



PLAN CONJUNTO:

INGENIERÍA SUPERIOR DE TELECOMUNICACIONES
INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso Académico 2014/2015

Proyecto Fin de Carrera

ELEARNT

Autor : José Ignacio Madan Frías

Tutor : Dr. Gregorio Robles

*Dedicado a
todas las personas que estuvieron
robando el tiempo
que pertenecía a la soledad*



Esta obra está sujeta a la licencia Reconocimiento-CompartirIgual 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/4.0/>.

Agradecimientos

En especial, a mi familia y a todas las personas que he considerado como tal a lo largo de mi vida. Por supuesto, a mi tutor y a mi querido consultor de dudas, Internet, y a todos los que a través de él, respuestas han hecho posibles.

*“A la hora de la verdad,
se verá la verdad”*

Un Don Nadie

Resumen

Como su propio nombre indica Elearnt (Electronic Learning Tool) nos encontramos con una herramienta para el aprendizaje electrónico. Siendo más específico, sirve de ayuda a un proyecto realizado por la Universidad Complutense de Madrid, eAdventure, cuya finalidad es la creación de aventuras educativas.

Para los propósitos especificados, se ha trabajado con un sistema operativo actualmente bastante popular, hablamos de Android, utilizando un lenguaje de programación también bastante conocido, como es el caso de Java. Con el fin de conseguir implementar una aplicación que proporcione cierta agilidad a la hora de desarrollar las bases de una aventura educativa.

Cabe mencionar que en ningún momento pretende tener un carácter sustitutivo, al contrario, pretende proporcionar soporte al proyecto inicial, siendo posible su utilización como se estime oportuno.

Summary

As its name suggests Elearnt (Electronic Learning Tool) we are found with a tool for e-learning. Being more specific, helps a project carried out by the Universidad Complutense de Madrid, eAdventure, whose purpose it is the creation of educational adventures.

For specified purposes, it has worked with an operating system currently quite people, we are talking about Android, using a programming language also quite known, as in the case of Java. In order to implement an application that provides some agility in developing the foundations of an educational adventure.

It is worth mentioning that at no time he intends to have a replacement character, on the contrary, aims to provide support to the initial project, it is possible to use as is deems appropriate.

Índice general

1. Introducción	1
1.1. ¿Qué es una aventura gráfica?	1
1.2. ¿Qué significa e-learning?	2
1.3. Proyectos similares	2
1.4. Tecnologías implicadas	3
1.4.1. Java [5]	4
1.4.2. Android [7]	5
1.4.3. Eadventure [4]	6
1.4.4. Eclipse IDE (Integrated Development Environment)	7
1.5. Estructura de la memoria	8
2. Objetivos	11
2.1. Descripción del problema	11
2.1.1. Comparación eLearnT-eAdventureEditor	11
2.2. Requisitos	14
2.3. Modelo de desarrollo	14
3. Diseño	17
3.1. Funcionamiento general	17
3.2. Funcionalidades	19
3.2.1. Aventura	19
3.2.1.1. Crear	20
3.2.1.2. Crear de Nuevo	20
3.2.1.3. Modificar	21

3.2.1.4.	Eliminar	22
3.2.1.5.	Enviar	23
3.2.1.6.	Finalizar	24
3.2.2.	Escena	24
3.2.2.1.	Obtener	25
3.2.2.2.	Modificar	25
3.2.2.3.	Eliminar	26
3.2.3.	Seleccionar Escena en la Galería	27
3.2.4.	Seleccionar Área de Salida	28
3.2.5.	Crear Salida	28
3.2.6.	Corregir Enlaces	29
3.2.7.	Enviar Ficheros	30
3.2.8.	Aclaraciones	31
3.3.	Interfaz gráfica	32
3.3.1.	Servidor	32
3.3.2.	Aplicación	32
4.	Implementación	33
4.1.	Estructura de datos	33
4.1.1.	Listas enlazadas	34
4.2.	Selección de área	35
4.3.	Interacción entre actividades	36
4.4.	Ficheros	39
4.4.1.	Guardando/Cargando Imágenes	40
4.4.2.	Generando descriptores xml	42
4.4.3.	Envío	45
4.5.	Interfaz gráfica	45
4.5.1.	Servidor	45
4.5.2.	Aplicación	46
5.	Publicación	49
5.1.	Tipo de licencia	49

<i>ÍNDICE GENERAL</i>	XI
6. Conclusiones	51
6.1. Lecciones aprendidas	51
6.2. Conocimientos aplicados	52
6.3. Trabajos futuros	52
A. Instalación	53
A.1. Requisitos	54
B. Manual de usuario	55
B.1. Comenzando	55
B.2. Crear Aventura	56
B.3. Crear Aventura de Nuevo	57
B.4. Modificar Aventura	58
B.5. Eliminar Aventura	58
B.6. Enviar Aventura	59
B.7. Obtener Escena	60
B.8. Modificar Escena	61
B.9. Eliminar Escena	63
B.10. Seleccionar Escena en la Galería	64
B.11. Seleccionar Área de Salida	64
B.12. Crear Salida	65
B.13. Corregir Enlaces	66
B.14. Finalizando	69
B.15. Preparar el servidor	69
B.16. Enviar Ficheros	70
Bibliografía	73

Índice de figuras

1.1. Arquitectura Android. Autor: Smieh. Licencia: Cc-by-sa-3.0-migrated, GFDL .	5
1.2. eAdventure Engine	6
1.3. eAdventure Editor	6
1.4. Eclipse IDE	8
3.1. Funcionamiento General	18
3.2. Diagrama de casos de uso general	19
3.3. Crear Aventura	20
3.4. Crear Aventura de Nuevo	21
3.5. Modificar Aventura	22
3.6. Eliminar Aventura	23
3.7. Enviar Aventura	23
3.8. Finalizar Aventura	24
3.9. Obtener Escena	25
3.10. Modificar Escena	26
3.11. Eliminar Escena	27
3.12. Seleccionar Escena en la Galería	27
3.13. Seleccionar Área de Salida	28
3.14. Crear Salida	29
3.15. Corregir Enlaces	30
3.16. Enviar Ficheros	31
4.1. Listas enlazadas	34
4.2. Diagrama de flujo surfaceView	35

4.3. Ciclo de vida de las actividades [12]	37
4.4. descriptor.xml	42
4.5. project.xml	43
4.6. Relación entre escenas (eAdventureEditor)	43
4.7. chapter1.xml	44
4.8. Utilizando Java Swing	45
4.9. Resoluciones soportadas	46
4.10. Gestión de diferentes resoluciones	46
B.1. Menú principal.	55
B.2. Crear aventura.	56
B.3. Seleccionar aventura existente.	57
B.4. Crear aventura de nuevo.	57
B.5. Modificar aventura.	58
B.6. Eliminar aventura.	59
B.7. Enviar aventura.	59
B.8. Obtener escena.	60
B.9. Hacer fotografía.	60
B.10. Seleccionar imagen.	61
B.11. Modificar escena.	62
B.12. Modificar escena (no existen escenas).	62
B.13. Eliminar escena.	63
B.14. Eliminar escena (no existen escenas).	63
B.15. Seleccionar escena en la galería.	64
B.16. Seleccionar área de salida.	65
B.17. Crear salida.	66
B.18. Crear salida (no existen escenas).	66
B.19. Corregir enlaces.	67
B.20. Corregir enlaces (no existen escenas).	67
B.21. Corregir salida.	68
B.22. Corregir salida (sin escenas enlazables).	68

B.23. Preparar el servidor.	69
B.24. Enviar ficheros.	71

Capítulo 1

Introducción

En este capítulo se introduce el proyecto.

En un primer apartado se realiza una breve introducción a las aventuras gráficas, explicando sus conceptos básicos. Posteriormente, nos centraremos en explicar el concepto de aprendizaje electrónico y los proyectos relacionados.

Por último, daremos a conocer las principales características de las tecnologías implicadas y la estructura de la memoria.

1.1. ¿Qué es una aventura gráfica?

Para conocer este concepto, vayámonos a su principal antecesor, hablamos de las aventuras conversacionales [1], que no es más que un género de los videojuegos, principalmente de ordenador, basado en la interacción con el jugador por medio de texto.

Teniendo clara esta primera idea, podemos ver las aventuras gráficas [2] como una evolución de dicho género, permitiendo al jugador, además, interactuar con personajes, objetos, rompecabezas e incluso ofrecer un personaje que controlar (3ª persona). Como ejemplos más ilustres existen bastantes aventuras pertenecientes a la época dorada del género, 90's (Maniac Mansion, Grim Fandango, Monkey Island, Simon the Sorcerer, Broken Sword, Hollywood Monsters, Day of the Tentacle, Indiana Jones and the Fate of Atlantis, ...).

1.2. ¿Qué significa e-learning?

Podemos definir el aprendizaje electrónico [3] (e-learning) como una alternativa a los sistemas educativos actuales, en la que, a diferencia de otros, no hace falta un estado presencial. En caso de tener un estado semipresencial estaríamos hablando de b-learning (blended learning) una combinación de los dos sistemas, tradicional con el aprendizaje electrónico.

Entre los recursos que ofrece para dar una educación a distancia nos encontramos con el concepto de web 2.0 o web social, que permite la interacción entre usuarios a la hora de compartir información. Además se hace uso de otras tecnologías como pueden ser: correo electrónico, páginas web, foros de discusión, mensajería instantánea, etc. Todas ellas relacionadas con la web 2.0 propiamente dicha.

Las principales ventajas que presenta son las siguientes:

- Desarrollo de habilidades en el uso de las nuevas tecnologías
- Se adquiere capacidad autodidacta
- Actualización de contenidos
- Eliminación de barreras espaciales
- Confrontación de conocimientos
- Gestión real del conocimiento

Frente a:

- Se requiere un conocimiento mínimo de las tecnologías implicadas
- Como toda tecnología nueva, requiere un tiempo de adaptación
- Empobrece la calidad educativa si se prescinde demasiado de la figura del profesor

1.3. Proyectos similares

Algunos de los proyectos relacionados con el aquí expuesto, entendiéndolo como herramienta para el aprendizaje electrónico [3]:

- WebCT
- TopClass
- Learning Space
- Virtual-U
- DigitalThink

Más en particular, el principal proyecto con el que tiene relación, entendiéndolo casi como una extensión o complemento de él, ya mencionado anteriormente, y del que se darán más detalles en la sección Tecnologías implicadas es el eAdventure. Definido como, cito textualmente [4]:

La plataforma eAdventure es un proyecto de investigación que aspira a facilitar la integración de juegos educativos y simulaciones basadas en juegos en procesos educativos en general y Entornos Virtuales de Aprendizaje (VLE) en particular. Está siendo desarrollado bajo los auspicios del grupo e-UCM en la Universidad Complutense de Madrid, con tres objetivos principales:

- *Reducción de los costes de desarrollo para juegos educativos.*
- *Incorporación de características educativas específicas en herramientas de desarrollo de juegos.*
- *Integración de los juegos resultantes con material educativo en el contexto de los Entornos Virtuales de Aprendizaje.*

Es más, actualmente se encuentran trabajando en un proyecto que cumple la funcionalidad básica de eLearnT además de otras opciones (insertar objetos, diálogos, personajes, ...) siendo en este caso una versión de eAdventure para Android. (<http://www.youtube.com/watch?v=A5Dt0cwe8kc>)

1.4. Tecnologías implicadas

A continuación pasaremos a mencionar las principales características de las tecnologías relacionadas con el proyecto a documentar. Téngase en cuenta que el principal lenguaje de pro-

gramación ligado al sistema operativo Android es Java y el proyecto padre es eAdventure, por lo tanto el margen de elección ha sido pequeño: entorno de desarrollo y creación del servidor.

Por comodidad (hacer uso de un mismo lenguaje de programación) y experiencia (entorno de desarrollo con el que estoy más familiarizado) se ha optado por realizar el servidor con Java y todo el proyecto con eclipse IDE.

1.4.1. Java [5]

Lenguaje de programación basado en c y c++, de más alto nivel, de propósito general, concurrente, orientado a objetos y basado en clases. De tipado fuerte (distingue entre mayúsculas y minúsculas). Requiere el uso de una máquina virtual que adecue el código compilado por las aplicaciones *bytecode* al código máquina asociado a la arquitectura del terminal en cuestión, es decir, nos encontramos con un lenguaje multiplataforma.

Las principales APIs (Application Program Interface) o conjunto de paquetes formado por clases, interfaces y excepciones relacionadas de las que dispone:

- Java ME (Micro Edition) para entornos con recursos limitados: móviles, PDAs (Personal Digital Assistant), etc.
- Java SE (Standard Edition) para gamas medias: PCs (Personal Computer).
- Java EE (Enterprise Edition) para entornos empresariales o Internet.

En el caso del servidor se hará uso de Java SE por tratarse de un entorno de red pequeño desarrollado sobre un PC. Podría pensarse en el uso de Java ME para la aplicación eLearnT, sin embargo, a pesar de que las aplicaciones de Android estén programadas en Java, poseen su propia máquina virtual Dalvik, que ejecuta un *bytecode* generado por una API específica de Android [6]. Como curiosidad:

Es frecuentada por algunos de los miembros del equipo. Pero no está claro si es un acrónimo o no, aunque algunas fuentes señalan que podría tratarse de las iniciales de sus creadores: James Gosling, Arthur Van Hoff, y Andy Bechtolsheim. Otros abogan por el siguiente acrónimo, Just Another Vague Acronym ("sólo otro acrónimo ambiguo más"). La hipótesis que más fuerza tiene es la de que Java debe su nombre a un tipo de café disponible en la cafetería cercana, de ahí que el icono de java sea una taza de café caliente. Un pequeño signo que da fuerza a

esta teoría es que los 4 primeros bytes (el número mágico) de los archivos.class que genera el compilador, son en hexadecimal, 0xCAFEBABE. A pesar de todas estas teorías, el nombre fue sacado al parecer de una lista aleatoria de palabras.

1.4.2. Android [7]

Sistema operativo basado en el kernel de Linux diseñado para una gran variedad de dispositivos (relojes, televisores, automóviles, tabletas...) inicialmente Smartphones, libre, gratuito y multiplataforma. Posee su propia máquina virtual, como se menciona en la subsección anterior. Además ofrece a los desarrolladores todo el conjunto de herramientas y documentación necesarias para la creación de aplicaciones <http://developer.android.com/intl/es/index.html>

De esta forma consigue ser un sistema operativo atractivo para todos los públicos, tanto para el usuario medio, como para aquel con conocimientos informáticos, ya que ofrece una gran variedad de aplicaciones, la mayoría accesibles desde Google play <http://play.google.com/store>

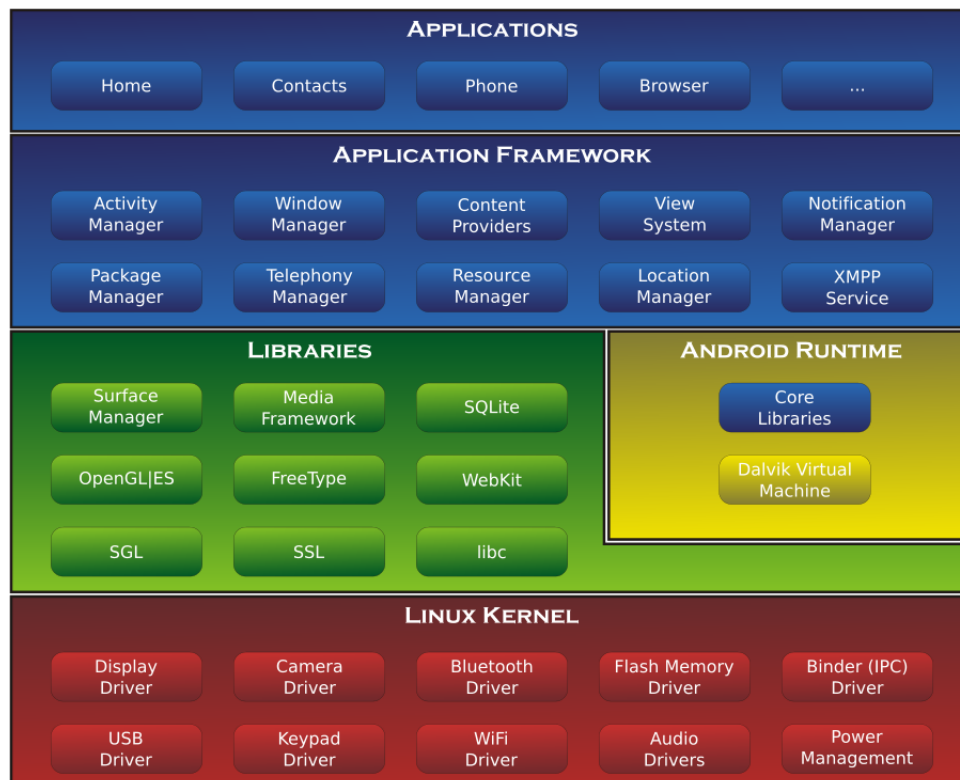


Figura 1.1: Arquitectura Android. Autor: Smieh. Licencia: Cc-by-sa-3.0-migrated, GFDL

1.4.3. Eadventure [4]

Proyecto ya definido anteriormente en la subsección **Proyectos similares**. Entrando en más profundidad, está formado por un motor gráfico destinado a la ejecución de las aventuras educativas creadas mediante un editor.

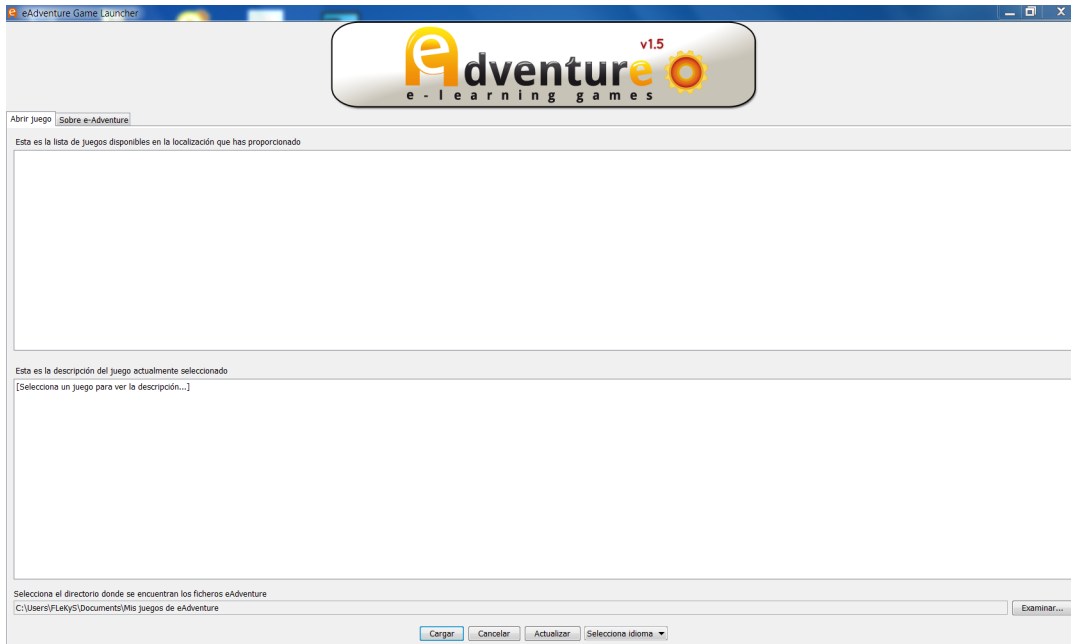


Figura 1.2: eAdventure Engine

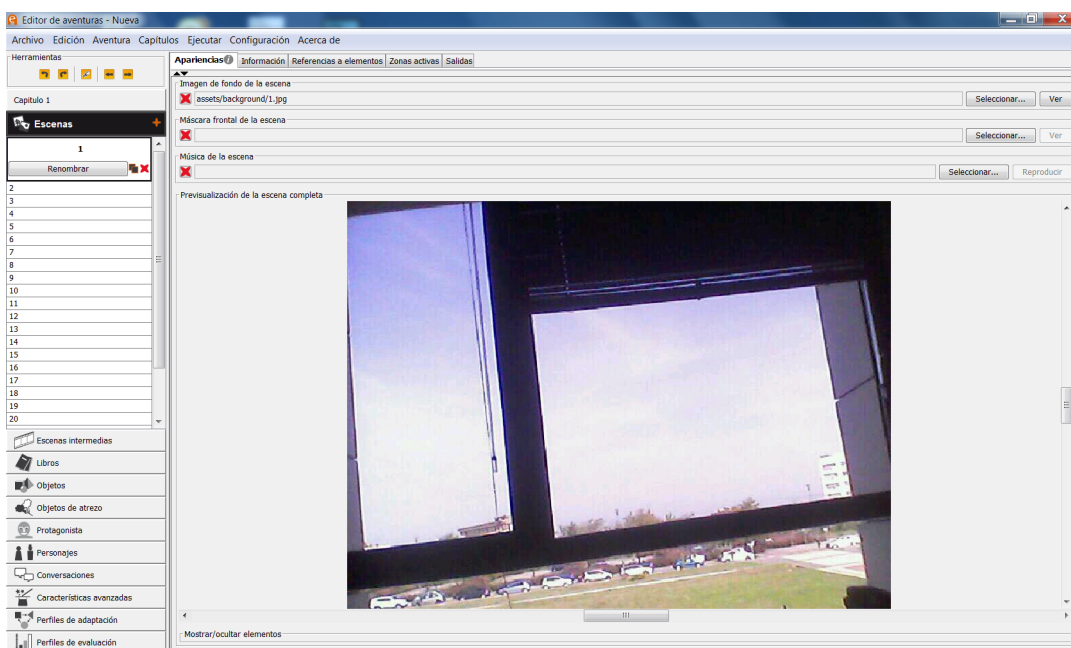


Figura 1.3: eAdventure Editor

Como puede apreciarse en la figura 1.3, el editor cuenta con una gran funcionalidad (crear escenas, salidas, objetos, protagonistas, personajes, conversaciones...). La configuración de la aventura será guardada en lenguaje de marcado XML, es decir, modificando dichos archivos se pueden alterar sus propiedades: *Los autores pueden usar el editor gráfico para crear juegos, o acceder directamente a los documentos fuente que describen las aventuras utilizando un lenguaje de marcado XML.*

Además ofrecen la posibilidad de exportar las aventuras en todo tipo de formatos, por ejemplo, .jar o ejecutables de java, objetos de aprendizaje (IMS CP, WebCT 4 CP, AGREGA, ...).

