



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



## Trabajo Terminal



TT2012-A035

# Plataforma interactiva para sitios de interés cultural mediante códigos QR

Que para cumplir con la opción de titulación curricular en la carrera de

“Ingeniería en Sistemas Computacionales”

### Presentan:

Gutiérrez Silva Eduardo  
Luna Pérez Carlos Gabriel  
Sosa Rivera Jesús Norberto

### Directores:

M. en C. Pescador Rojas Miriam

M. en C. Franco Martínez Edgardo

Soy politécnico para alcanzar las conquistas universales y ofrecerlas a mi pueblo. Soy politécnico porque mi respetada libertad de joven y estudiante me impone la razón de respetar este recinto.  
Soy politécnico porque aspiro a ser todo un hombre. Soy politécnico porque exijo mis deberes antes que mis derechos. Soy politécnico por convicción y no por circunstancia.  
Soy politécnico porque me dignifico y siento el deber de dignificar a mi institución. Soy politécnico porque me duele la patria en mis entrañas y aspiro a calmar sus dolencias.



**"La Técnica al Servicio de la Patria"**



**INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO**



No. de Registro: 2012 - A035

Abril de 2013

Documento Técnico

**Plataforma interactiva para sitios de interés cultural mediante códigos QR**

**Presentan**

**Gutiérrez Silva Eduardo<sup>1</sup>  
Luna Pérez Carlos Gabriel<sup>2</sup>  
Sosa Rivera Jesús Norberto<sup>3</sup>**

**Directores**

**M. en C. Miriam Pescador Rojas  
M. en C. Edgardo Franco Martínez**

**RESUMEN**

En este reporte se presenta la documentación técnica del Trabajo Terminal 2012-A035 titulado “Plataforma interactiva para sitios de interés cultural mediante códigos QR”, cuyo objetivo se centra en el planteamiento para el desarrollo e implementación de una plataforma que otorgue interactividad para la administración, manipulación y visualización de información de las exposiciones disponibles de un determinado sitio de interés cultural. La propuesta contempla una aplicación móvil que realice el procesamiento de códigos QR para brindar información del sitio a los visitantes, dicha información será alojada en un servidor local del lugar.

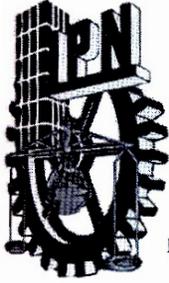
**Palabras clave** – Códigos QR, Dispositivos móviles, Programación web, Programación móvil, Bases de Datos.

---

<sup>1</sup> gt.silva.e@gmail.com

<sup>2</sup> gabo\_luna@msn.com

<sup>3</sup> alter32saos@gmail.com



**ESCUELA SUPERIOR DE CÓMPUTO**  
**SUBDIRECCIÓN ACADÉMICA**  
DEPARTAMENTO DE FORMACIÓN INTEGRAL E INSTITUCIONAL



**COMISIÓN ACADÉMICA DE TRABAJO TERMINAL**

México, D.F. a 3 de Junio de 2013

**ING. APOLINAR FCO. CRUZ LÁZARO**  
**PRESIDENTE DE LA COMISIÓN ACADÉMICA**  
**DE TRABAJO TERMINAL**  
**PRESENTE**

Por medio del presente, informamos que los alumnos que integran el **TRABAJO TERMINAL 2012-A035** titulado “**Plataforma interactiva para sitios de interés cultural mediante códigos QR**”, concluyeron satisfactoriamente su trabajo.

El empastado del Reporte Técnico Final y el Disco Compacto (CD) fueron revisados ampliamente por sus servidores y corregidos, cubriendo el alcance y el objetivo planeados en el protocolo original y de acuerdo a los requisitos establecidos por la comisión que Usted preside.

**ATENTAMENTE**

  
\_\_\_\_\_  
M. en C. Miriam Pescador Rojas

  
\_\_\_\_\_  
M. en C. Edgardo Franco Martínez

Directores

# 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 queda 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:

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

Sin duda alguna en el transcurso de estos cuatro años atravesé por muchos cambios, ya que todo lo que conlleva el estudiar en esta institución requiere de constancia, paciencia y muchas ganas de querer superarse. Las fuerzas necesarias para poder formar todas esas cosas no las conseguí sólo. Aunque se bien que por mi cuenta corría el tener la voluntad necesaria por continuar, se bien que recibí apoyo de muchas personas, desde quienes nunca imaginé que lo recibiría y obviamente de las personas que han estado conmigo desde que tengo memoria.

De mis padres aprendí a ser constante, a mantenerme siempre firme con mis capacidades. Ellos siempre trataron de comprender el hecho del poco tiempo que tenía disponible, pues en los casos en los que no podía estar con ellos, entendían que era un esfuerzo que debía realizar dada la importancia de estudios de este nivel.

Además del apoyo moral, el cual es el de mayor importancia para mí, agradezco a mis padres por el esfuerzo que ellos mismos pusieron para que tuviera las condiciones de desarrollarme correctamente durante toda la carrera. Madrugaban con migo y me daban el apoyo necesario para que no tuviera problemas en los recursos demandantes de esta carrera.

Mis amigos son otro pilar muy importante, de ellos aprendí que las oportunidades no se repiten y deben ser tomadas al instante. Al mismo tiempo me dieron los consejos necesarios para poder desarrollarme en mi vida personal y avanzar de manera correcta para mantener un equilibrio con lo laboral.

Agradezco todos sus consejos y apoyos.

De igual manera conocí a un sin número de grandes personas a lo largo de la carrera, siendo profesores, compañeros y algunos conocidos que me harían mejorar de diversas maneras, algunos de ellos se convertirían en nuevos amigos y me apoyarían a llegar hasta donde estoy ahora.

No me queda más que agradecer a todos los mencionados por su apoyo en los buenos y malos momentos y por último en una etapa de tan importante como lo han sido estos cuatro años en mi formación académica.

Eduardo Gutiérrez Silva

# Agradecimientos

A mis padres; María Eugenia Pérez Silva y Constanzo Luna Herrera, quienes son mi más grande admiración, gracias por todo el amor, cariño y comprensión que me han brindado, gracias por su dedicación y apoyo, por sacarme adelante, porque gracias a esto puedo cumplir cada una de mis metas. Gracias a ustedes hoy soy una persona de éxito y me encuentro feliz de que vean un logro más en mi vida. ¡Los amo!

A mi hermano Miguel Ángel Luna Pérez, por su apoyo incondicional en los momentos difíciles, por todos esos momentos llenos de felicidad y diversión. ¡Te quiero mucho hermano!

A Carolina Trinidad Vargas, mi novia, por llenar mi vida de amor y felicidad, por todo su apoyo en los momentos buenos y malos, y por su apoyo y comprensión que me brindo durante la carrera. ¡Te amo mi niña!

A mis tíos, que siempre me han demostrado su cariño y comprensión en todas las decisiones que he tomado, y por siempre brindarme su apoyo.

A mis profesores, porque me llevo la más grata experiencia de haber convivido con ellos y de haber aprendido lo que sé me servirá en mi futuro profesional. ¡Gracias!

¡SOY LO QUE SOY GRACIAS A TODOS Y CADA UNO DE USTEDES!

Carlos Gabriel Luna Pérez

# Agradecimientos

A mis padres; Angélica Rivera y Jesús Norberto Sosa García, quienes me han apoyado en cada etapa de mi vida.

A mi escuela, a mis profesores y sobre todo al Instituto Politécnico Nacional, por hacer posible que alumnos como yo pueda concluir sus estudios.

¡GRACIAS!

Sosa Rivera Jesús Norberto

## Índice

Índice.....	8
Índice de Figuras.....	12
Índice de Tablas.....	15
CAPÍTULO 1 INTRODUCCIÓN.....	17
1.1 Motivación.....	17
1.2 Descripción del problema.....	17
1.3 Solución propuesta.....	18
1.4 Objetivos.....	18
1.4.1 Objetivo general.....	18
1.4.2 Objetivos particulares.....	18
1.5 Complejidad del proyecto.....	19
1.6 Justificación.....	19
1.7 Alcances.....	19
1.7 Productos esperados.....	19
CAPÍTULO 2 MARCO TEÓRICO.....	20
2.1 Interactividad.....	20
2.2 Sitios de interés en México.....	20
2.2.1 Museos.....	21
2.3 Códigos QR.....	23
2.3.1 Características básicas de los códigos QR.....	24
2.3.2 Almacenamiento de la información.....	25
2.3.3 Recuperación de Errores.....	26
2.3.4 Impresión y marcaje del símbolo.....	27
2.3.5 Uso de códigos QR.....	27
2.4 Frameworks.....	29
2.4.1 Herramientas de desarrollo.....	30
2.4.2 El problema de los frameworks.....	30
2.5 Cómputo móvil.....	30
2.5.2 Dispositivos de cómputo móvil.....	31
2.5.3 Características.....	32
2.5.4 Conectividad.....	33

2.6 Estado del arte .....	33
2.6.1 Herramientas museológicas digitales .....	33
2.6.2 La interactividad en los museos .....	34
CAPÍTULO 3 ANÁLISIS .....	36
3.1 Metodología .....	36
3.1.1 Metodología en V .....	36
3.2 Códigos QR.....	37
3.2.1 Generación de códigos QR.....	37
3.2.1 Decodificación de códigos QR.....	40
3.3 Plataformas móviles .....	42
3.3.1 Estadísticas de uso.....	43
3.3.2 Arquitectura de las plataformas móviles .....	44
3.3.3 Comparativa de las plataformas móviles .....	48
3.3.4 Frameworks para desarrollo en plataformas móviles.....	49
3.4 Tecnologías del servidor .....	50
3.4.1 Programación del sitio web .....	50
3.4.2 Justificación de la elección de las tecnologías del servidor .....	51
3.5 Servidor de aplicaciones .....	51
3.5.1 Comparativa .....	51
3.5.2 Justificación de la elección del servidor de aplicaciones .....	52
3.6 Entornos de Desarrollo Integrados (IDEs).....	52
3.6.1 Comparativa .....	52
3.6.2 Justificación de la elección del IDE .....	53
3.7 Sistema de Gestión de Base de Datos .....	53
3.7.1 Comparativa .....	54
3.7.2 Justificación de la elección del SGBD .....	55
3.8 Servicios Web .....	55
3.8.1 Sistemas orientados a servicios.....	55
3.8.2 Definición de un servicio web.....	56
3.8.3 Estándares en los servicios web .....	56
3.8.4 Tipos de servicios web .....	57
3.8.5 Elección de servicios web .....	58

3.8.6 Comparativa de framework .....	58
3.8.7 Justificación de la elección del framework .....	58
3.9 Requerimientos .....	58
3.9.1 Requerimientos funcionales .....	59
3.9.2 Requerimientos no funcionales .....	60
3.10 Casos de uso .....	60
3.10.2 Diagramas de casos de uso .....	62
3.10.3 Especificación de casos de uso .....	66
3.11 Análisis de factibilidad .....	89
3.11.1 Factibilidad técnica .....	89
3.11.2 Factibilidad operativa .....	93
3.11.3 Factibilidad económica .....	97
3.12 Análisis de riesgos .....	98
3.12.1 Definición de riesgos .....	98
3.12.2 Tabla de riesgos .....	99
3.12.3 Plan de riesgos .....	99
CAPÍTULO 4 DISEÑO .....	105
4.1 Arquitectura del sistema .....	105
4.1.1 Arquitectura general del sistema .....	105
4.1.2 Arquitectura del sistema - sitio web .....	105
4.1.3 Arquitectura del sistema - Dispositivo móvil .....	107
4.2 Diagramas de secuencia .....	109
4.2.1 Sitio web .....	109
4.2.2 Dispositivo Móvil .....	118
4.3 Diseño de la base de datos .....	121
4.3.1 Diagrama Entidad-Relación .....	121
4.4 Diagrama de clases .....	122
4.4.1 Diagrama de clases Web .....	123
4.4.2 Diagrama de clases Service .....	124
4.4.3 Diagrama de clases Model .....	126
4.4.4 Diagrama de clases Exceptions .....	129
CAPÍTULO 5 DESARROLLO .....	130

5.1 Sitio web.....	130
5.2 Dispositivo móvil.....	141
5.3 Vistas del sistema.....	148
5.3.1 Sitio web.....	148
5.3.2 Dispositivo móvil.....	162
CAPÍTULO 6 PRUEBAS Y RESULTADOS.....	174
6.1 Métricas.....	174
6.1.1 Desarrollo y aplicación de cuestionarios.....	174
6.1.2 Aplicación de las 8 reglas del diseño de interfaces gráficas .....	174
6.2 Pruebas.....	175
6.3 Resultado de las Pruebas.....	177
6.4 Cuestionarios de Calidad.....	178
CAPÍTULO 7 CONCLUSIONES Y TRABAJO A FUTURO.....	180
7.1 Conclusiones .....	180
7.2 Trabajo a futuro.....	182
ANEXOS.....	183
I Diccionario de Datos .....	183
REFERENCIAS BIBLIOGRÁFICAS.....	188

## Índice de Figuras

Figura 1 Comparación entre código de barra y código QR .....	23
Figura 2 Código QR .....	23
Figura 3 Descripción de las partes de <i>quiet zone</i> de un código QR .....	25
Figura 4 Descripción de la metodología en V .....	36
Figura 5 Dirección de colocación de los módulos con base a las filas que forman el código .....	39
Figura 6 Colocación de los bits dentro de las filas .....	39
Figura 7 Colocación final de los bits dentro de la región de datos.....	39
Figura 8 Algoritmo de referencia para la lectura de códigos QR .....	40
Figura 9 Línea de escaneo en el patrón de detección de posición .....	41
Figura 10 Sistemas Operativos más usados en México. Sep 2011 a Sep 2012 .....	44
Figura 11 Distribución actual de Android. Basada en el número de dispositivos que han accedido a Google Play en un periodo de 14 días. ....	44
Figura 12 Arquitectura de Android .....	45
Figura 13 Capa de Abstracción de Hardware .....	46
Figura 14 Arquitectura de iOS.....	47
Figura 15 Diagrama de Caso de uso - General .....	62
Figura 16 Diagrama de Caso de uso - Sitio Web.....	63
Figura 17 Diagrama de Caso de uso - Gestión de Catálogo .....	63
Figura 18 Diagrama de Caso de uso - Gestión de Objeto .....	64
Figura 19 Diagrama de Caso de uso - Gestión de Usuarios .....	64
Figura 20 Diagrama de Caso de uso - Dispositivo Móvil .....	65
Figura 21 Arquitectura General del Sistema .....	105
Figura 22 Arquitectura del Sistema - Sitio Web .....	105
Figura 23 Arquitectura del sistema - Captura de objetos.....	106
Figura 24 Arquitectura del sistema - Gestión de usuarios.....	106
Figura 25 Arquitectura del sistema - Gestión de catálogos.....	107
Figura 26 Arquitectura del sistema - Dispositivo móvil.....	107
Figura 27 Arquitectura del sistema - Carga de información .....	108
Figura 28 Arquitectura del sistema - Procesamiento de la información .....	108
Figura 29 Diagrama de secuencia - Iniciar Sesión en Sitio Web .....	109
Figura 30 Diagrama de secuencia - Cerrar Sesión en Sitio Web .....	110
Figura 31 Diagrama de secuencia - Registrar Usuario .....	110
Figura 32 Diagrama de secuencia - Consultar Usuario .....	111
Figura 33 Diagrama de secuencia - Modificar Usuario .....	111
Figura 34 Diagrama de secuencia - Borrar Usuario .....	112
Figura 35 Diagrama de secuencia - Crear Catálogo .....	112
Figura 36 Diagrama de secuencia - Consultar Catálogo.....	113
Figura 37 Diagrama de secuencia - Modificar Catálogo.....	113
Figura 38 Diagrama de secuencia - Borrar Catálogo .....	114
Figura 39 Diagrama de secuencia - Registrar Objeto.....	114
Figura 40 Diagrama de secuencia - Consultar Objetos .....	115
Figura 41 Diagrama de secuencia - Modificar Objeto .....	115
Figura 42 Diagrama de secuencia - Borrar Objeto.....	116
Figura 43 Diagrama de secuencia - Generar código QR .....	116
Figura 44 Diagrama de secuencia - Generar código QR .....	117

Figura 45 Diagrama de secuencia - Generar código QR .....	117
Figura 46 Diagrama de secuencia - Iniciar Sesión en Dispositivo Móvil .....	118
Figura 47 Diagrama de secuencia - Cerrar Sesión en Dispositivo Móvil.....	118
Figura 48 Diagrama de secuencia - Escanear código QR .....	119
Figura 49 Diagrama de secuencia - Modificar Datos en el Editor .....	119
Figura 50 Diagrama de secuencia - Enviar Información a Correo.....	120
Figura 51 Diagrama Entidad – Relación.....	121
Figura 52 Diagrama de clases a nivel de paquetes .....	122
Figura 53 Diagrama de clases Web .....	123
Figura 54 Diagrama de clases Service .....	124
Figura 55 Diagrama de clases Service -utilities.....	125
Figura 56 Diagrama de clases Model – dao y jpa.....	126
Figura 57 Diagrama de clases Model - domain .....	127
Figura 58 Diagrama de clases Model - dto.....	128
Figura 59 Diagrama de clases Exceptions .....	129
Figura 60 Flujo de Registro de Objetos .....	139
Figura 61 Flujo de generación de código QR.....	140
Figura 62 Cuerpo html del Phonegap .....	142
Figura 63 Pantalla de Inicio de Sesión .....	148
Figura 64 Pantalla de bienvenida del administrador .....	149
Figura 65 Pantalla de bienvenida del capturista.....	150
Figura 66 Pantalla de gestión de usuarios.....	151
Figura 67 Pantalla de registro de usuarios.....	152
Figura 68 Pantalla de modificación de usuarios .....	153
Figura 69 Pantalla de gestión de catálogo .....	154
Figura 70 Pantalla de crear catálogo .....	155
Figura 71 Pantalla de modificación de catálogo.....	156
Figura 72 Pantalla de gestión de objetos (Administrador) .....	157
Figura 73 Pantalla de gestión de objetos (Capturista) .....	158
Figura 74 Pantalla de registrar objeto .....	159
Figura 75 Pantalla de modificación de objeto .....	160
Figura 76 Pantalla de generar código QR.....	161
Figura 77 Pantalla de bienvenida .....	162
Figura 78 Pantalla de registro de correo.....	163
Figura 79 Icono de la cámara .....	164
Figura 80 Icono del recorrido.....	164
Figura 81 Icono de editar .....	165
Figura 82 Icono de enviar.....	165
Figura 83 Icono de ajustes.....	166
Figura 84 Pantalla de inicio de sesión .....	167
Figura 85 Pantalla del escáner .....	168
Figura 86 Pantalla del recorrido .....	169
Figura 87 Pantalla del editor.....	169
Figura 88 Pantalla de la lista de objetos.....	170
Figura 89 Pantalla de envío de información al correo electrónico .....	171
Figura 90 Pantalla de principal de ajustes .....	172

<b>Figura 91 Pantalla de cambio de correo.....</b>	<b>172</b>
<b>Figura 92 Pantalla de ajustes (Ver nuevamente el Tutorial) y Pantalla de configuraciones avanzadas</b>	<b>173</b>
<b>Figura 93 Encargado del museo usando el sistema móvil.....</b>	<b>176</b>
<b>Figura 94 Encargado del museo usando el sistema web .....</b>	<b>176</b>
<b>Figura 95 Equipo de desarrollo durante las pruebas.....</b>	<b>177</b>
<b>Figura 96 Gráfica de resultado de la métrica de cuestionarios.....</b>	<b>179</b>

## Índice de Tablas

Tabla 1 Tipos de código y porcentaje de recuperación .....	27
Tabla 2 Comparación de las arquitecturas de Android y iOS .....	49
Tabla 3 Plataformas soportadas por PhoneGap .....	50
Tabla 4 Comparación de servidores de aplicaciones .....	52
Tabla 5 Comparación de IDEs .....	53
Tabla 6 Comparación de Sistemas de Gestión de Base de Datos .....	55
Tabla 7 Comparación de Frameworks de Servicios Web de tipo “Big” .....	58
Tabla 8 Requerimientos Funcionales - Sitio Web .....	59
Tabla 9 Requerimientos Funcionales - Dispositivo Móvil.....	60
Tabla 10 Requerimientos No Funcionales.....	60
Tabla 11 Descripción de Actores .....	61
Tabla 12 Caso de uso 1 - Iniciar Sesión en Sitio Web .....	66
Tabla 13 Caso de uso 2 - Cerrar Sesión en Sitio Web.....	67
Tabla 14 Caso de uso 3 – Registrar Usuario .....	68
Tabla 15 Caso de uso 4 – Consultar Usuarios .....	69
Tabla 16 Caso de uso 5 – Modificar Usuario .....	70
Tabla 17 Caso de uso 6 – Borrar Usuario .....	71
Tabla 18 Caso de uso 7 - Crear Catálogo.....	72
Tabla 19 Caso de uso 8 - Consultar Catálogo .....	73
Tabla 20 Caso de uso 9 - Modificar Catálogo .....	74
Tabla 21 Caso de uso 10 – Borrar Catálogo.....	75
Tabla 22 Caso de uso 11 - Registrar Objeto.....	76
Tabla 23 Caso de uso 12 - Consultar Objetos .....	77
Tabla 24 Caso de uso 13 – Modificar Objeto .....	78
Tabla 25 Caso de uso 14 – Borrar Objeto .....	79
Tabla 26 Caso de uso 15 – Generar código QR.....	80
Tabla 27 Caso de uso 16 – Cambiar Contraseña .....	81
Tabla 28 Caso de uso 17 – Recuperar Contraseña.....	82
Tabla 29 Caso de uso 18 - Iniciar Sesión en Dispositivo Móvil.....	83
Tabla 30 Caso de uso 19 – Cerrar Sesión en Dispositivo Móvil.....	84
Tabla 31 Caso de uso 20 – Escanear código QR .....	85
Tabla 32 Caso de uso 21 – Modificar Datos en el Editor .....	86
Tabla 33 Caso de uso 22 – Enviar Información a Correo .....	87
Tabla 34 Caso de uso 23 – Modificar Datos en el Editor .....	88
Tabla 35 Especificación de costos de software .....	89
Tabla 36 Especificaciones técnicas de laptop Dell Vostro 3460 .....	90
Tabla 37 Especificaciones técnicas de notebook Acer Aspire One 721-3155.....	90
Tabla 38 Especificaciones técnicas de laptop MacBook Pro.....	91
Tabla 39 Especificaciones técnicas de mini laptop Sony Vaio Serie E.....	91
Tabla 40 Especificaciones técnicas del dispositivo con Sistema Operativo Android .....	91
Tabla 41 Especificaciones técnicas del dispositivo con Sistema Operativo iOS .....	92
Tabla 42 Estimación de equipo de cómputo a utilizar .....	92
Tabla 43 Estimación de costos de servicios .....	92
Tabla 44 Factores de peso de complejidad para tipos de objetos .....	94

Tabla 45	Análisis de tipos de objetos en cuanto a cantidad y pesos totales .....	94
Tabla 46	Relación entre los ratios de productividad y pesos de acuerdo a su complejidad .....	95
Tabla 47	Evaluación de los ratios de productividad de acuerdo al proyecto .....	95
Tabla 48	Actividades y responsabilidades de las personas encargadas de la plataforma .....	96
Tabla 49	Descripción de actividades del modelo de desarrollo software y perfil de roles de desarrollador .....	97
Tabla 50	Hardware necesario para el desarrollo de las plataformas .....	97
Tabla 51	Especificación de costos de software .....	97
Tabla 52	Precio de los servicios/productos para desarrollo del sistema .....	98
Tabla 53	Costos Totales para el desarrollo del trabajo terminal.....	98
Tabla 54	Tabla de riesgo .....	99
Tabla 55	Plan de riesgo - PRT1.....	100
Tabla 56	Plan de riesgo - PRT2.....	100
Tabla 57	Plan de riesgo - PRT3.....	100
Tabla 58	Plan de riesgo - PRT4.....	101
Tabla 59	Plan de riesgo - PRE1 .....	101
Tabla 60	Plan de riesgo - PRE2.....	101
Tabla 61	Plan de riesgo – PRE3 .....	102
Tabla 62	Plan de riesgo - PRO1 .....	102
Tabla 63	Plan de riesgo - PRO2 .....	102
Tabla 64	Plan de riesgo - PRD1.....	103
Tabla 65	Plan de riesgo - PRD2.....	103
Tabla 66	Plan de riesgo - PRD3.....	104
Tabla 67	Tiempos de respuesta .....	177
Tabla 68	Métrica de evaluación de cuestionarios .....	178
Tabla 69	Resultado de métrica de cuestionarios .....	178
Tabla 70	Tablas del Sistema .....	183
Tabla 71	Tabla bitacora .....	183
Tabla 72	Indexes bitacora .....	184
Tabla 73	Tabla catalogo .....	184
Tabla 74	Indexes catalogo .....	184
Tabla 75	Tabla columnacatalogo .....	184
Tabla 76	Indexes columnacatalogo .....	185
Tabla 77	Tabla columnaobjeto.....	185
Tabla 78	Indexes columnaobjeto.....	185
Tabla 79	Tabla objeto.....	185
Tabla 80	Indexes objeto.....	185
Tabla 81	Tabla objetorecorrido .....	186
Tabla 82	Indexes objetorecorrido .....	186
Tabla 83	Tabla objetorecorrido_columnaobjeto .....	186
Tabla 84	Indexes objetorecorrido_columnaobjeto .....	186
Tabla 85	Tabla recorrido .....	187
Tabla 86	Indexes recorrido .....	187
Tabla 87	Tabla usuario.....	187
Tabla 88	Indexes recorrido .....	187

# CAPÍTULO 1 INTRODUCCIÓN

El uso de internet tiene un impacto profundo en el mundo laboral, el ocio y el conocimiento a nivel mundial. Gracias a la web, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información en línea. Esta tecnología ha llegado a gran parte de los hogares y de las empresas de los países desarrollados. No obstante, en el transcurso del tiempo se ha extendido el acceso a Internet en casi todas las regiones del mundo. Desde una perspectiva cultural del conocimiento, Internet ha sido una ventaja y una responsabilidad. Para la gente que está interesada en otras culturas, la red de redes proporciona una cantidad significativa de información y de una interactividad que sería inasequible de otra manera.

En los últimos años, el creciente desarrollo tecnológico del Internet ha provocado que las personas dejen de buscar información en medios a los que antes se acostumbraba, como los libros. Además, han dejado de concurrir a sitios de interés cultural en el país, por ejemplo a museos, zoológicos y/o monumentos. Debido a la facilidad de acceso a la información que ofrece el Internet sin salir de casa.

## 1.1 Motivación

La motivación para el desarrollo de este trabajo terminal involucra las visitas que se llevan a cabo dentro de los sitios de interés cultural, tomando en cuenta la experiencia del visitante al recorrer las exposiciones disponibles, así como la finalidad de su visita[1].

Existe un estrecho vínculo entre educación y cultura, podemos destacar que los jóvenes del país son los que tienen una mayor demanda de información. El hecho de que las personas que más asisten a espacios culturales sean jóvenes y estudiantes en su mayoría, representa una enorme ventana de oportunidades<sup>4</sup>.

Es así que podemos definir la principal motivación de como el mejoramiento de la experiencia de los visitantes a un sitio de interés cultural.

## 1.2 Descripción del problema

El problema planteado se encuentra relacionado con la experiencia del visitante a una exposición, la cual es diseñada con base en el tema y contenido que se presentará. Esta tarea comprende todo un proceso de administración de las muestras y su presentación. Dado que el sistema plantea una nueva forma de interactuar para el visitante con la información mostrada en la exposición, se debe tener la capacidad de gestionar los artículos en exposición, que bien pueden tener características variadas. Posteriormente se deberá manejar la integración del sistema con la exposición, el cual proporcionará información adicional al visitante mediante la lectura de códigos QR usando su propio dispositivo. El dispositivo móvil requerirá de una conexión local al sitio de interés para realizar este

---

<sup>4</sup> Encuesta nacional de prácticas y consumo culturales , CONACULTA, 2006

proceso y generar así un recorrido virtual de los artículos que han sido del interés del visitante.

La limitante de los dispositivos móviles son sus recursos por lo que la conexión móvil-servidor deberá ser permanente mientras se encuentre en el sitio de interés, teniendo la capacidad de manejar errores de conexión sin que los datos del usuario se pierdan, así como un envío de la información final evitando duplicados o datos innecesarios para el usuario.

La plataforma deberá gestionar grandes cantidades de información (generación de códigos QR e integración con la exposición), el procesamiento de diversas peticiones de los dispositivos móviles de los visitantes en el sitio de interés para la generación de recorridos virtuales independientes de cada visitante, el envío de la información capturada al correo del visitante, evitando duplicados con un control de errores sobre las posibles pérdidas de comunicación entre el móvil y el servicio local.

### **1.3 Solución propuesta**

Se plantea el desarrollo de una plataforma para la gestión de la información mostrada en exposición, así como de la introducción de una nueva forma de interactividad dentro de la experiencia del visitante del sitio de interés. La plataforma será capaz de generar códigos QR de cada una de los objetos pertenecientes a la exhibición según se requiera. El visitante interactuará con los datos mediante la lectura de los símbolos haciendo uso de la aplicación instalada en su propio dispositivo móvil. La aplicación móvil será capaz de acceder a la información perteneciente a objetos de interés del usuario, la cual se encontrará almacenada en el servidor.

### **1.4 Objetivos**

#### **1.4.1 Objetivo general**

- ✓ Desarrollar una plataforma para la administración de la información de las exposiciones de un sitio de interés cultural y el desarrollo de una plataforma interactiva mediante el uso de códigos QR y dispositivos móviles.

#### **1.4.2 Objetivos particulares**

Para cumplir con el objetivo general se propone lo siguiente:

- ✓ Desarrollar una herramienta web de administración de los objetos del sitio de interés cultural que harán uso de los códigos QR.
- ✓ Hacer uso eficiente de algoritmos de lectura de códigos QR en el dispositivo móvil con sistema operativo Android y/o iOS.
- ✓ Implementar un algoritmo que permita el manejo de datos entre el dispositivo móvil y el servidor.
- ✓ Implementar un módulo en el dispositivo móvil que permita la manipulación de la información del sitio de interés cultural.

- ✓ Implementar la propuesta en un caso de estudio real.

### **1.5 Complejidad del proyecto**

La propuesta de Trabajo Terminal es hacer uso de un sistema web, creando con una aplicación que administre la información de los sitios de interés cultural y así mantener informada a la población que haga uso de ésta.

Se pretende hacer uso de los dispositivos móviles para ofrecer a los usuarios una aplicación capaz de obtener la información que le parezca útil del sitio de interés cultural que visite y haga uso de esta plataforma, para posteriormente generarle algún reporte que pueda manipular a su gusto.

### **1.6 Justificación**

Aprovechar la tendencia al uso de las nuevas tecnologías, como los dispositivos móviles, capaces de tomar la lectura de los códigos QR, realizando una serie de procesos internos, aprovechando las capacidades de cómputo de estos dispositivos, así como de mantener comunicación entre el servidor web y la aplicación; como resultado final el visitante de los sitios de interés cultural tenga un fácil acceso a la información que proveen estos sitios, además de poder manipular esta información como le convenga al visitante. Así como de proveer una herramienta de gestión de la información al sitio de interés cultural para mantener un control de lo que se muestra a los visitantes y la información que se les entrega a los mismos.

### **1.7 Alcances**

El sistema a desarrollar está limitado en tiempo y costo, por lo que solo está planeado para un funcionamiento básico el cuál abarca hasta la etapa de aplicar la plataforma en un caso de estudio definido. Así mismo, la plataforma en los dispositivos móviles solo funcionará adecuadamente en equipos de gama media-alta.

### **1.7 Productos esperados**

Los artefactos de software que se presentarán son:

1. El desarrollo de la plataforma.
2. Ingeniería de Software analizada y planteada.
3. Manual técnico.
4. Manual de usuario.
5. Reporte de Trabajo Terminal.
6. Implementar la propuesta en un caso de estudio real.

## CAPÍTULO 2 MARCO TEÓRICO

El presente capítulo describe el estado del arte relacionado con las principales tecnologías que han sido incorporadas a las actividades que se llevan a cabo en sitios de interés y que involucran el concepto de interactividad, así como proyectos similares que se han o pretenden desarrollar y cuyos objetivos son similares a los ya presentados.

### 2.1 Interactividad

“Existen tres conceptos importantes: interacción, interactivo e interactividad. Una interacción involucra al menos dos participantes. En el contexto de la interacción humano-computador, el humano está interactuando con el computador. Se dice que algo es interactivo si permite la interacción. El término interactividad denota los aspectos interactivos de un objeto. El término interactividad puede ser usado para la descripción de un fenómeno en general o como una propiedad” [2].

Hoy en día la tecnología presenta diversas formas de interacción, principalmente las computadoras. Por ejemplo una computadora requiere la manipulación de un puntero en su pantalla. Otros ejemplos de artefactos que en su momento fueron nuevas formas de interacción son: La manipulación de objetos gráficos mediante un puntero de luz (SketchPad, 1963), ventanas, el mismo concepto que manejan muchos de los entornos de escritorio actualmente.

La interactividad se extiende a tipos de aplicaciones como los programas de dibujo, la edición de texto, el diseño asistido por computador y los videojuegos. Uno de los conceptos que cobra mayor fuerza en interactividad que se está popularizando es la realidad aumentada. Decimos únicamente que se populariza, pues sus orígenes datan de 1971[3].

