

PROTOTIPO FUNCIONAL DE APLICATIVO MOVIL PARA ORIENTACIÓN EN  
LAS INSTALACIONES DE LA UNIVERSIDAD SAN BUENAVENTURA MEDELLIN  
SEDE SAN BENITO

MATEO GALLO MARTINEZ

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL MEDELLÍN

FACULTAD DE INGENIERÍAS

INGENIERIA DE SISTEMAS

MEDELLIN

2014

PROTOTIPO FUNCIONAL DE APLICATIVO MOVIL PARA ORIENTACIÓN EN  
LAS INSTALACIONES DE LA UNIVERSIDAD SAN BUENAVENTURA MEDELLIN  
SEDE SAN BENITO

MATEO GALLO MARTINEZ

Proyecto presentado para optar al título de Ingeniero de Sistemas

Asesor

Claudia Elena Durango Vanegas, Ing. de Sistemas

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL MEDELLÍN

FACULTAD DE INGENIERÍAS

MEDELLIN

2014

## TABLA DE CONTENIDO

<b>1</b>	<b>JUSTIFICACIÓN</b> .....	<b>7</b>
<b>2</b>	<b>PLANTEAMIENTO DEL PROBLEMA</b> .....	<b>8</b>
<b>3</b>	<b>OBJETIVO GENERAL</b> .....	<b>14</b>
<b>4</b>	<b>OBJETIVOS ESPECÍFICOS</b> .....	<b>14</b>
<b>5</b>	<b>MARCO REFERENCIAL</b> .....	<b>15</b>
5.1	¿QUÉ ES ANDROID?.....	15
5.2	SQLITE.....	15
5.3	REALIDAD AUMENTADA.....	16
5.4	CODIGOS QR.....	16
5.5	GPS Y SUS CARACTERISTICAS.....	16
5.6	SISTEMA DE INFORMACIÓN GEOGRAFICO (SIG).....	17
<b>6</b>	<b>ESTADO DEL ARTE</b> .....	<b>18</b>
<b>7</b>	<b>DESCRIPCION DE LA PROPUESTA</b> .....	<b>20</b>
<b>8</b>	<b>METODOLOGIA</b> .....	<b>22</b>
<b>9</b>	<b>CRONOGRAMA</b> .....	<b>24</b>
<b>10</b>	<b>CASOS DE USO</b> .....	<b>25</b>
10.1	Buscar aula.....	25
10.1.1	Documentación Caso de Uso Buscar aula.....	26
10.1.2	Interfaces.....	27
10.2	Escanear código QR.....	29
10.2.1	Documentación Caso de Uso Escanear código QR.....	30
10.2.2	Interfaces.....	32
<b>11</b>	<b>CODIFICACIÓN</b> .....	<b>38</b>
11.1	ACTIVITY PANTALLA1.XML.....	38
11.2	PANTALLA1.JAVA.....	39
11.3	ACTIVITY PANTALLA2.XML.....	40
11.4	PANTALLA2.JAVA.....	41
11.5	ADMINSQLITEOPENHELPER.JAVA.....	42
11.6	INTENTRESULT.JAVA.....	43
11.7	INTENTINTEGRATOR.JAVA.....	44
<b>12</b>	<b>RESULTADOS DE CASOS DE PRUEBA</b> .....	<b>53</b>
12.1	Buscar Salón.....	53

12.2	Buscar salón .....	55
<b>13</b>	<b>FUTUROS RESULTADOS .....</b>	<b>56</b>
<b>14</b>	<b>CONCLUSIONES .....</b>	<b>57</b>
14.1	Anotaciones especiales.....	57
<b>15</b>	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>58</b>
<b>16</b>	<b>ANEXOS.....</b>	<b>59</b>
16.1	ANEXO#1 (encuesta Nro1).....	59
16.2	ANEXO#2(encuesta Nro2).....	59
16.3	ANEXO#3(Código QR Salón 409-c).....	60
16.4	ANEXO#4(Código QR Auditorio Principal) .....	60

## Listado de Ilustraciones

<i>Mapa de evacuación</i> .....	9
<i>señalización</i> .....	9
<i>Datos encuesta</i> .....	10
<i>Datos encuesta 2</i> .....	11
<i>Datos encuesta 3</i> .....	12
<i>Diagrama de flujo</i> .....	21
<i>Metodología</i> .....	22
<i>Caso de uso UML buscar aula</i> .....	25
<i>Pantalla1</i> .....	27
<i>Pantalla2</i> .....	28
<i>Caso de uso UML escanear QR</i> .....	29
<i>Pantalla3</i> .....	32
<i>Pantalla4</i> .....	33
<i>Pantalla5</i> .....	34
<i>Pantalla6</i> .....	34
<i>Pantalla7</i> .....	35
<i>Pantalla8</i> .....	36
<i>Pantalla9</i> .....	37
<i>Prueba1</i> .....	53
<i>Prueba2</i> .....	54
<i>Prueba3</i> .....	55
<i>QR [9]</i> .....	56
<i>Anexo1</i> .....	59
<i>Anexo2</i> .....	59
<i>QR1</i> .....	60
<i>QR2</i> .....	60

**Listado de Tablas**

*Datos encuesta* ..... 10  
*Datos encuesta2* ..... 11  
*Datos encuesta3* ..... 12  
*cronograma* ..... 24  
*Tabla buscar aula* ..... 26  
*Tabla escanear QR*..... 30

## 1 JUSTIFICACIÓN

La realización del proyecto tiene una gran importancia para la universidad, puesto que el desarrollo de aplicaciones para dispositivos móviles tiene una gran prospectiva, adicionalmente en un estudio previo del estado del arte de la producción bibliográfica en la Universidad no se encontraron desarrollos similares, en lo que se refiere a desarrollos para dispositivos móviles; en esta temática se presentó un proyecto sobre un sistemas de información para dispositivos móviles que permitiera la administración de la gestión metrológica. Pero no se tiene un proyecto que convine los sistemas de información geográfica y las aplicaciones para dispositivos móviles. Por lo que este proyecto tendría un impacto muy significativo en la oferta de nuevas temáticas de estudio para los semilleros en el programa de ingeniería de sistemas en la universidad de san buenaventura.

Es un proyecto innovador que busca satisfacer la necesidad de la demanda de aplicaciones y cuya importancia radica en tres ramas, primero la necesidad de la Universidad de ser competente y no quedar relegada a los avances en sistemas de información; segundo es importante puesto que la solución que se propone es innovadora y relacionada con el mercado actual de tecnologías; tercero por su enfoque social, que contribuye a la comunidad estudiantil de manera práctica y desde un enfoque de la ingeniería de sistemas a mejorar el proceso académico, evitando retrasos en sus actividades curriculares.

Es importante que los ingenieros tengan un enfoque social, dando soluciones ingeniosas a las necesidades del día a día. Este proyecto propone entonces solucionar el inconveniente que representa llegar tarde a las clases por desconocimiento del aula, el desarrollo va más enfocado a estudiantes de primer semestre y visitantes que no tiene un conocimiento de la infraestructura de la universidad.

## 2 PLANTEAMIENTO DEL PROBLEMA

Actualmente se observa el impacto que tienen los dispositivos móviles o teléfonos inteligentes (Smartphone) en la sociedad, pues su versatilidad y conectividad le confiere virtudes en el entretenimiento, la información y el geoposicionamiento global. Las aplicaciones para dispositivos móviles son utilizadas en diversas actividades cotidianas de las personas para facilitar la búsqueda de una dirección en la ciudad; partiendo de la facilidad y la tendencia actual de usar estos desarrollos, el proyecto se basa en que las personas que concurren en la sede San Benito de la Universidad de San Buenaventura puedan por medio de un aplicativo, encontrar las aulas u oficinas a las cuales se deben dirigir durante su estadía en la sede universitaria.

Cada semestre a la Universidad llegan nuevos estudiantes y visitantes que probablemente no conocen las instalaciones ni bloques de la sede y tienen la necesidad de dirigirse a las aulas u oficinas. Como se puede observar en la ilustración 1, en la portería principal no se cuenta con un mapa que muestre ni oriente al usuario claramente en las instalaciones, este mapa enseña un plano de evacuación, que no supe las necesidades específicas para la orientación y búsqueda de las aulas de sus respectivas clases.



Figura 1: Mapa de evacuación

El único medio de ayuda con el que cuenta la universidad es una señalización con flechas, como muestra la ilustración 2, que señalan el camino hacia algunos auditorios, pero no supe tampoco la necesidad de orientación y ubicación dentro de la universidad



Figura 2: señalización

La falta de orientación puede tener consecuencias negativas en el proceso de aprendizaje, ya que por falta de información se puede llegar tarde a una clase afectando al estudiante, tal como lo muestran los resultados de algunas encuestas. Se encuestaron a 100 personas que concurrían las instalaciones de la Universidad de San Buenaventura Medellín sede San Benito, entre las cuales se

encontraban estudiantes de pregrado y estudiantes del Sena que utilizan la sede por convenio con la universidad. En la encuesta se consultó si en algún momento el usuario había tenido dificultades para encontrar un aula, y en promedio de cuanto era el tiempo aproximado de retraso para encontrar un salón determinado.

De la población de personas encuestadas, el 83% indicó que en algún momento durante su estadía en la USB habían llegado tarde debido al desconocimiento de la ubicación del aula donde tenían clase. En la tabla 1 y la ilustración 3 se muestran los resultados obtenidos en el ejercicio.

