



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SISTEMA DE APOYO PARA VISITAS A MUSEOS UTILIZANDO DISPOSITIVOS
MÓVILES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN

FELIPE IGNACIO ALBERTO CAÑAS SABAT

PROFESOR GUÍA:
NELSON ANTRANIG BALOIAN TATARYAN

MIEMBROS DE LA COMISIÓN:
JOSÉ MIGUEL PIQUER GARDNER
EDUARDO SALVADOR GODOY VEGA

SANTIAGO DE CHILE
ABRIL 2011

Resumen

Tradicionalmente, las visitas a museos consisten en un recorrido a través de salas, mientras se observan obras, posiblemente contando con información adicional en forma de texto impreso, o pistas de audio en los casos más avanzados. Últimamente, con el avance y la masificación de los dispositivos móviles, se ha abierto una oportunidad para usar la tecnología para añadirle dinamismo e interacción a las visitas a museos, cosas de las cuales hoy en día carecen.

El objetivo de este trabajo fue el de abordar esta carencia, para lograr ofrecerle a las visitas una experiencia de museo más completa. Para lograr este objetivo, se abordó la interacción de la visita con el museo, y la manera en que la información le es presentada, con el objetivo de crear un canal bidireccional que le permita al visitante «hablarle» de vuelta al museo.

Se desarrolló un sistema para generar y visualizar contenido que represente virtualmente a un entorno de museo. El sistema cuenta con un editor de contenido, hecho para ser usado por el curador del museo, y con un visualizador de contenido, hecho para acompañar a las visitas. Se crearon estas dos aplicaciones ya que, aunque el objetivo era mejorar la experiencia de las visitas, era un paso imprescindible facilitarle la creación de contenido al curador del museo.

El editor permite acceder y modificar la información del proyecto. En un proyecto de museo se definen capturas, salas, y rutas. Se incluye una simulación 3D de las salas, mediante la cual es posible posicionar las obras para tener una vista previa de la exhibición en distintas configuraciones. Es posible asociarle datos adicionales a una obra, más allá de los básicos.

El visualizador fue desarrollado para el sistema operativo móvil Android, de forma que las visitas puedan utilizarla durante sus recorridos. Esta aplicación también permite enviar comentarios acerca de las obras, por lo cual se desarrolló un servidor que los recibe, para luego compartirlos con todos los visitantes que accedan a esa obra de allí en adelante.

Se llegó a un conjunto de aplicaciones que forman parte de un prototipo que funciona y cumple con los objetivos con que se desarrolló. Las pruebas de desarrollo se hicieron con una colección del Museo de Arte Contemporáneo, el cual se consultó para guiar el desarrollo con conocimiento de primera mano, en vez de apoyarse sólo en teoría.

El desarrollo deja una base sobre la cual se puede construir para aprovechar aún más las oportunidades que brinda la tecnología para el campo de los museos. Al haber sentado una fundación, se deja abierta la posibilidad de que se continúe el trabajo para encontrar métodos más novedosos y efectivos para mejorar la experiencia de una visita a un museo.

Agradecimientos

Gracias a mis padres, por darme esta oportunidad.

Gracias a mis hermanas, por su apoyo.

Gracias a mis amigos, por ayudarme en el camino.

Gracias a Paula, por distraerme cada vez que lo necesitaba.

No lo hubiera logrado sin ustedes.

Índice general

Resumen	I
Agradecimientos	II
Índice de figuras	V
Índice de cuadros	VII
1. Introducción	1
1.1. Nomenclatura	2
2. Motivación	4
2.1. Escenarios posibles	6
3. Objetivos	7
3.1. Objetivo general	7
3.2. Objetivos específicos	7
4. Trabajo relacionado	9
5. Metodología	12
6. Desarrollo del trabajo	15
6.1. Planificación	15
6.2. Formato de los proyectos	16
6.3. Aplicación para el curador	29
6.4. Aplicación para la visita	43
7. Resultados	50
7.1. bnac-editor	50
7.2. bnac-visitor	59
8. Conclusión	70
8.1. Trabajo futuro	72
Bibliografía y referencias	74
Apéndices	77

A.	Archivo meml.xsd	77
B.	Archivo info.xsd	82

Índice de figuras

6.1. Relación entre <i>Objeto Cultural</i> y <i>Captura</i>	16
6.2. Ejemplo de uso de Objeto Cultural y Captura.	17
6.3. Entidades con que se realizó el desarrollo.	20
6.4. Las entidades creadas para especificar las Salas.	24
6.5. Relación entre las Rutas y las Capturas.	24
6.6. Las entidades para soportar traducciones.	25
6.7. Entidades para soportar información adicional.	25
6.8. Parte de un archivo de especificación de proyecto.	27
6.9. Parte de un archivo de páginas de información.	28
6.10. Ejemplo de la estructura de un proyecto.	29
6.11. Las clases principales del editor.	31
6.12. Casos de uso que incluyen Objetos Culturales y Capturas.	32
6.13. Diagrama de clases de Objetos Culturales y Capturas.	32
6.14. Casos de uso que incluyen Salas y Capturas Posicionadas.	33
6.15. Diagrama de clases de Salas y sus clases asociadas.	34
6.16. Algoritmo usado para generar los Muros y Capturas bidimensionales en la simulación 3D.	35
6.17. Algoritmo usado para generar los pisos y techos de las salas en la simulación 3D.	36
6.18. Casos de uso que incluyen Puertas y Vínculos.	37
6.19. Diagrama de clases de Puertas y Vínculos.	37
6.20. Casos de uso que incluyen las Rutas.	38
6.21. Diagrama de clases de Ruta.	39
6.22. Casos de uso que incluyen Traducciones.	39
6.23. Manera de acceder a una traducción pasando por los tres diccionarios.	40
6.24. Casos de uso que incluyen Páginas de Información.	41
6.25. Diagrama de clases de las Páginas de Información.	41
6.26. Casos de uso para la visita que incluyen las Capturas y las Rutas.	45
6.27. Casos de uso para la visita que incluyen las Páginas de Información.	46
6.28. Casos de uso para la visita que representan su aporte de información.	47
6.29. Diagrama de flujo por las pantallas de la aplicación para visitas.	49
7.1. Ventana principal de la aplicación.	52
7.2. Vista de simulación.	52
7.3. Vista de simulación, con los elementos modificados.	53
7.4. Comparación de una captura seleccionada y no seleccionada.	53
7.5. Vista de capturas.	54

7.6. Diálogo de creación de captura.	54
7.7. Vista de capturas con nueva adición.	54
7.8. Vista de salas.	55
7.9. Vista de salas con una sala nueva.	55
7.10. Vista de plano.	56
7.11. Vista de plano mostrando como se mueve una esquina.	56
7.12. Vista de plano mostrando como se divide un muro.	57
7.13. Vista de navegación.	57
7.14. Diálogo de creación de un nuevo vínculo y de sus dos puertas.	58
7.15. Vista de rutas.	60
7.16. Diálogo de ruta.	60
7.17. Diálogo de ruta luego de presionar el botón «Agregar captura».	60
7.18. Herramienta de traducciones.	61
7.19. Herramienta de traducciones mostrando una traducción.	61
7.20. Herramienta de traducciones, en el caso de una ruta.	61
7.21. Herramienta de información.	62
7.22. Entrada de una página de información.	62
7.23. Entrada inexistente para una página en cierto idioma.	63
7.24. Entrada de una página de información mostrando la traducción a otro idioma.	63
7.25. Pantalla de inicio de la aplicación para la visita.	65
7.26. Pantalla de selección de ruta.	65
7.27. Pantalla de captura.	66
7.28. Elementos de la interfaz de la pantalla de capturas para que el usuario com- parta con el museo.	66
7.29. Comentario antes de enviar.	67
7.30. Comentario después de enviar.	67
7.31. Pantalla de información.	69
7.32. Pantalla de información mostrando una traducción a otro idioma.	69

Índice de cuadros

6.1. Atributos de un Objeto Cultural.	18
6.2. Atributos de una Captura.	19
6.3. Atributos de un Cuadro.	19
6.4. Atributos de una Captura Posicionada.	21
6.5. Atributos de una Puerta.	22
6.6. Atributos de un Punto de Partida.	23
6.7. Atributos de una Sala.	23

Capítulo 1

Introducción

Junto con el avance de la tecnología, ha ido cambiando la manera en que las personas acceden a la información. Hoy en día es muy fácil establecer canales de comunicación entre dos partes, por los cuales fluyen datos tanto uni- como bidireccionalmente, y tanto entre partes que se conocen como desconocidas entre sí. Es más, tener disponible una infraestructura que permita este tipo de intercambio de datos es actualmente simple, desde el punto de vista técnico, y poco exigente desde el punto de vista de los recursos necesarios. Esta combinación permite potenciar situaciones en las cuales ha sido tradicionalmente difícil entablar una comunicación efectiva entre dos partes, usando tecnología para encontrar soluciones a este tipo de problemas. Un contexto que se puede ver beneficiado de este tipo de soluciones es el de los museos, en particular las visitas a ellos.

Tradicionalmente, al visitar un museo, un visitante se ve limitado a recorrer las exposiciones presentadas en las distintas habitaciones. De esta situación, toda la información a la cual está expuesto el visitante es la que brinda el mismo objeto en exposición. Esto tiene la desventaja de que le es difícil a alguien no entendido en el tema de la exposición sacar provecho de ésta, ya que es probable que se le escapen detalles, o bien que no esté al tanto de los temas a los cuales se relaciona algún objeto en particular. Finalmente, es fácil que su experiencia no sea tan completa como podría ser si tuviera más información a su disposición.

A esta situación se le han encontrado soluciones parciales. Por ejemplo, es común que objetos de exposiciones cuenten con textos explicativos. Esta solución es simple, pero bastante limitada, ya que la cantidad de información que se puede disponer de esta manera es bastante acotada. Esto ocurre por varias razones, la principal de ellas siendo que la acción de leer el texto le quita la atención al visitante del objeto al cual describe. Adicionalmente, en el caso de niños o personas que no hablen el idioma en el cual está escrita la descripción, el texto explicativo no le será de utilidad alguna al visitante.

Otra solución común es la de entregarle a los visitantes dispositivos que reproducen pistas de audio, las cuales contienen descripciones de lo que está en exposición. Esta estrategia le permite al visitante concentrarse en el objeto mientras escucha la descripción, y es más fácil añadir pistas de audio con traducciones, lo cual elimina algunas desventajas presentes en el

caso del texto. Esta solución, sin embargo, comparte con el caso del texto que en ciertas situaciones, especialmente en aquellas con exhibiciones con muchos objetos, al visitante le es difícil identificar el objeto que se está describiendo. Adicionalmente, al utilizar guías de audio ocurre que los visitantes se sienten aislados de sus acompañantes [Woodruff et al., 2001a].

A pesar de sus desventajas, ambas estrategias efectivamente cumplen con brindar una mayor cantidad de información al visitante. Sin embargo, comparten una desventaja adicional que es clave: ambas son vías completamente unidireccionales de comunicación.

Al contar únicamente con una vía de comunicación de naturaleza unidireccional, el museo no tiene manera de saber si la información disponible es efectivamente útil para el visitante. Las visitas, a su vez, no tienen posibilidad de comunicarle al museo ningún tipo de comentario acerca del contenido que se les está brindando. De esta forma, el museo deja de recibir comentarios importantes sobre los cuales podría modificar la información que se le entrega a cada visita, o incluso modificar las mismas exposiciones.

Una solución que contempla un canal bidireccional de comunicación entre museo y visitante es aquella que incluye el uso de dispositivos móviles como herramienta que asiste durante las visitas. Los dispositivos móviles tienen las propiedades de ser portátiles y capaces de conservar conectividad mientras se desplazan dentro del recinto [Houser et al., 2002], lo cual le permite al visitante tener una conexión directa al museo durante todo su recorrido dentro de él. De esta forma, lo único que limita su interacción con el museo es la funcionalidad que éste último haga disponible mediante el dispositivo.

El uso de dispositivos móviles también abre oportunidades para aprovechar las ventajas de la educación móvil [Soloway et al., 2001]. Se le hace posible al museo presentar la información de manera distinta y novedosa al visitante. El uso de un dispositivo móvil permite salir de una situación en que la información sólo se presenta de forma estática, a una en la cual ésta se puede ofrecer de forma interactiva y con elementos multimedia, para así hacer la experiencia más interesante para el usuario, y la transferencia de conocimiento más eficaz.

Al aprovechar las oportunidades aquí descritas, se espera que el visitante se involucre en un mayor grado en las exhibiciones disponibles, con el fin de que su experiencia sea más enriquecedora. Es más, al utilizar dispositivos móviles, también existe la posibilidad de que los visitantes establezcan vías de comunicación entre ellos. Ya que la interacción con compañía es algo importante para la mayoría de los visitantes [Woodruff et al., 2001c], el abarcar este aspecto de la visita significa afectar una faceta importante de la experiencia.

1.1. Nomenclatura

Para evitar confusión, a continuación se definen algunos términos que se usarán en el presente informe. La lista incluye tanto siglas como términos a los cuales se les ha asignado un significado específico dentro del contexto del trabajo aquí descrito.

- El puesto de *curador* tiene asignadas varias responsabilidades dentro de un museo. Una de éstas, y la más relevante en el contexto de este informe, consiste en la selección y organización de piezas para formar exhibiciones, con el objetivo de tener cierto efecto sobre los visitantes.
- *Proyecto* se refiere a los archivos, directorios y recursos digitales de distinto tipo, dispuestos de forma tal que se pueda reconstruir un ambiente de museo virtual a partir de ellos.
- Ya que se consideran dos tipos principales de usuarios, curadores y visitas, al hablar de *distintos usuarios*, se hace referencia a los distintos tipos de usuarios, no a usuarios individuales.
- Un término común para los dispositivos móviles es el de *PDA*, por el inglés *Personal Data Assistant*, lo cual se traduce a «asistente personal digital».
- Se le llama XML a una manera de estructurar información en un archivo. Sus iniciales representan el inglés *eXtensible Markup Language*, lo cual en español significa «lenguaje de marcas extensibles».
- El formato XSD se utiliza para especificar la estructura que debe seguir un documento en formato XML. Esta sigla significa *XML Schema Definition*, o «definición de esquema XML» en español.
- MIME es una convención que ayuda a identificar el formato de un archivo, como por ejemplo si se trata de un archivo de texto plano, o una imagen en formato PNG. Sus siglas significan *Multipurpose Internet Mail Extensions*, o «extensiones multipropósito de correo de Internet».

Capítulo 2

Motivación

Dado el estado de la tecnología actual, que cuenta con dispositivos móviles con capacidad cercana (y a veces superior) a computadoras de escritorio, con sistemas de comunicaciones fácilmente desplegables, y que la complejidad técnica de mantener estos componentes sea baja, existe un potencial en el uso de estos elementos en el entorno de los museos que no se ha aprovechado al máximo.

La principal motivación de este trabajo es la de habilitar un canal de comunicación mediante el cual la información pueda fluir tanto desde el museo hacia las visitas como en la dirección opuesta. Al permitirle al visitante comunicarse con el museo, se le da la oportunidad de tomar un papel activo durante su visita, como también la de dejar una huella que otros puedan seguir posteriormente. Esto hace un contraste drástico con medios tradicionales, como pistas de audio, las cuales limitan a los visitantes al rol pasivo de simplemente escuchar.

Al darle tanto más participación como cierta trascendencia a los visitantes, se busca motivarlos a inmiscuirse en el mundo del museo, lo cual ayudaría a transmitirles los conceptos que se desean impartir mediante las obras, lo cual es lo que le interesa al curador. Por otra parte, contar con la opinión y otros comentarios de los visitantes le hace más fácil al museo modificar progresivamente las exposiciones y la forma en que las presenta para que los visitantes dejen el museo habiendo vivido una experiencia más completa, durante la cual habrán disfrutado tanto como aprendido más.

Al contar con dispositivos móviles, el museo dispondría de un medio en el cual es posible presentar el contenido de manera dinámica ante el usuario. La ventaja de esto tiene un fin más allá que simplemente impresionar al usuario. La libertad que brinda la oportunidad de presentar contenido mediante una PDA se puede usar de maneras que son imposibles de realizar físicamente. Por ejemplo, mientras que un catálogo con información acerca de las exhibiciones tiene su contenido limitado a lo que lleva impreso en sus páginas, una PDA es capaz de conectarse a un equipo remoto y descargar información adicional cada vez que el usuario la requiera. Con una PDA se tiene la posibilidad de mostrar contenido animado, lo cual es imposible en un catálogo. Estas ventajas se pueden utilizar para mejorar la experiencia del usuario.

El contenido expuesto y su presentación son determinados por los curadores de los museos. Un sistema de apoyo para visitas a un museo sería inútil si no se considerara su participación. Un buen sistema debe estar diseñado para que los curadores puedan fácilmente crear el contenido que luego será mostrado a los visitantes. La participación de los curadores es crucial, ya que ellos definen el contenido que se le hace disponible a las visitas, como también la manera en que se les presenta dicha información. Cualquier contenido adicional, como sugerencias de recorridos o información relacionada a las obras y colecciones, también es provisto por ellos. Sería imposible lograr los beneficios esperados sin contar con su participación activa.

El valor agregado que le brinda este sistema al curador a la hora de crear contenido es lo maleable que éste se torna al trabajar con una representación virtual, y no una física. Al manipular obras siempre se corre el riesgo de que resulten dañadas, mientras que la posibilidad de poder visualizar una exposición mediante una simulación permite probar distintas configuraciones sin poner en peligro las obras que incluye la colección. Es más, también existe la posibilidad de planificar el posicionamiento de las obras de una colección aún sin contar con ellas, como en el caso de una colección que se planea recibir como préstamo en una futura fecha. Al contar con una simulación virtual, se puede planificar con anticipación, en literalmente cualquier momento y lugar.

Se hizo mención del concepto de rutas dentro del museo. Éstas son sugerencias de un orden a seguir para recorrer una o más colecciones. Las rutas son otra herramienta para enriquecer la experiencia de las visitas. Mediante éstas, es posible dejar en evidencia factores como contrastes, tendencias, similitudes entre distintas obras, su evolución a través del tiempo, o bien cualquier otro concepto que el curador desee impartir. Las rutas pueden también ser creadas considerando distintos tipos de visitantes. Por ejemplo, una ruta para niños, otra para estudiantes de arte.

El usar una PDA como medio de presentación de contenido le permite al museo incluso mostrar contenido que no está físicamente en una exposición. Esto se puede dar por distintas razones, por ejemplo, algunas piezas pueden estar siendo restauradas, o bien simplemente las piezas están en otro museo. Esto perjudicaría la experiencia de un visitante, ya que no lograría ver la exhibición completa. Al usar un dispositivo móvil, se logra complementar parcialmente este déficit.

Un aspecto importante de las visitas a museos que se puede ver mejorada por el uso de dispositivos móviles es la interacción entre los visitantes que recorren juntos el museo. En el caso de guías de audio, es usual que las visitas se quejen porque el uso del dispositivo lo aísla de su acompañante. Se han hecho estudios con métodos para que esto no ocurra [Woodruff et al., 2001c], y se ha encontrado que las PDA son dispositivos que ofrecen la posibilidad de lidiar con este problema.

Un tipo particular de PDA cada vez más utilizado hoy en día es el *smartphone*, o *teléfono inteligente* [Stimac, 2010]. El hecho de que muchos de los visitantes cuenten con su propio dispositivo libera al museo de la responsabilidad de contar con un uno para cada visita, ya que se podría hacer posible la instalación de la aplicación para visitantes en sus propios dispositivos. Es más, la aplicación podría disponerse de forma que los dueños de *smartphones* puedan incluso descargar la aplicación desde sus hogares, pudiendo revisar los datos sin

necesidad de visitar el museo.

2.1. Escenarios posibles

Para complementar la motivación, se enuncian algunos casos de uso de un sistema como el que se describe en este documento.

- El curador del museo podrá crear simulaciones de exhibiciones, teniendo control sobre elementos como la forma de la sala de exhibición y la posición de las obras. De esta forma, se le dará facilidad para visualizar un montaje sin necesidad de armarlo en la realidad, pudiendo así considerar un abanico de opciones rápidamente y con poco esfuerzo.
- El curador del museo tendrá la posibilidad de hacer disponibles exhibiciones virtuales para el sistema de dispositivos móviles. Adicionalmente, podrá asociar distintos recursos a elementos de éstas, tal como archivos multimedia, referencias a otras obras relacionadas, información acerca del autor, etc.
- El curador tendrá la posibilidad de asociar traducciones a las exhibiciones creadas, para así beneficiar a muchas visitas que no entiendan el lenguaje con el cual fue creado el contenido originalmente.
- El curador podrá definir rutas para recorrer durante una visita. Estas rutas son sugerencias que ordenan las exhibiciones de tal manera que, a juicio del curador, las visitas sean más enriquecedoras que simplemente recorrer el museo en un orden aleatorio.
- Dado que distintos visitantes tienen distintas razones para visitar el museo, se podrán definir rutas para tipos específicos de visitantes. Ejemplos son estudiante de arte, niños en visita organizada por su escuela, visita casual, etc.
- Se podrán almacenar los comentarios relacionados a distintas obras, hechos por los visitantes, los cuales podrán ser vistos por otros visitantes. Se podrán también agregar «sugerencias», de tal forma que una visita pueda influir en el recorrido de una futura visita.
- Los visitantes podrán marcar las obras que más les guste, pudiendo así fácilmente identificar las obras más populares.
- Los visitantes podrán acceder a información audiovisual acerca de las obras o algún tema relacionado, en formatos como vídeos o juegos, por ejemplo.
- Sería posible incluso integrar el sistema del museo con alguna red social de forma que la interacción que un visitante tenga con el museo se vea reflejada en ella. De esta manera se podrían compartir, por ejemplo, comentarios, o resultados de alguna actividad.

Capítulo 3

Objetivos

3.1. Objetivo general

El objetivo general de este trabajo es el de diseñar un sistema de apoyo de visitas a museos que le permita a los visitantes tener una experiencia más completa durante su visita, tal que salgan de ella habiendo aprendido más acerca de los temas de las distintas exposiciones a las que hayan acudido. Este objetivo engloba tanto interacción del sistema con las visitas como con personal del museo, siendo los primeros los que *consumen* contenido, y los segundos los que lo *producen*. Por esta razón, se deben satisfacer los requerimientos tanto del personal del museo como de las visitas.

Los pilares del presente trabajo son dos; uno es la comunicación bidireccional, que permite que los visitantes puedan dar comentarios y observaciones acerca de lo que les es presentado, lo cual le permite al personal del museo saber qué y hasta cierto punto cómo cambiarlo. El segundo consiste en las posibilidades de utilizar distintos métodos de presentación para el contenido, las cuales existen gracias a los dispositivos con los cuales se trabaja.

3.2. Objetivos específicos

- I. Diseñar una aplicación mediante la cual el personal del museo pueda crear y modificar exhibiciones y rutas de recorrido de forma simple.
- II. Diseñar la aplicación que tomará el rol de interfaz con el visitante. Esta interfaz debe presentar la información definida por el personal del museo, como también permitirle al visitante ingresar distintos tipos de datos de manera definida por el museo.
- III. Incluir funcionalidad en la aplicación la cual permita que el usuario interactúe con representaciones virtuales de la exposición. Dichas interacciones deben ser definidas

por el personal del museo.

