



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Herramienta para facilitar la creación de actividades educativas posicionadas

AUTORES: Zimbello, Agustina Micaela

DIRECTOR: Dra. Cecilia Challiol

CODIRECTOR: Dra. Silvia Gordillo

ASESOR PROFESIONAL: Mg. Alejandra Beatriz Lliteras

CARRERA: Licenciatura en Sistemas

Resumen

En el dominio educativo las aplicaciones móviles se caracterizan, entre otras cosas, por ser usadas por los docentes como una herramienta adicional en el proceso de aprendizaje de los alumnos, se denominan *Aplicaciones Móviles Educativas*; dentro de estas se encuentran las *Aplicaciones Móviles Educativas basadas en posicionamiento*, donde brindan actividades educativas posicionadas a los alumnos. En este marco, la principal motivación de esta tesis es proveer a los docentes de una herramienta que les facilite la creación de *Actividades Educativas Posicionadas* para ser empaquetadas luego en aplicaciones móviles.

Palabras Clave

Herramienta de Autor; Actividades Educativas Posicionadas; Aplicaciones Educativas Móviles basadas en Posicionamiento; PhoneGap.

Trabajos Realizados

Se realizó una conceptualización de las características que debería cumplir una herramienta para permitir la creación de elementos educativos posicionados. Esta caracterización se realizó en base al análisis bibliográfico sobre la temática. Tomando como base la caracterización realizada, se desarrolló una herramienta para facilitar la creación de Actividades Educativas Posicionadas. Este prototipo permite definir que los elementos educativos posicionados junto con sus plantillas representativas, tanto en espacios indoor como outdoor. Dicho prototipo se realizó utilizando PhoneGap.

Conclusiones

Se presentó una herramienta para la facilitar la creación de Actividades Educativas Posicionadas que permite, a usuarios no expertos con especial foco a los docentes, a definir recorridos que conlleven un aprendizaje tanto en espacios indoor como outdoor, brindándole la posibilidad de no solo construir el orden en que se darán las tareas, sino también sus posibles plantillas representativas, de manera tal que el docente pueda definir tanto el contenido de las tareas como así también las indicaciones para llegar y realizar las mismas.

Trabajos Futuros

Proveer más variabilidad de plantillas relacionadas a diferentes tipos de tareas que podrían hacer los alumnos en cada posición.

Generar aplicaciones móviles empaquetadas que contengan una Actividad Educativa Posicionada.

Contar con un repositorio de capas de contenidos, posiciones o ya contenidos posicionados, esto facilitaría que la herramienta las tome, y se pueda arrancar a usar esto como base de la creación.

Fecha de la presentación: Agosto, 2019

Índice

1. Introducción.....	4
1.1 Motivación	4
1.2 Objetivo	5
1.3 Estructura de la tesina	6
2. Background	7
2.1 Herramientas de armado de maquetas visuales	7
2.2 Herramientas de construcción de aplicaciones móviles.....	11
2.3 Herramientas de creación de aplicaciones móviles con fines culturales para usuario finales	16
2.4 Herramienta para la creación de elementos posicionados	22
2.5 Análisis de las herramientas presentadas.....	27
3. Análisis de las características de las Actividades Educativas Posicionadas que involucran elementos concretos	30
4. Herramienta propuesta	39
4.1 Problemas y soluciones relacionadas a la extensión propuesta	40
4.2 Funcionalidad de la herramienta propuesta en esta tesina.....	44
<input type="checkbox"/> Barra Superior de acciones.....	45
<input type="checkbox"/> Selección de mapas.....	47
<input type="checkbox"/> Creación de Posiciones.....	48
<input type="checkbox"/> Visualización y Creación de Templates.....	55
<input type="checkbox"/> Creación de Contenidos Educativos	62
<input type="checkbox"/> Creación de Elementos Educativos Posicionados.....	63
4.3 Exportar las creaciones realizadas.....	68
<input type="checkbox"/> Exportar <i>Posiciones</i>	69
<input type="checkbox"/> Exportar <i>Contenidos Educativos</i>	70
<input type="checkbox"/> Exportar <i>Elementos Educativos Posicionados</i>	71
5. Ejemplo de uso de la Herramienta propuesta	72
6. Publicaciones realizadas relacionadas con la tesina	83
7. Conclusiones y Trabajos Futuros.....	85
Bibliografía.....	88
Anexo A: Relevamiento de otras herramientas de maquetado	90

A mi abuela Sara, a mi mamá, a Carolina y a Martín, por el apoyo, la motivación y por estar siempre presentes

Al laboratorio LIFIA, por el espacio y las herramientas brindadas. A mi directora de tesis, Cecilia Challiol por todo lo que me ha enseñado a lo largo de mi carrera, por su confianza, paciencia, y dedicación.

1. Introducción

1.1 Motivación

El avance de la tecnología ha traído en consecuencia la creciente y variada gama de dispositivos y junto a ellos la aparición de las aplicaciones móviles. Las cuales han ido evolucionando tanto en aspectos visuales/gráficos como así también en la complejidad de las funcionalidades. Esto ha permitido su incorporación como parte del uso cotidiano.

Las aplicaciones móviles, son creadas con distintas finalidades de uso: financieras, culturales, salud, divertimento, comerciales, incluso educativas (como se describe en [Hyo-Jeong et al., 2008]), entre otras. Y muchas de estas, utilizan las características del ambiente (contexto) para brindar su funcionalidad. En particular, para el foco de esta tesina son de interés aquellas aplicaciones móviles basadas en posicionamiento. Este tipo de aplicaciones móviles brindan al usuario, en determinadas posiciones de interés información (o contenido) acorde al dominio de dicha aplicación. Estas aplicaciones pueden tener como finalidad proveer servicios tanto en espacios indoor (es decir, espacios cerrados) como outdoor (espacios abiertos), o en una combinación de ambos. Según lo mencionado por [Lliteras et al., 2011] y [Emmanouilidis et al., 2013], 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 (Pol) cercano a esa ubicación. Es decir, para poder brindar al usuario asistencia en la movilidad, las aplicaciones móviles basadas en posicionamiento deben contar una forma de representación de dichos espacios para poder guiar al usuario mientras recorre los mismos.

Las aplicaciones móviles basadas en posicionamiento pueden estar definidas (de acuerdo a su naturaleza) para que el usuario recorra, las posiciones relevantes, de diferentes maneras, como lo puede ser libremente o siguiendo un recorrido específico (como se especifica en [Millard et al., 2013] para hipertextos basados en posicionamiento).

En el dominio educativo las aplicaciones móviles se caracterizan, entre otras cosas, por ser usadas por los docentes como una herramienta adicional en el proceso de aprendizaje de los alumnos [Johnson et al., 2011]. Estas aplicaciones, se denominan Aplicaciones Móviles Educativas, y permiten al alumno poder aprender en cualquier momento y en cualquier lugar. Dentro de estas se encuentran las Aplicaciones Móviles Educativas basadas en posicionamiento, donde brindan actividades educativas posicionadas a los alumnos.

Acorde a lo definido en [Lliteras, 2015], las Actividades Educativas Posicionadas son actividades educativas que están formadas por tareas (que poseen consignas) y estas tareas pueden estar relacionadas entre sí. Estas actividades educativas se desarrollan en un determinado espacio físico, y dichas tareas están posicionadas dentro del mismo. En [Lliteras, 2015] se propone un enfoque de modelado de Actividades Educativas Posicionadas que involucran elementos concretos en el cual se adopta la separación en capas, en particular de contenido y posicionamiento como se propone en [Lliteras et al., 2012]. Esta separación en capas facilita el reuso tanto del contenido como de las

posiciones. Este enfoque de modelado no cuenta aún con una herramienta que facilite la creación de este tipo de aplicaciones, esta es la motivación principal de esta tesina.

Actualmente existen herramientas que permiten la creación de aplicaciones educativas en espacios outdoor, por ejemplo, CASTOR [Pittarello & Bertani, 2012]. Esta aplicación permite construir narraciones in-situ (en el lugar) a chicos de siete a nueve años, permitiendo considerar características del ambiente como parte de la construcción. Otra herramienta es presentada en [Santos et. al, 2011] donde se permite crear preguntas posicionadas en un mapa para que luego los alumnos respondan in-situ las mismas. Sin embargo, estas herramientas están pensadas para espacios outdoors, y no presentan ninguna separación en capas entre el contenido y el posicionamiento. Es decir, no hay reuso de la información definida.

En [Alconada Verzini, 2015] se define una herramienta para creación de elementos posicionados tanto en espacios indoors como outdoors. La definición de los elementos posicionados se puede realizar tanto desktop como in situ (desde un dispositivo móvil) permitiendo a través de ésta última opción considerar características del ambiente. Cabe destacar que esta herramienta fue definida considerando la separación en capas mencionada en [Lliteras et al., 2012]. Sin embargo, los elementos que se definen con la herramienta definida en [Alconada Verzini, 2015] son: textos, imágenes, videos, etc. Por ende, no cuentan con las características de las actividades y tareas educativas definidas en el enfoque de modelado presentado en [Lliteras, 2015]. Este es un aspecto a incorporar en la tesina propuesta. Por otro lado, en [Alconada Verzini, 2015] el usuario no define ningún tipo de asistencia en la movilidad entre los elementos posicionados, este será otro aspecto a incorporar en esta tesina.

Acorde a esto, se tomará como base la herramienta definida en [Alconada Verzini, 2015] y se la extenderá para poder facilitar la creación de Actividades Educativas Posicionadas acorde a los conceptos propuestos en [Lliteras, 2015]. Logrando así contar con una herramienta destinada a usuarios no expertos que facilite la creación de este tipo de aplicaciones. En particular, considerando elementos concretos como parte de las actividades.

1.2 Objetivo

Como se mencionó anteriormente, en [Lliteras, 2015] se propone un enfoque de modelado de Actividades Educativas Posicionadas que involucran elementos concretos. Sin embargo, no se provee una herramienta que facilite la creación de este tipo de aplicaciones.

El objetivo principal de esta tesina es brindar una herramienta destinada a usuarios no expertos que facilite la creación de Actividades Educativas Posicionadas, acorde a los conceptos definidos en [Lliteras, 2015]. Esto implica, por ejemplo, definir con la herramienta el espacio físico donde se llevará a cabo una determinada actividad educativa como así también las posiciones de las tareas relacionadas a la misma. Además, permitir relacionar las tareas entre sí.

En [Lliteras, 2015] se describe el prototipo “*Aprendo Jugando*”, el cual presenta una *Actividad Educativa Posicionada* que involucran elementos concretos. Este prototipo es analizado para así poder detectar los componentes específicos que debe proveer la herramienta propuesta en esta

tesina. Este análisis permite poder definir aspectos generales que debería tener cualquier herramienta destinada a la creación de *Actividad Educativa Posicionada* que involucran elementos concretos.

Se utilizó como base la herramienta presentada en [Alconada Verzini, 2015] y [Alconada Verzini et al., 2015], la cual permite definir elementos posicionados. De esta herramienta se toma el concepto de posicionar elementos en el espacio físico pero se incorpora para esta tesina conceptos propios de actividades educativas, como así también la posibilidad de definir algún tipo de asistencia en la movilidad (algo que no contempla la herramienta presentada en [Alconada Verzini, 2015]). Es decir, se analiza de qué forma se puede brindar algún tipo de asistencia en las actividades generadas, por ejemplo, definir mapas para llegar de una posición a otra.

La herramienta definida en [Alconada Verzini, 2015] exporta a XML los datos definidos por los usuarios. Usando como base esto, en esta tesina se analizará que otra información se deberá generar en pos de facilitar la creación de Actividades Educativas Posicionadas.

1.3 Estructura de la tesina

En el Capítulo 2 se presenta un relevamiento de distintas herramientas algunas que permiten el maquetado visual, otras facilitan la construcción de aplicaciones móviles, se detallan algunas con fines culturales. Y en particular se describe la herramienta de creación de elementos posicionados presentada en [Alconada Verzini, 2015] y que se tomará de base en esta tesina.

En el Capítulo 3 se describe primero la problemática que se desea resolver, y luego se especifican los conceptos que deben ser considerados como parte de una herramienta que brinde soporte a la creación de Actividades Educativas Posicionadas que involucran elementos concretos.

En el Capítulo 4 se presenta la herramienta propuesta, detallando los aspectos que fueron extendidos a partir de la herramienta definida en [Alconada Verzini, 2015]. También se describen aquellos problemas que fueron surgiendo a lo largo de la extensión de la herramienta.

El uso de la herramienta propuesta es presentado en el Capítulo 5, donde se muestra como se crean los conceptos relacionados a las Actividades Educativas Posicionadas que involucran elementos concretos.

Las publicaciones realizadas en base a la presente tesina se describen en el Capítulo 6.

Las conclusiones son presentadas en el Capítulo 7, donde además se describen algunos trabajos futuros que se desprenden de la tesina desarrollada.

2. Background

En este capítulo se realizará un relevamiento de distintas herramientas que permiten crear o prototipar aplicaciones móviles, dicho relevamiento se focalizará en aquellas herramientas que existen actualmente tanto para la creación de distintos recursos visuales como así también aquellas que presentan la idea de posicionamiento entre sus componentes.

Cabe mencionar que el uso del término dispositivos móviles, se utilizara a lo largo del capítulo para referirse tanto a celulares como tabletas. El relevamiento será presentado a continuación en cuatro subsecciones, primero se presentarán distintas herramientas tanto de maquetado como de prototipado, luego se describirán algunas herramientas que están focalizadas en crear aplicaciones móviles con fines culturales, y por último se presentará una herramienta para la creación de elementos posicionados.

2.1 Herramientas de armado de maquetas visuales

En esta sección se realizará un relevamiento de aquellas herramientas que permiten la realización de prototipos de interfaces (del estilo mockups), es decir, maquetas. Algunas de estas herramientas pueden ser usadas tanto para crear aplicaciones web o desktop como así también para móviles, haciendo foco principalmente en estas últimas.

A la hora de destacar los aspectos más relevantes de cada herramienta, se utiliza siempre el mismo ejemplo, en el cual se utilizan las componentes del tipo: botón, input, texto y, de existir, del tipo mapa. También se presentarán en los ejemplos, en caso de que la herramienta lo posea, una característica particular que la diferencie de las otras. Esto permitirá ver aspectos comunes de las herramientas, como así también detectar las particularidades de cada una.

A la hora de realizar el relevamiento se encontraron varias herramientas de maquetado se tomó la decisión de mencionar en esta sección solo algunas herramientas representativas, el resto de las herramientas relevadas se mencionan y detallan brevemente en el Anexo A. Estas herramientas que se describen en el anexo mencionado no poseen una componente del tipo mapa, o bien, si la poseen esta presenta características semejantes a las que se relevaran en detalle esta sección.

Usando diferentes capturas de pantallas se irá presentando las distintas herramientas relevadas. Cabe destacar que solo se hará hincapié en aquellos aspectos relevantes para el foco de la tesina.

- **Fluidui**¹

Es una aplicación web que permite la creación de interfaces para dispositivos móviles y desktop. Se puede observar en la Figura 2.1 la interfaz de la herramienta *Fluidui* y su uso mediante un ejemplo. En este caso particular la figura muestra la creación de un texto junto con un input y un botón.

¹ Página de *Fluidui*: <https://www.fluidui.com> (Último acceso 14-10-2015)

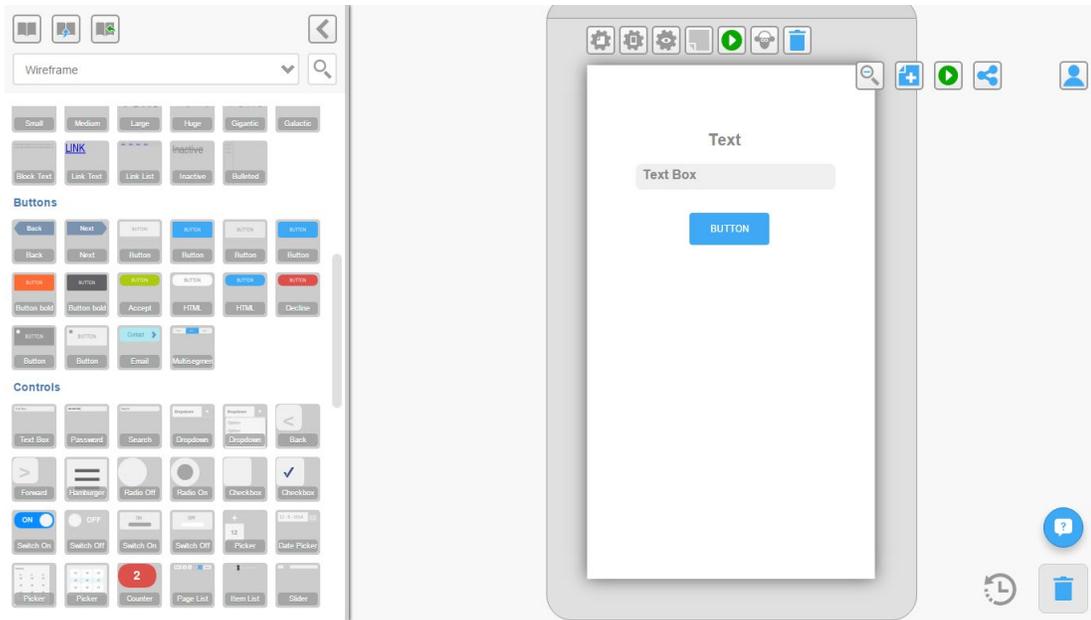


Figura 2.1: Herramienta Web de Prototipado FluidUI.

Fluidui cuenta con una componente mapa denominada “map tag”, esta representa un indicador de un punto de interés (POI) en un mapa, cuando se agrega a la maqueta, se está dejando un lugar en la interfaz para que se considere que debe mostrarse un mapa en su posterior refinamiento. En la Figura 2.2 se puede observar un ejemplo de utilización de este componente.

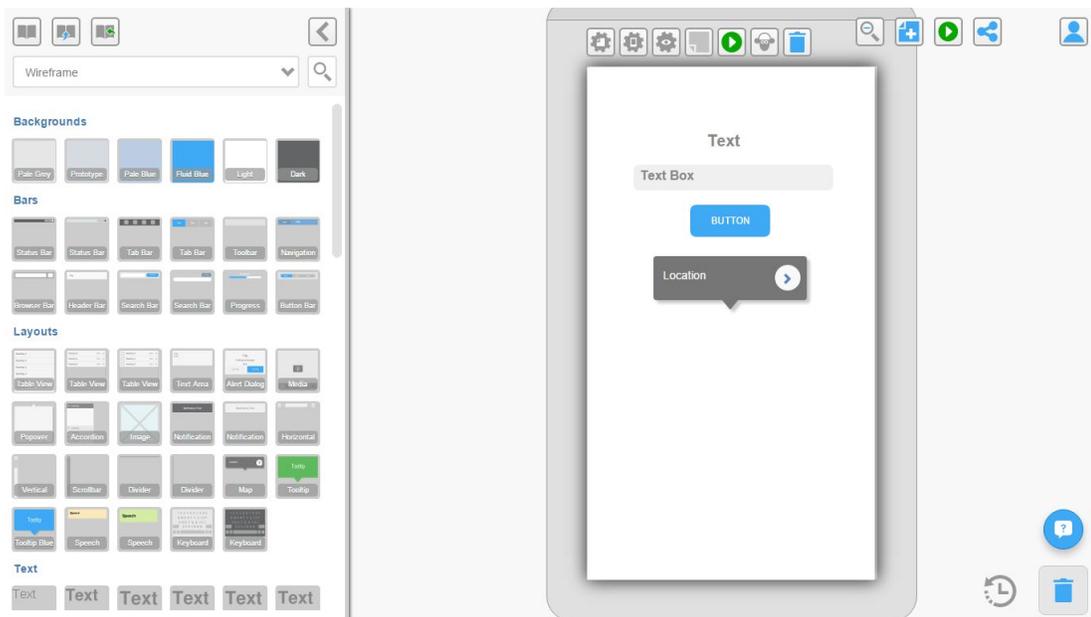


Figura 2.2: Herramienta Web de Prototipado FluidUI Componente Location.

- **Proto.io²**

Es una aplicación web que permite el prototipado de interfaces para dispositivos móviles, pero no genera interfaces desktops. Se puede observar esta herramienta en la Figura 2.3, donde se muestra la creación de un texto junto con un input y un botón. Esta herramienta que provee una variada gama de componentes visuales, además agrega características como el uso de una *Grid*, para facilitar la distribución de los componentes en el prototipo. En la Figura 2.4 se puede observar el panel de configuración asociado al componente *Grid*.

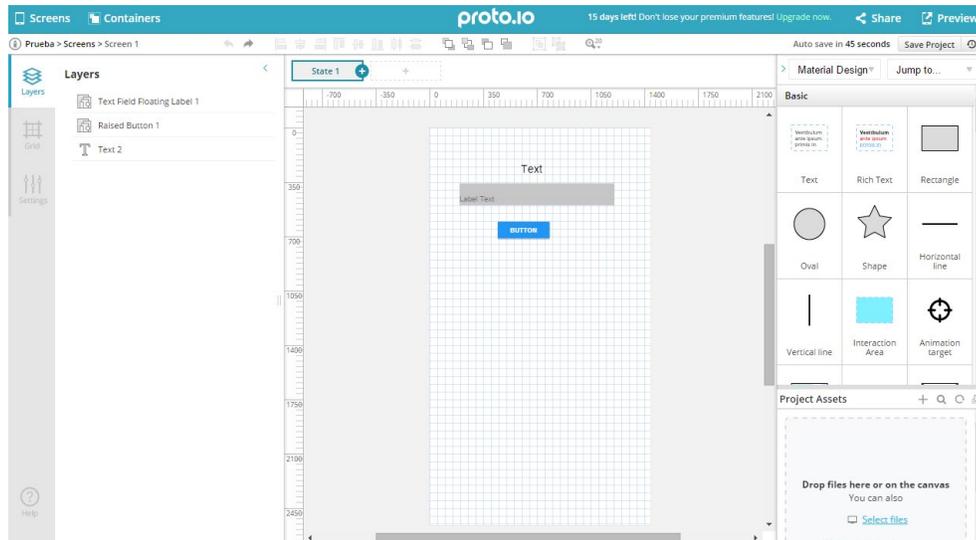


Figura 2.3: Herramienta Web de Prototipado *Proto.io*.

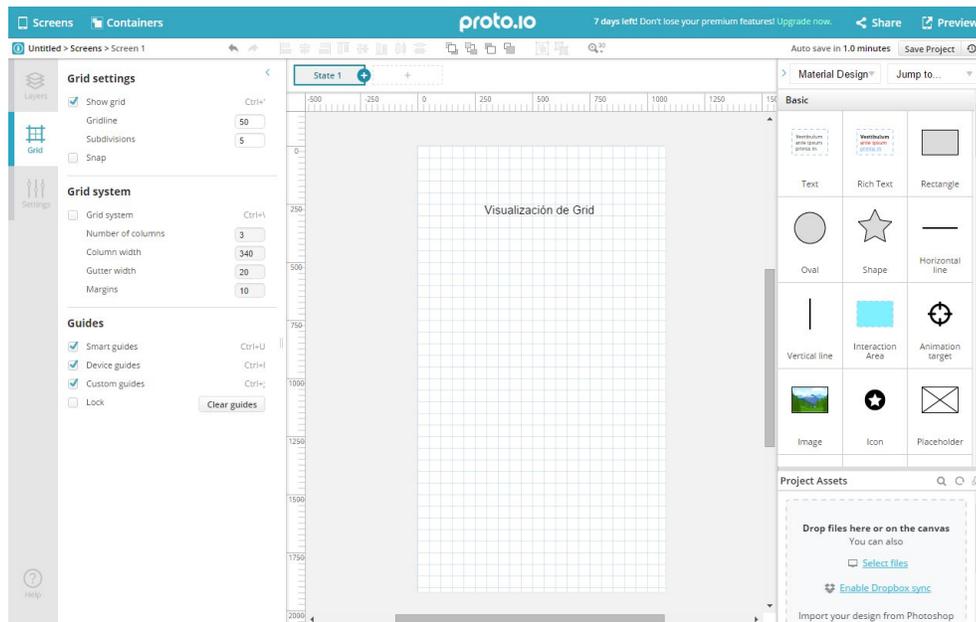


Figura 2.4: Componente *Grid* en la aplicación Web de Prototipado *Proto.io*.

² Página de *Proto.io*: <https://proto.io> (Último acceso 14-10-2016)

Proto.io no maneja el concepto de componentes mapas. Sin embargo, se puede destacar que esta herramienta brinda la posibilidad de iniciar un proyecto desde una plantilla preexistente (*templates*). Estas plantillas abarcan desde prototipos visuales para el uso de las redes sociales, como así también uso de menús. De esta manera, el usuario puede optar por inicializar su proyecto con una de estas plantillas, y continuar agregando las componentes necesarias para su dominio de aplicación. Un ejemplo de esta funcionalidad mencionada se puede observar en la Figura 2.5, donde se utiliza la plantilla “social”.

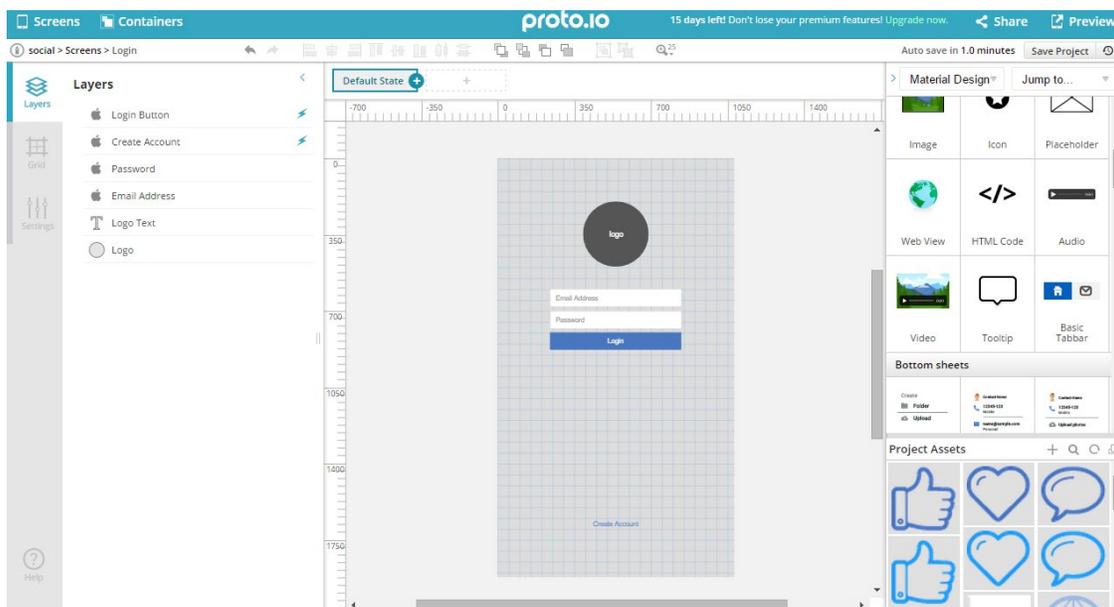


Figura 2.5: Uso de la plantilla social en la herramienta web de prototipo *Proto.io*.

- ***Justinmind Prototyper***³

Es una aplicación de tipo desktop, la misma es descargada desde una página web e instalada posteriormente. Permite que, al inicializar un proyecto, se pueda elegir algún *template* (plantilla) para guiar al usuario a la hora de utilizar la aplicación. También provee la posibilidad configurar el tamaño de la pantalla de prototipo. El tamaño de esta pantalla debería corresponderse con las dimensiones de pantalla del o los celulares para los que se piensa luego desarrollar. Se observa en la Figura 2.6 como es la interfaz de esta herramienta, en particular se muestra la creación de un texto junto con un input y un botón, al igual de como se venía probando con las herramientas anteriores.

Una funcionalidad que se destaca de esta herramienta, es que presenta la opción de incluir una componente de tipo mapa, este tipo de componente es simplemente una visualización de una imagen que representa un mapa. La diferencia es que, en este caso, el mismo se representa con una imagen de *Google Maps*⁴. Un ejemplo de esta componente se puede observar en la Figura 2.7.

³ Página de *Justinmind Prototyper*: <http://www.justinmind.com> (Último acceso: 14-10-2016)

⁴ Página de *Google Maps*: <https://maps.google.com> (Último acceso: 14-10-2016)

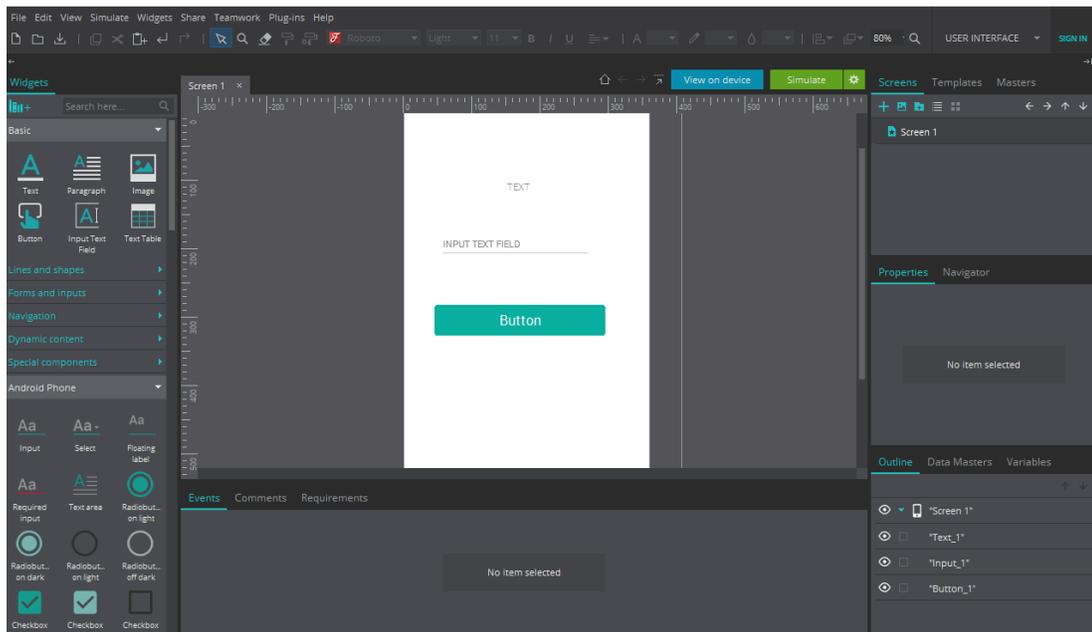


Figura 2.6: Herramienta Desktop *Justmind Prototyper*.

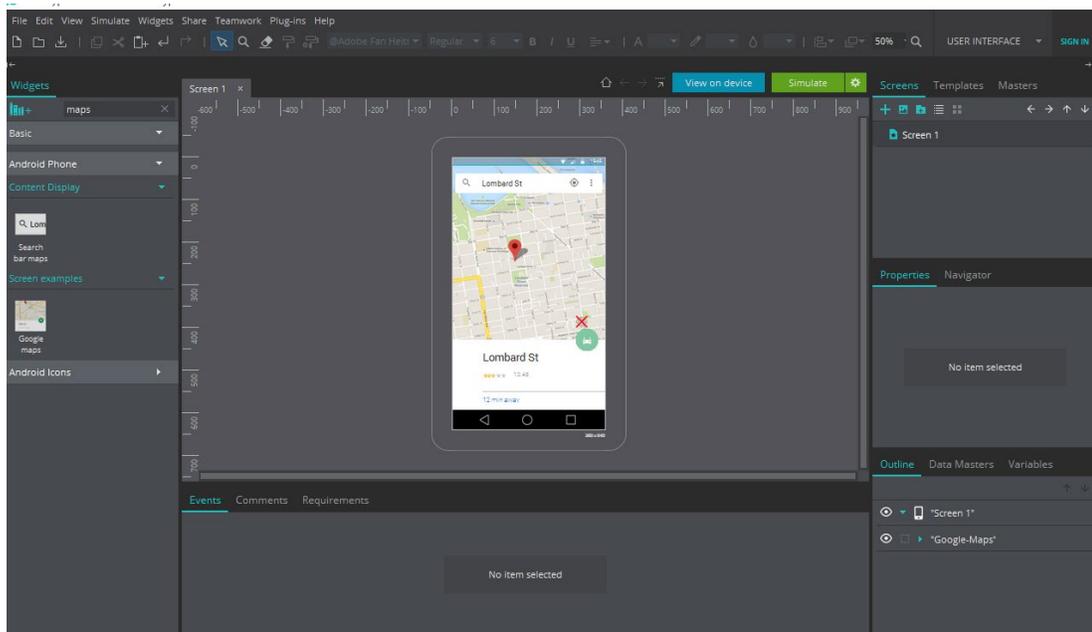


Figura 2.7: Ejemplo de componente Google Maps en *Justmind Prototyper*.

2.2 Herramientas de construcción de aplicaciones móviles

A la hora de realizar el relevamiento se encuentran otro tipo de herramientas, las cuales permiten no solo la construcción de maquetas visuales como las presentadas en la Sección 2.1, sino también permiten la creación de una aplicación con sus funcionalidades.

Para describir este tipo de herramienta, se mantiene el ejemplo que se venía mostrando en las herramientas presentadas en la Sección 2.1, donde se agregaba una componente del tipo texto, otra del tipo input y finalmente una del tipo botón, más allá de mostrar aspectos particulares de cada una de ellas.

- **Appery.io**⁵

Es una herramienta de construcción de aplicaciones móviles del tipo web. Al utilizarla, se puede ver un preview de la aplicación generada. Pero para observar las funcionalidades no se puede probar *on the fly* (en la misma herramienta), sino que hay una simulación más tardía porque hay que generar la aplicación ejecutable para probarla, en *Android*, *iOS* o *Windows Phone*. Otro punto a observar es que la herramienta se basa en conceptos de *Web Responsive*, es decir, un diseño adaptable al tipo de pantalla [Frain, 2012].