### 2.2 Sitios de interés en México

Dentro de los sitios de interés cultural en México, la siguiente lista de recintos culturales es considerada por CONACULTA: [4]

- ✓ Bibliotecas
- ✓ Museos
- ✓ Sitios arqueológicos
- ✓ Casas de cultura y centros culturales
- ✓ Cines
- ✓ Teatros
- ✓ Espacios para presentaciones de música
- ✓ Librerías
- ✓ Salas de lectura

Sin embargo, los sitios de interés a los que está orientado este trabajo terminal, en un principio no abarcan todos los mencionados. Entre los lugares en los que tendría un mejor

desempeño con base en su definición se tienen: museos, exposiciones temporales, sitios arqueológicos, casas de cultura y centros culturales, además de los zoológicos.

### **2.2.1 Museos**

Los museos son uno de los sitios más interesantes en el proceso de desarrollo de este sistema, ya que el ambiente en el que se presenta la información a sus visitantes es de vital importancia.

Un museo está definido por el Consejo Internacional de Museos (ICOM por sus siglas en inglés) como una institución permanente, sin finalidad lucrativa, al servicio de la sociedad y su desarrollo, abierta al público, que adquiere, conserva, investiga, comunica y exhibe para fines de estudio, educación y deleite, testimonios materiales del hombre y su entorno[5]. Dentro de esta misma definición se presentan los museos designados a las instituciones que presenten especímenes vivos, como jardines botánicos y zoológicos, acuarios, viveros, entre otros.

Partiendo de lo anterior y destacando como parte de sus objetivos la comunicación y exhibición, a continuación se explican conceptos de importancia en el planteamiento de actividades de un museo.

Tomando como base las técnicas de las que hacen uso los museos para la presentación de la información y organización de sus exposiciones, consideramos importante describir los conceptos que se manejan en este entorno.

#### **2.2.1.1 Museología**

“La Museología es la ciencia del museo; estudia la historia y la razón de ser de los museos, su función en la sociedad, sus peculiares sistemas de investigación, educación y organización, la relación que guarda con el medio ambiente físico y la clasificación de los diferentes tipos de museos”. [5]

De esta definición destaca la mención en los sistemas de organización, su relación con el medio ambiente físico y la clasificación de los distintos tipos de museos, ya que son las partes principales en las que se desempeña el sistema. Este trabajo terminal pretende analizar estas características y usarlas para desarrollar un software, en lo que posteriormente se conocerá como una herramienta museológica digital.

Destacamos la relación del museo con el ambiente físico, en nuestro caso la interacción de los visitantes a una exposición con el ambiente que esta crea, aportando la interactividad con el uso de códigos QR.

#### **2.2.1.2 Museografía**

La museografía es la técnica que expresa los conocimientos museológicos en el museo. Trata especialmente sobre la arquitectura y ordenamiento de las instalaciones científicas de los museos.

Lo anterior, nos exige tomar en cuenta para las consideraciones del sistema. Ya se cuenta con toda una serie de técnicas para la presentación de exposiciones en los museos, las cuales ya han sido explotadas con anterioridad y evolucionadas con la inclusión de las nuevas tecnologías, tal y como se verá más adelante.

Dentro de las funciones de la museografía destacamos las siguientes: [6]

1. Coordinar y apoyar la elaboración de proyectos expositivos haciendo énfasis en la creación de narrativas espaciales que generen experiencias en el visitante.
2. Coordinar y ejecutar los montajes de las exposiciones que se realicen dentro y fuera del museo.
3. Emitir consentimientos técnicos junto con el área de diseño para la realización de exposiciones por parte de personas, dependencias e instituciones en la sala temporal del museo.
4. Investigar acerca de la utilización de nuevos materiales que permitan generar nuevas alternativas en la producción de dispositivos de exhibición y conservación que faciliten el que hacer técnico, comunicativo, educativo y científico del museo.
5. Elaborar y coordinar proyectos de actualización de la exposición permanente del museo.

Los anteriores son puntos tomados como ejemplo que lleva a cabo el área de trabajo de museografía del Museo de Ciencias Naturales de La Salle.

Con ello comenzamos a orientar aspectos del sistema relacionados con que tipo de personas se encargan de estos aspectos dentro de un museo, así como ciertas actividades que se deben realizar en las exposiciones dentro de los sitios de interés.

### **2.2.1.3 Recursos museográficos**

Dentro de una exposición cultural los recursos museográficos son los que determinan el curso de la visita al sitio de interés. Un recurso museográfico es un elemento narrativo que desempeña un papel fundamental en la construcción de los discursos museográficos, en el que se aprovechan todas las opciones multimedia que lo componen[7].

La definición de recursos museográficos se puede especializar a lo audiovisual. Se consideran recursos audiovisuales cualquier equipamiento, dispositivo o medio que transmite un mensaje determinado utilizando los códigos y la gramática narrativa propios del lenguaje audiovisual y, en su caso, multimedia. Siendo la definición anterior cambiante de acuerdo a la exposición y herramientas tecnológicas que se incluyan y cataloguen como recurso museográfico en ella.

El sistema puede catalogarse como una fuente audiovisual, al mismo tiempo que una herramienta para el desarrollo del discurso museográfico de la exposición.

## 2.3 Códigos QR

El Código QR (*Quick Response*), o código de respuesta rápida, es un sistema de información que supone una evolución del código de barras. El código QR es una imagen bidimensional (ver Figura 2) que almacena la información en una matriz de puntos que, a diferencia de los códigos de barras que se leen en una única dirección, contienen información en dos direcciones: vertical y horizontal.



Figura 1 Comparación entre código de barra y código QR

Los códigos QR son símbolos matriciales consistentes de un arreglo de cuadrados nominales colocados a manera de patrón formando un cuadrado, incluye un patrón único de búsqueda ubicado en tres esquinas del símbolo, cuyo propósito es ayudar a ubicar su localización, posición, tamaño e inclinación fácilmente.

Los códigos QR se encuentran además estandarizados y fueron aprobados como estándar ISO internacional (ISO/IEC18004) en el año 2000[8].

En [8] tenemos que los códigos QR presentan además numerosas ventajas:

- ✓ Este código tiene un patrón de localización lo que permite que puedan ser leídos en cualquier posición (360°). Al no ser la ubicación del código esencial para su correcta lectura, como sí ocurre con los códigos de barras, se reduce el margen de errores y se simplifica el trabajo.



Figura 2 Código QR

- ✓ Los códigos QR poseen una alta capacidad para restaurar información y cuatro niveles diferentes de corrección de errores que el usuario puede elegir en el momento de generarlo. Hasta un 30% de los datos pueden ser recuperables si parte del código se ha alterado o se ha perdido, como se muestra en la Tabla 1.
- ✓ Soportan cualquier tipo de lenguaje y diversos tipos de caracteres.
- ✓ Sólo precisan una décima parte del espacio que un código de barras requeriría.

Este sistema permite almacenar mayor cantidad de información, si un código de barras puede contener unos 20 dígitos, el código QR almacena hasta 7089 caracteres numéricos, 4296 caracteres alfanuméricos y 2953 bytes. Reduce además los márgenes de error y de pérdida de datos, convirtiéndolo en un sistema de trabajo más seguro.

Existen dispositivos específicos pero no son necesarios para poder descifrar un código QR pues basta con un móvil con cámara de fotos para decodificarlos, ni siquiera es preciso que el dispositivo móvil sea de última generación para que funcione y la mayor parte de los móviles actuales de los principales fabricantes permiten ya esta opción. Esta, sorprende por la facilidad y velocidad de lectura, convirtiéndolo así en una opción versátil y fácil de usar. Así pues, es posible utilizar el código QR como un elemento para que los usuarios interesados obtengan más datos sobre aquellos objetos que sean de su interés.

Hay que tener en cuenta que cada vez son más los dispositivos móviles con conexión a Internet bien a través de 3G o a través de Wi-Fi, que puede ser proporcionado por el propio sitio de interés cultural. Un código QR puede albergar una URL que redirija al usuario a la web del sitio (si la tuviese) o a algún proceso dentro del servidor. Es una forma de facilitar el acceso al visitante a una mayor cantidad de información y de darle a conocer el espacio en Internet donde podrá obtener más datos de la colección.

### **2.3.1 Características básicas de los códigos QR**

Dentro del conjunto de datos que podemos codificar en un código QR pueden ser, datos numéricos (dígitos 0-9), datos alfanuméricos (dígitos 0-9; letras mayúsculas A-Z; otros 9 caracteres: espacio, \$ % \* + - . / :), datos en bytes de 8 bits, caracteres Kanji.

La representación de los datos es binaria, un módulo oscuro es un 1, mientras que un módulo claro es un 0 binario.

El tamaño del símbolo de un código QR (sin incluir la llamada “quiet zone”) desde los 21 x 21 módulos hasta los 177 x 177 módulos, donde todos los módulos tendrán el mismo tamaño. Por lo general la “*quiet zone*” está compuesta por 4X módulos, donde X es la dimensión de uno de los lados del módulo, aunque se han hecho pruebas en las que sólo se requeriría una zona con tamaño 1X, manteniendo la correcta lectura del código QR. Esto es importante si se busca una disminución en el tamaño de los códigos.

Los caracteres por símbolo de un código QR (para un tamaño máximo de símbolo, siendo este el de la Version 40 -L), en datos numéricos se tiene la capacidad de almacenar 7089 caracteres, datos alfanuméricos: 4296 caracteres, datos en bytes de 8 bits: 2953 caracteres, datos Kanji: 1817 caracteres.

En cuanto a la corrección de errores, la cual es elegible al momento de crear el código se tienen cuatro niveles, los que permiten recuperar en el nivel L, 7%; nivel M 15 %, Q 25%;

H 30% de los datos. Los códigos QR son de tipo matricial, y la orientación de la lectura de los datos es independiente.

Una de las características más interesantes de este tipo de códigos es su capacidad para representar archivos de datos de manera lógica y continua, en hasta 16 códigos QR. Estos pueden ser escaneados sin importar la secuencia para permitir la reconstrucción correcta de los datos originales.

### 2.3.2 Almacenamiento de la información

La forma como se almacena la información en un código QR se compone de cuadrados nominales, llamados módulos; colocados en un arreglo cuadrado, debe consistir de una región de codificación y patrones que tienen funciones, entre ellos el de búsqueda, separador, patrones de tiempos y de alineamiento. Los patrones de función no se deben emplear para la codificación de los datos. La construcción general del símbolo deberá estar rodeada por la llamada “quiet zone” (ver Figura 3).

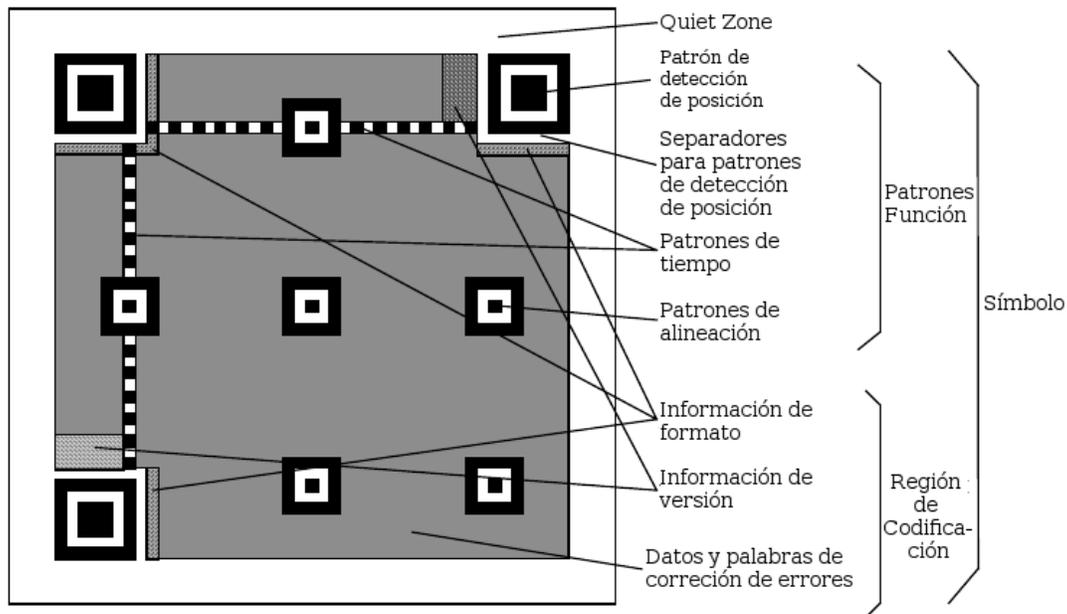


Figura 3 Descripción de las partes de quiet zone de un código QR

Existen cuarenta versiones distintas de códigos QR, nombrados *Version 1* hasta *Version 40*. La *Version 1* se compone de 21 módulos por 21 módulos, la *Version 2* mide 25 módulos x 25 módulos, cambiando así en incrementos de 4 módulos por lado hasta la *Version 40* que se compone de 177 x 177 módulos.

De igual forma y como parte del almacenamiento de la información en un código QR se describen las siguientes secciones (ver Figura 3):

✓ **Patrones de búsqueda**

Debe consistir de tres patrones de detección de posición idénticos, ubicados en las esquinas superior izquierda, superior derecha e inferior izquierda respectivamente. El símbolo se codifica de tal manera que patrones similares tengan una baja probabilidad de ser encontrados en cualquier otra parte del símbolo, habilitando así una rápida detección de un posible código QR en el campo de visión.

La ubicación de estos patrones, define la ubicación y orientación del código QR en el campo de visión.

✓ **Separadores**

Un separador de anchura de un módulo se coloca entre cada patrón de detección de posición y la región de codificación.

✓ **Patrones de tiempo**

Consiste en una fila o columna de un módulo de anchura, compuesta por módulos blancos y negros alternados, empezando y terminando con un módulo oscuro.

Estos patrones habilitan la detección de la densidad y versión del símbolo, al igual que datos de la posición para determinar las coordenadas de algún módulo.

✓ **Patrones de alineación**

El número de patrones de alineación dependen de la versión del símbolo y deberán ser colocados en cualquier símbolo correspondiente al *Model 2*, desde la *Version 2* en adelante.

✓ **Región de codificación**

Esta región debe contener los caracteres que representan los datos, sólo aquellos que representan los datos de corrección de errores, la información de la versión y de formato.

✓ ***Quiet zone***

Región de tamaño 4X módulos, la cual debe estar libre de cualquier otra marca, rodea al símbolo en los cuatro lados. Su reflectancia nominal debe ser igual a la de los módulos claros.

### **2.3.3 Recuperación de Errores**

Los códigos QR emplean la corrección de errores para generar una serie de datos de corrección que son añadidos a la secuencia codificada de la información, para permitir que se sobrepase el posible daño sin pérdida de datos. Se tienen cuatro niveles elegibles por el usuario ofreciendo la capacidad de recuperar los datos de las siguientes cantidades de errores. La Tabla 1 muestra los porcentajes de recuperación de cada uno de los niveles de los códigos QR.

TIPO DE CODIGO	PORCENTAJE DE RECUPERACION
Nivel L	Aproximadamente 7%
Nivel M	Aproximadamente 15%
Nivel Q	Aproximadamente 25%
Nivel H	Aproximadamente 30%

**Tabla 1 Tipos de código y porcentaje de recuperación**

Los datos de corrección de error pueden corregir dos tipos de datos erróneos, borraduras (palabras en posiciones conocidas) y errores (palabras erróneas en posiciones desconocidas). Una borradura es un carácter no escaneado o no legible. Un error es un símbolo carácter mal decodificado. Dado que los códigos QR son parte de la simbología matricial, un defecto en el que se convierta un módulo oscuro en uno claro o viceversa resultará en un carácter del símbolo leído como datos aparentemente válidos.

#### **2.3.4 Impresión y marcaje del símbolo**

En este apartado se revisan como primer punto las dimensiones de un código QR. Al tener como área representativa un cuadrado y tratarse de códigos matriciales, estos símbolos deben tener la misma dimensión tanto en ancho (X), como en alto (Y). Estas medidas están dadas por el tamaño de un módulo, el cual será especificado por la aplicación. Tomando en cuenta la tecnología de escaneo que se emplee.

Los códigos QR pueden almacenar miles de caracteres, una interpretación de estos datos no se considera práctica. La alternativa más usada, un texto descriptivo acompañando al código, es elegida como una mejor práctica. Como veremos más adelante, esta es la forma de uso más común en la presentación de información que incluye estos símbolos.

#### **2.3.5 Uso de códigos QR**

Desde sus primeras presentaciones, la capacidad con las que cuentan los códigos QR han sido motivo para su basta inclusión en diversos entornos en los que antes la información era presentada mayoritariamente por imágenes y texto, sin embargo, esta representación no presenta el dinamismo al que la sociedad actual se ha acostumbrado con el uso de herramientas tecnológicas.

Los usos de los códigos QR son variados, aunque están orientados más hacia el marketing, siendo distintas empresas las que los han empleado para dar un toque postmoderno.

Además de este uso, los códigos QR presentan un patrón similar de uso en distintos países, siendo los siguientes ejemplos, los más generales en el que los usuarios los crean y escanean.

- ✓ Cupones de descuento.
- ✓ Obtener más información sobre un tema.

### 2.3.5.1 códigos QR en exposiciones culturales

Este caso de uso es el que más nos interesa ya que se utilizan en cualquier tipo de exposiciones culturales, el cual ha sido bastante amplio, en el caso de museos.

Su uso es variado, puede ser guardar información sobre el objeto que se está observando, conocer obras relacionadas, interactuar con la obra y participar en actividades del tipo “búsqueda del tesoro”, como ejemplo de actividad recreativa e interactiva en conjunto con la exposición.

A continuación se mencionan casos nacionales e internacionales.

- ✓ **Museo Nacional de Artes Visuales de Uruguay:** El MINAV ha desarrollado un lector de códigos QR para ceibalitas<sup>5</sup>, instalado en su sitio web. Los códigos QR que se encuentran junto las reproducciones en los exhibidores, direccionan a nuestra web, posibilitando la navegación desde ceibalitas, teléfonos celulares, poniendo a disposición del público, materiales didácticos en línea.
- ✓ **Túnel de la ciencia:** En el transborde de la línea 5 de la estación del metro La Raza hacia la línea 3 se encuentra una exposición cultural con diversas imágenes en cuya descripción se encuentra un código QR con el que se puede acceder a más información respecto a la foto y tema expuesto.
- ✓ **London Broadgate's Art with QR Code Tour:** Broadgate Estates lanzó un tour de arte, dándole vida a algunas de las esculturas más famosas del Este de Londres con un tour permanente de códigos QR. Al escanear el código se tiene acceso a información del artista y la inspiración detrás de las esculturas en la colección pública de arte.
- ✓ **PowerHouse Museum - Gene Sherman Contemporary Japanese fashion:** Cada una de las etiquetas de los objetos mostrados está acompañada de un código QR y una URL en caso de que el código no se pueda usar. Este caso de uso en exposiciones es de 2009, por lo que se cuentan con varios problemas al llevar a cabo su implementación y es uno de los casos principales en el que el uso de un sistema como el que se está planteando permitiría una mejor gestión de la exposición así como de la información extra a la que se tiene acceso.
- ✓ **Brooklyn Museum:** 30 de los más de 3000 objetos dentro del museo han sido adicionados con un código QR que contiene datos de la obra, así como poemas que

---

<sup>5</sup> Las “ceibalitas” son laptops de uso educativo distribuidas gratuitamente por el gobierno uruguayo a todos los niños en edad escolar en el marco del Plan Ceibal

el usuario tiene acceso al escanear el código. De igual manera se tiene acceso a la versión web de la colección del museo.

- ✓ **Mattress Factory:** Museo que incorpora códigos QR en sus exposiciones, con el objetivo de reducir el gasto en papel que contempla el hacer catálogos para mostrar los contenidos.
- ✓ **Derby Musseum:** Implementación de QRpedia para extender la información de la exposición con artículos de la Wikipedia.

## 2.4 Frameworks

Conocidos como entornos de desarrollo, un entorno de desarrollo suele ser elegido como la base en el desarrollo de aplicaciones dado que estos modelan parte del dominio de la aplicación. Existe una vasta terminología que rodea este tema, y que fundamentalmente es usada para describir su estructura. En este apartado nos enfocaremos a los entornos de desarrollo orientados a objetos, dado que el sistema planteado planea ser dirigido a la abstracción de objetos presentes en la exposición del sitio de interés.

A pesar de los múltiples componentes de un entorno de desarrollo, estos se pueden generalizar a clases, tal y como sería el caso del paradigma orientado a objetos. Es de esta manera que definimos un entorno de desarrollo como una colección de clases y los algoritmos asociados a estas, de las cuales las aplicaciones desarrolladas pueden insertar código especializado propio mediante la construcción de subclases que trabajan juntas. En resumen un entorno de desarrollo consiste en clases abstractas, las operaciones que estas implementan y la expectación que recae sobre las subclases concretas.

Algunas características básicas de los entornos de desarrollo son:

- Son modulares, en parte por definición, pues están compuestos de clases abstractas con interfaces estables, encapsulando y localizando de esta manera los puntos variables del mismo. Pronto fueron reconocidos como un medio para lograr la reusabilidad a gran escala. Es aquí que cuando hablamos del dominio de una aplicación estamos haciendo un análisis de los problemas que esta puede abarcar. De esta manera es que la reusabilidad involucra tanto el análisis, diseño así como el código que se haya obtenido.
- La extensibilidad es otro término ligado fuertemente, a pesar de esto se da el caso en el que el entorno de desarrollo ya cuenta con clases que se pueden instanciar de forma directa, a este tipo de entornos de desarrollo se les conoce como *black-box framework*, aunque dentro de las actividades más frecuentes se encuentra la especialización a las necesidades de la aplicación, se les conoce como *white-box framework* a los basados primariamente en esta capacidad. Como se podría deducir, a aquellos que hacen uso de ambas propiedades tienen el término *gray-box framework*.

- Inversión de control. Este concepto es usado principalmente para diferenciar al entorno de desarrollo de por ejemplo las librerías. La inversión de control se refiere a la capacidad del entorno de desarrollo de llamar al código especializado y no como lo haría una aplicación que usa librerías al llamar al código definido dentro de las mismas.

#### **2.4.1 Herramientas de desarrollo**

Normalmente las aplicaciones no están construidas de un único entorno de desarrollo, esto da paso a los denominados *application frameworks* que se pueden definir como un entorno de desarrollo compuesto, esto con el propósito de cubrir la mayoría de los aspectos de una aplicación. Las aplicaciones o sistemas se vuelven una extensión del mismo, reúsan la arquitectura de software definida al igual que el domino del entorno de desarrollo.

Dentro de las principales ventajas de su uso están el diseño y código reusable, lo que aumenta la productividad en el desarrollo y reduce los tiempos de lanzamiento de un nuevo producto. Las aplicaciones tienden a contener menos errores (*bugs*) y pueden tener un aspecto más homogéneo. Podríamos mencionar que la interoperabilidad entre aplicaciones de una misma suite se vería considerablemente aumentada.

#### **2.4.2 El problema de los frameworks**

Uno de los retos clave al adoptar tecnologías de entorno de desarrollo es la curva de aprendizaje inclinada que los desarrolladores deben enfrentar antes trabajar productivamente. En términos de recursos, toma tiempo y buenos desarrolladores hasta que un proyecto o una organización puedan emplear un entorno de desarrollo de manera exitosa. Los entornos de desarrollo tienen curvas de aprendizaje empinadas ya que los desarrolladores no solo tienen que comprender clases aisladas, si no que deben tratar con diseños abstractos de distintas clases cuyas instancias colaboran en muchos fines. Es común que esta curva tenga un periodo de 6 a 12 meses de aprendizaje.

### **2.5 Cómputo móvil**

El cómputo móvil se puede definir como un ambiente de cómputo con movilidad física. Las posibilidades del usuario en un ambiente de este tipo se incrementan, su capacidad de acceso a la información, así como la realización de tareas pasan a ser independientes del sitio en el que se encuentre, este solo se ve restringido al dispositivo que emplea y las capacidades que este tenga. Por ejemplo, mientras el usuario se encuentra desplazándose de un sitio a otro, el dispositivo a emplear sería un *smartphone*, tablet o incluso una laptop, por otro lado si este se encuentra en casa o en la oficina, el dispositivo usado sería un computador de escritorio o alguna estación de trabajo. Cabe destacar que el cómputo móvil, no necesariamente hace referencia a la ejecución de aplicaciones en dispositivos móviles, que se puedan catalogar como portables o que cumplan ciertas características. El cómputo móvil está bastante ligado con el concepto de computación ubicua, en la que la tecnología que la hace posible, pasa desapercibida para el usuario.

En otras palabras el concepto de cómputo móvil abarca la facilidad de que el usuario extienda su información y conocimiento, así como la realización de tareas independientemente de su ubicación y por el momento, solo restringiéndose al acceso a la tecnología que posea.

Lo mencionado anteriormente es uno de los objetivos del sistema a desarrollar, siendo esta la capacidad de que el visitante de una exposición cultural o sitio de interés, se lleve parte del conocimiento, para que de esta manera, la experiencia de la exposición no sólo se quede en ella.

Algunas definiciones alternativas de cómputo móvil son:

1. **Información en cualquier lugar y momento:** Esta es la definición genérica de ubicuidad, donde la información está disponible en cualquier lugar todo el tiempo.
2. **Computación nómada:** El ambiente de cómputo es nómada y se mueve junto al usuario. Esto aplica tanto para servicios locales como remotos.
3. **Computación ubicua:** Ambos términos hacen referencia a la computación ubicua, en la que la computación se da por sentada, y su presencia no es notoria. Podría compararse a la electricidad, de la cual sólo notamos su ausencia.

### **2.5.2 Dispositivos de cómputo móvil**

Los dispositivos de cómputo móvil parecen tener una definición temporal, pues conforme las tecnologías avanzan, las características de estos se orientan hacia diversos aspectos presentes en la época. En términos de tiempos más recientes y en el que las computadoras se han integrado ya bastante a la vida cotidiana, un dispositivo móvil se puede considerar como un computador lo suficientemente pequeño como para sostenerlo en la mano. Sin embargo, con esta definición estaríamos omitiendo muchas de las características que actualmente asignaríamos a un dispositivo considerado como móvil. Dentro de los aspectos más distintivos de un sistema de cómputo móvil están, su conexión prevalente e inalámbrica a la red, su tamaño reducido, su naturaleza móvil de uso, las fuentes de poder que emplean y sus funcionalidades que están adaptadas especialmente a los usuarios móviles. Debido a estas características las aplicaciones de cómputo móvil son inherentemente diferentes a las aplicaciones escritas para uso en sistemas de cómputo estacionarios, aunque esto último está tratando de complementarse mediante el ajuste a sistemas operativos de escritorio para poder ser ejecutados en arquitecturas móviles.

Estos dispositivos pueden caer igualmente dentro de cuatro paradigmas que los describen: están fuertemente descentralizados, diversificados, conectados y son de uso simple. De igual manera, cumplen con ciertas características que los engloban.

### **2.5.2.1 Cámaras móviles**

La integración de cámaras en los dispositivos móviles como es el caso de los *smartphones* ha estado presente desde que se acuñó el término, hoy en día incluso el mercado de las cámaras han perdido terreno debido a su inclusión en estos dispositivos.

Las cámaras han incrementado las posibilidades de estos dispositivos, principalmente en el software que se es capaz de desarrollar, ya que las aplicaciones que son desarrolladas solo ocupan la cámara para tener acceso a una gran variedad de funciones, dentro de ellas la lectura de códigos QR, ya sea mediante la captura de una imagen o la decodificación en tiempo real.

Dentro de las características de las cámaras que incluyen los teléfonos se encuentran el auto balance de blancos, auto enfoque, y la exposición automática para mejorar la calidad de la imagen. La resolución de las cámaras va desde los 0,3 hasta los 12 megapíxeles en los teléfonos de última generación.

Estas prestaciones sumadas a las cualidades de los símbolos QR permiten que su lectura se realice con mayor facilidad aún bajo condiciones ambientales no tan favorables.

### **2.5.3 Características**

Entrando en más detalle, las características principales de los dispositivos móviles podrían generalizarse a las siguientes:

- Tamaño reducido.
- Conectividad parcial o total a un medio de información.
- Fuente de alimentación limitada
- Capacidad de cómputo (reducida).
- Cuentan con software específicamente diseñado para los bajos recursos de los que disponen.
- Los métodos de interacción con el usuario son variados.
- Capacidad de almacenamiento limitada.
- Llegan a poseer sensores para interactuar con el ambiente y el usuario.

Más allá de las características anteriores ya se hablaría de dispositivos modernos que son usados por la sociedad hoy en día, como es el caso de los teléfonos móviles (refiriéndose a *smartphones* principalmente) y tablets. Estos cumplen la mayoría de las características mencionadas, aunque como veremos más adelante una de las capacidades que los ha hecho tan populares, es la inclusión de componentes como lo es la cámara que integran.

### **2.5.4 Conectividad**

Existen diversos métodos por los que un dispositivo móvil, centrándonos en el área de los *smartphones* y teléfonos móviles, tiene acceso a una red; ya sea la red telefónica o la Internet. En México las compañías de telefonía celular proporcionan tecnologías de conectividad 3.5G, la cual provee una arquitectura abierta para servicios basados en multimedia, y aplicaciones de tecnologías como smart antenna, radio definido por software y TD-CMA, la estandarización ha cambiado su enfoque hacia la red. Al usar esta tecnología se facilitará el cambio hacia su sucesora, la 4G o LTE, la cual tiene un incremento en la velocidad de transmisión de datos.

Como otra opción en la conectividad y aún más usada que los planes de datos proporcionados por las compañías se encuentra el Wi-Fi, que se ha popularizado al aumentar los sitios que ofrecen gratuitamente una conexión.

Con la facilidad de acceso que se tiene a la contratación de un plan de datos, que abarca la primera parte descrita, los teléfonos móviles aumentan su tiempo de conexión a la red a un estado casi permanente.

## **2.6 Estado del arte**

Una vez repasado un marco teórico general de los conceptos a usar en el desarrollo del sistema, pasamos a hacer una relación entre las definiciones mencionadas en los contextos museológicos así como tecnológicos presentados.

### **2.6.1 Herramientas museológicas digitales**

Las herramientas museológicas digitales surgen de la relación que se genera entre lo museológico y lo digital, en el momento en el que se trata de resolver un problema propio de un museo.

Este término puede abarcar herramientas digitales que se emplean como recursos museográficos, tales como páginas web, multimedios interactivos, kioskos interactivos. Sin embargo las herramientas más interesantes son aquellas que se encargan de la gestión de las colecciones de los museos. En México se han implementado una gran cantidad de este tipo de software, aunque no se ha logrado que tengan una aceptación alta ni un estándar que replique su uso a los museos presentes en el país. Dentro de las más destacadas tenemos los siguientes ejemplos:

- Sistema Interactivo para el Programa de Catálogo de Bienes del Patrimonio Cultural de Propiedad Federal (Mac Masters, 1993) (SEDESOL).
- Sistema General de Registro de Obra Artística (SIGROA): la información está digitalizada, no está estandarizada ni normada, lo que significa que no es común para todos los museos del INBA, ni tampoco se puede acceder a ella en línea o en red.
- Museo Nacional del Virreinato (MNV): Sistema de Información y Gestión (SIM): era un sistema complejo y su operación requería mucho tiempo, lo cual lo convertía

en poco amigable. No generaba la idea de ser un sistema que administrara de mejor manera las funciones.

- Memoria Museológica Mexicana (Witker, 2006): Prototipo en desarrollo.
- Consultarte, un programa que permitía asentar, documentar y consultar más de 1 000 registros documentales del acervo del Museo Arocena, en Torreón.
- Teseo, Digital Art Project, propuesto por investigadoras curadoras de la Benemérita Universidad Autónoma de Puebla.
- Programa integral de información (MNC, Museo Nacional de las Culturas).

Aunque nuestro sistema no plantea un alcance similar al de estas herramientas, es de destacar su existencia como un registro de la inversión e intento de estandarización de la información que se maneja en los museos de nuestro país. Aclaramos de nuevo que el sistema no se orientará únicamente a museos, sin embargo, se trata de la institución que ha originado la mayor cantidad de técnicas para muestras culturales.

El sistema que se está planteando entra dentro de la categoría de herramientas digitales que se emplean como recursos museográficos, ya que lo que interesa es mejorar el discurso museográfico en el aspecto interactivo dentro de la exposición objetivo.

### **2.6.2 La interactividad en los museos**

Los sitios culturales en los que ha evolucionado más el concepto de interactividad y sus aplicaciones, son sin duda, los museos. En México contamos con museos interactivos en varios estados y principalmente en el Distrito Federal. En un principio se ofrecía contenido que el usuario pudiera comprar y llevarse como dato de la experiencia del museo, principalmente CD-ROMs.

De entre los primeros museos interactivos y el cual marcó ciertos procedimientos que se extenderían a la apertura de más museos tenemos al Universum, de la Universidad Nacional Autónoma de México. Para él se produjeron programas de cómputo y equipamiento que usaba los multimedios para que el visitante pudiera interactuar con el discurso museográfico de la exposición. Sus conceptos se expandieron a museos como el museo de la luz (1996), el Centro de Ciencias, de Sinaloa y el Museo de Ciencia y Tecnología, de Xalapa (1992), el Papalote Museo del Niño, en el Distrito Federal (1993), MIDE (1998) y el Museo Arocena, Torreón, Coahuila son algunos ejemplos.

#### **2.6.2.1 Los dispositivos móviles como recursos museográficos**

La presencia de dispositivos móviles en las exposiciones no es tan elevada. Los museos, por ejemplo, no ven la necesidad de invertir en este tipo de tecnología sólo para el uso interno en el discurso museográfico. No ha sido necesaria la inversión por parte del museo, ya que la tecnología y la reducción de su costo, están dando las condiciones para que las personas tengan acceso a dispositivos móviles de bajo costo pero con presentaciones medianamente altas.

El nuevo escenario, atiende a que la evolución tecnológica se dirige a la actualización de los contenidos desde la propia web y a la consideración de los dispositivos móviles como herramientas con las que interactuar en el museo[9], o donde se presente la oportunidad.