En los resultados arrojados en la encuesta se observa que tan solo el 17% de los encuestados indicó nunca haber llegado tarde a clase por este motivo mientras que el 83% restante indicó haber llegado tarde con diferentes lapsos de tiempo

Tabla 1: Datos encuesta

Tiempo de retraso	0 min	1-5 min	6-10 min	11-15 min	16 o mas
Número de personas	17	12	27	29	15

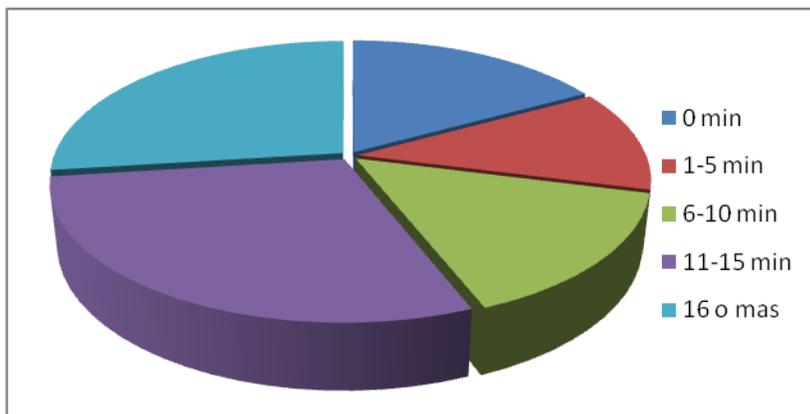


Figura 3: Datos encuesta

Del conjunto de personas que llegaron tarde, se encontró que el 14,45% tardó entre 1 y 5 minutos para ubicar su aula, el 18,07% se tardó entre 6 y 10 minutos, el 34,93% se tardó hasta 15 minutos, y el 32,53% mostró que se tardó 16 minutos o más en encontrar su salón. Según estos datos, la cifra es preocupante, ya que

alrededor del 67,49% llegaron en algún momento tarde a clase por la desorientación.

Se planteó una segunda encuesta. Se preguntó a una población de 100 personas concurrentes a la universidad si conocían los espacios físicos que se revelaban en las fotografías, con el propósito de verificar el conocimiento de las instalaciones por parte de los usuarios. Para esto se tomaron como referencia la oficina 409 C y la sala de cómputo # 11. (Imágenes en anexo)

En la tabla2 y la ilustración4 se muestran los resultados de la encuesta realizada sobre el reconocimiento de aulas y oficinas.

Tabla 2: Datos encuesta2

¿Reconoce en qué lugar de la universidad fue tomada la foto de la oficina 409 C?	SI	NO
Porcentaje	38%	62%

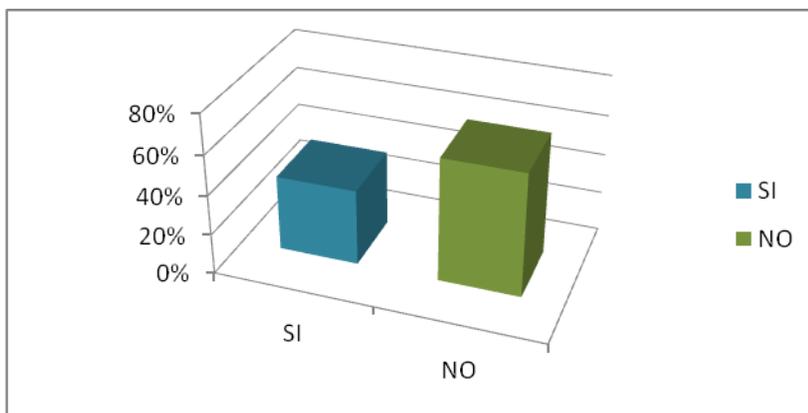


Figura 4: Datos encuesta 2

De los resultados arrojados en la segunda encuesta se encontró que el 62% de los encuestados indicó no conocer el la ubicación de la oficina 409 C y tan solo el 38% revelo saber dónde había sido tomada la fotografía.

Otro resultado puede verse en la tabla 3 y la ilustración 5 en la que se muestran las opiniones sobre la ubicación de la sala de cómputo # 11 a partir de una fotografía. Los resultados indicaron que solo el 26% de la población conoce la ubicación de la sala de cómputo, mientras que el 74% no reconoció en qué lugar fue tomada la fotografía.

Tabla 3: Datos encuesta 3

¿Reconoce donde se encuentra ubicada la sala de cómputo # 11?	SI	NO
Porcentaje	74%	26%

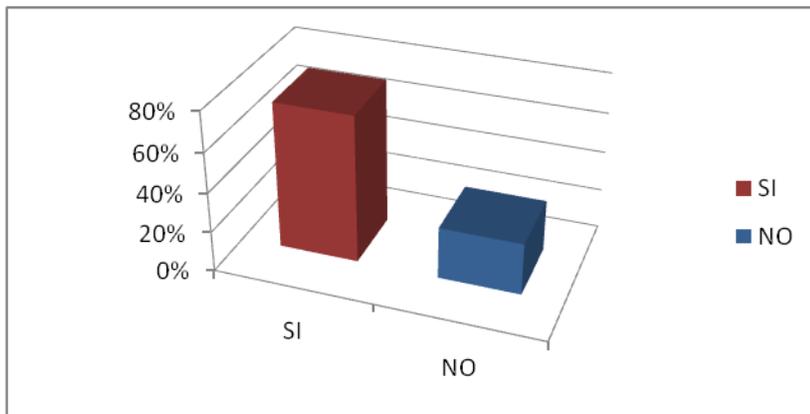


Figura 5: Datos encuesta 3

Con los resultados obtenidos se puede concluir que existe una gran parte de la comunidad estudiantil que tiene inconvenientes de ubicación dentro de la sede, lo que conlleva a inconvenientes como retrasos, y que existen locaciones que son de difícil ubicación y no muy conocidas por los estudiantes, por lo que sería de gran utilidad una aplicación para dar solución al problema de orientación de la sede.

Aunque la Universidad ya cuenta con algunos desarrollos enfocados en dispositivos móviles no hay aún un sistema posicionamiento, debido a factores que se presentan al pretender realizar este tipo de aplicación. Como el margen de

error de los GPS que pueden ser hasta de 20 metros, o la dificultad que supone el uso de la realidad aumentada. Para el proyecto se propone utilizar herramientas de georreferenciación que podrán ser modificadas según se vea la necesidad en el desarrollo de la aplicación.

Finalmente el desarrollo tecnológico ha conllevado a que la ingeniería de sistemas se plantee un nuevo desafío en las aplicaciones móviles para satisfacer las necesidades de las personas, por medio de aplicativos que faciliten actividades diarias, en este caso que se facilite su ubicación dentro de la Universidad San Buenaventura.

### **3 OBJETIVO GENERAL**

Desarrollar un prototipo funcional de una aplicación para dispositivos móviles mediante el sistema operativo Android, que le permita a los estudiantes encontrar los espacios físicos de la universidad san buenaventura Medellín

### **4 OBJETIVOS ESPECÍFICOS**

Identificar y definir los requisitos funcionales y no funcionales de la aplicación.

Implementar prototipos funcionales de forma iterativa y evolutiva obtener los ajustes de los requisitos y asegurar una mayor eficiencia de los datos.

Realizar las pruebas funcionales del sistema.

## 5 MARCO REFERENCIAL

### 5.1 ¿QUÉ ES ANDROID?

Android esencialmente es un sistema operativo para teléfonos inteligentes y tabletas, es decir que es el programa encargado de gestionar los recursos de los Smartphone, Android también permite a sus usuarios programar y desarrollar sus propias aplicaciones basado en un lenguaje de Java. Lo cual facilita el fin del proyecto, crear una aplicación para Android propia y ponerla a disposición de la universidad.

Android es un lenguaje de programación basado en Java y un sistema operativo para dispositivos embebidos y móviles, cuya parte superior de capas de Android están escritos en el lenguaje mismo. Como todo lenguaje, cuenta con una biblioteca basada en eventos prolongados y la inflación dinámica de vistas gráficas de los archivos de formato XML[1]

### 5.2 SQLITE

SQLite es una biblioteca en proceso que se implementa, sin servidor, sin configuración, el motor de base de datos transaccional de SQL es autónomo. El código para SQLite es de dominio público y por lo tanto es libre para el uso para cualquier propósito, comercial o privado. SQLite es un motor de base de datos SQL embebido. A diferencia de la mayoría de las otras bases de datos SQL, SQLite no tiene un proceso de servidor independiente. SQLite lee y escribe directamente en archivos de disco ordinarios. Una completa base de datos SQL con varias tablas, índices, triggers y vistas, está contenida en un archivo de disco único. El formato de archivo de base de datos es multiplataforma - usted puede copiar libremente una base de datos entre sistemas de 32 bits y de 64 bits o entre arquitecturas big-endian y little-endian. Estas características hacen de SQLite en una opción popular como un formato de archivo de la aplicación.[2]

Sqlite es el gestor de base de datos que se utilizara para el desarrollo de la aplicación y almacenar los salones oficinas y auditorios y la información que se desee almacenar de los espacios físicos de la universidad, aunque se podría trabajar con MySQL, PostgresSQL u Oracle, se toma la decisión de trabajar con Sqlite dado que Android permite trabajar sin un servidor de base de datos externo y dado que la base de datos requerida no es de gran tamaño Sqlite es la opción más óptima para trabajar con la aplicación

### **5.3 REALIDAD AUMENTADA**

Realidad aumentada se refiere a una imagen en directo y en el mundo real que se ha mejorado o disminuido por el contenido virtual a través de un interfaz de la cámara. La tecnología de la realidad aumentada tiene por objeto simplificar las tareas cotidianas, complementando la percepción del usuario y la interacción con el mundo real [3]

### **5.4 CODIGOS QR**

Un código QR (quick response code, «código de respuesta rápida») es un módulo útil para almacenar información en una matriz de puntos o un código de barras bidimensional. Se caracteriza por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector. Aunque inicialmente se usó para registrar repuestos en el área de la fabricación de vehículos, hoy los códigos QR se usan para administración de inventarios en una gran variedad de industrias. La inclusión de software que lee códigos QR en teléfonos móviles, ha permitido nuevos usos orientados al consumidor, que se manifiestan en comodidades como el dejar de tener que introducir datos de forma manual en los teléfonos. Las direcciones y los URLs se están volviendo cada vez más comunes en revistas y anuncios. El agregado de códigos QR en tarjetas de presentación también se está haciendo común, simplificando en gran medida la tarea de introducir detalles individuales de un nuevo cliente en la agenda de un teléfono móvil.[4]

Mediante la utilización de códigos QR para la aplicación se pretende que mediante un lector de códigos se pueda traer la información contenida en el QR de cómo se puede llegar al salón solicitado, estos códigos ofrecen una opción simple ágil y rápida para contener información que se encuentra almacenada en la nube o en cualquier URL.