Como se puede apreciar en la Figura 2.8, un texto, un input y un botón. Además, se puede apreciar en la figura en el menú de la izquierda, la lista de archivos relacionados con la aplicación que se va a generar, por ejemplo, se puede apreciar los *CSS* o *JavaScript*.

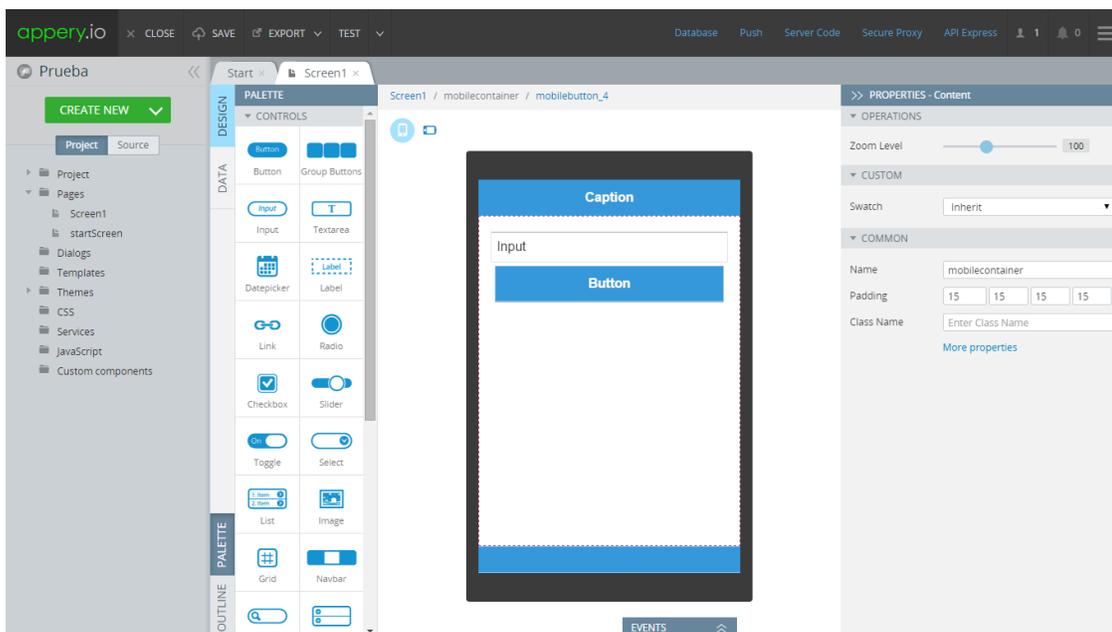


Figura 2.8: Herramienta de desarrollo de aplicaciones *Appery.io*.

Appery.io permite agregar componentes de tipo mapa, utilizando la API de *Google Maps*. Esta componente permite especificar las coordenadas (*latitud, longitud*) de un punto de interés en particular, no así la posición donde el usuario se encuentra ubicado. Es decir, este mapa no muestra la posición actual del usuario, solo el POI indicado en la componente como algo fijo que siempre se muestra al usuario. En la Figura 2.9 se puede apreciar el mapa agregado junto con un punto de interés posicionado en este (siendo uno el máximo permitido a utilizar), y en la columna de la derecha se observa que es posible configurar características relacionadas al

⁵ Página de *Appery.io*: <https://appery.io> (Último acceso: 14-10-2016)

Pol. Este componente sirve para prototipar mapas del estilo usado para indicar donde se encuentra un determinado Pol.

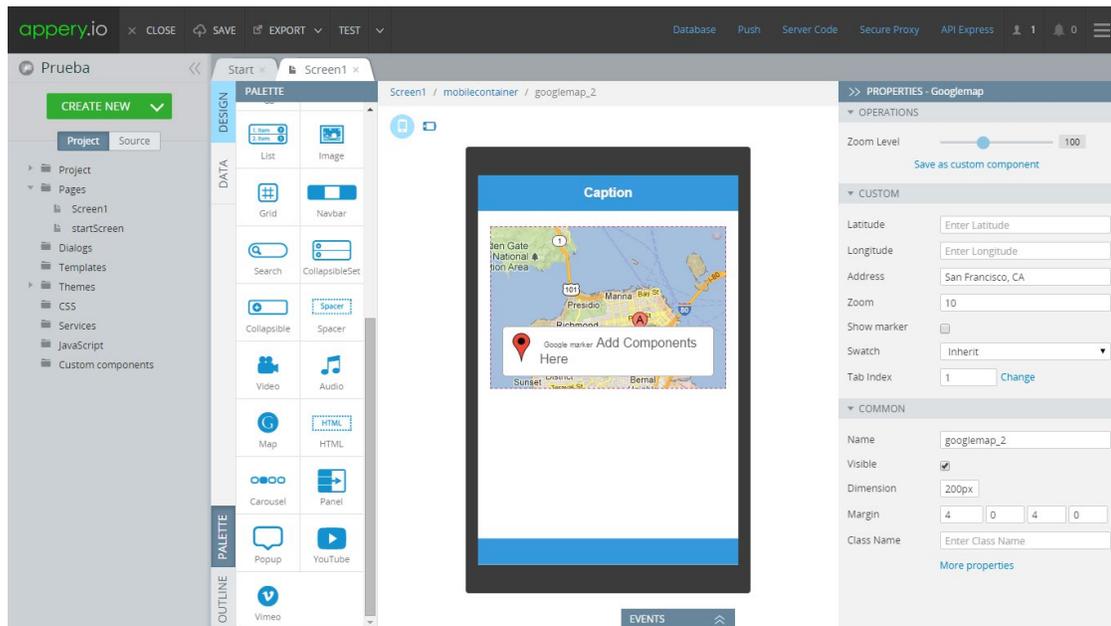


Figura 2.9: Herramienta de desarrollo de aplicaciones *Appery.io* utilización componente Map.

- ***IbuildApp***⁶

Es una aplicación web para la creación de aplicaciones móviles. Podemos observar la interfaz de la aplicación en la Figura 2.10, como primera característica sobresaliente encontramos que, a la hora de construir estas aplicaciones, las componentes no son del tipo *Drag and Drop* sino que el usuario debe poseer conocimientos de HTML5, CSS y, de ser necesario, JavaScript para poder definir sus interfaces. Es decir, es una herramienta destinada a usuarios expertos o con mínimos conocimiento de estas tecnologías.

Una de las características que posee esta herramienta es la existencia de la variedad de *templates* (plantillas), esto permite partir de una base ya armada, es decir, los archivos HTML5, CSS y JavaScript ya están especificados. Esto es de gran ayuda para usuarios no expertos con mínimos conocimientos, los cuales pueden partir de estas plantillas y llegar así de una forma más fácil a tener una aplicación funcionando.

En la Figura 2.11 se puede observar varios *templates* provistos por la aplicación *IbuildApp*. Estos están destinados a diferentes dominios, y cubren así una variada gama de plantillas. Estos dominios se pueden apreciar en el menú de la izquierda donde se listan las categorías disponibles de plantillas.

⁶ Página de *IbuildApp*: <http://es.ibuildapp.com> (Último acceso: 14-10-2016)

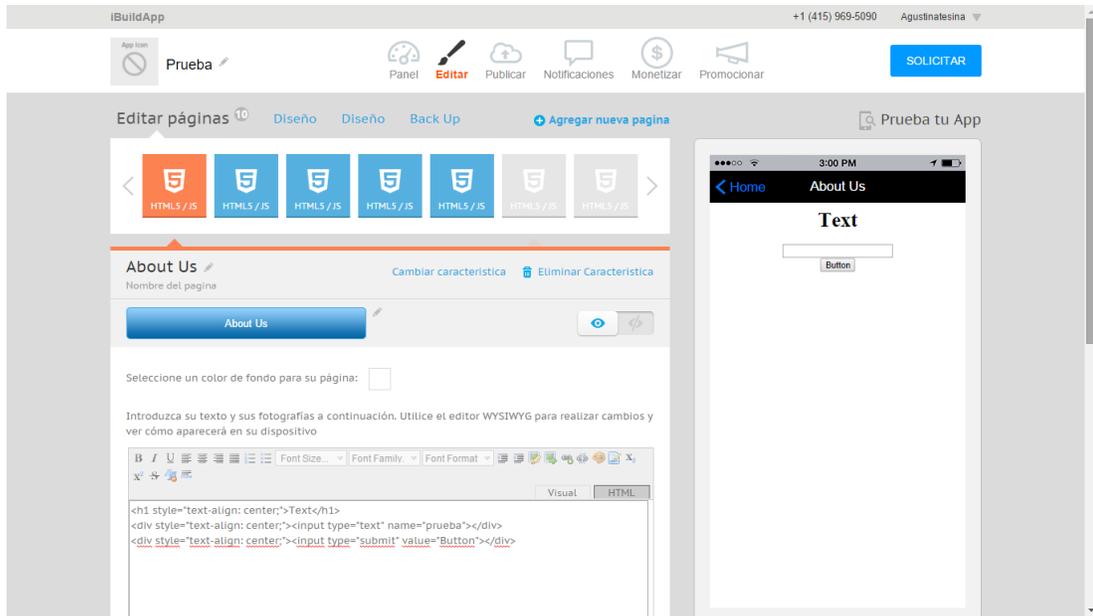


Figura 2.10: Herramienta de desarrollo de aplicaciones *IbuildApp*.

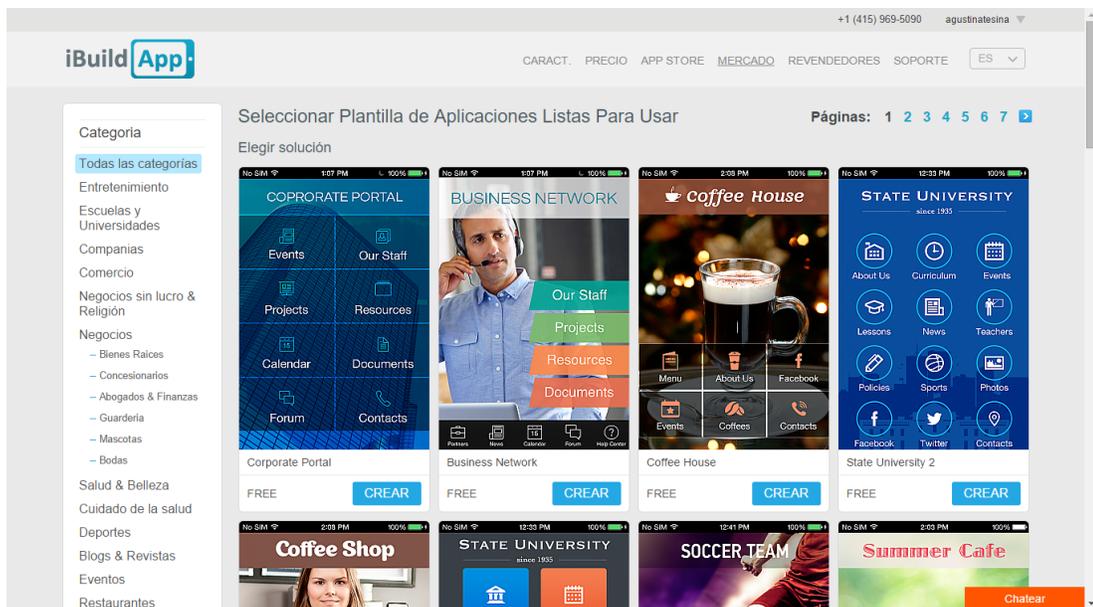


Figura 2.11: Ejemplos de template en *IbuildApp*.

IbuildApp maneja además componentes del tipo mapa. Esta componente es igual al que mencionamos que tenía *Appery.io*, esta implementada mediante *Google Maps*, donde se permite configurar las coordenadas (*latitud, longitud*) de un punto de interés específico, sin tener en cuenta la posición del usuario. Es decir, este mapa no muestra la posición actual del usuario, solo el Pol indicado, el cual es siempre visualizado por el usuario. En la Figura 2.12 podemos observar que esta componente está definida a través de la API de *Google Maps* y es utilizada para mostrar la posición, por ejemplo, de un comercio.

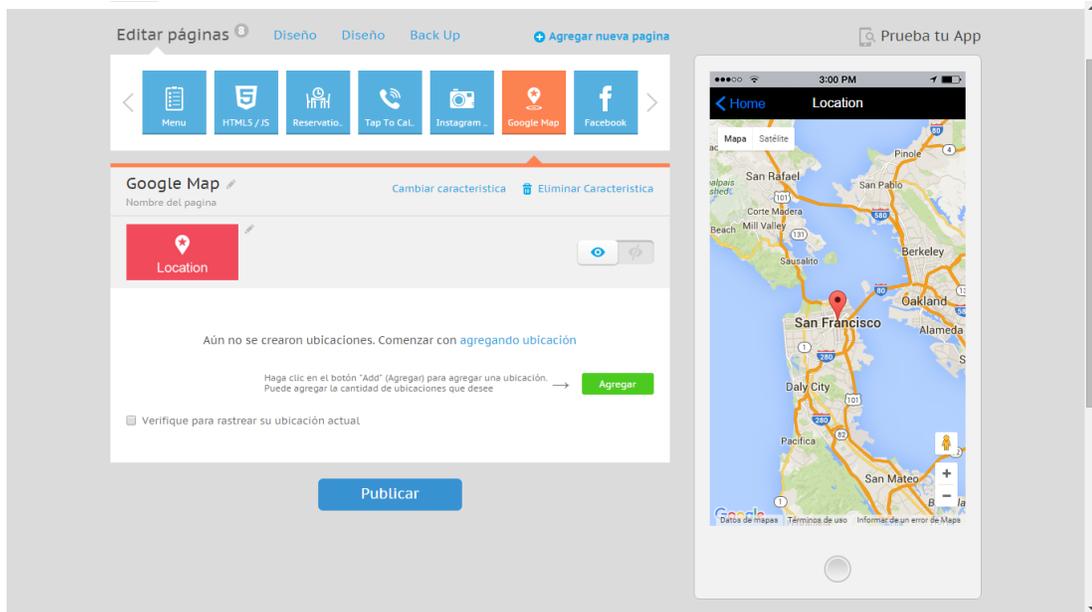


Figura 2.12: Componente mapa en *iBuildApp*.

En la Figura 2.13 podemos observar los distintos parámetros que se pueden configurar para la componente tipo mapa, como lo son una *latitud* y una *longitud*, donde se posiciona el POI como así también detalles del mismo.

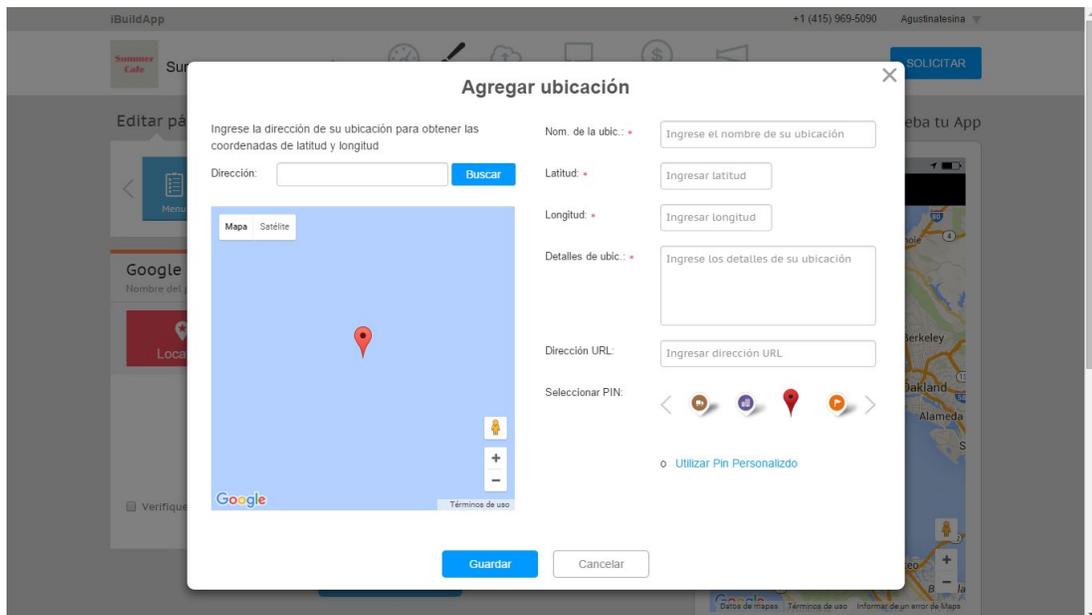


Figura 2.13: Configuración del componente mapa en *iBuildApp*.

2.3 Herramientas de creación de aplicaciones móviles con fines culturales para usuario finales

El dominio cultural es ampliamente usado por los usuarios de aplicaciones móviles. En el relevamiento, se encontró un paper que resume las herramientas de creación más desatadas en este dominio. En esta sección se presentarán las herramientas detalladas en [Fidas et al., 2015], donde se menciona la importancia que fueron tomando las aplicaciones móviles en el contexto del patrimonio cultural. Estas aplicaciones son utilizadas en los recorridos de los museos como así también en recorridos turísticos. En los últimos años ha crecido el número de aplicaciones que permitían a usuarios finales (no expertos) proporcionar contenido y modificar las mismas afectando la experiencia del usuario final.

Es interesante destacar, que la participación de los usuarios finales en el ciclo de desarrollo facilita la creación de aplicaciones, en particular, aplicaciones móviles, facilitando así que los usuarios personalicen sus aplicaciones. En esta sección se hará hincapié en herramientas de creación de aplicaciones móviles con fines culturales relevadas en [Fidas et al., 2015]. Estas herramientas cuentan con la particularidad de explotar funcionalidades de posicionamiento (geo-posicionamiento) para la creación de dichas aplicaciones móviles.

Cabe destacar que los autores en [Fidas et al., 2015] usan de base la definición de [Lieberman et al., 2006], donde estos últimos autores definen que el desarrollo por parte de los usuarios finales puede ser definido como un conjunto de métodos, técnicas y herramientas que permiten a los usuarios de sistemas de software, quienes están actuando de una manera no profesional (con respecto al desarrollo de software), en algún punto crear, modificar o extender un artefacto de software.

- **Hoppala: Plataforma de gestión contenidos de Realidad Aumentada**⁷ [Fidas et al., 2015]

Hoppala es una herramienta de autoría destinada a usuarios finales (end-user authoring tool), soporta la creación de contenidos para la producción/desarrollo y manejo de contenidos basado en posicionamiento (geo-based). Estos contenidos pueden ser usados en aplicaciones móviles culturales de realidad aumentada.

Como se puede observar en la Figura 2.14, desde un punto de vista conceptual el sistema de creación provee una interfaz de manipulación fácil de utilizar, la cual asocia objetos del mundo físico (regiones, edificios, monumentos, otros) con contenidos multimedia digitales. En este sentido permite a los usuarios finales, especificar puntos de interés usando una interfaz de geo

⁷ Acorde a [Fidas et al., 2015] podemos considerar la realidad aumentada (RA) es el término que se usa para definir una visión a través de un dispositivo tecnológico, directa o indirecta, de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real. Esta es la principal diferencia con la realidad virtual, puesto que no sustituye la realidad física, sino que sobreimprime los datos informáticos al mundo real.

posicionamiento integrada con *Google Maps* y correlacionar a estos con la información espacial y los elementos multimedia virtuales. Estos elementos virtuales pueden ser documentos, sitios web, videos, audios modelos 3D o imágenes y pueden ser comparados con el contenido subido por otros usuarios. Cada punto de interés es organizado en capas y los usuarios pueden utilizar elementos virtuales de un inventario el cual está compuesto por recursos que ya han sido cargados por otros usuarios y considerados como una solución de origen grupal.



Figura 2.14: Captura de pantalla de *Hoppala* [Fidas et al., 2015].

El resultado de esta herramienta es una aplicación basada en geo-posicionamiento (geo-based) de realidad aumentada. Si bien la herramienta cumple con algunas de las características enunciadas, se encontró en el relevamiento dos inconvenientes: el primero es que las últimas actualizaciones de la página de *Hoppala* fueron a mediados de junio del 2015⁸, y, el segundo, fue que no se pudo determinar si la creación de los puntos de interés se realizaba in-situ o solo sobre un mapa. Lo que termina siendo probado in-situ eran las aplicaciones generadas con *Hoppala*.

- ***ARIS: es una plataforma open-source para crear y vivenciar, juegos móviles, tours e historias interactiva*** [Fidas et al., 2015]

ARIS (Augmented Reality and Interactive Storytelling - Realidad Aumentada y Narración Interactiva) es una herramienta de creación de juegos de realidad alternativa (ARG) que utiliza el mundo real como una plataforma y utiliza narraciones para proporcionar experiencias

⁸ Algunos docentes mencionan que venían trabajando con *Hoppala*, y dejó de funcionar. <http://recursostic.educacion.es/blogs/malted/index.php/2013/02/05/aumentando-la-realidad-en-luarca> (Último acceso: 27-12-2016)

móviles que pueden ser alteradas por las ideas o acciones de los jugadores. En este contexto, esta herramienta permite producir rápidamente experiencias interactivas, posicionadas, interactivas, centradas en la narración y educativas.

Desde un punto de vista conceptual, esta herramienta está basada en dos pilares básicos, por un lado una aplicación web para herramienta de creación y por otro lado una aplicación para *iphone* la cual es requerida que sea descargada por los usuarios de los juego en orden de participar en juegos sensibles a la posición que han sido creados a través de la herramienta ARIS.

Como se puede ver en la Figura 2.15, el editor ARIS provee a sus usuarios finales diferentes funcionalidades como, crear y mantener una lista de juegos, identificar la posición en la cual se supone que el juego será jugado. Posteriormente, el usuario tiene que definir objetos/ítems (ejemplo: objetos multimedia, links a páginas web, sugerencias, otros) y posicionarlos en el mapa a través de coordenadas GPS específicas. Estos objetos, los cuales representan puntos de intereses para los jugadores, pueden ser escondidos, visibles públicamente o pueden ser activados de acuerdo a las distintas situaciones del juego. Además, la plataforma de edición permite a los usuarios finales crear y mantener personajes dentro de su diseño del juego. Es decir, es una herramienta de creación basada en mapas, y luego la aplicación generada se prueba in-situ.

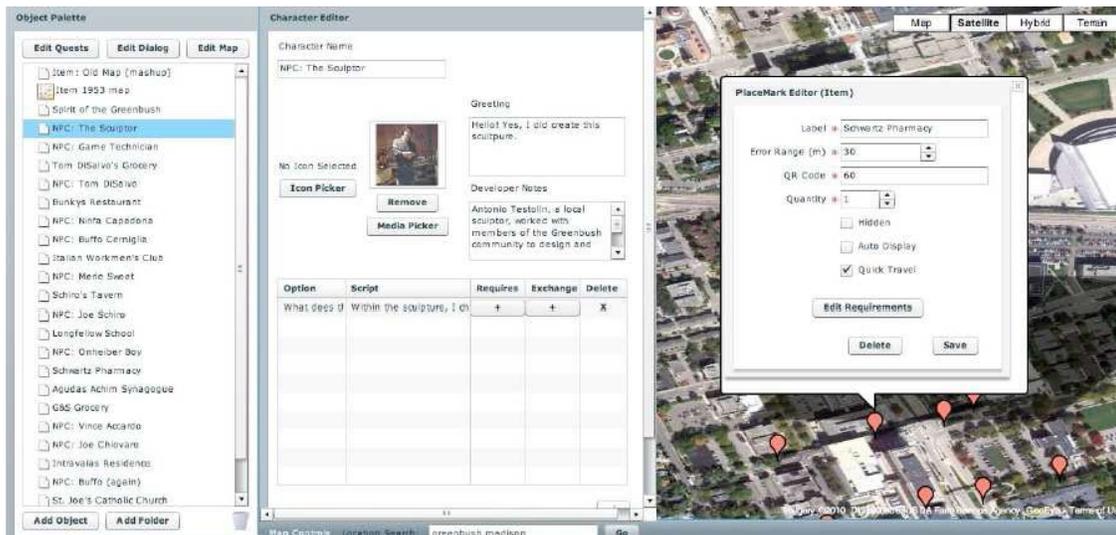


Figura 2.15: Captura de pantalla del editor ARIS [Fidas et al., 2015].

- **TaggingCreditor: herramienta para crear y compartir contenido para juegos de aprendizaje basados en posicionamiento** [Fidas et al., 2015]

TaggingCreditor es una herramienta de creación de contenidos indoor y outdoor para juegos basados en posicionamiento. Esta tiene como objetivo soportar actividades de aprendizaje interactivas en sitios con fines culturales. Está basada en la idea de las conexiones entre los contenidos digitales y el mundo físico. La idea general es atraer a los visitantes a participar en

juegos en línea, los cuales son un proceso de descubrimiento de relaciones entre las exhibiciones del museo y los objetos que no se encuentran en la exhibición pública.

Desde un punto de vista conceptual. La idea subyacente está basada en una ontología con diferentes capas de abstracciones las cuales correlacionan los elementos del mundo físico (objetos del mundo real, como exhibiciones de museos, monumentos o edificios, lugares históricos, otros) con sus respectivos elementos del mundo virtual (principalmente contenido virtual). Dentro del juego es necesario establecer las relaciones correctas entre las entidades del mundo físico y las del mundo virtual, ya que estas otorgan créditos al jugador. Esto de alguna manera etiquetar los elementos de ambos mundos y relacionarlos.

Desde el punto de vista de la interacción del usuario, como se puede observar en la Figura 2.16, la versión móvil de *TaggingCreditor* puede ser usada por un usuario final para: (a) definir objetos reales, (b) definir objetos virtuales, (c) crear la relación entre estos dos.

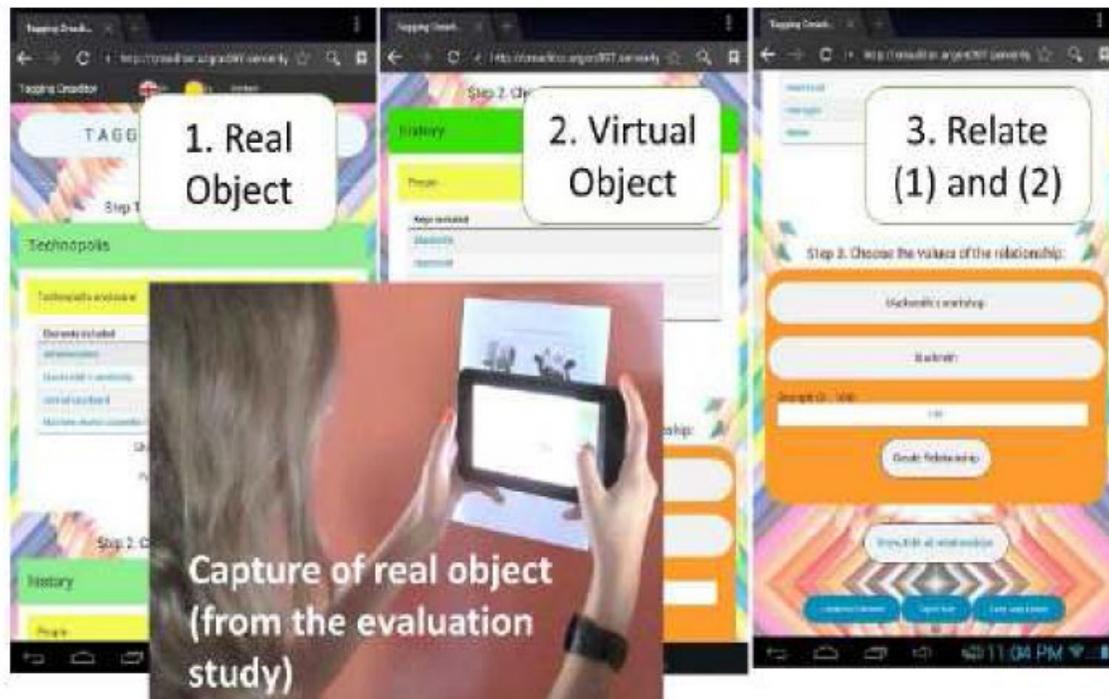


Figura 2.16: Captura de pantalla de *TaggingCreditor* [Fidas et al., 2015].

- **CHEF: Una perspectiva centrada en el usuario para frameworks empresariales de patrimonio cultural** [Fidas et al., 2015]

CHEF provee un ambiente visual de manipulación directa, donde el dominio del patrimonio cultural instancia esquemas de diseño con los contenidos multimedia apropiados y genera aplicaciones hipermedias de gran calidad para las diferentes plataformas sin la necesidad de aprender implementación de alguna tecnología específica. El objetivo final de *CHEF* es el de habilitar a los profesionales del patrimonio cultural a crear y mantener sus objetos hipermediales sin la necesidad de capacitarse o de un programador.

Desde un punto de vista conceptual esta herramienta aspira a proporcionar un framework empresarial el cual denota un esqueleto de aplicación “reusable”, “semi-completo” que puede adaptarse fácilmente para producir aplicaciones de patrimonio cultural a medida.

En la Figura 2.18, se puede observar una captura de una aplicación creada a través de CHEF. Se puede apreciar que la misma fue creada ya hace algunos años por el tipo de dispositivo usado para la prueba.



Figura 2.18: Captura de pantalla de una aplicación creada a través de CHEF. [Fidas et al., 2015].

- **LMAC (Creador de aplicaciones móviles basadas en posicionamiento): creando búsquedas del tesoro móviles educativos** [Fidas et al., 2015]

LMAC presenta un framework para crear diferentes aplicaciones móviles basadas en el contexto de posicionamiento, que ha sido utilizado para crear contenido para tours guiados o juegos del tipo búsqueda de tesoro en sitios de interés. Las aplicaciones pueden ser creadas o bien como simple aplicaciones con un marco informativo, o como una búsqueda del tesoro con elementos de diseño de juegos para hacer la experiencia más interesante. Estos juegos pueden ser diseñados de una manera competitiva o colaborativa. En juego colaborativo puede ser desarrollado creando diferentes aplicaciones de búsqueda del tesoro para cada usuario (o grupo de usuarios), ya que los conocimientos y habilidades de muchas personas son

requeridos para el encontrar el punto de interés final. En una configuración competitiva, los usuarios tratan de encontrar todos los puntos de interés por su propia cuenta y tratan de ser el primero (o más rápido) en encontrar el punto de interés final.

El software permite la creación de diferentes escenarios de aplicación. Esto puede ser o bien la creación de (1) una aplicación para ayudar a los turistas a obtener información sobre una ciudad, (2) una aplicación para obtener información sobre áreas desconocidas, tal como puede ser el tour a través de una gran y compleja universidad o (3) una aplicación para aprender acerca del entorno, como sería información histórica o atracciones geológica mientras se realiza una excursión/caminata.

En [Pirker et al., 2014] no se menciona la idea de la construcción in-situ. Por otro lado, menciona que las interfaces de la aplicación móvil creadas a partir de la plataforma web LMAC, son aplicaciones diseñadas con un foco en la compatibilidad multiplataforma. Las aplicaciones móviles resultantes, pueden ser accedidas o ingresando la URL especificando un ID que es único de cada aplicación o seleccionando una ruta desde un acceso abierto en una lista, como se puede ver en la Figura 2.19.

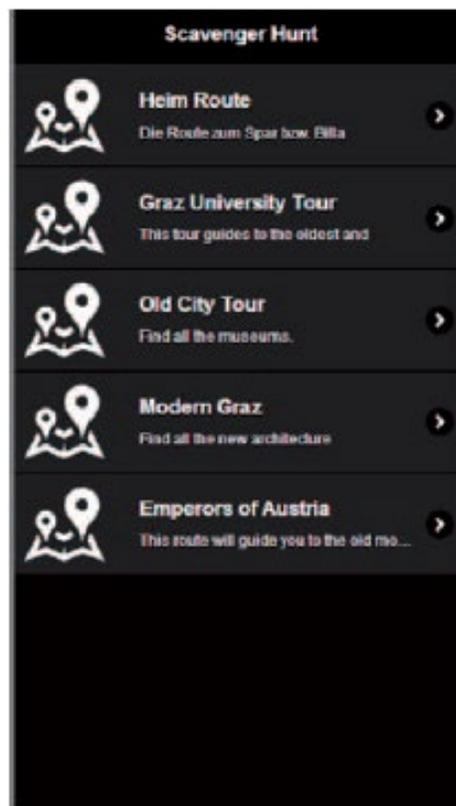


Figura 2.19: Selección de la lista de uno de los recorridos [Pirker et al., 2014].

Desde un punto de vista conceptual como se puede observar en la Figura 2.20, LMAC está diseñado como un editor online para crear aplicaciones móviles, el cual utiliza información geográfica y puede activar eventos cuando una posición en particular está cercana a un punto de interés. Cada punto de interés está relacionado a una actividad, la cual es desencadenada

cuando el usuario de la aplicación móvil está cerca a ese punto. La interface de la búsqueda del tesoro está basada en una vista de mapa, la cual muestra la posición y la distancia al punto de interés más cercado. Al usar la aplicación generada, cuando el usuario se acerca a ese punto, la actividad relacionada aparece. El usuario tiene un proceso de información (como, por ejemplo, mirar un video) o hacer alguna tarea (como responder una pregunta) y después de esto recibirá la posición del próximo punto de interés a donde debe dirigirse.