- IV. Incluir funcionalidad que le haga posible a las visitas interactuar con otras visitas, no necesariamente acompañantes. Las interacciones permitidas serán definidas por el personal del museo.
- V. Identificar los datos relevantes acerca de los distintos objetos culturales para incluir estos atributos al diseñar el formato de almacenamiento de éstos.
- VI. Identificar la información proveniente de las visitas que le interesa al personal de los museos, para facilitar el ingreso de ésta mediante la interfaz presentada a los visitantes.
- VII. Diseñar esquemas de almacenamiento por lo menos los siguientes elementos,
 - Objetos culturales
 - Salas de exposición
 - Rutas de recorrido
- VIII. Definir un procedimiento de captura de datos de objetos culturales que se pueda repetir por usuarios básicos de herramientas de software.
- IX. Recopilar información de un museo para verificar que el procedimiento llevado a cabo sea útil en una aplicación real, y no sólo en el prototipo.
- X. Definir un esquema de comunicación y almacenamiento de la información aportada por las visitas, manteniendo su asociación con las obras de tal forma que se puedan recuperar en un contexto apropiado.
- XI. Seleccionar el sistema y/o tipo de dispositivo para el cual implementar la aplicación para las visitas. Dado que existe una variedad de tanto dispositivos como sistemas operativos móviles, es imposible hacer una implementación para cada combinación. Se debe tomar una decisión con respecto a lo más conveniente.
- XII. Incluir funcionalidad que permita traducir contenido a distintos idiomas, para lograr beneficiar a usuarios que no entiendan el idioma en el cual se produjo el contenido original.

Capítulo 4

Trabajo relacionado

Este trabajo trata con dos áreas distintas de investigación. El área principal con la cual trata es la de visitas a museos guiadas con un dispositivo móvil. La segunda, que no es el énfasis, sin embargo cumple un rol importante, es el área de almacenamiento de datos de objetos culturales.

En el campo de guías electrónicas de museos, la influencia principal fue el trabajo presentado en [Woodruff et al., 2001c], donde se le asigna un rol conversacional a la guía. Donde normalmente una guía tendería a «monopolizar» la atención mediante largas descripciones, el tipo de guías propuestas en este trabajo se insertan como un interlocutor más en la conversación con los visitantes. Esto permite que las visitas puedan aportar su propio conocimiento, como también interactuar entre ellos, y así no sentirse aislados.

El mismo trabajo está relacionado con otros llevados a cabo por los mismos autores, los cuales comparten además la forma en que le presentan los datos a los visitantes. En los trabajos [Woodruff et al., 2001b] y [Woodruff et al., 2001a] se usa el mismo sistema de guías, el cual le presenta el contenido al usuario en forma de fotografías de lo que él o ella ve en ese lado de la habitación que están recorriendo. Se muestran entrevistas a usuarios en las cuales éstos elogian esta decisión, opinando que hace mucho más intuitiva la interacción con la guía. Esta es una práctica que se desearía emular, para lograr que el usuario entienda lo más intuitivamente posible la forma de operar del dispositivo.

En el trabajo [Aoki and Woodruff, 2000] se abarca el tema de selección de objetos. En éste se discuten los distintos métodos que se pueden seguir para permitirle al usuario seleccionar un ítem, o bien permitir que la guía automáticamente seleccione un elemento, basado en distintos criterios. Este trabajo propone un sistema concreto con hasta tres criterios a considerar al seleccionar un objeto. Qué criterios considerar depende del contexto en el cual se moverán los usuarios, y es la labor del desarrollador escoger los criterios de tal forma que la selección de elementos sea lo más fácil y menos molesto posible para la visita.

Una manera de diseñar las distintas partes de la aplicación guía es discutida en el trabajo [Abowd et al., 1997]. En este trabajo se propone un esquema en el cual se identifican

distintas labores particulares dentro de la guía, las cuales se deberían aislar y tratar por separado. Argumentan que este método ayuda a hacer las guías modulares y reutilizables, además de mencionar que distintos módulos se pueden utilizar en distintos dispositivos para satisfacer distintas necesidades.

El estudio mostrado en [Bellotti et al., 2002] demuestra la utilidad real de las guías multimedia. Durante el estudio se trabajó con visitas de un acuario, a las cuales se les facilitó una guía digital para su uso durante su recorrido, para ser entrevistados al finalizarlo. Se les encuestó con respecto a sus percepciones con respecto a distintos aspectos de la guía, como facilidad de uso, entretenimiento, distracción, y otros. Este estudio muestra con números concretos la percepción del aporte que brindan los dispositivos móviles en este contexto.

Se muestra en [Fleck et al., 2002a] una implementación de una guía multimedia en un museo interactivo. Aunque el museo con el cual se trabajará durante este trabajo no es interactivo, las ideas expuestas en el artículo son valiosas. Se mencionan funcionalidades y formas de uso, junto con la reacción que tuvieron ante éstas los usuarios, y cómo terminaron utilizando en realidad el dispositivo. La experiencia descrita en este documento se puede utilizar en otros trabajos, aunque éstos no se lleven a cabo en museos interactivos.

Se describe en [Fleck et al., 2002b] una aplicación que registra la trayectoria de un visitante durante su recorrido en un museo, para que éste pueda acceder a ésta a través de la Internet desde su hogar. Aunque esta no es una funcionalidad que se planea agregar al trabajo propuesto, el artículo menciona conceptos que se pueden utilizar para registrar los movimientos de las visitas como datos para el museo, para que su personal tenga una idea de los objetos más o menos vistos en las exposiciones.

El tema de la interacción entre visitas que recorren juntos el museo se vuelve a tocar en el trabajo [Grinter et al., 2002], donde se describe el concepto de «escuchar a escondidas»¹. Este método le permite a un visitante en cualquier momento escuchar lo que la guía de su acompañante está exponiendo en su propia guía. Dado que este es un aspecto importante, las conclusiones de este artículo son relevantes para el presente trabajo.

El artículo [Yatani et al., 2004] muestra un sistema diseñado para niños para asistirles mientras recorren un correo. El sistema se utiliza de a pares, y se presenta como un juego donde los niños deben responder preguntas como un equipo. El grupo objetivo del presente trabajo no serán niños, sin embargo los detalles de la implementación del sistema son interesantes si se desea realizar algo parecido. Se describe, adicionalmente, el procedimiento seguido para evaluar el sistema, lo cual también resulta útil para comparar resultados.

Del trabajo relacionado que tiene que ver con el almacenamiento de datos, se puede apreciar que el artículo [Mourkoussis et al., 2003] muestra algunas estrategias de cómo guardar datos de objetos culturales para exposiciones virtuales. Sin embargo, el énfasis de este trabajo está en exposiciones utilizando realidad virtual. A pesar de esto, las técnicas que muestra de almacenamiento se pueden aplicar en otros contextos.

¹Por el inglés *eavesdrop*.

El artículo [[Skov and Ingwersen, 2008](#)] trata del comportamiento de los usuarios para acceder a información disponible en un entorno virtual. Se describen las características de la manera en que los usuarios acceden a los datos, analizando profundidad en un sólo tema tanto como la cantidad de temas acerca de los cuales averiguan.

Capítulo 5

Metodología

Una característica importante que tuvo este trabajo es que se planeaba desarrollar dos aplicaciones, una destinada al curador, y otra destinada a la visita. Por esta razón, inicialmente se decidió separar el desarrollo en dos partes, para luego realizar lo necesario para su interacción antes de finalizar el desarrollo.

El desarrollo se planeó utilizando un modelo estilo cascada, ya que se trataba de un proyecto con objetivos claros y con una baja probabilidad de cambios en los requerimientos. Los pasos que se establecieron fueron los siguientes,

1. Análisis de requisitos.
2. Diseño de la representación de datos.
3. Diseño de la aplicación del curador.
4. Implementación de la aplicación del curador.
5. Diseño de la aplicación del visitante.
6. Implementación de la aplicación del visitante.
7. Pruebas.
8. Ajustes y depuración.

Adicionalmente, en el desarrollo se utilizó el lenguaje de diagramas UML¹ para definir ciertos aspectos del sistema. Su elección se debió a que consiste en un estándar bastante completo para definir los distintos aspectos del sistema, además de ser ampliamente conocido y utilizado en el mundo.

¹<http://www.uml.org>

La primera fase consistió en una toma de requerimientos. La bibliografía y referencias aportaron valiosa información para definir una lista inicial, pero se decidió que era necesario adicionalmente contactar un museo. La razón por la cual tomó esta determinación fue la ventaja de permitir que el proyecto cuente con el aporte de una fuente que trabaje en el medio que se desea beneficiar. Adicionalmente, también se buscaba la posibilidad de solicitar datos para realizar pruebas con lo que se lograra desarrollar. Con este fin se hizo contacto con el Museo de Arte Contemporáneo. Se llevaron a cabo entrevistas con el personal del museo para determinar los primeros requerimientos que tendría el sistema a desarrollar.

Se decidió comenzar por el formato que tendrían los proyectos, para definir tanto la estructura de su directorio como los archivos que especificarían su contenido para su reconstitución por una aplicación. Se estimó éste como el tema más conveniente para solucionar primero dado que los datos que se decida guardar acerca de las exhibiciones y sus obras le darían forma tanto a la aplicación de edición de contenido como a la aplicación de visualización de éste. De esta forma, con un conjunto fijo de campos de valores asignado cada objeto del entorno virtual, se evita la necesidad de hacer grandes cambios una vez considerada una aplicación como «terminada».

Se continuó con el desarrollo de la aplicación para el curador. Esta determinación se tomó por la simple razón de que, de comenzar con la aplicación de las visitas, no se tendrían datos para hacer las pruebas. Era necesario primero tener la aplicación que genera contenido, para luego hacer pruebas con su visualización.

Se comenzó, entonces, con el diseño de la estructura de la aplicación del curador solamente. Sin embargo, dado que se contemplaba la necesidad de intercambiar datos entre las dos aplicaciones, el diseño debía considerar una separación entre los módulos de lógica de la aplicación e interfaz gráfica, de forma que fuera posible reutilizar la implementación de lectura y escritura de los proyectos a almacenamiento no volátil en la aplicación de las visitas.

Con el diseño ya terminado, se comenzó la fase de implementación. Desde un principio, el enfoque fue la de llegar a un prototipo que sea suficiente para generar contenido para las visitas, no el de tener una aplicación pulida y acabada. Este enfoque se adoptó por razones de tiempo de desarrollo, ya que se consideró más importante la implementación de la funcionalidad que la apariencia de la interfaz, o de buscar errores de programación exhaustivamente.

Cuando se tuvo la aplicación del curador en un estado tal que cumplía con los objetivos, se comenzó el desarrollo de la aplicación para las visitas. Esta fase incluyó la decisión de qué plataforma utilizar, en particular el sistema operativo sobre el cual se podría ejecutar la aplicación. Durante este paso se reutilizó lo hecho para la aplicación del curador en cuanto a lectura de proyectos de museo, por lo tanto lo que se desarrolló en este paso fue esencialmente la interfaz gráfica para el usuario, junto con los ajustes necesarios para el sistema operativo móvil.

Se debió definir el esquema con que se compartiría el contenido. Existían por lo menos dos posibilidades: la de desplegar puntos de acceso a través de las salas, mediante los cuales se podría descargar contenido; o bien la posibilidad de cargar contenido en las PDA previo a

cada recorrido. Se debió decidir qué camino seguir.

Habiendo resuelto el tema de la comunicación entre las distintas partes, se procedió a hacer pruebas para comprobar la facilidad de uso de todas las aplicaciones desarrolladas. Para esto se utilizaron los datos provistos por el museo, con los cuales se creó una exposición virtual con la aplicación del curador, para luego visualizarla utilizando la aplicación de la visita.

Finalmente se llevaron a cabo las últimas correcciones. Es decir, se corrigieron los errores de programación que perjudicaban de mayor manera la experiencia de los usuarios y se le hicieron algunas mejoras a las interfaces de los usuarios para hacer más fácil el uso de las aplicaciones.

Capítulo 6

Desarrollo del trabajo

6.1. Planificación

Los objetivos de este trabajo contemplan dos tipos principales de usuarios: el grupo de los curadores, y el grupo de las visitas. Cada tipo de usuario tiene un rol definido en el sistema, siendo los curadores los generadores de contenido, y los visitantes los que consumen dicho contenido. Esta diferencia entre los casos de uso de cada tipo de usuario justificaba el desarrollo de una aplicación para los curadores, y otra para las visitas. A grandes rasgos, y reflejando el rol de cada tipo de usuario, la funcionalidad que se necesitaba de cada aplicación debía satisfacer las tareas de generación y visualización de contenido, respectivamente.

Adicionalmente, ya que se contemplaba la creación de contenido que debía ser distribuido entre varios usuarios, era necesario idear un formato estándar con el cual definir el contenido al almacenarlo. El formato debía lograr contener completamente cada proyecto, de forma que fuera posible de forma fácil y ordenada compartir contenido. Se debía considerar tanto los recursos incluidos en cada proyecto, por ejemplo imágenes o textos, como también la estructura con la cual serían vinculados a las distintas obras.

Considerando lo anterior, se decidió dividir el proceso en distintos pasos para desarrollar cada parte. Éstas se listan a continuación,

- El formato de los proyectos,
- la aplicación para el curador, y
- la aplicación para la visita.

6.2. Formato de los proyectos

Era necesario definir un formato con el cual especificar la estructura de las distintas partes de una exhibición virtual. Este formato debería definir desde las obras que forman parte de la exhibición, hasta las salas donde son exhibidas, junto con las conexiones de sala a sala, y todo lo demás. Se determinó que lo que se necesitaba especificar era lo siguiente,

- Obras de arte, junto con datos como su nombre y autor,
- salas de exposición, es decir su forma y cómo están dispuestos los elementos que contienen,
- las conexiones entre las distintas salas, representadas como puertas,
- rutas que definen un recorrido, obra por obra,
- información adicional, como descripciones o acerca de temas relacionados, y traducciones, y
- la estructura con que se almacenarían los proyectos, y los archivos necesarios para poder reconstruirlos desde una aplicación.

Adicionalmente, se debía decidir si se utilizaría algún estándar existente para la sintaxis de estos archivos.

Obras

El diseño de la especificación de las obras se basó en el estándar propuesto por el proyecto ARCO [Mourkoussis et al., 2003].

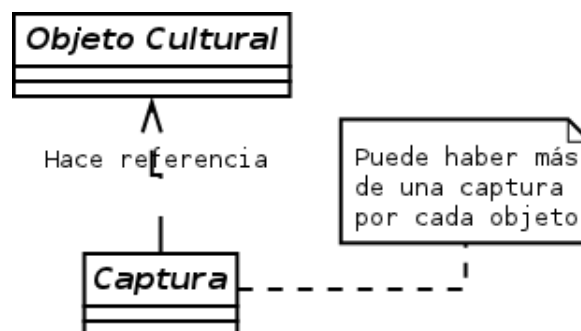


Figura 6.1: Relación entre *Objeto Cultural* y *Captura*

La figura 6.1 muestra las dos entidades básicas y la relación que existe entre ellas.

Un *Objeto Cultural* es una referencia al objeto del mundo real, conteniendo solamente información acerca de éste. Un *Objeto Cultural* no incluye una referencia a una representación

virtual del objeto al cual describe. Al contrario, una *Captura* incluye tanto una referencia a un Objeto Cultural como una referencia a una representación virtual del objeto.

Se tomó esta decisión considerando que, mientras que un objeto cultural del mundo real es único e irremplazable (dos atributos que hacen que tenga suficiente importancia como para ser incluido en una exhibición cultural), una captura de cualquier naturaleza es tanto repetible como probablemente no única. Es decir, un objeto cultural puede tener, y probablemente tiene, más de una captura. Las razones para esto varían de situación a situación, pero tres causas comunes son distintas técnicas, distintos formatos de almacenamiento, y distintos niveles de calidad.

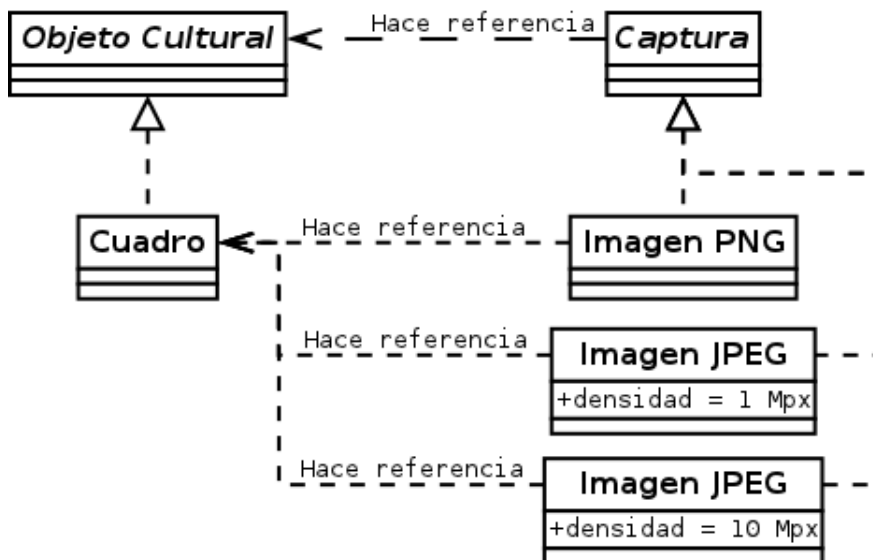


Figura 6.2: Ejemplo de uso de Objeto Cultural y Captura. El cuadro es un tipo de Objeto Cultural, mientras que las imágenes son tipos de Capturas. Se puede apreciar claramente que un mismo Objeto Cultural tiene más de una Captura asociada a él.

La figura 6.2 muestra un ejemplo concreto de esta relación. Existe un cuadro que se desea incluir en un proyecto. Primero se define su representación como Objeto Cultural, la cual se hace de tipo *Cuadro*. Posteriormente, se pueden producir cuantas capturas sean necesarias para luego asociarlas al objeto tipo *Cuadro*. En este caso hay de dos tipos; *Imagen PNG* e *Imagen JPEG*. Adicionalmente, el ejemplo muestra que existen dos capturas distintas de tipo *Imagen JPEG*. Su diferencia no es el formato, sino la calidad de la captura.

Era también necesario definir los atributos que describirían tanto los Objetos Culturales como las Capturas. Dado que era deseable mantener un nivel de simplicidad que permitiera avanzar rápidamente con el desarrollo, se escogió un conjunto mínimo de atributos con los cuales describir los elementos, con los cuales se logra manejar una cantidad de información suficiente para el alcance del trabajo.

El cuadro 6.1 muestra los atributos con que se define un Objeto Cultural. Se determinó que lo mínimo necesario para describir una obra era el autor y el título de ésta. Adicionalmente, se agregó el tipo de la obra como tercer atributo obligatorio. Finalmente, se le asignó el atributo

Cuadro 6.1: Atributos de un Objeto Cultural. Los atributos en letra negrita son obligatorios.

Nombre	Descripción	Comentarios
Autor	El nombre de la persona o grupo al que se le atribuye la autoría de la obra.	
Título	El nombre de la obra.	
Tipo	El material o materiales usados para elaborar la obra.	
Localización	El lugar en que se encuentra la obra actualmente.	Se espera la ciudad y país. En caso de que se encuentre en un museo, también se puede agregar este dato.

opcional de la localización.

Los atributos que describen a las Capturas se muestran en el cuadro 6.2. En este caso se incluyen el autor de la Captura y la fecha en que fue creada. Estos dos atributos son meramente descriptivos, mientras que los otros dos hacen referencia a archivos dentro del proyecto de museo. El atributo *Recurso* hace referencia al archivo, el cual puede ser un archivo regular o un directorio, donde se encuentran los datos que forman la captura. No se hace ninguna restricción sobre el formato o formatos de las capturas, y se le asigna la responsabilidad de cargar el recurso a la aplicación.

Dado que no se asegura que la aplicación sea capaz de visualizar el formato de algún recurso, se determinó necesario que todo recurso debería incluir una «vista previa» en un formato de imagen conocido, tales como PNG o JPEG. Adicionalmente, aunque la aplicación tenga la capacidad de visualizar el formato del recurso, es posible que dicha operación sea costosa en procesamiento o algún otro recurso del computador, por lo cual es conveniente tener una imagen para usar como vista previa.

Concretamente, en el proyecto se trabajó exclusivamente con obras de una colección fotográfica. Ya que una Obra Cultural no especifica ningún tipo concreto de obra, se creó una entidad adicional, llamada *Cuadro*. Esta entidad se creó con la función de especificar atributos de obras de este tipo, la cual también se usó para describir las fotografías.

Una entidad tipo Cuadro debe incluir todos los atributos que incluye un Objeto Cultural, y adicionalmente debe incluir la información tanto del ancho como el alto de la obra. El cuadro 6.3 muestra un resumen de esta información.

Para representar las imágenes, se creó la entidad tipo *Captura Bidimensional*, destinada a representar cualquier tipo de captura bidimensional. En este caso, no se identificó ningún atributo adicional necesario para su especificación. A pesar de esto, se decidió separar este tipo de entidad de una Captura general por la posibilidad de encontrar durante las pruebas

Cuadro 6.2: Atributos de una Captura. Los atributos en letra negrita son obligatorios.

Nombre	Descripción	Comentarios
Autor	La persona o grupo que produjo la captura.	
Fecha	La fecha de creación de la captura.	Limitada a año, mes y día.
Recurso	El archivo al que se hace referencia	Es posible que sea tanto un archivo regular como un directorio. La habilidad de cargar el recurso depende de la aplicación que se esté usando.
Vista previa	Una imagen para usar como vista previa del recurso.	Ya que existen recursos necesitan pasar por algún proceso antes de poder ser visualizados (como un modelo tridimensional), se exige una vista previa como una manera rápida de identificarlo.

Cuadro 6.3: Atributos de un Cuadro. Los atributos en letra negrita son obligatorios.

Nombre	Descripción	Comentarios
Ancho	El ancho de la obra.	Se mide en metros.
Altura	La altura de la obra.	Se mide en metros.
Nota: incluye los atributos de un Objeto Cultural.		

atributos que tuviera una Captura Bidimensional y no una general.

Finalmente, la figura 6.3 muestra las entidades con las que se realizó el trabajo, y la relación entre ellas.

Salas de exposición

Se determinó que lo más básico e indispensable de las salas son simplemente su forma y tamaño. Los atributos que se le asignaron a una entidad de tipo *Sala* para definir estas cualidades fueron una altura junto con una lista de puntos (en un plano bidimensional) para definir el lugar de las esquinas.

Para especificar las capturas incluidas en una sala, junto con su posición y orientación, se creó una entidad llamada *Captura Posicionada*. Existía la opción de establecer que las

Capturas se definen dentro de las salas donde se pretende posicionarlas, sin embargo no se tomó esta vía. La razón principal a favor de esta decisión fue consistió en que de esta manera sería incomodo y poco claro manejar los casos en que se quita una Captura de una sala para ponerla en otra, ya que el estado de «entre medio» no tendría representación. Habiendo tomado la decisión de definir las Capturas independientemente de cualquier sala, este problema se evita.

Se ganan ventajas adicionales con el método de separar la declaración de la Captura de su posicionamiento. Es posible definir Capturas en un proyecto que no se posicionan en ninguna sala, como también se hace posible posicionar una Captura más de una vez, si así se desea.

Los atributos de una Captura Posicionada definen no solamente el muro sobre el cual se encuentran y su posición sobre él, sino que también el ángulo de rotación en el eje perpendicular al muro sobre el cual se encuentra, además del factor por el cual se multiplican sus dimensiones en la visualización, si se desea hacerla más grande o más pequeña que su tamaño original. El cuadro 6.4 resume los atributos que se le asignan a esta entidad.

Las salas también pueden contener puertas que las conectan con otras salas, permitiendo simular un recorrido a través de los distintos recintos del museo. Una entidad tipo *Puerta* se puede definir dentro de una sala, y sus atributos son muy parecidos a una combinación de una Captura Bidimensional con una Captura Posicionada. Es decir, los atributos de una puerta tienen como función describir el tamaño y posición de la puerta.

Una Puerta no incluye una referencia a su destino. Se tomó esta determinación de diseño considerando, nuevamente, que una simulación virtual no tiene por qué verse restringida a las leyes del mundo real. Dentro de la simulación, una Puerta puede ser el destino de más de una Puerta (incluso puede ser el destino de sí misma), o puede tener como destino un lugar no necesariamente relacionado con otra puerta.

Para reflejar la libertad de poder definir el destino de una Puerta un lugar completamente aleatorio, definido sólo por el usuario que se encuentra creando el contenido, se crearon las entidades llamadas *Puntos de Partida*. Éstas describen un destino, al cual un usuario dentro de la simulación es «transportado». Su función es determinar la posición y la orientación con la que el usuario queda después de ser transportado desde otro lugar a través de la simulación.

