



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo Terminal

“Sistema Interactivo de Guía para Museos en Android”

2015-A036

Presentan

Marcos Ricardo García García

Emmanuel Montiel Martínez

José Eduardo Zamorano Aparicio

Directores:

M. en C. David Araujo Díaz

M. en C. Laura Méndez Segundo



ESCOM

Junio de 2016.



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA



No. de TT: 2015-A036

Junio de 2016

Documento Técnico

“Sistema Interactivo de Guía para Museos en Android”

Presentan

Marcos Ricardo García García¹

Emmanuel Montiel Martínez²

José Eduardo Zamorano Aparicio³

Directores

M. en C. David Araujo Díaz

M. en C. Laura Méndez Segundo

RESUMEN

En este trabajo terminal se implementó un sistema de intercambio de información para la orientación y asistencia de los usuarios en un museo. El desarrollo se compone de una aplicación móvil para dispositivos con la plataforma Android y soporte para la tecnología iBeacon además de un servicio web. Con la aplicación móvil, el usuario puede obtener información más detallada de las exposiciones que se encuentre visitando en el museo. También se cuenta con un sistema de administración de contenido para la información que el museo maneje.

Palabras clave – Aplicación móvil, iBeacon, Museos, Programación para dispositivos móviles.

¹ E-mail: mricardogg15@gmail.com

² E-mail: emmanuel_montt@hotmail.com

³ E-mail: jezamorano_0606@outlook.com



ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE FORMACIÓN INTEGRAL E
INSTITUCIONAL
COMISIÓN ACADÉMICA DE TRABAJO TERMINAL



México, D.F. a 9 de Junio de 2016

**DR. FLAVIO ARTURO SÁNCHEZ GARFIAS
PRESIDENTE DE LA COMISIÓN ACADÉMICA
DE TRABAJO TERMINAL
P R E S E N T E**

Por medio del presente, se informa que los alumnos que integran el TRABAJO TERMINAL: 2015-A036, titulado "Sistema Interactivo de Guía para Museos en Android" concluyeron satisfactoriamente su trabajo.

Los discos (DVDs) fueron revisados ampliamente por sus servidores y corregidos cubriendo el alcance y el objetivo planteados en el protocolo original y de acuerdo a los requisitos establecidos por la Comisión que Usted preside.

ATENTAMENTE

Araujo Diaz David

M. en C. David Araujo Díaz

Laura Méndez Segundo

M. en C. Laura Méndez Segundo

ADVERTENCIA

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.”

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000 Extensión 52000.

Agradecimientos

A mis padres:

Por estar siempre ahí a mi lado apoyándome en todos los aspectos, por brindarme siempre su apoyo en los momentos difíciles, darle un seguimiento a toda mi trayectoria académica, enseñarme a darle valor a las cosas y estar conmigo en estos 24 años de vida. Gracias a todos los valores y educación que me enseñaron para ser un persona de bien. Este logro también es de ustedes, ya que sin apoyo no se hubiera logrado, en verdad le agradezco a la vida por tenerlos y ser la guía en mi camino.

A mi familia:

A mi hermana Aideé por estar ahí, ella es mi motivación para seguir adelante, ya que no puedo dar un paso atrás, porque sé que ella me sigue y no le tengo fallar, a pesar de tener nuestras diferencias siempre hemos salido adelante, en verdad agradezco que estés aquí porque tú eres esa personita que hace que no caiga, ya que soy la guía para ti, te quiero mucho.

Mis abuelos que son un gran pilar para mí, ya que son mis segundos padres, gracias por estar ahí siempre y apoyarme en todo. A mi abuela Socorro que aunque no está aquí con nosotros ya, siempre estará cuidándome, este logro es para ti porque siempre me dijiste que querías estar ese día tan anhelado. Siempre tuve todo tu cariño y sé que te hubiera gustado estar aquí festejando con nosotros, pero por algo te fuiste y sé que serás mi guía para todas mis decisiones que tome en mi vida. A mi abuelo Modesto que es un gran padre, también este logro es tuyo porque sin tu apoyo y cariño no hubiera salido adelante, en verdad agradezco mucho por ser ese gran pilar inquebrantable en mi vida, por estar ahí desde que era niño y brindarme tus valores.

A mi tía Jazmín por esta conmigo desde que era un niño, que me aguantaba todas mis travesuras y me consentía, ahora me alegro mucho por nunca cambiar y seguir siendo así ahora con sus hijas, sé que será un gran guía para ellas ya que fue así conmigo y entre buenos resultados.

A mi tío Paco, que digo mi tío, mi hermano mayor que todo mundo desea, por estar ahí siempre, por ser mi ejemplo a seguir, por darme su gran cariño y estar conmigo en todo momento, te quiero muchísimo.

Por ultimo pero no el menos importante, mi primo hermano Iván que él ha estado ahí siempre conmigo, pasara el tiempo y aun seguiremos unidos, gracias por escucharme, apoyarme en mi vida y darme esos consejos duros pero verdaderos para que no caiga, por confiar en mí y ser el pilar mío y yo el tuyo.

A mis amigos:

Primeramente agradecerle a la vida por darme unos amigos maravillosos, porque sin ellos, esta vida sería gris y triste. A mi hermanazo Eduardo Zamorano por estar siempre ahí en las buenas y en las malas, en todo momento cuento con su apoyo y admirarlo por aguantarme estos 5 años en ESCOM, hemos pasado tantas cosas que nadie en la vida pensaría, y a pesar de todo ahí andamos con toda la actitud, en verdad muchas gracias por estar siempre ahí y eres mi ejemplo a seguir, después de tanto esfuerzo al fin podemos gritar ¡lo logramos!

A Emmanuel por ser una gran persona y un gran amigo, que desde primero nos conocimos y al día de hoy llegamos hasta el final. Gracias por estar siempre ahí apoyándonos y aportar grandes cosas a mi vida. A Vela por ser siempre ese amigo incondicional, que le dio chispa y color con todo su buen humor a ESCOM, sabes amiguito que se te quiere harto y gracias por estar ahí en todo momento, eres uno de mis hermanitos y siempre estaremos juntos en el camino. Omar Garduño mi hermano mayor no de sangre, que siempre estás ahí en cualquier momento, fuiste un gran apoyo en mi vida escolar y el mejor amigo que uno puede tener. Este logro también es tuyo hermano. Yarim mi hermanito también este logro es tuyo, desde la vocacional has estado ahí apoyándome y en verdad agradezco eso, por todos esos consejos y esas buenas palabras de señor has hecho que salga adelante, gracias por estar ahí siempre. A Fátima por ser mi apoyo desde la vocacional, se logró lo prometido, entramos juntos a ESCOM y salimos juntos, gracias por todo y por siempre estar ahí y sabes que siempre eres mi inspiración y a la persona que más admiro.

Verito gracias por ser esa amiga que siempre está ahí, con tus consejos, charlas, regaños pero sobretodo tu gran corazón y tu bondad, te quiero mucho. También porque gracias a ti conocimos a otra gran amiga que es Montse e hizo que nuestros días fueron mejores con su buena actitud y amistad. Fernando Vázquez le agradezco a la vida por conocerte ya que eres un amigo en el que se puede confiar y hacer tarugadas en todo momento, sabes que siempre andaremos para lo que sea. A mis amiguitos Erik, Rene, Víctor, Leonardo y Alejandro que se convirtieron en mi familia y que puedo contar con ellos siempre. A Deras que siempre me apoya en todo momento y en cualquier situación sin esperar nada a cambio, en verdad muchas gracias. A mis compañeritas y mejores amigas Miroslava y Mariana que siempre me escuchan, me dan consejos y que puedo contar con ellas siempre, las quiero muchísimo. A Jorge que siempre estaba ahí en los momentos difíciles y siempre lo solucionábamos con batallas. A Wendy que llego a alegrar con su buen humor mi vida a pesar de llevar corto tiempo nuestra amistad, siempre tan alegre, sincera, que me escucha todas mi tonterías, me brinda su apoyo y cariño, su gran compromiso de tener una vida saludable con sus sonsacadas de ir a correr y lo más importante por ser siempre tan *Picky*, jamás olvidare cuando nos conocimos, porque fue la actitud de como si nos conociéramos hace años. Por ultimo a Monse que por casualidades de la vida nos hablamos (obviamente por ser hartos ñoños) y a pesar de tener una corta pero gran amistad ha demostrado estar ahí cuando uno lo necesita, con su buena actitud y positivismo dio un giro muy bueno a nuestra vida, si incluyo demasiado a mi amiguito Zamo y nos sacó de nuestra monotonía en la escuela, gracias por brindarme tu confianza y hacernos conocer a mas amiguitas.

En verdad agradezco a esta gran institución que es el Politécnico por permitirme en poder desarrollarme en este camino del estudio, conocer maravillosas personas y a sus grandes profesores Alcántara, Halim, Luz María, Soto, Montiel, Asunción, Jessie, Nidia, Reyna, Pacheco. Ahora es tiempo de iniciar una nueva aventura, ¡gracias totales!

Marcos Ricardo García García

A mis padres:

Quienes a lo largo de mi vida me han apoyado, motivado en la vida y en mi formación académica y siempre se han preocupado por mí, gracias por todo el amor que me han dado por creer y confiar en mí, hoy con el apoyo de ustedes estoy terminando una etapa más de la vida, estoy muy orgulloso de ustedes y agradezco a Dios y a la vida de seguir disfrutando mis logros con ustedes. ¡Los amo! No tengo palabras para agradecer todo lo que han hecho por nosotros.

A mi familia:

A mi hermano Jaime por todas las enseñanzas que me ha dado desde que éramos pequeños, porque siempre fuimos inseparables, siempre hay algo que nos mantiene unidos, te quiero mucho gracias por todo el apoyo y confianza que me has brindado.

A mi primo Gerardo que considero un hermano, gracias por estar conmigo durante esta trayectoria, por escucharme y poder confiar en ti, por los momentos en que dejábamos todo y nos perdíamos jugando FIFA. En general quiero agradecer a todos mis primos y tíos que han estado al pendiente de esta trayectoria por todas esas palabras de aliento que me dieron para seguir adelante.

A mis amigos:

A todos mis amigos que he conocido durante la carrera, de los cuales siempre me apoyaron, estuvieron en los buenos y malos momentos y aprendí mucho de ustedes, gracias por darme esa confianza, por compartir tantas cosas dentro y fuera de ESCOM.

A mi mejor amigo Richard, ¡un gran hermano! por compartir momentos inolvidables a lo largo de toda la carrera, gracias por tu apoyo, tus palabras, tus consejos, tus regaños que bien me sirvieron para aprender de ello y salir adelante, siempre me escuchaste, por todo eso que hiciste por mí, por tu amistad incondicional. Hoy estamos terminando la carrera que gracias a nuestro esfuerzo y dedicación lo logramos, quien viera a Richard y Zamorano de primer semestre hoy ya somos todos unos ingenieros. Además quiero darte las gracias porque a pesar de ser un gran amigo fuiste un gran equipo de clases, de laboratorio, de servicio social y de TT.

A mi mejor amiga Miros, gracias por tu grandiosa amistad, no cabe duda que hemos compartido demasiadas cosas bonitas, pasado por buenos y malos momentos pero seguimos siempre fieles ante todo, gracias por apoyarme a cada instante de mi vida, por creer y confiar en mí, demostraste estar cuando lo necesitaba, por aguantarme todo este tiempo pero sobre todo gracias por haber llegado a mi vida y ser quien eres. ¡Te quiero mucho bonita!

Marianita, compañera, has sido una excelente amiga, lo has demostrado en cada palabra, en cada regaño, en cada abrazo, en cada acción, muchas gracias por la confianza y amistad, por los buenos consejos, nunca olvidaré el día en que nos conocimos, te quiero compañera.

A Vela quien agradezco toda su confianza, todo su apoyo, por los buenos momentos que pasamos durante todo este tiempo que llevamos de conocernos, gracias por ser un grandísimo amigo, siempre haciéndonos reír y preocupándote por nosotros. A Erik, Rene y Víctor que cada broma y comentario nos hacían morir de la risa y siempre apuntados para la fiesta, gracias por su amistad amiguitos.

A Emmanuel quien además de ser una excelente persona, lo considero un gran amigo, ¡eres un ejemplo a seguir! Te agradezco tu ayuda, paciencia, tiempo, esfuerzo y trabajo para que nuestra meta se llevara a cabo. A Marco Romero que ha sido un gran amigo desde la vocacional, siempre tan ocurrente y divertido, gracias por tu amistad y porque sé que cuento contigo.

¡Verito! Muchas gracias por tu grandiosa amistad, siempre ayudándonos en todo y muy amable, agradezco los buenos consejos, estar siempre pendiente de los amigos, admiro tu fortaleza para seguir adelante, por presentarnos a Montse que es una gran amiga que siempre nos contagia su energía y su ingenio para hacernos reír. Fercho, eres un gran amigo, un gran guía, alguien que siempre está dispuesto a ayudar, gracias por todo tu apoyo. Fátima, compañera, agradezco su amistad, el tiempo compartido con usted ha sido muy gratificante.

A Monse quien la vida me dio la oportunidad de conocerla, se ha convertido en una persona muy especial y ha estado en momentos importantes de mi vida, has demostrado ser una gran amiga, alegre, linda, sincera, me has dado la confianza necesaria para poder contarte mis traumas y sobre todo has demostrado tu gran cariño, ¡te quiero!... Amiguita Wendy ¡Picky! Siempre contagiando tu alegría y tu energía, gracias por el apoyo, confianza y tu amistad.

Agradezco de corazón las enseñanzas de todos y cada uno de mis maestros que me dieron clase a lo largo de toda mi carrera en ESCOM, de todos me llevo algo muy especial e importante.

José Eduardo Zamorano Aparicio

ÍNDICE GENERAL

ÍNDICE GENERAL	9
INTRODUCCIÓN	19
PROBLEMÁTICA.....	19
Descripción del Problema.....	19
Solución propuesta.....	20
JUSTIFICACIÓN	20
OBJETIVOS.....	21
Objetivo General.....	21
Objetivos Específicos.....	21
CAPÍTULO 1. MARCO TEÓRICO	23
1.1. Sistema Operativo Android.....	23
1.1.1. Características Sistema Android.....	24
1.1.2. Arquitectura Android.....	24
1.1.3. Comparación con otros Sistemas Operativos Móviles.....	25
1.1.4. Componentes principales Android.....	26
1.1.4.1. Actividades.....	26
1.1.4.1.1. Estado de una Actividad.....	26
1.1.4.1.2. Back stack.....	27
1.1.4.2. Fragmentos.....	27
1.1.4.3. Intención (intent).....	28
1.1.4.4. Layout.....	28
1.2. iBeacon	28
1.2.1. Características.....	28
1.2.2. Ventajas a otras Tecnologías.....	29
1.3. Localización en interiores.....	30
1.4. Bluetooth	30
1.4.1. Especificaciones.....	30
1.4.2. Bluetooth 4.0.....	31
1.5. Modelo Vista Controlador (MVC).....	31
1.6. REST	32
1.6.1. Ventajas.....	32
1.6.2. Desventajas	33
CAPÍTULO 2. ESTADO DEL ARTE.....	34

2.1. En el mundo.....	34
2.2. En México.....	35
CAPÍTULO 3. METODOLOGÍA.....	36
3.1. MDAM.....	36
CAPÍTULO 4. ANÁLISIS.....	38
4.1. Descripción General.....	38
4.2. Alcances.....	38
4.2.1. Productos esperados.....	39
4.3. Análisis de requerimientos	39
4.3.1. Requisitos funcionales.....	39
4.3.2. Requisitos no funcionales.....	40
4.4. Análisis de riesgos.....	42
4.4.1. Riesgos del desarrollo del sistema.....	42
4.4.2. Riesgos del sistema	44
4.5. Estudio de factibilidad	45
4.5.1. Factibilidad técnica	45
4.5.1.1. Herramientas para el desarrollo	45
4.5.1.2. Herramientas para pruebas e implementación	47
4.5.1.3. Requisitos de hardware y software	48
4.5.2. Factibilidad operativa	48
4.5.3. Factibilidad económica	48
4.6. Definición de roles	50
4.7. Propuesta de proceso general.....	51
CAPÍTULO 5. DISEÑO	53
5.1. Diagrama general del sistema	53
5.2. Diagrama a bloques.....	53
5.3. Diseño API	54
5.4. Glosario de términos	55
5.5. Reglas de negocio.....	55
5.6. Casos de uso	56
5.6.1. Casos de uso general	56
5.6.2. CU1: Visualizar home	59
5.6.3. CU2: Visualizar exhibiciones	61
5.6.4. CU3: Visualizar distribución del museo.....	64
5.6.5. CU4: Configurar aplicación	66

5.6.6.	CU5: Visualizar Ayuda	67
5.6.7.	CU6: Ver lista de material expuesto y exhibiciones	68
5.6.8.	CU7: Ver Notificaciones	74
5.6.9.	CU8: Iniciar sesión	76
5.6.10.	CU9: Gestión de Material	82
5.6.11.	CU10: Gestión de Exhibición	87
5.6.12.	CU11: Gestión de Archivos	91
5.6.13.	CU12: Gestión de Plantillas.....	93
5.6.14.	CU13: Gestión de variable de plantilla.....	95
5.6.15.	CU14: Recuperar contraseña	97
5.6.16.	CU15: Gestión de Museos	98
5.7.	Diagrama de base de datos	101
5.7.1.	Descripción de entidades	102
5.7.1.1.	Museum	102
5.7.1.2.	Material	102
5.7.1.3.	Exhibition.....	102
5.7.1.4.	Type_exhibition	103
5.7.1.5.	Template.....	103
5.7.1.6.	Template_variable.....	103
5.7.1.7.	File	103
5.7.1.8.	Key	104
5.7.1.9.	Type_variable	104
5.8.	Diagramas de clases	104
5.9.	Diagramas de secuencia	108
CAPÍTULO 6. DESARROLLO.....		117
6.1.	Desarrollo aplicación móvil.....	117
6.1.1.	Descarga e instalación Android Studio.....	117
6.1.2.	Configuración del entorno	117
6.1.3.	Actividad principal.....	117
6.1.3.1.	Barra de navegación.....	118
6.1.3.2.	Barra de menú	119
6.1.4.	Verificación de la conexión a Internet	119
6.1.5.	Verifica estado del bluetooth.....	119
6.1.6.	Icono de la aplicación.....	120
6.1.7.	ANDROID MANIFEST.....	120
6.1.7.1.	Permisos	120

6.1.8.	Lectura de datos del Beacon.....	121
6.1.8.1.	Android Beacon Library (AltBeacon).....	121
6.1.9.	Librería Volley.....	123
6.1.9.1.	Petición JSON (JsonObjectRequest, jsonArrayRequest).....	123
6.1.9.2.	Generar y añadir peticiones GET y POST	123
6.1.9.3.	Caché	124
6.1.10.	Fragmentos	125
6.1.10.1.	Fragmento Home.....	125
6.1.10.2.	Fragmento Exhibiciones y Distribución	126
6.1.10.3.	Fragmento Materiales y Exhibiciones	127
6.1.11.	Plantillas.....	129
6.1.12.	Archivos Descargables	131
6.1.13.	Configuración de la aplicación.....	132
6.1.14.	Codificación de Ayudas	134
6.2.	Desarrollo del Servidor Web	135
6.2.1.	Descarga e instalación MODX Revolution.....	135
6.2.2.	Configuración del entorno	135
6.2.3.	Módulos	136
6.2.4.	Acceso a MODx Revolution.....	138
6.2.5.	Connectors	139
6.2.6.	Core	139
6.2.7.	Manager	140
6.2.8.	Assets	140
6.2.9.	Code.....	140
6.2.10.	Modx.....	141
6.2.11.	TT	141
6.3.	Desarrollo API	144
6.3.1.	Requerimientos API	146
6.3.2.	Códigos de error	146
6.3.3.	POST museum/	146
6.3.4.	POST material/	148
CAPÍTULO 7. PRUEBAS DE FUNCIONAMIENTO		152
7.1.	Dispositivos reales.....	152
7.2.	Pruebas API	162
7.3.	Pruebas del servidor Web	163
7.4.	Prueba de Concurrencia.	168

7.5. Análisis de las 6 M's	169
CONCLUSIONES.....	170
TRABAJO A FUTURO.....	171
REFERENCIAS	172

ÍNDICE DE FIGURAS

Figura 1 Estadísticas sistemas operativos móviles.....	20
Figura 2 Arquitectura de Android	24
Figura 3 Estados de una Actividad	27
Figura 4 Representa la precisión de la localización de un dispositivo	30
Figura 5 Etapas de la metodología para el desarrollo de aplicaciones móviles.....	37
Figura 6 Diagrama de proceso general	52
Figura 7 Diagrama general del sistema	53
Figura 8 Diagrama a bloques del sistema.....	54
Figura 9 Diagrama general casos de uso de usuario	57
Figura 10 Diagrama general casos de uso administrador	58
Figura 11 CU1: Visualizar home	59
Figura 12 Pantalla home.....	60
Figura 13 Pantalla no hay conexión a internet.....	60
Figura 14 Pantalla Museo no encontrado	61
Figura 15 CU2: Visualizar exhibiciones.....	61
Figura 16 Pantalla exhibiciones.....	62
Figura 17 Pantalla no hay conexión a internet.....	63
Figura 18 Pantalla Exhibición no encontrada.....	63
Figura 19 CU3: Visualizar distribución del museo	64
Figura 20 Pantalla distribución del museo	65
Figura 21 Pantalla no hay conexión a internet.....	65
Figura 22 Pantalla distribución no encontrado.....	66
Figura 23 CU4: Configurar aplicación	66
Figura 24 Pantalla de configuración	67
Figura 25 CU5: Visualizar Ayuda.....	67
Figura 26 Pantalla de ayuda	68
Figura 27 CU6: Ver lista de material expuesto y exhibiciones	68
Figura 28 Pantalla de material expuesto y exhibiciones	71
Figura 29 Pantalla no hay conexión a internet.....	71
Figura 30 Información no encontrada.....	72
Figura 31 Plantilla imágenes	72
Figura 32 Plantilla texto	73
Figura 33 Plantilla video	73
Figura 34 Pantalla mensaje “Acércate al material o exhibición”	74
Figura 35 Pantalla de lista de archivos.....	74
Figura 36 CU7: Ver Notificaciones.....	74
Figura 37 Pantalla de notificaciones	76
Figura 38 CU8: Iniciar sesión	76
Figura 39 Pantalla inicio de sesión	78
Figura 40 Pantalla de Acceso.....	78
Figura 41 Pantalla campos incorrectos	79
Figura 42 Pantalla Campos Incompletos.....	79
Figura 43 Pantalla Principal.....	80
Figura 44 Pantalla Menú Sesión.....	80

Figura 45 Pantalla Cambiar contraseña	80
Figura 46 Pantalla Contraseña actualizada	81
Figura 47 Pantalla Contraseña no coincide	81
Figura 48 Pantalla Faltan Campos	81
Figura 49 Pantalla Cerrar Sesión	82
Figura 50 CU9: Gestión de material	82
Figura 51 Pantalla de Gestionar Material	86
Figura 52 Pantalla Detalle Material	87
Figura 53 CU10: Gestión de exhibición	87
Figura 54 Pantalla Gestionar Exhibición	90
Figura 55 Pantalla Detalle de Exhibición	91
Figura 56 CU11: Gestión de Archivos	91
Figura 57 Pantalla de Gestionar Archivo	93
Figura 58 Pantalla de Detalle de Archivo	93
Figura 59 CU12: Gestión de Plantilla	93
Figura 60 Pantalla Gestionar Plantilla	95
Figura 61 CU13: Gestión de variable de plantilla	95
Figura 62 Pantalla Gestionar Variable de Plantilla	97
Figura 63 CU14: Recuperar contraseña	97
Figura 64 CU15: Gestión de Museos	98
Figura 65 Diagrama de base de datos	101
Figura 66 Diagrama de clases servidor web (Modelo)	105
Figura 67 Diagrama de clases servidor web (Vista)	105
Figura 68 Diagrama de clases servidor web (Controlador)	106
Figura 69 Diagrama de clases servidor web (Objetos)	107
Figura 70 Diagrama de clases aplicación móvil	108
Figura 71 Diagrama de secuencia visualizar home	108
Figura 72 Diagrama de secuencia Visualizar exhibición	109
Figura 73 Diagrama de secuencia visualizar mapa	109
Figura 74 Diagrama de secuencia configurar aplicación	109
Figura 75 Diagrama de secuencia visualizar ayuda	110
Figura 76 Diagrama de secuencia ver lista de material expuesto y exhibiciones	110
Figura 77 Diagrama de secuencia ver notificaciones	111
Figura 78 Diagrama de secuencia iniciar sesión	111
Figura 79 Diagrama de secuencia gestión de material	112
Figura 80 Diagrama de secuencia gestión de exhibición	113
Figura 81 Diagrama de secuencia gestión de archivos	114
Figura 82 Diagrama de secuencia gestión de plantillas	114
Figura 83 Diagrama de secuencia gestión de variable	115
Figura 84 Diagrama de secuencia recuperar contraseña	115
Figura 85 Diagrama de secuencia gestionar museo	116
Figura 86 Configuración del entorno	117
Figura 87 Navigation Drawer Activity	118
Figura 88 Desarrollo de la Barra de navegación	118
Figura 89 Barra de menú	119
Figura 90 Verificación de la conexión a internet	119
Figura 91 Verifica bluetooth	120
Figura 92 Icono de la aplicación	120
Figura 93 Permisos	121

Figura 94 Permisos especiales Android 6.....	121
Figura 95 Agregando Altbeacon.....	121
Figura 96 Código obtener información del Beacon	122
Figura 97 Región Beacon.....	122
Figura 98 Constructor de peticiones tipo JsonObjectRequest.....	124
Figura 99 Envío de parámetros al servidor	124
Figura 100 Almacenar imágenes en caché.....	125
Figura 101 Fragmento Home	126
Figura 102 Fragmento Exhibiciones	126
Figura 103 Fragmento de Distribución.....	127
Figura 104 Fragmento de materiales y exhibiciones	128
Figura 105 Información del Beacon con material asignado.....	128
Figura 106 Plantilla texto	129
Figura 107 Plantilla imagen.....	130
Figura 108 Plantilla de video	130
Figura 109 Reproducción de video.....	131
Figura 110 Archivos descargables	132
Figura 111 Mensaje "Archivo ya existe!"	132
Figura 112 Configuración (Intent)	133
Figura 113 Pantalla configuración.....	134
Figura 114 Botón de ayuda	134
Figura 115 Sección de Ayuda	135
Figura 116 Permisos de instalación MODX Revolution	135
Figura 117 Opciones de Base de Datos para MODX Revolution.....	136
Figura 118 Ingresar datos de Administrador en MODX Revolution.....	136
Figura 119 Módulo Login de Inicio de Sesión.....	136
Figura 120 Módulo Login de cerrar sesión.....	136
Figura 121 Módulo Wayfinder	137
Figura 122 Módulo QuickCrumbs.....	138
Figura 123 Acceso a MODx Revolution	139
Figura 124 Directorio Raíz MODx Revolution.....	139
Figura 125 Directorio Assets	140
Figura 126 Directorio Code	140
Figura 127 Fragmento de la clase F.....	141
Figura 128 Directorio assets/code/tt	142
Figura 129 Fragmento de código en clases	142
Figura 130 Fragmento de código en controlador	143
Figura 131 Fragmento de código en modelo	143
Figura 132 Directorio assets/media	144
Figura 133 Directorio js	144
Figura 134 Fragmento de código en clase de js	144
Figura 135 Creación de JSON en API	145
Figura 136 Códigos de acceso.	145
Figura 137 Consulta SQL de API	146
Figura 138 Error dispositivo virtual	152
Figura 139 Opciones del programador Nexus 6.....	153
Figura 140 Seleccionar dispositivo para ejecutar.....	153
Figura 141 Pruebas conexión a internet Nexus 6.....	154
Figura 142 Prueba visualización de la información Nexus 6	155

Figura 143 Prueba visualización de la información Galaxy Tab S2	155
Figura 144 Prueba acceso a la ayuda en Nexus 6	156
Figura 145 Pruebas notificaciones Nexus 6.....	157
Figura 146 Pruebas notificaciones Galaxy Tab S2	157
Figura 147 Pruebas materiales y exhibiciones 1	158
Figura 148 Pruebas materiales y exhibiciones 2	159
Figura 149 Pruebas plantillas texto, imagen y video.....	159
Figura 150 Pruebas reproductor de video	160
Figura 151 Prueba lista de archivos descargables.....	160
Figura 152 Pruebas notificaciones descarga de archivos.....	161
Figura 153 Prueba apertura de archivos	161
Figura 154 Error de credenciales API en Android Studio	162
Figura 155 Prueba respuesta de la API en Android Studio	163
Figura 156 Error de credenciales API en navegador web	163
Figura 157 Prueba de respuesta de la API en navegador web.....	163
Figura 158 Login Servidor Web	164
Figura 159 Error de Login en Servidor Web	164
Figura 160 Pruebas menú principal del Servidor Web.....	165
Figura 161 Pruebas acceso al menú Plantillas	165
Figura 162 Prueba alta de información en el Servidor Web	165
Figura 163 Prueba alta de imágenes en el servidor web	166
Figura 164 Prueba cantidad de información a mostrar en el Servidor Web.....	166
Figura 165 Prueba muestra de la información en el Servidor Web	166
Figura 166 Prueba de la visualización de la información a detalle en el servidor web.....	167
Figura 167 Prueba editar Variable de Plantilla en el Servidor Web.....	167
Figura 168 Prueba de eliminación de información del Servidor Web	167
Figura 169 Ejecución ApacheBench	168
Figura 170 Respuesta de Peticiones.....	168

ÍNDICE DE TABLAS

Tabla 1 Versiones de Android	23
Tabla 2 Características de las principales plataformas móviles	26
Tabla 3 Información proporcionada por iBeacon	29
Tabla 4 Estados de la proximidad de un Beacon.....	29
Tabla 5 Clases de Bluetooth	31
Tabla 6 Aplicaciones en el mercado con IBeacon	35
Tabla 7 Aplicaciones en México con IBeacon.....	35
Tabla 8 Análisis de riesgos del desarrollo del sistema	44
Tabla 9 Análisis de riesgos del sistema	44
Tabla 10 Requisitos mínimos recomendados Ubuntu server	45
Tabla 11 Requisitos del cliente.....	48
Tabla 12 Requisitos del servidor	48
Tabla 13 Promedio del salario mensual	49
Tabla 14 Costo del desarrollo del sistema	49
Tabla 15 Servicios.....	49
Tabla 16 Costo del hardware	50
Tabla 17 Costo total del sistema	50
Tabla 18 Definición de roles.....	51
Tabla 19 CU1: Visualizar home.....	59
Tabla 20 CU2: Visualizar exhibiciones	61
Tabla 21 CU3: Visualizar distribución del museo	64
Tabla 22 CU4: Configurar aplicación	66
Tabla 23 CU5: Visualizar Ayuda.....	68
Tabla 24 CU6: Ver lista de material expuesto y exhibiciones	69
Tabla 25 CU7: Ver Notificaciones.....	75
Tabla 26 CU8: Iniciar sesión	76
Tabla 27 CU9: Gestión de material.....	83
Tabla 28 CU10: Gestión de exhibición	88
Tabla 29 CU11: Gestión de Archivos	91
Tabla 30 CU12: Gestión de Plantilla	94
Tabla 31 CU13: Gestión de variable de plantilla	95
Tabla 32 CU14: Recuperar contraseña.....	97
Tabla 33 CU15: Gestión de Museos	98
Tabla 34 Propiedades de módulo Login.....	137
Tabla 35 Parámetros de módulo Wayfinder	137
Tabla 36 Propiedades de módulo QuickCrumb.....	138
Tabla 37 Prueba de distancia espacio abierto	156
Tabla 38 Prueba de distancia espacio cerrado.....	157
Tabla 39 Análisis de las 6 M's.....	169

INTRODUCCIÓN

“Un recurso que posibilita la mejor adquisición por parte del alumnado de contenidos conceptuales, procedimentales y actitudinales puede ser el museo, y si este es interactivo, mucho mejor (Pérez, Díaz, Echevarría, Moretín y Cuesta, 1998)” pues bien sabido es que lo que se hace se retiene más y mejor que lo que simplemente se ve. [1]

El museo es un espacio de educación informal a través del cual se pretende que el alumnado y la población en general adquieran conocimientos mediante métodos que generan un complemento a la educación formal. Si este espacio cultural se presenta como un espacio de ocio –entendida la palabra ocio como el momento en el que las personas aumentan su riqueza personal- en el que la interactividad tiene un lugar privilegiado y el diálogo es más ágil y divertido, el museo tendrá con seguridad cumplida su meta de divulgación. [2]

Sin embargo, tomando en cuenta que el Sistema de Información Cultural de *CONACULTA* nos arroja que existen 1250 museos en todo el país [3], el uso de las tecnologías para la divulgación científica ha sido un recurso utilizado pero se podría decir, poco explotado. No solo eso, según el “Diagnóstico informático para el aprovechamiento de las tecnologías de información para el instituto nacional de antropología e Historia (*INAH*)” [4] se establece un *FODA* general de los museos que están asociados al *INAH*: la cual nos dice que la información de la organización está fragmentada, en algunas áreas existe la centralización de la información y en otras se mantiene por colaborador mediante un respaldo de la parte que trabaja en cada proyecto específico, y no existe el estímulo para compartirlo ni los mecanismos para aprovecharlo. [5]

Tomando como punto de partida la necesidad de los museos de México de implementar un sistema de interactividad y contar con un repositorio de información electrónica que permita la difusión del conocimiento entre áreas, se sugiere que una forma de conseguirlo es utilizando nuevas tecnologías tales como los teléfonos inteligentes conocidos como “*Smartphone*”. En la actualidad los *Smartphone* cumplen roles muy importantes en la vida cotidiana de las personas, ya que se han convertido en la herramienta indicada para el intercambio de información y el ocio.

PROBLEMÁTICA

En la siguiente sección se describe de manera general la problemática a la que nos enfrentamos y una solución propuesta.

Descripción del Problema

Actualmente algunos museos están diseñados como espacios didácticos con el fin de atraer a los visitantes para que aprendan e interactúen con lo expuesto en el lugar, sin embargo en ocasiones la información sobre las exposiciones o piezas que se encuentran en el museo no están actualizadas o incompletas y esto provoca desinformación a los visitantes.

Esto ocurre porque la información de los museos está fragmentada, y en algunas ocasiones la información se encuentra centralizada en diferentes áreas, otras se mantienen con los colaboradores que trabajan en un proyecto específico y no existe el estímulo para compartirlo ni los mecanismos para aprovechar toda la información.

Esto trae:

- Fuga del conocimiento de los museos acerca de los proyectos y las funciones desempeñadas de cada área o sala específica.
- Falta de disponibilidad de la información.
- Información incompleta e imprecisa.

Solución propuesta

Ofrecemos como solución en este Trabajo Terminal el desarrollo de una aplicación móvil y sistema web para así poder dar una visita guiada en un museo, dando los siguientes beneficios:

- Un sistema de administración y mantenimiento de la información de las exhibiciones y materiales del museo.
- Dotar a cada usuario con su propia guía y asistencia turística actualizada en un dispositivo móvil.
- Elaborar plantillas para mostrar la información de cada exhibición y materiales.

JUSTIFICACIÓN

Actualmente las estadísticas de los sistemas operativos móviles en México son:

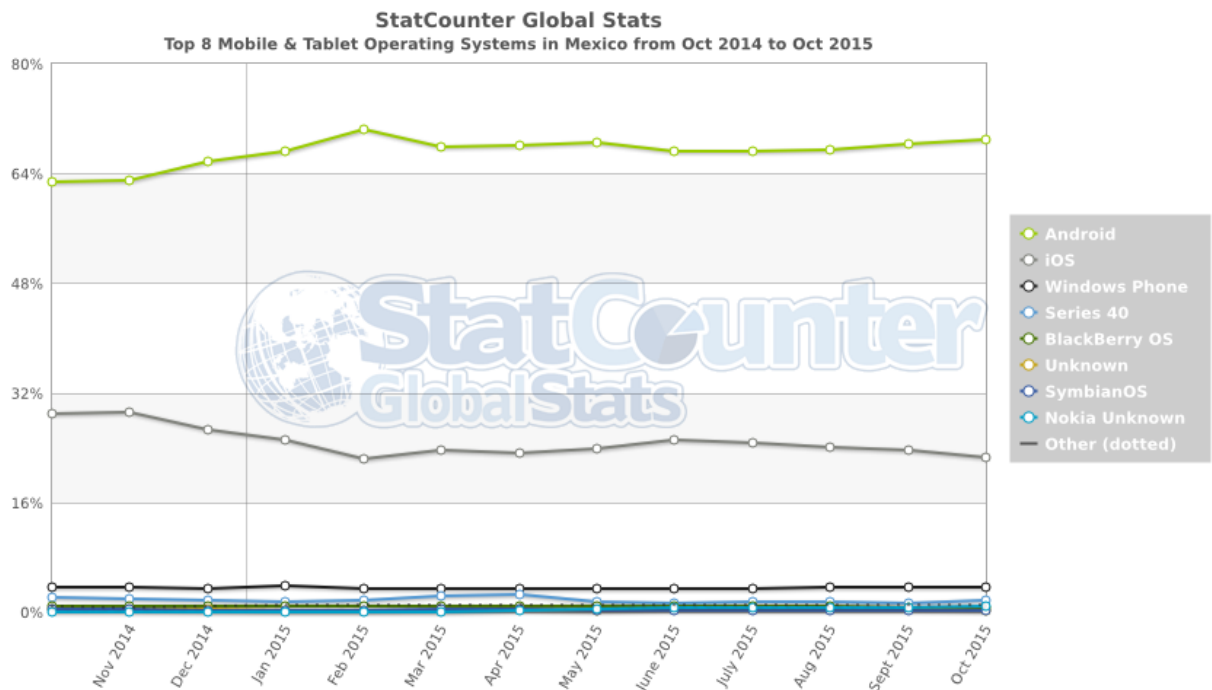


Figura 1 Estadísticas sistemas operativos móviles

[6]

Android da como ganador hasta la fecha con un mayor porcentaje sobre IOS, por esta razón se utiliza el sistema operativo Android para el desarrollo de la aplicación móvil.

¿Por qué iBeacon?

Se utiliza la tecnología **iBeacon** debido a que el uso de esta tecnología inalámbrica permite cubrir más distancia (hasta 50 m) y menor consumo de energía que la tecnología NFC (de corto alcance y alta frecuencia) y mejor precisión en geo localización en proximidad dentro de edificios (llamada *IPS-indoor positioning system*) con respecto a la tecnología GPS (que genera un alto consumo eléctrico). **iBeacon** por ser un sistema de posicionamiento de interiores (*IPS-indoor positioning system*) nos permite conocer con mucha exactitud la distancia a la cual se encuentra de uno o varios **Beacon** de manera múltiple, con esto se puede interpretar el interés de la persona aproximándose a las exposiciones del museo (pintura, escultura, etc.) con lo cual se iniciará una pre-descarga de los datos para que al llegar ya cuente con la información necesaria.

Diseño del Museo:

La utilización de **iBeacon** también influye por el diseño del museo en que se va a implementar.

Esta tecnología nos permite (además de las ventajas antes mencionadas) la eficacia del IPS contra el **GPS** y **NFC** en un museo con las siguientes características:

- Todas o la mayoría del material expuesto del museo se encuentre en el edificio interior. El GPS tiene problemas de conexión en áreas cerradas.
- Cuento con áreas subterráneas donde se obstruye o se imposibilita la comunicación con el GPS.
- El museo cuente con muchas visitas al día, lo cual podría ocasionar un conglomerado de personas en un o más materiales expuestos (pinturas, esculturas, salas). Se requiere que el intercambio de información se ajuste a la proximidad de las personas a la escultura. El NFC solo permite el intercambio de información a una distancia máxima de 20 cm.

OBJETIVOS

A continuación se describe el objetivo general del trabajo terminal así como también se enlistan los objetivos específicos de este.

Objetivo General.

Implementar un sistema de intercambio de información empleando la tecnología **iBeacon** para la generación de un sistema interactivo de guías y asistente turístico de museos mediante una aplicación móvil para sistema operativo Android, que proporcione al visitante información actualizada de las exposiciones y/o materiales que se encuentren dentro del museo.

Objetivos Específicos.

Los objetivos específicos son:

- Desarrollo de la aplicación móvil que permita la búsqueda, comunicación y recepción de información básica del **Beacon**.
- La aplicación móvil debe mostrar información sobre las exhibiciones y el material expuesto dentro del museo.

- La aplicación móvil realiza una pre descarga de la información (de un servidor web) dependiendo de la ubicación del usuario dentro del museo.
- La visualización de la información debe ser mediante plantillas.
- La conexión y descarga de la información al servidor web se realiza a través de una API (*Application Programming Interface*).
- Desarrollo de un sistema web para la administración de información del museo.

CAPÍTULO 1. MARCO TEÓRICO

Se desarrolla la teoría la cual va a fundamentar el trabajo terminal con base a la problemática que se presenta.

1.1. Sistema Operativo Android.

En el año 2005, Google adquiere a la compañía Android Inc. La cual acababa de ser creada y era orientada a la producción de aplicaciones móviles para terminales móviles.

En el año 2007 se crea *Handset Alliance* el cual es un consorcio con el objetivo de desarrollar estándares abiertos para móviles y está conformado por las grandes empresas Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel entre otros. Su objetivo es promover el diseño y la difusión de la plataforma Android la cual sería publicada como una parte importante de su propiedad intelectual como código abierto bajo la licencia Apache v2.0. [7]

En el 2007 se lanza la primera versión de Android SDK. En el 2008 aparece el primer dispositivo móvil con sistema Android (T-Mobile G1). Ese mismo año, Google libera el código fuente de Android bajo licencia de código abierto Apache (GPL v2 para el núcleo). En Octubre del mismo año se lanza el Android Market para la descarga de varias aplicaciones y desde esas fechas han sacado varias versiones del sistema.

La tabla 1 se muestra las versiones de Android.

Nombre	Versión	Nivel API	Fecha
Apple Pie	1.0	1	Septiembre 2008
Banana Bread	1.1	2	Febrero 2009
Cupcake	1.5	3	Abril 2009
Donut	1.6	4	Septiembre 2009
Éclair	2.0	5	Octubre 2009
Éclair	2.1	7	Enero 2010
Froyo	2.2	8	Mayo 2010
Ginger Bread	2.3	9	Diciembre 2010
Honeycomb	3.0	11	Febrero 2011
Honeycomb	3.1	12	Mayo 2011
Honeycomb	3.2	13	Julio 2011
Ice Cream Sandwich	4.0	14	Octubre 2011
Ice Cream Sandwich	4.0.3	15	Diciembre 2011
Ice Cream Sandwich	4.0.4	15	Marzo 2012
Jelly Bean	4.1	16	Julio 2012
Jelly Bean	4.2	17	Noviembre 2012
Jelly Bean	4.3	18	Julio 2013
KitKat	4.4	19	Octubre 2013
Lollipop	5.0	21	Noviembre 2014
Marshmallow	6.0	23	Septiembre 2015 [8]

Tabla 1 Versiones de Android

1.1.1. Características Sistema Android.

Las características del sistema Android son las siguientes:

- Plataforma realmente abierta. Es una plataforma de desarrollo libre, la cual está basada en Linux y es de código abierto. Su mayor ventaja es el poder usarlo y modificar el sistema sin necesidad de hacer un pago.
- Adaptable a cualquier tipo de Hardware. Este sistema tiene la cualidad de no estar diseñado exclusivamente para uso en teléfonos y tabletas, si no también podemos encontrarlo en relojes, cámaras y electrodomésticos. Esto conlleva un esfuerzo adicional a los programadores, ya que debe funcionar en dispositivos con gran variedad de tipos de entrada, pantalla, memoria entre otros.
- Portabilidad asegurada. Las aplicaciones son desarrolladas en Java, lo cual nos asegura que pueden ser ejecutadas en cualquier tipo de CPU, gracias al concepto de máquina virtual.
- Filosofía de dispositivo siempre conectado a Internet.
- Arquitectura basada en componentes inspirados en Internet.
- Servicios incorporados. Ejemplo de ello son, la localización en GPS, bases de datos con SQL, navegador, multimedia, etc.
- Nivel de Seguridad. Los programas se encuentran aislados, gracias a la ejecución dentro de una caja que hereda del sistema Linux y además cada aplicación tiene una serie de permisos que limitan su rango de actuación.
- Alta calidad de gráficos y sonido. Contiene gráficos vectoriales suavizados, animaciones en Flash, gráficos en OpenGL. [9]

1.1.2. Arquitectura Android

La siguiente figura muestra la arquitectura de Android, está conformada por cuatro capas y todas son basadas en software libre.

