



# TESINA DE LICENCIATURA

**Título:** Herramienta para la creación de elementos posicionados

**Autores:** Alconada Verzini, Federico Martín

**Director:** Dra. Cecilia Challiol

**Codirector:** Dra. Silvia Gordillo

**Asesor profesional:**

**Carrera:** Licenciatura en Sistemas

## Resumen

*Las aplicaciones móviles basadas en posicionamiento brindan a los usuarios, en determinadas posiciones relevantes, información (contenido) acorde al dominio de dicha aplicación. Estas aplicaciones pueden estar destinadas a proveer servicios tanto sea en espacios indoor como outdoor, o en una combinación de ambos. Esto genera la necesidad de que la aplicación cuente con la representación de dicho espacio para poder guiar al usuario mientras se mueve en el mismo. La aplicación puede estar planteada (acorde a su naturaleza) para que el usuario recorra posiciones, por ejemplo, libremente o siguiendo un recorrido secuencial.*

*Por otra parte, algunas de las herramientas existentes están acopladas al mecanismo de sensado, lo que restringe y limita el dominio de la aplicación y en otros casos, dan soporte para la construcción in-situ, es decir, permiten construir elementos en el lugar donde la aplicación correrá permitiendo al usuario considerar características del ambiente.*

*Es por lo mencionado anteriormente que surge la motivación de desarrollar una herramienta para usuarios no expertos que permita construir elementos posicionados tanto para espacios indoor como outdoor.*

## Palabras Claves

*Herramienta de Autor In-situ; Definición de elementos posicionados; Espacios Indoor-Outdoor; Separación de Concerns; Aplicaciones Móviles Basadas en Posicionamiento*

## Trabajos Realizados

*Primero se realizó una conceptualización de las características que debería cumplir una herramienta para permitir la creación de elementos posicionados. Esta caracterización se realizó en base al análisis bibliográfico sobre la temática.*

*Tomando como base la caracterización realizada, se desarrolló una herramienta para la creación de elementos posicionados, la cual permite que los mismos estén posicionados tanto en espacios indoor como outdoor. El usuario puede optar por usar como mecanismos de sensado GPS o códigos QR.*

## Conclusiones

*En esta tesis se presentó una herramienta de creación de elementos posicionados in-situ que permite, a usuarios no expertos, definir experiencias con contenidos multimedia posicionados (como por ejemplo, audios, videos, imágenes, textos o una combinación de ellos) tanto en espacios indoor como outdoor. Dicha herramienta está definida para poder ser extensible, y así brindar en un futuro nuevas funcionalidades. Se presentó un ejemplo de uso de creación para mostrar el funcionamiento de la misma.*

## Trabajos Futuros

*Combinar la herramienta con un framework capaz de generar aplicaciones móviles ejecutables en base a los elementos posicionados definidos en una experiencia.*

*Permitir que el usuario pueda subir tanto las capas de contenidos como las de posiciones a un repositorio.*

*Dar soporte a otros mecanismos de sensado, por ejemplo, NFC.*

*Soportar más servicios de mapas. Es decir, integrar la herramienta con otras APIs de mapas.*

A mis papás y hermanos, por el apoyo, la motivación y por estar siempre presentes.  
Al laboratorio LIFIA, en particular a Gustavo Rossi, por su ayuda para mis futuros proyectos.  
A mi directora de tesis, Cecilia Challiol, por la ayuda, guía, formación y dedicación de manera desinteresada, excediendo lo estrictamente académico.

# ÍNDICE

<b>1. INTRODUCCIÓN3</b>	
1.1 Motivación .....	3
1.2 Objetivos .....	3
1.3 Estructura de la tesis .....	4
<b>2. BACKGROUND .....</b>	<b>5</b>
2.1 Construcción aplicaciones móviles indoor-outdoor .....	5
2.1.1 CASTOR .....	6
2.1.2 QuesTInSitu .....	9
2.2 Librerías utilizadas .....	19
2.2.1 Leaflet .....	19
2.2.2 jsPlumb .....	21
2.2.3 XMLWriter para Javascript .....	23
<b>3. DEFINICIÓN CONCEPTUAL DE UNA HERRAMIENTA DE CREACIÓN IN-SITU DE APLICACIONES MÓVILES INDOOR-OUTDOOR.....</b>	<b>25</b>
3.1 Conceptos abstractos usados como base de la Herramienta .....	26
3.1.1 Capa de contenidos .....	27
3.1.2 Capa de posicionamiento .....	28
3.1.3 Relación entre las capas de contenido y posicionamiento.....	28
3.2 Interfaz para la construcción de elementos posicionados .....	29
<b>4. HERRAMIENTA PROTOTÍPICA.....</b>	<b>30</b>
4.1 Crear contenidos .....	30
4.2 Crear posiciones .....	35
4.3 Crear elementos posicionados .....	42
4.4 Importación/Exportación.....	44
4.5 Eliminación .....	53
<b>5. CASOS DE USO DEL PROTITIPO PROPUESTO.....</b>	<b>54</b>
<b>6. CONCLUSIONES Y TRABAJOS FUTUROS .....</b>	<b>67</b>

<b>7. REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>70</b>
<b>8. PUBLICACIONES REALIZADAS .....</b>	<b>72</b>

## **1. INTRODUCCIÓN**

### **1.1 Motivación**

Las aplicaciones móviles basadas en posicionamiento brindan a los usuarios, en determinadas posiciones relevantes, información (contenido) acorde al dominio de dicha aplicación. Estas aplicaciones pueden estar destinadas a proveer servicios tanto sea en espacios indoor como outdoor, o en una combinación de ambos. Esto genera la necesidad de que la aplicación cuente con la representación de dicho espacio para poder guiar al usuario mientras se mueve en el mismo. La aplicación puede estar planteada (acorde a su naturaleza) para que el usuario recorra posiciones, por ejemplo, libremente o siguiendo un recorrido secuencial. En [Pittarello & Bertani, 2012], se presenta una herramienta llamada CASTOR orientada a niños que permite la construcción in-situ de historias estructuradas y sensibles al contexto.

La mayoría de las herramientas que existen actualmente están orientadas a usuarios expertos como puede verse en [Bouvin et. al,2003], cuyos autores presentan HyCon, un framework orientada a usuarios expertos que da soporte a sistemas de Hipermedia context-aware que permite el manejo de links y tours guiados asociando ubicaciones. Es decir, este framework permite unir objetos digitales y/o físicos entre sí. Sin embargo, existen algunos desarrollos que están orientados a usuarios no expertos como [Pittarello & Bertani, 2012], [Santos et. al, 2011] y [Toursphere].

Tanto [Pittarello & Bertani, 2012], [Santos et. al, 2011] y [Toursphere] permiten trabajar solamente sobre espacios físicos outdoor, mientras que en [Realinho et. al, 2011a], [Realinho et. al, 2011b], [Realinho et. al, 2011c] y [Realinho et. al, 2012] se presenta la plataforma IVO que permite trabajar tanto en espacios outdoor como indoor.

Por otra parte, éstas herramientas están acopladas al mecanismo de sensado, lo que restringe y limita el dominio de la aplicación. Es decir, los datos que se generan están acoplados al mecanismo de sensado que terminará usando la aplicación resultante.

Algunas de las herramientas mencionadas dan soporte para la construcción in-situ en espacios outdoor, es decir, permiten construir elementos en el lugar donde la aplicación correrá permitiendo al usuario considerar características del ambiente.

Es por lo mencionado anteriormente que surge la motivación de desarrollar una herramienta para usuarios no expertos que permita construir elementos posicionados tanto para espacios indoor como outdoor. Esta herramienta permitirá definir los elementos posicionados de manera desacoplada del mecanismo de sensado.

### **1.2 Objetivos**

Se propone una tesina cuyo objetivo principal es facilitar la creación de elementos posicionados mediante el desarrollo de una herramienta prototípica. Esta herramienta está orientada a usuarios no expertos y permite definir, en posiciones específicas, diferentes contenidos como por ejemplo, textos, videos, imágenes, etc. Se utilizarán mapas de base para facilitar el posicionamiento de los elementos.

La herramienta permite la construcción de experiencias in-situ y dará soporte para la representación de espacios indoor, outdoor y espacios combinados (indoor-outdoor).

El usuario podrá indicar qué mecanismo de sensado estará asociado a la aplicación que se quiere generar, teniendo también diferentes alternativas como por ejemplo GPS o códigos QR.

La herramienta está pensada para ser utilizada tanto desktop como in situ (desde un dispositivo móvil) permitiendo a través de ésta última opción considerar características del ambiente. Se asiste al usuario para que el posicionamiento de los contenidos se realice acorde a su posición actual.

La herramienta permite la exportación de contenidos independientemente del espacio, esto posibilitará el reúso de los mismos para que sean posicionados en lugares físicos diferentes. Es decir, un usuario define los contenidos una sola vez y los puede usar para generar aplicaciones móviles que van a ejecutarse en espacios físicos diferentes. Esta misma flexibilidad se provee para el espacio de representación y posiciones, donde al exportar esta información, luego se puede asociar la misma en diferentes contenidos.

### **1.3 Estructura de la tesis**

La tesis se organiza de la siguiente forma:

- En la Capítulo 2 se describen la temática de la construcción de aplicaciones móviles indoor-outdoor citando algunas herramientas existentes.
- En la Capítulo 3, se presentan las características principales que debería tener una herramienta de creación in-situ para lograr crear aplicaciones móviles tanto para espacios indoor como outdoor.
- En la Capítulo 4 y teniendo en cuenta lo que se menciona en la sección anterior, se describe y explica el funcionamiento de una herramienta prototípica que permite la crear los conceptos mencionados, es decir, contenidos, posiciones o elementos posicionados.
- En la Capítulo 5, se ve un ejemplo de un caso de uso de la herramienta. Utilizando como base un tour existente, se crea una experiencia con la herramienta descrita en la Sección 4.
- En la Capítulo 6 se detallan las conclusiones de la investigación realizada así como también se listan algunos trabajos a futuro que surgen como mejoras a la herramienta y como ampliación de sus características.
- Por último, en la Capítulo 7 se presenta una publicación realizada de la herramienta presentada en este trabajo.

## **2. BACKGROUND**

En este apartado se tratará, por un lado, la temática de la construcción de aplicaciones móviles indoor-outdoor, citando algunas herramientas de autor existentes y, por otra parte, se explicarán las tecnologías existentes que pueden servir de base para la construcción de este tipo de herramientas.

### **2.1 Construcción aplicaciones móviles indoor-outdoor**

En la actualidad hay una variada gama de aplicaciones móviles que pueden correr en un dispositivo, en el marco de esta tesis nos interesa analizar aquellas que brindan información relevante acorde a la posición del usuario. A este tipo de aplicaciones se las conoce como aplicaciones móviles basadas en posicionamiento. Estas pueden ser creadas para ser usadas en espacios abiertos (outdoor), como por ejemplo una ciudad, o para espacios cerrados (indoor), por ejemplo un edificio. Según [Literas et. al, 2011], una característica relevante que poseen las aplicaciones móviles basadas en posicionamiento es asistir a sus usuarios para moverse en el mundo real. Para ello se valen, por ejemplo, de un mapa en el cual se muestra la posición actual del usuario y un conjunto de puntos de interés cercanos a esa posición.

Para obtener la posición actual del usuario se pueden usar diferentes mecanismos de sensado, estos varían de acuerdo al espacio físico (indoor-outdoor) en el que se encuentra dicho usuario, por ejemplo, para espacios outdoor, el sistema de posicionamiento GPS (Global Positioning System) es muy usado. Sin embargo, en espacios indoor [Hansen et. al, 2009] se usan otros sistemas de posicionamiento, por ejemplo WIFI, RFID [Gu, 2009] y códigos QR [Lai, 2010], dado que el GPS brinda una señal muy débil en este tipo de espacios.

La construcción de este tipo de aplicaciones móviles generalmente es realizada por expertos en el área y se presentan de manera ad-hoc, limitando así la construcción por parte de usuarios no expertos. En [Santos et. al, 2011] se presenta la herramienta QuesTInSitu, que tiene la ventaja de estar orientada a usuarios no expertos y permite crear, a profesores o docentes, actividades de evaluación asociadas a posiciones. A su vez, ofrece la derivación de esta construcción en una aplicación móvil dirigida a estudiantes para que éstos tomen las evaluaciones de manera in-situ.

Por otra parte, en [Pittarello & Bertani, 2012] se presenta una herramienta de autor denominada CASTOR que permite construir a usuarios no expertos, en particular a niños de 7 años, narraciones sensibles al contexto.

Una de las ventajas de CASTOR respecto a QuesTInSitu es que la construcción es realizada de forma in-situ, pudiendo así crear en el lugar donde luego se utilizará la aplicación derivada. De esta forma, la construcción se realiza teniendo noción de los diferentes aspectos del contexto que son relevantes para la aplicación móvil derivada, como ser el espacio físico, el clima, el tiempo, etc.

Estas herramientas representan los espacios físicos outdoor utilizando servicios de mapas como Google Maps pero no cuentan con soporte para espacios indoor.

Los autores de ([Realinho et. al, 2011a], [Realinho et. al, 2011b], [Realinho et. al, 2011c] y [Realinho et. al, 2012]) presentan la plataforma IVO (Integrated Virtual Operator), una arquitectura que permite a usuarios finales crear y derivar aplicaciones sensibles al contexto de manera rápida y a través del uso de teléfonos inteligentes para generar aplicaciones de diferentes dominios (guías turísticas, entretenimiento, etc.). A diferencia de CASTOR y QuestInSitu, la plataforma IVO permite también su uso en ambientes indoor.

Se detallan a continuación las herramientas mencionadas anteriormente: CASTOR, QuesTInSitu y la plataforma IVO.

### **2.1.1 CASTOR**

En [Pittarello & Bertani, 2012] se presenta CASTOR, una herramienta de autor orientada a niños de 7 años que permite la construcción directa e in-situ de historias estructuradas interactivas donde las diferentes dimensiones del contexto juegan un papel fundamental en la evolución de la *narración*.

#### **Modelo de la herramienta**

El modelo describe a la *situación* como el componente básico de la narración. Las situaciones son los componentes activos de la historia y son referidas a un *escenario* que caracteriza la *escena* de la historia. Una historia puede incluir una o más escenas, dependiendo de la necesidad de definir uno o más escenarios y cada situación se asocia a una posición específica y con un fragmento narrativo dado.

Un conjunto de situaciones es usualmente asociada a la escena, pero sólo un subconjunto de ellos está activo en un determinado momento para que el oyente de la historia pueda escuchar los fragmentos narrativos al ingresar en la posición asociada a éstos. De ahora en más, usaremos el término genérico *oyente* para hacer referencia al usuario que recibe el contenido de la historia a través de diferentes canales audiovisuales y el término *etapa* para referirnos a una situación. La regla para definir qué situaciones deben estar activas en determinado momento son definidos por el autor de la historia y pueden originar diferentes **estructuras** para la historia.

El modelo es lo suficientemente flexible como para acomodarse a la construcción tanto de narraciones con una estructura lineal como con una estructura no lineal.

#### **Estructura de las historias**

Dado que los usuarios de la herramienta son niños de 7 años, CASTOR limita la expresividad del modelo a la posibilidad de generar historias de tres tipos de estructuras (ver Figura 1):

1. Historia con una sola etapa
2. Historia secuencial
3. Historia con una ramificación que lleve a dos finales diferentes.

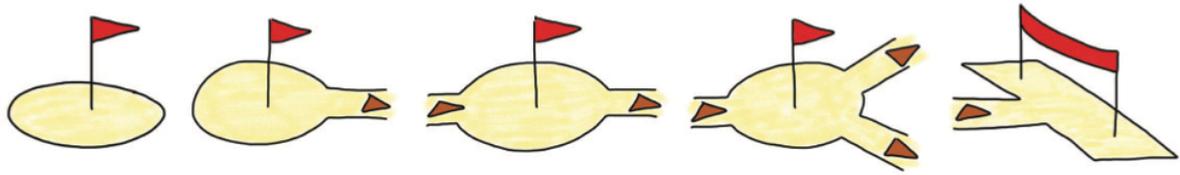


Figura 1. Íconos describiendo los diferentes tipos de etapas [Pittarello & Bertani, 2012].

### Dimensiones de las historias

De acuerdo al modelo descrito anteriormente, las narraciones comienzan su relato no sólo por la acción del usuario sino también por el contexto ambiental y social.

Las dimensiones son (ver Figura 2):

- Tiempo (amanecer, día, crepúsculo, noche)
- Estación (primavera, verano, otoño, invierno)
- Clima (soleado, parcialmente nublado, nublado, chubascos, tormenta, aguanieve)
- Oyentes (solo, en grupo)

Para cada parámetro de contexto se brinda la posibilidad de seleccionar el valor de 'siempre', para indicar que el contenido debe mostrarse para todos los valores de cierta dimensión.



Figura 2. Conjunto de íconos que describen las dimensiones del contexto [Pittarello & Bertani, 2012].

En lo que concierne al contexto social se puede elegir que el contenido sea mostrado ante la presencia de un solo oyente o de un grupo de oyentes.

### Diseño de la herramienta

La implementación está basada principalmente en el uso de tecnologías web. La aplicación cliente guía al usuario en los pasos necesarios para la construcción de una historia y se encarga de recolectar toda la información relacionado a al estructura de la historia, los contenidos audiovisuales asociados a las diferentes etapas y la especificación de los valores de contexto asociados a las situaciones. Se utiliza el módulo GPS de la tablet para obtener las posiciones asociadas a cada etapa. Los datos recibidos son periódicamente enviados al servidor que los

almacena para la futura elaboración y presentación de la narración.

En cuanto a la parte del servidor, tiene la capacidad de recuperar, desde la red, los valores actuales de las dimensiones del contexto ambiental para una determinada posición (por ejemplo, el clima). Durante la ejecución de la narración generada por algún alumno estos valores son fundamentales para poder presentar la narración de manera adecuada. En caso de que el clima esté soleado, este valor será brindado por el servidor a la aplicación y ésta mostrará aquellas etapas de la narración cuyos valores (de la dimensión clima) coincidan con dicho valor.

La implementación de la herramienta tiene la peculiaridad de haberse realizado utilizando el framework Phonegap [Phonegap], lo que permite que el usuario perciba la aplicación web como una aplicación nativa y que pueda ser descargada e instalada desde cualquier tienda de aplicaciones. Este framework permite utilizar un enfoque web para el desarrollo rápido de la interfaz de usuario y el lanzamiento de la aplicación en diferentes plataformas y dispositivos.

### **Funcionamiento de CASTOR**

La Figura 3 muestra una colección de screenshots de la interfaz de CASTOR y los posibles caminos de interacción para la construcción de una historia. Luego de la fase de autenticación, la interfaz guía a los usuarios en la construcción, lo que requiere especificar al inicio la información general (Figura 3.1) y los autores (Figura 3.2). Luego, la interfaz lleva a los autores a elegir un tipo de estructura para la historia y a componer incrementalmente las diferentes etapas que son parte de la narración.

Una vez generada una etapa se pueden generar más. La Figura 3.4 muestra cómo puede agregarse otra etapa o cómo terminar el proceso de construcción.

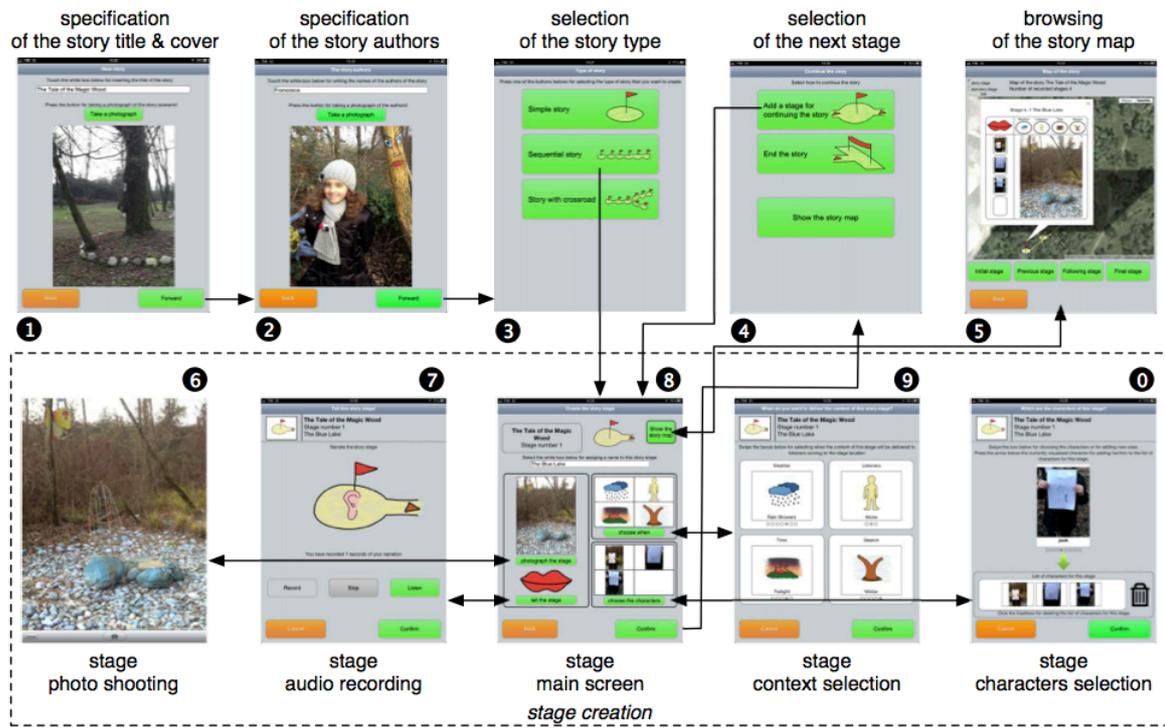
Finalmente, en la Figura 3.5 se muestra la interfaz de mapa de la historia, realizada utilizando la API de Google Maps. El mapa muestra una historia secuencial visualizada a través de un número de etapas conectadas y ordenadas. El autor puede seleccionar cualquiera de ellas para visualizarla.

En las Figuras 3.6 y 3.7 puede verse cómo el usuario puede agregar una foto o un audio a la composición de la etapa respectivamente. En la etapa de la toma de una foto, automáticamente se graba la posición actual del autor tomada por el GPS. Concurrentemente, la interfaz informa que la posición será utilizada por la aplicación para oír la historia, es decir, servirá como disparador de la narración en esa posición.

La Figura 3.8 muestra la pantalla principal donde se especifican todos los componentes de una situación. El ícono de la bandera que se ubica en la parte superior de la pantalla indica que ésta es la etapa inicial de la historia.

La interfaz permite a los usuarios indicar toda la información relacionada a la etapa presionando los botones asociados a los cuatro cuadrantes en el medio de la pantalla. Cada botón lleva a una pantalla auxiliar especializada para el tipo de data específico. Por ejemplo, la Figura 3.9 muestra la interfaz secundaria para la especificación de los valores de contexto.

Además, a través de las representaciones en miniatura de esta figura, se puede ver que toda la información especificada en la etapa: la foto, 4 valores de contexto, 3 personajes. La presencia de la boca roja indica que los datos de audio relacionados a la etapa han sido grabados y pueden ser oídos.



**Figura 3. Conjunto de capturas de pantalla de la interfaz y los posibles caminos de interacción para la construcción de una historia secuencial [Pittarello & Bertani, 2012].**

Si bien los autores de CASTOR no describe la forma en que las narraciones puede ser reproducidas, mencionan que éstas son almacenadas en un servidor de tal forma que en un futuro se desarrollarán una aplicación móvil que permita recuperarlas para que los usuarios puedan ver las narraciones generados por los niños.

### 2.1.2 QuesTInSitu

QuesTInSitu es una herramienta [Santos et. al, 2011] web que propone un nuevo tipo de evaluación de actividades basado en exámenes llamado “evaluación *in-situ*”, donde los usuarios utilizan un dispositivo móvil para seguir y responder una evaluación.

A diferencia de CASTOR, los autores de QuesTInSitu presentan tanto la herramienta para la construcción de evaluaciones como también la herramienta móvil que permite a los estudiantes ejecutar las evaluaciones creadas por los profesores. Si bien los estudiantes ejecutan las evaluaciones de manera *in-situ*, la construcción de éstas por parte de los profesores no lo es.

La arquitectura (ver Figura 4) comprende cuatro componentes diferentes:

1. Módulo de control: QuesTInSitu es una aplicación web que utiliza Java Server Pages (JSP<sup>1</sup>) utilizada para diseñar, jugar e implementar actividades de evaluación in situ. Su módulo de control fue desarrollado para que sea posible el manejo de las siguientes aplicaciones: QTI Engine, un servicio de mapas y un sistema basado en posicionamiento (location-based).
2. QTI Engine (NewAPIS): es el motor que interpreta las preguntas QTI compatibles con QTI v2.1. Este motor se caracteriza por ser capaz de tratar con las pruebas y el procesamiento de respuesta compleja. Ambos, QuesTInSitu y NewAPIS comparten la misma base de datos almacenando toda la información relacionada a las preguntas, los test-routes, los usuarios y los mapas.
3. Web Map Service (Google Maps): Este servidor provee los mapas web necesarios para el sistema para poder asociar preguntas con coordenadas reales y visualizarlas en un mapa. Google Maps puede ser sustituido en un futuro por otra aplicación de mapas web (por ejemplo, Yahoo!Maps u OpenStreetMap).
4. Location-Based System: QuesTInSitu es un sistema location-based. Esto es necesario para saber la posición del usuario y lanzar el evento con la pregunta asociada.

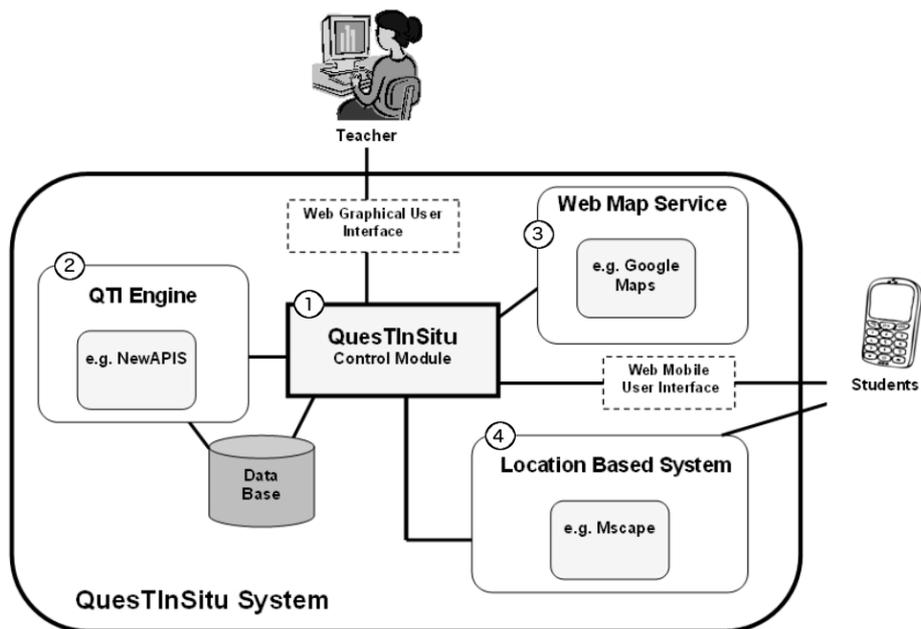


Figura 4. Arquitectura del sistema [Santos et. al, 2011].

QuesTInSitu permite a los estudiantes seguir un evaluación o ruta mientras caminan. Deben usar un teléfono inteligente con conexión a internet, GPS y cámara.

Mientras tanto, los profesores pueden monitorear el progreso de los estudiantes a través de la interfaz gráfica de usuario de QuesTInSitu.

<sup>1</sup> **JavaServer Pages (JSP)** es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

## Herramienta de construcción de actividades

Utilizando una interfaz gráfica de usuario (GUI), el autor (profesor) de estas evaluaciones tiene la capacidad de posicionar preguntas dentro de un espacio físico. Esta interfaz permite crear dos tipos de evaluaciones in situ: evaluaciones en lugares reales o virtuales.

Actualmente, el servicio de mapas utilizado para mostrar las preguntas es la API de Google Maps, pero puede ser extendida a otras aplicaciones web de mapas.

Las evaluaciones se basan principalmente en preguntas, donde cada una tiene asociada una coordenada geográfica, lo que las hace dependientes de la posición (location-based). Esto significa que un usuario debe estar posicionado en el lugar correcto para poder visualizar una pregunta. Las preguntas creadas con QuesTInSitu son compatibles con el estándar tecnológico educativo para evaluaciones llamado IMS Question & Test Interoperability<sup>2</sup> (QTI). Los estándares tecnológicos educativos para las evaluaciones son utilizados para dar soporte a la interoperabilidad y reusabilidad de los recursos de las evaluaciones.

QuesTInSitu integra un módulo que representa un motor de evaluación utilizado para interpretar preguntas QTI y para testear que son creadas con la sección de edición del sistema. QuesTInSitu no extiende QTI con nuevos tipos sino que utiliza preguntas QTI tradicionales como 'opción múltiple', 'de respuestas múltiples' y preguntas 'Sí/No'.

