen que llamar a una función que escriba dichas cuestiones. Los eventos de botón se encuentran dentro de las dos clases mencionadas anteriormente, sin embargo, para simplificar el código y no repetirlo en ambas clases, se ha procedido a que dentro de estas funciones se cree otra variable del tipo “escribir”, cuyas propiedades se definen en otra clase. Dentro de sus propiedades estarán las de escribir las cuestiones en el stage.

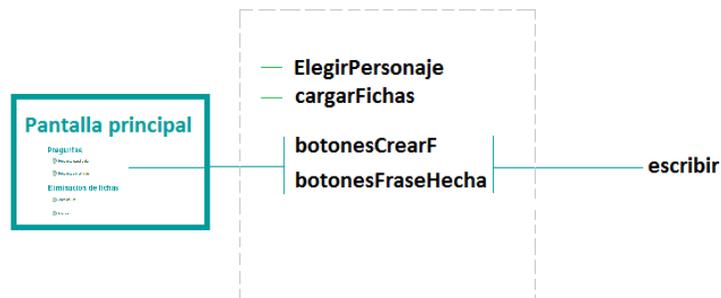


Fig. 58 Esquema de clases 2

También hay dos maneras de eliminar las fichas, por lo que se sigue el mismo criterio que con las preguntas. Se crean dos clases y según se haya seleccionado una opción u otra en la pantalla inicial se creará una variable de un tipo u otro.

Sin embargo, para la eliminación de fichas automática es necesario saber qué frase ha seleccionado el jugador por lo que la variable del tipo “eliminaciónAutomática” se definirá en las clases de los botones, como ocurre con la variable de tipo escribir.

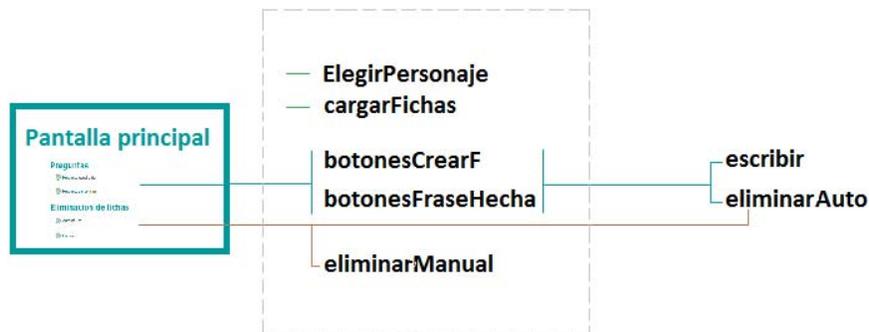


Fig. 59 Esquema de clases 3

Todas estas clases siguen la estructura general de las clases. Sin embargo, a continuación se procede a una breve descripción de todas ellas para que queden más claras sus funciones.

#### - Clase “botonesCrearF”

En esta clase se pueden distinguir dos partes. En la primera parte se cargan los botones de la pregunta sin construir empleando variables. Cada variable pertenece a un tipo de botón a la que se le cambian sus propiedades de posición para que se sitúe en la parte del stage deseada.

La segunda parte corresponde a los eventos y funciones de los botones que se definen en la parte anterior. En las funciones simplemente se cambia el valor a una variable llamada “pregunta”, se crea una variable de tipo “escribir” y por último se crea otra variable de tipo “eliminarAuto”.

La variable “pregunta” es una entrada para la clase “escribir” de esta forma se sabe qué botón ha sido pulsado y, por lo tanto, se conoce la palabra que se debe escribir en el stage. Las otras dos variables de describen más adelante.

- Clase “botonesFraseHecha”

Esta es muy similar a la clase anterior, solamente cambia los botones que se cargan pues en esta se cargan los botones con las preguntas ya construidas.

-Clase “eliminarAuto”

Las variables de tipo “emilinarAuto” se definen en las clases de los botones como se ha mencionado en la descripción de dichas clases por lo que tiene como entrada la variable “pregunta”. Según el valor que tenga esta variable, se cambia la propiedad alpha de unas fichas u otras para ello se ha empleado una estructura de control switch.

Hay que mencionar que las variables “eliminarAuto” poseen estas características de cambio de alpha únicamente si se ha seleccionado en la pantalla de inicio que la eliminación de fichas sea automática. Para que esto se cumpliera, se ha puesto una estructura de control if antes de definir las propiedades, Así pues, si no se ha seleccionado en la pantalla de inicio esta opción, la variable no poseerá ninguna propiedad, será una variable vacía.

- Clase “eliminarManual”

Las variables de este tipo solo se definen si en la pantalla inicial se selecciona la opción de eliminación de fichas manual. Tiene como entrada la variable “Fichas” que es la que carga todas las fichas en el stage y se ha descrito anteriormente.

Cuando se crea una variable de este tipo, se dota de interactividad a la fichas de los personajes pues se crean eventos de botón para estos MovieClips. De esta manera estos funcionan como si de botones se tratase.

Las funciones a las que llaman estos eventos son dos. Si es la primera vez que se hace clic sobre una ficha, se llama a una función que coloca sobre la ficha un óvalo blanco con alpha 0.80 modificando así el alpha de la ficha. Sin embargo, si se selecciona por segunda vez la ficha, se llama a otra función que elimina el óvalo añadido anteriormente.

#### - Clase “escribir”

El tipo de variables “escribir” se definen, como se ha comentado anteriormente, en las clases de los botones y tienen como entrada la variable “pregunta”. Esta tiene un valor u otro según los botones que se han seleccionado. Por lo tanto, esta clase consta de una estructura de control switch que, según el valor de la variable “pregunta”, escribe sobre el escenario la pregunta correspondiente. También controla que en los casos en los que las preguntas se escriben palabra por palabra las cuestiones estén bien formadas, si no es así escribe como respuesta “Frase mal construida”.

En el caso de que las cuestiones estén bien construidas, esta clase también escribe la respuesta. Para ello, comprueba el valor de la propiedad seleccionada en las propiedades del personaje misterioso. Si dicha propiedad es “true” la respuesta es “Si” mientras que si la propiedad es “false” la respuesta es “No”.

Una vez organizado y programado el funcionamiento general del juego, se piensa cómo se va a organizar el final del juego. Es decir, cuando el jugador sepa quién es el personaje misterioso una vez realizadas las oportunas preguntas.

Se ha decidido añadir al escenario un nuevo botón de resolver, diseñado siguiendo el mismo procedimiento que el resto. Cuando el jugador sepa quién es el personaje misterioso deberá pulsar sobre este botón y a continuación sobre la ficha del personaje que crea que es el personaje misterioso.

Al hacer clic sobre el botón resolver se ejecuta el método gotoAndStop() que hace que la película Flash salte y se pare en el siguiente fotograma. Se ha decidido que haga esto para separar el código explicado hasta el momento del código de resolución del juego.

Una vez en el siguiente fotograma se comprueba si el personaje seleccionado es el personaje misterioso o no. Para hacer esto, se recurre a otra clase; la clase “solución”. En esta clase, al igual que en la clase “EliminacionManual”, se crean eventos de botón para las Fichas dotándolas así de interactividad y proporcionándoles un funcionamiento similar al de los botones.

Cada evento de botón lleva asociado su función a pesar de que todas ellas tienen una estructura idéntica. Estas funciones inicialmente añaden en el stage la siguiente pregunta: “¿El personaje se llama \_\_\_\_\_?” Seguido del nombre del personaje seleccionado.



Fig. 60 Pregunta final

A continuación comprueban si el número aleatorio de la clase “ElegirPersonaje” coincide con el número del personaje que ha seleccionado el jugador. Si ambos números son iguales el jugador ha ganado la partida por lo que en el escenario aparecerá un MovieClip en el que se indica que el jugador ha resultado ganador. Por el contrario, si ambos números no coinciden el jugador ha perdido la partida por lo que en este caso en el escenario aparecerá un MovieClip en el que se indica la derrota del jugador. Ambos MovieClips mencionados se han diseñado previamente con las herramientas de texto de Flash.



Fig. 61 Respuestas finales

La variable de tipo “solución” también carga en el escenario un rectángulo transparente con alpha cero. La finalidad de este rectángulo invisible es que no se pueda continuar jugando pues se coloca encima de todos los botones que se hallaban en el escenario.

Cuando se hace clic sobre el rectángulo transparente se carga en el escenario el botón “volver a jugar” que permite, como su nombre indica, volver a jugar. Es decir, elimina todas las variables definidas y, mediante el método “gotoAndStop()”, vuelve al fotograma de la página principal.



**Fig. 62 Pantalla final**

Para añadir un poco de dificultad al juego, se han añadido vidas. Es decir, al iniciar el juego el jugador posee siete vidas y conforme se va preguntando las vidas se van eliminando. Por cada pregunta realizada, bien o mal, se pierde una vida. Así cuando el jugador se quede sin vidas, este habrá perdido la partida.

Esta pequeña dificultad se implementa muy fácilmente en el código explicado. Basta con crear un MovieClip con una imagen de una vida, en este caso de un corazón, y cargar tantos MovieClips como vidas se tienen al comienzo del juego en las clases de los botones. Así a cada función llamada por los eventos de botón se le añade código para que elimine un MovieClip cada vez que se ejecute.

También en cada función se añade código para que cuando ya no quede ninguna vida se termine el juego mostrando en el escenario un MovieClip con un mensaje de que las vidas se han acabado.

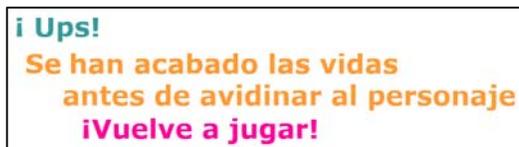


Fig. 63 Mensaje de no vidas

Lo único que falta para completar el juego es un botón que permita ir hacia atrás, hacia la pantalla de inicio, en cualquier momento. Por lo que en el segundo fotograma del código de Flash se ha definido un botón como el de la figura 63 situado en la esquina superior izquierda del escenario. Al pulsar sobre dicho botón ocurre lo mismo que al pulsar sobre el botón explicado anteriormente de “volver a jugar”. Se eliminan todas las variables definidas en el código anterior y mediante el método “gotoAndStop()” vuelve al fotograma de la página principal. De esta manera, el juego individual queda concluido.



Fig. 64 Flecha volver a inicio

Una vez finalizado el modo de juego individual, se construye el juego para jugar en parejas. Se ha duplicado el juego individual y a partir de la copia, se procede a desarrollar el juego colaborativo.

La primera modificación del juego se encuentra en la pantalla de inicio. Se ha decidido eliminar la opción en la que las fichas se eliminan automáticamente ya que los jugadores que jueguen en el modo en parejas antes han debido de jugar en modo individual y tienen un nivel de conocimiento adecuado para comprender las respuestas y eliminar las fichas correctamente.

Por esta razón, se han eliminado de la pantalla de inicio los elementos de selección del tipo “eliminación de fichas” y su respectivo código. También se ha eliminado la clase “EliminarAuto” y las variables de este tipo que se definían en las clases de los botones.

La siguiente modificación está relacionada con la elección del personaje misterioso. Ahora es el jugador 1 el que lo elige y no la máquina. Para ello, se añaden un par de fotogramas continuados de la pantalla inicial en los que se le da a elegir al jugador 1 entre todos los personajes para que elija al personaje misterioso.

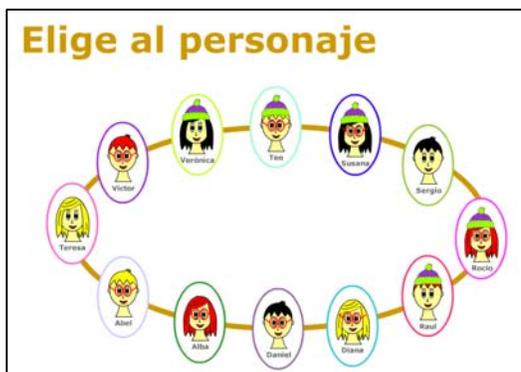


Fig. 66 Pantalla selección de personaje



Fig. 65 Pantalla comprobación de selección personaje

Para que quede clara la secuencia d juego, es decir qué jugador debe de jugar en cada momento, también se han añadido pantallas y bocadillos en los que se indica tanto a qué jugador le toca jugar como qué es lo que debe de hacer.



Fig. 67 Pantallas y bocadillos de ayuda

Sin embargo, la mayor modificación en cuanto al código se ha llevado a cabo en la clase “escribir”. En esta clase, ya no se escribe la respuesta a la pregunta sino que se carga en el escenario dos botones para que sea el jugador 1 el que responda a las preguntas.



Fig. 68 Botones de responder

En dicha clase también se encuentran los eventos de dichos botones, por lo que se comprueba cuando se pulsa un botón si la respuesta dada por el jugador 1 coincide verdaderamente con la respuesta correcta, que es la que se daba en el modo individual. Si ambas respuestas coinciden se escribe sobre el escenario la respuesta “Sí” o “No” correspondiente mientras que si ambas respuestas son diferentes se escribe un mensaje de respuesta incorrecta.

El resto del código e interfaz sigue exactamente igual que en el modo individual.

## 7.6. ADECUACION A VARIOS FORMATOS

Una vez implementadas las dos aplicaciones explicadas anteriormente, se procede a su adecuación para que funcionen correctamente en las plataformas mencionadas en el apartado 7.2, PC y tableta.

Como se ha mencionado en el citado apartado, para que las aplicaciones corran en el PC no es necesario efectuar ningún cambio ya que es el documento .swf el que se ejecuta en el ordenador. Sin embargo, para adaptar las aplicaciones para la tableta sí es necesario efectuar cambios.

Inicialmente se crea un documento .apk, paquete para el sistema operativo android. Su creación se consigue siguiendo los pasos explicados en el anexo I “Exportar archivo .apk de adobe Flash Professional CS5.5”.

Este archivo .apk es el que se instala en la tableta. Su instalación se explica en el anexo II “Importar archivo apk a un dispositivo android”.

Al seguir estos pasos las aplicaciones se instalaron correctamente en ambas plataformas por lo que los juegos están listos para ser usados.

## 7.7. PRUEBAS DE USUARIO

El lunes 25 de Junio se acudió a ANFAS, Asociación Navarra en favor de las personas con discapacidad intelectual, para hacer pruebas con usuarios finales.

Aunque los usuarios no correspondían exactamente con el perfil de los usuarios finales de la aplicación puesto que la mayoría no sabían leer o les resultaba muy complicada la comprensión lectora, su experiencia con la aplicación sirvió para dar una idea sobre lo que se debía de mejorar.

Además en dichas pruebas se contaba con la ayuda de varias expertas que también dieron su opinión sobre el juego tanto oralmente como mediante una encuesta que se les proporcionó al finalizar el encuentro.

El soporte en el que les presentó la aplicación fue la tableta ya que era el soporte más cómodo de transportar y debido al nivel de los usuarios, solo experimentaron con el juego individual. Por lo tanto, no emplearon todas las modalidades de juego aunque, como se ha mencionado anteriormente, a pesar de esto la experiencia fue suficiente como para observar los aspectos del juego que se debían perfeccionar.

Los usuarios que probaron la aplicación fueron cuatro. El primer usuario apenas sabía leer y solo pulsaba sobre los botones que le indicaba la experta puesto que no comprendía la dinámica del juego.

Por el contrario, el segundo usuario fue capaz de jugar solo correctamente. No obstante, este sufre un pequeño problema en la vista que le dificultó leer con facilidad las palabras.

El tercer usuario sí comprendió la dinámica del juego pero tenía dificultad en saber qué es lo que tenía que preguntar y qué fichas debía eliminar. Debido a esto necesitó una gran ayuda de la experta para comprender qué características poseían los personajes y sobre cuales tenía que preguntar. Así mismo, la experta también le tuvo que ayudar a la hora de la eliminación de las fichas.

El último usuario, al igual que el segundo, jugó solo correctamente y no presentó ninguna dificultad.

A continuación se muestran las encuestas completadas por las expertas de ANFAS.

# CUESTIONARIO

## MODO INDIVIDUAL

- |  |                                     |                                     |
|--|-------------------------------------|-------------------------------------|
| ¿Se entienden las opciones de configuración iniciales? (Para profesionales)                      | <input checked="" type="radio"/> Si | <input type="radio"/> No            |
| ¿Cambiaría estas opciones de juego? ¿Añadiría o eliminaría alguna?<br>En caso afirmativo, ¿Cuál? | <input type="radio"/> Si            | <input checked="" type="radio"/> No |
| Nada más comenzar el juego, ¿Se intuye qué es lo que hay que hacer?                              | <input checked="" type="radio"/> Si | <input type="radio"/> No            |
| ¿Queda claro cómo hay que formular las frases?   | <input checked="" type="radio"/> Si | <input type="radio"/> No            |
| Cuando está en modo eliminación de fichas manual, ¿Se entiende cómo hay que eliminar las fichas? | <input checked="" type="radio"/> Si | <input type="radio"/> No            |
| ¿Es intuitivo cómo resolver?   | <input checked="" type="radio"/> Si | <input type="radio"/> No            |

## MODO PAREJAS

- |  |                          |                          |
|--|--------------------------|--------------------------|
| ¿Se entienden las opciones de configuración iniciales? (Para profesionales)                      | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Ve interesante añadir la opción de juego "eliminación de fichas automática"?                    | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Cambiaría estas opciones de juego? ¿Añadiría o eliminaría alguna?<br>En caso afirmativo, ¿Cuál? | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Es fácil saber cuándo le toca actuar a cada jugador?  | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Queda claro cómo hay que formular las frases?   | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Queda claro que el jugador 1 tiene que responder a la frase formulada por el jugador 2?         | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Se deduce cómo hay que eliminar las fichas?   | <input type="radio"/> Si | <input type="radio"/> No |
| ¿Es intuitivo cómo resolver?   | <input type="radio"/> Si | <input type="radio"/> No |

# GENERAL

## Aspecto:

Puntúe el aspecto del juego:

No me gusta nada										Me gusta mucho
1	2	3	4	5	6	7	8	9	10	

¿Cambiaría algo del aspecto del juego?

Si  No

En caso afirmativo ¿El qué?

- Tamaño letra : más grande
- Iconos más grandes
- Que tenga sonido

## Plataforma:

¿Las plataformas seleccionadas (PC y tableta) son adecuadas?

Si  No

¿Qué plataforma prefiere para jugar?

PC  Tableta

¿Sería interesante proporcionar el juego en otra plataforma diferente?

Si  No

En caso afirmativo, ¿Cuál?

## Valoración:

¿Le ha gustado el juego?

No me gusta nada										Me gusta mucho
1	2	3	4	5	6	7	8	9	10	

¿En qué medida cree que este juego es útil para el aprendizaje?

Nada útil										Muy útil
1	2	3	4	5	6	7	8	9	10	

- No elimine preguntas cuando galle  
o que se puedan ver

# CUESTIONARIO

## MODO INDIVIDUAL

¿Se entienden las opciones de configuración iniciales? (Para profesionales)

Si  No

¿Cambiaría estas opciones de juego? ¿Añadiría o eliminaría alguna?  
En caso afirmativo, ¿Cuál?

Si  No

Nada más comenzar el juego, ¿Se intuye qué es lo que hay que hacer?

Si  No

¿Queda claro cómo hay que formular las frases?

Si  No

Cuando está en modo eliminación de fichas manual, ¿Se entiende cómo hay que eliminar las fichas?

Si  No

¿Es intuitivo cómo resolver?

Si  No

## MODO PAREJAS

¿Se entienden las opciones de configuración iniciales? (Para profesionales)

Si  No

¿Ve interesante añadir la opción de juego "eliminación de fichas automática"?

Si  No

¿Cambiaría estas opciones de juego? ¿Añadiría o eliminaría alguna?  
En caso afirmativo, ¿Cuál?

Si  No

¿Es fácil saber cuándo le toca actuar a cada jugador?

Si  No

¿Queda claro cómo hay que formular las frases?

Si  No

¿Queda claro que el jugador 1 tiene que responder a la frase formulada por el jugador 2?

Si  No

¿Se deduce cómo hay que eliminar las fichas?

Si  No

¿Es intuitivo cómo resolver?

Si  No

# GENERAL

## Aspecto:

Puntúe el aspecto del juego:

No me gusta nada										Me gusta mucho
1	2	3	4	5	6	7	8	9	10	

¿Cambiaría algo del aspecto del juego?  Si  No

En caso afirmativo ¿El qué?

- El tamaño de la letra: más grande
- El tamaño de los iconos: más grandes
- Que tenga sonido
- Que tenga refuerzo positivo
- Que lo diga.

## Plataforma:

¿Las plataformas seleccionadas (PC y tableta) son adecuadas?  Si  No

¿Qué plataforma prefiere para jugar?  PC  Tableta

¿Sería interesante proporcionar el juego en otra plataforma diferente?  Si  No

En caso afirmativo, ¿Cuál?

## Valoración:

¿Le ha gustado el juego?