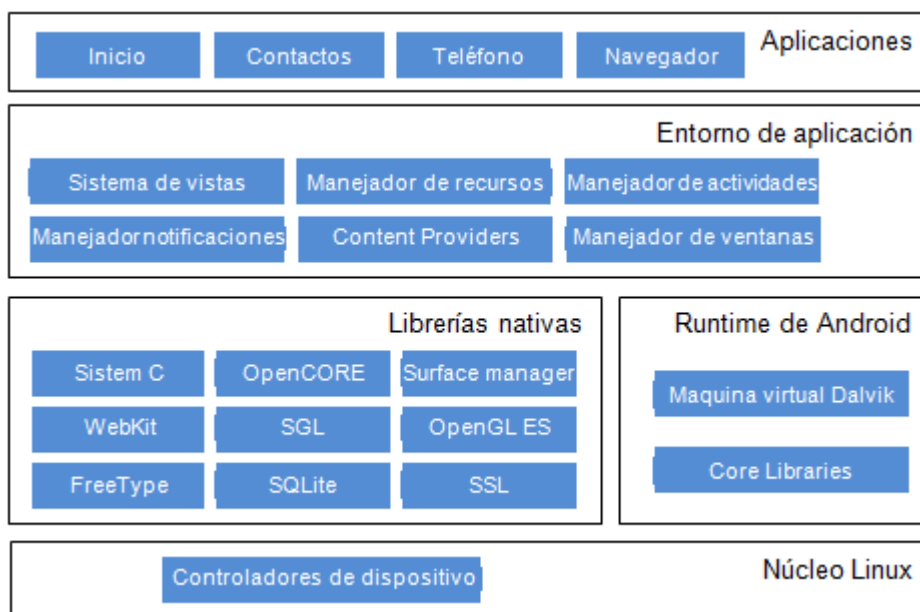


Figura 2 Arquitectura de Android

- Núcleo Linux. El núcleo está formado por el sistema operativo Linux en su versión 2.6. esta capa da servicios como la seguridad, memoria, multiproceso, protocolos y drivers para los dispositivos.
- Runtime de Android. Se utiliza la máquina virtual Dalvik, porque responde mejor a las limitaciones que da la máquina virtual de Java (poca memoria y procesador limitado). Las características con las que cuenta es que facilitan la optimización de recursos, como el ahorrador de energía (formato .dex). Otra de ellas es que cada aplicación corre su propio proceso con su instancia y el *kernel* de Linux delega algunas funciones como el *threading* y el manejo de memoria de bajo nivel.
- Librerías nativas. Contiene librerías en C/C++ las cuales son usadas en componentes Android y la mayoría son utilizadas en proyectos de código abierto. Algunas de ellas son: System C library, Media Framework, Surface Manager, Webkit, SGL, Librerías 3D, FreeType, SQLite, SSL, entre otras.
- Entorno de aplicación. Fue diseñada para simplificar la reutilización de componentes. También esta capa aprovecha el lenguaje de programación Java, y el SDK no puede ofrecer todo lo de su entorno pero es compatible con una fracción muy significativa de la misma. Algunos servicios que incluyen son los *views*, *resource manager*, *activity manager*, *notification manager*, *content providers*, etc.
- Aplicaciones. Está formado por el conjunto de aplicaciones instaladas en una maquina Android.

[10]

1.1.3. Comparación con otros Sistemas Operativos Móviles.

En la tabla 2 se describen las características de las principales plataformas móviles que existen actualmente.

Plataforma	Apple iOS 6	Android 5.0	Windows Phone 7	BlackBerry OS 7	Symbian 9.5
Compañía	Apple	Open Handset Alliance	Windows	RIM	Symbian Foundation
Núcleo S.O.	Mac OS X	Linux	Windows CE	Mobile OS	Mobile OS
Familia CPU soportada	ARM	ARM, MIPS, Power, x86	ARM	ARM	ARM
Lenguaje de programación	Objetive -C, C++	Java, C++	C#, muchos	Java	C++
Licencia de Software	Propietaria	Software Libre y abierto	Propietaria	Propietaria	Software Libre
Año de Lanzamiento	2007	2008	2010	2003	1997
Motor del navegador Web	Web Kit	Web Kit	Pocket Internet Explorer	Web Kit	Web Kit
Soporte Flash	No	Si	No	Si	Si
HTML 5	Si	Si	Si	Si	No
Tienda de Aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry App World	Ovi Store
Numero de Aplicaciones	400,000	300,000	50,000	30,000	50,000

Coste Publicación	\$99 / año	\$25 una vez	\$99 / año	Sin Coste	\$1 una vez
Plataforma de Desarrollo	MAC	Windows, MAC, Linux	Windows	Windows, MAC	Windows, MAC, Linux
Actualizaciones automáticas del SO	Si	Depende Fabricante	Depende Fabricante	Si	Si
Soporte memoria externa	No	Si	No	Si	Si
Fabricante Único	Si	No	No	Si	No
Variedad de Dispositivos	Modelo único	Muy alta	Baja	Baja	Muy alta
Tipo de Pantalla	Capacitiva	Capacitiva / Resistiva	Capacitiva	/ Resistiva capacitiva	Capacitiva / Resistiva
Aplicaciones Nativas	Si	Si	No	No	Si [11]

Tabla 2 Características de las principales plataformas móviles

1.1.4. Componentes principales Android.

El *framework* de Android se compone de algunos elementos importantes para la creación de aplicaciones.

1.1.4.1. Actividades

Una actividad un componente de Android que ofrece una pantalla con la que los usuarios pueden interactuar con la aplicación. Cada actividad tiene asociada una ventana en la que se dibuja la interfaz de usuario.

Los dos métodos *callback* más importantes en la creación de actividades son:

onCreate(): este método es invocado cuando se crea la actividad, dentro de este debemos iniciar todos los elementos esenciales de la actividad, como dibujar la interfaz de usuario empleando la función **setContent()**.

onPause(): el sistema llama a este método cuando el usuario detiene la actividad, esto no significa que la actividad se destruya. El programador debe guardar los cambios para que persistan en la siguiente sesión del usuario, porque esta aplicación no podrá volver a la actividad y se destruye.

[12]

1.1.4.1.1. Estado de una Actividad

Una actividad puede mantener los siguientes estados:

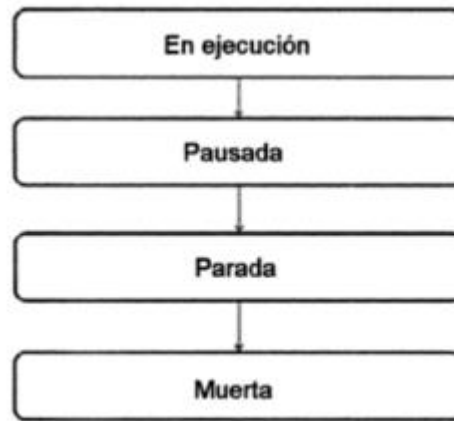


Figura 3 Estados de una Actividad

En ejecución: la actividad se encuentra en primer plano y recibe las interacciones del usuario.

Pausada: la actividad esta visible pero el usuario no puede interactuar con ella (oculta por un cuadro de diálogo, por ejemplo).

Parada: la actividad ya no es visible pero sigue en ejecución. Todos los datos relativos a su ejecución se conservan en memoria.

Muerta: se ha matado la actividad, ya no está en ejecución y todos los datos presentes en caché se eliminan. [13]

1.1.4.1.2. *Back stack*

Todas las actividades iniciadas se almacenan en una lista llamada *back stack*. Cuando se inicia una actividad nueva, esta se añade al principio de la *back stack*.

Cuando el usuario presiona la tecla de “atrás” la actividad en curso se cierra y la que se encontraba debajo de ella en la *back stack* se abre. [14]

1.1.4.2. *Fragmentos*

Este concepto se introdujo a partir de la versión 3.0 de Android. Permite construir interfaces más flexibles, dinámicas y fáciles de utilizar.

Un Fragmento es un componente que debe adjuntarse a una actividad para poder utilizarse, una actividad puede componerse de varios fragmentos. Su ciclo de vida se parece al de la actividad que lo contiene pero con algunas particularidades.

La primera etapa del ciclo de vida de un fragmento se corresponde con el instante en que el fragmento se adjunta a la actividad que lo contiene (**onAttach**). A continuación, el fragmento se inicializa en la llamada al método **onCreate**, seguido de la creación y de la carga de la interfaz del fragmento (método **onCreateView**).

Cuando un fragmento pasa a estar inactivo, la llamada al método **onPause** permite ejecutar las acciones adecuadas (deshabilitar actualizaciones, listener...).

Después de esta llamada se invoca al método **onStop**.

Si se destruye un fragmento, se invoca respectivamente los siguientes métodos:

- **onDestroyView:** destrucción de la vista.
- **onDestroy:** destrucción del fragment.
- **onDetach:** el fragmento se desvincula de la actividad que lo contiene. [15]

1.1.4.3. *Intención (intent)*

Una intención representa la voluntad de realizar alguna acción; como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que queramos:

- Lanzar una actividad
- Lanzar un servicio
- Enviar un anuncio de tipo broadcast
- Comunicarnos con un servicio

Los componentes lanzados pueden ser internos o externos a la aplicación móvil. También utilizaremos las intenciones para el intercambio de información entre estos componentes. [16]

1.1.4.4. *Layout*

Un *layout* es un conjunto de vistas agrupadas de una determinada forma. Vamos a disponer de diferentes tipos de *layouts* para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los *layouts* también son objetos descendientes de la clase *View*. Igual que las vistas, los *layouts* pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML. [17]

1.2. iBeacon

Entre las nuevas tecnologías que han salido, existe una en especial basada en **iBeacon**. Un **Beacon** es un transmisor en una ubicación conocida, que transmite una señal de radio continua o periódica con contenido limitado de información (por ejemplo, su identificación o ubicación), en una frecuencia de radio especificado. Cualquier tecnología de radio se puede utilizar para identificar la ubicación. [18]

Apple Inc. llama "**iBeacon**" a su sistema de posicionamiento de interiores (**IPS-*indoor positioning system***) y el mismo lo define como "una nueva clase de transmisores de baja potencia de bajo costo que pueden notificar iOS 7 dispositivos cercanos a su presencia, proporciona aplicaciones de un nuevo nivel de conocimiento de la ubicación." [19]. De los Smartphone actuales que lo soportan son: iPhone 4s y después, iPad 3, mini y después, también dispositivos móviles con Android con *Bluetooth* 4.0 y Android 4.3 y después como los Samsung Galaxy S3/S4/S4 Mini/, Samsung Galaxy Note II/III, HTC One, Nexus, etc. [18] Técnicamente **iBeacon** se basa en *Bluetooth Low Energy* (BLE), también conocida como *Bluetooth* 4.0 o *Bluetooth* inteligente.

1.2.1. Características

Un **iBeacon** proporciona la siguiente información a través de **Bluetooth Low Energy**:

Campo	Tamaño	Descripción
UUID	16 bytes	Los desarrolladores deben definir una especificación de UUID a la aplicación y el uso de su despliegue.
Mayor	2 bytes	Especifica un caso específico para el uso del Beacon. Por ejemplo, se podría definir una subregión dentro de una región definida por el UUID.
Minor	2 bytes	Permite una mayor subdivisión de la región o de casos de uso, especificado por el desarrollador de la aplicación.

Tabla 3 Información proporcionada por iBeacon

Los dispositivos con tecnología **iBeacon** pueden ser alimentados mediante pilas de botón durante uno o dos meses, o tienen la opción de ser alimentados por una fuente de alimentación fija, como un puerto USB de una terminal electrónica de punto de venta (POS).

El Beacon permite un nivel de flexibilidad para definir las regiones de una ubicación geográfica (latitud y longitud) con un identificador y un radio conocido como **geofence**. Esto permite que el Beacon que se fijan a los objetos no estén vinculados a una sola ubicación.

Cuando un dispositivo móvil detecta la señal del Beacon, es a través de la señal RSSI (*Received Signal Strength Indication*) y esto lo utiliza para determinar la proximidad y la precisión de su ubicación, entre más fuerte sea la señal, más próximo está el Beacon.

Ranging

iOS 7 introduce un nuevo conjunto de APIs para determinar la proximidad de un dispositivo que utiliza la tecnología **iBeacon**, esto es un proceso conocido como "ranging". Sobre escenarios de uso común, iOS aplica filtros para estimar la precisión y así determinar un estado de proximidad a un Beacon.

Estado de Proximidad	Descripción
Immediate	Indica que el dispositivo esta físicamente muy cerca del Beacon y representa un alto nivel de exactitud.
Near	Indica que el dispositivo está en una línea de visión clara con el Beacon, entre una proximidad de 1 a 3 metros.
Far	Indica que el dispositivo es detectado pero su nivel de exactitud es bajo para determinar su estado (cerca o inmediato).
Unknown	La proximidad del Beacon no se puede determinar. Esto indica que no hay medidas suficientes a determinar el estado.

Tabla 4 Estados de la proximidad de un Beacon

1.2.2. Ventajas a otras Tecnologías.

Difícil de utilizar en ambientes interiores: GPS generalmente no son adecuados para establecer lugares de interior, ya que las microondas serán atenuadas y reflejadas por los techos, paredes y otras obstrucciones. Por ejemplo, en los lugares como sótanos, subterráneos y túneles, se podría tener una señal raramente débil o incluso peor. Además, cuando la red móvil está congestionado o en algunos lugares que las señales de móviles están bloqueados a propósito, es imposible que usted cargue su dirección.

Cuando un dispositivo recibe una señal GPS muy clara, indica que el dispositivo está al aire libre y tiene una línea de visión abierta, esto quiere decir, que no hay obstáculos a los satélites GPS y trae consigo una ubicación más precisa, un ejemplo de ello, es la aplicación Google Maps, donde la precisión de la localización es representado por un círculo azul indicando su lugar actual. Este círculo azul cambia a un radio mayor cuando el dispositivo se encuentre en un interior o la línea de visión de los satélites se obstruye e indica esto una menor precisión de su ubicación, ya que podría estar situado en cualquier lugar del círculo azul.

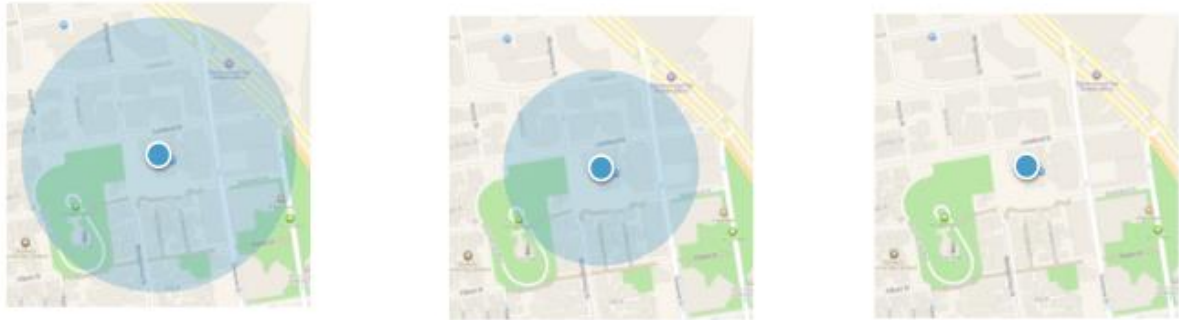


Figura 4 Representa la precisión de la localización de un dispositivo

1.3. Localización en interiores

La localización en interiores es una de las áreas más prometedoras en el campo de la computación móvil. Estos sistemas permiten desarrollar innumerables aplicaciones gracias al posicionamiento de objetos o personas en tiempo real. El número de ellas que actualmente requieren de un sistema de localización en los interiores de un edificio es elevado.

Algunas de las áreas de interés son las siguientes:

- **Ámbito sanitario:** control de localización en cada momento de pacientes y personal sanitario para la disminución del tiempo de respuesta ante emergencias.
- **Ocio y entretenimiento:** conocimiento de la localización de visitantes en emplazamientos tales como museos, exposiciones, recintos deportivos, para ofrecer un mejor servicio y dotarlos de medidas de seguridad en caso de extravío de personas.
- **Seguridad:** para definir zonas dentro de los recintos de forma que solo algunos usuarios autorizados puedan tener acceso a ellas. [20]

1.4. Bluetooth

Es una tecnología de corto alcance que pertenece a las WPAN, y tiene como estándar la IEEE 802.15.1, esta permite el intercambio de información entre dispositivos mediante un enlace por radiofrecuencia seguro, con una frecuencia de 2.4 GHz, una distancia de hasta de 10 metros y una fácil sincronización.

1.4.1. Especificaciones.

Algunas de las especificaciones del *bluetooth* son:

- Frecuencia de operación de 2.4 GHz.

- Red de dispositivos llamada Piconet y tiene un número máximo de dispositivos interconectados de ocho, tiene un dispositivo master que inicia y controla la comunicación y los siete restantes son pasados a esclavos.
- Pueden interconectarse más de ocho a la Piconet pero pasan a estado inactivo y son llamados Parked.
- Posee 79 canales y cada uno contiene un ancho de banda de 1 MHz.
- Tiene tres clases de *Bluetooth* :

Transmisor	Potencia Máxima de Transmisión (mW)	Potencia Máxima de Transmisión (dBm)	Alcance
Clase I	100 mW	20 dBm	100 m
Clase II	2.5 mW	4 dBm	10 m
Clase III	1 mW	0 dBm	10 cm

Tabla 5 Clases de Bluetooth

- Está diseñado con la finalidad de poder operar en ambiente con ruido y con ello realiza un rápido emparejamiento y utiliza saltos de frecuencia en la transmisión para garantizar una conexión robusta. [21]

1.4.2. Bluetooth 4.0

Tiene la capacidad de transferencia de datos de 2Mb/s hasta 26 Mb/s con la cualidad de ahorro de energía, ya que como está constantemente recolectando información necesita tener una duración larga de energía.

Otra cualidad que tiene es la compatibilidad con el NFC, cuyo chip habilita a los dispositivos para interactuar entre sí por medio de la proximidad. [22]

1.5. Modelo Vista Controlador (MVC)

La tarea principal de un programa orientado a objetos es modelar la aplicación de interés. Un modelo o simulación de una aplicación no es suficiente, sería sorda, ciega y tonta. Se necesita una manera de conectar al modelo con su usuario. El controlador es una manera de controlar, alterar posición y tamaño, entre otros y la aplicación o la acción de mostrarlo en ventana se le llama vista.

Por lo tanto toda programación orientada a objetos (POO) consiste de:

- Modelo
- Vista
- Controlador

El modelo en sí generalmente es invisible (excepto como código Java). La vista es visible en la pantalla de la computadora como una imagen gráfica, controles, gráficos, etc.

Los controles son visibles como barras de desplazamiento, botones, campos de texto, etc.

El controlador constituye las entradas para el modelo y la viste constituye las salidas del modelo.

[23]

1.6. REST

REST (Representational State Transfer) es un estilo de arquitectura para desarrollar servicios. Los servicios web que siguen este estilo deben cumplir con las siguientes premisas.

Arquitectura en Capas: Todos los servicios REST están orientados hacia la escalabilidad y un cliente REST no será capaz de distinguir entre sí está realizando una petición directamente al servidor, o se lo está devolviendo un sistema de caches intermedio o por ejemplo existe un balanceador que se encarga de redirigirlo a otro servidor.

Cliente/Servidor: Como servicios web son cliente servidor y definen un interface de comunicación entre ambos separando completamente las responsabilidades entre ambas partes.

El cliente inicia siempre una solicitud, pero ahora ésta no la produce el usuario en "bruto" hacia el servidor, sino que pasa por Javascript. El lenguaje del lado del cliente es el que solicitará al servidor un recurso al servidor.

El servidor y el cliente web se comunican en un formato de intercambio de información como puede ser *JSON*, aunque podría ser otro lenguaje como XML.

Lo importante es que el cliente no recibe HTML para renderizar, sino simplemente los datos que se han generado como respuesta. Es decir, el servidor no escribe HTML, sino únicamente genera los datos para enviarlos al cliente.

Sin estado: Son servicios web que no mantienen estado asociado al cliente. Cada petición que se realiza a ellos es completamente independiente de la siguiente. Todas las llamadas al mismo servicio serán idénticas.

Cache: El contenido de los servicios web REST ha se puede cachear de tal forma que una vez realizada la primera petición al servicio el resto puedan apoyarse en la cache si fuera necesario.

Servicios Uniformes: Todos los servicios REST compartirán una forma de invocación y métodos uniforme utilizando los métodos GET,POST,PUT ,DELETE.

[24]

1.6.1. Ventajas

a) Separación cliente/servidor

Al ser sistemas independientes puedes desarrollarlos proyectos autónomos, equipos autónomos. Al cliente le da igual cómo está hecha la API y al servidor le da igual qué vas a hacer con los datos que te ha proporcionado.

Si se necesita refactorizar uno de los dos, back o front, se puede hacer de manera separada, siempre que se mantenga la interfaz del API.

Puedes hacer varios front con un único backend.

b) Independencia de tecnologías / lenguajes

Se puede desarrollar en cualquier tipo de tecnología o lenguaje, para ajustar los tiempos de desarrollo y necesidades de un proyecto.

c) Fiabilidad, escalabilidad, flexibilidad

Solamente se debe tener correcto el nexo cliente / servidor. Se pueden hacer cambios en el servidor, lenguajes, bases de datos, etc, mientras se devuelvan los datos que corresponden.

d) REST requiere menos recursos del servidor

No mantiene el estado, no requiere memoria, se pueden atender más peticiones, no requiere escribir el HTML, por lo tanto se tiene menos procesamiento en el servidor.

1.6.2. Desventajas

- En un esquema REST se puede tener varios servidores donde unos no saben que los otros existen.
- No sabes si un usuario ha iniciado sesión en un servidor y si le has enviado ciertos datos.
- No se sabe realmente en qué servidor puede caer una solicitud. [25]

CAPÍTULO 2. ESTADO DEL ARTE.

A continuación se presentan distintos software que tienen características similares al trabajo propuesto, describiendo las cualidades con las que poseen.

2.1. En el mundo.

En la actualidad hay un museo en Bélgica que implementa la tecnología **iBeacon** y estas son sus características:

El Museo Rubens de Amberes ha confiado en los **Beacon** para ofrecer una experiencia diferente a las personas que acuden a la pinacoteca. Gracias a los Beacon ubicados en las diferentes salas o directamente junto a las obras de arte, los visitantes pueden interactuar con el propio edificio, conociendo más detalles sobre su historia o las actuaciones de ampliación que haya sufrido, y, por supuesto, con los cuadros. Visitando la exposición siguiendo un recorrido temático, particularidades de la vida de los artistas que tienen su reflejo en la técnica, los trabajos de restauración de las pinturas, etc. [26]

La tabla siguiente es un cuadro comparativo entre las diferentes aplicaciones lanzadas en el mercado con el uso de la tecnología **IBeacon** integrada en los *Smartphone*.

Software	S.O.	Tecnología de Comunicación	Descripción	Innovación
Estimote. [27]	iOS	<i>Bluetooth</i> 4.0	Se utiliza esta aplicación para cambiar de rango o el acceso incorporado en los sensores Beacon tales como un acelerómetro o sensor de temperatura.	Lista de Beacon agregada como una alternativa a los radares para visualizarlos con la posibilidad de filtrado (zona o propiedad).
Geogadgets. [28]	iOS	GPS	Permite implementar proyectos de movilidad con Beacon . Gracias esta plataforma, es posible procesar los datos obtenidos de los Beacon y la App para su análisis y generación de información de negocio que ayude en la toma de decisiones.	Mensajes de avisos y notificaciones, información ampliada sobre artículos, acciones posibles para el cliente sean métricas o analíticas, marketing y fidelización.
IBeacon Finder. [29]	Android	<i>Bluetooth</i> 4.0	Localiza todos los Beacon que estén cercanos a tu posición y en listará sus datos básicos, como por ejemplo potencia de la señal, distancia aproximada o su identificador único.	Identifica todos los Beacon que hay alrededor, puede asignar nombre al dispositivo y poder enviar los datos por e-mail.

Shopkick. [30]	IOS & Android	<i>Bluetooth</i> 4.0	Al hacer compras admitidos, escanear elementos, o incluso caminar a los minoristas destacados, podrás empezar a ganar "puntos", que pueden ser redimidos por tarjetas de regalo, mercancía gratis, etc.	Revoluciona el sistema de compras en Estados Unidos, como en tiendas Target, Best Buy, Old Navy, Macys, ToysRUs, American Eagle, Sports Authority y muchos más.
--------------------------	------------------	----------------------	---	---

Tabla 6 Aplicaciones en el mercado con IBeacon

2.2. En México.

En México no hay como tal un sistema empleando con los “**Beacon**”, solamente han sacado a la venta en Walmart el cual es el siguiente:

Producto	Características
Sistema de control remoto universal Griffin Beacon	<ul style="list-style-type: none"> • Transforma el iPhone, iPod Touch o iPad en un control remoto universal. • Armoniza discretamente con cualquier decoración. • Funciona con la aplicación del control remoto universal de Dijit, que se puede adquirir gratuitamente de App Store. • La configuración es simple e intuitiva. • Sin cables, funciona con cuatro baterías AA.

Tabla 7 Aplicaciones en México con IBeacon

[31]

CAPÍTULO 3. METODOLOGÍA.

Para el desarrollo de cualquier sistema de cómputo se deben tomar en cuenta las metodologías existentes que faciliten la obtención de los resultados establecidos en el objetivo. La Metodología para el desarrollo de un sistema de cómputo es un enfoque estructurado que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos, la metodología que usaremos es MDAM (Metodología para el Desarrollo de Aplicaciones Móviles) la cual se describe a continuación.

3.1. MDAM

La Metodología para el Desarrollo de Aplicaciones Móviles (MDAM) propuesta para el desarrollo de aplicaciones para móviles se fundamenta en la experiencia de investigaciones previas en aplicaciones móviles, la evaluación del potencial de éxito para servicios de tercera generación denominada 6 M, la ingeniería de software educativo con modelado orientado por objetos (ISE-OO), y principalmente en los valores de las metodologías ágiles.

De las metodologías ágiles se heredan los conceptos inmersos en los cuatro postulados o manifiesto ágil:

- Desarrollar software que funciona más que conseguir buena documentación
- La respuesta ante el cambio es más importante que el seguimiento de un plan.
- Colaboración con el cliente sobre negociación actual.
- Individuos e interacciones sobre procesos y herramientas.

La metodología se encuentra enmarcada en cinco fases como se muestra en la figura 5:

- I. **Análisis:** En esta fase se analizan las peticiones o requerimientos para la cual se desarrolla el servicio móvil “cliente”. El propósito es definir las características del mundo o entorno de la aplicación. Se realizan tres tareas:
 - Obtener requerimientos.
 - Clasificar requerimientos.
 - Personalizar el servicio.
- II. **Diseño:** Se debe plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa a integrar los aspectos técnicos, funcionales, sociales y económicos. A esta fase se retorna si no se obtiene lo deseado en la etapa de pruebas.
En esta fase se realizan cuatro actividades:
 - Definir el escenario.
 - Estructurar el software.
 - Definir tiempos
 - Asignar recursos.
- III. **Desarrollo:** En esta fase, el objetivo es implementar el diseño del producto de software. En esta etapa se realizan las siguientes actividades:

- Codificar.
- Pruebas unitarias.
- Documentación del código.
- Codificar ayudas.

IV. **Pruebas de funcionamiento:** En esta fase se debe verificar el funcionamiento de la aplicación en diferentes escenarios y condiciones, para esto se realizan las siguientes tareas:

- Emulación y simulación.
- Trabajo en dispositivos reales.
- Análisis de las 6 Ms

V. **Entrega:** Terminada la depuración de la aplicación y atendidos todos los requerimientos de última hora se da por finalizada la aplicación y se procede a la entrega del ejecutable, el código fuente, la documentación y el manual del sistema.

- Manuales
- Distribución

[32]

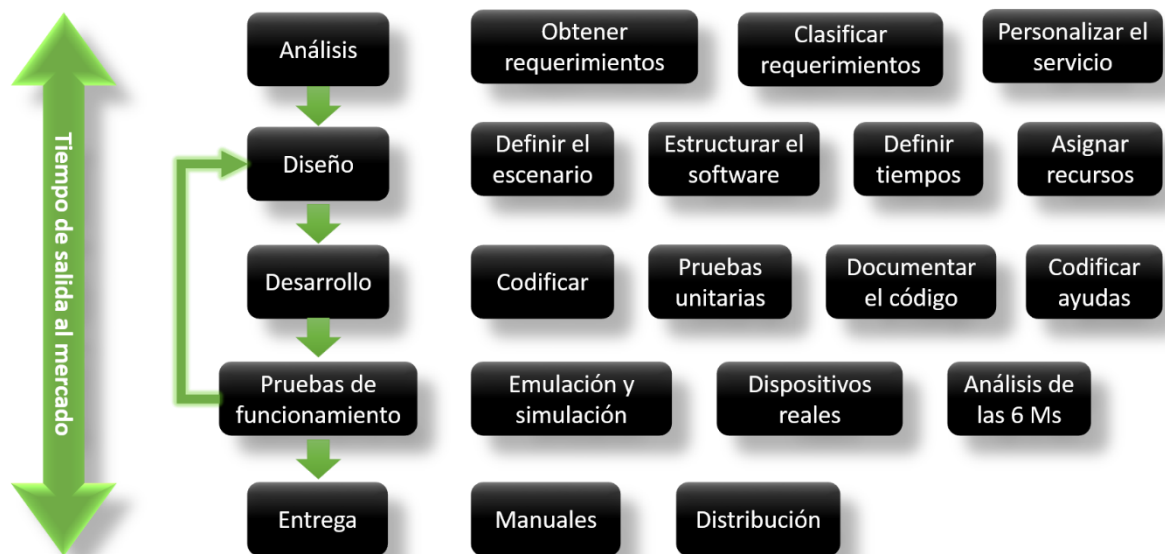


Figura 5 Etapas de la metodología para el desarrollo de aplicaciones móviles

CAPÍTULO 4. ANÁLISIS.

4.1. Descripción General

Este trabajo terminal consiste en el intercambio de información para dar asistencia a los usuarios y así tener su propia guía en un museo, implementando la tecnología **iBeacon** para obtener información más detallada de las exposiciones y material que se está exhibiendo (dependiendo de su ubicación en el museo), estas se muestran en una aplicación móvil con unas plantillas ya definidas (texto, imagen, video).

También se cuenta con un sistema web en el cual se puede administrar las exhibiciones y material expuesto en el museo.

El trabajo terminal consta de un servidor y una aplicación móvil (cliente).

a) El servidor cuenta con lo siguiente:

Sistema gestor de base de datos (SGBD) que contiene la Base de Datos, poblada con la información previamente recolectada.

Software que cumpla con las funciones de solicitud a la base de datos, respuesta a los clientes y las operaciones relacionadas con la gestión de la información almacenada.

Un servidor web que permite la conexión con la aplicación móvil por medio de la API.

Un administrador tendrá acceso al sistema web para la administración de exhibiciones y material expuesto del museo.

b) La aplicación móvil consta de lo siguiente:

La aplicación móvil detecta al **Beacon** con base a distancias previamente configuradas, establece una conexión para obtener los datos básicos establecidos y posteriormente envía peticiones al servidor por medio de la API. Obtiene la información del servidor y se la muestra al cliente por medio de plantillas, las cuales pueden ser: texto, imágenes y video.

Se establece la plataforma Android para el desarrollo de la aplicación cliente dado que es la plataforma de mayor difusión entre la población

4.2. Alcances

El proceso para la generación de un sistema interactivo de guía para museos basado en **iBeacon**, se divide en 3 módulos principales:

- Aplicación móvil: Sera la encargada de recaudar la información necesaria del Beacon que se localizó para posteriormente enviarlos al servidor web a través de internet y mostrar la información que llegue de respuesta.
La API para la comunicación de la aplicación móvil con los Beacon, se proporciona dependiendo los múltiples fabricantes que existen en el mercado. Además las funcionalidades básicas del API se encuentran en todos los sistemas a partir del Android 4.3.
- Interfaz de programación de aplicaciones (API): Este módulo es el encargado de recibir y procesar los datos enviados desde la aplicación y generar la respuesta. Esta conexión se realiza a través de Internet y contará con las siguientes características:

- Servidor Web: Este módulo es el encargado de administrar toda la información que se pueda almacenar y administrar de los museos.

La información se mostrará de acuerdo a plantillas definidas basadas en texto, imágenes y video.

4.2.1. Productos esperados

Al término del presente trabajo terminal, el equipo se compromete a entregar:

- Aplicación del servidor (código fuente).
- Aplicación móvil (código fuente).
- Documentación de análisis y diseño.
- Pruebas de funcionamiento.
- Manual de usuario.

4.3. Análisis de requerimientos

En la siguiente sección se describen los requerimientos del sistema, los cuales nos permite especificar las características operacionales del sistema y sus restricciones.

4.3.1. Requisitos funcionales

Requerimientos de la aplicación móvil

RF1 Mostrará información general del museo.

- Nombre
- Descripción
- Imagen
- Imagen de la distribución.
- Lista de exhibiciones

RF2 Mostrará información de las exhibiciones (salas, exposiciones, colecciones) en el museo.

RF3 Mostrará información de los materiales y exhibiciones expuestas en el museo dependiendo su proximidad con estos.

RF4 Proporcionar una lista de los materiales exhibiciones expuestas de los cuales se puede obtener información acorde a la ubicación del dispositivo móvil.

RF5 La lista contiene nombre, descripción corta de los materiales y nombre de la exhibición a la que pertenecen.

RF6 Visualizar la información más detallada del material expuesto al encontrarse cerca de éste.

RF7 La información detallada del material expuesto se mostrará con base en las siguientes plantillas:

- Texto
- Imagen
- Video

RF8 Configurar la aplicación móvil para habilitar o deshabilitar las notificaciones del material cercano (*Immediate*) y material a distancia media (*Near*).

RF9: Acceder a la lista de materiales y exhibiciones mediante la notificación desplegada.

RF10 Configurar la aplicación móvil para habilitar o deshabilitar la pre –descarga de información a distancia *Immediate*.

RF11 Descargar archivos de los materiales expuesto si se encuentran disponibles.

RF12 Mostrar un listado de los posibles archivos a descargar.

Requerimientos del Sistema web

RF13 El sistema web solo puede ser manipulado por un administrador.

RF14 El sistema web permite el acceso al administrador por usuario y contraseña.

RF15 Administrar la información de los materiales expuestos del museo:

- Alta
- Baja
- Actualización
- Lectura

RF16 La administración de la información de los materiales expuesto está sujeta a las plantillas disponibles.

RF17 Administrar la información de las salas y exhibiciones del museo

- Alta
- Baja
- Actualización
- Lectura

RF18 Administrar la información de los tipos de exhibiciones.

- Alta
- Baja
- Actualización
- Lectura

RF19 Administrar los archivos disponibles para descarga:

- Alta
- Baja

4.3.2. Requisitos no funcionales

Requerimientos del dispositivo móvil

RNF1 Requiere la versión de Android 4.3 o superior.

RNF2 Contar con *Bluetooth* 4 o superior

RNF3 Contar con una conexión a internet.

Requerimientos del servidor

RNF4 El servidor debe ser un servidor Linux.

RNF5 El sistema web debe estar montado en un servidor HTTP Apache.

RNF6 El sistema de gestión de base de datos es MySQL.

RNF7 El lenguaje de programación a utilizar es PHP 5 o superior.

RNF8 Se utiliza el CMS Modx revolution para el desarrollo del sistema web.

Requerimientos de Beacon

RNF9 El proveedor de los **Beacon** es Kontack.io

RNF10 La configuración básica de los **Beacon** es a través de la aplicación del proveedor.

Requerimientos de la aplicación móvil

RNF11 La aplicación móvil es desarrollada para Android en el lenguaje nativo *Java*.

RNF12 Las distancias utilizadas para mostrar la información son las establecidas por la tecnología **IBeacon**: *Immediate, Near y Far*.

RNF13 La lista de los materiales expuesto de los cuales se obtiene información es a partir de la proximidad Far a los materiales expuesto.

RNF14 Se inicia la descarga de información cuando encuentra la proximidad Immediate a un material expuesto

RNF15 Se inicia con la descarga cuando encuentra la proximidad Near a un material expuesto siempre y cuando este activada esta función.

RNF16 La aplicación móvil no registra datos personales de los usuarios.

RNF17 La aplicación móvil establece una conexión con el Beacon al estar dentro del rango de proximidades.

RNF18 La plantilla de Video cuenta con:

- Título
- Descripción
- Video

RNF19 La plantilla de imágenes cuenta con:

- Título
- Descripción
- Imágenes
- Autor

RNF20 La plantilla de texto cuenta con:

- Título
- Texto
- Autor

Requerimientos API

RNF21 La conexión de la aplicación móvil con el servidor se realiza a través de un API tipo REST.

RNF22 La conexión de la aplicación móvil con el servidor solo se realiza con credenciales autorizadas.

RNF23 El tiempo de respuesta de las peticiones depende de la calidad y velocidad de la conexión y el tamaño de respuesta de la petición

Requerimientos del Sistema Web

RNF24 El patrón de arquitectura de software es el modelo-vista-controlador (MVC)

4.4. Análisis de riesgos

Con base al marco de riesgos de TI [33] se clasifican en las tablas 8 y 9 los riesgos y consecuencias del desarrollo del sistema y del sistema.

1. Tipo respecto al sistema
 - Interno: Se genera dentro del sistema.
 - Externo: Se genera fuera del sistema.
2. Probabilidad de ocurrencia
 - Raro: La probabilidad de que ocurra es muy baja.
 - Poco probable: La probabilidad de que ocurra es baja.
 - Posible: La posibilidad de que ocurra es mediana.
 - Común: La posibilidad de que ocurra es moderadamente alta.
 - Altamente probable: La posibilidad de que ocurra es elevada.
3. Nivel de riesgo en el sistema
 - Bajo: Cuando ocurra, no afectará o afectará muy poco al sistema.
 - Moderado: Cuando ocurra, afectará poco al sistema.
 - Alto: Cuando ocurra, tendrá un efecto considerable al sistema.
 - Extremo: Cuando ocurra, afectará mucho al sistema o lo inutiliza.
4. Respuesta al riesgo
 - Evitar riesgo: Preferiblemente evitar acciones que impliquen el riesgo.
 - Reducción/mitigación de riesgos: Tomar medidas que minimicen el riesgo.
 - Riesgo compartido/transferencia: Distribuir las consecuencias del riesgo entre varios entes.
 - Aceptación del riesgo: Considerar que el riesgo ocurrirá sin importar las medidas aplicadas.

4.4.1. Riesgos del desarrollo del sistema

Nombre	Tipo	Descripción	Ocurrencia	Riesgo	Respuesta
Riesgo de desempeño	Interno	El Desarrollo del proyecto no proporcione los resultados esperados	Poco Probable. Se tiene una estructura de análisis y diseño ya definido.	Extremo. Si no se sigue lo planificado, el sistema no podrá ser desarrollado correctamente.	Evitar riesgo. Llevar acabo el análisis y diseño propuesto, y si es necesario se puede reestructurar.

Riesgo de falla componente de hardware	Interno	Algún componente de hardware deja de funcionar correctamente.	Posible. La probabilidad es medianamente posible, ya que se considera varias piezas de hardware.	Extremo. Si hay una falla en el hardware, sea servidor, Beacon o móvil, no se podrá disponer de la información.	Evitar riesgo. Considerar tener un mantenimiento programado cada cierto tiempo, para checar el estatus del hardware.
Riesgo Dispositivos Beacons no lleguen a tiempo.	Externo	Los dispositivos Beacon serán adquiridos en el extranjero, lo cual requiere de tiempo para su llegada al país.	Poco Probable. Se hará el encargo de los dispositivos en el mes de diciembre.	Extremo. La demora de estos dispositivos afectará el desarrollo de la programación y pruebas.	Reducción. Considerar el pedido de los dispositivos con un tiempo anticipado.
Riesgo de cambio de tecnología.	Externo	Alguna de las tecnologías sobre las que se esté desarrollando se sustituya por una nueva.	Raro. La tecnología elegida fue escogida a través de varios estudios realizados por el equipo y se adapta al sistema para desarrollarlo.	Extremo. El cambio de tecnología tomaría un tiempo considerable para la nueva estructura y desarrollo.	Riesgo Compartido. Se cuenta con una planeación ya estructurada, que funciona para esta tecnología.
Riesgo de baja de algún integrante del equipo.	Interno	Un integrante del equipo por cualquier motivo (personal, académico o desinterés) salga del proyecto.	Raro. Cada integrante del equipo cuenta con un alto nivel de compromiso para llevar a cabo este proyecto.	Extremo. La ausencia de un integrante causaría retraso ya dejaría trabajo y tareas inconclusas.	Evitar riesgo. Tener una sana convivencia y dar apoyo en todo lo necesario entre cada integrante para seguir teniendo el compromiso del proyecto.
Riesgo de falta de comunicación en el equipo.	Interno	La falta de comunicación en el equipo puede retrasar las tareas o crear disgusto entre los miembros.	Poco Probable. El equipo tiene reuniones frecuentes para el desarrollo del proyecto.	Extremo. Al no haber comunicación entre el equipo, no se podrá llegar a un fin común para el desarrollo óptimo del proyecto.	Evitar riesgo. Tener reuniones diarias para tocar puntos sobre el proyecto.

Riesgo de contingencia natural.	Externo	Cualquier tipo de desastre natural puede presentarse durante el desarrollo del sistema.	Posible. La naturaleza es impredecible.	Extremo. Al llegar a afectar la zona de donde se encuentre el equipo de desarrollo, se podrían varias fallas.	Reducción. Poner medidas de seguridad para evitar el mayor impacto posible cuando ocurra un suceso natural.
---------------------------------	----------------	---	---	---	---

Tabla 8 Análisis de riesgos del desarrollo del sistema

4.4.2. Riesgos del sistema

Nombre	Tipo	Descripción	Ocurrencia	Riesgo	Respuesta
Sobrecarga de Servidor	Interno	Sobrecarga de peticiones que provoca la caída del servidor	Común Pueden generarse un número muy elevado de peticiones al servidor.	Extremo Existe el riesgo de que el servidor colapse por el elevado número de Peticiones.	Evitar riesgo Considerar elevar el número de recursos para poder sostener un número de peticiones simultaneas
Falla en servicio de WiFi	Externo	Los dispositivos móviles se quedan sin Internet	Poco probable Se puede generar una falla en el servicio de WiFi	Extremo Si no se cuenta con servicio de WiFi no se podrá descargar la información.	Reducción Considerar que la conexión debe de ser estable
Falta de detección de proximidad de los dispositivos Beacon	Interno	Los dispositivos móviles cuentan con las características mínimas o tienen fallas.	Raro La probabilidad es muy baja	Extremo Si no se detecta algún Beacon el sistema no podrá obtener información.	Reducción Se requieren las características mínimas del dispositivo móvil.
Batería del Beacon se agota	Interno	La Batería del Beacon esta baja o se terminó	Raro La batería de los Beacon dura hasta 2 años.	Extremo No se detectaría ningún Beacon y no podrá obtener información.	Evitar riesgo Considerar el cambio de baterías oportunamente,

Tabla 9 Análisis de riesgos del sistema

4.5. Estudio de factibilidad

En el estudio de factibilidad se determina la infraestructura tecnológica y la capacidad técnica que implica la implantación del trabajo terminal.

4.5.1. Factibilidad técnica

4.5.1.1. Herramientas para el desarrollo

- Apache Http Server

El servidor *HTTP Apache* es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), *Microsoft Windows*, *Macintosh* y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Utilizado para enviar páginas web estáticas y dinámicas (principalmente) en la “WWW” (World Wide Web).

Características

- Multiplataforma: *Linux, Unix, Windows, Mac OS*.
- Modular: El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad. Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Costo: Está disponible para cualquier persona para descargar sin costo alguno.
- Código abierto (*Open source*): Puede ser modificado para adaptarse a sus necesidades individuales.

- Ubuntu Server

Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre. Está orientado al usuario novel y promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia de usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto.

Soporta oficialmente dos arquitecturas de hardware en computadoras personales y servidores: 32-bit (x86) y 64-bit (x86_64)⁴⁴ Sin embargo, extraoficialmente, Ubuntu ha sido portado a más arquitecturas: ARM, PowerPC, SPARC e IA-64.

Ubuntu Server 14.04

Tipo de instalación	CPU	RAM	Espacio en disco duro	
			Sistema base	Todas las tareas instaladas
Server (Standard)	1 giga Hertz	512 megabytes	1 gigabyte	1.75 gigabytes
Server (Minimal)	300 mega Hertz	192 megabytes	700 megabytes	1.4 gigabytes

Tabla 10 Requisitos mínimos recomendados Ubuntu server

- PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Características

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

- SGBD MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario

Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

- IDE de desarrollo Android Studio

Costo: Gratuito

Herramienta recomendada por Android para el desarrollo de aplicaciones de esta misma plataforma. Con ella se tienen herramientas de depuración, administración de distintos simuladores de dispositivos Android y la capacidad de manejar proyectos.

- Lenguajes de desarrollo para cliente móvil

Costo: Gratuito

Para el desarrollo de las aplicaciones en Android se utiliza el lenguaje de marcado XML junto con el lenguaje Java.

- IDE de desarrollo Aptana Studio

Costo: Gratuito

IDE de desarrollo web de código abierto que provee soporte para lenguajes como: Php, Python, Ruby, CSS, Ajax, HTML y Adobe AIR.

- CMS Modx Revolution.

MODx es un sistema de administración de contenidos y framework o CMF (Content Manager Framework) de código abierto, desarrollado en PHP. Su API y modelo de eventos le otorga una amplia capacidad de adaptación y personalización, haciéndolo muy flexible en la creación de sitios Web, a diferencia de otros CMS rígidos basados en bloques. Su alto soporte de estándares xhtml/css lo hace ideal para diseñadores Web.

MODx incorpora funcionalidades Web 2.0 a través de su API permitiendo búsquedas en "vivo", efectos Web y comunicaciones Ajax.

4.5.1.2. Herramientas para pruebas e implementación

- Simulador Android

Se encuentra incorporado en el IDE de Android Studio, este cuenta con personalizar las características de configuración de los equipos que simulara.

- Dispositivo Móvil Android Moto G

En este dispositivo se mostrara las demo del proyecto y así probar la funcionabilidad del cliente en el sistema y obtener la experiencia del usuario final.

4.5.1.3. Requisitos de hardware y software

Equipo Cliente.

Recurso	Requerimiento Mínimo
Procesador	1 GHz
RAM	1 Gb
Sistema Operativo	Android 4.3
Otros	Pantalla 3.5", WiFi/ 3G , <i>Bluetooth</i> 4.0

Tabla 11 Requisitos del cliente

Servidor

Características	GoDaddy Plan Inicial
Sitios web	1
Espacio en disco	30 GB
Ancho de banda mensual	Ilimitados
Usuarios de FTP	50
Bases de datos MySQL	1 x 1 GB
Copia de seguridad	Si
Restauración de bases de datos	Si
Acceso directo a las bases de datos	Si
Administración de DNS	Si
Sistema operativo	Linux
Precio mensual	\$38.99

Tabla 12 Requisitos del servidor

4.5.2. Factibilidad operativa

Se realizará un pronóstico del impacto que se dará a la sociedad de las personas que visitan un museo en la Ciudad de México.