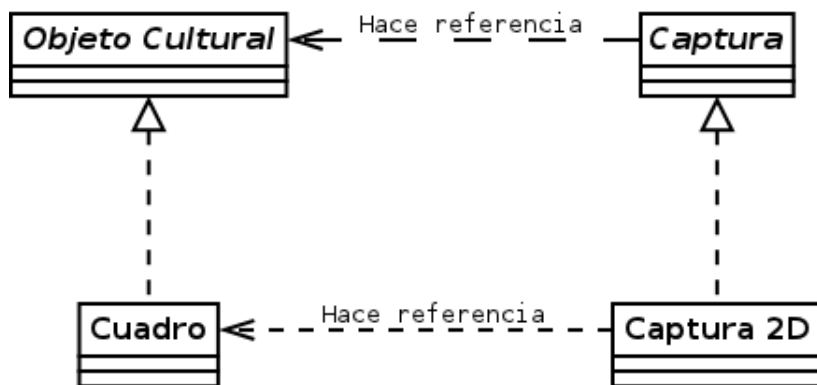


Figura 6.3: Entidades con que se realizó el desarrollo.

Cuadro 6.4: Atributos de una Captura Posicionada. Los atributos en letra negra son obligatorios.

Nombre	Descripción	Comentarios
Muro	El índice, comenzando de 0, del muro sobre el cual se encuentra.	El índice de los muros incrementa en sentido horario.
Eje x	El desplazamiento en el eje horizontal.	Se mide en metros.
Eje y	El desplazamiento en el eje vertical.	Se mide en metros.
Ángulo	El ángulo de rotación en el eje perpendicular al muro.	Medido en grados.
Escala	El factor por el cual se multiplican sus dimensiones para cambiar su tamaño en la visualización.	Se define el tamaño «original» de una captura, en el cual su Escala tiene el valor de 1,0, como el tamaño especificado en la descripción del Objeto Cultural que representa.

Ya que un mismo Punto de Partida puede ser destino de más de una Puerta, para representar este tipo de relación se creó la entidad llamada simplemente *Vínculo*. Una entidad de este tipo define una conexión entre una Puerta y un Punto de Partida, o bien entre dos Puertas. En el caso de una conexión entre una Puerta y un Punto de Partida, se podría decir que ésta es unidireccional ya que es imposible seleccionar un Punto de Partida. Al contrario, en el segundo caso el vínculo es bidireccional, ya que ambas Puertas pueden ser seleccionadas para indicar el deseo de transportarse al otro punto de la conexión.

Los cuadros 6.5 y 6.6 muestran los atributos de las Puertas y Puntos de Partida, respectivamente. Los Vínculos son sólo un contenedor de una referencia a una Puerta y un Punto de Partida, por lo cual no necesita atributos adicionales.

Finalmente, el cuadro 6.7 muestra los atributos de una Sala, mientras que la figura 6.4 muestra la relación entre una Sala y los elementos que pueden formar parte de ella.

Rutas

Era necesaria una manera de especificar rutas, para lo cual se creó la entidad *Ruta*.

Una Ruta consiste simplemente en un conjunto ordenado de obras, las cuales se visitan en orden de principio a fin. La complejidad de las rutas consistía en que existe más de una manera de hacer referencia a una obra. Es decir, existen las entidades de Obras Culturales, Capturas y además Capturas Posicionadas, y era necesario determinar cuál o cuales entidades

Cuadro 6.5: Atributos de una Puerta. Los atributos en letra negrita son obligatorios.

Nombre	Descripción	Comentarios
Muro	El índice, comenzando de 0, del muro sobre el cual se encuentra.	El índice de los muros incrementa en sentido horario.
Ancho	El ancho de la puerta.	Se mide en metros.
Altura	La altura de la puerta.	Se mide en metros.
Eje x	El desplazamiento en el eje horizontal.	Se mide en metros.
Eje y	El desplazamiento en el eje vertical.	Se mide en metros.
Ángulo	El ángulo de rotación en el eje perpendicular al muro.	Medido en grados.
Escala	El factor por el cual se multiplican sus dimensiones para cambiar su tamaño en la visualización.	

serían a las cuales las Rutas harían referencia.

Se tomó la decisión de hacer que las Rutas hagan referencia a las entidades tipo *Captura*. Las razones por las cuales se hizo esto se explican a continuación.

Se descartaron los Objetos Culturales ya que se determinó que las Rutas solamente debían incluir elementos con algún recurso que visualizar. Los Objetos Culturales no cumplen con esta restricción. Era deseable que se pudiera visualizar todo elemento de una ruta porque se estimó que, de no ser así, las rutas no tendrían sentido, ya que se alejarían demasiado de la naturaleza de un recorrido dentro de un museo.

Igualmente, se descartaron las Capturas Posicionadas por la fragilidad que esta relación le daría a las rutas. La «fragilidad» se refiere a la alta probabilidad de que una Captura Posicionada sea eliminada de un proyecto, en comparación con la baja probabilidad de que una Captura sea eliminada del mismo. No es deseable que las rutas se tengan que modificar cada vez que se modifica una Sala, ya que ambas cosas no tienen razón para estar tan estrechamente relacionadas. Por esta razón, se determinó que las rutas harían referencia a las mismas Capturas.

La figura 6.5 muestra la simple relación entre las Capturas y las Rutas. Es importante destacar que la relación entre estas dos entidades es una *agregación*, puesto que las Capturas no necesitan depender de una Ruta para existir de manera válida.

Cuadro 6.6: Atributos de un Punto de Partida. Los atributos en letra negrita son obligatorios.

Nombre	Descripción	Comentarios
Eje x	Posición en el eje x .	Se utiliza el mismo marco de referencia que las esquinas de la sala. Se mide en metros.
Eje y	Posición en el eje y .	Se utiliza el mismo marco de referencia que las esquinas de la sala. Se mide en metros.
Ángulo	El ángulo de rotación en el eje vertical (perpendicular al suelo) en que queda mirando al llegar al Punto de Partida.	Se mide en grados.

Cuadro 6.7: Atributos de una Sala. Los atributos en letra negrita son obligatorios.

Nombre	Descripción	Comentarios
Esquinas	Puntos en un plano bidimensional que definen la posición de las esquinas.	Es una lista de mínimo tres puntos. Las dimensiones se miden en metros.
Altura	La altura de la sala.	Se mide en metros.

Información adicional y traducciones

Ya que la información contenida en un Objeto Cultural era bastante limitada y acotada exclusivamente a datos precisos, se detectó la necesidad de contar con un mecanismo para agregar información «miscelánea» a los proyectos. La finalidad de este tipo de información es la de hacer disponibles datos adicionales acerca de las obras o de temas que se relacionen a ellas, ya que difícilmente se puede predecir todo tema que un curador desee asociar a una u otra obra para crear campos destinados a ello en un Objeto Cultural.

Asimismo, era necesaria una manera de definir traducciones para todo el contenido generado. Es común que un porcentaje importante de las visitas a museos sean turistas, los cuales muchas veces no manejan el idioma nativo de la región en la cual se encuentra el museo. Para que toda el contenido generado no sea inutilizado por esta barrera idiomática, resulta muy importante abordar el tema de las traducciones.

El método que se adoptó para definir traducciones de los datos definidos en Objetos Culturales, Capturas, y el resto de las entidades definidas anteriormente, fue el de simplemente formar una lista de valores ordenados por su idioma y el identificador del elemento al cual hacen referencia.

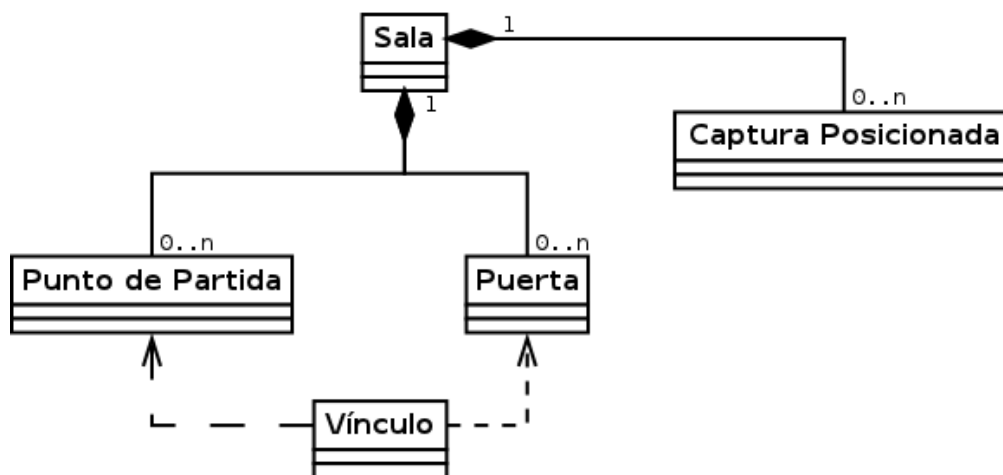


Figura 6.4: Las entidades creadas para especificar las Salas. Se incluyen también las relaciones entre ellas.



Figura 6.5: Relación entre las Rutas y las Capturas. Notar que la relación es sólo de agregación, no de composición.

Se definió el elemento llamado *Cadenas*, al cual se le asignó la función de contener todas las traducciones. Dentro de éste, se determinó que se agregarían entidades llamadas *Elemento*, las cuales tendrían la referencia al elemento para el cual almacenan traducciones. Las traducciones mismas se agregarían en entidades llamadas *Entrada*, las cuales además de contener el texto, también mantendrían un valor que indique el idioma de éste.

Para especificar el lenguaje se decidió adoptar el uso del estándar ISO 639-1¹ por ser un sistema simple de usar, además de ya ser usado en numerosos contextos distintos en el mundo.

La figura 6.6 muestra las entidades utilizadas para las traducciones, y las relaciones que existen entre ellas.

Se consideró apropiado tratar la información adicional por separado del resto de los datos, ya que ésta es completamente opcional. La manera de tratarla resultó muy parecida a la manera de tratar las traducciones, como se explica a continuación.

¹http://www.infoterm.info/standardization/iso_639_1_2002.php

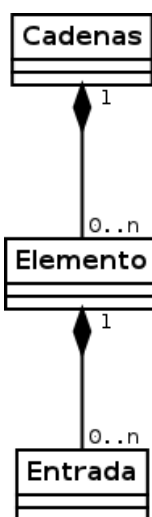


Figura 6.6: Las entidades para soportar traducciones.

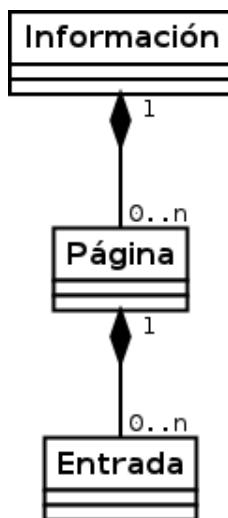


Figura 6.7: Entidades para soportar información adicional.

Se creó una entidad llamada, simplemente, *Información*. Ésta contiene toda la información adicional que se le desee agregar al proyecto, la cual está distribuida en *Páginas*. Las páginas se definieron como contenedores de algún texto, junto con una imagen opcional. El texto, a su vez, en realidad es contenido dentro de una *Entrada*, la cual especifica el lenguaje del texto que contiene. Nuevamente, se usa el estándar ISO 639-1 para especificar el lenguaje.

Las Páginas pueden, adicionalmente, definir una Página siguiente, para poder armar una secuencia que hable de algún tema en particular. Lo que se esperaba conseguir con esto era permitir la distribución de información acerca de un mismo tema en varias páginas, para no forzar a los generadores de contenido a crear páginas interminables. Esto resultaría incómodo en un dispositivo con un limitado espacio de visualización, como es el caso de las PDA.

La figura 6.7 muestra las entidades que especifican los recursos de información adicional, junto con sus relaciones.

Sintaxis

Se tomó la decisión de utilizar el formato XML para los archivos de definición de los proyectos. Esta determinación se tomó dada la facilidad de uso que brinda XML, ya que existen muchas implementaciones para recuperar la información guardada en este formato. Adicionalmente, existen validadores de sintaxis, lo cual es una gran ventaja, ya que hace innecesario implementar un mecanismo de validación de sintaxis para archivos de configuración. Basta con elaborar un archivo XSD, el cual se le entrega a un validador XML junto con el archivo por validar, para verificar su validez.

Los apéndices [A](#) y [B](#) muestran los esquemas que deben seguir las definiciones de los proyectos y las páginas de información adicional.

Estructura

En cuanto a la estructura de los proyectos, era deseable que el formato de los proyectos sea lo menos compleja posible, y que tuviera un mínimo de componentes obligatorias. Ya que se trataba de una primera versión de un sistema, era muy posible que se encontraran características que se podrían mejorar durante las pruebas. Un formato complejo y rígido es más difícil de adaptar a los cambios que uno simple y versátil.

Se determinó que los proyectos deberían estar contenidos en un sólo directorio, el cual se consideraría la raíz del marco de referencias de los recursos incluidos en el proyecto. No se puso ninguna restricción con respecto a la estructura a seguir; los autores de contenido se les dejó libertad absoluta para disponer de los recursos que incluyan en el proyecto.

El único elemento obligatorio que debe llevar todo proyecto es su archivo de descripción, el cual debe encontrarse en el primer nivel del directorio raíz del proyecto, y debe llevar el nombre de `project.xml`. Como ya se mencionó, éste debe seguir la sintaxis del estándar XML, y debe seguir el esquema definido para archivos de definición de proyectos, incluido en el apéndice [A](#).

Los archivos `project.xml` siguen un esquema creado siguiendo lo descrito anteriormente, al cual se le llamó «MEML», por el inglés de «Lenguaje de Marcado de Ambientes de Museo»². Dentro de este archivo se especifica toda la información acerca del museo virtual, con la excepción de las páginas de información adicional. Ya que éstas últimas son una parte opcional de un proyecto, se decidió manejarlas en un archivo separado.

La figura [6.8](#) muestra un ejemplo breve de una sección de un posible archivo de especificación de proyecto. Se muestra la definición de una obra de tipo Cuadro y una captura tipo Captura Bidimensional (*Painting* y *Capture2D* en inglés, respectivamente). Se muestra también la manera en que se expresan sus atributos, y como la captura hace referencia a la obra, mediante identificadores únicos en el documento.

²*Museum Environment Markup Language.*

```

<painting id="obj-secando_redes">
  <author>Antonio Quintana</author>
  <title>Secando Redes</title>
  <type>Impresión lambda montada en trovisel</type>
  <width>1.25</width>
  <height>1.25</height>
</painting>

<capture2D id="cap-secando_redes" ref="obj-secando_redes">
  <author>Felipe Cañas</author>
  <date>2009-11-15</date>
  <resource type="image/png">
    img/secando_redes.png
  </resource>
  <preview type="image/png">
    img/secando_redes.png
  </preview>
</capture2D>

<strings>
  <elem id="obj-secando_redes">
    <entry lang="en">
      <pair name="Author" value="Antonio Quintana"/>
      <pair name="Title" value="Drying Nets"/>
      <pair name="Type"
        value="Print mounted on plastic"/>
    </entry>
  </elem>
</strings>

```

Figura 6.8: Parte de un archivo de especificación de proyecto.

La manera en que se especifican los identificadores y las referencias en la especificación de los proyectos es siempre de la manera que se muestra en la figura 6.8, utilizando los atributos «id» y «ref» para los identificadores y las referencias, respectivamente. Esta forma de definir identificadores es útil, ya que se le puede asignar uno a cualquier elemento del proyecto. De esta manera es como se hace referencia desde las traducciones. En el ejemplo se puede apreciar como se le hace referencia a la obra, y se le asignan traducciones en inglés a sus atributos.

Otro método importante que se puede ver empleado en el ejemplo es la manera de hacer referencia a archivos dentro del proyecto. Se mencionó anteriormente que el proyecto tendría todos los archivos incluidos dentro de sí en un mismo directorio, el cual actuaría como el directorio raíz del proyecto. En este caso, si el directorio raíz se encuentra en el siguiente directorio,

/home/user/Museo/

entonces la imagen de vista previa debería encontrarse en la siguiente ruta,

```
/home/user/Museo/img/secando_redes.png
```

Adicionalmente, se puede ver que las capturas tienen un atributo que indica su formato. Para especificar esta propiedad se decidió utilizar la convención MIME. Se tomó esta decisión ya que éste es un método usado casi universalmente, lo cual ayuda a hacer el formato más familiar para cualquiera que entre en contacto con él. En el caso del ejemplo, se expresa el tipo de las imágenes como *image/png*.

Con respecto a las páginas de información adicional, ya se mencionó que éstas serían especificadas en un archivo aparte de la especificación del resto del proyecto. Se determinó que el nombre del éste archivo sería simplemente `info.xml`, y deberían también existir directamente bajo el directorio raíz del proyecto. La definición del formato que deben seguir, es decir el esquema, se encuentra en el apéndice [B](#).

```
<page id="antonio_quintana">
  <image src="extra/antonio_quintana.png">

  <entry lang="en">
    <title>Antonio Quintana</title>
    <description>
      Antonio Quintana was a chilean photographer.

      He was one of the main organizers of the
      "The Face of Chile" exposition.
    </description>
  </entry>

  <entry lang="es">
    <title>Antonio Quintana</title>
    <description>
      Antonio Quintana fue un fotógrafo chileno.

      Fue uno de los principales organizadores
      de la exposición «El Rostro de Chile».
    </description>
  </entry>
</page>
```

Figura 6.9: Parte de un archivo de páginas de información.

La figura [6.9](#) muestra una sección de un archivo `info.xml` a modo de ejemplo. Se muestra una página con una imagen y dos entradas, una en el idioma inglés y otra en español. Se puede apreciar el uso de el estándar ISO 639-1 para especificar el idioma de cada entrada, y también el hecho de que no se traduce solamente el contenido de la página, sino que también el título. La imagen que se incluye en la página es opcional, y su ruta se define de igual manera que en el caso de las Capturas, como se muestra anteriormente en la figura [6.8](#).

Finalmente, en la figura 6.10 se muestra la estructura del directorio de este proyecto de ejemplo.

```
/
|-- project.xml
|-- info.xml
|-- img/
|   '-- secando_redes.png
|
'-- extra/
    '-- antonio_quintana.png
```

Figura 6.10: Ejemplo de la estructura de un proyecto.

Teniendo definida la manera de especificar proyectos y sus elementos, se siguió con el desarrollo de la aplicación para el curador, la cual se encargaría de generar contenido utilizando el formato ya descrito.

6.3. Aplicación para el curador

El nombre con que se bautizó el proyecto completo, es decir todas las aplicaciones necesarias para generar y visualizar contenido, fue «**bnac**». Ya que la aplicación del curador debía ser la encargada de crear y editar contenido, se le llamó apropiadamente «**bnacEditor**».

La aplicación para el curador se desarrolló utilizando el lenguaje de programación Java, y la interfaz gráfica se desarrolló usando la biblioteca Swing. Para trabajar con archivos en formato XML, se utilizó la biblioteca de analizadores sintácticos SAX. Se utilizaron solamente dos bibliotecas adicionales, Java 3D³ y Vecmath⁴. Fue necesario incluir estas dos bibliotecas ya que era necesario trabajar con entornos virtuales tridimensionales, justamente el objetivo con que éstas fueron desarrolladas.

Esta sección se divide en dos partes para hacer más ordenada su lectura. Se divide en el diseño general y el detallado. El primero explica los patrones generales que se usaron a través de la aplicación, mientras que el segundo indica para cada parte del desarrollo las decisiones particulares que se tomaron.

Diseño general

El estilo de interfaz por el cual se decidió fue el de una *interfaz controlada de documento único*, o *SDI controlada* por sus siglas en inglés⁵. La interfaz planeada consistía en una ventana con

³<http://java3d.dev.java.net>.

⁴<https://vecmath.dev.java.net/>.

⁵*Single Document Interface*.

el menú de opciones, desde la cual el usuario podría abrir ventanas adicionales para controlar distintos aspectos del proyecto de museo. Así, cada proyecto tendría sus propias ventanas, pero el control de la aplicación en su totalidad estaría concentrado en una ventana solamente, sin importar el número de proyectos abiertos.

A la clase principal de la aplicación, la que tendría el control total sobre ésta y todos sus elementos, se le llamó `BnacEditor`. Se creó también una clase que representara a los proyectos individuales, a la cual se le dio el nombre de `Project`.

Ya que la clase `BnacEditor` representa la ventana controladora de la aplicación, se decidió utilizar el patrón del *singleton* para ésta. No así con la clase `Project`, ya que es necesaria una instancia para cada proyecto con que se esté trabajando.

Las ventanas de cada proyecto serían representadas por clases que hereden de la clase `ProjectView`. Ésta es una clase abstracta, que no implementa casi ninguna funcionalidad. Su objetivo es ser la base de todas las ventanas que muestren algún aspecto del proyecto. A estas ventanas se les denominó «vistas», lo cual explica el nombre de la clase base.

De manera casi idéntica al caso de las vistas, se creó la clase `ProjectDialogue` para manejar los diálogos que se le deberían mostrar al usuario en distintas situaciones. La diferencia que se determinó entre una vista y un diálogo es básicamente el tiempo de vida de cada interfaz. En el caso de las vistas, su tiempo de vida puede que sea todo el periodo durante el cual el usuario interactúa con el proyecto. En el caso de los diálogos, su tiempo de vida es mucho más breve, ya que su objetivo es usualmente el de ser la interfaz para que el usuario ingrese ciertos datos, o bien mostrarle algún mensaje o alguna información, tras lo cual el diálogo sería desechado.

Un aspecto importante de la aplicación consiste en los comandos con los cuales el usuario modifica el proyecto. Se decidió, como regla general, que al usuario se le debería dar la confianza de modificar el proyecto a su antojo, sin en ningún caso temer arruinar su trabajo por haber llevado a cabo alguna acción irreversible. Con este fin, se determinó que los comandos de la aplicación contarían con las ya comunes opciones de *deshacer* y *rehacer*.

Se creó la clase abstracta `ProjectCommand` como representación básica de un comando. Esta clase determina que todo comando debe implementar no solamente el método que lleva a cabo su modificación particular, sino que también debería implementar un método para deshacer su modificación. Este método sería el que se utilizaría en caso de que el usuario se arrepienta de haber ejecutado alguna modificación en el proyecto. Así, se le darían al usuario las ya comunes opciones de deshacer y rehacer una acción, llamadas usualmente *undo* y *redo*, respectivamente.

La figura 6.11 muestra las relaciones entre las clases abstractas (vistas, diálogos y comandos) y el proyecto al cual pertenecen, representado por una instancia de la clase `Project`. Adicionalmente, la figura muestra la relación entre `Project` y la clase de la ventana principal, `BnacEditor`.

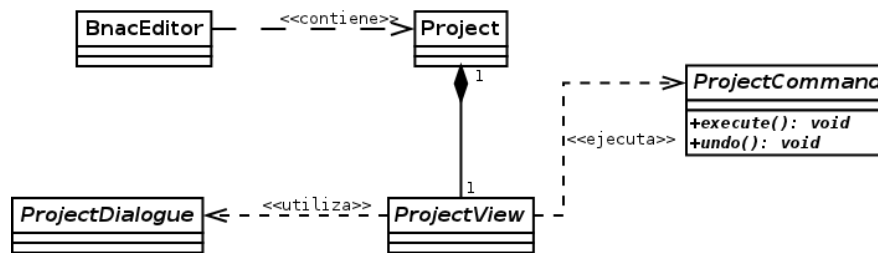


Figura 6.11: Las clases principales del editor. Notar que las vistas, los diálogos y los comandos son clases abstractas.