A continuación se presentan las principales funcionalidades de la herramienta de construcción:

Los profesores pueden utilizar QuesTInSitu para crear tres tipos de preguntas QTI: 'opción múltiple', 'de respuestas múltiples' y preguntas 'Sí/No' (ver Figura 5). Un editor amigable con el usuario muestra los pasos necesarios que se deben realizar para crear una pregunta geolocalizable. Los profesores deben introducir el texto de la pregunta, la/s respuesta/s y seleccionar la respuesta correcta y un feedback (mensaje que muestra cuál era la respuesta correcta y, además, explica los próximos pasos que el estudiante debe realizar para continuar). Luego de ello, deben geolocalizar la pregunta en una ventana de Google Maps, buscando la posición deseada y haciendo click sobre la posición de tal forma que una coordenada se asocie a la pregunta. Como resultado, la pregunta es creada y posicionada en el mapa.

---

<sup>2</sup> QTI (Question & Test Interoperability) es el estándar de facto para representar computacionalmente preguntas y evaluaciones. La especificación provee un modelo de datos interoperable para la representación de preguntas (items), evaluaciones, sus resultados y feedback.



Figure 5. QuesTnSitu creador de preguntas QTI geocalizables [Santos et. al, 2011].

Cuando los profesores han creado un conjunto de preguntas, pueden crear una evaluación (o una *ruta*, como es llamado en QuesTnSitu) utilizando la sección *Route creator*.

Primero, deben seleccionar un nombre para la evaluación y luego pueden:

1. Buscar y seleccionar en el mapa la pregunta que quieren o
2. seleccionar una pregunta desde un *combo box*.

Además, los autores pueden manejar las evaluaciones creadas, creando o eliminando preguntas. Para verificar que las preguntas de la evaluación están bien diseñadas, los autores pueden chequear éstas directamente desde la aplicación QuesTnSitu player. Para hacer esto, deben ir a la sección *Answer Questions* y seleccionar una evaluación (ruta) que quieran responder. Entonces, aparece un mapa con diferentes markers (uno por cada pregunta) y cuando uno de ellos es seleccionado, se muestra el texto con la pregunta (ver Figura 6). Utilizando esta herramienta de testeo, una vez que la pregunta es respondida, automáticamente es corregida. Cuando la respuesta es correcta, el marker se colorea en verde y, en caso contrario, se colorea en rojo. Antes de contestar una pregunta, el marker es blanco con un punto rojo en el centro..

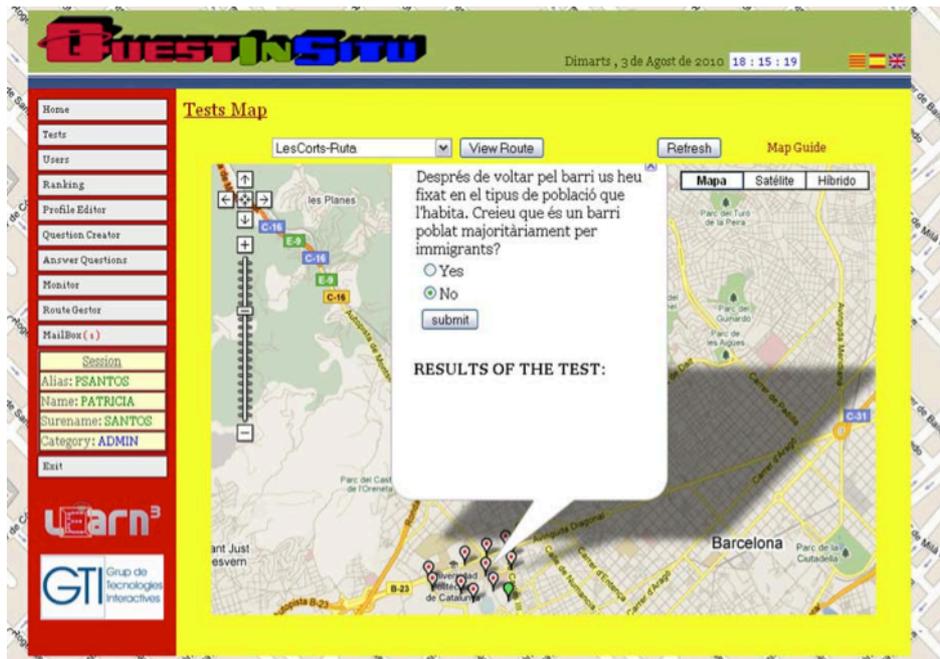


Figura 6. Testeo de preguntas [Santos et. al, 2011].

Los profesores pueden monitorear la actividad en tiempo real utilizando la función de *Monitor* de QuesTnSitu (ver Figura 7).

Luego de seleccionar una evaluación que quieran monitorear, un conjunto de markers (correspondientes a las preguntas de la evaluación elegida) aparece. Los markers coloreados en amarillo son preguntas que han sido respondidos por los estudiantes que están realizando esa evaluación. Seleccionando una de ellas, los profesores pueden ver la puntuación adquirida por estudiante en esa pregunta. Con esta información, pueden saber aproximadamente dónde los estudiante están y su progreso en la evaluación.



Figure 7. Monitoreo de la performance del grupo durante la evaluación [Santos et. al, 2011].

## Herramienta móvil de evaluación

QuesTInSitu tiene una versión web móvil que permite a los estudiantes seguir las evaluaciones de evaluación in situ, respondiendo las preguntas, obteniendo las respuestas y el puntaje de forma in situ así como también personalizando las preguntas con sus propias imágenes y comentarios.

Las preguntas deben ser respondidas *in situ* utilizando un dispositivo móvil. Luego, un espacio abierto, como una ciudad, es utilizado como canvas donde las preguntas son posicionadas y organizadas. La información de este espacio debe ser utilizada por los estudiantes para reflexionar sobre las preguntas y realizar las acciones necesarias para responder correctamente.

Si bien este sistema utiliza GPS para visualizar preguntas geolocalizables, otras tecnologías como las etiquetas NFC [Annika, 2007] o los códigos podrían ser también utilizadas durante la evaluación in situ.

A continuación se presentan las principales funcionalidades de la herramienta de evaluación:

Para usar QuesTInSitu en un lugar real, los estudiantes deben seguir la evaluación con sus teléfonos inteligentes. En el caso de que la ruta tenga infraestructura de señal GPS, cuando los estudiantes se posicionan en un radio que contiene a la pregunta geolocalizada, una pregunta aparece automáticamente en la pantalla del dispositivo móvil. Al contrario, si la zona no recibe bien la señal GPS, las preguntas deben seleccionarse manualmente por los estudiantes.

Una vez que los estudiantes responden una pregunta, el sistema la corrige y muestra el puntaje y la retroalimentación asociada a la pregunta (en verde cuando es correcta, en rojo cuando no lo es) (ver Figura 8).

La información de la pregunta contestada es guardada en el sistema. Los estudiantes pueden revisarla cuando quieran a pesar que la evaluación haya finalizado. Solo deben acceder al sistema QuesTInSitu con su usuario y contraseña y pueden ver si historial de preguntas y respuestas. Además, los estudiantes pueden enriquecer y personalizar las preguntas con comentarios e imágenes (tomadas durante la evaluación con su teléfono inteligente o cámara, o descargas de internet), ver Figura 8 (b). Estos datos son compartidos con el resto de los estudiantes que también contestaron la misma pregunta.



Figura 8. Interfaz web móvil de QuesTInSitu (a) Ejemplo de una pregunta QTI contestada visualizada desde un teléfono móvil (b) Ejemplo de una imagen y un comentario añadido por un estudiante asociados a la pregunta [Santos et. al, 2011].

### 2.1.3 Integrated Virtual Operator

IVO (Integrated Virtual Operator) ([Realinho et. al, 2011a], [Realinho et. al, 2011b], [Realinho et. al, 2011c] y [Realinho et. al, 2012]) es una arquitectura que permite a usuarios no expertos crear y ejecutar aplicaciones móviles sensibles al contexto.

IVO provee 2 herramientas de construcción (IVO Builder e IVO Outlook) que permiten crear y derivar las aplicaciones mencionadas y una aplicación cliente para ejecutar las aplicaciones derivadas. La Figura 9 muestra un gráfico de la arquitectura IVO.

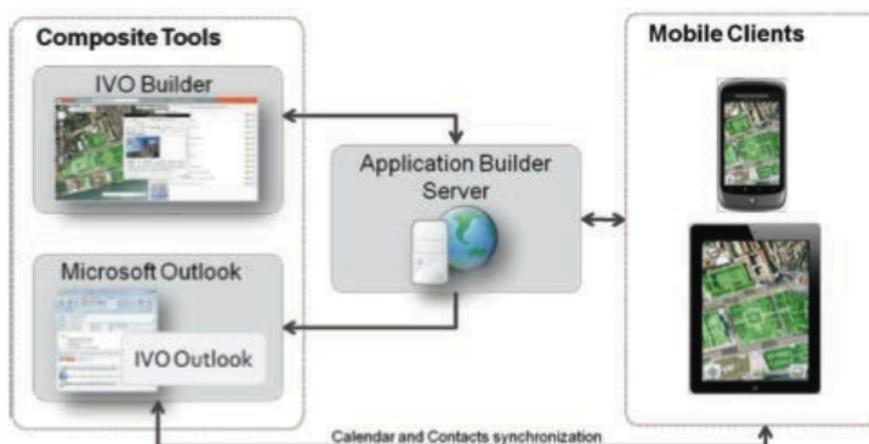
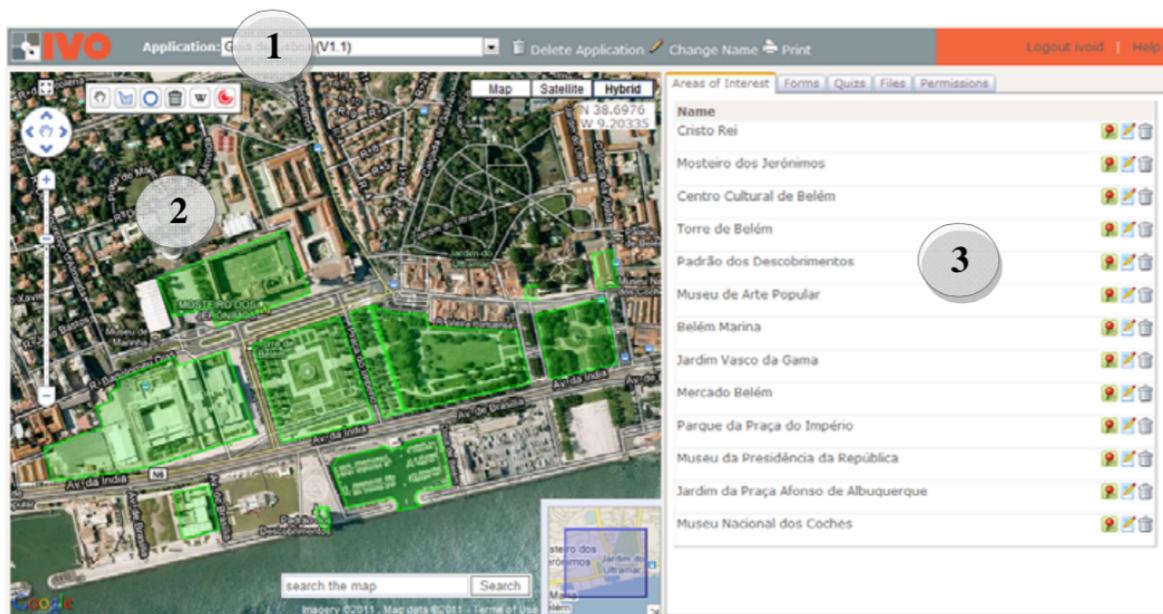


Figura 9. Arquitectura IVO [Realinho et. al, 2011].

A continuación se detallan los componentes de la arquitectura.

IVO Builder (herramienta de construcción): aplicación web que facilita, a usuarios no expertos, el desarrollo ágil de aplicaciones basadas en posicionamiento. Permite crear contextos espaciales y definir las acciones que deben ejecutarse al aproximarse a estos contextos. Por ejemplo, el usuario puede definir una aplicación que automáticamente muestre información referida a un área geográfica determinada cuando el usuario se aproxima a éste.

La Figura 10 muestra la interfaz de IVO Builder con una aplicación cargada. En la zona 1 de la figura aparecen las opciones que permiten al manejo de la aplicación (cargar, crear, cambiar, eliminar e imprimir). La zona 2 de la figura está reservada para la representación geográfica, a la cual se le superponen áreas de interés IVO (representadas en verde). La barra de tareas de este área da soporte para la construcción de áreas de interés. Por último, la zona 3 de la figura se reserva para la representación de información de la aplicación que se está construyendo, como ser, áreas de interés, cuestionarios, imágenes, audio, video, scripts.



**Figura 10. Interfaz de IVO Builder [Realinho et. al, 2011].**

Una vez creadas las áreas de interés, el usuario puede asociarlas con información de contexto. De esta manera, define los flujos de trabajo que este área de interés ejecutará cuando el usuario de la aplicación derivada ingresa o sale de ese área.

Los flujos de trabajo son flujos de actividades controlados por la información de contexto recuperada por el dispositivo móvil. La Figura 11 muestra las actividades que soporta IVO y que pueden ser ejecutadas en cada paso del flujo de trabajo.

Icon	Activity	Icon	Activity	Icon	Activity
	Change phone profile		Update status on facebook		Execute external web-service
	Make a call		Open a web page		Read IVO code (QR-code)
	Send a SMS		Enter a value		Detect nearby people or equipment
	Send a email		Speech the text entered (text-to-speech)		Run script
	Don't show information window		Show alert message		Run external application
	Play audio file		Show toast message		Navigate to a place
	Play video file		Complete form		Take a photo
	Post message on twitter		Execute quiz		Voice recognition

**Figura 11. Actividades disponibles en la construcción de un flujo de trabajo.**

Algunos ejemplos de posibles actividades son: cambiar el perfil del teléfono, ‘postear’ mensajes en redes sociales, usar la navegación del GPS, responder preguntas o usar el reconocimiento de voz.

IVO Builder permite crear cuestionarios y formularios que pueden ser usados como actividades. Un cuestionario puede contener múltiples preguntas con un máximo de 4 posibles respuestas.

IVO Outlook (herramienta de construcción): es un add-on para Outlook Calendar de Microsoft que permite definir contextos temporales y de proximidad, permitiendo así crear contextos más enriquecidos. El usuario define fácilmente condiciones temporales a través de la incorporación de flujos de trabajo en la agenda de Outlook Calendar, que son ejecutados “cuando un evento comienza” o “cuando un evento termina”. Estos flujos de trabajo tienen disponibles las mismas actividades que IVO Builder. Los usuarios pueden asociar un evento a un contexto de posición creado con IVO Builder o a una campo de dirección en los Contactos del usuario. De esta manera, el sistema puede manejar condiciones más complejas como “en este instante y cuando esté en esta posición con estas personas”. En la Figura 12 se ve la interfaz de IVO Outlook.

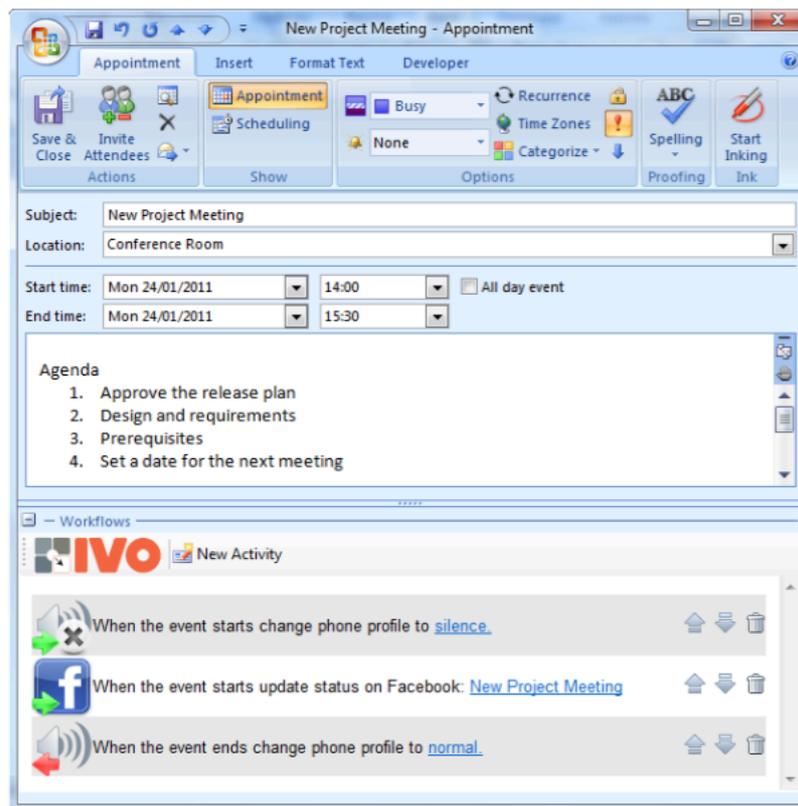


Figura 12. Interfaz de IVO Outlook [Realinho et. al, 2011].

IVO Client (aplicación cliente): Cuando una aplicación es creada, se almacena en un servidor web y queda disponible a través del cliente IVO. En la implementación existente, este cliente se ejecuta sobre dispositivos Android y su función es cargar las aplicaciones desarrolladas y se provee además, un motor de flujo de trabajo (Workflow Engine) para que coordine el flujo de actividades. Es decir, los flujos de trabajo deben ser iniciados sólo cuando el usuario está ante la presencia de un contexto pre-definido detectado por la aplicación.

Cuando el usuario entra en un área de interés, la información asociada con dicho área se muestra. Esta información incluye lo que fue definido con IVO Builder más información de contexto.

El botón de menú de la aplicación permite realizar algunas acciones:

- 1) Cargar de aplicaciones realizadas con IVO Builder o IVO Outlook
- 2) Acceder a un buscador de Realidad Aumentada
- 3) Buscar puntos de interés
- 4) Leer códigos IVO
- 5) Ejecución manual de cuestionarios y/o formularios

Los códigos IVO son códigos QR con semántica propia que pueden ser utilizados para ejecutar flujos de trabajo y son particularmente útiles cuando no funciona el GPS, por ejemplo en espacios indoor.

Las herramientas de construcción codifican las aplicaciones creadas en un formato propietario llamado IVO Markup Language (IVOML). IVOML puede ser embebido en KML<sup>3</sup>.

## 2.2 Librerías utilizadas

Se presentan a continuación las librerías que se usaron como base para la construcción de la herramienta que se propone esta tesis. La librería LeafletJS provee el manejo de mapas interactivos mientras que jsPlumb brinda la posibilidad de poder interconectar marcadores (que representarán contenidos multimedia). Estas librerías se explicarán con más detalles en la Secciones 2.2.1 y 2.2.2 respectivamente.

Por última, en la Sección 2.2.3 se menciona la herramienta XMLWriter que se utilizó para poder guardar el estado de las creaciones que va realizando el usuario de la herramienta.

### 2.2.1 Leaflet

Leaflet [LeafletJS] es una librería javascript open-source que ofrece una diversa cantidad de capas para hacer mapas web interactivos orientada a dispositivos móviles (aunque también permite su uso de forma desktop). Las capas son objetos que se muestran sobre el mapa, que constan de uno o más elementos independientes y que, sin embargo, se manipulan como una sola unidad. Representan normalmente conjuntos de objetos que se añaden sobre el mapa para designar una asociación común (por ejemplo, capa de tránsito de automóviles, capa del estado del tiempo, etc.).

Diseñada de manera simple, eficiente y hacienda foco en la usabilidad, esta librería ofrece una gran variedad de capas que permiten darle mayor interactividad a los mapas, incluyendo marcadores, pop-ups, superposiciones de imágenes y GeoJSON (formato para codificar variables geográficas).

Esta librería tiene soporte para ser utilizada en la mayoría de los navegadores Web, sean de escritorio o móviles, aprovechando HTML5 y CSS3 en los navegadores más modernos pero sin dejar de ser accesibles a otros navegadores. Leaflet provee aceleración de hardware para iOS, haciendo que el uso de la librería en el navegador se sienta como si se estuviese sobre una aplicación nativa.

El diseño modular de Leaflet permite reducir el tamaño de la librería para que sea aún más rápida. Con un peso de 35 KB, contiene todas las características necesarias para crear mapas interactivos online.

La capacidad de escalabilidad de la librería no es un dato menor. Pensaba para ser ampliada, Leaflet dispone de una gran cantidad de plugins que, por ejemplo permiten: dibujar formas complejas, superponer imágenes para georeferenciarlas, manipulación de capas, etc.

---

<sup>3</sup> **KML** (del acrónimo en inglés **Keyhole Markup Language**) es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones.

A continuación se presentan algunos de los métodos que ofrece esta librería para interactuar con mapas:

- Métodos de modificación del estado del mapa:
  - `setView(latlng, zoom)`: Ubica el mapa dada una latitud y longitud y un zoom.
  - `setZoom(zoom)`: Setea el zoom del mapa.
  - `Locate()`: Intenta ubicar la posición del usuario usando la API de Geolocalización, disparando el evento *locationfound* y, opcionalmente, ubica el mapa en la posición del usuario.
- Métodos para obtener el estado del mapa:
  - `getCenter()`: Retorna el centro geográfico de la vista del mapa.
  - `getZoom()`: Retorna el zoom actual del mapa.
  - `getBounds()`: Retorna los límites de la vista actual del mapa.
  - `getSize()`: Retorna el tamaño del contenedor del mapa.
- Métodos para capas:
  - `addLayer(layer)`: Agrega una capa dada al mapa.
  - `removeLayer(layer)`: Elimina la capa dada del mapa.

Más información detallada sobre la librería puede encontrarse en la documentación de ésta en la página oficial de LeafletJS [LeafletJS].

Como se menciona anteriormente, LeafletJS permite ampliar sus capacidad a través de los plugins. LeafletAffineImageOverlay [LeafletAffineImageOverlay] es un plugin para Leaflet que permite agregar una imagen sobre un mapa y modificarla por medio de tres puntos.

En la Figura 13 puede verse un ejemplo de una imagen azul con la letra “B” superpuesta sobre el mapa y los tres redimensionadores que permiten modificar su tamaño y posición.

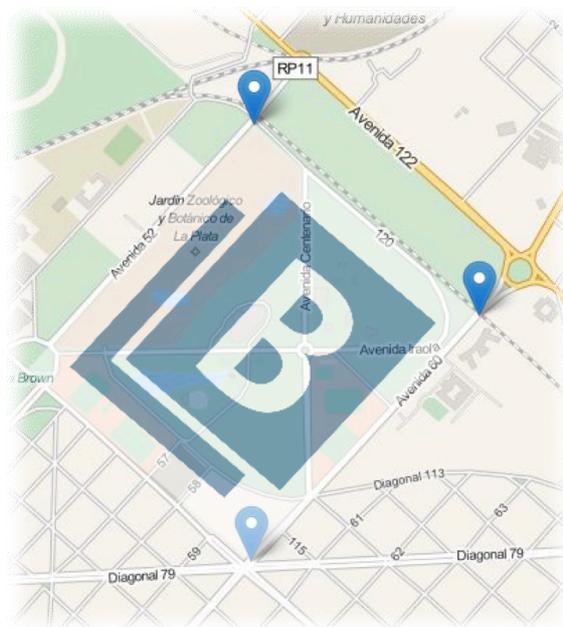


Figura 13. Ejemplo de imagen que se agrega sobre un mapa utilizando el plugin LeafletImageOverlay.

## 2.2.2 jsPlumb

jsPlumb [jsPlumb] es una librería javascript con licencia MIT que permite crear esquemas, diagramas o grafos de forma dinámica. jsPlumb provee un medio para conectar visualmente elementos del DOM<sup>4</sup> en una web.

Una conexión se compone de dos Endpoints, un Connector y cero o más Overlays que trabajan en conjunto para conectar dos elementos. Cada Endpoint tiene un Anchor asociado. Las conexiones se realizan, generalmente, a través de la interfaz gráfica. Manteniendo presionado el cursor sobre un fin de conexión y arrastrándose hacia un elemento, se genera la conexión.

En la Figura 14 se ve un ejemplo de un diagrama de una máquina de estado realizado utilizando esta librería. Se pueden observar dos conexiones (con\_5 y con\_11).

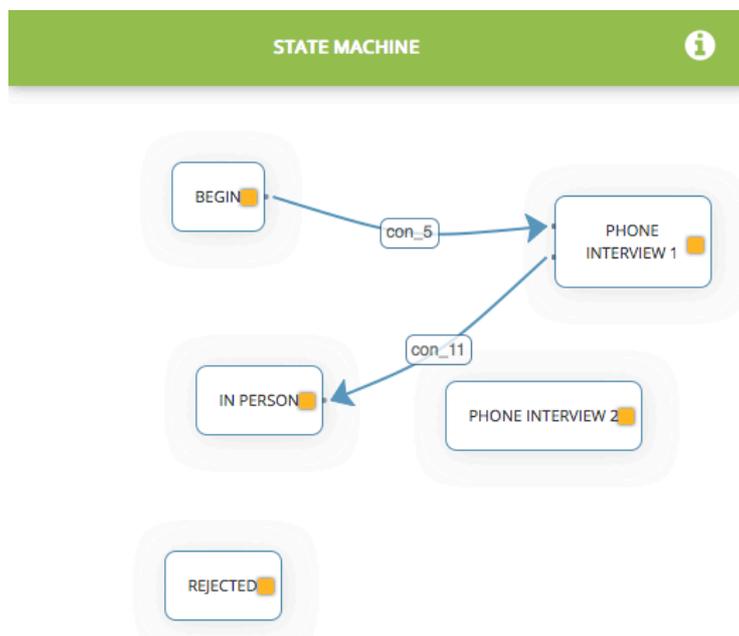


Figura 14. Ejemplo de una máquina de estados tomado del sitio oficial de jsPlumb [jsPlumb].

jsPlumb permite conectar cosas entre sí, siendo su elemento principal la **conexión**.

Una conexión se divide en cuatro conceptos:

- Anchor: un anchor representa un área que indica dónde, en un elemento, un conector puede conectarse (valga la redundancia). No tiene una representación visual sino que son posiciones lógicas.

<sup>4</sup> El **Document Object Model** o **DOM** ('Modelo de Objetos del Documento') es una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

- Fin de conexión (Endpoint): modela la apariencia y comportamiento de un fin de una conexión. jsPlumb viene con 4 tipo de fines de conexión: punto, rectángulo, espacio en blanco e imagen.
- Conector (Connector): es la representación visual de la línea que conecta dos elementos.
- Superposición (Overlay): es un componente de UI utilizado para decorar un conector, como ser un Label, Arrow, etc.

Puede observarse en la Figura 15 un ejemplo de conexión que se realiza sobre el diagrama de máquina de estado de la Figura 11. Se conecta el elemento 'Phone Interview 1' con el elemento 'Rejected'. La conexión queda identificada a través de una flecha azul que representa al conector (Connector). Los cuadrados amarillos dentro de cada elemento son los fines de conexión (Endpoints) y el label (rectángulo de puntas redondeadas) que tiene dentro el texto 'con\_24' es una superposición (Overlay).



Figura 15. Conexión de ejemplo realizada con la librería jsPlumb.

La Figura 16 muestra cómo queda modificada la máquina de estados de la Figura 11 con la inserción de la nueva conexión realizada en la Figura 13.

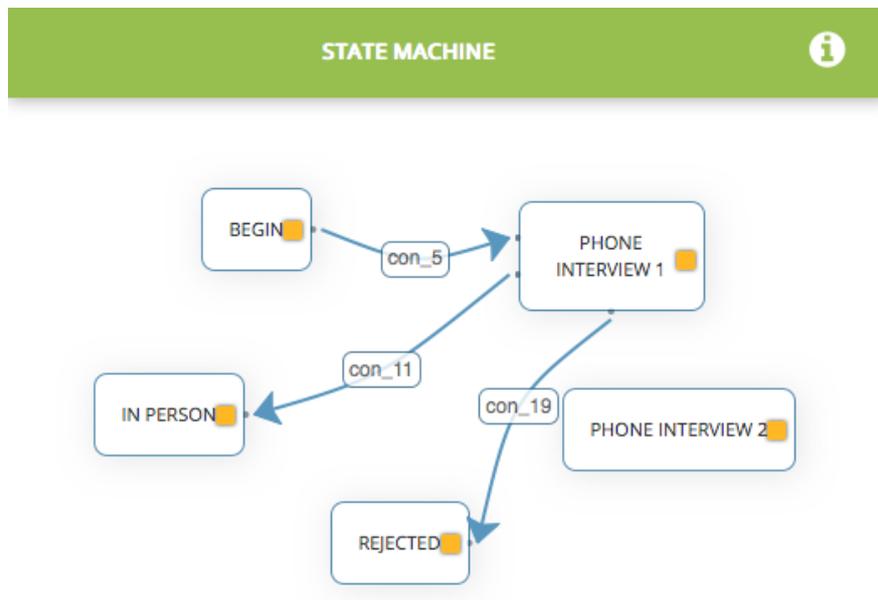


Figura 16. Se agrega una conexión a la máquina de estados de la Figura 14.