Hemos determinado que un problema actual en los museos es la desinformación de los visitantes, ya que en ocasiones se encuentra desactualizadas o incompletas e incluso no se cuenta con el personal necesario para guiarte o darte esa información faltante. Por lo cual, este sistema es propuesto para facilitar el acceso a una información más completa y personal, con ello tendremos los siguientes beneficios:

- Información precisa y detallada en algunas ocasiones de los materiales expuestos.
- Una guía y asistencia turística personal en su dispositivo móvil.
- Tener una noción de cercanía hacia el material expuesto a través de las alertas dadas por la aplicación móvil.

4.5.3. Factibilidad económica

Para determinar el estudio de factibilidad económica se tomó en cuenta:

- Costo de desarrollo

Con base a la Encuesta de Salarios de Software Guru [34], tomamos el salario promedio mensual para realizar los cálculos dentro de 10 meses por cada uno del personal que se

requiere para el desarrollo del sistema, cabe mencionar que las cifras y estadísticas que se manejan se generaron en base a una muestra de 1,550 respuestas obtenidas durante noviembre de 2014 en una encuesta abierta realizada por medio de internet.

A continuación se muestra la tabla el promedio del salario mensual y se realiza el cálculo de pago por hora tomando como referencia una jornada de trabajo de 8 horas diarias.

Rol	Salario mensual	Pago por hora
Análisis de Requerimientos	\$28,431	\$118.46
Arquitectura de sistemas	\$34,289	\$142.87
Programación de back end	\$24,980	\$104.08
Programación de front end	\$24,880	\$103.66

Tabla 13 Promedio del salario mensual

Acorde a nuestro cronograma se realiza la siguiente tabla, la cual nos muestra una relación de los días trabajados durante un periodo de tiempo y el costo por hora tomando en cuenta 2 horas de trabajo al día.

Cantidad	Personal	Periodo (meses)	Días trabajados aprox. 27 por mes	Tiempo aproximado en (horas)	Pago por persona por hora	Total
3	Análisis de Requerimientos	2	54	108	\$118.46	\$38,381.04
3	Arquitectura de sistemas	2	54	108	\$142.87	\$46,289.88
1	Programación de back end	4	108	216	\$104.08	\$22,481.28
2	Programacion de front end	4	108	216	\$103.66	\$44,781.12
Total:						\$151,933.32

Tabla 14 Costo del desarrollo del sistema

- Software

El software de desarrollo que utilizaremos es gratuito por lo que no implica algún costo.

- Servicios

En la siguiente tabla se muestran los servicios necesarios para la elaboración del trabajo terminal:

Servicio	Costo mensual	Costo total (10 meses)
Luz	\$22.23	\$222.37
Web Hosting (GoDaddy)	\$38.99	\$389.90
Internet (Telmex)	\$389.00	\$3,890.00
Total:	\$450.22	\$4,522.27

Tabla 15 Servicios

- Costo del hardware

A continuación se muestra el costo del hardware que se utilizara para el desarrollo del sistema.

Equipo/ dispositivo	Características principales	Precio total
Dispositivo Android Moto G 2da Generacion	-Pantalla 5.0 pulgadas -8GB memoria interna, 1GB RAM -Procesador Qualcomm MSM8226 Snapdragon 400 quad-core 1.2 GHz, GPU Adreno 305 -Android OS 5.0.2 Lollipop -Bluetooth v4.0 A2DP -Wi-Fi 802.11 b/g/n -3G HSDPA [35]	\$2,990
Acer Aspire V5 431-2420	-Procesador Intel Celeron 1,4 GHz -Memoria RAM 4 GB -Disco Duro 500GB -S.O Windows 10/Ubuntu 14.04 LTS	\$4,000
Acer Aspire One d255e-13894	-Procesador Intel Atom 1.66 GHz -Memoria RAM 1.0 GB -Disco Duro 120GB	\$2,500
Samsung Galaxy tab s2	-32GB memoria interna, 3GB RAM -Procesador octa-core (4x1.9GHz + 4x1.3GHz) -Android OS, v5.0 Lollipop -Bluetooth v4.1 -Wi-Fi 802.11 a/b/g/n/ac [36]	\$ 9,499
Kontakt.io Smart Beacon (3 Dispositivos)	-Batería de alta capacidad con 2 años vida útil -Funciona en ambos formatos IBeacon y Eddystone [37]	\$1,680
Total:		\$20,669

Tabla 16 Costo del hardware

Costo total del sistema

Concepto	Costos totales
Desarrollo	\$151,933.32
Servicios	\$4,522.27
Hardware	\$20,669
Total	\$177,124.59

Tabla 17 Costo total del sistema

4.6. Definición de roles

Se describen los diferentes roles y sus actividades que tienen dentro del sistema.

Rol	Descripción	Actividades
Usuario	Actor final que a través de un dispositivo móvil podrá acceder a los servicios del sistema mediante una aplicación.	Visualización de la información del museo. Visualización de la información mediante plantillas precargadas por el administrador del sistema. Descargar archivos si es que cuentan con disponibilidad de descarga.
Administrador	Actor final que se encargara de gestionar el contenido del sistema.	Administrar la información del museo. Administrar las exhibiciones y sus tipos. Añadir plantillas a cada material expuesto para su visualización en la aplicación móvil. Administrar la información del material expuesto. Administrar los archivos de descarga para la aplicación móvil.

Tabla 18 Definición de roles

4.7. Propuesta de proceso general

El siguiente diagrama describe el conjunto de actividades que realiza el sistema.

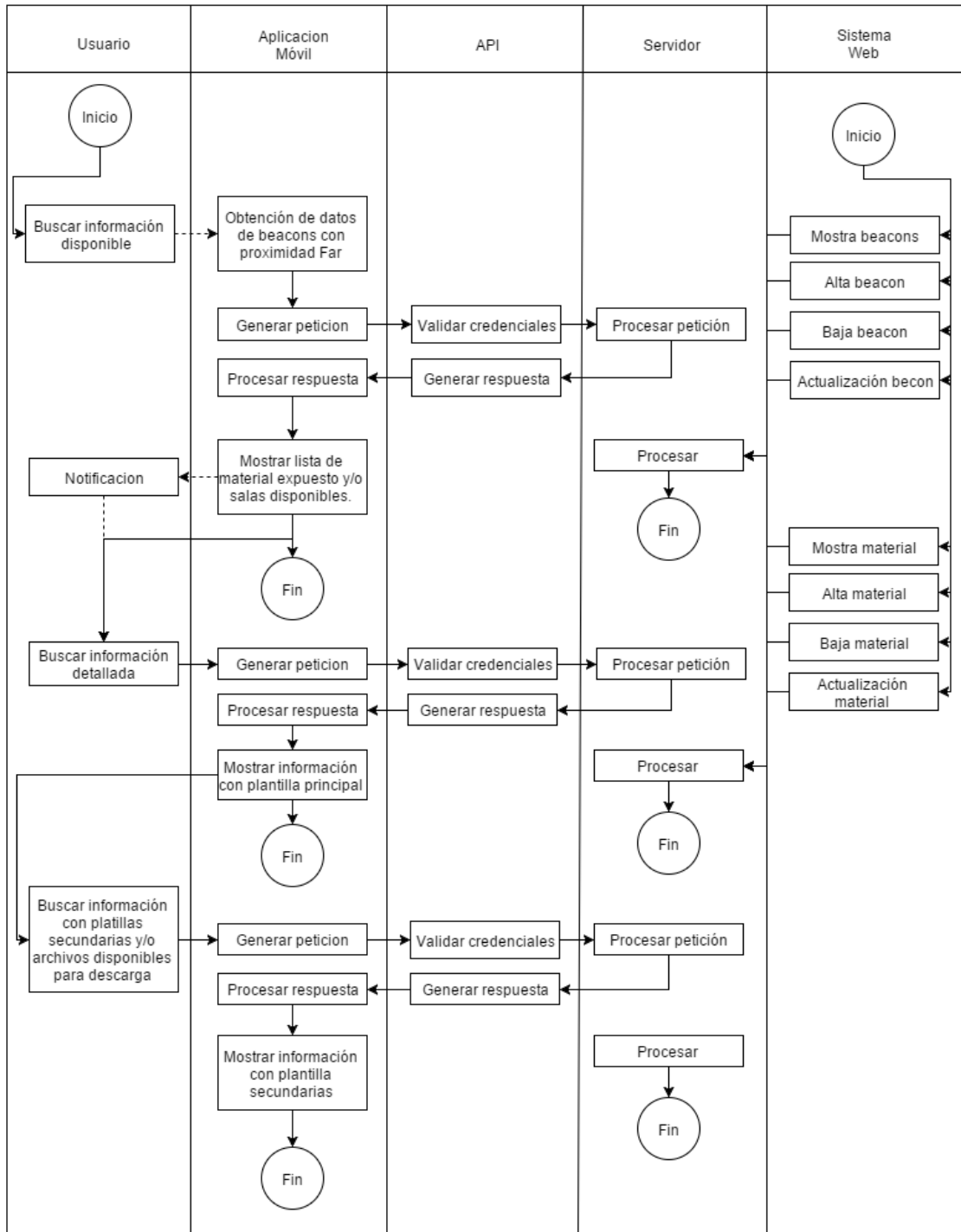


Figura 6 Diagrama de proceso general

CAPÍTULO 5. DISEÑO

5.1. Diagrama general del sistema

El siguiente diagrama describe el comportamiento del sistema.

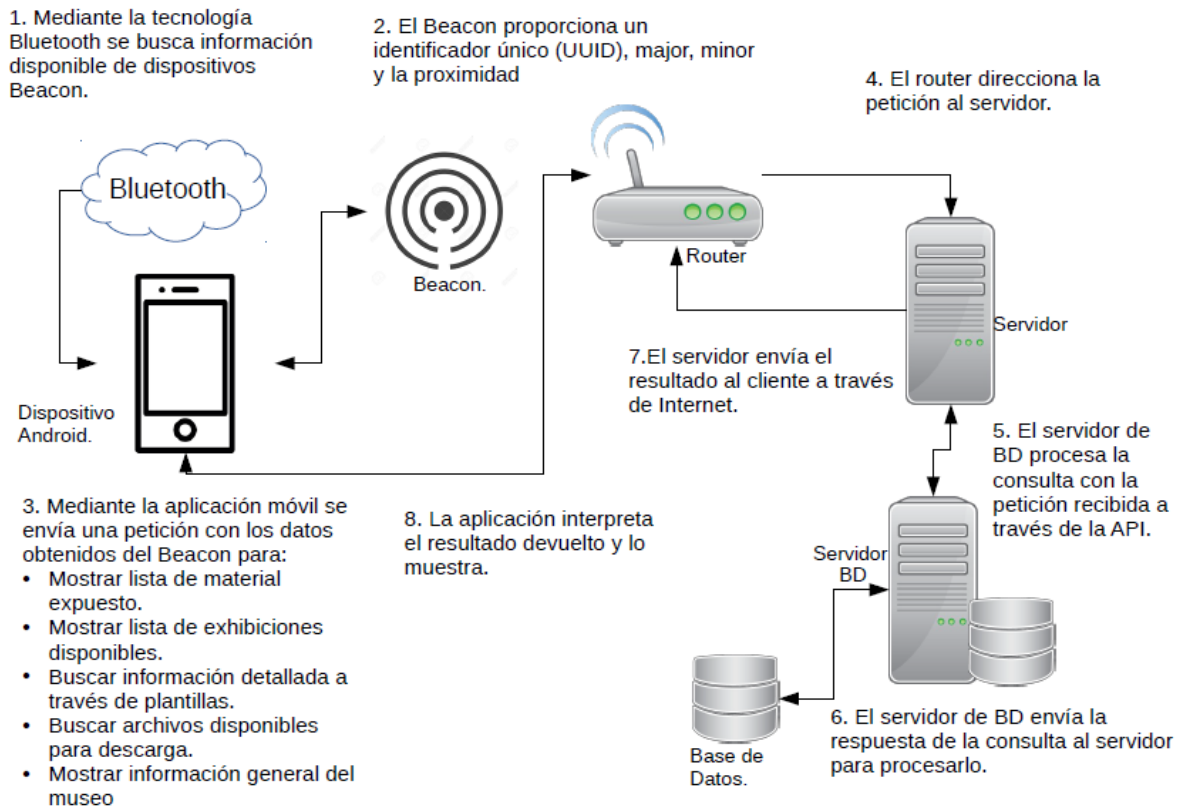


Figura 7 Diagrama general del sistema

5.2. Diagrama a bloques

El Diagrama de Bloques mostrado en la Figura 8, está dividido en dos: La aplicación móvil y el sistema web.

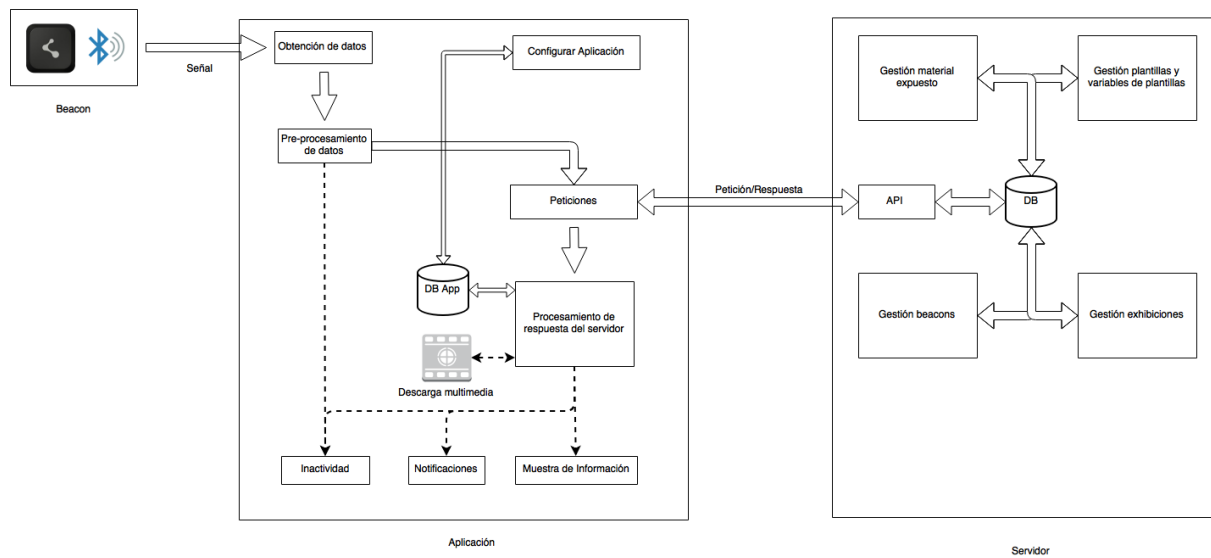


Figura 8 Diagrama a bloques del sistema

5.3. Diseño API

- i. Las llaves para la aceptación de la petición estarán en el header de la petición.

Parámetros Header

- secret_key
- customer_key

- ii. La API contesta a las peticiones en formato *JSON*.
- iii. La API tiene las siguientes direcciones con sus respectivas descripciones:

POST museum/

Regresa la información de un museo en específico.

URL

<http://dev.politicaenlinea.com/api/museum/>

Información de Petición

- Formato de respuesta: JSON
- Requiere autenticación: Si

Parámetros

- secret_key
- customer_key
- uuid

POST material/

Regresa la información de un material en específico.

La información templates dependerá de la proximidad enviada; Immediate =1, Near = 2, Far = 3.

URL

<http://dev.politicaenlinea.com/api/material/>

Información de Petición

- Formato de respuesta: JSON
- Requiere autenticación: Si

Parámetros

- secret_key
- customer_key
- Uuid
- Major
- Minor
- proximity

5.4. Glosario de términos

Beacon: Es un pequeño dispositivo que emite una señal en la onda corta de la tecnología *Bluetooth* Low Energy.

Servidor: Equipo que cuenta con un software que permite alojar la aplicación del sistema.

Usuario: Rol que utiliza el software en un dispositivo Android para consumir los servicios del servidor.

Administrador: Rol que se encarga de administrar la información del sistema utilizado el software que se ejecuta en el servidor.

5.5. Reglas de negocio

Las reglas de negocio describen las normas, operaciones, definiciones y restricciones sobre las cuales se llevará a cabo el desarrollo de sistema con el objetivo de normalizar y hacer claros los procesos.

RN1 Ingreso a la aplicación móvil

Se podrá ingresar a la aplicación móvil sin necesidad de un nombre de usuario o contraseña, es suficiente contar con los requerimientos descritos anteriormente.

RN2 Carga de videos al servidor

Los videos deben de ser en formato .mp4 y con un tamaño máximo de 30MB.

RN3 Carga de imágenes al servidor

Las imágenes deben ser en formato .png, .jpg, .jpeg y con un tamaño máximo de 3MB.

RN4 Permisos

El usuario debe proporcionar a la aplicación móvil permisos de acceso a la comunicación de red y almacenamiento.

RN5 Campos obligatorios

Se deben ingresar datos a todos los campos obligatorios en los formularios para poder ser validados.

RN6 Plantillas

Las plantillas están definidas por defecto y solo el administrador puede editarlas.

RN7 Gestionar Material

El administrador es el único usuario que puede gestionar el material.

RN8 Gestionar exhibición

El administrador es el único usuario que puede gestionar las exhibiciones.

RN9 Listado material expuesto y exhibiciones

La lista de material expuesto y exhibiciones debe contar con colores Verde, Amarillo y Rojo para indicar las distancias a las que el usuario se encuentra (*Near, Immediate y Far*) respectivamente.

5.6. Casos de uso

5.6.1. Casos de uso general

El siguiente diagrama muestra de forma general la interacción del usuario y el administrador con el sistema.

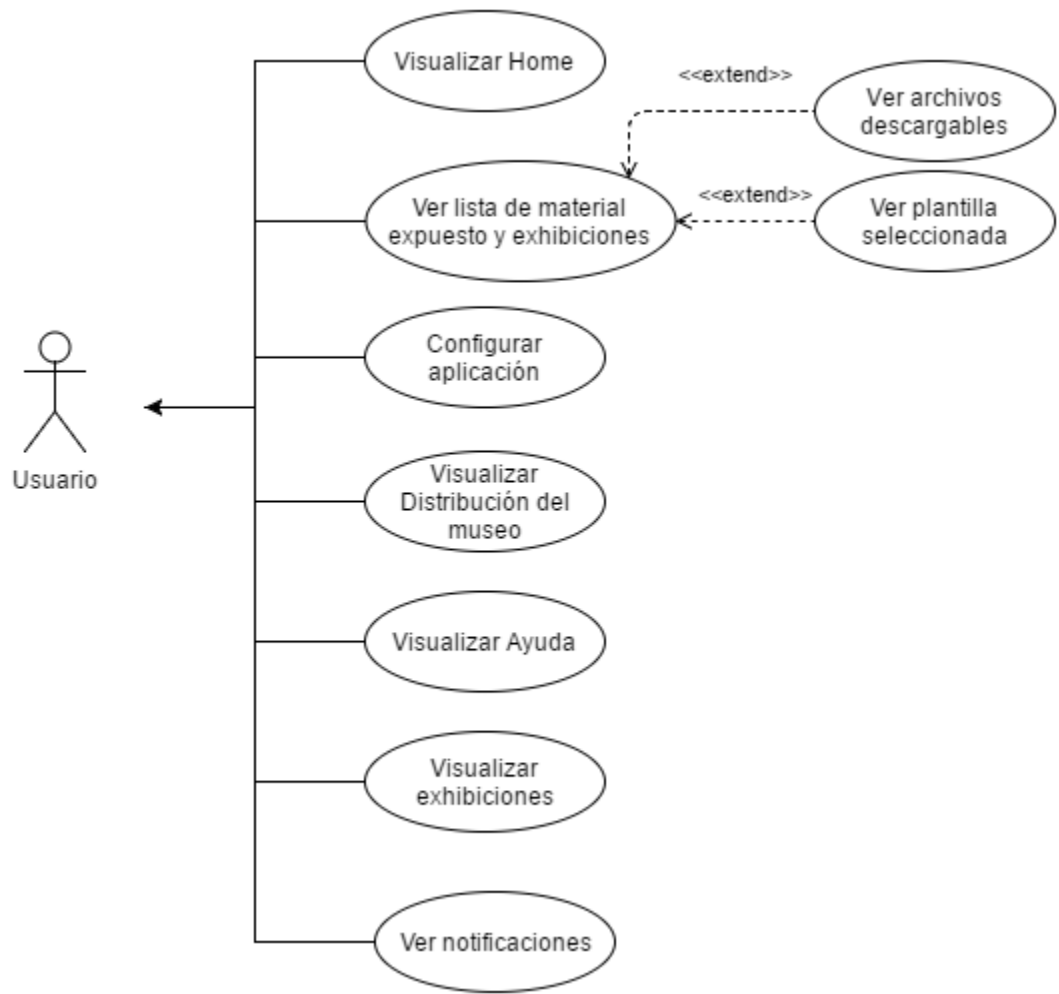


Figura 9 Diagrama general casos de uso de usuario

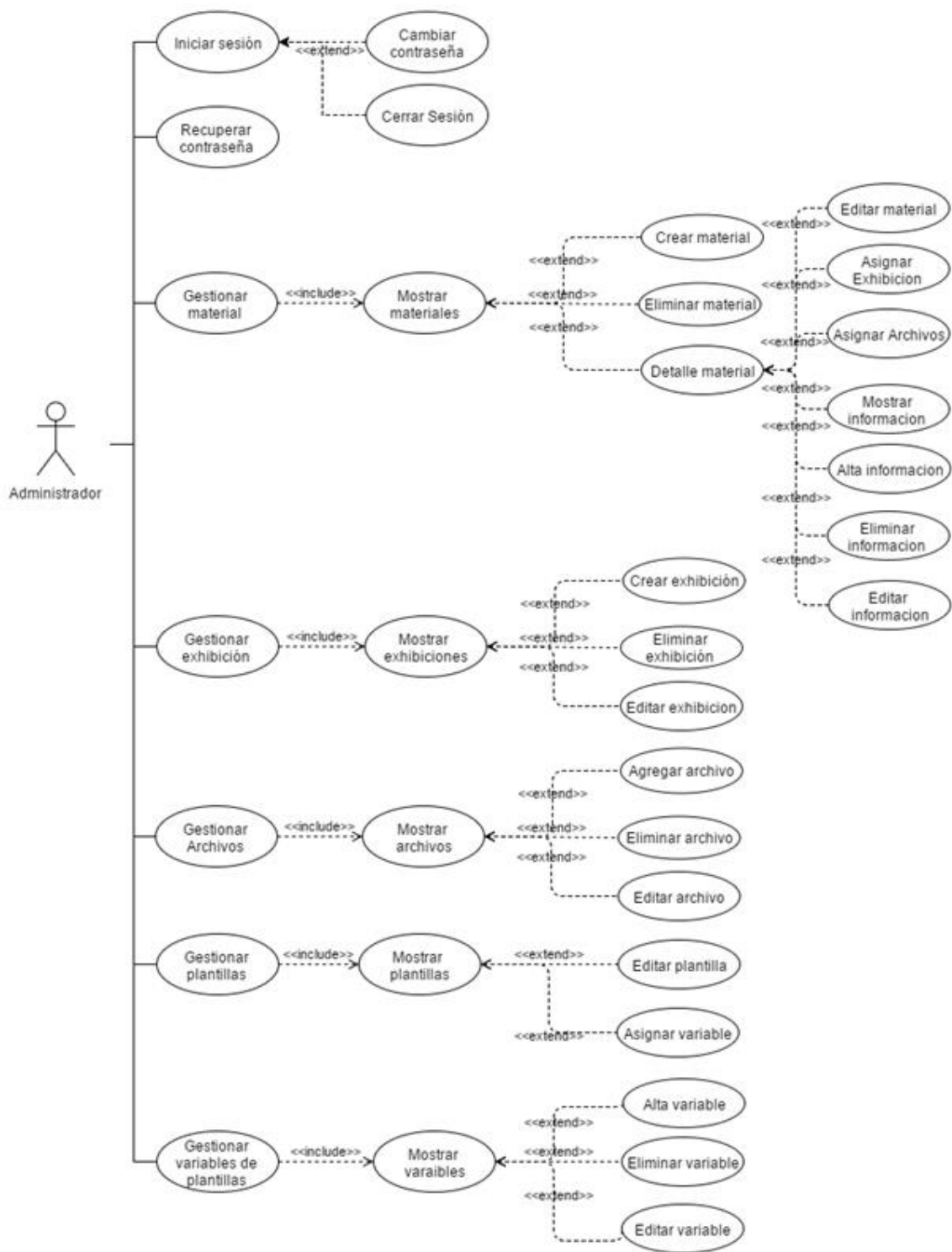


Figura 10 Diagrama general casos de uso administrador

5.6.2. CU1: Visualizar home

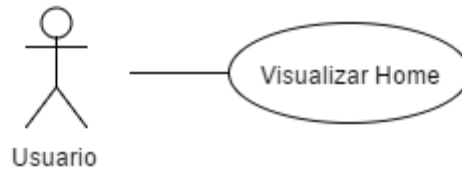


Figura 11 CU1: Visualizar home

Nombre:	CU1: “Visualizar home”
Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015
Propósito:	Ver información general del museo.
Actores:	Usuario.
Salidas:	Nombre, descripción e imagen del museo
Precondiciones:	El usuario debe encender el <i>bluetooth</i> y conectarse a una red WiFi.
Postcondiciones:	El usuario podrá acceder a la información
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Compruebe la conexión a internet” • “No hay conexión a internet”
Observaciones:	La ejecución de este caso de uso se realiza desde la pantalla principal de la aplicación móvil.

Tabla 19 CU1: Visualizar home

- Trayectorias del caso de uso

Trayectoria Principal

1. La aplicación móvil verifica que tenga conexión a internet [Trayectoria A]
2. La aplicación móvil verifica que el usuario este dentro del rango de algún **Beacon** [Trayectoria B]
3. La aplicación móvil muestra la información del museo (nombre, descripción e imagen)

--Fin de Trayectoria Principal.

Trayectoria alternativa A

1. La aplicación no detecta conexión a internet
2. La aplicación muestra el mensaje “No hay conexión a internet”
3. Volver al paso 1 de trayectoria principal.

--Fin de Trayectoria Alternativa A.

Trayectoria alternativa B

1. La aplicación no detecta ningún **Beacon**
2. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa B.

- Diseño de pantallas del caso de uso

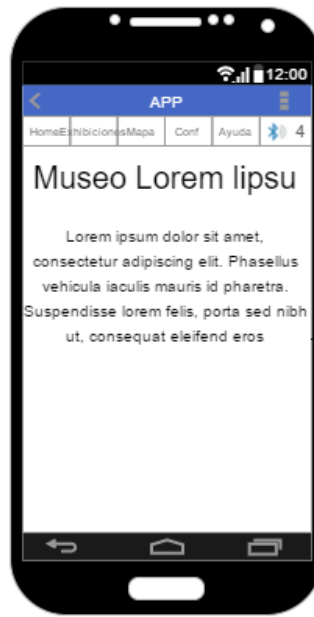


Figura 12 Pantalla home

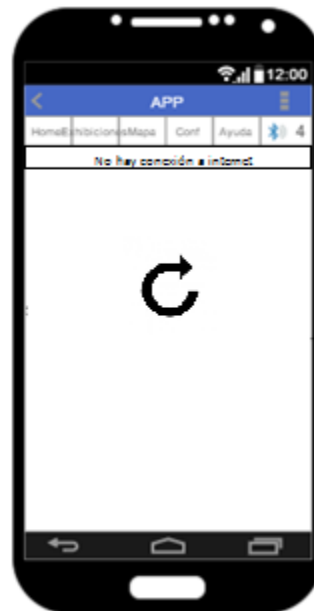


Figura 13 Pantalla no hay conexión a internet

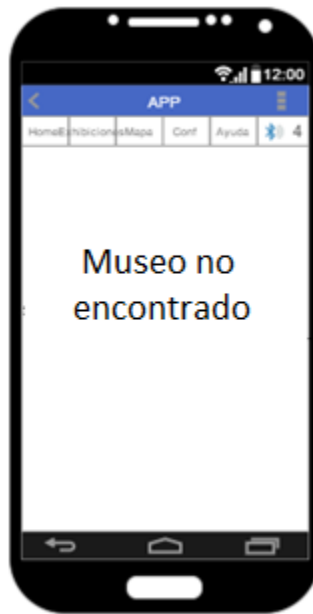


Figura 14 Pantalla Museo no encontrado

5.6.3. CU2: Visualizar exhibiciones

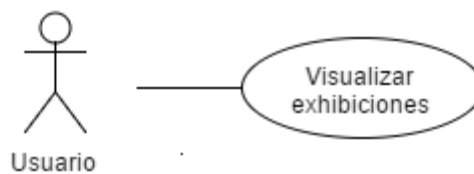


Figura 15 CU2: Visualizar exhibiciones

Nombre:	CU2: “Visualizar exhibiciones”
Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015
Propósito:	Ver un listado de los tipos de exhibiciones y sus nombres.
Actores:	Usuario.
Entradas:	Navegar a la pestaña de exhibiciones
Salidas:	Lista de las exhibiciones
Precondiciones:	El usuario debe encender el <i>bluetooth</i> y conectarse a una red WiFi.
Postcondiciones:	El usuario podrá acceder a la información.
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Comprueba tu conexión a internet” • “No hay conexión a internet”
Observaciones:	La ejecución de este caso de uso se realiza desde la pestaña de exhibición

Tabla 20 CU2: Visualizar exhibiciones

- Trayectorias del Caso de Uso.

Trayectoria Principal.

1. La aplicación móvil verifica que tenga conexión a internet [Trayectoria A]

2. La aplicación móvil verifica que el usuario este dentro del rango de algún **Beacon** [Trayectoria B]
3. La aplicación móvil muestra la información de las exhibiciones (Tipo de exhibición y nombre)

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. La aplicación no detecta conexión a internet
2. La aplicación muestra el mensaje “No hay conexión a internet”
3. Volver al paso 1 de trayectoria principal.

--Fin de Trayectoria Alternativa A.

Trayectoria alternativa B.

1. La aplicación no detecta ningún **Beacon**
2. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa B.

- Diseño de pantallas del caso de uso.



Figura 16 Pantalla exhibiciones

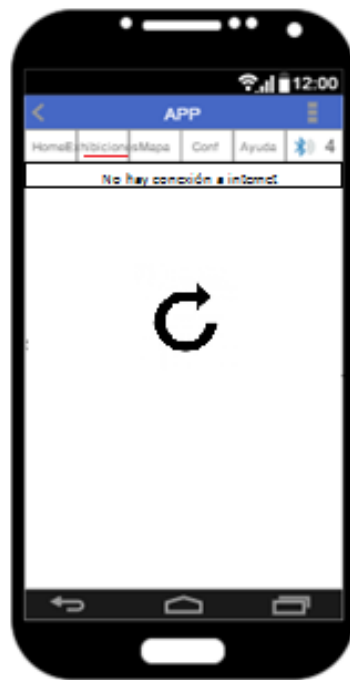


Figura 17 Pantalla no hay conexión a internet



Figura 18 Pantalla Exhibición no encontrada

5.6.4. CU3: Visualizar distribución del museo

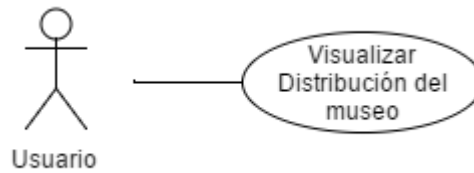


Figura 19 CU3: Visualizar distribución del museo

Nombre:	CU3: “Visualizar distribución del museo”
Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015
Propósito:	Ver la distribución general del museo.
Actores:	Usuario.
Entradas:	Navegar a la pestaña de Distribución
Salidas:	Se mostrará la distribución del museo.
Precondiciones:	El usuario debe encender el <i>bluetooth</i> y conectarse a una red WiFi.
Postcondiciones:	El usuario podrá visualizar la distribución del museo.
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Distribución no encontrado ” • “No hay conexión a internet”
Observaciones:	La ejecución de este caso de uso se realiza desde la pestaña de mapa.

Tabla 21 CU3: Visualizar distribución del museo

- Trayectorias del Caso de Uso.

Trayectoria Principal.

1. La aplicación verifica que tenga conexión a internet [Trayectoria A]
2. La aplicación verifica que el usuario este dentro del rango de algún **Beacon** [Trayectoria B]
3. La aplicación muestra la distribución del museo (Imagen de la distribución)

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. La aplicación no detecta conexión a internet
2. La aplicación muestra el mensaje “No hay conexión a internet”
3. Volver al paso 1 de trayectoria principal.

--Fin de Trayectoria Alternativa A.

Trayectoria alternativa B.

1. La aplicación no detecta ningún **Beacon**
2. La aplicación muestra el mensaje “Distribución no encontrado”
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa B.

- Diseño de pantallas del caso de uso.



Figura 20 Pantalla distribución del museo

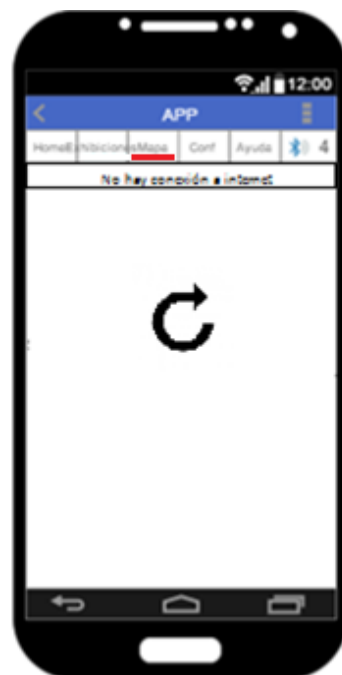


Figura 21 Pantalla no hay conexión a internet

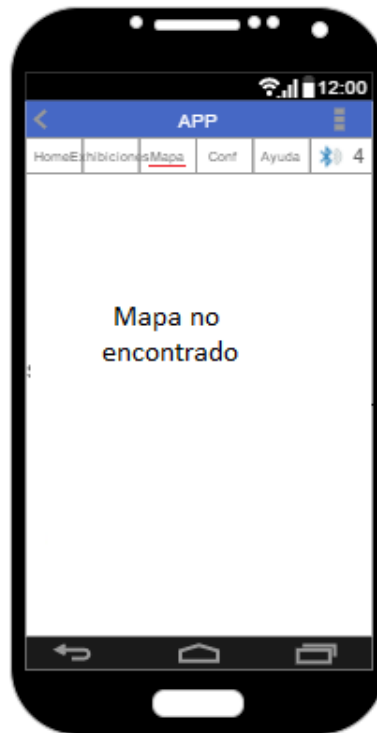


Figura 22 Pantalla distribución no encontrado

5.6.5. CU4: Configurar aplicación

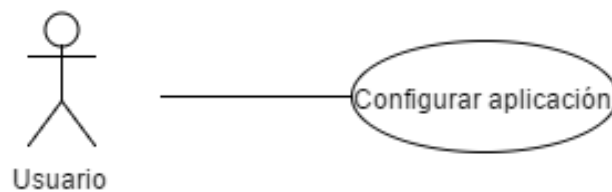


Figura 23 CU4: Configurar aplicación

Nombre:	CU4: “Configurar aplicación”
Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015
Propósito:	Cambiar la configuración de la aplicación.
Actores:	Usuario.
Entradas:	<ul style="list-style-type: none"> • Pre- descarga distancia media (si/no) • Notificación material cercano (si/no) • Notificación material a distancia media (si/no)
Salidas:	Se habilitara o deshabilitara la pre – descarga a distancia media, las notificaciones del material cercano o el material a distancia media
Postcondiciones:	El usuario cambiara la configuración de la aplicación
Observaciones:	La ejecución de este caso de uso se realiza desde la pestaña de configuración.

Tabla 22 CU4: Configurar aplicación

- Trayectorias del Caso de Uso.

Trayectoria Principal.

1. La aplicación muestra las opciones de configuración [Trayectoria A]

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. Activar o desactivar pre – descarga a distancia media.
2. Activar o desactivar notificaciones de material cercano
3. Activar o desactivar notificaciones de material a distancia media.

--Fin de Trayectoria Alternativa A.

- Diseño de pantallas del caso de uso.

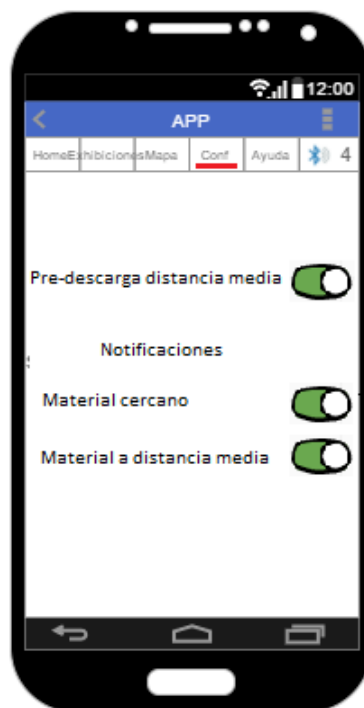


Figura 24 Pantalla de configuración

5.6.6. CU5: Visualizar Ayuda

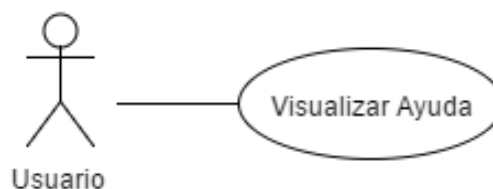


Figura 25 CU5: Visualizar Ayuda

Nombre:	CU5: “Visualizar ayuda”
Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015

Propósito:	Brindar al usuario una ayuda sobre el uso de la aplicación.
Actores:	Usuario.
Entradas:	Navegar a la pestaña de Ayuda
Salidas:	Se mostrará una pantalla de ayuda.
Postcondiciones:	El usuario podrá visualizar la ayuda.
Observaciones:	La ejecución de este caso de uso se realiza desde el botón de ayuda.

Tabla 23 CU5: Visualizar Ayuda

- Trayectorias del Caso de Uso.

Trayectoria Principal.

1. La aplicación muestra una ayuda sobre el uso de la aplicación.

--Fin de Trayectoria Principal.

- Diseño de pantallas del caso de uso.

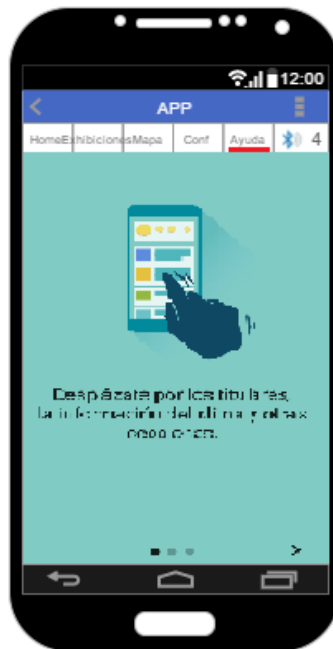


Figura 26 Pantalla de ayuda

5.6.7. CU6: Ver lista de material expuesto y exhibiciones



Figura 27 CU6: Ver lista de material expuesto y exhibiciones

Nombre:	CU6: “Ver lista de material expuesto y exhibiciones”
----------------	--

Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015
Propósito:	Ver la lista de material expuesto y exhibiciones que detecte la aplicación, poniendo un número en la pestaña de la cantidad de estas.
Actores:	Usuario.
Entrada:	Navegar a la pestaña de Beacon encontrados.
Salidas:	Lista del material expuesto y exhibiciones, con los datos: <ul style="list-style-type: none"> • Nombre • Descripción • Imagen en las exhibiciones
Precondiciones:	El usuario debe encender el <i>bluetooth</i> y conectarse a una red WiFi.
Postcondiciones:	El usuario podrá acceder a la información
Mensajes:	<ul style="list-style-type: none"> • “Acércate al material o exhibición”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Información no encontrada” • “No hay conexión a internet”
Observaciones:	La ejecución de este caso de uso se realiza desde la pantalla del icono de notificación.

Tabla 24 CU6: Ver lista de material expuesto y exhibiciones

- Trayectorias del caso de uso.

Trayectoria Principal.

1. La aplicación verifica que tenga conexión a internet [Trayectoria A]
2. La aplicación verifica que el usuario este dentro del rango de algún **Beacon** [Trayectoria B]
3. La aplicación muestra una lista del material expuesto [Trayectoria C,D,I]

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. La aplicación no detecta conexión a internet
2. La aplicación muestra el mensaje “No hay conexión a internet”
3. Volver al paso 1 de trayectoria principal.

--Fin de Trayectoria Alternativa A.

Trayectoria alternativa B.

1. La aplicación no detecta ningún **Beacon**
2. La aplicación muestra el mensaje “Información no encontrada”
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El usuario selecciona material expuesto o exhibición en la distancia “Far”
2. La aplicación muestra el mensaje “Acércate al material o exhibición”
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El usuario selecciona material expuesto cercano (Immediate)
2. El usuario selecciona la plantilla [Trayectoria E,F,G,H]
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa D.

Trayectoria alternativa E.

1. El usuario selecciona "Lista de archivos"
2. Se muestra lista de archivos a descargar
3. Usuario selecciona el archivo que desea descargar
4. Volver al paso 2 de trayectoria D.

--Fin de trayectoria alternativa E.

Trayectoria alternativa F.

1. El usuario selecciona la plantilla "Texto"
2. La aplicación muestra la plantilla Texto [Trayectoria E]
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa F.

Trayectoria alternativa G.

1. El usuario selecciona la plantilla "Imagen"
2. La aplicación muestra la plantilla Imagen [Trayectoria E]
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa G.

Trayectoria alternativa H.

1. El usuario selecciona la plantilla "Video"
2. La aplicación muestra la plantilla Video [Trayectoria E]
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa H.

Trayectoria alternativa I.

1. El usuario selecciona material expuesto Near.
2. El usuario selecciona la plantilla [Trayectoria E,F,G,H]
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa I.

- Diseño de pantallas del caso de uso.



Figura 28 Pantalla de material expuesto y exhibiciones

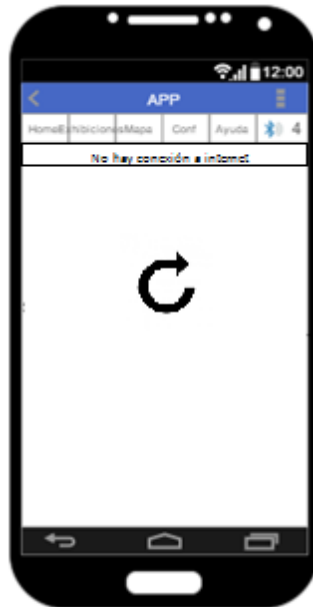


Figura 29 Pantalla no hay conexión a internet

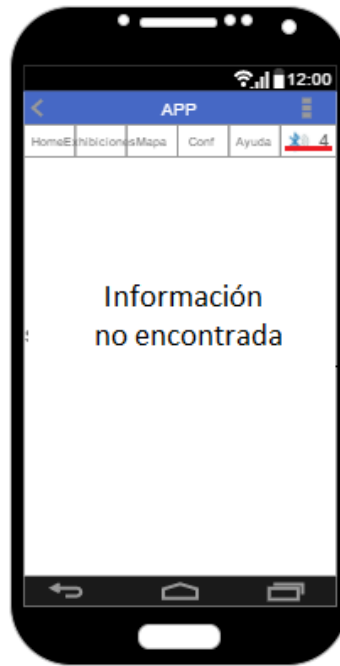


Figura 30 Información no encontrada



Figura 31 Plantilla imágenes



Figura 32 Plantilla texto



Figura 33 Plantilla video



Figura 34 Pantalla mensaje “Acércate al material o exhibición”



Figura 35 Pantalla de lista de archivos

5.6.8. CU7: Ver Notificaciones



Figura 36 CU7: Ver Notificaciones

Nombre:	CU7: “Ver notificaciones”
Autor:	José Eduardo Zamorano Aparicio
Fecha:	05 de Noviembre del 2015
Propósito:	Ver notificaciones que despliegue la aplicación.
Actores:	Usuario.
Salidas:	Notificaciones en el dispositivo móvil.
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe encender el <i>bluetooth</i> • Conectarse a una red WiFi. • El usuario debe configurar la aplicación para recibir notificaciones
Postcondiciones:	El usuario podrá acceder a la información seleccionando la notificación
Mensajes:	<ul style="list-style-type: none"> • “Acércate al material o exhibición”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Información no encontrada” • “No hay conexión a internet”

Tabla 25 CU7: Ver Notificaciones

- Trayectorias del caso de uso.

Trayectoria Principal.

1. El usuario hace click en la notificación que despliega la aplicación.
2. La aplicación muestra la lista de materiales [Trayectoria A,B,C,D]

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. El usuario selecciona “Lista de archivos”
2. Se muestra lista de archivos a descargar
3. Usuario selecciona el archivo que desea descargar.

--Fin de trayectoria alternativa A.

Trayectoria alternativa B.

1. El usuario selecciona la plantilla “Texto”
2. La aplicación muestra la plantilla Texto [Trayectoria A]

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El usuario selecciona la plantilla “Imagen”
2. La aplicación muestra la plantilla Imagen [Trayectoria A]

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El usuario selecciona la plantilla “Video”
2. La aplicación muestra la plantilla Video [Trayectoria A]

--Fin de trayectoria alternativa D.

- Diseño de pantallas del caso de uso.