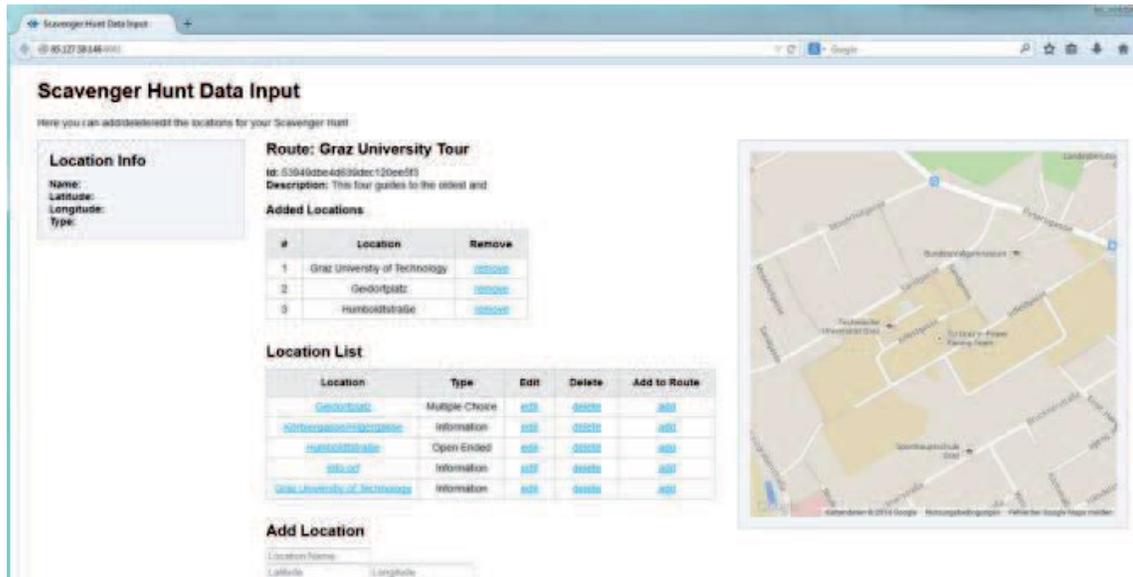


Figura 2.20: Intefaz web de LBMAC [Pirker et al., 2014].

2.4 Herramienta para la creación de elementos posicionados

En [Alconada Verzini, 2015] se desarrolló una herramienta prototípica *"eBuilder"* que permite crear contenidos, posiciones o elementos posicionados. Esta herramienta está orientada a usuarios no expertos y permite definir, en posiciones específicas, diferentes contenidos como, por ejemplo, textos videos, imágenes, entre otros. Utilizando mapas de base para facilitar el posicionamiento de los elementos. Permite la construcción de experiencias in-situ y da soporte para la representación de espacios indoor, outdoor y espacios combinados (indoor-outdoor). Es decir, esta característica in-situ permite que mientras el usuario se está moviendo por el espacio físico puede ir creando contenidos posicionados.

En la Figura 2.21, se puede observar el menú principal que se le presenta al usuario para crear los contenidos posiciones o elementos posicionados mencionados previamente.



Figura 2.21: Menú principal de la herramienta *eBuilder* [Alconada Verzini, 2015].

En la [Alconada Verzini, 2015] se implementan y se hace énfasis en las siguientes acciones:

- *Crear sólo capa de 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.

A la hora de seleccionar la opción de “*crear contenidos*” (del menú principal Figura 2.21) se abre una nueva pantalla con una grilla de fondo donde se ubicarán los contenidos a medida que se vayan creando. Esta acción se visualiza en la Figura 2.22.

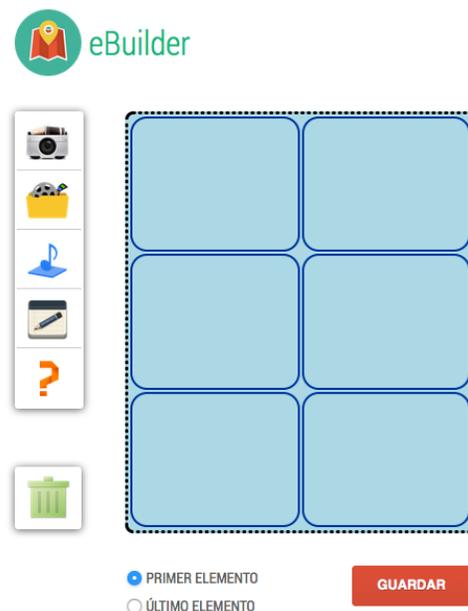


Figura 2.22: Creación de nuevo contenido [Alconada Verzini, 2015].

Cabe destacar que la herramienta permite al usuario conectar contenido entre sí formando alguna especie de estructura (lineal, grafo o conjunto) según sea la naturaleza del contenido que se está creando.

La herramienta, 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. Estos contenidos son representados por distintos iconos, se muestra en la Figura 2.23, la correspondencia entre la representación y el tipo de contenido.

IMAGEN	AUDIO	TEXTO	VIDEO	PREGUNTA	COMBINADO

Figura 2.23: Tabla de referencia entre la representación gráfica y el contenido [Alconada Verzini, 2015].

- *Crear sólo capa de posicionamiento*

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 realizar la opción se “*crear posiciones*” (del menú principal Figura 2.21) se abre una nueva pantalla que permite al usuario elegir qué tipo de espacio físico desea utiliza. La herramienta permite seleccionar tres tipos de mapas, un mapa 2D indoor, un mapa georeferenciado o un mapa satelital. Los tres tipos mencionados se pueden apreciar en la Figura 2.24.



Figura 2.24: Creación de posiciones [Alconada Verzini, 2015].

En la Figura 2.25 que se puede observar un ejemplo de creación de un mapa georeferenciado. En esta figura se puede apreciar que hay un mapa indoor superpuesto a la imagen satelital. Cabe mencionar que el mapa indoor se encuentra remarcado a fin de que se pueda distinguir

mejor el mismo. Además, se pueden distinguir tres posiciones relevantes tanto sobre el mapa indoor como sobre la imagen satelital.

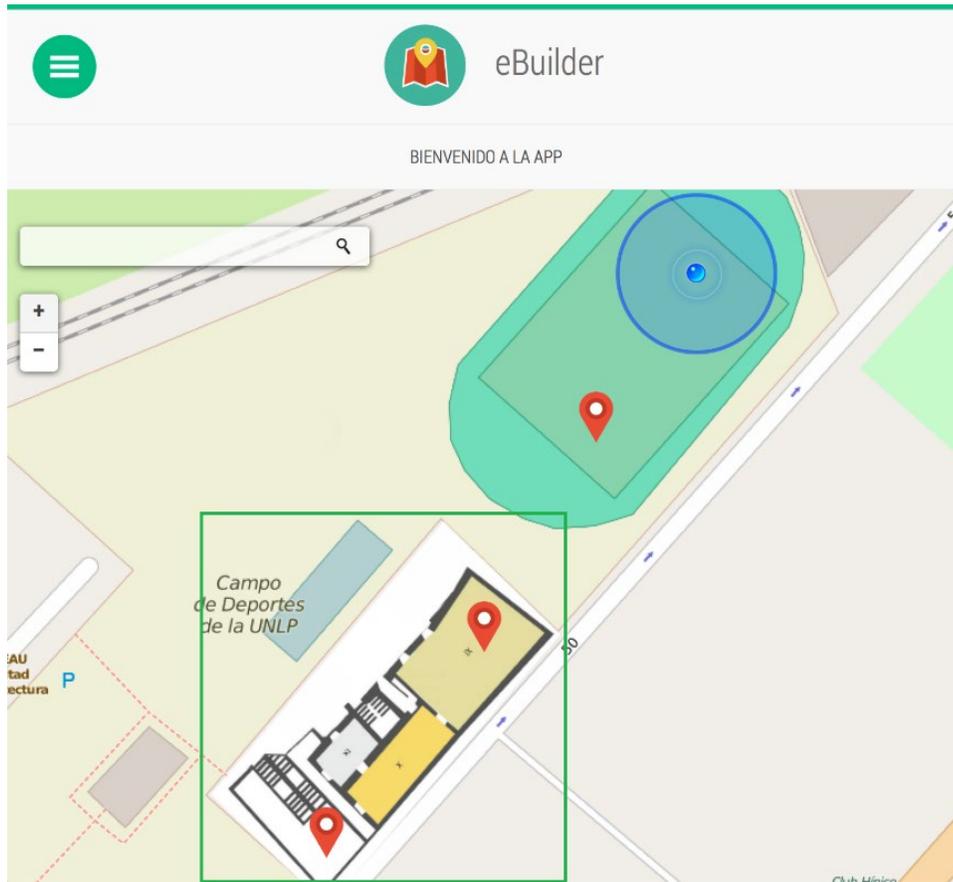


Figura 2.25: Ejemplo de mapa georeferenciado [Alconada Verzini, 2015].

- *Crear elementos posicionados:*

En *eBuilder* se define a los elementos posicionados como a aquellos que tienen contenido asociado una posición dentro de espacio físico. Para realizar la creación de un elemento posicionado, el usuario debe realizar una serie de pasos en las que se incluyen: seleccionar la opción de creación de elemento posicionado, la elección del tipo de mapa. Y una vez elegido el mapa, el usuario puede comenzar a crear elementos posicionados, así como también conectarlos entre sí siguiendo las mismas interacciones que el usuario utilizó para la creación de contenidos. Siguiendo estos pasos, un usuario puede crear los elementos posicionados directamente sobre el espacio físico elegido.

En la Figura 2.26 se puede observar un ejemplo de cómo se presenta el menú para la creación de los elementos posicionados. En este caso el usuario eligió trabajar sobre un mapa indoor.

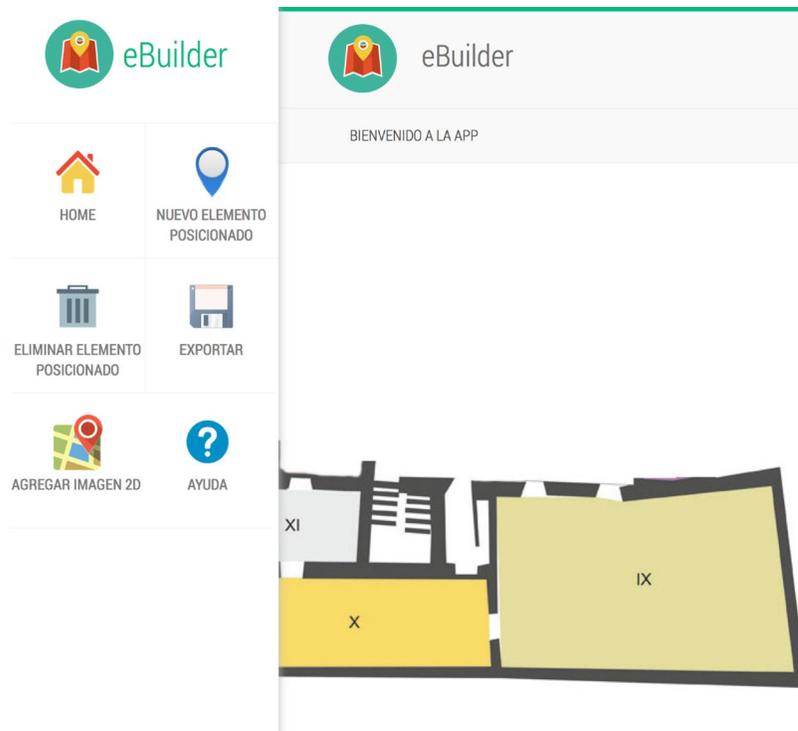


Figura 2.26: Menú de opciones al momento de crear un elemento posicionado [Alconada Verzini, 2015].

Cuando se están creando los elementos posicionados, se presenta al usuario la opción de establecer si se desea utilizar el GPS o códigos QR⁹ como mecanismo de sensado asociado a dicho elemento.¹⁰ Al estar creando contenidos posicionados (lo que llamamos elementos posicionados), el box de creación tendrá también la opción de establecer cuál de estos mecanismos de sensado se desea utilizar. Para hacer esto, en la esquina superior derecha del box de creación, aparece un ícono con una brújula (como se puede apreciar en la Figura 2.27.a). Esto indica que el mecanismo de sensado definido por defecto es el GPS. Si se quiere utilizar códigos QR, se debe apretar dicho botón y cambiará el ícono (ver Figura 2.27.b).

Cabe mencionar que este cambio de mecanismo de posicionamiento puede darse en cualquier momento, por lo tanto, pueden quedar algunos elementos posicionados a códigos QR mientras otros asociados a coordenadas de (latitud-longitud). Esto luego deberá ser interpretado por la aplicación que los utilice. Por ejemplo, una aplicación móvil que utilice lo creado por esta herramienta, podría funcionar de la siguiente manera: en el caso de definirse posicionado por código QR, el usuario deberá leer dicho código para poder recibir el contenido; mientras que en el caso de haber elegido GPS, cuando se detecte que el usuario está en esa posición se le mostrará automáticamente el contenido.

⁹ Código QR: El código QR (Quick Response) es un código bidimensional matricial que posee patrones de detección de posición en tres esquinas [Kato et al., 2010].

¹⁰ Cuando el usuario utiliza como mecanismo de sensado códigos QR, los generados por la aplicación son exportados cuando se exporta la aplicación, los mismos puede ser impresos y colocados de manera tal que se correspondan con los elementos posicionados creados.

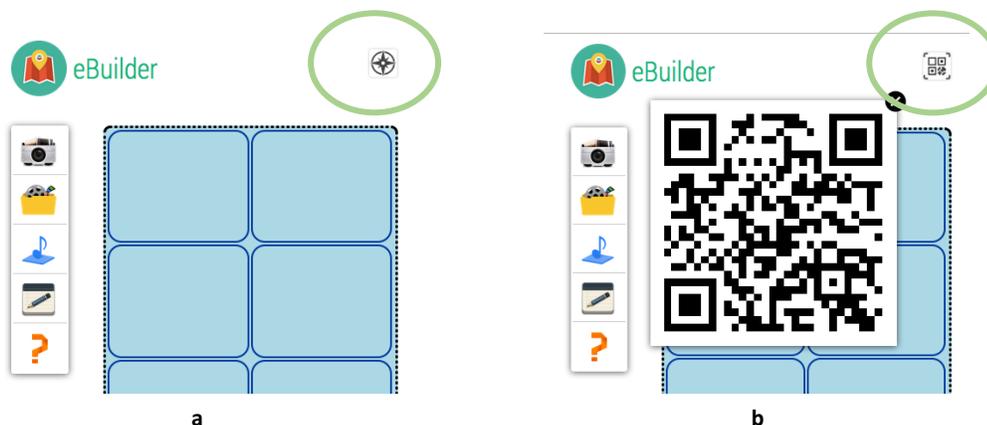


Figura 2.27: Mecanismos de sensado [Alconada Verzini, 2015].

A continuación, en la Figura 2.28 se muestra tres elementos posicionados conectados utilizando un mapa indoor. Se puede observar que cada uno de estos tiene una iconografía distinta, estas determinan el tipo de contenido que tiene el mismo (como se mostró en la Figura 2.23).

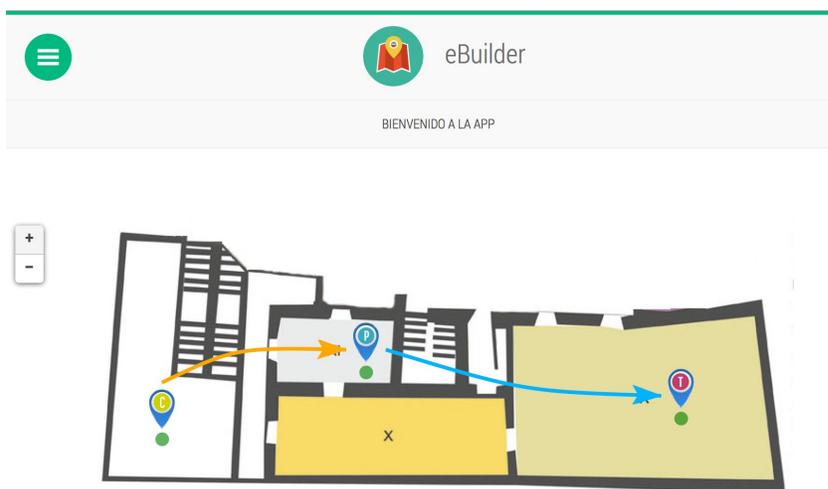


Figura 2.28: Ejemplo de elementos posicionados en un mapa indoor [Alconada Verzini, 2015].

Cabe destacar que la herramienta definida en [Alconada Verzini, 2015] provee un mecanismo que permite que las especificaciones que haga el usuario durante el uso de la herramienta puedan ser llevadas a un formato exportable. 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. Esto permite la futura reutilización de cada capa.

2.5 Análisis de las herramientas presentadas

Como se pudo apreciar en las Secciones 2.1 y 2.2, se presentan herramientas de maquetado o de aplicaciones móviles, pero las mismas no se focalizan en el posicionamiento de contenido o

informar. Algunas de las herramientas proveen componentes de mapas pero más que nada para indicar la posición, por ejemplo, de un Pol en particular. Una característica a destacar de estas herramientas es que varias proveen plantillas (*templates*) que facilitan al usuario el uso de las mismas. Estas herramientas han permitido detectar las características que deben de tener las herramientas destinadas a usuarios finales y como cada una aborda la asistencia a los mismos (por ejemplo, usando *templates*).

Recordemos que el objetivo de la tesina es poder brindar una herramienta que facilite la creación de Actividades Educativas Posicionadas, con lo cual, las herramientas presentadas en las Secciones 2.1 y 2.2 son muy generales, sin embargo las presentadas en las Secciones 2.3 y 2.4 se aproximan más a las necesidades requeridas para poder abordar el objetivo de la tesina. Acorde a esto, veamos a continuación una tabla de resumen comparativo entre estas herramientas de las Secciones 2.3 y 2.4.

Cabe mencionar que en [Alconada Verzini, 2015] se describen dos herramientas de creación de contenido posicionado. Estas no han sido incluidas en esta tesina dado que ya en [Alconada Verzini, 2015] se hace un análisis de cada una de ellas. Sin embargo, las mismas son interesantes para incluirlas en la tabla comparativa. A modo de ayudar al lector, a continuación se presentan una breve descripción de estas dos herramientas:

- *CASTOR* [Pittarello, 2011] [Pittarello & Bertani, 2012]: es una herramienta de autor orientada a niños de siete años que permite la construcción in-situ de historias interactivas contextuales outdoor. El contenido esta acoplado a las posiciones GPS.
- *QuesTInSitu* [Santos et. al, 2011]: es una herramienta basada en mapas, que permite a los docentes definir preguntas posicionadas en espacios outdoors. Las preguntas están acopladas a posiciones (latitud, longitud).

En la Tabla 2.1 se puede apreciar un resumen de las herramientas presentadas en las Secciones 2.3 y 2.4 y además se incorporaron las herramientas *CASTOR* y *QuesTInSitu* (descriptas en detalle en [Alconada Verzini, 2015]). Se puede observar que se eligieron cuatro criterios de análisis, estos son acorde al foco de la tesina. Por un lado, se determina cuál es la forma de construcción, la cual puede ser basada en mapa, es decir, que el usuario usando un mapa de base define ahí los contenidos posicionados; en este caso, el usuario puede estar en cualquier lugar usando la herramienta. O la construcción puede ser in-situ, el usuario posicionado en el lugar va definiendo el contenido, aprovechando las ventajas de estar en el lugar donde luego se brindara el contenido creado. En la tabla también se analiza para que tipo de espacio está destinada cada herramienta, ya sean espacios indoor u outdoor. Además, se analiza los mecanismos de sensado de posicionamiento involucros. Y por último, en la tabla se analiza una característica que se busca abordar en esta tesina, y es el desacople del contenido de las posiciones, permitiendo el resuo de estos conceptos.

Tabla 2.1: Comparación de las herramientas analizadas.

Nombre	Forma de Construcción	Permite crear contenido en espacios	Mecanismo de sensado de posicionamiento	Contenido desacoplado de las posiciones
Hoppala	*	Outdoor	GPS	No
ARIS	Basada en mapa	Outdoor	GPS	No
TaggingCreaditor	In-situ	Indoor/Outdoor	Φ	*
CHEF	*	Outdoor	GPS	No
LMAC	Basada en mapa	Outdoor	GPS	No
eBuilder	In-situ (también se puede usar como basada en mapa)	Indoor/Outdoor	GPS y Código QR	No
CASTOR	In-situ	Outdoor	GPS	No
QuesTInSitu	Basada en mapa	Outdoor	GPS	No

Se puede apreciar en la Tabla 2.1 que la única herramienta que considera el desacople del contenido y posiciones es *eBuilder*. Por otro lado, en la tabla se puede observar que solo dos herramientas están destinadas a espacios indoor-outdoor, una de ellas es *eBuilder*. Otra ventaja con la cuenta *eBuilder* es que permite la construcción in-situ, con la flexibilidad de poder ser usada como basada en mapa. Acorde a esto, esta fue la herramienta que se ha decidido usar como base para extenderla acorde a los objetivos de la tesina.

* No se especifica en el material analizado de dicha herramienta.

Φ Al ser una herramienta de realidad aumentada, usa más la identificación del objeto real por imagen, y de esa manera determina que se está en frente del mismo.

3. Análisis de las características de las Actividades Educativas Posicionadas que involucran elementos concretos

Recordemos que el objetivo de la tesina, consiste en brindar una herramienta, destinada a usuarios no expertos, que facilite la creación de Actividades Educativas Posicionadas. Para esto, como primer paso hay que diseñar dicha herramienta, es decir, entender conceptualmente que características debería contemplar la misma, esto implica entender los conceptos invocados en estas aplicaciones como así también que debería poder definir, por ejemplo, un docente que usa la herramienta.

Veamos primero algunos conceptos descriptos en [Lliteras, 2015], donde se propone un enfoque de modelado de Actividades Educativas Posicionadas que involucran elementos concretos. Estos conceptos sirven para poder identificar características relevantes de este tipo de aplicaciones. En [Lliteras, 2015], se define la capa de contenidos educativos, donde dentro de esta capa se cuenta con:

- Las actividades educativas están compuestas por tareas.
- Cada tarea posee una consigna (acorde a lo definido en [Riestra, 2004]) con un enunciado que puede implicar diferentes recursos (por ejemplo, texto o video). Estas tareas podrían involucrar, por ejemplo, la recolección de elementos concretos.
- Existe una posible relación entre las diferentes tareas involucradas en una actividad educativa.

Por otro lado, en [Lliteras, 2015] se propone una capa de posicionamiento de las actividades educativas, donde además de posicionar las mismas en un espacio físico, se posicionan las tareas y los elementos concretos. Considerando que la posición de estos últimos puede variar a lo largo de la actividad.

Por lo antes mencionado, se puede observar que en [Lliteras, 2015] se presenta una separación en capas bien clara, por lo cual se deberá tener en cuenta esto, en una herramienta que facilite la creación de este tipo de aplicaciones, para poder así reusar las distintas capas.

Estos son los conceptos identificados de manera general, veamos ahora las características involucradas en aplicaciones móviles que plantean a los alumnos Actividades Educativas Posicionadas. Para esto, a continuación, se analizarán los componentes presentados en el prototipo "*Aprendo Jugando*" descripto en [Lliteras, 2015]. Dicho prototipo contempla una Actividad Educativa Posicionada en las que las tareas posicionadas involucran elementos concretos que el alumno debe recolectar y posteriormente depositar de acuerdo a un criterio (en particular, relacionado el tipo de material). Este prototipo se encuentra implementado en *Android* y hace uso de un lector de código QR externo, el cual debe estar instalado previamente en el mismo dispositivo que el prototipo desarrollado. Este lector, permitirá reconocer códigos QR, los cuales serán usados para identificar los lugares en los que se brinden tareas de recolección y de depósito. Se usan, además, estos códigos, para identificar los elementos concretos recolectados o

depositados. En la Figura 3.1 se puede apreciar una mesa donde están los elementos concretos con sus correspondientes códigos QR, y a su vez el código QR de la mesa que permitirá que al leerla el alumno reciba la tarea relacionada.



Figura 3.1: Una mesa con los elementos concretos y el código QR de la tarea [LLiteras, 2015].

En la Figura 3.2 se puede apreciar un bosquejo general donde se muestran las tareas posicionadas. En dicha figura no solo se muestran las mesas sino también los depósitos donde los alumnos tienen que depositando los elementos recolectados. Se puede observar la separación de cada uno de los conceptos involucrados.

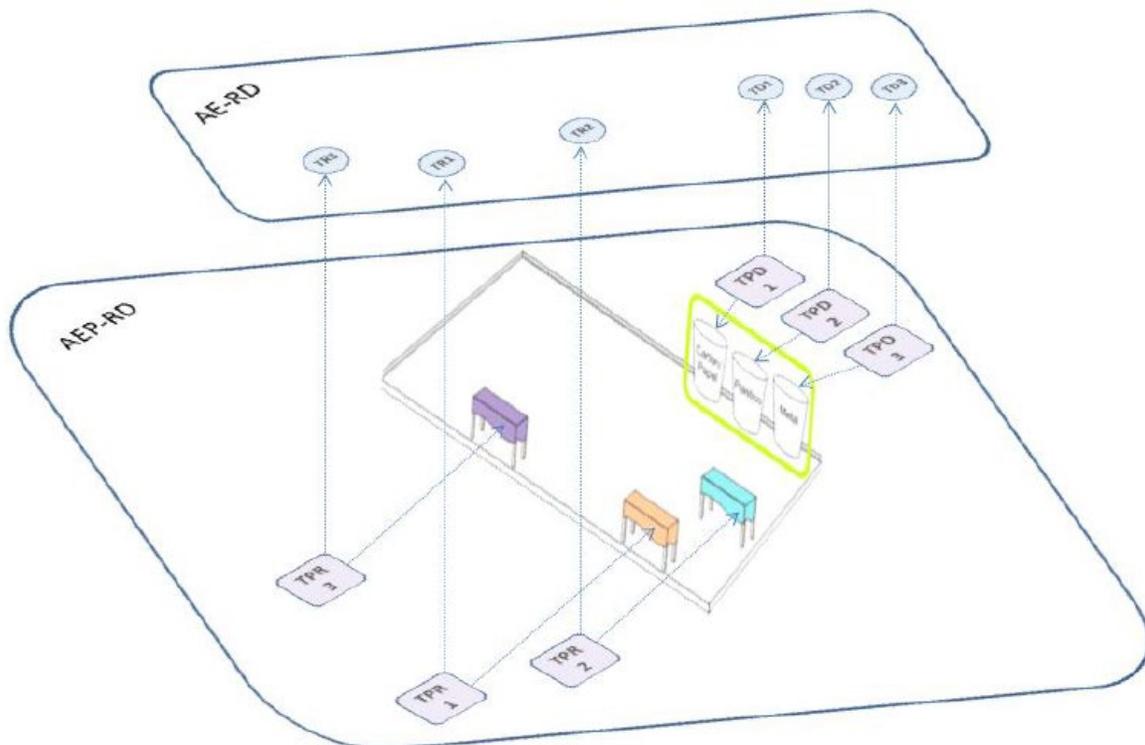


Figura 3.2: Tareas posicionadas [LLitera, 2015].

A continuación nos focalizaremos en las diferentes pantallas del prototipo “*Aprendo Jugando*”, observando en particular las diferentes características para poder ir detectando diferentes plantillas que se podrían llegar a proveer desde una herramienta de creación de este tipo de aplicaciones. Es decir, se ira presentado distintas pantallas del “*Aprendo Jugando*”, y en paralelo se irán describiendo mediante mockups (*maquetas*) posibles plantillas representativas.

➤ *Bienvenida/Inicio de Juego*

Toda aplicación tiene una pantalla de bienvenida que podría, como en el “*Aprendo Jugando*”, dar al usuario (en este caso, el alumno) una breve explicación del juego, permitiendo que el flujo de la aplicación móvil comience luego de que se presione el botón “*A jugar*”. Esto se puede apreciar en la parte izquierda de la Figura 3.3.

Acorde a esto, se podría identificar un posible mockup de plantilla que especifique: “*Título de bienvenidos al juego*”, “*Imagen representativa del juego*”, “*Párrafo Explicativo*” y “*Botón de inicio de juego*” como se puede apreciar en la parte derecha de la Figura 3.3. Esta plantilla sería editable y se podría completar acorde a las necesidades requeridas para cada aplicación de este tipo. También se podría contar con una plantilla que tenga exactamente el mismo texto que la pantalla del “*Aprendo Jugando*” y ser editable.

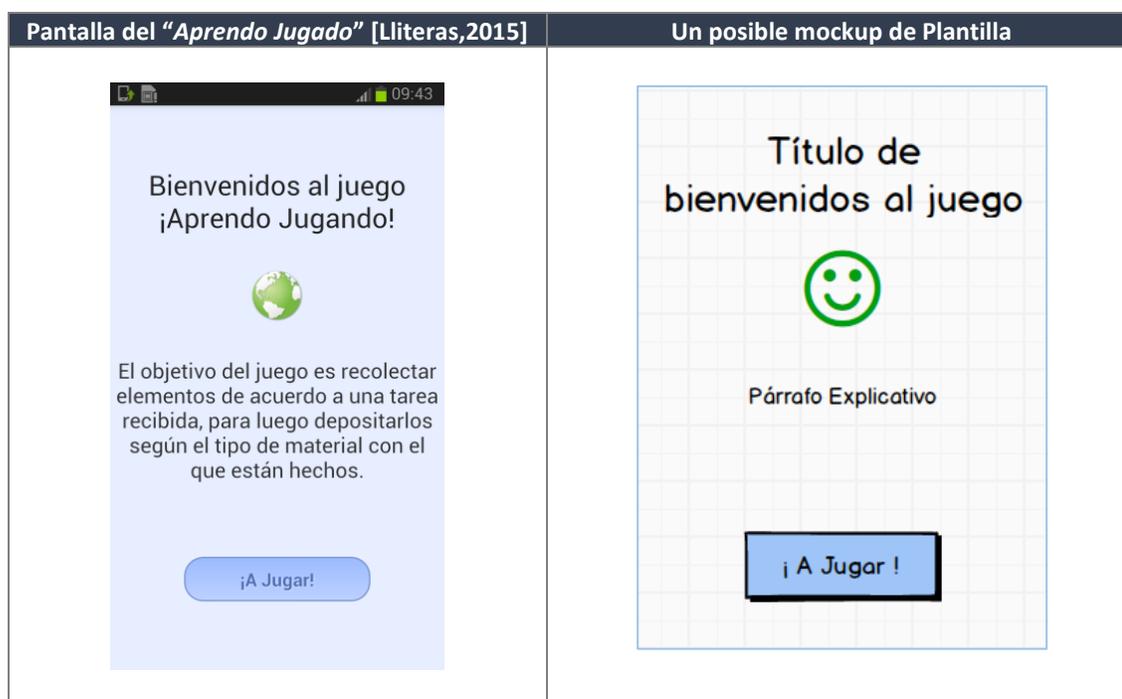


Figura 3.3: Pantalla de Bienvenida/Inicio de Juego del *Aprendo Jugando* y un posible mockup de plantilla.

➤ *Indicación para llegar a la posición de una Tarea*

Luego de la bienvenida, el “*Aprendo Jugando*”, brinda la indicación de que se debe seguir un mapa, y al llegar al lugar, se debe leer el código QR de dicha tarea. Esto se puede apreciar en la parte izquierda de la Figura 3.4.

En base a esta forma de indicación de asistencia en la movilidad brindada por el “*Aprendo Jugando*”, se podría identificar un posible mockup de plantilla que especifique: “*Párrafo indicativo para llegar al Pol*”, “*Mapa con el recorrido a un Pol*”, “*Texto explicativo para la lectura del código QR párrafo Explicativo*” y “*Botón de Lectura Código QR*”. Esta plantilla sería editable para ajustarse a para las necesidades de cada aplicación. Sin embargo, como esta pantalla presenta información general, la plantilla ya podría contar con todo esta información definida, como se puede apreciar en la parte derecha de la Figura 3.4. De esta manera, si un docente la usa (desde una herramienta de creación), se le agiliza el proceso de creación dado que solo tiene que hacer modificaciones si necesita algún nivel de detalle distinto en los textos. El mapa podría ser tomado automáticamente acorde al lugar donde se está creando la información posicionada.

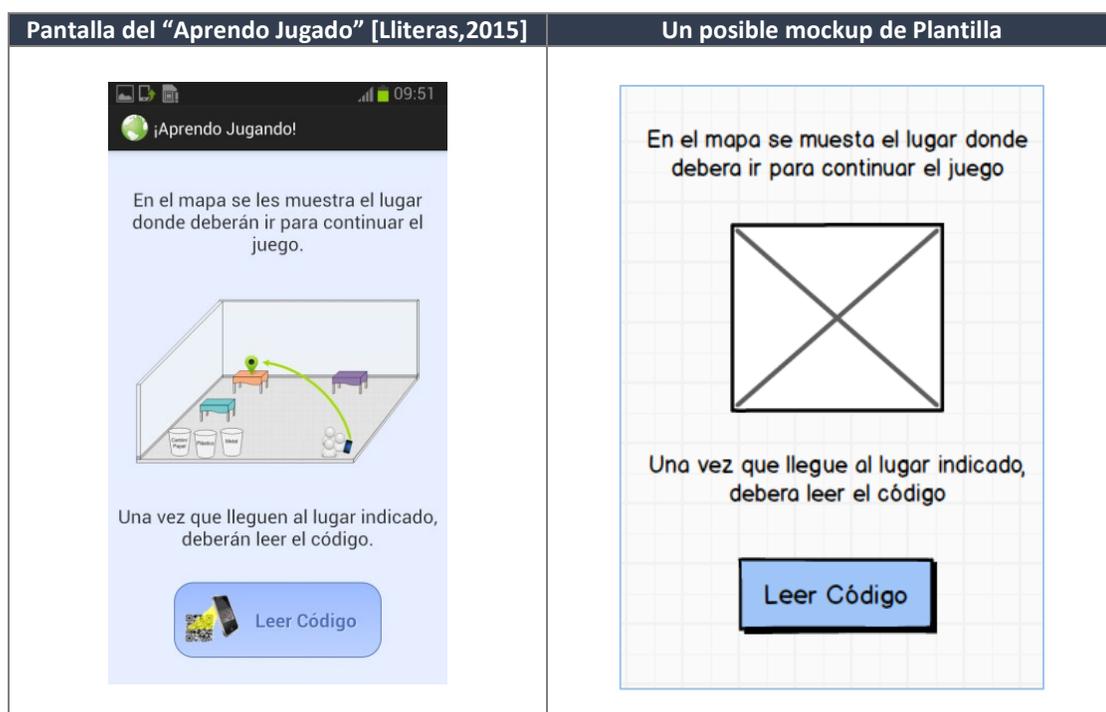


Figura 3.4: Indicación para llegar a la posición de una Tarea.