Las conexiones pueden realizarse también de forma programática muy fácilmente incluyendo un script de javascript e invocando al método `connect()`. Se muestra a continuación un ejemplo que conecta dos elementos denominados “element1” y “element2”:

```
jsPlumb.connect({source:"element1", target:"element2"});
```

jsPlumb soporta la personalización de los diferentes aspectos que la librería permite crear. Por ejemplo, la ubicación del anchor, el tipo de conector, el estilo de la flecha (color y ancho) o el tipo de fin de conexión (punto, rectángulo, espacio en blanco o imagen), entre otros. Esto puede ser realizado por el usuario de manera programática invocando a la función ‘importDefaults()’ y enviando los parámetros a modificar. En la Figura 17 se puede ver un ejemplo de personalización, donde se establece el color y el ancho de la flecha así como también que los fines de conexión sean puntos de radio 7.

```
jsPlumb.importDefaults({
  PaintStyle : {
    lineWidth:13,
    strokeStyle: 'rgba(200,0,0,0.5)'
  },
  Endpoints : [ [ "Dot", { radius:7 } ], [ "Dot", { radius:11 } ]
});
```

**Figura 17. Personalización en jsPlumb.**

### 2.2.3 XMLWriter para Javascript

XMLWriter [XMLWriter] es una librería que provee una manera rápida y directa de generar streams o archivos conteniendo datos XML conforme al W3C Extensive Markup Language (XML) 1.0 [XML].

Los métodos más utilizados:

- `writeStartDocument()`: Abre un nuevo documento.
- `writeEndDocument()`: Cierra el documento actual.
- `writeDocType(declarations)`: Agrega el doctype al documento.
- `writeStartElement(name)`: Crea un nuevo nodo elemento con un nombre dado y pasa a ser el elemento activo.
- `writeEndElement()`: Cierra el elemento activo y sube un nivel.
- `writeAttributeString(attr,value)`: Agrega un atributo al elemento activo.
- `getDocument()`: Genera un documento XML real.