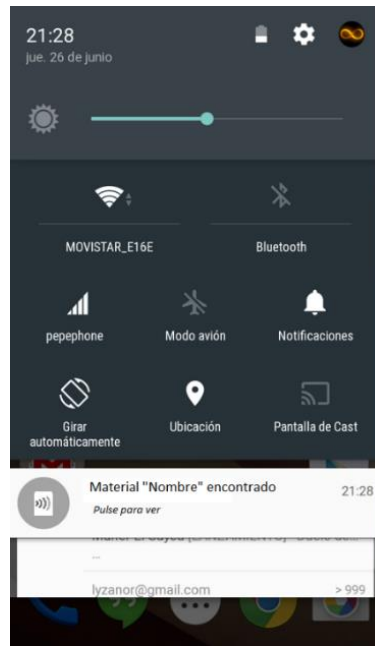


Figura 37 Pantalla de notificaciones

5.6.9. CU8: Iniciar sesión

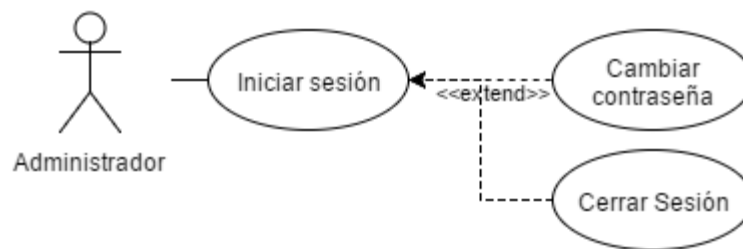


Figura 38 CU8: Iniciar sesión

Nombre:	CU8: “Iniciar Sesión”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador validar su identidad ante el sistema.
Actores:	Administrador.
Entradas:	Nombre de usuario y contraseña.
Salidas:	Validar si el usuario y contraseña son correctos para acceder al sistema.
Precondiciones:	El administrador debe estar registrado en la base de datos.
Postcondiciones:	El sistema permite el acceso al administrador.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Acceso Permitido, Bienvenido.”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “¡Error! La cuenta especificada no existe. Revisa el nombre de usuario y la contraseña y vuelve a intentarlo.”
Observaciones:	La ejecución de este caso de uso se realiza desde el inicio del sistema alojado en el servidor.

Tabla 26 CU8: Iniciar sesión

- Trayectorias del caso de uso.

Trayectoria Principal.

1. El administrador introduce su nombre de usuario y contraseña.
2. El sistema verifica que el usuario y contraseña sean correctas. [Alternativa A, B]
3. El sistema da acceso al servidor.
4. El sistema muestra la pantalla principal. [Alternativa C,F]

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. El sistema verifica que el usuario y/o contraseña son incorrectas.
2. El sistema muestra el mensaje “¡Error! La cuenta especificada no existe. Revisa el nombre de usuario y la contraseña y vuelve a intentarlo.”
3. Volver al paso 1 de trayectoria principal.

--Fin de Trayectoria Alternativa A.

Trayectoria alternativa B.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “¡Error! La cuenta especificada no existe. Revisa el nombre de usuario y la contraseña y vuelve a intentarlo.”
3. Volver al paso 1 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El administrador en la pantalla principal en el menú selecciona la opción de Administrador.
2. El administrador elige la opción de Cambiar Contraseña.
3. El sistema muestra la pantalla emergente para introducir los datos.
4. El administrador introduce los datos de la nueva contraseña.
5. El sistema verifica que los datos son correctos. [Alternativa D].
6. El sistema muestra el mensaje “OK! Su contraseña ha sido cambiada”.
7. Volver al paso 4 de trayectoria principal.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El sistema detecta que las contraseñas no coinciden.
2. El sistema muestra el mensaje “Las contraseñas no coinciden”.
3. Volver al paso 4 de la trayectoria C.

--Fin de trayectoria alternativa D.

Trayectoria alternativa E.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “Las contraseñas no coinciden”.
3. Volver al paso 4 de la trayectoria C.

--Fin de trayectoria alternativa E.

Trayectoria alternativa F.

1. El administrador en el menú selecciona la opción de Administrador.
2. El administrador elige la opción de Logout.
3. El sistema limpia toda la información de sesión del administrador.
4. El sistema muestra el mensaje “Sesión Cerrada”.
5. Volver al paso 1 de la trayectoria principal.

--Fin trayectoria alternativa F.

- Diseño de pantallas del caso de uso.

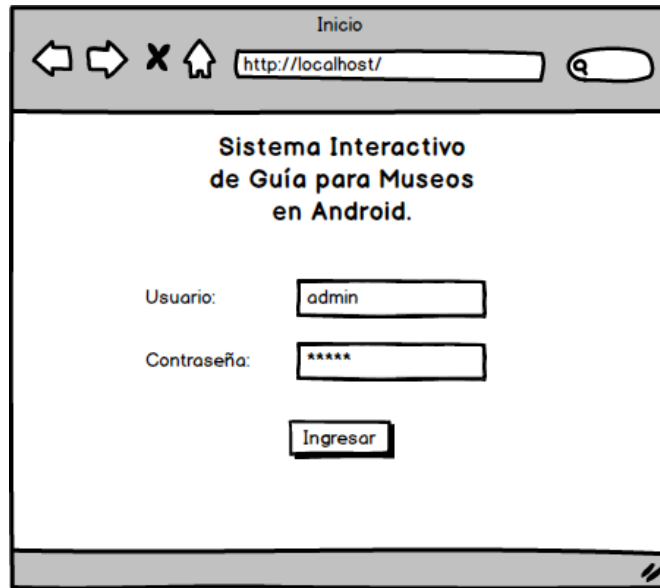


Figura 39 Pantalla inicio de sesión

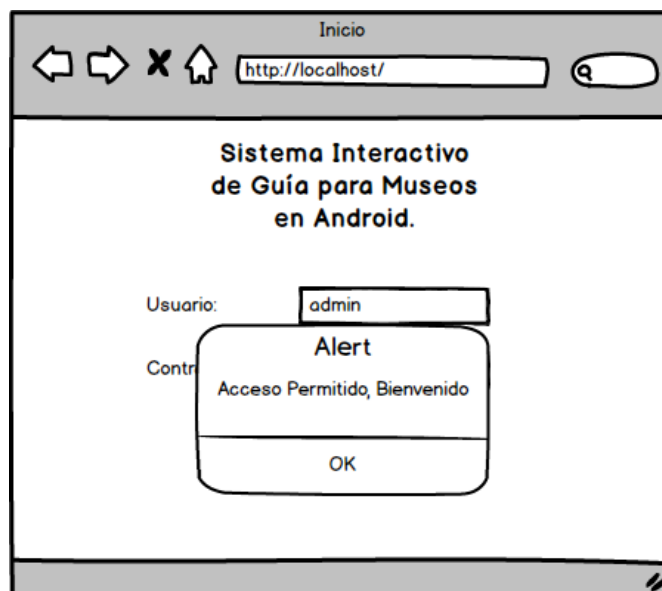


Figura 40 Pantalla de Acceso

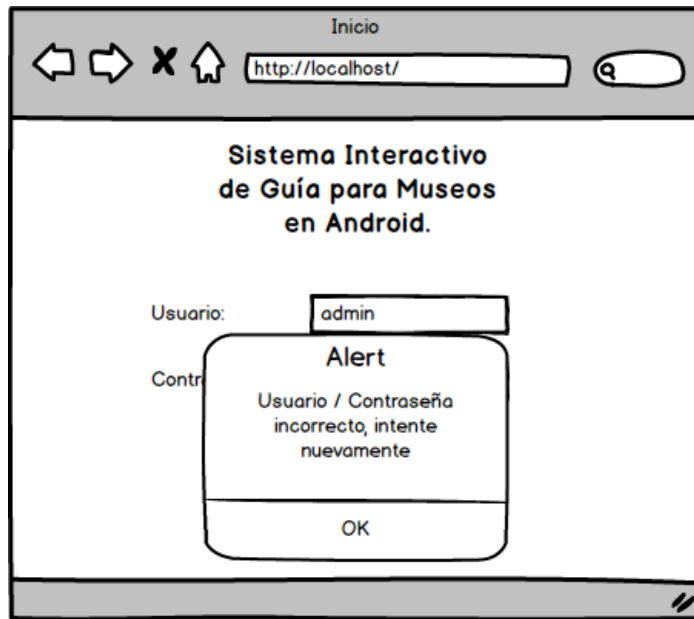


Figura 41 Pantalla campos incorrectos



Figura 42 Pantalla Campos Incompletos

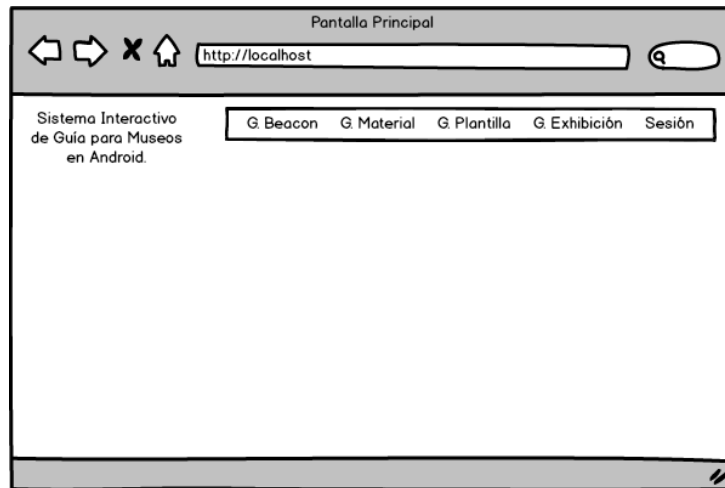


Figura 43 Pantalla Principal

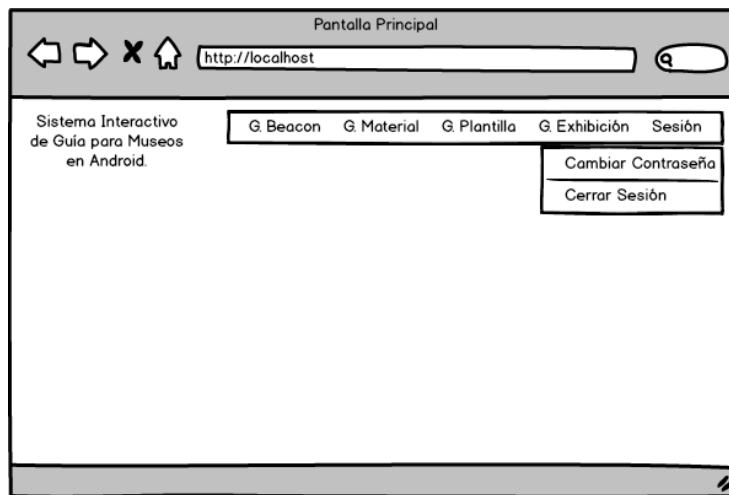


Figura 44 Pantalla Menú Sesión

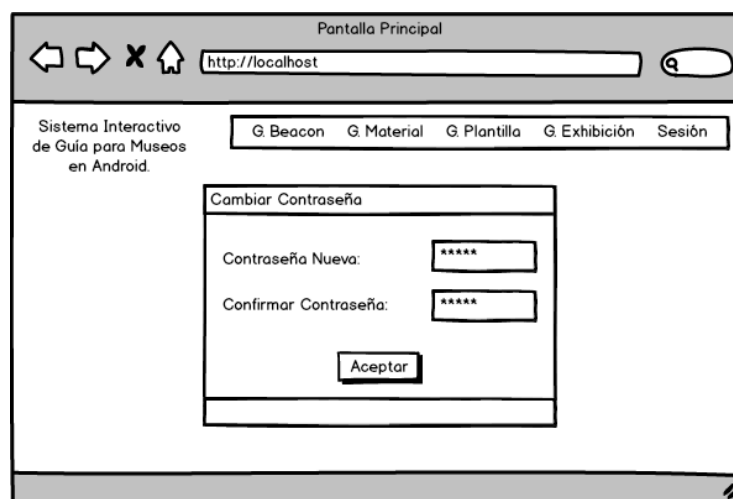


Figura 45 Pantalla Cambiar contraseña

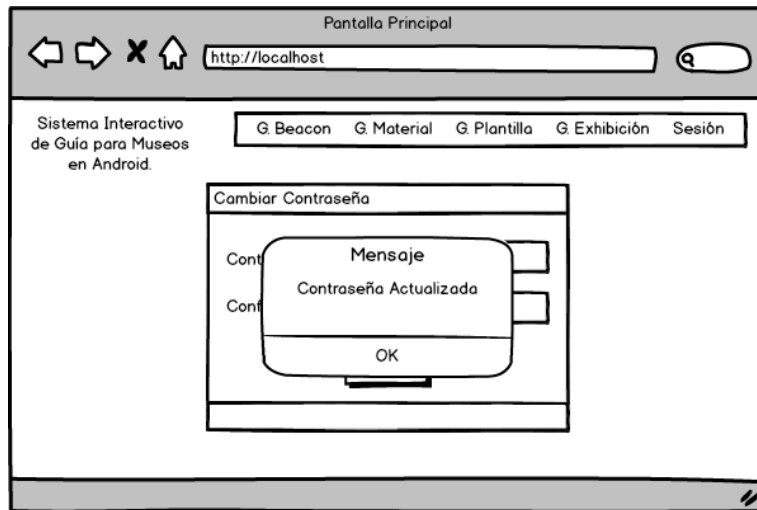


Figura 46 Pantalla Contraseña actualizada.

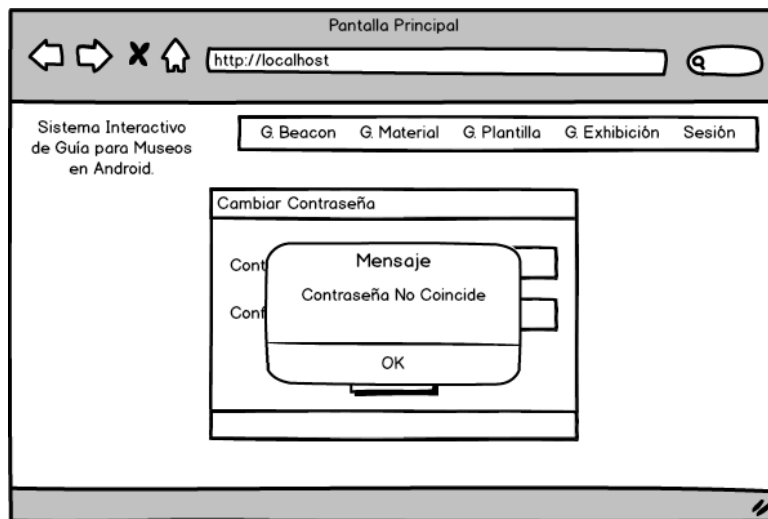


Figura 47 Pantalla Contraseña no coincide

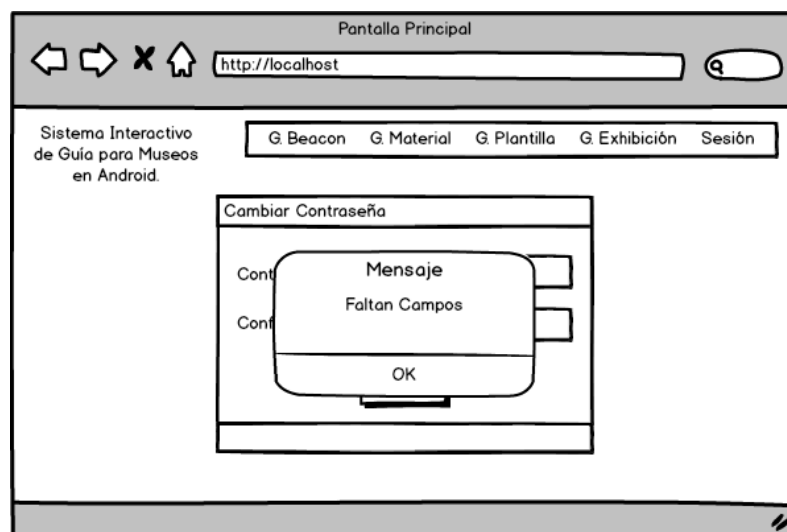


Figura 48 Pantalla Faltan Campos

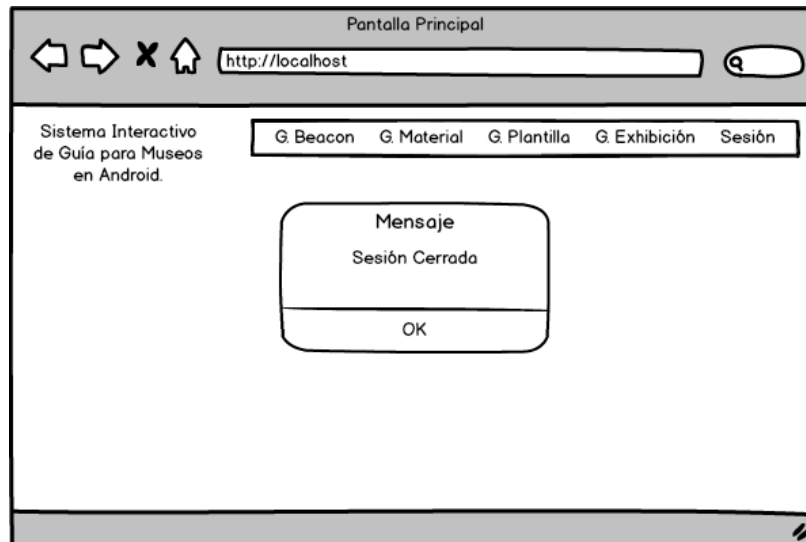


Figura 49 Pantalla Cerrar Sesión

5.6.10. CU9: Gestión de Material

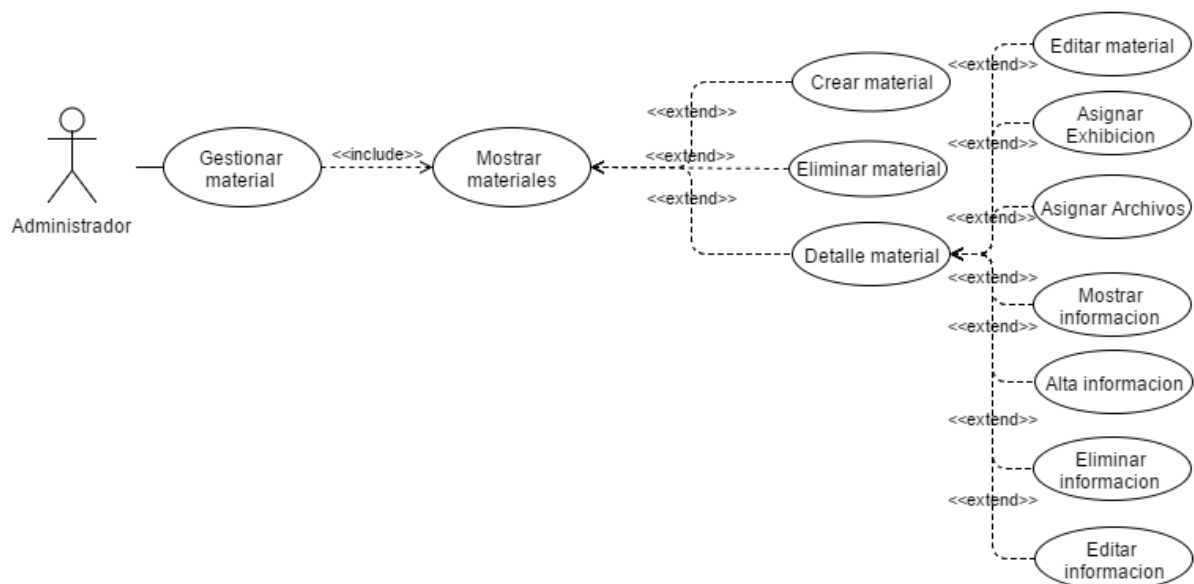


Figura 50 CU9: Gestión de material

Nombre:	CU9: “Gestión de Material”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador gestionar el material que se mostrara en la aplicación.
Actores:	Administrador.
Entradas:	Mostrará las opciones disponibles en el sistema para la gestión de material.
Salidas:	Validar las opciones de gestión del material.
Precondiciones:	El administrador debe tener sesión activa en el sistema.
Postcondiciones:	El sistema permitirá visualizar la información modificada.

Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Material dado de Alta”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Faltan campos por ingresar” • “Datos no válidos”
Observaciones:	La ejecución de este caso de uso se realiza desde el menú principal de la aplicación alojada en el servidor.

Tabla 27 CU9: Gestión de material

- Trayectorias del Caso de Uso.

Trayectoria principal.

1. El administrador en la pantalla principal en el menú selecciona la opción de Gestionar Material.
2. El sistema muestra la pantalla de Gestionar Material.
3. El sistema verifica que hay materiales registrados. [Trayectoria A].
4. El sistema muestra una lista con los materiales disponibles. [Trayectoria B].
5. El administrador selecciona un material de la lista. [Trayectoria D, E].

--Fin trayectoria principal.

Trayectoria alternativa A.

1. El sistema verifica que no hay materiales registrados.
2. El sistema muestra una lista vacía. [Trayectoria B].

--Fin de trayectoria alternativa B.

Trayectoria alternativa B.

1. El administrador presiona de pantalla “Gestionar Material” el botón de Alta.
2. El administrador introduce los datos del material.
3. El administrador presiona el botón Crear.
4. El sistema verifica que los datos son correctos. [Trayectoria C].
5. El sistema muestra el mensaje “Material dado de Alta”.
6. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “Faltan campos obligatorios”.
3. Volver al paso 2 de la trayectoria alternativa B.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El administrador presiona el botón Eliminar.
2. El sistema muestra el mensaje “Material eliminado”.
3. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa D.

Trayectoria alternativa E.

1. El administrador presiona el botón Detalles.
2. El sistema verifica los datos registrados del material.
3. El sistema verifica la exhibición registrada.
4. El sistema verifica los archivos registrados.
5. El sistema verifica las plantillas registradas.
6. El sistema verifica el museo registrado.
7. El sistema muestra la pantalla “Detalle de Material”. [Trayectoria F, I, J, K, L, O].
8. El administrador presiona el botón Regresar.
9. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa E.

Trayectoria alternativa F.

1. El administrador presiona de pantalla “Detalle de Material” de la sección de Material el botón de Editar.
2. El sistema muestra la pantalla emergente para introducir los datos.
3. El administrador introduce los datos del Material.
4. El administrador presionara el botón Finalizar.
5. El sistema verifica que los datos son correctos. [Trayectoria G, H].
6. El sistema muestra el mensaje “Material Editado”
7. Volver al paso 2 de trayectoria alternativa F.

--Fin de trayectoria alternativa F.

Trayectoria alternativa G.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “Faltan campos por ingresar”.
3. Volver al paso 3 de la trayectoria F.

--Fin de trayectoria alternativa G.

Trayectoria alternativa H.

1. El sistema detecta que hay datos no válidos en el formulario.
2. El sistema muestra el mensaje “Datos no Validos”.
3. Volver al paso 3 de la trayectoria alternativa F.

--Fin de trayectoria alternativa H.

Trayectoria alternativa I.

1. El administrador presiona de pantalla “Detalle de Material” de la sección de Museo el botón de Detalle de Museo.
2. El sistema muestra la pantalla de “Detalle de Museo” (Ir a CU15).

--Fin de trayectoria alternativa I.

Trayectoria alternativa J.

1. El administrador presiona de pantalla “Detalle de Material” de la sección Exhibición el botón de Detalle de Exhibición.
2. El sistema muestra la pantalla de “Detalle de Exhibición” (Ir a CU10).

--Fin de trayectoria alternativa J.

Trayectoria alternativa K.

1. El administrador presiona de pantalla “Detalla de Material” de la sección Archivo el botón de Asignar.
2. El sistema muestra en una ventana emergente la lista de archivos que se encuentran disponibles.
3. El administrador selecciona uno o más archivos de la lista.
4. El administrador presiona el botón de Agregar.
5. El sistema muestra el mensaje “Archivos Guardados”.
6. Volver al paso 2 de trayectoria alternativa E.

--Fin de trayectoria alternativa K.

Trayectoria alternativa L.

1. El administrador presiona de pantalla “Detalle de Material” de la sección de Plantillas el botón de Asignar.
2. El sistema muestra la pantalla emergente para seleccionar e introducir los datos.
3. El administrador introduce los datos de la Plantilla.
4. El administrador presionara el botón Finalizar.
5. El sistema verifica que los datos son correctos. [Trayectoria M, N].
6. El sistema muestra el mensaje “Datos Guardados en Plantilla”
7. Volver al paso 2 de trayectoria alternativa E.

--Fin de trayectoria alternativa L.

Trayectoria alternativa M.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “Faltan campos por ingresar”.
3. Volver al paso 3 de la trayectoria alternativa L.

--Fin de trayectoria alternativa M.

Trayectoria alternativa N.

1. El sistema detecta que hay datos no válidos en el formulario.
2. El sistema muestra el mensaje “Datos no Validos”.
3. Volver al paso 3 de la trayectoria alternativa L.

--Fin de trayectoria alternativa N.

Trayectoria alternativa O.

1. El administrador presiona de pantalla “Detalle de Material” de la sección de Archivos el botón de Asignar.
2. El administrador desasigna uno o varios archivos.
3. Volver al paso 2 de trayectoria alternativa E.

--Fin de trayectoria alternativa O.

Trayectoria alternativa P.

1. El administrador presiona de pantalla “Detalle de Material” de la sección de Archivos el botón de Ver.

2. El navegador web abre una nueva pestaña para poder visualizar el archivo PDF.
- Fin de trayectoria alternativa P.
- Diseño de pantallas del caso de uso.

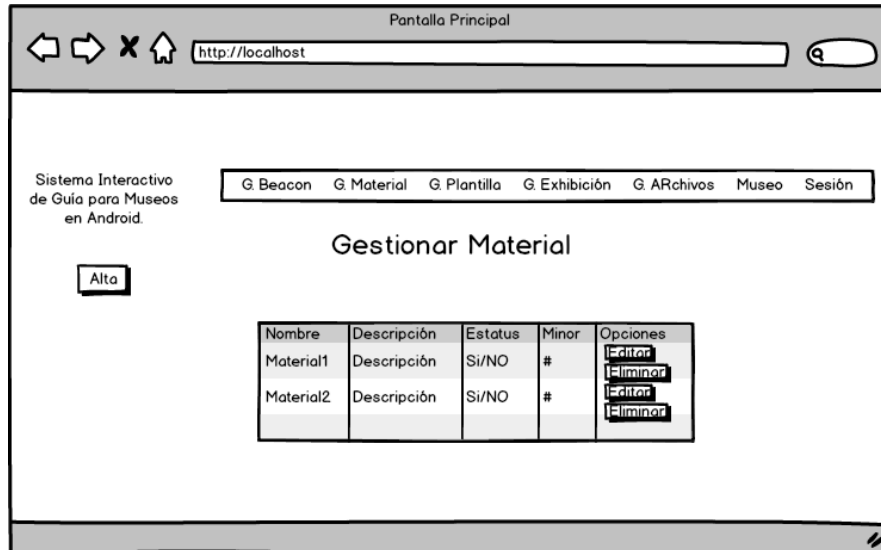


Figura 51 Pantalla de Gestionar Material.

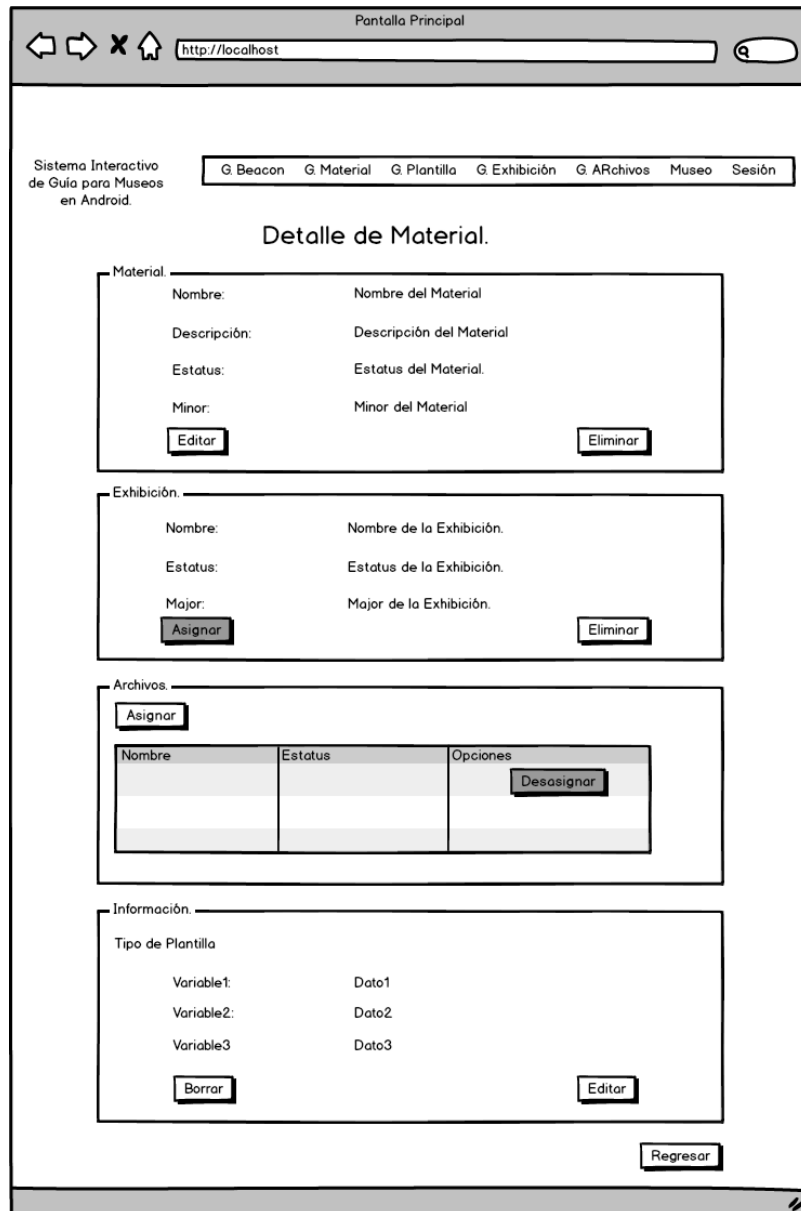


Figura 52 Pantalla Detalle Material.

5.6.11. CU10: Gestión de Exhibición

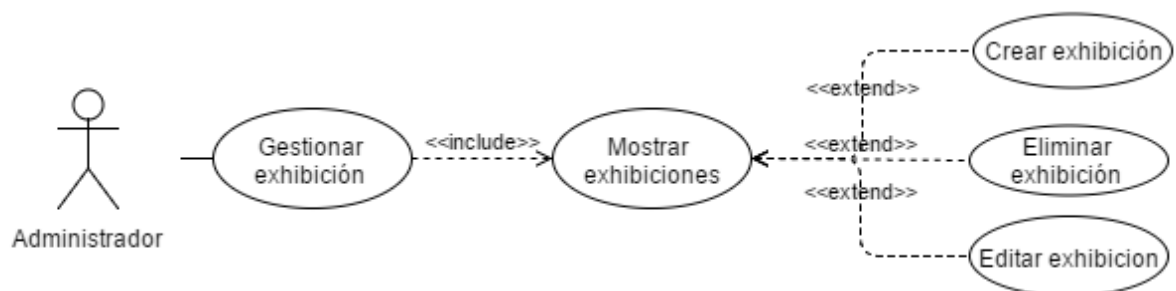


Figura 53 CU10: Gestión de exhibición

Nombre:	CU10: “Gestión de Exhibición”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador gestionar las exhibiciones.
Actores:	Administrador.
Entradas:	Mostrará las opciones disponibles en el sistema para la gestión de exhibiciones.
Salidas:	Validar las opciones de gestión de las exhibiciones.
Precondiciones:	El administrador debe tener sesión activa en el sistema.
Postcondiciones:	El sistema permitirá visualizar la información modificada.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Exhibición editada”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “¡Error!”
Observaciones:	La ejecución de este caso de uso se realiza desde el menú principal de la aplicación alojada en el servidor.

Tabla 28 CU10: Gestión de exhibición

- Trayectorias del Caso de Uso.

Trayectoria principal.

1. El administrador en la pantalla principal en el menú selecciona la opción de Gestionar Exhibición.
2. El sistema muestra la pantalla de Gestionar Exhibición.
3. El sistema verifica que hay exhibiciones registradas. [Trayectoria A].
4. El sistema muestra una lista con las exhibiciones disponibles. [Trayectoria B].
5. El administrador selecciona una exhibición de la lista. [Trayectoria D, E].

--Fin trayectoria principal.

Trayectoria alternativa A.

1. El sistema verifica que no hay exhibiciones registradas.
2. El sistema muestra una lista vacía. [Trayectoria B].

--Fin de trayectoria alternativa A.

Trayectoria alternativa B.

1. El administrador presiona de pantalla “Gestionar Exhibición” el botón de Alta.
2. El administrador introduce los datos de la exhibición.
3. El administrador presiona el botón Finalizar.
4. El sistema verifica que los datos son correctos. [Trayectoria C, D].
5. El sistema muestra el mensaje “Exhibición Creada”
6. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “¡Error!”.
3. Volver al paso 2 de la trayectoria alternativa B.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El sistema detecta que hay datos no válidos en el formulario.
2. El sistema muestra el mensaje “¡Error!”.
3. Volver al paso 2 de la trayectoria alternativa B.

--Fin de trayectoria alternativa D.

Trayectoria alternativa E.

1. El administrador presiona el botón Eliminar.
2. El sistema muestra el mensaje “Exhibición eliminada”
3. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa E.

Trayectoria alternativa F.

1. El administrador presiona el botón Detalles.
2. El sistema verifica los datos registrados de la Exhibición.
3. El sistema muestra la pantalla “Detalle de Exhibición”. [Trayectoria G, J, K].
4. El administrador presiona el botón Regresar.
5. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa F.

Trayectoria alternativa G.

1. El administrador presiona de pantalla “Detalle de Exhibición” de la sección de Exhibición el botón de Editar.
2. El sistema muestra la pantalla emergente para introducir los datos.
3. El administrador introduce los datos de la Exhibición.
4. El administrador presionara el botón Finalizar.
5. El sistema verifica que los datos son correctos. [Trayectoria H, I]
6. El sistema muestra el mensaje “ Exhibición Editada”
7. Volver al paso 2 de trayectoria alternativa F.

--Fin de trayectoria alternativa G.

Trayectoria alternativa H.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “¡Error!”.
3. Volver al paso 3 de la trayectoria G.

--Fin de trayectoria alternativa H.

Trayectoria alternativa I.

1. El sistema detecta que hay datos no válidos en el formulario.
2. El sistema muestra el mensaje “¡Error!”.
3. Volver al paso 2 de la trayectoria alternativa G.

--Fin de trayectoria alternativa I.

Trayectoria alternativa J.

1. El administrador presiona de pantalla “Detalle de Exhibición” de la sección de Museo el botón de Detalle de Museo.
2. El sistema muestra la pantalla de “Detalle de Museo” (Ir a CU15).

--Fin de trayectoria alternativa J.

Trayectoria alternativa K.

1. El administrador presiona de pantalla “Detalle de Exhibición” de la sección de Tipo de Exhibición el botón de Detalle de Tipo de Exhibición.
2. El sistema muestra la pantalla de “Tipo de Exhibición”.

--Fin de trayectoria alternativa K.

- Diseño de pantallas del caso de uso.

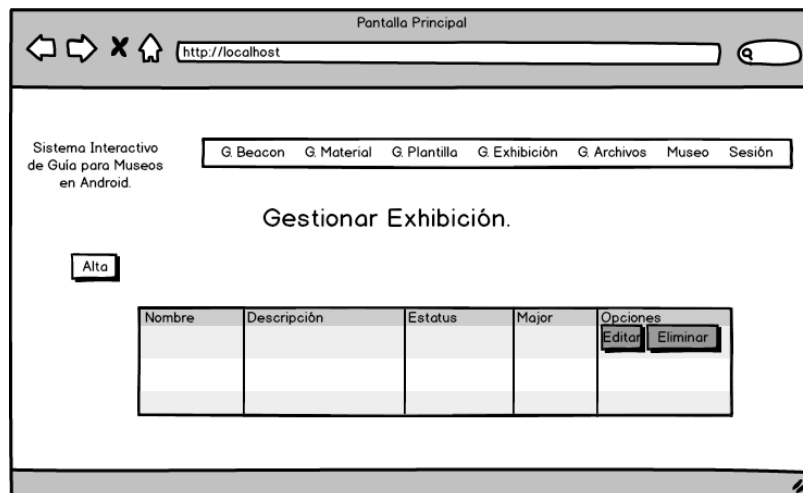


Figura 54 Pantalla Gestionar Exhibición



Figura 55 Pantalla Detalle de Exhibición

5.6.12. CU11: Gestión de Archivos



Figura 56 CU11: Gestión de Archivos

Nombre:	CU11: “Gestión de Archivos”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador gestionar los Archivos.
Actores:	Administrador.
Entradas:	Mostrará las opciones disponibles en el sistema para la gestión de Archivos.
Salidas:	Validar las opciones de gestión de los Archivos.
Precondiciones:	El administrador debe tener sesión activa en el sistema.
Postcondiciones:	El sistema permitirá visualizar la información modificada.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Archivo creado”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Faltan campos por ingresar”
Observaciones:	La ejecución de este caso de uso se realiza desde el menú principal de la aplicación alojada en el servidor.

Tabla 29 CU11: Gestión de Archivos

- Trayectorias del Caso de Uso.

Trayectoria principal.

1. El administrador en la pantalla principal en el menú selecciona la opción de Gestionar Archivos.
2. El sistema muestra la pantalla de Gestionar Archivos.
3. El sistema verifica que hay Archivos registrados. [Trayectoria A].
4. El sistema muestra una lista con los Archivos disponibles. [Trayectoria B].
5. El administrador selecciona un Archivo de la lista. [Trayectoria D, E].

--Fin trayectoria principal.

Trayectoria alternativa A.

1. El sistema verifica que no hay Archivos registrados.
2. El sistema muestra una lista vacía. [Trayectoria B].

--Fin de trayectoria alternativa A.

Trayectoria alternativa B.

1. El administrador presiona de pantalla “Gestionar Archivos” el botón de Crear.
2. El administrador introduce los datos del Archivo.
3. El administrador presiona el botón Finalizar.
4. El sistema verifica que los datos son correctos. [Trayectoria C].
5. El sistema muestra el mensaje “ Archivo Creado”
6. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “¡Error! Falta archivo”.
3. Volver al paso 2 de la trayectoria alternativa B.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El administrador presiona el botón Eliminar.
2. El sistema muestra el mensaje “ Archivo eliminado”
3. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa D.

Trayectoria alternativa E.

1. El administrador presiona el botón Ver.
2. El sistema verifica la dirección donde se encuentra guardado el Archivo.
3. El navegador web abre una nueva pestaña y muestra el archivo PDF.

--Fin de trayectoria alternativa E.

- Diseño de pantallas del caso de uso.

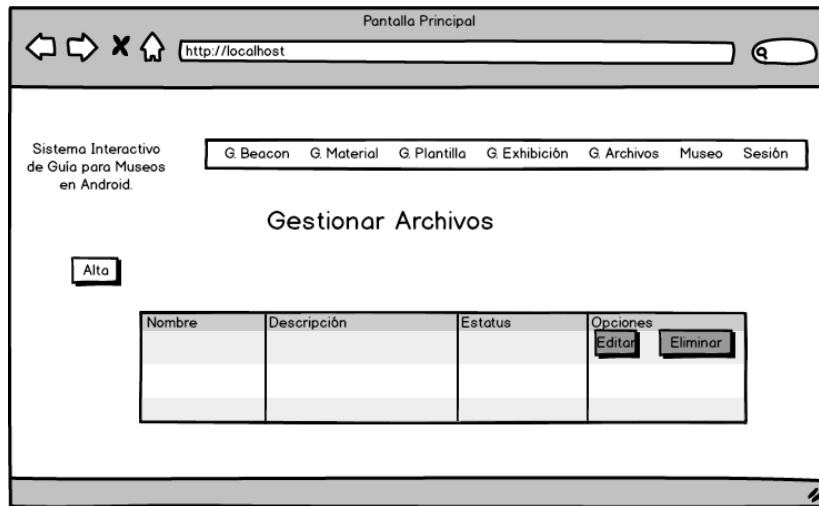


Figura 57 Pantalla de Gestionar Archivo

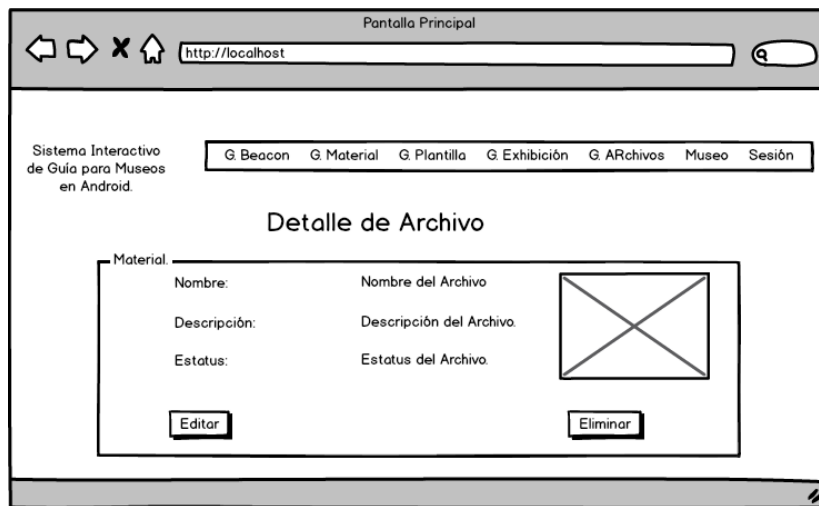


Figura 58 Pantalla de Detalle de Archivo

5.6.13. CU12: Gestión de Plantillas

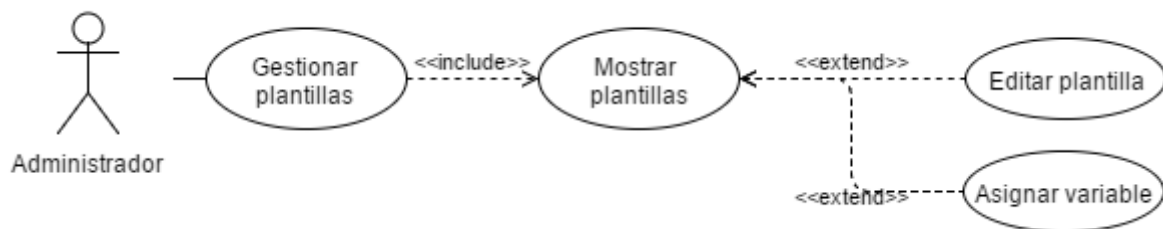


Figura 59 CU12: Gestión de Plantilla

Nombre:	CU12: “Gestión de Plantilla”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador gestionar las Plantillas.
Actores:	Administrador.
Entradas:	Mostrará las opciones disponibles en el sistema para la gestión de Plantillas.

Salidas:	Validar las opciones de gestión de las Plantillas.
Precondiciones:	El administrador debe tener sesión activa en el sistema.
Postcondiciones:	El sistema permitirá visualizar la información modificada.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Plantilla Editada”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Faltan campos por ingresar”
Observaciones:	La ejecución de este caso de uso se realiza desde el menú principal de la aplicación alojada en el servidor.

Tabla 30 CU12: Gestión de Plantilla

- Trayectorias del Caso de Uso.

Trayectoria principal.

1. El administrador en la pantalla principal en el menú selecciona la opción de Gestionar Plantilla.
2. El sistema muestra la pantalla de Gestionar Plantilla.
3. El sistema verifica que hay Plantillas registradas.
4. El sistema muestra una lista con las Plantillas disponibles.
5. El administrador selecciona un Archivo de la lista. [Trayectoria A].

--Fin trayectoria principal.

Trayectoria alternativa A.

1. El administrador presiona el botón Detalles.
2. El sistema verifica los datos registrados de la Plantilla.
3. El sistema muestra la pantalla “Detalle de Plantilla”. [Trayectoria B].
4. El administrador presiona el botón Regresar.
5. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa A.

Trayectoria alternativa B.

1. El administrador presiona de pantalla “Detalle de Plantilla” de la sección de Plantilla el botón de Editar.
2. El sistema muestra la pantalla emergente para introducir los datos.
3. El administrador introduce los datos de la Plantilla.
4. El administrador presionara el botón Finalizar.
5. El sistema verifica que los datos son correctos. [Trayectoria C]
6. El sistema muestra el mensaje “ Plantilla Editada”
7. Volver al paso 2 de trayectoria alternativa A.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “Faltan campos por ingresar”.
3. Volver al paso 3 de la trayectoria B.

--Fin de trayectoria alternativa C.

- Diseño de pantallas del caso de uso.

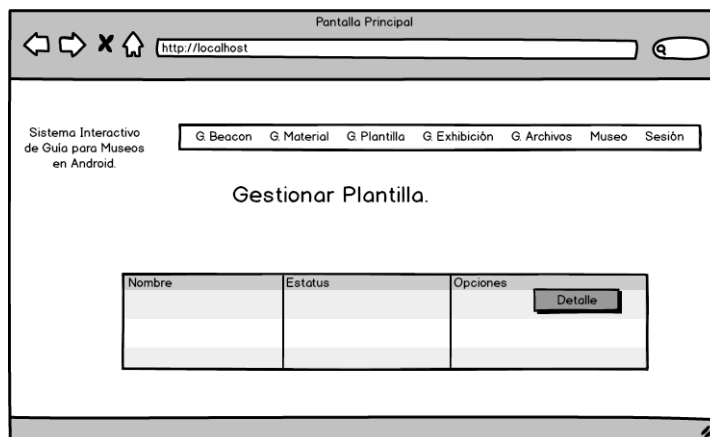


Figura 60 Pantalla Gestionar Plantilla

5.6.14. CU13: Gestión de variable de plantilla

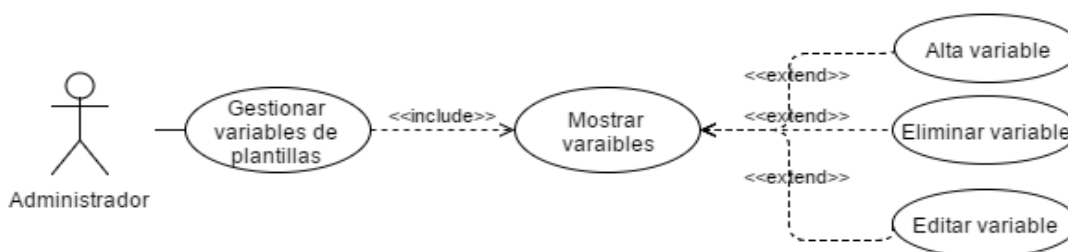


Figura 61 CU13: Gestión de variable de plantilla

Nombre:	CU13: “Gestión de Variable de Plantilla”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador gestionar las Variables de Plantillas.
Actores:	Administrador.
Entradas:	Mostrará las opciones disponibles en el sistema para la gestión de Variables de Plantillas.
Salidas:	Validar las opciones de gestión de las Variables de Plantillas.
Precondiciones:	El administrador debe tener sesión activa en el sistema.
Postcondiciones:	El sistema permitirá visualizar la información modificada.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Variable de Plantilla Creada”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Faltan campos por ingresar” • “Datos no Validos”
Observaciones:	La ejecución de este caso de uso se realiza desde el menú principal de la aplicación alojada en el servidor.