La plantilla tiene sentido en este caso, porque se utilizan códigos QR, en el caso de contar con otros mecanismos de sensado de posiciones esto podría variar. Por ejemplo, en el caso de usar GPS, solo hay que mostrarle la indicación y el mapa, dado que al arribar al lugar la tarea se podría brindar automáticamente sin requerir que el alumno haga ninguna acción.

➤ **Consigna de Tarea**

Una vez que el alumno llega a la posición de la tarea y lee el código QR de la misma, el “*Aprendo Jugando*”, brinda la tarea asociada, un ejemplo es mostrado en la parte izquierda de la Figura 3.5. Se puede apreciar que en este caso, al tratarse de una tarea de recolección se le indica mediante un botón que debe ir recolectando elementos (esto será mediante la lectura del código QR de cada elemento). Y se le da la opción de dar por finalizada una tarea.

Acorde a esto, se podría identificar un posible mockup de plantilla que especifique: “*Párrafo de la consigna*”, “*Botón de acción de tarea*”, “*Botón de finalización de tarea*”. Esta plantilla sería editable para ajustarse a las necesidades de cada aplicación. Sin embargo, como esta pantalla es representativa de cualquier tarea de recolección, la misma podría ser definida como se muestra en la parte derecha de la Figura 3.5. De esta manera, el docente solo tendría que definir el texto de la consigna de recolección.

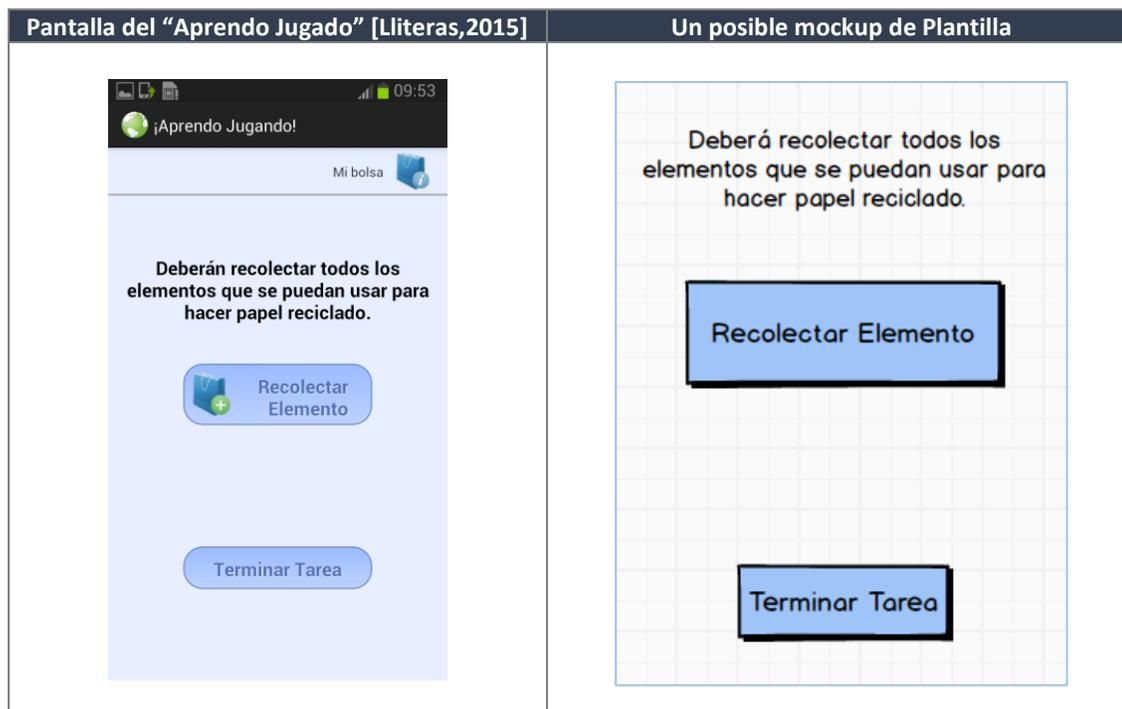


Figura 3.5: *Consigna de una Tarea de recolección.*

La plantilla especificada en la Figura 3.5 es útil cuando se trabaja con tareas de recolección, en el caso de una tarea con una pregunta de opción múltiple, la plantilla variaría. Acorde a esto, se puede observar que se podría contar con tantas plantillas como tipos de consignas se pudieran plantear como parte de una tarea.

➤ *Elección de como continuar luego de una tarea finalizada*

Una vez que el alumno selecciona el botón mostrado en la pantalla de la izquierda de la Figura 3.5, pueden suceder que haya más tareas para resolver, o sea la última tarea. A continuación se mostrarán estos dos casos:

- *Hay más tareas para realizar*

En este caso, el “*Aprendo Jugando*”, brinda la posibilidad de elegir cómo seguir, si ir a depositar lo recolectado o ir a la siguiente tarea, como se muestra en la parte izquierda de la Figura 3.6. Se podría identificar un posible mockup de plantilla que especifique: “*Párrafo indicando que ha finalizado la tarea*”, “*Texto explicativo indicando la decisión que debe realizar*”, “*Botón de Ir a depositar*” “*Botón Ir a la Siguiente Tarea*”. Esta plantilla sería

editable para ajustarse a las necesidades de cada aplicación. Sin embargo, como esta pantalla también es representativa de cualquier tarea de recolección, la misma podría ser definida como se muestra en la parte derecha de la Figura 3.6. De esta manera, el docente podría reusarla así como esta o editar el texto en el caso de ser necesario.

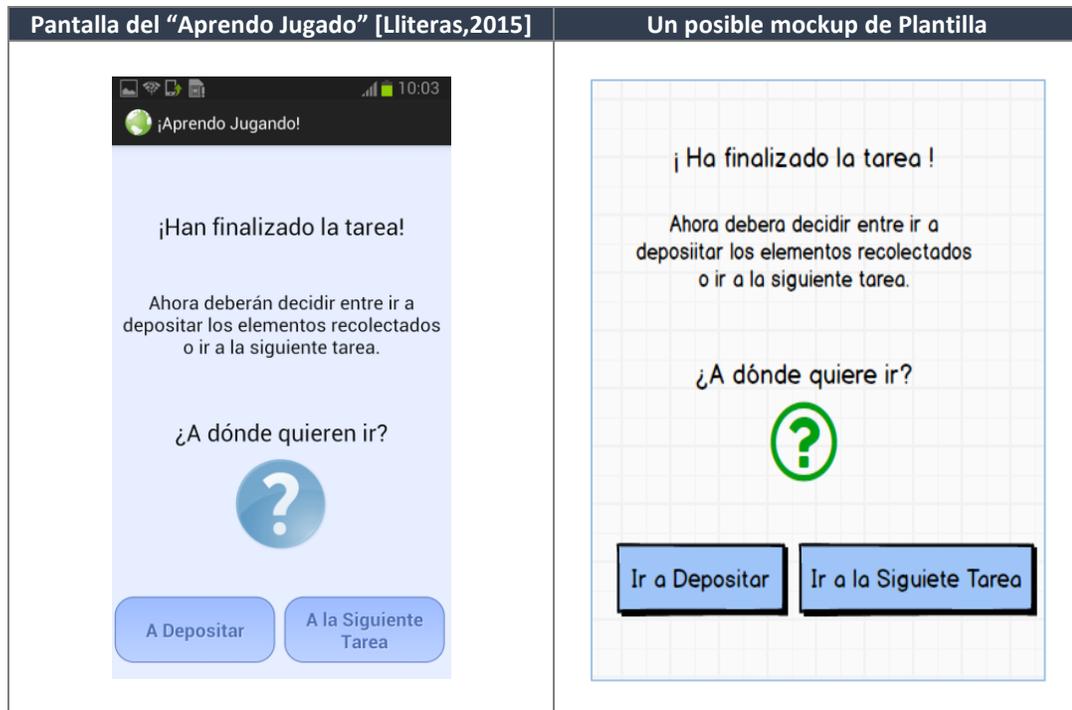


Figura 3.6: Finaliza una *Tarea de recolección* y hay más tareas.

Cabe destacar que el ejemplo de la Figura 3.6 es representativo de las tareas de recolección, sin embargo en el caso de tratarse de una tarea con una consigna de opción múltiple (u otro tipo de consigna), podría solo tener que especificar el botón de “*Ir a la Siguiente Tarea*”.

Sin importar cual opción elija de la pantalla de la izquierda de la Figura 3.6, en ambos casos se recibe una pantalla como la mostrada en la parte izquierda de la Figura 3.4 (el mapa de a dónde debe dirigirse).

- *Finalizo la última tarea*

En este caso, el “*Aprendo Jugando*”, brinda la posibilidad de elegir cómo seguir, optando por ir a depositar lo recolectado o finalizar el juego, como se muestra en la parte izquierda de la Figura 3.7. En este caso, se podría identificar un posible mockup de plantilla que especifique: “*Párrafo indicando que ha finalizado la tarea*”, “*Texto explicativo indicando la decisión que debe realizar*”, “*Botón de Ir a depositar*” “*Botón Finalizar Juego*”. Como pasaba con el caso de la Figura 3.6, como esta pantalla también es representativa de cualquier tarea de recolección, la misma podría ser definida como se muestra en la parte

derecha de la Figura 3.7. De esta manera, el docente podría reusarla así como esta o editar el texto en el caso de ser necesario.



Figura 3.7: Finaliza la última *Tarea de recolección*.

Al seleccionar el botón "*Finalizar Juego*" de la pantalla de la izquierda de la Figura 3.7, sale la pantalla mostrada en la Figura 3.8, donde se puede apreciar que se le indica que se finalizó el juego, se puede apreciar que eso podría ser representado con una generalización de la plantilla presentada en la Figura 3.7.

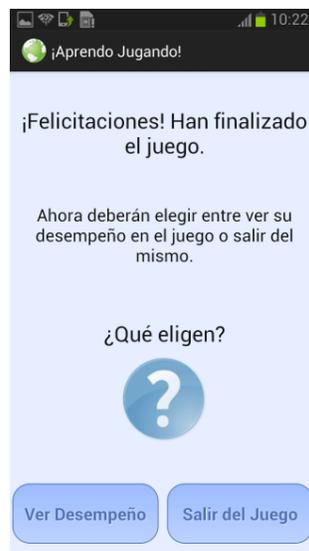


Figura 3.8: Pantallas asociada a la finalización del juego [Lliteras, 2015].

Acorde al análisis realizado anteriormente del prototipo “*Aprendo Jugando*” podemos identificar distintos plantillas relacionados con las Actividades Educativas Posicionadas. Sin embargo algunas tienen que ver con características relacionadas a los contenidos posicionados, otras a la asistencia, y otras son una continuación de algo posicionado. Acorde a esto podemos identificar:

- *Plantilla de bienvenida*, como se mostró en la Figura 3.3. Esta plantilla va a estar asociada a una posición en particular que es donde es el inicio de la actividad educativa.
- *Plantillas de asistencia a la movilidad*, en particular, mostrar el mapa para ir a un Pol, cuando el mecanismo de sensado usado es códigos QR (como se mostró en la Figura 3.4). Esta plantilla podría estar, por ejemplo, asociada a la flecha que une contenidos posicionados.
- *Plantillas de Tareas posicionadas*, por ejemplo, una tarea de recolección (como se mostró en la Figura 3.5). Es decir, que esta plantilla va a estar asociada a una posición en particular. En relación directa con estas plantillas, se dan las plantillas de elección de como continuar luego de una tarea finalizada, cuando:
 - *Hay más tareas para realiza* (como se mostró en la Figura 3.6)
 - *Finalizo la última tarea* (como se mostró en la Figura 3.7)

Para clarificar con que se asocia cada una de las plantillas mencionadas se muestra en la Figura 3.9 un ejemplo indicando a que elemento está asociado cada una de estas planillas.

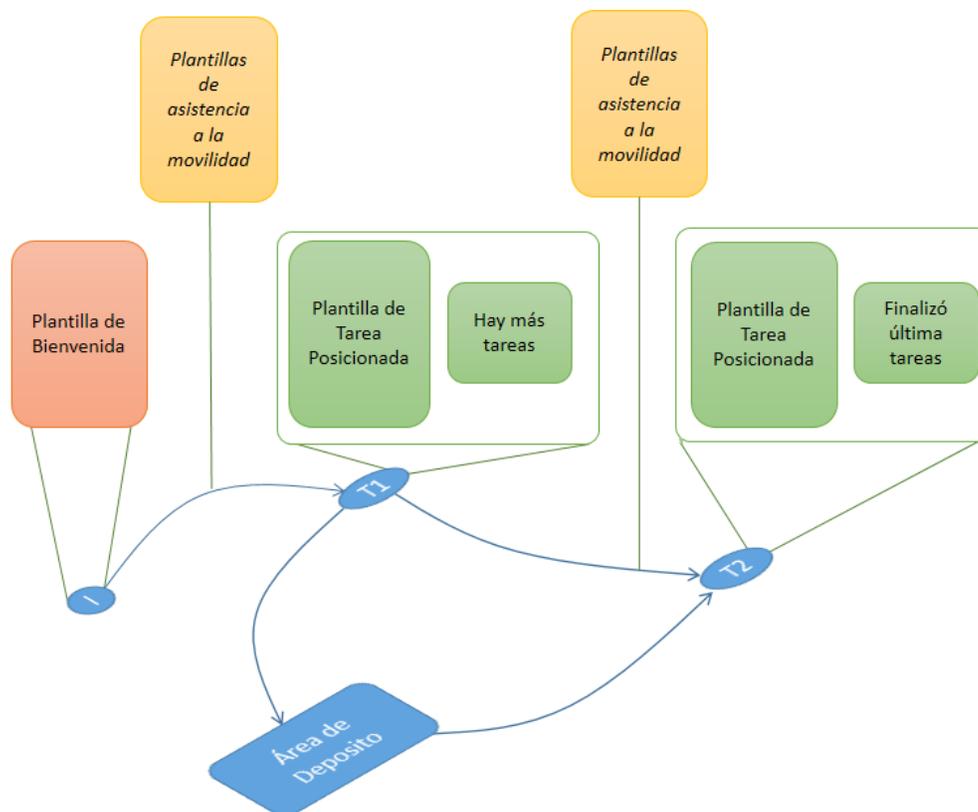


Figura 3.9: Bosquejo mostrando las plantillas identificadas del análisis del “*Aprendo Jugando*”.

Se puede apreciar de este análisis, que tanto las plantillas de las Figuras 3.6 y 3.7, solo tienen sentido al estar asociadas a la plantilla de la tarea posicionada. Acorde a esto, en un diseño inicial de la herramienta propuesta en esta tesina solo haremos hincapié en las tres tipos de plantillas principales: *Plantilla de bienvenida*, *Plantillas de asistencia a la movilidad* y *Plantillas de Tareas posicionadas*.

Mientras que las *Plantilla de bienvenida* y *Plantillas de Tareas posicionadas* están asociados a contenidos posicionados, la *Plantillas de asistencia a la movilidad* está asociada a la flecha que une dos contenidos posicionados.

Acorde a esto, de la herramienta definida en [Alconada Verzini, 2015] se incorporaran estas plantilla para poder definir estas como tipos particulares de contenidos posicionados. Y además, se incorporará a la fecha definida en [Alconada Verzini, 2015] la posibilidad de incorporar la *Plantillas de asistencia a la movilidad*.

En el siguiente capítulo se describe como se realizó la extensión a la herramienta presentada en [Alconada Verzini, 2015].

4. Herramienta propuesta

Como se mencionó en el objetivo de esta tesina se va a extender la herramienta presentada en [Alconada Verzini, 2015], la cual fue descrita y comparada con herramientas similares en el Capítulo 2; en dicho capítulo se pudo apreciar que esta herramienta contaba con las siguientes ventajas:

- *Desacopla el contenido de las posiciones*
- *Está destinada tanto a espacios indoor-outdoor*
- *Permite la construcción in-situ*

Por otro lado, en el Capítulo 3 se analizaron las características de las *Actividades Educativas Posicionadas*. Identificando que la herramienta definida en [Alconada Verzini, 2015] debe ser extendida con la siguiente funcionalidad para brindar:

- Al crear contenido posicionado, la posibilidad de optar por:
 - *Plantilla de Bienvenida ó*
 - *Plantillas de Tareas posicionadas*Pudiendo además editar las mismas.
- Al crear una flecha entre dos contenidos posicionados, poder indicar cuál será la *Plantilla de asistencia a la movilidad* asociada.

Por otro lado, mientras se analizaba para el Capítulo 3, el prototipo “*Aprendo Jugando*”, se pudo detectar que el mismo no requiere conectividad, facilitando así el uso en las pruebas con alumnos que se mencionan en [Lliteras, 2015]. Acorde a esto, se puede ver que la conectividad podría ser una dificultad, por ejemplo, si el docente quiere probar una herramienta como la propuesta en [Alconada Verzini, 2015], la cual es puramente Web.

Para poder brindar una herramienta que sea instalable en el dispositivo móvil, y además para espacios indoor no requiera conectividad. Y aprovechando todo el conocimiento adquirido en el trabajo descrito en [Zimbello and Challiol, 2016] en relación a *PhoneGap*¹¹. Se decidió embeber la herramienta presentada en [Alconada Verzini, 2015], en un entorno híbrido como es *PhoneGap*.

Esto en principio es viable, porque *PhoneGap* se basa en tecnología Web estándar, como es HTML5, *Javascript* y CSS. Sin embargo, al hacer la migración se detectaron algunos problemas tanto con las librerías de *Javascript* que usaba la herramienta [Alconada Verzini, 2015], como así también con las hojas de estilos (CSS) presentes. Estas librerías no se comportaban de la misma manera que en un navegador Web tradicional, debido a los distintos eventos que cambian en los dispositivos móviles. Por ejemplo, el `onClick()` deja de tener sentido en los dispositivos móviles. En la Sección 4.1 se detallan los problemas encontrados y como fueron resueltos para poder contar con la funcionalidad de la herramienta descrita en [Alconada Verzini, 2015] pero en nuestro caso embebida en *PhoneGap*.

¹¹ Página de *PhoneGap*: <https://phonegap.com> (Último acceso 06-08-2017)

En el caso de que el docente solo requiera definir tareas posicionadas en un espacio indoor, no requerirá ningún tipo de conectividad. Sin embargo, en el caso de definir tareas posicionadas en un espacio outdoor, requerirá conectividad para acceder a mapas externos, por una cuestión de simplicidad para esta primera versión de la herramienta propuesta en esta tesina no se dejan bajados previamente los mapas outdoor.

4.1 Problemas y soluciones relacionadas a la extensión propuesta

Como se mencionó en anteriormente, al realizarse la extensión de la herramienta elegida [Alconada Verzini, 2015] se detectaron problemas inesperados, por lo cual el desarrollo involucro dos etapas: una primera etapa (*Etapa I*) en la cual se realizaron los cambios necesarios para mantener la funcionalidad existente y una segunda etapa (*Etapa II*) donde se agregó la nueva funcionalidad, las cuales se presentan en la Sección 4.2.

A continuación se analizaran los problemas encontrados y las soluciones propuestas e implementadas para la Etapa I:

❑ **Hojas de estilo (CSS): Necesidad de un nuevo diseño Responsive**

Problemática:

Si bien la herramienta elegida [Alconada Verzini, 2015] contaba con un estilo *Web Responsive*, es decir, un diseño adaptable al tipo de pantalla [Frain, 2012]. Fue necesario realizar cambios para que se adaptaran correctamente a los dispositivos móviles, ya que en mucho de los casos la usabilidad de la aplicación se veía afectada, por ejemplo, los menú, botones tenían un tamaño que en mucho de los casos ocupaban gran espacio de la pantalla, dando esto lugar, a la perdida de foco o visualización por parte del usuario del mapa, el cual es uno de los elementos fundamentales de la aplicación.

Solución:

Se optó por utilizar un *template* de código libre basado en *Bootstrap*¹². Esto no solo permitió acomodar el tamaño de los elementos sino también la disposición de los mismos en las diferentes pantallas de la herramienta.

❑ **JqueryMobile: Necesidad de una considerar una nueva librería**

Problemática:

Al momento de migrar el código de la herramienta a una aplicación embebida. Nos encontramos que contar solo con la librería *jQuery*¹³, no era suficiente para el desarrollo propuesto en esta tesina, ya que para varias de las funcionalidades preexistentes y ha desarrollar, era necesario considerar, además, una librería que

¹² Página de *Bootstrap*: <https://getbootstrap.com> (Último acceso 16-7-2018)

¹³ Librería JavaScript que permite la manipulación, el manejo de eventos, de una manera más sencilla. Página de *jQuery*: <https://www.google.com/search?q=jQuery&ie=utf-8&oe=utf-8&client=firefox-b-ab> (Último acceso 16-7-2018)

tuviera un enfoque en las plataformas móviles, por ejemplo, para considerar nuevos eventos relacionados al manejo táctil del dispositivo.

Solución:

Como solución a esto, se utilizó una nueva librería *JqueryMobile*¹⁴, siguiendo el enfoque presentado en [Zibula and Majchrzak, 2012], en donde se construye una aplicación móvil embebida donde se utilizan las tecnologías HTML5, *PhoneGap*, tal como las propuestas para el desarrollo de la herramienta extendida presentada en esta tesina.

Esto no solo implicó la adición de nuevas referencias en los HTML, sino también cambios en la implementación del código legacy existente.

❑ ***JsPlumb el manejo de los eventos para la creación y/o modificación de las conexiones entre los Pol de la aplicación***

Problemática:

La herramienta de la que se extiende [Alconada Verzini, 2015], hace uso de una librería denominada *JsPlumb*¹⁵ que permite la construcción de flechas que representan la conexión de los distintos puntos de interés. Al migrar el código que permitía la creación de estas denominadas “*Conexiones*” la funcionalidad no se comportaba como se esperaba dentro de *PhoneGap*. Esto se debía a que gran parte de la implementación de este código estaba ligada a eventos de interacción del usuario por medio del mouse. Estos eventos son de UI y extienden de los “*DOM Event Objects*” definidos en el DOM. Los Eventos UI son aquellos típicamente implementados para el manejo de la interacción de lo usuario como son la entrada de mouse y teclado.¹⁶

Solución:

Distintas soluciones fueron probadas para esta problemática:

- Utilización de la librería *jQuery UI Touch Punch*¹⁷ esta librería convierte los eventos asociados al touch a los correspondientes eventos asociados al mouse. Si bien esta solución parecía prometedora, no siempre funcionaba de manera correcta; es decir, que en las pruebas con distintos dispositivos móviles se encontraba que la funcionalidad no siempre se comportaba de la manera esperada.
- Refactorización del código implementado en la herramienta existente que hacía uso de la librería *JsPlumb* mencionada anteriormente, con el fin de actualizar está con los últimos features presentes. Al realizar esto, se detectó que los

¹⁴ Página de JqueryMobile: <https://jquerymobile.com> (Último acceso 16-7-2018)

¹⁵ Página de *JsPlumb*: <https://jsplumbtoolkit.com> (Último acceso 16-7-2018)

¹⁶ Interpretación de la definición de UI events: <https://www.w3.org/TR/uevents/#events-UIevents> (Último acceso 16-7-2018)

¹⁷ Página de *jQuery UI Touch Punch*: <http://touchpunch.furf.com> (Último acceso 16-7-2018)

componentes a partir de los cuales se generan las flechas están posicionados adentro de un componente mapa, el cual utiliza distintos valores de márgenes, por ejemplo, respecto a una componente HTML simple. La solución existente tenía modificaciones a nivel de la librería, que utilizaba para solventar esto. Como esto involucraba un desarrollo de cero, y cómo no se podía determinar que esto funcione correctamente, esta opción no fue considerada.

- Utilización de la librería *Leaflet Polyline* y *Leaflet PolylineDecorator*, que permiten crear líneas con punta de flechas sobre el mapa. Para poder dibujar estas líneas, es necesario contar con las posiciones actualizadas de los Pol y conocer que puntos desea conectar el usuario de la aplicación. Con este fin, los Pol fueron representados con diferentes colores y letras del abecedario permitiendo de esta manera que sean identificados para cuando desean conectarse.

En conclusión, podemos decir que fue necesario reimplementar esta funcionalidad de la herramienta existente para poder embeberla en la aplicación móvil, acorde a la última opción mencionada.

❑ **Mapas Indoor: Carga de imágenes**

Problemática:

En [Alconada Verzini, 2015] una de las funcionalidades que se presentaba y que resultaba de interés mantener es la de permitir definir Pol tanto en mapas indoor como outdoor. Para esto, es necesario seguir exponiendo la acción de carga de imágenes sobre los mapas geolocalizados. En *eBuilder* se utiliza la librería *Dropzone.js* que presenta una forma cómoda de subir archivos mediante *Drag and Drop*. Dado que el enfoque de *eBuilder* solo era para proyecto Web, se necesitó implementar para la extensión una solución que permitiera cargar nuevas imágenes desde el dispositivo móvil en donde la aplicación fuera utilizada.

Solución:

Se utilizó el plugin de *PhoneGap ImagePicker*¹⁸. Los plugins de *PhoneGap* son aquellos que permiten que la aplicación se comunique con la plataforma nativa en la que se ejecuta. Y proveen el acceso a las funcionalidades y la plataforma del dispositivo móvil, lo cual normalmente no está disponible para las aplicaciones *Web-based*.

En consecuencia agregar el plugin de *PhoneGap ImagePicker* implicó realizar cambios en el código legacy del *eBuilder*.

❑ **Manejo de Permisos**

Problemática:

¹⁸ Página de *PhoneGap ImagePicker*: <https://www.npmjs.com/package/cordova-plugin-imagepicker> (Último acceso 16-7-2018)

Mantener las funcionalidades de:

- Permitir cargar mapas indoor (ver problemática *Carga de Imágenes para mapas indoor*) y
- Permitir crear posiciones en la posición actual del usuario.

Dado que *eBuilder* está implementado como una aplicación Web, varios aspectos fueron modificados para el correcto funcionamiento de estas características.

Solución:

Cuando se desarrolla una aplicación Android, una de las primeras consideraciones que se deben tener es el manejo de permisos.

Una aplicación básica de *Android* no tiene permisos asociados de manera predeterminada. Esto significa que no puede hacer nada que afecte negativamente la experiencia del usuario o los datos en el dispositivo. Para usar funciones protegidas del dispositivo, es necesario incluir una o más etiquetas en el `Android-Manifest.xml`¹⁹ El comportamiento de estos permisos varía de acuerdo a la versión del sistema operativo *Android* donde se vaya a instalar la aplicación; es decir, si el dispositivo tiene instalado *Android 6.0* (nivel de API23) o versiones posteriores y el atributo `targetSdkVersion` de la app es 23 o superior, la app solicita los permisos al usuario en el tiempo de ejecución. Sin embargo, si el dispositivo tiene instalado *Android 5.1* (nivel de API 22) o versiones anteriores, o el atributo `targetSdkVersion` de la app es 22 o un valor inferior, el sistema solicita al usuario que otorgue los permisos al instalar la aplicación.

De acuerdo, a lo mencionado anteriormente fue necesario agregar los siguientes permisos:

- Permisos relacionados con la ubicación:

```
android.permission.ACCESS_LOCATION_EXTRA_COMMANDS
android.permission.ACCESS_COARSE_LOCATION
android.permission.ACCESS_FINE_LOCATION
```

- Permisos relacionados con el acceso a la galería/almacenamiento del dispositivo:

```
android.permission.WRITE_EXTERNAL_STORAGE
```

❑ **Adaptación de la creación de contenidos**

Problemática:

Esta problemática tiene dos enfoques, uno de ellos es la implementación a nivel código, ya que esta funcionalidad en su totalidad estaba implementada con librerías con enfoque para desarrollo Web, y por otro lado, un enfoque desde lo conceptual,

¹⁹ Página de *Android-Manifest*: <https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es-419> (Último acceso 16-7-2018)

dado que la perspectiva cambia con la incorporación de la funcionalidad de *templates*, los cuales se presentarán en detalla en la Sección 4.21 Acorde a esto, los contenidos serán también visualizados como partes de los *templates*.

Solución:

Para la problemática de implementación a nivel código, se reimplementó la funcionalidad por completo, dejando como opción la creación de preguntas con una respuesta, preguntas con *multiple choice* y el contenido imagen que se desarrolla utilizando el mismo plugin *PhoneGap* mencionado en la descripción de la problemática de “*Mapas Indoor: Carga de imágenes*”.

Para el nivel conceptual de esta problemática, se optó por agregar una funcionalidad que también se encuentra relacionada con la mencionada creación de “*Templates*”, donde para este caso en particular se permite crear un *template* blanco en el cual se pueden agregar tanto componentes a un *template*, como contenidos previamente creados.

4.2 Funcionalidad de la herramienta propuesta en esta tesina

Tomando como base lo mencionado en los capítulos anteriores, y la herramienta existente [Alconada Verzini, 2015], se desarrolló una Aplicación Móvil prototípica, denominada “*Herramienta de autor*” que presenta las funcionalidades preexistentes en *eBuilder* (acorde a lo mencionado en la Sección 2.4), como así también funcionalidades nuevas tales como la creación de *templates*, asociación de estos a puntos de interés o a la conexión entre los mismos.

Además, se agregó la posibilidad de exportar en formato JSON toda la información necesaria para crear una Aplicación Educativa Móvil basada en posicionamiento construida mediante la herramienta propuesta, como así también el guardado de los *templates* mencionados para complementar esta creación. Cabe mencionar que la herramienta existente [Alconada Verzini, 2015] solo exportaba a XML las capas de contenidos, posiciones y elementos posicionados.

A continuación se explican las características en detalle de la herramienta presentada en esta tesina, su funcionamiento y sus modos de uso.

Teniendo en cuenta el análisis realizado en el Capítulo 3, y el menú presentado en la herramienta de la cual extiende esta tesina, al iniciar la aplicación móvil “*Herramienta de Autor*” aparece un menú principal el cual se puede apreciar en la Figura 4.1, donde se presentan cuatro acciones principales posibles: *Creación de Elementos Educativos Posicionados*, *Creación de Posiciones*, *Creación de Contenidos y Visualización* y *Creación de Templates*, como así también una barra superior con distintas funcionalidades.

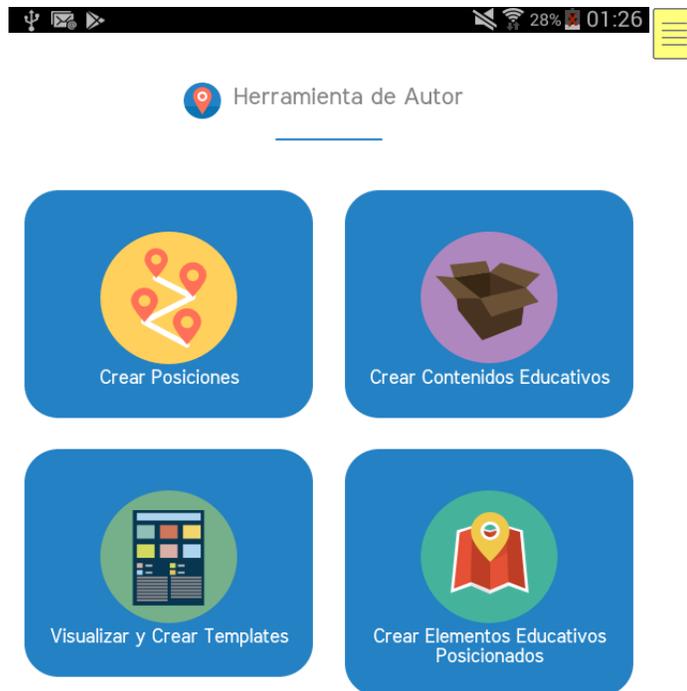


Figura 4.1: Menú de Inicio de la Aplicación Móvil prototípica “Herramienta de Autor”.

A continuación se describirá en detalle cada opción presentada en la Figura 4.1, explicando cómo es su funcionamiento como así también se detallarán algunos conceptos que incorpora la herramienta propuesta en esta tesina.

❑ Barra Superior de acciones

En la Figura 4.1 se puede apreciar la barra superior que se presenta en la herramienta, dicha barra va mostrando distintas funcionalidades dependiendo de las acciones que va eligiendo el usuario. Por ejemplo, cuando el usuario realiza las acciones correspondientes a la *Creación de Posiciones* o *Creación de Elementos Educativos Posicionados* se muestran opciones diferentes las cuales se pueden apreciar en las Figuras 4.2.a y 4.2.b respectivamente. Se puede apreciar que hay opciones comunes pero también opciones acordes a la opción seleccionada, ya sea *Creación de Posiciones* o *Creación de Elementos Educativos Posicionados*.



Figura 4.2: Barra Superior de acciones.

Veamos a continuación las opciones comunes de la barra superior de las Figuras 4.2.a y 4.2.b. En la Figura 4.3 se puede apreciar el *icono de limpieza*, este se encuentra representado por una goma de borrar. Cuando el usuario selecciona esta opción deshace todo lo creado hasta el momento. Es decir, en el caso de, por ejemplo, estar creando solo posiciones, se borrarán todas las posiciones que el usuario haya definido.



Figura 4.3: Barra Superior de Acciones - Opción de Limpieza.

Otra opción común en la barra superior de las Figuras 4.2.a y 4.2.b es el icono del menú, el cual se puede apreciar en la Figura 4.4. Como se puede apreciar en dicha figura el mismo está representado por una circunferencia azul con tres líneas horizontales blancas en su interior. Esta opción se ubicara siempre a la derecha de las pantallas definidas por la herramienta. Este menú es contextual las opciones que se muestren cuando sea seleccionado varía dependiendo de lo que el usuario está haciendo.



Figura 4.4: Barra Superior de Acciones - Opción de Menú.

En la Figura 4.2.a se puede apreciar además el icono de Descarga/Exportación, el cual se muestra en la Figura 4.5; este icono está representado con una flecha, cuando el usuario/docente seleccione esta opción descargara en formato JSON lo que haya definido en la herramienta. Para esto, se presentará un modal que mostrará el JSON a descargar y se pedirá la confirmación del usuario. Cabe aclarar que en caso de no haber definido nada aún se descargará un JSON vacío.



Figura 4.5: Barra Superior de Acciones - Opción de Descarga/Exportación.