En el caso de querer guardar la información de la Figura 17, uno podría usar esta librería, quedando representada la información como se visualiza en la Figura 18.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <Relations>
3   <Relation>
4     <ConnectionId>connector_1</ConnectionId>
5     <SourceId>element1</SourceId>
6     <TargetId>element2</TargetId>
7   </Relation>
8 </Relations>
```

Figura 18. Ejemplo de la conexión de la Figura 3 exportada en XML utilizando XMLWriter.

### 3. DEFINICIÓN CONCEPTUAL DE UNA HERRAMIENTA DE CREACIÓN IN-SITU DE APLICACIONES MÓVILES INDOOR-OUTDOOR

En este capítulo se presentarán las características principales que detectamos que debería tener una herramienta de creación in-situ para lograr crear aplicaciones móviles tanto para espacios indoor como outdoor. Como se busca que esta herramienta sea usada por usuarios no expertos, es fundamental que la interfaz de creación sea usable. Los usuarios interactuarán con esta interfaz para lograr la creación de sus aplicaciones.

En la Figura 19 presentamos de manera gráfica los componentes que debería tener una herramienta para la creación in-situ de aplicaciones móviles indoor-outdoor.



Figura 19. Componentes constitutivos de la herramienta.

Como se puede observar en la Figura 19, como primer y principal componente se puede ver la 'interfaz para la creación de elementos posicionados'. Esta interfaz sería la representación visual de los 'conceptos abstractos' y aquella que permitiría que el usuario pueda crear y/o manejar estos conceptos (que son explicados en detalle en la Sección 3.2).

Los componentes conectados con la interfaz visual tienen como objetivo brindar diferentes servicios o facilidades:

- El 'servicio de mapas' es aquel que permite representar el espacio en relación a mapas satelitales de manera simple y sencilla a través del uso de diferentes APIs como ser

GoogleMaps<sup>5</sup> u OpenStreetMap<sup>6</sup>. Por ejemplo, de esta manera el usuario podría definir posiciones en un mapa sin necesidad de saber cuál es la latitud y longitud del lugar donde quiera definir dicha posición.

- El componente de '*mecanismo de sensado*' se refiere a aquellos mecanismos que la interfaz puede ofrecer para soportar de manera transparente para el usuario el posicionamiento de elementos al momento de su creación. Por ejemplo, cuando uno usuario quiera crear un elemento posicionado dentro de un mapa satelital, el GPS del dispositivo móvil con el cual se esté trabajando, podría tomar automáticamente los datos y posicionar dicho elemento en la posición actual del usuario.

Se puede apreciar en la Figura 19 que la herramienta puede ser extendida con otros componentes que brinden servicios para la creación, por ejemplo, que permita el acceso a un repositorio de imágenes posicionadas.

### 3.1 Conceptos abstractos usados como base de la Herramienta

En esta sección presentamos un listado de los conceptos abstractos que la herramienta toma como base para su desarrollo. La herramienta que se quiere desarrollar contempla al menos, dos aspectos relevantes: por un lado el contenido brindado por la aplicación y por el otro, el posicionamiento de los contenidos dentro de un espacio. Se presenta entonces un enfoque de separación en capas de los aspectos de contenido y posicionamiento (espacio físico y posiciones). Esta idea de separación en capas es tomada de ([Literas et. al, 2012] y [Literales et. al, 2013]) y permite mejorar la reusabilidad y simplificar las problemáticas que vienen relacionadas con la evolución de cada aspecto antes mencionado.

En esta tesis, se tendrá en cuenta esta separación para aprovechar las ventajas mencionadas. Además, se cuenta con una relación entre estos aspectos para lograr así la composición entre ambos. Es decir, si bien, ambas capas se relacionan, no hay acoplamiento. En particular, la relación se da entre contenido y posiciones en una capa aparte.

La Figura 20 muestra esta separación en capas. Por un lado, los marcadores azules que representan a los contenidos y, por el otro, los marcadores rojos que representan a las posiciones dentro de un espacio. A la composición entre un contenido y una posición la llamaremos **elemento posicionado**.

---

<sup>5</sup> <https://developers.google.com/maps> (Accedido: 8/2/2015)

<sup>6</sup> <http://www.openstreetmap.org> (Accedido: 8/2/2015)

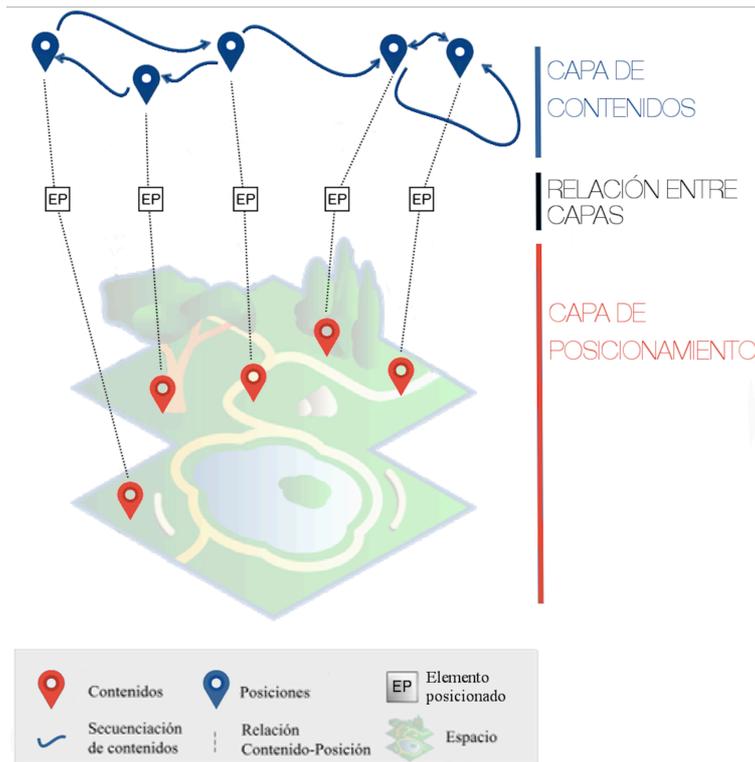


Figura 20. Separación en capas.

A continuación, se explican en detalle tanto la capa de contenidos como la de posicionamiento así como también se explica la relación entre ambas capas.

### 3.1.1 Capa de contenidos

Se define en ([Literas et. al, 2012] y [Literales et. al, 2013]) como contenido a algún tipo de elemento que puede estar o no relacionado a su vez con otro contenido, formando de esta manera una estructura.

En ([Literas et. al, 2012] y [Literales et. al, 2013]) se utilizan diferentes estructuras que pueden resultar de las conexiones entre los contenidos. Por un lado, una estructura secuencial que podría presentarse de manera lineal o con bifurcación (en forma de grafo) y por otro lado, una estructura de conjunto.

En la Figura 21, se puede visualizar gráficamente cada estructura identificada (estructura secuencial lineal, secuencial con bifurcación y de conjunto), definidas considerando las dos capas mencionadas (contenido y movilidad).

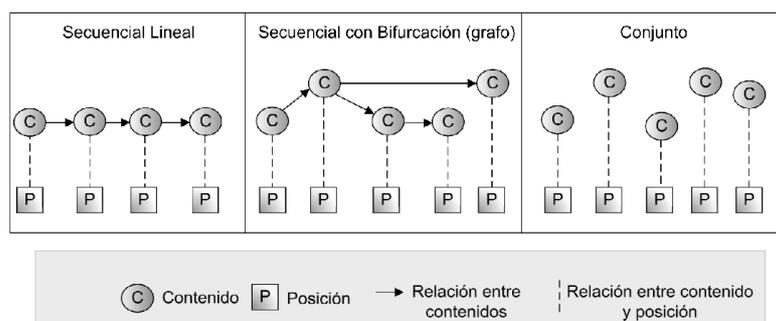


Figura 21. Conceptualización definida en capas [Literas et. al, 2012].

A continuación se explica de manera breve qué significa cada tipo de estructura:

- **Estructura secuencial lineal:** Para formar una estructura de este tipo, los contenidos deben relacionarse considerando un orden consecutivo para los mismos (un contenido seguido de otro).
- **Estructura secuencial con bifurcación (grafo):** En este caso, los contenidos deberán conectarse considerando la posibilidad de bifurcaciones entre los mismos, formando una estructura de tipo grafo.
- **Estructura de conjunto:** En este tipo de estructura los contenidos no tienen una relación explícita entre sí. No existe un orden preestablecido.

### 3.1.2 Capa de posicionamiento

Siguiendo los conceptos definidos en ([Literas et. al, 2012] y [Literales et. al, 2013]), la capa de posicionamiento está representada por posiciones asociadas a un espacio físico. El espacio físico podría ser cualquier espacio al que, de alguna forma, se le puedan asociar posiciones que representen una ubicación dentro del mismo y en el cual un usuario podría movilizarse.

Las posiciones no tienen relación entre sí y serían posibles lugares donde se brindará contenido.

Estos aspectos (las posiciones y el espacio físico) forman parte de la capa de posicionamiento.

Cuando se habla de una posición, no sólo se hace referencia a una coordenada satelital (latitud y longitud) sino a cualquier forma que permita ubicar a alguien o algo dentro de un espacio. Dicho espacio se determina en relación la orientación respecto a algo [Leonhardt, 1998]. Por ejemplo, la posición podría tratarse de coordenadas cartesianas dentro de un plano.

### 3.1.3 Relación entre las capas de contenido y posicionamiento

Como ya se mencionó anteriormente, a la relación entre contenido y posición se la denomina elemento posicionado. El hecho de poder tener desacopladas ambas capas permite que la herramienta tenga una mayor flexibilidad así como también da mayor libertad al usuario permitiéndole redefinir y reutilizar capas ya creadas. Por ejemplo, se podría definir una capa de posiciones y utilizarla con diferentes capas de contenidos para crear más de una aplicación móvil.

Supongamos el caso en el que se quiera crear una aplicación móvil para que los usuarios de esta aplicación puedan realizar un recorrido a través del interior de un museo y luego por los alrededores del mismo (exterior del museo). Si la conceptualización de los componentes no hubiese sido de esta forma general y no se contase con el desacople de las capas, no habría posibilidad alguna de reutilizar los diferentes componentes que constituyen a la aplicación en otro momento. Si, por el contrario, se hubiese construido esta aplicación con alguna herramienta que permita mantener las diferentes que capas desacopladas, se podría dar el caso hipotético que más adelante, la persona que creó la aplicación quiera reutilizar los contenidos en otro espacio geográfico, por ejemplo, en otro museo diferente y en los alrededores de este otro museo. O también, podría darse el caso de que quiera conservar el espacio geográfico original pero esta vez, modificar los contenidos.

### 3.2 Interfaz para la construcción de elementos posicionados

Habiendo conceptualizado los aspectos más importantes que son base para la interfaz, en esta sección se especifica qué sería necesario para que los conceptos abstractos mencionados en la Sección 3.1 puedan representarse de manera visual a través de una interfaz que permita a usuarios no expertos crear aplicaciones móviles basadas en posicionamiento tanto en ambientes outdoor como indoor. Si bien como trabajo a futuro se trabajará sobre la derivación de la aplicación móvil, esta tesis hará foco en la especificación de los contenidos, llevando a éstos a un formato exportable.

Siguiendo el enfoque de separación en capas mencionado anteriormente, podemos decir que esencialmente la interfaz deberá proveer algún mecanismo sencillo y amigable con el usuario para la creación tanto de la capa de contenidos como de la de posicionamiento; para la primera, se espera tener alguna especie de barra de herramientas donde el usuario pueda crear un contenido. De la misma forma, se podrá agregar a este contenido algún tipo de elemento audiovisual o cualquier otro que sea de interés para el usuario. En cuanto a la capa de posicionamiento, la interfaz deberá proveer alguna forma de poder seleccionar un tipo de espacio físico (outdoor o indoor) y a su vez poder definir posiciones en relación al espacio elegido.

Por otra parte, además de estas opciones de creación, se espera que la interfaz provea algún mecanismo transparente al usuario para la creación combinada de ambas capas (creación de elementos posicionados), es decir que, mientras creo un contenido ya lo estoy posicionando en un espacio físico previamente elegido. Por ejemplo, trabajando con un mapa outdoor y un mecanismo de sensado de tipo GPS, el usuario de la herramienta podría ir creando contenidos que automáticamente se posicionan en la posición actual del usuario.

En resumen, la interfaz debería poder crear:

- Capas de contenidos
- Capas de posicionamiento
- Capas de contenidos y posicionamiento simultáneamente

La interfaz debería permitir reusar capas ya definidas. Para poder realizar esto, debe ofrecer algún mecanismo de exportar e importación de capas. Por ejemplo, se podrían tener opciones para:

- Importar una capa de contenidos en una capa de posicionamiento existente
- Importar una capa de posicionamiento en una capa de contenidos existente

También, como se espera que la herramienta permita la derivación en una aplicación móvil, se espera que ésta pueda transformar las construcciones que el usuario de la herramienta realice a un formato exportable, por ejemplo, en XML.

#### 4. HERRAMIENTA PROTOTÍPICA

Tomando como base lo mencionado en el Capítulo 3, se desarrolló una herramienta prototípica que permite la crear los conceptos mencionados, es decir, contenidos, posiciones o elementos posicionados. A continuación se explican las características de dicha herramienta, su funcionamiento y sus modos de uso.

Teniendo en cuenta la separación en capas mencionada en el Capítulo 3 (capa de posicionamiento y capa de contenidos), al iniciar la herramienta se presenta al usuario el menú principal (ver Figura 22) con las seis posibles vías de acción:

1. Crear sólo capa de contenidos
2. Crear sólo capa de posicionamiento
3. Crear elementos posicionados
4. Importar contenidos y asignar posiciones
5. Importar posiciones y asignar contenidos
6. Importar contenidos y posiciones por separado y asociarlos



**Figura 22. Menú principal de la herramienta.**

A continuación se describirá cada vía de acción, cómo es su funcionamiento y se definirán algunos conceptos particulares de la herramienta.

##### 4.1 Crear contenidos

La herramienta permite crear una capa de contenidos sin necesidad de que cada uno de los contenidos se asocie inmediatamente a una posición ni a un espacio físico. Es decir, el contenido se crea independiente del lugar donde en un futuro puede ser utilizado. Esta opción facilita el reuso de este contenido, ya que el mismo posteriormente puede ser asociado a diferentes espacios físicos.

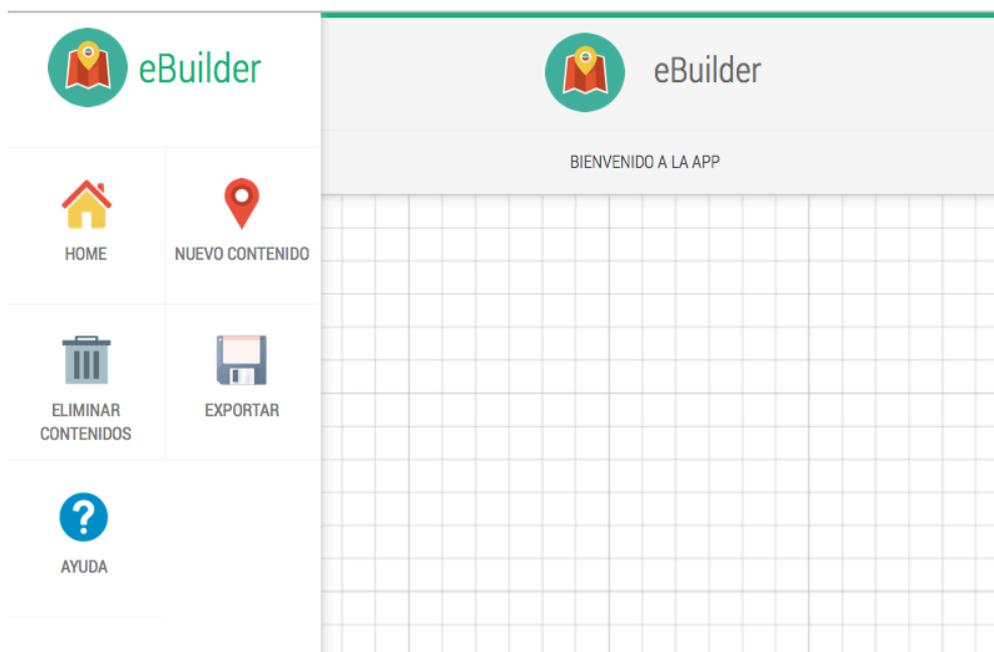
Al elegir la opción “CREAR CONTENIDOS” dentro del menú principal (como se mostró en la Figura 22), se abrirá una nueva pantalla con una grilla de fondo donde se ubicarán los contenidos a medida que se vayan creando.

El botón de menú (ver Figura 23) se ubica siempre en la esquina superior izquierda de las pantallas definidas por la herramienta. Está representado por una circunferencia verde con tres barras horizontales blancas en su interior. Acorde a la pantalla actual que el usuario este visualizando serán las opciones que se muestren cuando este botón sea seleccionado.



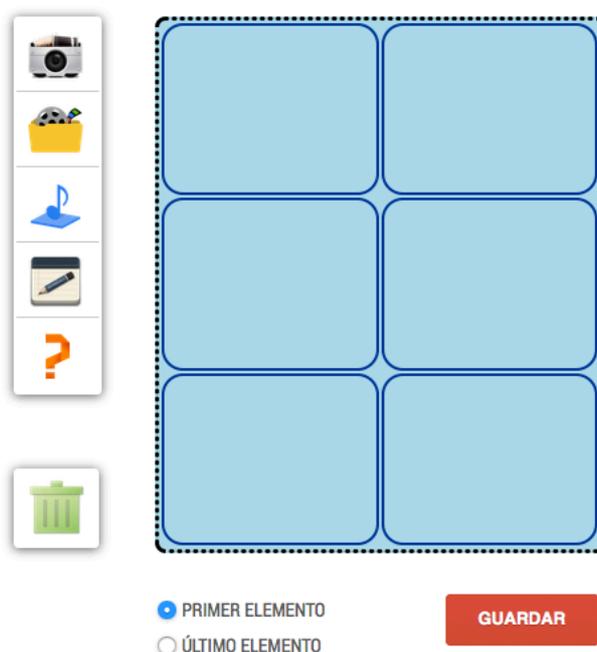
**Figura 23. Botón de menú.**

Para crear un contenido se debe apretar el botón de menú (Figura 23), lo que desplegará el menú con opciones a realizar (ver Figura 24). Se puede apreciar que el usuario tiene la posibilidad de volver al home (regresando a la pantalla mostrada en la Figura 22), puede crear contenidos, borrarlos, exportarlos o consultar la ayuda. En la Figura 24 se puede apreciar la grilla dónde se irán creando los contenidos.



**Figura 24. Menú de opciones desplegado.**

Haciendo click en la opción “NUEVO CONTENIDO” (de la Figura 24), un box de creación aparecerá que permitirá personalizar el nuevo contenido (ver Figura 25). Para la herramienta, se define como contenido a una imagen, video, audio, texto, pregunta o una combinación de éstos. En todo momento el usuario tiene la posibilidad de ir borrando lo que fue creando, para esto usa el ícono con forma de cesto de basura ubicado abajo a la izquierda.



**Figura 25. Box de creación.**

Se podrán añadir hasta un máximo de seis elementos audiovisuales por contenido. Para agregar uno de ellos se debe arrastrar cualquiera de estos hasta alguno de los cuadrantes azules que aparecen en la Figura 25.

Además, en caso que el usuario desee relacionar los contenidos con una estructura de tipo secuencia, deberá establecer cuál es el primer elemento y cuál es el último. Para hacer esto, notar que a la izquierda del botón “GUARDAR”, el usuario puede tildar la opción “PRIMER ELEMENTO” o “ÚLTIMO ELEMENTO” según corresponda<sup>7</sup>.

La herramienta (por el momento) permite crear preguntas de opción múltiple. Al arrastrar el signo de pregunta (de la Figura 25) hacia un cuadrante, aparece en pantalla un box de creación de pregunta (ver Figura 26). Las preguntas poseen un título (que hace referencia a la pregunta en sí) y las posibles respuestas. Las respuestas se marcan como correctas o incorrectas apretando sobre el *tilde*. Cuando el *tilde* está en color gris, significa que estamos creando una respuesta incorrecta y cuando se colorea en verde significa que la respuesta es correcta. El ícono verde que representa el signo + se utiliza para agregar más respuestas a la pregunta que se está creando. Una vez que se terminó de personalizar el nuevo contenido basta con apretar el botón “GUARDAR” para que el nuevo contenido sea creado y se añada a la grilla.

---

<sup>7</sup> Dado que la herramienta permite crear relaciones formando estructuras de grafo (ver Capítulo 3), surge la necesidad de establecer un orden, especificando cuál será el primer elemento y cuál el último. Esto sirve en los casos donde hay más de un elemento inicial o final en el grafo.



iconografías utilizadas por la herramienta para cada tipo de contenido que se crea. En caso de que se tenga sólo texto, el ícono del contenido mostrará una T (haciendo alusión a la palabra ‘texto’), en caso que sea sólo pregunta, mostrará una P y así sucesivamente. Si el contenido es una combinación de diferentes elementos audiovisuales, el ícono mostrará una C haciendo alusión a la palabra ‘combinado’.

IMAGEN	AUDIO	TEXTO	VIDEO	PREGUNTA	COMBINADO

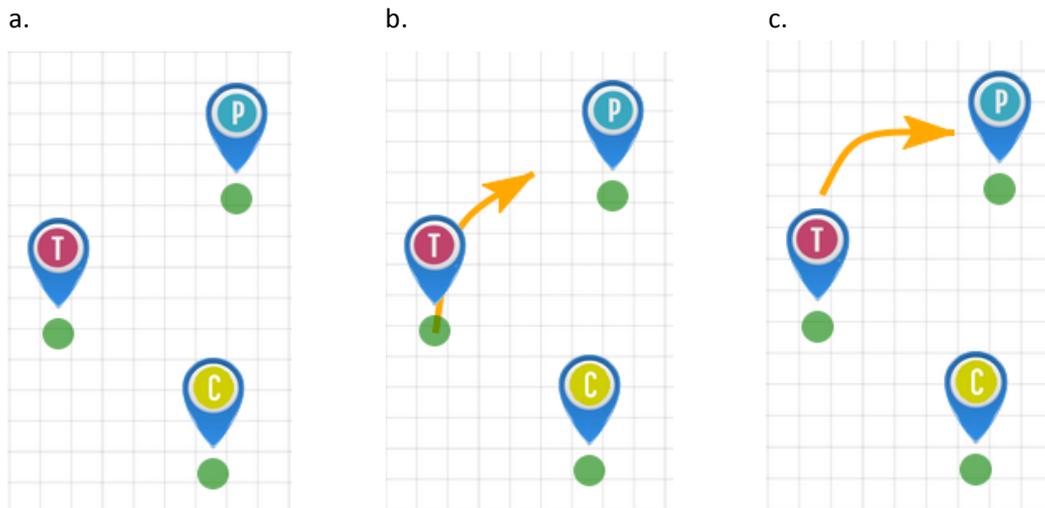
**Figura 28. Tabla de referencias ícono – tipo de contenido.**

El mismo proceso de creación se repite tantas veces como contenidos desea tener el usuario de la herramienta, formando así la capa de contenido. En la Figura 29 se puede apreciar 3 contenidos creados, uno de tipo texto, uno de tipo pregunta y uno de tipo combinado. Notar que no hay ningún espacio físico definido.



**Figura 29. Ejemplo de capa de contenidos creada.**

Como fuera mencionado anteriormente, el usuario puede conectar contenidos entre sí formando alguna especie de estructura (lineal, grafo o conjunto). Para realizar una conexión entre dos contenidos (ver Figura 30.a), basta con apretar el círculo verde debajo de cada contenido que se crea y arrastrar el cursor o el dedo (en caso de estar en un dispositivo móvil) haciendo el contenido destino. De esta forma se irá dibujando una flecha (ver Figura 30.b). Cuando se llega al contenido destino, la flecha queda asentada entre los dos contenidos estableciendo una conexión (ver Figura 30.c). Para eliminar una relación debe presionarse sobre la flecha y ésta será removida.

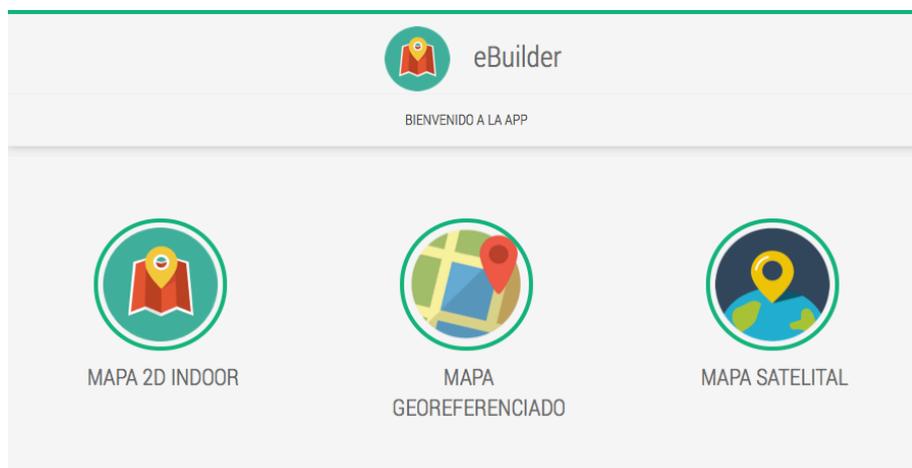


**Figuras 30.** En la Figura 30.a se ve la capa de contenidos si relaciones. La Figura 30.b muestra cómo el usuario empezó a crear una flecha y está moviéndose hacia el contenido con la letra T. Finalmente, en la Figura 30.c se muestra cómo queda establecida la conexión entre los dos contenidos.

#### 4.2 Crear posiciones

La herramienta permite crear una capa de posicionamiento sin necesidad de que cada uno de las posiciones se asocie inmediatamente a un contenido. Esta capa debe tener un tipo de espacio físico asociado. En este prototipo se da soporte para mapas indoor, mapas satelitales y mapas georeferenciados (mapa satelital con el agregado de un mapa indoor en él). Definir las posiciones de manera independiente al contenido permite al usuario identificar en que posiciones se va a brindar algo sin importar qué, luego esta misma capa se puede usar para combinar con diferentes tipos de contenidos. De esta manera, se logra el reusó de esta capa.

Al elegir la opción “CREAR POSICIONES” dentro del menú principal (el cual se mostró en la Figura 22), se abrirá una nueva pantalla que permitirá elegir al usuario qué tipo de espacio físico desea utilizar como se puede apreciar en Figura 31. Hasta el momento la herramienta permite seleccionar tres tipos de mapas, un mapa 2D indoor, un mapa georeferenciado o un mapa satelital. Más adelante se explica cada una de estas opciones.



**Figura 31.** Pantalla de selección de tipo de mapa.

En caso de querer utilizar mapas indoor, se debe seleccionar la opción “MAPA 2D INDOOR” (de la Figura 31). Haciendo esto, aparece un box que permite que el usuario cargue una imagen que represente al mapa que quiere utilizar (ver Figura 32). La imagen que se puede cargar puede ser de cualquier formato (.jpg, .png, .gif, etc.) y no debe exceder los 10MB.

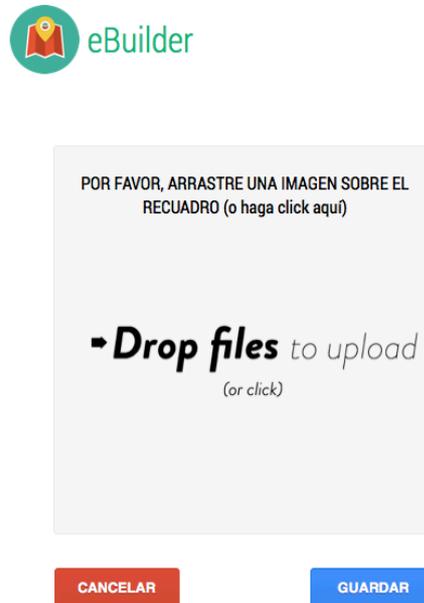


Figura 32. Box de carga de imagen de un mapa 2D.

Una vez cargada la imagen, se presiona el botón “GUARDAR” (Figura 32) para establecer dicha imagen como espacio físico y, de esta forma, se muestra en pantalla la imagen<sup>8</sup> (ver Figura 33). Notar que las imágenes cargadas de esta manera deben estar previamente definidas, lo único que hace la herramienta es tomarlas como espacio físico.

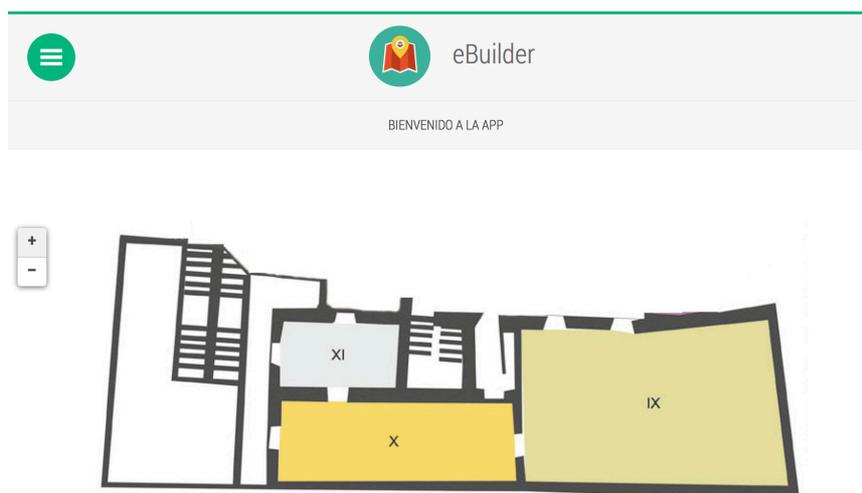
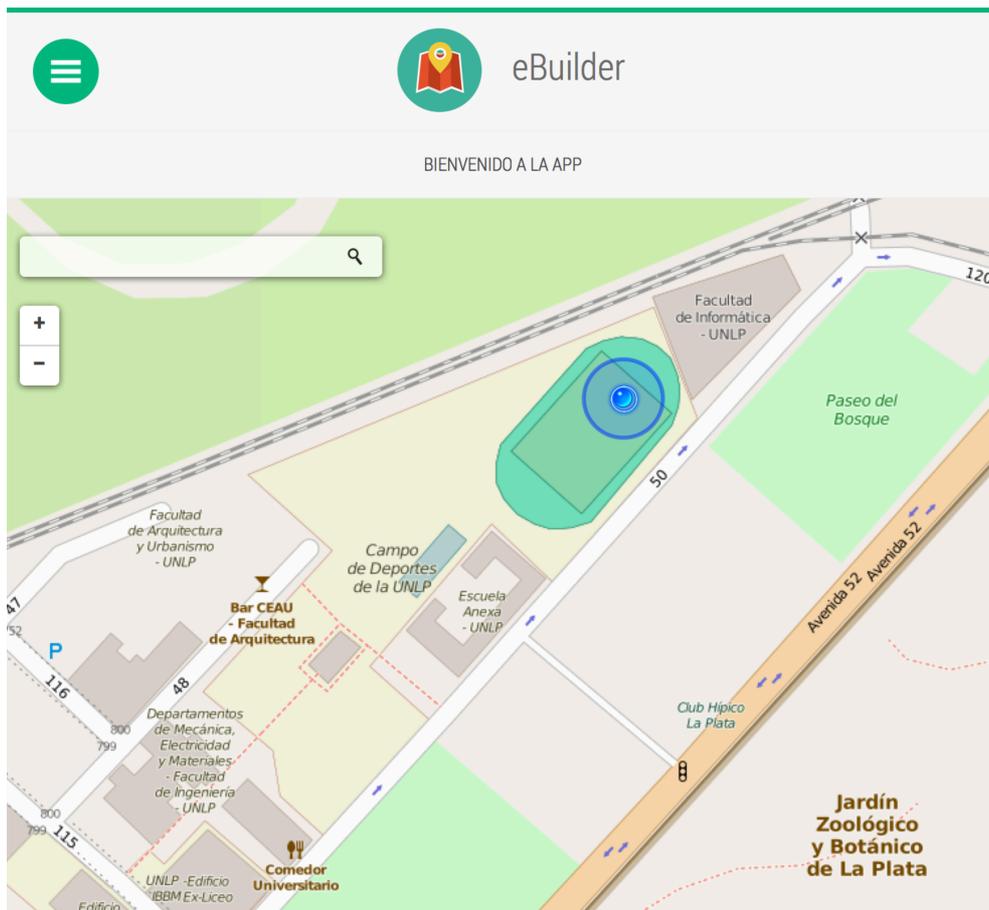


Figura 33. Mapa 2D insertado.

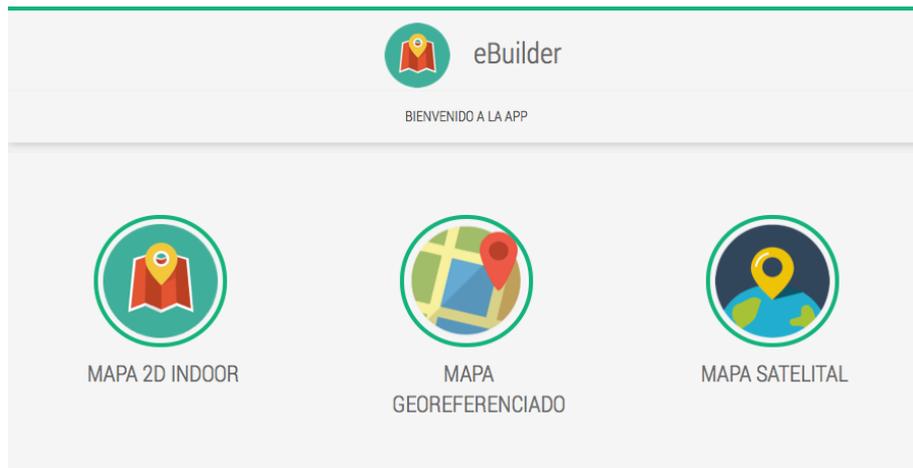
<sup>8</sup> Imagen extraída de la siguiente URL: <https://megamiami.files.wordpress.com/2014/01/sawgrass-mill-map.jpg> (Accedido: 8/2/2015)

Para utilizar mapas satelitales basta con seleccionar la opción “MAPA SATELITAL” (de la Figura 31) y aparecerán en pantalla los mapas de OpenStreetMaps (ver Figura 34). Hasta el momento la herramienta sólo implementa mapas de *OpenStreetMaps* en futuras evoluciones de la misma se podrían usar otro tipos de mapas como, por ejemplo, *GoogleMap*. Siempre que haya un mapa satelital y un mecanismo de sensado asociado a la herramienta (por ejemplo, GPS, 4G o WiFi), se podrá ver un punto azul en él que indicará la posición actual del usuario. Si el usuario de la herramienta se encontrase utilizando la herramienta sobre un dispositivo móvil el punto azul se irá moviendo a medida que el usuario recorra el espacio físico.



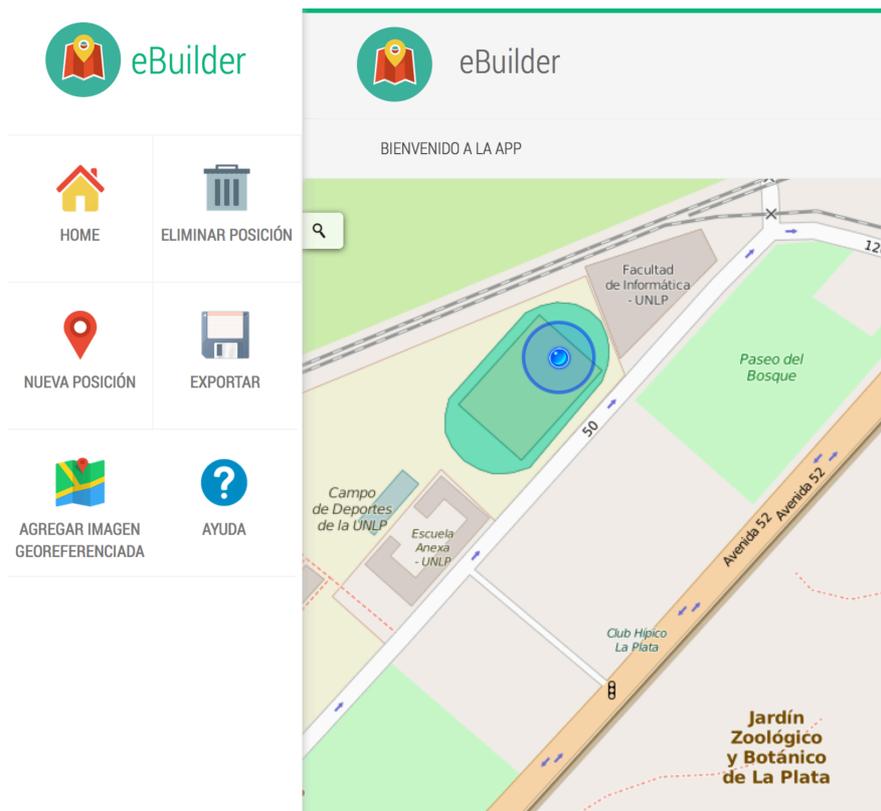
**Figura 34. Ejemplo utilizando mapas satelitales.**

Por último, la otra alternativa es utilizar mapas satelitales y superponer un mapa indoor a éstos. Para realizar esto, se debe elegir la opción “MAPA GEOREFERENCIADO” (ver Figura 35).



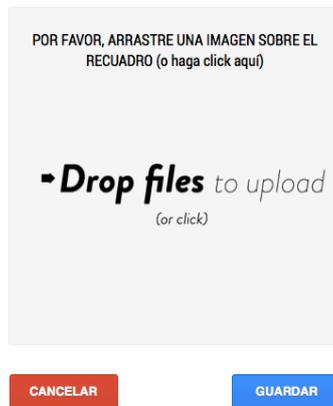
**Figura 35. Pantalla de selección de tipo de mapa.**

Una vez elegida esta opción se carga el mapa satelital. Para superponer un mapa indoor, se debe hacer click en el botón menú (ubicado en la esquina superior izquierda de la pantalla) y luego, dentro del menú que se despliega, hacer click en la opción "AGREGAR IMAGEN GEOREFERENCIADA" (ver Figura 36).



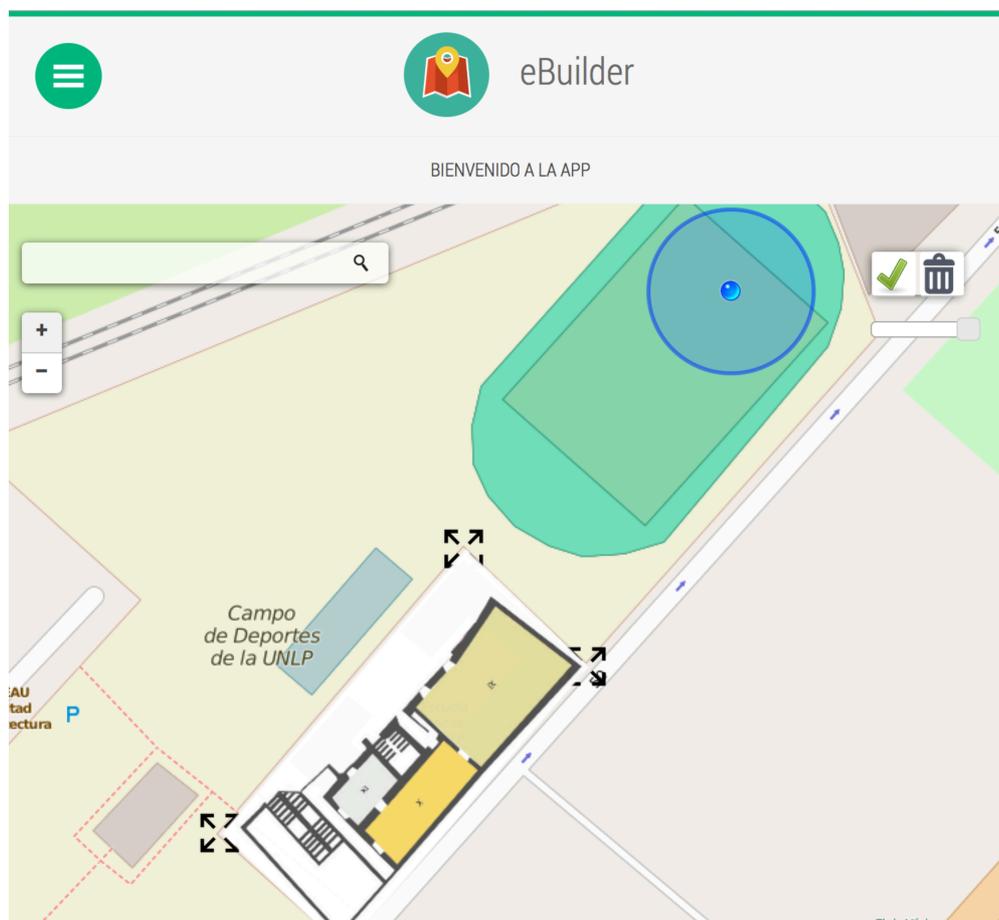
**Figura 36. Menú de opciones desplegado.**

Una vez hecho esto, aparece un box para cargar una imagen igual al box para realizar la carga de un mapa indoor (ver Figura 37).



**Figura 37. Box de carga de imagen de un mapa 2D.**

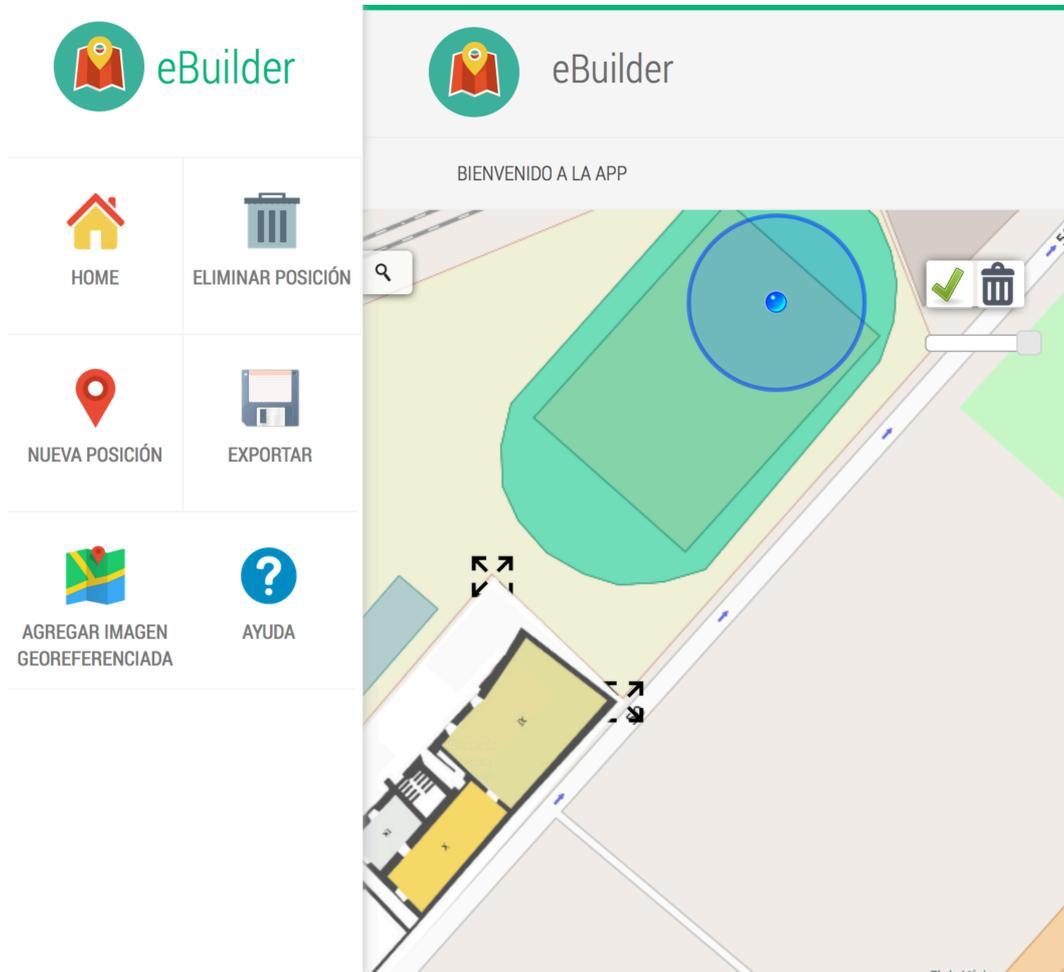
Una vez cargada la imagen, ésta aparece superpuesta sobre el mapa satelital. La herramienta permite mover la imagen, agrandarla o reducirla y además, a través de un toolbox que aparece en la esquina superior derecha de la pantalla y que permite ajustar su opacidad, eliminar la imagen georeferenciada y guardar la imagen. En la Figura 38 se puede ver una imagen de un mapa indoor superpuesta sobre un mapa satelital y el toolbox para personalizar la imagen.



**Figura 38. Ejemplo de mapa geofereenciado.**

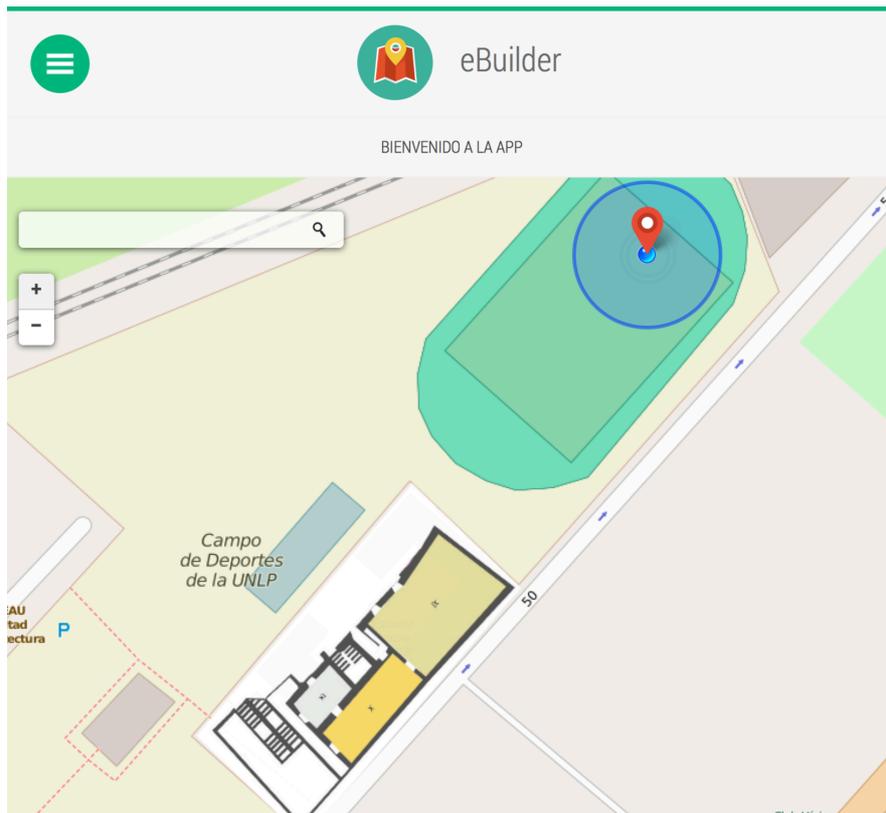
Una vez que el usuario terminó de definir qué tipo de mapa desea utilizar, puede comenzar a crear posiciones y ubicarlas en el mapa. El procedimiento para realizar esto es similar al de la creación de contenidos, mencionado en la Sección 4.1.

Sea cual sea el tipo de mapa que se eligió, para crear una posición se debe presionar sobre el botón de menú y luego apretar el botón “NUEVA POSICIÓN” (ver Figura 39).



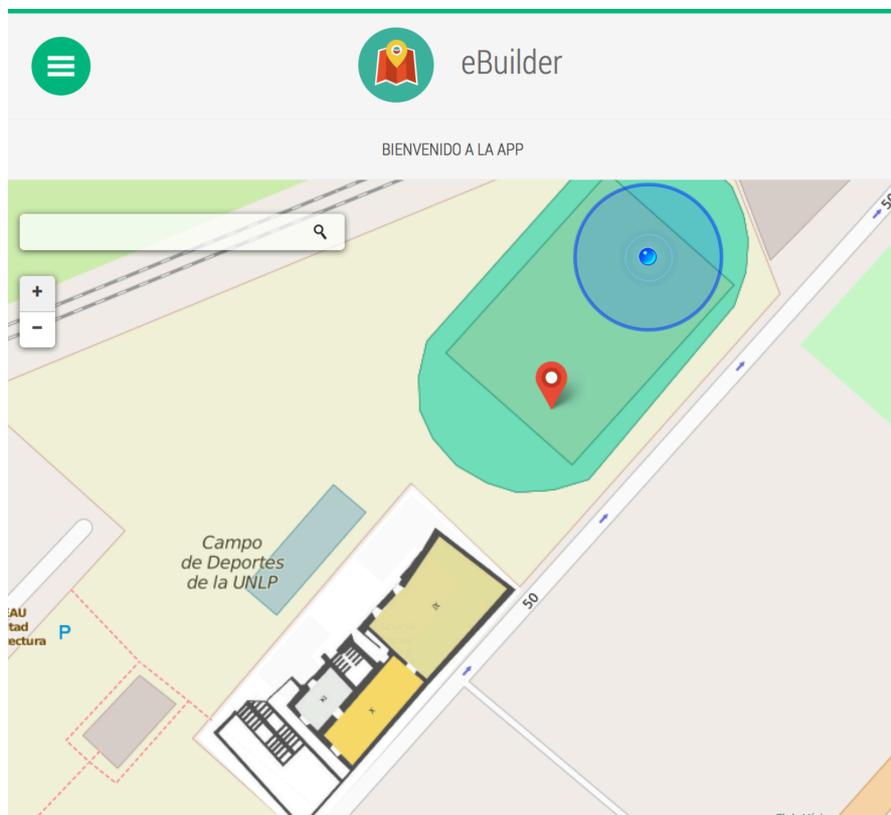
**Figura 39. Menú de opciones desplegado. Notar el botón “NUEVA POSICIÓN”.**

Al crear la nueva posición se sitúa automáticamente en la posición actual del usuario (ver Figura 40), tomando como mecanismo de sensado el GPS del dispositivo o el triangulado a partir de la señal Wi-Fi o de las redes de celulares. De no contar con ningún mecanismo, la nueva posición se sitúa en algún punto aleatorio del mapa. La característica que brinda la herramienta de poder ser utilizada in-situ, es justamente que la creación puede realizarse de una manera más práctica y automática, ya que a medida que el usuario se mueve sobre el espacio físico puede ir creando los elementos posicionados. Crear las posiciones o los elementos posicionados en el lugar donde luego se utilizará la aplicación permite tener una visión completa del espacio físico y teniendo completa seguridad acerca de la viabilidad de lo que se quiere generar.



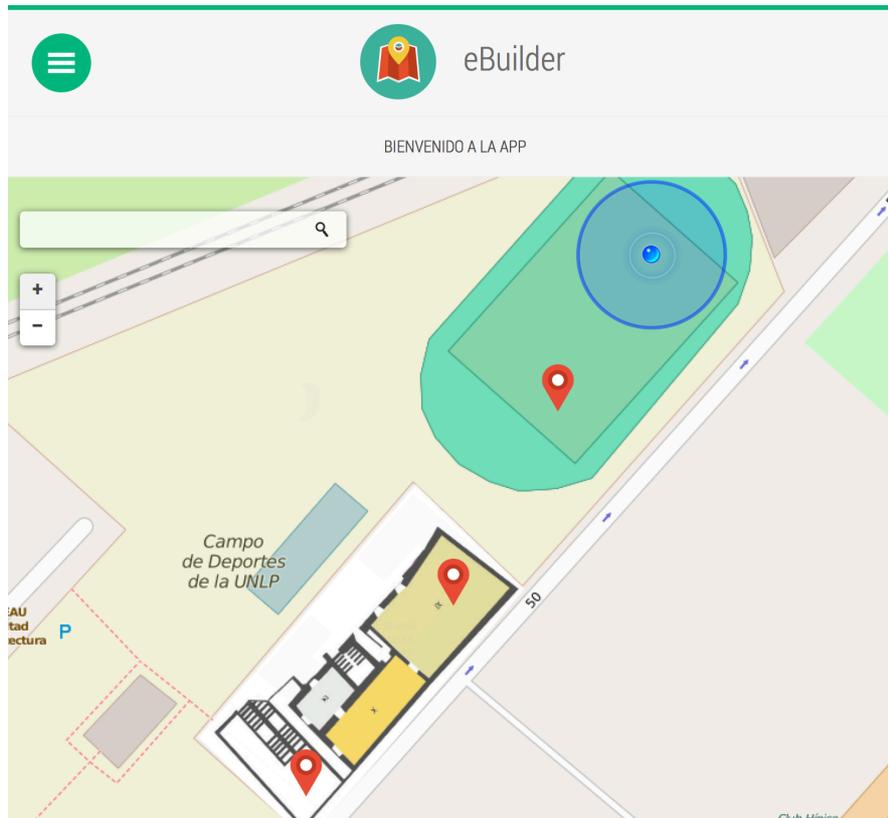
**Figura 40. Posición creada y situada en la posición actual del usuario.**

En ambos casos, el usuario puede arrastrar la posición manualmente hacia donde quiera (ver Figura 41).



**Figura 41. Posición movida manualmente por el usuario.**

En el ejemplo que se muestra en la siguiente figura (ver Figura 42) se pueden apreciar tres markers, dos ubicados dentro del mapa indoor y uno sobre el mapa satelital.



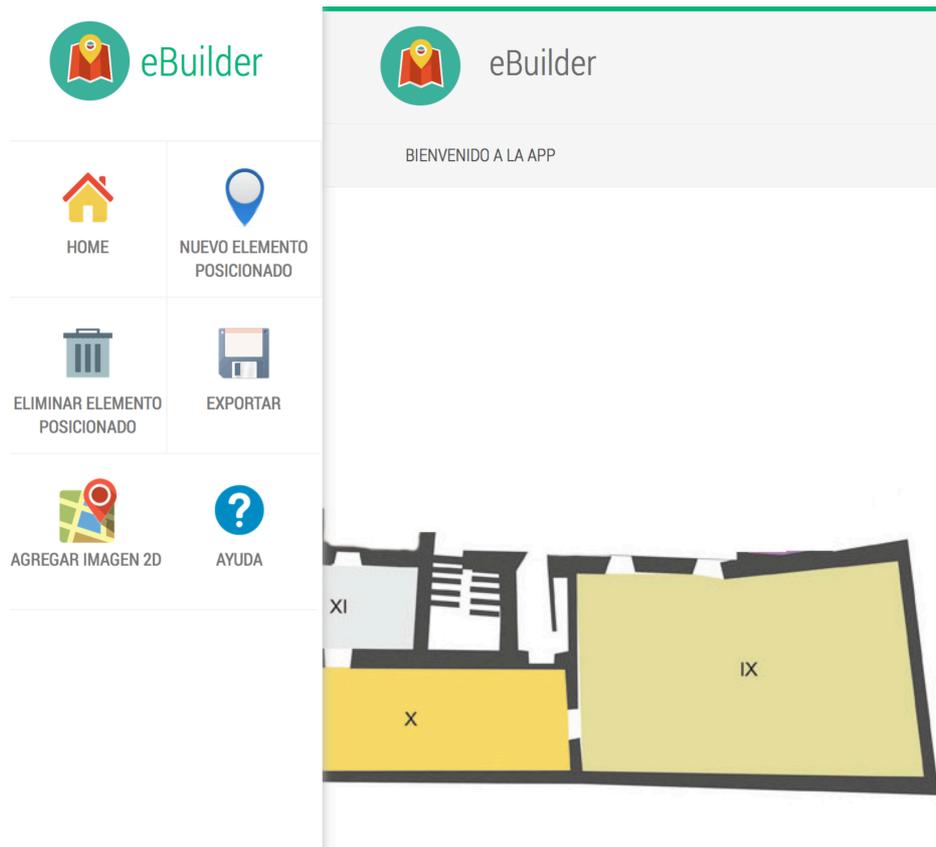
**Figura 42. Ejemplo de mapa geofereciado con markers posicionados en él.**

### **4.3 Crear elementos posicionados**

Definimos como elemento posicionado a aquellos contenidos que tienen asociada una posición dentro de espacio físico.

Para realizar la creación de un elemento posicionado, el usuario de la herramienta debe realizar los siguientes pasos:

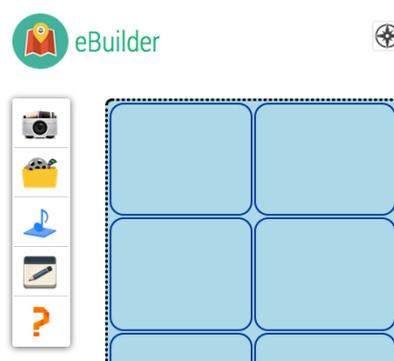
1. Seleccionar la opción "CREAR ELEMENTOS POSICIONADOS" del menú principal (como se mostró en la Figura 22).
2. Una vez realizado esto, al igual que durante la creación de posiciones se le pedirá al usuario que elija un tipo de mapa. El procedimiento para realizar esto es exactamente idéntico a lo descrito en la Sección 4.2.
3. Una vez elegido el mapa, el usuario puede comenzar a crear elementos posicionados así como también conectarlos entre sí siguiendo las mismas mecánicas que utilizó para la creación de contenidos en la Sección 4.1. Para la creación, en este caso el nombre del botón es "NUEVO ELEMENTO POSICIONADO" como se puede apreciar en la Figura 43.



**Figura 43. Menú de opciones desplegado mostrando el botón “NUEVO ELEMENTO POSICIONADO”.**

Siguiendo estos pasos, un usuario puede crear los elementos posicionados directamente sobre el espacio físico elegido.

Al estar creando contenidos posicionados (lo que llamamos elementos posicionados), el box de creación tendrá también la opción de establecer si se desea utilizar el GPS o códigos QR como mecanismo de sensado. Para hacer esto, en la esquina superior derecha del box de creación, aparece un ícono con una brújula (ver Figura 44 a). Esto indica que el mecanismo de sensado definido por defecto es el GPS. Si se quiere utilizar códigos QR, basta con apretar dicho botón y cambiará el ícono (ver Figura 44 b). Además, se abrirá el código QR definido para dicho elemento posicionado y el usuario tendrá la posibilidad de descargarlo a su dispositivo para eventualmente imprimirlo.



**Figura 44 a. Box de creación con GPS como mecanismo de sensado.**

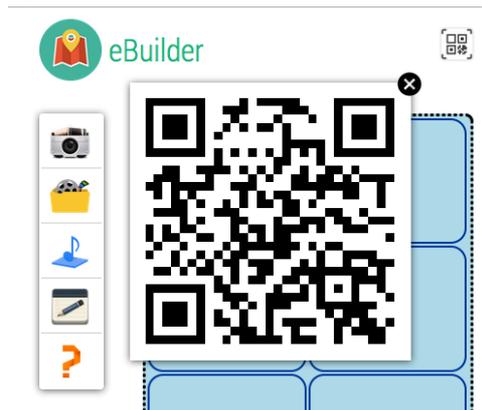


Figura 44 b. Box de creación con QR como mecanismo de sensado.

Por ejemplo, se puede apreciar en la Figura 45, tres elementos posicionados sobre un mapa indoor representado por una imagen modificada del plano de un museo<sup>9</sup>. Los elementos posicionados son: una pregunta, un texto y un elemento de tipo combinado (en el cual se generaron una pregunta y un texto). Estos elementos están relacionados linealmente entre ellos como se puede apreciar.

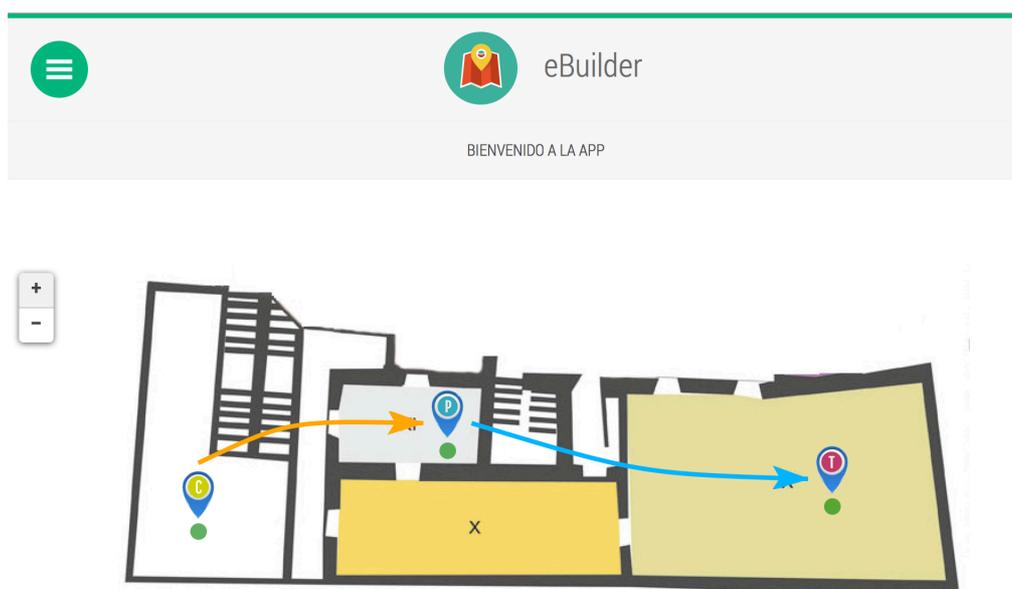


Figura 45. Ejemplo de capa de elementos posicionados utilizando un mapa indoor.

#### 4.4 Importación/Exportación

Como ya se mencionó en el Capítulo 3, el foco de esta tesis es la construcción tanto de las capas de contenidos como de las de posicionamiento y no la derivación de éstas en una

<sup>9</sup> <http://www.museocasalis.org/nuevaweb/visitanos/planos-museo> (Accedido: 10/02/2015)

aplicación móvil (lo cual quedará como trabajo a futuro). Sin embargo, la herramienta sí debe proveer al menos un mecanismo que permite que las especificaciones que haga el usuario durante el uso de la herramienta puedan ser llevadas a un formato exportable.

Esto por esto que, la herramienta da soporte para que el usuario pueda exportar sus creaciones a formato XML, ya sea de los contenidos, de las posiciones y su espacio físico o de ambas a la vez. En futuras evoluciones de la herramienta se podrán exportar a otro tipo de formato. Es por eso que la herramienta utiliza la librería XMLWriter (ver Sección 2.2.3) para generar archivos XML dinámicamente con toda la información que refiere a la creación que el usuario de la herramienta realizó.

Para realizar esto y una vez que el usuario terminó de crear, debe apretar el botón de menú, lo que desplegará el menú con opciones a realizar. Una vez allí, se debe seleccionar el botón “EXPORTAR” (ver Figura 46), lo que iniciará automáticamente la descarga de un archivo .XML con las especificaciones. Este botón siempre está disponible para el usuario independientemente de la pantalla en la que esté trabajando (ya sea creando contenidos, posiciones o elementos posicionados).

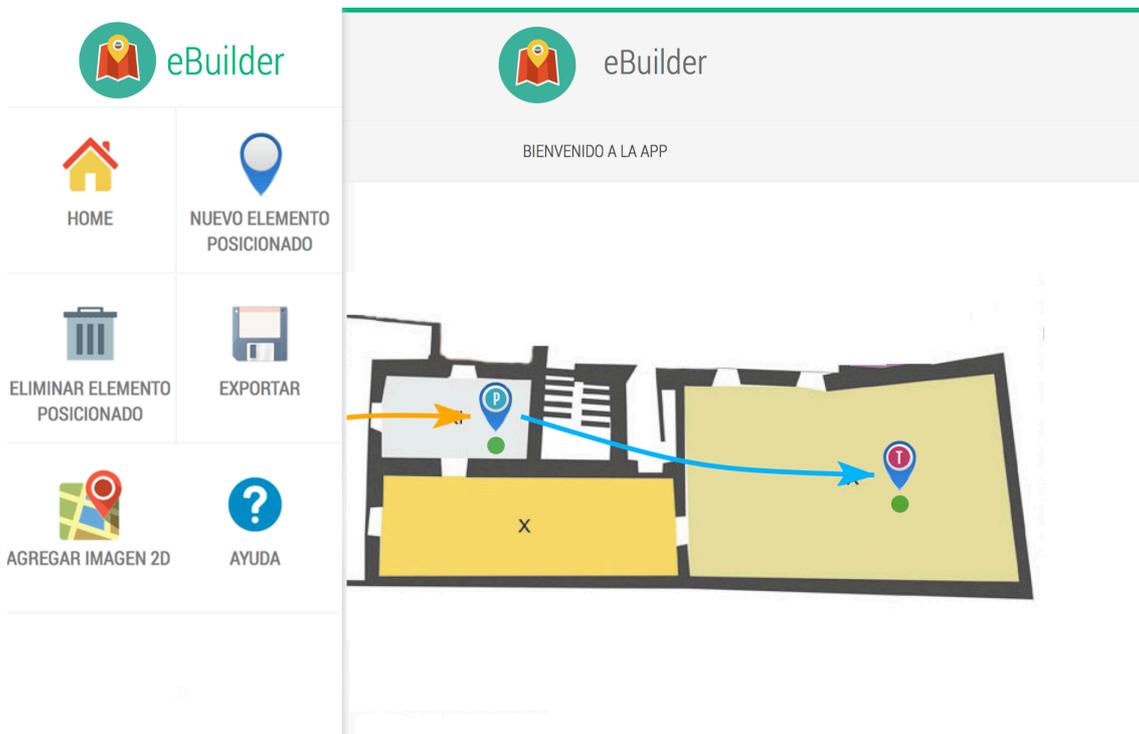


Figura 46. Menú de opciones desplegado. Notar el botón “EXPORTAR”.

Los XML exportados<sup>10</sup> pueden representar toda una experiencia completa que se genera con la herramienta. El tag que contiene toda la experiencia se denomina <Experience>. Dentro de este tag se distinguen seis subgrupos de tags:

<sup>10</sup> Los nombres de los tags usados están relacionados con la definición del framework presentado en [Alconada Verzini et. al, 2015b] para en un futuro, poderlos combinar fácilmente.

- Estructura: Define la estructura que tendrá la experiencia. El nombre del tag puede ser:
  - <ListStructure>, que corresponde a la estructura de secuencia lineal
  - <GraphStructure>, que corresponde a la estructura de tipo grafo
  - <SetStructure>, que corresponde a la estructura de tipo conjunto
  
- Contenidos: Se representa a través del tag <DigitalPointsOfInterest> y engloba a todos los <DigitalPointOfInterest>, este último tag se especifica a continuación:
  - Contenido: Se define dentro del tag <DigitalPointofInterest> y hace referencia a un contenido, tiene un identificador <Id> y una lista de elementos <Elements>. De acuerdo al orden, primero, intermedio o último, en que se han definido los contenidos, un elemento puede ser <FirstElement>, <InnerElement> o <LastElement>, respectivamente. A continuación se listan los tipos de elementos que dichos elementos pueden contener:
    - <TextInformation>: representa el contenido de tipo texto y contiene un identificador <Id>, un texto de contexto <ContextText> (este permite contextualizar el texto en cuestión si es necesario, por ejemplo, haciendo una introducción previa al mismo, para que dicho texto tenga sentido) y finalmente el tag <Text> con el texto propiamente dicho.
    - <ClosedAutoevaluableQuestion>: representa las preguntas de opción múltiple y este tag lleva un atributo llamado “dicotomic” que se setea automáticamente de acuerdo a si la pregunta es dicotómica o no. Al igual que en el caso anterior, las preguntas poseen un identificador <Id> y un texto de contexto (en este caso, la pregunta) <ContextText> y finalmente una lista de respuestas correctas <CorrectAnswers> y otra de respuestas incorrectas <IncorrectAnswers>. Cada respuesta se representa con el tag <Answer> y contiene un tag <Text> con el texto de la respuesta.
    - <Image>: representa al contenido de tipo imagen y está compuesto por un identificador <Id> y una URL a la ubicación donde estuviere almacenada la imagen con el atributo <Source>.
    - <Video>: representa al contenido de tipo video y está compuesto por un identificador <Id> y una URL a la ubicación donde estuviere almacenado el video con el atributo <Source>.

- `<CompositeContent>`: representa al contenido de tipo combinado y está compuesto por un identificador `<Id>`, un texto de contexto `<ContextText>` (este permite contextualizar todos los contenidos que integran este tag, por ejemplo, haciendo una introducción previa antes de listarlos) y una lista de contenidos `<Contents>` que puede ser cualquier de los mencionados anteriormente.
- Relaciones entre contenidos: Las relaciones se definen dentro del tag `<Relations>`, el cual engloba a todas las relaciones entre los puntos de interés digitales, es decir, entre los contenidos.
  - Relación: Representa la conexión entre un punto de interés digital con otro. Está representada con el tag `<Relation>` y se compone de un origen `<SourceId>` y un destino `<TargetId>`, cuyos valores están relacionados con los identificadores de los puntos de interés digitales.
- Posiciones: Se definen dentro del tag `<PhysicalPointsOfInterest>` y engloba a todos los `<PhysicalPointOfInterest>`, este último tag se especifica a continuación:
  - Posición: Cada posición se representa con el tag `<PhysicalPointOfInterest>` y hace referencia a las posiciones físicas dentro de una representación del espacio. Este tag está compuesto por un identificador `<Id>` y un tag `<Position>` que puede tener una o más representaciones de la posición. Estas representaciones pueden ser:
    - Simbólica, representa una etiqueta descriptiva, en este caso usada con códigos QR. Su tag es `<symbolic>`.
    - Georeferenciada, utiliza el posicionamiento global (latitud y longitud). Su tag es `<georeferenced>`.
    - Relativa, se utiliza para posicionar relativamente respecto de un punto X,Y. Su tag es `<relativeXY>`.
- Tipo de mapa: Especifica el tipo de mapa que utiliza la experiencia. El nombre de este tag es siempre `<Map>` y tiene un atributo llamado *type*. Los valores del *type* pueden ser:
  - *Satellite* cuando se utilizan mapas satelitales, en cuyo caso también se setea el valor del atributo *api*, que puede ser *OpenStreetMap* o *GoogleMaps*.
  - *2DMap*, cuando se utilizan imágenes georeferenciadas. Posee un atributo llamado *url* que especifica la ubicación donde estuviere almacenado la imagen del mapa.
  - *Georeferenced*, cuando se utilizan mapas satélites y se le superpone un mapa 2D. Al igual que en el caso de *Satellite*, también existen los atributos *api* (que

puede tomar los valores OpenStreetMap o GoogleMaps) y *url* (que especifica la ubicación donde estuviere almacenado la imagen del mapa).

- Elementos posicionados: Esta definido con el tag <Pols> y engloba a todos los elementos posicionados.
  - Elemento posicionado: Representa la asociación entre un contenido y una posición. Esta asociación esta definida a través del identificador del contenido <IdDigital> y el identificador de la posición <IdPhysical>.

A continuación se presenta, por partes, el XML generado a partir de la experiencia definida en la Figura 45.

En la Figura 47, se puede apreciar la definición de la estructura que, en este caso, es lineal.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Experience>
  <!-- STRUCTURE -->
  <ListStructure>
    <StructureName>ListStructure</StructureName>
  </ListStructure>
```