Tabla 31 CU13: Gestión de variable de plantilla

- Trayectorias del Caso de Uso.

Trayectoria principal.

1. El administrador en la pantalla principal en el menú selecciona la opción de Gestionar Variables de Plantillas.
2. El sistema muestra la pantalla de Gestionar Variables de Plantillas.
3. El sistema verifica que hay Variables de Plantillas registradas. [Trayectoria A]
4. El sistema muestra una lista con las Variables de Plantillas disponibles.
5. El administrador selecciona una Variable de Plantilla de la lista. [Trayectoria B, C].

--Fin trayectoria principal.

Trayectoria alternativa A.

1. El sistema verifica que no hay Variables de Plantillas registradas.
2. El sistema muestra una lista vacía.
3. Volver al paso 2 de la plantilla principal.

--Fin de trayectoria alternativa A.

Trayectoria alternativa B.

1. El administrador presiona el botón Eliminar.
2. El sistema muestra el mensaje “ Variable de Plantilla eliminado”
3. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El administrador presiona el botón Editar.
2. El sistema muestra la pantalla emergente para introducir los datos.
3. El administrador introduce los datos de la Variable de Plantilla.
4. El administrador presionara el botón Finalizar.
5. El sistema verifica que los datos son correctos. [Trayectoria D]
6. El sistema muestra el mensaje “ Variables de Plantilla Editada”
7. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “Faltan campos por ingresar”.
3. Volver al paso 3 de la trayectoria D.

--Fin de trayectoria alternativa G.

- Diseño de pantallas del caso de uso.

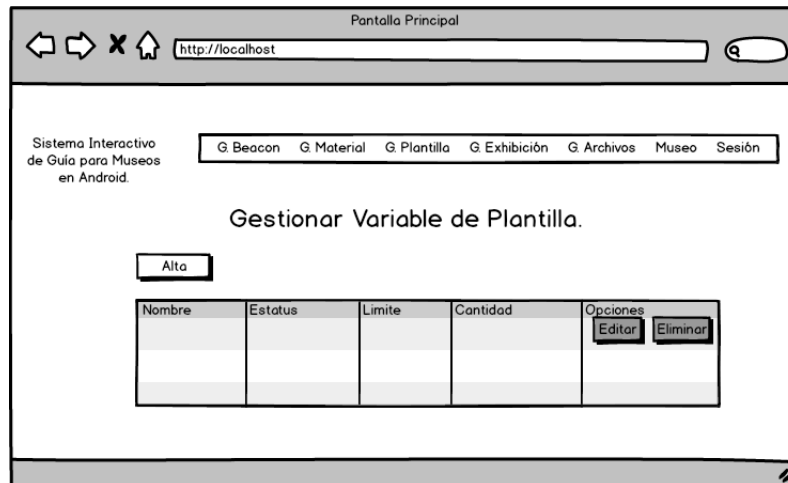


Figura 62 Pantalla Gestionar Variable de Plantilla

5.6.15. CU14: Recuperar contraseña

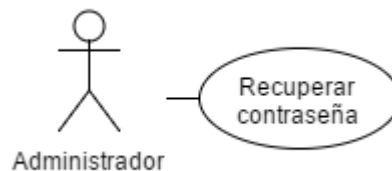


Figura 63 CU14: Recuperar contraseña

Nombre:	CU14: “Recuperar Contraseña”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador validar su identidad ante el sistema.
Actores:	Administrador.
Entradas:	Correo del Administrador.
Salidas:	Validar la recuperación de su contraseña.
Precondiciones:	El administrador debe estar registrado en la base de datos.
Postcondiciones:	El sistema permite la recuperación de contraseña al administrador.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Correo Enviado”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Correo no válido.”
Observaciones:	La ejecución de este caso de uso se realiza desde el inicio del sistema alojado en el servidor.

Tabla 32 CU14: Recuperar contraseña

- Trayectorias del caso de uso.

Trayectoria Principal.

1. El administrador presiona el botón de recuperar contraseña.
2. El sistema muestra una pantalla emergente para ingresar los datos.
3. El sistema verifica que los datos sean correctos [Alternativa A, B]

4. El sistema muestra el mensaje “Correo enviado”.

--Fin de Trayectoria Principal.

Trayectoria alternativa A.

1. El sistema verifica que los datos son incorrectos.
2. El sistema muestra el mensaje “Correo no válido”
3. Volver al paso 2 de trayectoria principal.

--Fin de Trayectoria Alternativa A.

5.6.16. CU15: Gestión de Museos

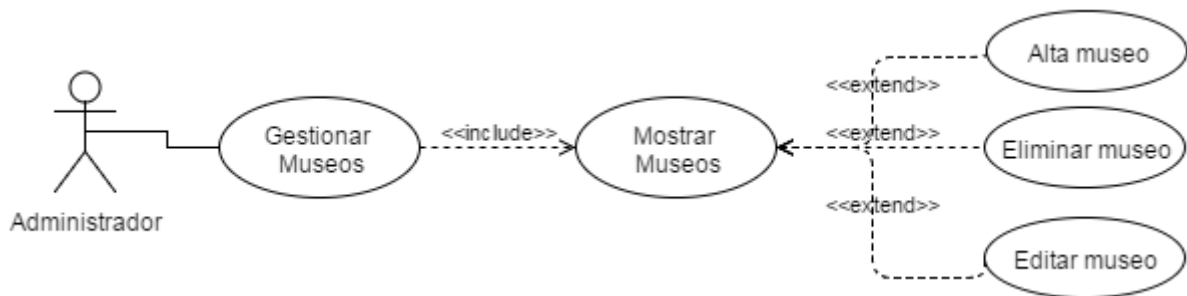


Figura 64 CU15: Gestión de Museos

Nombre:	CU15: “Gestión de Museos”
Autor:	Marcos Ricardo García García
Fecha:	05 de Noviembre del 2015
Propósito:	Permitir al administrador gestionar los Museos.
Actores:	Administrador.
Entradas:	Mostrará las opciones disponibles en el sistema para la gestión de Museos.
Salidas:	Validar las opciones de gestión de los Museos.
Precondiciones:	El administrador debe tener sesión activa en el sistema.
Postcondiciones:	El sistema permitirá visualizar la información modificada.
Mensajes:	Mensaje de notificación: <ul style="list-style-type: none"> • “Museo creado”
Errores:	Mensaje de error: <ul style="list-style-type: none"> • “Faltan campos por ingresar”
Observaciones:	La ejecución de este caso de uso se realiza desde el menú principal de la aplicación alojada en el servidor.

Tabla 33 CU15: Gestión de Museos

- Trayectorias del Caso de Uso.

Trayectoria principal.

1. El administrador en la pantalla principal en el menú selecciona la opción de Gestionar Museos.
2. El sistema muestra la pantalla de Gestionar Museos.
3. El sistema verifica que hay Museos registrados. [Trayectoria A].
4. El sistema muestra una lista con los Museos disponibles. [Trayectoria B].

5. El administrador selecciona un Museo de la lista. [Trayectoria D, E].

--Fin trayectoria principal.

Trayectoria alternativa A.

1. El sistema verifica que no hay Museos registrados.
2. El sistema muestra una lista vacía. [Trayectoria B].

--Fin de trayectoria alternativa A.

Trayectoria alternativa B.

1. El administrador presiona de pantalla “Gestionar Museos” el botón de Crear.
2. El administrador introduce los datos del Museo.
3. El administrador presiona el botón Finalizar.
4. El sistema verifica que los datos son correctos. [Trayectoria C].
5. El sistema muestra el mensaje “ Museo Creado”
6. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa B.

Trayectoria alternativa C.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “¡Error!”.
3. Volver al paso 2 de la trayectoria alternativa B.

--Fin de trayectoria alternativa C.

Trayectoria alternativa D.

1. El administrador presiona el botón Eliminar.
2. El sistema muestra el mensaje “ Museo eliminado”
3. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa D.

Trayectoria alternativa E.

1. El administrador presiona el botón Detalles.
2. El sistema verifica los datos registrados del Museo.
3. El sistema muestra la pantalla “Detalle de Museo”. [Trayectoria F].
4. El administrador presiona el botón Regresar.
5. Volver al paso 2 de trayectoria principal.

--Fin de trayectoria alternativa E.

Trayectoria alternativa F.

1. El administrador presiona de pantalla “Detalle de Museo” de la sección de Museo el botón de Editar.
2. El sistema muestra la pantalla emergente para introducir los datos.
3. El administrador introduce los datos del Museo.
4. El administrador presionara el botón Finalizar.
5. El sistema verifica que los datos son correctos. [Trayectoria G]
6. El sistema muestra el mensaje “ Museo Editado”
7. Volver al paso 2 de trayectoria alternativa E.

--Fin de trayectoria alternativa F.

Trayectoria alternativa G.

1. El sistema detecta que hay campos vacíos en el formulario.
2. El sistema muestra el mensaje “¡Error!”.
3. Volver al paso 3 de la trayectoria F.

--Fin de trayectoria alternativa G.

5.7. Diagrama de base de datos

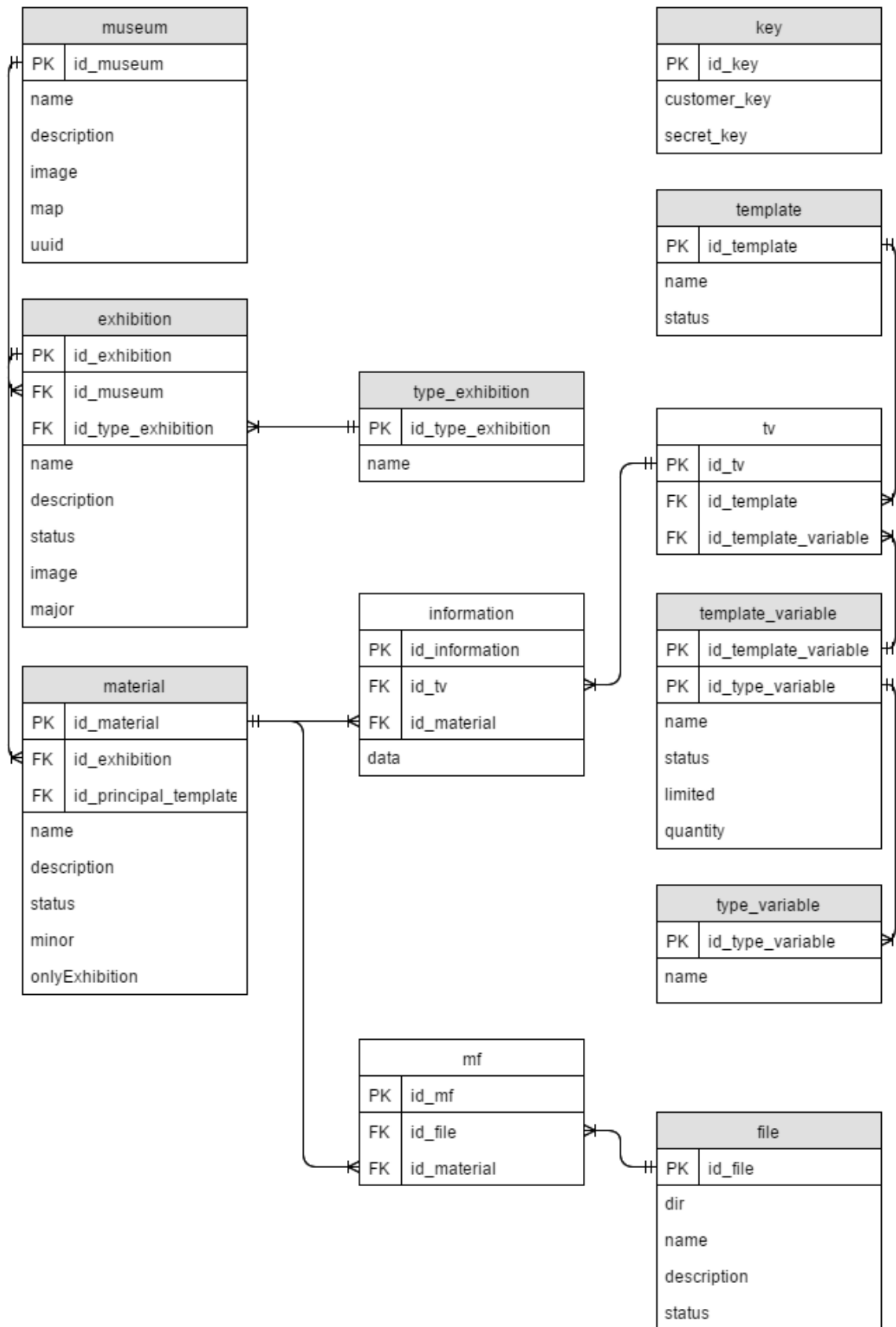


Figura 65 Diagrama de base de datos

5.7.1. Descripción de entidades

5.7.1.1. Museum

- Name
 - TEXT NOT NULL
- Description
 - TEXT NOT NULL
- Image
 - TEXT
- map
 - TEXT
- uuid
 - TEXT

5.7.1.2. Material

- Id_material (Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Id_principal_template (Llave foránea)
 - INT
- Id_exhibition (Llave foránea)
 - INT
- Name
 - TEXT NOT NULL
- Description
 - TEXT NOT NULL
- Status
 - TINYINT
- Minor
 - SMALLINT

5.7.1.3. Exhibition

- Id_exhibition (Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Id_museum (Llave foránea)
 - INT
- Id_type_exihition (Llave foránea)
 - INT
- Name
 - TEXT NOT NULL
- Description
 - TEXT NOT NULL
- Status
 - TINYINT
- Image
 - TEXT

- Major
 - SMALLINT

5.7.1.4. Type_exhibition

- id_type_exhibition (Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Name
 - TEXT NOT NULL

5.7.1.5. Template

- id_template (Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Name
 - TEXT NOT NULL
- Status
 - TINYINT

5.7.1.6. Template_variable

- id_template_variable (Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Name
 - TEXT NOT NULL
- Status
 - TINYINT
- Limited
 - TINYINT
- Quantity
 - TINYINT

5.7.1.7. File

- Id_file(Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Name
 - TEXT NOT NULL
- Description
 - TEXT NOT NULL
- Status
 - TINYINT
- Dir
 - TEXT NOT NULL

5.7.1.8. Key

- Id_key(Llave primaria)
 - INT PRIMARY KEY AUTO_INCREMENT
- Customer_key
 - TEXT NOT NULL
- Secret_key
 - TEXT NOT NULL

5.7.1.9. Type_variable

- Id_type_variable
 - INT PRIMARY KEY AUTO_INCREMENT
- Name
 - TEXT NOT NULL

5.8. Diagramas de clases

Descripción de módulos

- **Modelo**

Contiene las clases que tienen el control de la información con la cual el sistema opera, todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso.

- **Vista**

Contiene las clases que presentan el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

- **Controlador**

Contiene las clases que responden a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.

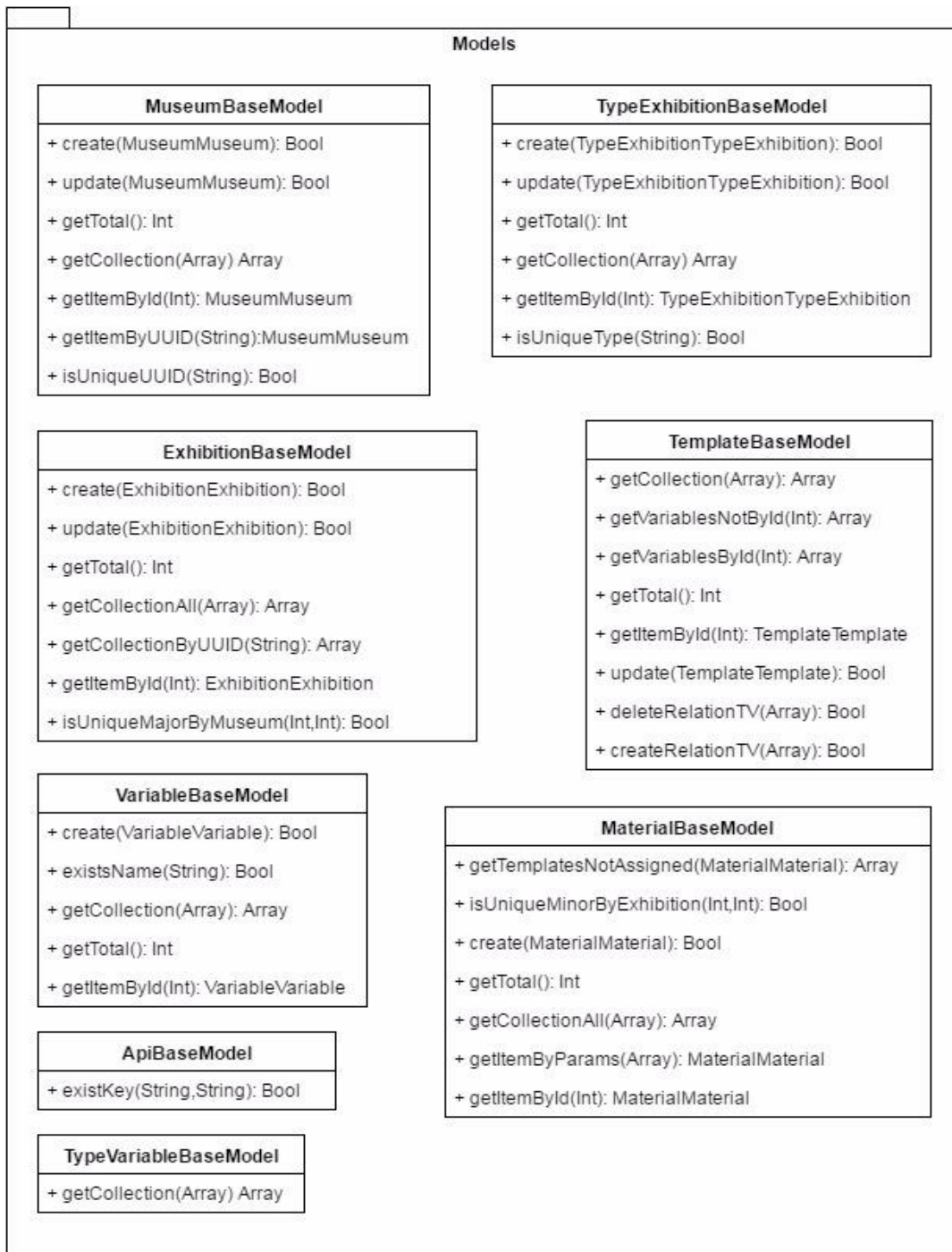


Figura 66 Diagrama de clases servidor web (Modelo)

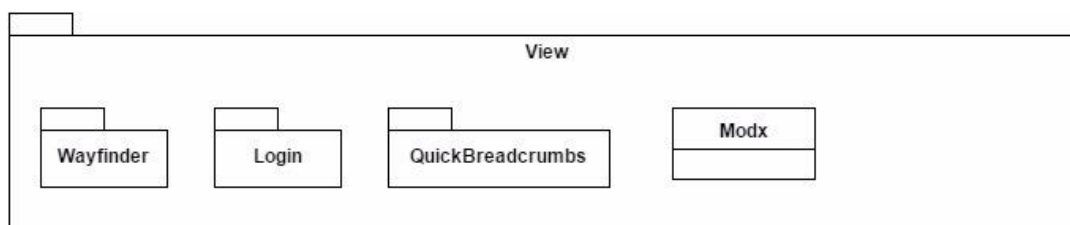


Figura 67 Diagrama de clases servidor web (Vista)

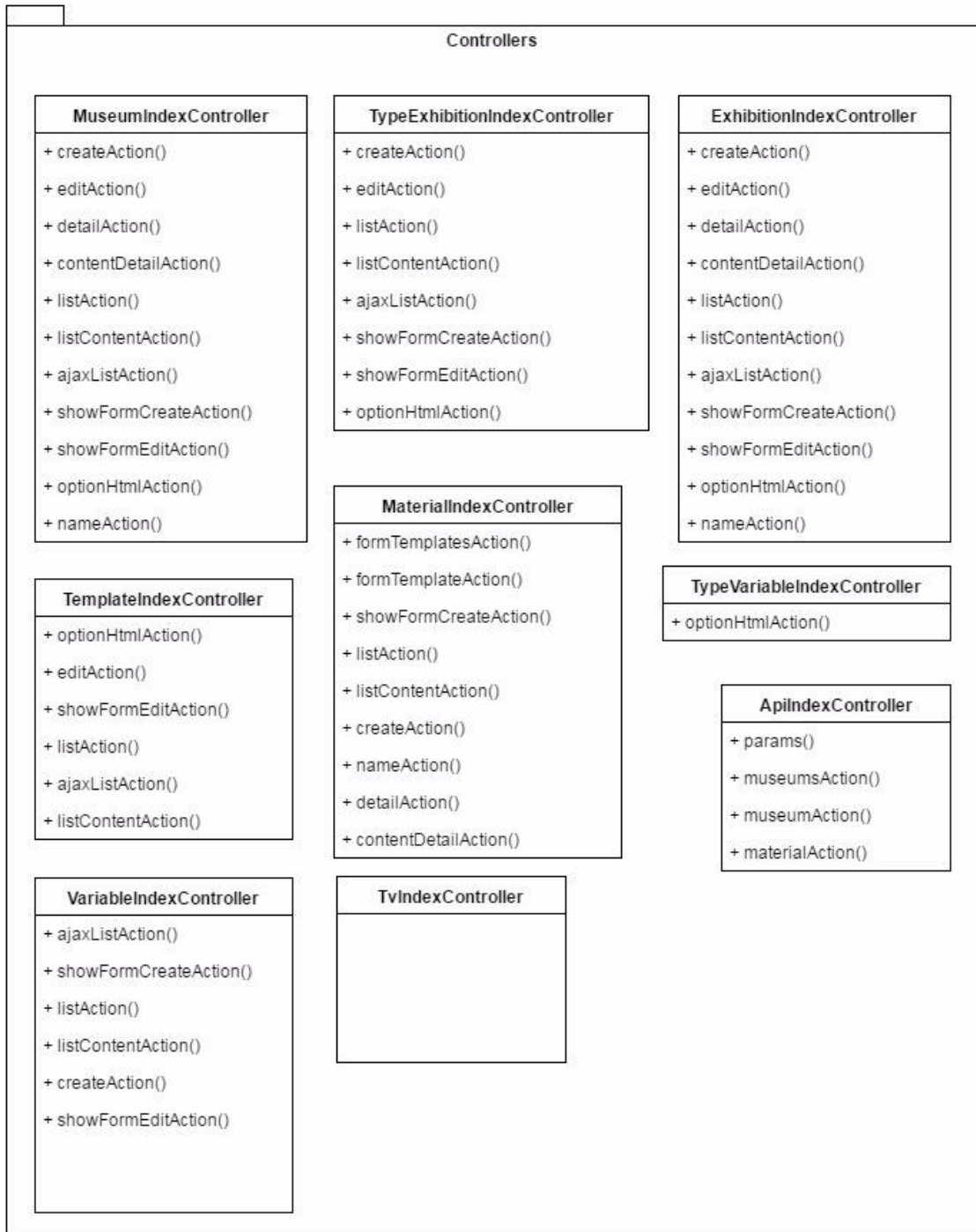


Figura 68 Diagrama de clases servidor web (Controlador)

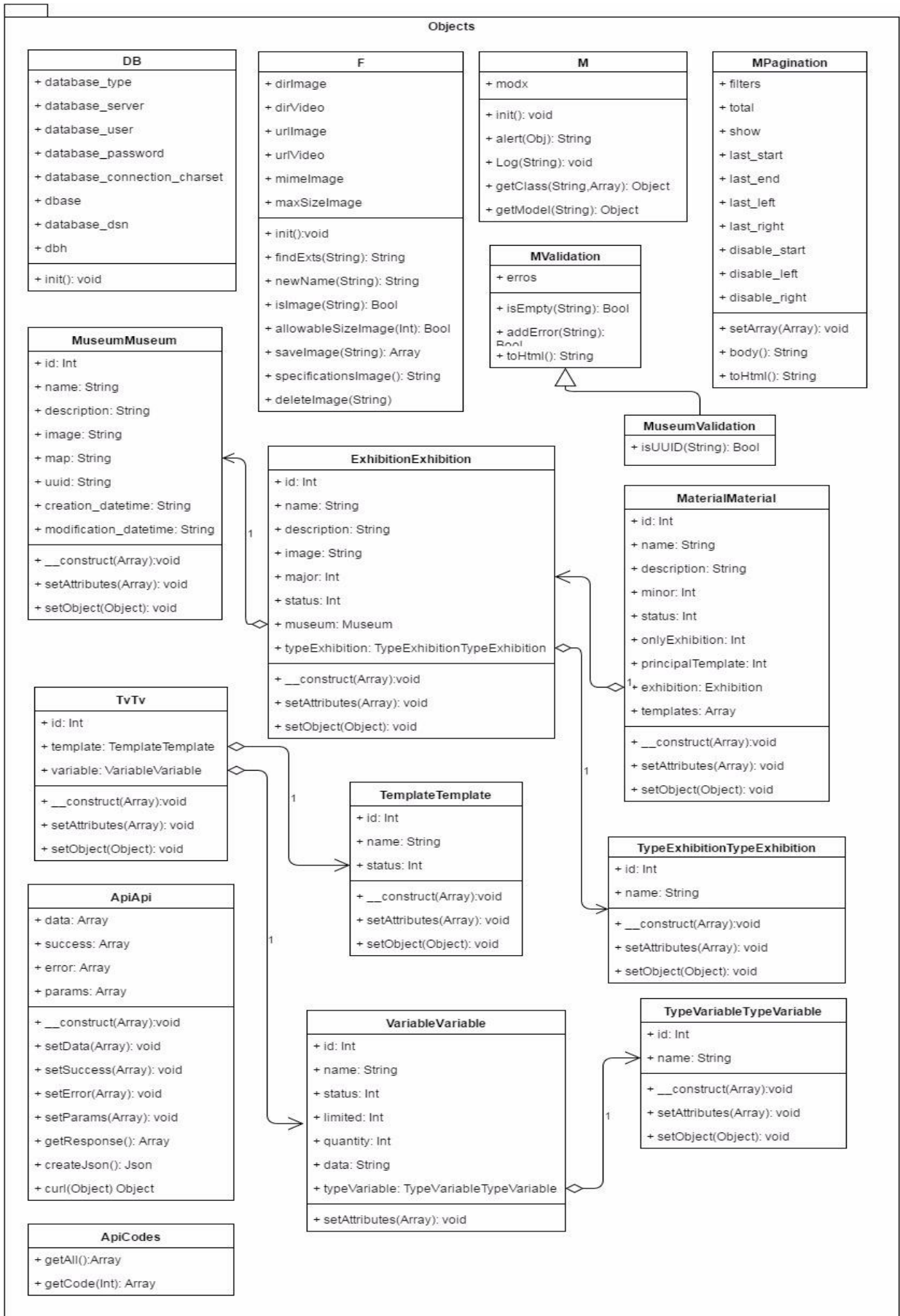


Figura 69 Diagrama de clases servidor web (Objetos)

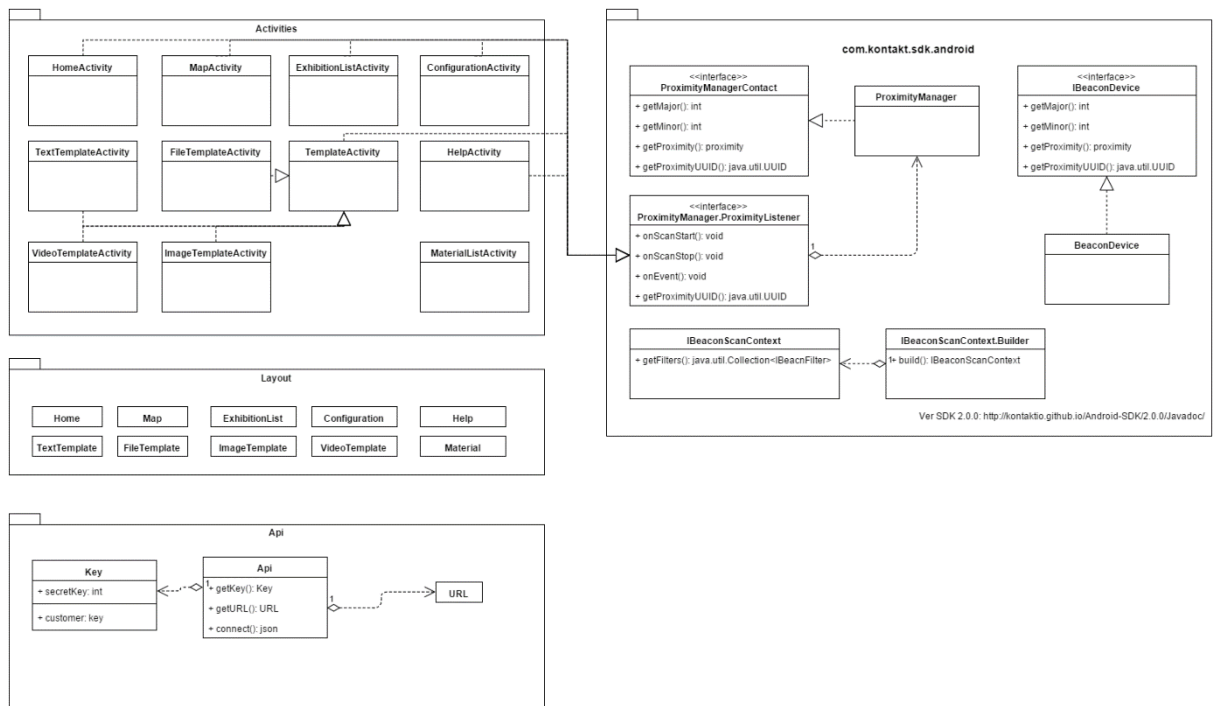


Figura 70 Diagrama de clases aplicación móvil

5.9. Diagramas de secuencia

Los diagramas de secuencia que a continuación se describen, hacen referencia a los casos de uso anteriormente descritos.