# CAPÍTULO 3 ANÁLISIS

## 3.1 Metodología

### 3.1.1 Metodología en V

La metodología que será usada es la llamada Metodología-V[10], que es una representación gráfica del ciclo de vida del desarrollo de sistemas (ver Figura 4). Esta metodología resume las principales medidas que deben adoptarse en relación con las prestaciones correspondientes en el marco del sistema informático de validación.

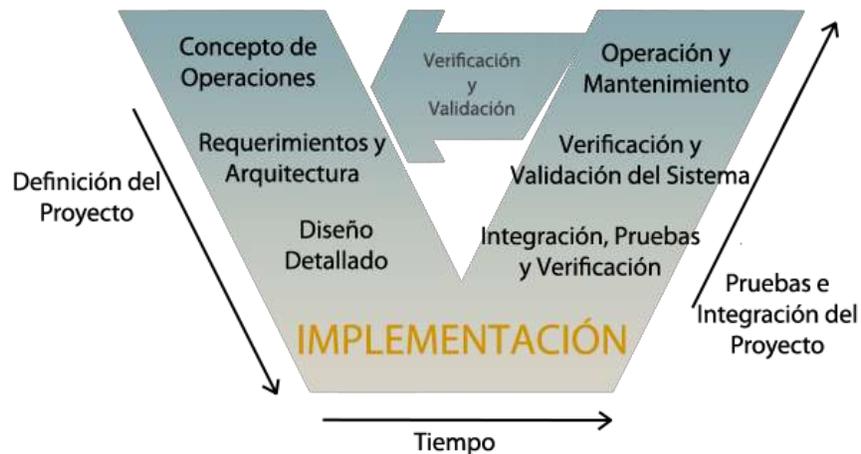


Figura 4 Descripción de la metodología en V

Es un conjunto de actividades que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Se describen las actividades y resultados que deben producirse durante el desarrollo del producto. El lado izquierdo de la V representa la descomposición de las necesidades y la creación de las especificaciones del sistema. El lado derecho de la V representa la integración de las piezas y su verificación. V significa «Verificación y Validación». A grandes rasgos es una metodología basada en el modelo en cascada pero iterando en cada etapa hasta que el desarrollo de cada una esté correcto.

En la propuesta solo nos enfocaremos a la parte izquierda de la V que es el diseño y desarrollo de la arquitectura, sin embargo intentaremos aplicarlo a un caso de estudio real con lo que llegaríamos a la segunda etapa del lado derecho de la V (Verificación y Validación del Sistema).

Esta metodología nos presenta grandes ventajas como son:

- ✓ Minimización de los riesgos del proyecto.
- ✓ Mejora y garantía de calidad.
- ✓ Reducción de los gastos totales como son tiempo y esfuerzo principalmente.
- ✓ Permite el desarrollo de proyectos con poco personal.

## 3.2 Códigos QR

La generación y la decodificación de los códigos QR son aspectos de importancia para el sistema. Cada una de estas actividades se realiza en partes separadas del sistema (más detalle sobre esto en la arquitectura del sistema), cada una con recursos y capacidades distintas. En esta sección se describen los algoritmos que definen la codificación y decodificación de los datos en formato QR.

### 3.2.1 Generación de códigos QR

La generación de los símbolos tiene un procedimiento detallada dentro de la especificación ISO/IEC 18004:2006. A continuación se describe en forma general el procedimiento:

- **Análisis de los datos:** Se analiza el flujo de datos de entrada identificando los diferentes caracteres a ser codificados. Se debe cambiar de entre los distintos modos de codificación para obtener la codificación más eficiente de los datos en una cadena binaria. Se selecciona el nivel de corrección de errores, si el usuario no especifica la versión a usar del símbolo, la versión más pequeña en la que quepan los datos será la elegida.
- **Codificación de datos:** Convertir los caracteres de datos en una cadena de bits que será separado por los distintos modos de codificación elegidos. La cadena de bits resultante se divide en palabras de 8 bits, se añaden caracteres de relleno con forme sea necesario para completar el número de palabras de la versión elegida.

La codificación empleada en la plataforma será de tipo alfanumérico, ya que sólo interesa el almacenamiento de direcciones a los recursos en el servidor.

Dentro de este apartado y antes de colocar los datos correspondientes a la cadena de caracteres debemos codificar el indicador de modo, en este caso el modo alfanumérico, por lo que los primeros bits de la cadena son 0010. Los identificadores para otros tipos de codificación están definidos en la especificación.

Para codificar los datos alfanuméricos se separa la cadena en pares de caracteres. Para cada par de caracteres se toma el valor ASCII del primer caracter multiplicándolo por 45 y sumando el valor ASCII del segundo. El resultado será convertido a una cadena de 11 bits. Si el número de caracteres es impar, el último caracter deberá ser convertido a una cadena de 6 bits de longitud.

Dependiendo de la versión del código que se elija, el número  $n$  de bits capaces de almacenarse y requeridos para generar el código es específico. Si el  $n$  generado es menor al definido, se completa la cadena con cuatro ceros, siempre y cuando no se sobre pase  $n$ , de ser así se agrega el número de ceros necesarios.

Posteriormente la cadena de bits se separa en palabras de 8 bits. Si alguna de las secciones no tiene el esta cantidad, será completada con ceros hasta completar la longitud de la palabra.

Si la cadena resultante de bits no llega a completar el valor necesario de bits ( $n$ ) especificado por la versión del código QR, esta se completará con un par de palabras definidas dentro de la especificación del símbolo: 11101100 y 00010001. Estas se agregarán de manera alternada comenzando con 11101100 y pasando 00010001, así hasta tener el número de bits necesarios ( $n$ ).

➤ **Codificación de la corrección de errores:** Se divide la secuencia de palabras en el número requerido de bloques para habilitar el procesamiento del algoritmo de corrección de errores. Se genera la palabra de corrección de errores para cada bloque, añadiéndola al final de la secuencia de palabras de corrección de errores.

Los códigos QR emplean el algoritmo *Reed-Solomon* para la corrección de errores, además de un número de bloques de corrección requeridos para este proceso. Dicho número está definido en relación a la versión del código QR empleado y el nivel que se elija. Por ejemplo, la *Version 1* con un nivel de corrección Q requiere de 13 bloques de datos y 13 bloques de corrección de palabras para la corrección errores.

El algoritmo *Reed-Solomon* emplea un polinomio generador, que se deriva de un concepto llamado campo finito, también conocido como campo de *Galois*. Los códigos QR emplean un campo de *Galois* de 256 elementos. El polinomio generador estará sujeto a una división polinomial, la cual emplea un cambio entre la notación  $\alpha$  (alfa) y enteros. Las alfas y los enteros se obtienen a partir de las tablas de logaritmos y anti logaritmos generadas a partir de operaciones aritméticas en un campo de *Galois* elevado a la segunda potencia.

El proceso de generación del código de corrección de errores (ECC) itera entre la conversión de notación alfa y enteros, la multiplicación del primer coeficiente de los polinomios generados con el polinomio generador y una operación XOR con los coeficientes de los polinomios.

Esto se repite hasta que el término final del polinomio tenga  $x^0$ . Los coeficientes resultantes en el polinomio serán las palabras para el ECC.

➤ **Estructurar el mensaje final:** Intercalar los datos y las palabras de corrección de errores añadiendo bits como sea necesario. Aquí se colocarán los datos generados en el paso de codificación de datos seguidos de las palabras generadas en el paso anterior.

➤ **Colocación de los módulos en la matriz:** Colocar la palabra correspondiente a cada módulo en la matriz, junto con los patrones de búsqueda, separadores, patrones de tiempos y patrones de alineación.

➤ **Enmascarado:** Aplicar los patrones de enmascarado correspondientes a la región de codificación del símbolo. Evaluar los resultados y seleccionar el patrón que optimice el balance de módulos y minimice la ocurrencia de patrones indeseables.

En esta parte del proceso se lleva a cabo la creación de los módulos dentro de la sección de almacenamiento de los datos, es decir el dibujo de la imagen para los módulos ya sean negros o blancos, indicando el bit que está en esa posición con base a la máscara elegida de entre las ocho disponibles por la especificación del código QR.

El orden de colocación de los bits comienza en la parte inferior derecha de la sección de datos, comenzando el recorrido a la izquierda de este módulo, posteriormente arriba a la derecha de este último. Este proceso se repite para toda la columna, siendo llenada hacia arriba. Cuando se ha completado se pasa a la columna inmediata izquierda. Se llena el primer módulo, pasando al que se encuentra a la izquierda de este y por último al ubicado abajo a la derecha de este. Al llenar esta columna se continúa con la inmediata izquierda a esta. Este proceso continua hasta llenar toda la sección de datos. Los últimos 7 módulos se rellenan con ceros. En las siguientes figuras se muestra el recorrido para colocar los bits.

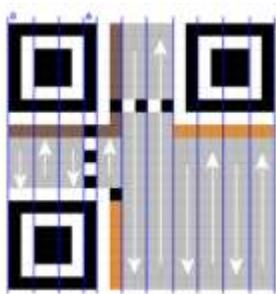


Figura 5 Dirección de colocación de los módulos con base a las filas que forman el código

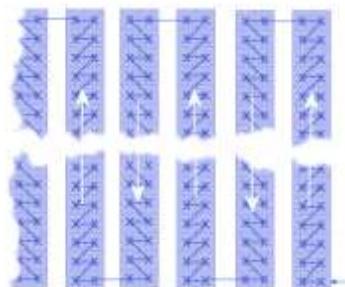


Figura 6 Colocación de los bits dentro de las filas

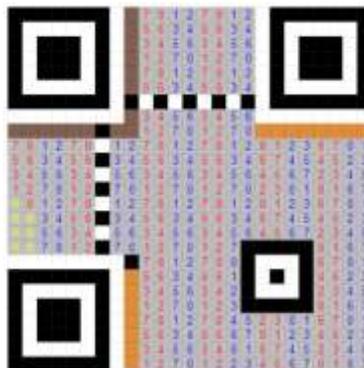


Figura 7 Colocación final de los bits dentro de la región de datos

- **Información de la versión y el formato:** Generar la información del formato y la versión y completar el símbolo.

### 3.2.1 Decodificación de códigos QR

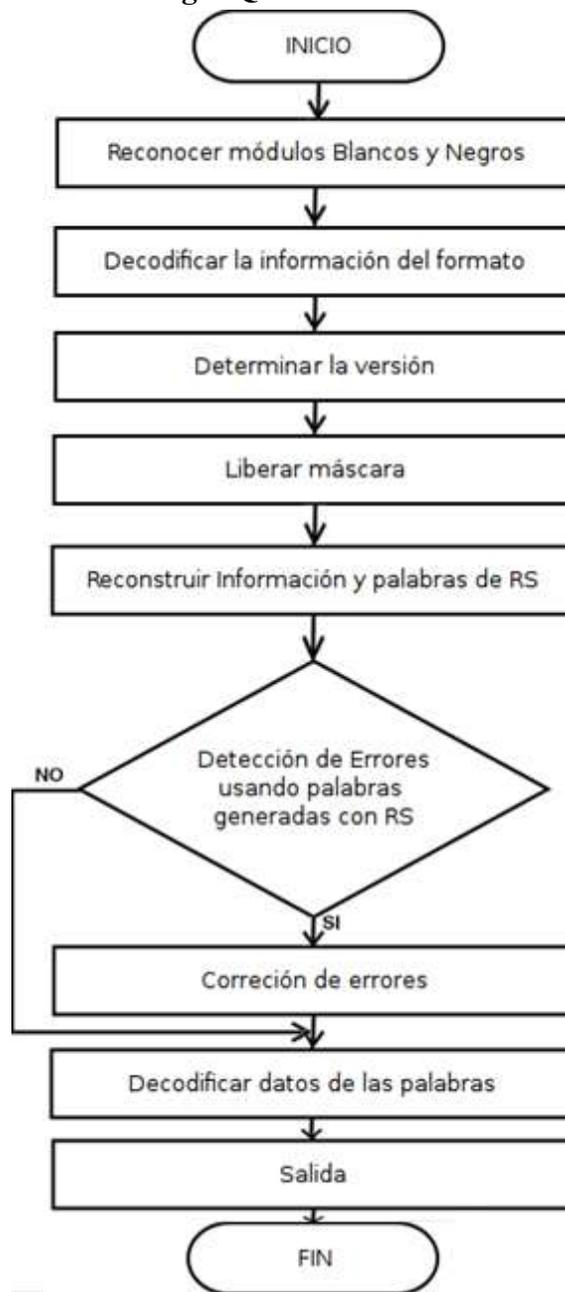


Figura 8 Algoritmo de referencia para la lectura de códigos QR

El siguiente algoritmo de decodificación encuentra el símbolo en una imagen y la decodifica. El algoritmo de decodificación hace referencia a los estados claros y oscuros en la imagen.

1. Determinar un umbral global, tomando una media de entre el máximo y el mínimo del valor de la reflectancia de la imagen. Con esto se convierte la imagen en un conjunto de píxeles blancos y negros usando el umbral global.
2. Ubicar el patrón de búsqueda. El patrón de búsqueda en un código QR consiste en

tres patrones de detección de posición ubicados en tres de las cuatro esquinas del símbolo.

- a) Cuando un área candidata es detectada se resalta la posición del primero y último de los puntos A y B respectivamente en los cuales la línea de píxeles en la imagen encuentra los bordes exteriores del patrón de detección. Se repite esto para las líneas de píxeles adyacentes en la imagen hasta que todas las líneas que cruzan el cuadrado central del patrón de detección en el eje x de la imagen sean identificadas.

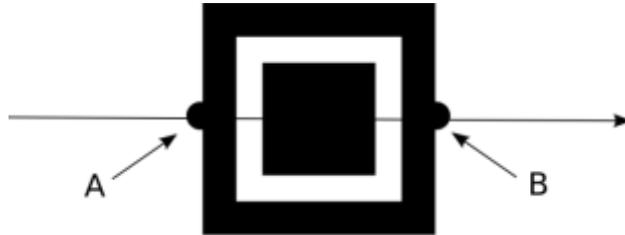


Figura 9 Línea de escaneo en el patrón de detección de posición

- b) Repetir el paso a) para las columnas de píxeles que cruzan el cuadrado central del patrón de detección en el eje y.
- c) Localizar el centro del patrón. Construir una línea a través del punto medio entre el punto A y B en los píxeles de fuera que atraviesan el cuadrado central del patrón de detección de posición en el eje x. Construir una línea similar que cruce los puntos A y B en los píxeles de fuera del cuadrado central en el eje y.
- d) Repetir los pasos a) a c) para localizar el centro de los otros dos patrones de localización.
- Determinar la orientación del símbolo mediante el análisis de las coordenadas de los centros de los patrones de posición, identificando cual es el patrón ubicado en la esquina superior izquierda en el símbolo y el ángulo de rotación del símbolo.
  - Determinar la distancia D que cruza la longitud total del símbolo entre los centros de los patrones de localización superior derecho y superior izquierdo, así como la longitud de ambos patrones  $W_{UL}$  y  $W_{UR}$ .
  - Calcular la dimensión nominal X del símbolo.
 
$$X = (W_{UL} + W_{UR}) / 14$$
  - Determinar provisionalmente la versión V del símbolo.
 
$$V = [(D / X) - 10] / 4$$
  - Si la versión provisional del símbolo es 6 o una menor, esta se especifica como la versión definida. Si la versión provisional es 7 o mayor, la decodificación se realiza de acuerdo a las reglas definidas en la especificación de los códigos QR en el apéndice E.7.
  - La decodificación de los símbolos con versión se realiza de acuerdo con los pasos 7 y 8 del algoritmo en el anexo M.14 de la especificación. La decodificación de los símbolos de versión 2 o mayor, requieren de las coordenadas centrales de cada uno de los patrones de alineación, determinados por las coordenadas definidas en 7.3.5 y

- el Anexo E para determinar la rejilla de muestreo.
9. Muestrear el pixel de la imagen en cada una de las intersecciones de las líneas de la rejilla y determinar si es claro u oscuro en base al umbral global. Construir una matriz de bits mapeando los pixeles oscuros a 1 y los pixeles claros a 0.
  10. Decodificar la información de formato adyacente al patrón de detección de posición ubicado arriba a la izquierda para obtener el nivel de corrección de errores y el patrón de enmascaramiento aplicado al símbolo. Si hay errores que excedan el nivel de corrección de errores, entonces aplicar el mismo procedimiento para decodificar la información de formato adyacente al patrón de detección de posición ubicado en la parte superior derecha.
  11. Realizar la operación XOR del patrón de enmascaramiento con la región codificada del símbolo, para restaurar los datos enmascarados y reconstruir los caracteres codificados que representan los datos y las palabras de corrección de errores. Esto revierte el efecto del proceso de enmascaramiento aplicado durante el proceso de codificación.
  12. Determinar las palabras que conforman al símbolo de acuerdo con las reglas de colocación definidas en la sección 8.7.3 de la especificación.
  13. Reubicar la secuencia de las palabras en bloques como se requiere para la versión y nivel de corrección de errores del símbolo.
  14. Seguir el procedimiento de detección y corrección de errores, definido dentro de la especificación, Anexo B para corregir errores de borrado hasta el máximo de la capacidad definida por la versión y nivel de corrección de errores del símbolo.
  15. Reconstruir la cadena original de bits mediante el ensamble de los bloques en secuencia.
  16. Sub-dividir los datos de la cadena de bits en segmentos cada uno comenzando con un indicador de modo y la longitud, los cuales son determinados por el indicador de conteo de caracteres seguido del indicador de modo.
  17. Decodificar cada segmento de acuerdo a las reglas para el modo aplicado.

### 3.3 Plataformas móviles

Las plataformas móviles se han masificado en dispositivos que han evolucionado a lo que hoy conoceríamos como *smartphone* y tablets. Todos estos sistemas tienen una tendencia que se ajusta al hardware que viene siendo incorporado recientemente a los dispositivos, con esto nos referimos a pantallas táctiles y sensores de posicionamiento y movimiento. Es por esto que los sistemas van orientando sus interfaces a paradigmas orientados a eventos, en los que son necesarias acciones del usuario. Orientándonos a la arquitectura de estos sistemas, tenemos que las plataformas más populares basan su software de bajo nivel en sistemas que originalmente fueron pensados para servidores o escritorio, esto en parte por las prestaciones que estos sistemas más desarrollados tienen así como el diseño que los originó, el cual aprovecha los recursos ajustándose a la escasez de los mismos.

Como se verá más adelante, existen sistemas que evolucionaron de versiones bastante viejas, que exclusivamente fueron desarrollados para el hardware de un dispositivo móvil y

a pesar de tener ya un largo tiempo a la venta, estos tienen gran parte del mercado, bien podríamos atribuir este caso al bajo costo de los equipos; por tanto conforme se reduzca el precio del hardware de los equipos de gama alta, las prestaciones de los sistemas más populares se verán aun mayormente masificadas.

### 3.3.1 Estadísticas de uso

Para comenzar a delimitar las plataformas presentes en los siguientes temas, se presentan estadísticas de uso de teléfonos móviles, así como de las plataformas más usadas en ellos.

En México existen cerca de 94,565,305[11] suscripciones a teléfonos móviles, donde cerca el 15.5% corresponde a teléfonos inteligentes un total de 14.7 millones. La causa principal del crecimiento de ventas de *smartphones* se debe al continuo descenso de los precios de la tecnología y por una mayor disponibilidad de equipos avanzados, la expansión en la oferta de planes de datos y el atractivo social de contar con un dispositivo de alta tecnología[12]. La mayoría de los usuarios que acceden a Internet mediante estos dispositivos, lo realizan con un acceso a redes Wi-Fi, mientras que sólo el 51% tiene un plan de datos. Se prevé que para 2015 siete de cada diez personas cuente con un teléfono inteligente.

De este porcentaje de *smartphones*, las plataformas más populares en nuestro país parecen diferir en diversos estudios. Por un lado tenemos a BlackBerry con un 39%; Apple alcanza un 21%, Samsung tiene un promedio de ventas del 12% de *smartphones*; y Nokia un 22%.

Las estadísticas provistas por el sitio *StatCounter* muestran un resultado un tanto diferente, aunque estas dependen de los sitios que tengan habilitado su plugin. La distribución de los sistemas operativos es de acuerdo a este servicio se observan en la Figura 10.

A su vez los mayores sistemas operativos, refiriéndonos a Android y iOS, se encuentran fragmentados en las distintas versiones que estas presentan. En el mercado de Android se encuentran presentes versiones como la 1.5 hasta la más reciente 4.1 funcionando. En la Figura 11 se muestra la distribución actual de este sistema operativo.

Por otro lado, los dispositivos con iOS van desde la primera versión hasta la más reciente iOS 6 funcionando. Si tuviésemos que elegir una plataforma en específico para el desarrollo de aplicaciones para este mercado, podemos seguir el patrón en el que se ha comportado la adopción de cada una de las nuevas versiones. Como ejemplo de la plataforma 4, esta se encontraba adoptada por más del 90% de los dispositivos tan sólo a 6 meses de su lanzamiento. Su tasa de adopción fue el siguiente:

- 1 mes tras el lanzamiento - 50% en tasa de adopción
- 3 mes tras el lanzamiento - 70% en tasa de adopción
- 6 mes tras el lanzamiento - 90% en tasa de adopción

Con esto se estima que para finales de Diciembre de 2012 cerca del 60% de los dispositivos que ejecutan iOS tendrán instalada la versión 5.

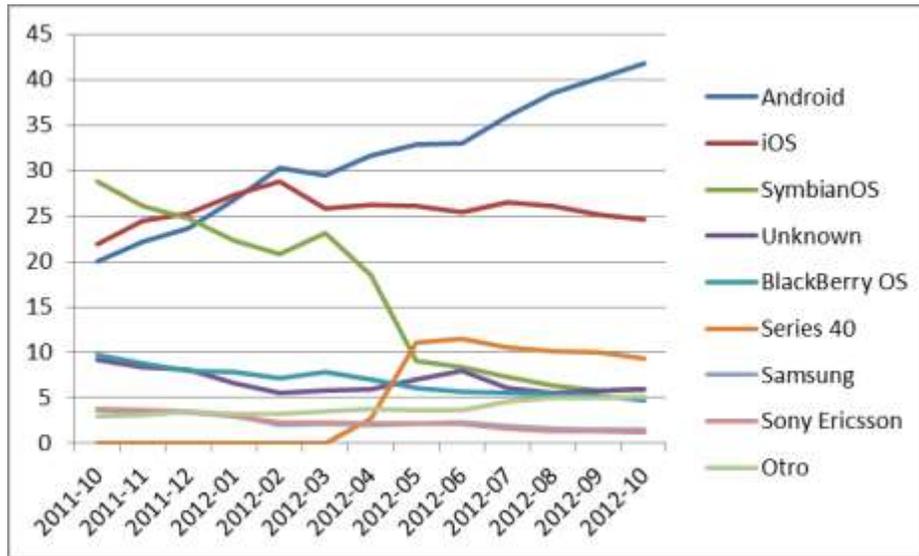


Figura 10 Sistemas Operativos más usados en México. Sep 2011 a Sep 2012

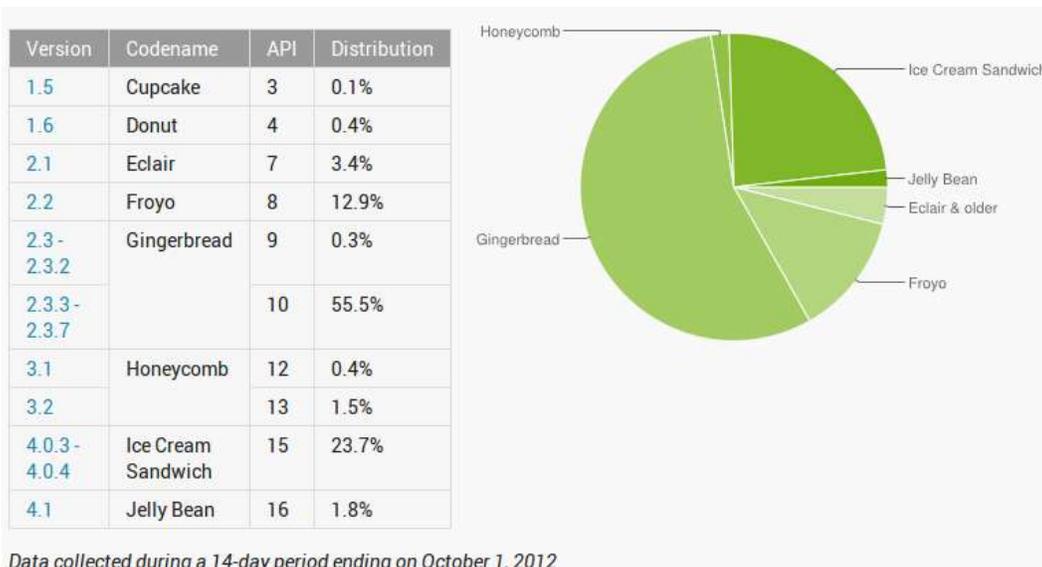


Figura 11 Distribución actual de Android. Basada en el número de dispositivos que han accedido a Google Play en un periodo de 14 días.

### 3.3.2 Arquitectura de las plataformas móviles

En este apartado describimos las arquitecturas respectivas de los dos sistemas operativos más usados, en los que coinciden las encuestas, además de ser los elegidos como sistemas objetivos del sistema que se plantea: Android y iOS.

### 3.3.2.1 Arquitectura de Android

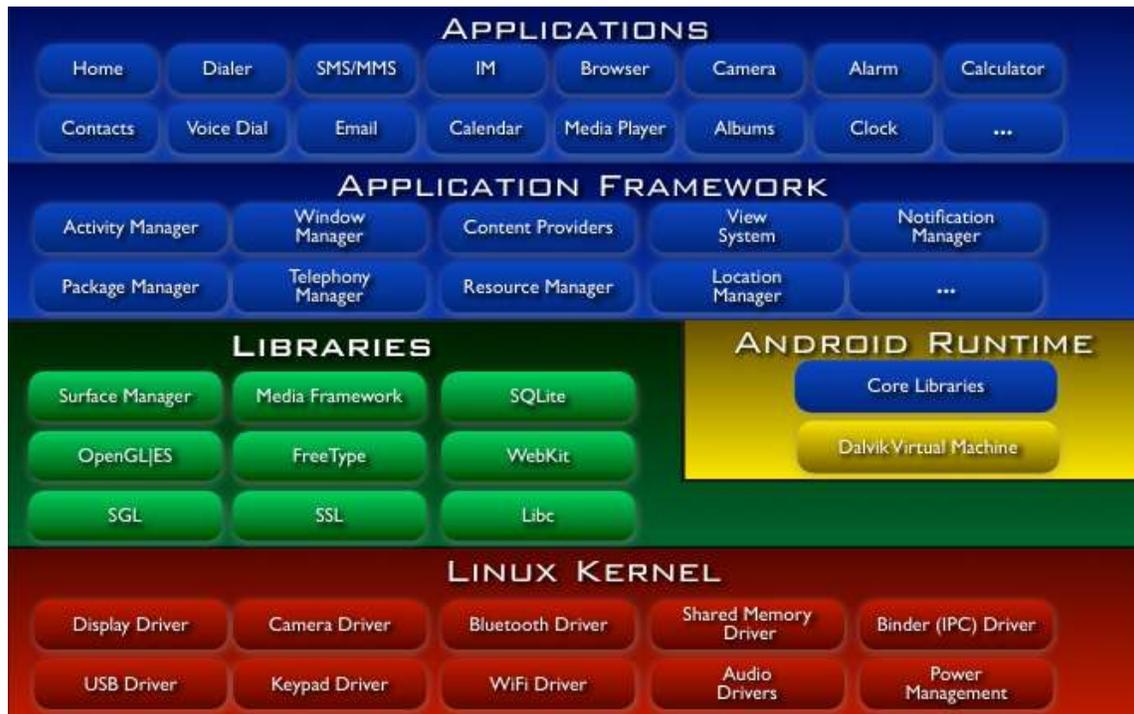


Figura 12 Arquitectura de Android

La arquitectura de este sistema operativo (mostrada en la Figura 12) está categorizada por bloques, en este apartado se describen de forma general estos componentes y las funciones para las que fueron pensadas. [13]

- **Kernel de Linux:** Android se construye a partir del kernel de Linux, sin embargo, Android no es Linux, no posee un sistema de ventanas nativo, como es el caso de la mayoría de las distribuciones que incorporan el servidor X11 como el estándar y sobre el que se han construido la mayoría de los gestores de ventanas y entornos de escritorio. Android tampoco incorpora soporte para glibc. Tampoco se incluye la mayoría de las utilidades estándar de Linux. Dentro de los motivos de la elección de este kernel están la buena gestión de memoria y procesos, el modelo de drivers con el que cuenta y el que se trata de software libre, por lo que se puede ajustar a las necesidades que se tengan. Una de las mejoras a destacar y que han sido incluidas a la rama oficial del kernel está la gestión de energía (Power Management) con los denominados *wake locks*.
- **Librerías:** Las librerías incorporadas por Android son:
  - *Bionic Libc:* Una implementación personalizada de libc, optimizada para uso embebido.
  - Librerías de función: Librerías nativas, estas proveen de ciertas características

funcionales al sistema, dentro de estas tenemos a webkit (motor de navegación web), un Media Framework, que incluye soporte audiovisual así como plugins para codecs de hardware y software. SQLite, que es un gestor de bases de datos ligero, es usado como el backend para el almacenamiento de datos en la plataforma.

- Capa de Abstracción de Hardware (HAL, Figura 13): Define la interfaz que requieren implementar los drivers del hardware que ejecuta Android. Se separa la lógica de la plataforma de la interfaz de hardware.



Figura 13 Capa de Abstracción de Hardware

- **Android Runtime:** La ejecución de las aplicaciones en Android fue pensada con el modelo de la máquina virtual. Para esta plataforma fue diseñada la máquina virtual *Dalvik*, la cual provee portabilidad de las aplicaciones y consistencia en el tiempo de ejecución. Está diseñada para entornos embebidos, cada proceso se ejecuta en su propia instancia de la máquina virtual[14]. Posee intérpretes de código altamente optimizados y emplea la memoria de tiempo de ejecución de manera efectiva. En este apartado también están las librerías del núcleo, es aquí donde tenemos las APIs del lenguaje Java, que proveen una plataforma de desarrollo familiar.
- **Framework de aplicaciones:** Se trata de los servicios básicos de la plataforma, las aplicaciones no acceden a ellos directamente, proveen una manera de acceso a las APIs de bajo nivel implementadas en hardware. Aquí tenemos a los servicios de localización, acceso a sensores o el servicio de telefonía.
- **Aplicaciones:** Bloque final de la arquitectura, es el área de la interfaz el usuario y con la que se producen eventos al interactuar con ella. Aplicaciones SMS, teléfono, cámara, alarma, etc.

Como vemos la arquitectura de Android es abierta y está bien definida, ha probado ser escalable a nivel de hardware y software pues se basa en fundamentos sólidos (kernel y librerías de código abierto) que permiten mejora continua.

### 3.3.2.2 Arquitectura de iOS



Figura 14 Arquitectura de iOS

La arquitectura de iOS es más simple a la vista, la realidad es que no se tiene mucha noción de su estructura interna, ya que a diferencia de Android, este no es un sistema abierto. Si bien está basado en Mac OS X, del cual el kernel (Darwin) surgió de la comunidad de desarrolladores, ha tenido muchas modificaciones privativas por parte de Apple. A los detalles que más acceso se tiene es a la sección del núcleo del sistema, por otra parte el bloque de Cocoa Touch es privativo y no se tienen muchos detalles de su funcionamiento interno. Como se vio con la arquitectura de Android, se detallarán los bloques que componen a iOS.

- **Núcleo del sistema:** iOS cuenta con hilos (cumplen los estándares POSIX[15]), su sistema de red está basado en Sockets de tipo BSD. Este bloque tiene acceso al sistema de archivos, entrada y salida estándar, servicios de red, servicios de ubicación y localización. Aquí se realizan todas las operaciones de asignación de memoria y los cálculos del SO.
- **Servicios del núcleo:** Concentra los servicios a los que tiene acceso el dispositivo, como son acceso a red de bajo nivel (*CFNetwork*), persistencia de datos mediante bases de datos con *SQLite*, los frameworks para acceso al núcleo que se basan básicamente en *wrappers* escritos en *Objective-C*. Servicios de localización, mediante GPS, célula, y Wi-Fi, también incorporan los frameworks de telefonía que se encargan de los eventos de llamadas. A este nivel también tenemos la habilidad para realizar compras desde las aplicaciones.
- **Media Services:** Concentran una base para la ejecución de la interfaz de usuario. Estas son tecnologías desarrolladas por Apple (*Core Graphics*, *Animation e Image*), se tiene acceso a *OpenGL* y *GLKit*, los que proporcionan la aceleración 3D de los gráficos mediante hardware. Otras implementaciones resguardadas por esta capa son el renderizado del texto, lectura y escritura de imágenes, acceso a contenido

audiovisual, ya sea local o remoto (*AirPlay*). Resumiendo, esta capa involucra a los componentes que permiten la manipulación de multimedia.

- **Cocoa Touch:** Es el framework encargado de todas las aplicaciones que se ejecutan en iOS. Estas están basadas en el paradigma orientado a eventos. Está escrito en *Objective-C*, asumiendo que es un lenguaje que proporciona una alta velocidad de respuesta, gracias a su rápida ejecución. Los frameworks que componen a *Cocoa Touch* son:
  - *Core Animation:* Permite crear interfaces de usuario, desde un modelo de programación basado en capas independientes con composición.
  - *Core Audio:* Procesamiento y captura de audio.
  - *Core Data:* Provee un manejo de los datos orientado a objetos que es fácil de usar y de entender. Pretende manejar los modelos de datos tanto de aplicaciones grandes como pequeñas (cualquier aplicación).