### **5.5 GPS Y SUS CARACTERISTICAS.**

Sistema de posicionamiento global o GPS por sus siglas en inglés (Global Positioning System) en esencia es un navegador que permite saber la ubicación en la que se encuentra una persona o permite mostrar direcciones y llegar al destino que el usuario desee, para este propósito el GPS se conecta con satélites en órbita y utiliza la triangulación de la coordenadas para mostrar la ubicación del objeto, o persona.

El GPS inspira el desarrollo de este proyecto ya que permite mostrar la ubicación del usuario, pero por si solo no permite solucionar el problema planteado, puesto que el GPS maneja un margen de error que puede ser de algunos metros, lo cual

representaría un problema a la hora de encontrar una ruta hacia un salón y teniendo en cuenta que la diferencia entre salones también es de pocos metros.

#### **5.6 SISTEMA DE INFORMACIÓN GEOGRAFICO (SIG).**

Es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica

## 6 ESTADO DEL ARTE

Realizando búsquedas en bases de datos especializadas sobre desarrollos, prototipos y aplicativos similares se encontró que sistemas de navegación similares aunque implementados con diferentes tecnologías ya ha sido aplicado y probados en un ámbito universitario.

El artículo de la universidad católica de Fu-Jen propone un problema que es como ayudar a los visitantes a familiarizarse con un campus de la universidad, lo cual solucionaron con una herramienta de implementación de teléfono inteligente o una aplicación, se desarrolló un prototipo de un sistema para el recorrido del campus en la Universidad Católica Fu-Jen, fue realizado y revisado por dos grupos de enfoque (estudiantes de primer semestre). Se recopilaron datos para el levantamiento de requerimientos como entrevistas, y los pensamientos y las reacciones de estos dos grupos fueron resumidos y documentados.

Resultados de la investigación revelan que recorriendo el campus con un teléfono inteligente con el sistema de realidad aumentada proporciona información oculta en un entorno real, dando asistencia instantánea de los novatos que pueden haberse perdido, y es una herramienta de aprendizaje móvil personal[5]

Como se puede observar en dicha investigación se probó el impacto positivo que tiene un aplicativo de estas características para la población estudiantil, principalmente para estudiantes de primer semestre que no conocen el campus.

Para este tipo de proyectos se deben realizar los estudios y encuestas correspondientes a los estudiantes que serían el grupo de partida para el desarrollo y poder medir el impacto positivo que se puede tener con este tipo de aplicaciones para los dispositivos móviles.

De igual manera que el proyecto anteriormente mencionado, se puede encontrar un trabajo similar realizado en un contexto más cercano al de la USB y que fue realizado en la Escuela de ingenierías de Antioquia donde se decidió realizar un prototipo funcional para ayudar a los estudiantes de la Escuela de Ingeniería a revisar su horario y su salón de clase con una aplicación móvil, esta aplicación utiliza realidad aumentada para dar indicaciones visuales al usuario de hacia dónde debe dirigirse para llegar al bloque donde está el salón[6]

Realizando una comparación respecto a los proyectos de la universidad de católica Fu-Jen y de la EIA, con el proyecto que se proponer realizar en la

Universidad de San Buenaventura, se observa que es un proyecto completamente viable y que se puede realizar con el respaldo de la universidad, se plantea una problemática similar en ambos trabajos que es como mejorar y ayudar a estudiantes y visitantes respecto a la ubicación, necesidad que fue corroborada en la USB por medio de investigaciones consultas y entrevistas a las personas que serían beneficiadas con el aplicativo.

## **7 DESCRIPCION DE LA PROPUESTA**

La propuesta es realizar una aplicación para dispositivos móviles con sistema operativo Android, el desarrollo permite a los usuarios ubicarse y buscar salones de clase y oficinas en la sede de san Benito de la universidad de san buenaventura, la aplicación se basa en el uso del GPS y de la realidad aumentada.

Los resultados del proyecto se pueden evaluar según el impacto positivo que se tenga en los estudiantes y usuarios de la aplicación, teniendo como referencia entrevistas y opiniones de los beneficiarios, según la satisfacción y verificando la disminución de retrasos de los estudiantes para hallar las aulas, la forma de verificar sería por medio de encuestas y entrevistas al grupo de enfoque que serían los estudiantes de primer semestre, comparando los datos de llegadas tardes del grupo de control con los datos de las primeras entrevistas.

El proyecto tiene un enfoque social más que económico, puesto que no se pretende tener lucro por cada descarga de la aplicación, lo que se intenta es brindar un servicio gratuito y de calidad a la comunidad universitaria para dar solución de forma práctica y eficiente a un inconveniente que se presenta en la sede para la ubicación de ciertos espacios físicos.

Para la realización del proyecto se debe investigar las diferentes herramientas tecnológicas desarrolladas para el sistema operativo Android que están disponibles en los distintos sitios de Internet, ya que se desea desarrollar un aplicativo con medios lógicos para establecer un protocolo básico que integre herramientas del teléfono con el sistema operativo; Con base a esto se desea optimizar el nivel de información que actualmente posee la universidad sobre las aulas, auditorios y espacios.

Es importante la opinión del usuario en el proceso de desarrollo del aplicativo, dado que se realiza el proyecto pensando en la satisfacción del usuario final, a continuación se muestra el diagrama de flujo con los procesos para el desarrollo del software.



Figura 6: Diagrama de flujo

Es necesario también invertir tiempo en la capacitación del uso del software. De manera que una de las finalidades del proyecto involucra profundamente en el ámbito educativo ya que se quiere impulsar nuevas ideas para que los estudiantes de la carrera se sientan atraídos en el desarrollo de aplicaciones móviles y logren potencializar nuevos proyectos.

## 8 METODOLOGIA

Este es un proyecto de investigación aplicada basada en la ingeniería del software para dispositivos móviles que incluye el ciclo de vida clásico como es pre análisis, análisis, diseño, programación, pruebas, implementación. Inicialmente se propone trabajar con una metodología RUP ágil (agile modeling) que incluye las fases de inicio o comienzo, elaboración, construcción, transición. Dentro de la metodología RUP ágil están definidas actividades o tareas relacionadas con los siguientes procesos

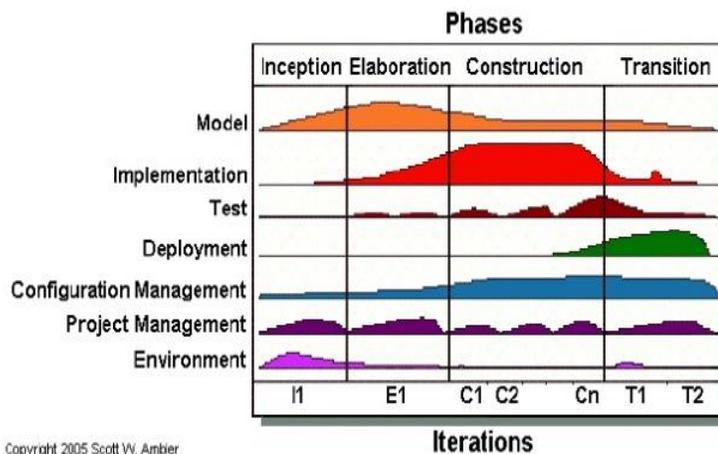


Figura 7: Metodología

### FASE 1

Identificar y definir los requisitos funcionales y no funcionales por medio de entrevistas y encuestas para determinar con certeza las funcionalidades del aplicativo.

Actividades: Diseñar entrevistas, encuestas, revisar documentación, aplicación de entrevistas, aplicación de encuestas, obtener datos espaciales, elaborar plan de desarrollo.

### FASE 2

Diseñar un modelo de datos y una arquitectura de software acorde a los requisitos para facilitar la reutilización.

Actividades:

Utilizar un editor de java y lenguajes de programación orientado a objetos y programar la aplicación, basado en una versión 2.2 de Android o posterior.

### FASE 3

Implementar prototipos funcionales de forma iterativa y evolutiva para facilitar la comunicación con el cliente y los obtener los ajustes de los requisitos.

Actividades:

Finalizar APK funcional, diseñar y actualizar la forma de visualización de datos para direccionar los usuarios a las aulas, revisar y actualizar la documentación.

Entrega del producto final

### FASE 4

Diseñar un plan de pruebas de software para asegurar la calidad de la aplicación.