No me gusta nada										Me gusta mucho
1	2	3	4	5	6	7	8	9	10	

¿En qué medida cree que este juego es útil para el aprendizaje?

Nada útil										Muy útil
1	2	3	4	5	6	7	8	9	10	



Tanto de lo observado durante la experiencia como de lo mencionado por las expertas en las encuestas se pueden nombrar unos cuantos aspectos que se pueden mejorar de la aplicación.

Por un lado, se debería de aumentar el tamaño de las palabras y los iconos pues puede que haya jugadores, como el usuario 2, que tengan problemas de la vista.

También sería interesante aumentar el tamaño de los botones ya que, aunque durante esta experiencia no se ha dado el caso, puede que haya jugadores con problemas de movilidad de las extremidades superiores que tengan menor precisión a la hora de seleccionar los botones.

Otro aspecto que podría ayudar al jugador sería añadir sonido al juego. Este sonido debería de reforzar positivamente al jugador cuando este acertara o hiciera las cosas bien.

También ayudaría a la comprensión del jugador que cuando este diera como solución final a un personaje erróneo posteriormente se pudiesen revisar de nuevo las preguntas y respuestas y así observar en qué se ha confundido.

Por último, otro detalle que se observó en la prueba con usuarios finales pero no mencionaron las expertas fue que a la hora de presentar el juego a los usuarios, las expertas contaban una pequeña historia para que estos captasen mejor el objetivo del juego. Por lo tanto, sería apropiado crear una breve introducción al juego en el que se contara una pequeña historia que dejara claro el objetivo del juego.

## 7.8. MEJORA DE LA APLICACIÓN DEFINITIVA.

La primera mejora llevada a cabo consiste en añadir una pequeña animación al inicio del juego en la que se cuenta una pequeña historia. Dicha historia tiene como finalidad ayudar a los usuarios a entender el objetivo del juego.

La historia es la siguiente:



Fig. 69 Animación inicial

Para que el resto del juego quede en concordancia con la historia narrada, se procede a realizar algunos cambios en el aspecto del juego. Por ejemplo la frase que aparecía como título: “¿Quién es el personaje misterioso?” Se ha sustituido por la cuestión: “¿Quién tiene la moneda?”. Del mismo modo en se ha modificado la cuestión final del juego y su respuesta como se puede observar en la figura 70.



Fig. 70 Preguntas y respuestas finales

Por un lado se han hecho más grandes los botones, palabras e iconos. Sin embargo, para que el aspecto del juego quede ajustado, se han modificado las posiciones de algunos elementos. Finalmente el aspecto de la interfaz queda como se muestra en las figuras 71,72 73 y 74.

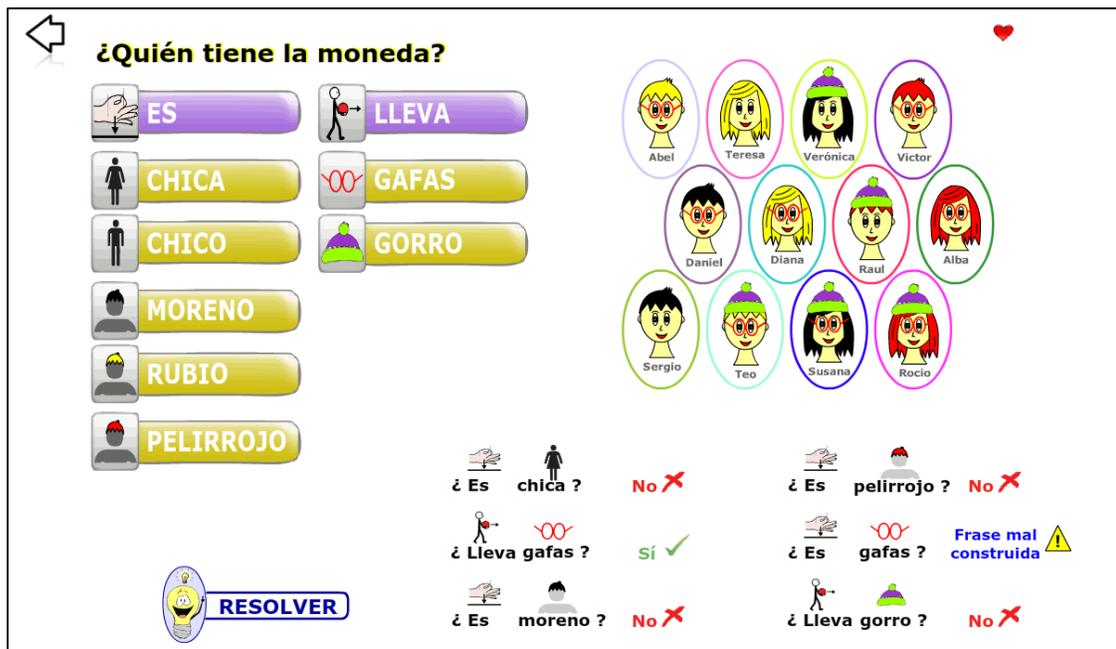


Fig. 71 Interfaz final Modo individual Preguntas sin construir

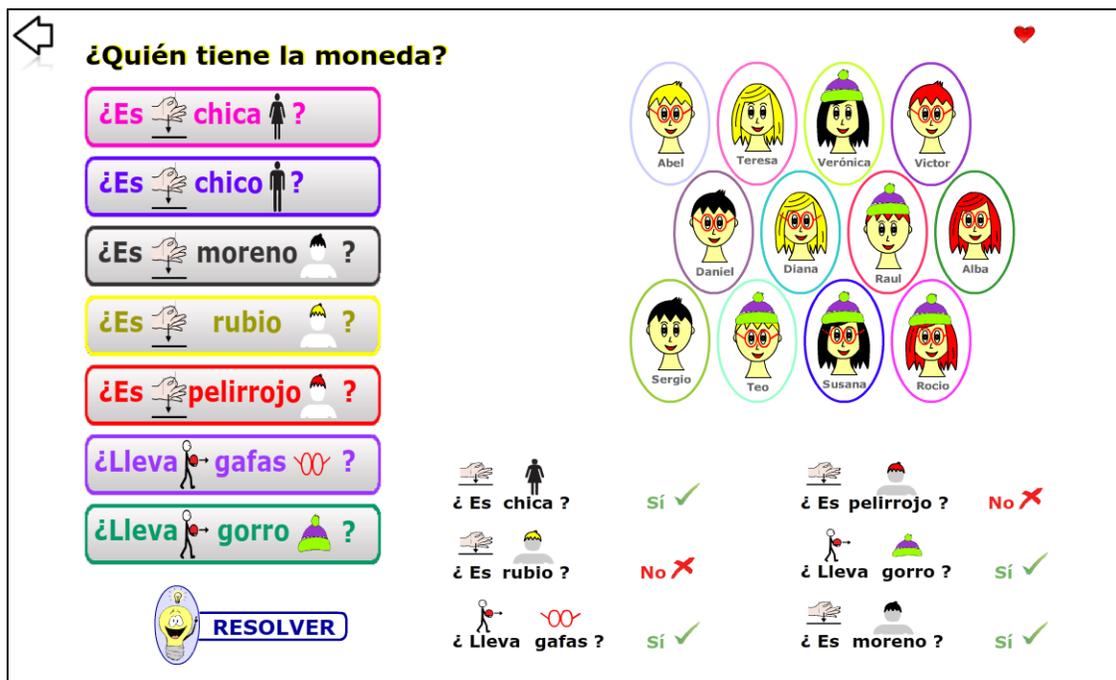


Fig. 72 Interfaz final Modo individual Preguntas construidas

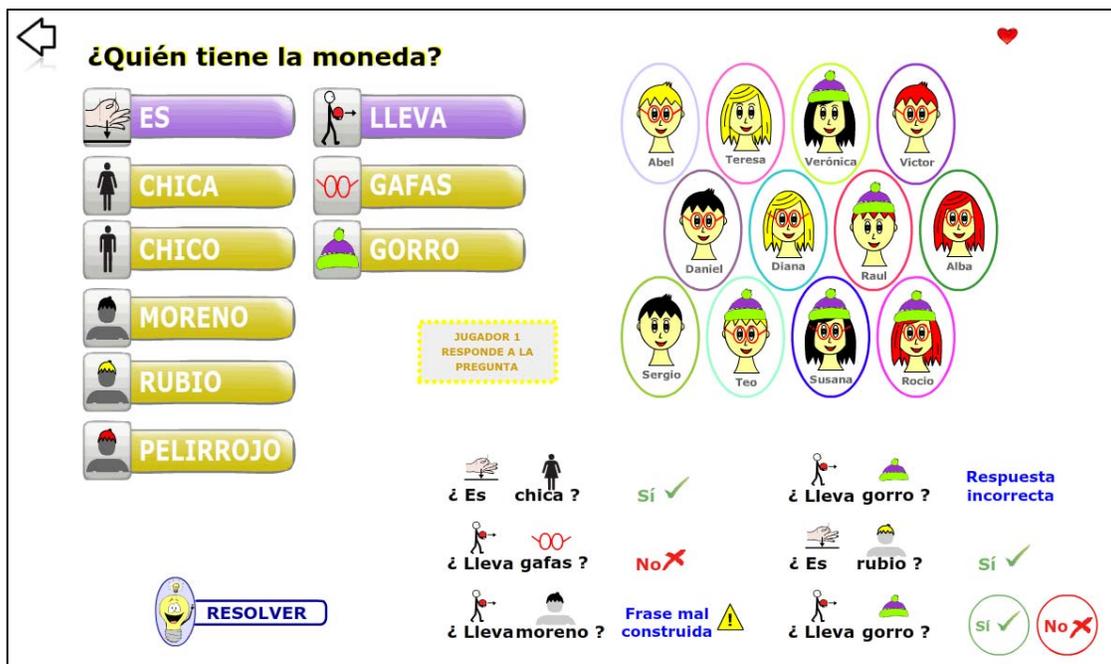


Fig. 71 Interfaz final Modo pareja Preguntas sin construir

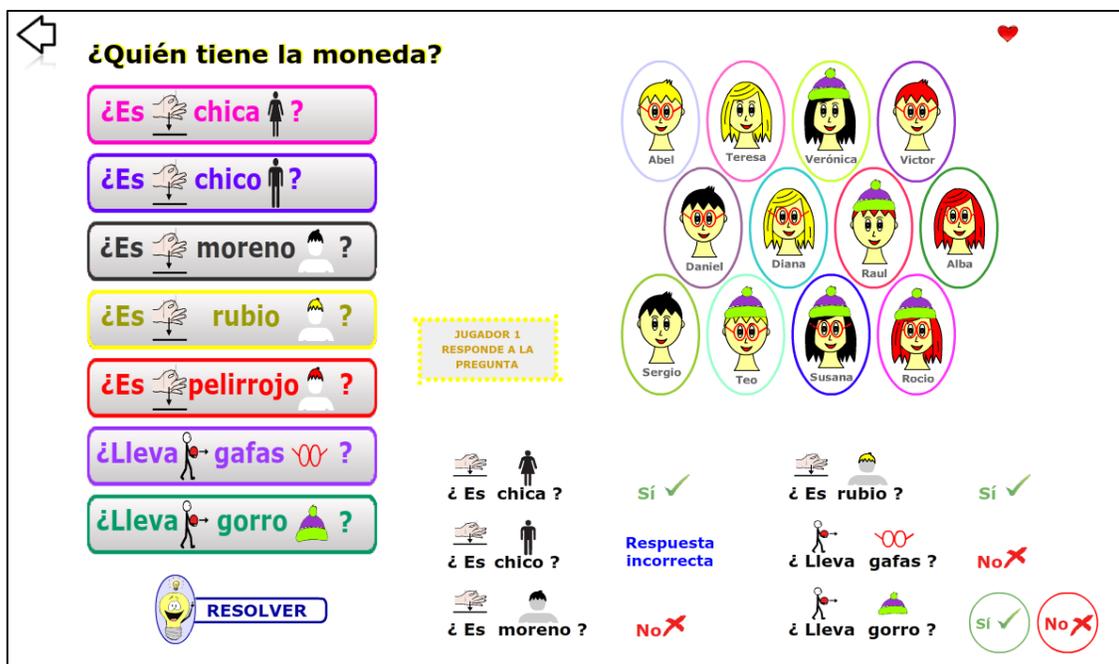


Fig. 72 Interfaz final Modo pareja Preguntas construidas

El último aspecto modificado ha sido que al terminar el juego se permite la opción de volver a ver las preguntas y respuestas de la jugada. Para efectuar dicho cambio, en la clase “solución” a la vez que se carga el botón “volver a jugar” se carga un nuevo botón. Este nuevo botón permite ver las preguntas realizadas y sus respectivas respuestas, esto es, elimina los MovieClips que aparecen en el escenario al crear la variable solución.

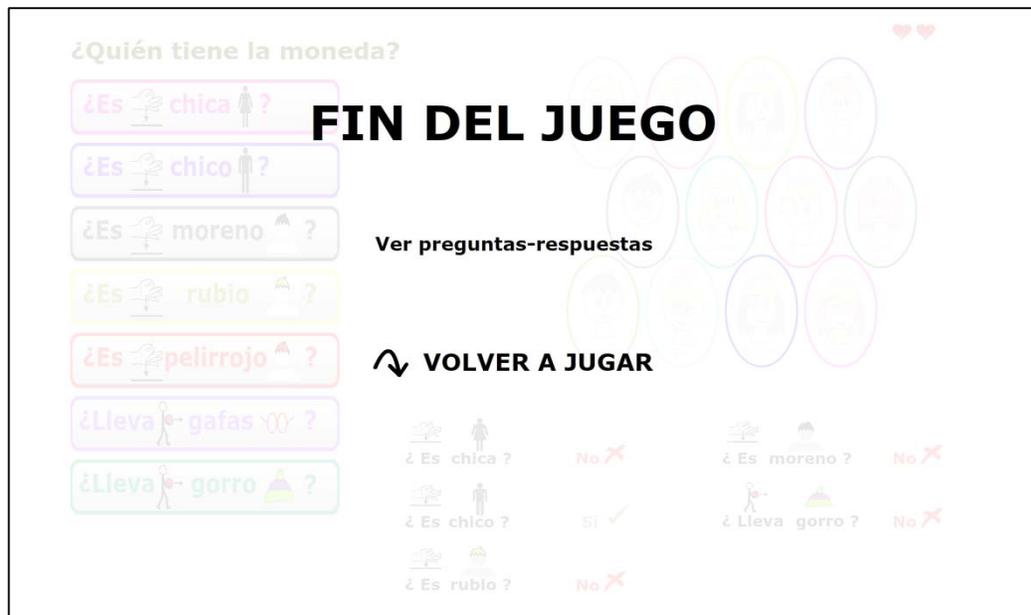


Fig. 73 Pantalla final definitiva

## 8. CONCLUSIONES Y LÍNEAS FUTURAS

Como conclusión general se puede señalar que se han cumplido con todos los objetivos especificados al inicio del proyecto.

Por un lado se han reforzado y aumentado los conocimientos sobre el software de desarrollo Adobe Flash Professional. A pesar de que se había trabajado en este entorno de trabajo en alguna asignatura de la carrera, en ninguna se había llegado a profundizar tanto ni se había llegado más allá de la creación de una simple animación.

Este proyecto también ha ayudado a la adquisición de conocimientos sobre un nuevo lenguaje de programación (ActionScript 3.0). Las mayores dificultades que se han encontrado en este proyecto, sin duda, han estado relacionadas con el aprendizaje de dicho lenguaje de programación pues nunca antes se había trabajado con un lenguaje de programación orientado a objetos.

Sin embargo, al finalizar el proyecto se ha conseguido un gran manejo de dicho lenguaje de programación y se ha adquirido seguridad como para introducirse en otro lenguaje de programación orientado a objetos.

Por otro lado, en este proyecto se ha conseguido diseñar e implementar una aplicación multimedia interactiva que ayuda a la enseñanza, individual y cooperativa, de personas con trastorno general de la comunicación.

El proyecto ha sido instalado con éxito en dos soportes informáticos, PC y tableta. Aunque la instalación en el PC ha resultado sencilla, la instalación en la tableta ha necesitado del estudio de ciertos aspectos del sistema operativo android. Dichos conocimientos adquiridos son interesantes y prácticos debido a la gran repercusión que dicho sistema operativo está teniendo sobre los teléfonos inteligentes y las tabletas hoy en día.

Quizás el objetivo más interesante de cumplir era el de evaluar el software desarrollado con personas de características similares a los destinatarios del producto ya que donde verdaderamente se observa si las expectativas del juego se cumplen es en un uso real de la aplicación.

Dicha evaluación se ha llevado a cabo exitosamente pues, aunque los usuarios no eran exactamente los usuarios destinatarios de la aplicación, ha dado una idea de cómo los usuarios finales pueden usar el software y se ha aprovechado dicha experiencia para mejorar los problemas que les podían surgir a los usuarios a la hora de emplear la aplicación.

Respecto a las líneas futuras de este proyecto se sugieren varias ideas. La más interesante es la relacionada con la implementación de esta aplicación en una mesa multitouch. Al comienzo del proyecto se pensó en implementar la aplicación en este soporte en vez de en la tableta. Sin embargo, la forma de unir el entorno de trabajo Flash con el protocolo TUIO y reactivision era una tarea dura que haría de este proyecto un proyecto mucho más complejo.

Otra idea que va a llevar a cabo un compañero es desarrollar la misma aplicación pero empleando otro lenguaje de programación más adecuado para android como es Eclipse. También este compañero modificará la estructura del programa para crear una especie de “plantilla” que se pueda modificar y que permita cambiar las fichas de personajes por fichas de otros tipos de familias como por ejemplo animales u oficios.

## 9. BIBLIOGRAFÍA

- <http://es.wikipedia.org/wiki/Wikipedia>
- <http://www.slideshare.net/ladbrito/02-principios-del-aprendizaje-multimedia>
- <http://sauce.pntic.mec.es/falcon/aprencooper.pdf>
- <http://www.hanseduca.com/categorias/area-de-comunicacion/6>
- <http://childtopia.com/index.php?module=home&func=juegos&idphpx=juegos-educativos-divertidos>
- <http://ictgames.com/literacy.html>
- <http://www.minijuegos.com/Who-Are-You/10229>
- [http://www.directoriow.com/pe\\_Quien\\_es\\_Quien\\_Espanol\\_Bluetooth\\_24\\_53240.html](http://www.directoriow.com/pe_Quien_es_Quien_Espanol_Bluetooth_24_53240.html)
- <http://aplicacionipad.com/juegos-ipad/juego-ipad-%C2%BFquien-es-quien/>
- <http://www.boe.es/boe/dias/2003/12/03/pdfs/A43187-43195.pdf>
- [http://help.adobe.com/es\\_ES/flash/cs/using/flash\\_cs5\\_help.pdf](http://help.adobe.com/es_ES/flash/cs/using/flash_cs5_help.pdf)
- [http://livedocs.adobe.com/flash/9.0\\_es/main/flash\\_as3\\_programming.pdf](http://livedocs.adobe.com/flash/9.0_es/main/flash_as3_programming.pdf)

# ANEXO I: EXPORTAR ARCHIVO .APK DE ADOBE FLASH PROFESSIONAL CS5.5

- 1º Crear un documento “AIR for android” donde se encontrará el código ActionScript 3.0 de la aplicación.

Para ello al abrir el programa Adobe Flash Professional seleccionar la opción “crear nuevo”-> “AIR for Android” como se muestra en la siguiente imagen.

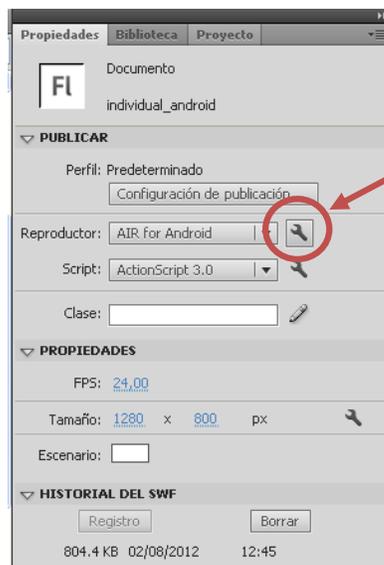


Fig. 74 Pantalla inicial de Flash CS5.5

- 2º Escribir el código de la aplicación con sus clases y los mc necesarios para su desarrollo.