CU1: Visualizar home

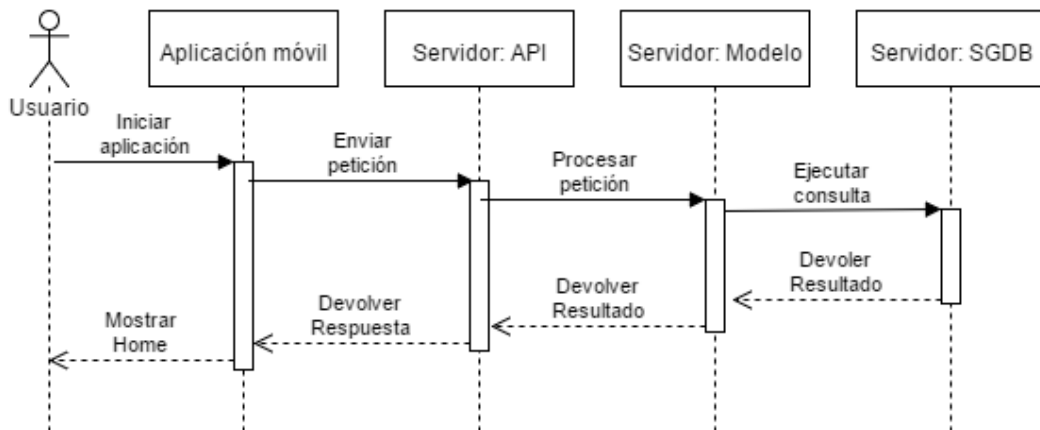


Figura 71 Diagrama de secuencia visualizar home

CU2 Visualizar exhibición

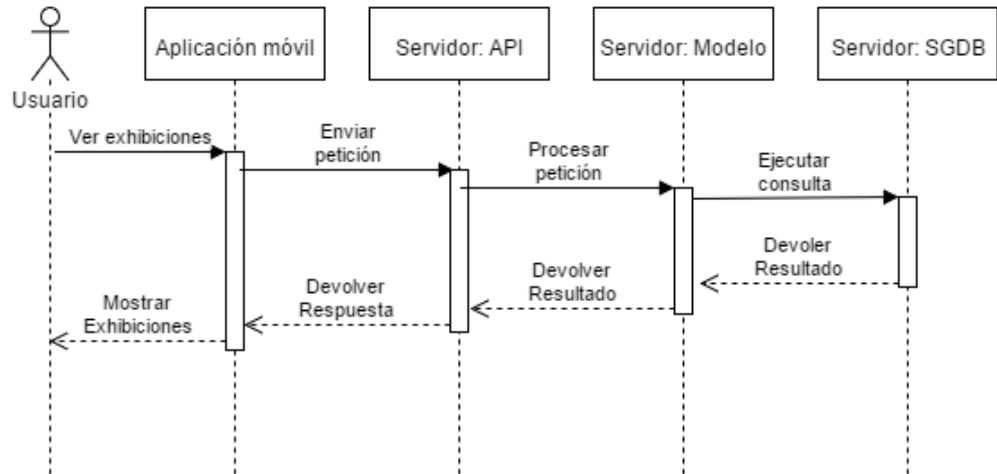


Figura 72 Diagrama de secuencia Visualizar exhibición

CU3 Visualizar mapa

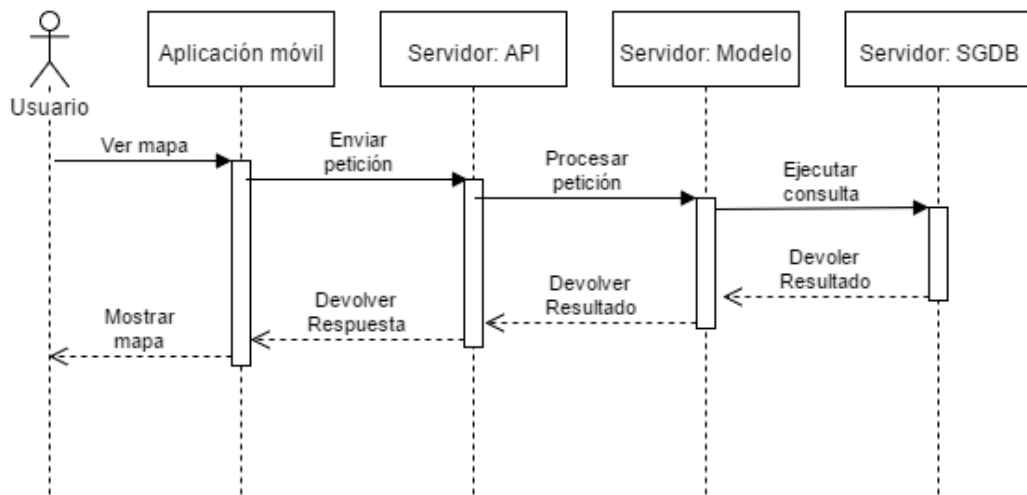


Figura 73 Diagrama de secuencia visualizar mapa

CU4 Configurar aplicación

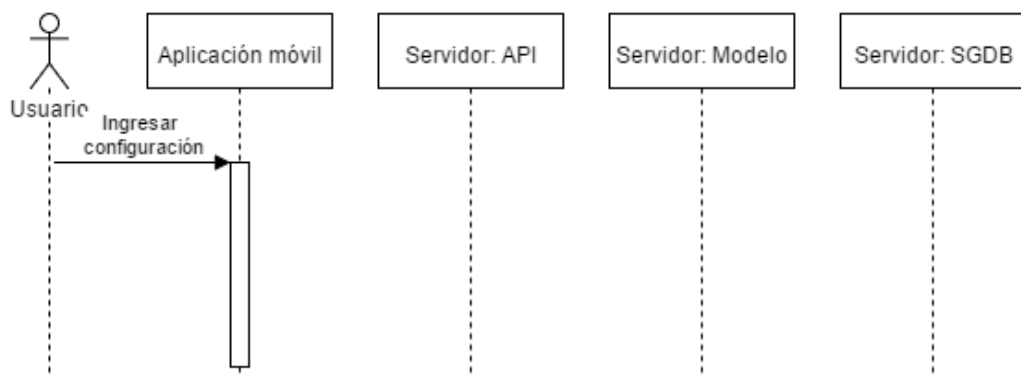


Figura 74 Diagrama de secuencia configurar aplicación

CU5 Visualizar ayuda

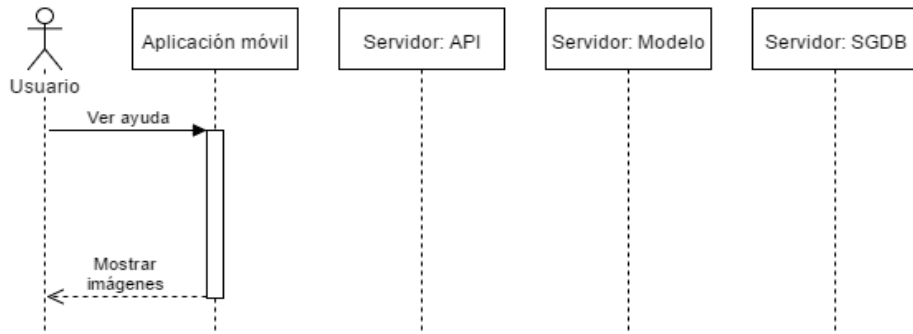


Figura 75 Diagrama de secuencia visualizar ayuda

CU6 Ver lista de material expuesto y exhibiciones

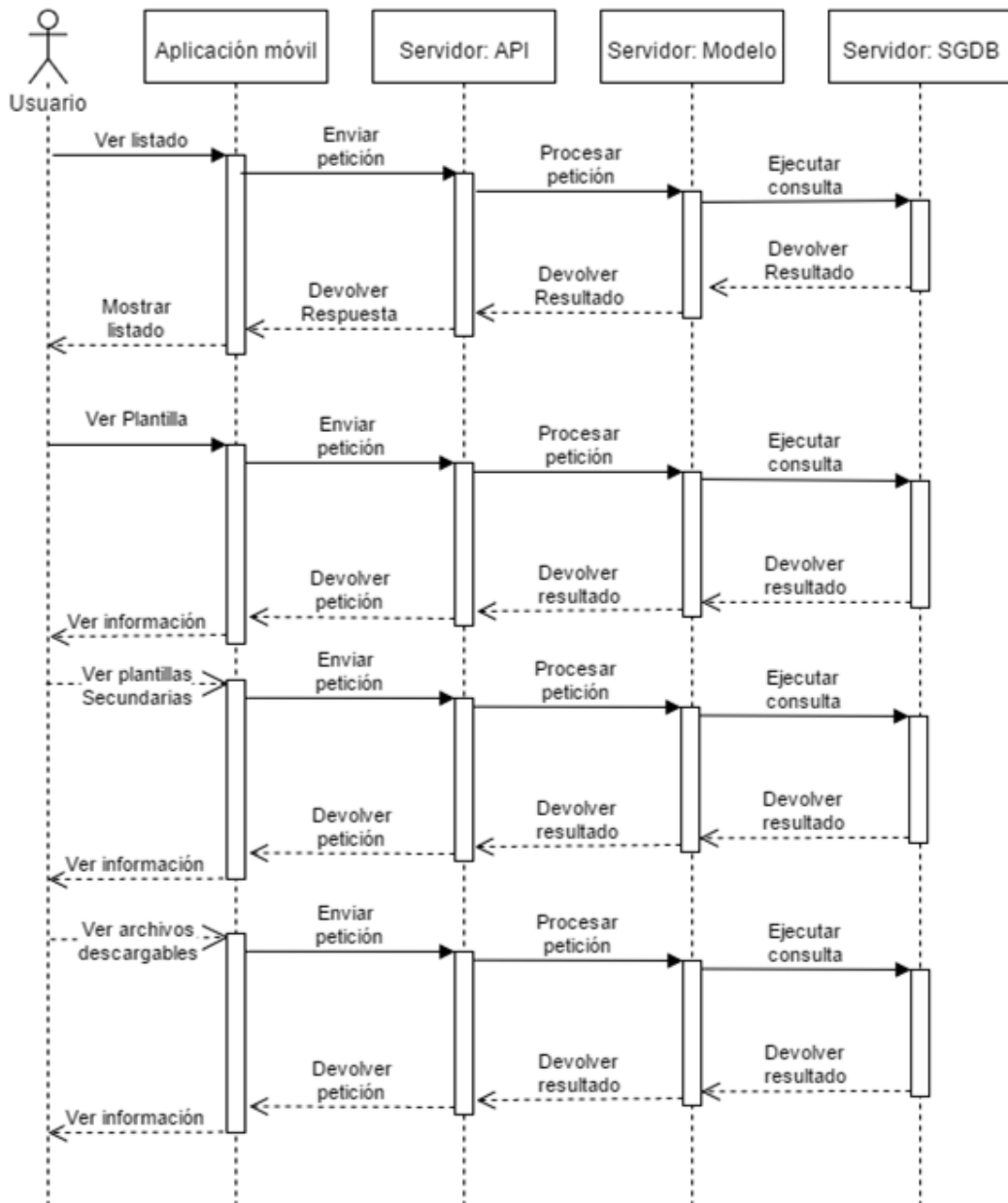


Figura 76 Diagrama de secuencia ver lista de material expuesto y exhibiciones

CU7 Ver notificaciones

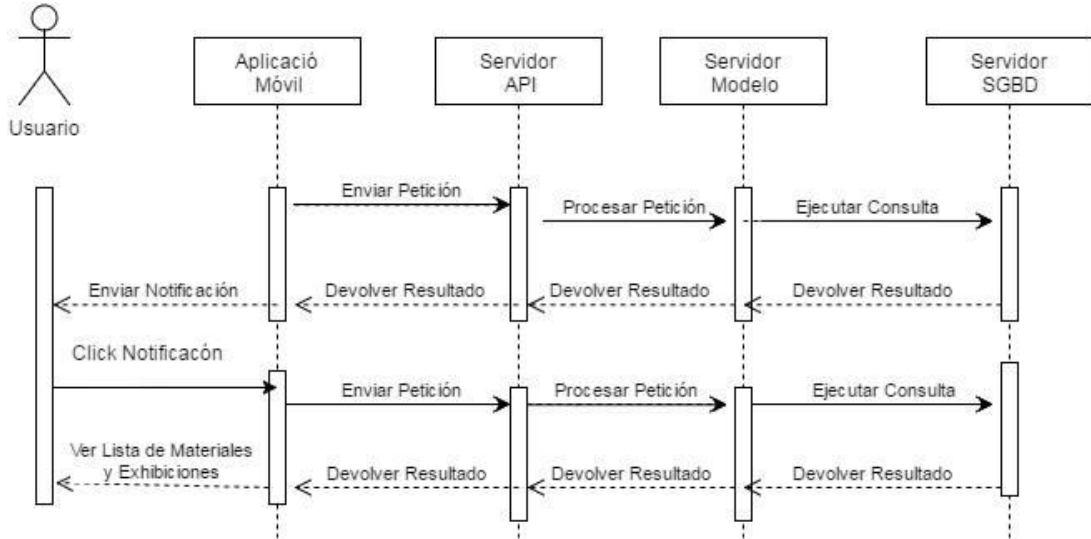


Figura 77 Diagrama de secuencia ver notificaciones

CU8 Iniciar sesión

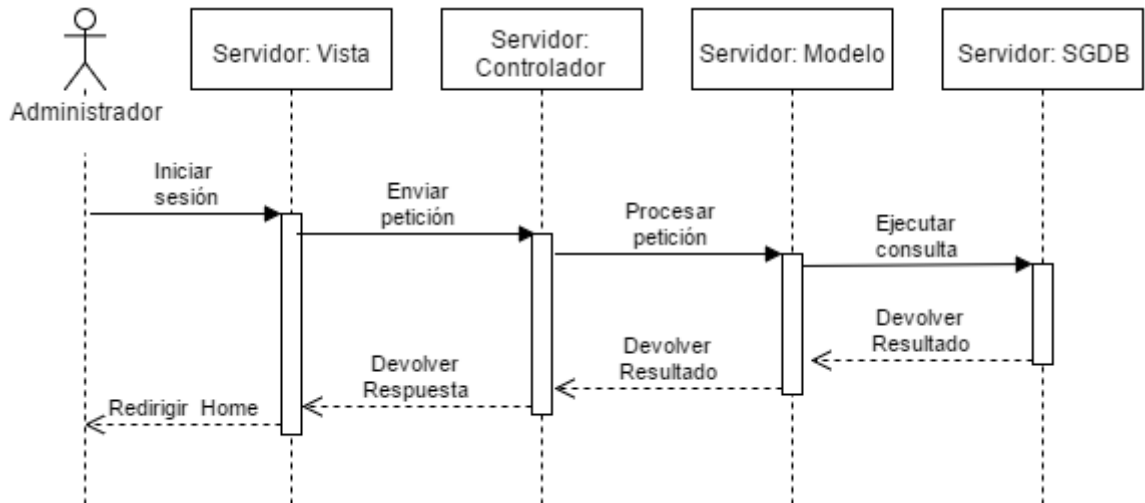


Figura 78 Diagrama de secuencia iniciar sesión

CU9 Gestión de material

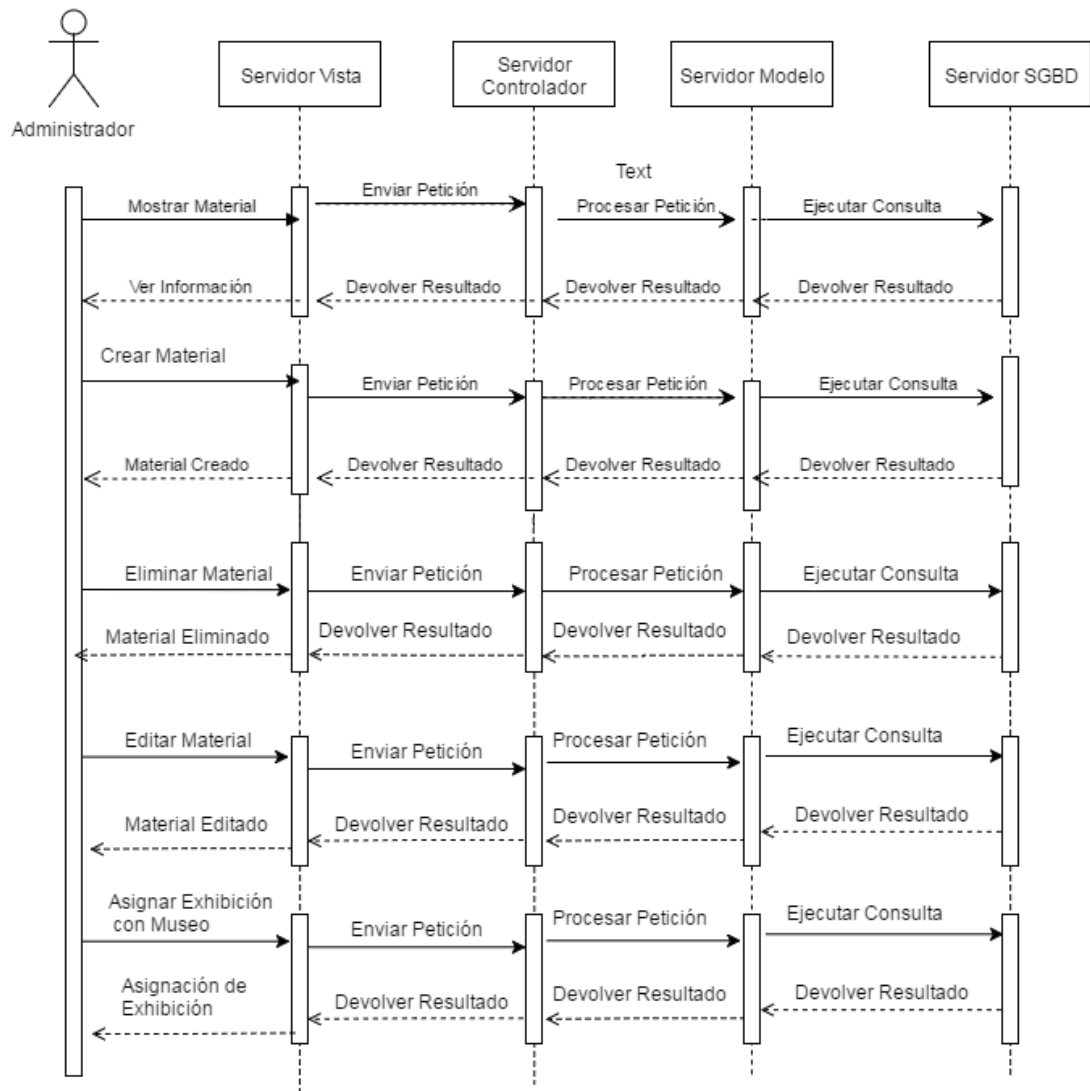


Figura 79 Diagrama de secuencia gestión de material

CU10 Gestión de exhibición

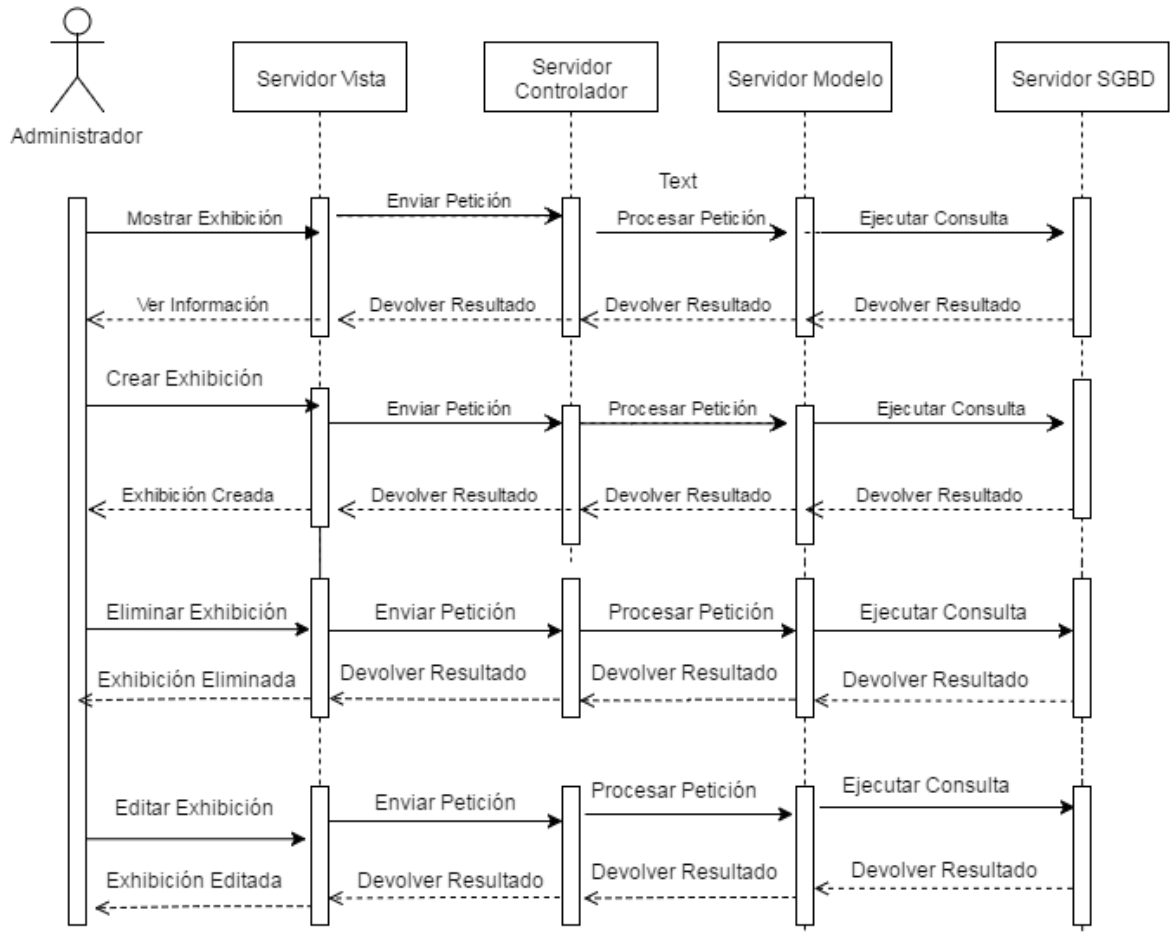


Figura 80 Diagrama de secuencia gestión de exhibición

CU11 Gestión de archivos

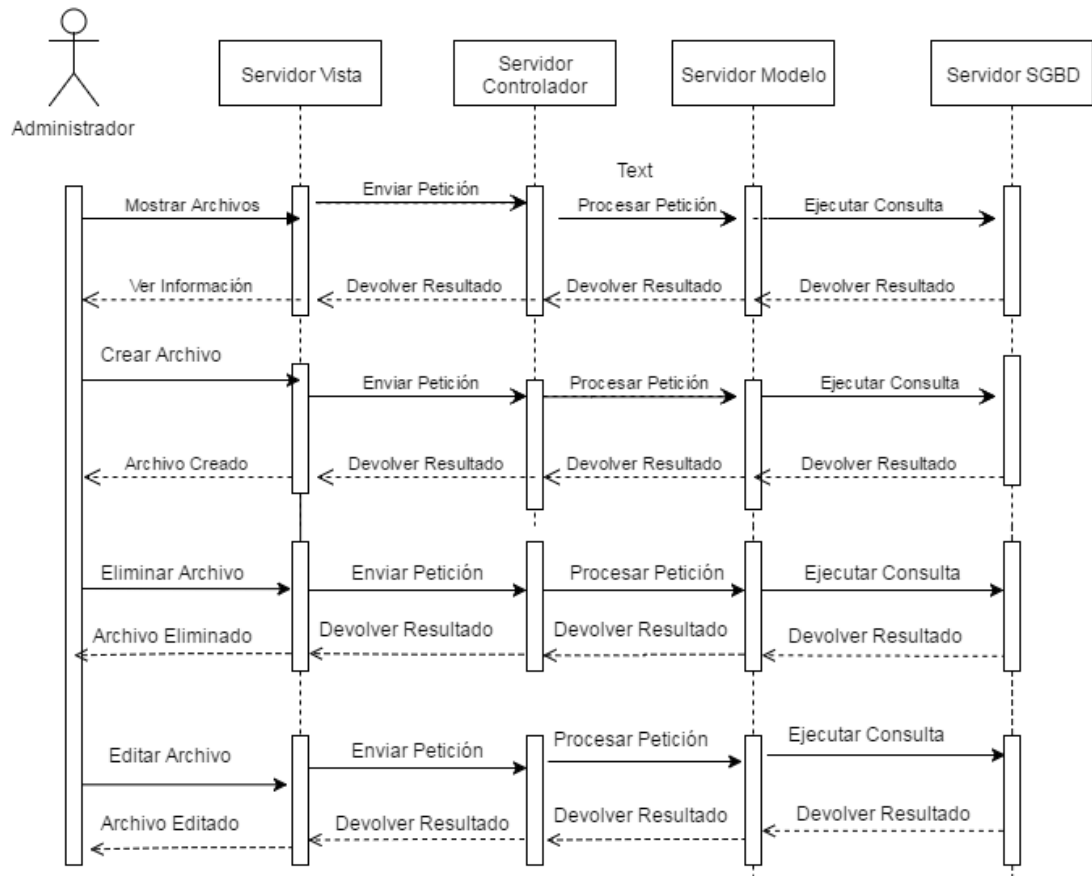


Figura 81 Diagrama de secuencia gestión de archivos

CU12 Gestión de plantillas

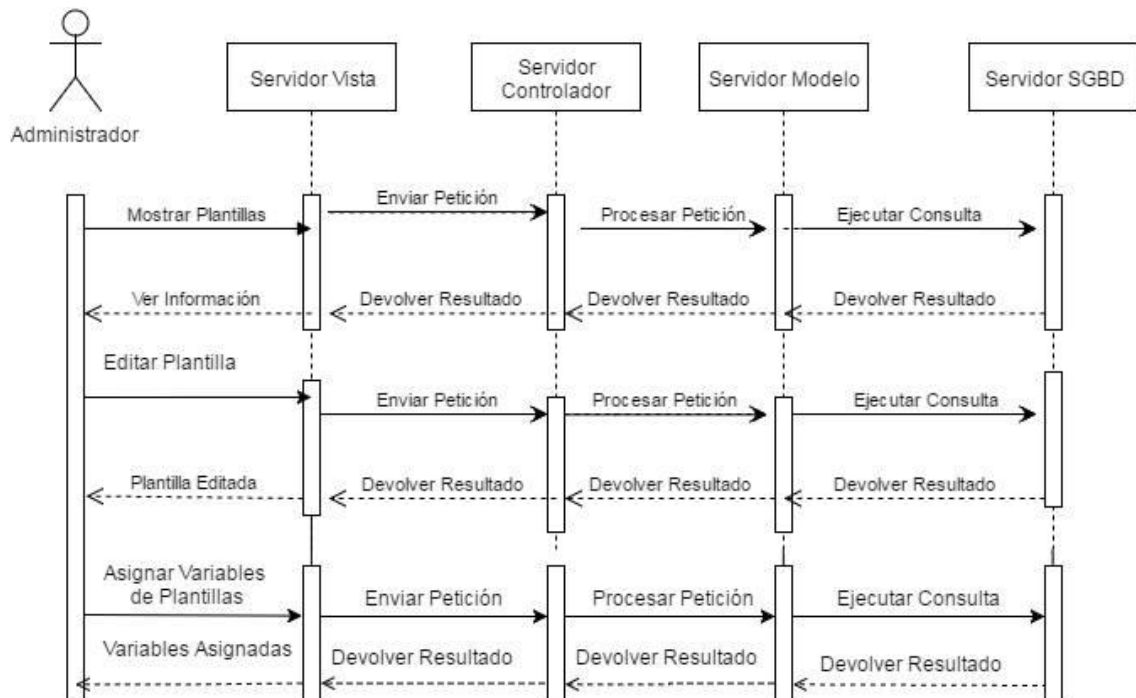


Figura 82 Diagrama de secuencia gestión de plantillas

CU13 Gestión de variable

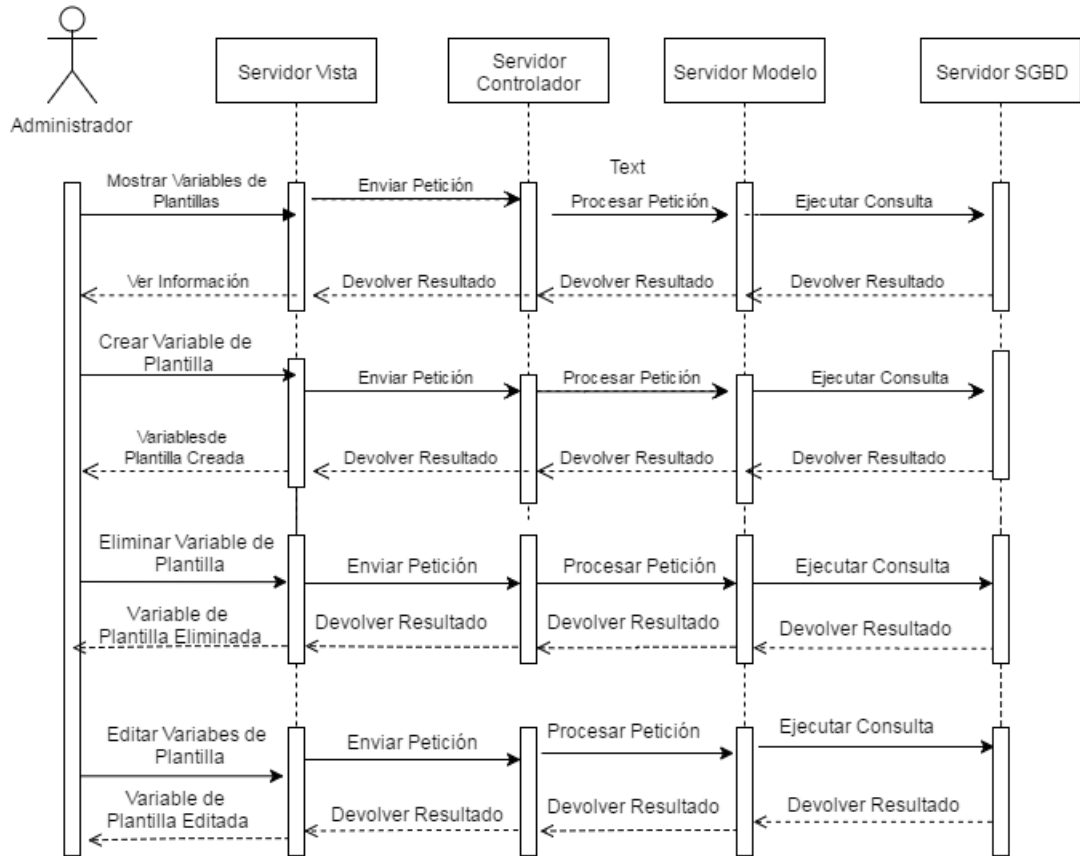


Figura 83 Diagrama de secuencia gestión de variable

CU14 Recuperar contraseña

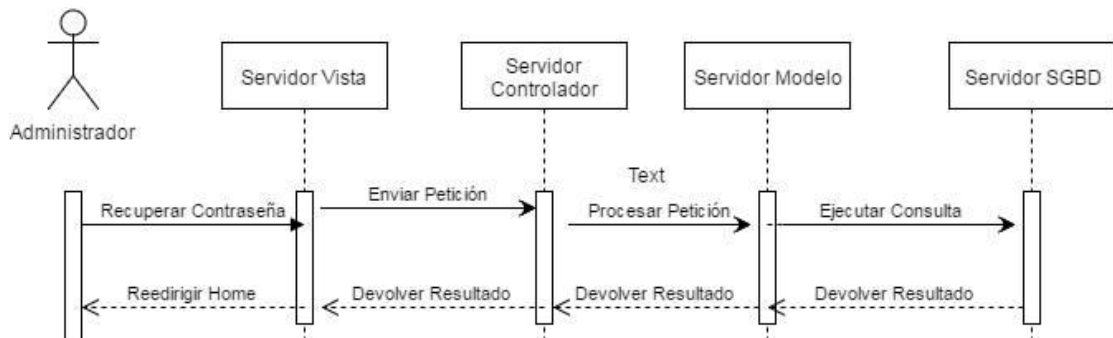


Figura 84 Diagrama de secuencia recuperar contraseña

CU15 Gestionar Museos

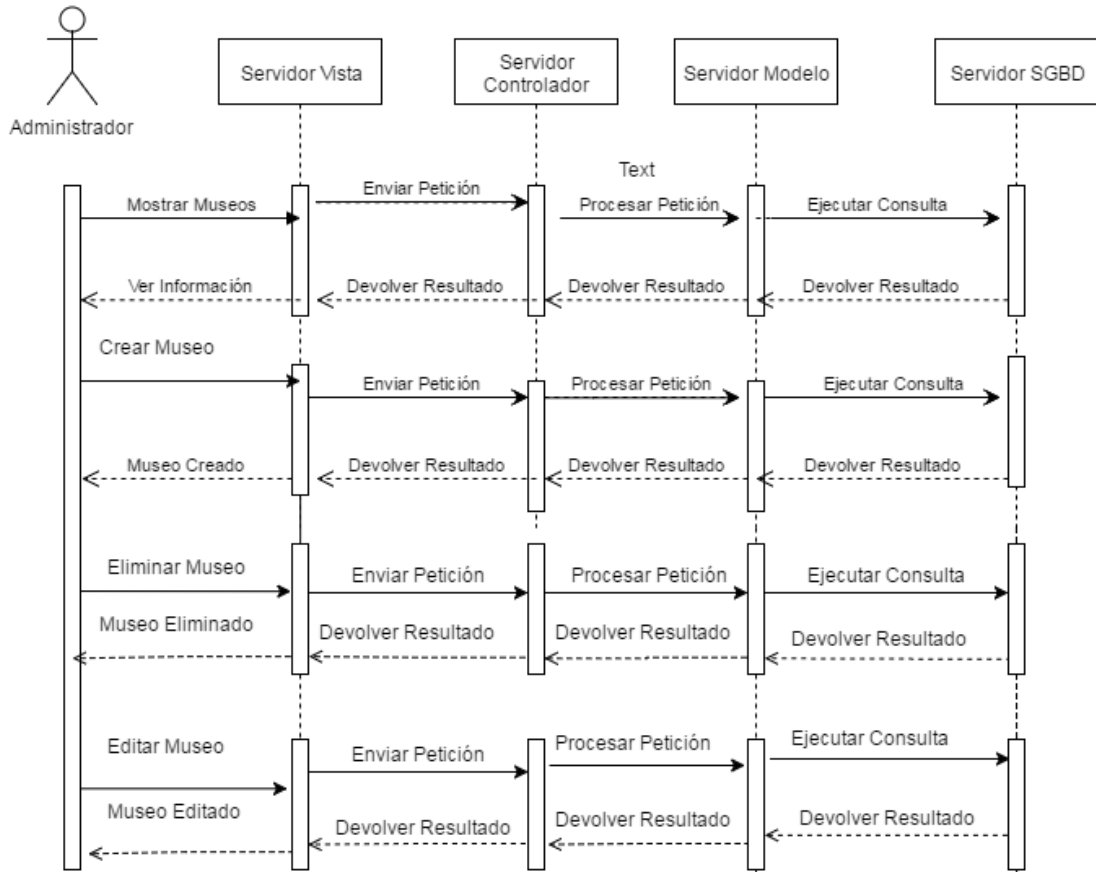


Figura 85 Diagrama de secuencia gestionar museo

CAPÍTULO 6. DESARROLLO

En esta sección se describe el desarrollo del sistema, el cual está integrado en tres módulos, la aplicación móvil, el Servidor web y por último la API para la conexión de la aplicación con el Servidor Web.

6.1. Desarrollo aplicación móvil

El desarrollo de la aplicación móvil consistió en los siguientes pasos.

6.1.1. Descarga e instalación Android Studio

Para poder desarrollar la aplicación móvil, necesitamos instalar el IDE de trabajo, el cual es Android Studio. Para ello se descargó de su página oficial, con el siguiente enlace: <http://developer.android.com/intl/es/sdk/index.html>, se instaló la versión 1.5.1 y se cuenta con el JDK 7 para así desarrollar para versiones Android 6.0 hacia más bajo.

6.1.2. Configuración del entorno

Creamos un proyecto en Android Studio y especificamos la versión mínima de Android que requerirá nuestra aplicación para funcionar, en este caso será la API 18: Android 4.3 Jelly Bean.

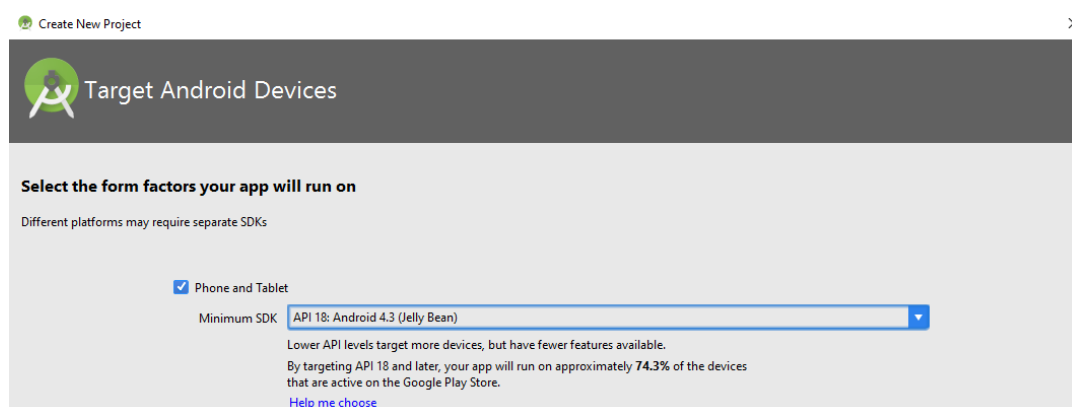


Figura 86 Configuración del entorno

6.1.3. Actividad principal

Para iniciar el desarrollo de la aplicación, se tomó como base la *Navigation Drawer Activity*, ya que esta se acopló más a nuestra idea del modelado que deseamos. Ya que cuenta con un menú lateral para así acceder a las opciones que dispondremos más adelante, además, cuenta con una barra y ahí tendremos las configuraciones y un botón flotante para asignarle una tarea más.

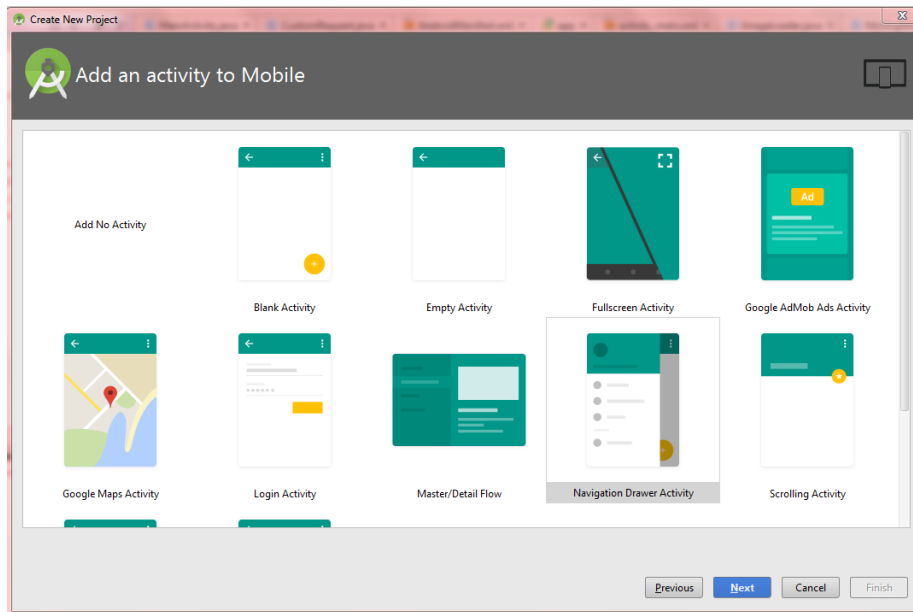


Figura 87 Navigation Drawer Activity

6.1.3.1. Barra de navegación

Al tener la Actividad base se fue diseñando, primeramente la barra de navegación, se le asignaron títulos e imágenes a cada opción.

Se contó primeramente con seis opciones en el menú, pero mediante se fue avanzando en el proyecto, se descartaron dos y una se movió una al botón flotante que contaba la pantalla principal. Se descartó la opción Salas porque se integró en la parte de Exhibiciones para que la información este más completa en esa pestaña, y la opción de compartir por el momento no se tomó en cuenta porque no está planeado agregar la aplicación a la tienda de Google llamada Play Store. La opción que se definió ahora en el botón flotante fue la de ayuda, ya que estará disponible en cualquier pestaña para que el usuario ingrese con mayor facilidad.

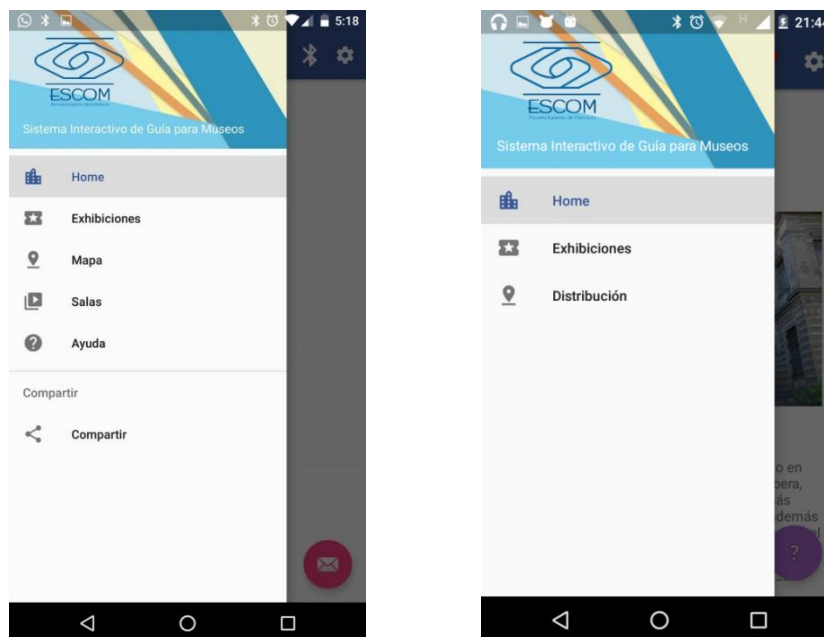


Figura 88 Desarrollo de la Barra de navegación

6.1.3.2. Barra de menú

En esta sección agregamos los iconos de configuración y *bluetooth*, en la primera como dice su nombre, se puede acceder a la configuración de la aplicación y la segunda muestra la lista de materiales expuestos y exhibiciones que se encontraron.

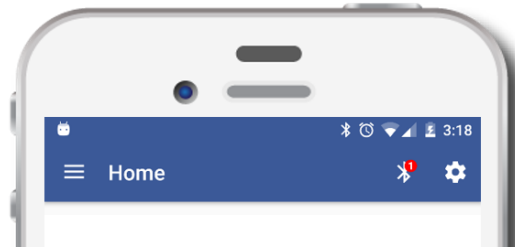


Figura 89 Barra de menú

6.1.4. Verificación de la conexión a Internet

Cuando la aplicación se ejecuta o se navega dentro de ella, verifica la conexión a internet (WiFi y Redes móviles) mediante una función que obtiene el estado de la clase *NetworkInfo*, sin una conexión a internet la aplicación no podrá mandar los datos obtenidos de los **Beacon**, y no podrá tampoco recibir datos del servidor web.

Si no detecta ninguna conexión a internet lanzara dos mensajes con lo siguiente:

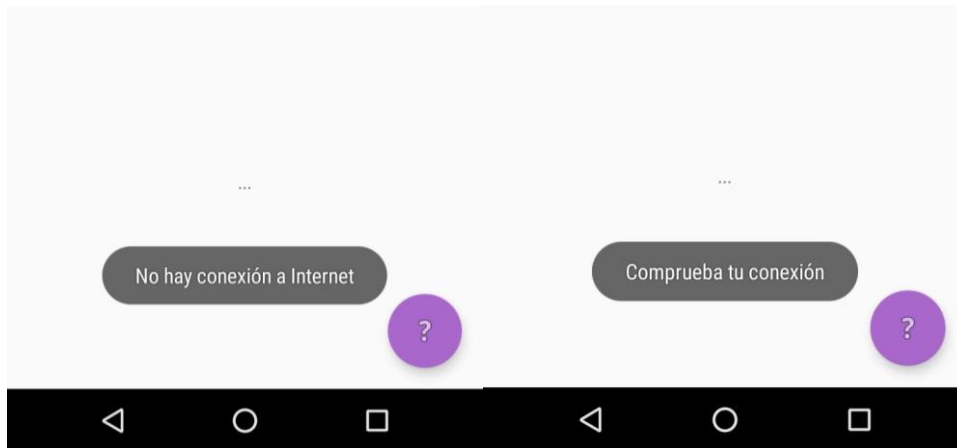


Figura 90 Verificación de la conexión a internet

6.1.5. Verifica estado del bluetooth

Se implementó una función la cual enciende el *bluetooth* cuando la aplicación se ejecuta. Esta función verifica el estado del mismo, si esta desactivado y recibe un valor de tipo boolean verdadero, esta procederá a activar el *bluetooth*, si recibe el valor verdadero y el *bluetooth* ya está activado simplemente mantendrá el mismo estado.

```

public static boolean enciendeBluetooth(boolean enable) {
    BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    boolean isEnabled = bluetoothAdapter.isEnabled();
    if (enable && !isEnabled) {
        return bluetoothAdapter.enable();
    }
    else if(!enable && isEnabled) {
        return bluetoothAdapter.disable();
    }
    // No se necesita cambiar el estado
    return true;
}

```

Figura 91 Verifica bluetooth

6.1.6. Icono de la aplicación

El diseño del icono de la aplicación se compone de imágenes representativas a la temática del trabajo terminal, durante el diseño de este obtuvimos tres imágenes candidatas y la última fue la de nuestro mayor agrado.



Figura 92 Icono de la aplicación

6.1.7. ANDROID MANIFEST

Las aplicaciones desarrolladas para la plataforma Android, deben contar con un archivo AndroidManifest.xml en su directorio raíz, en él se encuentran definidas las configuraciones del proyecto como actividades, intents, permisos, entre otros.

Este archivo es generado automáticamente por Android Studio al crear el proyecto.

6.1.7.1. Permisos

Para que la aplicación pueda tener acceso al uso del *bluetooth*, al internet y para almacenar datos es necesario escribir los siguientes permisos


```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

Figura 93 Permisos

Para la versión de Android 6 Marshmallow (SDK 23) es necesario declarar dos permisos más para poder detectar los **Beacon**.

```

<uses-permission-sdk-23 android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission-sdk-23 android:name="android.permission.ACCESS_FINE_LOCATION"/>

```

Figura 94 Permisos especiales Android 6

6.1.8. Lectura de datos del Beacon

Para la lectura de los datos propios de los dispositivos **Beacon** hicimos uso de la librería AltBeacon.

6.1.8.1. Android Beacon Library (AltBeacon)

Android Beacon Library proporciona la API para poder interactuar con los **Beacon** y poder obtener datos de ellos.

- **Agregando AltBeacon al proyecto**

Configurar el archivo build.grade de la app. Añadiendo la librería como dependencia como se especifica a continuación.

```

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile 'org.altbeacon:android-beacon-library:2+'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
    compile 'com.android.support:support-v4:23.1.1'
}

```

Figura 95 Agregando Altbeacon

- **Implementación de AltBeacon**

La aplicación móvil para recibir información de cualquier **Beacon**, debe generar una instancia de la clase **BeaconManager** dentro de la Actividad donde hará la detección de los **Beacon**.

```
beaconManager = BeaconManager.getInstanceForApplication(this);
```

Además para que sea capaz de detectar dispositivos de diferentes proveedores distintos a **AltBeacon**, se debe especificar lo siguiente para el caso de los **Beacon** de *Kontakt*.

```
beaconManager.getBeaconParsers().add(new BeaconParser().setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24,d:25-25"));
```

[38]

La clase *BeaconManager* la utilizamos para configurar la interacción de los **Beacon** en la Actividad.

Implementamos la interfaz *BeaconConsumer*, la cual mediante un *callback* nos dice cuando el Servicio está listo para hacer uso de él, sin este *callback* no será posible hacer la detección de los dispositivos **Beacon**.

Una vez listo el servicio llamamos al método **onBeaconServiceConnect()** el cual recibe las instrucciones a través de la instancia *beaconManager*.

Al tener permiso para acceder al *Bluetooth* dentro de la aplicación móvil, hace uso del método **onBeaconServiceConnect**, para así poder obtener datos de los **Beacon** y así mandar al servidor web mediante una petición *POST* el identificador *uuid* que leemos de los **Beacon**.

El siguiente código cuenta el número de dispositivos que ha encontrado.

```
@Override
public void onBeaconServiceConnect() {
    beaconManager.setMonitorNotifier(new MonitorNotifier() {
        @Override
        public void didEnterRegion(Region region) {
            Log.i(TAG, "Se encontro un beacon por primera vez");
        }
        @Override
        public void didExitRegion(Region region) {
            Log.i(TAG, "No se encuentra ningun Beacon");
        }
        @Override
        public void didDetermineStateForRegion(int state, Region region) {
            Log.i(TAG, "Determinando estado del beacon: " + state);
        }
    });

    try {
        beaconManager.startMonitoringBeaconsInRegion(new Region("RegionBeacons", null, null, null));
    } catch (RemoteException e) { }
    beaconManager.setRangeNotifier(new RangeNotifier() {
        @Override
        public void didRangeBeaconsInRegion(final Collection<Beacon> rangedBeacons, Region region) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    statusView.setText("Beacons encontrados : " + rangedBeacons.size());
                    beaconAdapter.replaceWith(rangedBeacons);
                }
            });
        }
    });
    try {
        beaconManager.startRangingBeaconsInRegion(new Region("RegionBeacons", null, null, null));
    } catch (RemoteException e) { }
}
}
```

Figura 96 Código obtener información del Beacon

Definimos una sola región para la lectura de los **Beacon**, es decir solo podrá leer los dispositivos con el *UUID* que asignamos en esa región.

```
region = new Region("regionId", Identifier.parse("252ea4f4-a959-4512-b43c-7edc533c70b0"),
    null, null);
```

Figura 97 Región Beacon

6.1.9. Librería Volley

Es una librería que tiene como objetivo hacer las llamadas a recursos online más fáciles y rápidas, sin necesidad de escribir gran cantidad de código. Las llamadas realizadas con Volley son asíncronas y algunas de las características de Volley destacan:

1. Gestión de prioridades y de la cola de peticiones.
2. Gestión de memoria y caché.
3. Cancelación de peticiones.

Se debe añadir los permisos en el `AndroidManifest` para tener conexión a Internet, como se mostró anteriormente.

6.1.9.1. Petición JSON (`JsonObjectRequest`, `JSONArrayRequest`)

JSON o Notación de Objetos de *Javascript* (*JavaScript Object Notation*) es un formato ligero de intercambio de datos y está constituido por dos estructuras: la primera es siendo un objeto que es una colección de pares de nombre y valor, la segunda es un arreglo con una lista ordenada de valores.

Volley ofrece una forma sencilla para hacer peticiones *JSON*, si se requiere obtener un objeto de tipo *JSON* en la respuesta se debe usar la clase `JsonObjectRequest` o si la respuesta es un arreglo se debe usar la clase `JSONArrayRequest`.

El uso de Volley comienza en la creación de una cola de peticiones, este nos servirá para gestionar automáticamente el envío de las peticiones, la administración de los hilos, gestión de cache y la publicación de los resultados.

```
final RequestQueue queue = Volley.newRequestQueue (this);
```

6.1.9.2. Generar y añadir peticiones GET y POST

Para poder realizar la petición del tipo `JsonObjectRequest` tenemos que crear un constructor, ya que el que tiene por defecto, al meter los parámetros para el método *POST* no envía ni recibe los parámetros, para resolver esto tenemos que construir una clase con el constructor para tener el método *POST* personalizado.

```

public CustomRequest(String url, Map<String, String> params,
                    Response.Listener<JSONObject> reponseListener, Response.ErrorListener
                    errorListener) {
    super(Method.GET, url, errorListener);
    this.listener = reponseListener;
    this.params = params;
}

public CustomRequest(int method, String url, Map<String, String> params,
                    Response.Listener<JSONObject> reponseListener, Response.ErrorListener
                    errorListener) {
    super(method, url, errorListener);
    this.listener = reponseListener;
    this.params = params;
}

protected Map<String, String> getParams()
    throws com.android.volley.AuthFailureError {
    return params;
};

```

Figura 98 Constructor de peticiones tipo *JsonObjectRequest*

Ahora mandaremos los parámetros para poder acceder al servidor, ya que implementamos unas llaves para tener un control de accesos y esto lo realizamos con el método *getParams()* el cual devuelve la lista de parámetros a enviar en formato clave-valor.

```

protected Map<String,String> getParams() throws AuthFailureError {
    Map <String,String> params = new HashMap<String,String>();
    params.put("secret_key", "H0oh9RjigoKtPeU1gkbZLTikkx90WaBUBE4toH1DYQOkByz1jK");
    params.put("customer_key", "eqEqDsAqnyjoZx6IjbCM73vNI");
    params.put("uuid", "252ea4f4-a959-4512-b43c-7edc533c70b0");
    return params;
};

```

Figura 99 Envío de parámetros al servidor

Para realizar peticiones y obtener imágenes, la librería incorpora un elemento llamado *NetworkImageView*, este nos ayuda a mostrar las imágenes que obtenemos a través de una URL.

6.1.9.3. Caché

La librería *Volley* incorpora un mecanismo para mantener caché de las peticiones y ahorrar un gran ancho de banda y reducir el tiempo de espera. Para ello tuvimos que crear una clase que extiende de *LruCache*.

```

public class BitmapLruCache
    extends LruCache<String, Bitmap>
    implements ImageLoader.ImageCache {

    public BitmapLruCache() { this(getDefaultLruCacheSize()); }

    public BitmapLruCache(int sizeInKiloBytes) { super(sizeInKiloBytes); }

    @Override
    protected int sizeOf(String key, Bitmap value) {
        return value.getRowBytes() * value.getHeight() / 1024;
    }

    @Override
    public Bitmap getBitmap(String url) { return get(url); }

    @Override
    public void putBitmap(String url, Bitmap bitmap) { put(url, bitmap); }

    public static int getDefaultLruCacheSize() {
        final int maxMemory =
            (int) (Runtime.getRuntime().maxMemory() / 1024);
        final int cacheSize = maxMemory / 8;

        return cacheSize;
    }
}

ImageLoader.ImageCache imageCache = new BitmapLruCache();
ImageLoader imageLoader = new ImageLoader(queue, imageCache);

NIV.setImageUrl(urlimage+jsonImage, imageLoader);

```

Figura 100 Almacenar imágenes en caché

6.1.10. Fragmentos

Para cada ítem de la barra de navegación (Home, Exhibiciones y Distribución), así como para la barra de menú en el icono de *bluetooth* (Materiales y exhibiciones) se crearon diferentes fragmentos, dentro de estos se realizan diferentes tareas.

6.1.10.1. Fragmento Home

Este fragmento es iniciado cada vez que la aplicación se abre o se selecciona en la barra de navegación el ítem de *home*, en él se envía la información requerida a través de la API al servidor y se recibe una petición de tipo *POST* con la información que se visualiza en los Fragmentos Home, Distribución y Exhibiciones.



Figura 101 Fragmento Home

6.1.10.2. Fragmento Exhibiciones y Distribución

Estos fragmentos son los encargados de visualizar la información que recibió de la petición a través del fragmento home y se muestra cada que el usuario haga clic sobre el ítem de la barra de navegación correspondiente.



Figura 102 Fragmento Exhibiciones



Figura 103 Fragmento de Distribución

6.1.10.3. Fragmento Materiales y Exhibiciones

Para poder acceder a este fragmento se le diseño un botón en la barra de menú, en el cual también aparece con número la cantidad de materiales que ha encontrado en ese momento, esto se logró haciendo uso de la clase *Drawable* por lo que nos permite crear gráficos haciendo uso de sus métodos.

Cada que se accede a la aplicación móvil, se hace la lectura de los **Beacon** y se envía información por cada uno de ellos a través de la API al servidor web, tal como: *UUID*, *Major*, *Minor* y la proximidad, además de las llaves (*secret key* y *customer key*), esto genera una respuesta, la cual recibe la aplicación móvil mediante una petición de tipo *POST*.

Cuando un nuevo **Beacon** es visible, es detectado por la aplicación móvil, esta realiza una petición con los valores obtenidos del dispositivo **Beacon** y la respuesta es visualizada en la aplicación móvil. En caso contrario de que deje de ser visible, solamente es eliminado de la lista.

Suponiendo que un **Beacon** cambia de proximidad, la aplicación móvil realiza una petición con los nuevos valores que ha leído y actualiza la lista con la información que ha traído de la petición.

Para poder visualizar la información creamos un adaptador, este nos proporciona una conexión entre el modelo de datos y la vista.



Figura 104 Fragmento de materiales y exhibiciones

Para verificar que la información sea correcta, es decir; que esta corresponda con el **Beacon** que se asignó en el servidor web, se realizó una nueva lista donde se puede observar los datos obtenidos del **Beacon** con su correspondiente información establecida en el servidor web.