Por otro lado, en la Figura 4.2.b se puede observar el icono de Guardado, el cual se muestra en la Figura 4.6. Este icono aparecerá cuando se va creando y también cuando está previsualizando un *template*. Como se puede apreciar en la Figura 4.6 esta opción se representa mediante un disquete que se encuentra en una circunferencia azul.



Figura 4.6: Barra Superior de Acciones - Opción de Guardado.

Hay otras opciones que van apareciendo en la barra superior de acciones y que no fueron visualizadas en las Figuras 4.2.a ni 4.2.b. Por ejemplo, cuando se está en modo *Creación de Contenidos Educativos o Creación de Templates* se muestra la opción de previsualización, el icono asociado a esta opción se puede apreciar en la Figura 4.7; el cual se representa mediante la imagen de un ojo que se encuentra en una circunferencia azul. Esto permite previsualizar lo creado antes de guardarlo.



Figura 4.7: Barra Superior de Acciones - Opción de Previsualización.

Otra opción que se muestra en la barra superior cuando se está realizando la *Creación de Contenidos Educativos* es *Capturar Pantalla actual*, esta opción permite capturar el mapa con los distintos contenidos educativos posicionados que el usuario está creando, y es utilizado en particular para cuando se asocia un *template* del tipo “*Indicación por medio de mapa*”. De esta manera el mapa se saca acorde a lo que ve el usuario en su pantalla. El icono asociado a la captura de pantalla se puede apreciar en la Figura 4.8. Cabe mencionar que la funcionalidad asociada a esta opción se describirá más adelante en detalle cuando se aborde la temática de *Creación de Elementos Educativos Posicionados*.



Figura 4.8: Barra Superior de Acciones - Opción de Capturar Pantalla.

❑ Selección de mapas

Cuando el usuario/docente seleccione las opciones *Crear Elementos Educativos* o *Crear Posiciones* de la Figura 4.1, recibirá la pantalla mostrada en la Figura 4.9 para la selección del mapa. Esta pantalla permite seleccionar el tipo de mapa con el cual se desea hacer las creaciones.

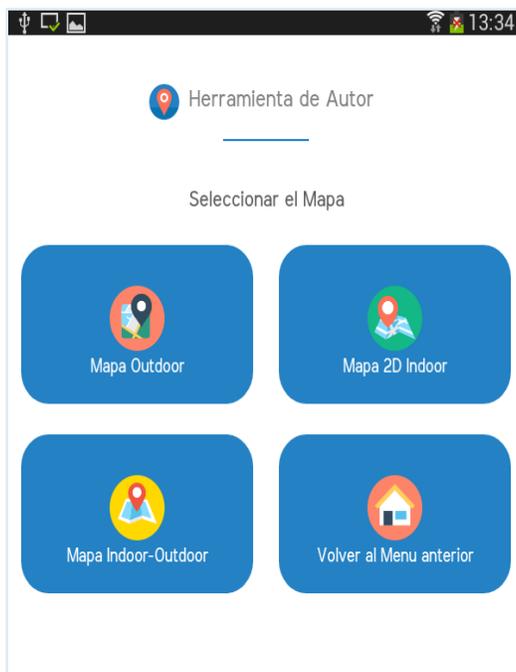


Figura 4.9: Menú de selección de Mapas.

Se puede apreciar que las opciones mostradas en la Figura 4.8 guarda relación con las opciones presentadas en la Figura 2.24 de la herramienta presentada en [Alconada Verzini, 2015].

A continuación se resumen las opciones que se despliegan en la Figura 4.8:

- *Mapa Outdoor*

Esta opción guarda relación con la opción “*Mapa Satelital*” de la herramienta [Alconada Verzini, 2015]. Al seleccionar esta opción se muestran mapas de *OpenStreetMap*²⁰. Asociado a esto se tendrá un mecanismo de sensado asociado a la herramienta (por ejemplo, GPS, 4G o WiFi), si el usuario acepta los permisos de posición al instalar la herramienta (aplicación), podrá ver un punto azul en el mapa que indicará la posición actual del usuario. Este punto azul se irá moviendo a medida que el usuario recorra el espacio físico.

- *Mapa 2D Indoor*

Esta opción permite al usuario cargar un mapa indoor que luego se usará para definir posiciones o *Elementos Educativos* posicionados. Para esta opción, el usuario debe tener un mapa guardado en la galería del dispositivo móvil en donde se está utilizando la aplicación. Este mapa puede estar guardado como una imagen.

- *Mapa indoor-outdoor*

Esta opción combina los dos tipos de mapas presentados previamente. Es decir, se toma los mapas de *OpenStreetMap* y además el usuario selecciona un mapa indoor de su galería, el cual se superpone en el mapa de *OpenStreetMap*, permitiendo marcar puntos sobre ambos mapas.

❑ Creación de Posiciones

En la Figura 4.1 se pudo apreciar la opción *Crear Posiciones*, esta opción funciona igual que la herramienta presentada en [Alconada Verzini, 2015]. Es decir, permite crear posiciones sin necesidad de asociar a la misma ningún tipo de contenido, ni en el caso de esta tesina, tampoco ningún *template*. Esto permite seguir manteniendo el concepto de separación en capas como se ha descrito en el Capítulo 3. Cabe mencionar que definir las posiciones de manera independiente, permitirá al docente identificar en que posiciones desea realizar la actividad educativa, sin importar el contenido que se encontrará asociado a estas. De esta manera, esta capa de posiciones podrá ser reusada.

Para poder crear una posición, el usuario debe elegir la opción *Crear Posiciones* de la Figura 4.1, ahí recibe una pantalla como la mostrada en la Figura 4.9, y luego debe elegir el mapa que será usado de base para definir las posiciones.

Supongamos que el usuario elige la opción “*Mapa Indoor-Outdoor*”, como respuesta recibe una pantalla similar a la Figura 4.10. Se puede apreciar que se muestra el mapa de

²⁰ Página de *OpenStreetMap*: <https://www.openstreetmap.org> (Último acceso 6-8-2017)

OpenStreetMap y las acciones que puede realizar en un menú de acciones en la parte inferior; las cuales van variando acorde al tipo de mapa elegido, en este caso, al ser un mapa *Indoor-Outdoor* se le presenta la opción de superponer un mapa indoor.



Figura 4.10: Menú de selección de Mapas.

Las opciones del menú inferior que se puede apreciar en la Figura 4.10 son, de izquierda a derecha: *Crear una Nueva Posición*, *Cargar un Mapa 2D Indoor* y *volver al menú anterior*.

Veamos cómo funciona la opción *Cargar un mapa 2D Indoor*, al seleccionar esta opción de la Figura 4.10 el usuario recibe una pantalla similar a la Figura 4.11; se puede apreciar que se abre la galería de imágenes del dispositivo móvil, y desde allí podrá elegir el mapa que desea superponer al mapa outdoor (*OpenStreetMap*).

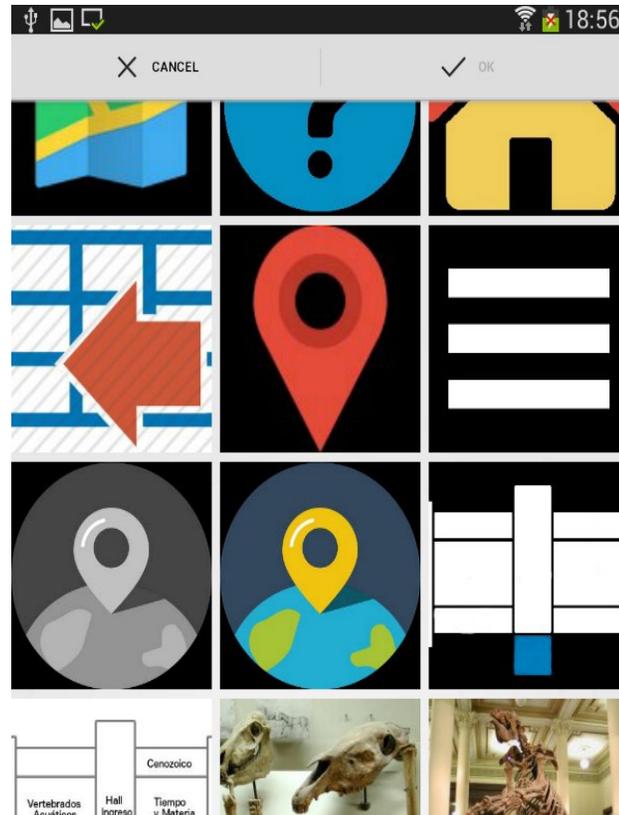


Figura 4.11: Selección de mapa de la galería.

Una vez cargada la imagen desde la Figura 4.11, ésta aparece superpuesta sobre el mapa de *OpenStreetMap* como se puede apreciar en la Figura 4.12. Si bien dicha imagen aparece en un lugar del mapa la misma se puede mover, agrandarla o reducirla.

Además, como se puede apreciar en la Figura 4.12 aparece un toolbox en la esquina superior derecha de la pantalla del mapa de *OpenStreetMap* que permite ajustar la opacidad de la imagen cargada, eliminarla o guardarla.

A modo de ejemplo en la Figura 4.12 se puede apreciar la imagen del primer piso del Museo de Ciencias Naturales La Plata, como se mencionó anteriormente está imagen debe estar previamente almacenada en el dispositivo móvil del usuario. Es decir, acorde a lo que quiera crear el docente, éste debe prever contar con los mapas de los lugares indoor en los cuales quiere definir posiciones o como se mostrará luego elementos educativos posicionados.

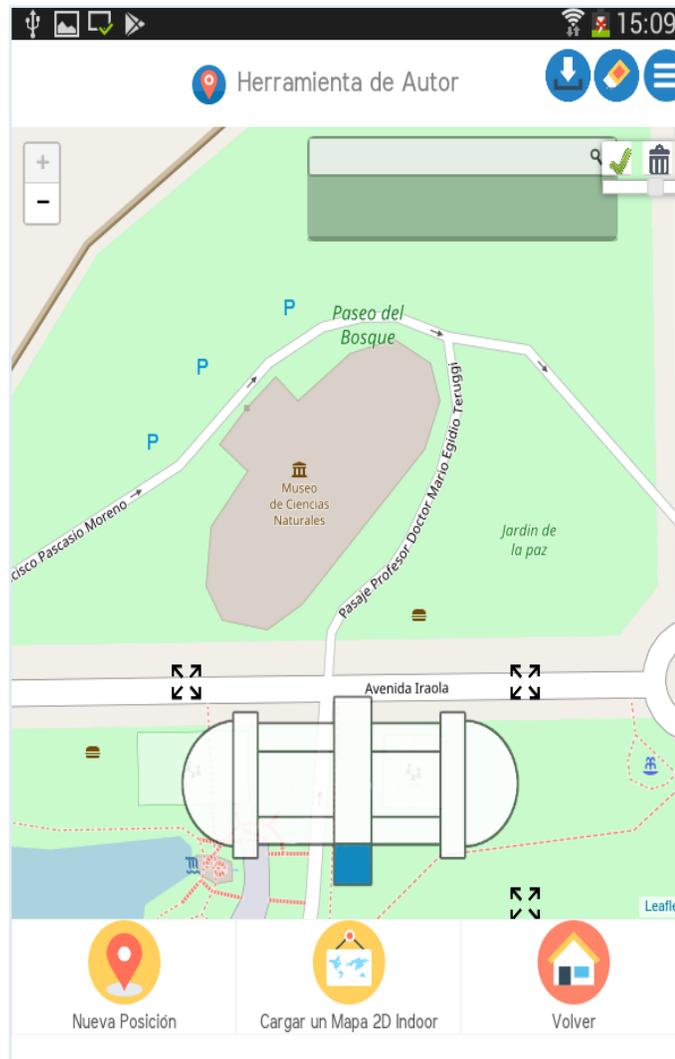


Figura 4.12: Mapa Indoor superpuesto con el mapa Outdoor.

Una vez que el usuario acomoda el mapa indoor 2D, podrá comenzar a definir posiciones, tanto en el interior como en el exterior de este del mismo. Para esto debe elegir la opción "*Nueva Posición*" que se visualiza en la Figura 4.12. Cuando selecciona esta opción, se crea una nueva posición como se puede apreciar en la Figura 4.13, para poder identificar cada una de estas posiciones las mismas son asignadas con un color y una letra del abecedario, las cuales como se mostrará más adelante pueden ser modificadas.

Cabe mencionar que el color y la letra son incorporaciones variando lo propuesto en la herramienta presentada en [Alconada Verzini, 2015], donde todas las posiciones tenían un mismo color y no tenían una forma de identificarlas visualmente. Esta incorporación permite un mejor manejo de las posiciones creadas para facilitar el prototipado dado que se debían conectar PoI de manera fácil. Podría ser que en un futuro esto sea modificado acorde a detectar alguna otra forma de ayudar mejor al docente a identificar las posiciones, dado que las letras podrían prestarse a confusión pensando que las mismas se asocian al tipo de

contenido que contienen como es el caso de los elementos posicionados presentados en [Alconada Verzini, 2015].



Figura 4.13: Creación de una nueva Posición.

Como se pudo observar en la Figura 4.13 junto con la nueva posición al tocarla aparecerá un icono representado con un “engranaje” que se mostrara del color correspondiente a la posición. Al seleccionar el icono de engranaje se abrirá el menú de acciones correspondiente. Dado que en este ejemplo solo se están creando posiciones al tocar sobre el icono mencionado se desplegará el menú que se puede observar en la Figura 4.14. Se puede apreciar que las opciones desplegadas en este caso son: *QR* (que permite asociar un QR a la posición), *Conectar* (permite asociar una posición a otra preexistente) y *Borrar* (que permite borrar dicha posición).



Figura 4.14: Menú de acciones asociada a una posición.

Al elegir la opción de QR de la Figura 4.14, se genera un QR y se asociará a ese punto. El usuario/docente podrá visualizar un cartel que le informará que el QR se generó exitosamente, y en consecuencia se asoció el QR; esto se puede observar Figura 4.15. Este QR luego es entregado al usuario para que lo pueda pegar en el lugar correspondiente.

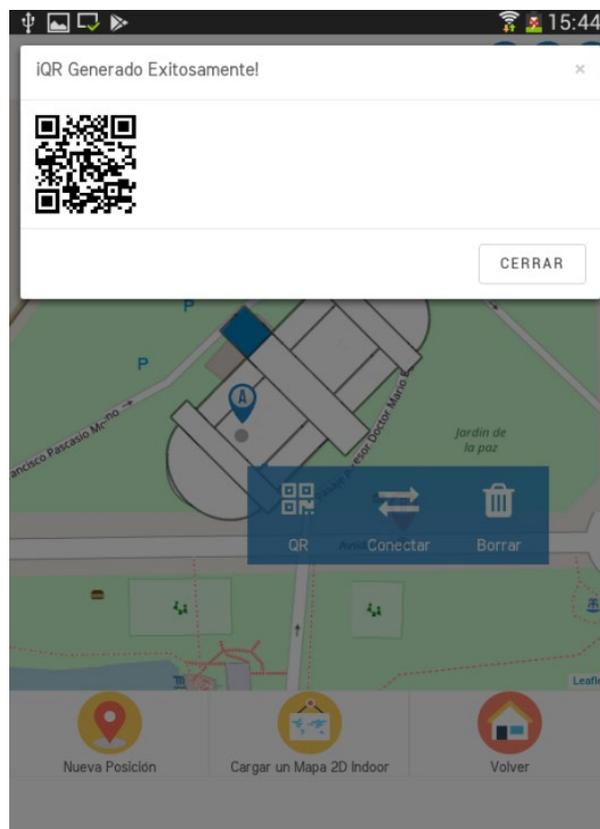


Figura 4.15: Creación de QR.

Para poder utilizar la función se *Conectar* mostrada en la Figura 4.14, se deben tener al menos dos posiciones creadas.

Supongamos que se crea otra posición como se visualiza en la Figura 4.16, donde se puede apreciar que se cuenta con una nueva posición, la cual tiene un color diferente y otra letra. Y se vuelve a seleccionar la primera posición creada; acorde a esto, aparece la opción del *engranaje* que permite realizar la acción de *Conectar* mostrada en la Figura 4.14. Como resultado de elegir esta opción (*Conectar*), el usuario recibe una pantalla similar a la Figura 4.17 donde puede seleccionar la posición con la cual se quiere hacer la posición.

La forma de contactar varía de lo propuesto en la herramienta presentada en [Alconada Verzini, 2015] donde se usaban flechas, este cambio se debe a que las librerías usadas en [Alconada Verzini, 2015] no funcionaban en *PhoneGap* como se describió en la Sección 4.1.



Figura 4.15: Mapa con dos puntos posicionados creados.

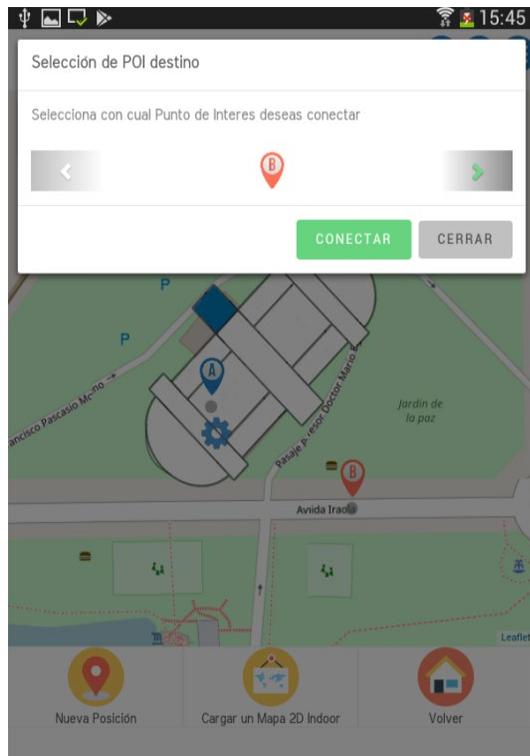


Figura 4.17: Realizar una conexión.

Una vez realizada la conexión en la Figura 4.17, el usuario recibe una pantalla similar a la Figura 4.18, donde se puede apreciar una flecha conectando ambas posiciones y la direccionalidad de dicha conexión.



Figura 4.18: Conexión entre dos posiciones.

De esta manera, el usuario/docente podrá definir todas las posiciones que desee, asignando si desea QR a las mismas, y conectando las posiciones mediante las flechas. Una vez que finalice podrá exportar las posiciones generadas para su posterior reuso.

❑ Visualización y Creación de Templates

Veamos ahora la funcionalidad asociada a la opción “Visualizar y Crear Templates” de la Figura 4.1. Al seleccionar esta opción se mostrará una pantalla similar a la Figura 4.19, donde se listan los templates existentes, en este caso asociados al “Aprendo Jugando” acorde a lo identificado en el Capítulo 3. Y además, la opción de crear un *template* blanco, el cual luego pueda ser utilizado en la creación de *Elementos Educativos Posicionados*.

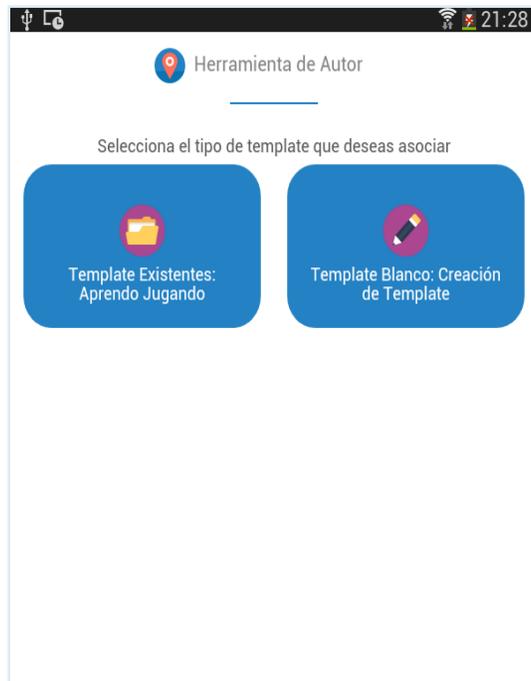


Figura 4.19: Visualización y Creación de *Templates*.

Al seleccionar la opción de “*Templates existentes: Aprendo Jugando*” de la Figura 4.19, el usuario/docente recibe la pantalla que se muestra en la Figura 4.20, donde se listan los *templates* definidos acorde al análisis realizado en el Capítulo 3.



Figura 4.20: *Templates* relacionados al *Aprendo Jugando*.

En la versión actual de la herramienta se permiten visualizar los *templates* creados, si bien se pueden editar los componentes de cada *template* no se puede modificar la estructura de los mismos. Es decir, por ejemplo, no se puede agregar otro texto o descripción a *templates* existentes.

Veamos a modo de ejemplo dos de los *templates* listados en la Figura 4.20 y luego la creación de un *template* blanco (cuando se selecciona dicha opción de la Figura 4.19).

➤ **Template de Bienvenida**

Como se puede apreciar en la Figura 4.21 en este *template* se cuenta con un título, una imagen representativa y una descripción. Esto es solo la visualización del *template* luego el mismo se puede usar cuando se crean tanto *Contenidos Educativos* como *Elementos Educativos Posicionados*.



Figura 4.21: Template de Bienvenida

Supongamos que el docente decide usar el *template* mostrado en la Figura 4.21 y desea editar el título, entonces deberá seleccionar el mismo, y al hacerlo, se visualizará el teclado del dispositivo móvil que permitirá ingresar el texto que se desee. Esto se puede apreciar en la Figura 4.22.

Algo similar ocurre para editar la imagen representativa del juego. Para esto se deberá tocar el icono del *template* y al hacerlo se abrirá una ventana que permitirá elegir entre un conjunto de imágenes definidas, esto se puede apreciar en la Figura 4.23.

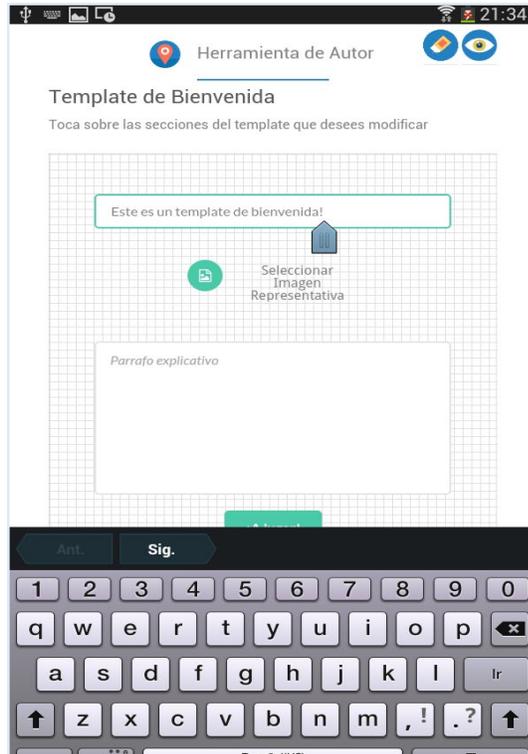


Figura 4.22: *Template* de Bienvenida.

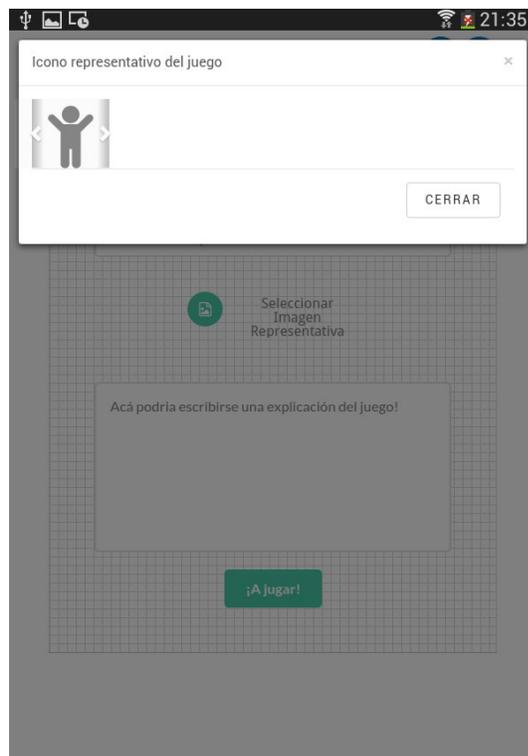


Figura 4.23: Edición de la imagen representativa *template* de Bienvenida.

Al elegir la opción previsualización del *template*, el docente recibirá una pantalla similar a la Figura 4.24 acorde a lo que fue editando.

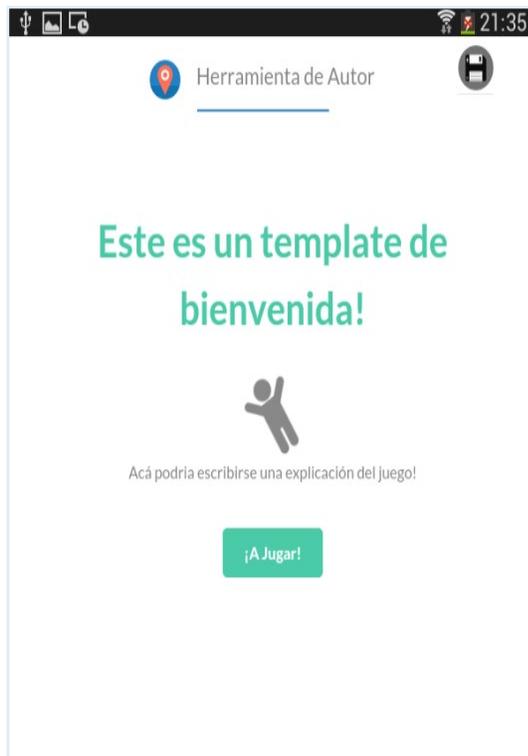


Figura 4.24: Previsualización del *template* de Bienvenida.

➤ **Template de Indicación por medio de un mapa**

Al seleccionar la opción “*Template de Indicación por medio de un mapa*” de la Figura 4.20, el usuario recibe un pantalla similar a la mostrada en la Figura 4.25. Cabe mencionar que este *template* se definió acorde al análisis realizado en el Capítulo 3.

Al igual que con el *template* de bienvenida, el usuario podrá ir modificando cada una de las secciones. Es decir, podrán definir los distintos textos de indicación, pero en particular, en este *template* podrá agregar una imagen ya sea de un mapa que representara como llegar de un punto a otro, o cualquier otro tipo de imagen que el usuario crea conveniente para indicar como llegar a determinado lugar.

Como parte de las características provistas por la herramienta presentada en esta tesina, el usuario podría elegir una imagen previamente capturada y recortada. Esta característica en particular se explicará más adelante en la temática de “*Creación de Elementos Educativos Posicionados*”.

Además, mientras va realizando la edición el usuario podrá ir visualizando el *template* para hacer los ajustes que crea necesarios.

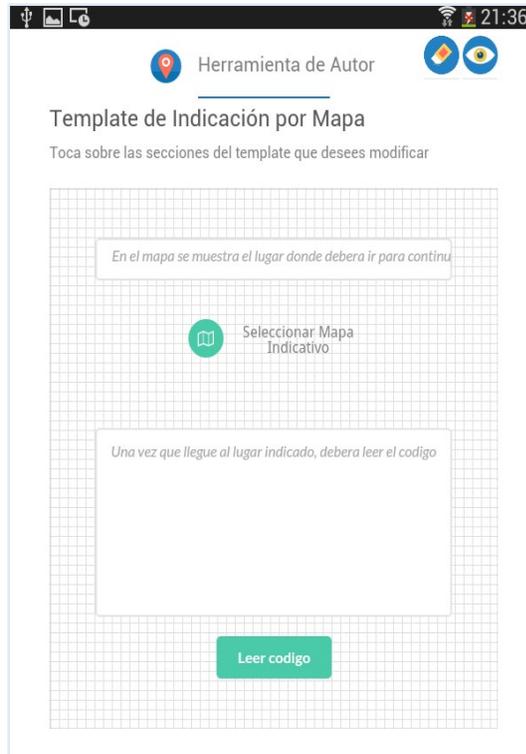


Figura 4.25: *Template* de indicación para llegar a la posición de otra tarea.

➤ Creación de un *template*

Esta opción surge de la Figura 4.19, cuando el usuario elige la creación de un *template* blanco. La herramienta propuesta en esta tesina facilita esta opción para permitirle al usuario contar con la posibilidad de definir sus *templates*.

Al elegir esta opción de creación el usuario recibe una pantalla como se puede observar en la Figura 4.26, la cual cuenta con un menú inferior con distintos componentes que pueden ser agregados al *template*. Además, se cuenta con una grilla (grid) en la cual se irán incorporando los elementos que el usuario decida.

Al momento de crear un nuevo *template*, el usuario debe ingresar un nombre el cual permitirá identificarlo cuando el mismo este listado.

En este tipo de creación, como se mencionaba anteriormente, el usuario podrá agregar los diferentes componentes al *template*, un ejemplo es mostrado en la Figura 4.27. Se puede apreciar en dicha figura que se agregó un elemento Título, un elemento Pregunta y la inserción de una imagen. Una vez que el usuario termina de agregar todos los componentes que desee al *template* deberá guardar los cambios, también puede borrar componentes si ve que no son adecuados para el *template* que está definiendo.

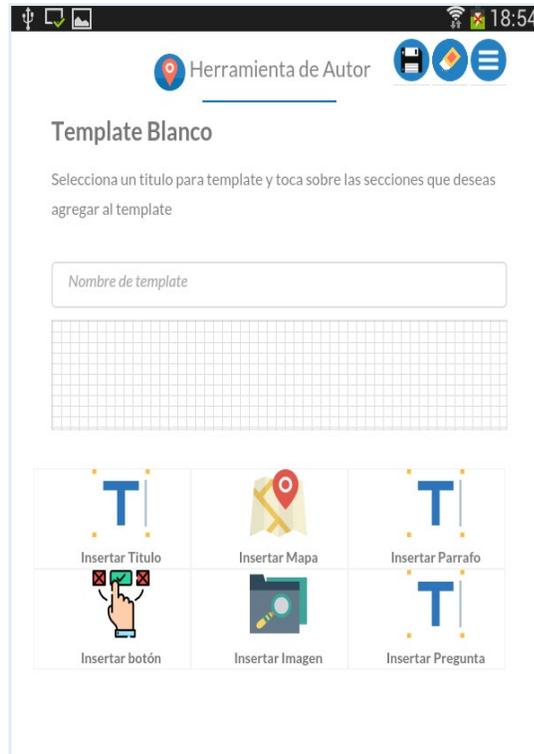


Figura 4.26: Creación de un *template*.

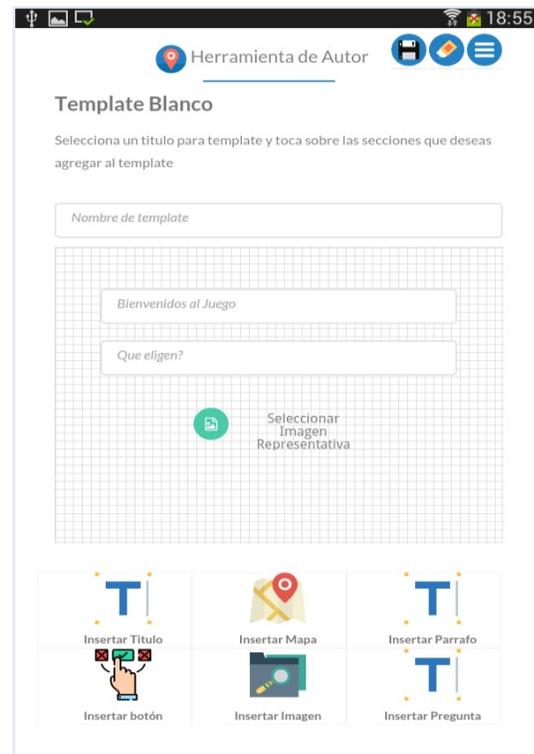


Figura 4.27: Definición de elementos dentro de un nuevo *template*.

❑ Creación de Contenidos Educativos

Al seleccionar la opción *Creación de Contenidos Educativos* de la Figura 4.1, se le permitirá al usuario crear contenidos. Es importante aclarar que desde la perspectiva de esta tesina, los contenidos están siempre embebidos adentro de un *template*. Considerando la separación de capas, se propone la creación de tres tipos de contenidos que luego podrán ser utilizados a la hora de definir un *template* en la creación de *Elementos Educativos Posicionados* (como se explicará más adelante).

Los tres tipos de contenidos que se pueden crear actualmente son la herramienta propuesta en esta tesina son:

- contenido de pregunta
- contenido de texto
- contenido de imagen

La pantalla que recibe el usuario cuando elige la opción *Creación de Contenidos Educativos* de la Figura 4.1 se puede apreciar en la Figura 4.28; donde se puede observar en el menú inferior los tres contenidos mencionados. Podría extenderse en un futuro otros contenidos relacionados a las aplicaciones educativas móviles.



Figura 4.28: Creación de Contenidos Educativos.

Al crear los diferentes contenidos, indistintamente del tipo, el usuario debe indicar el nombre de manera tal que sean identificable luego para su utilización. Es importante aclarar, que un contenido es independiente del otro, por lo cual tanto la creación como el guardado serán unitarios. Es decir, solo podremos crear un contenido por vez, una vez guardado o descartado podremos crear otro.

En la Figura 4.29 se puede apreciar la creación de los diferentes contenidos mencionados. En la Figura 4.29.a se muestra la creación de una pregunta, en este caso un elemento de tipo input se presentará en la pantalla, de manera tal que el usuario pueda ingresar la pregunta.

La creación de contenido texto se puede observar en la Figura 4.29.b, esto se realizará de la misma manera que el contenido pregunta, con la excepción que la cantidad de caracteres que podrá ingresar será más extensa.

En la Figura 4.29.c se muestra la creación de contenido imagen. Esta creación varia de las dos anteriores ya que se le solicita al usuario elegir un las imágenes de su galería.