### 3.3.3 Comparativa de las plataformas móviles

Ambas plataformas se pueden asociar tomando algunos de sus bloques y comparándolos, por ejemplo por equivalencia en funcionalidad. Pasamos a hacer una comparación de los bloques de las arquitecturas mostradas mediante la siguiente tabla:

Bloque	Equivalencia en		Comparativa
	Android	iOS	
Núcleo	Linux	Darwin	Ambos tienen principios basados en el software libre, pero cada uno a provisto sus implementaciones de los parches necesario para mejoras y como base para el desarrollo de sus tecnologías posteriores. Una comparativa más técnica debería abarcar las diferencias entre un sistema UNIX y Linux.
Servicio del núcleo	Librerías y Runtime	Servicios del núcleo	Además de las librerías y sus equivalentes, Android implementa una capa de abstracción del hardware, donde su principal objetivo es brindar soporte a múltiples dispositivos. Por otro lado iOS está específicamente diseñado para un hardware ya definido. Además de esto, esta capa sólo es la encargada de servir de interfaz entre las llamadas al núcleo y las aplicaciones que las generan. Retomando a Android, es aquí donde se ubica la máquina virtual que asegura la portabilidad del código y probablemente la causante de algunos problemas de respuesta en la

			interfaz de Android.
Servicios de aplicación	Framework de Aplicaciones	<i>Media Services</i>	IOS se enfoca más al apartado audiovisual, mientras que Android especifica que en esta capa residen servicios importantes como la ubicación y el control de los sensores, etc. Al final se trata de servicios que las aplicaciones emplean para generar contenido.
Interfaz de usuario	Aplicaciones	<i>Cocoa Touch</i>	Android presenta aplicaciones que son mostradas al usuario, mientras que Cocoa Touch es el hramework, mediante el cual se crean estas aplicaciones en iOS. En este apartado mencionamos el hecho de que la velocidad de respuesta de iOS con respecto de la interfaz de Android, se ve afectada debido a la prioridad con la que cada plataforma procesa los eventos de la interfaz. iOS prioriza la entrada del usuario y la velocidad con la que se debe responder a esta. Por su parte Android no define una prioridad específica para la interfaz o para los procesos de fondo.

Tabla 2 Comparación de las arquitecturas de Android y iOS

### 3.3.4 Frameworks para desarrollo en plataformas móviles

Se ha decidido que la mejor herramienta para desarrollar la aplicación en el dispositivo móvil es la llamada *PhoneGap*, ya que cuenta con la ventaja de que con una sola codificación del programa, éste tengo soporte para los principales sistemas operativos de los dispositivos móviles.

*PhoneGap* es un framework de código abierto para el desarrollo móvil producido originalmente por *Nitobi*, adquirido posteriormente por *Adobe Systems*. Permite a los desarrolladores construir aplicaciones para dispositivos móviles utilizando *JavaScript*, *HTML5* y *CSS3*, en lugar de lenguajes nativos del dispositivo, como es el caso de *Objective-C* para iOS. Las aplicaciones resultantes son híbridas, es decir, que no son ni totalmente lenguaje nativo ni puramente basado en la web[16].

#### 3.3.4.1 Plataformas Soportadas

Actualmente *PhoneGap* soporta los dispositivos móviles con sistema operativo: Apple iOS, Google Android , HP Web OS , Microsoft Windows Phone , Nokia Symbian OS , RIM BlackBerry y Tizen[17].

Característica	iPhone / iPhone 3G	iPhone 3GS +	Android 1,0 a 4,0	Windows Phone	Blackberry 5.x-6.0 +	Blackberry 4.6-4.7	Symbian	HP Web OS	Tizen
Acelerómetro	Sí	Sí	Sí	Sí	Sí	N/A	Sí	Sí	Sí
Cámara	Sí	Sí	Sí	Sí	Sí	N/A	Sí	Sí	Sí
Brújula	N/A	Sí	Sí	Sí	N/A	N/A	N/A	Sí	Sí
Contactos	Sí	Sí	Sí	Sí	Sí	N/A	Sí	N/A	N/A
Expediente	Sí	Sí	Sí	Sí	Sí	N/A	N/A	N/A	N/A
Geo localización	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Medios de comunicación	Sí	Sí	Sí	Sí	N/A	N/A	N/A	N/A	Sí
Red	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Notificación (alerta)	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Notificación (sonido)	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Notificación (vibración)	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	N/A
Almacenamiento	Sí	Sí	Sí	Sí	Sí	N/A	Sí	Sí	Sí
Escáner de código de Barras y códigos QR	Sí	Sí	Sí	N/A	Sí	Sí	N/A	N/A	N/A

Tabla 3 Plataformas soportadas por PhoneGap

Y es gracias a esta tabla en la que también se basa la decisión de utilizar PhoneGap como la herramienta que más conviene al proyecto debido a que se planea desarrollar la aplicación móvil en los dispositivos Android e iOS.

### 3.4 Tecnologías del servidor

#### 3.4.1 Programación del sitio web

El sitio web será desarrollado en tecnología Java mediante el framework Spring MVC con seguridad implementada en *Shiro*. El motor de base de datos será MySQL como se mencionó anteriormente; la base de datos será mapeada en nuestro sistema mediante JPA-Hibernate. Por último, para darle menos carga al servidor se utilizará Ajax implementado en JQuery.

Además, se utilizará un Web-Service desarrollado en Axis1 para conectar el dispositivo móvil con los datos de los objetos del sitio de interés cultural.

### 3.4.2 Justificación de la elección de las tecnologías del servidor

Se utilizará Spring MVC por la modularidad bastante completa que tiene además de brindar robustez a los sistemas; la seguridad en Shiro se utilizará para no permitir intrusos en el sistema y así brindar protección a la información.

La base de datos se mapeará mediante JPA/Hibernate para evitar inyecciones de código SQL además de evitar hacer muchos scripts en SQL, manteniendo todo configurado con unas cuantas líneas para así lograr una migración de base de datos en unos cuantos segundos.

Para evitar saturar el servidor de peticiones, se utilizará Ajax para así lograr la mayor eficiencia posible en el sistema.

Sobre el *Web Service*, se utilizará Axis ya que al estar basado en SOAP, nos garantiza que es una implementación estandarizada y con ellos nos evitaremos muchos inconvenientes que se pudieran generar. Además, es una tecnología libre y con bastante documentación de cómo utilizarla.

## 3.5 Servidor de aplicaciones

Se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones.

Como consecuencia del éxito del lenguaje de programación Java, el término servidor de aplicaciones usualmente hace referencia a un servidor de aplicaciones Java EE. Entre los servidores de aplicación Java EE privativos más conocidos se encuentran WebLogic de Oracle (antes BEA Systems) y WebSphere de IBM. Entre los servidores de aplicaciones libres se encuentran, JBoss AS de JBoss, Geronimo de Apache, TomEE de Apache y GlassFish de Oracle.

### 3.5.1 Comparativa

	WebLogic	JBoss	GlassFish
<b>Licencia</b>	Pago	Gratis	Gratis
<b>Idioma</b>	Inglés	Inglés	Español
<b>Facilidad de Uso</b>	No	No	Si
<b>Pool de Conexiones</b>	Si	Si	Si

<b>Reconfiguración dinámica</b>	Si	Si	Si
<b>Compatibilidad con IDEs</b>	Si	Si	Si
<b>Cumplimiento de Estándares</b>	Si	Si	Si

Tabla 4 Comparación de servidores de aplicaciones

### 3.5.2 Justificación de la elección del servidor de aplicaciones

En base a la tabla mostrada en la sección 3.5.1, podemos notar que GlassFish nos ofrece las mayores ventajas de uso, ya que es gratuito y además nos ofrece algo que los otros no, está en español.

Cabe mencionar, que además de las ventajas antes mencionadas, tenemos otro punto a nuestro favor, GlassFish es un servidor con el que estamos familiarizados, por lo cual evitaremos entrar en una curva de aprendizaje que solo nos llevaría a desperdiciar tiempo de desarrollo.

### 3.6 Entornos de Desarrollo Integrados (IDEs)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes[18].

#### 3.6.1 Comparativa

	NetBeans	Eclipse	IntelliJ
<b>Licencia</b>	Gratis	Gratis	De Pago
<b>Sistema Operativo</b>	Multiplataforma	Multiplataforma	Multiplataforma
<b>Plugins</b>	Si	Si	Si
<b>Español</b>	Si	Si	No
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>- Bastantes tutoriales en internet</li> <li>- Autocorrección de código</li> <li>- Actualización de versiones constantes</li> </ul>	<ul style="list-style-type: none"> <li>- Bastantes tutoriales en internet</li> <li>- Todo el código es modificable</li> <li>- Actualización de versiones constantes</li> <li>- Expandible mediante plugins</li> </ul>	<ul style="list-style-type: none"> <li>- Muy ligero</li> <li>- Todo el código es modificable</li> <li>- Soporte total por parte del desarrollador</li> </ul>
<b>Desventajas</b>	<ul style="list-style-type: none"> <li>- Genera código basura</li> <li>- No permite modificar ciertas partes de código generadas</li> </ul>	<ul style="list-style-type: none"> <li>- Consume muchos recursos</li> </ul>	<ul style="list-style-type: none"> <li>- Casi no hay información sobre su uso</li> </ul>

	<p>automáticamente</p> <p>- Oculta archivos y detalles de código al usuario</p>		
--	---	--	--

Tabla 5 Comparación de IDEs

### 3.6.2 Justificación de la elección del IDE

Primeramente, IntelliJ es el más veloz y el mejor IDE de los 3, pero el hecho que sea de pago hace que lo descartemos de este proyecto.

Por lo tanto, el IDE elegido para trabajar es Eclipse ya que a pesar de tener la desventaja de que es en inglés y que NetBeans nos ofrecía trabajar en nuestro idioma nativo, es con el que más estamos familiarizados, además de que en el desarrollo de aplicaciones para dispositivos móviles mediante PhoneGap nos ofrece el mejor soporte entre los 3 comparados mediante complementos.

Otro punto a destacar que NetBeans nos oculta demasiado código y no permite modificarlo y esto muchas veces ocasiona gasto de recursos innecesarios y queremos que el programa sea lo más eficiente posible.

### 3.7 Sistema de Gestión de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (*DataBase Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permiten definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos[19].

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL y/o MS SQL Server.

Un SGBD debe permitir:

- ✓ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- ✓ Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- ✓ Manipular la base de datos: realizar consultas, actualizarla, generar informes.

### 3.7.1 Comparativa

SISTEMAS GESTORES DE BASES DE DATOS (SGBD)			
NOMBRE DEL SGBD	CARACTERÍSTICAS Y PROPIEDADES	REQUERIMIENTOS DE HARDWARE	PLATAFORMA
<b>PostgreSQL</b>	Alta concurrencia, además provee nativamente números de precisión arbitraria, texto de largo ilimitado, figuras geométricas y direcciones IP.	Al igual que en la mayoría de los SGBD este consume una gran cantidad de recursos, 2GB RAM, disco duro superior a 4 GB, procesador de 2GHz.	Multiplataforma (Windows, Mac, Unix, Linux, etc.)
<b>MySQL</b>	Es relacional, multihilo y multiusuario, posee un amplio subconjunto de ANSI SQL 99, procedimientos almacenados, disparadores, soporta SSL.	Consumidor medio/bajo de recursos.	Multiplataforma
<b>Firebird</b>	Arquitectura Cliente/ Servidor sobre protocolo TCP/IP, soporte de transacciones ACID, versión autoejecutable.	Bajos de requerimientos de hardware, contrario a los otros.	Multiplataforma
<b>Interbase</b>	Bases de datos en ficheros independientes, soporta el ANSI de SQL, distintos protocolos de red, arquitectura multigeneracional.	Bajo consumo de recursos (cuando mucho 256 de RAM, procesador no tan veloz, aproximadamente 40 MB en disco.)	Linux, Microsoft Windows y Solaris
<b>Oracle Database</b>	Eliminación de redundancia ociosa del centro de datos, precio muy elevado, reducción del BDA del riesgo de cambio.	A diferencia del anterior requiere alto consumo de recursos.	Multiplataforma
<b>Microsoft SQL Server</b>	Soporte de transacciones, estabilidad, seguridad, soporta procedimientos almacenados, permite trabajar en modo cliente-servidor.	Requiere una gran cantidad de recursos (2 GB de RAM para trabajar a gusto y sin preocupaciones, procesador de preferencia Dual Core).	Microsoft Windows
<b>Adaptive Server Enterprise (ASE)</b>	Técnicas de particionamiento semántico de tablas que aumentan la velocidad de acceso a los datos, columnas cifradas, arquitectura VSA de Sybase, conexiones seguras de SSL, conectividad cliente-servidor.	Consumidor medio de recursos (afortunadamente las PC de hoy tienen capacidades gigantescas).	Windows, Linux, Sun Solaris, IBM AIX, HP-UX, Mac OS, Silicon Graphics IRIX
<b>Informix</b>	Ejecución de múltiples hilos y procesadores virtuales que trabajan en paralelo, encriptación en integridad de mensajes, incluye volcado de seguridad.	Recursos mínimos	Multiplataforma

<b>Apache Derby</b>	APIs para JDBC y SQL, su código mide alrededor de 2000 KB comprimido, soporta cifrados completos, es multilenguaje, transacciones y recuperaciones ante errores ACID.	Recursos mínimos (512 de RAM, etc.)	Multiplataforma
<b>DBMan</b>	Personalización y servicios de apoyo, proceso de instalación automatizado basado en la web, puede ser alimentados por otros SGBD como MySQL, Oracle, etc.	Recursos mínimos	Multiplataforma

**Tabla 6 Comparación de Sistemas de Gestión de Base de Datos**

### 3.7.2 Justificación de la elección del SGBD

Con base a la anterior tabla y a experiencias personales, nos decidimos por utilizar MySQL en este proyecto, ya que nos ofrece opciones que satisfacen las de este proyecto y consume muy pocos recursos del sistema.

También, es el que más hemos utilizado en nuestros proyectos tanto personales como laborales y eso nos dará mucho tiempo de ahorro en el desarrollo.

## 3.8 Servicios Web

En la siguiente sección se describen las principales características de los servicios web para demostrar la importancia de la implementación de los mismos en el sistema.

### 3.8.1 Sistemas orientados a servicios

El acceso remoto a la información ha sido una actividad constante desde la inclusión del Internet a las actividades de la industria, así como de las actividades cotidianas de sus clientes. Si una empresa era capaz de poner a disposición de sus clientes esta información en un formato web, esta podría accederse mediante un navegador. Sin embargo, esto no era suficiente, pues si alguna aplicación externa requería hacer uso de dicha información, el acceso a los almacenamientos de la empresa se volvía complicado.

Como una solución a este problema se propone la idea de un servicio web, mediante el cual, las organizaciones que deseen hacer accesible la información a otros programas, puedan hacerlo definiendo y publicando una interfaz de servicio web. Esta interfaz define los datos disponibles y como se puede acceder a ellos.

De forma más general un servicio web es una representación estándar para cualquier recurso computacional o de información que pueda ser usado por otros programas[20].

La esencia de un servicio, es que la provisión de servicio es independiente de la aplicación que usa el servicio. Los proveedores de servicios pueden desarrollar servicios

especializados y ofertarlos a un cierto número de usuarios de servicios desde diferentes organizaciones. Las aplicaciones pueden construirse enlazando los servicios desde varios proveedores utilizando bien un lenguaje de programación estándar o bien un lenguaje de instrumentación de servicios.

### **3.8.2 Definición de un servicio web**

Dadas las características principales de un servicio web previamente mencionadas, podemos pasar a la definición que nos presentará una mejor idea de estos.

La definición dada por la W3C es la siguiente:

“Una aplicación de software identificada por una URI<sup>6</sup> cuyas interfaces y enlaces se pueden definir, describir y descubrir como una representación en XML. Un servicio web soporta interacción directa con otros agentes de software que usan intercambio de mensajes XML mediante los protocolos de Internet”.

Esta definición enfatiza que los servicios web deben de poder ser definidos, descritos y descubiertos, por lo tanto clarificando el significado de “accesible” y haciendo más concreta la noción de “orientado a Internet, interfaces basadas en estándares”. También menciona que los servicios web deben ser servicios similares al middleware<sup>7</sup> convencional. No sólo deben encontrarse en ejecución, sino que deben ser descritos y promocionados, de tal forma que sea posible escribir clientes que se enlacen e interactúen con ellos. En otras palabras, los servicios web son componentes que pueden ser integrados en aplicaciones distribuidas de mayor complejidad.

### **3.8.3 Estándares en los servicios web**

La iniciativa de los servicios web ha estado basada en estándares desde su implementación inicial y los estándares que incluyen muchos aspectos de los servicios web aún están desarrollándose. Los tres estándares fundamentales que permiten la comunicación entre servicios web se explican a continuación:

#### **3.8.3.1 SOAP**

El *Simple Object Access Protocol*, define una organización para el intercambio de datos estructurados entre servicios web. SOAP es una especificación basada en XML, está diseñada para ser simple y extensible. SOAP ayuda en el intercambio de información de ambientes descentralizados y distribuidos. De entre sus mejores características se encuentra

---

<sup>6</sup> *Uniform Resource Identifier*. Identifica el nombre y ubicación de un archivo o recurso en un formato uniforme.

<sup>7</sup> Software de conectividad que consiste en un conjunto de servicios que permiten interactuar a múltiples procesos que se ejecutan en distintas máquinas a través de una red. Ocultan la heterogeneidad y proveen de un modelo de programación conveniente para los desarrolladores de aplicaciones.

la habilidad de poder enviar datos XML sobre el medio de intercambio, es decir el Internet. Especifica un conjunto de reglas para la aplicación en el sistema objetivo para iniciar una serie de acciones una vez que se ha entregado la información. Algunos protocolos similares a SOAP son CORBA, IIOP y RMI[21].

### **3.8.3.2 WSDL**

*Web Service Description Language*, es el protocolo que define como puede representarse las interfaces de un servicio web. Es otro vocabulario basado en XML que forma uno de los bloques principales de los servicios web. Como el propio nombre lo designa, WSDL es la descripción en XML del servicio. Toda la información requerida del servicio, que incluye los tipos de dato enviados en los mensajes, el nombre del servicio, los parámetros de intercambio, los protocolos de enlace, etc. Se codifican en un documento XML[21].

### **3.8.3.3 UDDI**

*Universal Description, Discovery and Integration*. Es un estándar de búsqueda que define como puede organizarse la información de descripción de servicios, usada por los solicitantes de los servicios para encontrar servicios[22].

Las arquitecturas de aplicaciones de servicios web son arquitecturas débilmente acopladas en donde el enlace a los servicios puede cambiar durante la ejecución. Algunos sistemas se construyen únicamente usando servicios web y otros mezclarán servicios web desarrollados localmente.

## **3.8.4 Tipos de servicios web**

A nivel técnico, los servicios web pueden ser implementados de varias maneras. Los dos tipos de servicios web que se describen a continuación pueden distinguirse como servicios web “Big” y servicios web REST.

### **3.8.4.1 Big**

Los servicios web de este tipo emplean mensajes XML que siguen los estándares del protocolo SOAP. Tales sistemas contienen por lo general una descripción de las operaciones ofrecidas por el servicio, escrita por lo general en WSDL.

El mensaje en formato SOAP y la definición de las interfaces en el lenguaje WSDL han ganado una gran adopción[22].

### **3.8.4.2 REST**

*Representational State Transfer* (RESTful) por sus siglas en Inglés. Están diseñados para escenarios específicos. Estos servicios web se integran mejor con http que los servicios

basados en SOAP, no necesitan de mensajes en XML o definición de la API de los servicios en WSDL.

Dado que los servicios web REST emplean estándares bien conocidos definidos por la W3C y la IETF (HTTP, XML, URI, MIME) y tienen una infraestructura ligera que permite a estos servicios ser construidos con un mínimo de herramienta, su desarrollo es poco costoso y tiene pocas limitantes en su adopción.

### 3.8.5 Elección de servicios web

En el desarrollo de la aplicación se emplea un servicio web de tipo “Big”, ya que para consumir este tipo de servicios web se requiere de un mínimo de operaciones, así como la generación de los objetos desde el XML[22]. Con esto, podemos cargarlo fácilmente desde el lenguaje javascript y sería totalmente compatible con el framework de Phoneygap (citado anteriormente) que utilizaremos.

### 3.8.6 Comparativa de framework

En la siguiente tabla se muestran algunos frameworks que son del tipo “Big”.

FRAMEWORKS			
NOMBRE DEL FRAMEWORK	MODELO DE MENSAJEJO	LENGUAJE PROGRAMACIÓN	AÑO DE DESARROLLO
Apache Axis	Cliente/Servidor	Java/C++	2004-2006
Apache Axis 2	Cliente/Servidor/Asíncrono	Java/C	2006-Fecha Actual
gSOAP	Cliente/Servidor/Asíncrono	C/C++	1992-Fecha Actual
Zend	Cliente/Servidor	PHP	2006-Fecha Actual
.NET Framework	Cliente/Servidor	C#/VB/.NET	2002-Fecha Actual

Tabla 7 Comparación de Frameworks de Servicios Web de tipo “Big”

### 3.8.7 Justificación de la elección del framework

Como se puede observar en la tabla anterior, los únicos frameworks que nos sirven para este proyecto son Apache Axis y Apache Axis 2, y escogimos la segunda opción ya que es más actualizada y sigue teniendo soporte.

## 3.9 Requerimientos

Los requerimientos para el sistema representan la descripción de los servicios que puede proporcionarnos, así como las limitantes que este puede tener, reflejan las necesidades de un usuario para poder resolver un problema.

Los requerimientos del usuario están escritos en lenguaje natural y algunos diagramas los cuales definen las necesidades que el sistema debe cubrir y las limitaciones sobre las cuales debe funcionar el sistema.

Los requerimientos del sistema establecen a detalle las funciones, servicios y restricciones en que opera el sistema.

Es importante definir 2 conceptos que se utilizarán de ahora en adelante:

- ✓ **Catálogo:** Por catálogo nos referimos al conjunto de características que tendrán los objetos dentro del sitio de interés cultural. Por ejemplo: año, autor, etc.
- ✓ **Objeto:** Es cada ente en exposición dentro del sitio de interés cultural.

### 3.9.1 Requerimientos funcionales

Los requerimientos funcionales describen los servicios que debe proporcionar el sistema, la manera en que este debe comportarse ante diferentes factores de entrada y lo que debe realizar bajo circunstancias particulares.

#### 3.9.1.1 Sitio web

Etiqueta	Nombre	Descripción
<b>RF1</b>	Control de acceso de usuarios.	El sistema permitirá controlar el acceso y finalizar sesión de los usuarios.
<b>RF2</b>	Gestión de usuarios.	El sistema tendrá la capacidad de agregar, eliminar, buscar y modificar usuarios.
<b>RF3</b>	Administración de privilegios.	El administrador de sistema podrá asignar privilegios a los demás usuarios, habrá 2 clases de usuarios: administradores y capturistas.
<b>RF4</b>	Gestión de catálogos de información de los objetos.	El sistema tendrá la capacidad de agregar, eliminar, buscar y modificar el catálogo de la información que tendrán los objetos del sitio de interés que harán uso de los códigos QR.
<b>RF5</b>	Gestión de objetos.	El sistema tendrá la capacidad de agregar, eliminar, buscar y modificar objetos del sitio de interés que harán uso de los códigos QR.
<b>RF6</b>	Generación de códigos QR.	El sistema tendrá la capacidad de generar un código QR único para cada objeto del sitio de interés que hará uso de los códigos QR.
<b>RF7</b>	Recuperación de Contraseña.	El sistema tendrá la opción para que los usuarios recuperen su contraseña en caso de perderla.
<b>RF8</b>	Bitácora.	El sistema tendrá la capacidad de guardar las acciones de los usuarios en una bitácora.

Tabla 8 Requerimientos Funcionales - Sitio Web

#### 3.9.1.2 Dispositivo móvil

Etiqueta	Nombre	Descripción
<b>RF9</b>	Lectura de códigos QR.	El sistema será capaz de escanear los códigos QR.
<b>RF10</b>	Cargar información del objeto a través del código QR.	El sistema al escanear el código QR cargará la información del objeto en pantalla.

<b>RF11</b>	Servicio web	Se debe implementar un servicio web que conecte los datos de los objetos del sitio de interés cultural con el sistema.
<b>RF12</b>	Manipulación de la información.	El sistema podrá manipular la información cargada en pantalla, es decir, se podrá borrar, modificar o agregar información.
<b>RF13</b>	Envío de información a correo.	El sistema tendrá la capacidad de dar por concluida la visita y enviar la información recolectada a un correo en específico dado por el usuario.

**Tabla 9 Requerimientos Funcionales - Dispositivo Móvil**

### 3.9.2 Requerimientos no funcionales

Los requerimientos no funcionales describen las restricciones o funciones que ofrece el sistema, estas restricciones son comúnmente aplicadas en los tiempos, desarrollo y estándares del sistema.

Etiqueta	Descripción
<b>RNF1</b>	El sistema administrativo será una plataforma web para su fácil acceso y disponibilidad en cualquier momento que se requiera.
<b>RNF2</b>	El sistema contará con seguridad en el sitio web para evitar intrusos, debido a la información de los objetos con derechos de autor que maneja el sitio de interés cultural.
<b>RNF3</b>	El sistema, tanto web como móvil, tendrá una interfaz gráfica intuitiva y usable.
<b>RNF4</b>	El servidor soportará varias conexiones de los dispositivos móviles.
<b>RNF5</b>	El sistema en el dispositivo móvil será eficiente para evitar desbordes de memoria.
<b>RNF6</b>	En caso de fallas de algún componente, no debe haber pérdida de información.
<b>RNF7</b>	Se debe estructurar el código de una manera consistente y predecible, para su futura mantenibilidad.
<b>RNF8</b>	El sistema debe ser construido e implantado de tal manera que un cambio en los parámetros de negocio no obligue a la generación de una nueva versión del módulo.

**Tabla 10 Requerimientos No Funcionales**

### 3.10 Casos de uso

En esta sección se mostrarán los Casos de uso del sistema, tanto sus diagramas como las especificaciones de los mismos. Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores.

Cabe mencionar, que primero se mostrarán los diagramas de casos de uso y posteriormente la especificación de cada uno de ellos.

#### 3.10.1.1 Descripción de actores

Actor	Descripción
<b>Administrador</b>	Es aquél que llevará a cabo las funciones administrativas del sitio web, como son la

	gestión de capturistas del sistema y la gestión de catálogos de las características de los objetos del sitio de interés. Además, podrá realizar las funciones relacionadas a la generación de los códigos QR.
<b>Capturista</b>	Es aquél que llevará a cabo las funciones del sitio web que están relacionadas a la generación de los códigos QR. Administrará los objetos del sitio de interés en el sistema tal como su información claramente detallada, para después generar un código QR único para cada objeto.
<b>Visitante</b>	Es aquél que a través de su dispositivo móvil podrá escanear los códigos QR y cargar la información de los objetos en pantalla para manipularlos y después poder enviarlo a su correo o guardarlo.

**Tabla 11 Descripción de Actores**

### 3.10.2 Diagramas de casos de uso

A continuación, mediante diagramas, se describirá los casos de uso del sistema.