```

MAC CF:A5:F0:B8:E8:DF
UUID 252ea4f4-a959-4512-b43c-7edc533c70b0
Major 1
Minor 1
Distancia 2.414
Proximidad NEAR
Nombre: Esqueleto mamut
Descripción: El esqueleto del mamut es una
culminación de diferentes investigaciones a lo largo de
muchos años de trabajo de varios investigadores.

```

Figura 105 Información del Beacon con material asignado

Se accede al material seleccionando un botón disponible el cual pertenece a una plantilla (Texto, Imagen, Video), estas serán habilitadas dependiendo si en el servidor web se encuentran asignadas a un material, cabe mencionar que, si se desea acceder a las plantillas en la proximidad *Near*, se debe tener activada en la configuración la pre descarga a distancia media. Si el usuario se encuentra a una proximidad *Immediate* de algún material, podrá acceder a cualquier plantilla.

En la proximidad *Far*, no estará disponible ninguna plantilla, solamente se visualizará la información del material en la lista.

Para tener claro también en que proximidad se encuentra el usuario de los materiales, se le asignó un color a cada proximidad, tenemos que en *Immediate* es de color verde, en *Near* es color amarillo y *Far* es de color rojo.

6.1.11. Plantillas

Para la elaboración de las plantillas se creó una actividad por cada una de ellas, dentro de estas se insertaron los componentes necesarios para poder visualizar la información de la petición que se ha hecho.

Dentro de cada plantilla, es posible acceder al menú de ayuda, mediante el botón flotante que fue asignado para la esta tarea, esto con la finalidad de que el usuario pueda acceder desde cualquier pantalla a la ayuda.

En la plantilla de video se le insertó un botón el cual al presionarlo se abre el reproductor del teléfono móvil en pantalla completa para cargar el video.

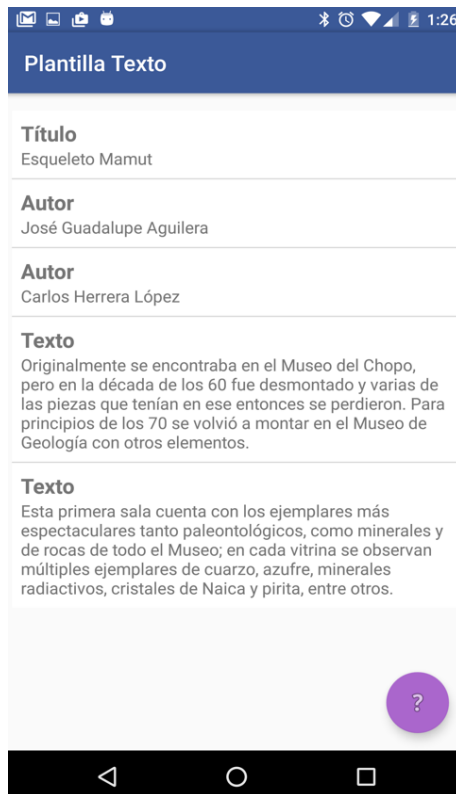


Figura 106 Plantilla texto



Figura 107 Plantilla imagen

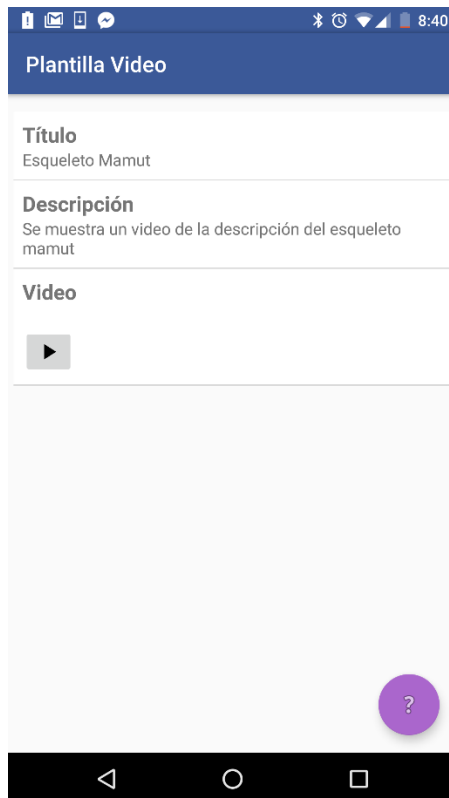


Figura 108 Plantilla de video



Figura 109 Reproducción de video

6.1.12. Archivos Descargables

Se creó una nueva actividad con el fin de mostrar los archivos que estén disponibles para cada material, puede tener uno o más archivos, los cuales el usuario los visualizará mediante alguna aplicación de lectura de archivos en formato PDF o tendrá la opción de descargarlos a su dispositivo móvil. La descarga puede ser visualizada en la barra de notificaciones.

Para lograr la descarga de los archivos creamos una clase llamada *DownloadFileFromURL* que extiende de *AsyncTask* para poder sobre escribir varios de sus métodos, los cuales nos ayudan a realizar esta tarea en segundo plano. Los métodos son los siguientes:

- *onPreExecute()*: Este método se ejecuta antes del código principal de nuestra tarea, aquí se prepara la ejecución de nuestra tarea, en este caso inicia una notificación en la barra de notificaciones.
- *doInBackground()*: Aquí contiene el código principal de nuestra tarea de descarga que básicamente son 3 pasos principales que se realizan.
 - Abre una nueva conexión mediante la clase *URLConnection*
 - Leemos el archivo con el método *getInputStream()* de la clase *URLConnection*. Obteniendo este archivo podemos hacer el cálculo de nuestra barra de progreso en base a su tamaño, la cual se muestra en la barra de notificaciones.
 - Finalmente creamos un archivo haciendo uso de la clase *FileOutputStream*.
- *onProgressUpdate()*: Este método se ejecuta cada que llamamos al método *publishProgress()* explícitamente dentro del método anterior.
- *onPostExecute()*: Se ejecuta cuando finaliza nuestra tarea, el cual muestra un mensaje “Archivo descargado” y lanza una notificación con el mensaje “Descarga completada”.

En la siguiente figura se muestra la lista de los archivos que contiene un material en específico, en la lista se puede observar el nombre del archivo, una breve descripción y dos

botones; el primero es para visualizar el archivo mediante una aplicación de lectura de archivos en formato PDF que el usuario tenga instalada en su dispositivo móvil y el segundo botón es para la descarga del archivo.

De igual manera que en las plantillas, es posible acceder al menú de ayuda mediante el botón flotante que se encuentra dentro de la actividad.

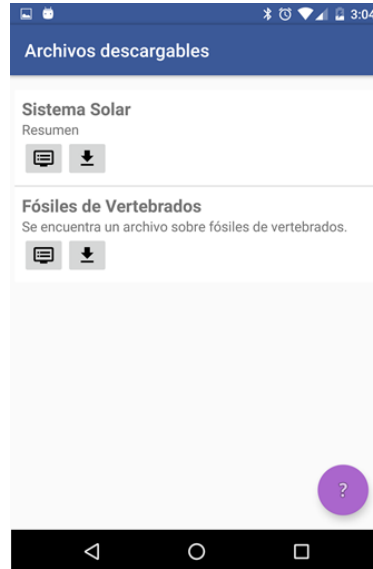


Figura 110 Archivos descargables

Se consideró el hecho de cuando el usuario descargue un archivo y vuelve a presionar el botón de descarga, la aplicación móvil notificará mediante un *Toast* que el archivo ya existe y por lo tanto no se realizará la descarga.

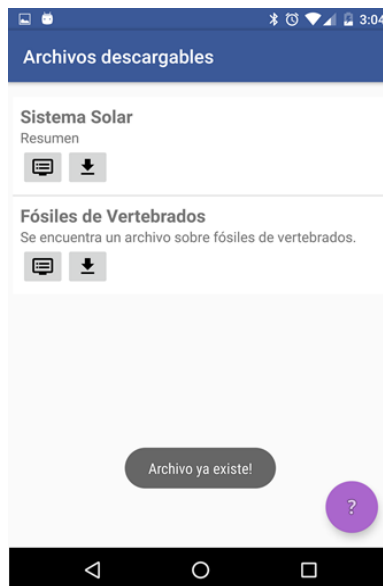


Figura 111 Mensaje "Archivo ya existe!"

6.1.13. Configuración de la aplicación

Para la configuración, se creó una nueva actividad, la cual es llamada *PreferenciasActivity* que extiende de *PreferenceActivity* e implementa el método de *SharedPreferences.OnSharedPreferenceChangeListener*.

El método `onSharedPreferenceChanged()` realiza una acción cuando alguna preferencia es cambiada, estas preferencias se diferencian una de otra por una llave (key).

Esta actividad dispone de varias opciones para poder configurar la aplicación móvil, una de ellas es la pre-descarga, aquí se habilita/deshabilita la pre-descarga del material, al hacer la acción se muestra un mensaje en el dispositivo, también se cuenta con la configuración de Notificaciones, aquí se habilita/deshabilita cuando el usuario se encuentre dentro del rango de un material cercano o a distancia media.

En la sección de Configuración General, se encuentran las conexiones, las cuales el usuario puede acceder con un clic a la configuración del WiFi, configuración de Redes Móviles y la configuración de *Bluetooth*. Esto se realiza mediante “*Intents*” los cuales realizan la acción de lanzar las opciones de configuración para cada uno, como se muestra a continuación.

```
<PreferenceCategory
android:title="General">

<Preference android:title="Configuración WiFi"
  android:icon="@drawable/wifi">
  <intent android:action="android.settings.WIFI_SETTINGS" />
</Preference>

<Preference android:title="Configuración Redes Móviles"
  android:icon="@drawable/redes_mov">
  <intent android:action="android.settings.DATA_ROAMING_SETTINGS" />
</Preference>

<Preference android:title="Configuración Bluetooth"
  android:icon="@drawable/bluetooth_negro">
  <intent android:action="android.settings.BLUETOOTH_SETTINGS" />
</Preference>

</PreferenceCategory>
```

Figura 112 Configuración (Intent)

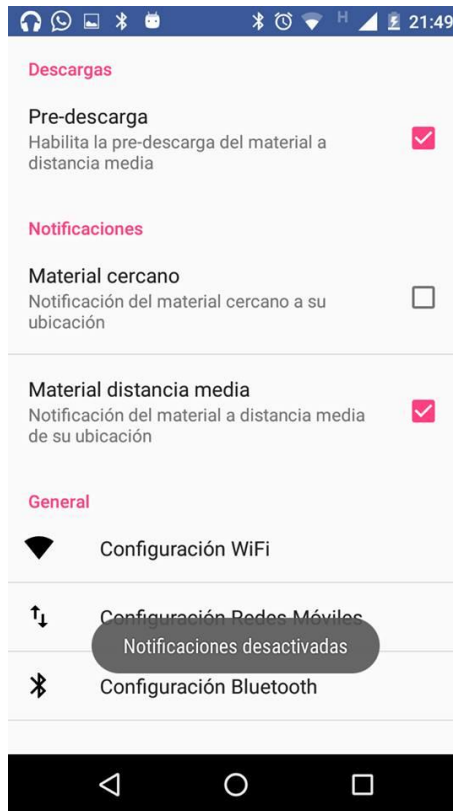


Figura 113 Pantalla configuración

6.1.14. Codificación de Ayudas

De acuerdo con la metodología, realizamos la codificación de ayuda la cual el botón flotante de la aplicación fue asignado para brindar la ayuda para el usuario, como se había comentado anteriormente fue puesto ahí para que esté disponible y visible en todo momento de la aplicación.

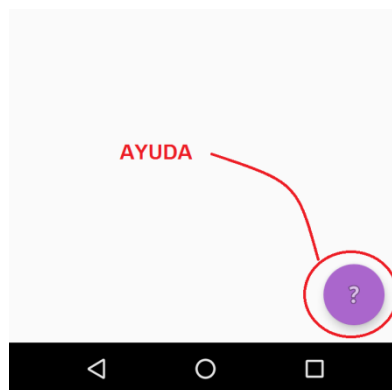


Figura 114 Botón de ayuda

Al presionarlo nos despliega una nueva actividad, mostrándonos un pequeño tutorial de cómo usar la aplicación, para ello utilizamos un *ViewPager* para poder cambiar las ventanas deslizando hacia la derecha o izquierda, según sea el caso. Se implementó en conjunto con un *Fragmento* el cual contiene un *WebView* para mostrar la ayuda la cual está desarrollada con *HTML* y *CSS*.

Al término de la ayuda se muestra un botón de salida y nos regresará a la actividad principal.

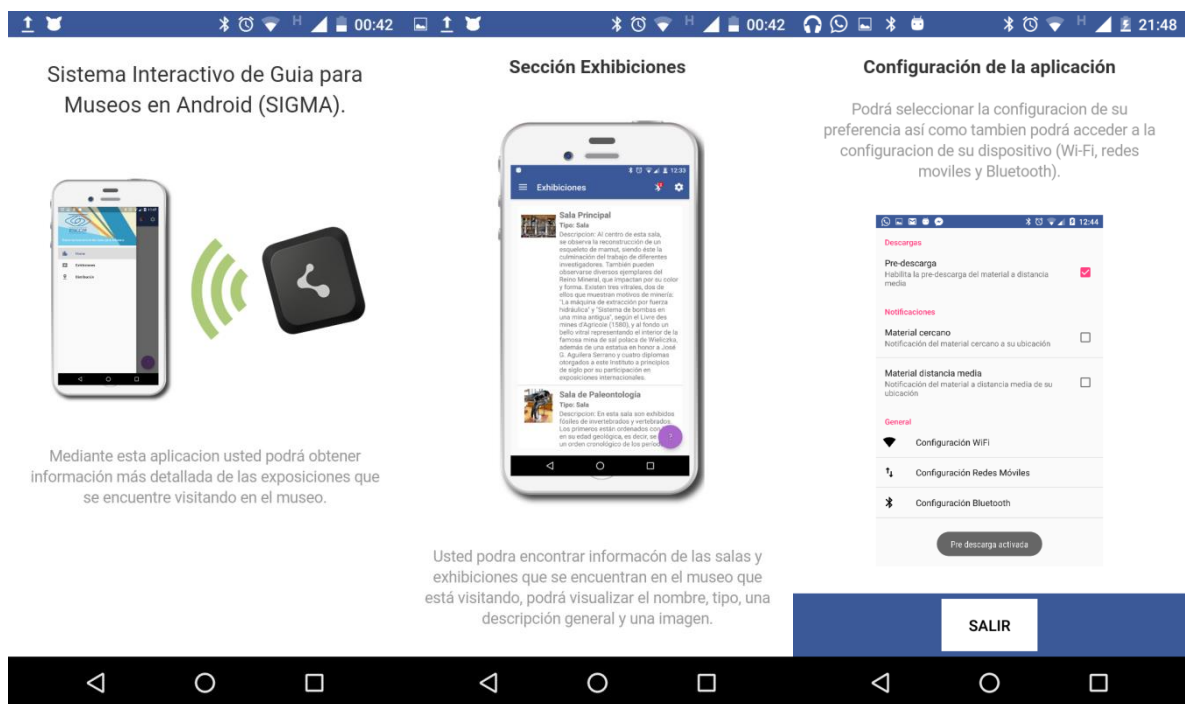


Figura 115 Sección de Ayuda

6.2. Desarrollo del Servidor Web

6.2.1. Descarga e instalación MODX Revolution

Para instalar el *MODX Revolution* en el servidor se descargó de su página oficial con el siguiente enlace: <http://modx.com/download/>, se instaló en su versión 2.4.4 y se requirió lo siguiente: un sistema operativo *Linux*, un servidor *Apache*, tener *PHP 5.3.3*, una base de datos en *MySQL* y un buscador como *Google Chrome*.

6.2.2. Configuración del entorno

Para iniciar el proceso de instalación, es mediante el navegador *web* y hay que ejecutar un script de configuración accediendo a *setup/folder*. Se selecciona un idioma y se muestra la pantalla de bienvenida y las opciones de instalación, en este caso la opción que usaremos es la opción de nuevo y nos genera los permisos por default.

Advanced Options

<input type="text" value="0755"/> New folder permissions	You can override the permissions new folders created via MODx will use, e.g. 0775 or 0777.
<input type="text" value="0644"/> New file permissions	You can override the permissions new files created via MODx will use, e.g. 0664 or 0666.

Figura 116 Permisos de instalación MODX Revolution

El siguiente paso son las opciones de Bases de Datos, saldrá un formulario y se llena con la información de la base que se usará.

Database type:	mysql ▾
Database host:	localhost
Database login name:	<input type="text"/>
Database password:	<input type="password"/>
Database name:	modx
Table prefix:	modx_

– [Test database server connection and view collations.](#)

Figura 117 Opciones de Base de Datos para MODX Revolution

Por último, nos pide ingresar los datos del administrador

Checking database:Success!

Default Admin User

Now you'll need to enter some details for the main administrator account. You can fill in your own name here, and a password you're not likely to forget. You'll need these to log into Admin once setup is complete.

Administrator username:	<input type="text"/>
Administrator email:	<input type="text"/>
Administrator password:	<input type="password"/>
Confirm password:	<input type="password"/>

Figura 118 Ingresar datos de Administrador en MODX Revolution

6.2.3. Módulos

Para desarrollo del servidor web se utilizaron módulos en el *MODX Revolution* para tener un mejor desarrollo en su funcionalidad.

Login

Se requirió un *Login* para tener un extra de seguridad en *MODX Revolution*, este nos permite en las aplicaciones tener procesos de autorización a los usuarios como asignar un inicio de sesión, un registro y la funcionalidad al olvido de contraseña.

Para inicio de sesión se usó lo siguiente:

```
[[!Login? &tplType=`modChunk` &loginTpl=`myLoginChunk`]]
```

Figura 119 Módulo Login de Inicio de Sesión

Para cerrar sesión se usó lo siguiente:

```
<a href= "[[~21? &service=logout]]" title= "Logout"> Salir </a>
```

Figura 120 Módulo Login de cerrar sesión

Las propiedades más importantes que maneja este módulo son:

Nombre	Descripción
actionKey	Es la variable de petición que indica qué acción tomar. Por defecto es "servicio".
loginKey	Su acción es de inicio de sesión. Por defecto es 'login'.
logoutKey	Su acción es de cierre de sesión. Por defecto es 'Salir'.
rememberMeKey	Como su nombre del campo lo indica es para preservar el estado de inicio de sesión.
loginResourceParams	Un objeto JSON en los parámetros es para anexar a la URL de inicio de sesión y se redirecciona.
loginMsg	Puede usar un mensaje opcional para la acción de inicio de sesión.

Tabla 34 Propiedades de módulo Login

[39]

Wayfinder

Wayfinder es un fragmento que escanea una porción específica de la estructura del documento en MODX, se encarga de encontrar todos los documentos que cumplan con ciertos criterios (determinado por los parámetros), y emite una lista con formato de dichos documentos. El formato de la salida es basado en plantillas, y puede contener cualquier combinación de *HTML*, *CSS* y *JavaScript*, produciendo un enorme grado de flexibilidad. Su objetivo principal es generar menús de navegación que se actualizan automáticamente para reflejar los cambios realizados en la estructura del documento.

En *MODX Revolution*, *Wayfinder* puede ser descargado a través de su administración de paquetes.

Para llamar un *Wayfinder* básico que da una salida *HTML* para un multi-nivel, esto nos da una lista desordenada de la estructura del documento.

```
[[Wayfinder? &startId=`0`&level=`1` ]]
```

Figura 121 Módulo Wayfinder

Los parámetros que tiene este módulo son los siguientes:

Nombre	Descripción
&startId	Es el punto de partida (ID de documento) para iniciar el menú en la lista de documentos.
&displayStart	Mostrar el documento como referencia por &startID y en el menú
&level	Profundidad (número de niveles) para construir el menú.
&ignoreHidden	El parámetro de límite que provoca Wayfinder sólo es para procesar el número de elementos especificados por nivel.
&ph	Nombra un marcador de posición para ajustar con los resultados de salida, en lugar de volver directamente a la salida.
&debug	Se pone a '1' para activar el modo de depuración para la resolución de problemas adicionales.
&sortOrder	Pone en orden de clasificación, 'ASC' o 'DESC'

Tabla 35 Parámetros de módulo Wayfinder

QuickCrumbs

Es una ruta de navegación rápida y eficiente para la generación de fragmentos en *MODX Revolution*. Es compatible con *Chunks TPL* de los *Crumbs* y tiene un separador configurable.

Los fragmentos de *QuickCrumbs* pueden ser llamados con la siguiente instrucción:

```
[[QuickCrumbs]]
```

Figura 122 Módulo QuickCrumbs

Las propiedades disponibles para estos fragmentos son:

Nombre	Descripción
Tpl	Nombre de un <i>chunk</i> que sirve como <i>template</i> para un <i>crumb</i> de recursos
siteStartTpl	Sirve como <i>template</i> para la <i>site_start</i> .
selfTpl	Sirve como <i>template</i> para el <i>crumb</i> de recursos actual.
outerTpl	Sirve como <i>wrapper template</i> para la salida de <i>crumb complete</i> .
Separator	Una cadena que utiliza como separador entre los <i>crumbs</i> y por defecto es `»`
toPlaceholder	Si no está vacía, la salida se guarda en un marcador de posición con el nombre especificado en lugar de regresar directamente del fragmento.
Debug.	Si es verdad, la información de depuración se enviará al registro de <i>MODX</i> .

Tabla 36 Propiedades de módulo QuickCrumb

6.2.4. Acceso a MODx Revolution

Para poder acceder al *MODx Revolution* se debe ingresar la dirección en donde se encuentra instalado, la cual es <http://dev.politicaenlinea.com/manager/>

Al acceder a la dirección ingresaremos el nombre y contraseña puesto anteriormente en la configuración del *MODx Revolution*.



Figura 123 Acceso a MODx Revolution

Al ingresar en la parte izquierda del navegador se muestra el directorio raíz del proyecto, ahí se colocan los paquetes a utilizar dependiendo las subdirecciones de destino, ya que cada uno tiene su propio conjunto de tareas.

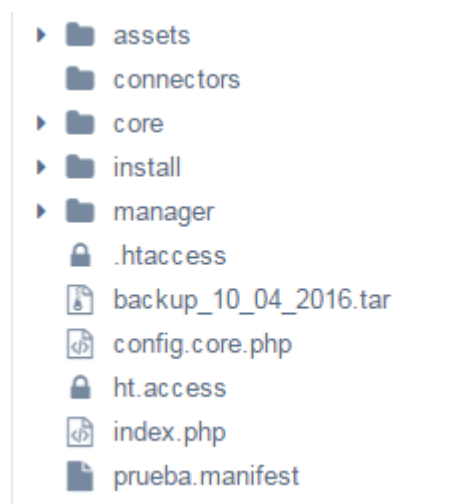


Figura 124 Directorio Raíz MODx Revolution

6.2.5. Connectors

Los conectores son esencialmente puntos de entrada de las solicitudes de *AJAX* en *MODx Revolution*. No hacen ningún tipo de manipulación de bases de datos por su cuenta; simplemente basta con cargar la clase principal y luego manejar la petición de que apunta al archivo de procesador adecuado.

6.2.6. Core

Es la base para todas las bibliotecas del *MODx Revolution*, con la excepción de los archivos del gestor y los archivos de instalación, se encuentran en este directorio.

6.2.7. Manager

Es el área administrativa para la creación de recursos, gestión de usuarios, y la realización de tareas de mantenimiento general del sitio.

[42]

6.2.8. Assets

En este directorio agrega la funcionalidad de gestión de imágenes, *CSS*, *JavaScript*, etc. Ya que almacena todos los datos en una tabla de base de datos personalizada, puede buscar fácilmente por criterios o manipular las consultas de búsqueda en informes personalizados.

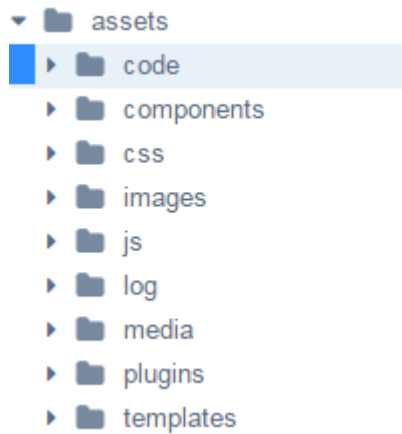


Figura 125 Directorio Assets

6.2.9. Code

Se creó el subdirectorio code, ahí se crearon las paquetes de *MODX* y *tt*, aquí estarán las clases de los objetos a trabajar, como se muestra en el diagrama de clases del servidor (Ver figura 66).

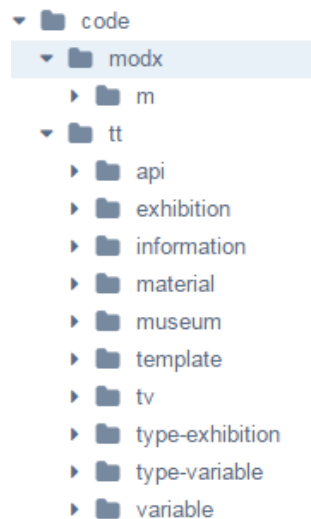


Figura 126 Directorio Code

6.2.10. Modx

En el directorio `code/modx/m` se encuentran las clases para la conexión a la Base de Datos llamada BD, la validación de los archivos de imagen y video que se guardan en el servidor esto se encuentra en la clase F, el cargar los objetos del *MODx Revolution* en la clase M y las validaciones de los datos a guardar en la base de datos en la clase *Mvalidation*.

```
class F{

    static $dirImage;
    static $dirVideo;
    static $urlImage;
    static $urlVideo;
    static $mimeImage=array('image/gif' => 'gif',
                            'image/jpeg' => 'jpeg',
                            'image/png' => 'png',
                            //'application/x-shockwave-flash' => 'swf',
                            //'image/psd' => 'psd',
                            'image/bmp' => 'bmp',
                            //'image/tiff' => 'tiff',
                            //'image/tiff' => 'tiff',
                            'image/jp2' => 'jp2',
                            //'image/iff' => 'iff',
                            //'image/vnd.wap.wbmp' => 'bmp',
                            //'image/xbm' => 'xbm',
                            //'image/vnd.microsoft.icon' => 'ico'
                            );

    /*
    * Max Size for image upload in bytes
    */
    static $maxSizeImage= 3145728; // 3 Megabytes

    static function init()
    {
        F::$dirImage = MODX_ASSETS_PATH."media/image/";
        F::$dirVideo = MODX_ASSETS_PATH."media/video/";
        F::$urlImage = "/assets/media/image/";
        F::$urlVideo = "/assets/media/video/";
    }
}
```

Figura 127 Fragmento de la clase F

6.2.11. TT

En este subdirectorio se encuentran las clases de los objetos que se utilizan para la el servidor web.

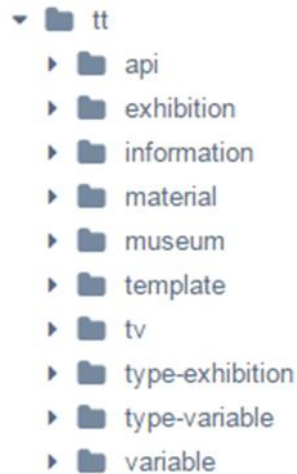


Figura 128 Directorio assets/code/tt

Cada objeto tiene tres categorías, esas son las clases, controladores y modelos. En clases se encuentran y se declaran los atributos de cada objeto y si tienen una relación con alguna otra clase de otro objeto, en los controladores está el índice de control de cada objeto, estos son las acciones que realizan como dar altas, bajas, leer los datos que tiene, borrar los datos y también mandar las peticiones *POST* a respectiva dirección que proporciona la clase M del objeto *modx*. Por ultimo en la categoría modelo se encuentran las sentencias *SQL* para la base de datos, al igual que en el controlador aquí se pueden hacer las acciones de dar de alta, baja, leer datos y actualizarlos.

```

function setObject($obj)
{
    $this->id=(INT) $obj->id_exhibition;
    $this->name=$obj->name;
    $this->description=$obj->description;
    $this->image=$obj->image;
    $this->major=$obj->major;
    $this->status=$obj->status;

    $this->museum=M::getClass("tt/museum_museum",array(
        "id"=>$obj->id_museum,
        "name"=>$obj->M_name,
        "uuid"=>$obj->M_uuid,
        "description"=>$obj->M_description,
        "image"=>$obj->M_image,
        "map"=>$obj->M_map,
        "creation_datetime"=>$obj->M_creation_datetime,
        "modification_datetime"=>$obj->M_modification_datetime
    ));

    $this->typeExhibition=M::getClass("tt/type-exhibition_type-exhibition",array(
        "id"=>$obj->id_type_exhibition,
        "name"=>$obj->TE_name
    ));
}

```

Figura 129 Fragmento de código en clases

```

function showFormCreateAction()
{
    return M::$modx->getChunk("FormExhibition");
}

function showFormEditAction()
{
    $model=M::getModel("tt/exhibition_base");

    $exhibition=$model->getItemById($_POST["id"]);

    return M::$modx->getChunk("FormExhibition",
        array(
            "name"=>$exhibition->name,
            "description"=>$exhibition->description,
            "major"=>$exhibition->major,
            "srcImage"=>($exhibition->image)? F::$urlImage.$exhibition->image:"",
            "showImage"=>($exhibition->image)? "s":"" ,
            "status"=>($exhibition->status)? "checked":"" ,
            "museum"=>$exhibition->museum->id,
            "typeExhibition"=>$exhibition->typeExhibition->id
        )
    );
}

```

Figura 130 Fragmento de código en controlador

```

function create($exhibition)
{
    $image=$exhibition->image;

    $sql="insert into tt_exhibition(id_museum,id_type_exhibition,name,description,image,major,status)
    values(:id_museum,:id_type_exhibition,:name,:description,:image,:major,:status)";

    $sth = DB::$dbh->prepare($sql);

    $sth->bindParam(':id_museum', $exhibition->museum->id, PDO::PARAM_INT);
    $sth->bindParam(':id_type_exhibition', $exhibition->typeExhibition->id, PDO::PARAM_INT);
    $sth->bindParam(':name', $exhibition->name, PDO::PARAM_STR);
    $sth->bindParam(':description', $exhibition->description, PDO::PARAM_STR);
    $sth->bindParam(':image', $image, PDO::PARAM_STR);
    $sth->bindParam(':major', $exhibition->major, PDO::PARAM_INT);
    $sth->bindParam(':status', $exhibition->status, PDO::PARAM_INT);

    if( $sth->execute() )
    {
        return true;
    }

    M::log($sth->errorInfo());
    M::log($exhibition);

    return false;
}

```

Figura 131 Fragmento de código en modelo

En el directorio de *assets/media* se encuentran guardados todas las imágenes y videos que encontraremos en el servidor.

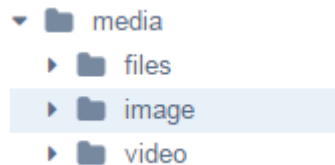


Figura 132 Directorio assets/media

En el directorio *assets/js* se encuentran todas las clases utilizadas en el sitio web hechas en *JavaScript*.

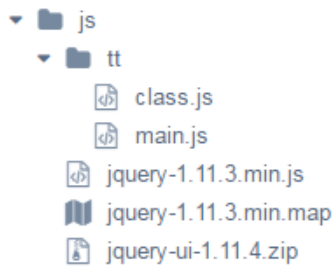


Figura 133 Directorio js

```
function newUUID() {
  function s4() {
    return Math.floor((1 + Math.random()) * 0x10000).toString(16).substring(1);
  }

  return s4() + s4() + '-' + s4() + '-' + s4() + '-' + s4() + '-' + s4() + s4() + s4();
}

function getVariable(variable)
{
  var query = window.location.search.substring(1);
  var vars = query.split("&");
  for (var i=0;i<vars.length;i++) {
    var pair = vars[i].split("=");
    if(pair[0] == variable){return pair[1];}
  }
  return(false);
}

function showQuantity(){
  console.log("quantity");
  if($("#limited").is(':checked'))
  { $(".div_quantity").show("slow");
  }else{
    $(".div_quantity").hide("slow");
  }
}
```

Figura 134 Fragmento de código en clase de js

6.3. Desarrollo API

Al igual que todos los objetos tiene sus tres categorías (clases, controladores y modelo), este se encarga de crear el objeto *JSON* que recibe la aplicación móvil, recibe los parámetros igualmente de la aplicación y da los códigos de acceso.


```

function createJson(){
    header('Content-Type: application/json');
    return json_encode($this->getResponse());
}

function curl($obj){

    //$url=$obj->url;
    //$headers=$obj->headers;

    $ch = curl_init();

    $headers = array(
        "Content-Type: application/json",
        "Secret_id: AA3f3g34g3-fg3g432",
        "Customer_id: J3J56JK6754H54K76");

    $options = array(
        CURLOPT_URL => "http://dev.politicaenlinea.com/api/museum/",
        CURLOPT_RETURNTRANSFER=> true,
        CURLOPT_HEADER => false,
        CURLOPT_HTTPHEADER => $headers,
        CURLOPT_POSTFIELDS => $this->data
    );
}

```

Figura 135 Creación de JSON en API

```

function getAll()
{
    return [
        [
            "code"=>200,
            "text"=>"Ok",
            "description"=>"Success!"
        ],
        [
            "code"=>401,
            "text"=>"No autorizado",
            "description"=>"Las credenciales de autenticación faltan o son incorrectas."
        ],
        [
            "code"=>402,
            "text"=>"Faltan parametros de busqueda.",
            "description"=>"Faltan parametros para ejecutar la busqueda."
        ],
        [
            "code"=>410,
            "text"=>"Museo no encontrado",
            "description" =>"El UUID proporcionado no corresponde con ningun museo."
        ],
        [
            "code"=>411,
            "text"=>"Material no encontrado",
            "description" =>"Los parametros proporcionados no corresponde con ningun material."
        ],
        [
            "code"=>412,
            "text"=>"Material no activado",
            "description" =>"El material se encuentra desactivado."
        ]
    ];
}

```

Figura 136 Códigos de acceso.

```

class ApiBaseModel{
    function existKey($secret_key,$customer_key){
        $sql="select count(*) as 'count' from tt_key where secret_key=:secret_key and customer_key =:customer_key";
        $sth = DB::$dbh->prepare($sql);
        $sth->bindParam(':secret_key', $secret_key,PDO::PARAM_STR);
        $sth->bindParam(':customer_key', $customer_key,PDO::PARAM_STR);
        $sth->execute();
        $obj=$sth->fetchObject();
        return (INT) $obj->count;
    }
}

```

Figura 137 Consulta SQL de API

6.3.1. Requerimientos API

Todas las peticiones al *API* deberán de enviar las credenciales:

- *Secret_key*
- *Customer_key*

6.3.2. Códigos de error

401: Las credenciales proporcionadas *secret_key* y/o *customer_key* son incorrectas o faltan.

402: Faltan parámetros de búsqueda.

410: El *UUID* proporcionado no corresponde con ningún museo.

411: Los parámetros proporcionados no corresponden con ningún material.

412: El material se encuentra desactivado.

Ejemplo:

```

{
  "error": {
    "code": 401,
    "text": "No autorizado",
    "description": "Las credenciales de autenticación faltan o son incorrectas."
  }
}

```

6.3.3. POST museum/

Regresa la información de un museo en específico.

URL

<http://dev.politicaenlinea.com/api/museum/>

Información de Petición

- Formato de respuesta: JSON
- Requiere autenticación: Si

Parámetros

- uuid

Posibles errores

- 401
- 402
- 410

Ejemplo de resultado

```
{
  "success": {
    "code": 200,
    "text": "Ok",
    "description": "Success!"
  },
  "data": {
    "museum": {
      "id": 20,
      "name": "Museo de Geolog\u00eda",
      "description": "El Museo de Geolog\u00eda de la UNAM ...",
      "image": "1457975743.jpg",
      "map": "1459381571.png",
      "uuid": "252ea4f4-a959-4512-b43c-7edc533c70b0",
      "creation_datetime": "2016-02-28 04:19:36",
      "modification_datetime": "2016-03-30 23:46:11"
    },
    "exhibitions": [
      {
        "id": 1,
        "name": "Sala Principal",
        "description": "Al centro de esta sala, se observa la reconstrucción ...",
        "image": "1457999470.jpg",
        "major": "1",
        "status": 1,
        "museum": null,
        "typeExhibition": {
          "id": 8,
          "name": "Sala"
        }
      }, {
        "id": 2,
        "name": "Sala de Paleontolog\u00eda",
        "description": "En esta sala son exhibidos f\u00f3siles de
        invertebrados..."
      }
    ]
  }
}
```

```

        "image": "1458961097.jpg",
        "major": "10",
        "status": 1,
        "museum": null,
        "typeExhibition": {
            "id": 8,
            "name": "Sala"
        }
    ]
}

```

6.3.4. POST material/

Regresa la información de un material en específico.

La información templates dependerá de la proximidad enviada; Immediate = 1, Near = 2 y Far = 3

URL

<http://dev.politicaenlinea.com/api/material/>

Información de Petición

- Formato de respuesta: JSON
- Requiere autenticación: Si

Parámetros

- Uuid
- Major
- Minor
- proximity

Posibles errores

- 401
- 402
- 411
- 412

Ejemplo de resultado

Proximidad *Near* y *Far* con *onlyExhibition* = 0

```

{
    "success": {
        "code": 200,
        "text": "Ok",
        "description": "Success!"
    },
    "data": {

```

```

    "material": {
      "id": 6,
      "name": "Ejemplo material",
      "description": "Descripción del material ",
      "minor": "43",
      "status": "1",
      "onlyExhibition": "0",
      "principalTemplate": null,
      "exhibition": null,
      "templates": null
    }
  }
}

```

Proximidad *Near* y *Far* con *onlyExhibition* = 1

```

{
  "success": {
    "code": 200,
    "text": "Ok",
    "description": "Success!"
  },
  "data": {
    "material": {
      "id": 3,
      "name": "lorem",
      "description": "fds",
      "minor": "4",
      "status": "1",
      "onlyExhibition": "1",
      "principalTemplate": null,
      "exhibition": {
        "id": 10,
        "name": "Colecci\u00f3n de exhibici\u00f3n",
        "description": "La Colecci\u00f3n de exhibici\u00f3n incluye
todas ....",
        "image": "1459214724.jpg",
        "major": "1245",
        "status": 1,
        "museum": null,
        "typeExhibition": null
      },
      "templates": null
    }
  }
}

```

Proximidad *Immediate*

```

{
  "success": {
    "code": 200,
    "text": "Ok",

```

```

    "description": "Success!"
  },
  "data": {
    "material": {
      "id": 2,
      "name": "F\u00f3siles de vertebrados",
      "description": "La mayor parte de los espec\u00edmenes procede de lechos f\u00f3siles de vertebrados en su mayor\u00eda son del Pleistoceno, excepto el dinosaurio Pico de Pato y el ictiosauro que son del mesozoico y el cr\u00e1neo del dinoterio, de edad terciaria.",
      "minor": "9",
      "status": "1",
      "onlyExhibition": "0",
      "templates": [{
        "template": "image",
        "information": [{
          "title": "T\u00edtulo",
          "data": "Galer\u00eda F\u00f3siles de vertebrados",
          "type": "texto"
        }, {
          "title": "Descripci\u00f3n",
          "data": "Se muestran los diferentes tipos de F\u00f3siles de vertebrados",
          "type": "texto"
        }, {
          "title": "Fecha de creaci\u00f3n",
          "data": "34/02/2004",
          "type": "texto"
        }, {
          "title": "Autor",
          "data": "Emmanuel Montiel Martinez",
          "type": "texto"
        }
      ]
    }, {
      "template": "text",
      "information": [{
        "title": "Descripci\u00f3n",
        "data": "Los primeros vertebrados se originaron alrededor de 525 millones de a\u00f1os durante la explosi\u00f3n c\u00e1mbrica, que vio el aumento de la diversidad organismo. El primer vertebrado conocido se cree que es la Myllokunmingia, se asemejaba a los mixinos actuales; carec\u00eda de mand\u00edbulas (agnato) y tanto su cr\u00e1neo como su esqueleto eran cartilagosos. Otro vertebrado es el ercaicunensis Haikouichthys. ",
        "type": "texto"
      }, {
        "title": "Autor",
        "data": "Marcos%%",
        "type": "texto"
      }
    ]
  }
}

```