Características educativas:

- *Soporte para escenarios de aprendizaje adaptativos en tiempo real*
- *Juegos empaquetados con metadatos estándares (IEEE Learning Object Metadata, LOM-ES)*
- *Integración con LMS a través de la implementación de diversos estándares educativos (SCORM 1.2 y 2004 e IMS Content Packaging), exportación especial para la integración con WebCT 4.0 y soporte para IMS Learning Design*
- *Integración en el repositorio de contenidos educativos digitales Agrega.*

1.4.4. Eclipse IDE (Integrated Development Environment)

Entorno de desarrollo multiplataforma para una gran variedad de lenguajes de programación (Java, c/c++) e instalando los plugins adecuados (Python, Ada, Pascal, Fortran, Haskell, Latex, Java Script...). Existen plugins específicos para el desarrollo de proyectos (Java Swing, Android, etc.).

En el proyecto que nos concierne se ha utilizado el lenguaje de programación Java, además de los plugins necesarios para el desarrollo de la aplicación (Android SDK) y el servidor (WindowBuilder).

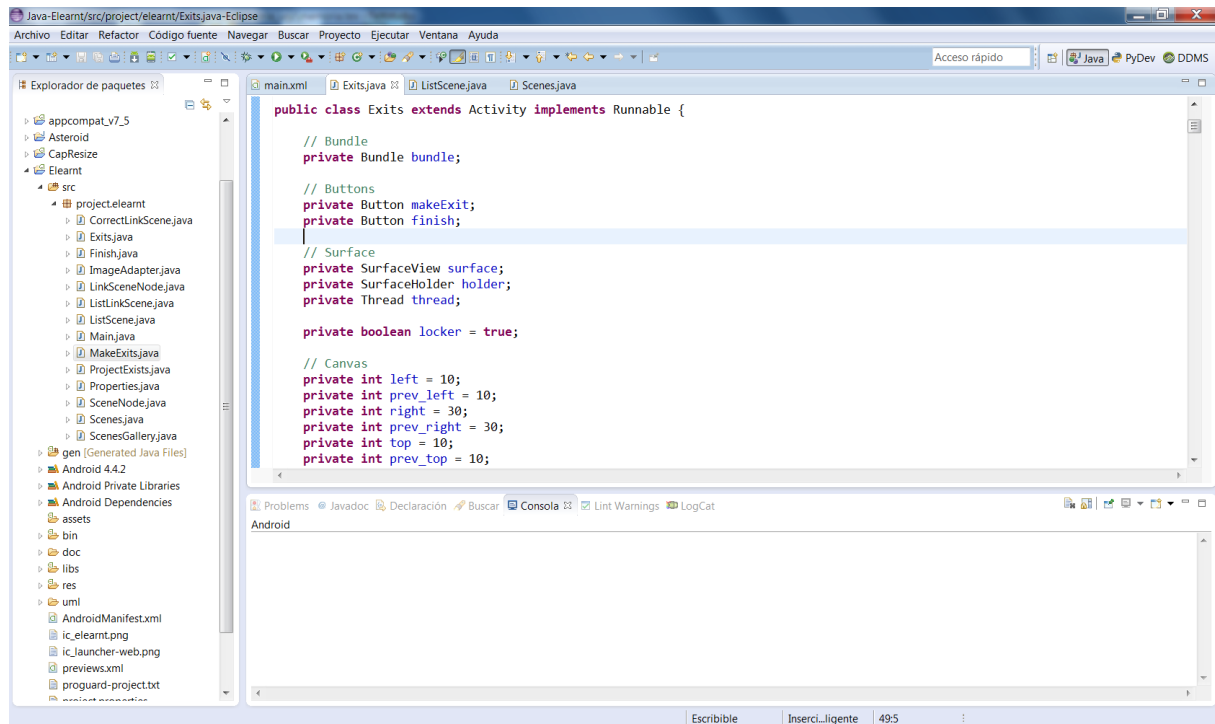


Figura 1.4: Eclipse IDE

Cuenta con un editor de texto, generación automática de código, depuración, corrección de errores, distinción de partes del código mediante colores, etc. Para más información visite la web oficial <http://eclipse.org/home/index.php>

1.5. Estructura de la memoria

Podemos dividir la memoria en tres grandes bloques:

- **I.** Introducción y objetivos.
- **II.** Diseño e implementación.
- **III.** Publicación y conclusiones.

En el primer bloque se hace una breve introducción al proyecto, explicando conceptos fundamentales, tecnologías implicadas y proyectos similares. Además se exponen los principales problemas a los que se enfrenta el proyecto, justificados mediante una prueba realizada, así como los requisitos necesarios. Por último, se especifica el modelo de desarrollo utilizado, justificándolo.

En el segundo bloque nos centraremos en dar respuesta a las siguientes preguntas, ¿qué se ha hecho? y ¿cómo se ha hecho? o en otras palabras, que diseño se ha seguido a la hora de realizar la aplicación y como se ha llevado a cabo.

El tercer bloque está formado por los términos bajo los cuales se hará público el proyecto, además de los conceptos aprendidos y experiencia adquirida.

Se ha añadido un apéndice para indicar al usuario los pasos a seguir a la hora de instalar A y utilizar B la aplicación y el servidor.

Capítulo 2

Objetivos

2.1. Descripción del problema

Podría verse poco útil o incluso inútil el desarrollo de este proyecto, ya que existe actualmente un editor con mucha más funcionalidad y se está trabajando en un proyecto que supone una versión avanzada de este. Sin embargo, se juega con distintos factores, el primero de ellos es la comodidad que ofrece frente al editor actual, agilizando una parte de la funcionalidad básica, la obtención de escenarios y enlazado entre ellos. Por otra parte, a pesar de que se esté trabajando en adaptar el editor para Android en paralelo, no deja de ofrecer alternativas al diseño e implementación, dando lugar a una fuente de información alternativa.

2.1.1. Comparación eLearnT-eAdventureEditor

VISITA VIRTUAL A TU HOGAR

Imagínate por un momento que desearas vender tu casa, y en vez de colgar simplemente unas imágenes en la red, te gustaría ofrecer la posibilidad de realizar un recorrido virtual por ella para hacer más atractiva su venta.

Conteste a las siguientes preguntas:

1) *Rendimiento*

- *¿Te ha requerido mucho tiempo? ¿En qué parte has dedicado más tiempo?*
- *¿Has notado algún tipo de demora en la ejecución?*

2) Usabilidad

- *¿Has podido realizar todas las acciones que has querido con comodidad?*
- *¿Hechas en falta alguna opción?*

3) Aspecto

- *¿Qué te parece el diseño?*
- *¿Te parece intuitivo su uso?*

4) Errores

- *¿Se ha producido algún error? ¿En qué parte?*

5) Mejoras

- *¿Qué te gustaría que se pudiera hacer demás?*

Especificaciones:

- *Usuario medio (conocimientos básicos de informática)*
- *Casa de dimensiones medias ($\approx 100 \text{ m}^2$)*
- *Numero de escenas (≈ 10)*

eLearnT**1) Rendimiento**

- *No, 15 minutos aproximadamente. En la parte de crear los enlaces.*
- *No.*

2) Usabilidad

- *Sí, excepto a la hora de crear los enlaces, ya que, a veces, al elegir el tamaño del cuadro para crear el enlace no responde bien.*

- *Saber los enlaces creados.*

3) Aspecto

- *Sencillo.*
- *Desde el móvil sí, desde el programa del ordenador considero que no es tan intuitivo.*

4) Errores

- *Sí. Al principio, cuando fui a realizar la primera imagen, la cámara no la reconocía.*

5) Mejoras

- *Imágenes panorámicas, así, si se quiere mostrar una casa no hace falta crear varias imágenes de una parte, por ejemplo el salón, si quiero que se vea todo.*

eAdventure

1) Rendimiento

- *No. 25 minutos. Enlazar imágenes.*
- *No.*

2) Usabilidad

- *Sí.*
- *Si una escena tiene que ser enlazada por más de una, poder indicarlo para que se creen los enlaces de varias escenas a esa misma.*

3) Aspecto

- *Sencillo.*
- *Considerablemente.*

4) Errores

- *No.*

5) Mejoras

- *Que saliera el nombre de la escena con la que enlaza en el cuadro de selección.*

2.2. Requisitos

El proyecto deberá cumplir ciertos requisitos básicos:

- Definir una interfaz fácil e intuitiva.
- Adaptar escenas a la resolución específica del eAdventure Editor (800x600)
- Cargado adecuado de imágenes de gran resolución.
- Proceso automatizado al máximo.
- Tiempo de ejecución asequible.

2.3. Modelo de desarrollo

El modelo de desarrollo que más se ajusta al proyecto es el iterativo, ya que se ha dividido en pequeños subproyectos (Capturar y ajustar resolución, Dibujar rectángulo modificable, Mostrar galería de escenas) todos ellos rediseñados y probados hasta formar parte del proyecto total (eLearnT). De esta forma se ofrecen versiones de las distintas funcionalidades del proyecto, que se prueban por separado, mejorando su calidad, encontrándose con el problema de tener versiones demasiado simples inicialmente, hasta que se consiguen ajustar adecuadamente a las especificaciones del cliente.

Entre las principales ventajas que nos encontramos cabe destacar [8]:

- *En el desarrollo de este modelo se da la retroalimentación muy temprano a los usuarios.*
- *Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.*
- *El producto es consistente y puntual en el desarrollo.*
- *Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.*
- *Se obtiene un aprendizaje en cada iteración que es aplicado en el desarrollo del producto y aumenta las experiencias para próximos proyectos.*

Pero [8]:

- *La entrega temprana de los proyectos produce la creación de sistemas demasiado simples que a veces se ven un poco monótonos a los ojos del personal que lo recibe.*
- *La mayoría de los incrementos se harán en base de las necesidades de los usuarios. Los incrementos en si ya son estipulados desde antes de la entrega del proyecto, sin embargo hay que ver cómo se maneja el producto para ver si necesita otros cambios además de los estipulados antes de la entrega del proyecto. Este problema no se ve frecuentemente ya que la mayoría de las veces los incrementos estipulados suplen satisfactoriamente al usuario.*
- *Los incrementos no deben constar de muchas líneas de código ya que la idea de los incrementos es agregar accesorios al programa principal (o funcional), para que este tenga una y mil formas de desenvolverse en su tarea; llenar los incrementos de muchas líneas de código provocaría que se perdiera la objetividad o base de lo que se trata el desarrollo incremental.*
- *Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.*
- *El trato con el cliente debe basarse en principios éticos y colaboración mutua, más que trabajar cada parte independientemente, defendiendo sólo su propio beneficio.*
- *La entrega de un programa que es parcial pero funcional puede hacer vulnerable al programa debido a la falta de robustez en su sistema, provocando que agentes ajenos puedan interferir con el correcto funcionamiento del programa en sí.*
- *Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales sobre el promedio.*
- *Sufre fuertes penalizaciones en proyectos en los cuales los requerimientos están previamente definidos, o para proyectos “todo/nada.”^{en} los cuales se requiere que se completen en un 100 % el producto para ser implementado (por ejemplo, licitaciones) otro punto muy importante es asegurarnos de que el trabajo se pueda cumplir tomando en cuenta los costos que podamos usar en nuestros propios recursos.*

Téngase en cuenta que el principal cliente del proyecto ha sido el propio desarrollador, a excepción de las pruebas realizadas en la subsección **Comparación eLearnT-eAdventureEditor**, en la que intervinieron dos clientes y cuya participación ayudó gratamente en la mejora y actualización del proyecto.

La fase de pruebas no se incluirá en la documentación por falta de tiempo para desarrollar casos suficientemente representativos.

Capítulo 3

Diseño

Se ha intentado definir un diseño que se ajuste a los requisitos y, posteriormente, a las necesidades del cliente. El diseño se ha ido redefiniendo a lo largo de todo el ciclo de vida del software, por lo que, se ha trabajado en paralelo diseño-implementación. Teniendo en cuenta esta afirmación, puede observarse como se recoge mayor complejidad en los diversos diagramas empleados.

Para tener una visión global de la aplicación se ha utilizado un diagrama de estados que recoge todas las posibles interacciones que el usuario puede realizar.

En el caso de las funcionalidades específicas, es preferible utilizar diagramas de secuencia que den más detalles del funcionamiento interno de la aplicación.

Por último, mencionar las principales características deseables en las interfaces (aplicación y servidor).

3.1. Funcionamiento general

Se ha intentado aportar la máxima funcionalidad posible (Crear/Modificar/Eliminar escenas y además Enviar en el caso de las aventuras). Cada aventura tiene asociadas una serie de características y está formada por escenas. Para facilitar la manipulación de escenas, así como la creación de enlaces, existirá una galería con todas las escenas que se vayan añadiendo a la aventura. Además contará con la opción de corregir los enlaces previamente creados para garantizar el correcto funcionamiento de la aventura.

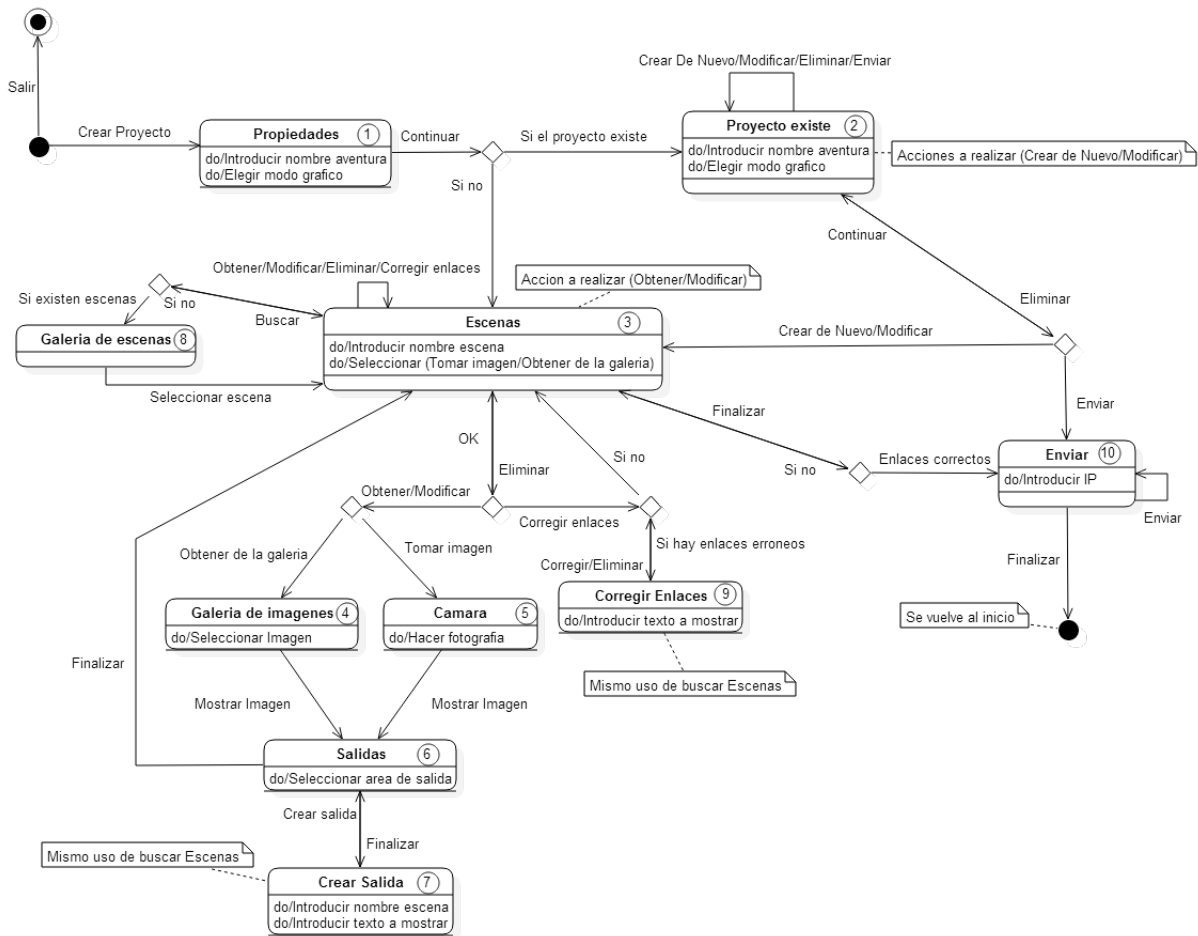


Figura 3.1: Funcionamiento General

Aclaraciones:

2)

- Una vez realizada la acción Continuar, según el tipo de acción previamente seleccionada (Crear De Nuevo/Modificar/Eliminar/Enviar) se pasará al estado correspondiente.

3)

- Una vez realizada la acción OK, según el tipo de acción previamente seleccionada (Obtener/Modificar/Eliminar/Corregir enlaces) se pasará al estado correspondiente.
- Se accederá a la Galería de escenas siempre y cuando exista al menos una.
- Si existen enlaces erróneos no se podrá acceder al estado Enviar.

9)

- Mientras existan enlaces erróneos no se volverá al estado Escenas.

3.2. Funcionalidades

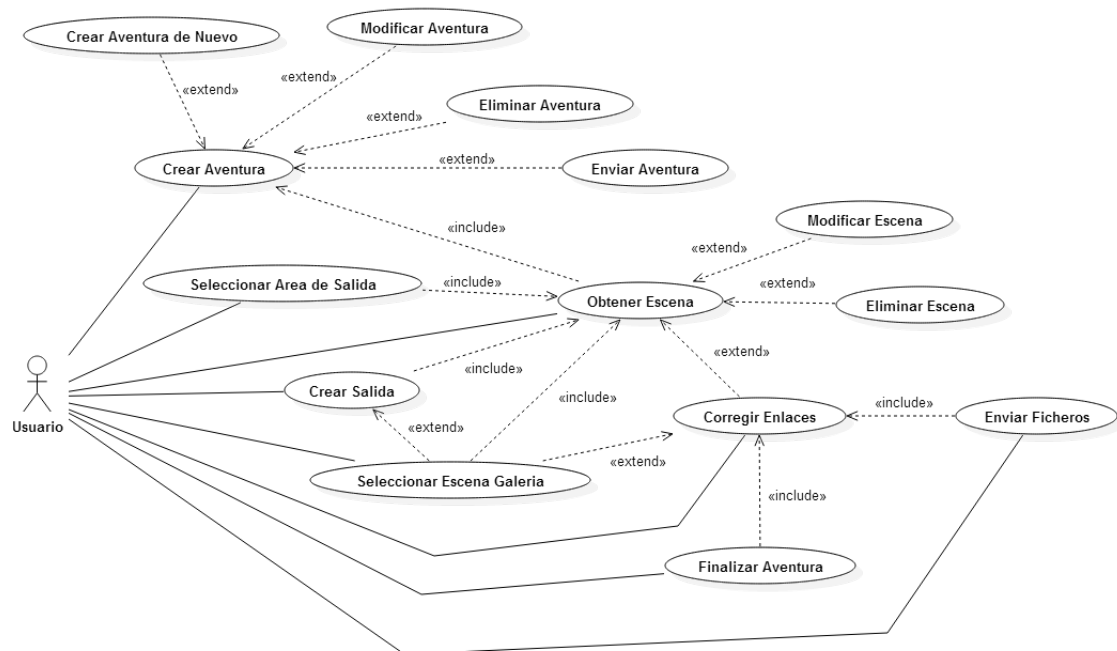


Figura 3.2: Diagrama de casos de uso general

Opcionalmente, el usuario podrá crear de nuevo, modificar, eliminar o enviar una aventura previamente creada. Si quisiera obtener una escena, necesariamente tendría que existir un proyecto creado. No se podrá enviar los ficheros contenidos en la aventura hasta que no estén todos los enlaces corregidos. Para seleccionar una escena en la galería necesariamente tiene que existir al menos una.

3.2.1. Aventura

Estructura central que alberga un conjunto de escenas, enlazadas o no entre ellas, en primera persona (no aparece el personaje en la escena, es el propio jugador) y con un par de propiedades básicas:

- Nombre
- Modo gráfico: ventana o ventana con fondo en negro, y pantalla completa (solo para Windows)

3.2.1.1. Crear

El usuario podrá crear una aventura, asignándole o no un nombre (se pone por defecto) y un modo gráfico (ventana por defecto). Se hará distinción entre la existencia o no de previas aventuras con el mismo nombre, en cuyo caso, se ofrecerá un abanico de posibilidades a realizar.