#### 3.10.2.1 General

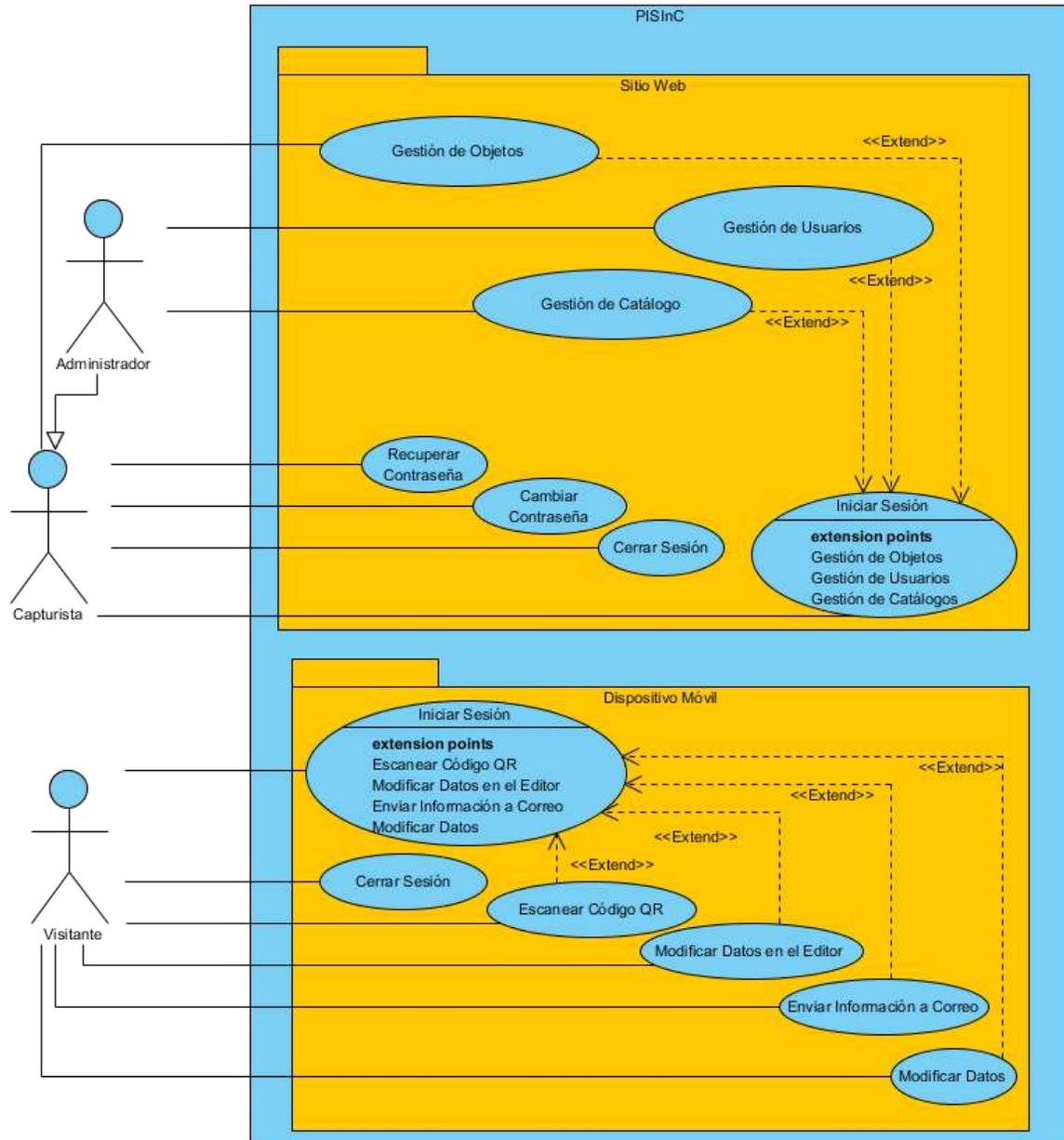


Figura 15 Diagrama de Caso de uso - General

### 3.10.2.2 Sitio web

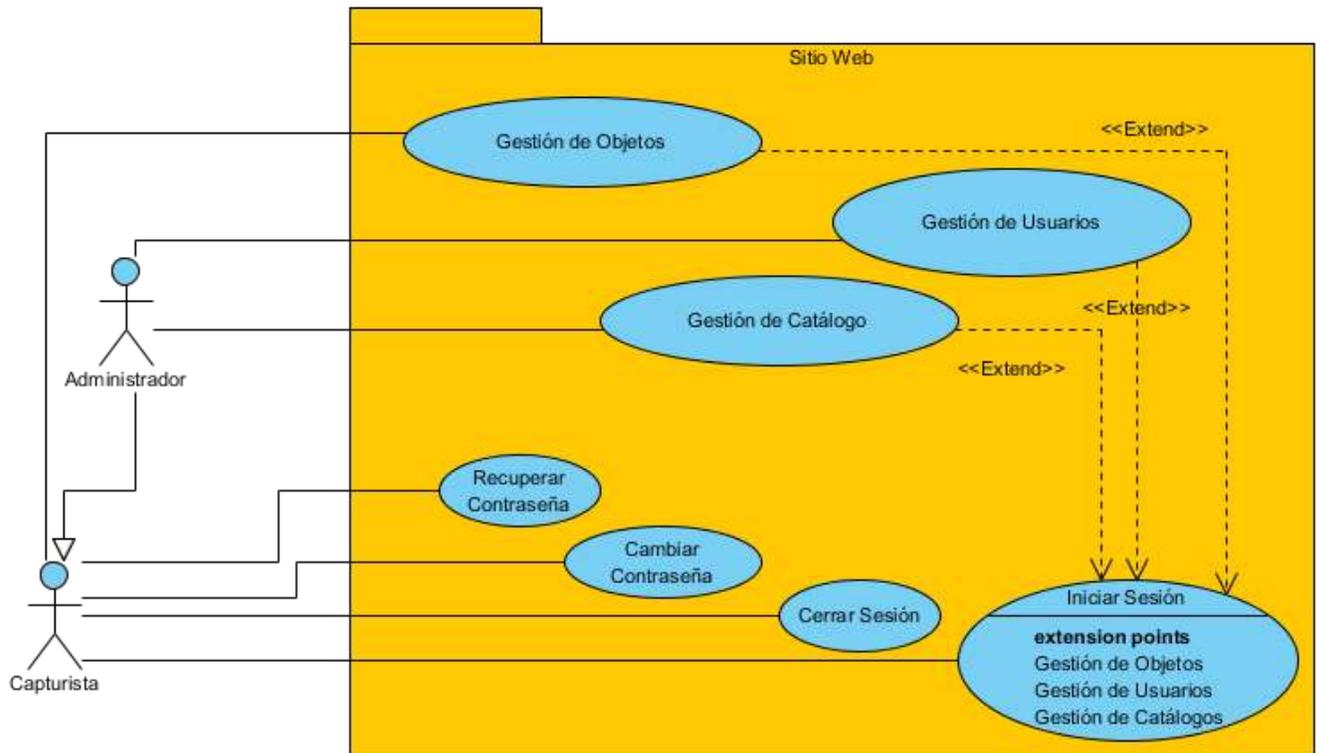


Figura 16 Diagrama de Caso de uso - Sitio Web

### 3.10.2.2.3 Gestión de catálogo

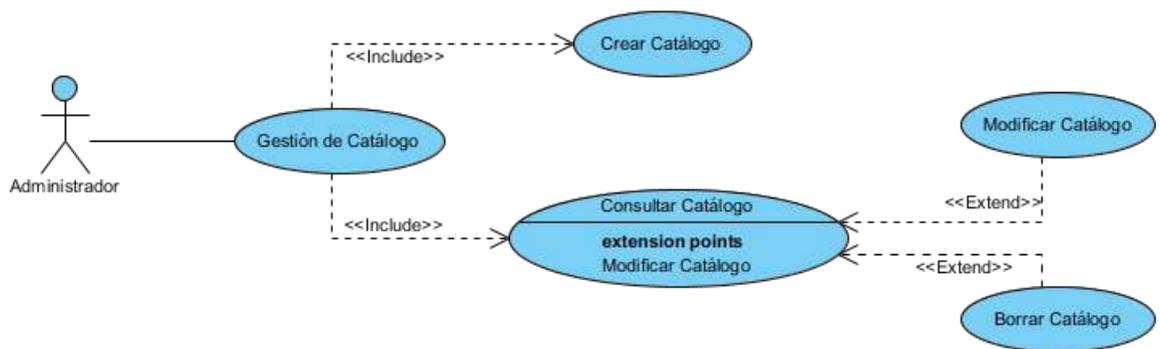


Figura 17 Diagrama de Caso de uso - Gestión de Catálogo

### 3.10.2.2.4 Gestión de objetos

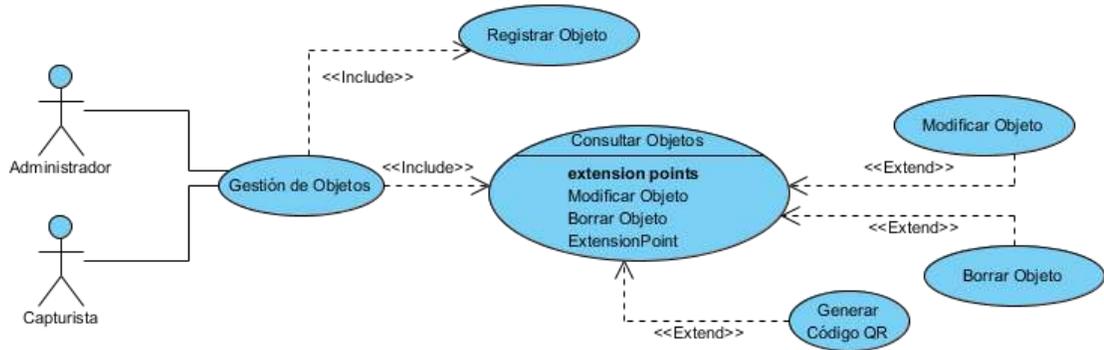


Figura 18 Diagrama de Caso de uso - Gestión de Objeto

### 3.10.2.2.5 Gestión de usuarios

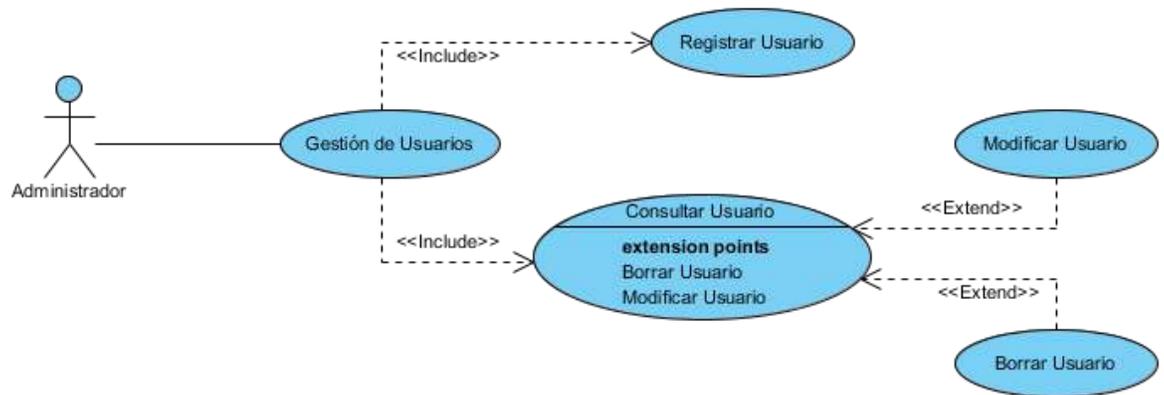


Figura 19 Diagrama de Caso de uso - Gestión de Usuarios

### 3.10.2.3 Dispositivo móvil

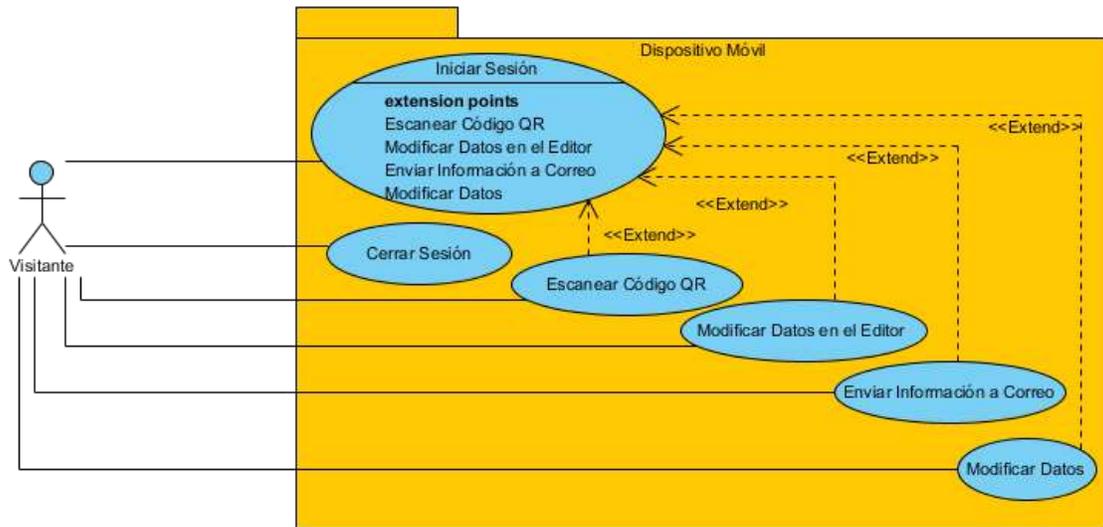


Figura 20 Diagrama de Caso de uso - Dispositivo Móvil

### 3.10.3 Especificación de casos de uso

La especificación de casos de uso, detalla y explica los diagramas de casos de uso y además define la trayectoria del sistema.

#### 3.10.3.1 Iniciar Sesión en Sitio Web

##### Descripción:

<b>CASO DE USO</b>	Iniciar Sesión en Sitio Web.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El usuario desea iniciar sesión para conectarse al sistema en el sitio web.	
<b>PRECONDICIÓN</b>	El usuario debe estar registrado en el sistema.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Introduce su usuario y su contraseña. 2. Envía los datos al sistema.	3. Confirma la validez de los datos introducidos en el paso 1. 4. Inicia una sesión para el usuario. 5. Muestra las opciones disponibles de acuerdo al rol del usuario.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>3a. Los datos introducidos en el paso 1 son inválidos:</b> 4. No inicia sesión. 5. Envía mensaje a usuario notificando que el usuario y/o la contraseña son incorrectos.
<b>POSTCONDICIÓN</b>	Se creó una sesión para el usuario. El usuario está conectado al sistema y tiene acceso a las funcionalidades de acuerdo a su rol.	

Tabla 12 Caso de uso 1 - Iniciar Sesión en Sitio Web

### 3.10.3.2 Cerrar Sesión en Sitio Web

**Descripción:**

<b>CASO DE USO</b>	Cerrar Sesión en Sitio Web.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El usuario desea cerrar sesión para desconectarse al sistema en el sitio web.	
<b>PRECONDICIÓN</b>	El usuario debe tener una sesión iniciada.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de salir del sistema.	2. Cierra la sesión del usuario y lo desconecta del sistema.
<b>POSTCONDICIÓN</b>	Se cerró la sesión para el usuario.	

**Tabla 13 Caso de uso 2 - Cerrar Sesión en Sitio Web**

### 3.10.3.3 Registrar Usuario

#### Descripción:

<b>CASO DE USO</b>	Registrar Usuario.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea registrar un usuario nuevo para que ingrese al sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona registrar nuevo usuario.  3. Escribe y envía los datos al sistema.	2. Pide el nombre, usuario, contraseña y rol del usuario a registrar.  4. Confirma la validez de los datos introducidos en el paso 3. 5. Registra el usuario.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 3 son inválidos:</b> 5. No registra al usuario. 6. Envía mensaje a usuario notificando que los datos son incorrectos.
	<b>3a. Cancela el registro de usuario:</b>	4. No registra al usuario.
<b>POSTCONDICIÓN</b>	El usuario queda registrado en el sistema y podrá iniciar sesión inmediatamente.	

Tabla 14 Caso de uso 3 – Registrar Usuario

### 3.10.3.4 Consultar Usuarios

**Descripción:**

<b>CASO DE USO</b>	Consultar Usuarios.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea consultar todos los usuarios del sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de consultar los usuarios.	2. Se muestran todos los usuarios registrados en el sistema.
<b>POSTCONDICIÓN</b>	Se deben observar todos los usuarios registrados en el sistema.	

Tabla 15 Caso de uso 4 – Consultar Usuarios

### 3.10.3.5 Modificar Usuario

#### Descripción:

<b>CASO DE USO</b>	Modificar Usuario.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea modificar los datos de un usuario registrado en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador. El administrador debe haber hecho el Caso de uso 3.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el usuario a modificar.  3. Escribe y envía los nuevos datos al sistema.	2. El sistema carga la información del usuario en el formulario con los campos nombre, usuario, contraseña y rol.  4. Confirma la validez de los datos introducidos en el paso 3. 5. Modifica los datos del usuario.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 3 son inválidos:</b> 5. No modifica los datos del usuario. 6. Envía mensaje al administrador notificando que los datos son incorrectos.
	<b>3a. Cancela la modificación de usuario:</b>	4. No modifica los datos del usuario.
<b>POSTCONDICIÓN</b>	Los datos del usuario son modificados y son verificables inmediatamente.	

Tabla 16 Caso de uso 5 – Modificar Usuario

### 3.10.3.6 Borrar Usuario

#### Descripción:

<b>CASO DE USO</b>	Borrar Usuario.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea eliminar a un usuario registrado en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador. El administrador debe haber hecho el Caso de uso 3.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el usuario a eliminar.  3. Confirma la eliminación del usuario.	2. Muestra una advertencia para continuar con la eliminación del usuario. 4. Elimina al usuario.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. El usuario a eliminar es el último administrador del sistema:</b> 5. No elimina al usuario. 6. Envía mensaje al administrador notificando que desea eliminar al último administrador del sistema.
	<b>3a. Cancela la eliminación del usuario:</b>	4. No elimina al usuario.
<b>POSTCONDICIÓN</b>	El usuario se elimina del sistema y ya no puede acceder al sistema.	

Tabla 17 Caso de uso 6 – Borrar Usuario

### 3.10.3.7 Crear Catálogo

#### Descripción:

<b>CASO DE USO</b>	Crear Catálogo.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea crear un catálogo de descripción de los objetos en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona crear nuevo catálogo.  3. Escribe y envía los datos al sistema.	2. Pide la descripción, imagen y campos del catálogo a registrar.  4. Confirma la validez de los datos introducidos en el paso 4.  5. Registra el catálogo en el sistema.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 3 son erróneos:</b>  5. No registra el catálogo.  6. Envía mensaje al administrador notificando que los datos son incorrectos.
	<b>3a. Cancela el registro de catálogo:</b>	4. No registra el catálogo.
<b>POSTCONDICIÓN</b>	Se crea el catálogo de la descripción de los objetos y a partir de este momento se pueden crear objetos. Este caso de uso deja de existir ya que solo puede existir un catálogo en el sistema.	

Tabla 18 Caso de uso 7 - Crear Catálogo

### 3.10.3.8 Consultar Catálogo

#### Descripción:

<b>CASO DE USO</b>	Consultar Catálogo.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea consultar el catálogo de descripción de los objetos del sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de consultar el catálogo.	3. Muestra el catálogo registrado en el sistema.
<b>POSTCONDICIÓN</b>	Se debe observar el catálogo registrado en el sistema.	

Tabla 19 Caso de uso 8 - Consultar Catálogo

### 3.10.3.9 Modificar Catálogo

#### Descripción:

<b>CASO DE USO</b>	Modificar Catálogo.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea modificar los datos del catálogo de descripción de los objetos del sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador.  El administrador debe haber hecho el Caso de uso 8.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el catálogo a modificar.  3. Escribe y envía los nuevos datos al sistema.	2. El sistema carga la información del catálogo en el formulario con los campos descripción, imagen y campos.  4. Confirma la validez de los datos introducidos en el paso 1.  5. Modifica los datos del catálogo.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 3 son inválidos:</b>  4. No modifica los datos del catálogo.  5. Envía mensaje al administrador notificando que los datos son incorrectos.
	<b>3a. Cancela la modificación de catálogo:</b>	4. No modifica los datos del catálogo.
<b>POSTCONDICIÓN</b>	Los datos del catálogo se han modificado y son verificables inmediatamente, los objetos que estaban registrados con el catálogo anterior, pierden los datos de las columnas modificadas.	

Tabla 20 Caso de uso 9 - Modificar Catálogo

### 3.10.3.10 Borrar Catálogo

**Descripción:**

<b>CASO DE USO</b>	Borrar Catálogo.	
<b>ACTOR</b>	Administrador.	
<b>DESCRIPCIÓN</b>	El administrador desea eliminar a un catálogo registrado en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador. El administrador debe haber hecho el Caso de uso 8.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el catálogo a eliminar.  3. Confirma la eliminación del catálogo.	2. Muestra una advertencia para continuar con la eliminación del catálogo.  4. Elimina al catálogo y todos los objetos que corresponden al catálogo.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	<b>3a. Cancela la eliminación del usuario:</b>	4. No elimina al usuario.
<b>POSTCONDICIÓN</b>	El usuario se elimina del sistema y ya no puede acceder al sistema.	

Tabla 21 Caso de uso 10 – Borrar Catálogo

### 3.10.3.11 Registrar Objeto

#### Descripción:

<b>CASO DE USO</b>	Registrar Objeto.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El administrador/capturista desea registrar un nuevo objeto en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador ó Capturista.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona registrar un nuevo objeto.  3. Escribe y envía los datos al sistema.	2. Pide el catálogo, nombre y campos del catálogo del objeto a registrar.  4. Confirma la validez de los datos introducidos en el paso 3.  5. Registra el objeto.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 3 son inválidos:</b>  4. No registra el objeto.  5. Envía mensaje al administrador/capturista notificando que los datos son incorrectos.
	<b>3a. Cancela el registro de objeto:</b>	4. No registra el objeto.
<b>POSTCONDICIÓN</b>	El objeto queda registrado en el sistema y a partir de este momento se puede generar un código QR único de éste.	

Tabla 22 Caso de uso 11 - Registrar Objeto

### 3.10.3.12 Consultar Objetos

#### Descripción:

<b>CASO DE USO</b>	Consultar Objetos.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El administrador/capturista desea consultar todos los objetos del sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador ó Capturista.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de consultar los objetos.	2. Se muestran todos los objetos registrados en el sistema.
<b>POSTCONDICIÓN</b>	Se deben observar todos los objetos registrados en el sistema.	

Tabla 23 Caso de uso 12 - Consultar Objetos

### 3.10.3.13 Modificar Objeto

#### Descripción:

<b>CASO DE USO</b>	Modificar Objeto.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El administrador/capturista desea modificar los datos de un objeto registrado en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador ó Capturista. El administrador/capturista debe haber hecho el Caso de uso 12.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el objeto a modificar. 2. Escribe los nuevos datos del objeto. 3. Envía los datos al sistema.	4. Confirma la validez de los datos introducidos en el paso 2. 5. Modifica los datos del objeto.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 2 son inválidos:</b> 5. No modifica los datos del objeto. 6. Envía mensaje al administrador/capturista notificando que los datos son incorrectos.
	<b>3a. Cancela la modificación de objeto:</b>	4. No modifica los datos del objeto.
<b>POSTCONDICIÓN</b>	Los datos del objeto se han modificado y son verificables inmediatamente.	

Tabla 24 Caso de uso 13 – Modificar Objeto

### 3.10.3.14 Borrar Objeto

**Descripción:**

<b>CASO DE USO</b>	Borrar Objeto.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El administrador desea eliminar a un objeto registrado en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador ó Capturista. El administrador/capturista debe haber hecho el Caso de uso 12.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el objeto a eliminar.  3. Confirma la eliminación del objeto.	2. Muestra una advertencia para continuar con la eliminación del objeto. 4. Elimina al objeto.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	3a. Cancela la eliminación del objeto: 4. No elimina al objeto	
<b>POSTCONDICIÓN</b>	El objeto se elimina del sistema y si existía un código QR de éste, deja de ser válido.	

Tabla 25 Caso de uso 14 – Borrar Objeto

### 3.10.3.15 Generar código QR

#### Descripción:

<b>CASO DE USO</b>	Generar código QR.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El administrador/capturista desea generar el código QR único de un objeto registrado en el sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador ó Capturista.  El administrador/capturista debe haber hecho el Caso de uso 12.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el objeto del cual se generará el código QR.	2. Genera el código QR a partir del identificador clave del objeto.  3. Muestra el código QR.
<b>POSTCONDICIÓN</b>	Se genera un código QR único que identificará al objeto.	

Tabla 26 Caso de uso 15 – Generar código QR

### 3.10.3.16 Cambiar Contraseña

#### Descripción:

<b>CASO DE USO</b>	Cambiar Contraseña.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El administrador/capturista desea cambiar su contraseña de acceso al sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar con una sesión iniciada y tener el rol de Administrador ó Capturista.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción cambiar contraseña.  3. Escriba y envía los datos al sistema.	2. El sistema pide la contraseña actual y la nueva contraseña.  4. Confirma la validez de los datos introducidos en el paso 3. 5. Modifica la contraseña.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. Los datos introducidos en el paso 3 son inválidos:</b> 5. No modifica la contraseña. 6. Envía mensaje al administrador/capturista notificando que los datos son incorrectos.
	<b>3a. Cancela la modificación de la contraseña:</b>	4. No modifica la contraseña.
<b>POSTCONDICIÓN</b>	La contraseña se modifica y para volver a iniciar sesión en el sistema debe ingresar la nueva contraseña.	

Tabla 27 Caso de uso 16 – Cambiar Contraseña

### 3.10.3.17 Recuperar Contraseña

**Descripción:**

<b>CASO DE USO</b>	Recuperar Contraseña.	
<b>ACTOR</b>	Administrador, Capturista.	
<b>DESCRIPCIÓN</b>	El usuario desea recuperar su contraseña de acceso al sistema.	
<b>PRECONDICIÓN</b>	El usuario debe estar registrado en el sistema y tener un correo válido.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción recuperar contraseña.  3. Escriba y envía el usuario.	2. El sistema pide el usuario.  4. Confirma la validez del dato introducido en el paso 3. 5. Envía al correo del usuario la contraseña.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>4a. El dato introducido en el paso 3 son inválidos:</b> 5. No envía correo. 6. Envía mensaje al administrador/capturista notificando que el dato es incorrecto.
	<b>3a. Cancela la recuperación de la contraseña:</b>	4. No envía correo.
<b>POSTCONDICIÓN</b>	-	

Tabla 28 Caso de uso 17 – Recuperar Contraseña

### 3.10.3.18 Iniciar Sesión en Dispositivo Móvil

#### Descripción:

<b>CASO DE USO</b>	Iniciar Sesión en Dispositivo Móvil.	
<b>ACTOR</b>	Visitante.	
<b>DESCRIPCIÓN</b>	El usuario desea iniciar sesión para conectarse al sistema en el dispositivo móvil.	
<b>PRECONDICIÓN</b>	El usuario debe estar conectado a la red local del sitio de interés.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	<ol style="list-style-type: none"> <li>1. Introduce su nombre y su correo electrónico.</li> <li>2. Envía los datos al sistema.</li> </ol>	<ol style="list-style-type: none"> <li>3. Confirma la validez de los datos introducidos en el paso 1.</li> <li>4. Inicia una sesión para el usuario.</li> </ol>
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<p><b>3a. Los datos introducidos en el paso 1 son inválidos:</b></p> <ol style="list-style-type: none"> <li>4. No inicia sesión.</li> <li>5. Envía mensaje a usuario notificando que el nombre y/o correo electrónico son incorrectos.</li> </ol>
<b>POSTCONDICIÓN</b>	<p>Se creó una sesión para el usuario.</p> <p>El usuario está conectado al sistema y tiene acceso a todas las funcionalidades del sistema en el dispositivo móvil.</p>	

Tabla 29 Caso de uso 18 - Iniciar Sesión en Dispositivo Móvil

### 3.10.3.19 Cerrar Sesión en Dispositivo Móvil

#### Descripción:

<b>CASO DE USO</b>	Cerrar Sesión en Dispositivo Móvil.	
<b>ACTOR</b>	Visitante.	
<b>DESCRIPCIÓN</b>	El usuario desea cerrar sesión para desconectarse al sistema en el dispositivo móvil.	
<b>PRECONDICIÓN</b>	El usuario debe estar conectado a la red local del sitio de interés. El usuario debe tener una sesión iniciada.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de salir del sistema. 3. Confirma que desea salir del sistema.	2. Muestra una advertencia para continuar con el cierre de sesión. 4. Elimina la información en su editor, cierra la sesión del usuario y lo desconecta del sistema.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	<b>3a. Cancela el cierre de sesión:</b>	4. No cierra la sesión.
<b>POSTCONDICIÓN</b>	Se cerró la sesión para el usuario.	

Tabla 30 Caso de uso 19 – Cerrar Sesión en Dispositivo Móvil

### 3.10.3.20 Escanear código QR

#### Descripción:

<b>CASO DE USO</b>	Escanear código QR.	
<b>ACTOR</b>	Visitante.	
<b>DESCRIPCIÓN</b>	El usuario desea escanear un código QR de algún objeto del sitio.	
<b>PRECONDICIÓN</b>	El usuario debe estar conectado a la red local del sitio de interés. El usuario debe tener una sesión iniciada.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de escanear código. 3. Apunta la cámara al código QR.	2. Se activa la cámara. 4. Escanea el código. 5. Valida el código QR y lo procesa. 6. Carga la información del objeto en el editor del dispositivo móvil.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
		<b>5a. El código QR es incorrecto:</b> 6. No carga ninguna información. 7. Envía mensaje a usuario notificando que el código QR es incorrecto.
<b>POSTCONDICIÓN</b>	La información guardada en el servidor sobre el objeto se carga correctamente en el dispositivo móvil.	

Tabla 31 Caso de uso 20 – Escanear código QR

### 3.10.3.21 Modificar Datos en el Editor

#### Descripción:

<b>CASO DE USO</b>	Modificar Datos en el Editor.	
<b>ACTOR</b>	Visitante.	
<b>DESCRIPCIÓN</b>	El usuario desea modificar la información cargada en su editor.	
<b>PRECONDICIÓN</b>	El usuario debe estar conectado a la red local del sitio de interés. El usuario debe tener una sesión iniciada.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Modifica la información mostrada en el editor del dispositivo móvil.	
<b>POSTCONDICIÓN</b>	Los datos cargados en el editor se modifican.	

Tabla 32 Caso de uso 21 – Modificar Datos en el Editor

### 3.10.3.22 Enviar Información a Correo

**Descripción:**

<b>CASO DE USO</b>	Enviar Información a Correo.	
<b>ACTOR</b>	Visitante.	
<b>DESCRIPCIÓN</b>	El usuario desea enviar a su correo toda la información recopilada dentro del sitio en su editor.	
<b>PRECONDICIÓN</b>	El usuario debe estar conectado a la red local del sitio de interés.  El usuario debe tener una sesión iniciada.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona la opción de enviar información a su correo.  3. Confirma que desea enviar la información a su correo.	2. Muestra una advertencia indicando que se eliminarán todos los datos y se enviará la información al correo.  4. Envía la información a su correo.  5. Se ejecuta el Caso de uso 16.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	<b>3a. Cancela el envío de información:</b>	4. No envía la información a su correo.
<b>POSTCONDICIÓN</b>	Se envía la información y se sale del sistema.	

Tabla 33 Caso de uso 22 – Enviar Información a Correo

### 3.10.3.23 Modificar Datos

**Descripción:**

<b>CASO DE USO</b>	Modificar Datos.	
<b>ACTOR</b>	Visitante.	
<b>DESCRIPCIÓN</b>	El usuario desea modificar sus datos en el sistema móvil.	
<b>PRECONDICIÓN</b>	El usuario debe estar conectado a la red local del sitio de interés. El usuario debe tener una sesión iniciada.	
<b>CURSO NORMAL</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona modificar datos.  3. Modifica sus datos. 4. Envía los datos al sistema.	2. Muestra un formulario con los datos cargados.  5. Modifica los datos.
<b>CURSO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	<b>3a. Cancela la modificación de sus datos:</b>	4. No modifica los datos.
<b>POSTCONDICIÓN</b>	Se modifican los datos del visitante.	

Tabla 34 Caso de uso 23 – Modificar Datos en el Editor

### 3.11 Análisis de factibilidad

Para la plataforma interactiva con los sitios de interés cultural utilizando los dispositivos móviles, se estudian algunos factores referentes a los recursos y restricciones que se pueden presentar en este con ayuda del estudio de factibilidad, el cual comprende de lo siguiente:

- ✓ **Factibilidad Técnica:** es aquél que verifica que el software y el hardware necesario para el desarrollo del sistema pueda ser desarrollado o en dado caso adquirido para dar solución al problema.
- ✓ **Factibilidad Operativa:** es una medida la cual nos indica si el proyecto puede ponerse en operación.
- ✓ **Factibilidad Económica:** permite señalar si el proyecto puede ser desarrollado desde el punto de vista financiero y además de pronosticar si los costos y tiempo de duración se justifican de acuerdo a los beneficios que conlleva.

#### 3.11.1 Factibilidad técnica

En esta parte se hace referencia al lenguaje de programación que se utilizará para el desarrollo del software de la aplicación, que en su totalidad son gratuitos y no requieren de licencias para poder desarrollarse e implementarse en los dispositivos móviles. Aunque para que la aplicación, en el caso de que se desee subir a *App Store* (aplicación de iOS) o bien a la de *Google Play* (para Android), se deben tomar en consideración los costos de la licencia para poder realizar estas acciones, así como las licencias necesarias para la instalación del software necesario en los equipos de cómputo requeridos. Los costos de dichas licencias se muestran en la tabla siguiente:

Plataforma	Precio Unitario (M.N)	Cantidad	Subtotales (M.N)
Android ( <i>GooglePlay</i> )	\$250.25*	1	\$250.25
iOS ( <i>App Store</i> )	\$990.99**	1	\$990.99
Visual Paradigm for UML Enterprise Edition	\$14,003.99***	1	\$14,003.99
Microsoft Office 2010	\$1,618.51	1	\$1,618.51
Adobe Photoshop CS6	\$6,996.99	1	\$6,996.99
Camtasia Studio	\$2,992.99	1	\$2,992.99
Servidor Dell PowerEdge R210 II	\$18,619.00	1	\$18,619.00
<b>Total</b>			\$45,502.73

Tabla 35 Especificación de costos de software

\* 25,00 USD es el costo real de la licencia, se considera el tipo de cambio al día: 21-October-2012

\*\* 99,00 USD es el costo real de la licencia, tipo de cambio considerado el 12-October-2012

\*\*\*1399,00 USD es el costo real de la licencia

\*\*\*\*699,00 USD es el costo real de la licencia

Además de las herramientas de software, es necesario contemplar el equipo de cómputo en el cual se va a desarrollar el sistema; para ello, se necesita el siguiente listado de hardware descrito en las tablas que aparecen a continuación:

Marca	<b>Dell Vostro 3460</b>
Sistema operativo instalado	Windows 7 Ultimate original 64 bit
Tipo de procesador	Intel Core i5-3210M
Ranuras de memoria	2 SDRAM DDR3
Monitor	LED HD 1366 x 768, 14.0"
Puertos	1 para audífono; 1 para micrófono; 3 USB 3.0; 1 LAN; 1 salida de línea de audio
Unidades ópticas	Unidad de bandeja 8x (Doble capa DVD +/- RW)
Peso del producto	2.23 kg
Dimensiones del producto	28 x 30,1 x 34,5 cm

**Tabla 36 Especificaciones técnicas de laptop Dell Vostro 3460**

Marca	<b>Acer Aspire One 721-3155</b>
Sistema operativo instalado	Windows 7 Ultimate original 32 bit
Tipo de procesador	AMD Athlon II Neo
Ranuras de memoria	DDR
Monitor	11.6"
Puertos	3 USB 2.0, 1 HDMI, 1 VGA, 1 entrada para micrófonos y audífonos
Unidades ópticas	No
Peso del producto	1.3 kg
Dimensiones del producto	29 x 283 x 196

**Tabla 37 Especificaciones técnicas de notebook Acer Aspire One 721-3155**

Marca	<b>MacBook Pro</b>
Sistema operativo instalado	OS X Mountain Lion
Tipo de procesador	Intel Core i5 de 2.5 GHz
Ranuras de memoria	2 DDR3
Monitor	LED 13.3"

Puertos	3 USB 3.0, Ethernet, Firewire, Thunderbolt
Unidades ópticas	SuperDrive de 8x
Peso del producto	2.06 kg
Dimensiones del producto	2.41 x 32.5 x 22.7cm

**Tabla 38 Especificaciones técnicas de laptop MacBook Pro**

Marca	<b>Sony Vaio Serie E</b>
Sistema operativo instalado	Windows 7 Ultimate Original 64-bits
Tipo de procesador	AMD E2-1800 APU
Ranuras de memoria	2 DDR3
Monitor	11"
Puertos	2 USB 2.0 y uno 3.0, 1 HDMI, Ethernet, Microfono y audifonos
Unidades ópticas	No
Peso del producto	1.5kg
Dimensiones del producto	29 x 20.3 x 2.3

**Tabla 39 Especificaciones técnicas de mini laptop Sony Vaio Serie E**

<b>Marca</b>	<b>Samsung Galaxy Ace</b>
<b>Sistema operativo instalado</b>	Android 2.2
<b>Tipo de procesador</b>	A 800MHz
<b>Ranuras de memoria</b>	N/A
<b>Pantalla</b>	3.5"
<b>Peso del producto</b>	113g
<b>Dimensiones del producto</b>	112.4 x 59.9 x 11.5 mm

**Tabla 40 Especificaciones técnicas del dispositivo con Sistema Operativo Android**

<b>Marca</b>	<b>iPod Touch 5ta Generación</b>
<b>Sistema operativo instalado</b>	iOS 6
<b>Tipo de procesador</b>	A5
<b>Ranuras de memoria</b>	N/A
<b>Pantalla</b>	4"

<b>Peso del producto</b>	88g
<b>Dimensiones del producto</b>	123.4m x 58.6m x 6.1m

Tabla 41 Especificaciones técnicas del dispositivo con Sistema Operativo iOS

De acuerdo a las especificaciones mostradas en las tablas anteriores, a continuación se muestra una tabla de estimación de los costos para los equipos a utilizar.

Producto	Precio unitario (M.N)	Cantidad	Subtotal (M.N)
<b>Dell Vostro 3460</b>	\$14,999.00	1	\$14,999.00
<b>Acer Aspire One 721-3155</b>	\$6,000.00	1	\$6,000.00
<b>MacBook Pro</b>	\$20,999.00	1	\$20,999.00
<b>Sony Vaio Serie E</b>	\$7,500.00	1	\$7,500.00
<b>Samsung Galaxy S3 mini</b>	\$6,329.00	1	\$6,329.00
<b>iPoud Touch 5ta. Generación</b>	\$4,799.00	1	\$4,799.00
<b>Total</b>			\$61,126.00

Tabla 42 Estimación de equipo de cómputo a utilizar

Algunos factores que se deben de tomar en cuenta son los servicios de los cuales se va a hacer uso por tal motivo en la tabla, mostrada a continuación, se muestran algunos servicios con que es indispensable contar para el desarrollo del sistema.

Producto / Servicio	Precio Unitario	Cantidad Mensual	Cantidad total	Subtotales
<b>Energía eléctrica</b>	\$1.08 / kwh	350 kwh	(\$350) x 5 bimestres	\$1,900.00
<b>Agua</b>	\$5.01 / m <sup>3</sup>	2 m <sup>3</sup>	(\$10.02) x 5 bimestres	\$50.00
<b>Línea telefónica / Internet</b>	\$349.00	3	(\$349) x 10 meses x 3	\$10,470.00
<b>Papelería</b>	\$1,500.00	\$150.00	(\$150) x 5 bimestres	\$1,500.00
<b>Total</b>				\$13,920.00

Tabla 43 Estimación de costos de servicios

En la tabla anterior se muestran los precios unitarios en relación al mes, aunque en la cantidad total pueden manejarse algunos costos bimestralmente. El presupuesto se determinó contemplando un tiempo de duración de 10 meses.

Una vez realizado el análisis, podemos concluir que lo referente al equipo necesario para el desarrollo, así como los entornos y software necesarios para el mismo, arroja un resultado

factible para el desarrollo de la plataforma interactiva para sitios de interés cultural. Cabe destacar que en realidad los servicios y productos que se plantearon en la última tabla de esta sección, serán cubiertos por parte de la Escuela Superior de Cómputo ya que en ésta se desarrollará el Trabajo terminal propuesto.