Diseño detallado

La funcionalidad que se debía implementar para esta aplicación era bastante extensa, por lo cual ésta se abordó por partes. Cada parte se ocupó de toda la interacción necesaria con cierto tipo de elementos, la cual usualmente consistía en las operaciones básicas de creación, edición y eliminación. Las partes en las cuales se dividió el desarrollo de la aplicación se listan a continuación,

- Objetos Culturales y Capturas,
- Salas de museo,
- Puertas y Vínculos,
- Rutas,
- Traducciones, y
- Páginas de Información.

Las partes se abordaron en este orden porque de esta manera resultaba más fácil y conveniente el proceso de pruebas de consistencia de la aplicación. Por ejemplo, la fase de las Salas incluía implementar la funcionalidad de visualizar Capturas. Hubiera sido imposible probar esta funcionalidad si al implementar las Salas, no se hubieran contado con Capturas disponibles para agregarle. Lo mismo para los Vínculos y las Salas, etc.

Objetos Culturales y Capturas

Como primer paso, se debía implementar la funcionalidad que le permita al usuario crear Objetos Culturales y Capturas para incluirlas en las exhibiciones virtuales. La figura 6.12 muestra los casos de uso que se definieron con respecto a estos elementos.

Anteriormente se definió qué atributos son los que definen tanto un Objeto Cultural como una

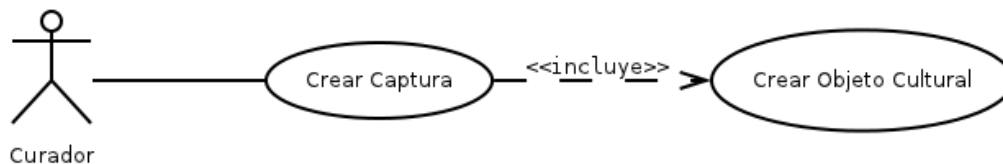


Figura 6.12: Casos de uso que incluyen Objetos Culturales y Capturas. Notar que para el curador la creación de Objetos Culturales es completamente transparente. El usuario sólo está consciente de que crea una Captura.

Captura. Siguiendo estas reglas, se crearon las clases `CObject`⁶ y `Capture`, para representar a los Objetos Culturales y las Capturas, respectivamente. Aunque se habló de ellas como clases abstractas, se decidió implementarlas como clases concretas para agilizar el desarrollo.

Adicionalmente, ya que se pretendía trabajar con fotografías para hacer las pruebas, se implementaron las clases `Painting` para representar los Cuadros, y `Capture2D` para representar las Capturas Bidimensionales. Estas clases tienen como atributos adicionales el alto y ancho, lo necesario para una visualización apropiada.

La figura 6.13 muestra un diagrama de clases con lo aquí descrito. Se muestran los atributos de cada clase, siguiendo lo establecido anteriormente en las figuras 6.1, 6.2 y 6.3.

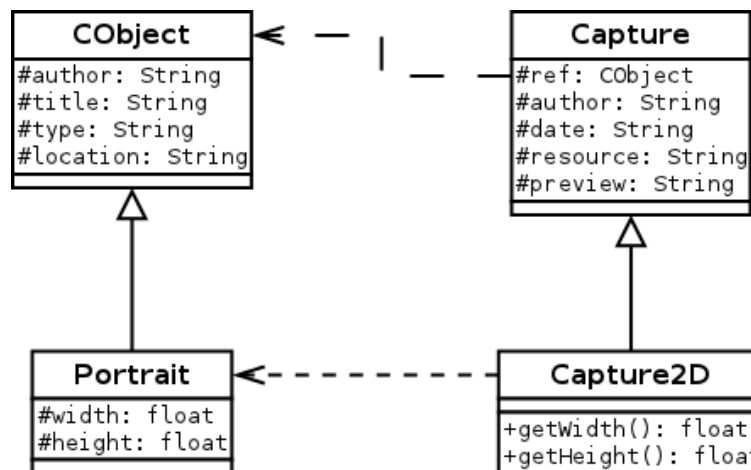


Figura 6.13: Diagrama de clases de Objetos Culturales y Capturas. Se incluye también el Cuadro y la Captura Bidimensional, las cuales heredan del Objeto Cultural y de la Captura, respectivamente.

Para hacer la interacción con el usuario lo más simple posible, se tomó la decisión de ocultarle el concepto de Objetos Culturales. Ya que éstos son elementos imposibles de visualizar, se consideró que al usuario le confundiría tener dos entidades que representen una misma obra (Objeto Cultural y Captura). Por lo tanto, al usuario tiene la posibilidad de crear Capturas, mediante lo cual automáticamente se crea un Objeto Cultural con los datos ingresados.

Se tomó la decisión de no implementar la edición ni eliminación de Capturas, ya que estas operaciones introducen una complejidad no trivial a la aplicación. Ya que los archivos a los

⁶No se le llamó `Object` para no entrar en conflicto con la clase `java.lang.Object`.

que se hace referencia como Recurso y Vista Previa desde las Capturas se almacenan dentro del directorio del proyecto, es posible que el usuario no disponga de copias de éstos en otras ubicaciones en su disco, o en equipos remotos. Si se permitieran la edición o eliminación de una Captura, se corre el peligro de que el usuario elimine o modifique de forma irreversible uno de estos archivos. Ya que esto va en contra de la regla general de no incluir acciones irreversibles, las opciones eran o no incluir estas opciones, o hacerlas de tal forma que no fueran irreversibles. Ya que se estimó que una solución alternativa tomaría mucho tiempo y no aportaría verdaderamente a los objetivos del desarrollo, se decidió omitir tanto la edición como la eliminación de Capturas, y por asociación, también en el caso de los Objetos Culturales.

Salas de museo

Ya contando con representaciones para los Objetos Culturales y para las Capturas implementadas, se prosiguió con la implementación de la representación de Salas de museo. La figura 6.14 muestra los casos de uso que se definieron para esta etapa.

La funcionalidad que se pretendía implementar en esta fase constaba de aquella necesaria para crear, modificar, eliminar y visualizar Salas, como también la funcionalidad requerida para llevar a cabo las mismas operaciones con las Capturas Posicionadas. Dado que la interacción que se planeaba era la de navegación a través de una representación virtual tridimensional de las salas y su contenido, en esta fase se tuvieron que solucionar todos los requerimientos de la aplicación que tenían que ver con el tema de gráficos 3D.

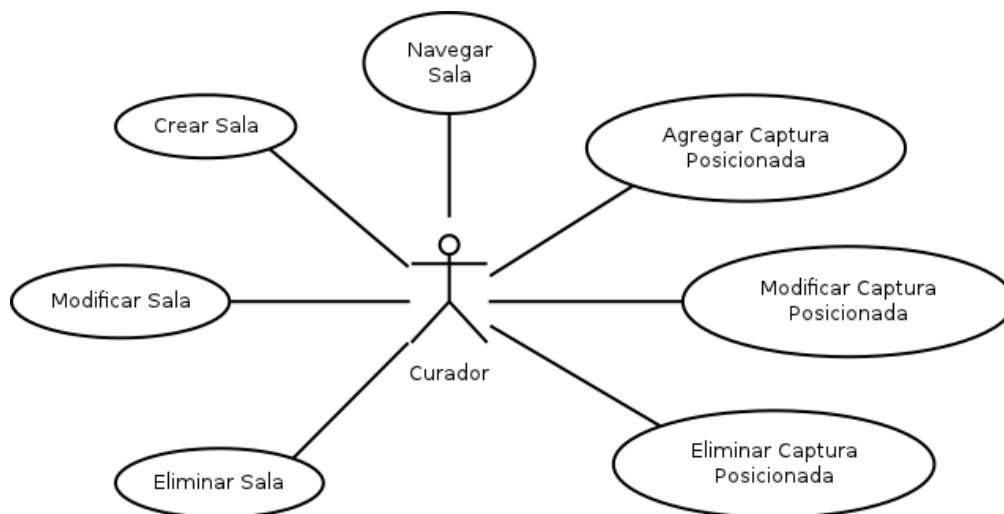


Figura 6.14: Casos de uso que incluyen Salas y Capturas Posicionadas.

Se creó la clase `Hall` para representar a las Salas. Éstas contienen un conjunto de instancias de la clase `Wall` para representar sus muros, los cuales tienen como objetivo contener las Capturas Posicionadas, representadas a su vez por la clase `PlacedCapture`. Adicionalmente, se creó la clase `StartingPoint` para representar los Puntos de Partida de cada Sala.

La figura 6.15 muestra las clases descritas anteriormente. Las clases se muestran con sus

atributos más importantes. Se muestra también la relación entre la clase `PlacedCapture` y la clase `Capture`. La clase `javax.vecmath.Point2f`, representada por su nombre corto en el diagrama, es parte de la biblioteca `Vecmath`, y no hace más que representar un punto con coordenadas en un plano, almacenadas en formato de punto flotante.

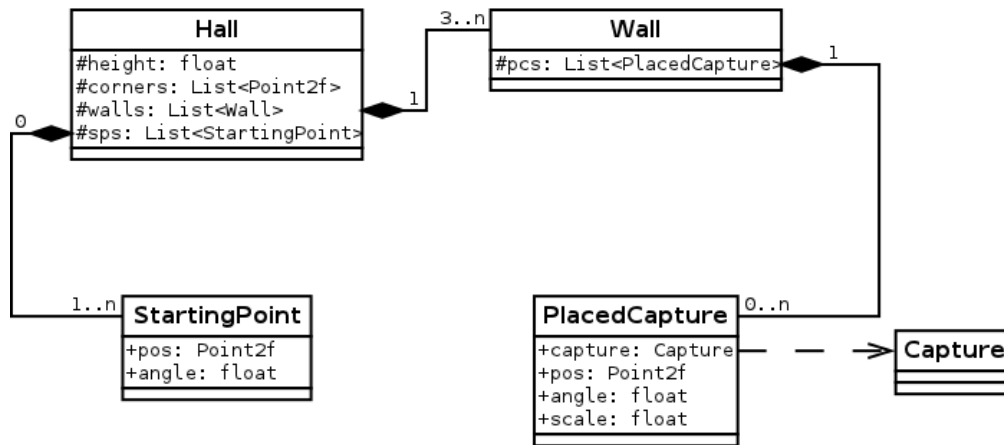


Figura 6.15: Diagrama de clases de Salas y sus clases asociadas. Se muestran adicionalmente los atributos más importantes de cada uno.

Muchas de los atributos de las clases tienen como explicación directa lo establecido durante la etapa de elaboración del formato de los proyectos. Por ejemplo, se definió que los Puntos de Partida deben contener una posición y un ángulo para indicar su orientación, lo cual es exactamente lo que muestra el diagrama. Sin embargo, otros atributos necesitan una explicación más explícita.

La clase `Wall` se creó, como ya se mencionó, para contener las instancias de `PlacedCapture`. Sin embargo, ésta no es su única función. Esta clase se creó para facilitar la generación de los entornos virtuales. El Muro almacena la información de la posición y la dimensión que toma en el espacio de simulación. Se obtiene esta información desde la lista de esquinas de la Sala y de su altura, lo cual se podría llevar a cabo cada vez que se «arma» la Sala. Sin embargo, se estimó que tomar esta vía generaría un código más confuso (y, por tanto, más difícil de mantener), además de hacer a la aplicación más costosa desde el punto de vista de operaciones de procesador, lo cual era innecesario.

Aunque se permite que las Salas no cuenten con ningún Punto de Partida, en realidad todas tienen uno implícito. Éste se obtiene calculando el punto que cae en el punto medio de tanto la extensión en el eje X como el eje Y de una Sala. Se utiliza este Punto de Partida en el caso que la Sala se use como aquella de inicio de un proyecto. En este caso, si no existen otros Puntos de Partida, la posición inicial quedaría indeterminada. Para evitar este problema, se define el Punto de Partida implícito.

Gráficos 3D

La generación de gráficos 3D no fue trivial, por lo cual a continuación se explica con algo de detalle los algoritmos usados para producir los gráficos deseados.

En el caso de los muros y las obras, una invariante importante era que éstos elementos siempre serían planos rectangulares. Por lo tanto, el algoritmo que se utilizó fue el de simplemente dividir el rectángulo en otros más pequeños, pero con igual proporción entre sus lados, para generar una malla de rectángulos. La figura 6.16 muestra gráficamente esta idea.

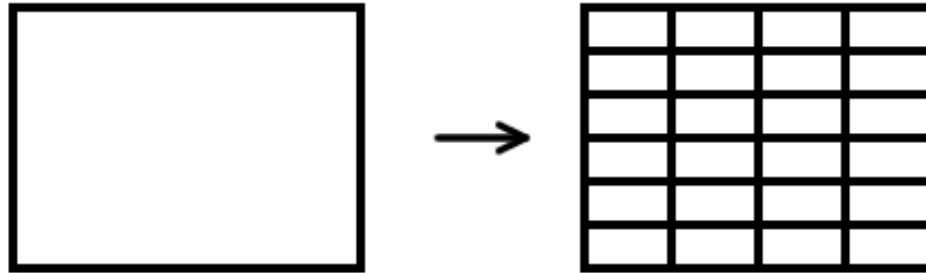


Figura 6.16: Algoritmo usado para generar los Muros y Capturas bidimensionales en la simulación 3D. La proporción entre los lados de los rectángulos pequeños es la misma del rectángulo original.

Esta división era necesaria si se deseaba darle un poco de realismo a la escena. Al momento de agregar fuentes de luz a la simulación, si los Muros fueran solamente un gran rectángulo, las luces no causarían el efecto esperado de zonas más iluminadas que otras, sino que cada muro sería de un sólo color. Al dividir los Muros en muchos rectángulos, se logra incorporar este efecto a la escena.

El efecto del sombreado se ve afectado por la cantidad de divisiones que se le da a cada muro. Muy pocas divisiones daría un efecto muy poco realista, mientras que muchas divisiones harían la generación de las escenas muy costosas en términos de operaciones de procesador, llegando a un punto tras el cual el beneficio adicional es imperceptible. Se decidió establecer el número de divisiones en cincuenta, ya que esta cantidad brinda muros que mantienen un efecto de realismo bastante aceptable, y no carga el procesador de sobremanera.

El caso del piso y el techo resultaba un poco más complejo. La causa de esta complejidad es el hecho de que éstos no cuentan con la invariante que establece que deben ser siempre rectangulares. En el caso del piso y el techo, la invariante es más relajada, y sólo establece que deben ser polígonos (ni siquiera se establece que deben ser polígonos convexos). Sin embargo, de igual manera debían ser generados mediante mallas, ya que el efecto de sombreado de los Muros también se quería presente en el caso del piso y el techo.

Para solucionar este problema, se utilizó como inspiración la idea de una «tela de araña». El algoritmo consiste en tomar el polígono que forman las esquinas de la Sala, y calcular su centro. Luego se calculan las líneas que van desde el centro hasta cada vértice, dibujando secciones triangulares alrededor del punto central. Luego, estas secciones se dividen, dejando una malla de cuadriláteros con dos de sus lados siendo segmentos de una línea que yace entre dos secciones radiales, mientras que los otros dos lados son paralelos a los lados del polígono original. La figura 6.17 muestra esta idea gráficamente.

Este algoritmo se utilizó tanto para el piso como el techo de cada sala. La única diferencia

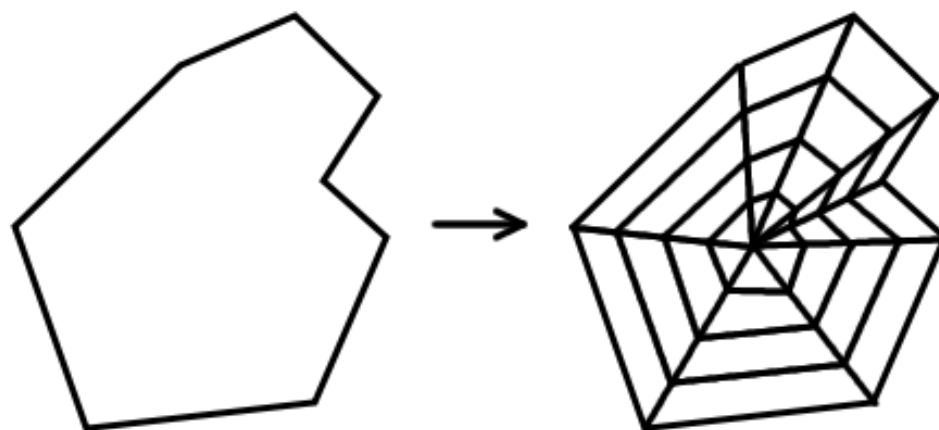


Figura 6.17: Algoritmo usado para generar los pisos y techos de las salas en la simulación 3D.

entre los dos es que el sentido en el que se dibujan es opuesto, ya que a uno se le mira «desde arriba», y al otro «desde abajo». Ya que el sentido en el cual se listan los puntos de los polígonos afecta la dirección desde la cual éste es visible, fue necesario invertir el orden de los puntos en un caso con respecto al otro.

El grado de fineza de la malla afecta al techo y piso de la misma forma que afecta a los muros, como ya fue descrito. En este caso, lo que varía al cambiar el grado de fineza de la malla es la cantidad de segmentos en los cuales se divide cada sección radial. Por ejemplo, en caso del polígono que se muestra en la figura 6.17, se hizo una malla con secciones divididos en cuatro fragmentos cada uno. Al igual que en el caso de los muros, se decidió dejar el número de divisiones en cincuenta

Puertas y Vínculos

Una vez que se implementaron las Salas, se continuó el desarrollo con el tema de Puertas y Vínculos para poder navegar de una Sala a otra. La figura 6.18 muestra los casos de uso que se determinaron suficientes para esta fase.

Lo que era necesario implementar en esta fase era lo necesario para permitirle al usuario crear y eliminar vínculos entre Salas. Estos vínculos se representarían con Puertas, las cuales serían la representación «física» de los Vínculos, ya que éstos últimos no son elementos que se visualicen dentro de la escena. Las Puertas también debían ser modificables por el usuario, permitiéndole moverlas, girarlas y cambiarles sus dimensiones.

Se crearon las clases `Door` y `Link` para representar las Puertas y los Vínculos, respectivamente. La relación entre estas clases es bastante clara; `Link` contiene dos campos que almacenan el identificador de cada extremo de la conexión que representa, de los cuales por lo menos uno debe hacer referencia a una Puerta para que el Vínculo tenga sentido. Se muestra en el diagrama, adicionalmente, que el Vínculo puede hacer referencia a un Punto de Partida,

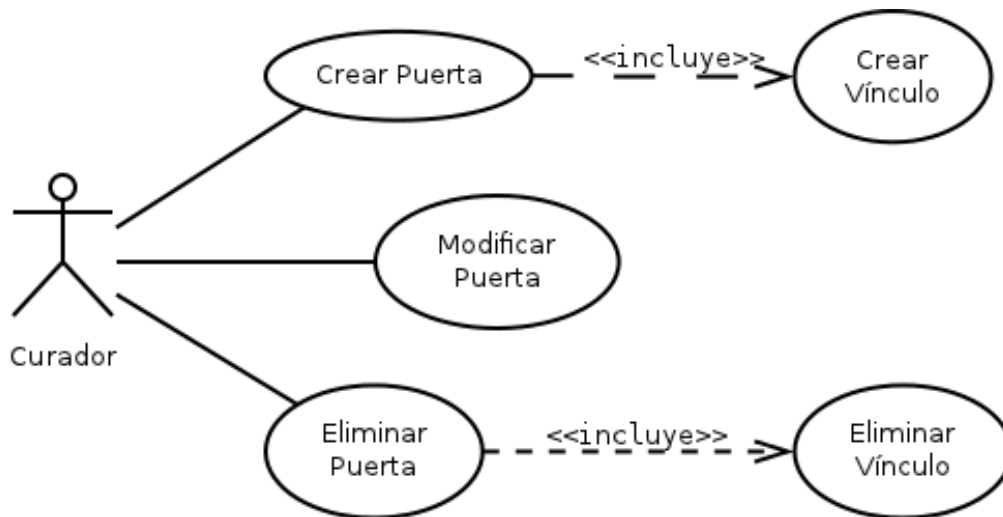


Figura 6.18: Casos de uso que incluyen Puertas y Vínculos.

lo cual representaría una Puerta que, al pasar a través de ella, transportaría al usuario a la posición indicada por el Punto de Partida, también dejando al usuario con la orientación definida por éste.

La figura 6.19 muestra las clases que se crearon en esta etapa. Se muestran también sus atributos, a excepción del caso de `StartingPoint`, ya que a esta clase se le describió en un diagrama anterior. Se puede apreciar como se modificó el diseño de la clase `Wall` para incluir también una lista de Puertas, representada por una lista de instancias de la clase `Door`.

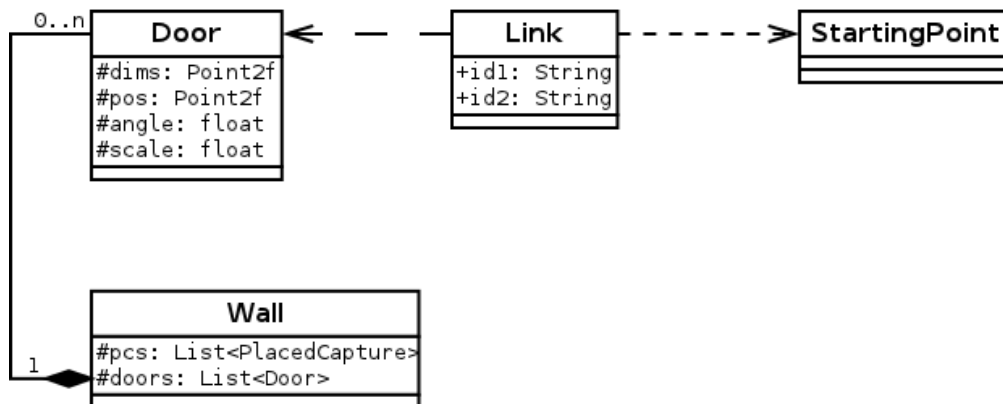


Figura 6.19: Diagrama de clases de Puertas y Vínculos. Se muestra adicionalmente la relación entre los Vínculos y los Puntos de Partida, ya que también se les puede hacer referencia a éstos últimos. La clase `Wall` fue modificada para también incluir una lista de instancias de la clase `Door`.

La clase `Door` tiene atributos casi idénticos a los de una `Captura Posicionada`. Esto es porque su visualización es casi idéntica. Las Puertas cuentan con atributos para indicar sus dimensiones, posición sobre el muro en el cual están puestas, el ángulo de su orientación y la escala que se usa para mostrarla más grande o más pequeña de lo que indican sus dimensiones. Gráficamente, se decidió representar a las Puertas simplemente con una cara blanca.

El diagrama de casos de uso mostrado en la figura 6.18 muestra que el usuario no crea Vínculos directamente. Esto fue diseñado de esta manera para mantener la interfaz lo más simple posible para el usuario. El caso más usual es que el usuario desee crear un Vínculo que vaya de una Puerta a otra. Se decidió, entonces, que al usuario se le permitiría crear Puertas y, mediante esta acción, automáticamente se crearían Vínculos para conectarlas. De igual forma, al eliminar una Puerta, la aplicación automáticamente eliminaría tanto la segunda puerta del Vínculo como éste.

Nada impide que para un mismo Vínculo se definan dos Puntos de Partida como extremos de la conexión. Sin embargo este vínculo sería inútil, ya que no hay forma de interactuar directamente con un Punto de Partida mediante la simulación, por lo cual no habría forma de utilizar dicho Vínculo.

Rutas

Era necesario implementar una manera de representar Rutas, y una interfaz que le permita al curador crearlas, modificarlas y eliminarlas. La figura 6.20 muestra los casos de uso que se definieron con este fin.