**Figura 47. Definición de la estructura.**

En la Figura 48 se definieron dos contenidos: un texto y una pregunta de opción múltiple (con una opción correcta y otra incorrecta). Se puede observar que el <DigitalPointOfInterest> posee su identificador y, a su vez, cada uno de los contenidos, define un identificador propio.

```

<!-- DIGITAL POINTS -->
<DigitalPointsOfInterest>
  <DigitalPointOfInterest>
    <Id>d-1</Id>
    <Elements>
      <InnerElement>
        <TextInformation>
          <Id>c-1</Id>
          <ContextText></ContextText>
          <Text>Este es un ejemplo de texto.</Text>
        </TextInformation>
      </InnerElement>
    </Elements>
  </DigitalPointOfInterest>
  <DigitalPointOfInterest>
    <Id>d-2</Id>
    <Elements>
      <InnerElement>
        <ClosedAutoevaluableQuestion dicotomic="false">
          <Id>c-2</Id>
          <ContextText>
            Texto de la pregunta
          </ContextText>
          <CorrectAnswers>
            <Answer>
              <Text>Ejemplo de respuesta correcta</Text>
            </Answer>
          <CorrectAnswers />
          <IncorrectAnswers>
            <Answer>
              <Text>Ejemplo de respuesta incorrecta</Text>
            </Answer>
          </IncorrectAnswers>
        </ClosedAutoevaluableQuestion>
      </InnerElement>
    </Elements>
  </DigitalPointOfInterest>

```

Figura 48. Definición de un texto y una pregunta.

En la Figura 49 se definió un contenido combinado. Se puede observar que el <CompositeContent> posee su identificador pero, en este caso, sus contenidos no definen un identificador propio.

```

<DigitalPointOfInterest>
  <Id>d-3</Id>
  <Elements>
    <InnerElement>
      <CompositeContent>
        <Id>c-3</Id>
        <ContextText></ContextText>
        <Contents>
          <TextInformation>
            <ContextText></ContextText>
            <Text>Este es un ejemplo de texto</Text>
          </TextInformation>
          <ClosedAutoevaluableQuestion dicotomic="false">
            <ContextText>
            </ContextText>
            <CorrectAnswers>
              <Answer>
                <Text>Ejemplo de respuesta correcta</Text>
              </Answer>
            <CorrectAnswers />
            <IncorrectAnswers>
              <Answer>
                <Text>Ejemplo de respuesta incorrecta</Text>
              </Answer>
            </IncorrectAnswers>
          </ClosedAutoevaluableQuestion>
        </Contents>
      </CompositeContent>
    </InnerElement>
  </Elements>
</DigitalPointOfInterest>
</DigitalPointsOfinterest>

```

Figura 49. Definición de un contenido combinado.

En la Figura 50 se definen las relaciones entre los contenidos usando los identificadores especificados en la definición de cada uno de estos.

```

<!-- RELATIONS -->
<Relations>
  <Relation>
    <SourceId>d-3</SourceId>
    <TargetId>d-2</TargetId>
  </Relation>
  <Relation>
    <SourceId>d-2</SourceId>
    <TargetId>d-1</TargetId>
  </Relation>
</Relations>

```

Figura 50. Definición de relaciones entre contenidos.

En la Figura 51, se definen las posiciones relevantes. Se puede observar que cada punto de interés físico se puede posicionar tanto relativamente como con una posición simbólica, que en este caso se corresponderá con el valor que tomé el código QR. En este caso, el tipo de mapa es un mapa indoor y los (X,Y) son relativos a dicho mapa.

```

<!-- PHYSICAL POINTS -->
<PhysicalPointsOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-1</Id>
    <Position>
      <Relative>
        <X>10</X>
        <Y>10</Y>
      </Relative>
      <Symbolic>Posición 1</Symbolic>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-2</Id>
    <Position>
      <Relative>
        <X>38</X>
        <Y>25</Y>
      </Relative>
      <Symbolic>Posición 2</Symbolic>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-3</Id>
    <Position>
      <Relative>
        <X>80</X>
        <Y>17</Y>
      </Relative>
      <Symbolic>Posición 3</Symbolic>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
</PhysicalPointsOfInterest>
<!-- MAP TYPE -->
<Map type="2DMap" url="/Users/usuario/plano_museo.png"/>

```

Figura 51. Definición de posiciones y tipo de mapa.

En la Figura 52 se definen las relaciones entre los contenidos y las posiciones, determinando así los puntos de interés (elementos posicionados).

```

<!-- POINTS OF INTEREST -->
<PoIs>
  <PoI>
    <IdDigital>d-1</IdDigital>
    <IdPhysical>f-3</IdPhysical>
  </PoI>
  <PoI>
    <IdDigital>d-2</IdDigital>
    <IdPhysical>f-2</IdPhysical>
  </PoI>
  <PoI>
    <IdDigital>d-3</IdDigital>
    <IdPhysical>f-1</IdPhysical>
  </PoI>
</PoIs>
</Experience>

```

**Figura 52. Asociación entre contenidos y posiciones.**

Todo lo que el usuario de la herramienta puede generar con ella (elementos posicionados, posiciones, contenidos, conexiones entre elementos posicionados o entre contenidos, selección del espacio, etc.) sería conveniente que pueda ser empaquetado de alguna forma para que así la herramienta permita enviar dicha información empaquetada hacia algún framework capaz de derivar los datos en una aplicación móvil. Esto queda fuera del alcance de este trabajo, la herramienta por ahora solo exporta a XML.

Como se vio anteriormente (ver Figura 22), el menú principal de la herramienta ofrece tres opciones para importar:

- *Importar espacio físico y crear contenidos:* Esta opción permite al usuario importar una capa de posiciones previamente definida y definir contenidos para cada posición que se haya importado. De esta manera, un usuario podría utilizar un mismo recorrido realizado sobre un espacio físico particular, y generar diferentes tipos de contenidos para cada una de las posiciones.
- *Importar contenidos y crear posiciones:* En este caso es la capa de contenidos la que se importa. En primer lugar, el usuario define un tipo de mapa a utilizar y luego importa la capa de contenidos, donde cada uno de ellos se importa como un elemento posicionado aleatoriamente dentro del mapa definido. Con este tipo de importación, se pueden reutilizar los mismos contenidos en diferentes espacios físicos.
- *Importar contenidos y posiciones:* En este caso, tanto la capa de contenidos como la capa de posiciones se importan. En primer lugar, el usuario de la herramienta importa la capa de posiciones para indicarle a la herramienta qué tipo de mapa se utilizará así como también se importan las posiciones definidas en la capa. Estas posiciones importadas son ubicadas sobre el mapa. Luego, se importa la capa de contenidos, que se ubica a un costado del mapa. Una vez realizado esto, el usuario es responsable de arrastrar un contenido sobre una posición de tal forma que se conforme el elemento posicionado. El usuario podría usar las capas importadas en su totalidad o elegir usar solo algunos contenidos o posiciones.

#### 4.5 Eliminación

Para eliminar un contenido, una posición o un elemento posicionado el usuario debe apretar el botón de menú. Una vez que éste se despliega debe presionar sobre el botón llamado “ELIMINAR CONTENIDOS”, “ELIMINAR POSICIONES” o “ELIMINAR ELEMENTOS POSICIONADOS” según corresponda (ver Figura 53).



Figura 53. Botones de eliminación.

De esta se activará el modo eliminación mostrando unos íconos cuadrados de color rojo con una cruz blanca en su interior sobre cada contenido, posición o elemento posicionado (ver Figura 54b). Finalmente, el usuario debe presionar cualquiera de estas cruces para eliminar dicho elemento.

Las Figuras 54a, 54b y 54c muestran un ejemplo de eliminación de un elemento posicionado. En la Figura 54a se ven 3 elementos posicionados sobre un mapa satelital relacionados entre sí. En la Figura 54b se muestra el resultado de haber presionado el botón “ELIMINAR ELEMENTO POSICIONADO”. Finalmente, en la Figura 54c se ve cómo queda el mapa con la eliminación de uno de éstos elementos posicionados así como también las relaciones que salían o llegaban a éste.

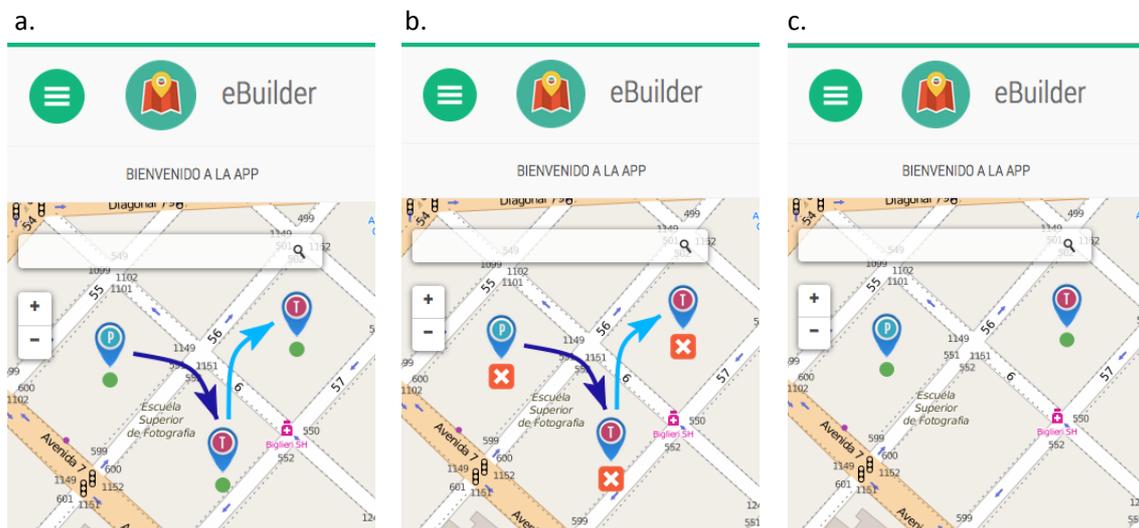


Figura 54. En la Figura 54a, pueden verse tres elementos posicionados sobre un mapa satelital. Luego, en la Figura 54b, el modo de eliminación se encuentra activo, mostrando los cuadrantes rojos con la cruz blanca. Por último, en la Figura 54c se ve el resultado de haber eliminado uno de los tres elementos posicionados y el modo de eliminación desactivado.

## 5. CASOS DE USO DEL PROTITPO PROPUESTO

En este capítulo se presentará un ejemplo de un caso de uso de la herramienta. Este ejemplo mostrará cómo crear contenido posicionado. Para este tipo de creación el usuario de la herramienta debe tener claro el dominio del contenido que quiere crear como el espacio donde se posicionará el mismo.

Supongamos que un usuario decide crear un recorrido en base a un tour virtual del sitio web del museo de Ciencias Naturales de La Plata. Este tour virtual está descrito en el sitio web<sup>11</sup> pero no brinda asistencia móvil, sino que alguien desde cualquier lugar donde se encuentre puede acceder al sitio web del museo y ver la información del mismo. El tour está inspirado en distintos fragmentos del libro "*El viaje del Beagle*" de *Charles Darwin*, muestra en diferentes salas del museo 12 fósiles mencionados en dicho libro. Para cada fósil se cita un fragmento del libro y una foto de dicho animal.

El usuario, al ver el tour virtual descrito anteriormente, puede apreciar que algunos animales todavía no se han extinguido, y además sabe que actualmente hay algunos especímenes en el Zoológico de La Plata.

Acorde a lo descrito anteriormente, piense qué interesante sería poder contar con una aplicación móvil que permita combinar, en un recorrido físico, la información provista por *Darwin*. Es decir, tener un recorrido físico que vaya mostrando aquellos animales extinguidos en el museo y los demás en el zoológico. Esto involucra un espacio abierto como es el zoológico cómo así también un espacio cerrado como es el museo. Para lograr una aplicación móvil como la descrita anteriormente se debe especificar el contenido de la misma como así también el espacio físico donde se posicionarán estos contenidos. Se usará la herramienta para mostrar cómo un ejemplo de este estilo puede ser especificado.

A continuación se brindarán más detalles de esta especificación: a fines prácticos, el recorrido del museo se limitará a 2 animales de los 12 originales, y 2 dentro del zoológico (tomados también de los 12 originales pero en este caso mostrando animales vivos en vez de los embalsamados del museo). Es decir, cada elemento posicionado tendrá información de cada animal, particularmente, una imagen y un texto descriptivo (acorde a los fragmentos del libro de *Darwin* presentados por el museo).

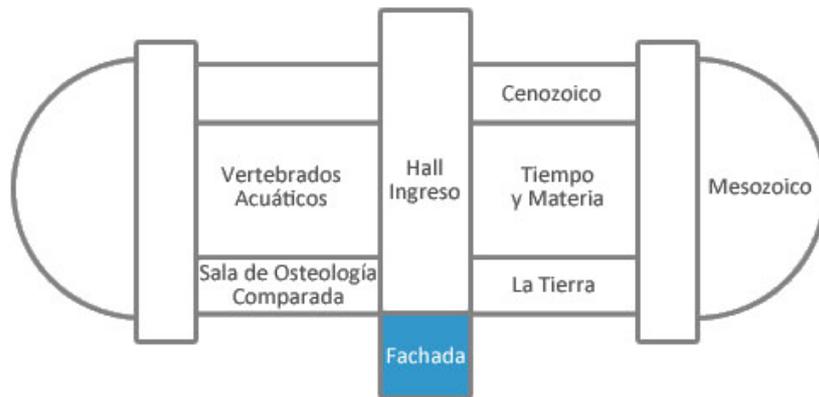
Supongamos que el usuario decide que la estructura del recorrido sea de tipo secuencial, donde el punto de partida será una de las salas del museo y el punto final será un lugar dentro del Zoológico.

La creación será realizada por el usuario de manera in-situ, es decir que dicho usuario se encontrará físicamente presente en el lugar, utilizando una tablet con soporte GPS como dispositivo móvil.

---

<sup>11</sup> <http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/darwinenmuseo.html> (Accedido: 10/02/2015)

El usuario se va a valer del mapa satelital que ofrece la herramienta y además contará con un plano del museo<sup>12</sup> (ver Figura 55) que el usuario deberá cargar a través de la herramienta.



**Figura 55. Plano del museo de Ciencias Naturales de la Plata.**

Supongamos que el usuario quiere especificar el recorrido de la siguiente manera: comenzará en la *Sala Mesozoico* y continuará en el *Hall de Ingreso*. Luego, saldrá del museo para ir al zoológico, donde tendrá como primer destino el *Aviario* y finalizará en la *Granja*, donde concluye el recorrido.

Se detallan a continuación los diferentes contenidos con los cuales el usuario detallará el recorrido. Tanto para aquellos contenidos dentro del museo como para los del zoológico, la información fue extraída desde la página del tour virtual mencionada anteriormente.