### 3.11.2 Factibilidad operativa

El estudio de factibilidad operativa describe el esfuerzo necesario para la construcción de la plataforma, de lo anterior entendemos por esfuerzo a la cantidad de personas requeridas para el desarrollo del proyecto. Por tal motivo para realizar dicha estimación se utilizará el modelo de estimación de costos COCOMO II. Este modelo involucra tres jerarquías de desarrollo las cuales son:

- ✓ **Modelo de composición de aplicación:** es utilizada dentro de las primeras etapas del proyecto donde las interfaces de usuario, el rendimiento del sistema y la tecnología con que se cuenta en ese momento son prioridad para el proyecto.
- ✓ **Modelo de fase de diseño previo:** es aplicada cuando se tiene ya definidos los requisitos del sistema, así como una arquitectura básica del software.
- ✓ **Modelo de fase posterior a la arquitectura:** es utilizada durante la fase de la construcción del software[23].

Analizando los modelos propuestos, se determinó desarrollar la estimación de acuerdo al modelo de composición de aplicación ya que se adapta a la fase en la que se encuentra el desarrollo del sistema.

#### 3.11.1.1 Modelo de composición de aplicación

Para realizar la estimación de este modelo es necesario encontrar el número de personas que se verán involucradas en la construcción del sistema, es por ello que se necesita evaluar la ecuación siguiente:

$$PM = \frac{NOP}{PROD}$$

En donde PM (Personas-Mes) se calcula obteniendo el cociente de la división de los NOP (Nuevos Puntos Objeto) entre PROD (Ratios de Productividad). Antes de realizar dicho cálculo es necesario obtener los nuevos puntos objeto y los ratios de productividad.

- ✓ **Nuevos puntos objeto.**

Los nuevos puntos objeto es una relación entre los puntos objetos y el porcentaje de reutilización del proyecto. Los puntos objetos pueden describirse a partir de tres aspectos: la cantidad de pantallas del sistema, de informe y de programación. En la siguiente tabla se

observa el peso de los tipos de objetos antes mencionados de acuerdo a la complejidad que representan.

Tipo de objeto	Peso de la complejidad		
	Básico	Intermedio	Avanzado
<b>Pantalla</b>	2	4	6
<b>Informe</b>	4	6	8
<b>Programación</b>		8	10

Tabla 44 Factores de peso de complejidad para tipos de objetos

De acuerdo al análisis de los tipos de objetos de la plataforma se cuenta con la información de la tabla siguiente:

Tipo de objeto	Complejidad Básica		Complejidad Intermedia		Complejidad Alta		Total de tipo de objeto
	Cantidad	Peso	Cantidad	Peso	Cantidad	Peso	
<b>Pantalla</b>	1	2	2	8	1	6	16
<b>Informe</b>	1	4	1	6	1	8	18
<b>Programación</b>	0	0	2	16	2	20	36
<b>Total</b>	2	6	5	30	4	34	70

Tabla 45 Análisis de tipos de objetos en cuanto a cantidad y pesos totales

En la tabla anterior, se observa la cantidad de pantallas, informes y programación que son necesarios de acuerdo a la complejidad que estos tienen, así como el cálculo de los pesos de estos. De lo anterior podemos destacar que el sistema cuenta con un total de 2 tipos de objeto con complejidad básica teniendo un peso de 6, 5 de complejidad intermedia con un peso de 30 y 4 tipos de objetos de complejidad alta con un peso de 34; teniendo un total de 11 tipos de objeto con un peso de 70.

Siguiendo con la estimación se procede a realizar el cálculo de los nuevos puntos objeto los cuales involucran el porcentaje de reutilización del proyecto, así como el peso total de los tipos de objeto antes calculado, para ello se utiliza la siguiente ecuación:

$$NOP = \frac{(Puntos Objeto) * (100 - \%Reutilización)}{100}$$

Por lo tanto sustituyendo obtenemos lo siguiente:

$$NOP = \frac{(70) * (100 - 50)}{100} = 35$$

Así que para el sistema obtenemos como resultado 35 que manifiestan los nuevos puntos objeto y es a partir de ellos que se realiza la estimación de la cantidad de personas por mes requeridas.

✓ **Ratios de productividad**

Los ratios de productividad quedan definidos por la capacidad y experiencia de los desarrolladores, así como la capacidad y madurez de ICASE, en la tabla siguiente se aprecian las valoraciones que pueden tener estos dos aspectos.

<b>Capacidad y experiencia de los desarrolladores</b>	Muy baja	Baja	Nominal	Alta	Muy alta
<b>Capacidad y madurez de ICASE</b>	Muy baja	Baja	Nominal	Alta	Muy alta
<b>Ratio de productividad</b>	4	7	13	25	50

**Tabla 46 Relación entre los ratios de productividad y pesos de acuerdo a su complejidad**

Tomando en cuenta los ratios de productividad de la Tabla 16 se evalúan dichas características, de acuerdo a las propiedades del sistema obteniendo los valores de la tabla siguiente:

Ratio de productividad	Complejidad	Peso
<b>Capacidad y experiencia de los desarrolladores</b>	Nominal	13
<b>Capacidad y madurez de ICASE</b>	Nominal	13

**Tabla 47 Evaluación de los ratios de productividad de acuerdo al proyecto**

Una vez calculado tanto el total de puntos objetos, así como los ratios de productividad es posible realizar el cálculo del número de personas-mes necesarias para el desarrollo del proyecto, para ello se utiliza la siguiente fórmula:

$$PM = \frac{NOP}{PROD}$$

Sustituyendo dicha fórmula con los valores encontrados se obtiene lo siguiente:

$$PM = \frac{35}{13} = 2.69$$

Considerando el cálculo que se realizó mediante el modelo de estimación COCOMO II arroja como resultado que para el desarrollo del sistema es necesario contar con tres personas-mes aproximadamente. Cabe destacar que para dicho cálculo se ha considerado que la parte de programación esté representada como un objeto. A continuación se presentan dos tablas, correspondientes a las actividades a desarrollar por cada una de las

personas involucradas en el desarrollo de la plataforma, así como la descripción de esos roles.

Persona	Actividad
<b>Persona 1</b>	<ul style="list-style-type: none"> <li>• Comunicación con el vínculo.</li> <li>• Planificación.</li> </ul>
<b>Persona 2</b>	<ul style="list-style-type: none"> <li>• Análisis.</li> <li>• Diseño.</li> </ul>
<b>Persona 3</b>	<ul style="list-style-type: none"> <li>• Construcción y adaptación.</li> <li>• Pruebas.</li> </ul>

Tabla 48 Actividades y responsabilidades de las personas encargadas de la plataforma

Actividad	Descripción	Perfil de roles desarrollador.
<b>Comunicación con el vínculo</b>	Establece una comunicación entre los desarrolladores y vínculo (museo).	<ul style="list-style-type: none"> <li>• Facilidad de palabra.</li> <li>• Conocimientos sobre ingeniería de software.</li> <li>• Conocimientos de cultura general</li> <li>• Habilidades sociales</li> <li>• Habilidades analíticas</li> <li>• Habilidades interpretativas</li> </ul>
<b>Planificación.</b>	Realiza las tareas necesarias para definir recursos, tiempo y la planificación del proyecto.	<ul style="list-style-type: none"> <li>• Habilidad para elaboración y control de planes.</li> <li>• Expresión oral y escrita de forma clara.</li> <li>• Comprensión y solución de problemas</li> </ul>
<b>Análisis.</b>	Actividades relacionadas para la comprensión y solución del problema, definir funciones del software y definir interacción del software.	<ul style="list-style-type: none"> <li>• Conocimientos sólidos de ingeniería de software.</li> <li>• Comprender el dominio de la información del software.</li> <li>• Comprensión de las funciones, comportamiento, rendimiento e interconexión del software.</li> </ul>
<b>Diseño.</b>	Proceso el cual se centra en la estructura de los datos, arquitectura del software, representaciones de interfaz y desarrollo procedimental.	<ul style="list-style-type: none"> <li>• Conocimientos sólidos de ingeniería de software.</li> <li>• Conocimientos acerca del Lenguaje Unificado de Modelado (UML)</li> <li>• Conocimientos acerca de arquitecturas de software.</li> </ul>
<b>Construcción y adaptación.</b>	Actividades relacionadas con la construcción, instalación y mantenimiento del software.	<ul style="list-style-type: none"> <li>• Conocimientos en lenguaje HTML5.</li> <li>• Conocimientos en lenguaje CSS3.</li> <li>• Conocimientos en lenguaje de programación JavaScript.</li> <li>• Conocimientos en el Framework Spring MVC</li> <li>• Conocimientos en SGBD</li> </ul>

<b>Pruebas.</b>	Proceso mediante el cual el software es sometido a diferentes escenarios donde se evalúa la forma en que éste se comporta en dichas condiciones.	<ul style="list-style-type: none"> <li>• Conocimientos en la tecnología de desarrollo PhoneGap.</li> <li>• Conocimientos en WebServices</li> <li>• Experiencia en el uso de casos de prueba.</li> <li>• Conocimiento en procedimientos de prueba.</li> </ul>
-----------------	--	--

**Tabla 49 Descripción de actividades del modelo de desarrollo software y perfil de roles de desarrollador**

### 3.11.3 Factibilidad económica

De acuerdo a los presupuestos realizados en el estudio de factibilidad técnica, se han elegido los recursos tanto software, hardware, así como los servicios necesarios para el desarrollo del sistema, a continuación se describen estos. Como bien se sabe, en la parte del software, las licencias, si se llega a tomar en cuenta la parte de poder subir las aplicaciones desarrolladas a las respectivas tiendas disponibles, esto conllevaría el costo necesario de la licencia.

Producto	Precio unitario (M.N)	Cantidad	Subtotal (M.N)
<b>Dell Vostro 3460</b>	\$14,999.00	1	\$14,999.00
<b>Acer Aspire One 721-3155</b>	\$6,000.00	1	\$6,000.00
<b>MacBook Pro</b>	\$20,999.00	1	\$20,999.00
<b>Sony Vaio Serie E</b>	\$7,500.00	1	\$7,500.00
<b>Samsung Galaxy Ace</b>	\$6,329.00	1	\$6,329.00
<b>iPod Touch 5ta Generación</b>	\$4,799.00	1	\$4,799.00
<b>Total</b>			\$61,126.00

**Tabla 50 Hardware necesario para el desarrollo de las plataformas**

Plataforma	Precio Unitario (M.N)	Cantidad	Subtotales (M.N)
<b>Android (GooglePlay)</b>	\$250.25	1	\$250.25
<b>iOS (App Store)</b>	\$990.99	1	\$990.99
<b>Visual Paradigm for UML Enterprise Edition</b>	\$14,003.99	1	\$14,003.99
<b>Microsoft Office 2010</b>	\$1,618.51	1	\$1,618.51
<b>Adobe Photoshop CS6</b>	\$6,996.99	1	\$6,996.99
<b>Camtasia Studio</b>	\$2,992.99	1	\$2,992.99
<b>Total</b>			\$26,853.72

**Tabla 51 Especificación de costos de software**

Los servicios que son necesarios para el desarrollo del sistema y que hasta cierto punto no son requeridos en algunos casos por este, aunque sí por los desarrolladores del mismo se describen en la siguiente tabla.

Producto / Servicio	Precio Unitario	Cantidad Mensual	Cantidad total	Subtotales
<b>Energía eléctrica</b>	\$1.08 / kwh	350 kwh	(\$350) x 5 bimestres	\$1,900.00
<b>Agua</b>	\$5.01 / m <sup>3</sup>	2 m <sup>3</sup>	(\$10.02) x 5 bimestres	\$50.00
<b>Línea telefónica / Internet</b>	\$349.00	3	(\$349) x 10 meses x 3	\$10,470.00
<b>Papelería</b>	\$1,500.00	\$150.00	(\$150) x 5 bimestres	\$1,500.00
<b>Total</b>				\$13,920.00

Tabla 52 Precio de los servicios/productos para desarrollo del sistema

Ahora bien, los recursos económicos necesarios para el desarrollo de la Plataforma interactiva para sitios de interés cultural mediante códigos QR vienen reflejados en la siguiente tabla:

Concepto	Costo total
<b>Hardware</b>	\$61,126.00
<b>Software</b>	\$26,853.72
<b>Servicios/Productos</b>	\$13,920.00
<b>Total</b>	\$100,899.72

Tabla 53 Costos Totales para el desarrollo del trabajo terminal

De acuerdo a la estimación realizada en el estudio de factibilidad económica se ha demostrado que el desarrollo de la “Plataforma interactiva para los sitios de interés cultural haciendo uso de los códigos QR” es factible, es importante mencionar que los servicios que se reportan en este trabajo, tales como agua, luz, etc.; son cubiertos por la Escuela Superior de Cómputo, debido a que el proyecto en cuestión será desarrollado dentro de las instalaciones de la misma. Algunos de los equipos de cómputo que se reportan aquí, de igual manera son recursos con los que ya se cuenta.

### 3.12 Análisis de riesgos

A continuación se describirá el Análisis de Riesgos para el desarrollo de este proyecto. Un Análisis de Riesgos es el estudio de las causas de las posibles amenazas y probables eventos no deseados y los daños y consecuencias que éstas puedan producir.

#### 3.12.1 Definición de riesgos

Evento o circunstancia cuya probabilidad de ocurrencia es incierta, pero que, en caso de aparecer, tiene un efecto (positivo o negativo) sobre los objetivos de un proyecto.

### 3.12.2 Tabla de riesgos

<b>Id</b>	<b>Nombre</b>	<b>Descripción</b>
<b>Técnicos</b>		
1	Cambian los requisitos en base a los del caso de estudio a implementar.	La institución donde se llevará a cabo el caso de estudio requiere que el sistema cuente con características nuevas.
2	Contratiempos con el framework.	El framework no facilita el desarrollo de la aplicación para ambas plataformas.
3	Falla el hardware empleado	Existe el riesgo de que el equipo empleado (laptops, servidor, móviles) para el desarrollo tenga fallas en su funcionamiento.
4	Robo de hardware.	Perdida del hardware empleado para desarrollar el sistema.
<b>Externos</b>		
1	No se obtiene la vinculación	No se obtiene la vinculación con algún museo para aplicar el caso de estudio.
2	Falta de infraestructura.	La Institución del caso de estudio no cuenta con la infraestructura necesaria.
3	Pérdida de vinculación	La vinculación con el Instituto que permitiría el caso de uso se rompe.
<b>De la organización</b>		
1	Falta de equipo para el desarrollo.	Requerimiento de más equipos para el desarrollo
2	Falta de recursos.	Falta de recursos para compra de equipo, licencias.
<b>Dirección de proyectos</b>		
1	Falta de comunicación en el equipo.	Falta de comunicación en el equipo, por tiempos o poca organización.
2	Desintegración	Desintegración del equipo, ya sea por el retiro de uno o todos los integrantes.
3	Módulos incompletos.	No terminar alguno de los módulos del sistema

Tabla 54 Tabla de riesgo

### 3.12.3 Plan de riesgos

#### 3.12.3.1 Técnicos

<b>PRT-1</b>	
<b>Riesgo</b>	Cambian los requisitos en base a los del caso de estudio a implementar.
<b>Probabilidad</b>	Media
<b>Impacto</b>	N/A
<b>Descripción</b>	La institución donde se llevará a cabo el caso de estudio requiere que el sistema cuente con características nuevas.
<b>Contexto</b>	Institución del caso de estudio, puede tener ciertos requerimientos que no estén implementados o establecidos en el sistema.
<b>Supervisión</b>	Comprobar que los requisitos que hemos planteado sean los adecuados para el caso de estudio que se implantará, esto hablando con los encargados del sitio de interés.

<b>Plan de contingencia</b>	El sistema está pensado en módulos, de tal manera que los cambios requeridos por la Institución serían modulares, pudiendo así repartir el trabajo entre los integrantes para una pronta atención de los cambios pedidos.
-----------------------------	---

Tabla 55 Plan de riesgo - PRT1

PRT-2	
<b>Riesgo</b>	Contratiempos con el framework.
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Alto
<b>Descripción</b>	El framework no facilita el desarrollo de la aplicación para ambas plataformas.
<b>Contexto</b>	El desarrollo de la aplicación móvil se ve afectado por las pocas prestaciones que el framework otorga.
<b>Supervisión</b>	Realizar prueba, desarrollando aplicaciones demo de los requerimientos que se tiene, viendo así la efectividad que tiene el framework. Se ha realizado una investigación previa de estas tecnologías, de tal manera que se ha comprobado su eficacia en el desarrollo.
<b>Plan de contingencia</b>	Buscar un framework con prestaciones equivalentes, o desarrollar las aplicaciones de forma nativa para las plataformas que ya se han especificado.

Tabla 56 Plan de riesgo - PRT2

PRT-3	
<b>Riesgo</b>	Falla de hardware empleado
<b>Probabilidad</b>	Muy Baja
<b>Impacto</b>	Alto
<b>Descripción</b>	Existe el riesgo de que el equipo empleado (laptops, servidor, móviles) para el desarrollo tenga fallas en su funcionamiento.
<b>Contexto</b>	El desarrollo de la aplicación móvil se ve afectado por la una falla inesperada en los equipos que se están usando.
<b>Supervisión</b>	Mantener copias de seguridad de los avances que se van teniendo del proyecto, de esta manera, si el hardware llegase a fallar, se podrá recuperar el progreso llevado hasta el momento. Su supervisión sólo queda en términos del cuidado que se le da al equipo.
<b>Plan de contingencia</b>	Se obtendría un préstamo, ya sea en efectivo para conseguir equipo nuevo, o en equipo físico para continuar con el desarrollo, en el tiempo en el que se llegue a la entrega, para posteriormente remplazar el equipo. Parte del equipo estará en periodo de garantía, por lo que se minimiza este riesgo.

Tabla 57 Plan de riesgo - PRT3

PRT-4	
<b>Riesgo</b>	Robo de hardware.
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Alto
<b>Descripción</b>	Existe el riesgo de que el equipo empleado (laptops, servidor, móviles) para

	el desarrollo se pierda debido a robo. Lo anterior, debido a que se tratan en su mayoría de equipo móvil que nos acompaña en transportes al sitio de interés, escuela y casa.
<b>Contexto</b>	El desarrollo de la aplicación móvil se ve afectado por la pérdida del equipo que se está empleando para su desarrollo
<b>Supervisión</b>	Resguardar de la mejor manera posible el hardware, esto en las instalaciones de la escuela, o si se mantendrá en las instalaciones del sitio de interés cultural.
<b>Plan de contingencia</b>	Parte del equipo se encontrará en garantía, sin embargo, para remplazar parte del otro hardware, se debería de hacer una petición de préstamo, ya sea monetario para conseguir nuevo equipo o con equipo físico.

Tabla 58 Plan de riesgo - PRT4

### 3.12.3.2 Externos

PRE-1	
<b>Riesgo</b>	No se obtiene la vinculación
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Muy alto
<b>Descripción</b>	No se obtiene la vinculación con algún museo para aplicar el caso de estudio.
<b>Contexto</b>	Ninguna de las instituciones posibles para un caso de estudio muestra interés en el sistema, por lo que no se nos permite probar su efectividad real como aplicación interactiva con el usuario.
<b>Supervisión</b>	Mantener contacto con las posibles instituciones que podrían prestar sus instalaciones para el caso de estudio, estar en constante comunicación con ellas y ofrecer el sistema como una solución para la interacción de los visitantes.
<b>Plan de contingencia</b>	Buscar sitios con menor cantidad de información, que involucre directamente a un museo, pero si a personas interesadas en montar una exposición, como es el caso de las exposiciones que se llevan a cabo en el metro de la Ciudad de México.

Tabla 59 Plan de riesgo - PRE1

PRE-2	
<b>Riesgo</b>	Falta de infraestructura
<b>Probabilidad</b>	Media
<b>Impacto</b>	Alto
<b>Descripción</b>	La Institución del caso de estudio no cuenta con la infraestructura necesaria.
<b>Contexto</b>	El lugar del caso de estudio.
<b>Supervisión</b>	Conocer las características del equipo con el que se cuenta en el sitio de interés y los cambios que se realizan a estos desde que se logró el convenio.
<b>Plan de contingencia</b>	Conseguir por nuestra parte, equipo que tenga las funciones que se buscan para desplegar el sistema en el sitio del caso de estudio.

Tabla 60 Plan de riesgo - PRE2

PRE-3	
<b>Riesgo</b>	Pérdida de vinculación
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Muy alto
<b>Descripción</b>	La vinculación con el Instituto que permitiría el caso de uso se rompe.
<b>Contexto</b>	El lugar del caso de estudio.
<b>Supervisión</b>	Mantener una comunicación estrecha con los encargados de la gestión del sitio de interés, mostrando avances y haciendo pruebas por lo menos una vez cada 15 días, esto fijando horarios para no interrumpir las actividades que se llevan a cabo en el lugar.
<b>Plan de contingencia</b>	Dependerá del tiempo en el que esto pudiera llegar a suceder, si se presenta en los comienzos de las pruebas, se buscaría una nueva vinculación; en el plazo medio o en la terminación del proyecto sería difícil encontrar una nueva vinculación y ajustarla a las necesidades del nuevo sitio.

Tabla 61 Plan de riesgo – PRE3

### 3.12.3.3 De la organización

PRO-1	
<b>Riesgo</b>	Falta de equipo para el desarrollo.
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Medio
<b>Descripción</b>	Requerimiento de más equipos para el desarrollo.
<b>Contexto</b>	Desarrollo y el sitio en donde se lleva a cabo el caso de estudio.
<b>Supervisión</b>	Se ajustará el requerimiento de equipo nuevo por parte de la institución (puntos de acceso, servidor, por ejemplo) en cuanto se realice el convenio y se tengan las características de su infraestructura tecnológica.
<b>Plan de contingencia</b>	Se realizará la petición de un préstamo para poder conseguir el equipo que sea necesario y poder instalarlo en el sitio donde es requerido.

Tabla 62 Plan de riesgo - PRO1

PRO-2	
<b>Riesgo</b>	Falta de recursos monetarios.
<b>Probabilidad</b>	Media
<b>Impacto</b>	Medio
<b>Descripción</b>	Falta de recurso monetario para la compra de software (licencias, cuentas, etc.) y equipo para desarrollar y hacer pruebas.
<b>Contexto</b>	Desarrollo y pruebas del sistema.
<b>Supervisión</b>	Emplear software gratuito y con suficiente documentación disponible para evitar la necesidad de pagar por soporte, de igual manera se planearon los requerimientos de hardware que se tendrían para evitar contratiempos.
<b>Plan de contingencia</b>	Se realizará la petición de un préstamo para poder conseguir el equipo que sea necesario.

Tabla 63 Plan de riesgo - PRO2

### 3.12.3.4 Dirección de proyectos

PRD-1	
<b>Riesgo</b>	Falta de comunicación en el equipo.
<b>Probabilidad</b>	Media
<b>Impacto</b>	Alto
<b>Descripción</b>	Falta de comunicación en el equipo, por tiempos o poca organización.
<b>Contexto</b>	Desarrollo y pruebas del sistema. Visita a directores y sinodales para entrega de avances.
<b>Supervisión</b>	Se plantea que el equipo se reúna por lo menos una vez a la semana, para avanzar, ver el estado del proyecto y realizar comentarios sobre el estado actual de la aplicación. De igual forma se planea realizar visitas a los directores de TT por lo menos cada 15 días, manteniendo la comunicación por vía electrónica y acudiendo a las citas que se nos indique con forme sea necesario.
<b>Plan de contingencia</b>	Si en el primer mes de trabajo terminal 2, no se cuenta con avances considerables y de los que se han planteado para ese periodo de tiempo, se realizará una junta para comprobar el estado y disposición de cada uno de los integrantes en cuanto al desarrollo del trabajo terminal. En ella se decidirá si es recomendable o no seguir trabajando en la manera establecida.

Tabla 64 Plan de riesgo - PRD1

PRD-2	
<b>Riesgo</b>	Desintegración del equipo.
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Alto
<b>Descripción</b>	Desintegración del equipo, ya sea por el retiro de uno o todos los integrantes.
<b>Contexto</b>	Desarrollo y pruebas del sistema. Visita a directores y sinodales para entrega de avances.
<b>Supervisión</b>	Habrán un registro de los entregables que ha realizado cada miembro del equipo. Dado el caso en el que uno o más de los integrantes no estén desarrollando el proyecto, se llevará a cabo una reunión, entre los integrantes para conocer los motivos, para posteriormente reunirse con los directores y hacer las notificaciones que sean necesarias.
<b>Plan de contingencia</b>	Quedando dos integrantes del trabajo terminal, se ajustarían las características de los módulos, quitando opciones o características que tal vez no sean del todo necesarias para una posible entrega. En dado caso de que un sólo integrante quede en el equipo, se pasará a discutir el futuro del trabajo terminal con los directores y analizar si es posible que se ajusten requerimientos para poder entregarlo. En el peor de los casos se cancelará el trabajo.

Tabla 65 Plan de riesgo - PRD2

<b>PRD-3</b>	
<b>Riesgo</b>	Módulos incompletos.
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Alto
<b>Descripción</b>	No terminar alguno de los módulos del sistema.
<b>Contexto</b>	Desarrollo del sistema, pruebas en el caso de uso.
<b>Supervisión</b>	Se tiene un plan de entrega para cada uno de los módulos, por lo que el tiempo de desarrollo y entrega estará controlado mediante este mecanismo. Con ello estaremos cocientes de la existencia de retrasos y del tiempo que debemos dedicar a cada uno de los módulos para evitar así retrasos.
<b>Plan de contingencia</b>	Dependerá del módulo que quede incompleto, se dará mayor prioridad para poder terminarlo trabajando un mayor tiempo en él, y al tratarse de un riesgo, deberán ignorarse otras tareas pendientes para poder terminar el módulo.

**Tabla 66 Plan de riesgo - PRD3**

# CAPÍTULO 4 DISEÑO

## 4.1 Arquitectura del sistema

En esta sección se muestra la arquitectura de la Plataforma interactiva para sitios de interés cultural mediante códigos QR, se puede observar que está dividida en 2 módulos: Sitio Web y Dispositivo Móvil.

### 4.1.1 Arquitectura general del sistema



Figura 21 Arquitectura General del Sistema

### 4.1.2 Arquitectura del sistema - sitio web



Figura 22 Arquitectura del Sistema - Sitio Web

#### 4.1.2.1 Captura de objetos

Este módulo será el encargado de recibir la información de los objetos del sitio de interés y con ellos generar un código QR único para poder identificarlo después a través del dispositivo móvil. A grandes rasgos solo cuenta con 2 pasos: el registro de la información del objeto y la generación del código QR.

Se implementará en el sitio web y la información será capturada por los usuarios “Administrador” y “Capturista”.

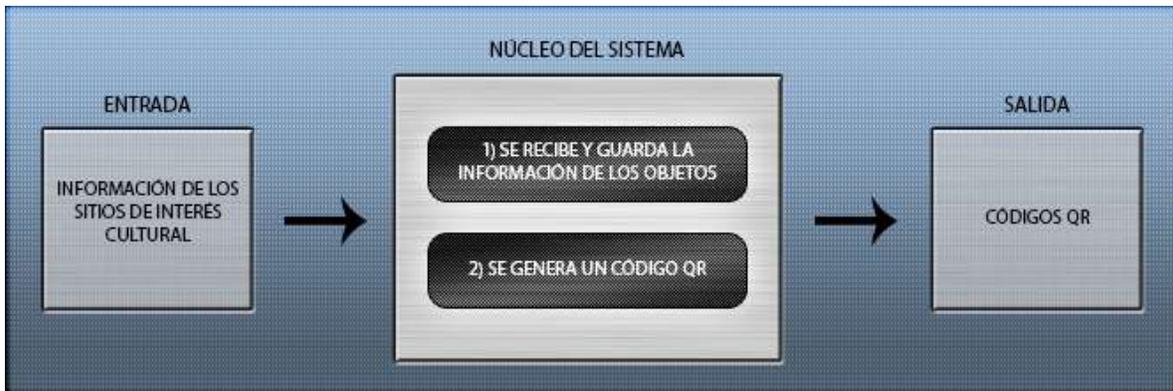


Figura 23 Arquitectura del sistema - Captura de objetos

#### 4.1.2.2 Gestión de usuarios

Este módulo será el encargado de administrar los usuarios, tanto “Administrador” como “Capturista”, para llevar un control correcto de estos y evitar intrusos en el sistema.

El módulo lo utilizará el usuario “Administrador” y las acciones que se podrán realizar son altas, bajas y cambios.

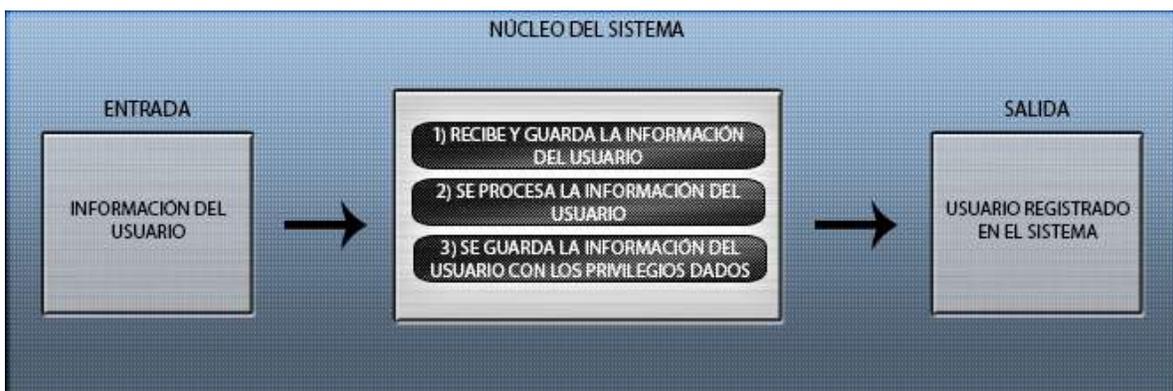


Figura 24 Arquitectura del sistema - Gestión de usuarios

### 4.1.2.3 Gestión de catálogos

La función de este módulo es primordial ya que definirá las características de los objetos del sitio de interés para poder guardar la información de estos posteriormente. Esta funcionalidad hace que nuestro sistema se convierta en una plataforma y pueda servir para cualquier sitio de interés.

Así mismo, se podrán gestionar las características con acciones como altas, bajas y cambios; estas acciones serán realizadas por el usuario “Administrador”.

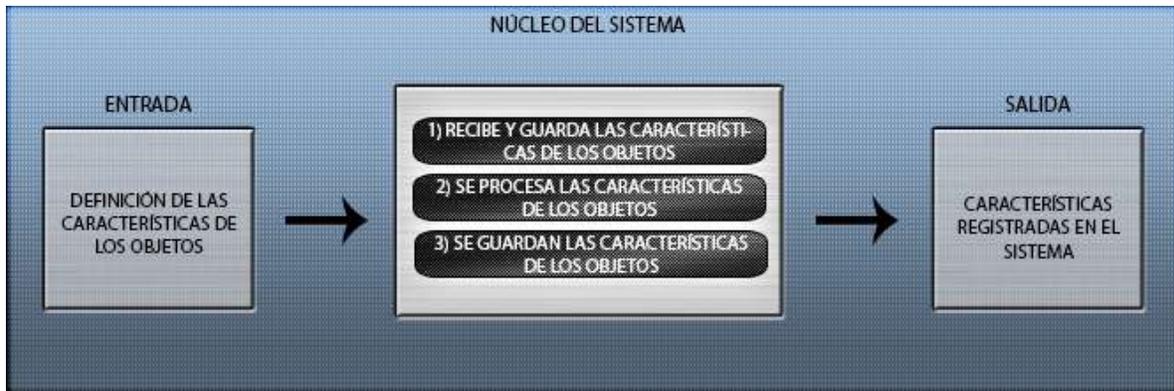


Figura 25 Arquitectura del sistema - Gestión de catálogos

### 4.1.3 Arquitectura del sistema - Dispositivo móvil



Figura 26 Arquitectura del sistema - Dispositivo móvil

#### 4.1.3.1 Carga de información

Este módulo será el más complejo a desarrollar ya que incluye toda la lógica de escanear el código QR, cargar la información y manipularla borrando, modificando o agregando información. El resultado final de este módulo es la información ya manipulada y con el formato que el usuario desee.

Se implementará en el dispositivo móvil y el usuario que realizará este proceso es “Usuario final del sitio de interés”.

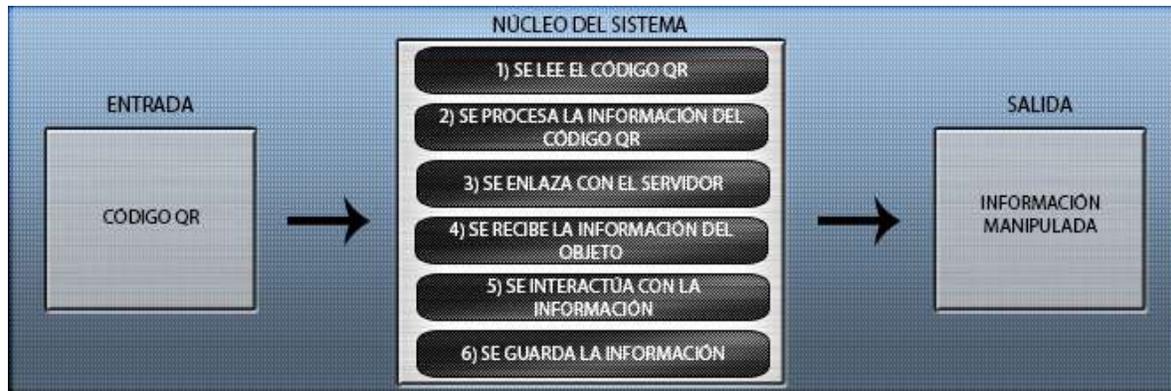


Figura 27 Arquitectura del sistema - Carga de información

#### 4.1.3.2 Procesamiento de la información

Este módulo será el paso final para el “Usuario final del sitio de interés”, ya que se enviará la información, que recolecto y manipuló en todo el sitio de interés, a un correo en específico dado por él. La información primeramente se enviará al servidor, éste la procesará y la convertirá a formato PDF, éste archivo se enviará al correo.