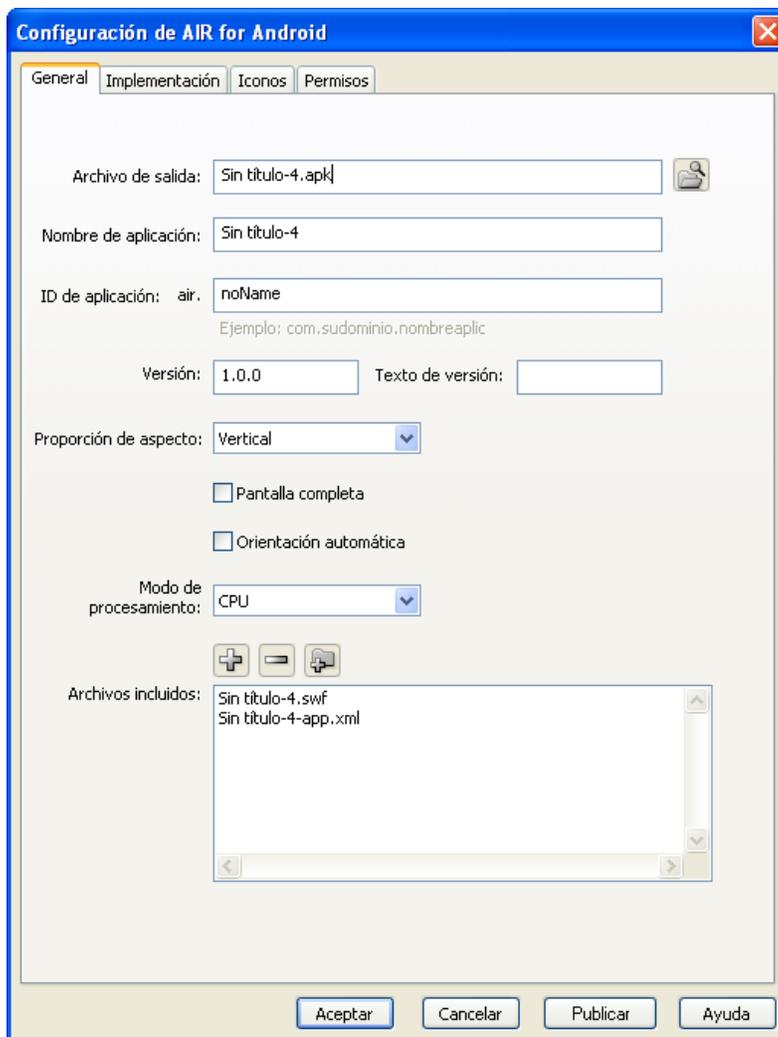
Una vez acabado de escribir el código, se procede a extraer el archivo .apk. Para ello se llevan a cabo los pasos explicados posteriormente.

- 3º En la opción “Propiedades”, sección “Publicar”, pulsar sobre la llave que hay en la opción “Reproductor”.



**Fig. 75 Pantalla propiedades**

Y se abre la siguiente ventana.



**Fig. 76 Configuración Air Android**

En la sección “GENERAL”:

- Rellenar el campo “Nombre de la aplicación”.
- Seleccionar en el campo “Proporción de aspectos” la opción “Auto”.
- Seleccionar la opción “Pantalla completa” y “Orientación automática”.
- Seleccionar en el campo “Modo de procesamiento” el modo “Auto”.

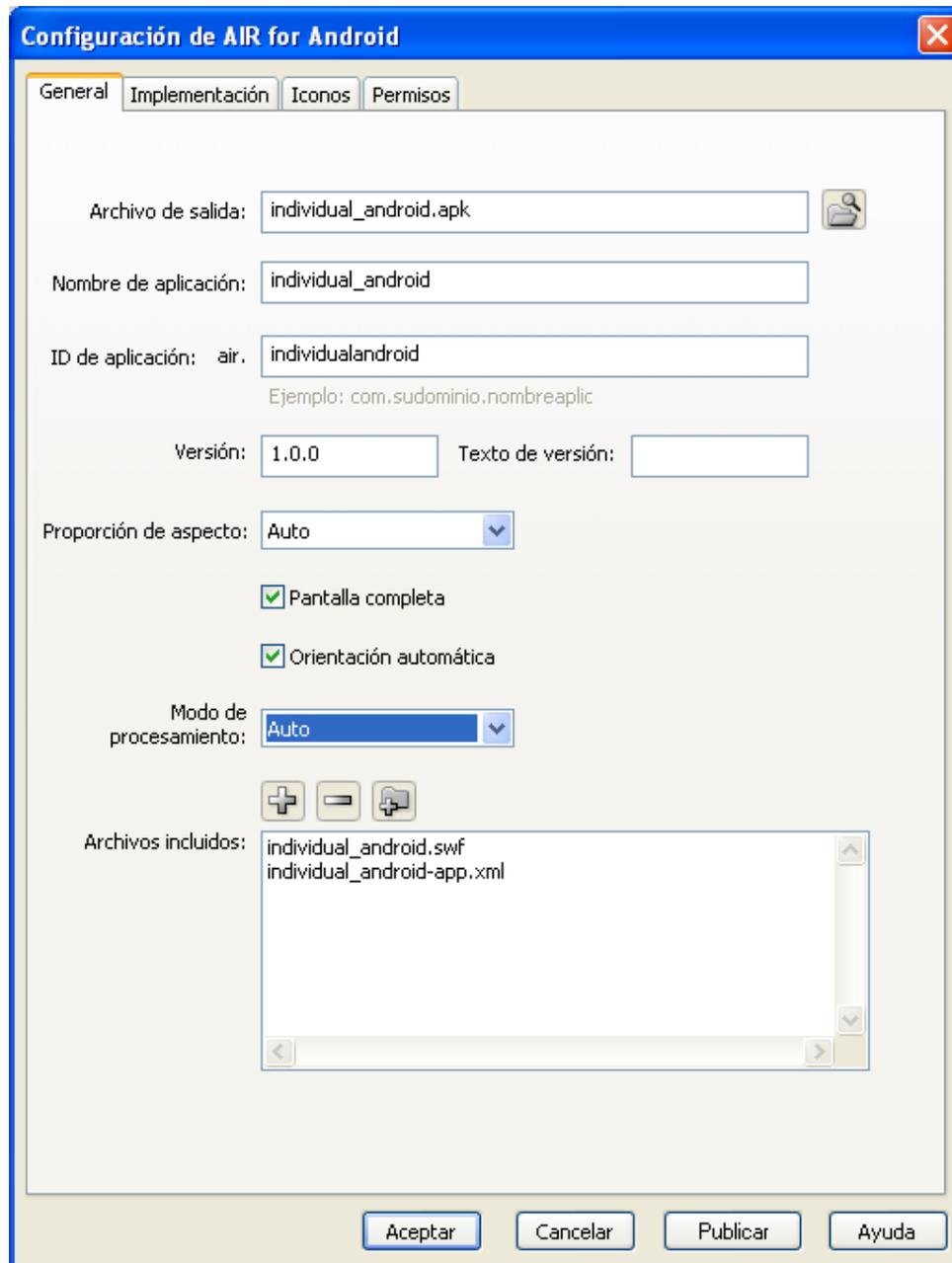


Fig. 77 Pantalla general de configuración air android

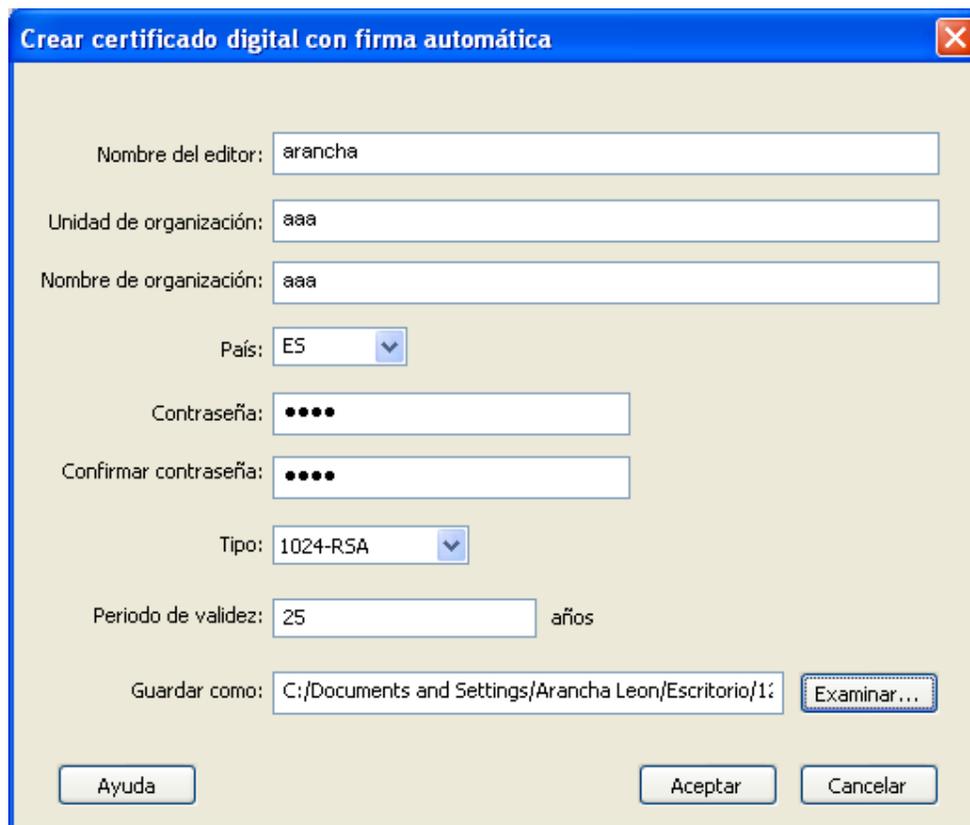
En la sección “IMPLEMENTACIÓN”:

- Pulsar sobre el botón “Crear certificado” y se abrirá una ventana como la que se muestra a continuación. En ella:

Rellenar todos los campos

El periodo de validez no debe de ser muy elevado o dará error al publicar. Se recomienda dejar el que viene por defecto (25 años).

Guardar en la carpeta donde se encuentran todos los archivos de la aplicación



Crear certificado digital con firma automática

Nombre del editor: arancha

Unidad de organización: aaa

Nombre de organización: aaa

País: ES

Contraseña: ●●●●

Confirmar contraseña: ●●●●

Tipo: 1024-RSA

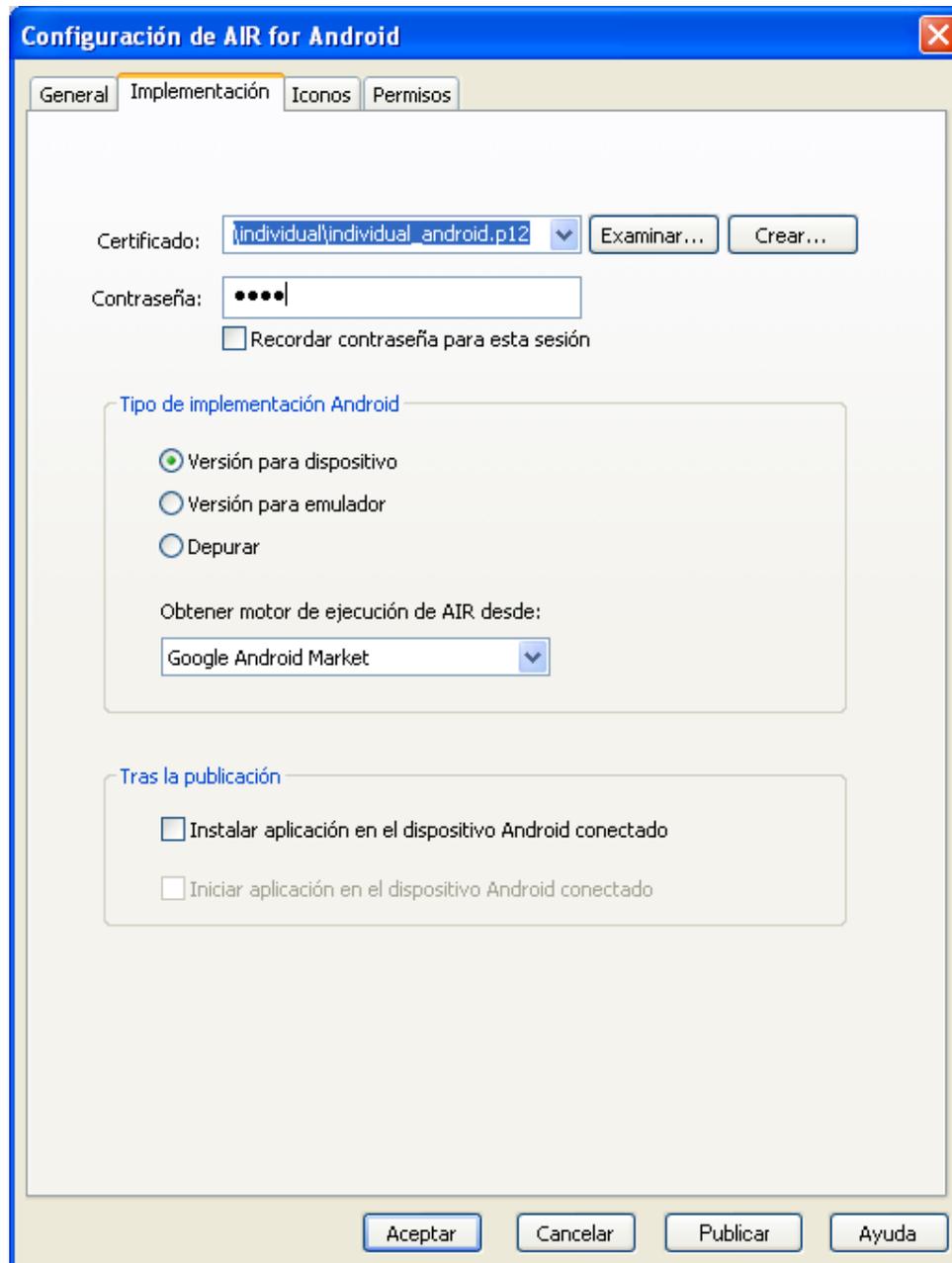
Periodo de validez: 25 años

Guardar como: C:/Documents and Settings/Arancha Leon/Escritorio/1: Examinar...

Ayuda Aceptar Cancelar

Fig. 78 Pantalla crear certificado digital

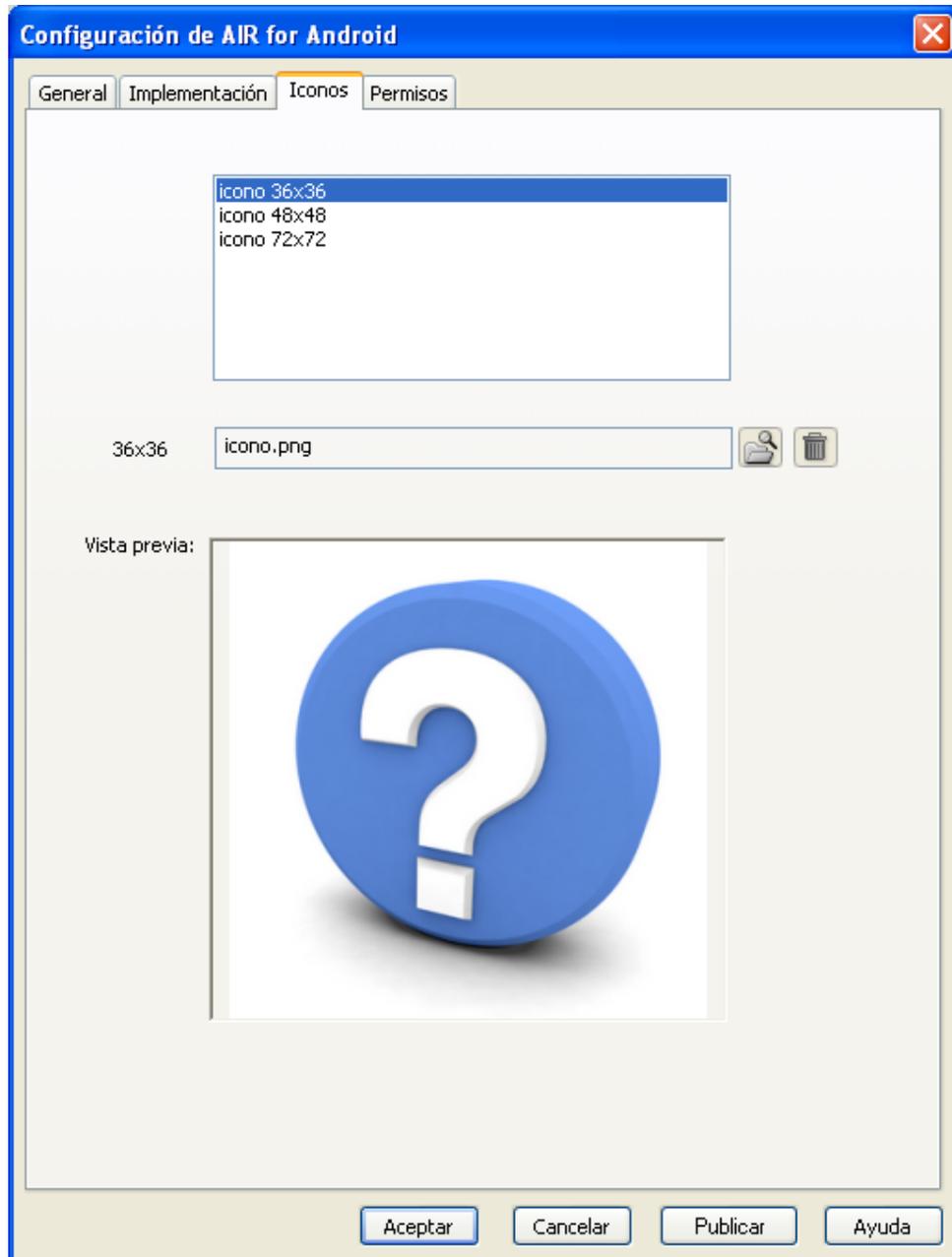
- Rellenar contraseña
- Seleccionar “Versión para dispositivo”.



**Fig. 79** Pantalla implementación de configuración air android

En la sección “ICONO”:

- Seleccionar el tamaño que se desea que tenga el icono.
- Seleccionar la imagen que desea colocar como icono.
- La imagen seleccionada debe tener el mismo tamaño que el tamaño seleccionado para el icono.



**Fig. 80 Pantalla iconos de configuración air android**

En la sección “PERMISOS”:

- Seleccionar todos los permisos que se requieran para la aplicación.

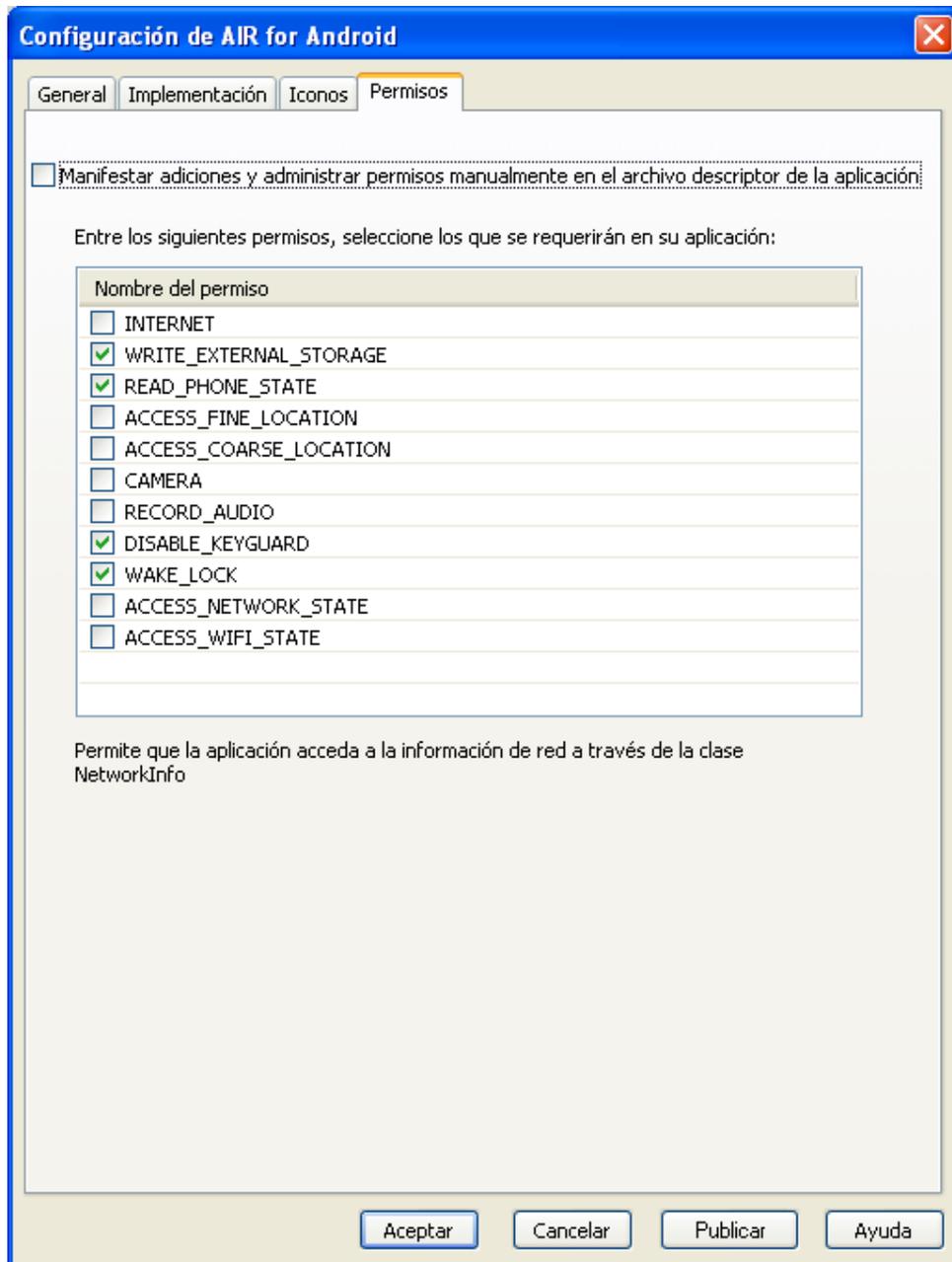


Fig. 81 Pantalla permisos de configuración air android

4º Pulsar sobre el botón Publicar.