Actividades:

Realizar Testing de la aplicación

## 9 CRONOGRAMA

CRONOGRAMA DE ACTIVIDADES																		
Título del trabajo:																		
MESES	MES 1				MES 2				MES 3				MES 4				MES 5	
SEMANAS	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	
ACTIVIDADES DEL PROYECTO																		
Presentación del anteproyecto	x																	
Aprobación del anteproyecto		x																
Firma acta de inicio y presentación al asesor del anteproyecto			x															
Inicio del código de la aplicación, y formulación de layouts				x														
Corrección de los layouts presentados				x	x													
Implementación del lector de QR en el aplicativo					x	x												
Testeo del lector de QR y correcciones							x											
Creación de base de datos con la información de los salones								x	x									
Creación de códigos QR con la información de las aulas y oficinas (Muestra)									x	x								
Testeo de la aplicación											x							
Entrega del aplicativo e informe del proyecto												x	x					
Sustentación del proyecto															x			

Tabla 4: cronograma

## 10 CASOS DE USO

### 10.1 Buscar aula

Es la primera actividad que realiza el usuario dentro del sistema. El usuario indica el salón al cual se quiere dirigir y obtiene como respuesta de sistema 2 opciones, la primera es que el salón no existe y la segunda en caso de que exista el salón retorna la información del bloque en el que se encuentra al aula solicitada.

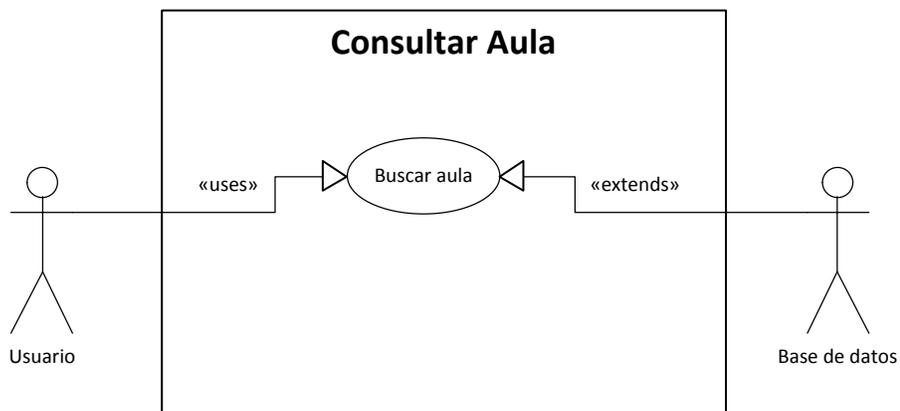


Figura 8: Caso de uso UML buscar aula

## 10.1.1 Documentación Caso de Uso Buscar Aula

Tabla 5: Tabla buscar aula

<b>Nombre Caso de Uso</b>	Buscar aula			
<b>Código Caso de Uso</b>	01			
<b>Descripción Caso de Uso</b>	El usuario solicita buscar un aula			
<b>Nombre Interacción</b>	Buscar aula			
<b>Descripción Interacción</b>	El usuario ingresa y puede realizar la consulta del aula que desee buscar			
<b>Actor(es)</b>	Cliente, sistema			
<b>Pre-condiciones</b>	N/A			
<b>Dependencias anteriores</b>	N/A			
<b>Camino principal</b>	<b>Acción actor</b>	<b>Reglas del negocio</b>	<b>Acción/respuesta sistema</b>	<b>Resultado de la acción</b>
	El usuario ingresa a la aplicación		Habilita la interfaz de búsqueda del aula	
	El usuario ingresa el aula que desea buscar		El sistema busca en su base de datos el aula que el usuario ingreso	
			El sistema retorna el bloque en el cual se encuentra el salón buscado	Mostrar bloque del salón solicitado
<b>Caminos Alternativos</b>	El usuario ingresa un aula que no se encuentra registrada en la base de datos		El sistema indica que el aula solicitada no existe	
<b>Dependencias posteriores</b>	Ninguna			
<b>Post-condiciones</b>	Información valida,			
<b>Autor</b>	Mateo Gallo Martínez			
<b>Fecha modificaciones</b>	Ninguna			
<b>Observaciones</b>	Ninguna			

## 10.1.2 Interfaces



Figura 9:Pantalla1



Figura 10: Pantalla2

## 10.2 Escanear código QR

Una vez el usuario indico el salón y el sistema retornó la información del bloque en el cual se encuentra el aula y se confirma que el aula está registrada en la base de datos, el usuario procede a iniciar el escaneo del código QR que contiene la información que el usuario solicita para dirigirse a la ruta deseada.

Como precondiciones en el móvil debe estar instalado un lector de códigos que será invocado por la aplicación, en caso tal que la aplicación no detecte preinstalado el lector se le indicara al usuario que si desea instalarlo y lo direccionara directamente al sitio de descargas de Android.

Una vez se tiene el lector, el usuario procede a escanear el código QR el cual le indicara como llegar a la ruta deseada.

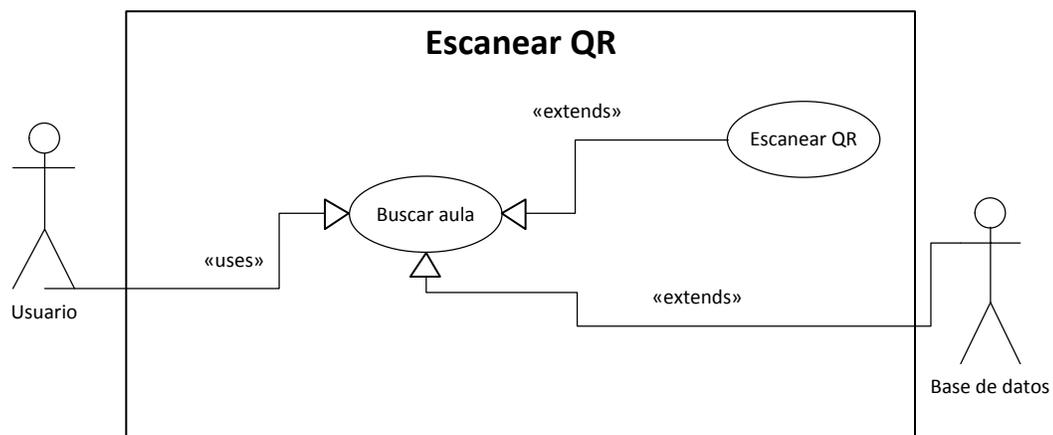


Figura 11: Caso de uso UML escanear QR

## 10.2.1 Documentación Caso de Uso Escanear código QR

Tabla 6: Tabla escanear QR

<b>Nombre Caso de Uso</b>	Escanear código QR			
<b>Código Caso de Uso</b>	02			
<b>Descripción Caso de Uso</b>	El usuario escanea código QR			
<b>Nombre Interacción</b>	Escanear QR			
<b>Descripción Interacción</b>	El usuario lee el código QR con la información del aula que desee buscar			
<b>Actor(es)</b>	Cliente, sistema			
<b>Pre-condiciones</b>	Lector de códigos QR instalado en el movil			
<b>Dependencias anteriores</b>	Ninguna			
<b>Camino principal</b>	<b>Acción actor</b>	<b>Reglas del negocio</b>	<b>Acción/respuesta sistema</b>	<b>Resultado de la acción</b>
	El usuario selecciona la opción Iniciar Escaneo		El sistema llama el lector de códigos QR	
	El usuario da clic en la imagen para iniciar el lector		El sistema muestra en pantalla la interfaz del lector	
	El usuario escanea el código del salón al cual desea dirigirse		El sistema muestra la información contenida en el código QR	Retorna URL almacenada en el QR
	El usuario abre por medio del explorador de su celular la URL que trae la aplicación luego del escaneo		El sistema muestra la ruta con la señalización para dirigirse al lugar deseado	Se muestra la ruta al salón indicado
<b>Caminos Alternativos</b>	El usuario selecciona la opción de iniciar el escaneo		El sistema trae la pantalla para iniciar el escaneo	
	El usuario da clic en la imagen para iniciar el lector		La aplicación no encuentra el lector instalado en el celular y le indica al usuario si desea instalar el Scanner para continuar.	

<b>Nombre Caso de Uso</b>	Escanear código QR			
	El usuario indica que si desea instalar el lector		El sistema direcciona a la tienda de descargas de Android	
	El usuario descarga la aplicación de la tienda de Android			Se instala en el celular el lector de códigos QR
			El sistema puede proceder con el escaneo de códigos	
<b>Dependencias posteriores</b>	Ninguna			
<b>Post-condiciones</b>	Ninguna			
<b>Autor</b>	Mateo Gallo Martínez			
<b>Fecha modificaciones</b>	Ninguna			
<b>Observaciones</b>	Ninguna			