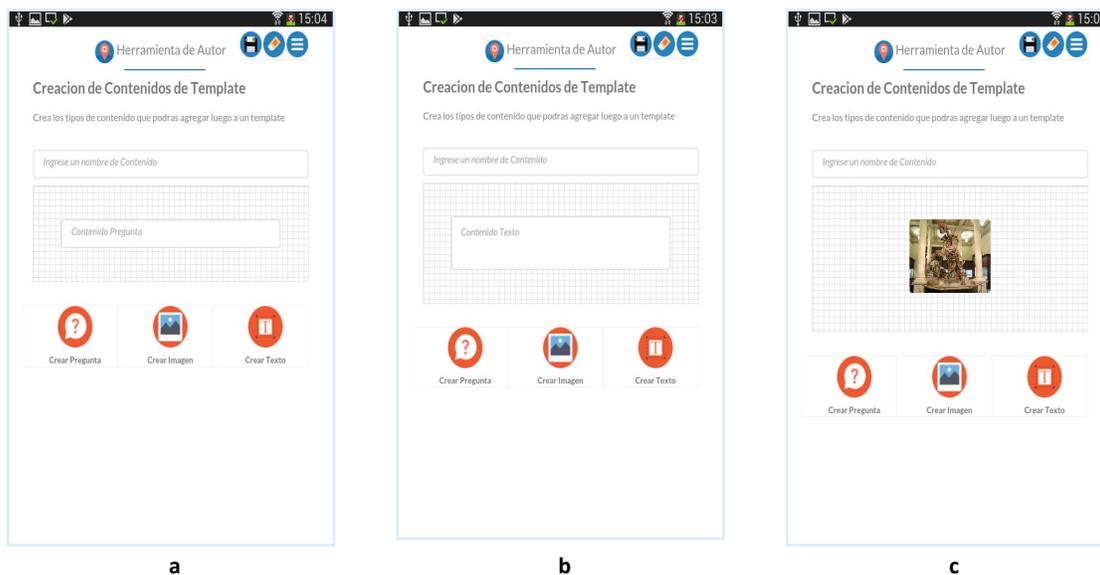


Figura 4.29: Pantallas de creación de contenidos.

En los tres casos presentados en la Figura 4.29, el usuario podrá guardar el contenido para reutilizarlo más adelante.

❑ Creación de Elementos Educativos Posicionados

La herramienta propuesta en esta tesina permite la creación de *Elementos Educativos Posicionados* (opción mostrada en la Figura 4.1). Estos elementos representan contenidos educativos que son asociados a una posición, será posible asignarles su correspondiente *template*.

Cuando el usuario selecciona la opción *Crear Elementos Educativos Posicionados* de la Figura 4.1, se muestra primero las opciones de selección de tipo de mapa (ver Figura 4.9).

Supongamos que el usuario elige un tipo de *Mapa Indoor-Outdoor*. Como resultado recibe una pantalla como se muestra en la Figura 4.30.

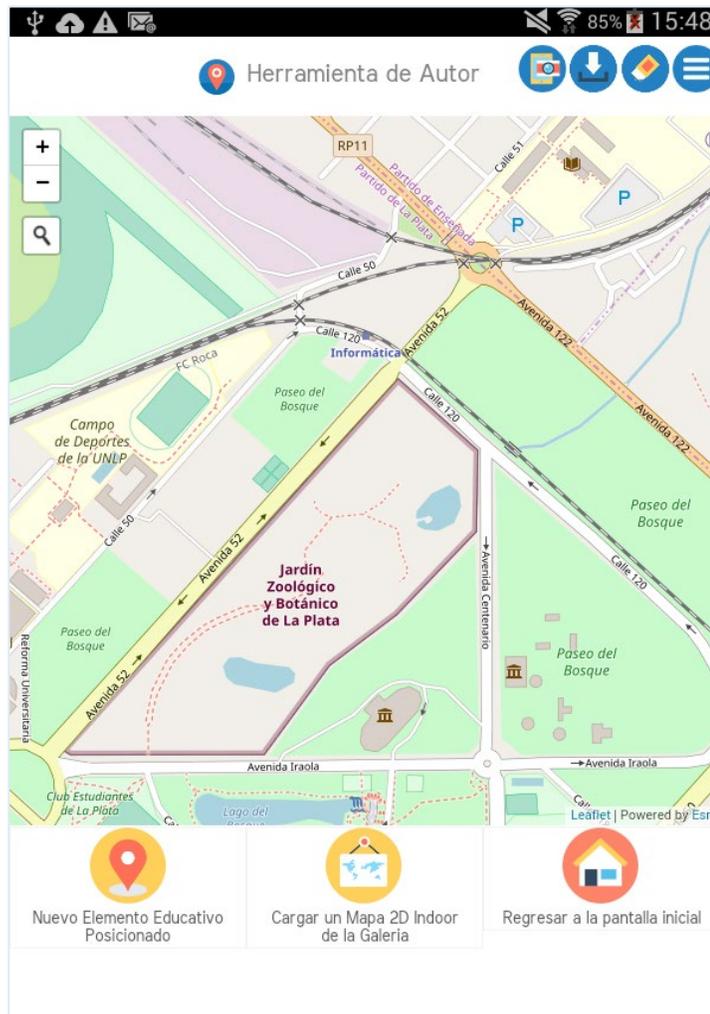


Figura 4.30: Menú de *Creación de Elementos Educativos Posicionados*.

Supongamos que el usuario elige de la Figura 4.30 la opción “*Nuevo Elemento Educativo Posicionado*”, que al igual que en la creación de posiciones será identificado mediante un color y una letra del abecedario; al elegir este icono aparecerá también un *engranaje*, pero las acciones que se podrán realizaran variaran respecto del menú de posiciones, esto puede apreciarse en la Figura 4.31. Se puede observar las siguientes acciones asociadas a un *Elemento Educativo Posicionado*:

- ❖ **QR**, asigna a un *Elemento Educativo Posicionado* su correspondiente código QR (al igual que la asignación de QR para posiciones).
- ❖ **Conectar**, permite asociar el *Elemento Educativo Posicionado* actual, con otro existe (al igual que la conexión entre posiciones). Al presionar sobre esta acción, se abrirá el menú que permitirá elegir entre los elementos existentes para conectar.

- ❖ **Borrar**, permite borrar el *Elemento Educativo Posicionado*.
- ❖ **Template**, al seleccionar esta opción se visualizará la pantalla de *templates* mostrada en la Figura 4.19 (relacionada al tema “*Visualización y Creación de Templates*”). El usuario deberá seleccionar el *template*, para esto podrá editar y utilizar un *template* del “*Aprendo Jugando*” o uno de los *templates blancos* creados previamente (en caso de haber definido alguno).

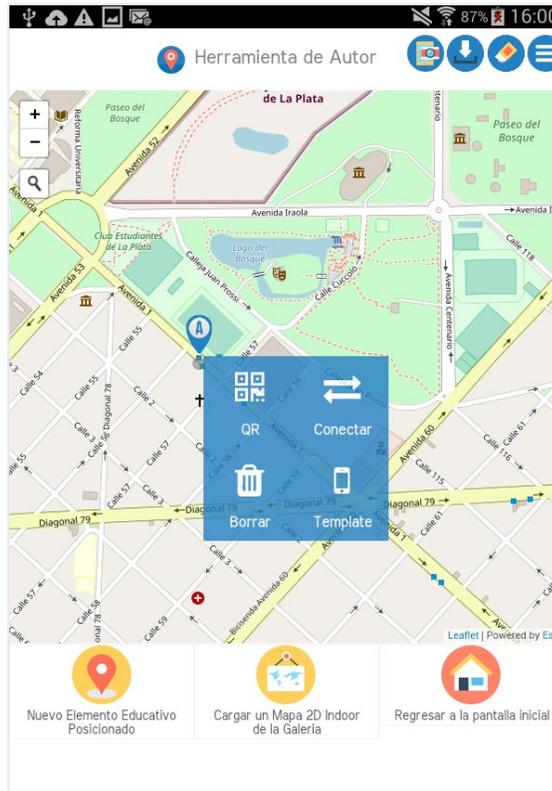


Figura 4.31: Menú de Creación de Elementos Educativos posicionados.

En el caso de elegir conectar dos *Elementos Educativos Posicionados* usando la opción *Conectar* de la Figura 4.31, el usuario recibe una pantalla similar a la Figura 4.32.

Se puede apreciar que al seleccionar la flecha de la Figura 4.32 se despliega un menú con tres opciones.

Supongamos que el usuario desea usar el *template* “*Indicación por medio de un mapa*”, para el mismo debía completar la indicación usando una imagen, esta imagen puede ser realizada mediante una captura del mapa actual usando el icono mostrado en la Figura 4.8.

Si el usuario selecciona el icono de captura de pantalla mostrado en la Figura 4.8, recibirá como respuesta una pantalla con un modal donde se le recomendará que haya hecho el zoom que crea necesario sobre el mapa que quiera visualizar en su captura. Esto se puede apreciar en la Figura 4.33. En este punto el usuario tendrá dos opciones, capturar o volver para realizar los ajustes necesarios.

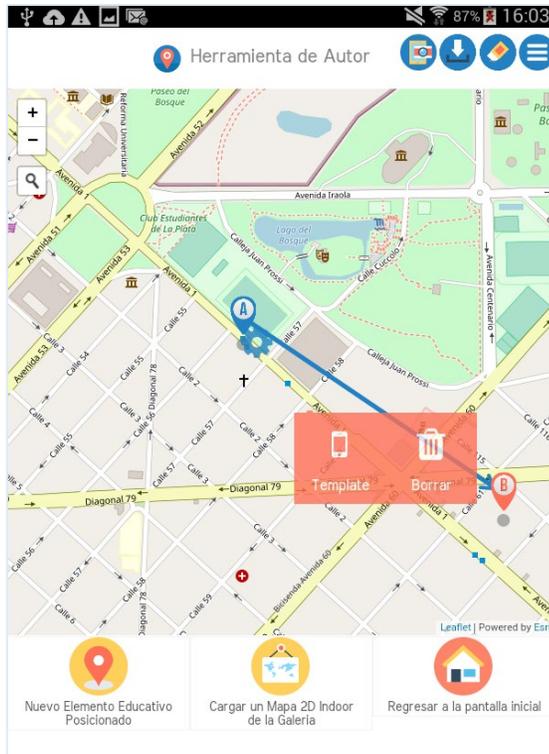


Figura 4.32: Menú en la conexión de los Elementos Educativos Posicionados.

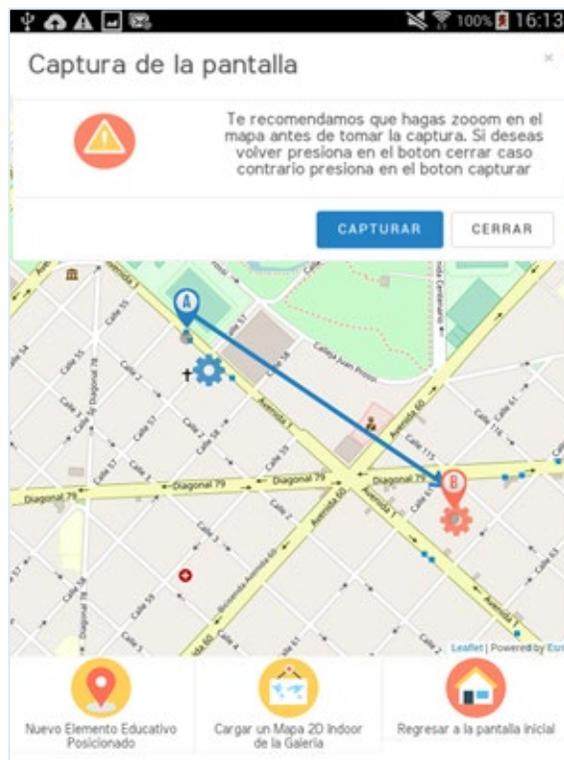


Figura 4.33: Modal de captura de pantalla.

Si se presiona la opción *Capturar* que se visualiza en la Figura 4.33, recibirá una pantalla como se muestra en la Figura 4.34, donde debe seleccionar el área que desee que se guarde como mapa de indicación. Una vez realizado el ajuste del área de interés debe seleccionar la opción guardar.

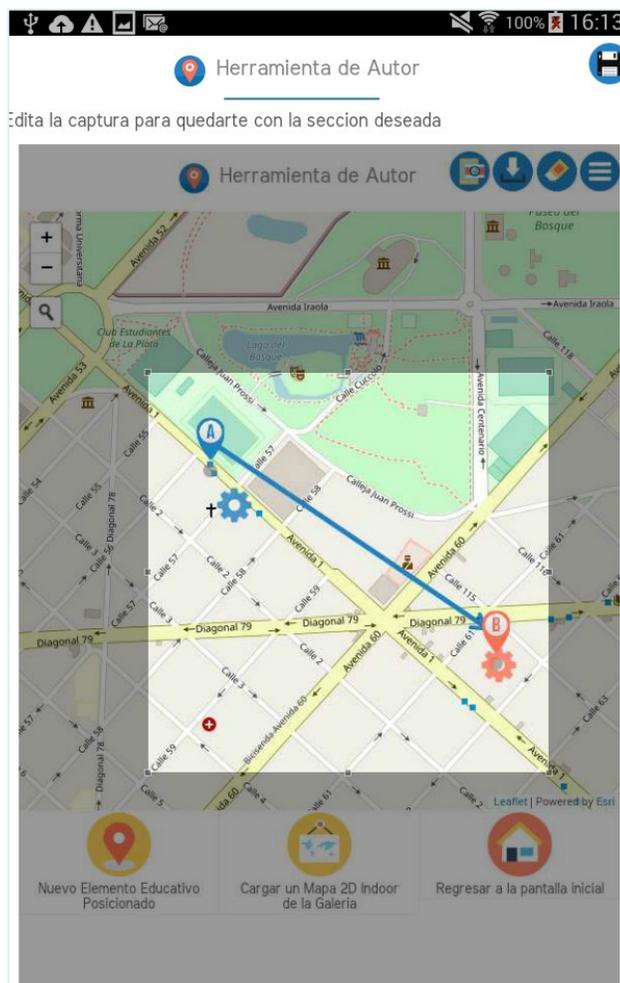


Figura 4.34: Captura de pantalla.

Una vez presionado el botón guardar de la Figura 4.34, la imagen se guarda en el dispositivo móvil, para que a la hora de usar el *template* “Indicación por medio de un mapa” esta imagen pueda ser utilizada, ya que la misma puede ser seleccionada de su galería de imágenes.

Supongamos que el usuario tomo la imagen de la indicación, y luego elige de la Figura 4.32 la opción *template*, y de ahí puede elegir el *template* “Indicación por medio de un mapa”, como resultado recibirá el *template* para empezar a completarlo como se puede apreciar en la Figura 4.35. Se puede observar que en este caso fue completado con la imagen que fue capturada.

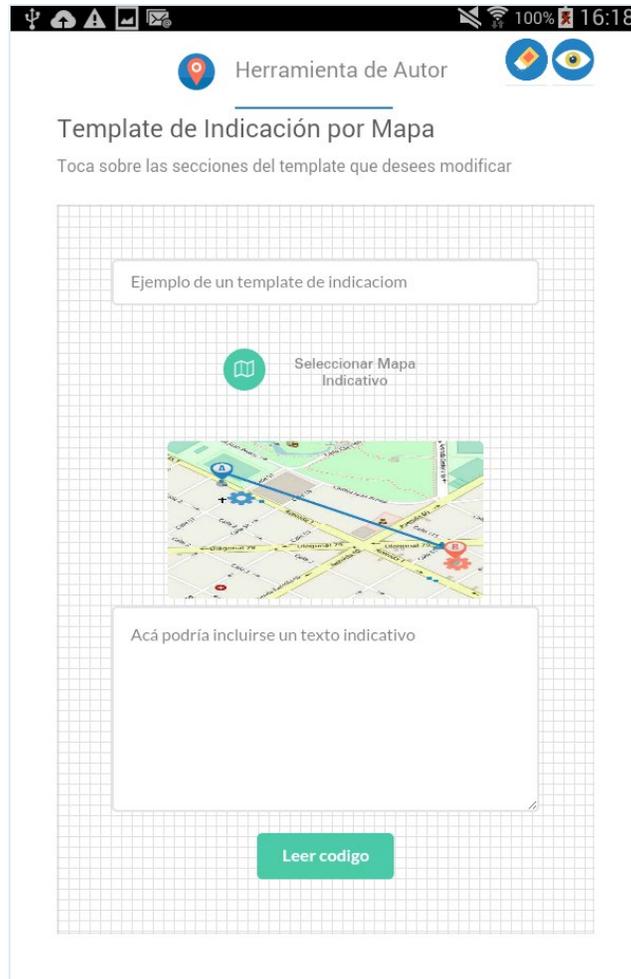


Figura 4.35: *Template* de indicación por medio de mapa con la captura realizada

Como se mencionó anteriormente el docente también podrá asignar *templates* a los distintos *Elementos Educativos Posicionados*, no solo los preexistentes sino también alguno de los *templates* generados mediante la acción de *Template Blanco*. Lo que cambia es que en vez de elegir los *template* del “*Aprendo Jugando*” se eligen los creados.

De esta manera el docente, podrá definir los *Elementos Educativos Posicionados* deseados, asignándole QR cuando corresponda. Con este contenido creado, es posible exportar en formato JSON lo definido, de manera que podría crearse luego una aplicación REST o usar un framework que permitiera consumir esta información y a partir de esta genera un prototipo de aplicación funcional.

4.3 Exportar las creaciones realizadas

En esta sección se presentarán los detalles respecto a la exportación que puede realizarse usando la herramienta presentada en esta tesina. Acorde a esto, la herramienta da soporte para exportar las creaciones realizadas a formato JSON.

Una vez que el usuario terminó de crear, debe apretar el botón de descargar/exportar, presentado en la Figura 4.5. En caso de seleccionar la opción descargar/exportar, se iniciará automáticamente la descarga de un archivo .JSON con las especificaciones. Cabe mencionar que esta opción siempre está disponible para el usuario en las pantallas: *Creación de Posiciones*, *Creación de Contenidos* y *Creación de Elementos Educativos Posicionados*. A continuación se describe como es la exportación en cada uno de estos tres casos.

❑ Exportar *Posiciones*

Cuando se exportan solo capas de posiciones se obtiene un archivo JSON con el formato mostrado en la Figura 4.36.

```
1 {
2   "authoringTool":{
3     "map":{
4       "mapType":,
5       "indoorMap":
6     },
7     "position":{
8       "position":{
9         "positionId":,
10        "positionQR":,
11        "positionLat":,
12        "positionLong":,
13        "connections":{
14          "connection":{
15            "connectionPosition":,
16            "connectionPositionLat":,
17            "connectionPositionLong":,
18            "templateId":
19          }
20        }
21      }
22    },
23    "templates":{
24      "template": {
25        "templateId":,
26        "templateHTML":,
27        "contentId":
28      }
29    }
30  }
31 }
32
```

Figura 4.36: JSON de *Posiciones* exportadas.

Se puede apreciar en la Figura 4.36 el archivo JSON formado por un elemento “*map*” que representara el mapa que fue seleccionado para la creación de estas posiciones, en el campo “*mapType*” es donde se tendrán los valores “*Outdoor*”, “*Indoor*”, “*Indoor-Outdoor*”. Y un campo “*indoorMap*” que solo tendrá un valor si se ha seleccionado un mapa “*Indoor*” o “*Indoor-Outdoor*”. Además, se puede apreciar el elemento “*position*” que tendrá un array de las posiciones creadas, el tamaño del mismo será ilimitado y estará compuesto por un campo “*positionId*” que representara una identificación de la posición creada, un “*positionQR*” que tendrá un valor si el usuario le asigno un QR a la posición, y una “*positionLat*” y “*positionLong*” que representa la *latitud* y *longitud* asociados a dicha posición. Dentro de “*position*” además se tiene un array denominado “*connections*” que representara las conexiones existentes para ese punto; estas conexiones, siendo “*connectionPoi*” el id del POI con el cual se conecta, “*connectionPOILat*”, “*connectionPOILong*” representaran los puntos cardinales del punto a

conectarse. Y un campo *“templateId”* que permitirá identificar un template en caso que haya sido agregado por el usuario.

Ademas, se agregara el elemento *“templates”* el cual será un JSONArray de *templates*, el cual será ilimitado y contara con los siguientes campos: *“TemplateID”*, el cual permitirá identificar el *template*; *“templateHTML”* que su valor será un html que contendrá todo el HTML de acuerdo a lo definido por el usuario; y *“content”* que en caso de contener un contenido, tendrá el *contentID* del mismo.

❑ **Exportar Contenidos Educativos**

Cuando se exportan solo capas de contenidos se obtiene un archivo JSON con el formato mostrado en la Figura 3.37. En ese caso el JSON está conformado por el elemento *“contents”*, el cual está compuesto por los *contenidos* que se hayan definido. Para cada contenido se define el elemento *“contentName”* que representara el nombre del contenido, asignado por el usuario a la hora de crearlo, lo que permitirá diferenciarlo de los otros. Un elemento *“contentType”* que representa el tipo de contenido: que podrá tener tres valores posibles: imagen (*image*), texto (*text*) ó pregunta (*question*). Y por último un elemento *“contentValue”* que contendrá en el caso del texto, el texto correspondiente, en el caso de la pregunta, la pregunta que haya ingresado el usuario y en el caso de la imagen la URL que representa la ruta donde esta almacenada dentro del dispositivo móvil. A su vez, cada contenido tiene un Id en particular, este si bien, es de uso interno permite identificar unívocamente a ese contenido, por lo tanto, consideramos que para una posible herramienta que consuma este JSON este dato tendrá mayor relevancia.

```
1 {
2   "authoringTool":{
3
4     "contents":{
5       "content1":{
6         "contentId":"content1",
7         "contentName":"Contenido texto",
8         "contentType":"text",
9         "contentValue":"Contenido texto"
10      },
11      "content2":{
12        "contentId":"content2",
13        "contentName":"Contenido Pregunta",
14        "contentType":"question",
15        "contentValue":"Contenido texto"
16      },
17      "content3":{
18        "contentId":"content3",
19        "contentName":"Contenido Imagen",
20        "contentType":"image",
21        "contentValue":"url/imagen.jpg"
22      }
23    }
24  }
25 }
26 }
27 }
```

Figura 4.37: Ejemplo de un archivo JSON de *Contenidos Educativos* exportados.

❑ Exportar *Elementos Educativos Posicionados*

En el caso de exportar *Elementos Educativos Posicionados* se obtiene un archivo JSON con el formato mostrado en la Figura 3.38. Este sería el tipo de exportación más completa para poder generar a partir del archivo JSON, Aplicaciones Educativas Móviles basadas en posicionamiento funcionales. Esto se debe a que no solo se cuenta con las posiciones sino también con los contenidos asociados a las mismas, y con los *templates* asociados a cada contenido se puede lograr mostrar información en una aplicación.

```
1 {
2   "authoringTool":{
3     "pois":{
4       "poi":{
5         "poiId":,
6         "poiQR":,
7         "poiLat":,
8         "poiLong":,
9         "templateId":,
10        "connections":{
11          "connection":{
12            "connectionPOI":,
13            "connectionPOILat":,
14            "connectionPOILong":,
15            "templateId":
16          }
17        }
18      },
19    },
20    "templates":{
21      "template": {
22        "templateId":,
23        "templateHTML":,
24        "contentId":
25      }
26    },
27    "map":{
28      "mapType":,
29      "indoorMap":
30    }
31  }
32 }
33 }
```

Figura 4.38: Ejemplo de un archivo JSON de *Elementos Educativos Posicionados* exportados.

Se puede apreciar en la Figura 4.38 que se cuentan con elementos similares a la exportación de posiciones y contenidos mostradas anteriormente.

Para el caso del Poi se agrega un nuevo elemento "*templateId*" dentro del elemento "*poi*", este permite indicar si un *template* fue asignado al Poi, el valor de ese campo será el id del *template*.

Cabe mencionar que estos HTML podrían ser usados en una aplicación móvil funcional para poder generar las pantallas al usuario.

5. Ejemplo de uso de la Herramienta propuesta

En este capítulo se presenta un ejemplo de la herramienta definida y descrita en el Capítulo 4. Supongamos que un docente desea diseñar una *Actividad Educativa Posicionada* del estilo del "Aprendo Jugando" similar a la descripción del Capítulo 3. Es decir, poder combinar contenidos educativos en determinados lugares de un espacio físico y establecer un recorrido físico entre los mismos. Para esto, debe definir:

- los templates (pantallas),
- el contenido e indicaciones que se representan por medio de estos templates
- el espacio físico donde se posicionarán estos contenidos

A continuación se brindarán más detalles de cómo usando la herramienta propuesta en esta tesina se puede definir una *Actividad Educativa Posicionada*. Supongamos que nos limitamos a dos tareas a fines explicativos y para simplificar el ejemplo.

Cada tarea posicionada tendrá la información necesaria para llegar hacia ella y cómo proseguir luego de haber realizado la indicación, particularmente, mediante un texto descriptivo y mediante el uso de un mapa indicativo.

Para poder comenzar se ha dibujado un plano con el fin de representar el espacio físico en el cual se posicionará la actividad educativa, el mismo se puede apreciar en la Figura 5.1. Por simplicidad el mismo será posicionado desde la herramienta en el campo de deportes de la UNLP contiguo de La Facultad de Informática de la UNLP.



Figura 5.1: Plano del espacio físico.

Supongamos que el docente desea especificar el recorrido de la siguiente manera:

1. Definir la pantalla de bienvenida.
2. Definir la primera indicación por medio de mapa, para llegar a la primera tarea educativa.
3. Definir la consigna de la tarea número 1, junto con la definición de la pantalla le asociara a la misma un código QR que permitirá identificar la posición con la indicación.
4. Definir la finalización de una tarea.

5. Definir la siguiente tarea.
6. Definir la opción de depósito.
7. Definir la finalización del juego.

A continuación se especifican los pasos que deben realizarse para poder crear la experiencia mencionada.

- a. Selección de la opción “*Crear Elementos Educativos Posicionados*” en el menú principal de la aplicación (como se describió en la Sección 4.2 del Capítulo 4).
- b. Selección del tipo de mapa, supongamos que se selecciona la opción de “*Mapa Indoor-Outdoor*” (como se describió en la Sección 4.2 del Capítulo 4 y se visualizó en la Figura 4.10). Al estar in-situ, y si la aplicación cuenta los permisos necesarios el mapa automáticamente se centrara en la posición actual del usuario.
- c. Desde el menú de opciones de mapa cargar el Mapa 2D Indoor de la galería que se corresponde con el espacio físico (como se describió en la Sección 4.2 del Capítulo 4 y se visualizó en las Figura 4.10 y 4.11). Se puede hacer uso de los redimensionadores (para mover, agrandar o reducir la imagen) y del toolbox para ajustar su opacidad. En la Figura 5.2 se puede observar el espacio físico posicionado sobre el campo contiguo a la facultad de Informática.



Figura 5.2: Plano del espacio físico posicionado en un espacio outdoor.

- e. Creación del primer *Elemento Educativo Posicionado*, que representara la pantalla de bienvenida. El docente decide ubicar este elemento en la posición que crea más adecuada para arrancar la aplicación. Para asignarle el correspondiente *Template* el usuario deberá

definir la pantalla de bienvenida, siguiendo los pasos explicados en el Capítulo 4 de esta tesina. Podemos observar en la Figura 5.3.a la creación del elemento, en la Figura 5.3.b la edición del *template* y en la Figura 5.3.c la previsualización y guardado del *Template* de bienvenida.

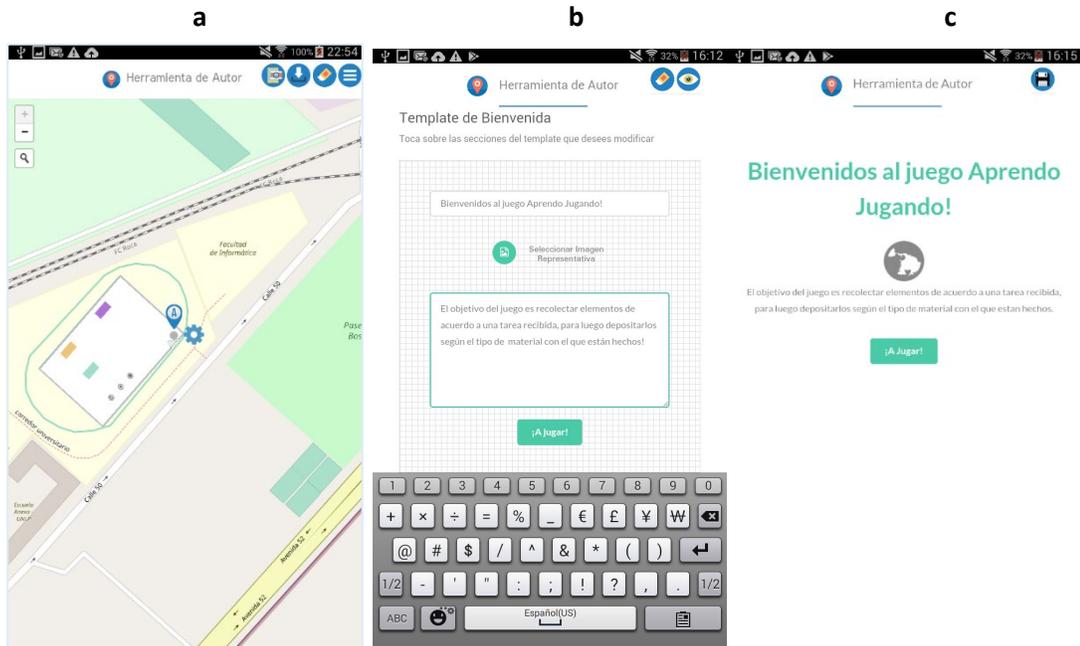


Figura 5.3: Creación del primer *Elemento Educativo Posicionado*: Pantalla de Bienvenida.

- f. Creación del segundo *Elemento Educativo Posicionado* que se corresponderá con la primera tarea educativa, asignarle el template correspondiente, para esto tendrá que definir el template que tendrá las indicaciones correspondientes a la Tarea 1, tocar sobre el icono de engranaje que permitirá expandir el menú correspondiente, y ahí seleccionará la opción de “*Template de indicación de Tarea*”, donde se editará y guardará el template.

En la Figura 5.4 se puede apreciar en la pantalla de la izquierda el menú correspondiente al *Elemento Educativo Posicionado*, donde una de las opciones es *Template*, al elegir la opción “*Template de indicación de Tarea*” se despliega la pantalla de la derecha para editar el template.

Para este *Elemento Educativo Posicionado* se debe definir además de su template, el código QR que se corresponderá con esta tarea. En la Figura 5.5 se aprecia nuevamente el menú del *Elemento Educativo Posicionado*, en este caso se debe elegir la opción QR.

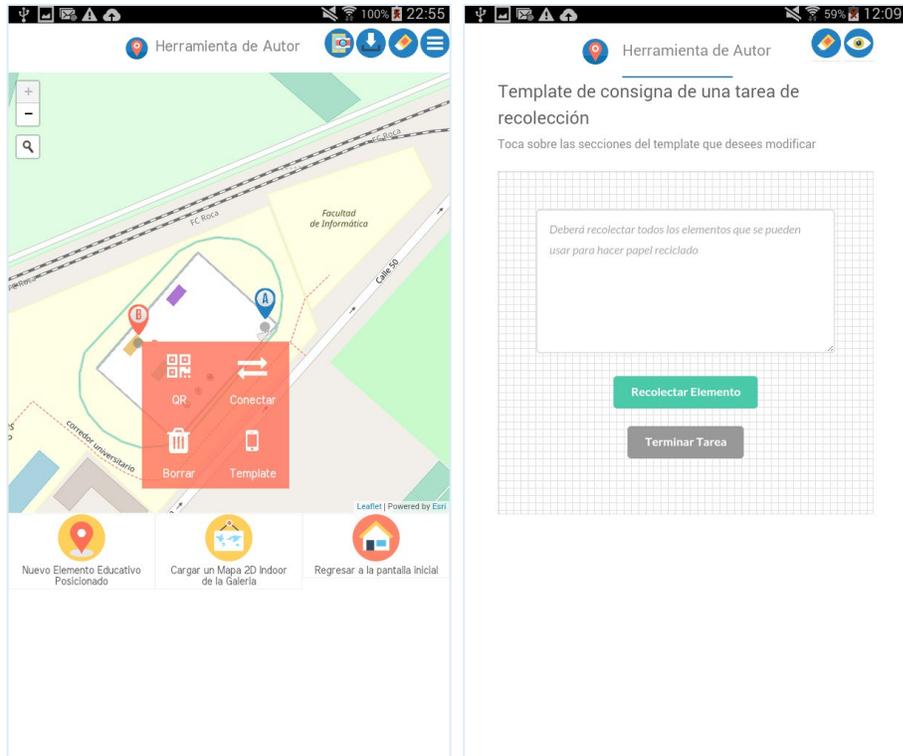


Figura 5.4: Creación del segundo *Elemento Educativo Posicionado*: Primera Tarea Educativa.

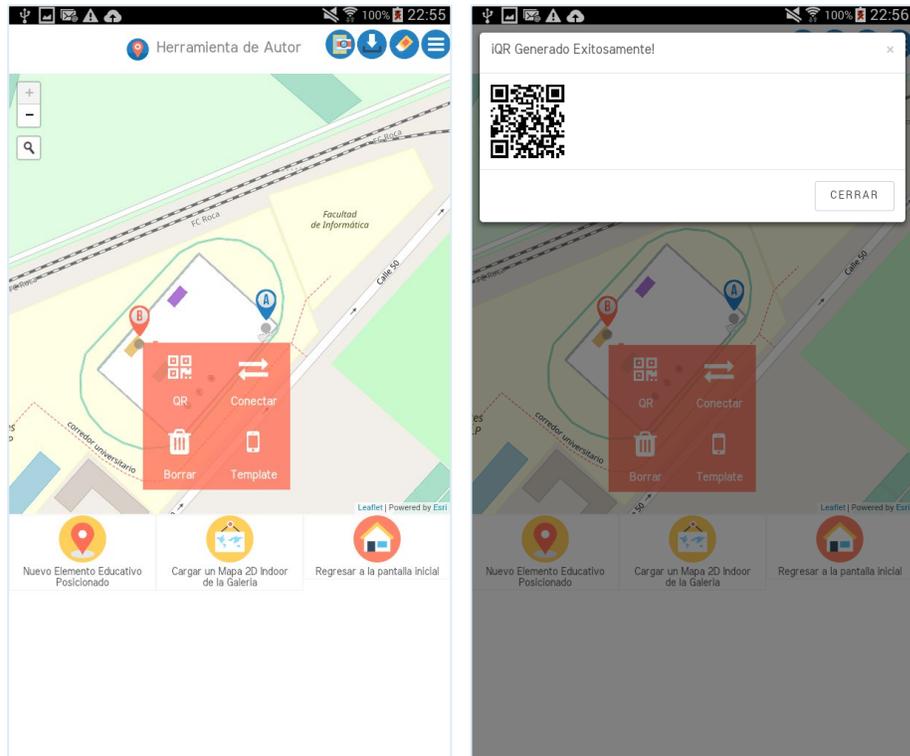


Figura 5.5: Asignación del código QR a la primera tarea educativa.