- Contenidos dentro del museo
  - TOXODON<sup>13</sup>
    - Ubicación: Sala Mesozoico
    - Imagen de fósil de caballo
    - Texto descriptivo: "...el toxodon, tal vez uno de los más extraños animales que hayan sido descubiertos; en la talla es igual al elefante o megaterio... juzgando por la posición de sus ojos, oídos y narices, era probablemente acuático como el dugong y el manatí, con el que tiene gran parentesco."
  - MEGATERIO<sup>14</sup>
    - Ubicación: Hall de Ingreso
    - Imagen de fósil de megaterio
    - Texto descriptivo: "Eran perezosos terrestres, parientes de los actuales, que habitaron América desde el Plioceno hasta el Holoceno. Se extinguieron hace 8000 años. Medían 6 m de altura parados en sus

<sup>12</sup> [http://www.argentinavirtual.educ.ar/localhost/images/stories/museos/MCNLP/Salas-Planos/01\\_Fachada.jpg](http://www.argentinavirtual.educ.ar/localhost/images/stories/museos/MCNLP/Salas-Planos/01_Fachada.jpg) (Accedido: 10/02/2015)

<sup>13</sup> <http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/piezas/pieza3.html> (Accedido: 10/02/2015)

<sup>14</sup> <http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/piezas/pieza4.html> (Accedido: 10/02/2015)

patas traseras. Las patas delanteras tenían grandes garras para la búsqueda de tubérculos y para la defensa. Sus fuertes mandíbulas constaban de 16 molares (8 en cada maxilar) carentes de esmalte. Eran mamíferos herbívoros poco sociables, aunque actualmente no se descarta una dieta omnívora. Eran animales terrestres, sin duda por su tamaño se veían imposibilitados de ser arborícolas como sus parientes actuales. Se alimentaban en posición bípeda.”

- Contenidos dentro del zoológico
  - BENTEVEO<sup>15</sup>
    - Ubicación: Aviario
    - Imagen de benteveo
    - Texto: "Por las noches el \*Saurophagus\* se posa en un arbusto, y repite continuamente y sin cambios un canto agudo y un tanto agradable que remeda palabras articuladas. Los españoles dicen se parece a las palabras "bien te veo", y le han bautizado con este nombre.”.
  
  - CABALLO<sup>16</sup>
    - Ubicación: Granja
    - Imagen de caballo
    - Texto: "Y también hallé dientes de \*Toxodon\* y mastodonte, junto con el diente de un caballo..."  
"tuve escrupuloso cuidado de comprobar con toda certeza el hecho de haber quedado sepultado al mismo tiempo con los otros fósiles...¡Ciertamente es un hecho maravilloso de la historia de los mamíferos que en Sudamérica haya vivido y desaparecido un caballo indígena, sucedido en edades posteriores por los introducidos por los colonos españoles!"

A continuación se especifican los pasos que se deben realizar para poder crear la experiencia que se definió al comienzo de esta sección:

1. Selección de la vía de acción en el menú principal: se seleccionará la opción “Crear elementos posicionados” (como ya se mostró en la Figura 22 del Capítulo 4).
2. Selección del tipo de mapa: se seleccionará la opción de “MAPA GEOREFERENCIADO” (como ya se mostró en la Figura 35 del Capítulo 4).  
*Al estar in-situ, el mapa automáticamente se centra en la posición actual del usuario.*
3. Desde el menú de opciones (como ya se mostró en la Figura 36 del Capítulo 4), agregamos la imagen georeferenciada del plano del *Museo de Ciencias Naturales de La*

---

<sup>15</sup> <http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/piezas/pieza5.html> (Accedido: 10/02/2015)

<sup>16</sup> <http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/piezas/pieza8.html> (Accedido: 10/02/2015)

Plata y la ubicamos sobre el museo que figura en el mapa satelital, haciendo uso de los redimensionadores (para mover, agrandar o reducir la imagen) y del toolbox (para ajustar su opacidad). Esto se puede apreciar en la Figura 56.

En este ejemplo, todos los elementos serán de tipo combinado, ya que contarán con una imagen y un texto. Además, en los cuatro elementos que posee el ejemplo, una vez que el usuario los define, presiona el botón GUARDAR y la herramienta posicionará dicho elemento sobre el mapa.

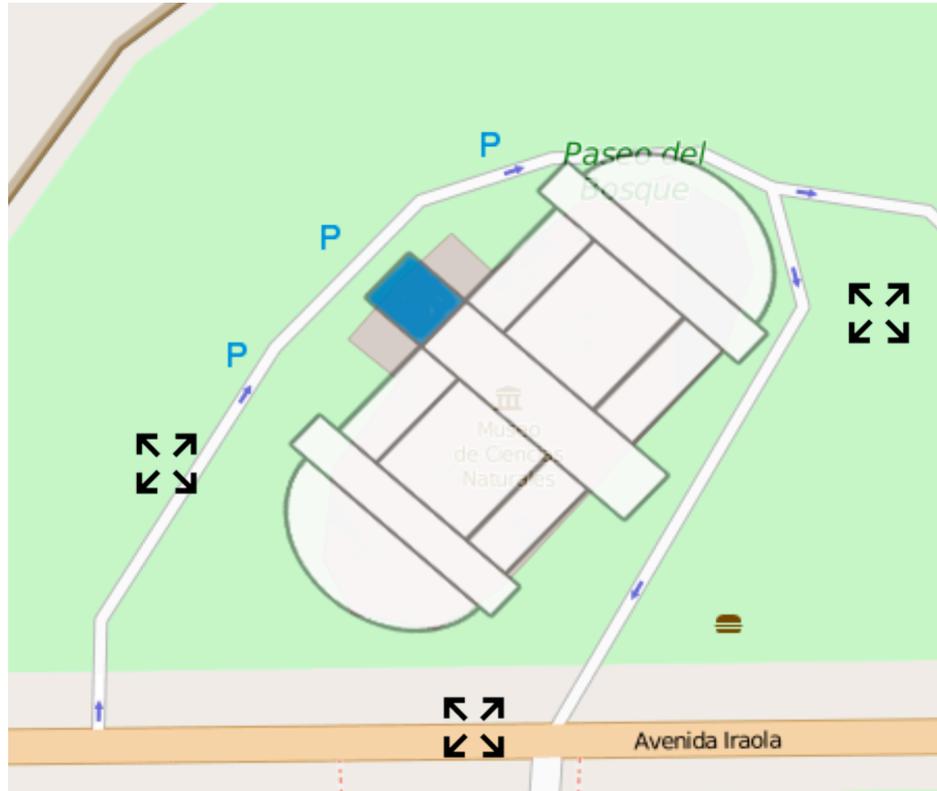


Figura 56. Imagen georeferenciada del museo sobre el mapa satelital.

4. Creación del primer elemento posicionado: en este ejemplo, el primer elemento será el correspondiente al *Toxodon* (ver Figura 57). El elemento se ubicará en la Sala *Mesozoico*. En este caso, el mecanismo de sensado fue configurado a código QR (como puede verse en la esquina superior derecha de la Figura 57). El usuario es el encargado de ajustar la posición del elemento para que quede ubicado en la dicha sala. En la Figura 58 se puede apreciar cómo queda el elemento ubicado sobre el mapa.



- PRIMER ELEMENTO
- ÚLTIMO ELEMENTO
- ELEMENTO INTERMEDIO

GUARDAR

Figura 57. Creación del elemento Toxodon.

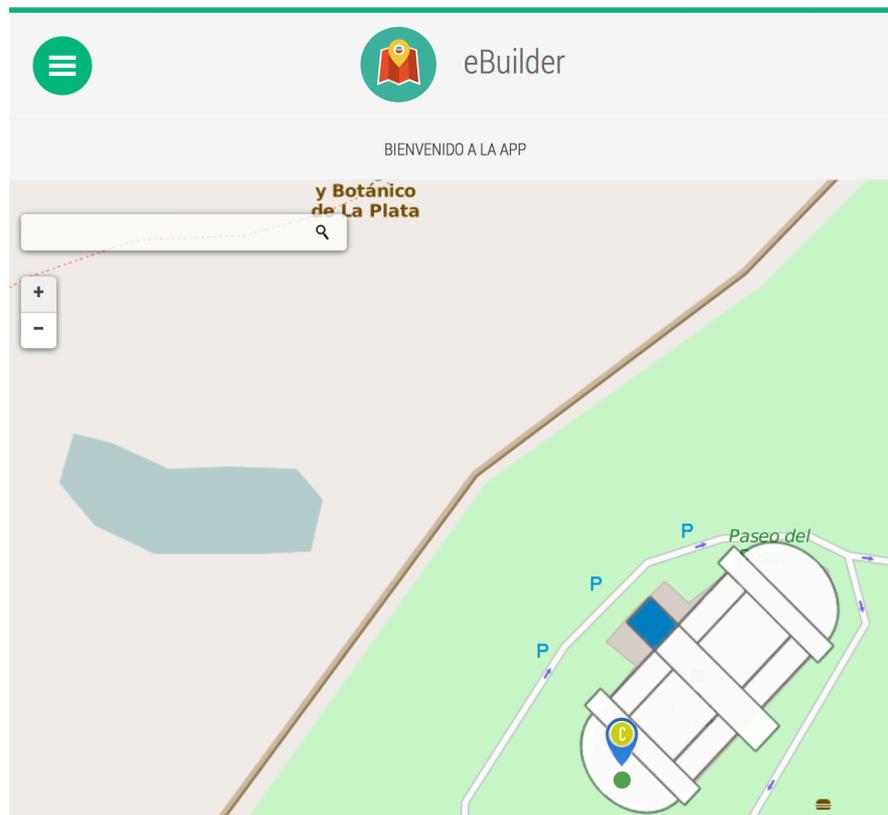


Figura 58. Primer elemento posicionado.

5. Creación del segundo elemento posicionado: este elemento será el correspondiente al *Megaterio* (ver Figura 59). Se ubicará en el *Hall de Ingreso* y, al igual que en el elemento anterior, el usuario deberá ajustar la posición del elemento para ubicarlo en la sala correspondiente. En la Figura 60 se puede apreciar cómo queda ubicado este nuevo elemento posicionado creado, y la relación lineal creada entre ambos.

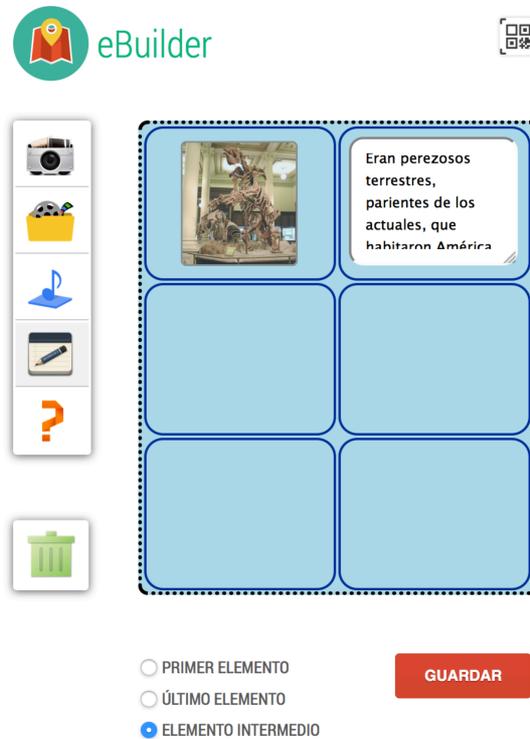


Figura 59. Creación del elemento Megaterio.

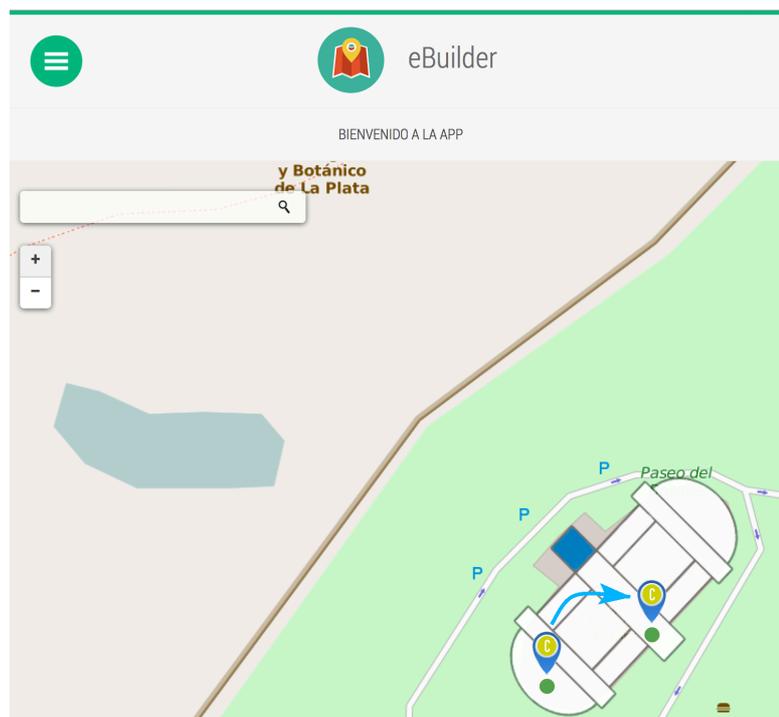


Figura 60. Segundo elemento posicionado.

6. Creación del tercer elemento posicionado: este elemento será el correspondiente al *Benteveo* (ver Figura 61). En este caso, el mecanismo de sensado es el GPS, por lo que el elemento se ubicará automáticamente en la posición del usuario, en este caso, en el *Aviario del Zoológico*. (ver Figura 62, donde se muestra además la relación lineal agregada).

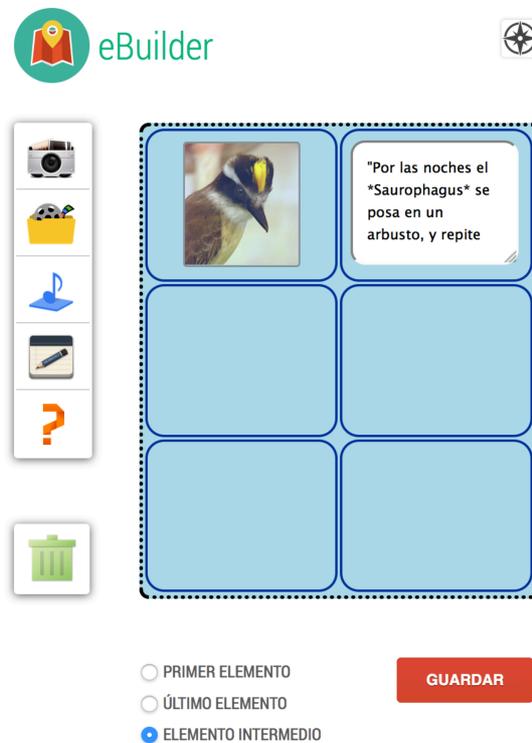


Figura 61. Creación del elemento Benteveo.

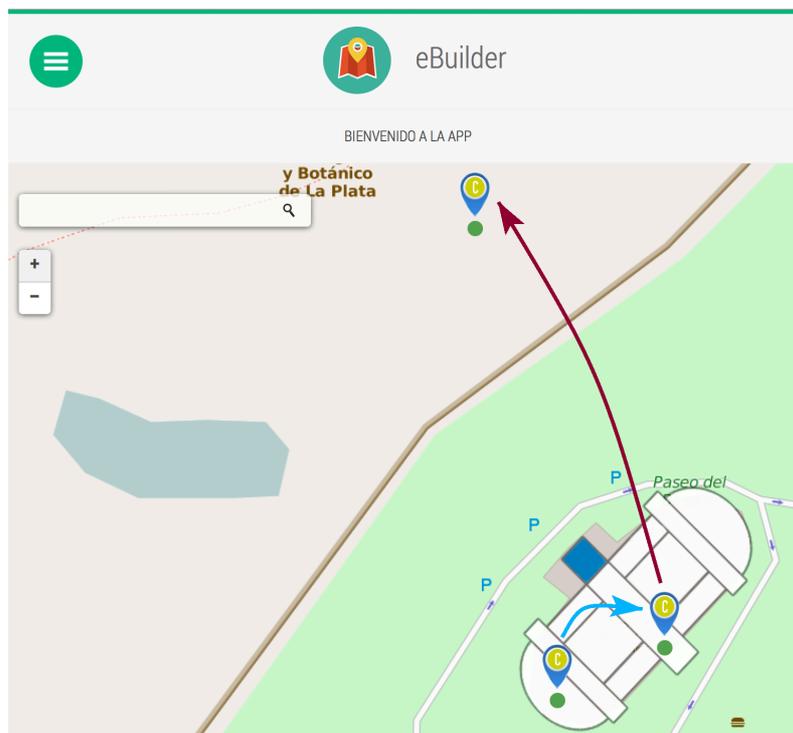


Figura 62. Tercer elemento posicionado.

7. Creación del cuarto elemento posicionado: este elemento será el correspondiente al *Caballo* (ver Figura 63). En este caso, se mantiene el GPS como mecanismo de sensado, por lo que el elemento se ubicará automáticamente en la posición del usuario, en este caso, en la *Granja del Zoológico*. En la Figura 64 se puede apreciar el elemento posicionado y además se agregó la relación lineal entre los elementos.

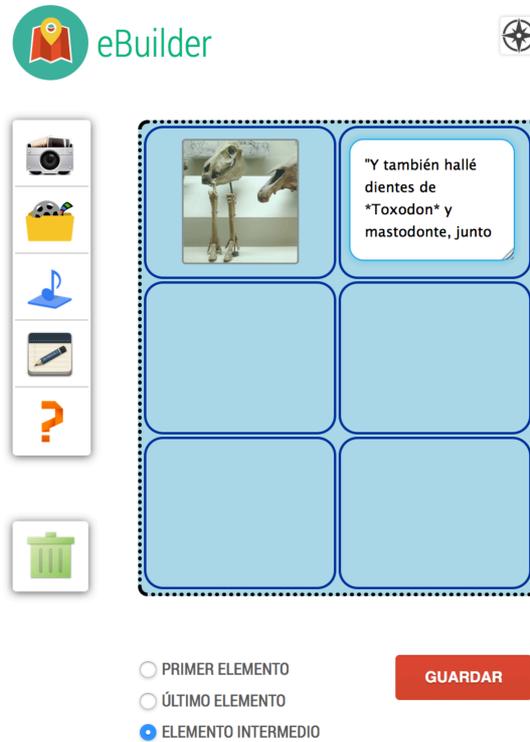


Figura 63. Creación del elemento Caballo.

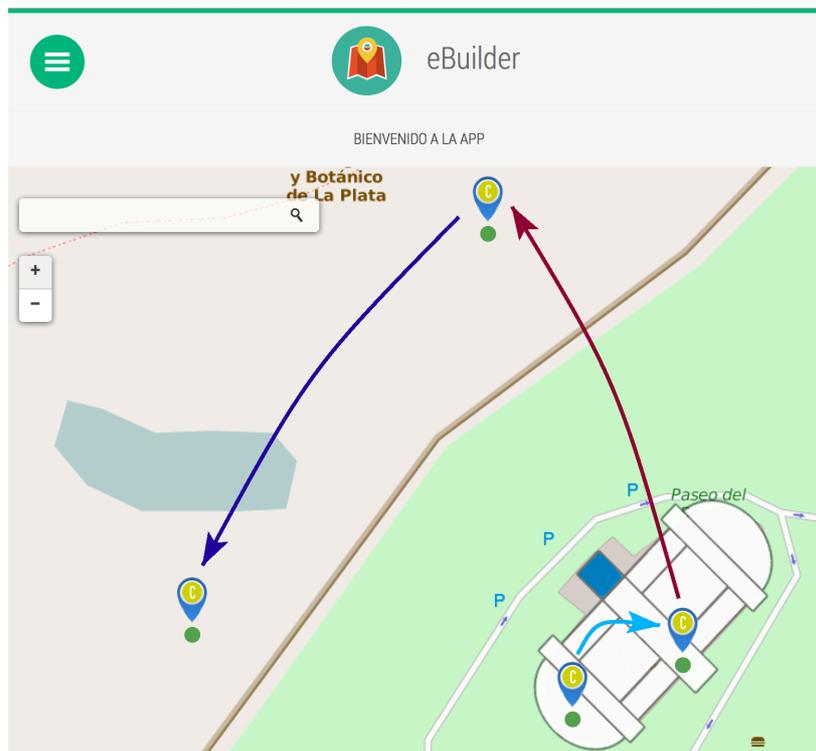


Figura 64. Cuarto elemento posicionado.

De esta manera, quedó creada un ejemplo de experiencia creado con la herramienta desarrollada para esta tesis. Cabe destacar que el usuario puede ir armando el recorrido lineal a medida que crea los elementos posicionados o, si no, estas relaciones las puede especificar al final, es indistinto el momento en el cual se especifican estas relaciones.

A continuación, se presenta el XML generado en función al caso de uso mencionado. En la Figura 65 se define la estructura del caso de uso que, en este caso, es lineal.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Experience>
  <!-- GAME STRUCTURE -->
  <ListStructure>
    <StructureName>ListStructure</StructureName>
  </ListStructure>
```

**Figura 65. Definición de la estructura.**

En la Figura 66 se define el primer contenido acorde a lo especificado para el Toxodon. La imagen asociada tiene la URL a una imagen del museo.

```
<!-- DIGITAL POINTS -->
<DigitalPointsOfInterest>
  <DigitalPointOfInterest>
    <Id>d-1</Id>
    <Elements>
      <InnerElement>
        <CompositeContent>
          <Id>c-1</Id>
          <ContextText></ContextText>
          <Contents>
            <Image>
              <Source>http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/imagenes/toxodon.jpg</Source>
            </Image>
            <TextInformation>
              <ContextText></ContextText>
              <Text>
                "...el toxodon, tal vez uno de los más extraños animales que hayan sido descubiertos; en la talla es igual al elefante o megaterio... juzgando por la posición de sus ojos, oídos y narices, era probablemente acuático como el dugong y el manatí, con el que tiene gran parentesco."
              </Text>
            </TextInformation>
          </Contents>
        </CompositeContent>
      </InnerElement>
    </Elements>
  </DigitalPointOfInterest>
```

**Figura 66. Definición del primer contenido.**

En la Figura 67 se define el segundo contenido acorde a lo especificado para el Megaterio. La imagen asociada tiene la URL a una imagen del museo.

```

<DigitalPointOfInterest>
  <Id>d-2</Id>
  <Elements>
    <InnerElement>
      <CompositeContent>
        <Id>c-2</Id>
        <ContextText></ContextText>
        <Contents>
          <Image>
            <Source>http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/
              imagenes/megaterio.jpg</Source>
          </Image>
          <TextInformation>
            <ContextText></ContextText>
            <Text>
              "Eran perezosos terrestres, parientes de los actuales, que
                habitaron América desde el Plioceno hasta el Holoceno. Se
                extinguieron hace 8000 años. Medían 6 m de altura parados
                en sus patas traseras. Las patas delanteras tenían
                grandes garras para la búsqueda de tubérculos y para la
                defensa. Sus fuertes mandíbulas constaban de 16 molares
                (8 en cada maxilar) carentes de esmalte. Eran mamíferos
                herbívoros poco sociables, aunque actualmente no se
                descarta una dieta omnívora. Eran animales terrestres,
                sin duda por su tamaño se veían imposibilitados de ser
                arborícolas como sus parientes actuales. Se alimentaban
                en posición bípeda."
            </Text>
          </TextInformation>
        </Contents>
      </CompositeContent>
    </InnerElement>
  </Elements>
</DigitalPointOfInterest>

```

Figura 67. Definición del segundo contenido.

En la Figura 68 se define el tercer contenido acorde a lo especificado para el Benteveo. La imagen asociada tiene la URL a una imagen del museo.

```

<DigitalPointOfInterest>
  <Id>d-3</Id>
  <Elements>
    <InnerElement>
      <CompositeContent>
        <Id>c-3</Id>
        <ContextText></ContextText>
        <Contents>
          <Image>
            <Source>http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/
              imagenes/benteveo.jpg</Source>
          </Image>
          <TextInformation>
            <ContextText></ContextText>
            <Text>
              "Por las noches el *Saurophagus* se posa en un arbusto, y
                repite continuamente y sin cambios un canto agudo y un
                tanto agradable que remeda palabras articuladas. Los
                españoles dicen se parece a las palabras "bien te veo", y
                le han bautizado con este nombre."
            </Text>
          </TextInformation>
        </Contents>
      </CompositeContent>
    </InnerElement>
  </Elements>
</DigitalPointOfInterest>

```

Figura 68. Definición del tercer contenido.

En la Figura 69 se define el cuarto contenido acorde a lo especificado para el Caballo. La imagen asociada tiene la URL a una imagen del museo.

```

<DigitalPointOfInterest>
  <Id>d-4</Id>
  <Elements>
    <InnerElement>
      <CompositeContent>
        <Id>c-4</Id>
        <ContextText></ContextText>
        <Contents>
          <Image>
            <Source>http://www.fcnym.unlp.edu.ar/museo/educativa/darwin/
              imagenes/caballo.jpg</Source>
          </Image>
          <TextInformation>
            <ContextText></ContextText>
            <Text>
              "Y también hallé dientes de *Toxodon* y mastodonte, junto con
              el diente de un caballo..."
              "tuve escrupuloso cuidado de comprobar con toda certeza el
              hecho de haber quedado sepultado al mismo tiempo con los
              otros fósiles...¡Ciertamente es un hecho maravilloso de la
              historia de los mamíferos que en Sudamérica haya vivido y
              desaparecido un caballo indígena, sucedido en edades
              posteriores por los introducidos por los colonos
              españoles!"
            </Text>
          </TextInformation>
        </Contents>
      </CompositeContent>
    </InnerElement>
  </Elements>
</DigitalPointOfInterest>
</DigitalPointsOfinterest>

```

**Figura 69. Definición del cuarto contenido.**

En la Figura 70 se definen las relaciones entre los contenidos utilizando los identificadores de cada uno de ellos.

```

<!-- RELATIONS -->
<Relations>
  <Relation>
    <SourceId>d-1</SourceId>
    <TargetId>d-2</TargetId>
  </Relation>
  <Relation>
    <SourceId>d-2</SourceId>
    <TargetId>d-3</TargetId>
  </Relation>
  <Relation>
    <SourceId>d-3</SourceId>
    <TargetId>d-4</TargetId>
  </Relation>
</Relations>

```

**Figura 70. Definición de las relaciones entre los contenidos.**

En la Figura 71 se definen las posiciones relevantes. Se puede observar que todos los puntos están georeferenciados y, aquellos definidos dentro del museo, además, poseen una posición simbólica, que en este caso se corresponderá con el valor que tomé el código QR. En este caso, el tipo de mapa es un mapa georeferenciado (indoor-outdoor).