## 10.2.2 Interfaces



Figura 12:Pantalla3

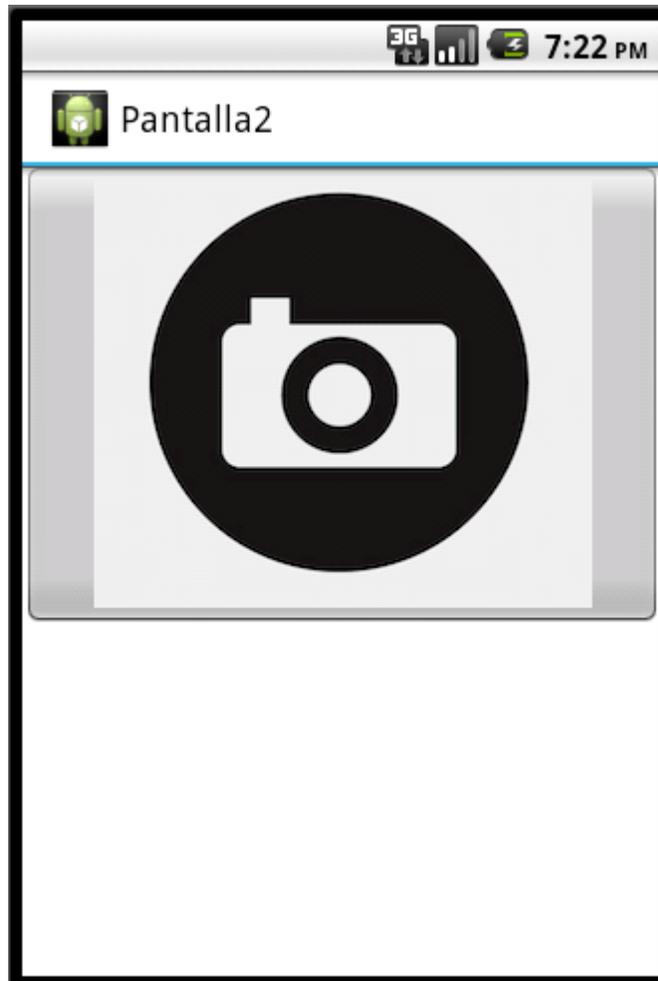


Figura 13:Pantalla4



Figura 14: Pantalla5

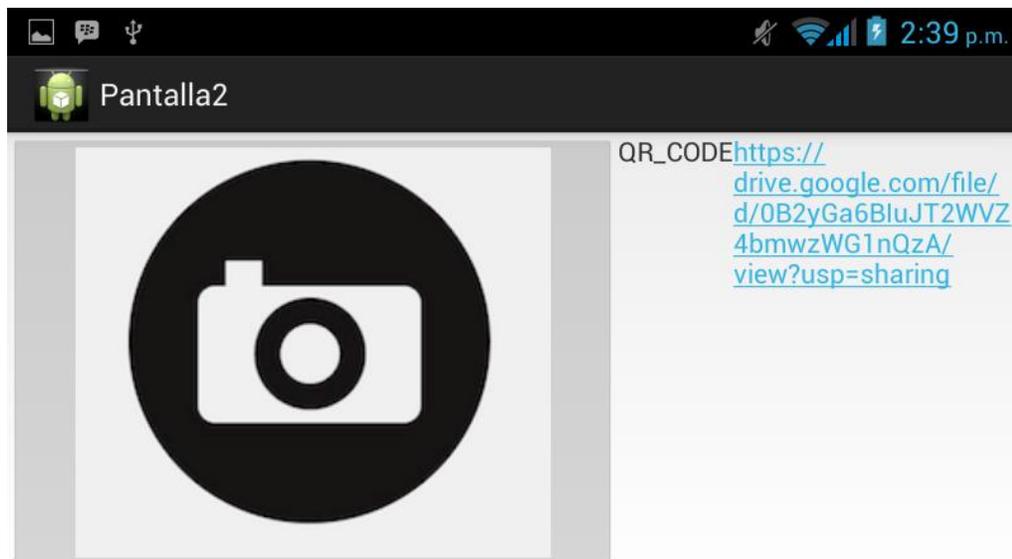


Figura 15:Pantalla6

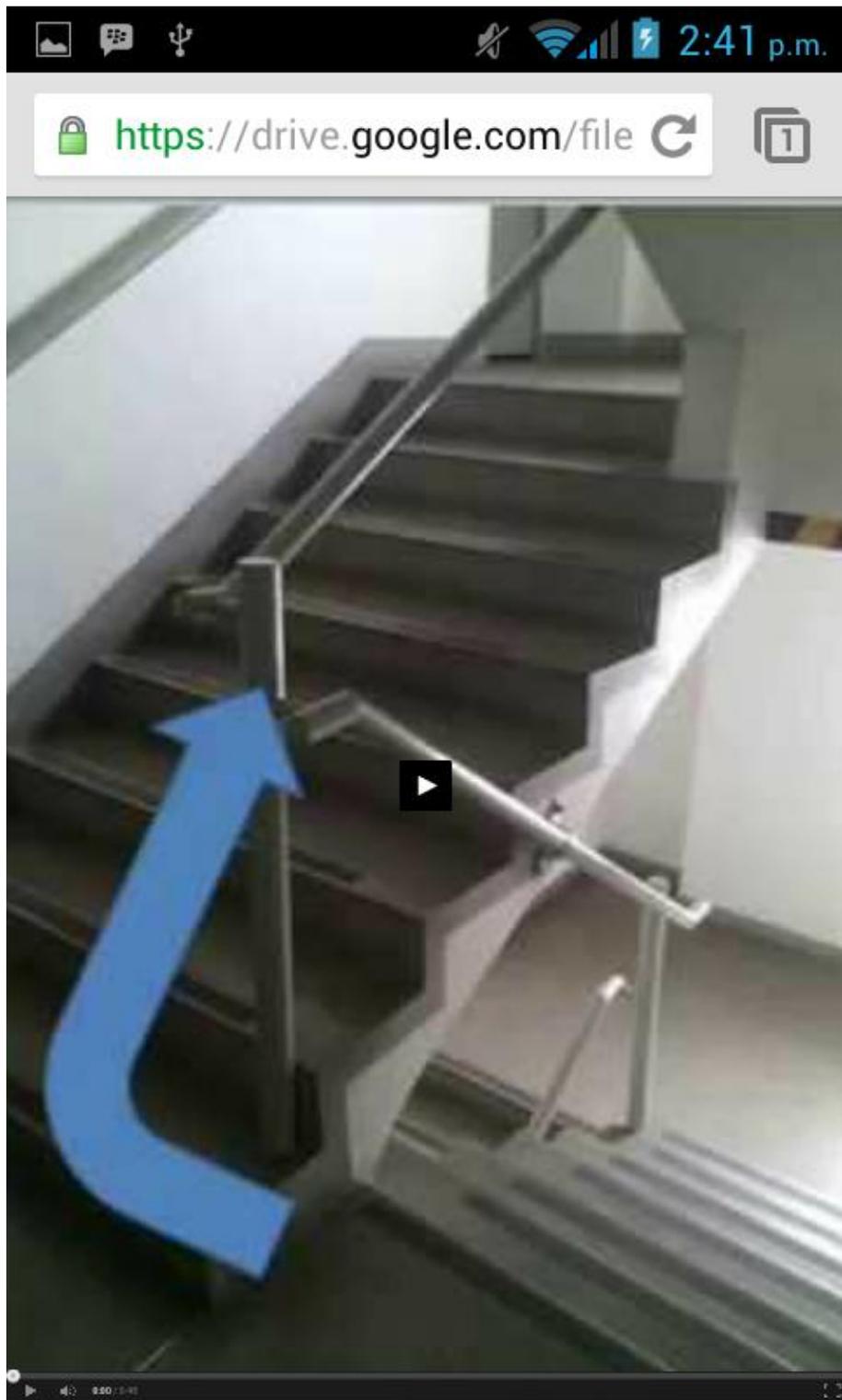


Figura 16: Pantalla7

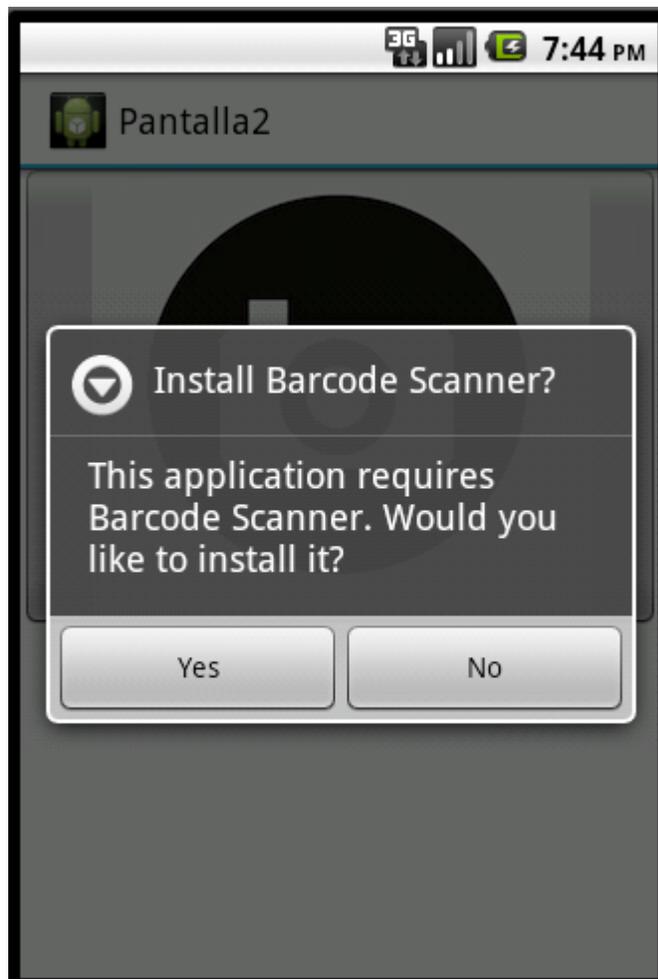
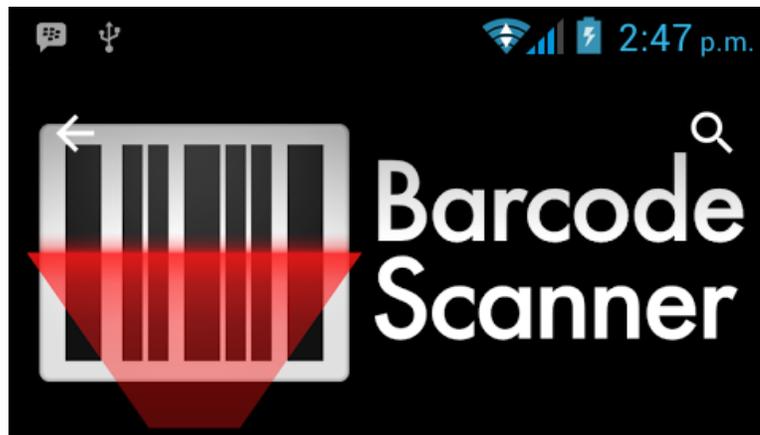