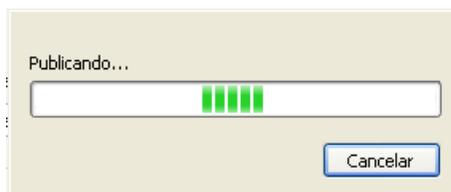


Fig. 82 captura pantalla de publicar

Una vez finalizado este proceso ya se ha creado un archivo .apk en la carpeta donde se encuentran los demás archivos de la aplicación.

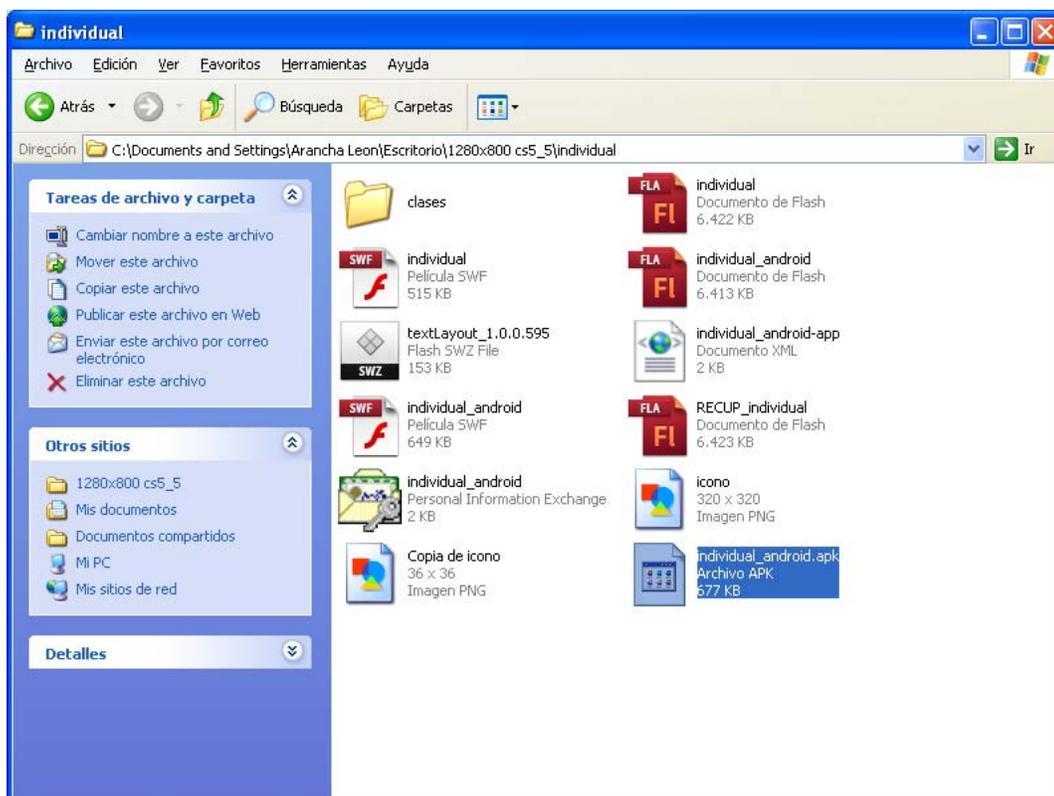


Fig. 83 Archivo creado .apk

## ANEXO II: IMPORTAR ARCHIVO APK A UN DISPOSITIVO ANDROID

En este anexo se describen los pasos a seguir de una de las maneras para pasar un archivo .apk desde un PC con sistema operativo Windows a un dispositivo android.

1º Instalar sdk de android

1º Descargar android SDK (kit de desarrollo de software).

En la página de android existen versiones disponibles del SDK de Android para Windows, Mac Os X y Linux. Seleccionar la versión para Windows y descargar.

<http://developer.android.com/sdk/index.html>

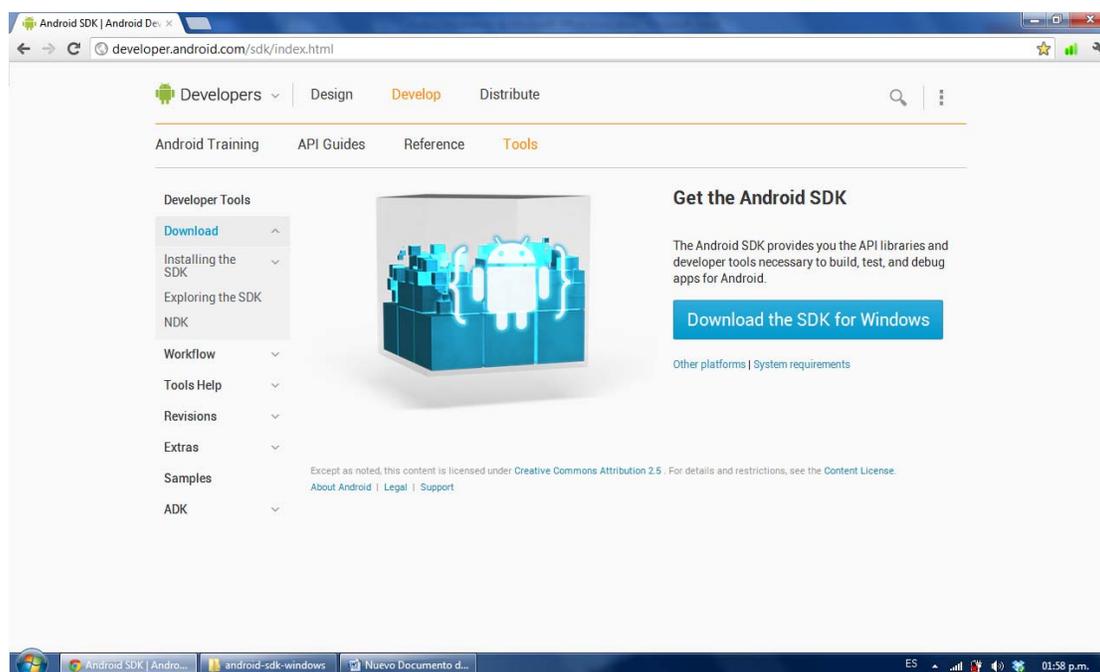


Fig. 84 Página de descarga android SDK

- 2º Descomprimir el fichero descargado en el directorio deseado.  
En este caso se ha decidido instalarlo en el escritorio.
- 3º Instalar siguiendo los pasos del instalador.



Fig. 85 Pantalla instalador

- 2º Tener el dispositivo android en modo depuración de USB  
Ajustes-> aplicaciones-> depuración de USB

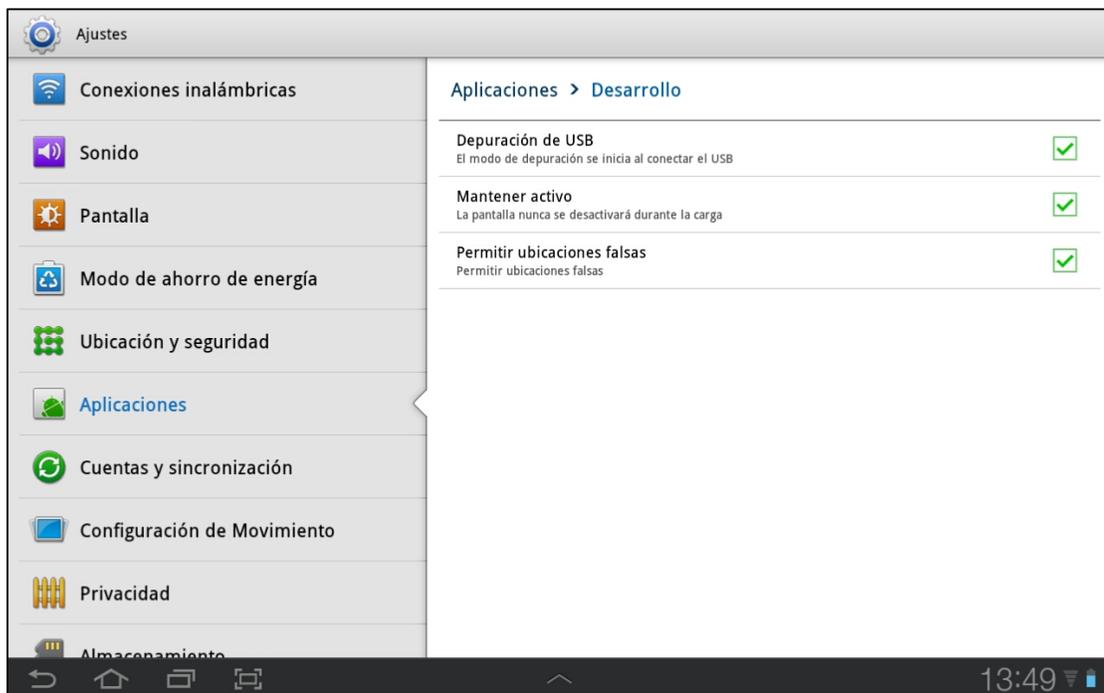


Fig. 86 Pantalla ajustes Samsung Galaxy Tab

3º Conectar el dispositivo android al PC mediante un cable USB



**Fig. 87 Conexión física Tableta-Pc**

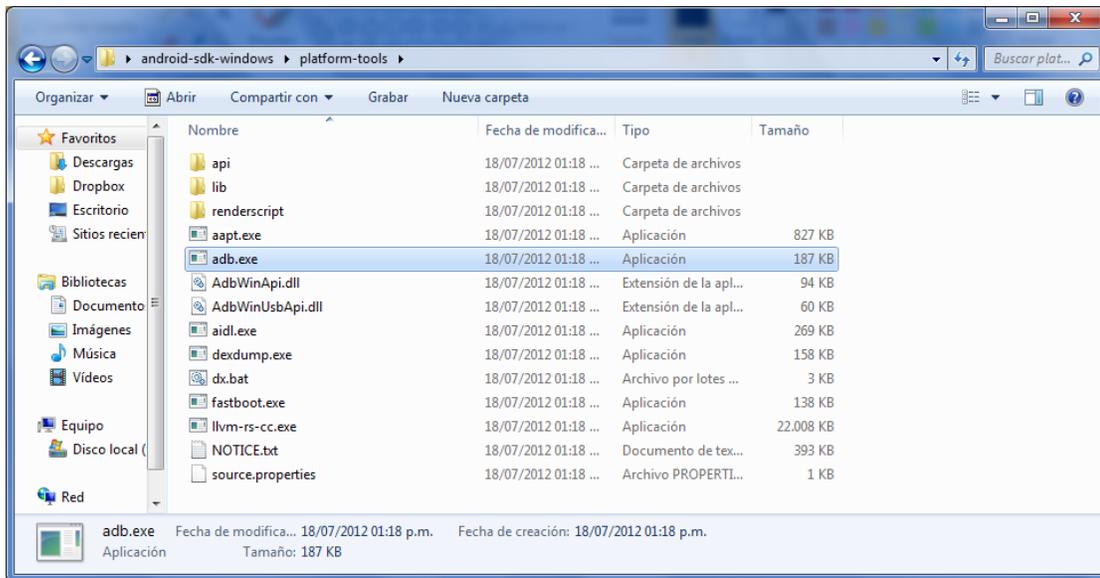
Para que el dispositivo android sea detectado correctamente por el PC hay que asegurar que están instalados en el PC todos los drivers del dispositivo android. Por ejemplo, en el caso del Samsung galaxy tab es necesario instalar el programa “Samsung Kies” que es el que contiene dichos drivers.

4º Localizar el ejecutable adb y añadirlo al path.

1º Abrir Símbolo del sistema.

2º Cambiar el directorio al directorio donde se encuentra el archivo adb.exe

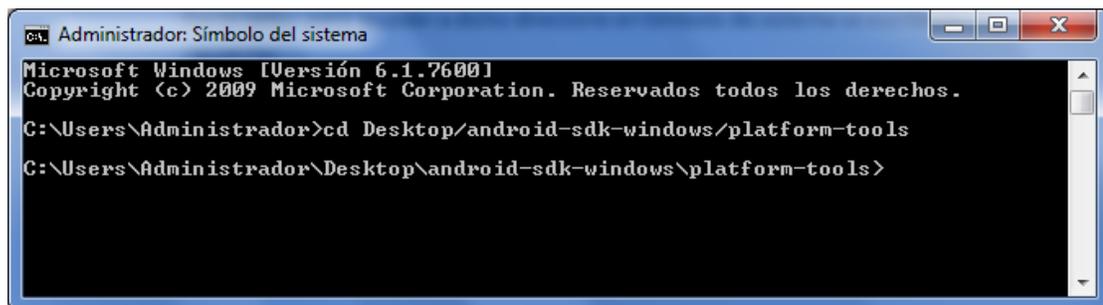
En este caso, como el SDK android se ha instalado en el escritorio, el archivo adb.exe se encuentra en: Desktop -> android-sdk-windows -> platform-tools; como se muestra en la siguiente imagen.



**Fig. 88 Directorio donde se encuentra adb.exe**

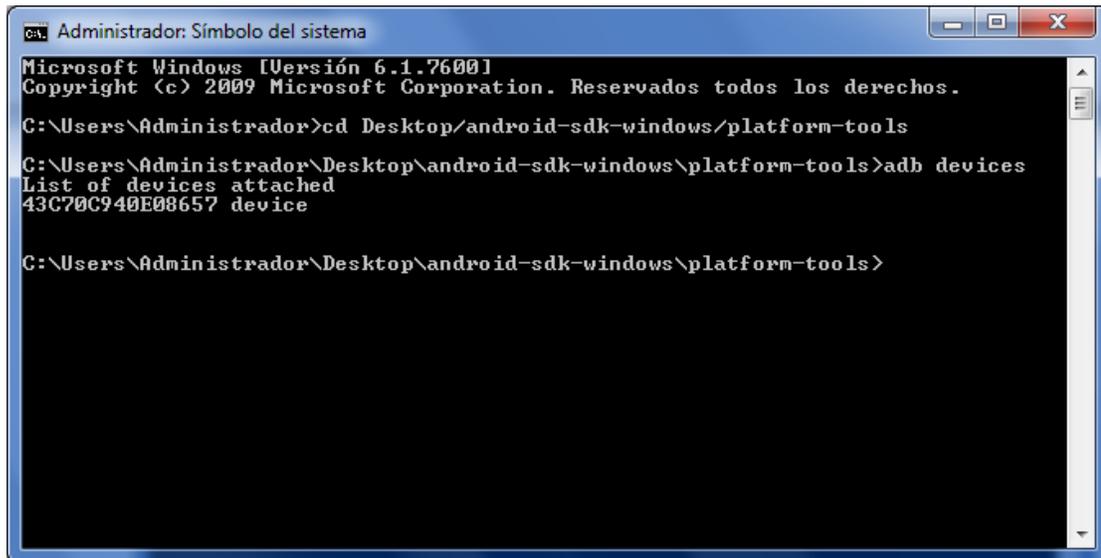
Por lo tanto, para acceder a dicho directorio en Símbolo de sistema se escribe lo siguiente:

```
cd Desktop/android-sdk-windows/platform-tools
```



**Fig. 89 Comandos para cambiar de directorio**

- 5º Escribir en Símbolo de sistema: adb devices



```
C:\Users\Administrador>cd Desktop/android-sdk-windows/platform-tools
C:\Users\Administrador\Desktop\android-sdk-windows\platform-tools>adb devices
List of devices attached
43C70C940E08657 device

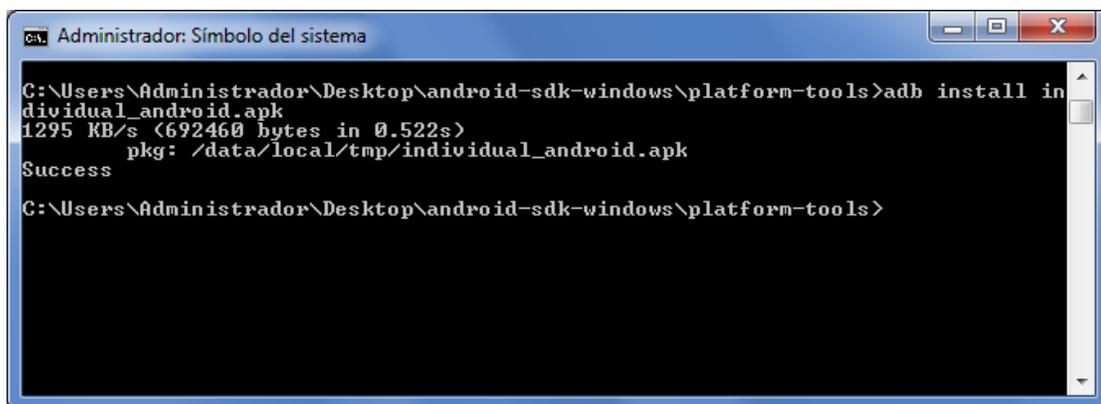
C:\Users\Administrador\Desktop\android-sdk-windows\platform-tools>
```

Fig. 90 Comando para reconocer tableta

- 6º Colocar el archivo .apk en la misma carpeta que se encuentra el archivo abd.exe

- 7º Instalar archivo .apk

Para ello escribir: adb install nombreaplicación.apk

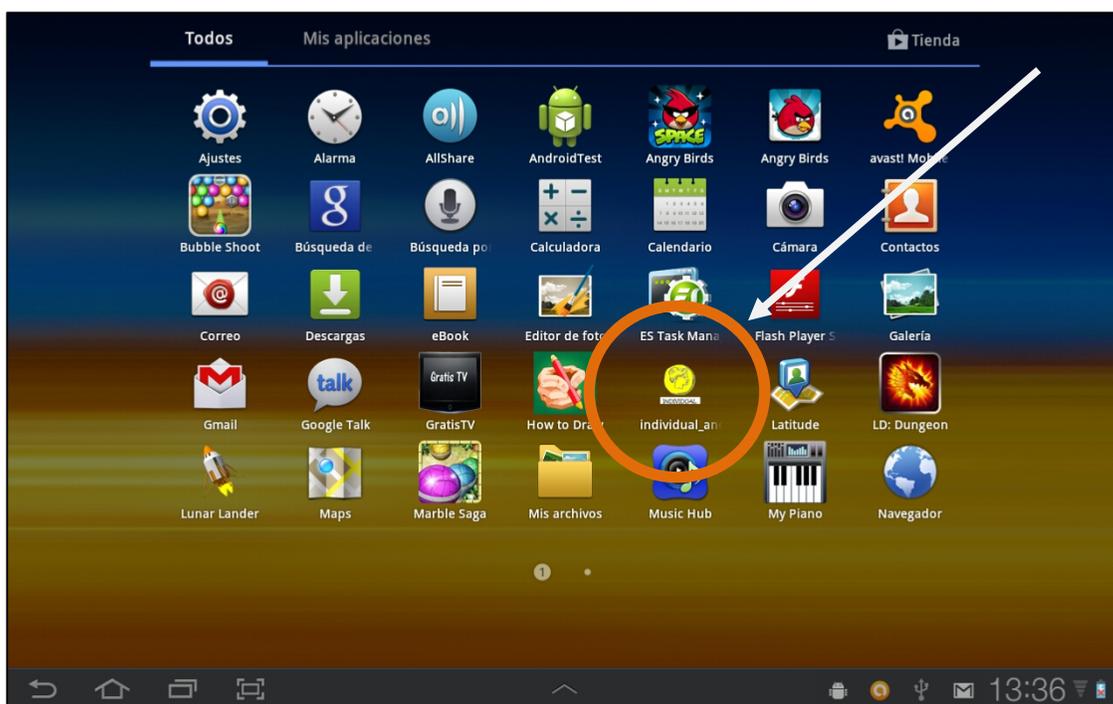


```
C:\Users\Administrador\Desktop\android-sdk-windows\platform-tools>adb install individual_android.apk
1295 KB/s (692460 bytes in 0.522s)
Success
pkg: /data/local/tmp/individual_android.apk

C:\Users\Administrador\Desktop\android-sdk-windows\platform-tools>
```

Fig. 91 Comando instalar archivo .apk

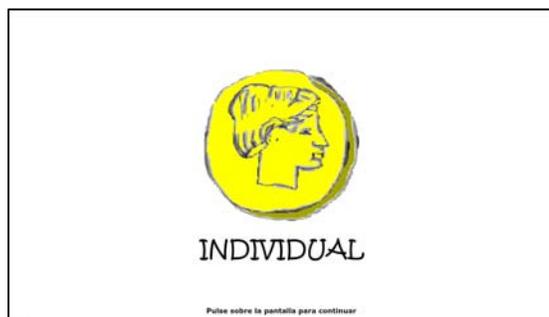
De esta forma la aplicación ya ha sido instalada en el dispositivo android y aparecerá en el menú de aplicaciones de dicho dispositivo como se puede ver en la siguiente imagen.