- g. Una vez asignado el QR, el docente debe conectar el elemento educativo que representa la pantalla de bienvenida con la primera tarea educativa posicionada, esto permitira indicarle a los alumnos a donde deben dirigirse para comenzar con la primer tarea número una vez que recibieron la pantalla de bienvenida.

Para realizar esto el docente debería presionar sobre la opción conectar que se visualiza en el menu que se despliega al presionar sobre el icono del engranaje del *Elemento Educativo Posicionado* representaba por la letra A, del menu de opciones elegir el *Elemento Educativo Posicionado* por la letra B. Estas acciones se pueden visualizar en la Figura 5.6.

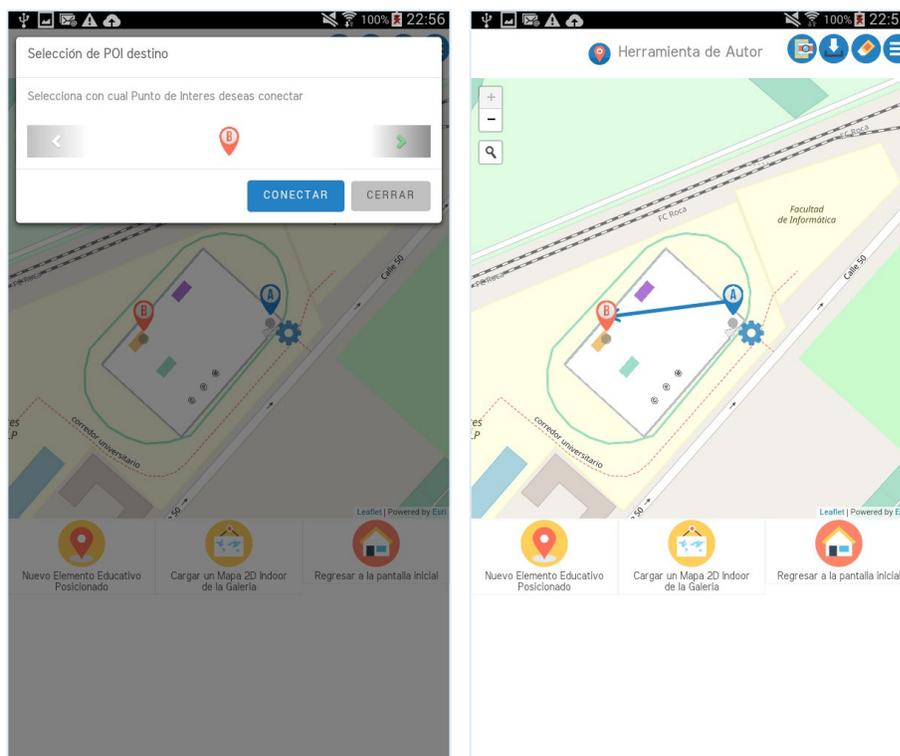


Figura 5.6: Conexión de la pantalla de bienvenida con la primera tarea educativa.

Luego de tener la conexión generada, el usuario puede asignarle un *Template*. Para este caso, el docente puede elegir usar la funcionalidad de captura de la sección del mapa (tal como se mencionó en el Capítulo 4, a través del uso del botón representado con el ícono de la Figura 4.8) o la cargar una imagen de su propia galería.

A fines de poder mostrar la funcionalidad de esta herramienta, definiremos el *Template* de indicación utilizando la opción de captura de pantalla provista por la herramienta presentada en esta tesina. Para poder tomar la captura, el usuario presionara sobre el icono que se corresponde con la Figura 4.8, seleccionara el área que considere de mayor claridad para indicarle a los usuarios como llegar hacia la primera tarea educativa, y guardara esta Imagen. Estos pasos pueden visualizarse en la Figura 5.7.

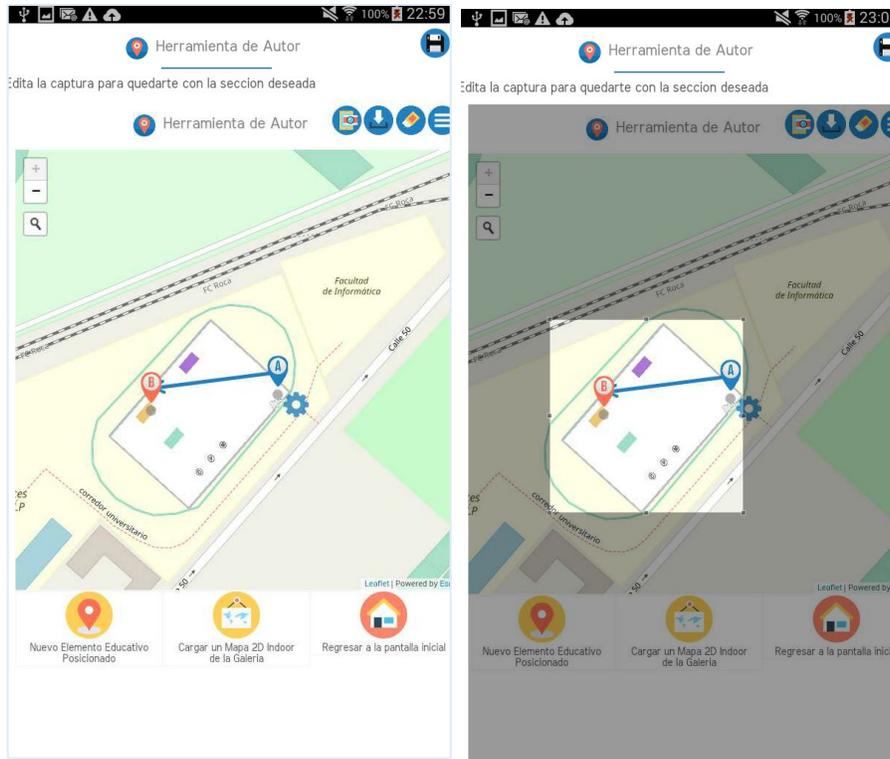


Figura 5.7: Captura que indica cómo llegar a la primera Tarea Educativa.

Una vez tomada la captura, y hecha la selección del mapa el docente deberá asignar en la conexión el *template* de indicación por medio de mapa, utilizando en su definición la captura recién tomada. Para esto deberá tocar sobre la fecha que identifica la conexión, nuevamente, tocar sobre el engranaje y seleccionar la opción de *template*. Una vez en la selección de *template* el docente deberá elegir el de “*Template por indicación por medio de mapa*”, editarlo y guardarlo. Estos pasos pueden verse en la secuencia de imágenes de la Figura 5.8.

El docente ahora debería crear la pantalla de finalización de una tarea y asociarla al punto B creado previamente. Para indicar esta pantalla deberá usar el *Template de Finalización de Una Tarea*.

La versión actual de la herramienta asocia una sola pantalla en cada posición, acorde a esto para indicar la pantalla de finalización de una tarea se debe crear un nuevo Elemento Educativo Posicionado, asignarle el *Template de Finalización de Una Tarea* y luego asignar la indicación desde el punto B (primera tarea educativa) a la finalización de la misma. Estos pasos se pueden apreciar en la Figura 5.9.

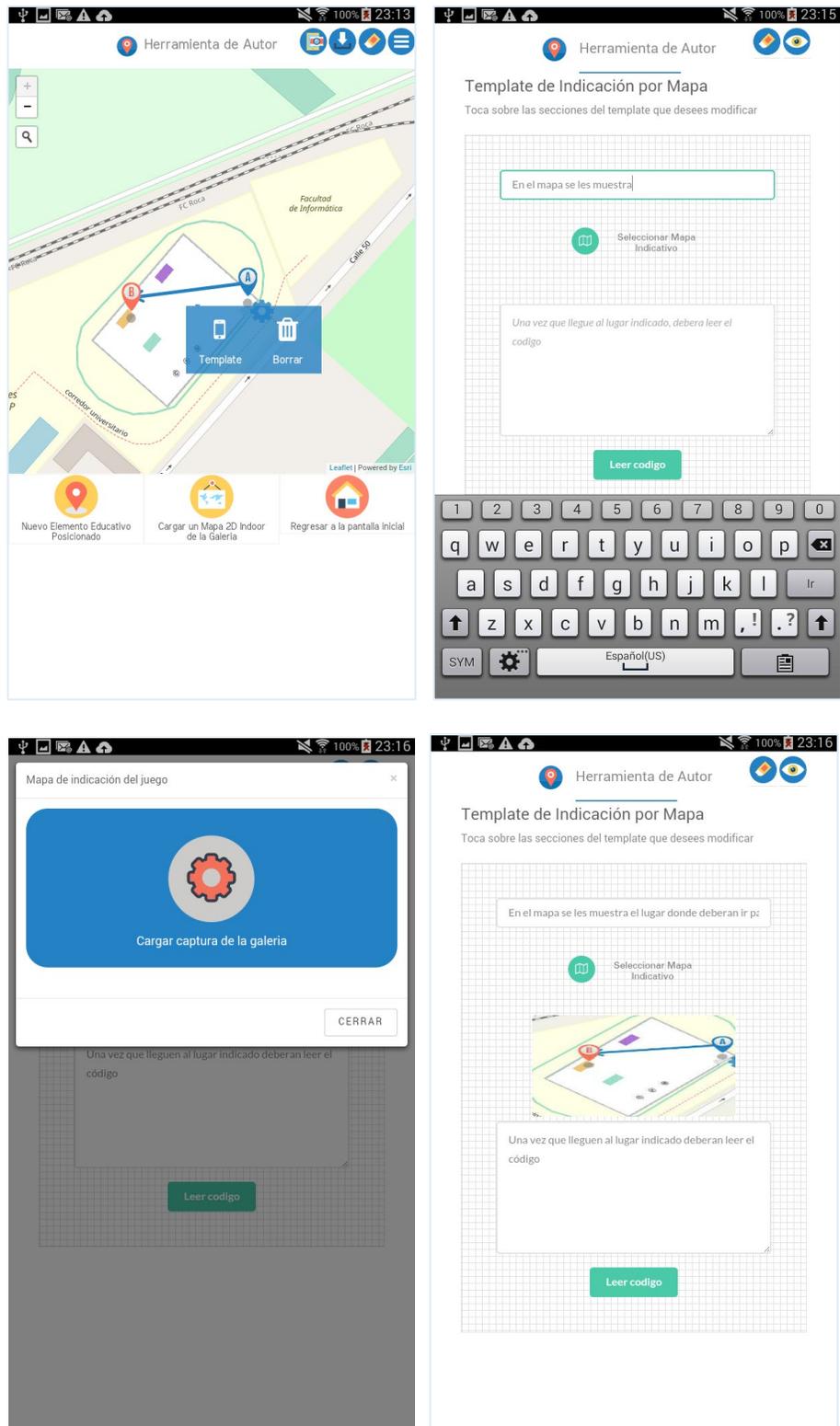


Figura 5.8: Asignación del *Template de Indicación por Mapa*.

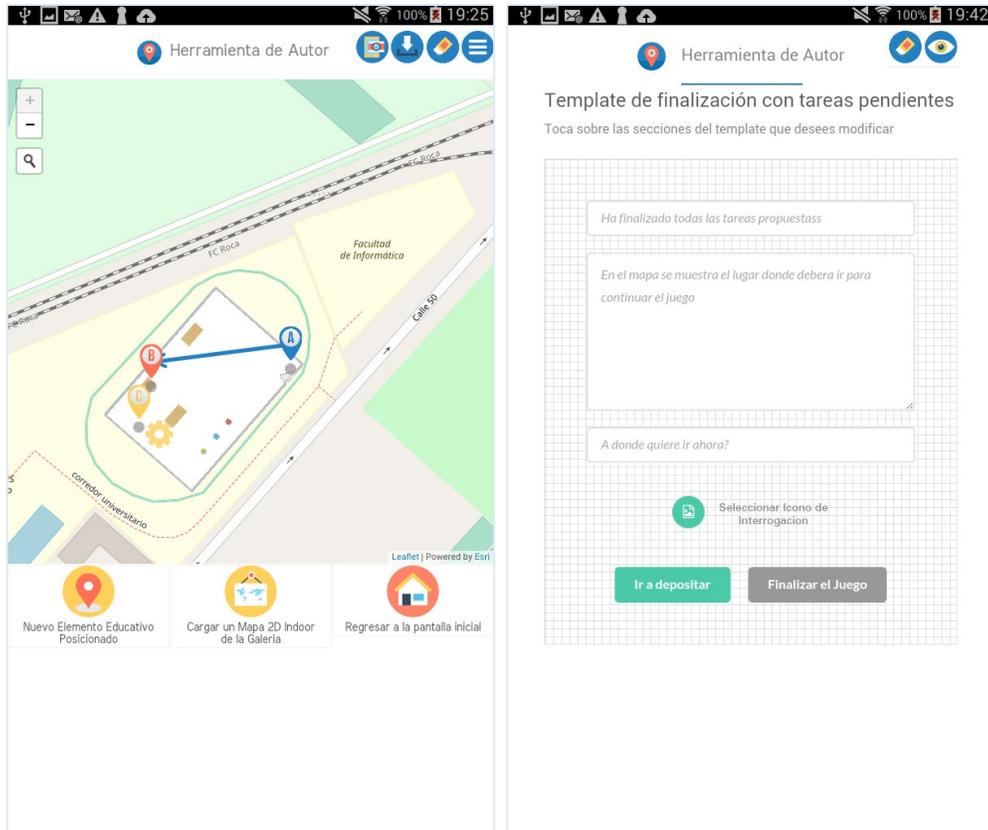


Figura 5.9: Creación del Elemento Educativo Posicionado con el Template de Finalización de una tarea.

Como se puede apreciar en la Figura 5.9, el template de finalización de una tarea muestra al alumno dos opciones, acorde a esto, el docente debería crear dos nuevos *Elementos Educativos Posicionados* y asignarles sus correspondientes templates, uno relacionado con el depósito otro con la finalización del juego. Los pasos relacionados a estos son similares a la asignación de un template de finalización de una tarea.

- h. Supongamos que se crean la segunda tarea educativa, la tarea de depósito y la finalización del juego, todo esto siguiendo los pasos mencionados anteriormente, lo que va variando son los templates que se van usando.

De esta manera, se mostró mediante un ejemplo como ir creado un ejemplo con la herramienta desarrollada para esta tesis. Cabe destacar que el docente puede ir armando el recorrido 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 JSON generado en función al caso de uso mencionado, cuando se desea exportar por completo la creación del usuario. En la Figura 5.10 se puede visualizar la estructura de dicho JSON.

```

1  {
2    "authoringTool": {
3      "pois": {***
34    },
35    "templates": {***
52    },
53    "map": {***
56    }
57  }
58 }

```

Figura 5.10: Estructura del JSON generado por la aplicación al exportar el ejemplo de caso de uso mencionado.

En la Figura 5.11 se visualiza el primer elemento educativo posicionado creado acorde a lo especificado en el caso de uso propuesto. En este se observa, el identificador de QR que se guardó en el dispositivo bajo el mismo nombre, el identificador de *template* asociado, la conexión con el segundo elemento educativo posicionado creado, donde además se describe el identificador del *template* de la conexión.

```

3  ▼ "pois": {
4  ▼   "educative_positioned_content_a": {
5     "poiId": "educative_positioned_content_a",
6     "poiQR": "authoringTool_19102018162244.png",
7     "poiLat": -34.90407444489416,
8     "poiLong": -57.93895159520256,
9     "templateId": "1_template",
10 ▼   "connections": {
11 ▼     "content_a_educative_positioned_content_b": {
12      "connectionPOI": "educative_positioned_content_b",
13      "connectionPOILat": -34.90423012894435,
14      "connectionPOILong": -57.93934106826783,
15      "templateId": "3_template"
16    }
17  }
18 }

```

Figura 5.11: JSON del elemento educativo posicionado (A) al exportar el ejemplo de caso de uso mencionado.

De la misma manera en la Figura 5.12 se visualizan los datos de los otros dos elementos educativos posicionados (b y c). En el JSON se puede observar que estos elementos se encuentran georeferenciados, y, aquellos definidos dentro del mapa del espacio delimitado, además, poseen una posición simbólica, que en este caso se corresponderá con el valor que tomó el código QR. Se define además la asociación con los *templates* que se encuentran presentes en la estructura de este JSON.

```

19  "educative_positioned_content_b": {
20      "poiId": "educative_positioned_content_b",
21      "poiQR": "qr",
22      "poiLat": -34.90423012894435,
23      "poiLong": -57.93934106826783,
24      "connections": {},
25      "templateId": "2_template"
26  },
27  "educative_positioned_content_c": {
28      "poiId": "educative_positioned_content_c",
29      "poiQR": "qr",
30      "poiLat": -34.90437970877864,
31      "poiLong": -57.93956637382508,
32      "connections": {}
33  }
34  },

```

Figura 5.12: Elementos educativos posicionados (B y C) al exportar el ejemplo de caso de uso mencionado.

En la Figura 5.13 se muestra a modo de ejemplo, uno de los *templates* generados por el JSON, estos están compuestos por el Identificador que permite la correcta asociación con los elementos educativos posicionados, como así también el código HTML que representa la plantilla generada por el docente, esto permitirá (que junto con los correspondientes CSS) a futuro, cuando la herramienta evolucione y permita generar aplicaciones móviles ejecutables estas plantillas pueden ser integradas fácilmente.

```

35  "templates": {
36      "1_template": {
37          "templateId": "1_template",
38          "templateHTML": "\n<div id=\"wrapper\">\n<div id=\"main\">\n<section
id=\"\">\n<div class=\"container\">\n<header class=\"major\">\n<h2
class=\"alignCenter\" id=\"wt_inputTitle\">¡Bienvenidos al juego
Aprendo Jugando!</h2>\n<div></div>\n</header>\n<div class=\"
container alignCenter marginBottom\">\n<i id=\"iconSelect\" class=\"
fa-spin fa fa-globe fa-5x fa-refresh\"></i>\n</div>\n<p class=\"
alignCenter\" id=\"wt_inputParagraph\">El objetivo del juego es
recolectar elementos de acuerdo a una tarea recibida, para luego
depositarlos segun el tipo de material con el que están hechos!</p>\n
<div class=\"container\">\n<div class=\"row uniform\">\n<div class=\"
12u\">\n<ul class=\"actions alignCenter\">\n<li><input id=\"wt_button
\" type=\"submit\" class=\"special\" value=\"¡A Jugar!\"></li>\n</ul>
\n</div>\n</div>\n</div>\n</div>\n</section>\n</div>\n</div>\n"
39  },
40  "2_template": {

```

Figura 5.13: Ejemplo de template exportado en el JSON.

Como último elemento del JSON se lista el elemento que representa el tipo de mapa y el mapa indoor asociado, como se puede visualizar en la Figura 5.14.

```

52  },
53  "map": {
54      "mapType": "GeoMap",
55      "indoorMap": "aulaTesis.png"
56  }
57  }
58  }

```

Figura 5.14: Elemento map al exportar el ejemplo de caso de uso mencionado.

Por último, en la Figura 5.15 se muestran los códigos QR correspondientes a los dos primeros elementos educativos posicionados del caso de uso, dichos códigos QR deberán ser ubicados en las posiciones adecuadas.



Figura 5.15: Códigos QRs generados por el ejemplo de caso de uso mencionado.

Hasta el momento la versión actual de la herramienta solo deja definir un *template* por posición, en un futuro se podrá tener *template* compuestos para que el usuario no tenga que moverse hasta otra posición solo para recibir una pantalla indicativa.

6. Publicaciones realizadas relacionadas con la tesina

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

- **Authoring Tool for Location-Aware Learning Experiences [Zimbello et al., 2017]**

Resumen: In this paper, we present some features of our in-situ indoor-outdoor authoring tool for location-based learning experiences. For example, how to reuse locations in different experiences or how to design from scratch in places without any location sensing mechanisms. In addition, we describe some features in which we are working on. We expect to create a space for discussion about these kind of tools and to reflect on the products created with them.

Publicado en: MobileSoft 2017

- **Relación con la tesina presentada:**

La publicación [Zimbello et al., 2017] está íntegramente relacionada con esta tesina, ya que se presenta una primera versión inicial de la herramienta propuesta en el Capítulo 4. Esto permitió poder evaluar con la comunidad científica la utilidad de este tipo de herramienta. Más que nada, al presentarse en *MobileSoft* donde el público no es experto en educación, entonces se pudo recibir feedback relacionado a características de funcionalidad de este tipo de aplicaciones, sobre con una visión internacional y actual del uso de aplicaciones móviles.

- **Experimentando un uso no tradicional de PhoneGap [Zimbello and Challiol, 2016]**

Resumen: En este trabajo presentamos un plugin de PhoneGap para Android que permite una comunicación genérica entre los Componentes visual es de una aplicación PhoneGap y una clase que funciona como punto de entrada de algún modelo Java. Dicho plugin fue utilizado para la creación de una aplicación móvil basada en posicionamiento. En este trabajo se brindan detalles de nuestro plugin, como así también aspectos aprendidos durante el desarrollo y futuros usos del mismo.

Publicado en: CACIC 2016

- **Relación con la tesina presentada:**

La publicación [Zimbello and Challiol, 2016] permitió explorar características avanzadas del uso de *PhoneGap*, esto fomentó la decisión de llevar la herramienta propuesta en [Alconada Verzini, 2015] a *PhoneGap*. Además, sirve de base para poder proyectar a futuro poder exportar con la herramienta propuesta generar aplicaciones *PhoneGap* empaquetadas con lo que el docente crea con nuestra herramienta, y donde además se utilice el modelo de *Actividades Educativas Posicionadas* propuesto en [Lliteras, 2015]. Es decir, la publicación [Zimbello and Challiol, 2016] sentó las bases

para plantear la herramienta propuesta (Capítulo 4) y además en un futuro facilitará el empaquetado de aplicaciones generadas con nuestra herramienta.

Se adjunta al final de la presente tesina ambos trabajos.

7. Conclusiones y Trabajos Futuros

En esta tesis se presentó una herramienta para la facilitar la creación de Actividades Educativas Posicionadas que permite, a usuarios no expertos con especial foco especialmente en docentes, definir recorridos que conlleven un aprendizaje tanto en espacios indoor como outdoor; brindándole la posibilidad de no solo construir el orden en que se darán las tareas, sino también sus posibles plantillas representativas, de manera tal que el docente pueda definir tanto el contenido de las tareas como así también las indicaciones para llegar y realizar las mismas.

Se realizó en el Capítulo 2, un relevamiento de las herramientas que existían en la actividad, que permitió la selección de una de ellas para ser extendida. Se presentó, en el Capítulo 3, un análisis de las características de las *Actividades Educativas Posicionadas* que involucran elementos concretos, a partir de ello se consideró un esquema con los conceptos constitutivos de la herramienta que hace foco en la reusabilidad y extensibilidad de los mismos.

Luego del análisis se presentó bosquejo mostrando las plantillas identificadas del análisis del “*Aprendo Jugando*”. De esta manera, se identificó cuáles son los *templates* que iban a ser brindados en la primera versión de la herramienta presentada en esta tesina.

En el Capítulo 4 se describieron las problemáticas encontradas y como fueron resueltas en relación a la herramienta que se decidió extender. A su vez en dicho capítulo se presentaron las características de la herramienta desarrollada usando *PhoneGap*. En el Capítulo 5 se mostró mediante un ejemplo como hacer uso de la herramienta desarrollada.

Además, de mostrar cómo se utilizar tanto los templates provistos por la herramienta como la generación de unos nuevos. La herramienta presentada en este trabajo fue parte del material presentado en dos publicaciones, las cuales están estrechamente relacionadas con la tesina presentada [Zimbello and Challiol, 2016] y [Zimbello et al., 2017]. En el Capítulo 6 se describe con más detalle la relación de cada uno de estas publicaciones con esta tesis.

En ambos trabajos [Zimbello and Challiol, 2016] y [Zimbello et al., 2017] la autora de la presente tesina ha sido la persona encargada en ir a presentar tanto en “*4th International Conference on Mobile Software Engineering and Systems*”, como así también en Congreso Argentino de Ciencias de la Computación, esto permitió tomar contacto con trabajos similares, y poder apreciar que todavía es una área abierta de trabajo, que aun cuenta con mucho potencial de crecimiento.

Cabe mencionar que con este tema de tesina se ha aplicado a una BECA de La Agencia Nacional de Promoción Científica y Tecnológica, a través del Fondo Fiduciario de Promoción de la Industria del Software (FONSOFT), dicha beca fue adjudicada conforme surge la Resolución N° RESOL-2017-106-APN-DANPCYT#MCT.

A continuación se mencionan distintos trabajos futuros que se pueden desprender de la tesina presentada:

- El primer paso a seguir es hacer una prueba con docentes para poder ver la utilidad de la herramienta creada. Si bien en esta tesina se abordaron características más técnicas o tecnológicas, luego, los docentes son los que terminan adoptando este tipo de prácticas con lo cual es fundamental retroalimentarse con su feedback. Logrando de esta manera, una herramienta que les pueda ser útil.
- La herramienta presentada también podría ser utilizada por alumnos para ser creadores de contenidos posicionados, en este caso, se deberá explorar aquellos aspectos relacionados con la interfaz visual para poder así adaptarla al perfil del usuario que la utiliza, ya sea docentes o alumnos.
- La herramienta podría extenderse con la siguiente funcionalidad:
 - Proveer más variabilidad de plantillas relacionadas a diferentes tipos de tareas que podrían hacer los alumnos en cada posición. Por ejemplo, plantilla para preguntas de opción múltiple.
 - Proveer plantillas para brindar soporte a las distintas estructuras relacionadas con las *Actividades Educativas Posicionadas* que se pueden crear, por ejemplo, no es lo mismo que tenga un grafo de decisión, esto genera que la planilla tenga que mostrar todas las opciones que podría optar el alumno. Para esto, se podría primero analizar las estructura mencionadas en [Hargood et al. 2016] y a partir de la mismas determinar cómo plantear las diferentes plantillas.
 - Generar aplicaciones móviles empaquetadas que contengan un *Actividad Educativa Posicionada*. Estas aplicaciones luego podrían ser instaladas por los alumnos y lograr así que los docentes sean independientes pudiendo general todo el material que necesitan sin necesitar de un experto en tecnología para poder crear la aplicación. Para esto, se deberá analizar cuál será el tipo de aplicación que se generará porque no es trivial la derivación a código y que este sea flexible a todo lo que el docente quiere crear.
 - Agregar otros mecanismos de sensores de posiciones, por ejemplo, *beacons*²¹. Los cuales están actualmente en auge como sensor indoor. Contar con funciones previamente definidas las cuales puedan ir siendo agregadas por el usuario, esta posible extensión proviene de que existe comportamiento repetitivo e identificable en este tipo de Aplicaciones, pudiendo facilitar aun más la creación de las Actividades Educativas Posicionadas.
- Se podría contar con un repositorio de capas de contenidos, posiciones o ya contenidos posicionados, esto facilitaría que la herramienta las tome, y se pueda arrancar a usar esto como base de la creación.

²¹ Página de uno de los vendedores más populares de *Beacons*: www.estimote.com (Último acceso 6-8-2017)

- Actualizar la herramienta para que soporte versiones recientes de Android, como así también evaluar la posibilidad de extenderlo a otras plataformas móviles.

Bibliografía

- [Alconada Verzini, 2015] Alconada Verzini, F.M.: Herramienta para la creación de elementos posicionados. Tesina de grado en Licenciatura en Sistemas, Facultad de Informática (UNLP), 2015.
- [Emmanouilidis et al., 2013] Emmanouilidis, C., Koutsiamanis, R.A., & Tasidou, A.: Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications*, 36(1), 103-125. 2013.
- [Fidas et al., 2015] Fidas, C., Sintoris, C., Yiannoutsou, N., & Avouris, N.: A survey on tools for end user authoring of mobile applications for cultural heritage. In 6th International Conference on Information, Intelligence, Systems and Applications (IISA). IEEE, pp. 1-5, 2015. [Frain, 2012] Frain, B.: Responsive web design with HTML5 and CSS3. Packt Publishing Ltd, 2012.
- [Hargood et al., 2016] Hargood, C., Hunt, V., Weal, M.J., Millard, D.E.: Patterns of sculptural hypertext in location based narratives. In: 27th ACM Conference on Hypertext and Social Media, pp. 61-70. ACM, New York (2016)
- [Hyo-Jeong et al., 2008] Hyo-Jeong, S.O., Insu, K.I.M., & Chee-Kit, L.O.O.I. Seamless mobile learning: Possibilities and challenges arising from the Singapore experience. *Educational Technology International*, 9(2), 97-121. 2008
- [Johnson et al., 2011] Johnson, L., Smith, R., Willis, H., Levine, A., Haywood, K.: The 2011 Horizon Report. The New Media Consortium. 2011.
- [Lieberman et al., 2006] Lieberman, H., Paternò, F., Klann, M., & Wulf, V.: End-user development: An emerging paradigm. In *End user development*. Springer Netherlands, pp. 1-8, 2006.
- [Llitas, 2015] Llitas, A.B.: Un enfoque de modelado de Actividades Educativas Posicionadas que contemplan elementos concretos. Tesis de Magister en Tecnología Informática Aplicada en Educación, Facultad de Informática (UNLP). 2015.
- [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., & Gordillo, S. E. (2012). Juegos Educativos Móviles basados en posicionamiento: una guía para su conceptualización. In XLI JAIIO (Jornadas Argentinas de Informática).
- [Millard et al., 2013] Millard, D.E., Hargood, C., Jewell, M. O., & Weal, M. J.: 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*, ACM, pp. 109-118, 2013.
- [Pirker et al, 2014] Pirker, J., Gütl, C., Weiner, P., Garcia-Barrios, V.M., & Tomintz, M: Location-based Mobile Application Creator creating educational mobile scavenger hunts. In 2014 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), IEEE, pp. 160-164, 2014.
- [Pittarello, 2011] Pittarello, F. (2011). Designing a context-aware architecture for emotionally engaging mobile storytelling. *Human-Computer Interaction—INTERACT 2011*, 144-151.

- [Pittarello & Bertani, 2012]** Pittarello, F., & Bertani, L.: CASTOR: learning to create context-sensitive and emotionally engaging narrations in-situ. In Proceedings of the 11th International Conference on Interaction Design and Children, ACM pp. 1-10, 2012.
- [Santos et al., 2011]** Santos, P., Pérez-Sanagustín, M., Hernández-Leo, D., & Blat, J.: QuesTInSitu: From tests to routes for assessment in situ activities. Computers & Education, 57(4), 2517-2534. 2011.
- [Zibula and Majchrzak, 2012]** Zibula, A., & Majchrzak, T. A.: Cross-platform development using HTML5, jQuery Mobile, and PhoneGap: Realizing a smart meter application. In International Conference on Web Information Systems and Technologies, 2012, pp. 16-33.
- [Zimbello and Challiol, 2016]** Zimbello, A., & Challiol, C. (2016). Experimentando un uso no tradicional de PhoneGap. In XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016).
- [Zimbello et al., 2017]** Zimbello, A.M., Alconada Verzini, F.M., Challiol, C., Lliteras, A. B., & Gordillo, S.E.: Authoring tool for location-based learning experiences. In Proceedings of the 4th International Conference on Mobile Software Engineering and Systems, 2017, pp. 211-212.

Anexo A: Relevamiento de otras herramientas de maquetado

En este anexo se describen otras herramientas de maquetado relevadas, que al no aportar ningún concepto significativo, no fueron agregadas en la Sección 2.1. En la Tabla A.1 se puede apreciar una descripción de las mismas y se deja disponible su URL.

Tabla A.1: Otras herramientas de maquetado relevadas

Nombre de la herramienta	Descripción	URL
InVision	Es una herramienta del tipo web. Si bien no se permiten crear diseños directamente, se pueden cargar propios en la herramienta y luego añadir los puntos de acceso para transformarlos en maquetas interactivas. Tiene una amplia gama de herramientas para facilitar el flujo de trabajo, donde compañeros de equipo, clientes, o cualquier persona involucrada en la maqueta tiene la posibilidad de dejar comentarios y discusiones específicas en cada lugar.	https://www.invisionapp.com (Último acceso: 14-10-2016)
Axure	Es una herramienta del tipo desktop. Diseñada para prototipos avanzadas, permite crear prototipos de web responsives y de aplicaciones móviles. Permite agregar interacciones, se pueden importar imagines agregar colores, y elegir la transparencia de cada objeto.	http://www.axure.com (Último acceso: 14-10-2016)
Moqups	Es una herramienta del tipo web, es gratis (cuenta con la idea de usuarios Premium) y bastante simple para crear prototipos. En la versión gratis encontramos 31 componentes disponibles, tales como gráficos, iconos, botones o figuras. También permite usar imágenes propias. En la versión Premium se menciona que se tiene la oportunidad de trabajar en múltiples proyectos, compartirlos con otros usuarios, comentar los prototipos, crear Master Pages/ paginas maestras, y exportar los proyectos finalizados en formato pdf o png.	https://moqups.com (Último acceso: 14-10-2016)

<p>Mockplus</p>	<p>Es una herramienta del tipo desktop, que se puede descargar en nuestra computadora y se permite trabajar con ella sin conexión a internet.</p> <p>En Mockplus se pueden dibujar y crear fácilmente diagramas funcionales, como así también diseñar mediante el método drag and drop prototipos interactivos. Posee alrededor de 200 componentes confeccionados junto con un conjunto de iconos con el propósito de facilitar el trabajo.</p> <p>Permite realizar un preview del prototipo creado mediante la lectura de un código QR, el cual puede ser compartido para que otros accedan a él.</p>	<p>http://www.mockplus.com (Último acceso: 14-10-2016)</p>
<p>Marvel</p>	<p>Es una herramienta web, permite crear prototipos de páginas web y de aplicaciones móviles. No presenta componentes ya confeccionados, sino que tenemos la opción de subir nuestras imágenes en formato psd, png o gifs para luego transformarlas en prototipos interactivos y clickeables con una serie de gestos, animaciones y transiciones.</p>	<p>https://marvelapp.com (Último acceso: 14-10-2016)</p>

Experimentando un uso no tradicional de PhoneGap

Agustina M. Zimbello¹ and Cecilia Challiol^{1,2}

¹ LIFIA, Facultad de Informática, UNLP, 50 y 120 Primer Piso,
La Plata, Argentina
{azimbello,ceciliac}@lifia.info.unlp.edu.ar
<http://www.lifia.info.unlp.edu.ar>

² CONICET, Argentina

Resumen. En este trabajo presentamos un plugin de *PhoneGap* para Android que permite una comunicación genérica entre los componentes visuales de una aplicación *PhoneGap* y una clase que funciona como punto de entrada de algún modelo Java. Dicho plugin fue utilizado para la creación de una aplicación móvil basada en posicionamiento. En este trabajo se brindan detalles de nuestro plugin, como así también aspectos aprendidos durante el desarrollo y futuros usos del mismo.