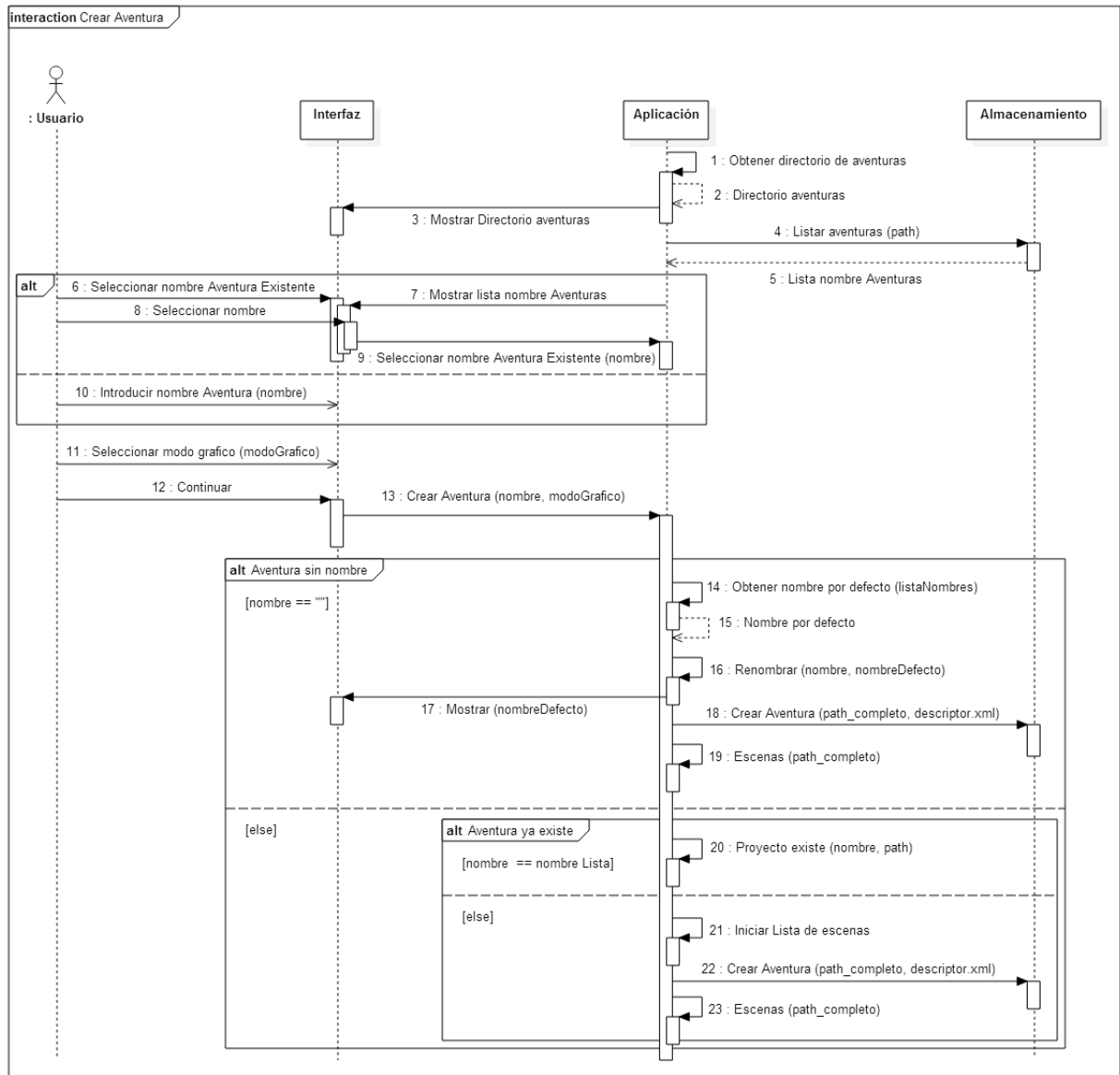


Figura 3.3: Crear Aventura

3.2.1.2. Crear de Nuevo

Si ya existe una aventura previa con el nombre que el usuario específico, y selecciona a continuación la opción crear de nuevo, se destruirá la aventura previamente creada, dando lugar

a una nueva, con el mismo nombre y modo gráfico, o los que elija el usuario.

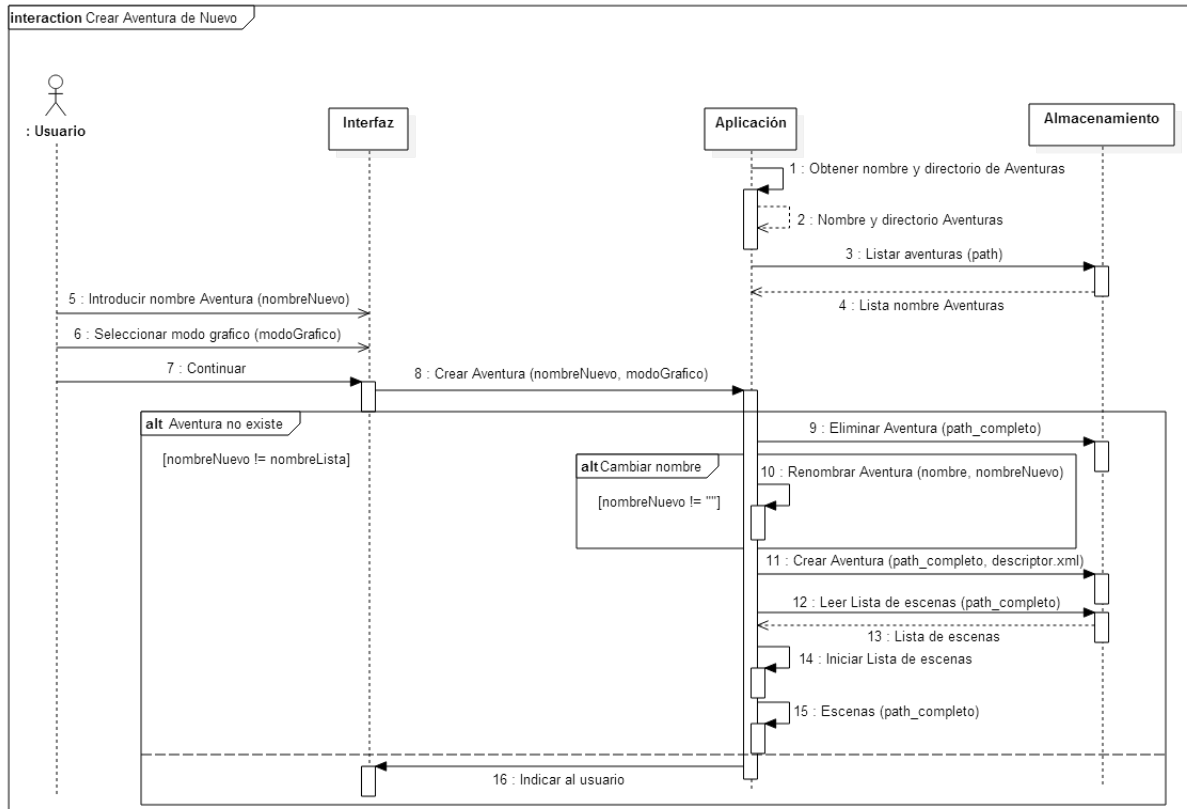


Figura 3.4: Crear Aventura de Nuevo

3.2.1.3. Modificar

En caso de detectar una aventura previamente creada, el usuario podrá cambiar sus propiedades, conservando sus escenas y enlaces anteriormente creados.

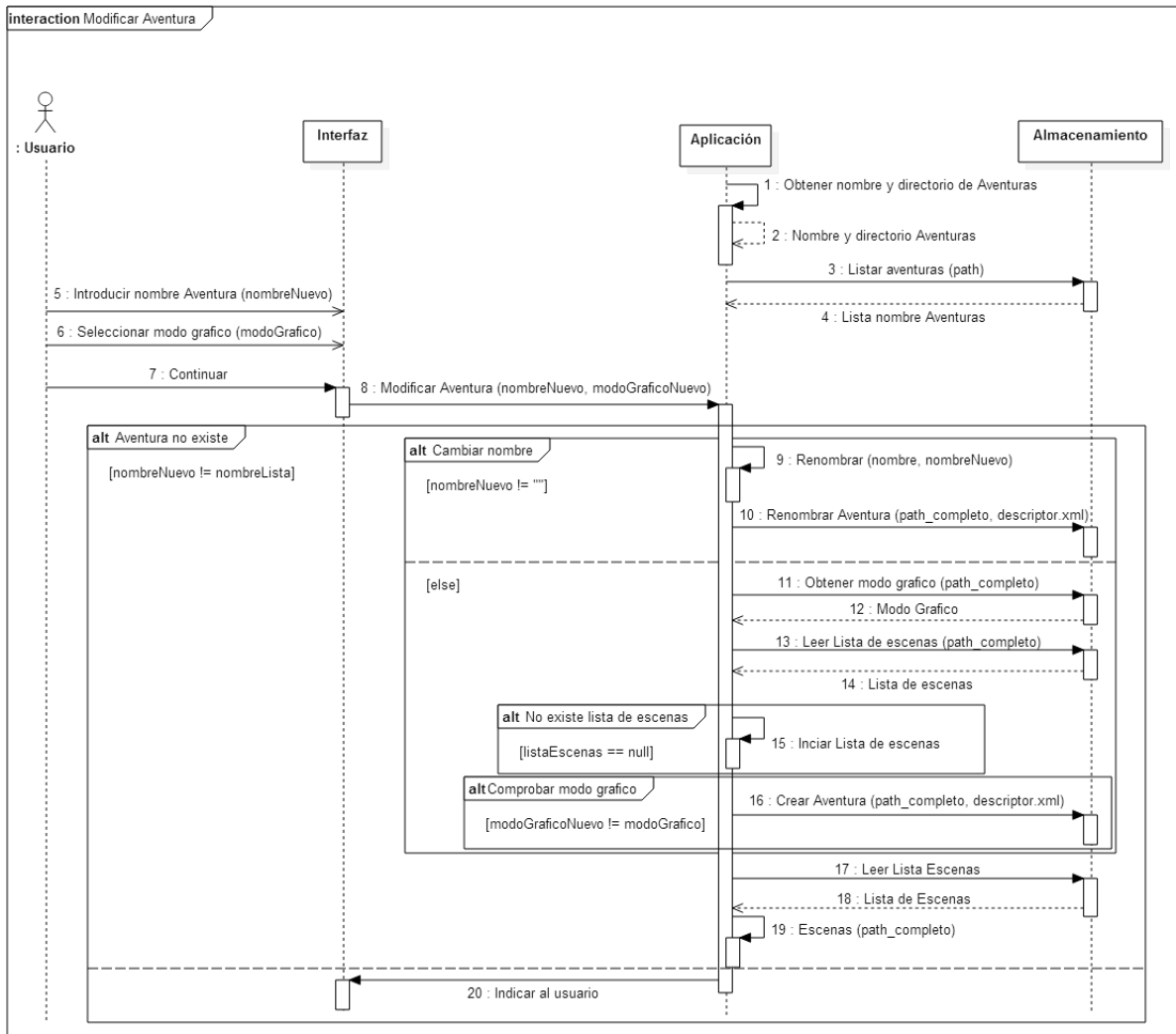


Figura 3.5: Modificar Aventura

3.2.1.4. Eliminar

Se podrá eliminar por completo una aventura previamente creada.

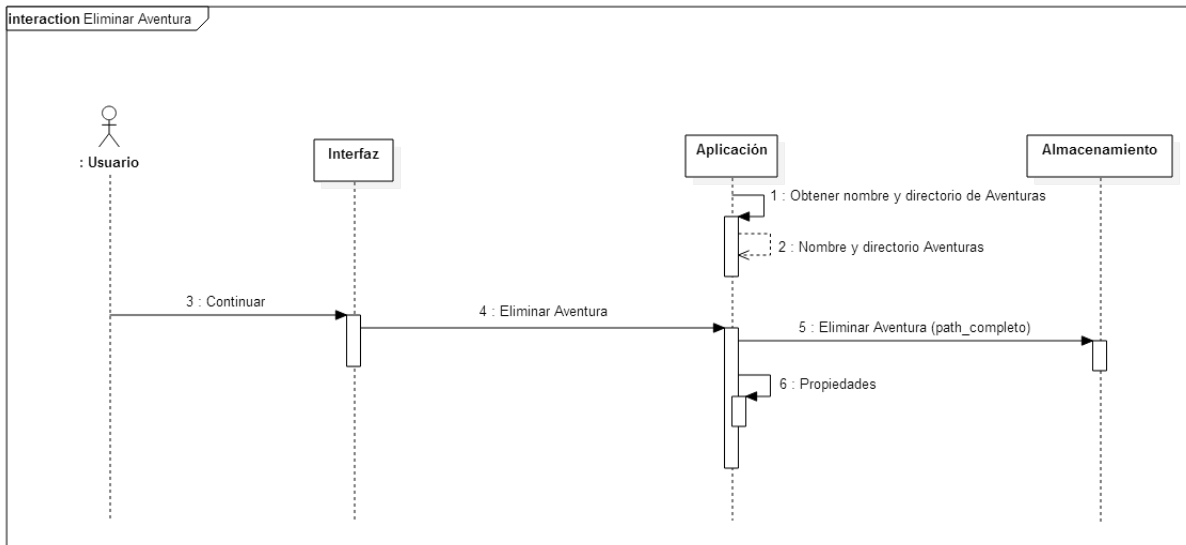


Figura 3.6: Eliminar Aventura

3.2.1.5. Enviar

En caso de no querer modificar la estructura interna de la aventura (escenas y enlaces), existe la posibilidad de enviarla directamente al servidor.

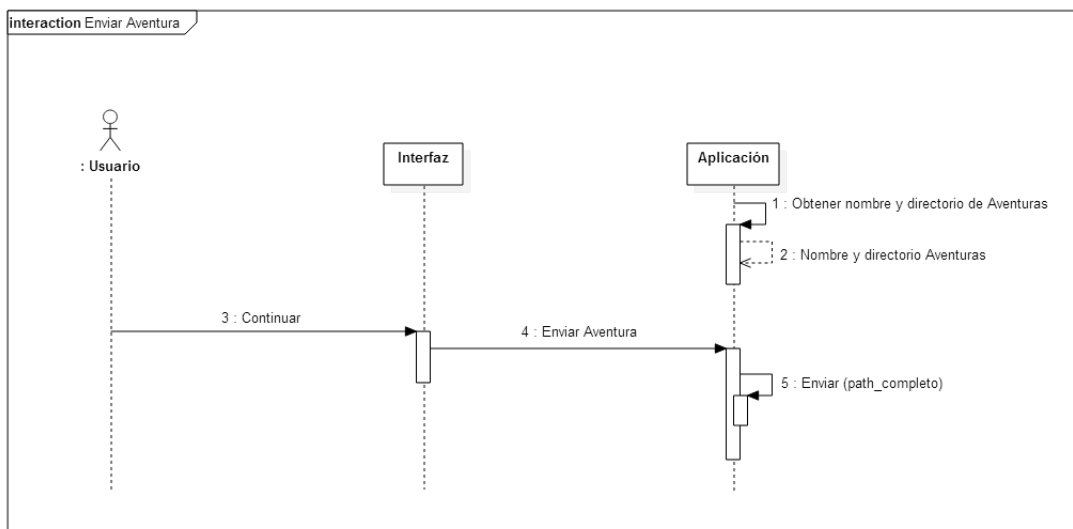


Figura 3.7: Enviar Aventura

3.2.1.6. Finalizar

Cuando haya terminado la edición de la aventura, la aplicación se asegurará de que todas las escenas y enlaces creados sean correctos.

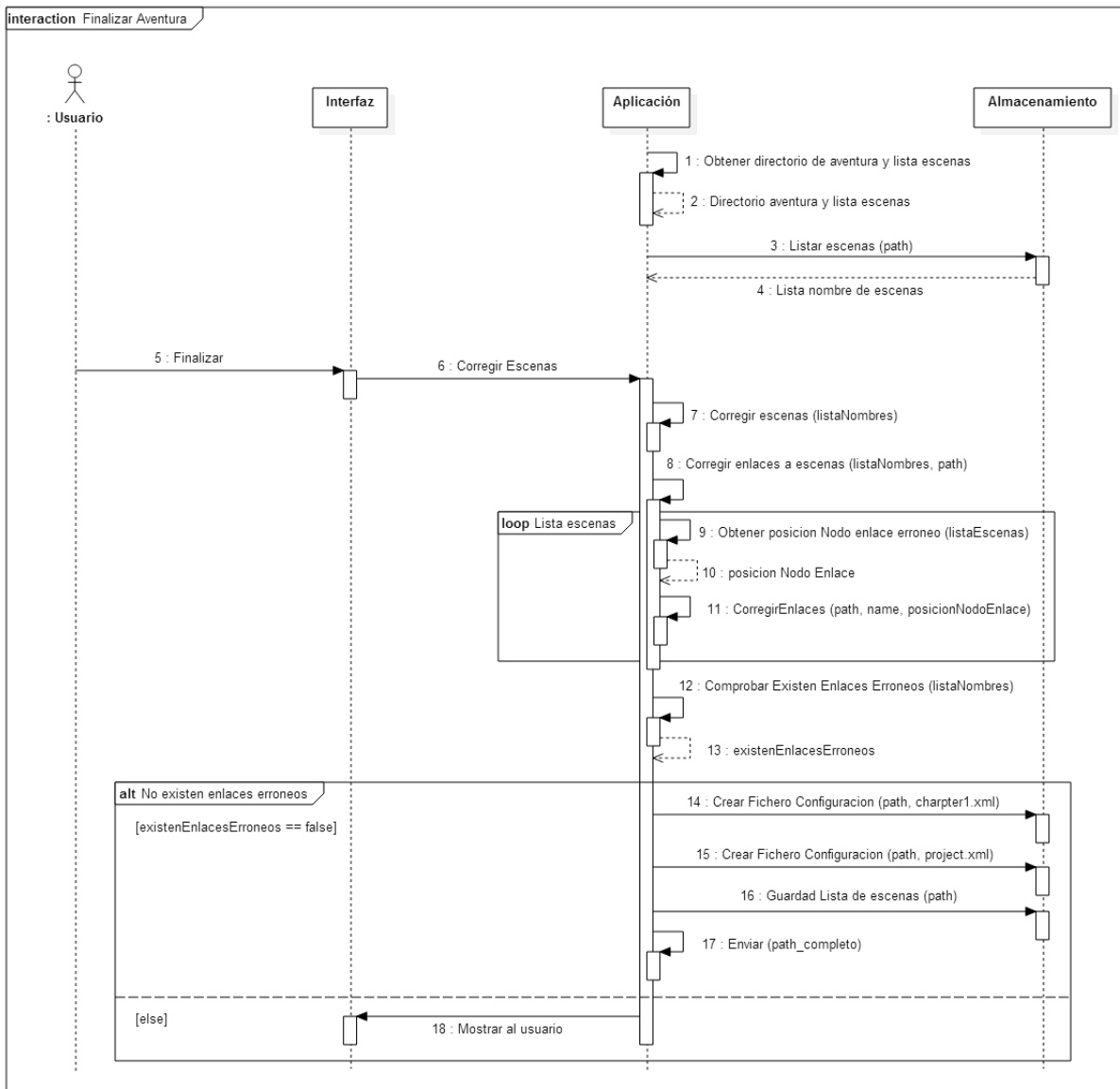


Figura 3.8: Finalizar Aventura

3.2.2. Escena

Imagen que representa cada una de las subpartes en las que se divide la aventura. Pueden estar enlazadas o no entre ellas, dicho enlace, visualmente representado por una zona interna dentro de la imagen (con nombre o no).

3.2.2.1. Obtener

Adquisición de la escena, con su nombre correspondiente, en caso de no especificarse se asignará por defecto. El usuario podrá elegir entre realizar una fotografía o seleccionar una imagen de la galería del dispositivo.

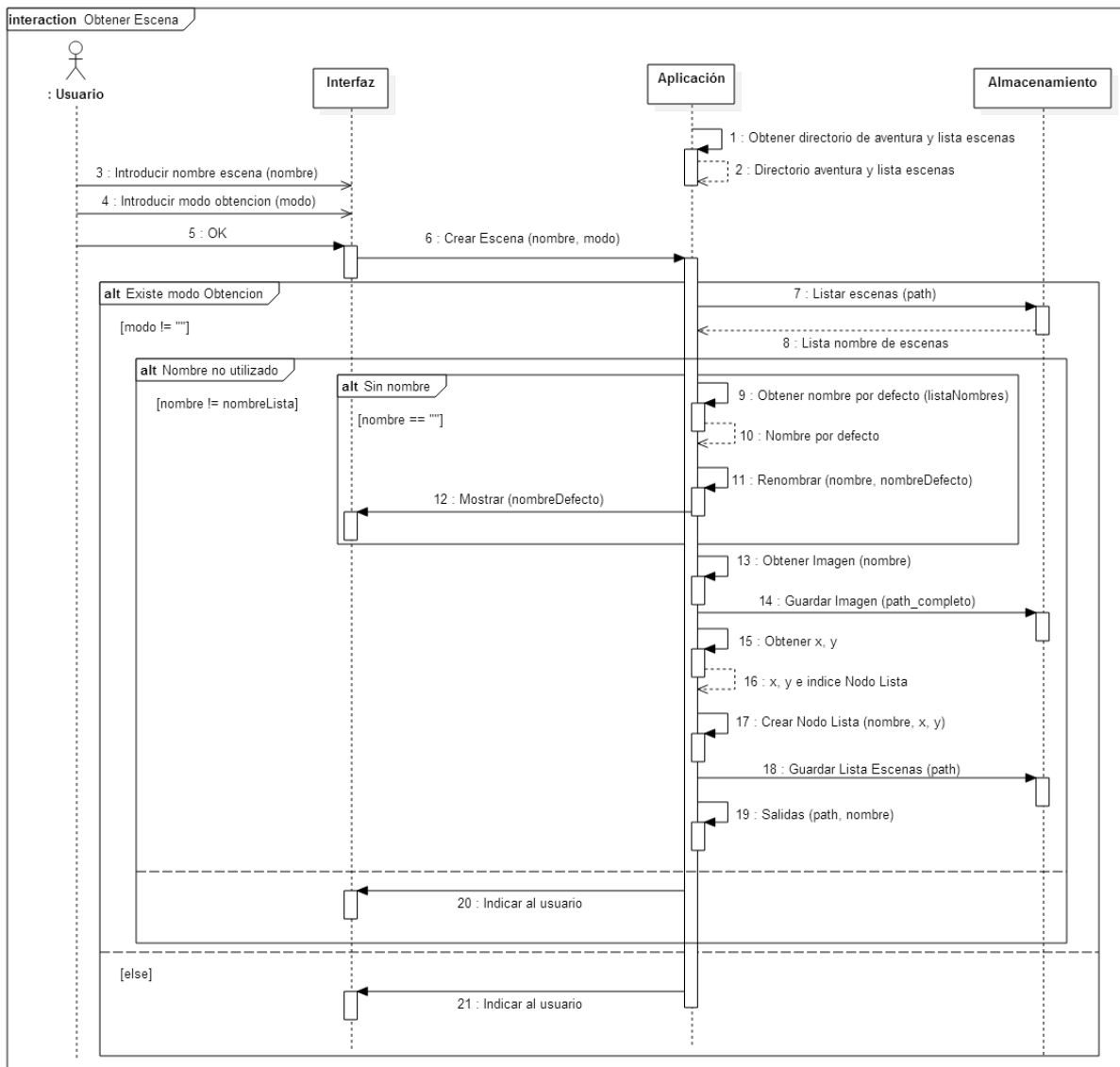


Figura 3.9: Obtener Escena

3.2.2.2. Modificar

Si se quisiera modificar parcialmente la escena (su nombre, crear nuevos enlaces) o incluso la imagen por completo que representa la escena (eliminando todos los enlaces previamente

creados) bastaría con seleccionar esta opción.