## ANEXO III MANUAL DE USUARIO

### ➤ MODO INDIVIDUAL

Al abrir el juego aparece una pantalla como la figura 92 que indica el modo de juego abierto, en este caso el modo individual. Pulsar sobre cualquier parte de la pantalla para pasar a la siguiente.



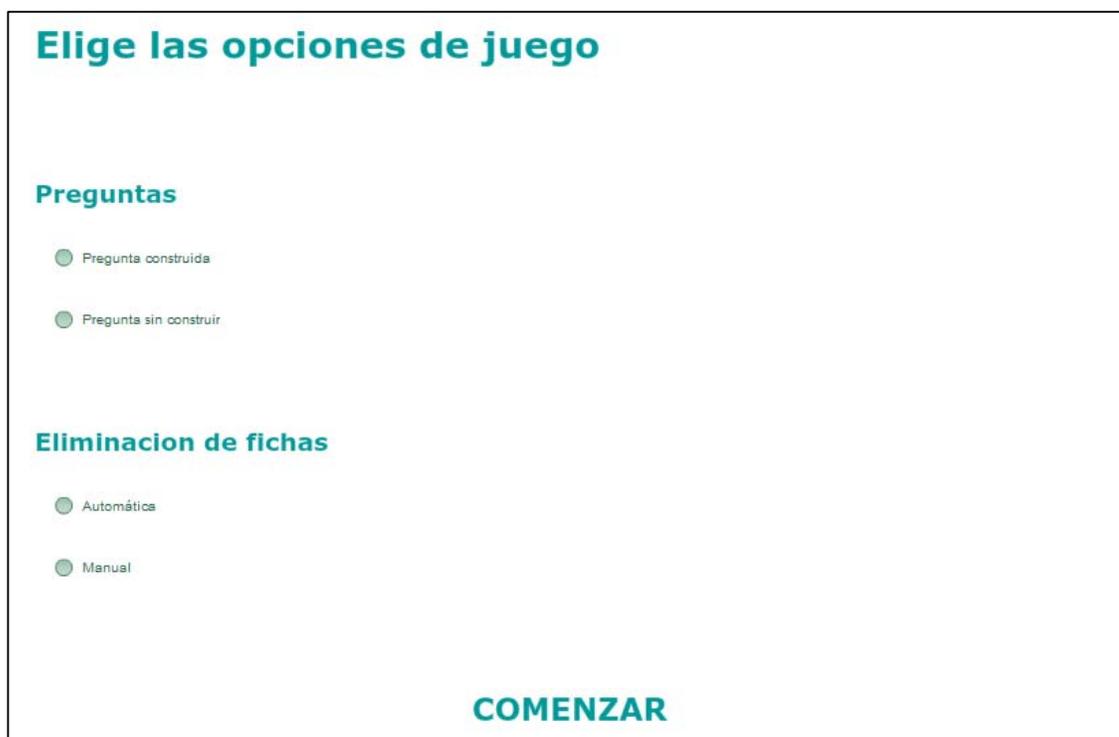
**Fig. 92 Portada, modo de juego individual**

A continuación, aparecen una serie de pantallas con pequeñas animaciones contando una pequeña historia. Leer el texto y pulsar sobre cualquier parte de la pantalla para ir pasando de una a otra.



**Fig. 93 Animación inicial**

Después de esta pequeña historia, aparece la pantalla inicial donde el profesional debe seleccionar el nivel de la jugada.



**Elige las opciones de juego**

**Preguntas**

Pregunta construida

Pregunta sin construir

**Eliminacion de fichas**

Automática

Manual

**COMENZAR**

**Fig. 94 Pantalla inicial**

El profesional tiene cuatro combinaciones entre las que elegir:

- pregunta construida + eliminación de fichas automática
- pregunta construida + eliminación de fichas manual
- pregunta sin construir + eliminación de fichas automática
- pregunta sin construir + eliminación de fichas manual

Para seleccionar la combinación deseada basta con pulsar sobre los radio buttons hasta que se ponga un pequeño círculo negro en el centro del círculo de la opción deseada. Una vez que se ha seleccionado el nivel deseado, pulsar sobre el botón “Comenzar” y aparecerá la interfaz del nivel seleccionado.

La interfaz de los niveles “pregunta construida + eliminación de fichas automática” y “pregunta construida + eliminación de fichas manual” se muestra en la figura 95. Mientras que el interfaz de los niveles “pregunta sin construir + eliminación de fichas automática” y “pregunta sin construir + eliminación de fichas manual” es la que se muestra en la figura 96.

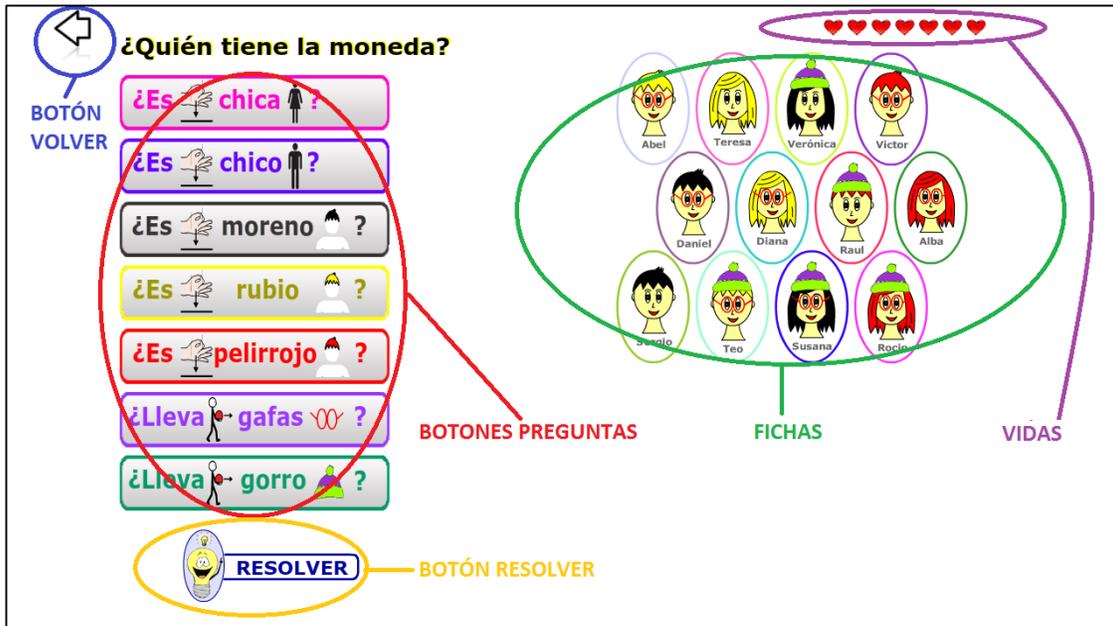


Fig. 95 Pantalla pregunta construida eliminación de fichas automática o manual

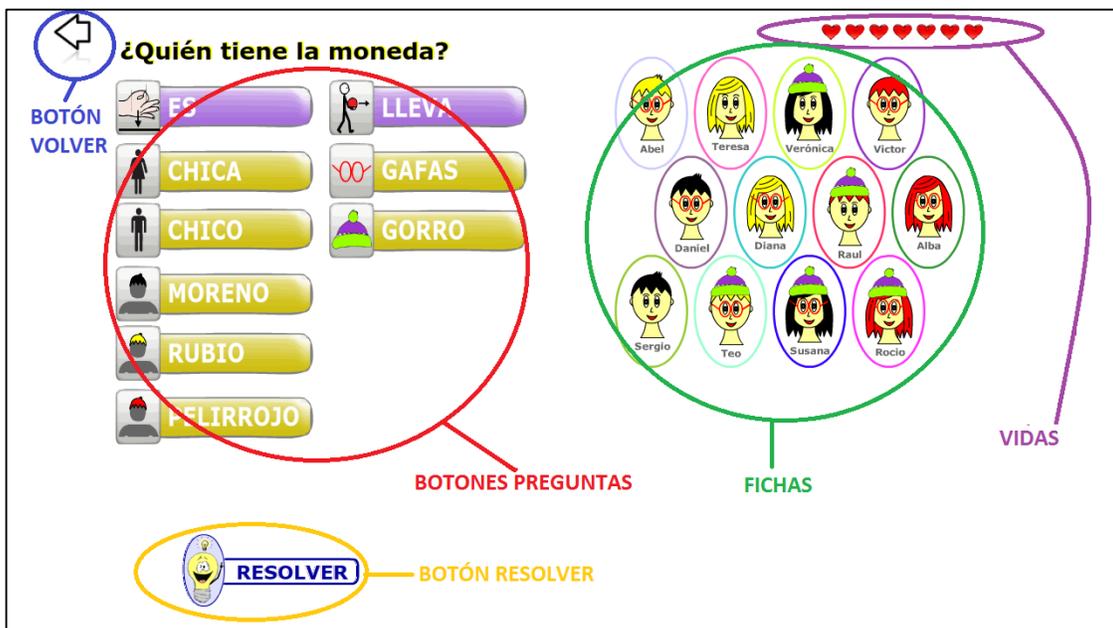


Fig. 96 Pantalla pregunta construida eliminación de fichas automática o manual

Como se puede observar en las figuras anteriores, aunque son cuatro niveles de dificultad cada uno distinto al resto, tienen las partes de la interfaz común. También los pasos a seguir por el jugador para jugar son los mismos en los cuatro casos.

Si se pulsa sobre el botón “Volver” el juego y volverá a la pantalla inicial que se muestra en la figura 94.

El jugador debe de pulsar sobre los “botones preguntas” hasta adivinar el personaje que tiene la moneda. Para ello seguirá los siguientes pasos.

#### 1º Realizar pregunta.

El jugador debe de realizar una pregunta sobre el aspecto del personaje a adivinar. Para ello, si se ha elegido la opción “Pregunta construida”, pulsará sobre un botón pregunta.

Si por el contrario se ha escogido la opción “Pregunta sin construir” deberá de pulsar sobre dos botones. Estos dos botones deben de formar correctamente una frase, es decir, primero se pulsará sobre un botón que contenga un verbo y después un botón que contenga un complemento adecuado.

Conforme se pulse sobre los botones se escribirá la pregunta formulada y su respectiva respuesta debajo de las fichas. Si la frase se ha construido de una forma errónea saldrá como respuesta a la cuestión un mensaje de error.

#### 2º Eliminar fichas

Observado la pregunta formulada y la respuesta dada, si se ha escogido la opción “Eliminación de fichas manual” el jugador tiene que eliminar las fichas que no correspondan con las características del personaje que tiene la moneda. Por ejemplo, si la pregunta formulada es “¿Es chica?” y la respuesta en “No”, se deben de eliminar todas las fichas que tengan chicas. Por el contrario, si frente a la misma

pregunta la respuesta es “Sí” se deberán de eliminar todas las fichas que contengan chicos.

Para eliminar las fichas solo tiene que pulsar una vez sobre las fichas que desea eliminar. Si por un casual el jugador se confunde al eliminar una ficha, basta con volver a pulsar sobre ella para que esta aparezca.

Si en la opción de “Eliminación de fichas” se ha escogido “automática” este paso lo hará automáticamente la máquina, es decir, el jugador solo tiene que observar y comprender por qué la máquina ha eliminado las fichas que ha eliminado.

- 3º Repetir los pasos 1º y 2º hasta que se solo quede una ficha. Esta última ficha será la ficha que contenga el personaje que tiene la moneda, es decir, el resultado.

Hay que prestar atención para llegar al resultado antes de que el jugador se quede sin vidas pues cada vez que se formula una pregunta se elimina una vida. Si esto ocurre se terminará el juego con un mensaje como el que se muestra en la figura 97.

Para salir del juego y volver a la pantalla inicial pulsar sobre cualquier parte del escenario.



Fig. 97 Mensaje sin vidas

#### 4º Resolver

Una vez se conoce el resultado, el jugador debe pulsar sobre el botón resolver y a continuación pulsar sobre la ficha que contiene el personaje que tiene la moneda. Al hacer esto se eliminarán las preguntas y respuestas escritas debajo de las fichas y en su posición aparecerá una pregunta que dice “¿Tiene la moneda\_\_\_” seguido del nombre del personaje que se ha seleccionado como resultado.

Al lado de dicha cuestión también aparece el resultado de dicha pregunta. Si el resultado es positivo, el jugador ha ganado mientras que si el resultado es negativo, el jugador ha perdido la partida.

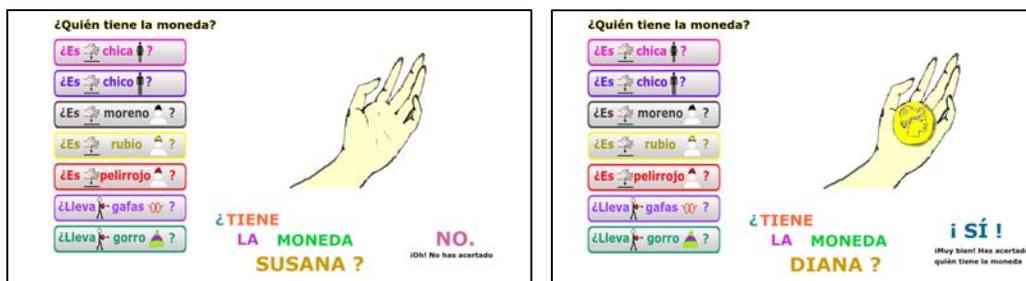


Fig. 98 Posibles resultados de la partida

#### 5º Finalizar el juego

Para salir de esta pantalla, una vez conocido si el jugador ha ganado o no la partida, hay que pulsar sobre cualquier parte del escenario y a continuación aparecerá otra pantalla con el aspecto de la figura 99.



### **Fig. 99 Pantalla fin del juego**

Dicha pantalla contiene dos botones. Si se pulsa sobre el botón “Ver preguntas-respuestas” se vuelve a la pantalla anterior al paso 4º (Resolver). Sin embargo, no se puede continuar jugando. Si se pulsa sobre cualquier parte de la pantalla volverá a aparecer la pantalla de la figura 99.

Si por el contrario se pulsa sobre el botón “Volver a jugar” aparece de nuevo la pantalla de inicio de la figura 94 para comenzar a jugar de nuevo otra partida distinta.

## ➤ MODO PAREJA

Al abrir el juego aparece una pantalla como la figura 100 que indica el modo de juego abierto, en este caso el modo pareja. Pulsar sobre cualquier parte de la pantalla para pasar a la siguiente.



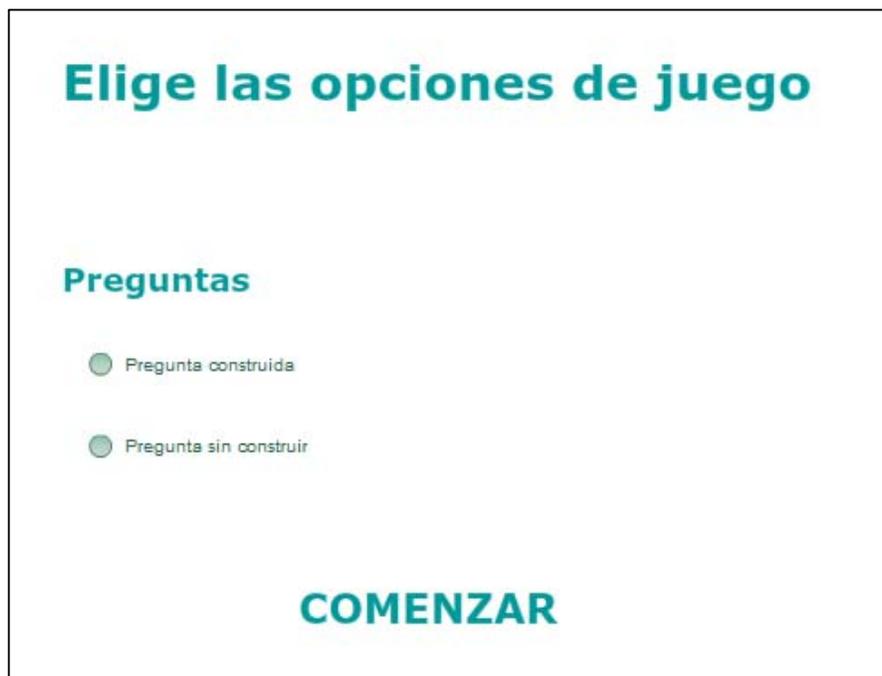
Fig. 100 Portada, modo de juego pareja

A continuación aparece la misma historia corta que en el modo individual. Leer el texto y pulsar sobre cualquier parte de la pantalla para ir pasando de una a otra.



Fig. 101 Animación inicial

Después de esta pequeña historia, aparece la pantalla inicial donde el profesional debe seleccionar el nivel de la jugada.



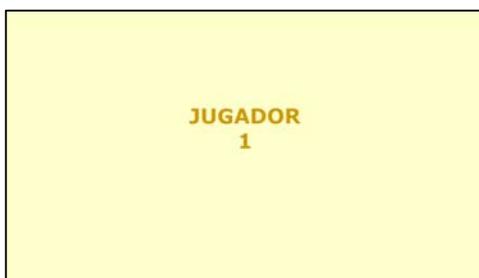
**Fig. 102 Pantalla inicial**

El profesional tiene dos niveles donde elegir.

- pregunta construida
- pregunta sin construir

Para seleccionar la opción deseada basta con pulsar sobre el radio buttons hasta que se ponga un pequeño círculo negro en el centro del círculo de la opción deseada. Una vez que se ha seleccionado el nivel, pulsar sobre el botón “Comenzar”.

A continuación aparecerá la pantalla de la figura 103 que indica que le toca jugar solo al jugador 1.



**Fig. 103 Jugador 1**

Para pasar de pantalla, el jugador 1 debe de pulsar sobre cualquier punto de la pantalla. Es importante que hasta que se diga lo contrario el jugador 2 no mire la pantalla.

En la siguiente pantalla el jugador 1 debe de seleccionar el personaje que desea que se guarde la moneda. Para ello, debe de presionar sobre el personaje elegido y pulsar el botón “Guardar” que aparece cuando hay un personaje seleccionado.

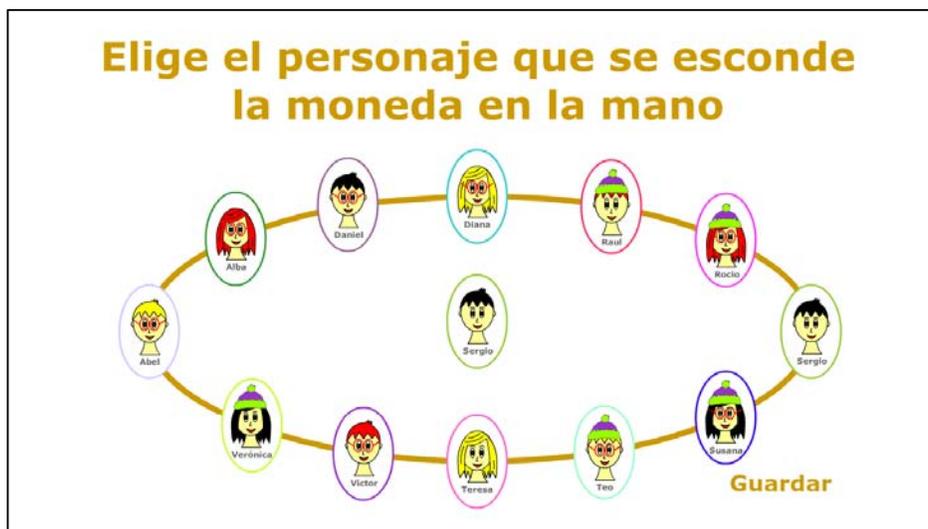


Fig. 104 Selección de personaje

Posteriormente, aparece una pantalla para asegurar que se ha escogido el personaje que realmente se deseaba.

Si se desea elegir otro personaje, pulsar sobre el botón “No”. De esta forma aparecerá la pantalla anterior y en ella, se podrá escoger de nuevo otro personaje. Si se selecciona el botón “Sí” se continúa con el juego.