Palabras Claves: PhoneGap, Aplicaciones Móviles basadas en posicionamiento, Prototipado, Computación Móvil.

1 Introducción

El avance de la tecnología ha traído en consecuencia la creciente y variada gama de dispositivos y plataformas móviles disponibles (Android, iOS, etc.). Esto genera un desafío a la hora de desarrollar aplicaciones móviles, sobre todo poder cubrir esta variada gama en corto tiempo. Varios autores (por ejemplo, [3] y [13]) coinciden en que se pueden identificar tres posibles enfoques de desarrollo: aplicaciones móviles nativas, aplicaciones web móviles o aplicaciones móviles híbridas. La elección de cada uno de ellos conlleva ventajas y desventajas [3]. Es decir, las características propias de la aplicación móvil a desarrollar determinarán cuál es el enfoque que se ajuste mejor a la solución buscada.

Las aplicaciones móviles basadas en posicionamiento, y más aún aquellas sensibles al contexto [8], poseen aspectos particulares que están asociados, en muchos casos, con los datos que proveen los sensores de los dispositivos móviles. Varios autores coinciden ([3],[11],[13]), que para lograr una solución multiplataforma y además poder acceder a los sensores, la mejor opción es optar por desarrollar aplicaciones móviles híbridas. Actualmente, existen diferentes plataformas para crear este tipo de aplicaciones (móviles híbridas). Varios autores (por ejemplo, [3],[4],[7],[9],[12] y [13]) presentan y describen comparaciones de algunas de ellas. En todos estos trabajos, los

autores coinciden en que *PhoneGap*¹ (distribución libre de *Apache Cordova*²) es la plataforma más popular, con más documentación y cuenta con la posibilidad de ser extendida mediante la definición de plugins. *PhoneGap* es un framework que se basa en tecnología web estándar como: HTML, JavaScript y CSS. Cabe destacar que usar esta tecnología permite realizar prototipos rápidos.

En nuestro grupo de investigación hace varios años que venimos trabajando en la temática de aplicaciones móviles sensibles al contexto ([5],[6]). En el 2014 nos invitaron a participar del evento *TEDx Diagonal73*³, que se desarrolló en La Plata. La participación consistía en poner en práctica durante el evento una aplicación móvil basada en posicionamiento, la cual denominamos *Caminos Alternativos*. Para darle más atractivo a la aplicación la combinamos con actuaciones teatrales, las cuales estaban relacionadas con las consignas que se le presentaban al usuario desde la aplicación. En la puesta en práctica de la aplicación había actores que interpretaban escenas en diferentes lugares del edificio. Más detalles de la puesta en práctica de esta aplicación se mencionan en [1]. Es importante mencionar que esta aplicación fue desarrollada usando *PhoneGap* de una manera no tradicional. La motivación de este trabajo es precisamente esta, poder compartir el aprendizaje que experimentamos durante este desarrollo.

El objetivo del trabajo es describir un uso no tradicional de *PhoneGap*. En este trabajo presentaremos un plugin de *PhoneGap*, para Android, que permite una comunicación genérica entre los componentes visuales de una aplicación *PhoneGap* y una clase que funciona como punto de entrada de algún modelo de dominio implementado en Java. Este plugin fue desarrollado motivados por la necesidad de contar con aplicaciones puramente clientes que hicieran usos de aspectos internos del dispositivo móvil (por ejemplo, la cámara). Y, además, poder reusar modelos de dominios existentes (implementados en Java) con funcionalidad específica para aplicaciones móviles basadas en posicionamiento. La necesidad de contar con un plugin viene dada por la forma en que funciona *PhoneGap*, ya que requiere de plugins para comunicar la vista con clases Java. En nuestro caso particular, queríamos conectar las vistas con modelos de dominios que ya teníamos implementados. Y dado que estos modelos tenían diferentes puntos de entradas (clases Java específicas) queríamos encontrar una forma genérica de solución, y ahí es donde surge el plugin genérico que presentamos en este trabajo. La decisión de elegir *PhoneGap* fue aprovechar la ventaja de poder desarrollar usando las tecnologías HTML, JavaScript y CSS, sin tener que aprender aspectos particulares de la definición de las vistas, por ejemplo, en Android. En este trabajo se brindan detalles de nuestro plugin como así también aspectos aprendidos que pueden servir para otros desarrolladores.

La estructura del trabajo es la siguiente. En la Sección 2 se presentan las características de la problemática que se busca resolver. En la Sección 3 se presenta el plugin de *PhoneGap* para Android propuesto. Los resultados obtenidos y las lecciones aprendi-

¹ Página de *PhoneGap*: <http://phonegap.com> (Último Acceso: 13-5-2016)

² Página de *Apache Cordova*: <https://cordova.apache.org> (Último Acceso: 13-5-2016)

³ Página de *TEDx Diagonal7*: <https://www.ted.com/tedx/events/10077> (Último Acceso: 13-5-2016)

das se presentan en la Sección 4. Las conclusiones y trabajos futuros se presentan en la Sección 5.

2 Características de la Problemática a Resolver

En esta sección se describirán algunas características del contexto en el cuál trabajamos y cómo llegamos a usar *PhoneGap* de una forma no tradicional. Como mencionamos anteriormente venimos investigando temáticas afines a las aplicaciones móviles basadas en posicionamiento, en particular, distintos aspectos relacionados al diseño y modelado de las mismas. Por lo tanto, en la aplicación *Caminos Alternativos* presentada en [1] queríamos poner en práctica aspectos de modelado en los cuales veníamos trabajando, por ejemplo, la separación en capas de los contenidos de las posiciones.

Como mencionamos anteriormente *PhoneGap* se basa en HTML, JavaScript y CSS, por lo tanto, toda la lógica de control queda contenida en los JavaScript. Este es el uso tradicional de *PhoneGap*, sin embargo, esto nos generaba pasar nuestros modelos de clases implementados en Java a código JavaScript. Estos modelos de clases los venimos desarrollando como parte de un framework Java que estamos definiendo para brindar soporte en la creación de aplicaciones móviles basadas en posicionamiento [2].

Por otro lado, *PhoneGap* provee el concepto de plugin para extender funcionalidad. Estos plugins proveen una interfaz JavaScript a los componentes nativos del dispositivo (por ejemplo, para acceder a los sensores), permitiendo así que la aplicación pueda usar características nativas del dispositivo más allá de lo que está disponible para aplicaciones web puras. Cada plugin es desarrollado para una plataforma particular, por ejemplo, Android. Los plugins para Android se definen con una clase Java (que posee la funcionalidad para acceder, por ejemplo, a un sensor), un archivo JavaScript (que especifica cómo se debe llamar el plugin desde los HTMLs) y, por último, un archivo XML (el cual permite especificar la relación entre el llamado del plugin con la clase puntual a ejecutarse). Actualmente, *PhoneGap* provee plugins generales que pueden ser descargados y utilizados, por ejemplo, para acceder a la cámara del dispositivo o comunicarse con una base de datos.

Tomando este concepto de plugins, como se mencionó en la Sección 1, nos surgió la idea de usarlo para acceder a clases Java que pudieran servir como punto de entrada de nuestros modelos, y así poder reusar nuestros modelos de clases. Los plugins en *PhoneGap* están definidos para acceder a una clase puntual invocando un método puntual con sus parámetros. Esto nos generaba que por cada método que fuera punto de entrada de nuestro modelo debíamos crear un nuevo plugin, algo que claramente no escalaba. Esto nos motiva a crear una solución genérica, y la misma es presentada en este trabajo en la siguiente sección.

3 Plugin Desarrollado para PhoneGap

El plugin general que desarrollamos está implementado para Android. Al definirlo de manera genérica, puede ser usado para comunicar cualquier JavaScript de la aplicación con alguna clase Java (que sea parte de un modelo). Esta característica nos permite que pueda ser usado con cualquier modelo de dominio. Como se mencionó en la sección anterior, cualquier plugin en *PhoneGap* para Android se debe definir con alguna clase Java, un archivo JavaScript y un archivo XML.

Para entender mejor el funcionamiento del mismo se presenta en la Figura 1 la comunicación que se establece entre la vista (HTML/JavaScript), nuestro plugin y un modelo. La característica genérica nos permite tener definida tanto la vista como el modelo acorde a los requerimientos propios de cada aplicación.

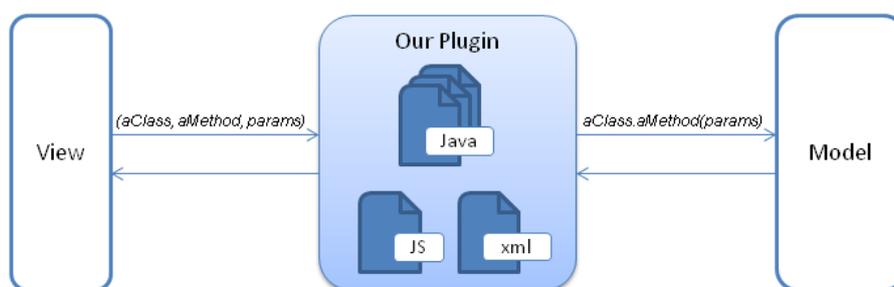


Fig. 1. Comunicación relacionada a nuestro plugin.

Se puede apreciar en la Figura 1 que nuestro plugin es invocado con: *aClass* (representa el nombre de la clase a la cual se desea invocar un método), *aMethod* (representa el nombre del método que se desea ejecutar, el mismo debe existir en el momento de invocarlo) y *params* (representa los parámetros a enviarse en el método indicado, es una colección con la forma `<tipo, valor>`). Esta información que llega a nuestro plugin es usada con `reflection`⁴ de Java, para invocar el método de la clase con los parámetros correspondientes. La respuesta brindada por esta clase es enviada a la vista que realizó la invocación. Por ahora, el plugin está implementado para enviar y recibir información en formato JSON (*JavaScript Object Notation*), pero este aspecto está diseñado para ser extendido por cualquier otro formato.

Cabe destacar que se realizó un análisis para determinar si `reflection` era la mejor forma de solucionar nuestro problema. Si bien `reflection` puede ocasionar errores en tiempo de ejecución, que no son detectados en compilación, la flexibilidad de invocar cualquier método de cualquier clase en forma dinámica se ajustaba perfectamente a la

⁴ Para más información sobre `reflection` se puede consultar:

<https://docs.oracle.com/javase/tutorial/reflect> (Último Acceso: 13-5-2016).

problemática que teníamos que dar solución. Por ende, este fue el motivo que prevaleció en la elección del uso de reflection.

A continuación, se presenta el código relacionado a los archivos JavaScript y XML de nuestro plugin como así también la clase Java del mismo.

En la Figura 2 se puede apreciar la definición del archivo JavaScript; se puede observar que el plugin debe ser invocado usando una función que recibe una función de *success* y otra de *error* (estas funciones son invocadas para devolver el control al JavaScript que invocó nuestro plugin), como así también los parámetros ya mencionados en la Figura 1.

```
var pluginHandler = {
  pluginHandler: function(successCallback, errorCallback, theClass, theMethod, arrayParameters) {
    cordova.exec(successCallback,
      errorCallback,
      "PluginHandler",
      "executePlugin",
      [
        {
          "theClass": theClass,
          "theMethod": theMethod,
          "arrayParameters": arrayParameters,
        }
      ]
    );
  }
}
module.exports = pluginHandler;
```

Fig. 2. Archivo JavaScript de nuestro plugin.

En la Figura 3 se puede apreciar la clase Java que representa nuestro plugin, la cual denominamos *PluginHandler*. Esta clase extiende de *CordovaPlugin*, que es la clase que se debe extender para representar un plugin personalizado en *PhoneGap*. Se debe implementar el método `execute()` como se puede visualizar en la figura. Si bien contamos con otras clases que colaboran con la clase *PluginHandler*, a fin de entender el funcionamiento y para simplificar la descripción no entraremos en detalle.

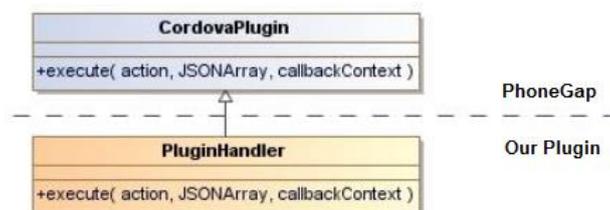


Fig. 3. Clase *PluginHandler*.

En la Figura 4 se puede apreciar el código del archivo XML de nuestro plugin, donde se establece la relación entre el archivo JavaScript y la clase *PluginHandler*.

```

<plugin xmlns="http://www.phonegap.com/ns/plugins/1.0"
  id="com.lifia.plugin.pluginHandler"
  version="0.1.0">
  <name>pluginHandler</name>
  <description> </description>
  <license> </license>
  <keywords>pluginHandler</keywords>
  <js-module src="www/pluginHandler.js" name="pluginHandler">
  <lobbers target="pluginHandler"/>
  </js-module>
  <!--android-->
  <platform name="android">
  <config-file target="res/xml/config.xml" parent="/*">
  <feature name="PluginHandler">
    <param name="android-package"
      value="com.lifia.plugin.pluginHandler.PluginHandler" />
  </feature>
  </config-file>
  <source-file src="src/android/PluginHandler.java"
  target-dir="src/com/lifia/plugin/pluginHandler" />
  </platform>
</plugin>

```

Fig. 4. Archivo XML de nuestro plugin.

De esta manera, queda definido nuestro plugin de *PhoneGap* genérico (para Android) que puede ser usado con cualquier modelo de dominio, más allá del que probamos, en particular, para la aplicación presentada en [1].

4 Resultados Alcanzados y Lecciones Aprendidas

Para lograr que nuestro plugin interactuara con cualquier clase Java (que sea punto de entrada de un modelo), tuvimos que restringir a que los métodos invocados reciban datos simples (por ejemplo, *String*) o a lo sumo colecciones con datos simples. Como mencionamos anteriormente la variable *params* que recibe el plugin al ser invocado es una colección con la forma <tipo, valor>. Esto nos permite transformar el *String* que se recibe en el JSON al tipo correspondiente y así hacer la invocación del método (de la clase Java) con los parámetros con el tipo adecuado. Sin esta especificación todas las clases Java (del modelo a invocar) debían definir sus métodos recibiendo solo *String*. Con esta característica damos más flexibilidad a las clases que son invocadas usando nuestro plugin.

Una desventaja de usar plugins, es que se pierde la característica multiplataforma que propone *PhoneGap*, ya que en nuestro caso solo definimos una solución para Android. Es decir, para brindar soporte multiplataforma se debería desarrollar un plugin específico para cada una de ellas. Esto requiere incorporar conocimientos de cada una de las plataformas para las cuales se quiera desarrollar un plugin. En nuestro caso, al querer usar este desarrollo para el prototipado solo nos limitamos a desarrollarlo para Android.

Nuestro plugin se pudo probar en una aplicación específica cómo se presentó en [1]. Esto nos permitió probarlo con un modelo particular, pero además poder proyectar su uso en otro tipo de aplicaciones como son aquellas aplicaciones móviles educati-

vas⁵ basadas en posicionamiento que es un área que también venimos trabajando. Cabe destacar que la aplicación presentada en [1], además de usar nuestro plugin, hace uso de un plugin provisto por *PhoneGap*, que permite la lectura de códigos de barra (llamado *BarcodeScanner*⁶). Este último plugin, abre la cámara, detecta que se está leyendo un código y devuelve el valor leído. Usamos códigos QR (Quick Response) para determinar dónde está posicionado el usuario y acorde a esto brindarle información o servicios.

Contar con el plugin para Android presentado en este trabajo nos permitirá en un futuro focalizarnos solo en la parte visual y de modelado, mientras que la comunicación entre estos estará dada por nuestro plugin. Esto nos permitirá agilizar los tiempos de prototipado, y a su vez reusar modelos que ya tenemos definidos para dominios específicos. Por otro lado, podremos trabajar con tecnologías HTML, JavaScript y CSS para definir las vistas, sin requerir aprender aspectos particulares de la definición de vistas en Android.

Varios de los prototipos que hemos desarrollado son probados en lugares donde no hay conectividad (acceso a internet), lo cual, nos llevó a desarrollar aplicaciones clientes, en particular, en Android. Hasta el momento, las aplicaciones que venimos prototipando están haciendo uso solo del posicionamiento del usuario, en un futuro se incorporará el uso de otros sensores. Esto seguramente requerirá un análisis del plugin presentado para lograr también agilizar el uso de dichos sensores. La ventaja que tiene *PhoneGap* en este sentido es que al contar con el concepto de plugin se puede acceder a cualquier aspecto interno del dispositivo, por ejemplo, sensores. Otro aspecto a mencionar, es que las aplicaciones que hemos desarrollado no requieren intercambio de información entre usuarios; esto es un tema a explorar, y también requerirá un análisis del plugin presentado para poder incorporar esta característica.

Es recomendable antes de hacer cualquier desarrollo, evaluar cuáles de las soluciones disponibles se ajusta mejor a los requerimientos de la aplicación que se quiere desarrollar. Se recomiendan lecturas actuales como, por ejemplo, [10] para tener en cuenta otros aspectos que pueden ser críticos para algunos desarrollos como pueden ser el consumo de energía o la seguridad de la información.

6 Conclusiones y Trabajos Futuros

En este trabajo se presentó un plugin de *PhoneGap* para Android que permite una comunicación genérica entre la vista de una aplicación y una clase que funciona como

⁵ Venimos trabajando en el modelado de este tipo de aplicaciones, más información se detalla en Lliteras, A.B.: “*Un enfoque de modelado de Actividades Educativas Posicionadas que contemplan elementos concretos*”. Tesis de Magister en Tecnología Informática Aplicada en Educación, Facultad de Informática (UNLP), 2016

<http://sedici.unlp.edu.ar/handle/10915/50030> (Último Acceso: 13-5-2016)

⁶ Más información sobre el plugin *BarcodeScanner* se detalla en:

<https://build.phonegap.com/plugins/951> (Último Acceso: 13-5-2016)

punto de entrada de un modelo Java. Se describió el funcionamiento del mismo, como así también se indicó que dicho plugin se utilizó en la aplicación *Caminos Alternativos* (la puesta en práctica de la misma se presentó en [1]).

El trabajo presentado es producto de haber detectado ciertos aspectos repetitivos en el prototipado de aplicaciones móviles basadas en posicionamiento, y acorde a eso, lograr agilizar esta tarea, en pos de reducir el tiempo de desarrollo. Además, se buscaba no tener que aprender aspectos puntuales de Android, por ejemplo, cómo definir las vistas; usando *PhoneGap* podemos usar tecnologías como HTML, JavaScript y CSS. En un futuro, con la solución presentada en este trabajo, nos focalizaremos solo en la especificación de las vistas como así también del modelo de la aplicación en sí.

Todavía nos queda explorar otros aspectos como, por ejemplo, la comunicación entre usuarios. Para esto se analizará, por ejemplo, el uso de NFC para que dos usuarios puedan intercambiar información sin requerir conectividad (acorde al tipo de aplicaciones que venimos probando). Otro tema a explorar es el uso de sensores, para esto, se tomará como punto de partida los plugin que ya provee *PhoneGap* para Android. Se analizará si los mismos se necesitan generalizar y cuál es la mejor forma para combinarlos con nuestro plugin genérico. Esto nos permitirá crear aplicaciones móviles sensibles al contexto, por ejemplo, usando sensores internos del dispositivo.

Estamos trabajando para que el plugin presentado nos agilice el empaquetado de aplicaciones móviles basadas en posicionamiento definidas usando la herramienta presentada en [2]. Por ejemplo, agilizar el prototipado de aplicaciones móviles educativas basadas en posicionamiento, para poder realizar pruebas más fácilmente y con distintas características.

Referencias

1. Alconada Verzini, F.M., Tonelli, J.I., Challiol, C., Lliteras, A.B., Gordillo, S.E.: Combing Location-Aware Applications with in-situ Actors Performances. In: the 2015 Workshop on Narrative & Hypertext, pp. 27-31. ACM, New York (2015)
2. Alconada Verzini, F.M., Tonelli, J.I., Challiol, C., Lliteras, A.B., Gordillo, S.E.: Authoring Tool for Location-Aware Experiences. In: the 2015 Workshop on Narrative & Hypertext, pp. 21-25. ACM, New York (2015)
3. Brucker, A. D., Herzberg, M.: On the Static Analysis of Hybrid Mobile Apps. In: Engineering Secure Software and Systems, pp. 72-88. Springer International Publishing (2016)
4. Dalmasso, I., Datta, S. K., Bonnet, C., Nikaen, N.: Survey, comparison and evaluation of cross platform mobile application development tools. In: 9th International Wireless Communications and Mobile Computing Conference, pp. 323-328. IEEE (2013)
5. Fortier, A., Challiol, C., Fernández, J.L., Robles, S., Rossi, G., Gordillo, S.: Exploiting personal web servers for mobile context-aware applications. *The Knowledge Engineering Review* 29(02), 134-153 (2014)
6. Fortier, A., Rossi, G., Gordillo, S., Challiol, C.: Dealing with variability in context-aware mobile software. *Journal of Systems and Software* 83(6), 915-936 (2010)

7. Heitkötter, H., Hanschke, S., Majchrzak, T.A.: Evaluating cross-platform development approaches for mobile applications. In: Web information systems and technologies, pp. 120-138. Springer-Verlag Berlin Heidelberg (2012)
8. Hong, J.Y., Suh, E.H., & Kim, S.J.: Context-aware systems: A literature review and classification. *Expert Systems with Applications* 36 (4), 8509-8522 (2009)
9. Jobe, W.: Native Apps vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies* (4), 27-32 (2013)
10. Nagappan, M., Shihab, E.: Future trends in software engineering research for mobile apps. In: 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering. (2016) <http://www.se.rit.edu/~mei/publications/pdfs/Future-Trends-in-Software-Engineering-Research-for-Mobile-Apps.pdf>
11. Palmieri, M., Singh, I., Cicchetti, A.: Comparison of cross-platform mobile development tools. In: 16th International Conference on Intelligence in Next Generation Networks, pp. 179-186. IEEE (2012)
12. Lim, S.: Experimental Comparison of Hybrid and Native Applications for Mobile Systems. *Int. J. Multimed. Ubiquitous Eng* 10(3), 1-12 (2015)
13. Xanthopoulos, S., Xinogalos, S.: A comparative analysis of cross-platform development approaches for mobile applications. In: 6th Balkan Conference in Informatics, pp. 213-220. ACM, New York (2013)

Authoring Tool for Location-based Learning Experiences

Agustina M. Zimbello¹, Federico M. Alconada Verzini¹, Cecilia Challiol^{1,2}, Alejandra B. Lliteras^{1,3},
Silvia E. Gordillo^{1,3}

¹UNLP, Facultad de Informática, LIFIA
La Plata, Buenos Aires, Argentina

²CONICET
Argentina

³CICPBA
Buenos Aires, Argentina

e-mails: {azimbello, falconada, ceciliac, lliteras, gordillo}@lifia.info.unlp.edu.ar

Abstract—In this paper, we present some features of our in-situ indoor-outdoor authoring tool for location-based learning experiences. For example, how to reuse locations in different experiences or how to design from scratch in places without any location sensing mechanisms. In addition, we describe some features in which we are working on. We expect to create a space for discussion about these kind of tools and to reflect on the products created with them.

Location-based Learning Experiences; In-situ Authoring Tool; Reuse Locations; Mobile Learning

I. INTRODUCTION

The technological advances have increased the scenarios of new learning practices [1]. In particular, location-based learning experiences allow learners to explore the environment and receive, on their mobile devices, learning activities in different locations. So, these kind of experiences are in-situ, what means that learning activities can be more challenging [1-2]. In some cases, the accuracy of the location is vital for the learner. For example, if the learner has to observe a specific plant, the activity should be triggered in front of it [1]. Then, some activities might require observing or manipulating physical objects in specific locations (as mentioned in [3]). This is why the design process has to consider not only location-based features but also physical objects.

Location-based learning experiences can be designed virtually or in-situ, as mentioned in [2]. Virtual design is based in a map to set the relevant locations in which the learners receive the learning activities [2]. On the other hand, in-situ design (as in [4]) implies visiting the real physical place, creating more precise experiences (by allowing teachers, for example, to observe the place from the learners' perspective [5] or to better estimate the time to move from one location to another [6]). Note that all of these mentioned works are focused on outdoor experiences and contents are couple with GPS locations, hence, reuse of both contents and locations separately is not possible.

Some authors explore implications related to in-situ authoring in outdoor environments, as in [7], where it is mentioned that this is yet an open area to research. Moreover, in-situ indoor authoring tools have not been thoroughly researched so this is one of our motivations and

why we are working on an in-situ indoor-outdoor authoring tool for location-based learning experiences which is an extension of [8].

The purpose of this paper is to create a space to discuss some features of the in-situ authoring tools. We use our authoring tool to show some features that we have been working on (e.g. reusable locations) and to describe a possible solution in places without location sensing mechanisms. Thus, the design is built from scratch. In addition, we hope that this paper could make a contribution to reflect on the products created with authoring tools (proceed from ad-hoc experiences to reusable solutions [9]).

The paper is structured as follows. In Section 2, we present and discuss some features of our in-situ indoor-outdoor authoring tool for location-based learning experiences. Conclusions and some future works are presented in Section 3.

II. IN-SITU AUTHORING TOOL

In this section, we present and discuss some features of our in-situ authoring tool which is in an initial phase (only tested by developers). As mentioned before, it is an extension of [8], so it has been defined with a web responsive design. We use the concept of separation of concerns to define learning activities and locations. These concerns could be seen as separate layers that can be used in different location-based learning experiences. Then, each experience defines their points of interest (PoI) which associate learning activities with locations.

In Fig. 1 an example of a location layer inside of La Plata museum is shown. Each location is defined in an independent way, regardless of any domain or sensing mechanism and being able to reuse them in different location-based learning experiences (what is not possible on other existing tools such as [2] and [4]). So far, the tool uses an image to represent indoor spaces (as shown in Fig. 1), what can be easily done by teachers. In the future, we will expect to evolve to more precise representations such as, for example, Google Maps Indoor [10]. This begs the question: would these different space representations be easy to use by teachers in terms of the usability? And in a broader sense, how can these authoring tools be developed so as their users (e.g. teachers) do not require technical knowledge and can easily be used by them?

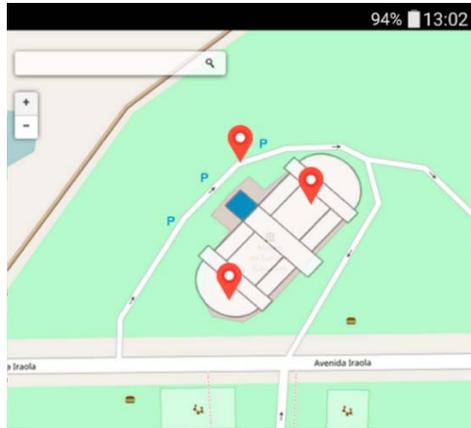


Figure 1. Example of location layer.

Our tool provides the possibility to set the mechanism to trigger learning activities for each location (currently, some locations could be triggered using GPS but others using QR codes). Then, this information is used in a location-based learning experience to trigger the learning activity in each location. These trigger mechanisms are defined in a separate layer, in order to reuse the location layer with different mechanisms on each experience. Suppose that in Fig. 1 the selected trigger mechanism for all locations is QR codes. Then, the tool will create as many QR codes as locations defined and teachers will be able to use these codes to set the physical place for each location. This is a simple way to create a location-based learning experience from scratch (for example, in places without any location sensing mechanisms). In addition, QR codes provide more accuracy to outdoor design by being able to observe a part of the physical place. Considering that indoor spaces may already count with sensing mechanisms, dynamically integrating them with these tools is definitely an open question.

Different structures to relate the contents have been identified in [11]. The current version of our tool provides the option to create linear, graph or set structures.

We have defined a model solution for location-based learning experiences which involves physical objects manipulation to resolve learning activities [12]. For the time being, our tool exports both content and location layers into XML files. We are working on integrating this model to generate executable experiences with our tool. Note that, the design of these specific learning activities requires to define a way to identify each physical object. To do that, we are developing a way to provide, from our tool, QR codes for each physical object associated to each learning activity. Using the prototype *LearningPlaying* (“*Aprendo Jugando*”) defined in [12], we detected some templates related to these kind of experiences and we will be adding this as part of our tool to contribute on the creation and design.

III. CONCLUSIONS AND FUTURE WORK

In this paper, we have mentioned some features of our in-situ indoor-outdoor authoring tool for location-based

learning experiences. We have described how to use separation of concerns to reuse the locations in different experiences. In addition, we have presented the use of QR codes as a way to build from scratch in places without any location sensing mechanisms. Furthermore, we have described some features in which we are working on and we are planning to incorporate other ways of handling indoor positioning.

In the near future, we will organize some tests of proof with teachers to help us to complete the design of our tool (in a way that they are involved in the design process [5], in this case, in the one of our tool). We will analyze different roles identified with the design phase (as mentioned in [3], for example, learner, instructor, researcher and developer) and how these roles could be supported by our tool.

REFERENCES

- [1] H.C. Chu, G.J. Hwang, C.C. Tsai, and J. Tseng, “A two-tier test approach to developing location-aware mobile learning systems for natural science courses,” *Computers & Education*, vol. 55, no. 4, Dec. 2010, pp. 1618-1627, doi: 10.1016/j.compedu.2010.07.004.
- [2] P. Santos, D. Hernández-Leo, and J. Blat, “To be or not to be in situ outdoors, and other implications for design and implementation, in geolocated mobile learning,” *Pervasive and Mobile Computing*, vol. 14, Oct. 2014, pp. 17-30, doi: 10.1016/j.pmcj.2013.09.001.
- [3] E. Gilman, I.S. Milara, M. Cortés, and J. Riekkii, “Towards user support in ubiquitous learning systems,” *IEEE Transactions on Learning Technologies*, vol. 8, no. 1, Dec. 2015, pp. 55-68, doi: 10.1109/TLT.2014.2381467.
- [4] F. Pittarello and L. Bertani, “Castor: Designing and experimenting a context-aware architecture for creating stories outdoors,” *Journal of Visual Languages & Computing*, vol. 25, no. 6, Dec. 2014, pp. 1030-1039, doi: 10.1016/j.jvlc.2014.10.013.
- [5] R. Cober, E. Tan, J. Slotta, H.J. So, and K.D. Könings, “Teachers as participatory designers: Two case studies with technology-enhanced learning environments,” *Instructional Science*, vol. 43, no. 2, Mar. 2015, pp. 203-228, doi: 10.1007/s11251-014-9339-0.
- [6] T.Y. Hsu, C.K. Chiou, J.C. Tseng, and J.G. Hwang, “Development and evaluation of an active learning support system for context-aware ubiquitous learning,” *IEEE Transactions on Learning Technologies*, vol. 9, no. 1, June 2016, pp. 37-45, doi: 10.1109/TLT.2015.2439683.
- [7] D.E. Millard and C. Hargood, C., “Location Location Location: experiences of authoring an interactive location-based narrative,” *Proc. 9th International Conference on Interactive Digital Storytelling (ICIDS 2016)*, Springer, Nov. 2016, pp. 419-422, doi:10.1007/978-3-319-48279-8_40.
- [8] F.M. Alconada Verzini, J.I. Tonelli, C. Challiol, A.B. Lliteras, and S.E. Gordillo, “Authoring tool for location-aware experiences,” *Proc. 2015 Workshop on Narrative & Hypertext (NHT '15)*, ACM, Sep. 2015, pp. 21-25, doi: 10.1145/2804565.2804570.
- [9] D. Weyns, M. Caporuscio, B. Vogel, and A. Kurti, “Design for Sustainability = Runtime Adaptation \cup Evolution,” *Proc. 2015 European Conference on Software Architecture Workshops (ECSAW '15)*, ACM, Sep. 2015, pp.62-69, doi: 10.1145/2797433.2797497.
- [10] <https://www.google.com/maps/about/partners/indoormap>
- [11] C. Hargood, V. Hunt, M.J. Weal, and D.E Millard, “Patterns of sculptural hypertext in location based narratives,” *Proc. 27th ACM Conference on Hypertext and Social Media (HT '16)*, ACM, July 2016, pp. 61-70, doi: 0.1145/2914586.2914595.
- [12] A.B. Lliteras, “Un enfoque de modelado de actividades educativas posicionadas que contemplan elementos concretos,” Master Thesis, Faculty of Informatics, National University of La Plata, Argentina, Nov. 2015, <http://hdl.handle.net/10915/50030>