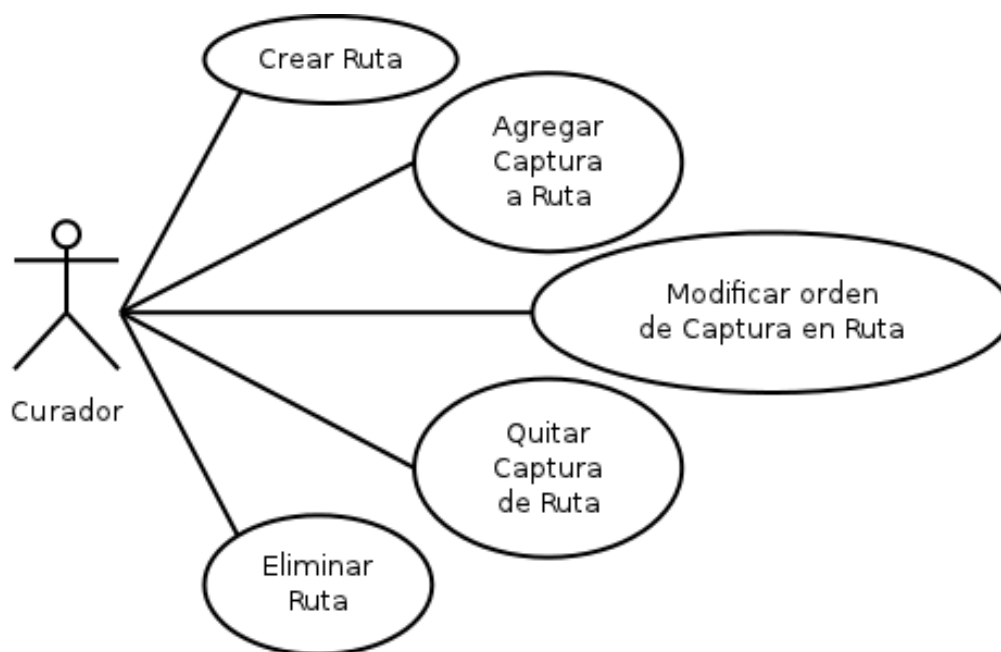


Figura 6.20: Casos de uso que incluyen las Rutas.

Las Rutas consisten simplemente en una lista de Capturas, las cuales un usuario puede revisar en orden, avanzando o retrocediendo. El objetivo es que el orden en que están dispuestas las Capturas dentro de la Ruta refleje el orden en que la visita recorre la exhibición en el mundo real. Durante su recorrido, el dispositivo lo acompañará brindando información adicional.

Se creó la clase `Route`, la cual refleja la simpleza de una Ruta conteniendo únicamente una lista de Capturas. El número de Capturas que puede contener una Ruta no tiene ninguna

restricción en lo absoluto, así como una misma Captura puede estar en cualquier número de Rutas, o dentro de ninguna. La posibilidad de hacer Rutas vacías y Capturas «huérfanas» existe, pero su utilidad es bastante limitada.

En la figura 6.21 se muestra el diagrama de clases que representa a una Ruta, junto con la relación que comparte con las Capturas. Además de ser una relación «muchos a muchos», no es una relación obligatoria. Tanto las Capturas pueden existir sin ser parte de una Ruta, como una Ruta puede existir sin contener Capturas.

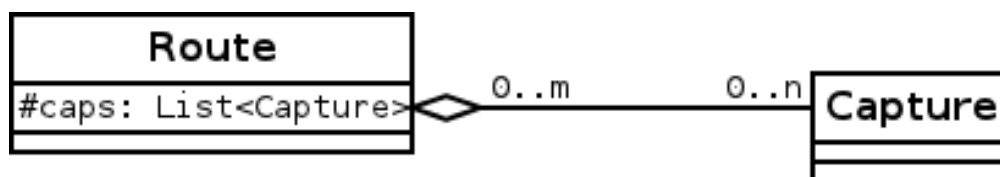


Figura 6.21: Diagrama de clases de Ruta.

Traducciones

Uno de los objetivos con los que debía cumplir la aplicación era la creación de traducciones para distintos elementos de las exhibiciones. Se determinó que lo más importante era la información acerca de las Obras Culturales, las Capturas y los nombres de las Rutas. La figura 6.22 muestra los casos de uso definidos para esta etapa.

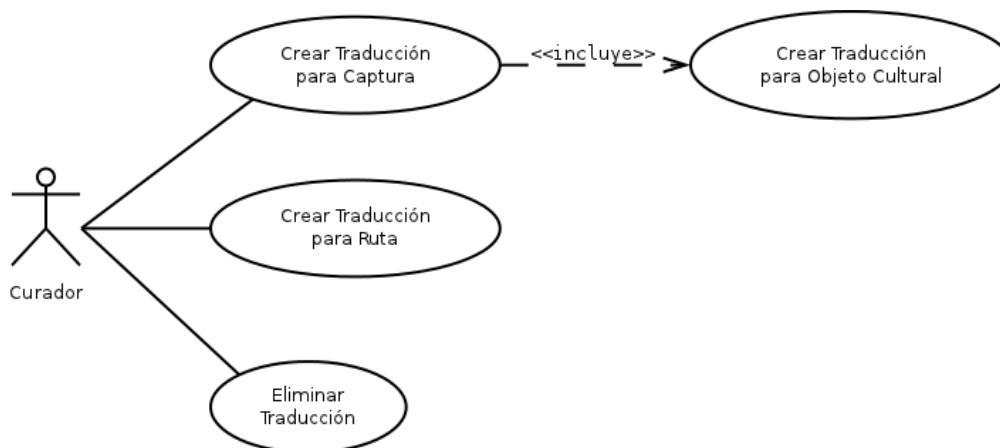


Figura 6.22: Casos de uso que incluyen Traducciones.

Para manejar las traducciones no se creó una clase nueva, sino que simplemente se utilizaron diccionarios anidados. Se definió que las traducciones se clasificarían según los siguientes valores, en el siguiente orden,

- Identificador del elemento al cual le pertenece el atributo traducido,
- el nombre del atributo traducido, y
- el idioma al cual se hace la traducción.

Estas tres llaves son las que definen una traducción, por lo cual se anidaron tres diccionarios para almacenarlas. La figura 6.23 ilustra la forma en que se llega a una traducción en particular.

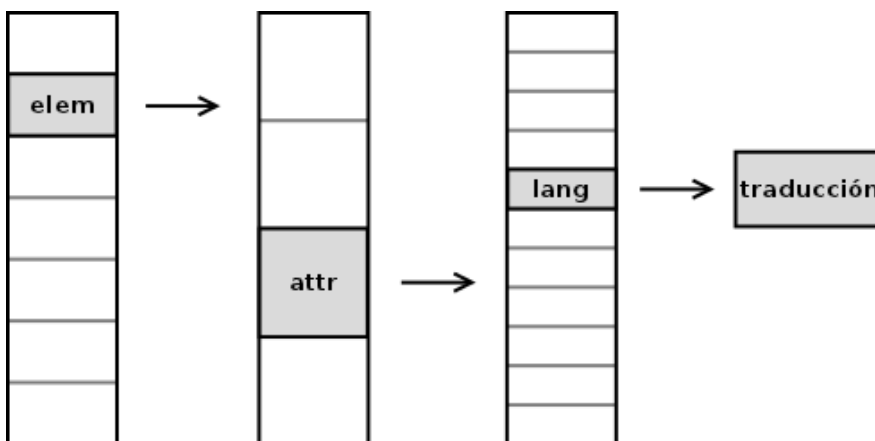


Figura 6.23: Manera de acceder a una traducción pasando por los tres diccionarios. Se necesita el identificador único del elemento al cual pertenece el atributo traducido, el nombre del atributo y el idioma en el cual está la traducción deseada para recuperarla.

Dado que la existencia de los Objetos Culturales se le esconde al curador, en el caso de las traducciones se debe ser consistente con este comportamiento, por lo cual efectivamente el usuario no puede traducir directamente Objetos Culturales. El curador accede a la interfaz que maneja traducciones para traducir información acerca de una Captura, y al traducir una, automáticamente se traduce el Objeto Cultural al cual hace referencia.

Aunque se definió que las Rutas no tienen atributos más allá de la lista de Capturas que las componen, se determinó que sería útil para el curador asignarle un nombre traducible a cada una. Esto fue porque, ya que se los objetivos contemplaban la creación de múltiples Rutas con distintos segmentos objetivo entre las visitas, se decidió que el curador se beneficiaría de poder identificar por nombre las Rutas (para no olvidar el contexto dentro del cual trabaja), y también les ayudaría a las visitas a identificar la Ruta que más le atrae al ver la lista incluida en cada proyecto.

Páginas de información

Una vez que se contaba con todo el contenido «básico» de un proyecto de museo, era el turno de la funcionalidad para agregarle información libre al proyecto. Como se especificó anteriormente, el método usado para esta etapa sería la de Páginas de Información. La figura 6.24 muestra los casos de uso que se definieron para esta fase.

Se creó la clase `InfoPage` para representar a las Páginas de Información, las cuales tendrían instancias de la clase `InfoEntry`, que representarían la entradas en los distintos idiomas. Las Páginas, a su vez, estarían contenidas dentro de un objeto de tipo `InfoSet`, el cual contendría todas las Páginas del proyecto.

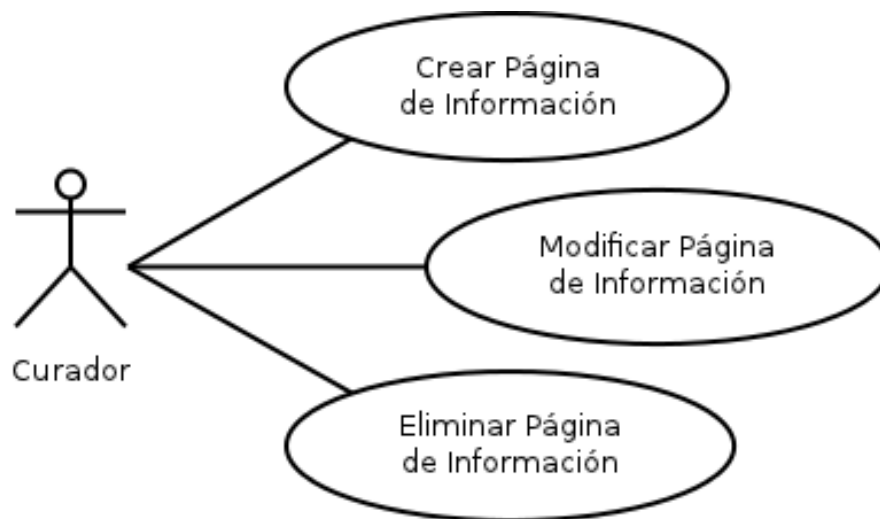


Figura 6.24: Casos de uso que incluyen Páginas de Información. Notar la similitud con el esquema utilizado para guardar traducciones.

La figura 6.25 muestra las clases descritas y sus relaciones. En particular, notar que las Páginas no están ligadas específicamente a una Captura. Esto es porque es posible definir una traducción asociada a cualquier elemento, haciendo la referencia mediante el identificador único de éste.

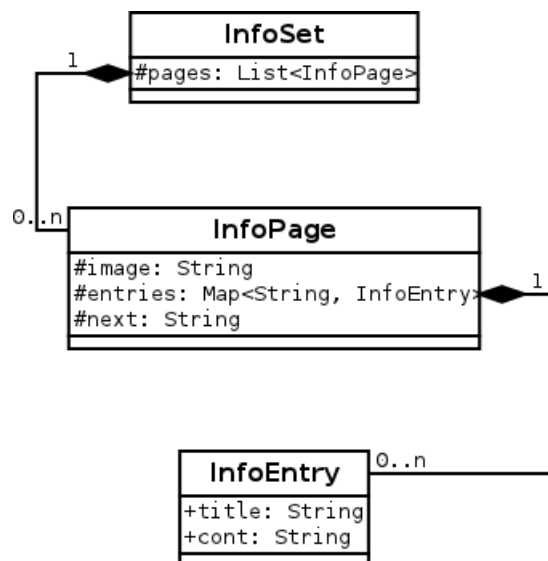


Figura 6.25: Diagrama de clases de las Páginas de Información.

Las relaciones que se muestran en el diagrama de clases son bastante simples, y son exactamente aquellas definidas en las secciones anteriores. El atributo llamado *next* perteneciente a las Páginas se definió para hacer posible la creación de secuencias de Páginas de Información. Esta variable lo que hace es almacenar el identificador único de la página que le sigue. Si la Página no tiene sucesora, entonces el campo se deja en blanco, y se interpreta como que no hay más Páginas en ese tema.

La forma en que se accede a las traducciones específicas de cada Página (llamadas Entradas)

es muy parecida a la manera en que se accede a las Traducciones de elementos generales del museo, como fue explicado en la sección anterior. Se accede a una Entrada primero buscando la Página mediante su identificador único, y luego se accede a la Entrada usando el lenguaje deseado como llave en el diccionario.

La forma en que se implementó la interfaz de creación de Páginas fue forzando al curador a crearlas en grupos asociados a alguna Captura. Se hizo esto, nuevamente, para simplificar la aplicación tanto para el curador como para la visita. El curador, en vez de manejar un conjunto de Páginas «huérfanas», maneja grupos de Páginas asociadas a alguna Captura, donde la primera Página es automáticamente dispuesta como la Página «de entrada» a la información adicional, y el resto apropiadamente encadenado según el orden definido mediante la interfaz. De esta manera, el curador no tiene la carga de estar pendiente de los identificadores únicos de cada Página, sólo se encarga de agregarle contenido al proyecto sin mayores complicaciones.

Analizadores sintácticos

Para poder recuperar la información de proyectos guardada en disco, y cargarla en la aplicación, era necesario implementar un analizador sintáctico tanto para los archivos de especificación de proyecto como para los archivos que contienen las páginas de información adicional.

Se crearon dos clases para estas labores, la clase `MemlParser` para la especificación de proyectos, y la clase `InfoParser` para la información adicional. De igual manera, se implementaron las clases `MemlWriter` e `InfoWriter` para generar archivos XML a partir del proyecto cargado en la aplicación y su información adicional, respectivamente.

Desacoplamiento del código fuente

Se mencionó anteriormente que parte de lo hecho para la aplicación del curador debería ser reutilizable para la aplicación de las visitas. Específicamente, se determinó que todo lo que no tenía que ver directamente con la interfaz gráfica se separaría a una biblioteca independiente, llamada `bnac-common`.

La biblioteca `bnac-common` fue donde se agruparon todas las clases que forman el «núcleo» del proyecto, todo aquello que tiene que ver con la lógica particular del proyecto, y nada que ver con la interfaz gráfica. Así, si se desea hacer alguna otra aplicación que interactúe con proyectos de museo que sigan el formato usado en este trabajo, entonces se podría utilizar `bnac-common`, y no se tendrían que incluir todas las clases de una biblioteca de interfaces gráficas que no se fueran a utilizar.

La interfaz gráfica de la aplicación del curador se agrupó en el paquete `bnac-editor`. Éste utiliza los contenidos de la biblioteca `bnac-common` para llevar a cabo toda la lógica que tiene que

ver con los proyectos de museo. El paquete `bnac-editor` sólo provee la interfaz gráfica para interactuar con esta capa inferior.

6.4. Aplicación para la visita

Se decidió que sería lo más conveniente utilizar como plataforma para la aplicación para las visitas el sistema operativo Android⁷. Se ha notado una clara tendencia al alza de la participación de mercado de este sistema operativo móvil [Cheng, 2011], por lo cual se estimó que al escogerlo, se llegaría a un público bastante amplio, y que no se estaría desarrollando para una plataforma pronta a quedar obsoleta. Se contaba también con la ventaja de que la plataforma Android cuenta con herramientas de desarrollo de buena calidad, que se pueden descargar y utilizar sin costo alguno.

El lenguaje de programación usado para desarrollar en la plataforma Android es Java. Las bibliotecas disponibles son en su gran parte idénticas a aquellas disponibles en la edición estándar de Java, por lo cual se podría decir que en los aspectos más básicos no se nota diferencia alguna entre desarrollo para un computador de escritorio y para un dispositivo móvil. La gran diferencia yace en la biblioteca usada para generar las interfaces gráficas. Mientras que en la aplicación para el curador se utilizó Swing, en este caso se utilizó la biblioteca de *widgets* provista por la biblioteca base de Android.

No se disponía de dispositivos con el sistema operativo Android durante el desarrollo, por lo cual se utilizó el emulador incluido en el conjunto de herramientas para el desarrollo de software de Android⁸ para hacer todas las pruebas.

Siguiendo con la idea de separar las clases que tratan con la lógica de los proyectos de museo de aquellas que forman la interfaz gráfica, se decidió separar en dos paquetes el desarrollo para los dispositivos móviles. Al que contendría la lógica de los proyectos de museo se le llamó `bnac-android-common`, mientras que al paquete con la interfaz gráfica, que le da forma a la aplicación para las visitas, se le llamó `bnac-visitor`.

Para cumplir con los objetivos de esta aplicación, se dividió la labor en los siguientes pasos,

- Carga de proyectos,
- visualización de las Capturas,
- visualización de las Páginas de Información, y
- interacción de las visitas.

⁷<http://www.android.com>

⁸<http://developer.android.com>

Carga de proyectos

Lo primero que se debía lograr para poder hacer pruebas en la aplicación de la visita era implementar un mecanismo para cargar los proyectos.

Se consideró la posibilidad de implementar un servidor del cual la aplicación pudiera automáticamente descargar proyectos y cargarlos para que las visitas pudieran acceder a la información contenida en ellos. Se decidió no seguir esta vía, ya que esto impondría una restricción sobre los museos que deseen implementar el sistema, ya que se necesitarían servidores accesibles mediante la red inalámbrica (WiFi) para poder utilizarlo apropiadamente. Ya que en esta etapa se deseaba optar siempre por lo más simple, se decidió tomar la ruta de la descarga y carga manual.

El mecanismo que se implementó para la carga de proyectos consiste en encargarle al usuario la tarea de conseguir el proyecto y hacerlo disponible desde la aplicación cargándolo en la tarjeta de memoria inserta en el dispositivo móvil, o bien en la memoria interna del mismo. Desde allí, la aplicación lo recuperaría utilizando el programa llamado OI File Manager⁹. Se recurrió a esta aplicación ya que la complejidad particular de navegar el sistema de archivos en el sistema operativo móvil se escapaba del alcance del proyecto. Por lo tanto, se decidió utilizar una aplicación desarrollada por terceros para esto. Una vez seleccionado el proyecto, se utilizarían las clases del paquete `bnac-android-common` para recuperar la información almacenada dentro de él.

Visualización de Capturas

Una vez que el usuario haya definido el proyecto con el cual está trabajando, le debía ser posible acceder a las Capturas y las Rutas allí definidas. El diagrama 6.26 muestra la interacción que se planeó implementar en esta etapa.

Uno de los casos de uso definía que la visita debería ser capaz de escoger la Ruta que le gustaría recorrer. Esto se llevaría a cabo mediante una simple lista, en la cual las Rutas estarían representadas por sus nombres. Dado que los nombres de las Rutas contaban con traducciones, el nombre de la Ruta variaría según el lenguaje que el usuario tenga definido en su aparato. Lo óptimo, aunque imposible de verificar, es que cada Ruta tenga un nombre que deje en evidencia su objetivo, o por lo menos el tema que trata. De esta manera, las visitas podrían fácilmente escoger la Ruta que se le adecúe más a su estilo y estado de ánimo.

La interfaz que se decidió implementar para la visualización de las Capturas fue una bastante simple, contando con una imagen de la Captura, junto con sus datos básicos, y dos botones: uno para avanzar en la Ruta, y otro para retroceder. Se decidió agregarle una imagen a la interfaz que muestra la información de una Captura a pesar de que la visita usualmente cuenta con la versión real de la obra al frente suyo, simplemente para darle al usuario una manera más fácil e inmediata para identificar la Captura que tiene seleccionada en su PDA.

⁹<http://www.openintents.org/en/filemanager>.

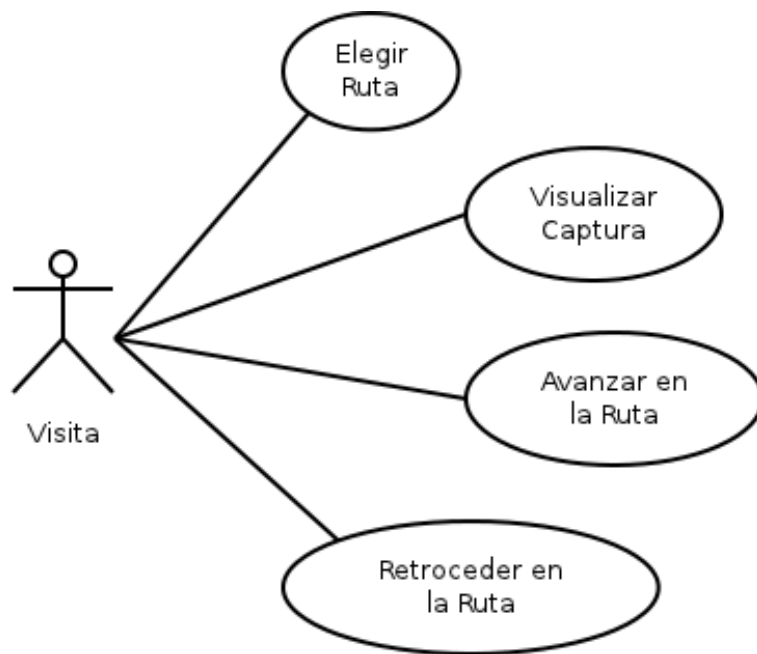


Figura 6.26: Casos de uso para la visita que incluyen las Capturas y las Rutas.

A esta pantalla que contiene los datos de una Captura se le denominó la *página principal* de una Captura.

Visualización de Páginas de Información

La figura 6.27 muestra un diagrama de casos de uso bastante simple para mostrar lo que se planeaba implementar en esta fase.

Para visualizar las Páginas de Información incluidas en el proyecto, se realizó una interfaz muy parecida a la de las mismas Capturas. La interfaz, se determinó, consistiría en la imagen opcional de cada página, seguida por el contenido de la entrada junto con su título, ambos textos traducidos al idioma con el que está configurado el dispositivo. Adicionalmente, dos botones permitirían avanzar y retroceder en la cadena de Páginas de Información asociadas a alguna obra.

Se definió, también, una manera de acceder en primer lugar a las Páginas de Información. Lo que se definió fue que cada Captura tendría un botón que, al ser presionado, le muestra al usuario la primera página de la cadena de información asociada a la Captura. A su vez, cada Página tendría un botón para volver a la captura, si es que el usuario decide simplemente seguir con la Ruta e ignorar el resto de la información.

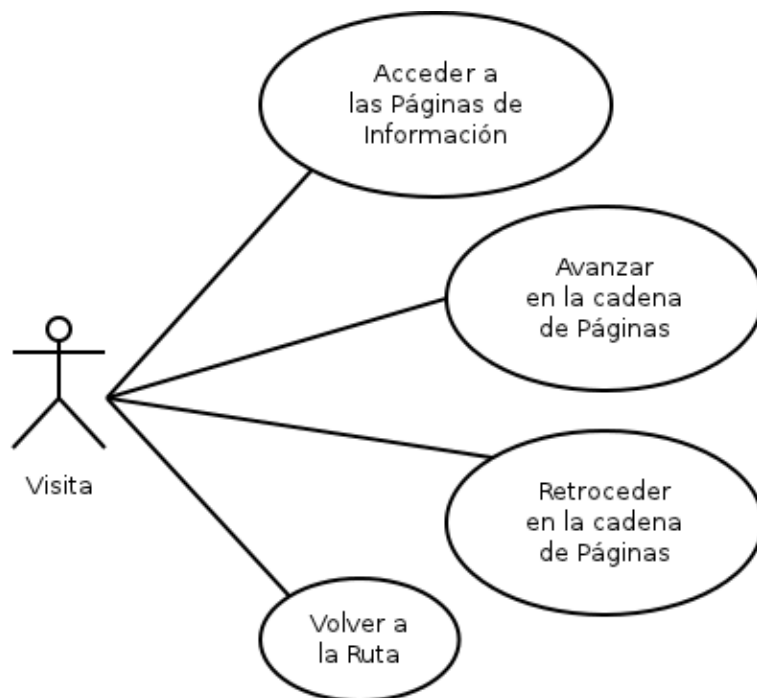


Figura 6.27: Casos de uso para la visita que incluyen las Páginas de Información.