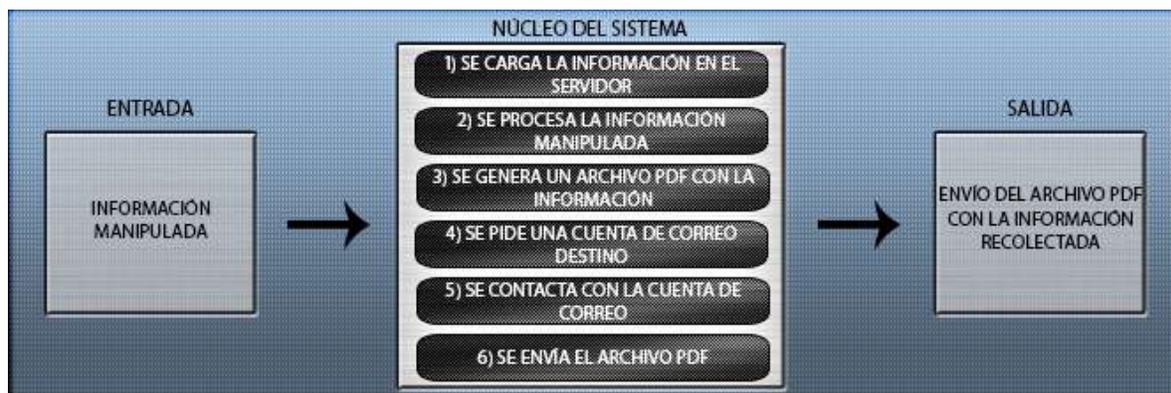


Figura 28 Arquitectura del sistema - Procesamiento de la información

## 4.2 Diagramas de secuencia

En esta sección se muestran los diagramas de secuencia del sistema, divididos en 2 partes: Sitio Web y Dispositivo Móvil. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

### 4.2.1 Sitio web

A continuación se pueden ver los diagramas de secuencia correspondientes al Sitio Web del sistema.

#### 4.2.1.1 Iniciar Sesión en Sitio Web

El administrador o capturista Iniciarán sesión en la interfaz web (Sistema), siendo estos validados con los registros en la Base de Datos (BD), se cargará la página que corresponda de acuerdo al rol que ha iniciado sesión. Si los datos son inválidos se notificará al usuario.

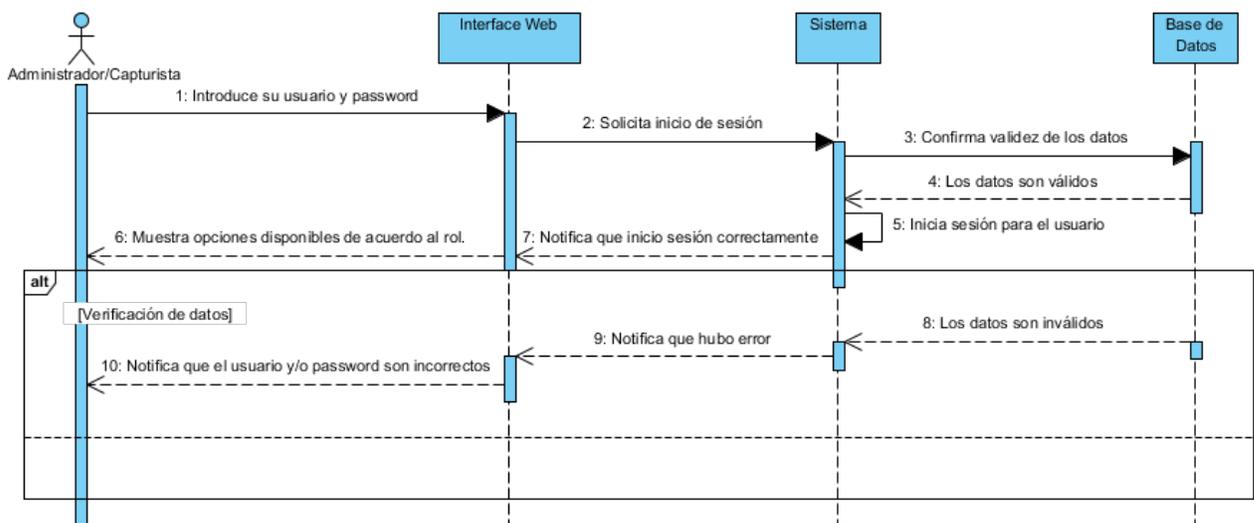


Figura 29 Diagrama de secuencia - Iniciar Sesión en Sitio Web

### 4.2.1.2 Cerrar Sesión

El administrador o capturista podrán elegir la opción de cerrar sesión, para dar por terminada su actividad en el sistema, quien los desconecta y notifica en la interfaz web.

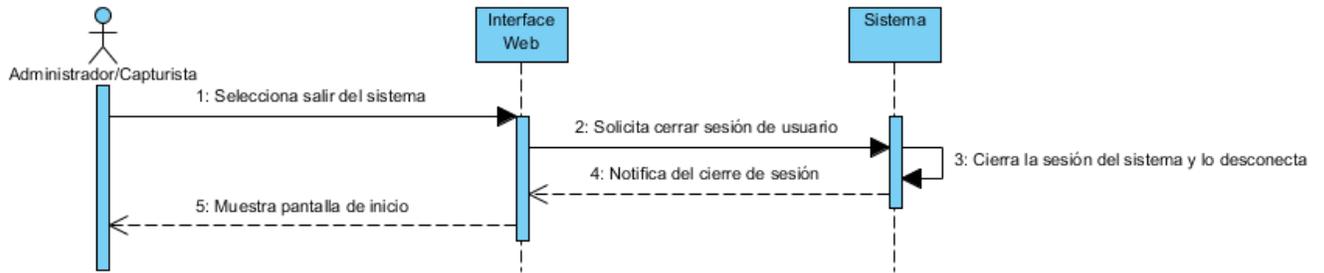


Figura 30 Diagrama de secuencia - Cerrar Sesión en Sitio Web

### 4.2.1.3 Registrar Usuario

El administrador o capturista ingresan los datos del usuario en la interfaz web, los cuales son enviados y validados por el sistema, por último se notificará al usuario si el registro se realizó o no correctamente.

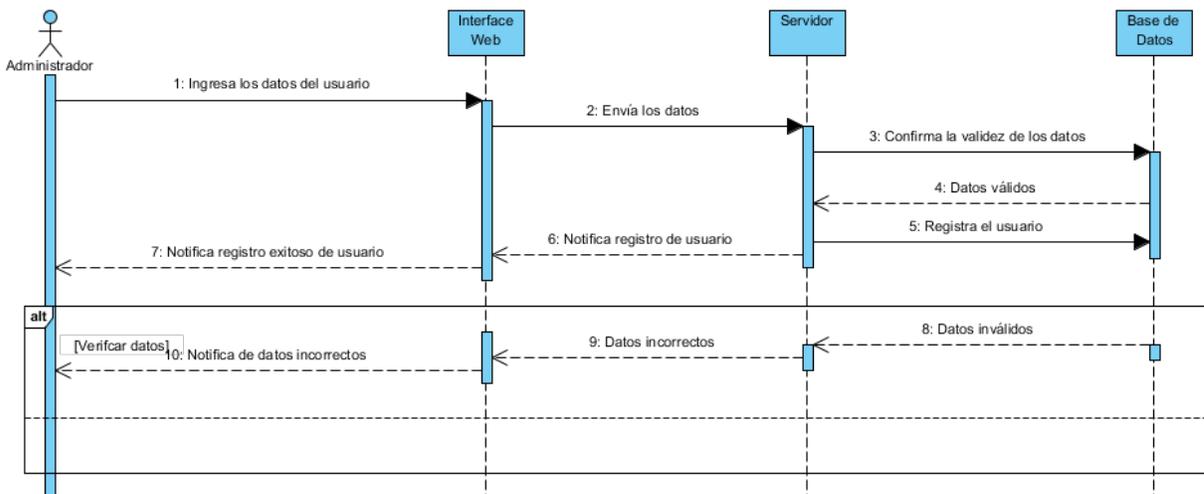


Figura 31 Diagrama de secuencia - Registrar Usuario

#### 4.2.1.4 Consultar Usuario

La consulta de usuarios se realiza mediante la elección de la opción “Consultar usuarios”, cargando una página con los usuarios realizados por el sistema, esta es iniciada únicamente por el administrador.

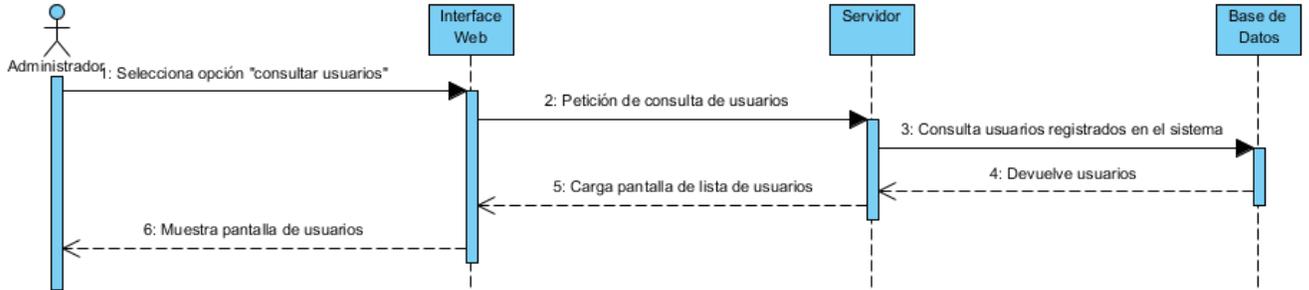


Figura 32 Diagrama de secuencia - Consultar Usuario

#### 4.2.1.5 Modificar Usuario

El administrador selecciona el usuario a modificar, el sistema cargará los datos en la pantalla de modificación de los datos donde estos podrán ser editados por el administrador y enviados al servidor para su actualización.

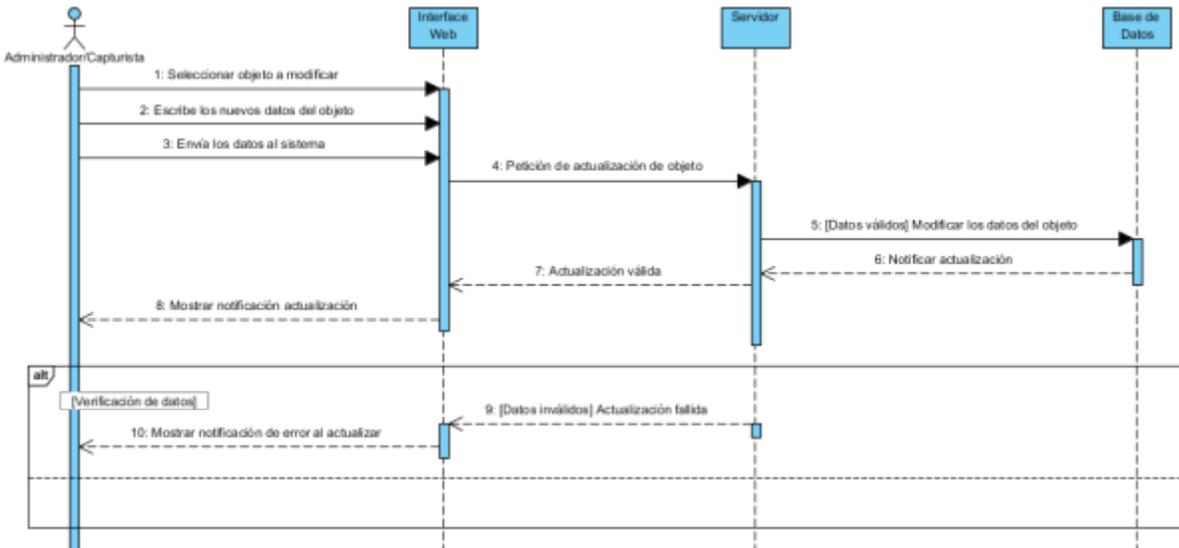


Figura 33 Diagrama de secuencia - Modificar Usuario

### 4.2.1.6 Borrar Usuario

El administrador selecciona al usuario a ser eliminado del sistema, el sistema verificará que no se trate del último administrador registrado, por lo que notificará al usuario si así fuese.

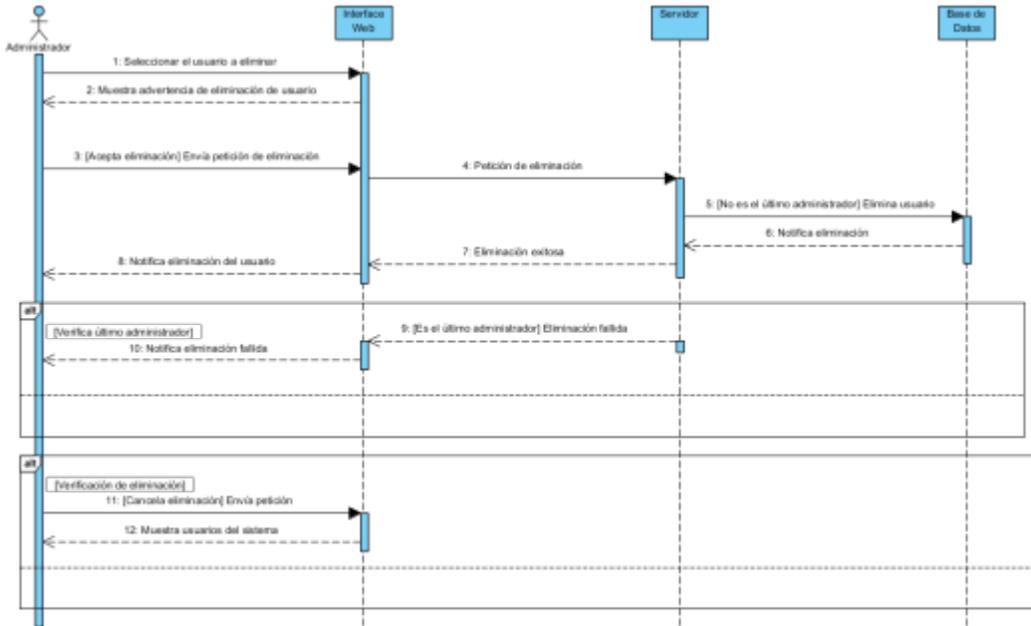


Figura 34 Diagrama de secuencia - Borrar Usuario

### 4.2.1.7 Crear Catálogo

El administrador se encargará de registrar los nuevos catálogos proporcionando una descripción de los objetos que este contendrá, mediante los tipos de datos que estén permitidos en el sistema para la creación de los catálogos.

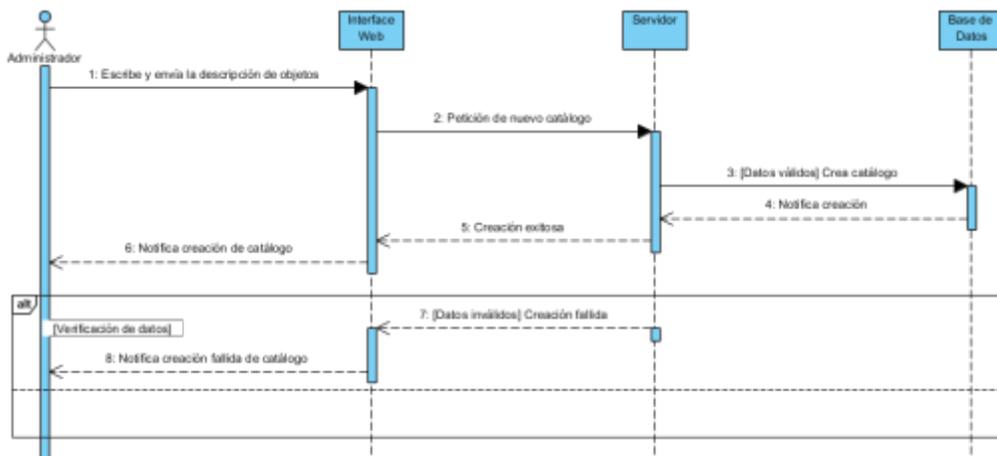


Figura 35 Diagrama de secuencia - Crear Catálogo

#### 4.2.1.8 Consultar Catálogo

Se selecciona la opción de consulta de los catálogos por parte del administrador, con lo que se carga la lista de catálogos disponibles en el sistema.

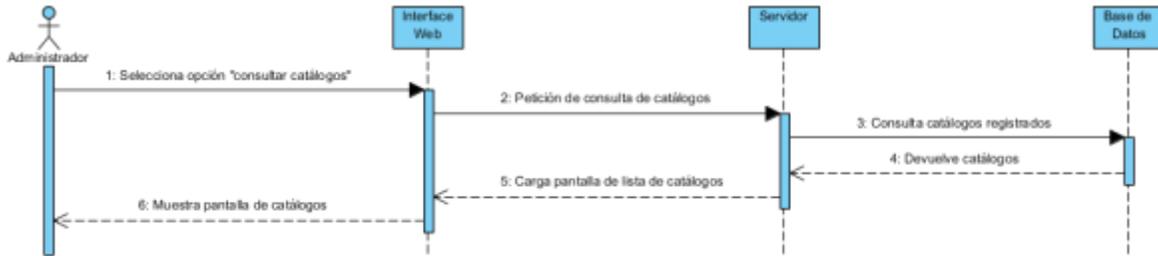


Figura 36 Diagrama de secuencia - Consultar Catálogo

#### 4.2.1.9 Modificar Catálogo

El catálogo seleccionado por el administrador es cargado en la interfaz que le permite modificar los tipos de dato dentro del mismo, el sistema validará los cambios realizados y notificará si los cambios pudieron o no ser realizados.

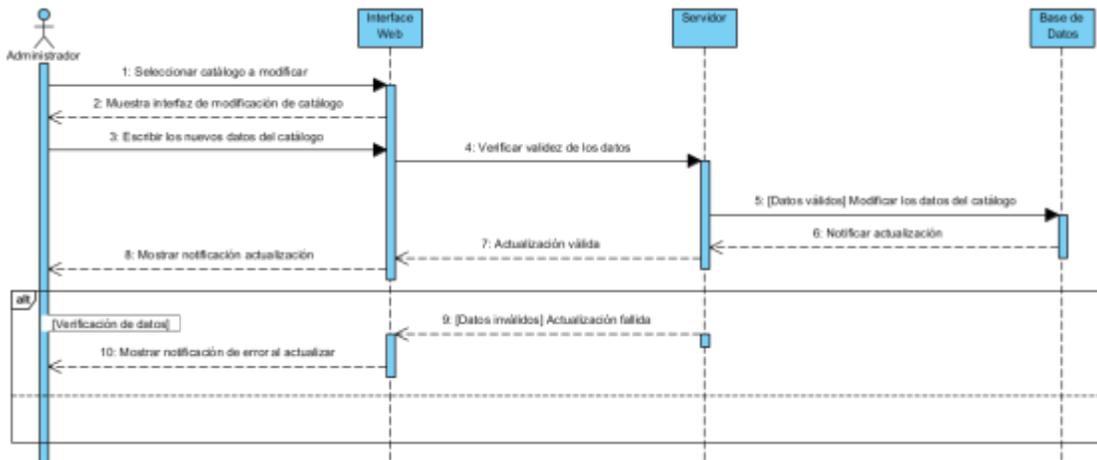


Figura 37 Diagrama de secuencia - Modificar Catálogo

#### 4.2.1.10 Borrar Catálogo

El administrador o capturista seleccionan el catálogo a eliminar, se presentará una advertencia verificando si es realmente lo que quiere realizar. El usuario podrá cancelar esta petición o continuar.

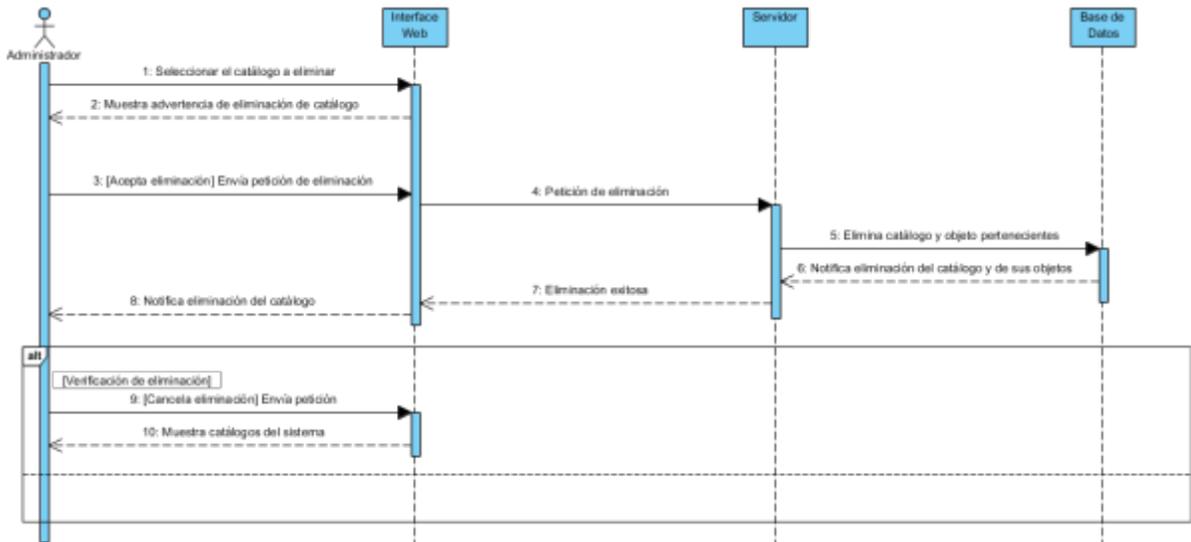


Figura 38 Diagrama de secuencia - Borrar Catálogo

#### 4.2.1.11 Registrar Objeto

El usuario selecciona los datos para el objeto con base en los catálogos disponibles. El sistema validará estos datos para así registrar el nuevo objeto dentro del catálogo correspondiente. De otro modo se notificará al usuario del error.

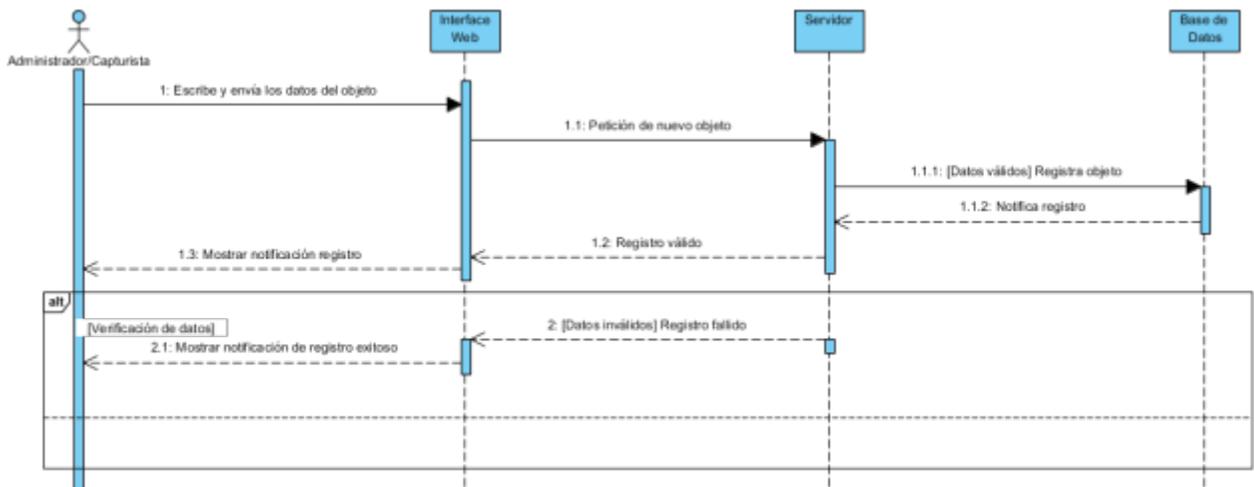


Figura 39 Diagrama de secuencia - Registrar Objeto

#### 4.2.1.12 Consultar Objetos

Se selecciona la opción de consulta de objetos, en la que se cargará una pantalla con todos los objetos correspondientes a la selección del usuario.

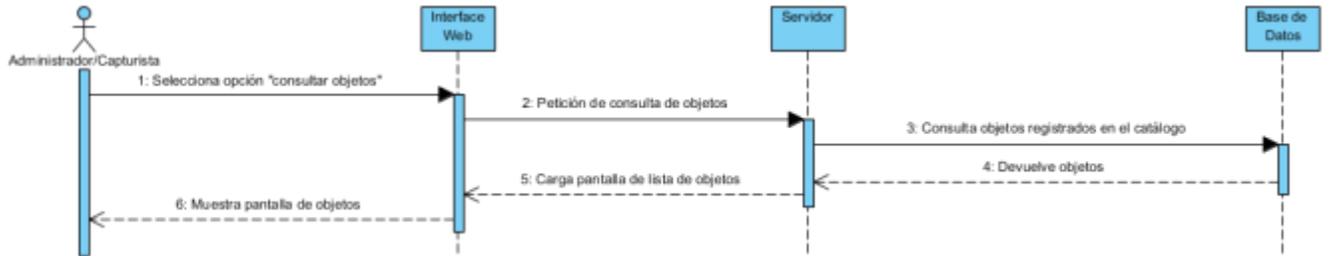


Figura 40 Diagrama de secuencia - Consultar Objetos

#### 4.2.1.13 Modificar Objeto

El administrador y capturista modificarán alguno de los objetos que elija dentro de los presentes en la lista de objetos disponibles. El sistema verificará la validez de estos cambios y se notificará al usuario de ello.

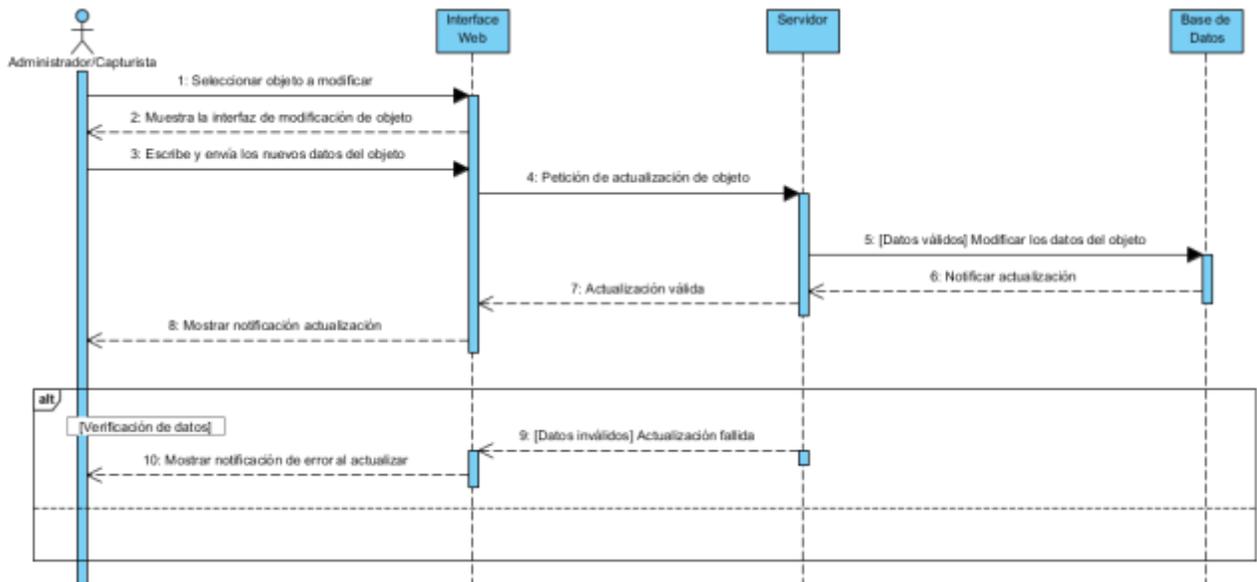


Figura 41 Diagrama de secuencia - Modificar Objeto

#### 4.2.1.14 Borrar Objeto

El administrador o capturista seleccionan el objeto a eliminar, se presentará una advertencia verificando si es realmente lo que quiere realizar. El usuario podrá cancelar esta petición o continuar.

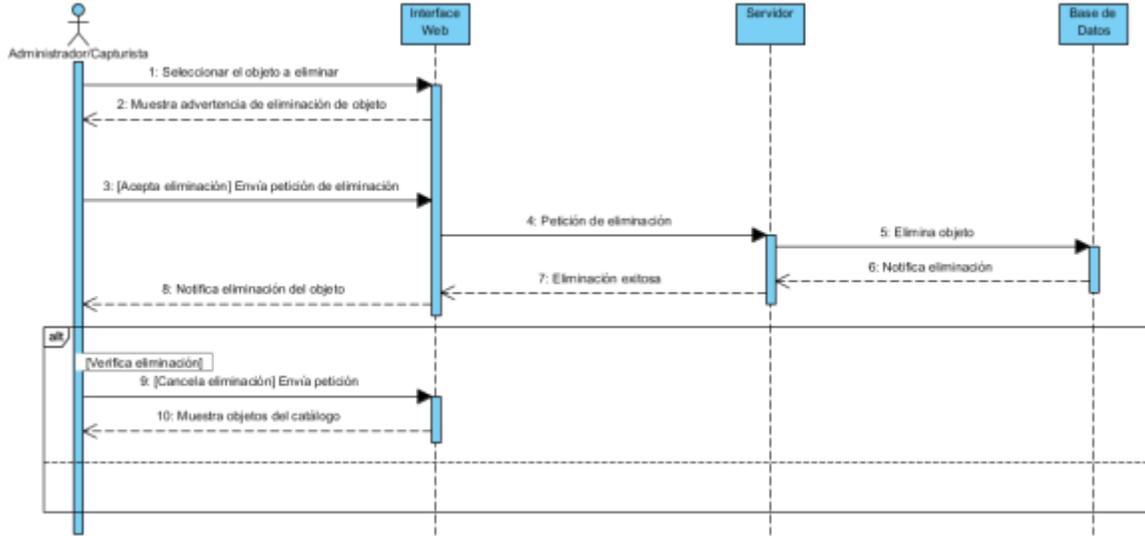


Figura 42 Diagrama de secuencia - Borrar Objeto

#### 4.2.1.15 Generar código QR

La generación de los códigos se lleva a cabo por el administrador o el capturista, se obtiene la información correspondiente respecto a los recursos del objeto y se genera el código QR respectivo.

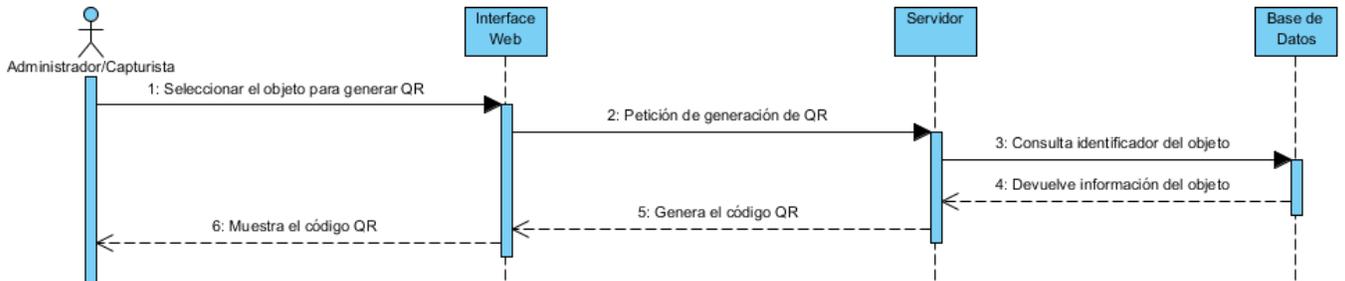


Figura 43 Diagrama de secuencia - Generar código QR

### 4.2.1.17 Cambio de contraseña

El administrador o capturista solicitan el cambio de contraseña.

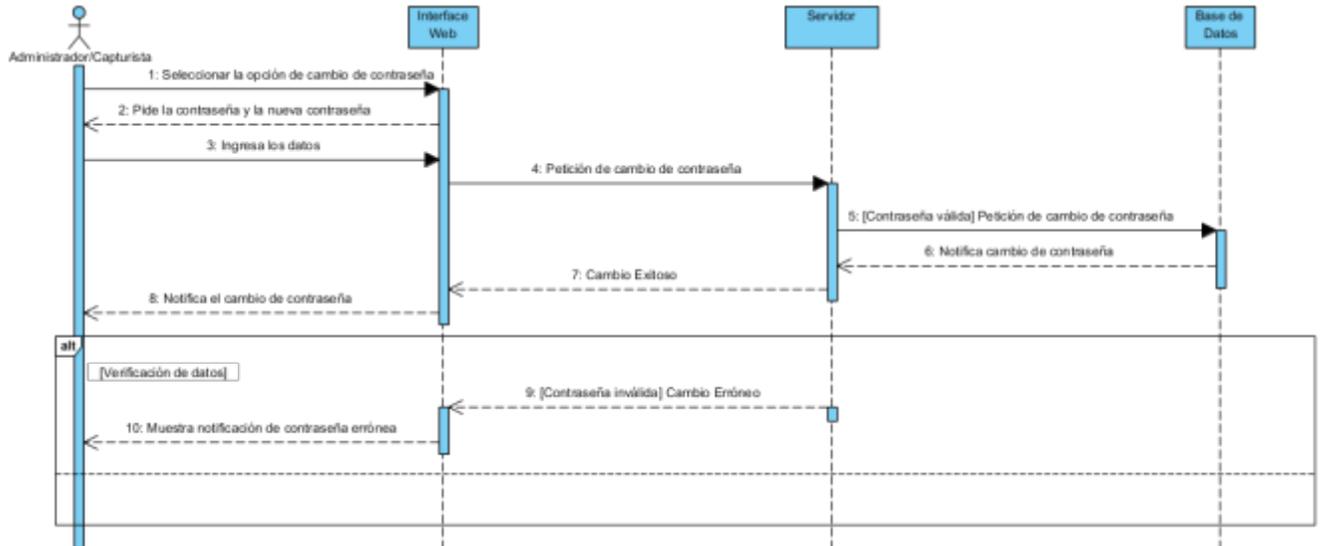


Figura 44 Diagrama de secuencia - Generar código QR

### 4.2.1.18 Recuperación de contraseña

El administrador o capturista solicitan la recuperación de contraseña.