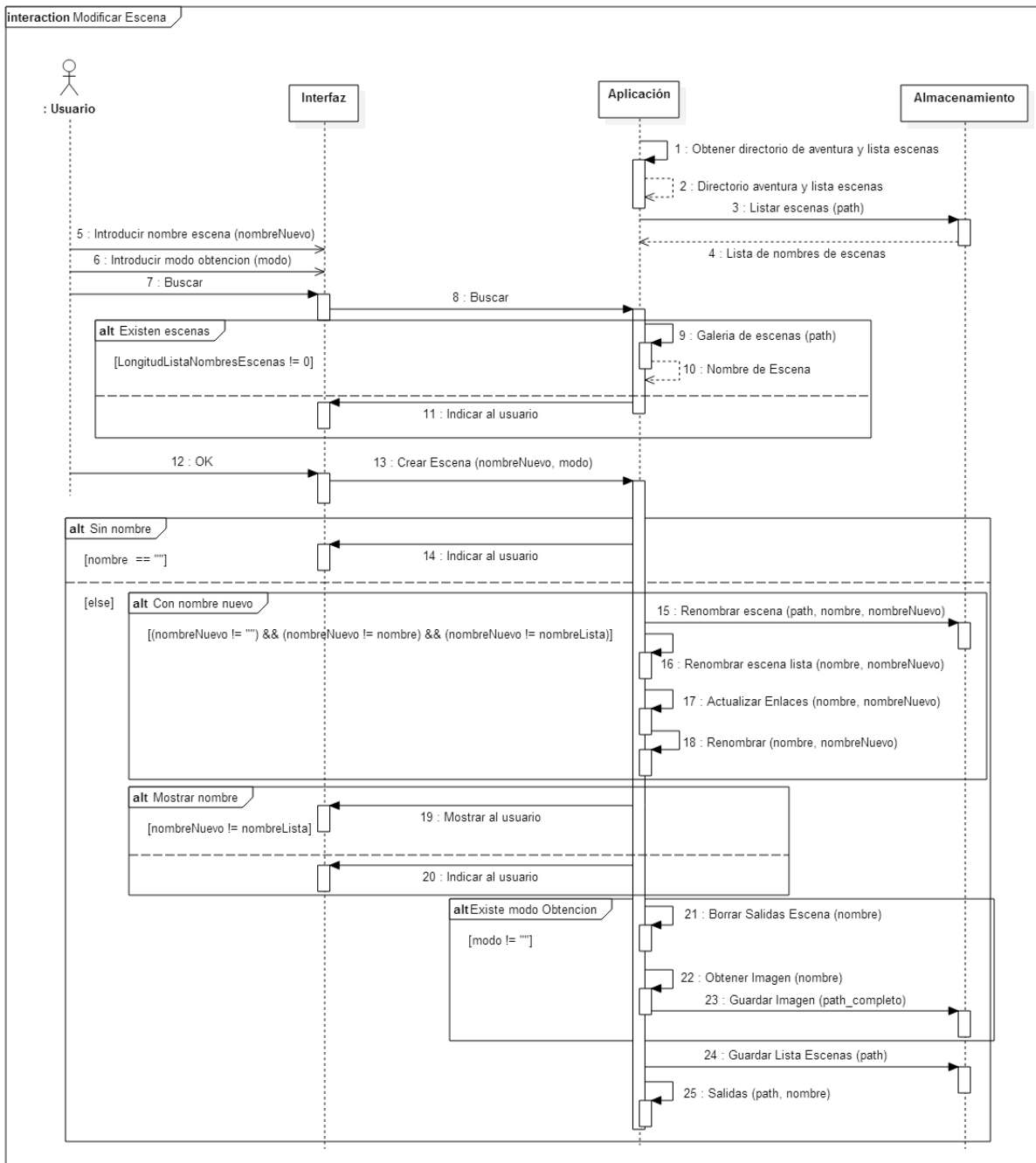


Figura 3.10: Modificar Escena

3.2.2.3. Eliminar

Toda escena no deseada podrá ser eliminada. Para ello será necesario seleccionarla dentro de la galería de escenas previamente creadas.

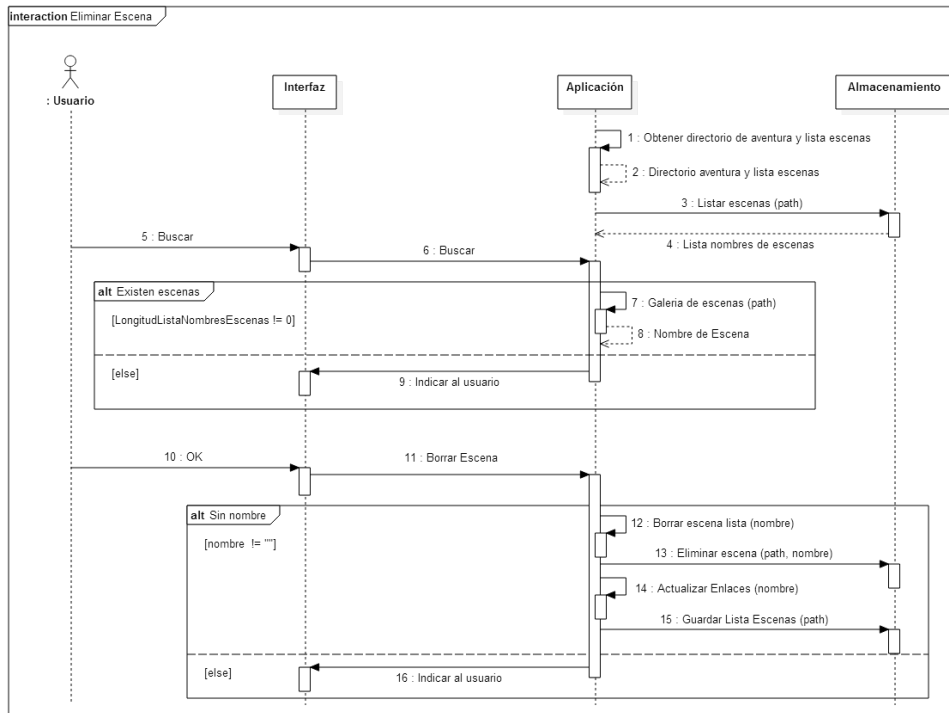


Figura 3.11: Eliminar Escena

3.2.3. Seleccionar Escena en la Galería

Recorriendo la galería de escenas previamente creadas, el usuario podrá seleccionar en cualquier momento la escena que desea para posteriormente procesarla como consideré adecuado.

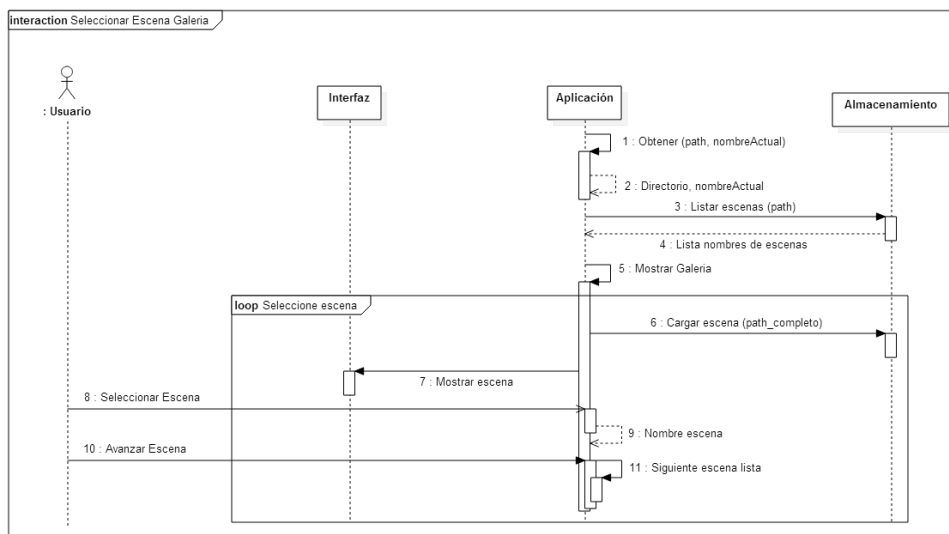


Figura 3.12: Seleccionar Escena en la Galería

3.2.4. Seleccionar Área de Salida

A través de las dimensiones del área que selecciona una zona dentro de la escena se creará un enlace con las coordenadas correspondientes, a su vez, se puede salir de la edición de enlaces siempre que se desee.

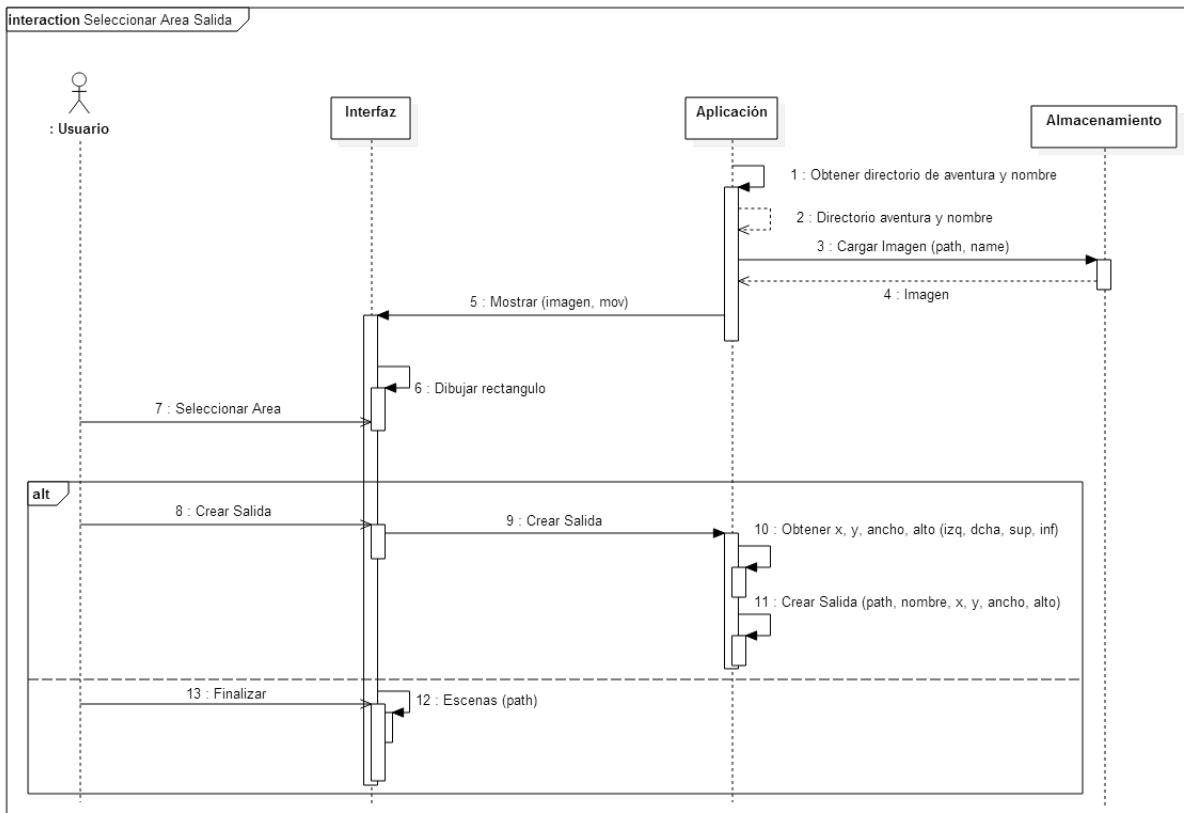


Figura 3.13: Seleccionar Área de Salida

3.2.5. Crear Salida

No es necesario indicar el nombre de la escena con la que enlaza (se asignará uno por defecto), tampoco el texto (no se mostrará nada al pasar por la zona). Se cuenta con la opción bastante cómoda de seleccionar una escena dentro de la galería 3.2.3, para evitar tener que recordar todas las escenas previamente creadas.

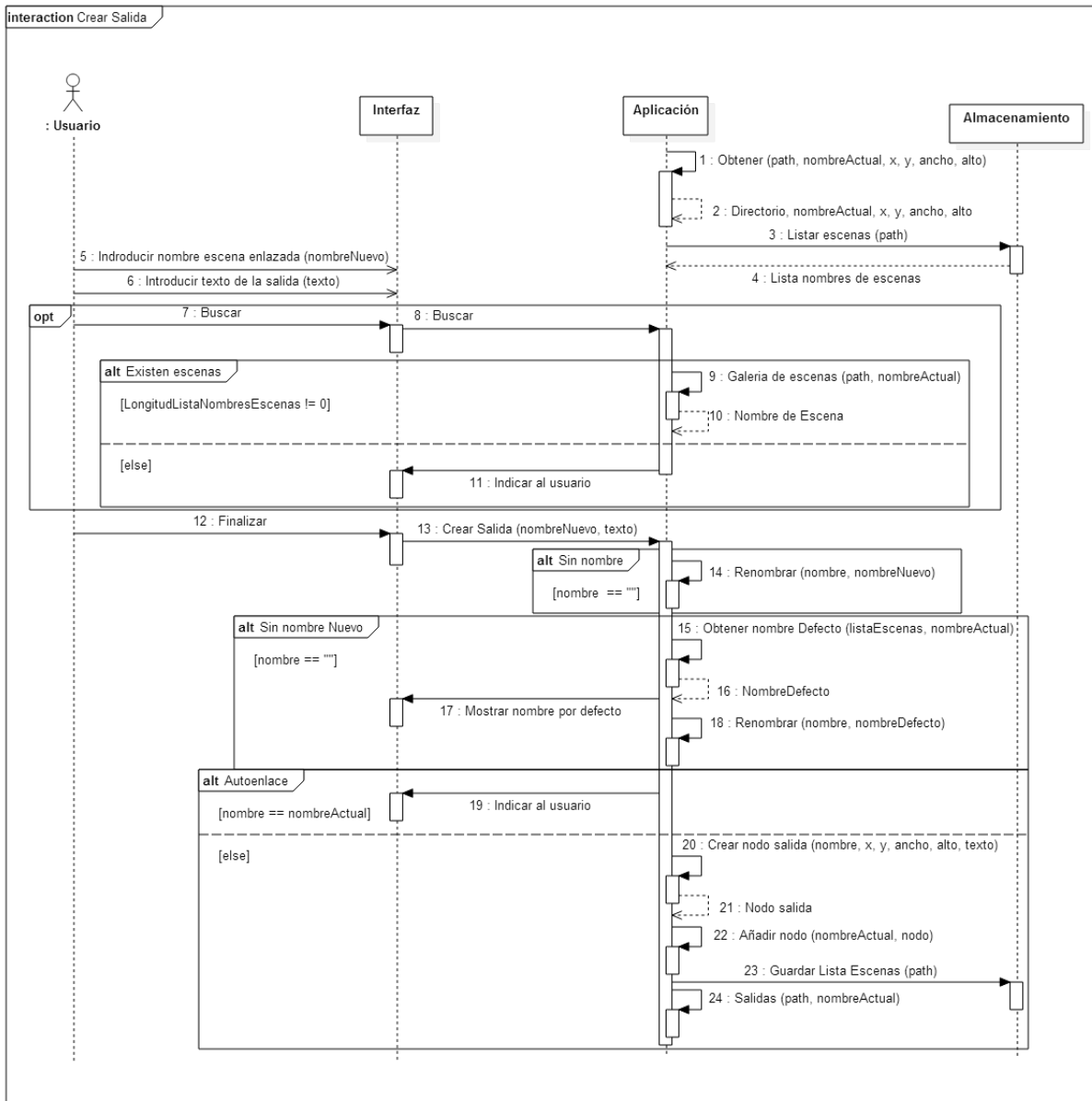


Figura 3.14: Crear Salida

3.2.6. Corregir Enlaces

La aplicación permitirá corregir los enlaces de una escena determinada previamente seleccionada en la galería.

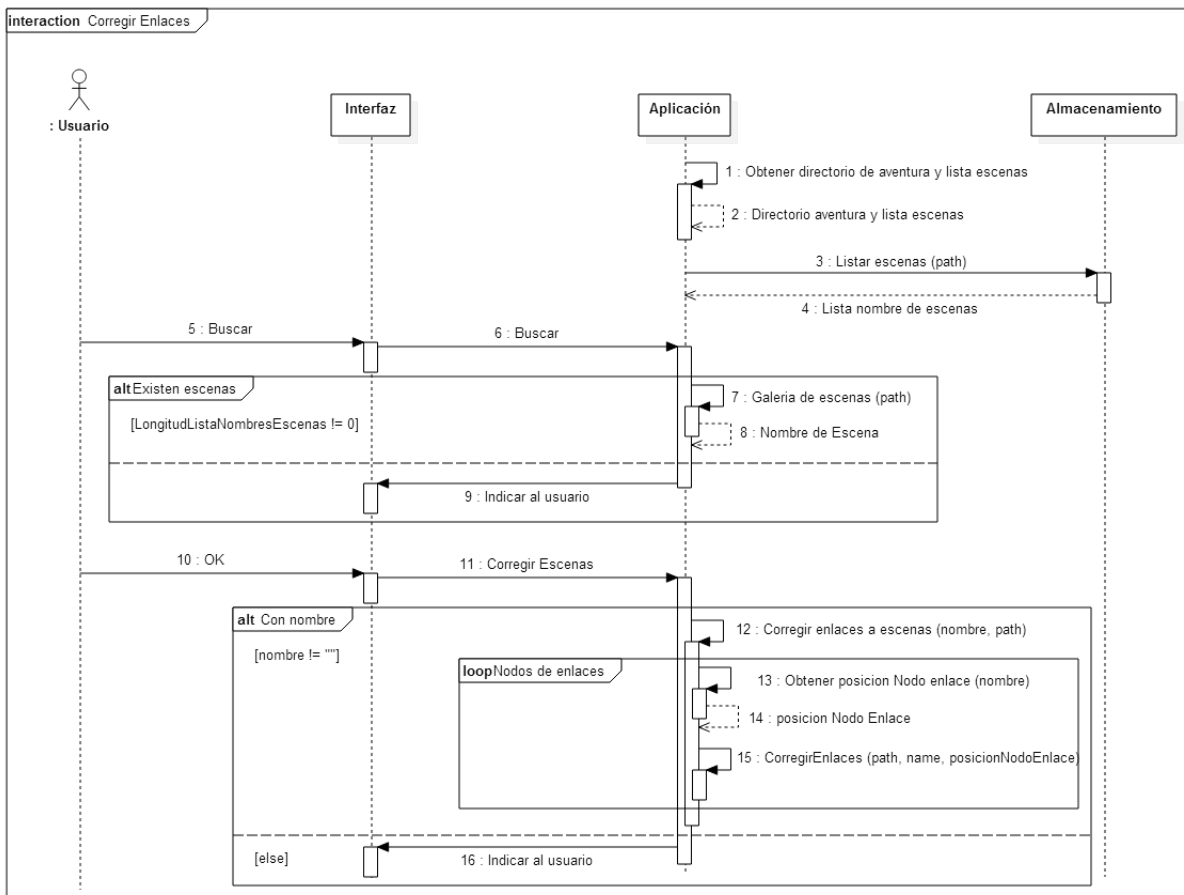


Figura 3.15: Corregir Enlaces

3.2.7. Enviar Ficheros

Una vez finalizada la aventura, se establecerá la conexión con el servidor mediante la dirección IP y se procederá al envío de los ficheros (descriptores de propiedades e imágenes).

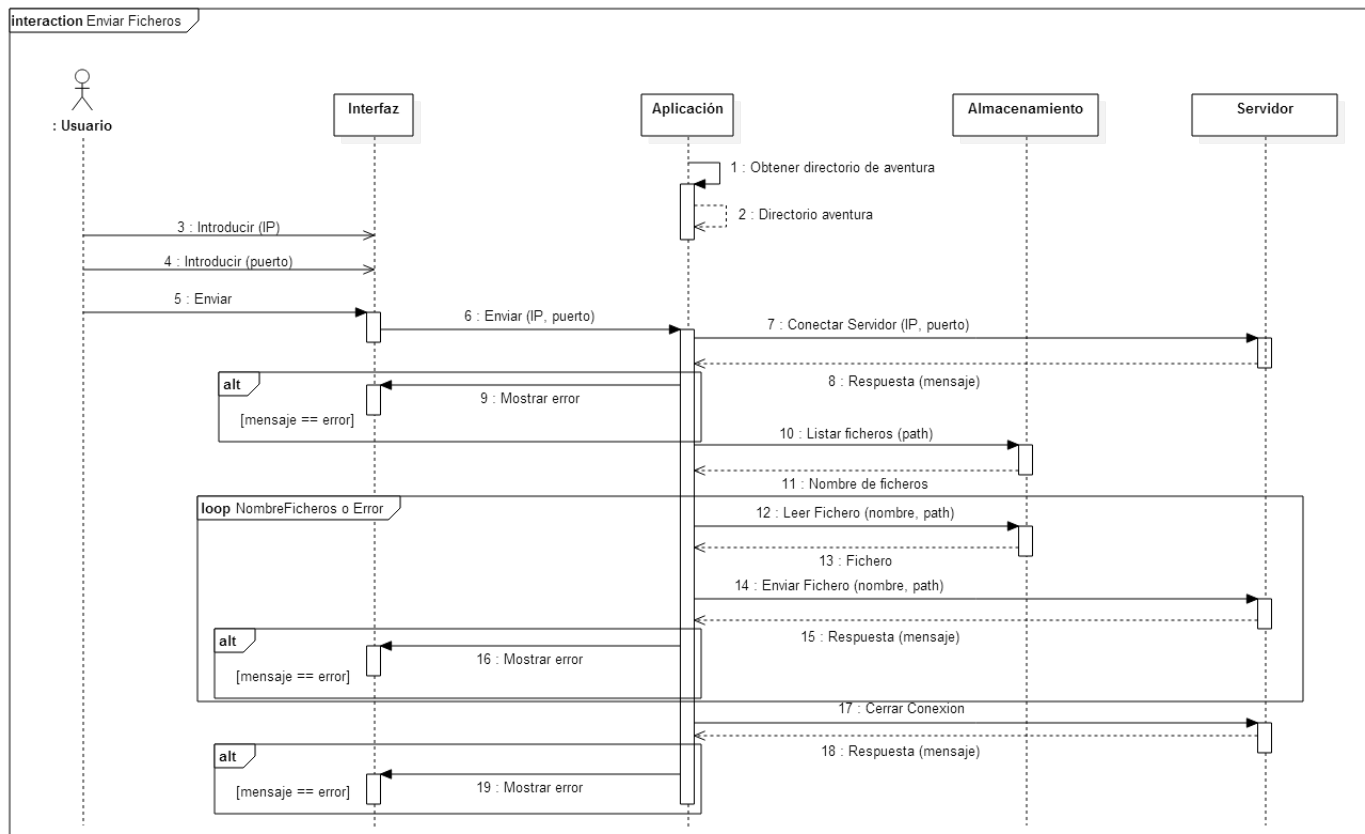


Figura 3.16: Enviar Ficheros

3.2.8. Aclaraciones

Ténganse en cuenta los siguientes datos a la hora de analizar los diagramas de secuencia correspondientes a cada funcionalidad:

- Por path completo se refiere al directorio de la aventura añadiéndole el nombre de la escena 3.2.2 o al directorio de aventuras más el nombre de la aventura 3.2.1
- Se mostrarán todo tipo de indicaciones al usuario (avisos, instrucciones, errores, etc.), se ha intentado que estén contenidos en los respectivos diagramas, en caso de faltar, el motivo reside en no añadir mayor complejidad
- Distíngase x e y (3.2.2.1, 3.2.5), en el primer caso hace referencia a la posición de las escenas dentro del cuadro de relación entre escenas del eAdventure 4.4.2, en el segundo, a la posición de la esquina superior izquierda del área de salida B.11 respecto al eje XY

3.3. Interfaz gráfica

Se pretende conseguir un aspecto sencillo e intuitivo, que no distraiga demasiado al usuario, pero que proporcione toda la comodidad máxima posible.