Fig. 105 Asegurar personaje

Es importante que el jugador 1 recuerde en todo momento qué personaje ha seleccionado.

La siguiente pantalla indica que ya juegan ambos jugadores, tanto el jugador 1 como el jugador 2 pueden mirar la pantalla. Pulsar sobre cualquier punto de esta para pasar a la siguiente.



Fig. 106 Jugador 1 y jugador 2

La interfaz de la siguiente pantalla depende del nivel de juego escogido. Si el nivel ha sido "Pregunta construida" la interfaz de la pantalla es la que se muestra en la figura 107 mientras que si el nivel escogido ha sido "Pregunta sin construir" es la de la figura 108. Ambas, en un principio son similares a las del juego individual con la pequeña diferencia de que estas tienen un bocadillo en el que indica a qué jugador le toca jugar.

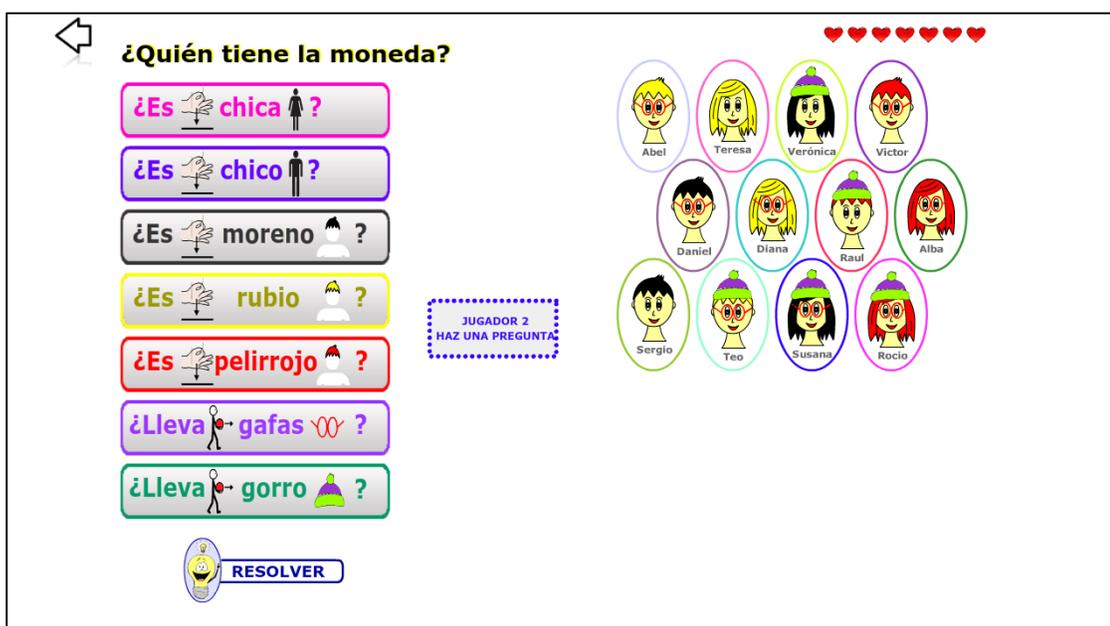


Fig. 107 Nivel pregunta construida

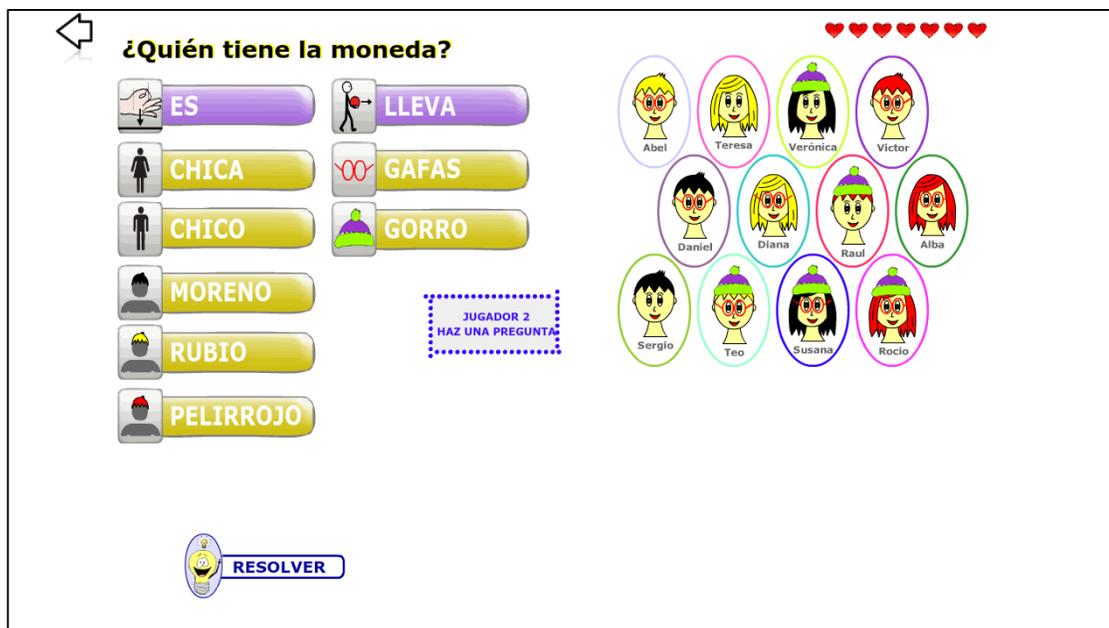


Fig. 108 Nivel pregunta sin construir

Los pasos a seguir por el jugador 2 son los pasos descritos en el modo individual, por lo que no se van a volver a describir.

Por lo tanto, lo único que varía en este modo de juego, a parte de las pantallas anteriormente explicadas, es que entre los pasos “Realizar pregunta” y “Eliminar fichas” realizados por el jugador 2, el jugador 1 añade un paso intermedio que es el de responder a la cuestión.

A diferencia de en el modo individual, ahora la máquina no responde a las preguntas realizadas sino que es el jugador 1 en que lo hace. Para ello solo tiene que pulsar sobre el botón “Sí” o “No” que aparece seguido de la pregunta realizada por el jugador 2 como se muestra en la figura 109.

← **¿Quién tiene la moneda?** ♥♥♥

¿Es  **chica**  ?

¿Es  **chico**  ?

¿Es  **moreno**  ?

¿Es  **rubio**  ?

¿Es  **pelirrojo**  ?

¿Lleva  **gafas**  ?

¿Lleva  **gorro**  ?

 **RESOLVER**

JUGADOR 1  
RESPONDE A LA  
PREGUNTA

  ¿ Es **chica** ? **Sí** ✓        ¿ Lleva **gorro** ? **Sí** ✓ **No** ✗  
  ¿ Es **moreno** ? **No** ✗  
  ¿ Lleva **gafas** ? **No** ✗

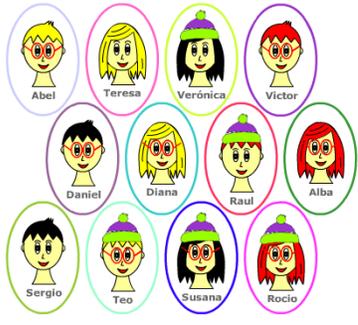


Fig. 109 Responder a la cuestión

Si el jugador 2 responde mal a alguna pregunta, en vez la respuesta “Sí” o “No” aparece un mensaje de error.

# “DESARROLLO DE UN SOFTWARE INTERACTIVO DE AYUDA AL APRENDIZAJE”

Arancha León Muñoz

Tutor: Alfredo Pina

Asesora externa: Isabel Sánchez

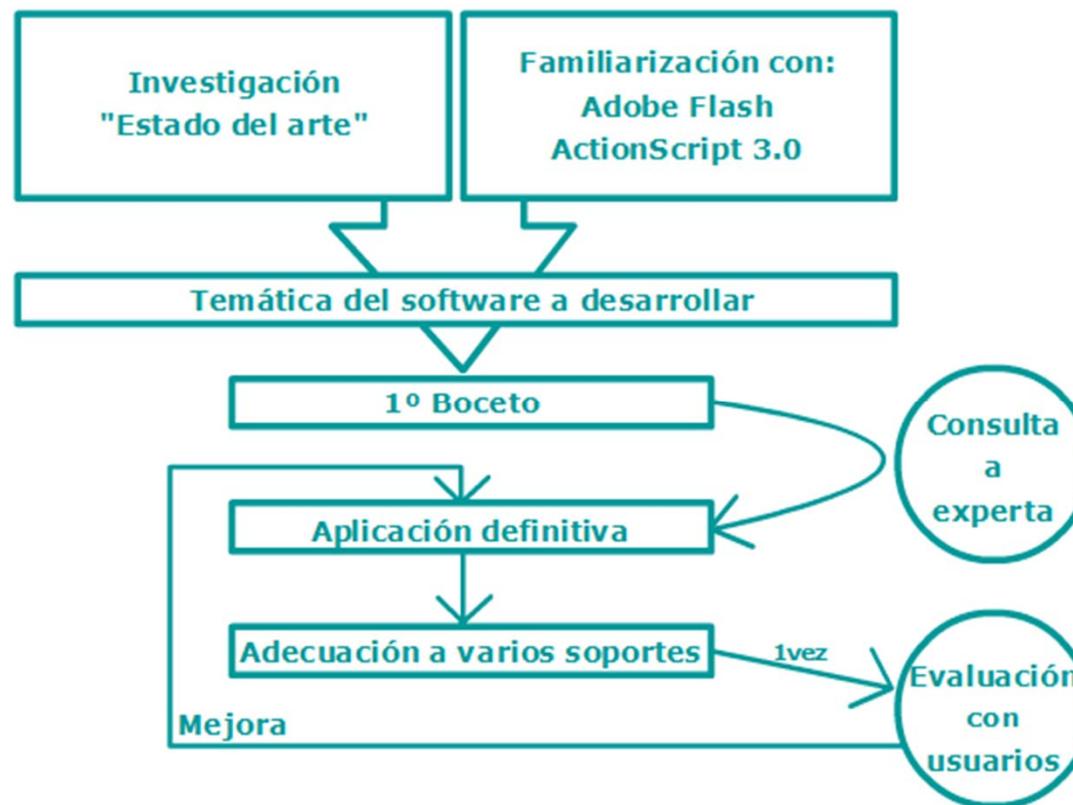
Pamplona, 10 de Septiembre del 2012

# Objetivos

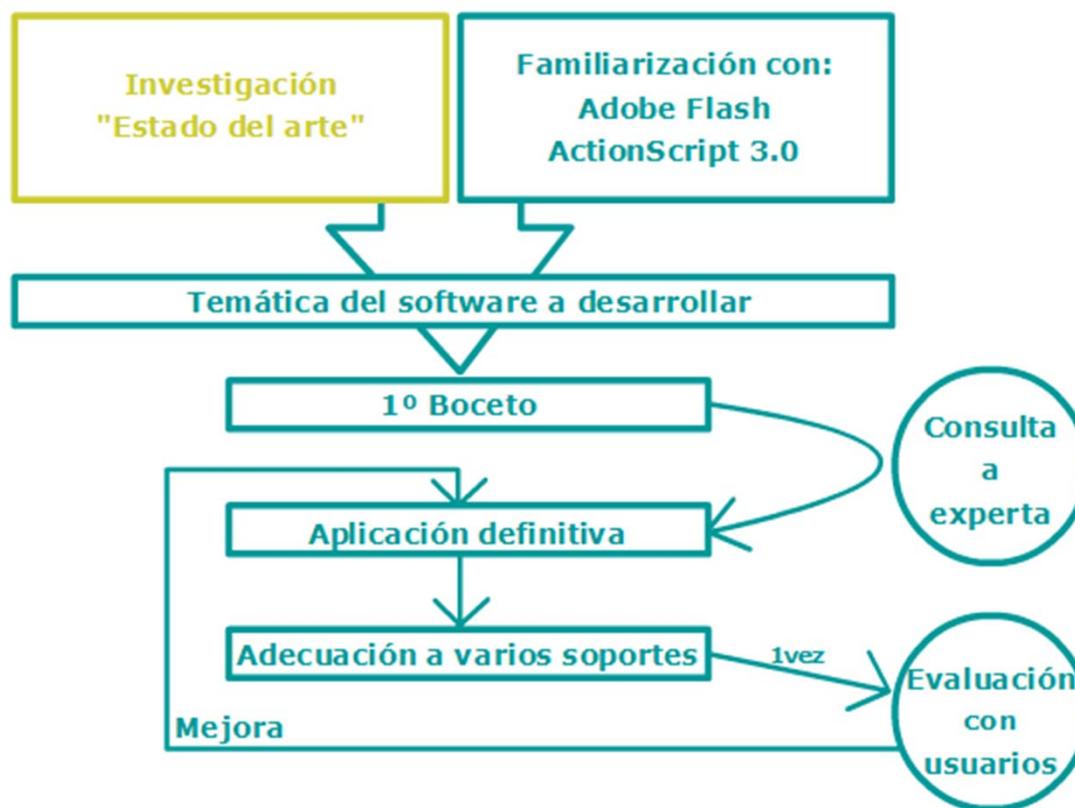
**Diseño, implementación y evaluación** de una **aplicación multimedia interactiva** que **ayude a la enseñanza, individual y cooperativo**, de **personas con trastornos generales de la comunicación**.

- Adquisición de conocimientos sobre el software de desarrollo Adobe flash Professional.
- Aprendizaje del lenguaje de programación ActionScript 3.0.
- Diseño e implementación de una aplicación destinada a la enseñanza de personas con trastornos generales de la comunicación.
- Adecuación del software a varios soportes (PC y tableta)
- Evaluación del software con personas de características similares a los destinatarios del producto.

# Fases del proyecto



# Investigación y “estado del arte”



# Estado del arte

## Recursos analógicos

### Pictogramas

El  malapata, tenía muy mala suerte, y todo le salía mal.

Un día secuestró a una , pero cuando se disponía a huir, le dispararon con un  , hundieron su  y la  pudo escapar.

### Tarjetas de relación



### Tabú

**futbolista**

1. Balón
2. Jugar
3. Real Madrid
4. Fútbol
5. Estadio

### ¿Quién es quién?

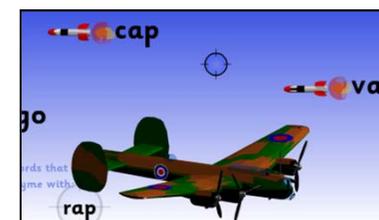


## Recursos digitales

### Busca la palabra



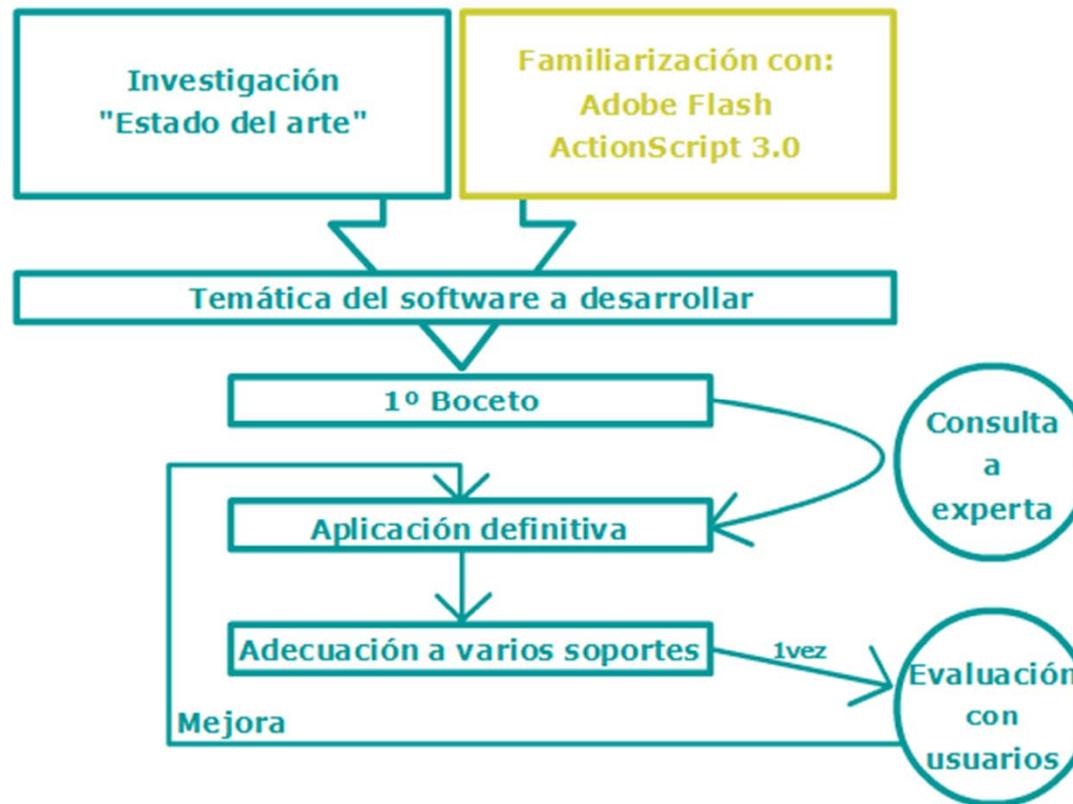
### Cohetes que riman



### ¿Quién es quién?



# Entorno de trabajo y lenguaje de programación



# Adobe Flash

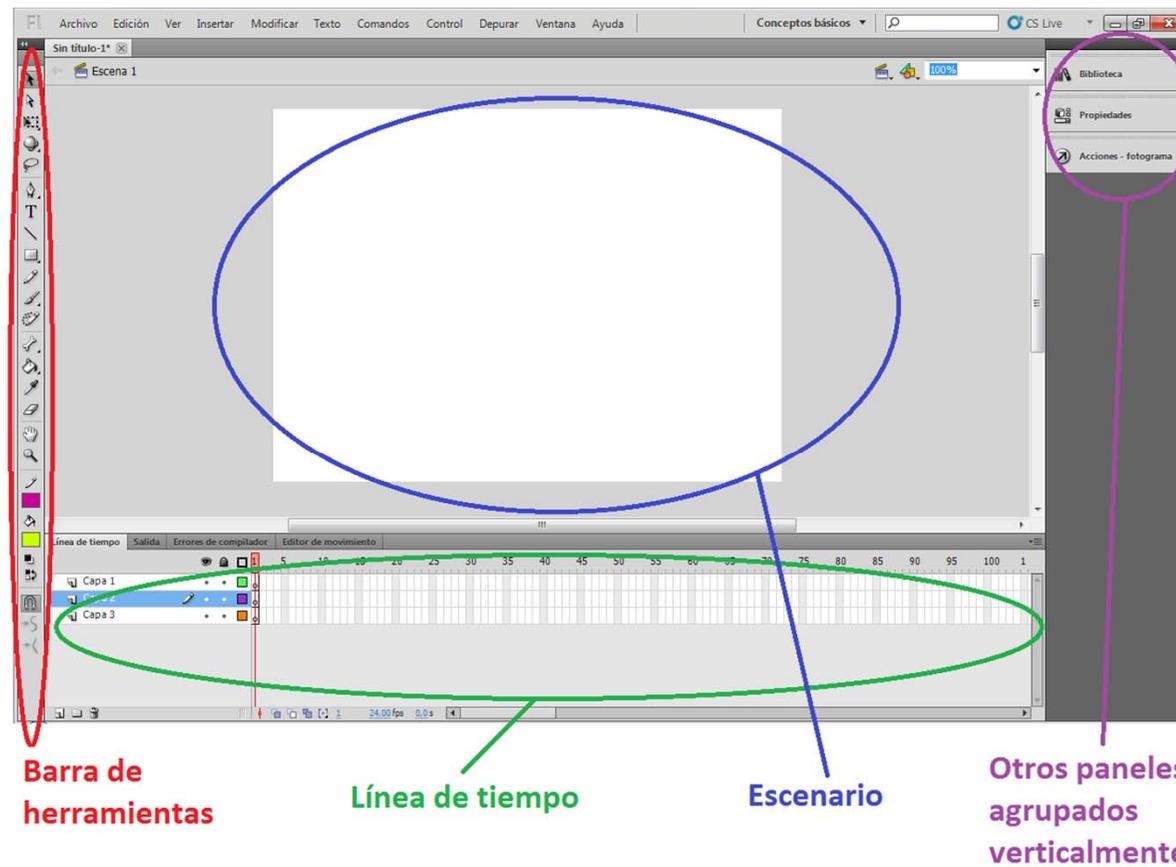


- Uno de los programas más populares de Adobe.
- Crea principalmente elementos multimedia e interactivos para Internet.
- Para ejecutar sus archivos en una plataforma cualquiera basta con tener instalado adobe flash player.
- Versiones empleadas:
  - Adobe Flash Professional CS5
  - Adobe Flash Professional CS5.5
- Tipo de archivos
  - .fla
  - .swf