Figura 17: Pantalla8



**Barcode Scanner**  
ZXing Team

INSTALAR



Descargas



648.246 



Compras



Similar

Los códigos de barras de  
exploración sobre los productos, los  
códigos QR, y más

Figura 18: Pantalla9

## 11 CODIFICACIÓN

### 11.1 ACTIVITY PANTALLA1.XML

Pantalla principal de la aplicación.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.prototipo1.Pantalla1" >
```

<ImageView

```
    android:id="@+id/imageView1"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:src="@drawable/usb" />
```

<ImageView

```
    android:id="@+id/imageView2"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_alignLeft="@+id/imageView1"
    android:layout_alignRight="@+id/imageView1"
    android:layout_below="@+id/imageView1"
    android:src="@drawable/imagen3" />
```

<Button

```
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/salon"
    android:layout_centerHorizontal="true"
    android:text="@string/button1" />
```

<TextView

```
    android:id="@+id/salon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="17dp"
    android:text="@string/bloque" />
```

<EditText

```
    android:id="@+id/editText1"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/Button2"
        android:layout_centerHorizontal="true"
        android:ems="10" />

<Button
    android:id="@+id/Button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:onClick="consulta"
    android:src="@drawable/abc_ic_search"
    android:text="@string/button2" />

</RelativeLayout>

```

## 11.2 PANTALLA1.JAVA

```

public class Pantalla1 extends ActionBarActivity {

    EditText et1;
    Cursor cursor;
    TextView texto;
    SimpleCursorAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pantalla1);
        Button orderButton=(Button)findViewById(R.id.button1);
        Button Button=(Button)findViewById(R.id.Button2);
        et1 = (EditText)findViewById(R.id.editText1);
        texto=(TextView)findViewById(R.id.salon);

        //--- BOTON QUE REDIRECCIONA A OTRA ACTIVIDAD---//
        orderButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Intent intent = new Intent(FirstActivity.this,
OrderScreen.class);
                Intent button_uno = new Intent (Pantalla1.this,
Pantalla2.class);
                startActivity(button_uno);
            }
        });
        //---END BOTON---//
    }
}

```

```

public void consulta(View v) {
    DataBaseManager admin = new DataBaseManager(this,
        "salon", null, 1);
    SQLiteDatabase bd = admin.getWritableDatabase();
    String salon = et1.getText().toString();
    Cursor c = bd.rawQuery(
        "select bloque from salon where nombre= '" + salon
        + "'", null);

    if (c.moveToFirst()) {
        texto.setText(c.getString(0));
    } else
        Toast.makeText(this, "No existe salón",
            Toast.LENGTH_SHORT).show();

    bd.close();
}
}

```

### 11.3 ACTIVITY PANTALLA2.XML

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="321dp"
        android:layout_height="wrap_content"
        android:src="@drawable/camara" />

    <TextView
        android:id="@+id/tvFormat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

    </TextView>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

    </TextView>

</LinearLayout>

```

## 11.4 PANTALLA2.JAVA

Pantalla para llamar lector de QR

```
public class Pantalla2 extends ActionBarActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pantalla2);
        configureButtonReader();
    }

    private void configureButtonReader() {
        final ImageButton buttonReader =
        (ImageButton)findViewById(R.id.imageButton1);
        buttonReader.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                new IntentIntegrator(Pantalla2.this).initiateScan();
            }
        });
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent intent)
    {
        final IntentResult scanResult =
        IntentIntegrator.parseActivityResult(requestCode, resultCode, intent);
        handleResult(scanResult);
    }

    private void handleResult(IntentResult scanResult) {
        if (scanResult != null) {
            updateUITextViews(scanResult.getContents(),
            scanResult.getFormatName());
        } else {
            Toast.makeText(this, "No se ha leído nada :(",
            Toast.LENGTH_SHORT).show();
        }
    }

    private void updateUITextViews(String scan_result, String
    scan_result_format) {
        ((TextView)findViewById(R.id.tvFormat)).setText(scan_result_format);
        final TextView tvResult = (TextView)findViewById(R.id.tvResult);
        tvResult.setText(scan_result);
        Linkify.addLinks(tvResult, Linkify.ALL);
    }
}
```

## 11.5 ADMINSQLiteOPENHELPER.JAVA

Clase encargada de la administración de la base de datos

```
public class DataBaseManager extends SQLiteOpenHelper {

    public DataBaseManager(Context context, String nombre, CursorFactory
factory, int version) {
        super(context, nombre, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table salon(id integer primary key
autoincrement, nombre text, bloque text)");

        ContentValues valores = new ContentValues();

        valores.put("nombre", "409-c");
        valores.put("bloque", "Bloque c");
        db.insert("salon", null, valores);
        valores.put("nombre", "miniauditorio1");
        valores.put("bloque", "Bloque c");
        db.insert("salon", null, valores);
        valores.put("nombre", "miniauditorio2");
        valores.put("bloque", "Bloque c");
        db.insert("salon", null, valores);
        valores.put("nombre", "rectoria");
        valores.put("bloque", "Bloque e");
        db.insert("salon", null, valores);
        valores.put("nombre", "decanatura");
        valores.put("bloque", "Bloque e");
        db.insert("salon", null, valores);
        valores.put("nombre", "auditorio principal");
        valores.put("bloque", "Bloque c");
        db.insert("salon", null, valores);
        valores.put("nombre", "auditorio fray arturo calle restrepo");
        valores.put("bloque", "Bloque c");
        db.insert("salon", null, valores);

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int versionAnte, int versionNue)
{
        db.execSQL("drop table if exists salon");
        db.execSQL("create table salon(id integer primary key
autoincrement, nombre text, bloque text)");
    }
}
```

## 11.6 INTENTRESULT.JAVA

Clase para invocar lector de códigos QR

```
public final class IntentResult {

    private final String contents;
    private final String formatName;
    private final byte[] rawBytes;
    private final Integer orientation;
    private final String errorCorrectionLevel;

    IntentResult() {
        this(null, null, null, null, null);
    }

    IntentResult(String contents,
                String formatName,
                byte[] rawBytes,
                Integer orientation,
                String errorCorrectionLevel) {
        this.contents = contents;
        this.formatName = formatName;
        this.rawBytes = rawBytes;
        this.orientation = orientation;
        this.errorCorrectionLevel = errorCorrectionLevel;
    }

    /**
     * @return raw content of barcode
     */
    public String getContents() {
        return contents;
    }

    /**
     * @return name of format, like "QR_CODE", "UPC_A". See {@code BarcodeFormat}
     for more format names.
     */
    public String getFormatName() {
        return formatName;
    }

    /**
     * @return raw bytes of the barcode content, if applicable, or null otherwise
     */
    public byte[] getRawBytes() {
        return rawBytes;
    }

    /**
     * @return rotation of the image, in degrees, which resulted in a successful
     scan. May be null.
     */
}
```

```

    */
    public Integer getOrientation() {
        return orientation;
    }

    /**
     * @return name of the error correction level used in the barcode, if
    applicable
    */
    public String getErrorCorrectionLevel() {
        return errorCorrectionLevel;
    }

    @Override
    public String toString() {
        StringBuilder dialogText = new StringBuilder(100);
        dialogText.append("Format: ").append(formatName).append('\n');
        dialogText.append("Contents: ").append(contents).append('\n');
        int rawBytesLength = rawBytes == null ? 0 : rawBytes.length;
        dialogText.append("Raw bytes: (").append(rawBytesLength).append("
bytes)\n");
        dialogText.append("Orientation: ").append(orientation).append('\n');
        dialogText.append("EC level: ").append(errorCorrectionLevel).append('\n');
        return dialogText.toString();
    }
}

```

## 11.7 INTENTINTEGRATOR.JAVA

Clase para invocar lector de códigos QR

```

public class IntentIntegrator {

    public static final int REQUEST_CODE = 0x0000c0de; // Only use bottom 16 bits
    private static final String TAG = IntentIntegrator.class.getSimpleName();

    public static final String DEFAULT_TITLE = "Install Barcode Scanner?";
    public static final String DEFAULT_MESSAGE =
        "This application requires Barcode Scanner. Would you like to install
it?";
    public static final String DEFAULT_YES = "Yes";
    public static final String DEFAULT_NO = "No";

    private static final String BS_PACKAGE = "com.google.zxing.client.android";
    private static final String BSPLUS_PACKAGE = "com.srowen.bs.android";

    // supported barcode formats
    public static final Collection<String> PRODUCT_CODE_TYPES = List("UPC_A",
"UPC_E", "EAN_8", "EAN_13", "RSS_14");
    public static final Collection<String> ONE_D_CODE_TYPES =
        List("UPC_A", "UPC_E", "EAN_8", "EAN_13", "CODE_39", "CODE_93",
"CODE_128",