3.3.1. Servidor

Dividirlo en pasos (interacción con eAdventure, ejecutar el servidor) será el principal objetivo a la hora de realizar su diseño. A su vez, se tiene que mostrar claramente la IP del servidor, así como el proceso de envío de ficheros.

3.3.2. Aplicación

Los elementos dentro las actividades de la aplicación deberán:

- Estar lo suficientemente separados para dar una visión global del entorno.
- Orientados para automatizar la interacción del usuario.
- Aquellos destinados a la transición entre actividades deberán estar colocados en zonas claramente detectables por el usuario.

Capítulo 4

Implementación

A continuación entraremos a analizar cómo se ha llevado a cabo el diseño de manera eficaz y produciendo resultados satisfactorios.

Para ello empezaremos dando una descripción de una parte fundamental de la puesta en marcha del proyecto, hablamos del almacenamiento de datos 4.1, ya que sin una estructura bien definida no se podrían modificar las aventuras creadas posteriormente.

A continuación pasaremos por otra parte fundamental a la hora de implementar el proyecto, seleccionar el área de las escenas 4.2 correctamente a través de la interacción con el usuario. Si no se define un método de obtención de áreas adecuado pudiera verse afectada la subjetividad crítica del usuario.

Además se explicará como interactúan las actividades de la aplicación entre sí para tener una visión global de la comunicación entre ellas 4.3.

Después se describirá el tratamiento de ficheros, ya sea su almacenamiento 4.4.1, 4.4.2 o su envío 4.4.3.

Y por último, indicar algunos aspectos básicos de la implementación de la interfaz gráfica del servidor 4.5.1 y de la aplicación 4.5.2.

4.1. Estructura de datos

Es importante manejar y almacenar la información básica de la aplicación para garantizar su usabilidad. Tras sopesarlo deliberadamente se optó por usar una estructura básica, como es las listas enlazadas 4.1.1, que ofrezca una implementación sencilla pero a su vez permita realizar

todo tipo de cambios con facilidad (téngase en cuenta que la API de java 1.4.1 recoge todas las funcionalidades del tipo de datos *ArrayList*).

4.1.1. Listas enlazadas

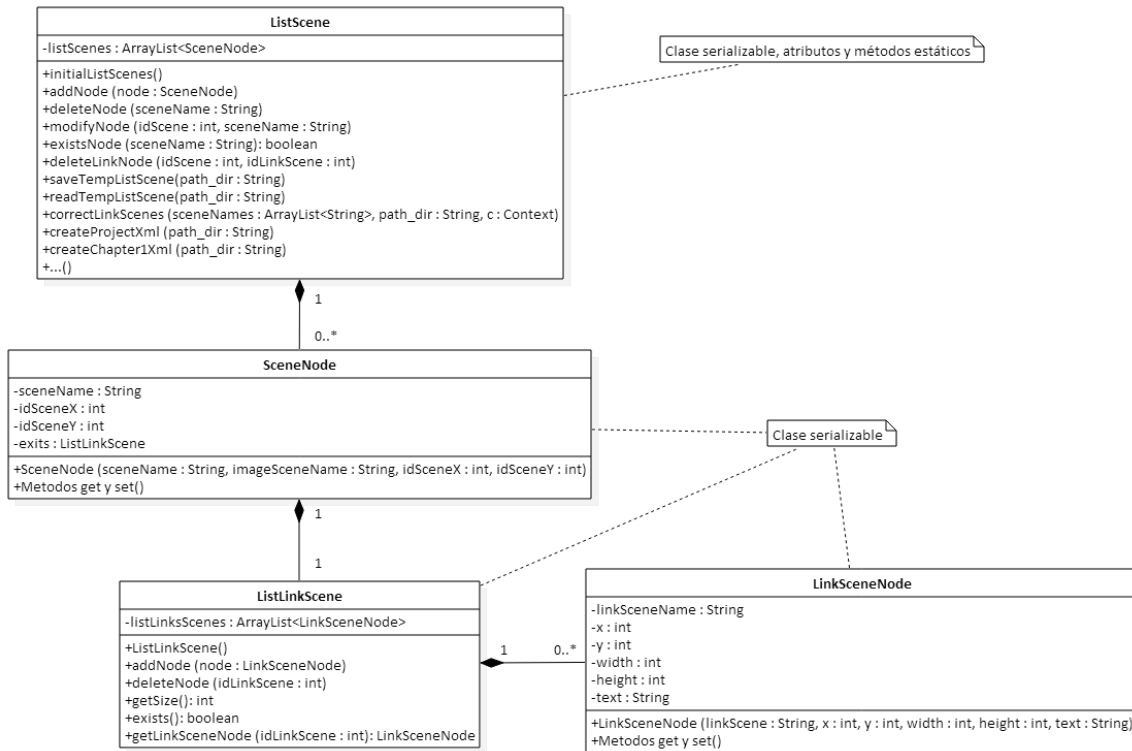


Figura 4.1: Listas enlazadas

Cada proyecto tendrá asociada una lista de escenas que a su vez se enlazarán, por cada escena, con una lista de salidas. Dicha lista de escenas será guardada (archivo *listScene.tmp* dentro de la carpeta del proyecto) a lo largo de su edición para garantizar la estabilidad de la aplicación en caso de error.

- Obtener escena (3.2.2.1 y B.7)
- Modificar escena (3.2.2.2 y B.8)
- Eliminar escena (3.2.2.3 y B.9)
- Crear salida (3.2.5 y B.12)
- Corregir enlaces (3.2.6 y B.13)

Se cargará cuando se seleccione la opción de modificar la aventura (3.2.1.3 y B.4) eliminará cuando se elimine la aventura (3.2.1.4 y B.5) e iniciará cuando se cree la aventura (3.2.1.1, B.2 y 3.2.1.2, B.3).

Es de acceso global en la aplicación y ofrece todo tipo de métodos para garantizar la manipulación de los diferentes tipos de nodos (escenas y salidas).

4.2. Selección de área

Para entender el proceso de selección de áreas de salida dentro de una escena empecemos primero comprendiendo como interactúa Canvas “is a container that holds various drawing elements (lines, shapes, text, frames containing other elements, etc.)” [9] con surfaceView “is a special subclass of View that offers a dedicated drawing surface within the View hierarchy”. [10]

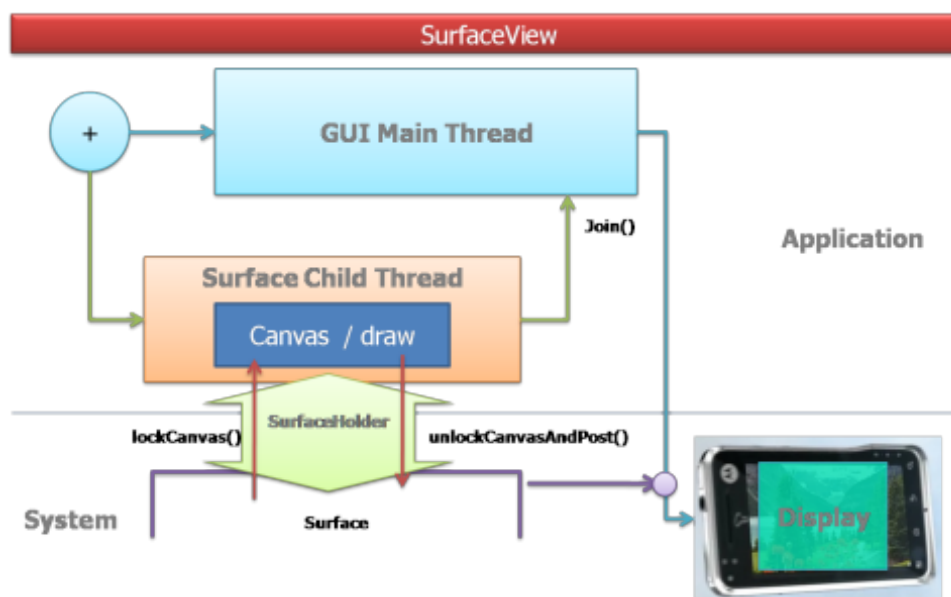


Figura 4.2: Diagrama de flujo surfaceView

Como puede observarse en la figura 4.2, existe un hilo especializado *Surface Child Thread* encargado de actualizar *Surface* mediante el uso de *Canvas*. Primeramente se prepara el *Canvas* `lockCanvas()`, a continuación se realizan todas las operaciones de dibujo `draw` y por último se devuelve el control a *Surface* `unlockCanvasAndPost()` y se muestra al usuario el resultado en el *Display*. El hilo principal se encarga de recoger toda la actividad del usuario y actualizar las posiciones de los píxeles a dibujar en el *Display*.

Una vez comprendido el funcionamiento, solo falta por explicar cómo se actualizan las coordenadas de los píxeles a través de la interacción con el usuario en el caso específico que nos encontramos, la aplicación eLearnT.

La API (Application Program Interface) de Android distingue entre distintos tipos de eventos táctiles [11], en especial, los utilizados para la recogida de datos han sido:

- **ACTION_DOWN**: primer dedo pulsado.
- **ACTION_UP**: primer dedo levantado.
- **ACTION_POINTER_DOWN**: se ha pulsado un dedo adicional al primero (p.ej. el segundo dedo).
- **ACTION_POINTER_UP**: se ha levantado un dedo diferente al primero (p.ej. el segundo dedo)
- **ACTION_MOVE**: se mueve algún dedo pulsado.

Y se han utilizado tres modalidades de actualización de coordenadas:

- **Move**: movimiento a lo largo de la pantalla con un dedo pulsado.
- **Zoom Zone**: dejando un dedo fijo (referencia) utilizar el otro en la dirección a ampliar, siempre alineado con el primero, es decir, por ejemplo, para expandir el lado vertical derecho sería necesario colocar el dedo de referencia a la izquierda del dedo secundario, y este a su vez, a lo largo del eje horizontal.
- **Zoom**: convencional. Utilizar los dos dedos para expandir el área de selección por completo o contraerlo. Ambos dedos tienen que estar alineados respecto a un mismo eje.

Con todo esto es posible dar al usuario un método sencillo y manejable de selección de áreas de salida.

4.3. Interacción entre actividades

En especial, lo que nos interesa para la aplicación es entender cómo iniciar, gestionar cuando reinicie o destruir una actividad.

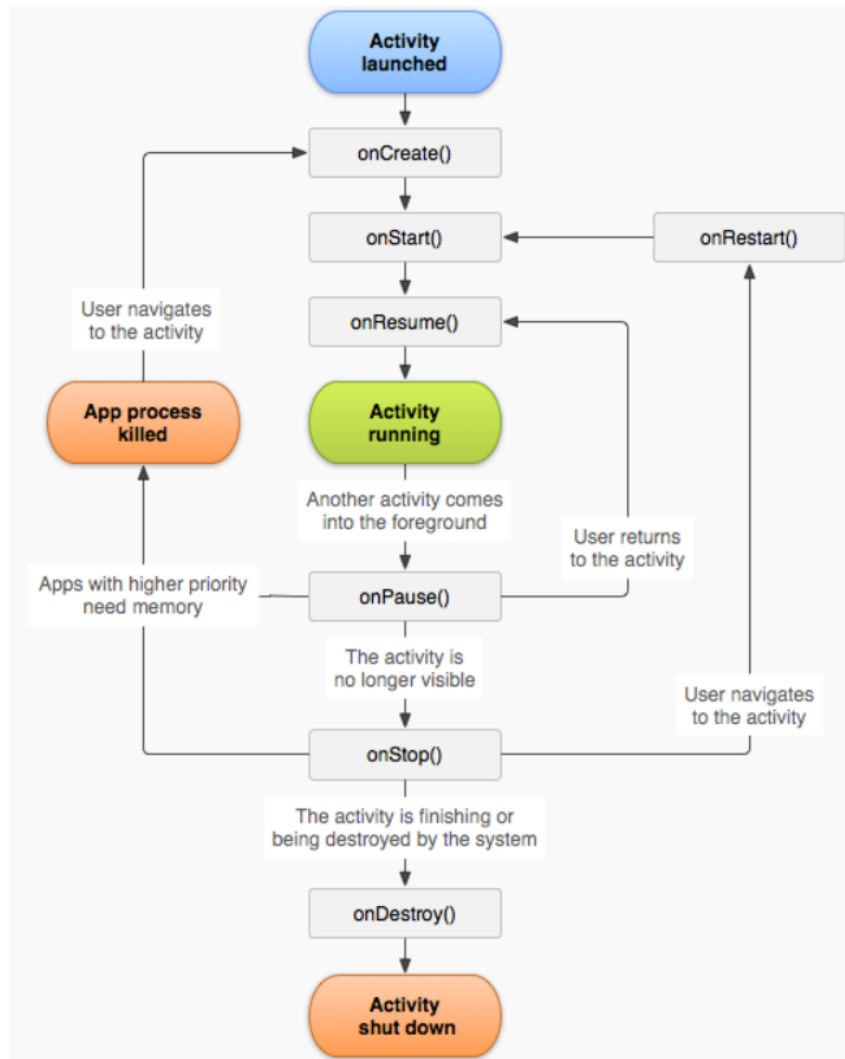


Figura 4.3: Ciclo de vida de las actividades [12]

Iniciar actividad:

```

// Iniciar intent con el contexto de la actividad actual y la clase de la actividad a iniciar
Intent i = new Intent(Context packageContext, Class<? > cls);
// Si se desea intercambiar datos entre actividades
i.putExtra(String name, String value);
// Lanzar la actividad
startActivity(i);

```

Reiniciar actividad:

```
/**
 * Metodo para gestionar el reinicio de una actividad
 */
@Override
protected void onRestart() {
    super.onRestart();
    // do actions
}
```

Finalizar actividad:

```
finish();
```

Además, también conviene entender cómo gestionar su creación y en el caso de la selección de áreas de salida 4.2 su pausa y puesta en marcha de nuevo.

Crear actividad:

```
/**
 * Gestionar la creacion de una actividad
 *
 * @param savedInstanceState para guardar los valores de la actividad
 */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Mapear componentes de la actividad definidos en el fichero xml
    setContentView(R.layout.nameActivityXml);
}
```



```
    // do actions
}
```

Pausar actividad:

```
/**
 * Metodo para gestionar la pausa de una actividad
 */
@Override
protected void onPause() {
    super.onPause();
    // do actions
}
```

Reanudar actividad:

```
/**
 * Metodo para gestionar la reanudacion de una actividad
 */
@Override
protected void onResume() {
    super.onResume();
    // do actions
}
```

4.4. Ficheros

Para tener un correcto funcionamiento del proyecto en general, es necesario realizar una gestión de ficheros adecuada que garantice el trámite de información entre sus diversas partes (aplicación, servidor y eAdventureEditor).

4.4.1. Guardando/Cargando Imágenes

Tema bastante polémico, ya que realizar una decodificación inadecuada de las imágenes pudiera producir un excesivo uso de memoria y sus consiguientes errores.

Para evitarlo se ha optado por reducir la calidad al obtener la imagen de la galería del dispositivo B.7 (opción 2):

```
// Load image
is = getContentResolver().openInputStream(selectedImage);
BufferedInputStream bis = new BufferedInputStream(is);

// Change to RGB_565 to down memory usage
BitmapFactory.Options options = new BitmapFactory.Options();
options.inPreferredConfig = Config.RGB_565;
Bitmap bitmap = BitmapFactory.decodeStream(bis, null, options);
```

Exactamente se ha realizado el mismo procedimiento al cargar las imágenes para mostrarlas en la aplicación (3.2.3, B.11) ya que es preferible un funcionamiento fluido en contraposición con un aspecto visual extremadamente detallado. Además, se han adaptado las imágenes a las resoluciones necesarias para su posterior uso:

scaled = Bitmap.createScaledBitmap(bitmap, width, height, true); (donde width y height corresponden a las dimensiones en píxeles)

A su vez, bajar demasiado la resolución pudiera causar en el usuario una insatisfacción visual innecesaria a la hora de probar la aventura en el editor de eAdventure.

Para ello, se ha aumentado ligeramente la calidad cuando las imágenes son tomadas desde la cámara B.7 (opción 1):

```
// Change quality
BitmapFactory.Options options = new BitmapFactory.Options();
options.inScaled = false;
options.inDither = false;
options.inPreferredConfig = Bitmap.Config.ARGB_8888;
```

```
Bitmap bm = BitmapFactory.decodeFile(path_dir + "/" + name, options);
```

Rotando la imagen si fuera necesario:

```
// Get Bounds
BitmapFactory.Options bounds = new BitmapFactory.Options();
bounds.inJustDecodeBounds = true;
BitmapFactory.decodeFile(path_dir + "/" + name, bounds);

// Read EXIF Data
ExifInterface exif = null;
try {
    exif = new ExifInterface(path_dir + "/" + name);
} catch (IOException e) {
    error = true;
}

// Get orientation
String orientString = exif.getAttribute(ExifInterface.TAG_ORIENTATION);
int orientation =
    orientString != null ? Integer.parseInt(orientString) : ExifInterface.ORIENTATION_NORMAL;
int rotationAngle = 0;
if (orientation == ExifInterface.ORIENTATION_ROTATE_90) rotationAngle = 90;
if (orientation == ExifInterface.ORIENTATION_ROTATE_180) rotationAngle = 180;
if (orientation == ExifInterface.ORIENTATION_ROTATE_270) rotationAngle = 270;

// Rotate Bitmap
Matrix matrix = new Matrix();
matrix.setRotate(rotationAngle, (float) bm.getWidth() / 2, (float) bm.getHeight() / 2);
Bitmap rotatedBitmap =
    Bitmap.createBitmap(bm, 0, 0, bounds.outWidth, bounds.outHeight, matrix, true);
```

Por último, se han gestionado los posibles errores de exceso de memoria, mostrando al

usuario el siguiente aviso *Memoria ocupada, espere un momento*. Se ha optado por esta opción para no tener que reducir aún más la calidad de las imágenes.

4.4.2. Generando descriptores xml

Existen tres tipos de ficheros *.xml* que recogen las propiedades de la aventura (figura 4.4) y las relaciones entre escenas (figura 4.5 y 4.7).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE game-descriptor SYSTEM "descriptor.dtd">
- <game-descriptor versionNumber="25">
  <title>3</title>
  <description/>
  - <configuration perspective="regular" keyboard-navigation="disabled"
    keepShowing="no" dragBehaviour="considerNonTargets"
    defaultClickAction="showDetails">
    <gui type="contextual" inventoryPosition="none"/>
    <mode playerTransparent="yes"/>
    <graphics mode="fullscreen"/>
  </configuration>
  - <contents>
    - <chapter path="chapter1.xml">
      <title>Capitulo 1</title>
      <description/>
    </chapter>
  </contents>
</game-descriptor>
```

Figura 4.4: descriptor.xml

Cuando creamos una aventura (B.2, B.3), se creará el fichero *descriptor.xml*. De entre todas las etiquetas, póngase énfasis especialmente en dos, *title* y *graphics* (atributo *mode*), es aquí donde se recoge la selección del usuario. En el caso de la figura 4.4:

- Título de lo aventura: **3**
- Modo gráfico: **fullscreen**

Otro detalle a tener en cuenta es que, al no hacer uso de objetos en la aventura, se ha deshabilitado la barra de inventario *inventoryPosition="none"* para tener un mayor área de selección de salidas.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
- <properties>
  <comment>Project Configuration</comment>
  <entry key="Chapter0.6.Visible">true</entry>
  <entry key="Chapter0.2.Y">30</entry>
  <entry key="Chapter0.2.X">168</entry>
  <entry key="Chapter0.3.Visible">true</entry>
  <entry key="Chapter0.1.Visible">true</entry>
  <entry key="Chapter0.4.Y">30</entry>
  <entry key="Chapter0.4.X">488</entry>
  <entry key="Chapter0.6.Y">150</entry>
  <entry key="Chapter0.6.X">8</entry>
  <entry key="autosave">yes</entry>
  <entry key="Chapter0.1.Y">30</entry>
  <entry key="Chapter0.1.X">8</entry>
  <entry key="Chapter0.5.Visible">true</entry>
  <entry key="Chapter0.4.Visible">true</entry>
  <entry key="Chapter0.3.Y">30</entry>
  <entry key="Chapter0.3.X">328</entry>
  <entry key="Chapter0.2.Visible">true</entry>
  <entry key="Chapter0.5.Y">30</entry>
  <entry key="Chapter0.5.X">648</entry>
</properties>

```

Figura 4.5: project.xml

A lo largo de la edición de la aventura, cada vez que se cree una escena, esta, será posicionada dentro de un cuadro de relaciones inter-escena (figura 4.6), para ello se le asignarán unas coordenadas X e Y debidamente equiespaciadas (figura 4.5).