```

<!-- PHYSICAL POINTS -->
<PhysicalPointsOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-1</Id>
    <Position>
      <Georeferenced>
        <Lat>-34.90929809390054</Lat>
        <Lng>-57.935811281204224</Lng>
      </Georeferenced>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-2</Id>
    <Position>
      <Georeferenced>
        <Lat>-34.909020947652444</Lat>
        <Lng>-57.935290932655334</Lng>
      </Georeferenced>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-3</Id>
    <Position>
      <Georeferenced>
        <Lat>-34.90727006560451</Lat>
        <Lng>-57.93622434139251</Lng>
      </Georeferenced>
      <Symbolic>Posición 3</Symbolic>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
  <PhysicalPointOfInterest>
    <Id>f-4</Id>
    <Position>
      <Georeferenced>
        <Lat>-34.90884498129508</Lat>
        <Lng>-57.93759763240815</Lng>
      </Georeferenced>
      <Symbolic>Posición 4</Symbolic>
    </Position>
    <Space />
  </PhysicalPointOfInterest>
</PhysicalPointsOfInterest>
<!-- MAP TYPE -->
<Map type="Georeferenced" api="OpenStreetMap" url="/Users/user/museo.png"/>

```

**Figura 71. Definición de las posiciones.**

En la Figura 72, se definen las asociaciones entre contenidos y posiciones usando los identificadores previamente definidos.

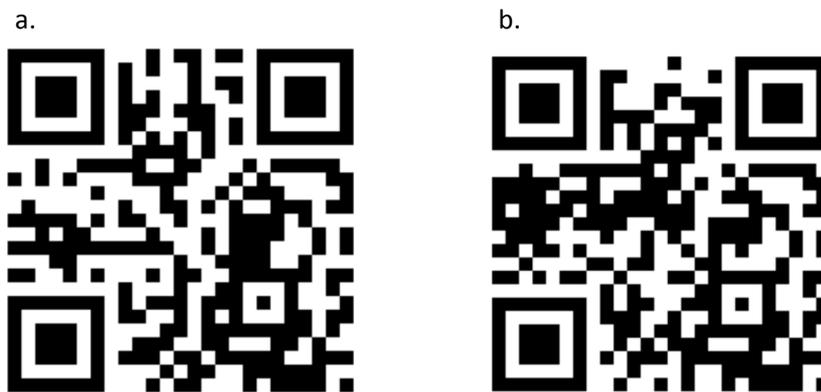
```