Interacción de las visitas

Durante las reuniones con los representantes del museo, se definió que la interacción de parte de las visitas que se implementaría sería aquella de dejar comentarios y la de marcar obras como «favoritas». Estas dos acciones son bastante simples, y también son muy conocidas hoy en día, estando presente en diversos sitios, desde *blogs* de particulares hasta artículos de noticias en línea de periódicos establecidos. También se debía implementar la funcionalidad necesaria para que las visitas pudieran acceder a los comentarios dejados por otras visitas, así como poder ver el contador de visitas que han marcado una Captura como «favorita».

La figura 6.28 muestra los casos de usos que se deberían implementar en esta etapa.

La interfaz que se diseñó para que el usuario envíe sus comentarios acerca de una Captura, fue la de una típica «área de texto» dispuesta en su página principal. Para marcar Capturas como favoritas, se agregaron botones a las páginas de cada Captura, los cuales al ser presionados la marcarían como tal.

Servidor de comentarios

Era necesario implementar un servidor para almacenar todos los comentarios y los contadores de preferencia de cada Captura. Ninguno de estos dos tipos de datos pertenecían dentro de la especificación de un proyecto, por lo cual no se generaba ninguna contradicción al almacenar de forma centralizada los aportes de las visitas, mientras que los proyectos se

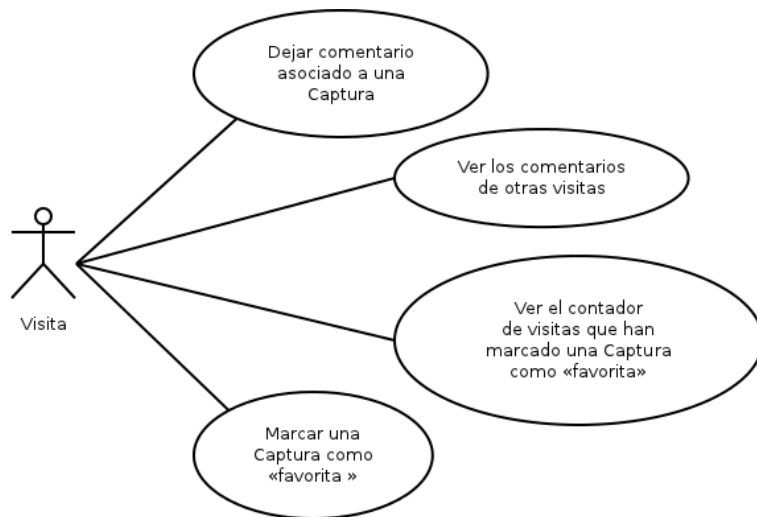


Figura 6.28: Casos de uso para la visita que representan su aporte de información.

seguirían distribuyendo a cada dispositivo.

La comunicación entre la aplicación para la visita y el servidor sigue un esquema tipo Transferencia de Estado Representacional, o REST por sus siglas en inglés¹⁰. La comunicación consiste en cuatro comandos posibles, y durante cada conexión sólo se atiende un comando, luego de lo cual el servidor la cierra. Los cuatro comandos que componen el protocolo con el cual se comunican el servidor y la aplicación de las visitas se listan a continuación,

- **PutLike**: para sumarle uno al contador de usuarios que marcan a una Captura como favorita. Como argumento se le entrega el identificador único de la Captura. Una sesión de ejemplo:

```
C: PutLike
S: OK
C: cap-secando_redes
S: BYE
```

- **PutComment**: para enviar un comentario asociado a alguna Captura. Como argumento se entregan el identificador único de la Captura, el nombre del autor del comentario, y el contenido de éste. El contenido puede incluir múltiples líneas, y su final se marca con un punto. El servidor le agrega la fecha y hora en que fue emitido el comentario. Una sesión de ejemplo:

```
C: PutComment
S: OK
C: cap-secando_redes
  Felipe Cañas
  Me gusta mucho esta obra.
```

¹⁰Representational State Transfer.

Me hubiera gustado ver más.

S: BYE

- **Likes:** para recuperar el número de visitantes que han marcado la obra como favorita. El argumento que se entrega es el identificador único de la Captura, y se recibe simplemente un número. Una sesión de ejemplo:

C: Likes

S: OK

C: cap-secando_redes

S: 20

BYE

- **Comments:** para recuperar todos los comentarios asociados a una Captura. El cliente entrega como argumento el identificador único de la Captura, y el servidor devuelve la lista con todos los comentarios asociados a ella. El orden en que se listan las partes de los comentarios es: *timestamp* de su momento de creación (representada por el número de milisegundos que han pasado desde medianoche del 1 de enero de 1970), nombre del autor, y contenido. Se marca el final de cada comentario con una línea compuesta sólo por un punto, y se marca el final de la transferencia simplemente cerrando la conexión. Una sesión de ejemplo:

C: Comments

S: OK

C: cap-secando_redes

S: OK

1298949777

Felipe Cañas

¡Hola!

Esta fue la obra que más me gustó.

.

1298423222

Gabriel Rojas

El viaje valió la pena :)

.

El motor de bases de datos que se utilizó para almacenar la información de forma no volátil fue SQLite¹¹. Se decidió utilizar esta biblioteca por la simpleza de su uso. No fue necesario utilizar un motor muy sofisticado, ya que las operaciones que se planearon eran cortas y simples, y el uso de un motor más sofisticado sólo hubiera hecho más complejo el proyecto, sin avanzar hacia el cumplimiento de los objetivos.

Al introducir un servidor al sistema, se tuvo que agregar un caso de uso a la aplicación de la visita. Éste establece que desde la aplicación debe ser posible definir la dirección IP y el

¹¹<http://www.sqlite.org/>.

puerto mediante los cuales se puede establecer la conexión al servidor de comentarios.

Navegación a través de la aplicación

Por último, la figura 6.29 muestra las pantallas por las que se pasa al utilizar la aplicación, y las conexiones entre unas y otras.

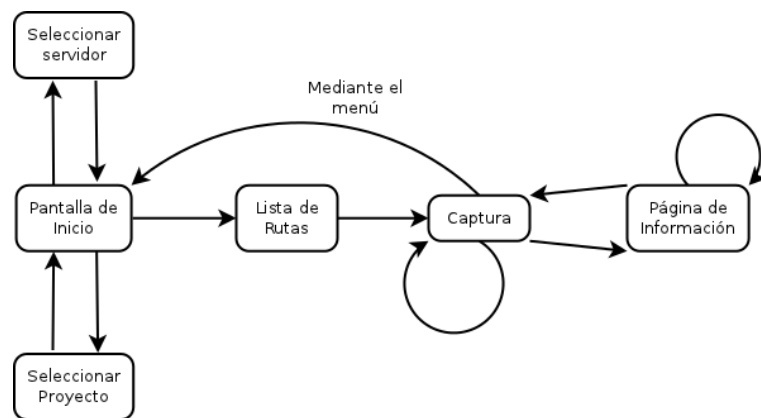


Figura 6.29: Diagrama de flujo por las pantallas de la aplicación para visitas.

Se muestran las conexiones de una pantalla a otra como fueron descritas anteriormente, más algunas que no se habían explicitado. Por ejemplo, se puede volver desde la pantalla de las Capturas a la pantalla de inicio mediante una opción en el menú. De igual forma, ciertas opciones de menú permiten ir desde la pantalla de inicio a las pantallas de selección de servidor y de proyecto.

Capítulo 7

Resultados

En este capítulo se muestran las interfaces de las aplicaciones para el curador y la visita en su estado final, con el objetivo de mostrar la manera en la cual se despliega el contenido para cada usuario.

7.1. **bnac-editor**

La aplicación del curador utiliza varias ventanas, llamadas «vistas», para permitirle al usuario visualizar y modificar los distintos tipos de elementos de un proyecto.

Ventana principal

La ventana principal de la aplicación contiene un menú con opciones para abrir, grabar y crear nuevos proyectos. Desde esta ventana se pueden hacer visibles las distintas vistas de cada proyecto, así como también las herramientas para modificar las traducciones y las páginas de información. Adicionalmente, se muestra el nombre del proyecto activo en ese momento, junto a un botón que define si al activar una vista se activa su proyecto automáticamente. Un proyecto «activo» es el que se ve afectado al hacer una llamada a las opciones de *deshacer* o *rehacer*.

Vista de simulación

La Vista de simulación sirve como la interfaz mediante la cual el usuario puede navegar dentro de la representación virtual de la exhibición. En esta interfaz se puede interactuar con Capturas Posicionadas y Puertas: se pueden seleccionar utilizando el botón izquierdo del ratón sobre ellas, tras lo cual se pueden mover, cambiar de tamaño, girar e incluso quitar.

El movimiento a través de la simulación se indica exclusivamente usando el teclado. Se utiliza el esquema WASD para moverse, es decir, se utilizan las teclas W, A, S y D para moverse hacia adelante, hacia la izquierda, hacia atrás y hacia la derecha, respectivamente. Para girar, se utilizan las teclas U e I para girar hacia la izquierda y derecha, respectivamente.

Vista de capturas

La Vista de capturas es la que el usuario utiliza para agregar nuevas capturas al proyecto, como también para ver la lista de capturas que contiene. Mediante esta vista se pueden crear capturas posicionadas para agregar a la sala activa en la Vista de simulación. Al seleccionar una captura para su incorporación a una sala, se muestra un diálogo para seleccionar el muro sobre el cual se desea «colgar» la imagen.

Vista de salas

Esta vista se utiliza para administrar las salas. Desde esta vista se pueden agregar nuevas salas y para ver la lista de las que pertenecen al proyecto. Cada ítem de la lista tiene un botón rotulado «Ir» el cual sirve para hacer que la Vista de simulación utilice esa sala para generar la escena.

Vista de plano

La Vista de plano es la interfaz mediante la cual se puede modificar el «plano» de una sala, es decir, la posición de sus esquinas en el plano. A través de esta vista las esquinas se pueden agregar, mover y eliminar.

La interacción con esta vista se hace exclusivamente con el ratón. Con el botón izquierdo se seleccionan las esquinas o los muros que se desean dividir (creando una nueva esquina), y se arrastra el cursor sin soltar el botón hasta que se deja la esquina seleccionada en la posición deseada. Al soltar, se llevan cabo los cambios. En el caso que se desee eliminar una esquina, se utiliza el botón derecho del ratón.

Vista de navegación

La interfaz mediante la cual se administran los enlaces entre salas es la Vista de navegación. Esta vista muestra el listado de los enlaces existentes, mostrando dónde se encuentran ambas puertas de cada vínculo, como también ofrece las opciones de creación y eliminación de enlaces.

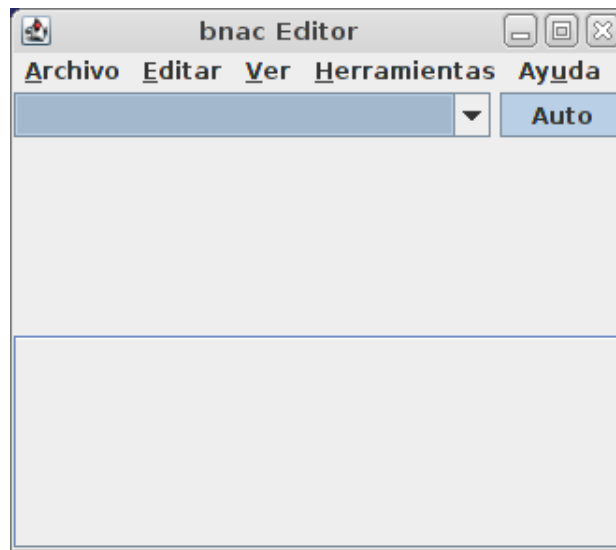


Figura 7.1: Ventana principal de la aplicación. El menú «Ver» contiene opciones para mostrar las distintas vistas de cada proyecto.



Figura 7.2: Vista de simulación. Se pueden ver dos cuadros junto con una puerta.



Figura 7.3: Vista de simulación, con los elementos modificados. Se muestra la misma escena que en la figura 7.2, pero con sus elementos modificados.



Figura 7.4: Comparación de una captura seleccionada y no seleccionada. Al ser seleccionada, una obra se pinta de rojo, como se muestra en el lado derecho de la imagen.



Figura 7.5: Vista de capturas. Se muestran dos capturas, se pueden ver los atributos principales con los cuales fueron definidas.

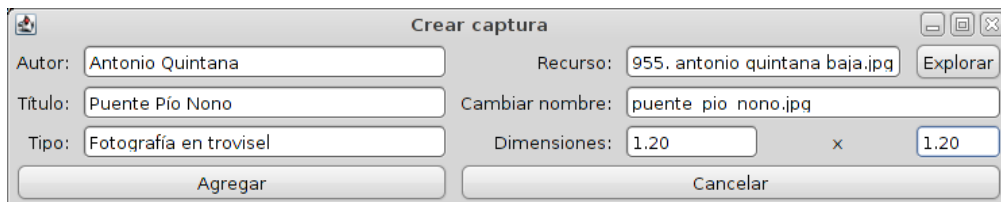


Figura 7.6: Diálogo de creación de captura.

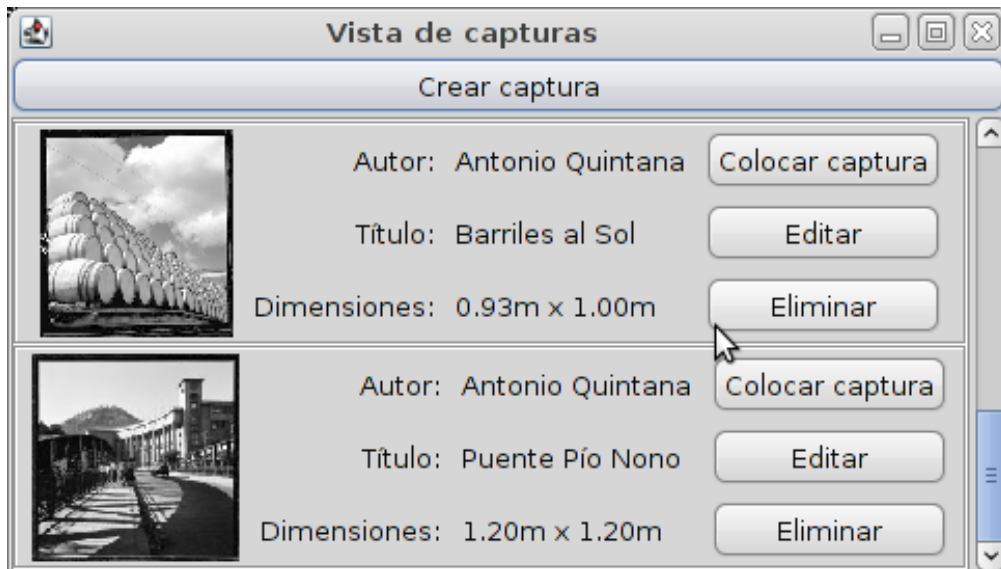


Figura 7.7: Vista de capturas con nueva adición. La captura «Puente Pío Nono» fue agregada según lo definido en la figura 7.6.

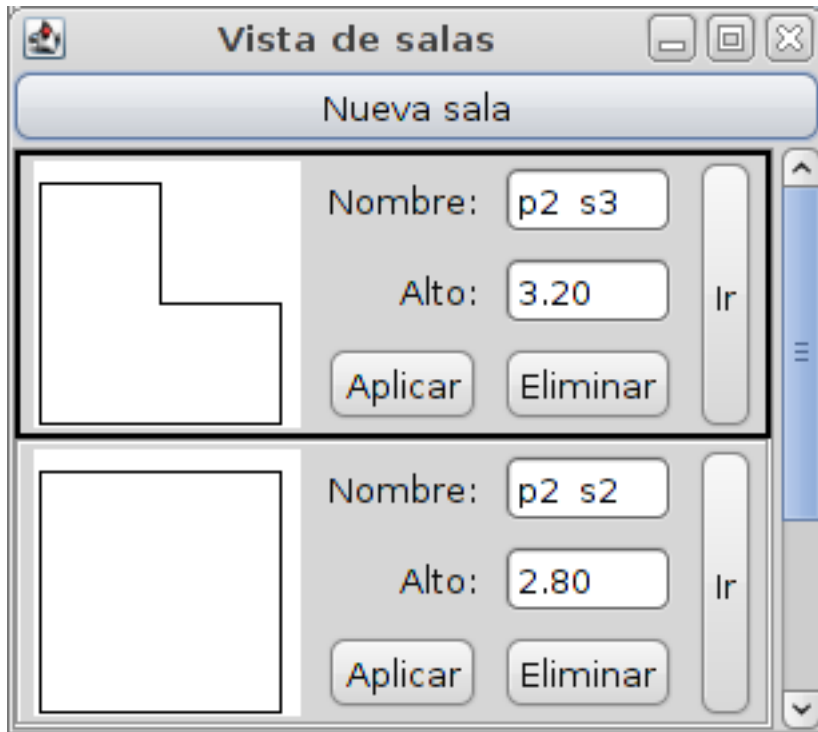


Figura 7.8: Vista de salas. Se muestran dos salas, junto con su altura (en metros) y el polígono que forman sus esquinas. La sala con borde negro indica que es la sala en ese momento presente en la Vista de simulación.

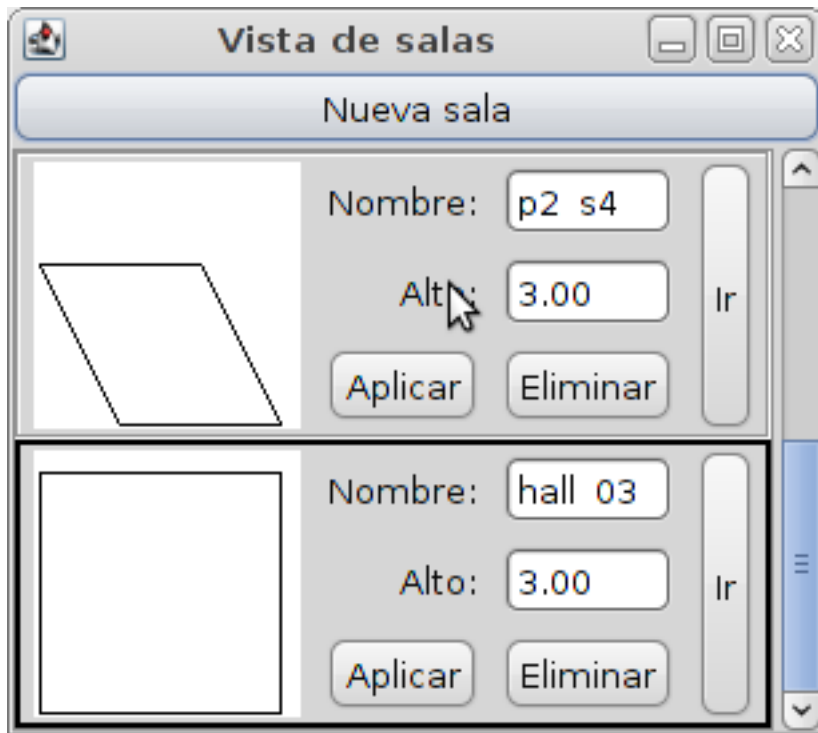


Figura 7.9: Vista de salas con una sala nueva. Las salas nuevas siempre se generan como salas cuadradas de tres metros de alto. En la Vista de salas se puede ver que, al crear una nueva sala, la vista de simulación automáticamente carga esta sala.

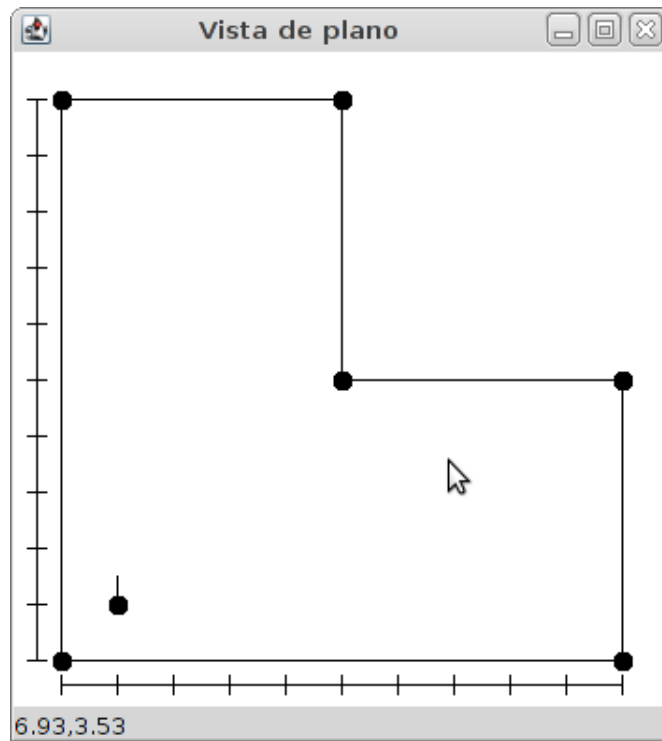


Figura 7.10: Vista de plano. Los números en la esquina inferior izquierda indican las coordenadas sobre las cuales se encuentra el puntero en ese momento. Las líneas que denotan los ejes cuentan con divisiones de un metro.

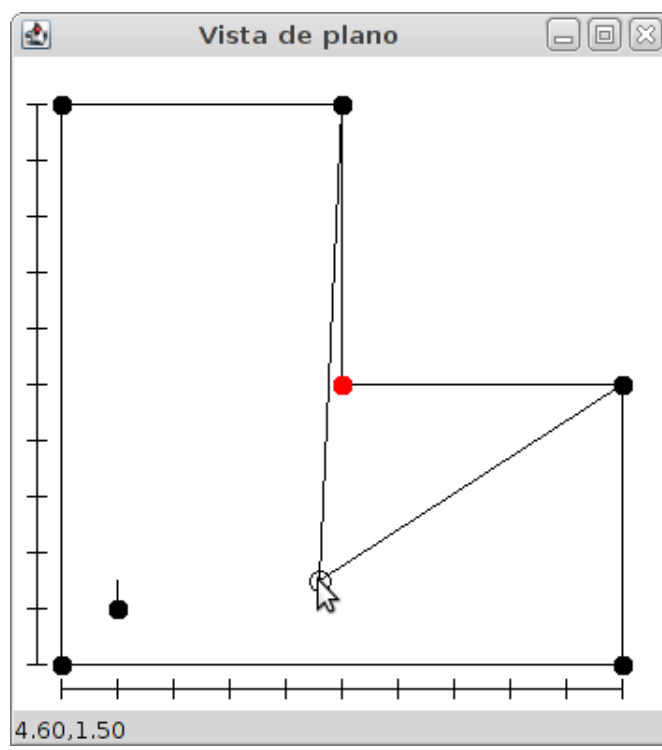


Figura 7.11: Vista de plano mostrando como se mueve una esquina. La vista mantiene la configuración original de las esquinas mientras se ejecuta el movimiento para poder comparar la modificación con el estado original.

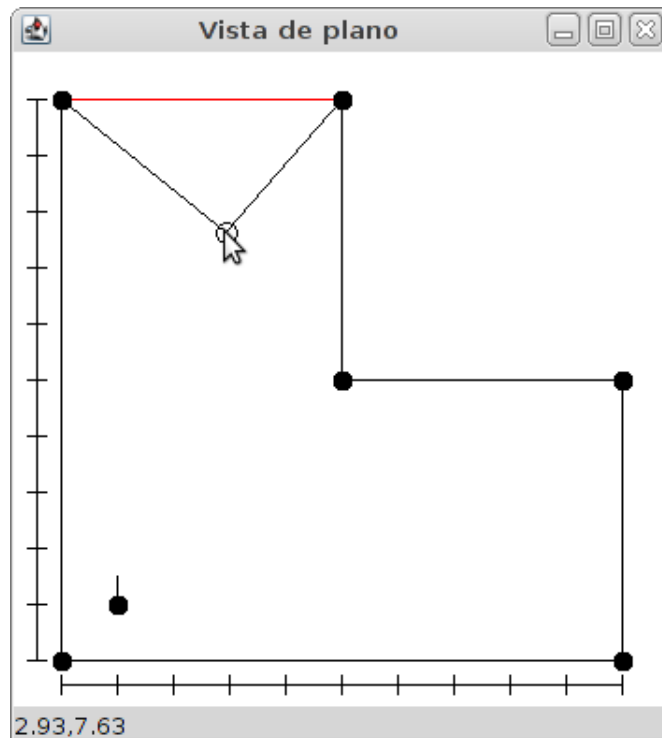


Figura 7.12: Vista de plano mostrando como se divide un muro. La manera de crear una nueva esquina es dividiendo un muro. Nuevamente, la interfaz muestra simultáneamente los cambios y el estado original para darle la facilidad al usuario de poder compararlos.