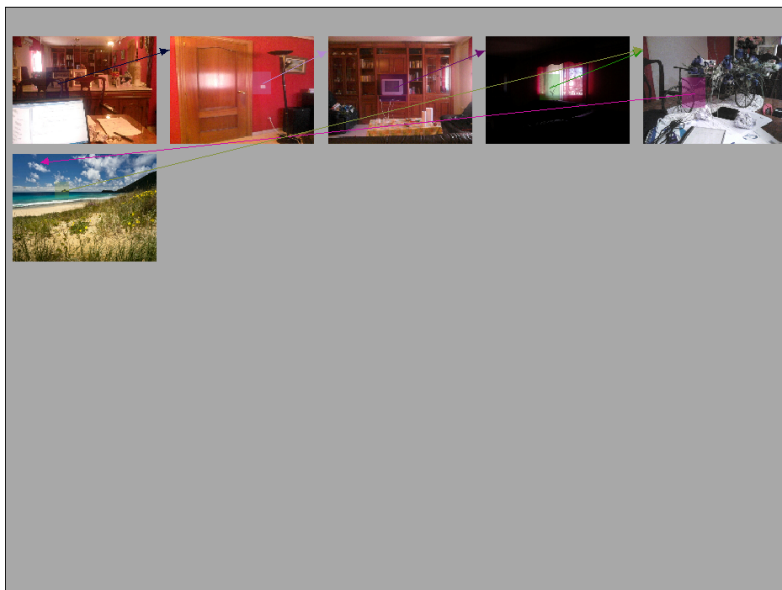


Figura 4.6: Relación entre escenas (eAdventureEditor)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eAdventure SYSTEM "eAdventure.dtd">
- <eAdventure>
  <comment>Project Configuration</comment>
  - <scene id="1" playerLayer="-1" playerScale="1.0" start="yes">
    - <resources name="No name">
      <asset type="background" uri="assets/background/1.jpg"/>
    </resources>
    <name/>
    <default-initial-position x="400" y="300"/>
    - <exits>
      - <exit x="188" y="177" destinyX="-2147483648" destinyY="-
        2147483648" hasInfluenceArea="no" height="172" idTarget="2" not-
        effects="no" rectangular="yes" transitionTime="0" transitionType="0"
        width="133">
        <exit-look text=""/>
      </exit>
    </exits>
  </scene>
  + <scene id="2" playerLayer="-1" playerScale="1.0" start="no">
  + <scene id="3" playerLayer="-1" playerScale="1.0" start="no">
  + <scene id="4" playerLayer="-1" playerScale="1.0" start="no">
  + <scene id="5" playerLayer="-1" playerScale="1.0" start="no">
  + <scene id="6" playerLayer="-1" playerScale="1.0" start="no">
  - <player>
    + <resources name="No name">
      - <textcolor bubbleBkgColor="#FFFFFF" bubbleBorderColor="#00000"
        showsSpeechBubble="no">
        <frontcolor color="#FFFFFF"/>
        <bordercolor color="#000000"/>
      </textcolor>
      - <description>
        <name/>
        <brief/>
        <detailed/>
      </description>
      <voice name="" synthesiseAlways="no"/>
    </player>
  </eAdventure>

```

Figura 4.7: chapter1.xml

Cada escena tiene asociado un id, nombre de imagen (con extensión) y salidas correspondientes. Las salidas tendrán asociado un id (escena con la que enlazan) y un texto a mostrar. En la figura 4.7:

- Id de la escena: etiqueta *scene* atributo *id* valor **1**
- Nombre de la imagen: etiqueta *asset* atributo *uri* valor **1.jpg**
- Id de la escena con la que enlaza la salida: etiqueta *exit* atributo *idTarget* valor **2**
- Texto a mostrar en la salida: etiqueta *exit-look* atributo *text* valor **""** (no muestra nada)

4.4.3. Envío

Al tratarse de una conexión TCP (Transmission Control Protocol), orientada a conexión y que garantiza la ordenación de los paquetes (*segmentos TCP*) a lo largo de la transmisión, se ha empleado un algoritmo bastante sencillo que consiste en leer los bloques del fichero a través de un *buffer* y a continuación ir enviándolos al servidor a través de un *OutputStream* (**PrintStream**). Previamente se enviará el nombre del fichero y se esperara su confirmación (*br.readLine()*), esto es así, para garantizar la sincronización a la hora de realizar el envío de bloques del fichero. Por cada fichero se establecerá una conexión TCP y al finalizar la transmisión se cerrará para indicarle al bucle de lectura de datos en el servidor que ya ha terminado la transmisión.

4.5. Interfaz gráfica

En las siguientes subsecciones daremos detalles de la implementación gráfica del servidor y de la aplicación.

4.5.1. Servidor

Para la implementación de la interfaz gráfica del servidor se ha hecho uso de *Java Swing* “Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegados y tablas.” [13]

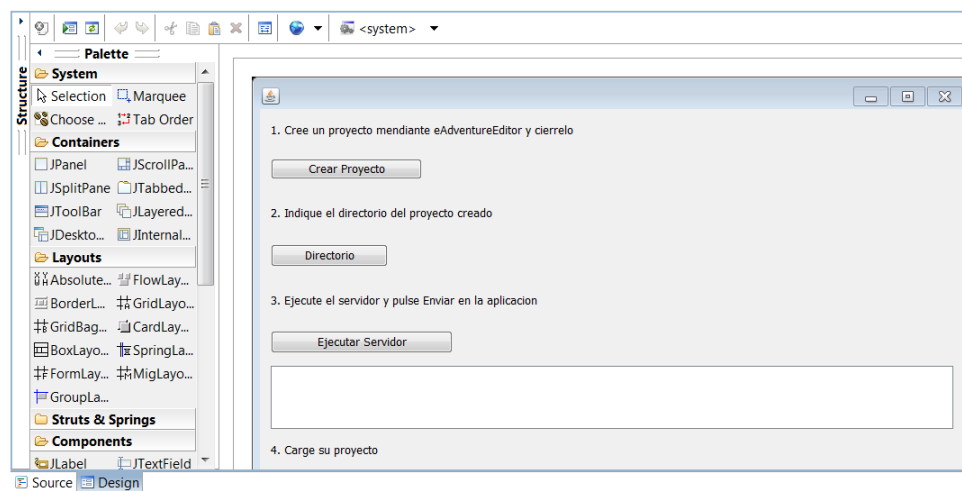


Figura 4.8: Utilizando Java Swing

4.5.2. Aplicación

Se ha adaptado la interfaz gráfica a distintas resoluciones para que soporte diferentes dispositivos de Android. Téngase en cuenta que al no disponer de suficientes dispositivos (físicamente) no se puede garantizar su correcta visualización en todas las resoluciones.

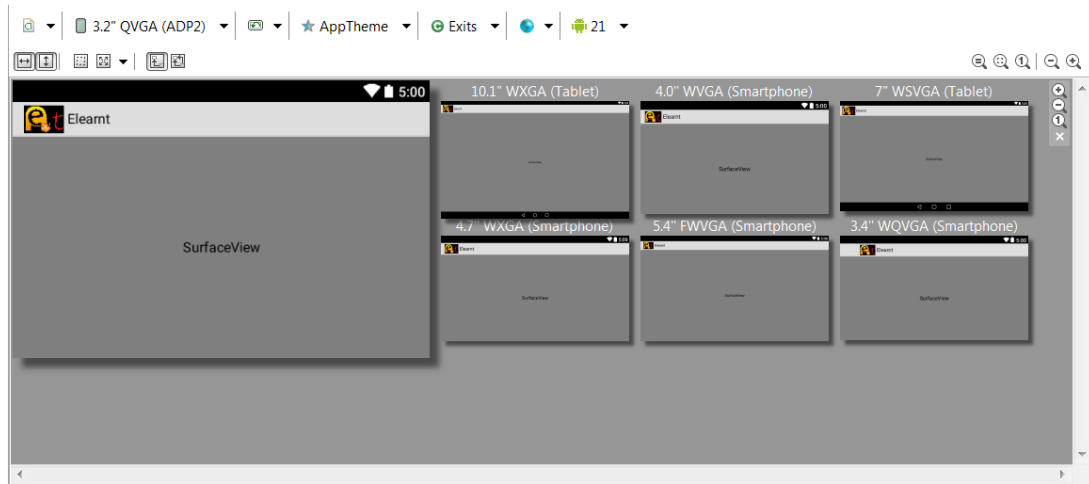


Figura 4.9: Resoluciones soportadas

Como cada dispositivo tiene asociada una resolución, destinando un fichero *strings.xml* para cada una de ellas es posible adaptar las *dimensiones* (internas o relativas) de los distintos componentes (botones, textos editables, imágenes, títulos, etc.).

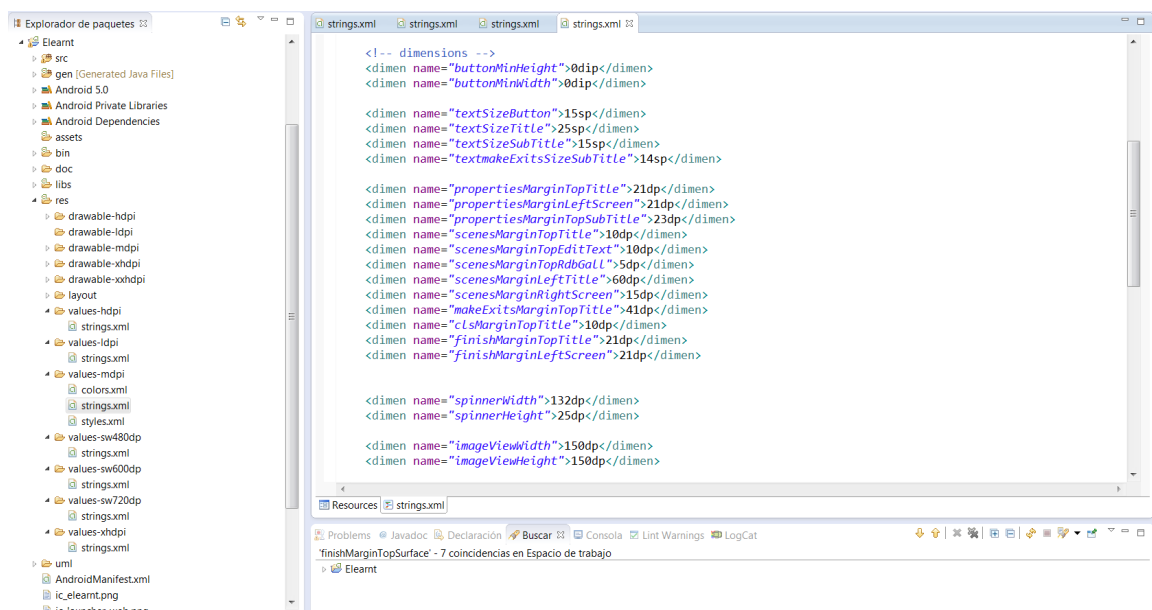


Figura 4.10: Gestión de diferentes resoluciones

Dispositivos utilizados:

- Sony Xperia Tipo ST21i (3.2") **dispositivo base**
- Sony Xperia Z1 (4.7")
- Wolder miTablink (7")
- Arnova 10C G3 (10.1")

Capítulo 5

Publicación

Para cumplir uno de los principales objetivos del proyecto 2.1 se procederá a publicar la aplicación y el servidor bajo una licencia que garantice los principios del software libre <http://www.gnu.org/philosophy/free-sw.es.html>. Esto es, derecho a modificar, utilizar y distribuir el programa (tanto modificado como original) libremente para el uso que se desee, garantizando ciertos derechos de autor, como exigir el uso de la misma licencia o no negar la autoría.

5.1. Tipo de licencia

La licencia elegida ha sido General Public License de GNU versión 3, ya que cumple con los criterios anteriormente definidos en la sección, evitando restricciones innecesarias en la liberación del software.

Para ello se ha incluido en cada fichero fuente (tanto de la aplicación como del servidor) un resumen de la licencia y el copyright (*Copyright (C) 2014 José Ignacio Madan Frías*), también indicado en las interfaces, véase <http://www.gnu.org/licenses/gpl-3.0.html#howto>. Además se ha añadido un fichero (*COPYING*) que contiene todas las especificaciones de la licencia.

Enlace a repositorio gitHub: <https://github.com/FLeKyS/Workspace>

Capítulo 6

Conclusiones

En el marco general del aprendizaje electrónico aporta una herramienta con gran utilidad, más en concreto, supone para eAdventure una alternativa a la hora de crear aventuras educativas sencillas bastante útil, manejable y funcional. Espero ante todo que finalicen con éxito el proyecto eAdventure-Android, que supone un gran avance dentro del entorno del aprendizaje electrónico móvil, *M-learning* (móviles, tabletas, iPod, dispositivos pequeños en general) y no menos importante, que el proyecto aquí presente les sirva de utilidad.

Como opinión personal, a pesar de haberme introducido plenamente en las nuevas tecnologías y su concepción de interacción usuario-dispositivo mediante la pantalla táctil, sigo pensando que no es una gran idea, por tener que multiplexar en tiempo la pantalla, está bien el concepto de integrar al máximo las partes que componen un terminal, pero el precio que se paga es alto, la visibilidad. Como alternativa, existen medios de interacción externos al dispositivo, pero a la vez integrados en él, es decir, no utilizan espacio de la pantalla, por ejemplo, teclado y ratón láser. O incluso separar la pantalla mediante un proyector (*picoprojector*). Todos ellos pueden ser una buena extensión para nuestros dispositivos actuales, y puntos intermedios para lo que, a mi parecer, podría formar parte de nuestras vidas en cierto momento, me refiero a la integración de captación de datos en el propio usuario.

6.1. Lecciones aprendidas

Fueron muchas las lecciones aportadas a través de la ejecución del proyecto, por lo que puede considerarse una experiencia formativa enriquecedora. Aquí se presentan algunas de ellas:

- Aprendizaje efectivo y constituyente del manejo de las APIs de Android
- Creación de interfaces gráficas sencillas e intuitivas
- Interacción desarrollador-cliente
- Actualización de software
- Acuerdo diseño-implementación
- Creación de prototipos

6.2. Conocimientos aplicados

Principalmente informáticos, más en particular, entendimiento y correcta utilización del lenguaje de programación *Java*, encontrándome con la dificultad de tener que trabajar con APIs nuevas, y por lo tanto, tener que enfrentarme a dicho problema. Además, estar familiarizado con el entorno de desarrollo integrado *Eclipse* ha facilitado mucho el proceso de creación del proyecto (tanto de la aplicación como del servidor).

Desde un punto de vista más centrado en las telecomunicaciones, la concepción cliente-servidor, en específico, el entendimiento del protocolo TCP (*Transmission Control Protocol*) a garantizado el correcto funcionamiento de la parte final de la aplicación, poder enviar la aventura previamente creada para probarla con *eAdventureEditor* como alternativa al uso de medios físicos (cable USB *Universal Serial Bus*).

6.3. Trabajos futuros

Estaría bastante bien añadir más funcionalidad a la aplicación, es decir, poder crear aventuras educativas completas incluyendo personajes, diálogos, objetos, visión en tercera persona, etc. Afortunadamente, se está realizando un proyecto que cumple estas características 1.3. Contando con que son los creadores de la plataforma *eAdventure*, con lo que poseen toda la experiencia y conocimientos acerca de su funcionamiento, sé que estará en buenas manos.