```

```

        "ITF", "RSS_14", "RSS_EXPANDED");
    public static final Collection<String> QR_CODE_TYPES =
Collections.singleton("QR_CODE");
    public static final Collection<String> DATA_MATRIX_TYPES =
Collections.singleton("DATA_MATRIX");

    public static final Collection<String> ALL_CODE_TYPES = null;

    public static final List<String> TARGET_BARCODE_SCANNER_ONLY =
Collections.singletonList(BS_PACKAGE);
    public static final List<String> TARGET_ALL_KNOWN = List(
        BSPLUS_PACKAGE, // Barcode Scanner+
        BSPLUS_PACKAGE + ".simple", // Barcode Scanner+ Simple
        BS_PACKAGE // Barcode Scanner
        // What else supports this intent?
    );

    private final Activity activity;
    private final Fragment fragment;

    private String title;
    private String message;
    private String buttonYes;
    private String buttonNo;
    private List<String> targetApplications;
    private final Map<String, Object> moreExtras = new HashMap<String, Object>(3);

    /**
     * @param activity {@link Activity} invoking the integration
     */
    public IntentIntegrator(Activity activity) {
        this.activity = activity;
        this.fragment = null;
        initializeConfiguration();
    }

    /**
     * @param fragment {@link Fragment} invoking the integration.
     * {@link #startActivityForResult(Intent, int)} will be called on the {@link
    Fragment} instead
     * of an {@link Activity}
     */
    public IntentIntegrator(Fragment fragment) {
        this.activity = fragment.getActivity();
        this.fragment = fragment;
        initializeConfiguration();
    }

    private void initializeConfiguration() {
        title = DEFAULT_TITLE;
        message = DEFAULT_MESSAGE;
        buttonYes = DEFAULT_YES;
        buttonNo = DEFAULT_NO;
    }

```

```

    targetApplications = TARGET_ALL_KNOWN;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public void setTitleByID(int titleID) {
    title = activity.getString(titleID);
}

public String getMessage() {
    return message;
}

public void setMessage(String message) {
    this.message = message;
}

public void setMessageByID(int messageID) {
    message = activity.getString(messageID);
}

public String getButtonYes() {
    return buttonYes;
}

public void setButtonYes(String buttonYes) {
    this.buttonYes = buttonYes;
}

public void setButtonYesByID(int buttonYesID) {
    buttonYes = activity.getString(buttonYesID);
}

public String getButtonNo() {
    return buttonNo;
}

public void setButtonNo(String buttonNo) {
    this.buttonNo = buttonNo;
}

public void setButtonNoByID(int buttonNoID) {
    buttonNo = activity.getString(buttonNoID);
}

public Collection<String> getTargetApplications() {
    return targetApplications;
}

```

```

}

public final void setTargetApplications(List<String> targetApplications) {
    if (targetApplications.isEmpty()) {
        throw new IllegalArgumentException("No target applications");
    }
    this.targetApplications = targetApplications;
}

public void setSingleTargetApplication(String targetApplication) {
    this.targetApplications = Collections.singletonList(targetApplication);
}

public Map<String,?> getMoreExtras() {
    return moreExtras;
}

public final void addExtra(String key, Object value) {
    moreExtras.put(key, value);
}

/**
 * Initiates a scan for all known barcode types with the default camera.
 *
 * @return the {@link AlertDialog} that was shown to the user prompting them
to download the app
 * if a prompt was needed, or null otherwise.
 */
public final AlertDialog initiateScan() {
    return initiateScan(ALL_CODE_TYPES, -1);
}

/**
 * Initiates a scan for all known barcode types with the specified camera.
 *
 * @param cameraId camera ID of the camera to use. A negative value means "no
preference".
 * @return the {@link AlertDialog} that was shown to the user prompting them
to download the app
 * if a prompt was needed, or null otherwise.
 */
public final AlertDialog initiateScan(int cameraId) {
    return initiateScan(ALL_CODE_TYPES, cameraId);
}

/**
 * Initiates a scan, using the default camera, only for a certain set of
barcode types, given as strings corresponding
 * to their names in ZXing's {@code BarcodeFormat} class like "UPC_A". You can
supply constants
 * like {@link #PRODUCT_CODE_TYPES} for example.
 *
 * @param desiredBarcodeFormats names of {@code BarcodeFormat}s to scan for

```

```

    * @return the {@link AlertDialog} that was shown to the user prompting them
    to download the app
    * if a prompt was needed, or null otherwise.
    */
    public final AlertDialog initiateScan(Collection<String>
desiredBarcodeFormats) {
        return initiateScan(desiredBarcodeFormats, -1);
    }

    /**
     * Initiates a scan, using the specified camera, only for a certain set of
     barcode types, given as strings corresponding
     * to their names in ZXing's {@code BarcodeFormat} class like "UPC_A". You can
     supply constants
     * like {@link #PRODUCT_CODE_TYPES} for example.
     *
     * @param desiredBarcodeFormats names of {@code BarcodeFormat}s to scan for
     * @param cameraId camera ID of the camera to use. A negative value means "no
     preference".
     * @return the {@link AlertDialog} that was shown to the user prompting them
     to download the app
     * if a prompt was needed, or null otherwise
     */
    public final AlertDialog initiateScan(Collection<String>
desiredBarcodeFormats, int cameraId) {
        Intent intentScan = new Intent(BS_PACKAGE + ".SCAN");
        intentScan.addCategory(Intent.CATEGORY_DEFAULT);

        // check which types of codes to scan for
        if (desiredBarcodeFormats != null) {
            // set the desired barcode types
            StringBuilder joinedByComma = new StringBuilder();
            for (String format : desiredBarcodeFormats) {
                if (joinedByComma.length() > 0) {
                    joinedByComma.append(',');
                }
                joinedByComma.append(format);
            }
            intentScan.putExtra("SCAN_FORMATS", joinedByComma.toString());
        }

        // check requested camera ID
        if (cameraId >= 0) {
            intentScan.putExtra("SCAN_CAMERA_ID", cameraId);
        }

        String targetAppPackage = findTargetAppPackage(intentScan);
        if (targetAppPackage == null) {
            return showDownloadDialog();
        }
        intentScan.setPackage(targetAppPackage);
        intentScan.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intentScan.addFlags(Intent.FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET);
    }

```

```

        attachMoreExtras(intentScan);
        startActivityForResult(intentScan, REQUEST_CODE);
        return null;
    }

    /**
     * Start an activity. This method is defined to allow different methods of
    activity starting for
     * newer versions of Android and for compatibility library.
     *
     * @param intent Intent to start.
     * @param code Request code for the activity
     * @see android.app.Activity#startActivityForResult(Intent, int)
     * @see android.app.Fragment#startActivityForResult(Intent, int)
     */
    protected void startActivityForResult(Intent intent, int code) {
        if (fragment == null) {
            activity.startActivityForResult(intent, code);
        } else {
            fragment.startActivityForResult(intent, code);
        }
    }

    private String findTargetAppPackage(Intent intent) {
        PackageManager pm = activity.getPackageManager();
        List<ResolveInfo> availableApps = pm.queryIntentActivities(intent,
        PackageManager.MATCH_DEFAULT_ONLY);
        if (availableApps != null) {
            for (String targetApp : targetApplications) {
                if (contains(availableApps, targetApp)) {
                    return targetApp;
                }
            }
        }
        return null;
    }

    private static boolean contains(Iterable<ResolveInfo> availableApps, String
    targetApp) {
        for (ResolveInfo availableApp : availableApps) {
            String packageName = availableApp.activityInfo.packageName;
            if (targetApp.equals(packageName)) {
                return true;
            }
        }
        return false;
    }

    private AlertDialog showDownloadDialog() {
        AlertDialog.Builder downloadDialog = new AlertDialog.Builder(activity);
        downloadDialog.setTitle(title);
        downloadDialog.setMessage(message);
    }

```

```

        downloadDialog.setPositiveButton(buttonYes, new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        String packageName;
        if (targetApplications.contains(BS_PACKAGE)) {
            // Prefer to suggest download of BS if it's anywhere in the list
            packageName = BS_PACKAGE;
        } else {
            // Otherwise, first option:
            packageName = targetApplications.get(0);
        }
        Uri uri = Uri.parse("market://details?id=" + packageName);
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        try {
            if (fragment == null) {
                activity.startActivity(intent);
            } else {
                fragment.startActivity(intent);
            }
        } catch (ActivityNotFoundException anfe) {
            // Hmm, market is not installed
            Log.w(TAG, "Google Play is not installed; cannot install " +
packageName);
        }
    }
});
downloadDialog.setNegativeButton(buttonNo, null);
downloadDialog.setCancelable(true);
return downloadDialog.show();
}

/**
 * <p>Call this from your {@link Activity}'s
 * {@link Activity#onActivityResult(int, int, Intent)} method.</p>
 *
 * @param requestCode request code from {@code onActivityResult()}
 * @param resultCode result code from {@code onActivityResult()}
 * @param intent {@link Intent} from {@code onActivityResult()}
 * @return null if the event handled here was not related to this class, or
 * else an {@link IntentResult} containing the result of the scan. If the
user cancelled scanning,
 * the fields will be null.
 */
public static IntentResult parseActivityResult(int requestCode, int
resultCode, Intent intent) {
    if (requestCode == REQUEST_CODE) {
        if (resultCode == Activity.RESULT_OK) {
            String contents = intent.getStringExtra("SCAN_RESULT");
            String formatName = intent.getStringExtra("SCAN_RESULT_FORMAT");
            byte[] rawBytes = intent.getByteArrayExtra("SCAN_RESULT_BYTES");