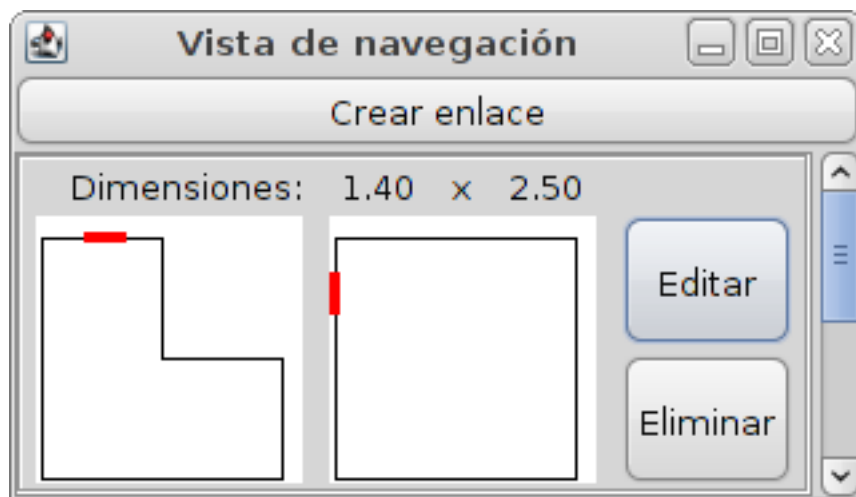


Figura 7.13: Vista de navegación. Las puertas de cada extremo del vínculo se marcan en rojo en el dibujo del plano de la sala en la cual se encuentra cada una.

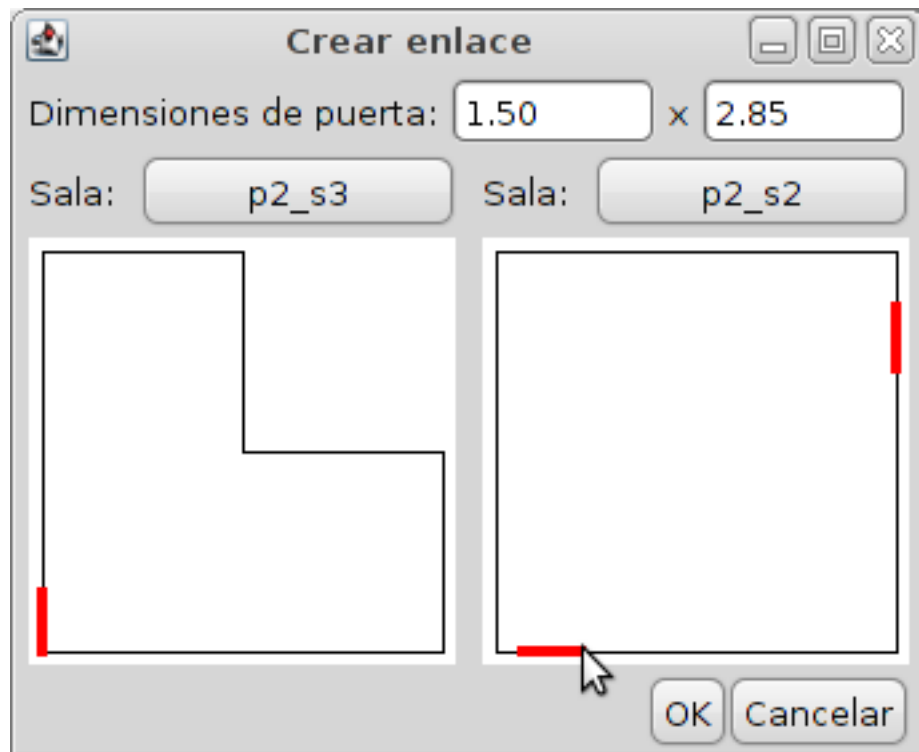


Figura 7.14: Diálogo de creación de un nuevo vínculo y de sus dos puertas. Se utiliza una interfaz muy parecida a la de la lista de enlaces. El usuario puede ver cómo quedará la puerta en la posición sobre la cual se mueve el puntero del ratón, y es posible definir un lugar como el deseado mediante el botón derecho del ratón.

Vista de rutas

La Vista de rutas tiene como objetivo ser la interfaz del usuario para la administración de las rutas. Mediante ésta, el usuario puede crear nuevas rutas, eliminar antiguas, o bien modificarlas.

Herramienta de traducciones

Tal como indica su nombre, esta herramienta tiene como función administrar las traducciones del proyecto. Se decidió que los elementos que el usuario tendría la posibilidad de traducir serían las capturas y las rutas. En el caso de las capturas, el usuario tendría la posibilidad de traducir su información básica, mientras que en el caso de las rutas se podría traducir su nombre.

Al usuario no se le ponen restricciones con respecto al idioma a usar para crear el contenido, se puede usar el idioma más cómodo para el que esté llevando a cabo la generación de contenido. Una vez que se desee traducir, se puede utilizar la opción «Valores originales» para cargarlos y asociarlos al idioma indicado. Se creó esta opción para no obligar al usuario a ingresar la misma información más de una vez.

Herramienta de información

La herramienta de información es la interfaz que el usuario tiene a su disposición para definir páginas de información acerca de las capturas que forman parte de una ruta. El usuario puede definir páginas y sus traducciones a distintos idiomas. Adicionalmente, a cada página se le puede agregar una imagen, la cual será mostrada independientemente del idioma.

La técnica que usa para asociar una página a una captura es mediante su identificador único. Lo que se hace es definirle el identificador a la primera página de la cadena asociada a una captura de tal forma que éste es idéntico al de la captura asociada. Todas las páginas siguientes pueden tener un identificador independiente de la captura, ya que se le hace referencia utilizando el campo «next» de la página anterior.

7.2. **bnac-visitor**

La aplicación de la visita corre dentro de un dispositivo móvil, por lo cual su interfaz es muy distinta a la de **bnac-editor**.

La figura 6.29 muestra el flujo de las pantallas de la aplicación, ya que el espacio reducido de la pantalla del dispositivo hace inviable el uso de ventanas múltiples, como es el caso de los

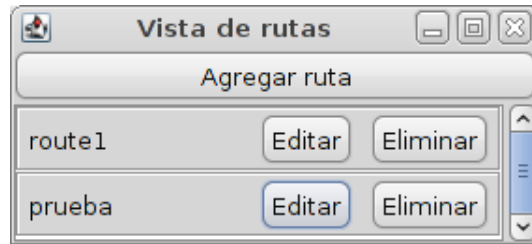


Figura 7.15: Vista de rutas. Se muestran dos rutas, rotuladas con su identificador único.



Figura 7.16: Diálogo de ruta. Éste muestra la lista de capturas incluidas en la ruta, junto al orden con el cual se incluyen. Las capturas se pueden mover hacia arriba o hacia abajo en la lista, o incluso eliminarse, mediante los botones asociados a cada una.

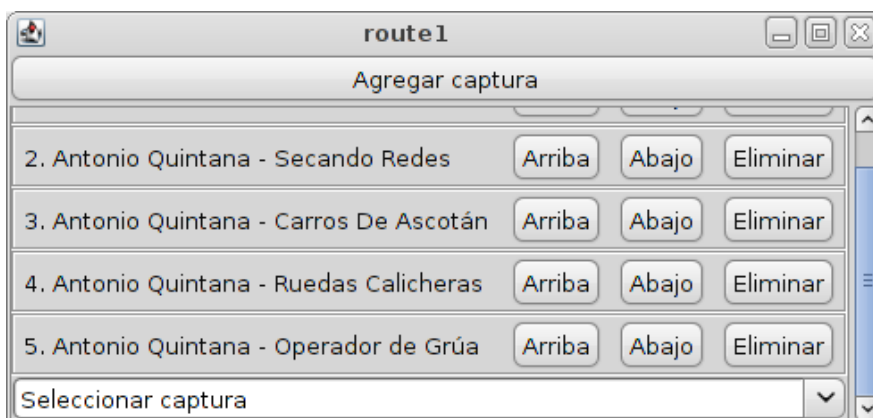


Figura 7.17: Diálogo de ruta luego de presionar el botón «Agregar captura». Se puede observar el campo desplegable en la parte inferior de la lista, el cual contiene una lista con todas las capturas del proyecto que no se encuentran ya en la ruta. El usuario selecciona una, y ésta se agrega a la ruta en la posición final.

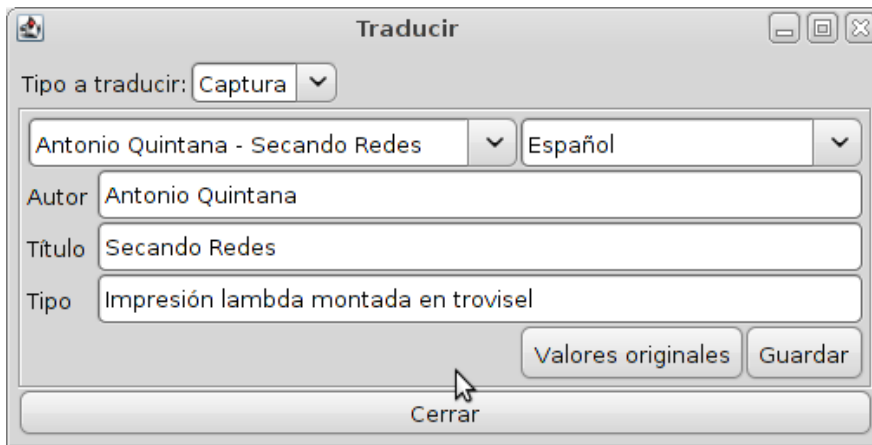


Figura 7.18: Herramienta de traducciones. En este ejemplo se usó la opción «Valores originales», con lo cual se cargaron los valores de cada atributo con que se creó la captura. Éstos estaban en español, por lo cual se asociaron a ese idioma.

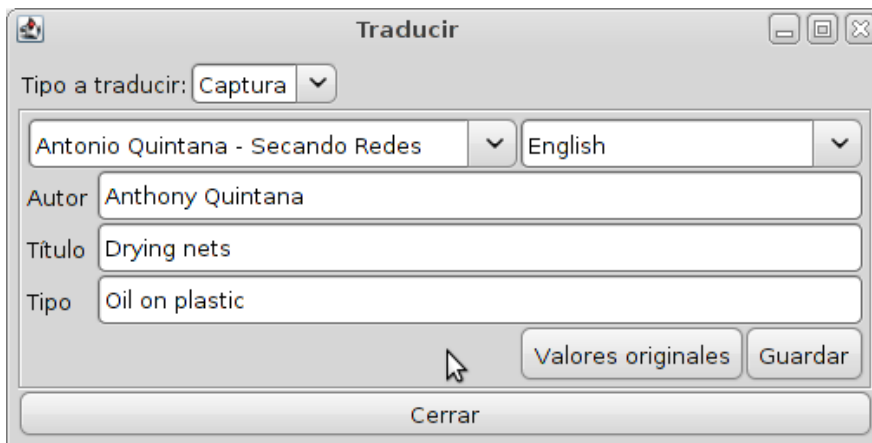


Figura 7.19: Herramienta de traducciones mostrando una traducción. Se muestra la misma captura del caso anterior, pero con su traducción al inglés.

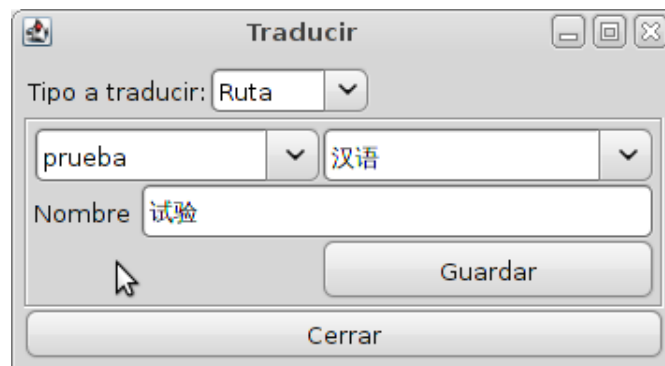


Figura 7.20: Herramienta de traducciones, en el caso de una ruta. Se muestra la interfaz en el caso de una ruta, en el cual el único atributo a traducir es el nombre de ésta. En este ejemplo se muestra su traducción al chino mandarín.

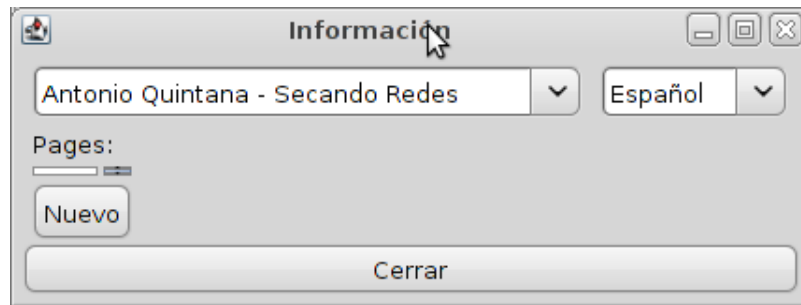


Figura 7.21: Herramienta de información. Se muestra el estado inicial de una captura, es decir sin ninguna página definida, en ningún idioma.

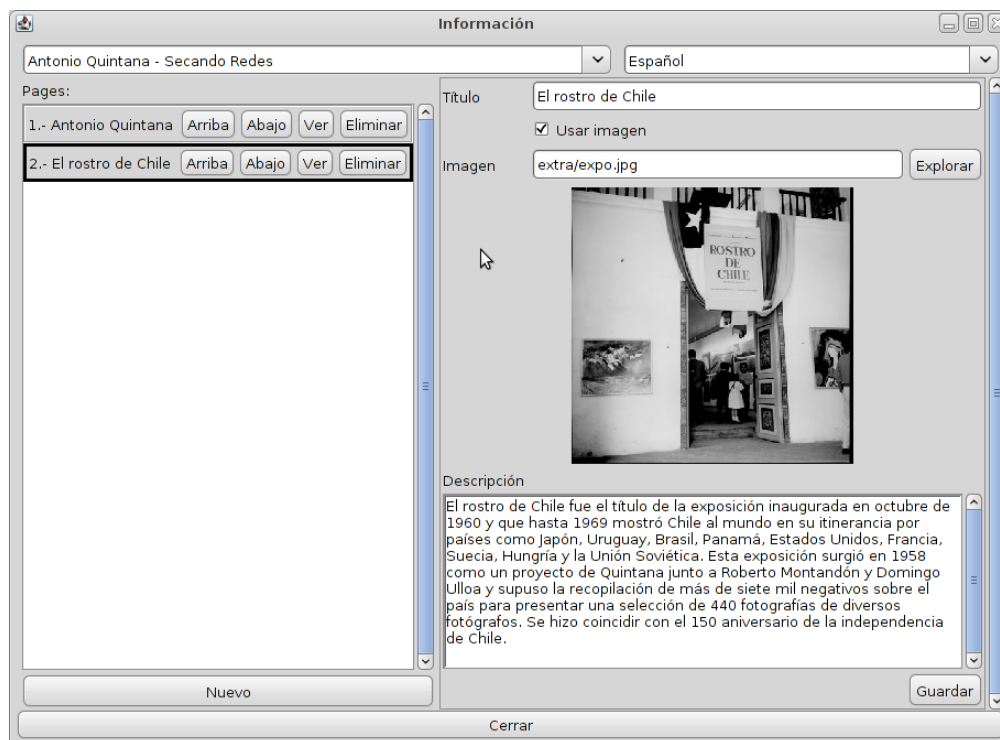


Figura 7.22: Entrada de una página de información. Se muestra la entrada en el idioma español de una página. En la lista de la izquierda de la ventana se muestran los títulos de las páginas en el idioma seleccionado. La página seleccionada se indica mediante un borde negro, como se muestra el caso de «El rostro de Chile».



Figura 7.23: Entrada inexistente para una página en cierto idioma. Se muestra una página al seleccionar un idioma para el cual no hay una entrada definida. El botón rotulado «Crear entrada» se debe presionar para crearla y comenzar su edición.

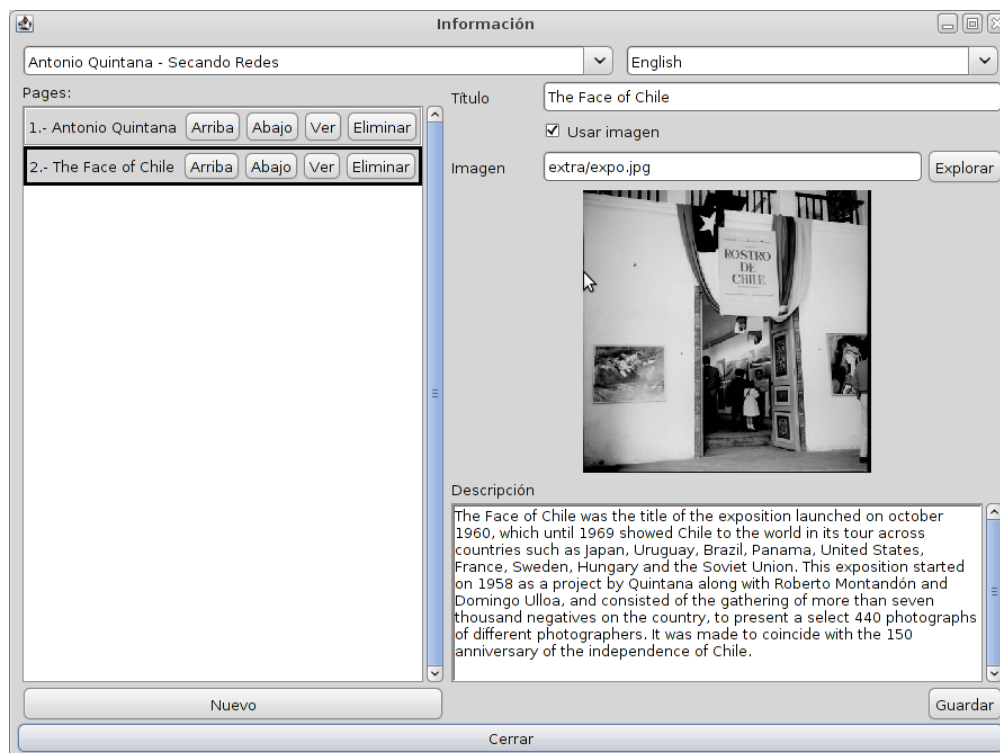


Figura 7.24: Entrada de una página de información mostrando la traducción a otro idioma. Se muestra la misma página de la figura 7.22, pero esta vez traducida al inglés. Se puede apreciar que la imagen es la misma para ambos casos.

programas de computadores de escritorio. Cada pantalla tiene un objetivo claro, y un diseño para lograr esto de la mejor manera posible. A continuación se explica con un poco más de detalle cada una.

Pantalla de inicio

La pantalla de inicio contiene solamente un botón para pasar a la lista de rutas. La razón por la cual se incluyó fue para asignarle un menú con dos opciones importantes para la operación de la aplicación: selección de proyecto y la selección de servidor de comentarios.

Al presionar el botón rotulado «Comenzar», se intenta cargar el proyecto para desplegar su información. Si se tiene éxito, se pasa a la pantalla de selección de ruta.

Pantalla de selección de rutas

Esta pantalla tiene como objetivo exactamente lo que dice su título, permitirle al usuario seleccionar la ruta que recorrer. Los nombres de las rutas se muestran en el idioma escogido por el usuario para su sistema, y en el caso de que no haya un nombre disponible en el idioma seleccionado, se muestra el identificador único de la ruta en su lugar.

Pantalla de captura

La pantalla de captura muestra la información básica de una captura. Es decir, ésta muestra el recurso de la captura (o la vista previa, en caso de que cargar el recurso sea imposible) y muestra sus atributos. Adicionalmente, esta pantalla contiene un botón para acceder a información adicional, provista por páginas de información, como también los elementos destinados a captar información de parte de la visita.

La figura 7.28 muestra los dos elementos de la interfaz mediante los cuales la visita puede compartir información con el museo. El primer elemento es el botón rotulado «Me gusta», el cual deja registrado que el usuario mostró cierta preferencia por la obra en cuestión por sobre otras.

El segundo elemento mediante el usuario puede compartir con el museo es el grupo de áreas de texto para crear un comentario y mandarlo. Luego de mandarlo, el comentario aparece en la lista, la cual se encuentra sobre los campos para escribir el comentario.

La figura 7.29 y 7.30 muestran como se anota un comentario y, luego de enviarlo, éste se agrega a la lista. También muestran lo que ocurre luego de presionar el botón rotulado «Me gusta»: el botón desaparece, y el contador de «personas a las que les gusta esto» se incrementa en uno.



Figura 7.25: Pantalla de inicio de la aplicación para la visita. Se muestra el menú, el cual normalmente no es visible, con las opciones de selección de proyecto y servidor.

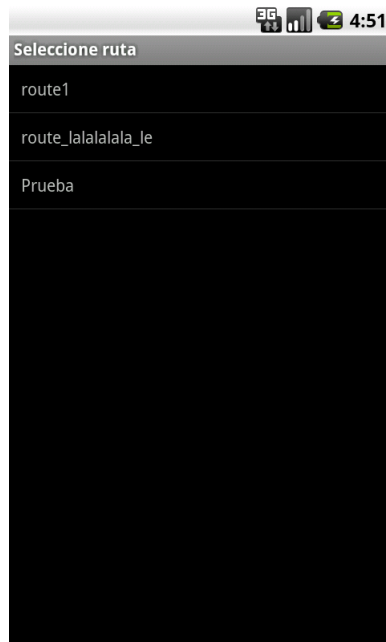


Figura 7.26: Pantalla de selección de ruta. Se muestran tres rutas, de las cuales sólo la última tiene una traducción para su nombre al idioma español, el cual es el idioma seleccionado en el sistema. Por esta razón, las primeras dos rutas están rotuladas con su identificador único, mientras que la última despliega su nombre en español en pantalla.



Figura 7.27: Pantalla de captura. Se muestra información básica acerca de la captura. Para acceder a más información, se puede acceder a la pantalla de información presionando el botón que se muestra seleccionado en la imagen.

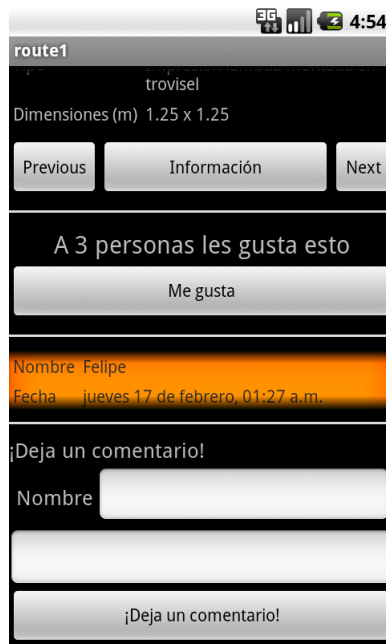


Figura 7.28: Elementos de la interfaz de la pantalla de capturas para que el usuario comparta con el museo. Éstas son el botón «Me gusta», y las áreas de texto para dejar un comentario asociado a la obra.