En un caso particular, este proyecto puede ser un buen impulsor para introducirme dentro del mundo del desarrollo de aplicaciones para Android.

Apéndice A

Instalación

Primero empecemos con la aplicación. Antes de nada, es necesario activar la opción **Orígenes desconocidos** que permite al usuario instalar aplicaciones que no pertenecen al Market (Google play entre otros), para ello diríjase dentro de su dispositivo a Ajustes/Seguridad o Ajustes/Aplicaciones (según la versión de Android que tenga).

Después copie el fichero Elearnt.apk en el dispositivo Android, ya sea a través del cable USB o estableciendo una conexión cliente-servidor utilizando una aplicación específica, por ejemplo, AirDroid <http://airdroid.com/>. Una vez copiado el fichero, búsquelo dentro del dispositivo con cualquier gestor de ficheros, por ejemplo, Root Browser <https://play.google.com/store/apps/details?id=com.jrummy.root.browserfree> (si no posee uno propio el terminal), a continuación ejecútelo para que proceda la instalación aceptando los permisos que requiere.

Estos son:

Almacenamiento *modificar/eliminar contenido de la tarjeta SD*

Comunicación de red *acceso integro a Internet*

Pasemos al servidor, basta con extraer el fichero *ElearntServer.rar* y ejecutar *ElearntServer.jar*, téngase en cuenta que la carpeta */bin* contenida dentro del archivo comprimido tiene que estar en el mismo directorio que el archivo *ElearntServer.jar*

Si se encuentra en *Linux*, tendrá que cambiar los permisos del fichero *ElearntServer.jar* para que permita la ejecución, para ello abra un terminal y ejecute:

cd (Directorio donde se encuentra ElearntServer.jar)

chmod 700 ElearntServer.jar

A.1. Requisitos

Los requisitos previos a la puesta en marcha de las herramientas son:

- *Sistema operativo Android.*

Es necesario que el dispositivo tenga instalado al menos la versión 2.2 *Froyo*. Aunque durante el desarrollo ha sido probada con versiones superiores 4.0 *Ice Cream Sandwich* y 4.4 *KitKat*, por no disponer de terminales específicos. Además incluye soporte con la última versión 5.0 *Lollipop* (no ha sido probada).

- *Máquina virtual de Java.*

Al menos será necesario tener instalada la versión 1.7 de la máquina virtual de Java *jre 7*.

- *Misma subred.*

Para poder realizar la transferencia de ficheros (imágenes y propiedades) es necesario que tanto el servidor como el dispositivo Android, estén conectados a la misma red, esto es, a través de un router en común.

- *eAdventure.*

Tener instalada la plataforma de creación de juegos educativos eAdventure, actualmente con versión 1.5-Release Candidate 4 <http://e-adventure.e-ucm.es/download/>

Apéndice B

Manual de usuario

Antes de comenzar asegúrese de tener correctamente instalados la aplicación y el servidor A, además de cumplir los requisitos necesarios A.1.

B.1. Comenzando



Figura B.1: Menú principal.

Pulse el botón **Crear Proyecto** para pasar a la edición de las propiedades de la aventura. Sí desea salir de la aplicación basta con pulsar el botón **Salir**, alternatively, pulse dos veces el botón *atrás*, la primera vez que lo pulse será informado *Pulse atras de nuevo si desea salir*.

B.2. Crear Aventura

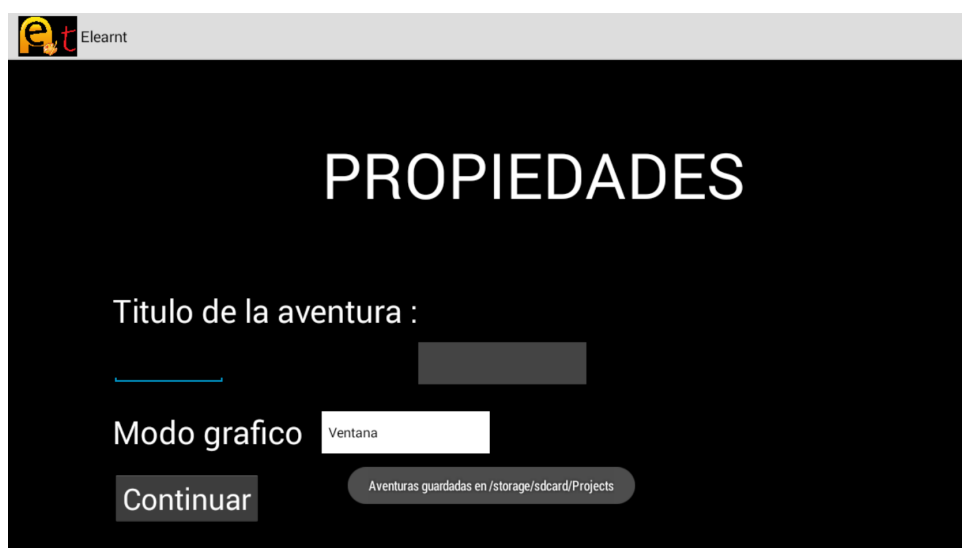


Figura B.2: Crear aventura.

Todas las aventuras serán guardadas dentro de la carpeta indicada en el mensaje de aviso, en el ejemplo **/storage/sdcard/Projects**

Introduzca el nombre que le quiere asignar a la aventura (en caso de no hacerlo le será asignado uno por defecto) y seleccione la opción de visualización (ventana por defecto):

- Ventana (de tamaño estándar)

- Ventana con el fondo en negro

- Pantalla completa (opción solo válida para el sistema operativo Windows)

Si prefiere editar una aventura que ya creo anteriormente, selecciónela en la lista desplegable situada junto a la parte para introducir el nombre de la aventura, en la figura B.3 el cuadro con el texto *Manual de usuario*.

Para ambas situaciones, a continuación presione en **Continuar**. Si crea una nueva aventura sin nombre, se le indicará el nombre por defecto que se le ha puesto *Nombre de la aventura por defecto: x* (donde x es el nombre asignado).



Figura B.3: Seleccionar aventura existente.

B.3. Crear Aventura de Nuevo

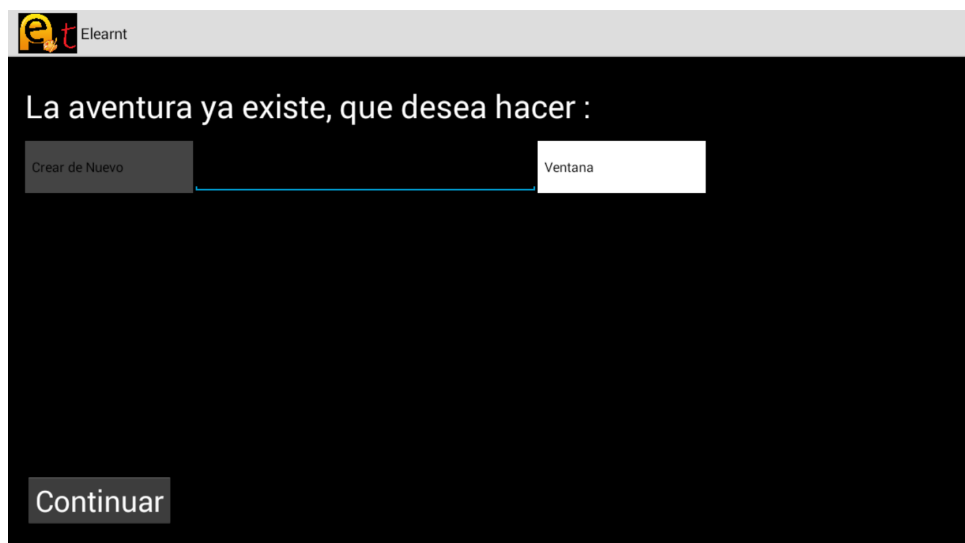


Figura B.4: Crear aventura de nuevo.

Si desea cambiar el nombre de la aventura y su modo gráfico, indíquelo introduciendo el texto y seleccionando el modo de visualización 3.2.1.1. Por defecto, se asignará el mismo nombre y modo gráfico que tenía la aventura.

A continuación presione en **Continuar**, la aventura será creada de nuevo, por lo que no existirán escenas ni enlaces.

B.4. Modificar Aventura

Si desea cambiar el nombre de la aventura y su modo gráfico, indíquelo introduciendo el texto y seleccionando el modo de visualización 3.2.1.1. Por defecto, se asignará el mismo nombre que tenía la aventura y modo gráfico.

A continuación presione en **Continuar**, la aventura conservará todas las escenas y enlaces previamente creados.

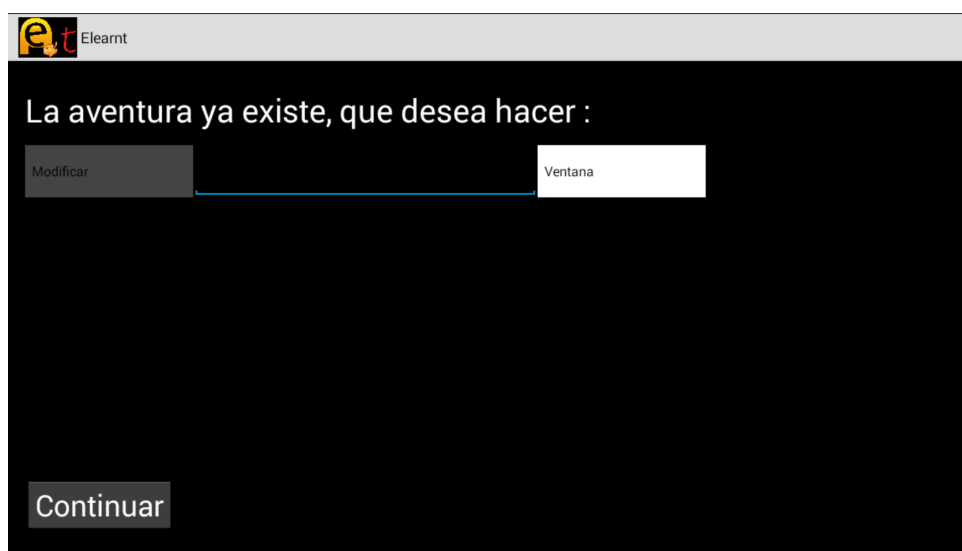


Figura B.5: Modificar aventura.

B.5. Eliminar Aventura

Presione en **Continuar**, para eliminar la aventura por completo, esto es, además de perder todas la escenas y enlaces creados, no aparecerá en la lista de aventuras previamente creadas, ya que se habrá borrado del directorio de aventuras 3.2.1.1.

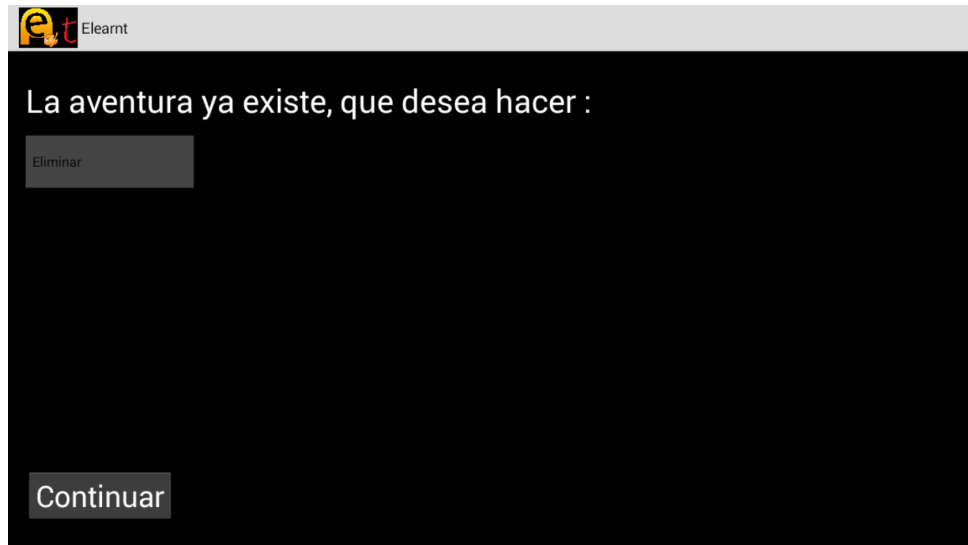


Figura B.6: Eliminar aventura.

B.6. Enviar Aventura

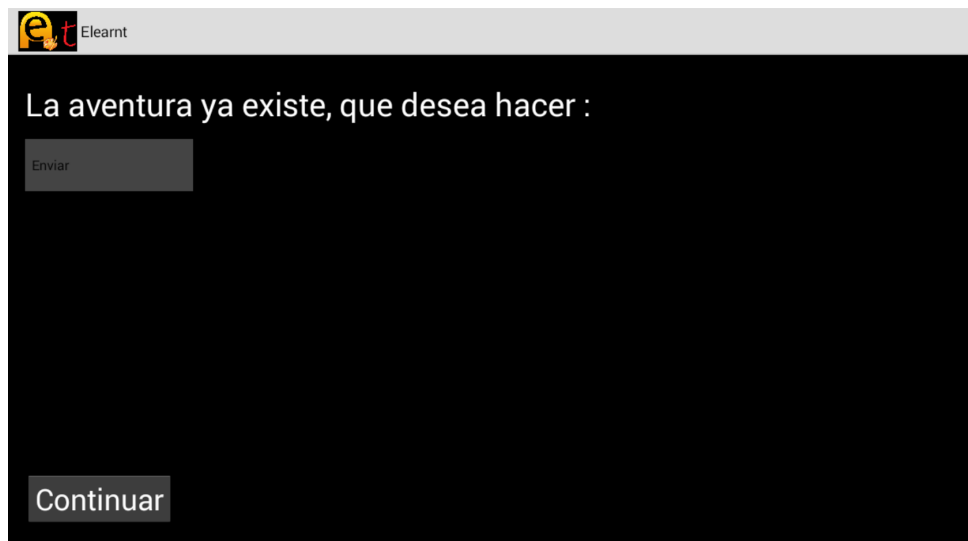


Figura B.7: Enviar aventura.

Si lo que desea es pasar a la ventana donde poder enviar la aventura B.16 presione **Continuar**.

B.7. Obtener Escena

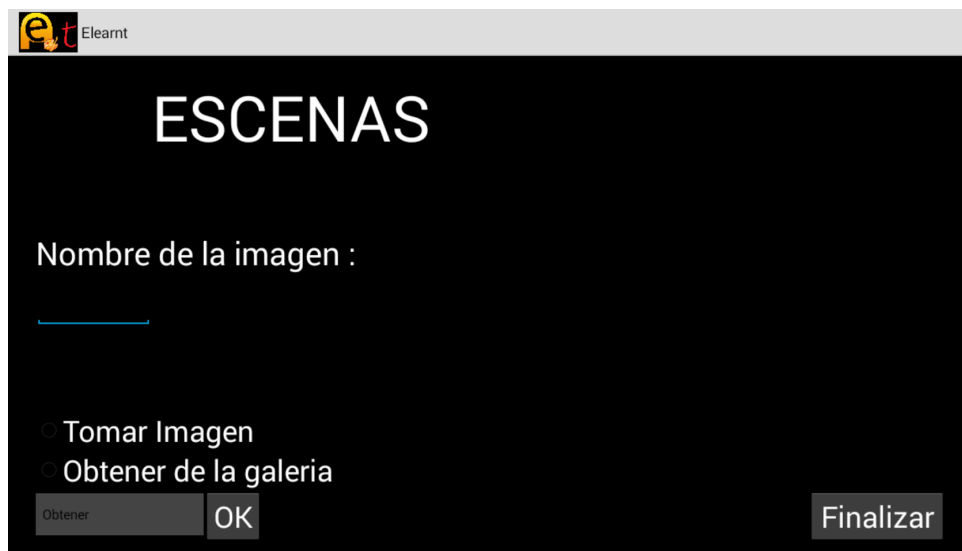


Figura B.8: Obtener escena.

Introduzca el nombre de la escena si lo desea (no es necesario, se asigna un nombre por defecto). Después seleccione el modo de obtención de la escena:

1) Tomar Imagen (se obtendrá la escena mediante la realización de una fotografía con la cámara de su dispositivo)

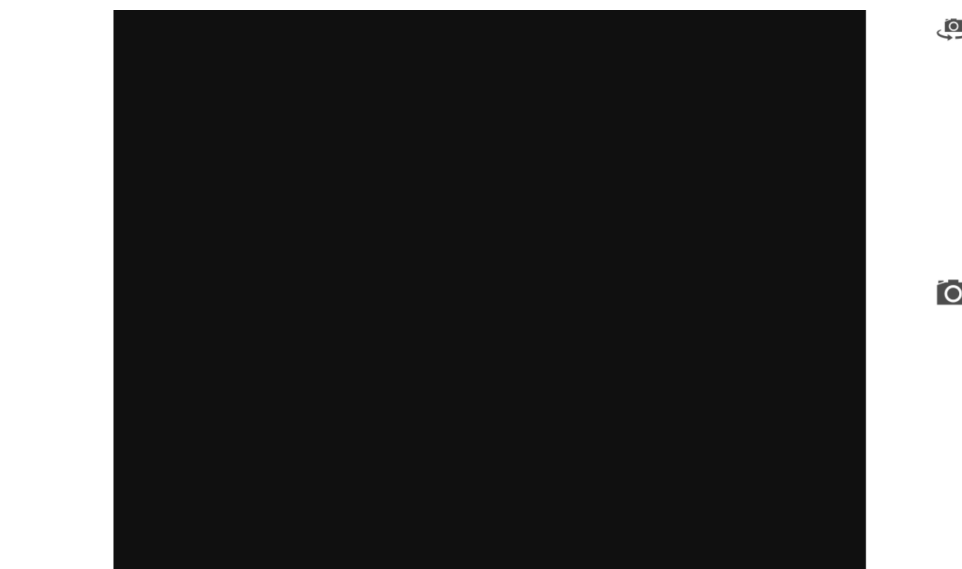


Figura B.9: Hacer fotografía.

2) Obtener de la galería (se obtendrá de la galería de imágenes del dispositivo)

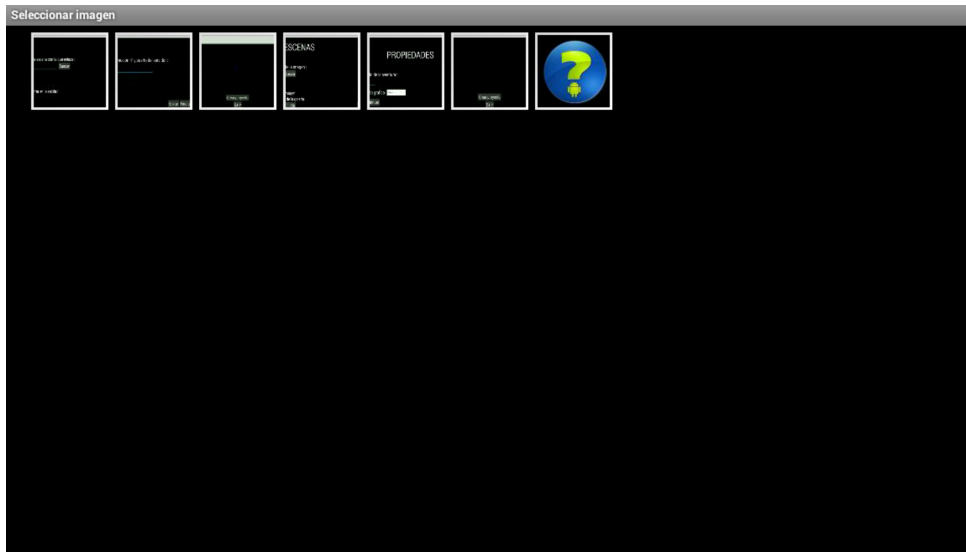


Figura B.10: Seleccionar imagen.

Por último, presione el botón **OK**. Si el nombre introducido ya está en uso le será indicado mediante el siguiente aviso *El nombre de la escena ya esta utilizado*, a su vez, también le será indicado si no ha seleccionado ningún modo de obtención de la escena *Seleccione Tomar Imagen u Obtener de la galería*. Si no se le ha asignado ningún nombre a la escena, aparecerá el siguiente mensaje *Nombre de la escena por defecto: x* (donde x es el nombre que se le ha asignado).

B.8. Modificar Escena

Pulse en **Buscar** para seleccionar la escena a modificar en la galería de escenas B.10. A continuación, introduzca el nombre (si desea cambiar el actual).

Seleccione el modo de obtención de la escena B.7 (si desea obtenerla de nuevo).

Por último presione en **OK**, después se abrirá una nueva ventana con la escena a editar B.11. Si se ha producido el cambio de nombre de manera satisfactoria le será indicado *Nombre de la escena: x* (donde x es el nuevo nombre de la escena), en caso contrario se mostrará *El nombre de la escena ya esta utilizado*; si no se ha introducido un nombre nuevo, también le será indicado de la misma forma para tener constancia del nombre actual de la escena. En caso de no haber seleccionado ninguna escena previa de la galería, se mostrará el siguiente aviso *Por favor, seleccione la escena a modificar en la lista*.

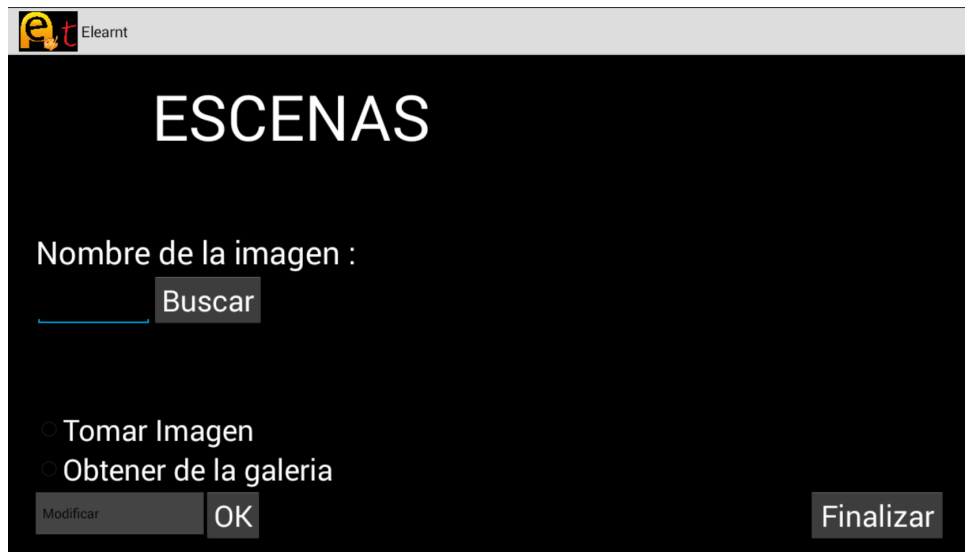


Figura B.11: Modificar escena.

En caso de no existir escenas, figura B.12, si pulsa **OK** le será indicado *No existen escenas creadas*.

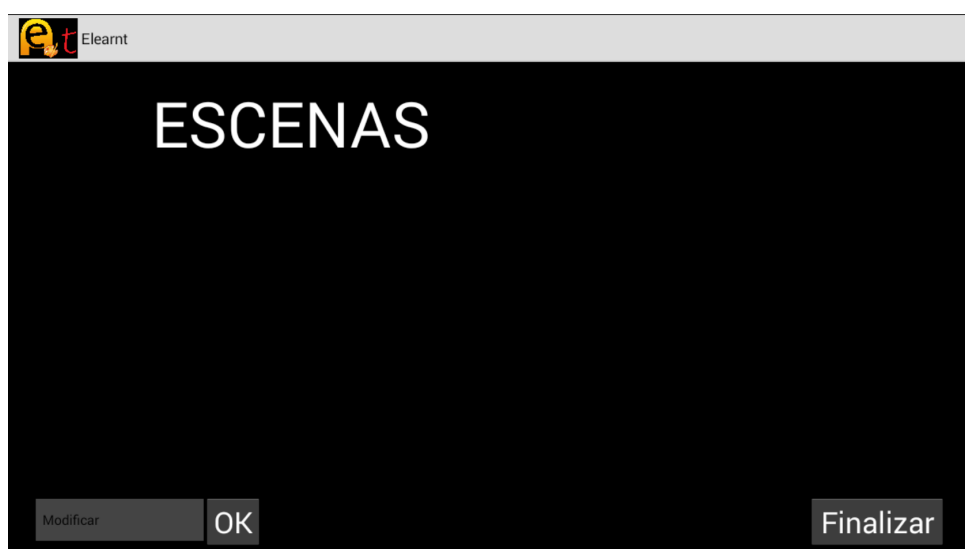


Figura B.12: Modificar escena (no existen escenas).

B.9. Eliminar Escena

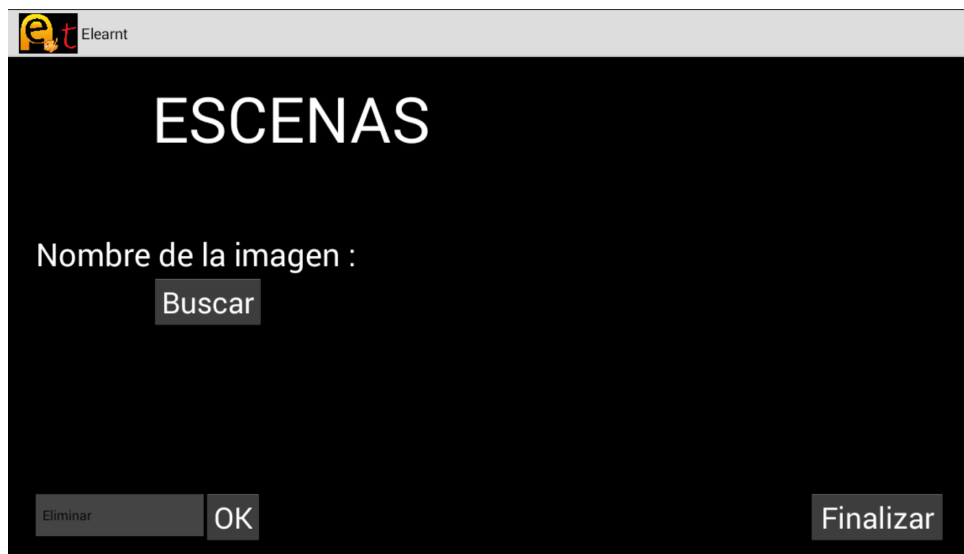


Figura B.13: Eliminar escena.