```
}},  
"files": [{  
    "id": 2,  
    "name": "Sistema Solar",  
    "description": "Resumen",  
    "file": "1463107450.pdf"  
}, {  
    "id": 3,  
    "name": "Sistema Solar 2",  
    "description": null,  
    "file": "1463110112.pdf"  
}]  
}}}
```

CAPÍTULO 7. PRUEBAS DE FUNCIONAMIENTO

En esta sección se verificó el funcionamiento de la aplicación en diferentes escenarios y condiciones; primeramente se trató de realizar las pruebas en el emulador, lo cual dadas las características de hardware de nuestro equipo de cómputo, las pruebas en el emulador no fueron posibles realizar; al intentar cargar el emulador, la computadora se volvía demasiado lenta y el dispositivo virtual se quedaba en negro.



Figura 138 Error dispositivo virtual

7.1. Dispositivos reales

Las pruebas fueron realizadas en un *Motorola Nexus 6* y en una *Samsung Galaxy Tab S2*, que cuentan con las características necesarias para que la aplicación móvil pueda ser ejecutada y un sistema operativo Android 6.0.1 y 5.0.1 respectivamente.

Para ello fue necesario conectar el *Smartphone* y la *Tablet* en modo desarrollador.

Configuración > Opciones del programador > Activar

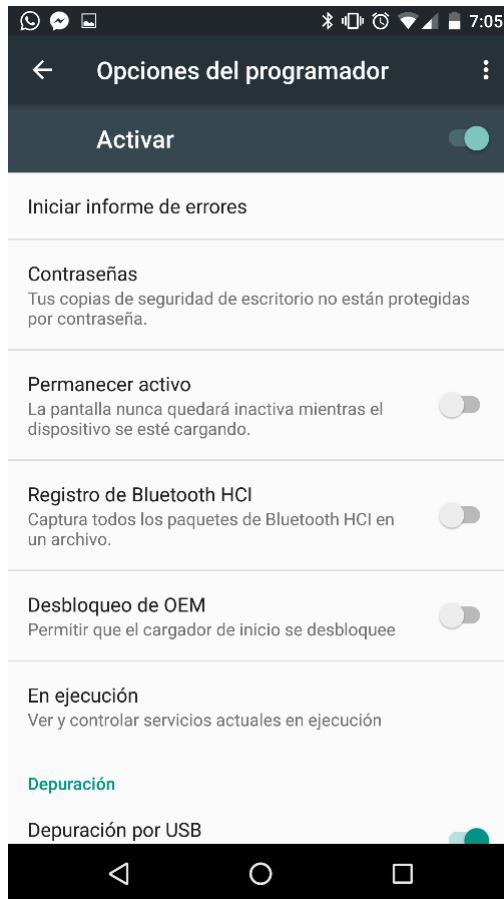


Figura 139 Opciones del programador Nexus 6

Una vez activadas las opciones del programador, conectamos el teléfono a la computadora y en el Android Studio ejecutamos la aplicación móvil para poder visualizar su ejecución.

Run > Debug 'app'

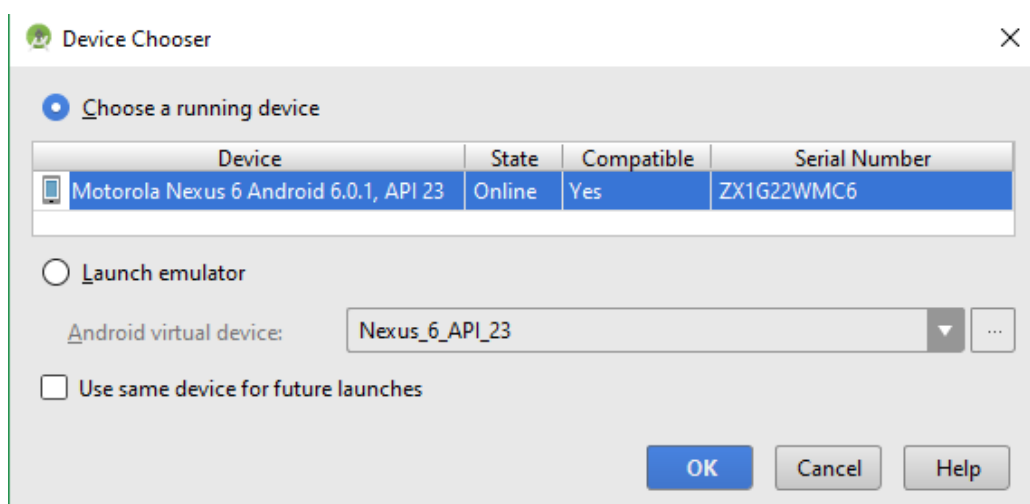


Figura 140 Seleccionar dispositivo para ejecutar

Las pruebas que se realizaron sobre la aplicación móvil fueron las siguientes:

- **Se comprobó la conexión a internet y el encendido del *bluetooth***

Para realizar las pruebas necesarias de conexión a Internet, primeramente desconectamos cualquier acceso a internet para hacer la comprobación, al hacer esto, la aplicación móvil no detecta la señal de internet y nos despliega dos *toast* “No hay conexión a Internet” y “Comprueba tu conexión”, al tener este problema la pantalla quedara cargando y no muestra ningún tipo de información, ya que no manda ni recibe los datos de la API que realiza la conexión al servidor.

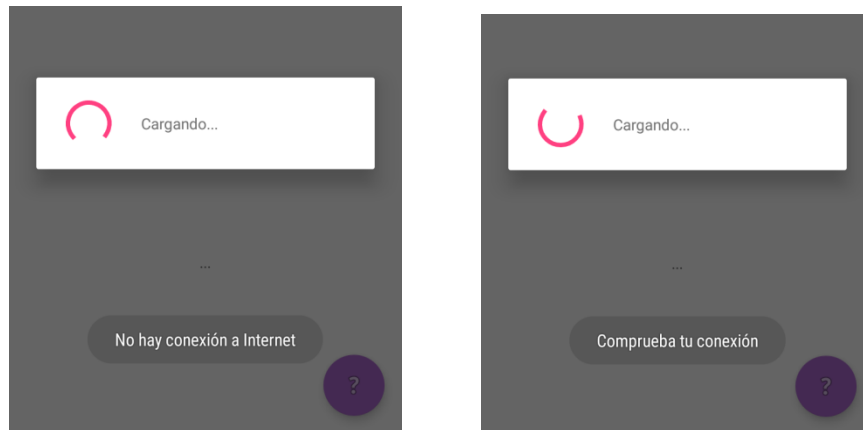


Figura 141 Pruebas conexión a internet Nexus 6

Para su correcto funcionamiento, se debe contar con la conexión a Internet para el mandar los datos a través de la API al servidor y recibir la información del fragmento home.

También comprobamos que al iniciar la aplicación móvil, el Bluetooth del dispositivo se encendiera, este es una parte muy importante del proyecto, porque este nos recopilara los datos de los Beacon (*UUID, Major, Minor y distance*), para poderlos enviar al servidor a través de la API.

- **Visualización de la información y acceso a la ayuda de la aplicación móvil.**

En este proceso de pruebas se verificó que la información fuera mostrada correctamente, mediante la conexión a la API (Ver sección 10.2).



Figura 142 Prueba visualización de la información Nexus 6

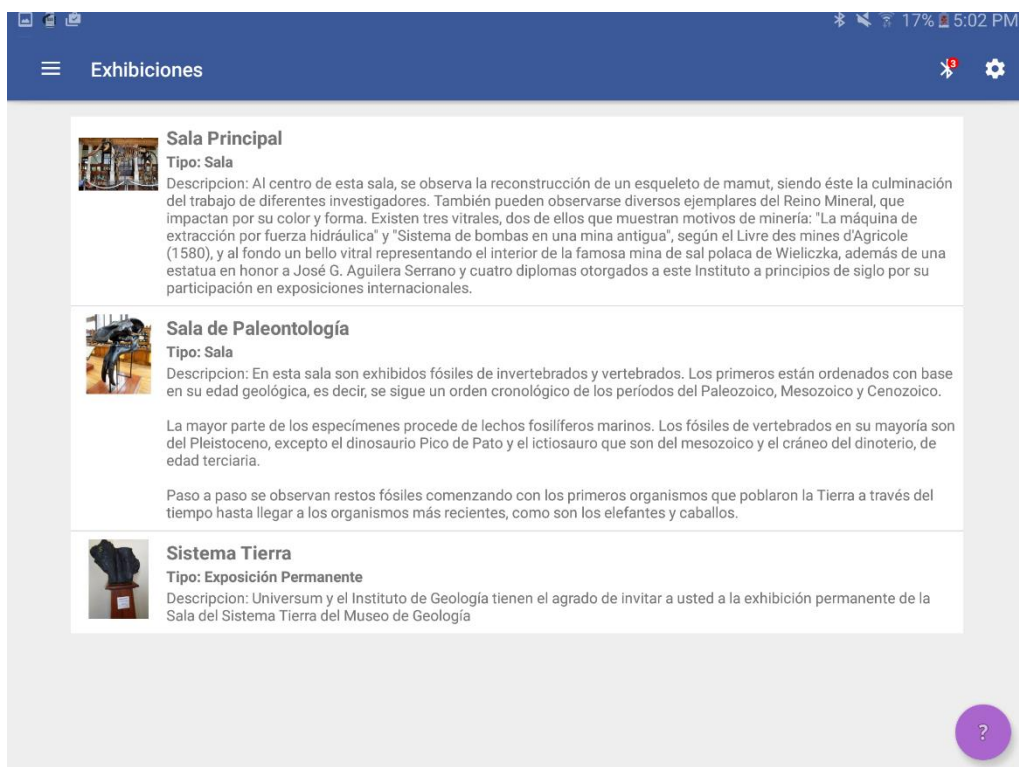


Figura 143 Prueba visualización de la información Galaxy Tab S2

También se corroboró que al presionar el botón flotante de la ayuda se mostrara la actividad correspondiente para esta acción.



Sistema Interactivo de Guía para Museos en Android (SIGMA).



Mediante esta aplicación usted podrá obtener información más detallada de las exposiciones que se encuentre visitando en el museo.



Figura 144 Prueba acceso a la ayuda en Nexus 6

- **Lectura de una sola región de Beacons**

La región de Beacon que se utiliza es solamente definida por el *UUID*, ya que este será utilizado para distinguir cada museo que se encuentre en la base de datos

La prueba que se hizo fue cambiarle el *UUID* a uno de los tres **Beacon** y observamos que solamente detectó dos, los cuales estaban dentro de la región asignada, por lo tanto esto no afecta en los datos de la aplicación móvil ya que ni siquiera ese dispositivo es leído.

- **Proximidad de Beacons**

Se comprobó el rango de proximidades (*Immediate*, *Near*, *Far* y *Unknown*) y se realizaron distintas pruebas.

Como establecimos durante el desarrollo del sistema, la proximidad *Immediate* abarca un radio de 0 a 1.5 metros, en *Near* de 1.5 a 5 metros y por último *Far* está definida de 5 a rango máximo, en este caso, en las pruebas nos dieron un máximo de 20 metros aproximadamente.

En la siguiente tabla se muestran las variaciones que obtuvimos al estar probando los **Beacons** en espacio abierto sin ningún tipo de obstáculo.

ESPACIO ABIERTO (SIN OBSTACULOS)	
Distancia real	Distancia obtenida por la app.
1.80 m.	1.76 – 1.84 m.
5 m.	5 – 5.30 m.
10 m.	10.5 – 11 m.
15 m.	17 – 19 m.

Tabla 37 Prueba de distancia espacio abierto

La siguiente tabla muestran las pruebas que se realizaron en un espacio cerrado con obstáculos como vidrios, estantes y personas.

ESPACIO CERRADO (CON OBSTACULOS)	
Distancia real	Distancia obtenida por la app.
1.60 m.	1.65 m.
4.8 m.	4.6 – 5.8 m.
7 m.	5.9 – 7 m.
7.3 m.	7 – 8 m.

Tabla 38 Prueba de distancia espacio cerrado

- Se verificó se lanzaran las notificaciones correspondientes dependiendo de la configuración de la aplicación móvil.

En el menú de configuración, se tiene el activar o desactivar las notificaciones, esto es, que cada vez que la aplicación móvil encuentra un Beacon en una distancia IMMEDIATE (cercano) o NEAR (distancia media), nos llega una notificación al dispositivo móvil.

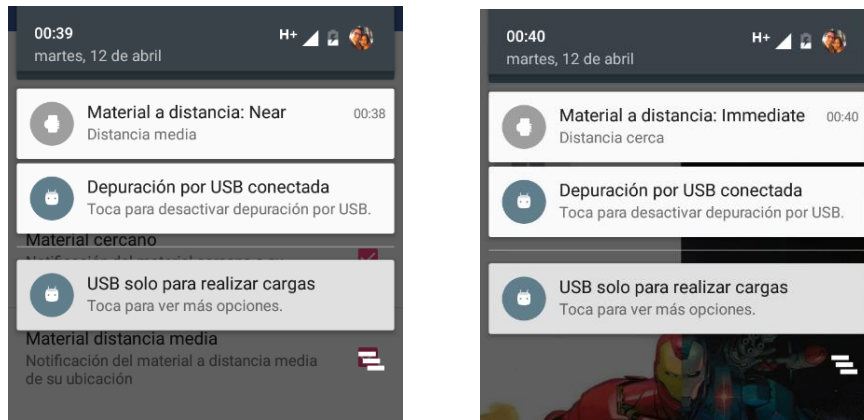


Figura 145 Pruebas notificaciones Nexus 6

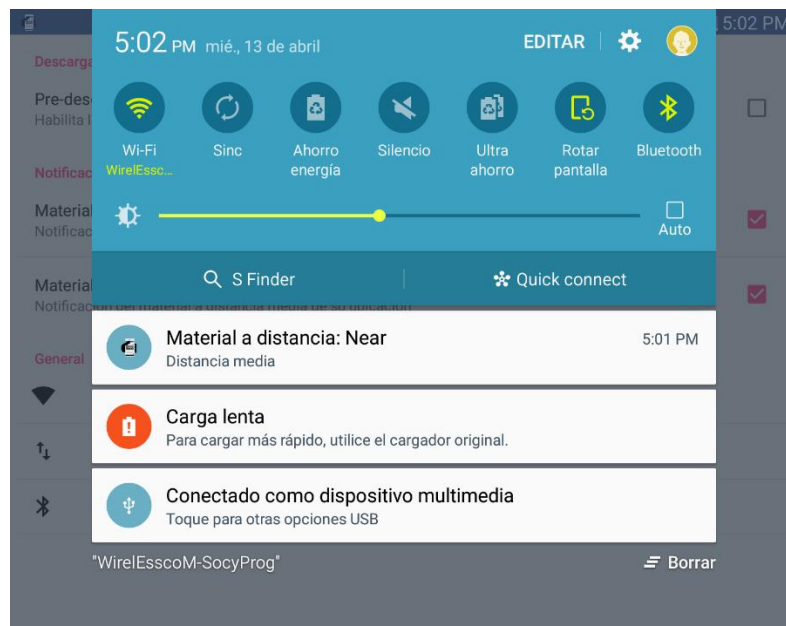


Figura 146 Pruebas notificaciones Galaxy Tab S2

Al estar la aplicación móvil en segundo plano continuará mandando notificaciones hasta que sea cerrada totalmente.

- **Listado de materiales y exhibiciones**

En esta sección se corroboró que el listado que se muestra en el Fragmento de materiales y exhibiciones, correspondiera con la información que se encuentra en el servidor web.

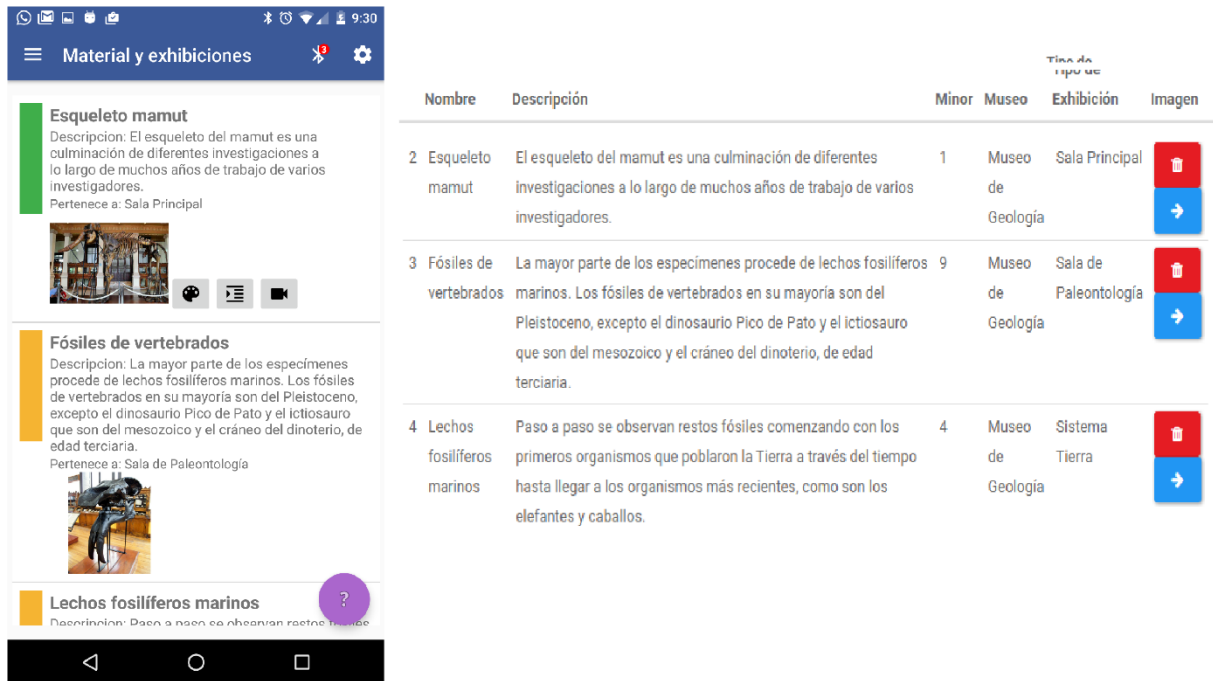


Figura 147 Pruebas materiales y exhibiciones 1

Se verificó que se habilitaran los botones de las plantillas correspondientes para cada material y que la lista se ordenara de acuerdo a la proximidad de cada material, también se verificó la actualización de la lista cada vez que hubiera un cambio de proximidad o que entrara o saliera un **Beacon** de la región.

Otra de las pruebas realizadas fue la pre descarga a distancia media, cuando esta estuviera activada, el material a distancia *Near* debe mostrar las plantillas asignadas a dicho material.

En la siguiente figura, del lado izquierdo se muestra la opción “Pre- descarga” habilitada y del lado derecho se observa que el material “Esqueleto Mamut” está a una distancia *Near* y se encuentran habilitadas las plantillas que tiene asignado dicho material. El material que se encuentra debajo de “Esqueleto Mamut” no cuenta con alguna plantilla en el servidor web.

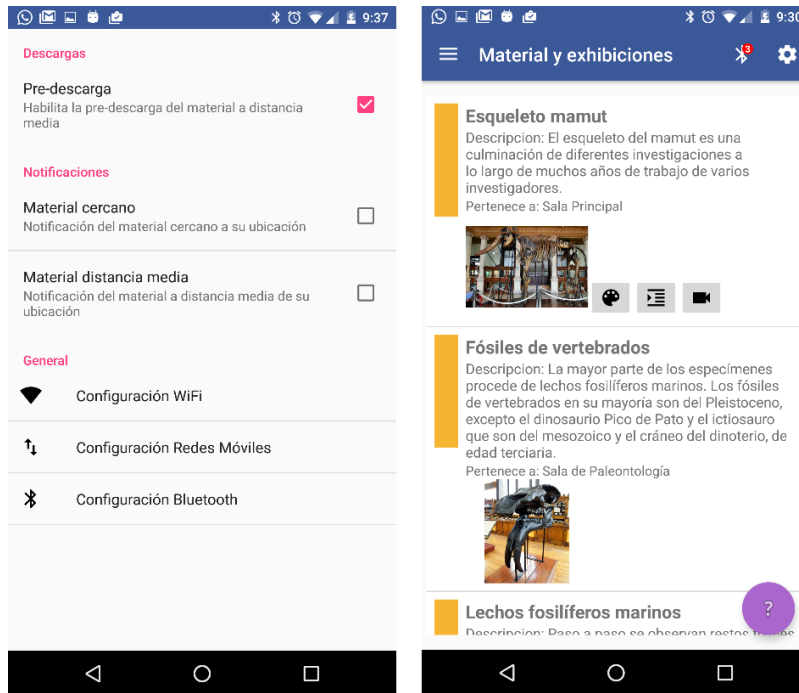


Figura 148 Pruebas materiales y exhibiciones 2

- **Plantillas texto, imagen y video**

Para comprobar el buen funcionamiento de las plantillas se revisó que cada una de ellas mostrara la información que se le fue asignada:

- Plantilla de texto: Título, el autor o autores designados y una o varias descripciones del material.
- Plantilla imagen: Título, el autor o autores designados, una o varias descripciones y una o más imágenes del material.
- Plantilla video: Título, una o varias descripciones y el botón para reproducir el video.

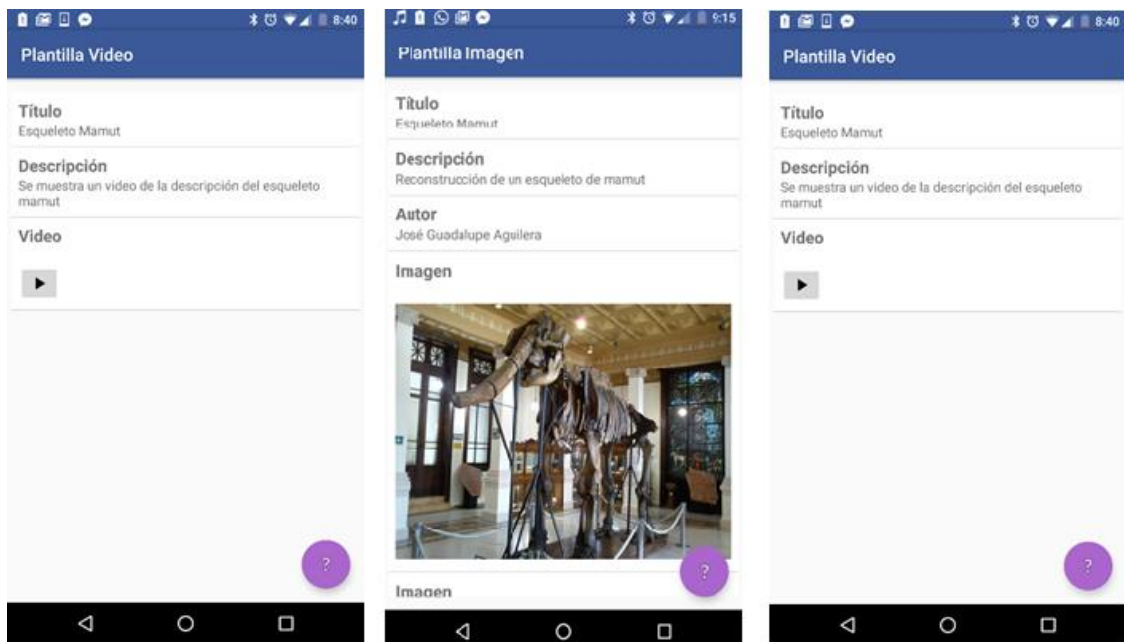


Figura 149 Pruebas plantillas texto, imagen y video

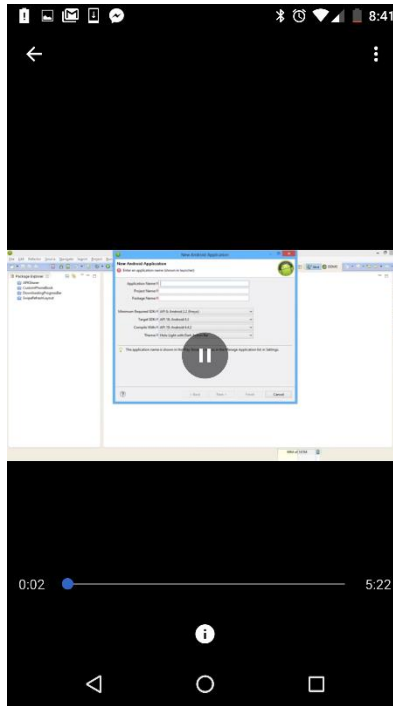


Figura 150 Pruebas reproductor de video

- **Lista de archivos descargables**

Las pruebas que se realizaron en esta sección fueron que se mostrara la lista de archivos (nombre y descripción) al hacer clic en el botón de archivos sobre el material y que esta lista concidiera con los archivos asignados en el servidor web. También se verificó que cada uno de los botones asignados (Ver y descargar PDF) cumpliera con su tarea, así como se mostraran las notificaciones y el proceso de descarga en la notificación.

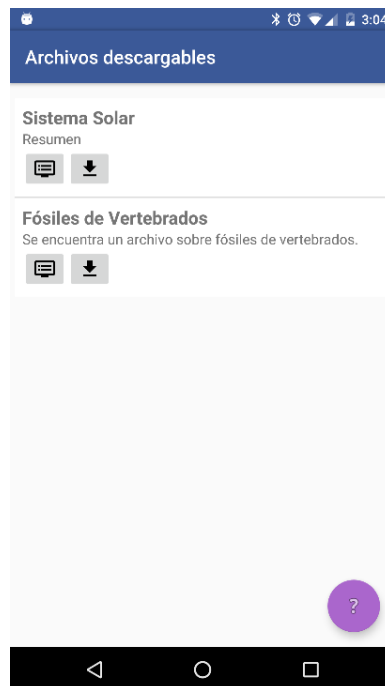


Figura 151 Prueba lista de archivos descargables

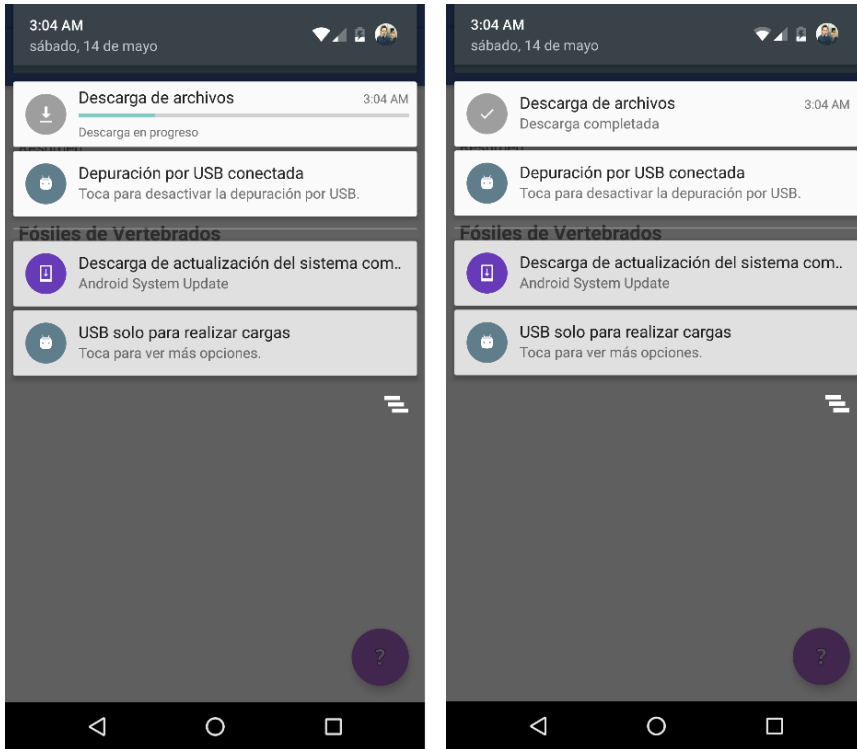


Figura 152 Pruebas notificaciones descarga de archivos

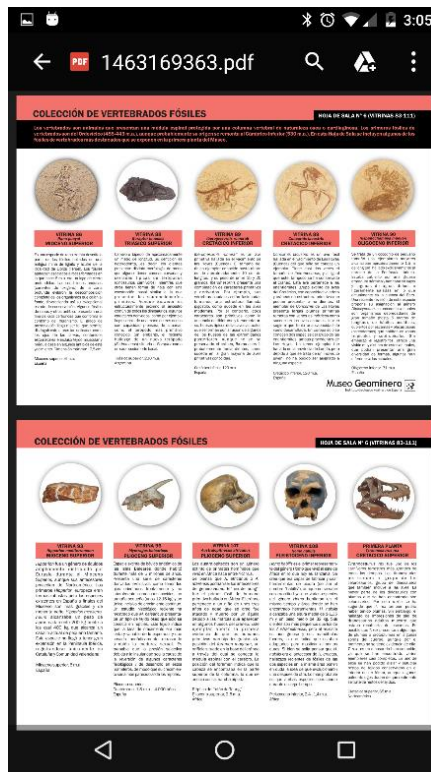


Figura 153 Prueba apertura de archivos

7.2. Pruebas API

La API requiere de una *secret_key* y *customer_key* para poder tener acceso a ella y dependiendo la API que se utilizara nos pedirá otros campos, tales como UUID, Major, Minor, entre otros.

Al no enviar los campos correspondientes o enviar datos erróneos, la API nos puede regresar alguno de los siguientes errores, dependiendo el caso:

401: Las credenciales proporcionadas *secret_key* y/o *customer_key* son incorrectas o faltan.

402: Faltan parámetros de búsqueda.

410: El UUID proporcionado no corresponde con ningún museo.

411: Los parámetros proporcionados no corresponden con ningún material.

412: El material se encuentra desactivado.

En la siguiente imagen se puede observar el error que muestra el IDE de desarrollo (*Android Studio*) cuando alguna de las dos credenciales son incorrectas

```
{
  "error": {
    "code": 401,
    "text": "No autorizado",
    "description": "Las credenciales de autenticación faltan o son incorrectas."
  }
}
```

Figura 154 Error de credenciales API en Android Studio

En la siguiente imagen se muestra la respuesta cuando los parámetros enviados a la API son correctos, esta respuesta son datos en formato *JSON*.

```

"success": {
  "code": 200,
  "text": "Ok",
  "description": "Success!"
},
"data": {
  "museum": {
    "id": 20,
    "name": "Museo de Geología",
    "description": "El Museo de Geología de la UNAM, ubicado en el corazón de la colonia Santa María La Ribera,
    "image": "1457975743.jpg",
    "map": "1459381571.png",
    "uuid": "252ea4f4-a959-4512-b43c-7edc533c70b0",
    "creation_datetime": "2016-02-28 04:19:36",
    "modification_datetime": "2016-03-30 23:46:11"
  },
  "exhibitions": [
    {
      "id": 1,
      "name": "Sala Principal",
      "description": "Al centro de esta sala, se observa la reconstrucción de un esqueleto de mamut, siendo éste l
      "image": "1457999470.jpg",
      "major": "1",
      "status": 1,
      "museum": null,
      "typeExhibition": {
        "id": 8,
        "name": "Sala"

```

Figura 155 Prueba respuesta de la API en Android Studio

Otras de las pruebas realizadas fueron mediante el navegador *Google Chrome* y un sistema operativo *Microsoft Windows 7*.

```

dev.politicaenlinea.com/api/museums/?demo=1&secret_key=fH0oh9RjiqoKtPeU1gkbZLTikx90WaBUBE4toH1DYQOkByz1JK&customer_key=eqEqDsAqnyjoZx6jbcM73vNI&uuid=252ea4f4-a5
{"error":{"code":401,"text":"No autorizado","description":"Las credenciales de autentificaci\u00f3n faltan o son incorrectas."}}
```

Figura 156 Error de credenciales API en navegador web

Por último, se muestra la respuesta cuando los parámetros enviados a la API son correctos, esta respuesta son datos en formato *JSON*.

```

dev.politicaenlinea.com/api/museums/?demo=1&secret_key=H0oh9RjiqoKtPeU1gkbZLTikx90WaBUBE4toH1DYQOkByz1JK&customer_key=eqEqDsAqnyjoZx6jbcM73vNI
{"success":{"code":200,"text":"Ok","description":"Success!"},"data":
[{"name":"Museo de Arte moderno","uuid":"37317e07-0031-8350-8a0d-1552b80075dd","image":"1458698048.jpg","map":"1455754548.png"},
{"name":"Museo de Geolog\u00eda","uuid":"252ea4f4-a959-4512-b43c-7edc533c70b0","image":"1457975743.jpg","map":"1459381571.png"},
{"name":"Museo de Historia Natural","uuid":"a3e0ef4e-5499-7194-5cde-baa3dd966dcc","image":"1458767213.jpg","map":"1459214986.jpg"}]}
```

Figura 157 Prueba de respuesta de la API en navegador web

7.3. Pruebas del servidor Web

Las pruebas fueron realizadas en un navegador web, en este caso se usó *Google Chrome* y en un sistema operativo *Windows 7*.

- **Inicio de Sesión**

Para acceder al sistema, se debe ingresar la dirección en donde se encuentra el servidor, la cual es <http://dev.politicaenlinea.com/admin/>

Al acceder a la dirección nos pide ingresar un nombre y una contraseña, en el caso de las pruebas el administrador tiene como nombre “*admin_tt*” y como contraseña “*admin_tt*”.

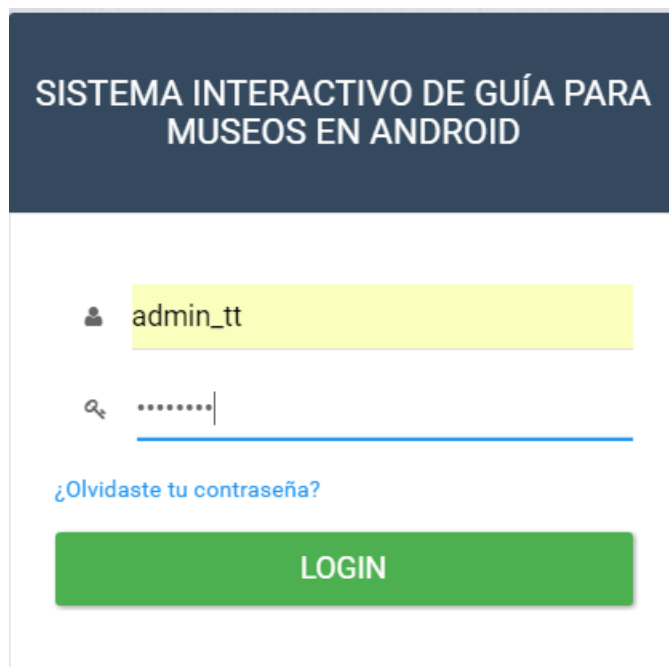


Figura 158 Login Servidor Web

Si se ingresa incorrectamente un campo, se despliega un mensaje de error.

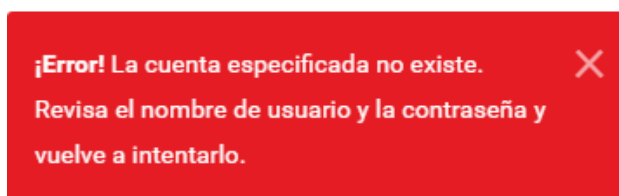


Figura 159 Error de Login en Servidor Web

- **Menú principal**

El servidor cuenta con un menú principal, ahí se tienen las opciones que se pueden gestionar, las cuales son:

- Museos
- Tipos
- Exhibiciones
- Plantillas
- Variables de Plantillas
- Materiales
- Archivos

Primeramente se comprobó que cada opción nos enviara a su respectiva sección.

Figura 160 Pruebas menú principal del Servidor Web

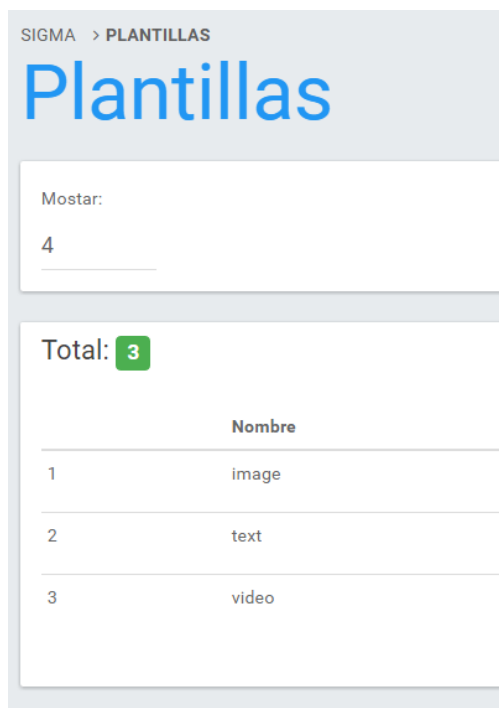


Figura 161 Pruebas acceso al menú Plantillas

- **Dar de Alta.**

Ya estando en una sección, se verificó que se pudieran dar de alta algún elemento a la base de datos, ya sea un museo, exhibición, material, etc.



Figura 162 Prueba alta de información en el Servidor Web

- **Alta de un Imagen.**

Además al sistema se podrá dar de alta no solo información, si no también imágenes. Cuando el campo de la imagen se encuentra vacío, su botón representativo estará de color gris, para hacer el alta, se presiona en el botón y saldrá para buscar la imagen para subir, se selecciona y en la opción de abrir. Con esto se comprueba que la imagen queda dada de alta en el servidor web. Al tener una imagen su botón se muestra de color morado.

Major	Museo	Tipo de Exhibición	Imagen
125	Museo de Arte moderno	Exposición Temporal	
15	Museo de Arte moderno	Exposición Temporal	
1267	Museo de Arte moderno	Exposición Temporal	

Figura 163 Prueba alta de imágenes en el servidor web

En caso de que no contar una imagen algún museo, el botón de imagen se muestra de color gris y no de color morado como se visualiza en la imagen.

- **Leer Datos.**

Al estar en una sección, se debe de leer los datos que se encuentran dentro de nuestra base de datos.

Se colocó un botón para buscar, en el cual seleccionas el número de información que se desea visualizar por pantalla, cuenta con unas flechas para poder navegar entre ellas.



Figura 164 Prueba cantidad de información a mostrar en el Servidor Web

La visualización de la información se vera de la siguiente manera

Total: 8		< INICIO >
Nombre		
1	Sala Principal	✖ 🔍
2	Exposición temporal	✖ 🔍
3	Exposición Permanente	✖ 🔍
4	Sala Especial	✖ 🔍

Figura 165 Prueba muestra de la información en el Servidor Web

Sera posible ver información más detallada presionando el botón de detalle.



Figura 166 Prueba de la visualización de la información a detalle en el servidor web

- **Editar Campos**

Una parte importante del servidor es que nos permite editar y actualizar la información de cada sección, estos cambios se ven reflejados, en el servidor web y en la aplicación móvil.

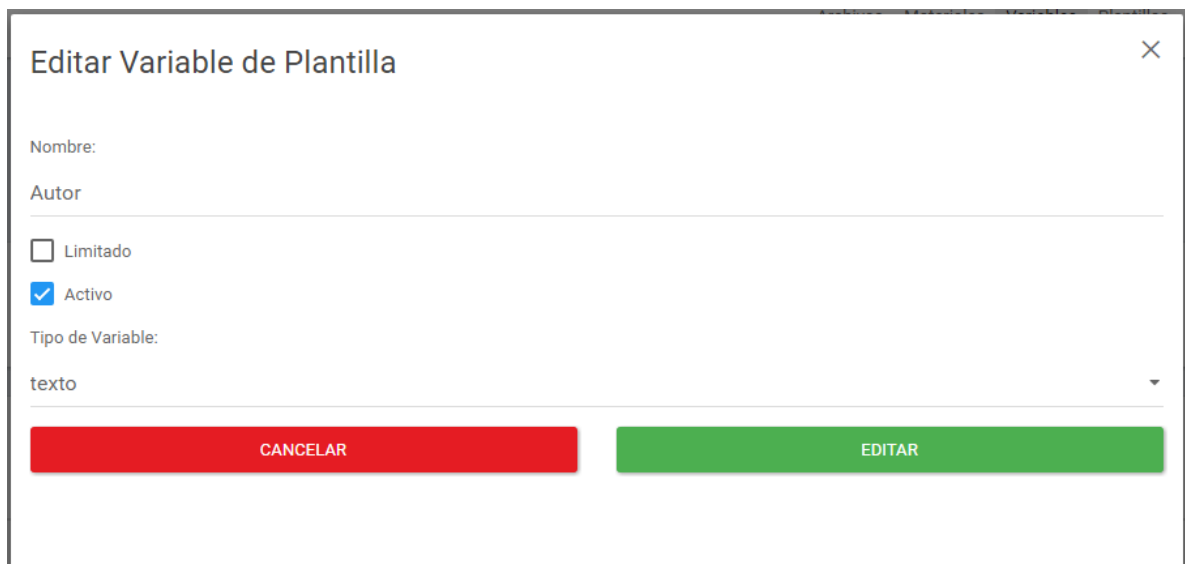


Figura 167 Prueba editar Variable de Plantilla en el Servidor Web

- **Eliminar Datos.**

Por ultimo tenemos el dar de baja la información que requiera el administrador, solamente se tendrá que presionar el botón de eliminar y la información seleccionada será borrada de la base de datos.



Figura 168 Prueba de eliminación de información del Servidor Web

7.4. Prueba de Concurrencia.

El sistema web se encuentra alojado en un Servidor Apache, para ello se utilizó la herramienta ApacheBench para testear y medir el rendimiento del mismo.

Para ejecutarlo se usó el comando **ab** con los parámetros del número de llamadas que en este caso fue de 400, y los hilos utilizados que fueron 60 para después poner la URL con sus parámetros de búsqueda.

```
root@PENDES02:~# ab -n 400 -c 60 http://dev.politicaenlinea.com/api/material?secret_key=H0oh9RjikoKtPeUlgkbZLTikkx90WaBUBE4toH1DYQ0kByz1jK
[1] 10270
[2] 10271
[3] 10272
[4] 10273
[5] 10274
[6] 10275
root@PENDES02:~# This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking dev.politicaenlinea.com (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Finished 400 requests
```

Figura 169 Ejecución ApacheBench

Esta capacidad de concurrencia nos permite comprobar las condiciones de carrera o bloqueos.

Los resultados obtenidos fueron los siguientes:

```
Server Software:      Apache/2.4.12
Server Hostname:     dev.politicaenlinea.com
Server Port:         80

Document Path:       /api/material?secret_key=H0oh9RjikoKtPeUlgkbZLTikkx90WaBUBE4toH1DYQ0kByz1jK
Document Length:     230 bytes

Concurrency Level:   60
Time taken for tests: 28.182 seconds
Complete requests:   400
Failed requests:     4
  (Connect: 0, Receive: 0, Length: 4, Exceptions: 0)
Non-2xx responses:   1
Total transferred:   283423 bytes
HTML transferred:   91711 bytes
Requests per second: 14.19 [#/sec] (mean)
Time per request:    4227.258 [ms] (mean)
Time per request:    70.454 [ms] (mean, across all concurrent requests)
Transfer rate:       9.82 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    66    71   3.3     71     82
Processing: 244 3997 1207.0  4399  5623
Waiting:    244 3996 1207.1  4398  5623
Total:      313 4067 1206.5  4466  5692

Percentage of the requests served within a certain time (ms)
 50%    4466
 66%    4850
 75%    4952
 80%    5025
 90%    5179
 95%    5306
 98%    5586
 99%    5627
100%    5692 (longest request)
```

Figura 170 Respuesta de Peticiones

Como se puede observar en la imagen, se muestra el nivel de concurrencia, el tiempo que tomo la prueba, los fallos de las pruebas, los bytes de transferencias que se usaron y el porcentaje de las peticiones en el tiempo determinado.

En resumen tuvimos que el nivel de concurrencia fue de 60 con 400 peticiones, en todas ellas hubo 4 fallos y se transfirieron 283423 bytes en total.

7.5. Análisis de las 6 M's

A continuación se muestra el Análisis de las 6M's de la aplicación móvil SIGMA.

Atributo	Calificación	Justificación
Movimiento	5	Para el funcionamiento de la aplicación es necesario estar en el museo de Geología y contar con un servicio de Internet para la obtención de información.
Momento	4	El usuario podrá abrir la aplicación en cualquier momento, pero la respuesta es variada dependiendo la ubicación en el museo.
Yo	4	La aplicación tiene una personalización de grado media, ya que el usuario podrá escoger entre las plantillas disponibles la que desea ver, los archivos si desea descargarlos o solo para su lectura, la configuración de poder recibir notificaciones cuando encuentre un material y poder ver las plantillas en distancia media.
Multi-Usuario	1	No hay ninguna interacción entre usuarios del sistema, ya que es una guía personal.
Dinero	5	Aunque la aplicación es de distribución gratuita, ésta genera ingresos al operador al ocasionar tráfico en la red de datos. También genera más visitas al museo de Geología y esto provoca un ingreso más para dar algún servicio al museo.
Maquinas	5	El servicio es soportado por dispositivos móviles que cuenten con la versión 4.3 de Android en adelante, una conexión a la red para mandar y recibir datos. Además de contar con Bluetooth 4.0. El usuario puede hacer uso de la aplicación sea en un celular o una tableta electrónica.

Tabla 39 Análisis de las 6 M's

CONCLUSIONES

Tras finalizar el desarrollo de este Trabajo Terminal, llegamos al momento de realizar el análisis y los resultados finales del proyecto.

Mediante este trabajo terminal logramos identificar los problemas que afectan generalmente a los museos, esto con el fin de poder ayudar estos espacios culturales en la integración de nuevas tecnologías y un mayor interés de la población, ya que estos espacios siguen siendo la fuente de información más fiable para los visitantes de cualquier tipo y condición.

La principal ventaja de nuestro sistema es que el uso de una aplicación móvil se correlaciona positivamente con los visitantes ya que desempeña funciones importantes y complementarias a la hora de cubrir una gran variedad de necesidades de información de las exposiciones.

A continuación mostraremos el análisis de los objetivos y si se han cumplido llegando a este parte final del proyecto:

Se ha logrado cumplir el objetivo general del proyecto, es decir, cumple con la funcionalidad que se deseaba dar en un inicio, ya que presenta la solución para un problema importante dentro del museo.

En los objetivos específicos también tenemos que se cumplieron satisfactoriamente ya que en la parte de la aplicación móvil nos permite la visualización sobre las exhibiciones y materiales que les fueron asignados a los dispositivos Beacon mediante las plantillas asignadas.

Por lo tanto, en la parte del servidor web se cumple con el poder administrar la información del museo, como lo son las Exhibiciones, Tipos de Exhibiciones, Materiales, Plantillas asignadas a cada material junto con sus variables y los Archivos que estarán disponibles.

Por ultimo en la parte de la conexión y descarga de toda la información que se requiere, la API que se creó cumple con las funciones que se requiere para poder realizar dicha tarea.

A pesar de que se hayan cumplido los objetivos marcados al inicio del proyecto, podemos concluir que el sistema puede crecer aún más, para poder aprovechar las tecnologías usadas como lo es el sistema operativo Android y los dispositivos Beacon.

También se presentaron varias complicaciones durante el desarrollo del proyecto y tuvimos la capacidad de resolverlos de una forma eficaz, ya que debemos estar preparados para cuando surja una situación similar en el ámbito laboral.

TRABAJO A FUTURO.

A continuación se describen las posibles mejoras propuestas como trabajo a futuro para mejorar la aplicación:

Interacción con redes sociales.

Para aprovechar el uso de las redes sociales, se propone usarlas para poder compartir su visita al museo, foto o una calificación hacia el mismo.

Ubicación dentro del Museo.

Se cuenta en la aplicación solamente la imagen de la distribución del museo, la propuesta para un futuro es poder indicar mediante un mapa la ubicación del usuario mientras está haciendo el recorrido en el museo.

Desarrollo para otros sistemas operativos móviles.

Actualmente la aplicación fue desarrollada para el sistema operativo Android y para que la aplicación pudiese llegar a manos de la gran mayoría de la población que cuenta con Smartphone, la propuesta es desarrollar para otros sistemas operativos como iOS o Windows Phone.

REFERENCIAS

- [1] Corina Varela Calvo¹ y Erik Stengler. “Los museos interactivos como recurso didáctico: El Museo de las Ciencias y el Cosmos” Consultada el 5 de Septiembre del 2014. Disponible en: http://reec.uvigo.es/volumenes/volumen3/REEC_3_1_2.pdf
- [2] Claudia Esther. “Interactividad y museos la experiencia del Museo Interactivo de Economía (MIDE) en la Ciudad de México”. Consultada el 5 de Septiembre del 2014. Disponible en: [http://www.gabinetecomunicacionyeducacion.com/files/adjuntos/Interactividad%20y%20museos%20%20la%20experiencia%20del%20museo%20interactivo%20de%20econom%C3%A9%20\(mide\)%20en%20la%20ciudad%20de%20M%C3%A9xico.pdf](http://www.gabinetecomunicacionyeducacion.com/files/adjuntos/Interactividad%20y%20museos%20%20la%20experiencia%20del%20museo%20interactivo%20de%20econom%C3%A9%20(mide)%20en%20la%20ciudad%20de%20M%C3%A9xico.pdf)
- [3] Sistema de Información Cultural. Consultada el 5 de Septiembre del 2014. Disponible en: http://sic.conaculta.gob.mx/index.php?estado_id=0&municipio_id=0&table=museo&disciplina=
- [4] *Diagnóstico informático para el aprovechamiento de las tecnologías de información para el instituto nacional de antropología e Historia (INAH)*. Consultada el 12 de Julio del 2015. Disponible en: http://www.inah.gob.mx/im7ages/stories/Transparencia/2014/diagnostico_unam%20_2008.pdf
- [5] Sistema de Información Cultural. Consultada el 5 de Septiembre del 2014. Disponible en: http://sic.conaculta.gob.mx/index.php?estado_id=0&municipio_id=0&table=museo&disciplina=
- [6] StarCounter Global Stats, “Top 8 Mobile & Tablet Operating Systems in Mexico from Oct 2014 to Oct 2015“. Consultada el 21 de Octubre del 2015”. Disponible en: <http://gs.statcounter.com/#mobile+tablet-os-MX-monthly-201410-201510>
- [7], [9], [10], [11] "El gran libro de Android", Jesús Tomas Girones, 3ª Edición, Alfaomega, Paginas (22, 23, 25, 27-29).
- [8] Evolución versiones de Android. Disponible en: https://www.android.com/intl/es-419_mx/history/
- [12] “Programación en Android”, Clodoaldo Robledo Sacristán y David Robledo Fernández, Ministerio de Educación, 2012, Paginas (93 y 94).
- [13], [14], [15]. ”Android 4: Principios del desarrollo de aplicaciones Java”, Nazim Benbourahala, Ediciones ENI, 2013, Paginas (206,207).
- [16], [17] Diploma de Especialista en Desarrollo de Aplicaciones para Android, “Componentes de una aplicación”, disponible en: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/149-componentes-de-una-aplicacion>
- [18] Magnus Jern. “Bluetooth Beacons vs. Wifi vs NFC”. Consultada el 8 de Septiembre del 2014. Disponible en: http://www.goldengekko.com/?blog_post=bluetooth-beacons-vs-wifi-vs-nfc

- [19] Apple Developers. "Submit your IOS 7 apps today". Consultada el 8 de Septiembre del 2014. Disponible en: <https://developer.apple.com/ios7/>
- [20] "Getting Started with iBeacon", Version 1.0. Disponible: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>
- [21] "Tecnología de comunicación de campo cercano (NFC) y sus aplicaciones" - Daniel Antonio Chavarría - http://eie.ucr.ac.cr/uploads/file/proybach/pb2011/pb2011_012.pdf
- [22] Beneficios Bluetooth 4.0. Disponible en: <https://www.unocero.com/2012/03/13/en-que-nos-beneficiara-el-bluetooth-4-0/>
- [23] D. Bell, M. Parr y A. V. R. Elizondo, Java para estudiantes, 2003.
- [24] arquitecturajava, "Introducción a servicios REST", Disponible en: <http://www.arquitecturajava.com/servicios-rest/>
- [25] desarrolloweb.com "Ventajas e inconvenientes de API REST para el desarrollo", Disponible en: <http://www.desarrolloweb.com/articulos/ventajas-inconvenientes-apirest-desarrollo.html>
- [26] Museo Rubens huis. Consultada el 11 de Septiembre del 2014. Disponible en: <http://blog.mobivery.com/2014/02/17/ibeacons-la-nueva-revolucion-esta-cerca/#more-4712>
- [27] iTunes Store, Estimote App. Consultada el 9 de Septiembre del 2014. Disponible en: <https://itunes.apple.com/mx/app/estimote/id686915066?mt=8>.
- [28] iTunes Store, Geogadgets App. Consultada el 9 de Septiembre del 2014. Disponible en: <https://itunes.apple.com/es/app/geogadgets/id855073677?l=en&mt=8>
- [29] Play Store, iBeacon Finder App. Consultada el 9 de Septiembre del 2014. Disponible en: <https://play.google.com/store/apps/details?id=com.mobisfera.prod.ibeaconfinder>
- [30] Shopkick App. Consultada el 9 de Septiembre del 2014. Disponible en: <http://www.shopkick.com/shopbeacon>
- [31] Walmart, Sistema de control remoto universal Griffin Beacon, Disponible en: <http://www.walmart.com.mx/Detalle-del-articulo/00068538732947/www.walmart.com.mx/Detalle-del-articulo/00068538732947/Sistema%20de%20Control%20Remoto%20para%20Apple%20Griffin%20Beacon%20Negro>
- [32] Metodología para el desarrollo de aplicaciones móviles – Maira Cecilia Gasca Mantilla - dialnet.unirioja.es/descarga/articulo/4778503.pdf
- [33] Risk IT, ISACA 2009- páginas 17, 18, 22, 27, 28
- [34] SG Buzz, Conocimiento Para Crear Software Grandioso "Estudio de Salarios 2014", Disponible en: <http://sg.com.mx/revista/46/estudio-salarios-2014#.VkGK5vkvfIU>
- [35] smartGSM, "Motorola Moto G (2nd Gen)", Disponible en: <http://www.smart-gsm.com/moviles/motorola-moto-g-2014>
- [36] smartGSM, "Samsung Galaxy Tab S2", Disponible en: <http://www.smart-gsm.com/moviles/samsung-galaxy-tab-s2-97>

[37] kontakt.io, “Kontakt.io Smart Beacon”, Disponible en: <https://store.kontakt.io/our-products/23-eddystone-smart-beacon.html>

[38] kontakt.io, “Knowledge Base” <https://support.kontakt.io/hc/en-gb/articles/201492492-iBeacon-advertising-packet-structure>

[39] MODX Revolution, “Login”, Disponible en: <https://rtfm.modx.com/extras/revo/login>

[40] MODX Revolution, “Wayfinder”, Disponible en: <https://rtfm.modx.com/extras/revo/wayfinder>

[41] MODX Revolution, “QuickCrumbs”, Disponible en: <https://rtfm.modx.com/extras/revo/quickcrumbs>

[42] MODX Revolution “Estructura de directorio”, Disponible en: <https://rtfm.modx.com/revolution/2.x/getting-started/an-overview-of-modx/glossary-of-revolution-terms/explanation-of-directory-structure#ExplanationofDirectoryStructure-assets%2F>