# Adobe Flash



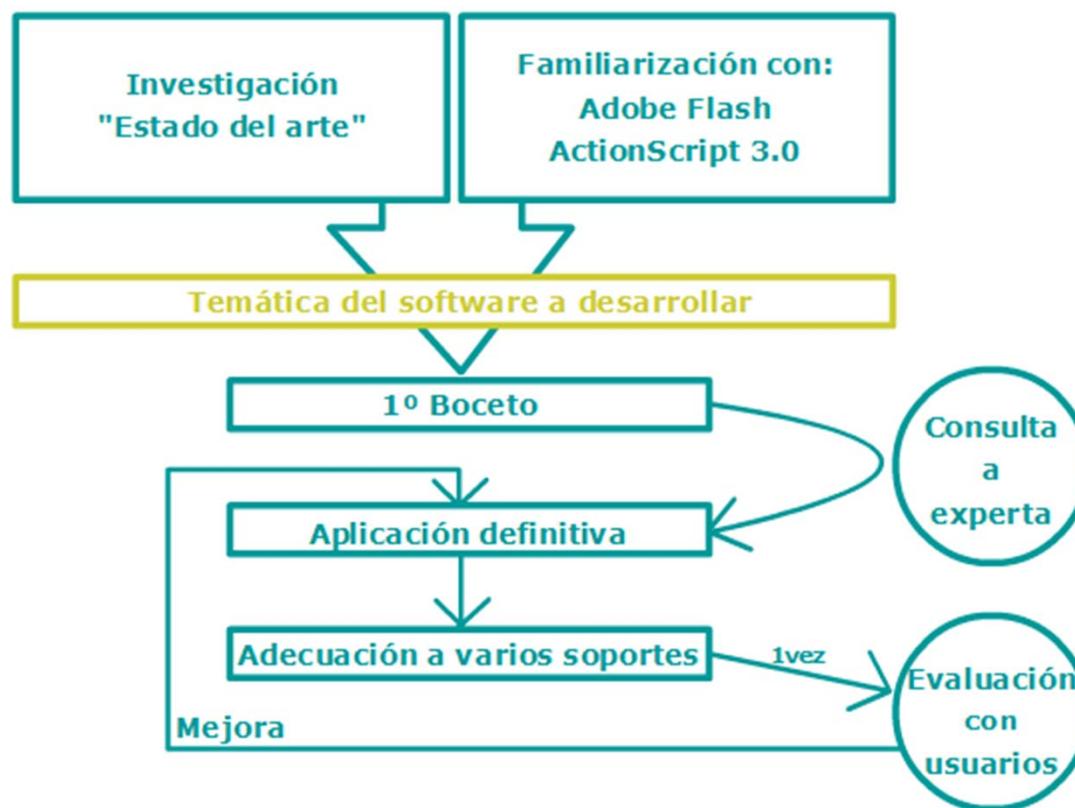
- Interfaz



# ActionScript

- Desarrollado como una forma para que los desarrolladores programen de forma más interactiva.
- Versión: 3.0
  - Mejora en el manejo de programación orientada a objetos (Ajuste al EDMA262).
  - Introduce mejoras de rendimiento
- Trabaja con:
  - Objetos
  - Variables
  - Métodos y funciones
  - Eventos
  - Estructuras de control
  - Clases

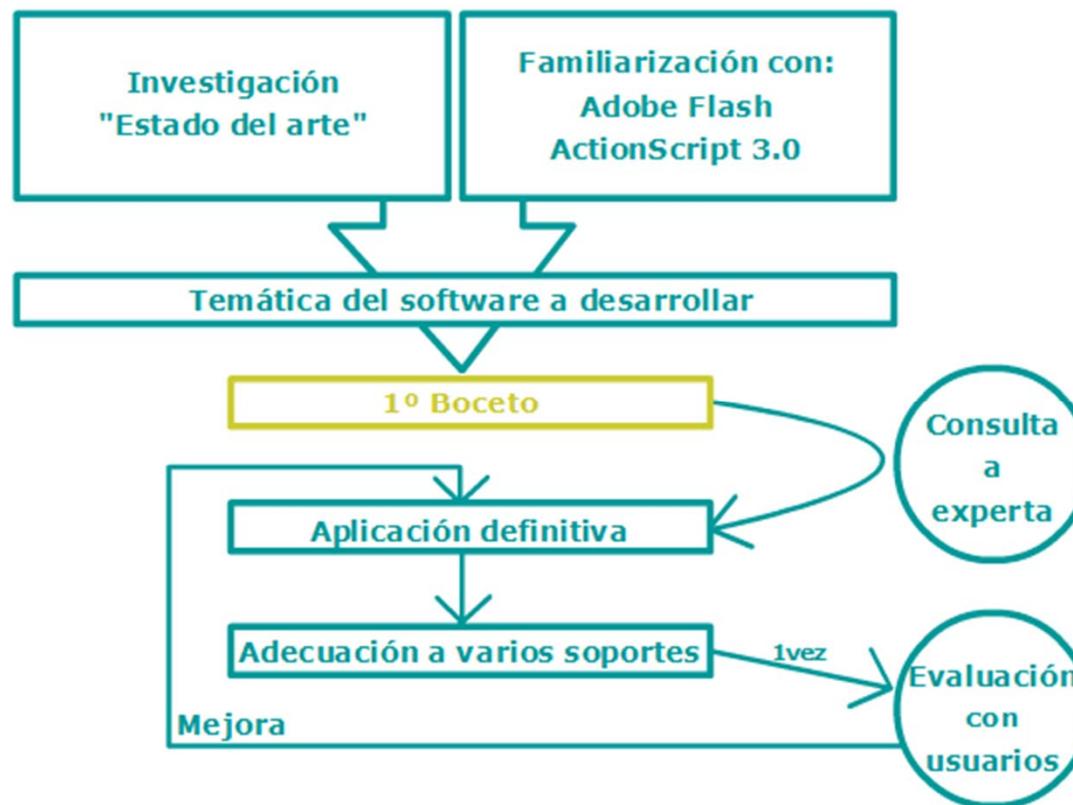
# Temática escogida



# Temática escogida

- Modalidad del juego **¿Quién es quién?**
  - Modalidades existentes no aportan nada nuevo a la versión analógica.
  - Bueno para mejorar la expresión y comprensión.
  - Juego analógico → juego competitivo  
Juego a desarrollar → juego individual y colaborativo
- Dos aplicaciones:
  - Individual
  - Cooperativo

# Primer boceto



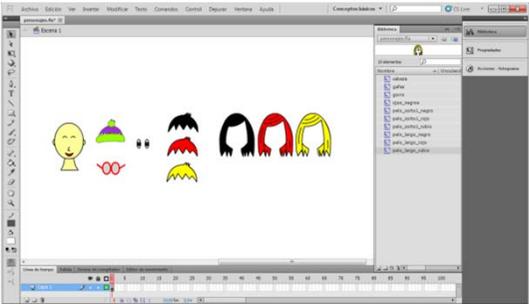
# Primer boceto

- Da idea de lo que se pretende hacer y su interfaz
- Solo modo individual
- Funcionamiento
- Eliminación de fichas

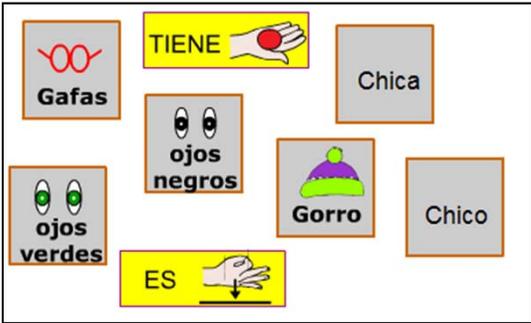
# Primer boceto

- Diseño interfaz

Fichas



Botones



Frases



# Primer boceto

Adobe Flash Player 10

Archivo Ver Control Ayuda

### ¿Quién es el personaje misterioso?

TIENE ES

Gafas Chica

ojos verdes Chico

Gorro ojos negros

Mi personaje TIENE GORRO

Mi personaje NO ES CHICA

Mi personaje NO TIENE GAFAS

Abel Teresa Verónica Victor

Daniel Diana Raul Alba

Sergio Teo Susana Rocio

Mi personaje TIENE los OJOS NEGROS

# Primer boceto

- Código
  - Simple
  - Sin clases
  - 3 partes:

## 1. Carga de fichas

```
1
2 ////////////////////////////////////////////////// CARGAR FICHAS //////////////////////////////////////
3
4
5 var FiAbel:FichaAbel= new FichaAbel();           FiAbel.x = 550;           FiAbel.y = 100;           addChild(FiAbel);
6 var FiTeresa:FichaTeresa = new FichaTeresa();     FiTeresa.x = 650;           FiTeresa.y = 100;           addChild(FiTeresa);
7 var FiVeronica:FichaVeronica = new FichaVeronica(); FiVeronica.x = 750;       FiVeronica.y = 100;       addChild(FiVeronica);
8 var FiVictor:FichaVictor = new FichaVictor();   FiVictor.x = 850;           FiVictor.y = 100;           addChild(FiVictor);
9
10 var FiDaniel:FichaDaniel = new FichaDaniel();   FiDaniel.x = 500;           FiDaniel.y = 225;           addChild(FiDaniel);
11 var FiDiana:FichaDiana = new FichaDiana();     FiDiana.x = 600;           FiDiana.y = 225;           addChild(FiDiana);
12 var FiRaul:FichaRaul = new FichaRaul();         FiRaul.x = 700;           FiRaul.y = 225;           addChild(FiRaul);
13 var FiAlba:FichaAlba = new FichaAlba();         FiAlba.x = 800;           FiAlba.y = 225;           addChild(FiAlba);
14
15 var FiSergio:FichaSergio = new FichaSergio();   FiSergio.x = 550;           FiSergio.y = 350;           addChild(FiSergio);
16 var FiTeo:FichaTeo = new FichaTeo();           FiTeo.x = 650;           FiTeo.y = 350;           addChild(FiTeo);
17 var FiSusana:FichaSusana = new FichaSusana();   FiSusana.x = 750;           FiSusana.y = 350;           addChild(FiSusana);
18 var FiRocio:FichaRocio = new FichaRocio();     FiRocio.x = 850;           FiRocio.y = 350;           addChild(FiRocio);
19
```

# Primer boceto

## 2. Elección de personaje y carga de sus propiedades

```
//////////////// ELEGIR PERSONAJE MISTERIOSO //////////////////////  
var numeroAleatorio:Number;  
numeroAleatorio= Math.floor(Math.random() * 12);
```

```
//////////////// CARGAR PROPIEDADES //////////////////////  
var gorro:Boolean = new Boolean;  
var gafas:Boolean = new Boolean;  
var ojosVerdes:Boolean = new Boolean;  
var ojosNegros:Boolean = new Boolean;  
var chico:Boolean = new Boolean;  
var chica:Boolean = new Boolean;  
  
switch(numeroAleatorio){  
  case 0: gorro = false; gafas = true; ojosVerdes = true; ojosNegros = false; chico = true; chica = false; break;  
  case 1: gorro = false; gafas = true; ojosVerdes = false; ojosNegros = true; chico = false; chica = true; break;  
  case 2: gorro = true; gafas = true; ojosVerdes = true; ojosNegros = false; chico = true; chica = false; break;  
  case 3: gorro = true; gafas = false; ojosVerdes = true; ojosNegros = false; chico = false; chica = true; break;  
  case 4: gorro = true; gafas = true; ojosVerdes = false; ojosNegros = true; chico = true; chica = false; break;  
  case 5: gorro = true; gafas = true; ojosVerdes = true; ojosNegros = false; chico = false; chica = true; break;  
  case 6: gorro = false; gafas = true; ojosVerdes = true; ojosNegros = false; chico = true; chica = false; break;  
  case 7: gorro = true; gafas = true; ojosVerdes = false; ojosNegros = true; chico = false; chica = true; break;  
  case 8: gorro = true; gafas = false; ojosVerdes = false; ojosNegros = true; chico = true; chica = false; break;  
  case 9: gorro = true; gafas = false; ojosVerdes = true; ojosNegros = false; chico = false; chica = true; break;  
  case 10: gorro = false; gafas = true; ojosVerdes = false; ojosNegros = true; chico = false; chica = true; break;  
  case 11: gorro = true; gafas = false; ojosVerdes = false; ojosNegros = true; chico = true; chica = false; break;  
}
```

# Primer boceto

## 3. Parte interactiva

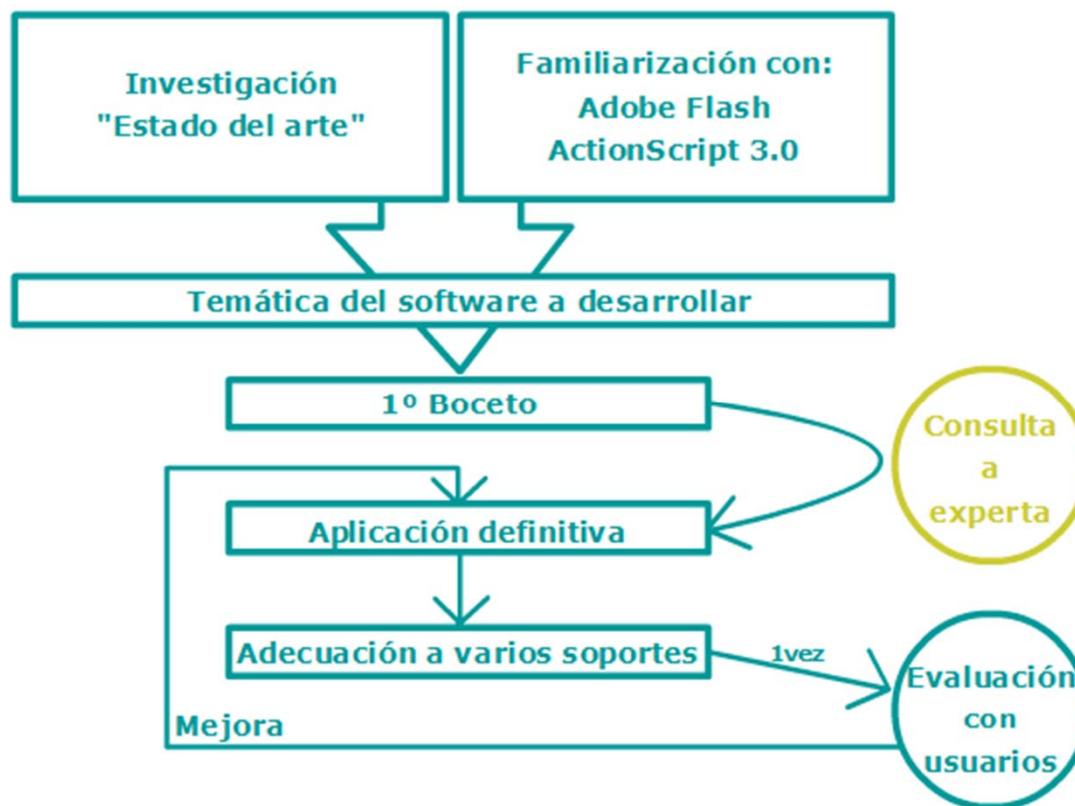
- Acciones de botón y funciones
- Sencillo

```
53 //////////////// ESCRIBIR FRASE Y ELIMINAR FICHAS ////////////////
54
55 btn_gorro.addEventListener(MouseEvent.CLICK, funcion1);
56 btn_gafas.addEventListener(MouseEvent.CLICK, funcion2);
57 btn_ojosVerdes.addEventListener(MouseEvent.CLICK, funcion3);
58 btn_ojosNegros.addEventListener(MouseEvent.CLICK, funcion4);
59 btn_chico.addEventListener(MouseEvent.CLICK, funcion5);
60 btn_chica.addEventListener(MouseEvent.CLICK, funcion6);
61
62 btn_tiene.addEventListener(MouseEvent.CLICK, funTiene);
63 btn_es.addEventListener(MouseEvent.CLICK, funEs);

```

```
96 function funcion1(event:MouseEvent) {
97
98     var Go:go = new go();
99     Go.x = posicionX2 + horiz;
100    Go.y = posicionY + vertical;
101    addChild(Go);
102
103    switch(gorro) {
104
105        case false:
106
107            var NoTiene:noTiene = new noTiene();
108            NoTiene.x = posicionX1 + horiz;
109            NoTiene.y = posicionY + vertical;
110            addChild(NoTiene);
111
112            FiDaniel.alpha = .15;
113            FiDiana.alpha = .15;
114            FiRaul.alpha = .15;
115            FiRocio.alpha = .15;
116            FiSusana.alpha = .15;
117            FiTeo.alpha = .15;
118            FiTeresa.alpha = .15;
119            FiVictor.alpha = .15;
120            break;
121
122        case true:
123
124            FiAbel.alpha = .15;
125            FiAlba.alpha = .15;
126            FiSergio.alpha = .15;
127            FiVeronica.alpha = .15;
128            break;
129    }
}
```

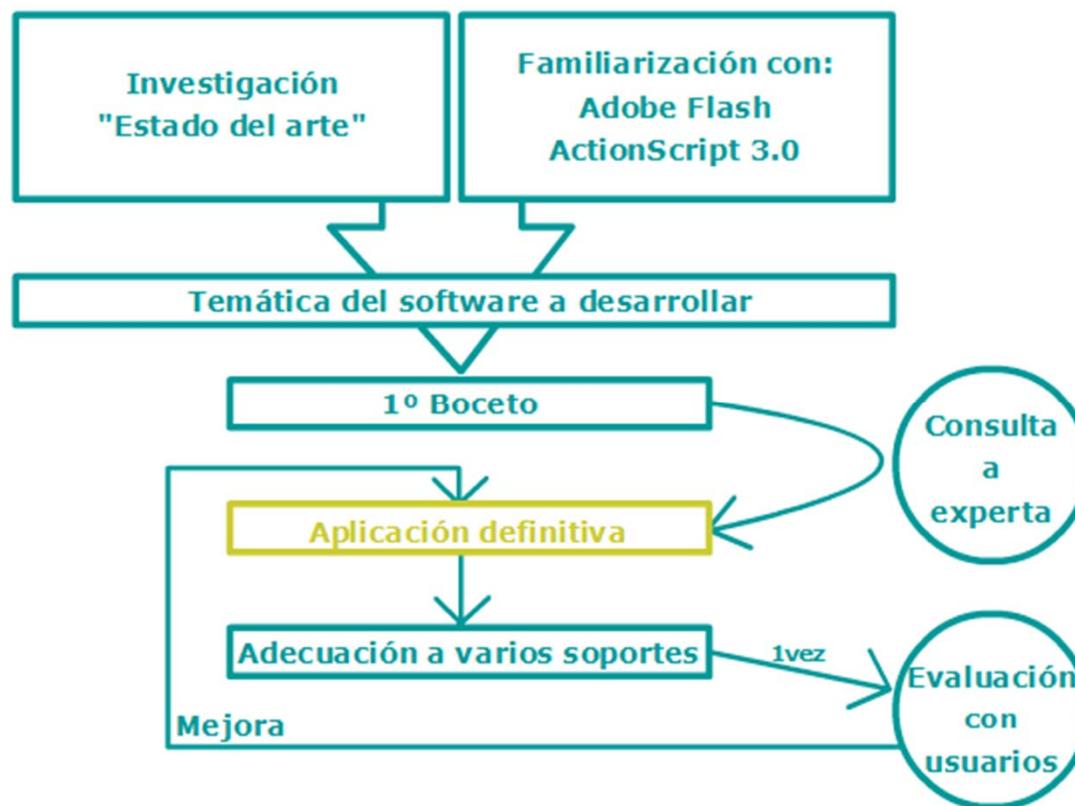
# Consulta a la experta



# Consulta a la experta

- ✓ temática elegida
- ✓ diseño
- ✓ uso de pictogramas con las palabras.
- Botones ordenados
- Separar pregunta de respuesta
- Eliminar las fichas manualmente
- Nivel de juego cero.