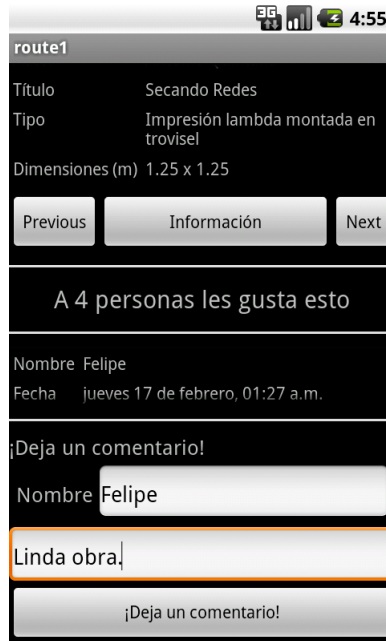


Figura 7.29: Comentario antes de enviar. Adicionalmente, se puede apreciar que el botón «Me gusta» ha sido ocultado y el contador de «personas a las que les gusta esto» ha incrementado en uno, lo que significa que el usuario presionó el botón.

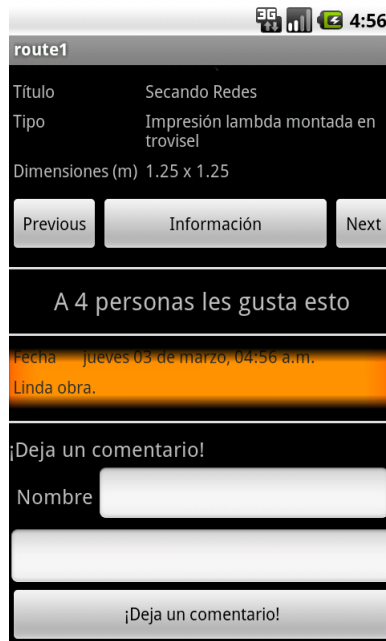


Figura 7.30: Comentario después de enviar. El comentario aparece en la lista.

Pantalla de información

La pantalla de información tiene como objetivo mostrar toda la información adicional que el curador quiso agregar a las capturas, información que no se podía agregar como atributo básico.

La información, como se mencionó anteriormente, se segmenta por páginas, las cuales pueden contar con una imagen, y con traducciones para distintos idiomas.

A cada captura se le puede asignar una serie de páginas de información, las cuales la visita luego puede revisar en orden. En cualquier momento, la visita puede presionar el botón que lo lleva de vuelta a la página de la captura, para seguir avanzando por la ruta.



Figura 7.31: Pantalla de información. Se muestra el ejemplo correspondiente a la figura 7.22. Se puede apreciar claramente tanto la imagen, como el contenido y el título de la página.



Figura 7.32: Pantalla de información mostrando una traducción a otro idioma. Se muestra la misma página que en la figura 7.31, pero esta vez con el inglés como idioma seleccionado. Se puede apreciar que tanto el contenido como el título cambiaron, sin embargo la imagen sigue siendo la misma.

Capítulo 8

Conclusión

En este trabajo se desarrollaron dos aplicaciones, una destinada a ser usada por el curador del museo, y otra para ser usada por las visitas del museo. Se implementó la funcionalidad necesaria para permitirle al curador generar contenido que compartir con las visitas de forma que éstas disfruten de una mejor experiencia en el museo. Adicionalmente, a la aplicación de las visitas se le agregó funcionalidad que les hace posible comunicarse con el museo, y poder dejar una huella de su paso por allí, a la cual pueden acceder todas las futuras visitas.

El desarrollo de la aplicación del curador incluyó elementos de modelado de gráficos 3D, lo necesario para lograr generar escenas tridimensionales que representen las salas de exhibición del museo. El objetivo de esto fue facilitarle al curador la labor de armar las exhibiciones, mediante el uso de modelos tridimensionales fáciles y rápidos de generar.

La aplicación para las visitas se desarrolló con las herramientas de desarrollo para el sistema operativo móvil Android. Dada la variedad de plataformas móviles en existencia, y la imposibilidad de desarrollar una aplicación para todas ellas por razones de tiempo, se escogió la plataforma Android por su creciente popularidad y por su cualidad de ser un sistema abierto para el desarrollo.

Se desarrolló, adicionalmente, un servidor encargado de recibir los datos desde las visitas. En esta etapa para el servidor solamente se implementó la recepción y el envío de comentarios y el contador de veces que se ha marcado la captura con «Me gusta», ya que éstas fueron las interacciones definidas para las visitas en este trabajo.

Se definió un formato para los proyectos de museo, el cual especifica la estructura que debe seguir el directorio en el cual se encuentren los recursos del proyecto, como también los archivos de especificación de sus elementos. El formato de los archivos de especificación se definió utilizando XML, dada su gran aceptación a nivel mundial, a su vez causada por la facilidad de su uso.

Los resultados del desarrollo muestran que el objetivo general de enriquecer la experiencia de las visitas dentro del museo fue efectivamente abordado. Se desarrolló un conjunto de

aplicaciones que permite, finalmente, darle a la visita elementos adicionales con los cuales interactuar durante su recorrido, los cuales se enfocan no en opacar la experiencia de apreciar las obras presentes en la exhibición, sino de complementarla.

Aunque el estado en que quedó el conjunto de aplicaciones al momento de confeccionar el presente informe estuvo más cerca de un prototipo que de un producto acabado, era tal que se podría fácilmente iniciar su uso en un museo. Ignorando el servidor de comentarios, bastaría que un museo dejara contenido acerca de alguna exhibición junto con la aplicación para las visitas disponible para descarga en su sitio de Internet, y todos los visitantes con un dispositivo móvil con sistema operativo Android podrían acceder al contenido. Esto, además de ser completamente posible, no tiene costo monetario alguno, ni es necesario desplegar dispositivos de ningún tipo a través del museo para el correcto funcionamiento del sistema.

El hecho de que los usuarios puedan descargar la aplicación por su cuenta, y que no dependan en que el museo les provea un dispositivo móvil dentro del cual ejecutar la aplicación, es una gran ventaja de este sistema. Tradicionalmente, si existen dispositivos para complementar las visitas (típicamente reproductores de audio), éstos son provistos por el museo. Esto implica problemas desde abastecimiento hasta congestión en los puntos de entrega de dichos dispositivos, y la grave posibilidad de su robo o avería. Al tener la posibilidad de traspasarle esta responsabilidad a la visita, el sistema se hace mucho más simple de administrar para el museo. Claramente, esto no significa que de igual forma el museo pueda contar con dispositivos para aquellas visitas que no dispongan de ellos.

El formato utilizado para describir los proyectos se diseñó de tal forma que fuera lo más simple posible, sin dejar de cumplir con los requerimientos mínimos, ya que se existía la posibilidad de que un tercero decida implementar su propia aplicación que la utilice. Esto es algo que no se desea evitar por ningún motivo. El hecho de hacer el estándar abierto y simple permite que, por ejemplo, un grupo de desarrolladores pueda implementar una aplicación para el iPhone que pueda cargar los mismos proyectos que la aplicación desarrollada para este trabajo, y que ambas implementaciones muestren el contenido de la misma manera.

Los proyectos también brindan una manera de guardar un registro de la configuración de distintas exhibiciones. Si se utiliza la aplicación del curador para planificar la posición de todos los elementos de una exhibición, luego el proyecto así generado, en vez de ser descartado, se podría guardar no sólo por razones de documentación, sino que se podría utilizar en caso de que la exhibición se vuelva a poner a disposición del público en el museo, o incluso se podría enviar a un museo que fuera a recibir la colección de obras contenidas en el proyecto como préstamo. Este segundo museo podría, entonces, disponer de las obras de forma de emular su exhibición original.

Finalmente, se puede decir que se desarrolló una base para un sistema de visitas a museos apoyadas por dispositivos móviles, y se formó un contacto con un museo importante, lo cual consiste en un primer paso importante para el proyecto. La computación brinda oportunidades muy interesantes para el contexto de los museos, y el trabajo acá hecho sienta una base sobre la cual se puede construir para concretarlas.

8.1. Trabajo futuro

Como se mencionó, el objetivo de este trabajo era implementar la funcionalidad planeada en un prototipo, ya que era más importante observar el funcionamiento que contar con una aplicación depurada y acabada, tanto funcional como estéticamente. Por lo tanto, se sacrificaron detalles y se dejó de profundizar en ciertos aspectos con el fin de cumplir con todos los objetivos.

Existen varias líneas que se podrían seguir para mejorar y extender lo desarrollado en este trabajo. A continuación se listan líneas que se podrían tomar al construir sobre lo desarrollado,

- Algo necesario y muy beneficioso para el proyecto sería una serie de pruebas de usabilidad del sistema, es decir todas sus partes, contando con la participación de personas no relacionadas al desarrollo. Los datos recolectados de estas pruebas serían muy útiles para corregir errores tanto de programación como de diseño de interfaz, y para determinar a qué líneas de desarrollo darles mayor prioridad.
- En este trabajo, se trató únicamente con imágenes como recursos. Puede que esto sea suficiente para describir obras como cuadros o fotografías, pero el objetivo es que también se puedan visualizar obras de otros tipos, como esculturas, audio, vídeo, o cosas más inusuales como sistemas de luces u obras no estáticas. Cada tipo adicional de recurso que es posible manejar tanto con la aplicación del curador como la de la visita, hace mucho más importante el sistema, tanto por la cantidad de obras que se pueden incluir en los proyectos, como por las nuevas posibilidades de interacción que se le hacen disponibles a los usuarios (especialmente a las visitas).
- Mientras que la definición de un nuevo tipo de captura no obliga a hacer cambios en el formato de los archivos de especificación del proyecto (`mem1.xsd`), es posible que se determine que le falta o le sobra algo. El personal del museo podría determinar que es necesario agregar más atributos a las obras y capturas, puede que se ideen mejores maneras de especificar los distintos elementos, o puede que se quiera asemejarlo más a algún otro estándar en existencia, como ARCO. Ninguno de los formatos, ni `mem1.xsd` ni `info.xsd`, son invariables e intocables.
- Se le podría dedicar tiempo a la mejora de las interfaces de `bnac-editor` y `bnac-visitor`. Especialmente si se hace tras pruebas de usabilidad, se puede determinar que hay funciones que requieren accesos rápidos por su frecuencia de uso, que hay funciones que siempre se usan en conjunto y que conviene fusionarlas, etc. Básicamente, esta línea consiste en pulir y depurar la interfaz usuario para facilitar su uso.
- Se le podría agregar funcionalidad a `bnac-editor`. En el caso de la Vista de simulación, el estado actual de la aplicación permite generar escenas bastante básicas. Por ejemplo, no hay control sobre las fuentes de luz, o los colores y/o texturas usadas para los muros, el techo y el piso, y así muchos otros aspectos de la simulación. Hay bastante espacio para hacer más realistas las escenas, lo cual beneficiaría al curador del museo al momento

de planificar una exhibición.

- La aplicación para las visitas tiene mucho potencial para otros tipos de interacción que no están limitados a texto e imágenes. Las posibilidades están limitadas en parte por el dispositivo sobre el cual se ejecuta la aplicación, pero algunas posibilidades son hacer posible la reproducción de audio y vídeo, e incluso visualización tridimensional de las salas, de igual forma que en la Vista de simulación de la aplicación del curador.
- Hay muchos tipos distintos de interacción que puede tener la visita con el museo mediante el dispositivo. En este trabajo se trató con texto e imágenes estáticas, pero sería interesante implementar contenido dinámico. Elementos como cuestionarios, rompecabezas, o cualquier juego educativo, resultan posibilidades muy importantes y potencialmente muy enriquecedoras para la experiencia de las visitas.
- La interacción entre visitas implementadas en este trabajo consistió en acceder a comentarios asociados a capturas y a los contadores de «Me gusta». Este tipo de interacción es bastante básico, mientras que hay posibilidades mucho más dinámicas y atractivas. Por ejemplo, se podrían considerar juegos multiusuario en tiempo real durante los recorridos, o también recorridos en conjunto utilizando dispositivos separados, etc.
- Con respecto al servidor de comentarios, se le podrían agregar funciones adicionales. Por ejemplo, se podría cambiar el protocolo para permitir la actualización de los proyectos de museo en los dispositivos de las visitas, por ejemplo para acceder a más páginas de información.

Otro aspecto importante es la seguridad e integridad de los datos que maneja. A medida que se exponga a más público, aumenta la probabilidad de que sea víctima de un ataque digital, por lo cual sería necesario implementar medidas para protegerlo de estos ataques.

- Por último, recordando que el objetivo principal de la aplicación de las visitas es ser usado durante el recorrido, físicamente dentro del museo, sería útil para la visita contar con ayuda con respecto a su ubicación en el museo, o bien la ubicación de distintas obras que desee encontrar.

Para esto se podría considerar, por ejemplo, agregar rótulos con código de barras próximos a las obras, de forma que la visita lea el código con su dispositivo, y éste automáticamente le muestre la información de la obra que señaló. Esto también se podría usar para ayudarlo a encontrar su ubicación, mostrándole un plano del museo con su posición estimada.

Bibliografía y referencias

- [Abowd et al., 1997] Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., Pinkerton, M., and Centre, U. (1997). Cyberguide: A mobile context-aware tour guide. **9**
- [Aoki and Woodruff, 2000] Aoki, P. M. and Woodruff, A. (2000). Improving electronic guidebook interfaces using a task-oriented design approach. In *Proc. DIS 2000*, pages 319–325. ACM Press. **9**
- [Bellotti et al., 2002] Bellotti, F., Berta, R., De Gloria, A., and Margarone, M. (2002). User testing a hypermedia tour guide. *IEEE Pervasive Computing*, 1(2):33–41. **10**
- [Carrozzino et al., 2008] Carrozzino, M., Evangelista, C., Scucces, A., Tecchia, F., Tennirelli, G., and Bergamasco, M. (2008). The virtual museum of sculpture. In *DIMEA '08: Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, pages 100–106, New York, NY, USA. ACM.
- [Cheng, 2011] Cheng, J. Android tops everyone in 2010 market share; 2011 may be different [en línea]. (2011). Disponible en: <http://arstechnica.com/gadgets/news/2011/01/android-beats-nokia-apple-rim-in-2010-but-firm-warns-about-2011.ars> [consulta: 18/02/2011]. **43**
- [Codocedo and Astudillo, 2008] Codocedo, V. and Astudillo, H. (2008). No mining, no meaning: relating documents across repositories with ontology-driven information extraction. In *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, pages 110–118, New York, NY, USA. ACM.
- [Fleck et al., 2002a] Fleck, M., Frid, M., O'Brien-Strain, E., Kindberg, T., Rajani, R., and Spasojevic, M. (2002a). From informing to remembering: Deploying a ubiquitous system in an interactive science museum. In *in an Interactive Science Museum", IEEE Pervasive Computing Magazine, April-June*, pages 13–21. **10**
- [Fleck et al., 2002b] Fleck, M., Frid, M., O'Brien-Strain, E., Kindberg, T., Rajani, R., and Spasojevic, M. (2002b). Rememberer: A tool for capturing museum visits. In *Proc. Ubicomp 2002*, pages 48–55. Springer-Verlag. **10**
- [Grinter et al., 2002] Grinter, R. E., Aoki, P. M., Hurst, A., Szymanski, M. H., Thornton,

- J. D., and Woodruff, A. (2002). Revisiting the visit: Understanding how technology can shape the museum visit. In *In Proc. ACM Conf. on Computer Supported Cooperative Work*, pages 146–155. ACM Press. 10
- [Houser et al., 2002] Houser, C., Thornton, P., and Kluge, D. (2002). Mobile learning: Cell phones and pdas for education. In *ICCE '02: Proceedings of the International Conference on Computers in Education*, page 1149, Washington, DC, USA. IEEE Computer Society. 2
- [Mourkoussis et al., 2003] Mourkoussis, N., White, M., Patel, M., Chmielewski, J., and Walczak, K. (2003). Ams: metadata for cultural exhibitions using virtual reality. In *DCMI '03: Proceedings of the 2003 international conference on Dublin Core and metadata applications*, pages 1–10. Dublin Core Metadata Initiative. 10, 16
- [Skov and Ingwersen, 2008] Skov, M. and Ingwersen, P. (2008). Exploring information seeking behaviour in a digital museum context. In *IIX '08: Proceedings of the second international symposium on Information interaction in context*, pages 110–115, New York, NY, USA. ACM. 11
- [Soloway et al., 2001] Soloway, E., Norris, C., Blumenfeld, P., Fishman, B., Krajcik, J., , and Marx, R. (2001). Log on education: Handheld devices are ready-at-hand. *Commun. ACM*, 44(6):15–20. 2
- [Stimac, 2010] Stimac, B. Smartphones account for almost 20% of all shipped phones [en línea]. (2010). Disponible en: <http://www.intomobile.com/2010/10/26/smartphones-adoption-phones-penetration/> [consulta: 18/02/2011]. 5
- [Sylaiou et al., 2008] Sylaiou, S., Economou, M., Karoulis, A., and White, M. (2008). The evaluation of ARCO: a lesson in curatorial competence and intuition with new technology. *Comput. Entertain.*, 6(2):1–18.
- [Tordai et al., 2007] Tordai, A., Omelayenko, B., and Schreiber, G. (2007). Thesaurus and metadata alignment for a semantic e-culture application. In *K-CAP '07: Proceedings of the 4th international conference on Knowledge capture*, pages 199–200, New York, NY, USA. ACM.
- [Walczak and Wojciechowski, 2005] Walczak, K. and Wojciechowski, R. (2005). Dynamic creation of interactive mixed reality presentations. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 167–176, New York, NY, USA. ACM.
- [Wojciechowski et al., 2004] Wojciechowski, R., Walczak, K., White, M., and Cellary, W. (2004). Building Virtual and Augmented Reality museum exhibitions. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, pages 135–144, New York, NY, USA. ACM.
- [Woodruff et al., 2001a] Woodruff, A., Aoki, P. M., Hurst, A., and Szymanski, M. H. (2001a).

Electronic guidebooks and visitor attention. In *Proceedings of the International Conference on Cultural Heritage and Technologies in the Third Millennium*, pages 437–454. 2, 9

[Woodruff et al., 2001b] Woodruff, A., Aoki, P. M., Hurst, A., and Szymanski, M. H. (2001b). The guidebook, the friend, and the room: visitor experience in a historic house. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 273–274, New York, NY, USA. ACM. 9

[Woodruff et al., 2001c] Woodruff, A., Szymanski, M. H., Aoki, P. M., and Hurst, A. (2001c). The conversational role of electronic guidebooks. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 187–208, London, UK. Springer-Verlag. 2, 5, 9

[Yatani et al., 2004] Yatani, K., Onuma, M., Sugimoto, M., and Kusunoki, F. (2004). Musex: A system for supporting children’s collaborative learning in a museum with pdas. *Syst. Comput. Japan*, 35(14):54–63. 10

Apéndices

A. Archivo mem1.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Utility types -->

  <xs:simpleType name="percentage">
    <xs:restriction base="xs:decimal">
      <xs:minInclusive value="0.0"/>
      <xs:maxInclusive value="1.0"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="attribute">
    <xs:attribute name="name" type="xs:string"/>
    <xs:attribute name="value" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="point2D">
    <xs:attribute name="x" type="xs:decimal" use="required"/>
    <xs:attribute name="y" type="xs:decimal" use="required"/>
  </xs:complexType>

  <!-- dimension in metres -->
  <xs:simpleType name="dimension">
    <xs:restriction base="xs:decimal">
      <xs:minInclusive value="0.0"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="resource">
    <xs:simpleContent>
      <xs:extension base="xs:string">
```

```

        <xs:attribute name="type" type="xs:string" use="required"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>

<!-- Abstract types -->

<xs:complexType name="object">
    <xs:sequence>
        <xs:element name="attribute" type="attribute"
            minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="author" type="xs:string"/>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="type" type="xs:string"/>
        <xs:element name="location" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>

<xs:complexType name="capture">
    <xs:sequence>
        <xs:element name="attribute" type="attribute"
            minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="author" type="xs:string"/>
        <xs:element name="date" type="xs:date"/>
        <xs:element name="resource" type="resource"/>
        <xs:element name="preview" type="resource"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="ref" type="xs:IDREF" use="required"/>
</xs:complexType>

<!-- Object children -->

<xs:complexType name="painting">
    <xs:complexContent>
        <xs:extension base="object">
            <xs:sequence>
                <xs:element name="width" type="dimension"/>
                <xs:element name="height" type="dimension"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="sculpture">
    <xs:complexContent>

```

```

    <xs:extension base="object">
      <xs:sequence>
        <xs:element name="width" type="dimension"/>
        <xs:element name="height" type="dimension"/>
        <xs:element name="depth" type="dimension"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Capture children -->

<xs:complexType name="capture2D">
  <xs:complexContent>
    <xs:extension base="capture"/>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="capture3D">
  <xs:complexContent>
    <xs:extension base="capture"/>
  </xs:complexContent>
</xs:complexType>

<!-- Hall types -->

<xs:complexType name="placedCapture">
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="ref" type="xs:IDREF" use="required"/>
  <xs:attribute name="wall" type="xs:nonNegativeInteger"
    use="required"/>
  <xs:attribute name="x" type="dimension" use="required"/>
  <xs:attribute name="y" type="dimension" use="required"/>
  <xs:attribute name="angle" type="xs:decimal" default="0.0"/>
  <xs:attribute name="scale" type="dimension" default="1.0"/>
</xs:complexType>

<xs:complexType name="door">
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="wall" type="xs:nonNegativeInteger"
    use="required"/>
  <xs:attribute name="w" type="dimension" default="1.4"/>
  <xs:attribute name="h" type="dimension" default="2.5"/>
  <xs:attribute name="x" type="dimension" default="0.7"/>
  <xs:attribute name="y" type="dimension" default="1.25"/>
  <xs:attribute name="angle" type="xs:decimal" default="0.0"/>
  <xs:attribute name="scale" type="dimension" default="1.0"/>

```

```

</xs:complexType>

<xs:complexType name="link">
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="door1" type="xs:IDREF" use="required"/>
  <xs:attribute name="door2" type="xs:IDREF" use="required"/>
</xs:complexType>

<xs:complexType name="startingPoint">
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="x" type="dimension" use="required"/>
  <xs:attribute name="y" type="dimension" use="required"/>
  <xs:attribute name="angle" type="xs:float" use="required"/>
</xs:complexType>

<xs:complexType name="hall">
  <xs:sequence>
    <xs:element name="corner" type="point2D" minOccurs="3"
      maxOccurs="unbounded"/>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="door" type="door"/>
      <xs:element name="placedCapture" type="placedCapture"/>
    </xs:choice>
    <xs:choice minOccurs="1" maxOccurs="unbounded">
      <xs:element name="startingPoint" type="startingPoint"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="height" type="dimension" use="required"/>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="route">
  <xs:sequence>
    <xs:element name="capture" type="xs:IDREF"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>

<!-- Translation types -->
<xs:complexType name="entry">
  <xs:sequence>
    <xs:element name="pair" type="attribute"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="lang" type="xs:string" use="required"/>

```

```

</xs:complexType>

<xs:complexType name="elem">
  <xs:sequence>
    <xs:element name="entry" type="entry" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="strings">
  <xs:sequence>
    <xs:element name="elem" type="elem" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- Top-level tags -->

<xs:element name="mem1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="attribute" type="attribute"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="painting" type="painting"/>
        <xs:element name="sculpture" type="sculpture"/>
      </xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="capture2D" type="capture2D"/>
        <xs:element name="capture3D" type="capture3D"/>
      </xs:choice>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="hall" type="hall"/>
      </xs:sequence>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="link" type="link"/>
      </xs:sequence>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="route" type="route"/>
      </xs:sequence>
      <xs:sequence minOccurs="0">
        <xs:element name="strings" type="strings"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

B. Archivo info.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="image">
    <xs:attribute name="src" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="entry">
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="lang" type="xs:string"
      use="required"/>
  </xs:complexType>

  <xs:complexType name="page">
    <xs:sequence>
      <xs:element name="image" type="image" minOccurs="0"/>
      <xs:element name="entry" type="entry" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="next" type="xs:IDREF"/>
  </xs:complexType>

  <xs:element name="info">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="page" type="page"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```