Pulse en **Buscar** para seleccionar la escena a eliminar en la galería de escenas B.10. A continuación presione **OK**, la escena será borrada por completo junto a sus enlaces (en caso de no haberse seleccionado *Por favor, seleccione la escena a eliminar en la lista*). Si no existiesen escenas creadas, figura B.14, si pulsa **OK** le será indicado *No existen escenas creadas*.

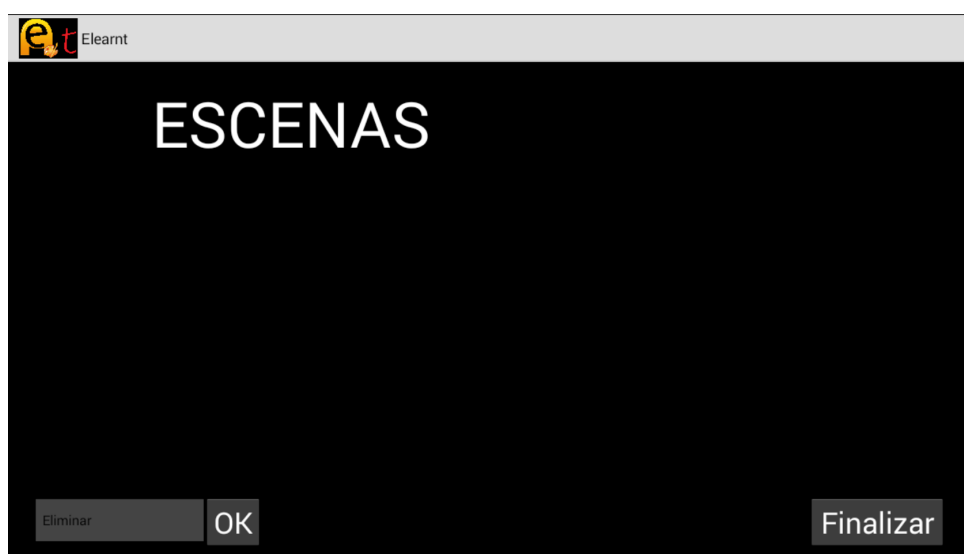


Figura B.14: Eliminar escena (no existen escenas).

B.10. Seleccionar Escena en la Galería

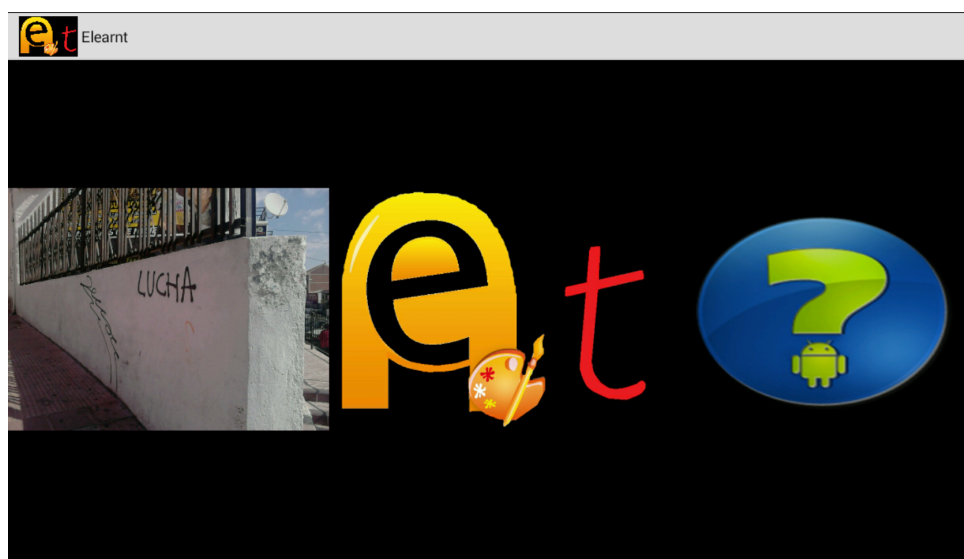


Figura B.15: Seleccionar escena en la galería.

Muévase a lo largo de la galería en busca de la escena que desea seleccionar, a continuación púlsela.

B.11. Seleccionar Área de Salida

La primera vez que entre en la ventana de edición de crear enlaces de escena (salidas), le aparecerán las siguientes instrucciones:

- *Con un dedo fijo, mueva el otro para ampliar la zona que dese.*
- *Presione dos veces seguidas dentro del area para crear la salida.*
- *Mueva los dos dedos a la vez para hacer zoom.*
- *Presione atras para finalizar.*

Puede ir probando su funcionamiento sobre la marcha. En el caso del *zoom por zonas*, siempre ha de tener un dedo de presionado que actuará como referencia, para cambiar de dedo base, es necesario, no solo dejar pulsado el próximo dedo que actuará de referencia, sino también levantar el dedo base anterior para actualizar la referencia; una vez fijado el dedo base, **primer**

dedo presionado fijo, puede desplazarlo a lo largo de la pantalla para cambiar el punto de referencia. En cuanto al *zoom* convencional, podrá ampliar exteriormente el área alejando ambos dedos y viceversa si desea acercarla.

Por último, presione el botón **atrás** si ha finalizado la edición o dos veces seguidas dentro del área si desea crear una salida B.12.

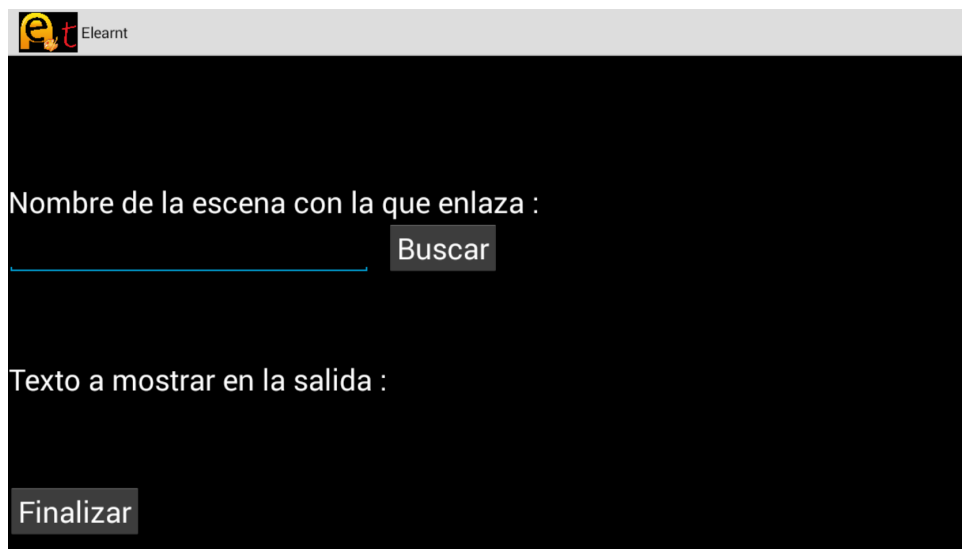


Figura B.16: Seleccionar área de salida.

B.12. Crear Salida

Busque (presione en **Buscar**) la escena con la que quiere enlazar en la galería de escenas B.10, si no se hubiese creado todavía introduzca el nombre de la escena manualmente (de no hacerlo se asignará por defecto *Escena a la que enlaza por defecto: x*, donde x es el nombre de la escena) una vez creada la verdadera escena con la que se quería enlazar siempre podrá corregir el enlace, véase B.13. A continuación indique el texto a mostrar en la salida cuando el usuario este situado sobre ella (por defecto no se mostrará nada).

Por último, presione en **Finalizar** para crear la salida o **atrás** en caso contrario.



Nombre de la escena con la que enlaza :

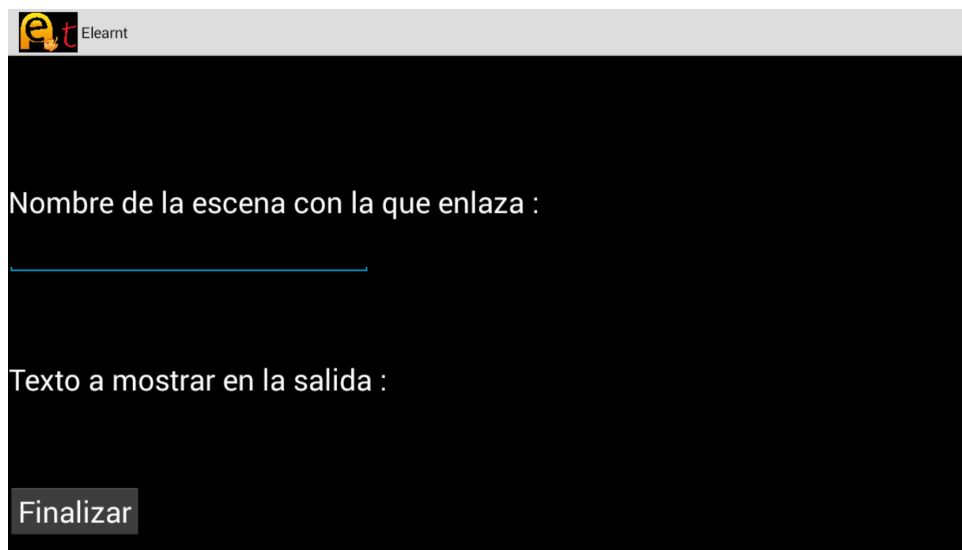
Buscar

Texto a mostrar en la salida :

Finalizar

Figura B.17: Crear salida.

Si no existiesen salidas la opción **Buscar** estaría deshabilitada, figura B.18. Téngase en cuenta que no está permitido el auto-enlace.



Nombre de la escena con la que enlaza :

Buscar

Texto a mostrar en la salida :

Finalizar

Figura B.18: Crear salida (no existen escenas).

B.13. Corregir Enlaces

Seleccione la escena en la cual quiere corregir los enlaces (pulse **Buscar**), de no hacerlo le será indicado *Por favor, seleccione la escena a corregir en la lista*, a continuación presione **OK**.

De no existir enlaces creados en la escena se mostrará, *No existen enlaces creados*.

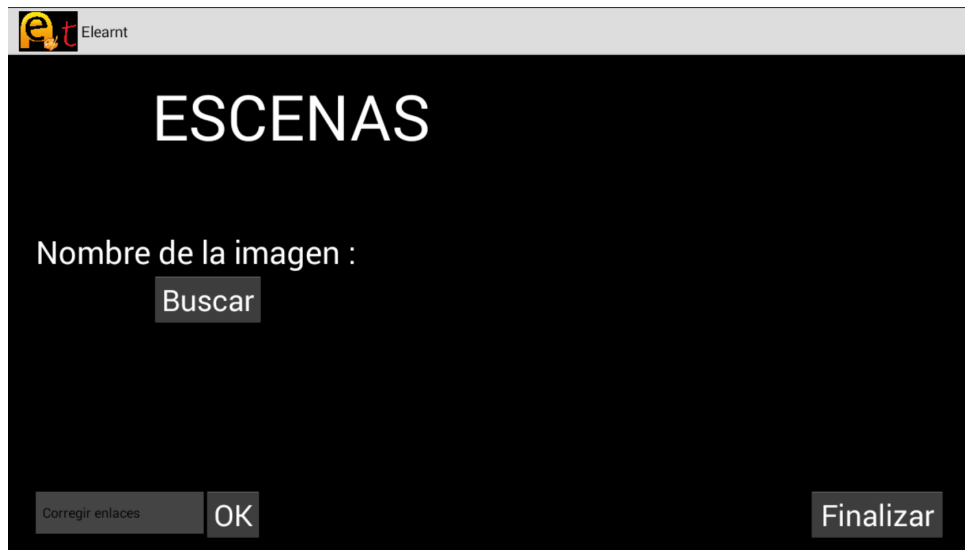


Figura B.19: Corregir enlaces.

De no existir escenas creadas le será informado *No existen escenas con las que enlazar*. Ver figura B.20.

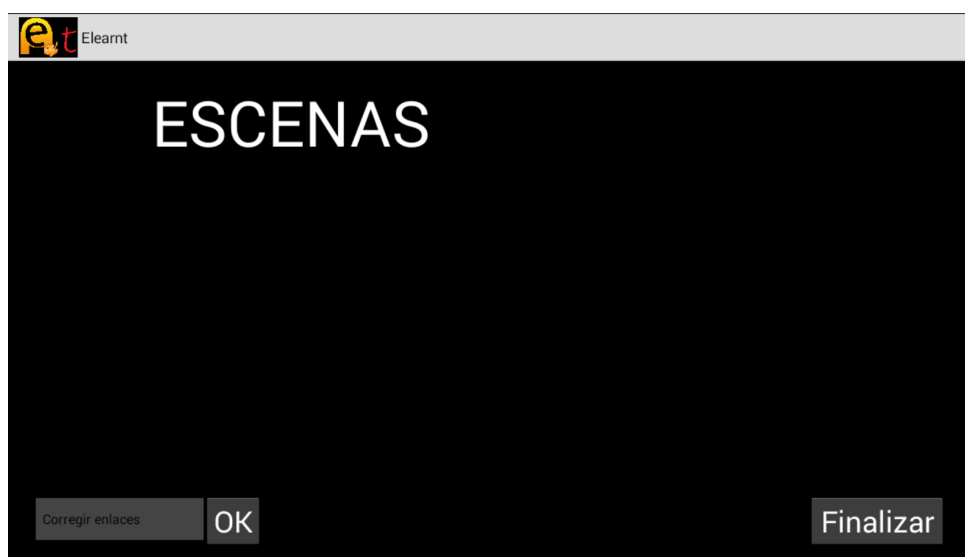


Figura B.20: Corregir enlaces (no existen escenas).

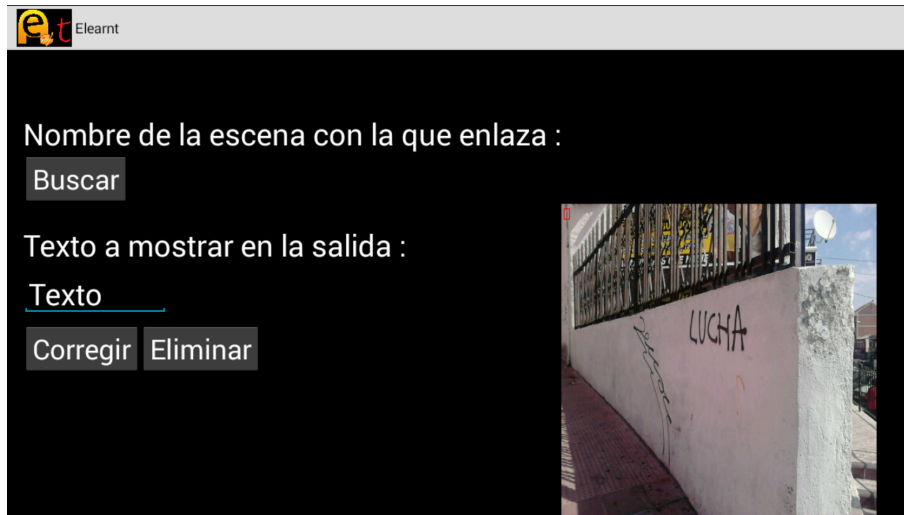


Figura B.21: Corregir salida.

A continuación se abrirá una ventana por cada enlace creado, donde se muestra la escena con el área de salida en rojo, ver figura B.21. Selecciona la escena con la que enlazar (pulse el botón **Buscar**) e introduzca el texto a mostrar en la salida B.12 (por defecto se introducirá el texto que asigno previamente). A continuación presione en **Corregir**. De no haber seleccionado una escena enlazable en la lista se mostrará el siguiente aviso *Por favor, seleccione la escena con la que enlazar en la lista*. Si por el contrario quiere eliminar el enlace, pulse en **Eliminar**. Para conservar el enlace con las propiedades actuales presione **atrás**.

Si no existiesen escenas válidas con las que enlazar solo se podrá eliminar el enlace, figura B.22.

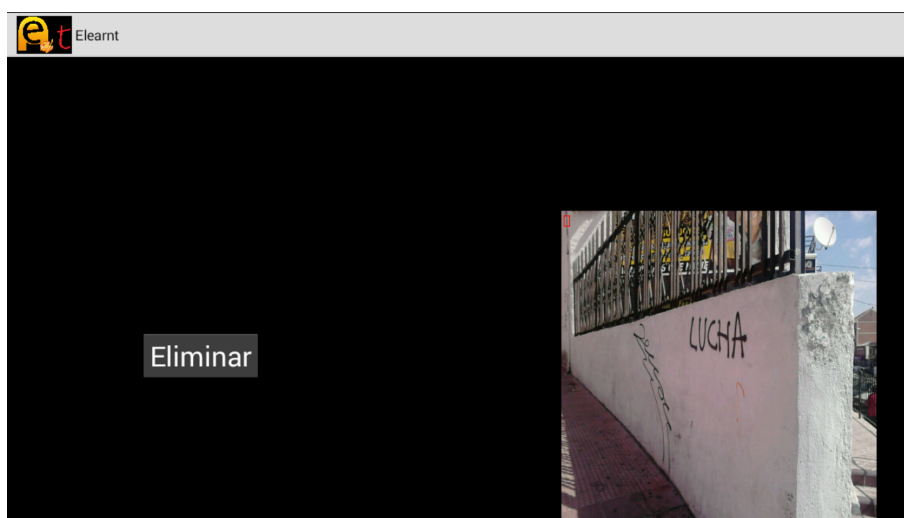


Figura B.22: Corregir salida (sin escenas enlazables).

B.14. Finalizando

Una vez finalizada la aventura dentro de la ventana de edición de escenas (B.7, B.8, B.9, B.13) presione en **Finalizar**. Si no existen enlaces erróneos pasará a la ventana donde podrá enviar los ficheros de la aventura B.16. En caso contrario hasta que no corrija todos los enlaces no se le dejará terminar (*Existen enlaces erróneos, pulse Finalizar de nuevo y corríjalos*). Idénticamente pasará si pulsa **atrás**, poniendo en el aviso *atras* en vez de *Finalizar*. Esto es así para evitar enviar aventuras inconsistentes.

B.15. Preparar el servidor

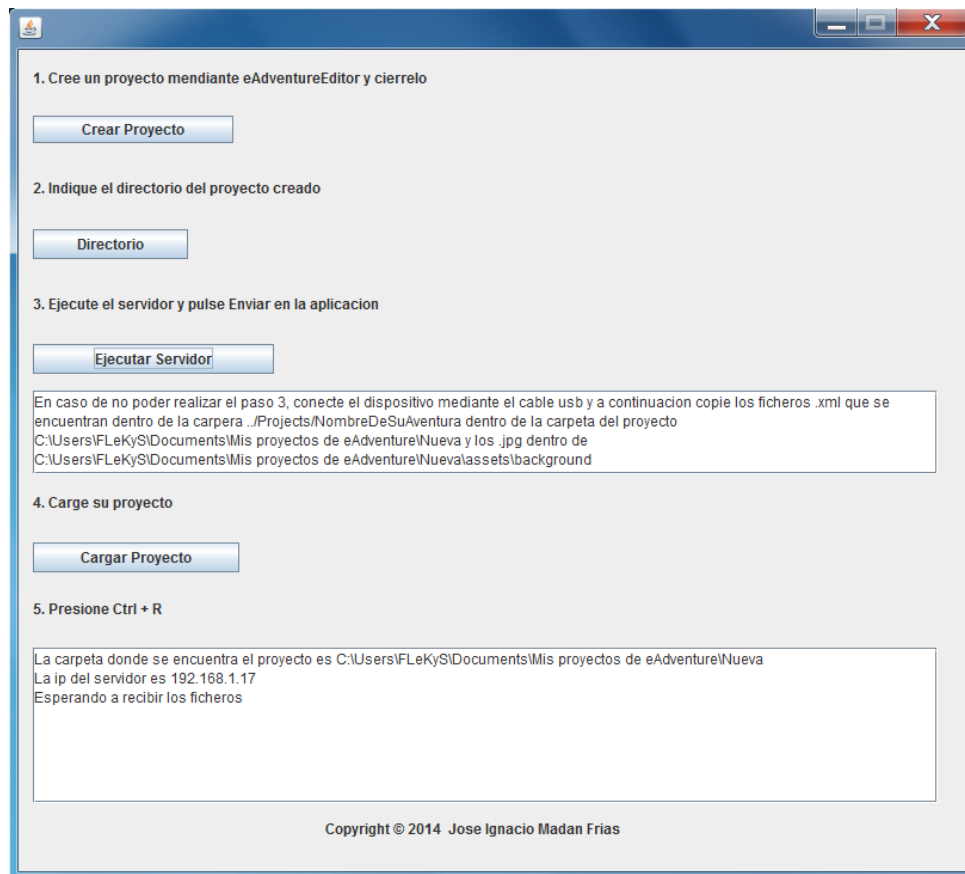


Figura B.23: Preparar el servidor.

Para garantizar el correcto funcionamiento de la puesta en marcha del servidor siga los pasos indicados teniendo en cuenta:

- Si tiene el proyecto abierto de anteriores veces, ciérrelo y vuelva a abrirlo para garantizar la actualización de las propiedades y enlaces creados.
- Para crear/cargar el proyecto, punto 1 y 4, tendrá que buscar el ejecutable *eAdventureEditor* al menos una vez.
- Por defecto, los directorios con los que se abrirá el buscador de archivos serán “*..\Program Files\eAdventure*” (Windows) “*/usr/bin*” (Linux) (punto 1 y 4) y “*..\Users\..\Documents\Mis proyectos de eAdventure*” (Windows) “*/home/./Projects*” (Linux) (punto 2). Si el ejecutable de *eAdventureEditor* o la carpeta donde se encuentra la aventura residen en directorios diferentes, tendrá que buscarlos manualmente.
- Si selecciona un ejecutable distinto a *eAdventureEditor* le será informado *El ejecutable no corresponde a eAdventureEditor*.
- Para ejecutar el servidor (punto 3) ha de haber seleccionado un directorio que contenga la aventura (*No existe la aventura en el directorio seleccionado*).
- Siempre podrá abrir la aventura manualmente a través de la interacción con su sistema operativo (buscando el archivo ejecutable de la aventura o abriendo *eAdventureEditor* y cargándolo).
- En caso de que tenga que copiar los ficheros manualmente mediante USB, consulte la sección B.2 para conocer más detalles sobre el directorio donde se encuentra la aventura.
- Probado su correcto funcionamiento en los sistemas operativos *Windows 7* y *Ubuntu 14.04*

B.16. Enviar Ficheros

Una vez preparado el servidor B.15 introduzca la dirección IP que le aparece y presione en **Enviar**. Puede observar cómo van llegando los ficheros en el servidor (se mostraran sus nombres). Cualquier error durante el envío le será informado, ya sea desde el servidor o desde la aplicación *Error al conectar con el servidor*, *Error al transferir el fichero*, etc.

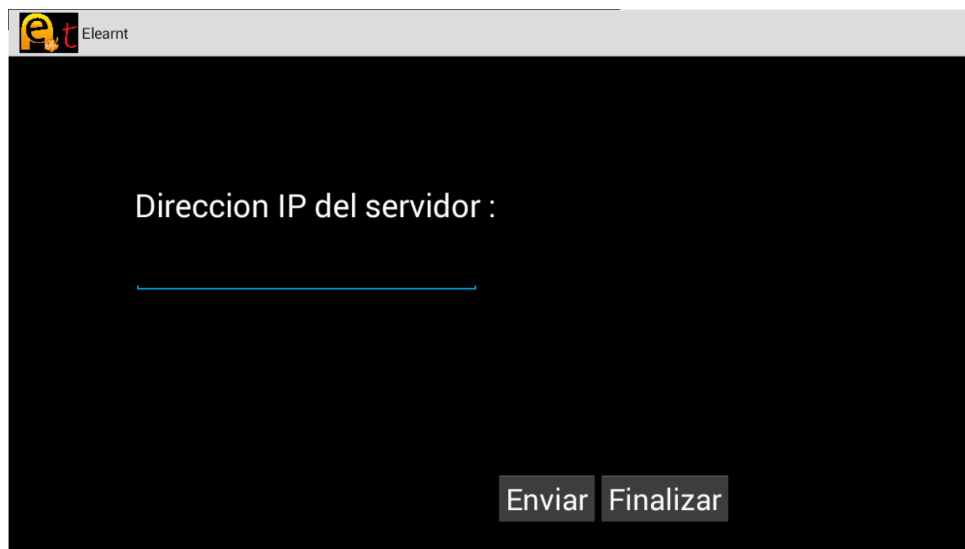


Figura B.24: Enviar ficheros.

Bibliografía

[1] Aventura conversacional.

http://es.wikipedia.org/wiki/Aventura_conversacional

[2] Aventura gráfica.

http://es.wikipedia.org/wiki/Aventura_grafica

[3] Aprendizaje electrónico.

http://es.wikipedia.org/wiki/Aprendizaje_electronico

[4] eAdventure.

<http://e-adventure.e-ucm.es/>

[5] ¿Qué es Java?

[http://es.wikipedia.org/wiki/Java_\(lenguaje_de_programacion\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programacion))

[6] Comparación entre la API de Android y Java.

http://en.wikipedia.org/wiki/Comparison_of_Java_and_Android_API

[7] ¿Qué es Android?

<http://es.wikipedia.org/wiki/Android>

<http://www.xatakandroid.com/sistema-operativo/que-es-android>

[8] Desarrollo iterativo.

http://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente

[9] ¿Qué es Canvas?

[http://en.wikipedia.org/wiki/Canvas_\(GUI\)](http://en.wikipedia.org/wiki/Canvas_(GUI))

[10] Draw with a Canvas on a surfaceView

<http://developer.android.com/intl/es/guide/topics/graphics/2d-graphics.html>

[11] Eventos táctiles en Android.

<http://developer.android.com/reference/android/view/MotionEvent.html>

[12] Actividades en Android.

<http://developer.android.com/guide/components/activities.html>

[13] ¿Qué es Java Swing?

[http://es.wikipedia.org/wiki/Swing_\(biblioteca_grafica\)](http://es.wikipedia.org/wiki/Swing_(biblioteca_grafica))