# Aplicación "definitiva"



# Aplicación

- Botones ordenados



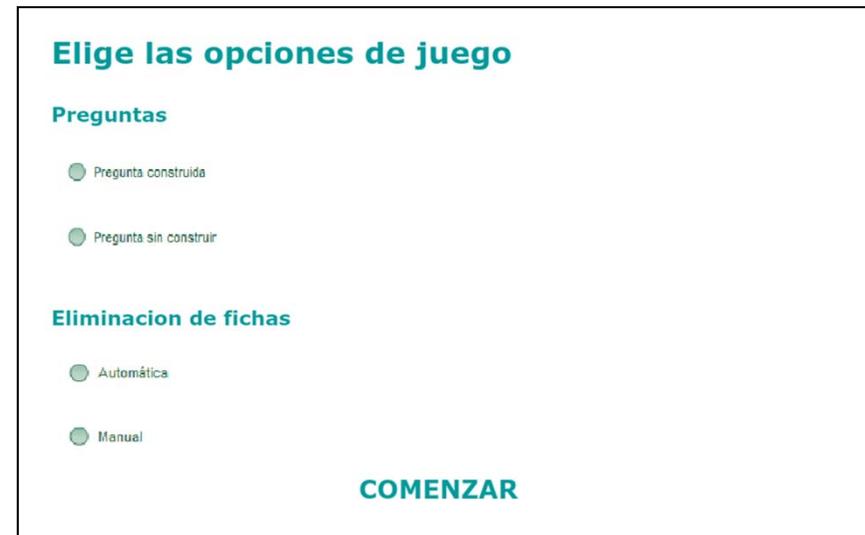
- Mejora del código del boceto
  - 1º parte del código → clase “cargarFichas”
  - 2º parte del código → clase “ElegirPersonaje”

```
2 import clases.*;
3
4 ///////////////////////////////////////////////////CARGAR FICHAS ////////////////////////////////////////
5
6 var Fichas:cargarFichas = new cargarFichas();
7 addChild(Fichas);
8
9
10 ///////////////////////////////////////////////////ELEGIR PERSONAJE Y SUS PROPIEDADES ////////////////////////////////////////
11
12 var propiedades:elegirPersonaje = new elegirPersonaje ();
13
```

- 3º parte del código → borrar

# Aplicación- Modo individual

- Pantalla inicial
  - Niveles de juego
  - Interfaz: radioButtons
  - Código: aparte del resto  
Sencillo

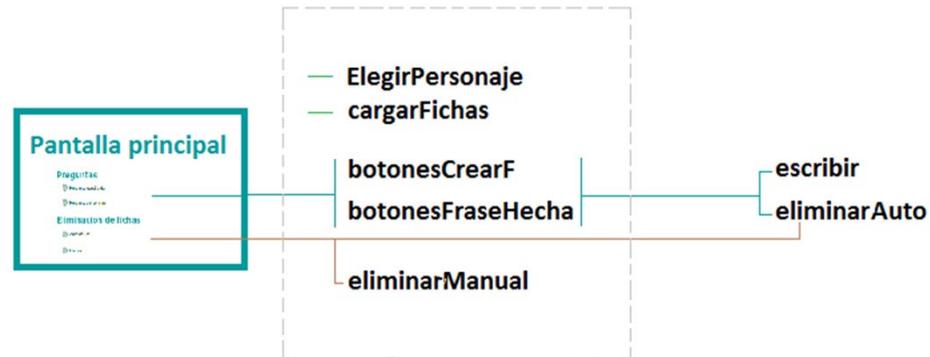


- Botones pregunta construida



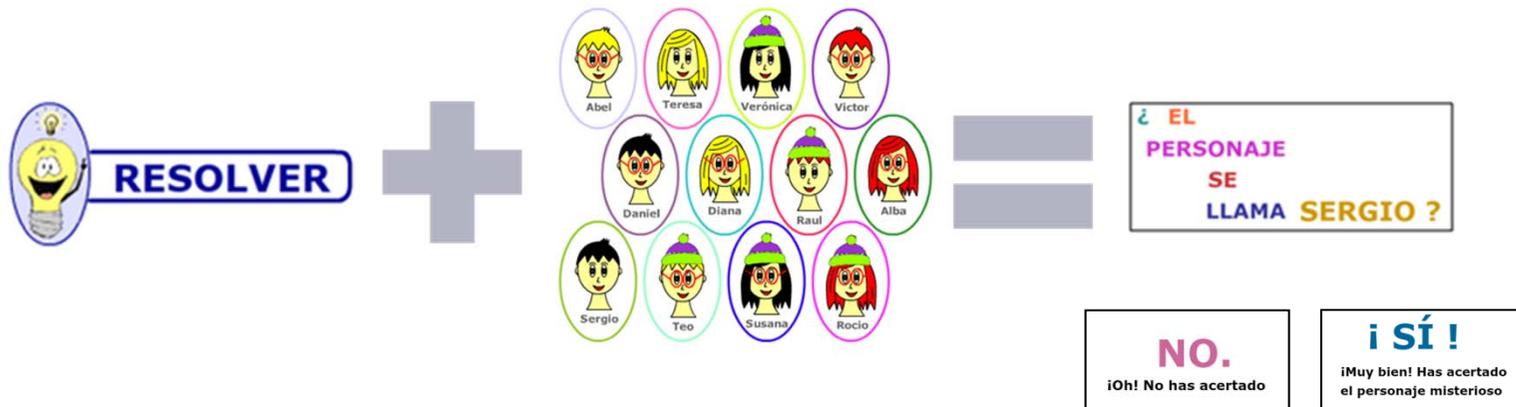
# Aplicación- Modo individual

- Clases



- Final juego

- Clase "solución"



# Aplicación- Modo individual

- Pantalla final



- Botón volver



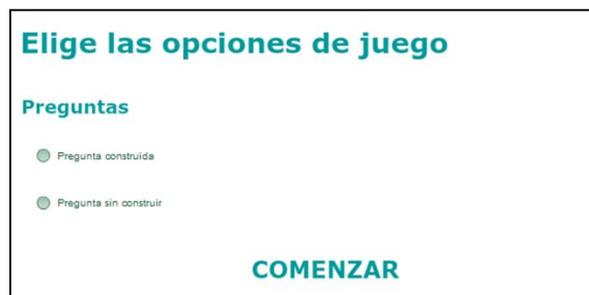
- Vidas 

**¡ Ups!**  
**Se han acabado las vidas**  
**antes de adivinar al personaje**  
**¡Vuelve a jugar!**

# Aplicación- Modo pareja

- Copia del modo individual con modificaciones

## 1. Pantalla principal



## 2. Elección de personaje

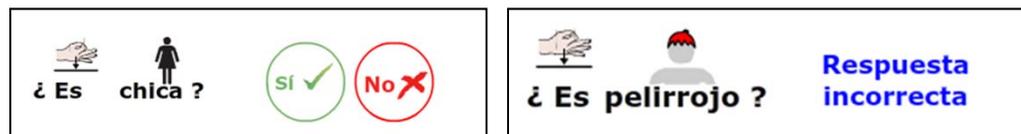


# Aplicación- Modo pareja

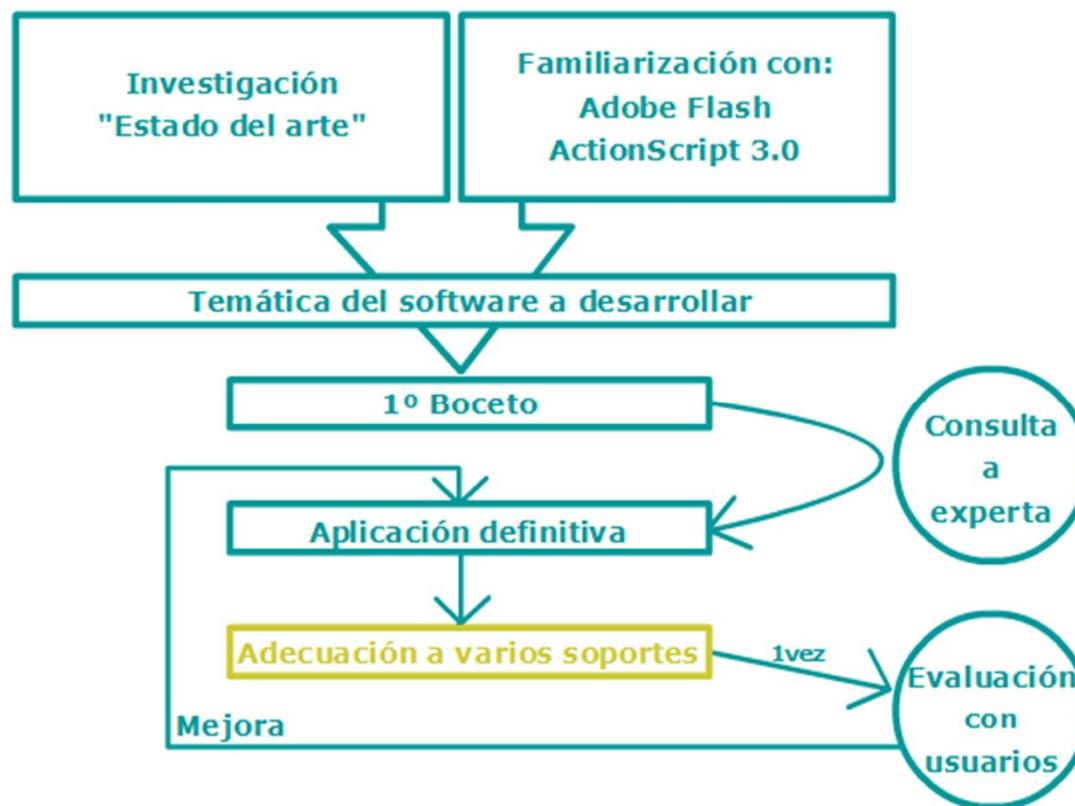
## 3. Pantallas y bocadillos de ayuda



## 4. Responder a las preguntas (Clase "escribir")



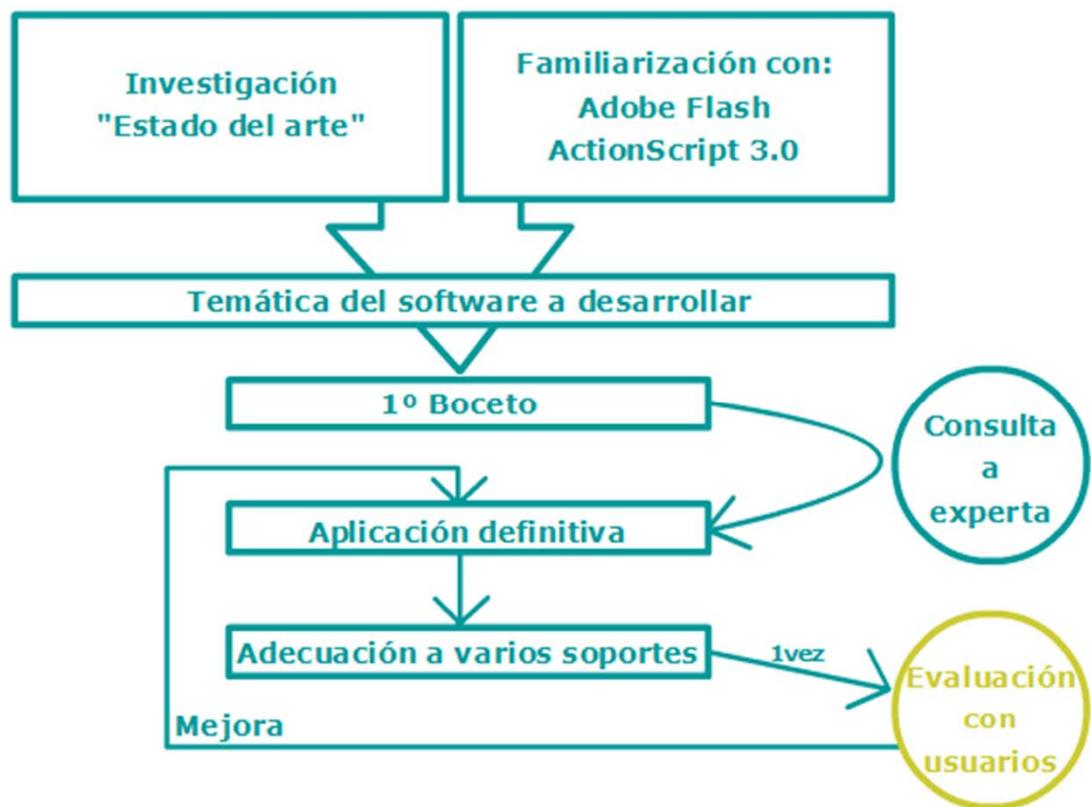
# Adecuación a varios soportes



# Adecuación a varios soportes

- PC
  - No necesario ningún cambio
- Tableta
  - Creación de un archivo .apk
  - Instalación del archivo .apk en la tableta

# Evaluación con usuarios



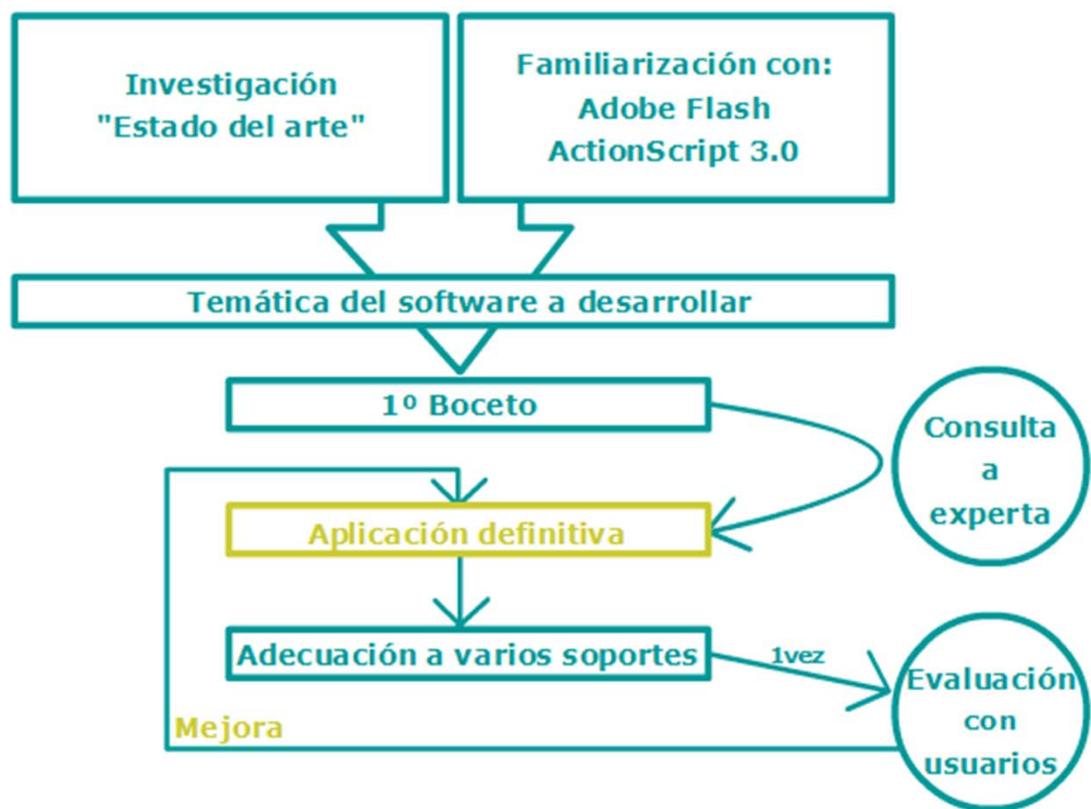
# Evaluación con los usuarios

- Lugar: ANFAS
- Objetivo: localizar aspectos del juego que se puedan mejorar
  
- Tableta
- Modo individual
  
- 4 usuarios
  - **Usuario 1:**
    - Apenas sabe leer.
    - No entendió la dinámica del juego.
  - **Usuario 2:**
    - Jugó solo correctamente.
    - Posee problema de vista → problema al leer las palabras.
  - **Usuario 3:**
    - Entendió dinámica de juego.
    - Necesitaba ayuda para preguntar y eliminar fichas.
  - **Usuario 4:**
    - Ninguna dificultad para jugar.

# Evaluación con los usuarios

- Cosas a cambiar
  - Aumentar tamaño palabras e iconos
  - Aumentar tamaño de botones
  - Añadir sonidos
  - Poder visualizar preguntas y respuestas para comprobar dónde se ha fallado
  - Añadir introducción

# Mejora y aplicación definitiva



# Aplicación definitiva

- Añadir animación inicial



- Cambiar resto de frases para que tenga concordancia



# Aplicación definitiva

- Más grandes botones, palabras e iconos

¿Quién tiene la moneda?

¿ Es chica ? No  No

¿ Es pelirrojo ? No  No

¿ Lleva gafas ? Si  Si

¿ Es moreno ? No  No

¿ Lleva gorro ? No  No

¿ Es pelirrojo ? No  No

¿ Lleva gafas ? Frase mal construida  No

¿ Lleva gorro ? No  No

¿Quién tiene la moneda?

¿ Es chica ? Si  Si

¿ Es rubio ? No  No

¿ Lleva gafas ? Si  Si

¿ Es pelirrojo ? No  No

¿ Lleva gorro ? Si  Si

¿ Es moreno ? Si  Si

¿Quién tiene la moneda?

¿ Es chica ? Si  Si

¿ Lleva gafas ? No  No

¿ Lleva moreno ? Frase mal construida  No

¿ Lleva gorro ? Respuesta incorrecta  No

¿ Es rubio ? Si  Si

¿ Lleva gorro ? Si  Si

¿Quién tiene la moneda?

¿ Es chica ? Si  Si

¿ Es chico ? Respuesta incorrecta  No

¿ Es moreno ? No  No

¿ Es rubio ? Si  Si

¿ Lleva gafas ? No  No

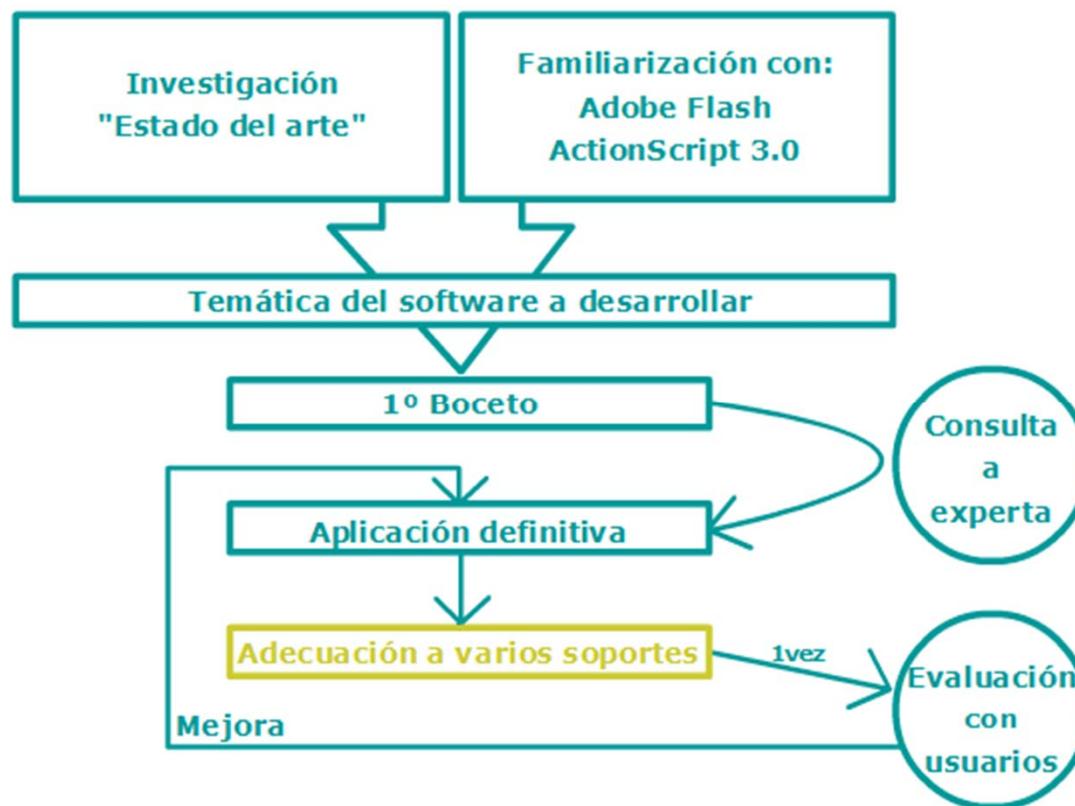
¿ Lleva gorro ? Si  Si

# Aplicación definitiva

- Opción de ver preguntas – respuestas
  - Modificación clase solución



# Adecuación a varios soportes



# Conclusiones

- Se han cumplido todos los objetivos especificados al inicio.
- Se han reforzado y aumentado los conocimientos sobre el software de desarrollo Adobe Flash Professional.
- Se han adquirido conocimientos sobre un nuevo lenguaje de programación (ActionScript 3.0).
- Se ha diseñado e implementado una aplicación multimedia interactiva que ayuda a la enseñanza, individual y cooperativa, de personas con trastorno general de la comunicación.
- El proyecto ha sido instalado con éxito en dos soportes informáticos, PC y tableta.
- Se ha evaluado el software desarrollado con personas de características similares a los destinatarios del producto.

# Líneas futuras

- Implementación en una mesa multitouch
  - Se pensó en hacer al principio del proyecto
  - Unir Flash con TUIO y reactivision → proyecto más complejo
- Desarrollar aplicación empleando un kit de desarrollo de android (Eclipse)
  - Crear “plantilla” para cambiar fichas de personajes por fichas de otras familias.

# “DESARROLLO DE UN SOFTWARE INTERACTIVO DE AYUDA AL APRENDIZAJE”

Arancha León Muñoz

Tutor: Alfredo Pina

Asesora externa: Isabel Sánchez

Pamplona, 10 de Septiembre del 2012