```

```

        int intentOrientation = intent.getIntExtra("SCAN_RESULT_ORIENTATION",
Integer.MIN_VALUE);
        Integer orientation = intentOrientation == Integer.MIN_VALUE ? null :
intentOrientation;
        String errorCorrectionLevel =
intent.getStringExtra("SCAN_RESULT_ERROR_CORRECTION_LEVEL");
        return new IntentResult(contents,
                                formatName,
                                rawBytes,
                                orientation,
                                errorCorrectionLevel);
    }
    return new IntentResult();
}
return null;
}

/**
 * Defaults to type "TEXT_TYPE".
 *
 * @param text the text string to encode as a barcode
 * @return the {@link AlertDialog} that was shown to the user prompting them
to download the app
 * if a prompt was needed, or null otherwise
 * @see #shareText(CharSequence, CharSequence)
 */
public final AlertDialog shareText(CharSequence text) {
    return shareText(text, "TEXT_TYPE");
}

/**
 * Shares the given text by encoding it as a barcode, such that another user
can
 * scan the text off the screen of the device.
 *
 * @param text the text string to encode as a barcode
 * @param type type of data to encode. See {@code
com.google.zxing.client.android.Contents.Type} constants.
 * @return the {@link AlertDialog} that was shown to the user prompting them
to download the app
 * if a prompt was needed, or null otherwise
 */
public final AlertDialog shareText(CharSequence text, CharSequence type) {
    Intent intent = new Intent();
    intent.addCategory(Intent.CATEGORY_DEFAULT);
    intent.setAction(BS_PACKAGE + ".ENCODE");
    intent.putExtra("ENCODE_TYPE", type);
    intent.putExtra("ENCODE_DATA", text);
    String targetAppPackage = findTargetAppPackage(intent);
    if (targetAppPackage == null) {
        return showDownloadDialog();
    }
}

```

```

intent.setPackage(targetAppPackage);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET);
attachMoreExtras(intent);
if (fragment == null) {
    activity.startActivity(intent);
} else {
    fragment.startActivity(intent);
}
return null;
}

private static List<String> list(String... values) {
    return Collections.unmodifiableList(Arrays.asList(values));
}

private void attachMoreExtras(Intent intent) {
    for (Map.Entry<String, Object> entry : moreExtras.entrySet()) {
        String key = entry.getKey();
        Object value = entry.getValue();
        // Kind of hacky
        if (value instanceof Integer) {
            intent.putExtra(key, (Integer) value);
        } else if (value instanceof Long) {
            intent.putExtra(key, (Long) value);
        } else if (value instanceof Boolean) {
            intent.putExtra(key, (Boolean) value);
        } else if (value instanceof Double) {
            intent.putExtra(key, (Double) value);
        } else if (value instanceof Float) {
            intent.putExtra(key, (Float) value);
        } else if (value instanceof Bundle) {
            intent.putExtra(key, (Bundle) value);
        } else {
            intent.putExtra(key, value.toString());
        }
    }
}
}

}

[7]

```

## 12 RESULTADOS DE CASOS DE PRUEBA

### 12.1 Buscar Salón

El usuario debe ingresar el salón siempre en letra minúscula



Figura 19:Prueba1



Figura 20:Prueba2

## 12.2 Buscar salón

No se debe dejar un espacio luego de finalizar de copiar el salón



Figura 21:Prueba3

### 13 FUTUROS RESULTADOS

Se entrega el proyecto y se espera que se continúe el progreso del mismo por medio del semillero en ingeniería de software, el paso a seguir sería implementar los códigos QR con realidad aumentada para una visualización de 3D de un modelo en google Sketchup (es un programa de diseño gráfico y modelado en (3D) tres dimensiones basado en caras. Para entornos de arquitectura, ingeniería civil, diseño industrial, diseño escénico, GIS, videojuegos o películas. Es un programa desarrollado por @Last Software, empresa adquirida por Google en 2006 y finalmente vendida a Trimble en 2012.[8]) para lo cual se necesitaría un trabajo conjunto entre la facultad de ingeniería y la facultad de artes integradas para el diseño del modelo de la universidad.

De igual manera se podría implementar el proyecto en la sede de Salento y ampliar este servicio a todas las sedes de la USB.

De igual manera se puede realizar modificaciones de código para al momento de realizar la consunta se pueda traer cualquier información que se desee.

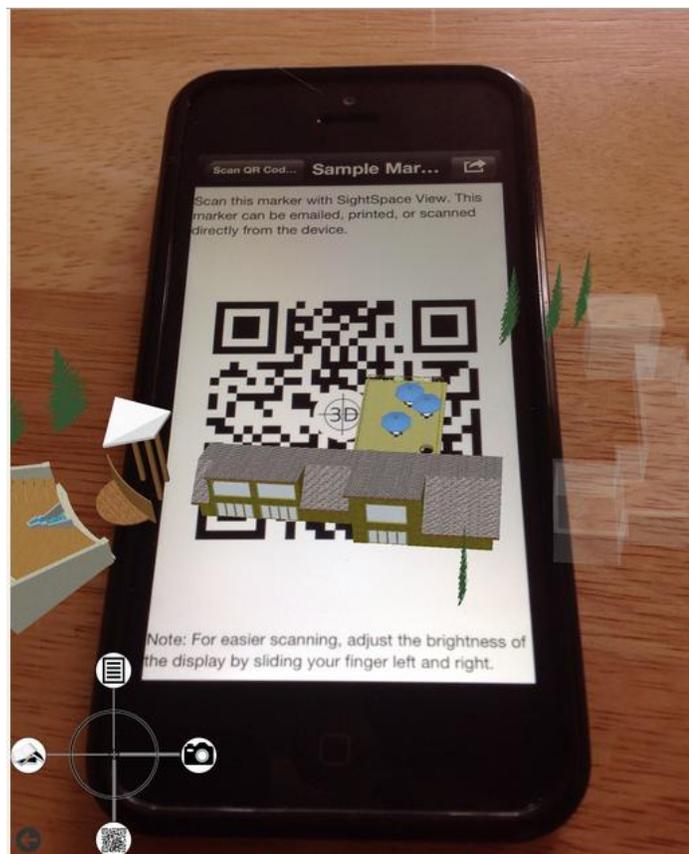


Figura 22: QR [9]

## 14 CONCLUSIONES

Con este trabajo pretendemos promover y continuar con el énfasis que le queremos dar a nuestro semillero de ingeniería del software contribuyendo con el desarrollo de aplicaciones móviles y motivando a nuevos estudiantes a este nuevo reto tecnológico.

El alcance del proyecto llega hasta la fase de creación de prototipo funcional, y queda el proyecto como parte del semillero de ingeniería de Software para futuros trabajos de grado de estudiantes que deseen continuar de forma activa con el proyecto

Se debe continuar con la enseñanza y las capacitaciones a los estudiantes del semillero con el fin de contribuir al trabajo y seguir el proceso comenzado, y se debe realizar desde el semillero ya que dentro del pensum visto ninguna materia fue enfocada a la parte de dispositivos móviles, situación que debiera cambiar ya que este tema es un tema actual y con gran proyección a futuro.

### 14.1 Anotaciones especiales.

La información a la cual direcciona los códigos QR está contenida en una cuenta de GOOGLE DRIVE creada para seguir almacenando la información de los salones de la universidad.

Cuenta: [prototipo.usbmed@gmail.com](mailto:prototipo.usbmed@gmail.com)

Pass: usb06medellin

## 15 REFERENCIAS BIBLIOGRÁFICAS

- [1] É. Payet and F. Spoto, "Static analysis of Android programs," *Inf. Softw. Technol.*, vol. 54, no. 11, pp. 1192–1201, Nov. 2012.
- [2] "Sqlite." [Online]. Available: <http://www.sqlite.org/about.html>.
- [3] A. Shatte, J. Holdsworth, and I. Lee, "Mobile augmented reality based context-aware library management system," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2174–2185, Apr. 2014.
- [4] "Wikipedia." [Online]. Available: [http://es.wikipedia.org/wiki/C%C3%B3digo\\_QR](http://es.wikipedia.org/wiki/C%C3%B3digo_QR).
- [5] T.-L. Chou and L.-J. ChanLin, "Augmented Reality Smartphone Environment Orientation Application: A Case Study of the Fu-Jen University Mobile Campus Touring System," *Procedia - Soc. Behav. Sci.*, vol. 46, pp. 410–416, Jan. 2012.
- [6] Á. eugenio Ardila Pérez, "Prototipo funcional para el uso de realidad aumentada en la EIA usando dispositivos móviles," Escuela de ingeniería de Antioquia, 2012.
- [7] F. J. Arroyo, "Integrar Barcode Scanner en nuestra aplicación Android." [Online]. Available: [http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=qr\\_reader\\_android](http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=qr_reader_android).
- [8] "SketchUp." [Online]. Available: <http://es.wikipedia.org/wiki/SketchUp>.
- [9] "No Title." [Online]. Available: <http://s3.amazonaws.com/lci-static/View - AR.PNG>.

## 16 ANEXOS

### 16.1 ANEXO#1 (encuesta Nro1)

Encuesta ubicación de aulas universidad de san buenaventura sede san Benito

Marque con una X su respuesta

¿Alguna vez ha llegado tarde a una clase por tener dificultades al encontrar el aula?

Si \_\_\_\_\_

No \_\_\_\_\_

¿Cuándo ha llegado tarde cual es el tiempo que se demora para hallar el salón?

0 minutos \_\_\_\_\_

Entre 1 y 5 minutos \_\_\_\_\_

Entre 6 y 10 minutos \_\_\_\_\_

Entre 11 y 15 minutos \_\_\_\_\_

16 Minutos o más \_\_\_\_\_



UNIVERSIDAD DE  
SAN BUENAVENTURA

Figura 23: Anexo1

### 16.2 ANEXO#2(encuesta Nro2)

Encuesta ubicación de aulas universidad de san buenaventura sede san Benito

Marque con una X su respuesta.

¿Conoce en que lugar de la universidad fue tomada la siguiente fotografia de la oficina 409 C?



Si: \_\_\_\_\_

No: \_\_\_\_\_

¿Conoce en que lugar de la universidad se encuentra la sala de computo # 11?



Si: \_\_\_\_\_

No: \_\_\_\_\_



UNIVERSIDAD DE  
SAN BUENAVENTURA

Figura 24: Anexo2

**16.3 ANEXO#3(Código QR Salón 409-c)**



Figura 25: QR1

**16.4 ANEXO#4(Código QR Auditorio Principal)**



Figura 26:QR2