<!-- POINTS OF INTEREST -->
<PoIs>
  <PoI>
    <IdDigital>d-1</IdDigital>
    <IdPhysical>f-1</IdPhysical>
  </PoI>
  <PoI>
    <IdDigital>d-2</IdDigital>
    <IdPhysical>f-2</IdPhysical>
  </PoI>
  <PoI>
    <IdDigital>d-3</IdDigital>
    <IdPhysical>f-3</IdPhysical>
  </PoI>
  <PoI>
    <IdDigital>d-4</IdDigital>
    <IdPhysical>f-4</IdPhysical>
  </PoI>
</PoIs>
</Experience>

```

**Figura 72. Asociación entre contenidos y posiciones.**

En la Figura 73a y en la Figura 73b se muestran los códigos QR correspondientes a los dos primeros elementos posicionados del caso de uso. A futuro, cuando la herramienta evolucione y permita generar aplicaciones móviles ejecutables, dichos códigos QR deberán ser ubicados en las posiciones adecuadas.



**Figura 73a. Código QR correspondiente al primer elementos posicionado. Figura 73b. Código QR correspondiente al segundo elemento posicionado.**

## 6. CONCLUSIONES Y TRABAJOS FUTUROS

En esta tesis se presentó una herramienta de creación de elementos posicionados in-situ que permite, a usuarios no expertos, definir experiencias con contenidos multimedia posicionados (como por ejemplo, audios, videos, imágenes, textos o una combinación de ellos) tanto en espacios indoor como outdoor. Se presentó, en el Capítulo 3, un esquema con los conceptos constitutivos de la herramienta que hace foco en la reusabilidad y extensibilidad de los mismos.

Se utilizó el concepto de *separación en capas* para separar los aspectos de contenido y posicionamiento. De esta manera, se facilita la reutilización de los aspectos mencionados así como también la evolución de la herramienta en general. Nombramos a la relación entre un contenido y una posición como elemento posicionado.

Se pudo apreciar mediante un ejemplo concreto como llevar a cabo la definición de elementos posicionados en un ambiente indoor-outdoor. Además, de mostrar cómo se puede cambiar el mecanismo de sensado que se utiliza para tomar las posiciones de los mismos.

La herramienta presentada en este trabajo fue parte del material publicado en dos publicaciones (valga la redundancia), las cuales están estrechamente relacionadas con la tesina presentada ([Alconada Verzini et. al, 2015a] y [Alconada Verzini et. al, 2015b]). Al final de la tesis se describe con más detalle la relación de cada uno de estas publicaciones con esta tesis. Cabe destacar que he sido la persona en ir a presentar ambos trabajos al workshop "*Narrative and Hypermedia 2015*", esto permitió tomar contacto con trabajos similares, y poder apreciar que todavía es una área abierta de trabajo, en la cual falta mucho por explorar.

A partir del desarrollo realizado para esta tesis, surgen algunas ideas, mejoras y trabajos a futuro. A continuación, se listan algunos trabajos a futuro que permitirían ampliar las funcionalidades existentes de la herramienta:

- Permitir que el usuario pueda subir tanto las capas de contenidos como las de posiciones a un repositorio, de tal forma que otros usuarios sean capaces de descargarlas y reutilizarlas para definir sus propias experiencias.
- Combinar la herramienta con un framework capaz de generar aplicaciones móviles en base a los elementos definidos en una experiencia (acorde a lo definido en [Alconada Verzini et. al, 2015b]). De esta forma, la herramienta sería capaz de interactuar con el framework para hacer el "deploy" de aplicaciones, y así brindarle al usuario la posibilidad de generar ellos mismos sus propias aplicaciones.
- Dar soporte a otros mecanismos de sensado, por ejemplo, Bluetooth o NFC. Para cada uno se deberá analizar la características de los mismos, y acorde a ellos, determinar la mejor manera de integrarlos a la herramienta.
- Agregar más servicios de mapas. Esto requerirá el análisis de las distintas APIs de mapas para integrarlas a la herramienta.

- Agregar la posibilidad de definir relaciones entre las posiciones, generando así posibles caminos entre las mismas.

Además, a continuación se listan algunas funcionalidades y mejoras que podrían añadirse como nuevas características para la herramienta:

- Enriquecer los tipos de contenido que un usuario puede crear (por ejemplo, permitir la toma de fotografías utilizando la cámara del dispositivo móvil así como la grabación de videos).
- Enriquecer la herramienta para la definición de contextos a la hora de la creación de contenido, esto puede implicar tomar datos de otros sensores. Esto requerirá un análisis exhaustivo de los contextos, sensores, y encontrar un lenguaje amigable para que el usuarios los pueda definir en una forma fácil.
- Enriquecer la herramienta para poder acomodar visualmente los contenidos creados, haciendo una especie de mockup reales de la visualización que le va a quedar al usuario.
- Enriquecer la herramienta con simulación on-the-fly de lo que se va creando, como tener un modo edición y un modo simulación para apreciar como lo recibe el usuario.
- Agregar condiciones a las relaciones, esto implica poder definir las en un lenguaje natural. Estas condiciones determinarán luego el flujo que puede seguir el usuario a la hora de usar la aplicación.

Es importante destacar que la representación de los espacios indoor es todo un campo de investigación. En la herramienta, dichos espacios son representados a través de imágenes georeferenciadas, pero podría pensarse, por ejemplo, en representaciones 3D o cualquier otra forma de representación más compleja [Zlatanova et. al, 2013]. La mayoría de los espacios indoor no cuentan con mapas con información del lugar, a veces son solo bosquejos, con lo cual el usuario construir el mapa, si estos son solo imágenes, luego será complejo poder brindar asistencia, por ejemplo, para determinar por donde se puede caminar. Esto es necesario, por ejemplo, si desde la herramienta se define una relación entre dos elementos posicionados, se podría determinar si hay un camino para llegar entre ambos, y si no, avisarle al creador que luego el usuario no va a poder llegar a dicho lugar.

Hasta el momento, la herramienta sólo ha sido evaluada por los desarrolladores involucrados en su definición. Pero sabemos que es importante que sea evaluada con usuarios finales. Para alcanzar esto, se está organizando un experimento para que usuarios finales utilicen la herramienta de tal forma que puedan crear aplicaciones 'location-aware' que involucren espacios indoor y outdoor. Estas pruebas son importantes para obtener resultados en cuanto a la usabilidad y funcionalidad de la herramienta así como también para la detección de bugs y errores. También, del feedback obtenido pueden surgir nuevas ideas para la mejora de la herramienta.

Diferentes autores destacan la importancia de realizar pruebas in-situ. En [Pittarello, 2011] los autores destacan que la percepción que tienen los usuarios del ambiente físico puede variar,

por ejemplo, acorde al momento del día. Esto puede ser un factor importante a tener en cuenta al momento de definir, por ejemplo, contenidos posicionados ya que no será la misma la percepción del usuario acorde, por ejemplo, al momento del día en que lo defina. Más aun deberán tener en cuenta en qué momento del día se va a usar la aplicación creada. Por ejemplo, supongamos que un usuario usa la herramienta durante el día, define elementos posicionados en un espacio indoor usando el mecanismo de sensado por QR. Y dicha información luego es usada para un recorrido nocturno, puede ser que la ubicación elegida para los QR puede no persistir si el lugar no tiene buena iluminación. Estos son aspectos que en un futuro habrá que analizar para lograr una herramienta que considere estas situaciones.

## 7. REFERENCIAS BIBLIOGRÁFICAS

- [AffineImageOverlay]. AffineImageOverlay:  
<https://github.com/thatjpk/LeafletAffineImageOverlay>
- [Alconada Verzini et .al, 2015a] Alconada Verzini, A.M. Tonelli, J.I., Challiol, C., Llitas, A.B., & Gordillo S.E. Authoring Tool for Location-Aware Experiences. In *Proceeding of 2015 Narrative and Hypertext Workshop*. ACM, pp. 21-25. ISBN 978-1-4503-3797-7/15/09, DOI: <http://dx.doi.org/10.1145/2804565.2804570>.
- [Alconada Verzini et. al, 2015b] Alconada Verzini, A.M., Tonelli, J.I., Challiol, C., Llitas, A.B., & Gordillo S.E. Combing Location-Aware Applications with in-situ Actors Performances. In *Proceeding of 2015 Narrative and Hypertext Workshop*. ACM, pp. 27-31. ISBN 978-1-4503-3797-7/15/09, DOI: <http://dx.doi.org/10.1145/2804565.2804571>.
- [Annika, 2007] Annika Paus.: *Near Field Communication in Cell Phones*, 2007.
- [Gu, 2009] Gu, Y., Lo, A., Niemegeers, I.G.: A Survey of Indoor Positioning Systems for Wireless Personal Networks. *IEEE Communications Surveys and Tutorials*, 11 (1), 13-32 (2009)
- [Hansen et. al, 2009] Hansen, R., Wind, R., Jensen, C.S., Thomsen, B.: Seamless Indoor/Outdoor Positioning Handover for Location-Based Services in Streamspin. In *Procs. of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*. IEEE Computer Society, Washington, DC, USA, pp. 267-272. (2009)
- [Hansen, 2012] Hansen, F. A., Kortbek, K. J., and Grønbaek, K. 2012. Mobile urban drama: interactive storytelling in real world environments. *New Review of Hypermedia and Multimedia*18, 1-2, (Mar.-June 2012), 63-89.
- [jsPlumb]. jsPlumb: <https://jsplumbtoolkit.com>
- [Kjeldskov and Paay, 2007] Kjeldskov, J., and Paay, J. 2007. Augmenting the City with Fiction: Fictional Requirements for Mobile Guides. In *Proceedings of Mobile Interaction with the Real World 2007/5th Workshop on HCI in Mobile Guides* (Singapore, September 9, 2007). MIRW 2007. ACM, New York, NY, 1- 6
- [Lai, 2010] Lai, Y.C., Han, F., Yeh, Y.H., Lai, C.N., Szu, Y.C.: A GPS navigation system with QR code decoding and friend positioning in smart phones. In: *2nd International Conference on Education Technology and Computer*, 5, pp. 55-66. IEEE Press (2010)
- [LeafletJS]. LeafletJS: <http://leafletjs.com>
- [Leonhardt, 1998] Leonhardt, U.: *Supporting Location Awareness in Open Distributed Systems*. Ph.D.Thesis, Dept. of Computing, Imperial College, London. (1998)
- [Llitas et. al, 2011] Llitas, A. B., Challiol, C., Mostaccio, C., & Gordillo, S. Representaciones enriquecidas para la navegación indoor-outdoor en aplicaciones móviles. In *XVII Congreso Argentino de Ciencias de la Computación*. 2011
- [Llitas et. al, 2012] Llitas, A.B., Challiol, C. y Gordillo, S. Juegos Educativos Móviles Basados en Posicionamiento: Una Guía para su Conceptualización . 41 JAIIO. Agosto de 2012.

- Facultad de Informática, UNLP. Con referato. In Proceedings of ASSE 2012 Argentine Symposium on Software Engineering. ISSN: 1850-2792, pp. 164-175. 2012
- [Llitas et. al, 2013] Llitas, A.B., Challiol, C. y Gordillo, S. Juegos Educativos Móviles: Aspectos Involucrados. In *XVIII Congreso Argentino de Ciencias de la Computación*, pp. 671-680. 2013
- [Phoneygap] Phoneygap. Open Source Mobile Framework. <http://phoneygap.com>
- [Pittarello, 2011] Pittarello, F. 2011. Designing a context-aware architecture for emotionally engaging mobile storytelling. In Proceeding Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction (Lisbon, Portugal, September 5-9, 2011). INTERACT '11. Springer-Verlag Berlin, Heidelberg, 144-151.
- [Pittarello & Bertani, 2012] Pittarello, Bertani.: CASTOR: Learning to create Context-Sensitive and Emotionally Engagings In-Situ. IDC 2012, June 12-15, 2012, Bremen, Germany.
- [Realinho et. al, 2011a] Realinho, Romão, Birra, Dias.: Building Mobile Context-Aware Applications for Leisure and Entertainment. ACE'2011 – Lisbon, Portugal.
- [Realinho et. al, 2011b] Realinho, Romão, Birra, Dias.: Rapid Development of Mobile Context-Aware Applications with IVO. ACE'2011 – Lisbon, Portugal.
- [Realinho et. al, 2011c] Realinho, Romão, Dias.: Testing the Usability of a Platform for Rapid Development of Mobile Context-Aware Applications. P. Campos et. al (Eds.): INTERACT 2011, Part III, LNCS 6948, pp. 521–536, 2011
- [Realinho et. al, 2012] Realinho, Romão, Dias.: An Event-Driven Workflow Framework to Develop Context-Aware Mobile Applications. *MUM'12*, December 04–06, 2012, Ulm, Germany.
- [Rouillard, 2008] Rouillard, J.: Contextual QR Codes. Proceedings of the Third International Multi-Conference on Computing in the Global Information Technology. ICCGI 2008. July 27 – August 1, 2008 – Athens, Greece
- [Santos et. al, 2011] Santos, Pérez-Sanagustín, Hernández-Leo, Blat.: QuesTInSitu: from tests to routes for assessment in situ activities.
- [XML]. W3C Extensive Markup Language: <http://www.w3.org/XML/>
- [XMLWriter]. XMLWriter: <http://flesler.blogspot.com.ar/2008/03/xmlwriter-for-javascript.html>
- [Zlatanova et. al, 2013] Zlatanova, Liu, Sithol.: A conceptual framework of space subdivision for indoor navigation. *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness* (Orlando, Florida, USA, November 5-8, 2013). ISA 2013. ACM, New York, NY

## 8. PUBLICACIONES REALIZADAS

A continuación se listan los trabajos publicados que se relacionan con la temática de la tesis realizada, se detalla de cada uno de estos trabajos un resumen del mismo y los aspectos relacionados a la tesis.

- *Authoring Tool for Location-Aware Experiences* [Alconada Verzini et. al, 2015a]

Resumen Traducido: En este paper presentamos un enfoque para crear experiencias 'location-based'. Utilizamos el concepto de 'separation of concerns' para representar tanto la capa de contenidos y la capa de posicionamiento. Esta separación permite reusar las capas independientemente una de otro. El foco de este paper es proveer una herramienta de usuario final para crear aplicaciones 'location-based' in-situ. A través del uso de esta herramienta, las experiencias 'location-aware' pueden ser definidas tanto en espacios indoor como en espacios outdoor. Presentamos también un ejemplo de cómo nuestra herramienta se utiliza y describimos algunos puntos de discusión que surgieron al definir la herramienta con estas características.

*Relación con la tesis presentada:*

En esta publicación se presenta la herramienta visual descrita en esta tesis, reflejando todo lo anteriormente descrito sobre las características de las mismas. Esto permitió que dicho trabajo sea evaluado por la comunidad científica para poder obtener así un feedback respecto de los potenciales y/o falencias de dicha herramienta.

- *Combing Location-Aware Applications with in-situ Actors Performances* [Alconada Verzini et. al, 2015b]

Resumen Traducido: En este paper se presenta un modelo para definir experiencias 'location-aware' teniendo sus diferentes aspectos (contenidos, posiciones, etc.) desacoplados unos de otros. Creamos una aplicación siguiendo este modelo que se combinó con actuaciones de actores reales. Esta aplicación define tres posiciones relevantes dentro de un edificio específico. Al leer un código QR, el usuario indica que ha llegado a una de esas posiciones. En ese momento, la aplicación muestra algunas preguntas relacionadas a las actuaciones de los actores. Entonces, el usuario debe ver todas las actuaciones para poder contestar las preguntas. La aplicación asiste al usuario para moverse a través de las diferentes posiciones. Presentamos también la evaluación de esta aplicación y luego discutimos algunos aspectos interesantes que surgieron como resultado de la evaluación in-situ de la aplicación.

*Relación con la tesis presentada:*

En esta publicación se presenta un enfoque de modelado para aplicaciones sensibles a la posición, y además una experiencia de uso concreta desarrollada con dicho enfoque. La experiencia reflejada en el paper permitió entender las necesidades de la creación de este tipo de aplicaciones, y esto sirvió de base a la

hora de plantear la herramienta presentada en esta tesis. Además, el enfoque de modelado en un futuro será combinado con la herramienta presentada para poder generar aplicaciones finales ya ejecutables.

Por la integra relación de los papers [Alconada Verzini et. al, 2015a] y [Alconada Verzini et. al, 2015b] con esta tesis, a continuación se presentan los mismos.

# Authoring Tool for Location-Aware Experiences

Federico M. Alconada Verzini

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

falconada@lifa.info.unlp.edu.ar

Juan I. Tonelli

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

jtonelli@lifa.info.unlp.edu.ar

Cecilia Challiol<sup>‡</sup>

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

ceciliac@lifa.info.unlp.edu.ar

Alejandra B. Lliteras

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

lliteras@lifa.info.unlp.edu.ar

Silvia E. Gordillo<sup>§</sup>

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

gordillo@lifa.info.unlp.edu.ar

## ABSTRACT

In this paper, we present an approach to create location-aware experiences. We use the concept of separation of concerns to represent the content layer and the location layer. This separation allows reusing the layers independently one of each other. The focus of this paper is to provide an end-user tool to create location-aware applications in-situ. By using our tool, the location-aware experiences can be defined in both indoor and outdoor spaces. We present an example of how our tool is used and we describe some discussion points that have occurred to us while defining a tool with these characteristics.

## Categories and Subject Descriptors

D.2.2 [Software Engineering]: Design Tools and Techniques – *Object-oriented design methods, User interfaces.*

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *User-centered design*

## General Terms

Design

## Keywords

In-situ Authoring Tool; Location-Aware Experiences; Indoor-Outdoor Space; Separation of Concern; Mobile Applications

## 1. INTRODUCTION

The increasing capabilities of mobile devices (e.g. GPS, NFC, etc.) are leading to a new way of creating mobile applications in

which end-users are getting involved. Nowadays, users not only create contents (e.g. taking pictures, creating comments, etc.) but also mobile applications; in particular Location-Aware applications [13].

Location-Aware applications can be used for different purposes and no matter which domain they have, the user of the application receives information based on his physical location. These domains can vary, going from education to games (more examples are mentioned in [7]).

Up until the last few years this kind of applications has been being created by developers. But this has changed with the growing use of mobile devices and users have become part of these creations.

According to [12], end-users have two different ways to design location-aware applications: digital or physical. In the first case (digital design), a map is used as a reference to define the relevant locations and contents but users are not physically located there. For example, [10] and [12] approaches provide authoring tools that use maps to define relevant locations (and its associated content). In the second case (physical design), the user is physically located in the place where the relevant locations and contents will be defined. In the literature, this last way of designing is generally known as “in-situ creation” ([4] and [9]). As mentioned in [11] and [13], being in-situ enables users to identify relevant environmental aspects of the real world. This can be important not only to choose a more suitable location but also to define a more accurate content associated with each place.

In general, in-situ tools create a content coupled with a location and are only focused on outdoor applications. For example, CASTOR [9] associates a GPS location with each content. Another similar example is TOTEM [4], which allows choosing a GPS or an NFC location. Both approaches provide support to create only outdoor mobile applications.

In this paper, we present an authoring tool to create location-aware applications not only for outdoor spaces but also for indoor ones. This tool has been defined in a generic way so it can be used for any domain (education, entertainment, etc.). In addition, we have focused on providing a solution to decouple contents

<sup>‡</sup> Another affiliation for Cecilia Challiol: CONICET, Argentina.

<sup>§</sup> Another affiliation for Silvia E. Gordillo: CIC, Buenos Aires, Argentina.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*NHT '15*, September 1, 2015, Guzelyurt, TRNC, Cyprus

© 2015 ACM. ISBN 978-1-4503-3797-7/15/09... \$15.00

DOI: <http://dx.doi.org/10.1145/2804565.2804570>

from locations by using the concept of separation of concerns which we have been analysing and modelling for mobile applications in our previous works ([1] and [2]). Furthermore, we present an example of how our tool is used and we describe some issues that have sprung to our minds while defining a tool with these characteristics.

This paper is organised as follows. In Sections 2 we describe some related works. Our approach is presented in Section 3. Our authoring tool is shown in Section 4. Conclusions and future work are mentioned in Section 5.

## 2. RELATED WORK

In [3], methods and tools for producing location-based Mobile Urban Dramas are presented. In this kind of applications, users become the main characters in a play. The authors define a framework to create these dramas. In particular, the authors present six dramas with different purposes, for example, tourism and learning. Note that these dramas are created by developers.

Structuring location-aware narratives is described in [7]. The authors present three patterns characterised using [5] and, based on this, they provide a sculptural location-based hypertext model. This model defines the concept of location-query as an attribute of different classes. Until now, this approach is focused on modelling aspects of these kinds of applications and not to provide an end-user tool.

As it was previously mentioned, end-users can design location-aware applications in two ways, digital or physical [12]. The former uses a map as a reference to define the relevant locations and contents but users are not physically located there. The latter, requires the user to be physically located in the place in which the application will be used later. Concrete examples of both approaches are described next.

In [12], a digital design tool called QuesTInSitu is presented. This tool enables users to define learning activities in specific outdoor locations. The activities defined can be run (by a student) using the mobile web application's user interface. The tool provides a map that uses a web map service (e.g. Google Maps), which is used by teachers, to create in-situ activities in relevant geolocations. The concept of in-situ activities means that, when the students reach one of these particular geolocations (using their mobile device's GPS), they receive the corresponding activity that is automatically triggered by the tool.

Another digital design is presented in [10]. The authors describe the IVO platform, which allows end-users to build and deploy mobile context-aware applications. To support this kind of applications, IVO platform uses an event-driven model. As part of the IVO platform, the authors also provide the IVO Builder and the IVO Client (Android application). In [10] is shown a tourist guide application created by using this platform.

Regarding to the physical design, two approaches are presented. In [9] the author describes CASTOR, (Context Aware STORytelling) an authoring system oriented to children that enables them to create structured stories in-situ. This system supports structuring the content both with a sequential logic or using the concept of branching. This system uses the mobile device's GPS to retrieve a specific geolocation that is associated to different contents (text, pictures, etc.). Moreover, in each location students specify contextual conditions used to trigger the content when the final mobile application is running.

The other similar in-situ authoring tool is presented in [4]. This is an Android application called TOTEM.Scout that allows users to create location-based content (image, video, etc.). The authors provide the concept of Shapes that are templates users can use to define contents (associated to relevant locations). These templates provide, for example, the possibility to associate a GPS or NFC location with the content. In addition, the authors provide the TOTEM.Design (a web-based tool) that uses the created content to build location-aware experiences. Up to now, the authors only support Google Maps and they mentioned as the future work to add floor plants to their tool.

In conclusion, all these mentioned approaches are focused on the production of different kinds of outdoor mobile applications. Additionally, the contents defined on these approaches are coupled with a specific location (generally a GPS location). Taking into account the contribution of this paper, we present a tool to define indoor-outdoor location-aware experiences, which models contents and locations in a decoupled way (providing reusability of both contents and locations).

## 3. OUR APPROACH

In Figure 1, a general schema of our approach is shown. We provide a responsive web-based interface used to define location-aware experiences. Our framework is used to provide the functionality required to build an experience (Figure 1 shows that the framework returns a final mobile experience). Web Services are used to communicate both parts (interface and framework). Our interface is capable of communicating both with map services (e.g. for outdoor representations) and sensing mechanisms such as GPS.

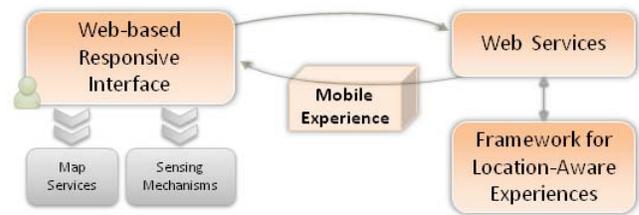


Figure 1. General schema of our approach.

Our approach defines general concepts used by our end-user tool and framework. We have identified two different layers (or concerns), one with contents (text, image, video, multiple choice question, etc.) and its relationships, and the other one that defines relevant locations (related with an indoor or outdoor space). Locations can be defined in different ways, such as in a symbolic or geometric way, as mentioned in [6]. The relationship between contents and locations define a point of interest (PoI). These are shown in the Figure 2.

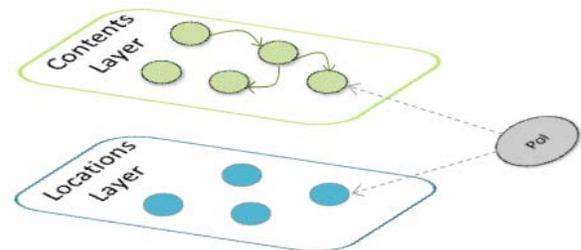


Figure 2. General concepts used in our approach.

Using the concepts defined in Figure 2, contents and locations can be part of different PoI, allowing reuse. This can be interesting and useful for different situations. For instance, if a museum used QR-codes associated to each object with our approach, the location layer could be reused to provide different location-aware experiences. Each experience could define different contents (related with their locations) with different purposes, such as learning activities or tour guides.

#### 4. AUTHORING TOOL

In this section, we present our visual tool to create location-aware experiences. This tool is defined by using a responsive web-based interface so it can be used in any web browser (mobile or desktop). Up to now, our tool provides two different ways of defining locations, using GPS or QR-code. But its flexible design supports the addition of any other location sensing mechanism. Our tool enables users to define:

- Contents Layers
- Locations Layers (relative to an indoor or outdoor space)
- PoI (a content within a location forms a point of interest)

In this paper, we only focus on the creation of PoI. To obtain the location, a user can configure the location sensing mechanism. By default, GPS is chosen but he can switch into QR-codes whenever he wants. In the case of GPS, the behaviour is similar to the related work presented in Section 2. If QR-codes are used, a new code is generated for each content (in a future approach of the tool, we will consider the possibility of reading existing QR-codes). Each code is related to an indoor-outdoor location inside the tool's map. In the final application, QR-codes are used to obtain the contents associated to them. Either using GPS or QR-codes we create a separation layer with these locations.

In the following sections, we show an example of the usage of our tool and some discussion points.

##### 4.1 Using our tool

A first version of our tool is presented in this section. The tool supports defining PoI in an indoor, outdoor or a mixed space (indoor-outdoor). The most interesting example is the latter, as it uses both indoor and outdoor spatial representations. In this first version, we have decided to use an image of the indoor space (the museum's plan) to represent the indoor space. Figure 3 shows how the image is overlaid over the outdoor map.

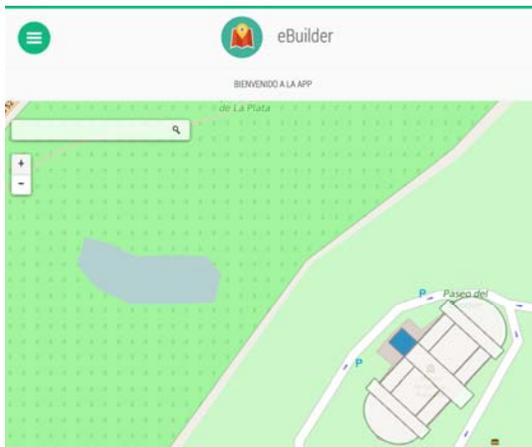


Figure 3. Indoor-Outdoor Map using our tool.

In Figure 4, the different types of content provided by our tool are shown, each one with one icon.



Figure 4. Icons representing contents in our tool.

Suppose a user is interested in creating a guided tour showing different mammal animals. In La Plata city (Argentina), the Zoo (outdoor) and the Natural Science Museum (indoor) are very close. The user wants to show some extinct animals (exhibited in the Museum) and some non-extinct animals (which are in the Zoo).

Using our tool the user can visit the Zoo and the Museum, and in each place define PoI. At the Zoo, the user takes advantage of the GPS sensing mechanism to create the different PoI. As he walks through the different places, he creates the different PoI and the markers representing each one of them are automatically placed on his/her current position.

Figure 5 shows some PoI created in the outdoor space (the zoo) and a linear relationship between them. For simplicity, assume that for each PoI only a text is defined.

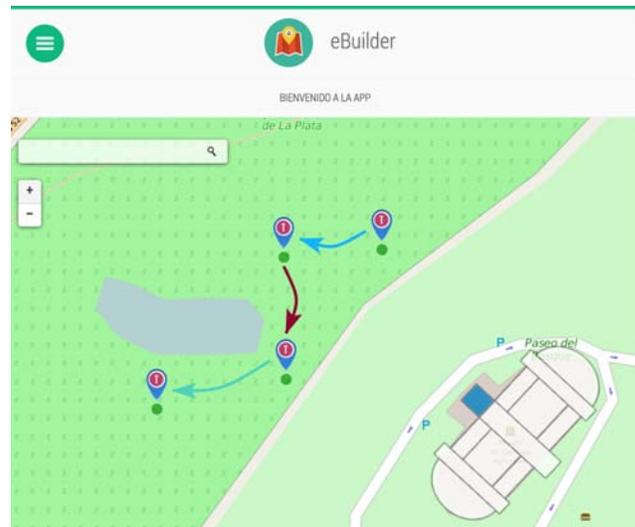


Figure 5. PoI defined in the outdoor map.

When the user is inside the Museum, he decides to switch to the QR-code sensing mechanism. As in this case there is no GPS coordinate representing the user's current position, the tool randomly places the PoI anywhere in the map. And this is why the user must manually drag the markers to the correct places on the tool's map. Every time the user creates a new PoI, a QR-code is generated as shown in Figure 6.

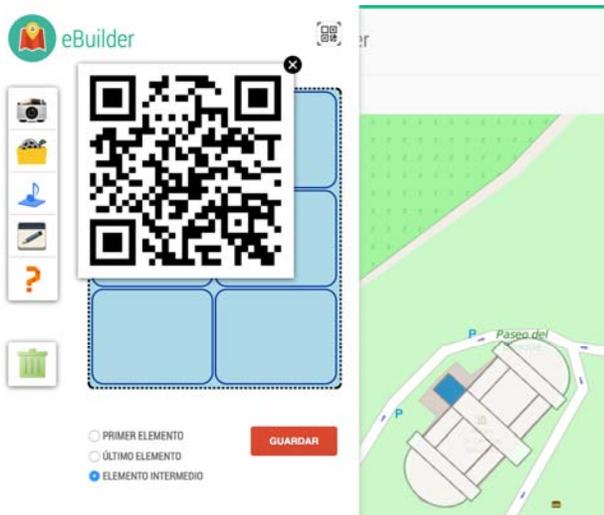


Figure 6. QR-code associated to a PoI.

Figure 7 shows all the PoI created by the user. Note that one of the outdoor PoI is connected with one of the indoor PoI.

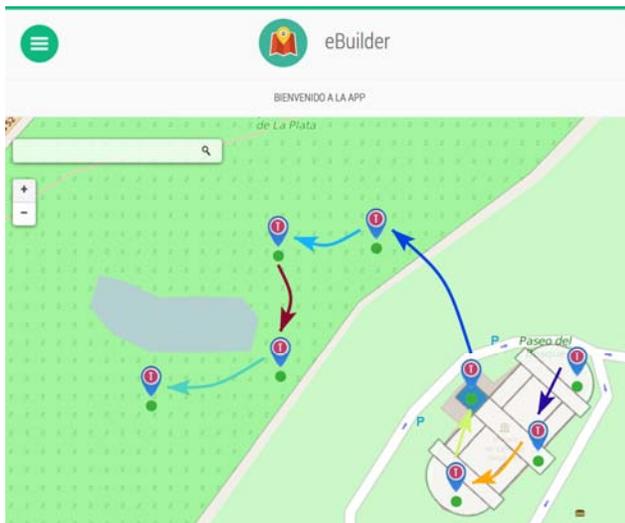


Figure 7. An indoor-outdoor tour defined with our tool.

As part of the final application, all QR-codes (if there are) are given to the user, who will be in charge of placing those codes in the correct physical locations. The framework builds the experiences as web applications with the possibility of using Phonegap [8] to create hybrid applications.

## 4.2 Discussions

Nowadays, most existing authoring tools do not provide any location sensing mechanism for indoor spaces. Consequently, not having this location assistance is the reason why creating mobile applications by end-users is more complex. The use of QR-codes is a first approach to define relevant locations. But this also depends on settling that code in the correct place.

The indoor representation is a topic by itself. In our first version of the tool, we only consider a referenced image to simplify it.

But we know that this is not scalable in order to build an indoor space with more than one floor. In these cases, the tool will should be defined with a more complex representation as shown in [14].

The complexity of the tool increases when context features are defined as part of each PoI, as presented in [9]. Up to now, the first version of our tool only allows defining simple contents such as: text, figure, audio, video, multiple-choice questions or a composite of them.

Another interesting point to mention is the layer's reusability. In this case, location layer's reusability is possible because they just define relevant locations in an indoor-outdoor space. But the reuse of contents is not always possible if it is location-dependent (as is presented in [13]). In other words, if the content mentions any characteristic of the environment, the reuse is limited.

It is important to note that, with our tool, it is possible to define the three structure patterns mentioned in [7]. These structures are dynamically defined according to the relationships that users define between the contents.

## 5. CONCLUSION AND FUTURE WORK

We have presented an approach to create location-aware experiences. As mentioned before, we have used the concept of separation of concerns to represent the content layer and the location layer. To relate content and location, we have used the concept of PoI. We have presented a first version of our tool, which creates location-aware applications in-situ. Our tool uses indoor-outdoor spaces to define PoI and relationships between them. These are shown in the example presented in this paper. Some discussion points have been presented to show that this is the starting point of an area that requires much more research and evaluation.

Up to the moment, the tool has only been evaluated by the developers involved in its definition. We know that it is crucial to evaluate it with end-users. To achieve this, we are currently organising an experiment to create a specific location-aware application that involves indoor-outdoor places to be evaluated by end-users.

We are also working on providing more complex indoor representations as it is defined in [14]. According to that, we have evaluated different kinds of interfaces to ease the definition of PoI when the indoor space has more than one floor.

As future work, the tool will support more sensing mechanisms such as NFC. Furthermore, the tool will define more complex contents. For example, users will be able to record videos or take pictures (as is described in [4]).

## 6. REFERENCES

- [1] Fortier, A., Challiol, C., Fernández, J. L., Robles, S., Rossi, G., and Gordillo, S. 2014. Exploiting personal web servers for mobile context-aware applications. *The Knowledge Engineering Review* 29, 2, (Mar. 2014), 134-153. DOI=<http://dx.doi.org/10.1017/S026988914000022>
- [2] Fortier, A., Rossi, G., Gordillo, S. E., and Challiol, C. 2010. Dealing with variability in context-aware mobile software. *Journal of Systems and Software* 83, 6, (June 2010) 915-936. DOI=<http://www.sciencedirect.com/science/article/pii/S0164121209002830>

- [3] Hansen, F. A., Kortbek, K. J., and Grønbaek, K. 2012. Mobile urban drama: interactive storytelling in real world environments. *New Review of Hypermedia and Multimedia* 18, 1-2, (Mar.-June 2012), 63-89. DOI=<http://www.tandfonline.com/doi/abs/10.1080/13614568.2012.617842>
- [4] Jurgelionis, A., Wetzel, R., Blum, L., and Oppermann, L. 2013. TOTEM.Scout: A mobile tool for in-situ creation of location-based content. In *Proceedings of Games Innovation Conference 2013* (Vancouver, BC, September 23-25, 2013). IGIC 2013. IEEE, 89-96.
- [5] Kjeldskov, J., and Paay, J. 2007. Augmenting the City with Fiction: Fictional Requirements for Mobile Guides. In *Proceedings of Mobile Interaction with the Real World 2007/5th Workshop on HCI in Mobile Guides* (Singapore, September 9, 2007). MIRW 2007. ACM, New York, NY, 1-6.
- [6] Leonhardt, U. 1998. *Supporting location-awareness in open distributed systems*. Doctoral dissertation, Imperial College.
- [7] Millard, D. E., Hargood, C., Jewell, M. O., and Weal, M. J. 2013. Canyons, deltas and plains: towards a unified sculptural model of location-based hypertext. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media* (Paris, France, May 01-03, 2013). Hypertext 2013. ACM, New York, NY, 109-118. DOI=<http://dl.acm.org/citation.cfm?doid=2481492.2481504>
- [8] Phonegap Home Page: <http://phonegap.com/>
- [9] Pittarello, F., and Bertani, L. 2012. CASTOR: learning to create context-sensitive and emotionally engaging narrations in-situ. In *Proceedings of the 11th International Conference on Interaction Design and Children* (Bremen, Germany, June 12-15, 2012). IDC 2012. ACM, New York, NY, 1-10. DOI=<http://dl.acm.org/citation.cfm?doid=2307096.2307098>
- [10] Realinho, V., Romão, T., Birra, F., & Dias, A. E. 2011. Building mobile context-aware applications for leisure and entertainment. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (Lisbon, Portugal, 8-11 November, 2011). ACE 2006. ACM, New York, NY, Article No. 29. DOI=<http://dl.acm.org/citation.cfm?doid=2071423.2071459>
- [11] Rogers, Y., Connelly, K., Tedesco, L., Hazlewood, W., Kurtz, A., Hall, R. E., and Toscos, T. 2007. Why it's worth the hassle: The value of in-situ studies when designing ubicomp. In *Proceeding of the 9th international conference on Ubiquitous Computing (Innsbruck, Austria, September 16-19, 2007)*. UbiComp 2007. Springer Berlin Heidelberg, 336-353.
- [12] Santos, P., Hernández-Leo, D., and Blat, J. 2014. To be or not to be in situ outdoors, and other implications for design and implementation, in geolocated mobile learning. *Pervasive and Mobile Computing* 14, (Oct. 2014), 17-30.
- [13] Weal, M. J., Hornecker, E., Cruickshank, D. G., Michaelides, D. T., Millard, D. E., Halloran, J., and Fitzpatrick, G. 2006. Requirements for in-situ authoring of location based experiences. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (Helsinki, Finland, September 12-15, 2006). Mobile HCI 2006. ACM, New York, NY, 121-128. DOI=<http://dl.acm.org/citation.cfm?doid=1152215.1152241>
- [14] Zlatanova, S., Liu, L., and Sithole, G. 2013. A conceptual framework of space subdivision for indoor navigation. In *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness* (Orlando, Florida, USA, November 5-8, 2013). ISA 2013. ACM, New York, NY, 37-41. DOI=<http://dl.acm.org/citation.cfm?doid=2533810.2533819>

# Combing Location-Aware Applications with in-situ Actors Performances

Federico M. Alconada Verzini

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

falconada@lifa.info.unlp.edu.ar

Juan I. Tonelli

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

jtonelli@lifa.info.unlp.edu.ar

Cecilia Challoi<sup>‡</sup>

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

ceciliac@lifa.info.unlp.edu.ar

Alejandra B. Lliteras

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

lliteras@lifa.info.unlp.edu.ar

Silvia E. Gordillo<sup>§</sup>

LIFIA, Fac. de Informática, UNLP.

Calle 50 y 120, La Plata,

Buenos Aires, Argentina.

+54 221 422-8252

gordillo@lifa.info.unlp.edu.ar

## ABSTRACT

In this paper we present a model to define location-aware experiences having its different aspects (contents, location, etc) decoupled one from another. We have created an application following this model, which is also combined with actors' performances. This application defines three relevant locations inside a specific building. By reading a QR-code the user indicates that he/she has arrived at one of those locations. In that moment, the application triggers some questions related to the actors' performances. Hence, the user is required to see all the plays in order to be able to answer such questions. The application assists the user in moving throughout the different locations. We present the evaluation of this application and then we discuss interesting aspects that have shown up as a result of the in-situ evaluation of this application.

## Categories and Subject Descriptors

D.2.13 [Software Engineering]: Reusable Software - *Reuse models*

H.5.2 [Information Interfaces and Presentation]: User Interfaces - *Prototyping*

**General Terms:** Design, Experimentation

**Keywords:** Location-Aware Applications, In-situ Actors Performance, In-situ evaluation, Prototype, Modelling Location-Aware Applications, Separation of Concerns, Mobile Computing

## 1. INTRODUCTION

Mobile Applications have been increasing in the last few years covering different domains such as education, entertainment,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

NHT'15, September 1, 2015, Guzelyurt, TRNC, Cyprus.

© 2015 ACM. ISBN 978-1-4503-3797-7/15/09...\$15.00DOI:

<http://dx.doi.org/10.1145/2804565.2804571>

tourist guides, etc. In particular, those applications that reference some relevant aspects of the environment (e.g. location-aware application) have only been tested in an in-situ fashion. Different authors ([2], [3], [14] and [16]) have come to the conclusion that evaluating prototypes in-situ is essential. In [12], the authors reinforce this idea by explaining that the way in which a person perceives a physical space can change according to the time of the day (morning, afternoon, evening), for instance.

There are different ways of creating mobile applications. In particular, location-aware applications can be defined by developers (e.g. [8], [10]) or by end-user ([13] and [15]). In both cases, according to what was mentioned before, the final mobile application is required to be tested in-situ to obtain real evaluation results.

Mobile Urban Dramas are a particular type of mobile application introduced in [7] and defined as Location-Aware Narratives. In each location defined within a narrative, the user receives a multimedia content, such as an audio or an image, and then he/she continues to the next location. Recent works of these authors present six different dramas [8], which are defined with specific domains (e.g. education or entertainment). Inspired by these dramas we came up with the idea of creating a location-aware application combined with plays performed by real actors. For each relevant location, the application shows a specific content and several actors perform a play. The same play can be performed in different locations (the content received by the user remains the same but location and physical space change). For a rapid prototype of this kind of applications, it is important to represent the contents independently of each location.

We have been working on different modelling aspects of mobile applications ([5] and [6]). In particular, we have used the concept of separation of concerns to handle them in an independent way. For instance, having content and location concerns decoupled one another. This experiences that combine location-aware applications with actors' performances are faster to prototype if we use the concept of separation of concerns.

In this paper, we present a model to define location-aware experiences where content aspects are decoupled from location

ones. To test our model, we create a location-aware application prototype. We describe an in-situ evaluation of this application, which combines location-aware content with actors' performances. We present various results and discussions points of this evaluation.

This paper is organised as follows. In Section 2 we describe some related work. A modelling approach for location-aware experiences is presented in Section 3. In Section 4, we describe a prototype which was created by using our model. An evaluation of this prototype is presented in Section 5. Conclusions and future work are mentioned in Section 6.

## 2. RELATED WORK

In [4], the authors analyse different kinds of Mobile Guides and give the taxonomy of relevant contexts in these kinds of applications, such as the location, that is one of the most useful contexts. The authors describe different location techniques and environment types. In [3], the authors mention the importance of evaluating prototypes in an in-situ way.

An event-driven model is used in [13] to create mobile context aware-applications. The authors build a mobile tourist guide using IVO platform (retrieving locations using GPS), which is tested in-situ.

In [10], the authors present a sculptural location-based hypertext model. This model is defined through a characterisation of three patterns (Canyons, Deltas and Plains) that authors have detected in this kind of applications. This model defines the concept of location-query as an attribute of different classes. To test their model, the authors have created GeoYarn, a web service application for Android implemented with this model which provides stories in different locations. Motivated by this idea, we have tested our model in the same way.

Six Mobile Urban Dramas are described in [8]. The authors of this paper present a conceptual framework that includes two specific models (user and environment) to define dramas whose plot is described by means of graphs with branches. The framework detects the user's locations through 2D barcodes, RFID or GPS and, when the user reaches a specific location, he/she receives a multimedia content (audios, images, animations or videos). Each drama is deployed as a mobile application both in iOS and Android platforms.

The authors in [8], [10] and [13] use different models to represent the content and the relevant locations. But in all of them, these concepts are coupled. We use these approaches as an inspiration and as a starting point to define a new model, where the content and location aspects are decoupled.

## 3. MODELLING LOCATION-AWARE EXPERIENCES

In Figure 1, we present a general schema of our model that allows representing location-aware experiences. The abstract class *Content* represents a general content. For simplicity we have decided not to show each concrete subclass (as Text, Image, Video, etc). We have decoupled the content from the structure that contains it. To do that, we have created the concept of *Element*. This enables us to reuse a content more than once in different structures (or patterns as mentioned in [10]). The location is defined as an *Interface* (following the idea of [9] in

which a location can be represented, for example, in a symbolic or geometric way). The class *PoI* acts as a weaving between elements and locations. Note that the same content or position could be used by different *PoI*.

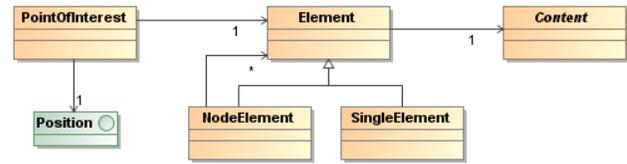


Figure 1. Decoupled contents, structures and locations.

Other relevant classes are required to be able to define the experience itself. To achieve that, we have created the class *MobileExperience* that contains all the relevant *PoI*. To show how we achieve the decoupling of some relevant aspects, we point out here the classes *SensingManager* and *SpaceRepresentationManager*. These classes are shown in Figure 2. This is a simplified model which aims at presenting the general idea.

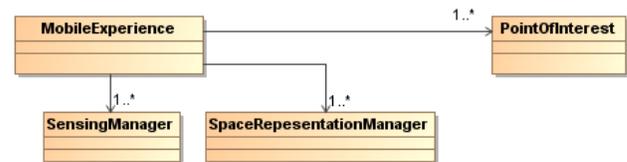


Figure 2. Simplified model to define location-aware experiences.

## 4. OUR PROTOTYPE

In 2014, we were invited by one of the local TED talks (TEDxDiagonales73) to create a mobile application to be used among the participants. Using the model depicted in Section 3, we decided to create a location-aware application prototype called CaminosAlternativos (Alternative Paths). We defined the indoor space by drawing a plan of the building, including some relevant points such as the elevator and the stairs.

When we first thought about this prototype, we agreed to make a web application and hosted it in a server, so that the participants could access it through the Internet. Later, when we tested the WiFi connection of the building where the TED talk was going to be held, we found out that it was not working properly. Therefore, we decided that the fewer HTTP connections the prototype required, the better performance it would be able to achieve. This is why we decided to use the framework Phonegap [11] to create a hybrid mobile web application for Android. In this way, users would install the mobile application in their phones and the prototype would only have one HTTP connection to send the result generated by each user.

In order to create a more interesting application we thought about combining this application with acting performances in each location. To do so, we contacted Lorena Velazquez, headmaster of the ECAE School of Performing Arts, who helped us to define each acting performance and coordinate the actors' participation. Lorena proposed us doing an experience in which three performances would be defined, each one based on a popular TED talk phrase taken from previous TED events (these phrases were selected by the organisers of the TEDxDiagonales73 talk). A

user would see the plays and answer, in one word, what has he/she felt while seeing it. Once the user finishes seeing the three plays, he/she would have to create a phrase trying to use the three words he/she has answered. The user with the most creative phrase would be the winner of a prize at the end of the TEDxDiagonales73 talk.

To achieve this, we visited the building (in which the talk was going to be held) and defined three relevant locations (corresponding to three rooms or scenes), two on the first floor and the other one on the second floor. The way to get around the three locations was without a predetermined order. The first QR-code selected by the user determined the first play he/she would see, which is carried by an actor as shown in Figure 3 (so, there are three actors, one for each scene). Even though this first-scene selection could have been achieved by a technical solution, we decided to do it in this way in order to make a more exciting experience.



**Figure 3. Deciding the first place.**

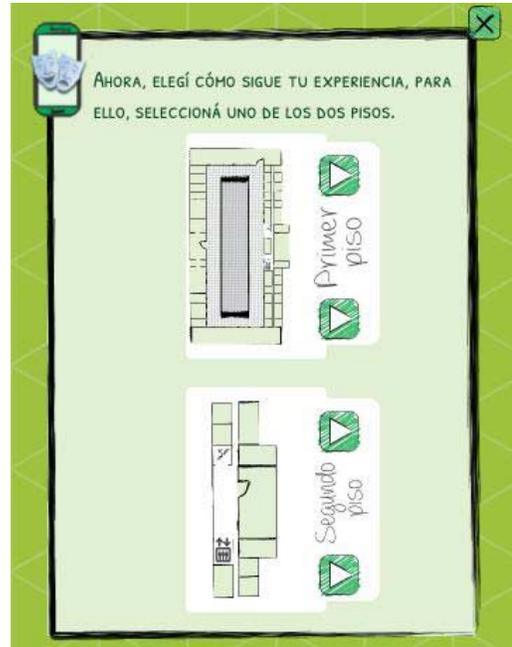
After reading the first QR-code (Figure 3), the application shows a map to the user indicating the path to reach the specific room where one of the performances will take place. The user walks to that room and finds another QR-code, as shown in Figure 4. When the user reads this code a question appears and the user is intended to answer a word regarding to his/her feelings about the performance.



**Figure 4. QR-code of a scene.**

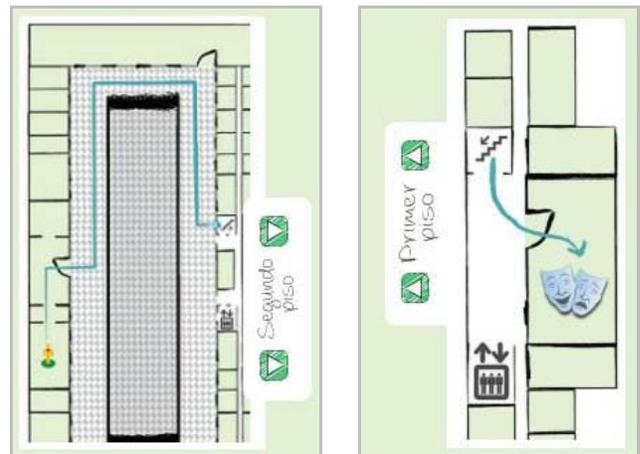
Once the play is over (all of them last 5 minutes) and the user has answered the question, the application shows the map of the building with the other two scenes that the user still has to see. If the two remaining plays are the ones on the first floor, a map only

of this floor is displayed to the user. Otherwise, the user has to choose which floor he/she wishes to go next (see Figure 5).



**Figure 5. Two options to follow with the experience.**

After choosing one of these options (Figure 5), the user receives a map with the path to its destination. If the path involves moving to the other floor, he/she receives the relevant instructions through the maps shown in Figure 6. The map on the left indicates the path to the stairs. Finally, the map on the right assists the user to reach the correct room.



**Figure 6. Instruction to move from one floor to the other.**

The same process which took place in the first scene is repeated. Since the application was defined with only three scenes, the path to the last scene is automatically provided when the second question is answered. When the user completes the last question, he/she receives a final question to generate a phrase about how the performances have made him/her feel.

Note that the pictures shown in Figure 3 and 4 are taken during the in-situ evaluation with TEDxDiagonales73 participants.

## 5. EVALUATION

In this section we present the in-situ evaluation of the prototype described in Section 4. The participants of TEDxDiagonales73 that used the application were 34 (ages ranged from 20 to 47). All participants used their own mobile devices to install the prototype.

### 5.1 Analysis and Design

We created a document to describe general characteristics of the prototype and what the installation process in their mobile devices was like. This information was available one day before the TEDxDiagonales73 talk. In this way, the user had the chance to download it and have it installed by the time of the talk.

We made identification cards with a QR-code and gave them to the users when the experience started. To start running the prototype users were required to read the code of their cards so as to be able to receive the map to go to the place where the three actors were. This identification step was made both to prevent users from running the prototype before the time the evaluation began as well as to create more expectations. Figure 7 shows users reading their identification codes.



Figure 7. QR-code for identification.

We defined a SUS questionnaire [1] to evaluate the usability of the system. In addition, we defined a general form with two questions:

- Mark the grade of complexity on the use of the application (scores 1 to 5, 1 means low difficulty and 5 high difficulty)
- The assistance provided by the application was enough to move from one place to another (scores 1 to 5, 1 means strongly disagree and 5 strongly agree)

These forms were completed by the participants at the end of the experience.

### 5.2 Result and Discussions

The 34 participants completed both forms mentioned in the previous section. The result of the SUS was 76,0 (over 100), which is an acceptable score for a prototype and considering that the indoor space was represented only with and static image that limited the usability of the user.

The result of the general form is described below. The average score for the question “Mark the grade of complexity on the use of

the application” was 2,14 (SD=1,23). This means that the complexity to use the prototype was low. The other question “The assistance provided by the application was enough to move from place to another” had an average score of 3,79 (SD=1,27). According with these values the assistance is more open to discussion because it was useful to most of the users but for others it was not. This point requires further improvement in future prototypes.

In this section we discuss some interesting points. To prevent all the users from going to the same play, each actor (of the three initial actors shown in Figure 3) counted the number of participants that were reading his QR-code. When this number reached the number of 12 users, the actor hid the code so that no other user could read it. This tactic was thought to balance the number of users in each room (or scene) preventing each room from getting overcrowded.

Another issue came up at the time of choosing the second play to see. Participants that were on the first floor (where two performances were being held), preferred their next choice to be the one of the first floor. In this situation, there were fewer people on the second floor than on the first one, where most participants were.

All three performances required a clockwork synchronisation to start simultaneously. To solve this, in each scene there were assistants who saw to this and decided when to start each performance, so as to do it all at the same time. The assistants’ mission was also guaranteeing that there were no participants in the corridor or stairs.

The QR-code identifications were general, so there was no need to give any personal information. This was an advantage due to the fact that being anonymous increased the chances that users would answer the questions.

Both the content and the structure can be reused as the application was built following the model presented in the Section 3. To reuse them, it will only be necessary to define a new physical space to be able to test this prototype.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a model to location-aware experiences. We described the main classes and focused on providing decoupled aspects of these kinds of applications, such as contents, structure and locations.

We have described a novel prototype, which was combined with actors’ performances. We have presented and analysed the result of the evaluation in-situ. The discussion points mentioned in this paper are part of the learning process in evaluation in-situ. This type of evaluations (with real end-user people) are the best way to understand how these kind of applications behave.

We have presented the evaluation of this application and then we discussed some interesting aspects that were learnt as part of the in-situ evaluation.

We are working to enrich our model and create more complex applications. One of the aspects to improve is the map the application provides. In indoor spaces it is essential to be precise in the assistance to ensure that the user arrives at his/her destination as easily and straightforward as possible. Moreover, based on this model, we are currently developing a tool that enables end-users to create these location-aware experiences.

As a future work, we will evaluate this model with other prototypes to consider other relevant aspects. For example, the collaboration between users to create contents.

## 7. ACKNOWLEDGMENTS

We extend our thanks to Lorena Velazquez for creating and preparing each acting performance, and to the actors of ECAE (*Espacio Creativo de las Artes Escénicas*). The authors thank to Agustina Zimbello and Leandro Vilas for their collaboration on the organisation of this experience.

## 8. REFERENCES

- [1] Brooke, J. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189,194, 4-7.
- [2] Consolvo, S., Harrison, B., Smith, I., Chen, M. Y., Everitt, K., Froehlich, J., and Landay, J. A. 2007. Conducting in situ evaluations for and with ubiquitous computing technologies. *International Journal of Human-Computer Interaction* 22, 1-2, (Dec. 2007), 103-118.
- [3] Crabtree, A., Chamberlain, A., Davies, M., Glover, K., Reeves, S., Rodden, T., and Jones, M. 2013. Doing innovation in the wild. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI* (Trento, Italy, September 16-19, 2013). CHI Italy '13. ACM, New York, NY, Article No. 25. DOI=<http://dl.acm.org/citation.cfm?doid=2499149.2499150>
- [4] Emmanouilidis, C., Koutsiamanis, R. A., and Tasidou, A. 2013. Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications* 36, 1, (Jan. 2013), 103-125.
- [5] Fortier, A., Challiol, C., Fernández, J. L., Robles, S., Rossi, G., and Gordillo, S. 2014. Exploiting personal web servers for mobile context-aware applications. *The Knowledge Engineering Review* 29, 2, (Mar. 2014), 134-153. DOI=<http://dx.doi.org/10.1017/S0269888914000022>
- [6] Fortier, A., Rossi, G., Gordillo, S. E., and Challiol, C. 2010. Dealing with variability in context-aware mobile software. *Journal of Systems and Software* 83, 6, (June 2010) 915-936. DOI=<http://www.sciencedirect.com/science/article/pii/S0164121209002830>
- [7] Hansen, F. A., Kortbek, K. J., and Grønbaek, K. 2008. Mobile Urban Drama—Setting the Stage with Location Based Technologies. In *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling* (Erfurt, Germany, November 26-29, 2008). ICIDS '08. Springer-Verlag Berlin, Heidelberg, 20-31.
- [8] Hansen, F. A., Kortbek, K. J., and Grønbaek, K. 2012. Mobile urban drama: interactive storytelling in real world environments. *New Review of Hypermedia and Multimedia* 18, 1-2, (Mar.-June 2012) 63-89. DOI=<http://www.tandfonline.com/doi/abs/10.1080/13614568.2012.617842>
- [9] Leonhardt, U. 1998. *Supporting location-awareness in open distributed systems*. Doctoral dissertation, Imperial College.
- [10] Millard, D. E., Hargood, C., Jewell, M. O., and Weal, M. J. 2013. Canyons, deltas and plains: towards a unified sculptural model of location-based hypertext. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media* (Paris, France, May 01-03, 2013). Hypertext 2013. ACM, New York, NY, 109-118. DOI=<http://dl.acm.org/citation.cfm?doid=2481492.2481504>
- [11] Phoneygap Home Page: <http://phoneygap.com/>
- [12] Pittarello, F. 2011. Designing a context-aware architecture for emotionally engaging mobile storytelling. In *Proceeding Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction* (Lisbon, Portugal, September 5-9, 2011). INTERACT '11. Springer-Verlag Berlin, Heidelberg, 144-151.
- [13] Realinho, V., Romão, T., Birra, F., and Dias, A. E. 2011. Building mobile context-aware applications for leisure and entertainment. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (Lisbon, Portugal, 8-11 November, 2011). ACE 2006. ACM, New York, NY, Article No. 29. DOI=<http://dl.acm.org/citation.cfm?doid=2071423.2071459>
- [14] Rogers, Y., Connelly, K., Tedesco, L., Hazlewood, W., Kurtz, A., Hall, R. E., and Toscos, T. 2007. Why it's worth the hassle: The value of in-situ studies when designing ubicomp. In *Proceeding of the 9th international conference on Ubiquitous Computing* (Innsbruck, Austria, September 16-19, 2007). UbiComp 2007. Springer Berlin Heidelberg, 336-353.
- [15] Santos, P., Hernández-Leo, D., and Blat, J. 2014. To be or not to be in situ outdoors, and other implications for design and implementation, in geolocated mobile learning. *Pervasive and Mobile Computing* 14, (October, 2014), 17-30.
- [16] Weal, M. J., Hornecker, E., Cruickshank, D. G., Michaelides, D. T., Millard, D. E., Halloran, J., and Fitzpatrick, G. 2006. Requirements for in-situ authoring of location based experiences. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (Helsinki, Finland, September 12-15, 2006). Mobile HCI 2006. ACM, New York, NY, 121-128. DOI=<http://dl.acm.org/citation.cfm?doid=1152215.1152241>

---

‡ Another affiliation for Cecilia Challiol: CONICET, Argentina.

§ Another affiliation for Silvia E. Gordillo: CIC, Buenos Aires, Argentina.