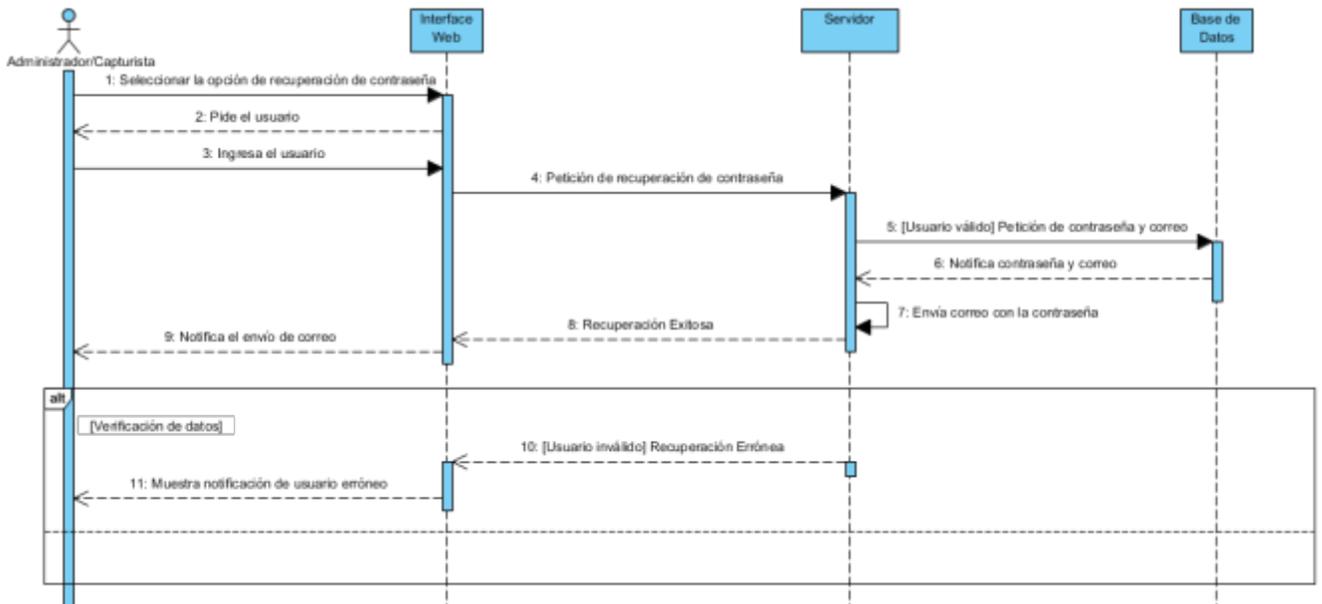


Figura 45 Diagrama de secuencia - Generar código QR

## 4.2.2 Dispositivo Móvil

A continuación se pueden ver los diagramas de secuencia correspondientes al sistema en el Dispositivo Móvil.

### 4.2.2.1 Iniciar Sesión

El usuario iniciará sesión proporcionando su correo dentro de la aplicación móvil, el sistema verifica los datos e inicia la sesión, en caso de no ser válidos se le notificará.

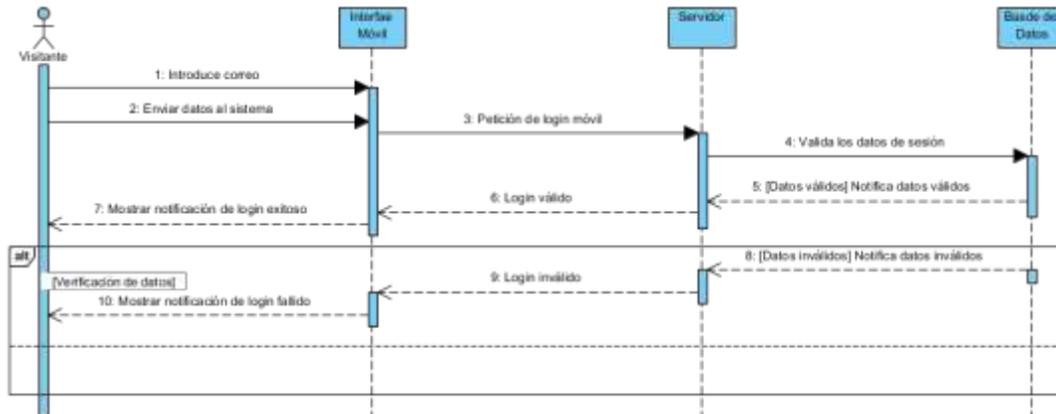


Figura 46 Diagrama de secuencia - Iniciar Sesión en Dispositivo Móvil

### 4.2.2.2 Cerrar Sesión

El visitante elige la opción de cierre de sesión en el sistema, confirmará que al cerrar la sesión su información se eliminará. El editor eliminar la información y se cierra la sesión en el servidor.

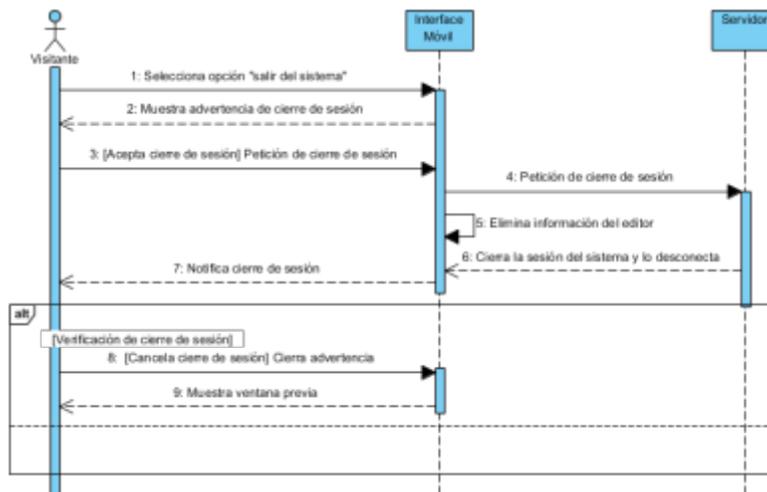


Figura 47 Diagrama de secuencia - Cerrar Sesión en Dispositivo Móvil

### 4.2.2.3 Escanear código QR

Empleando su dispositivo móvil, el visitante escaneará el código, la aplicación descifra la URL del recurso, se accede a la información y se carga en el editor. En caso de no poder leerlo se notifica al usuario.

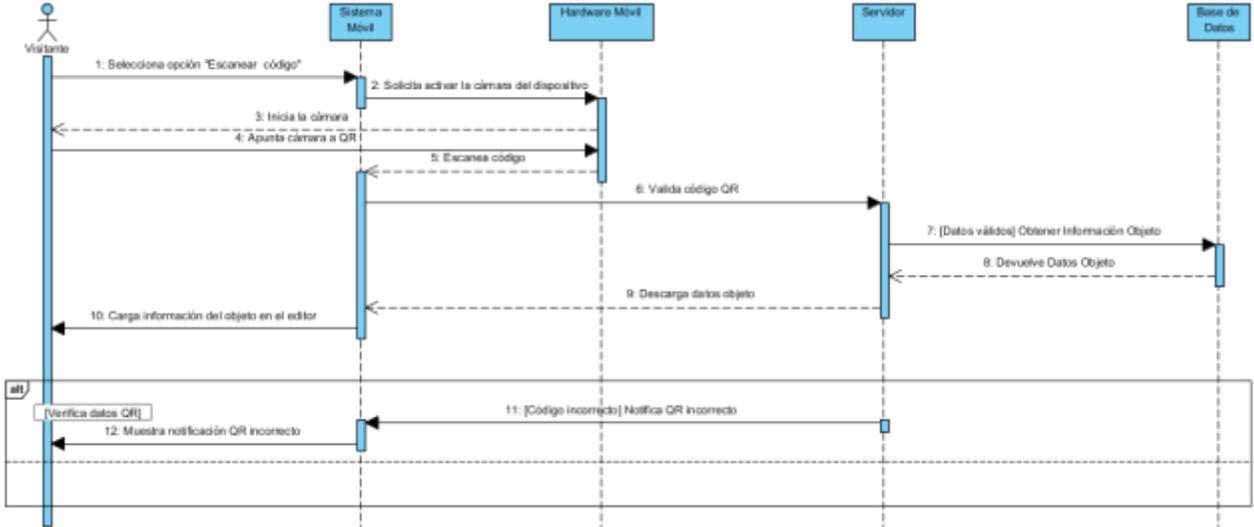


Figura 48 Diagrama de secuencia - Escanear código QR

### 4.2.2.4 Modificar Datos en el Editor

El usuario podrá manipular la información que se ha cargado en el editor, ya sea la selección de texto, el eliminar imágenes o elegir sólo datos que le interesen.

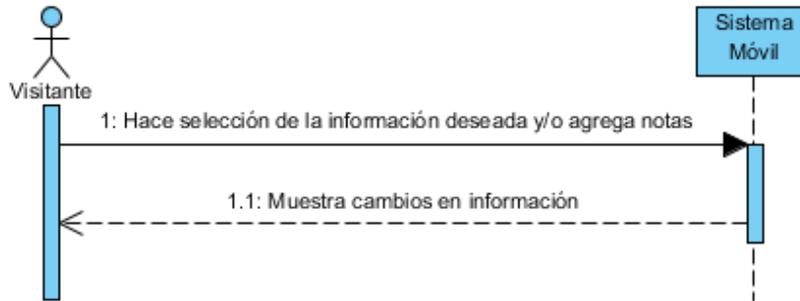


Figura 49 Diagrama de secuencia - Modificar Datos en el Editor

#### 4.2.2.5 Enviar Información a Correo

El visitante elije la opción de envío de la información, la aplicación notificará que su información será enviada y borrada del dispositivo. Se notificará si la información se pudo o no enviar correctamente.

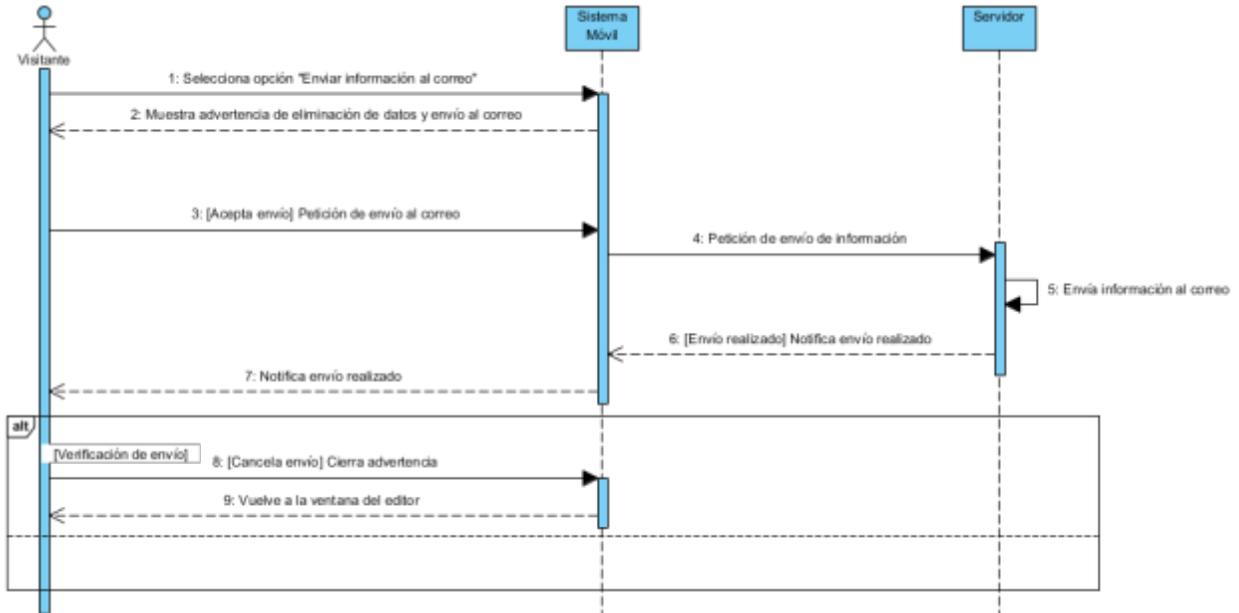


Figura 50 Diagrama de secuencia - Enviar Información a Correo

### 4.3 Diseño de la base de datos

En esta sección se muestra el diagrama Entidad-Relación de nuestra base de datos que se implementará en el sistema. Un diagrama de Entidad-Relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

Además, en el Anexo I se incluye el Diccionario de Datos de la base de datos.

#### 4.3.1 Diagrama Entidad-Relación

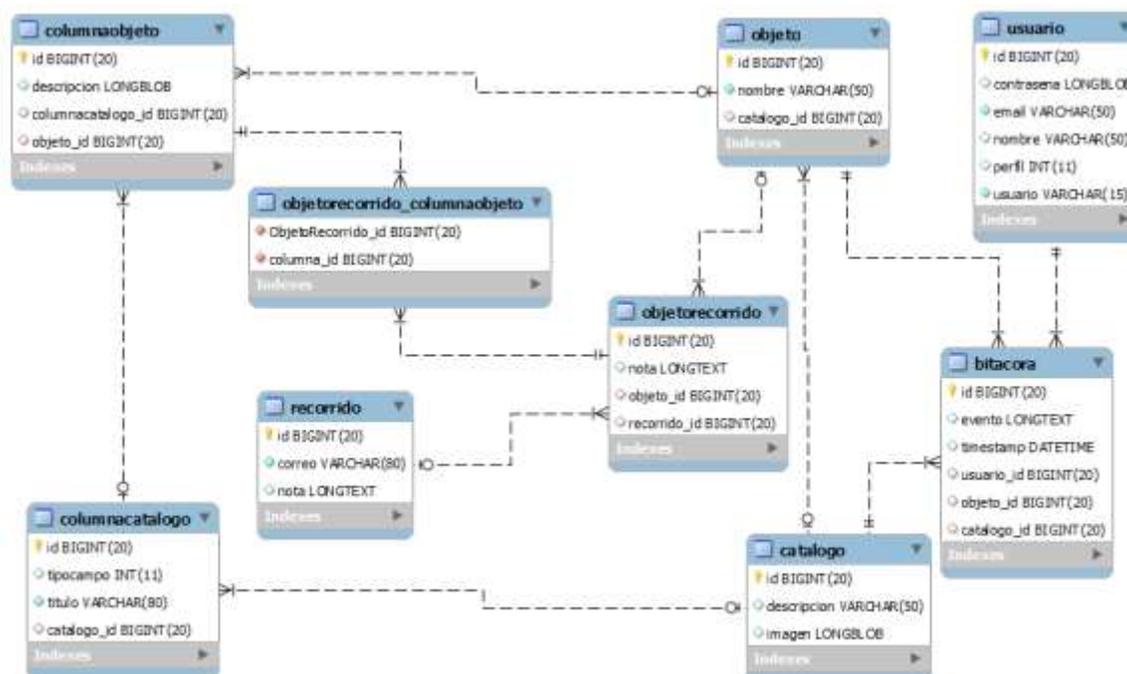


Figura 51 Diagrama Entidad – Relación

En el diagrama anterior, se muestra el diseño de la base de datos con la cual trabajará el sistema desarrollado. Está normalizada en 3FN aunque se realizaron unas conexiones extras entre las tablas para que las consultas fueran más optimizadas para la parte de Hibernate.

En dicho diagrama, podemos ver 9 tablas, cada una con sus propios atributos y relaciones; lo más destacable es la parte de catálogo que tiene 2 relaciones y será la base del sistema, sin esas relaciones todo el sistema estaría mal diseñado.

#### 4.4 Diagrama de clases

A continuación se muestra una vista del diagrama de clases del sistema a nivel de paquetes, las cuales son 4: Web, Service, Model y Exceptions. (Para verlo más a detalle ir a las subsecciones 4.5.1, 4.5.2, 4.5.3 y 4.5.4). Si se desea ver el diagrama de clases general del sistema con todas las relaciones, dentro del CD Anexo en la carpeta “Diagramas UML/ Clases” hay un archivo llamado “mx.ipn.pisinc.png”.

Se está usando el patrón de diseño de MVC en Spring, donde en la parte de Web (Izquierda) están los controladores; en la parte de Service (Centro Superior), están ubicados los servicios que comunican la base de datos con las vistas; y por último, en la parte de Model (Derecha) se ubica el modelado de la base de datos en entidades de JPA y los Data Transfer Object (DTO) que nos servirán para transportar los datos de la base de datos a través de las 3 capas principales. La última capa, de Exceptions (Centro Inferior), nos ayuda a manejar los posibles errores que pudieran ocurrir en la aplicación.

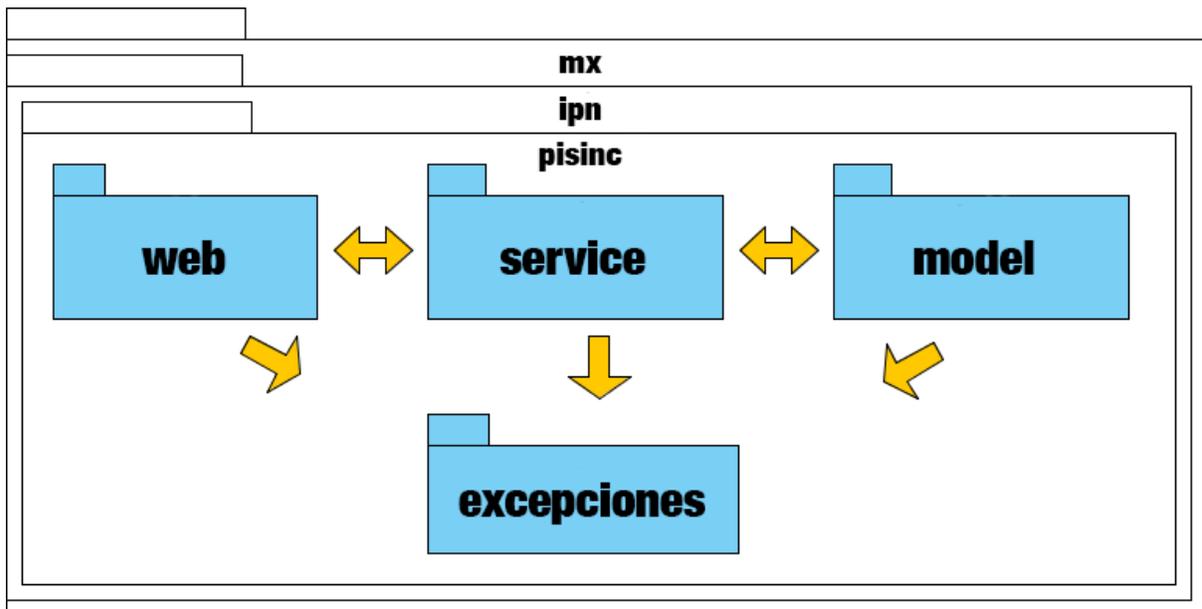


Figura 52 Diagrama de clases a nivel de paquetes

#### 4.4.1 Diagrama de clases Web

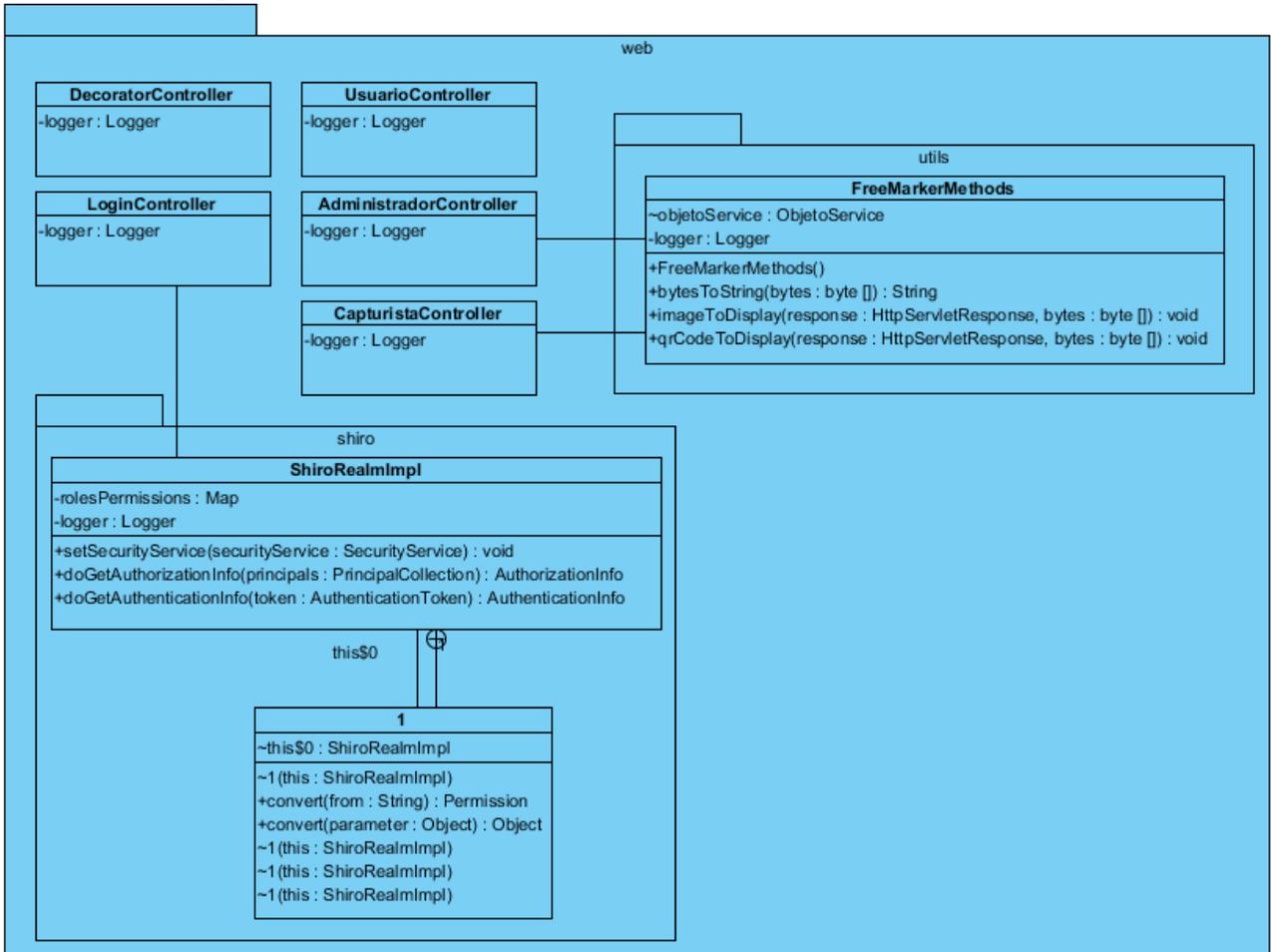


Figura 53 Diagrama de clases Web

En la imagen anterior podemos ver la capa de Web, donde tenemos los controladores de las vistas del sistema y en la parte inferior observamos un paquete llamado Shiro, el cual es el encargado de la seguridad del sitio.

## 4.4.2 Diagrama de clases Service

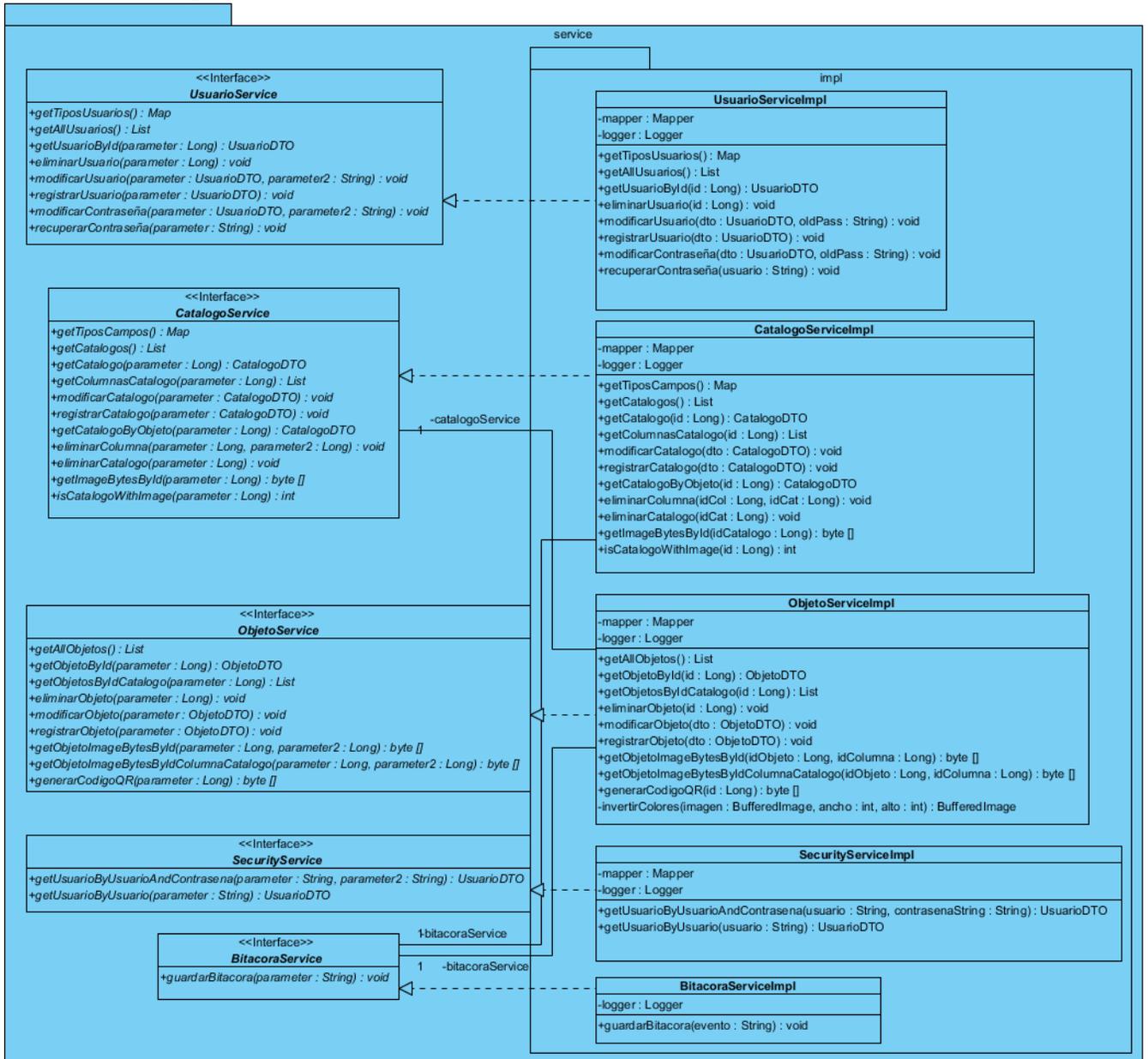


Figura 54 Diagrama de clases Service

En el diagrama antes mostrado, observamos la capa de Service la cual consta de una interface por cada clase de Servicio dentro de la aplicación.

Esta capa es la encargada de las funciones y transacciones de las vistas con la base de datos, aquí se llevan a cabo todas las operaciones del servidor tales como las consultas, modificación y/o eliminación de datos.

Además, hay un paquete que contiene algunas utilidades funcionales del sistema como es el envío de correos electrónicos.

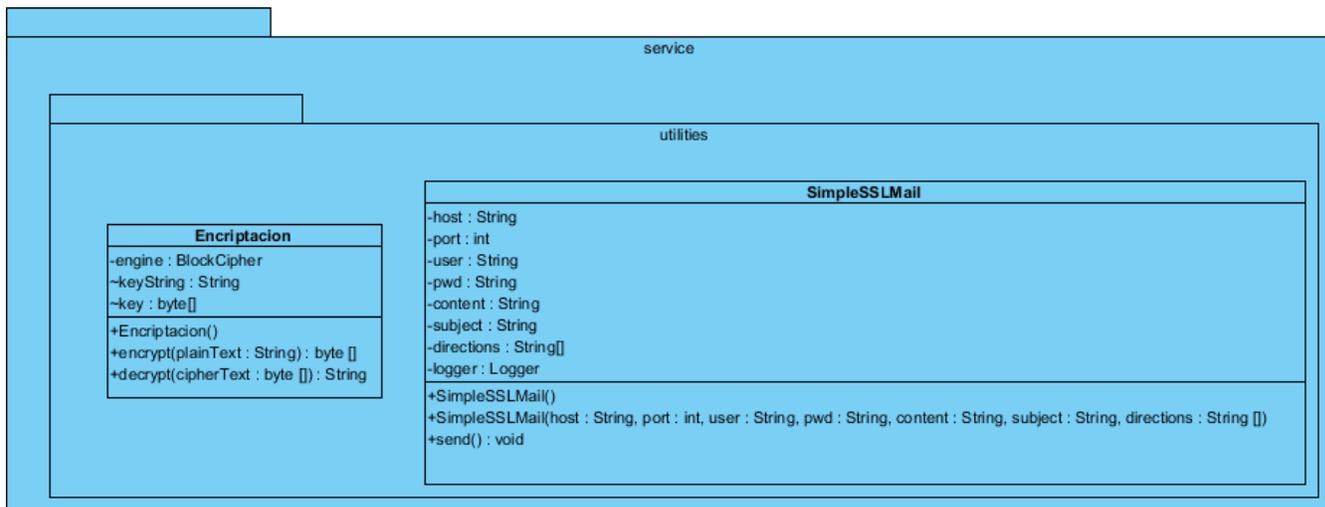


Figura 55 Diagrama de clases Service -utilities

### 4.4.3 Diagrama de clases Model

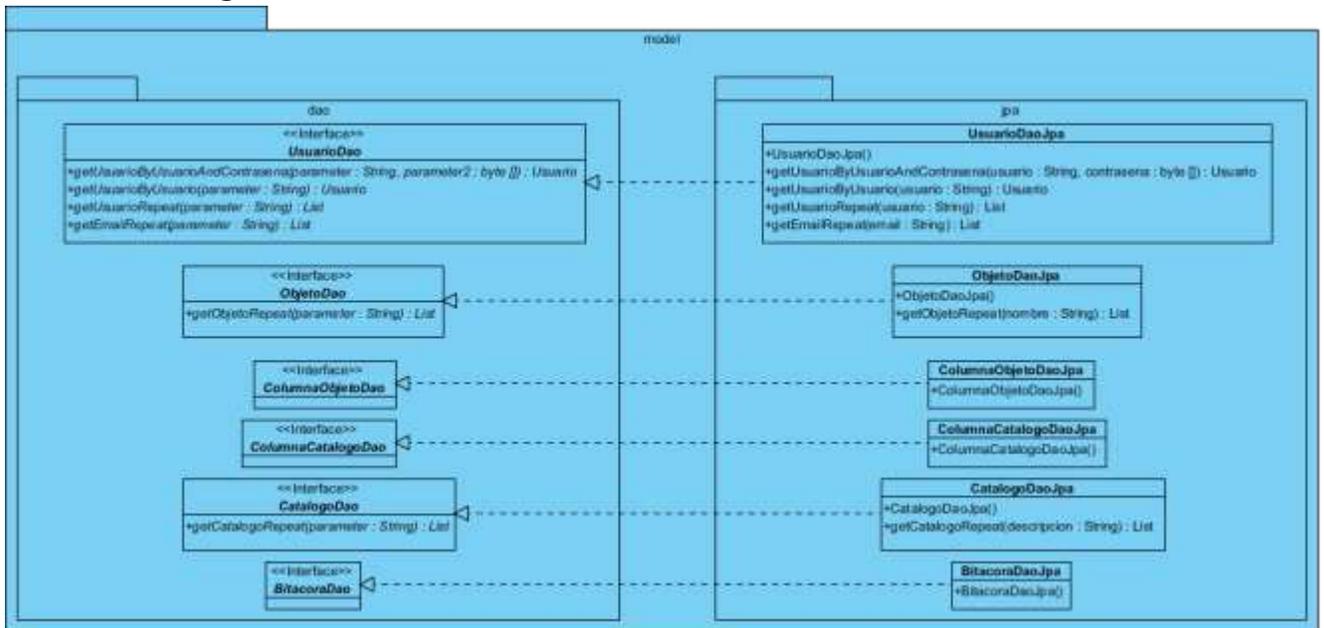


Figura 56 Diagrama de clases Model – dao y jpa

En la imagen anterior, está reflejada 2 subpaquetes de la capa de Model del sistema: dao y jpa.

El dao, nos refleja las interfaces a utilizar por el paquete jpa y nos ayudan a mantener la persistencia de los datos en la base de datos a través de Hibernate/JPA.

El dto, como se mencionó anteriormente, sirve de transporte de los datos de la base de datos por todas las capas del sistema.

En las siguientes 2 imágenes se muestran el subpaquete domain y dto, en el primero están contenidas las entidades de la base de datos dando transaccionabilidad con el motor de base de datos a través de JTA y en el segundo el medio de transporte de los datos a través de todas las capas del sistema.

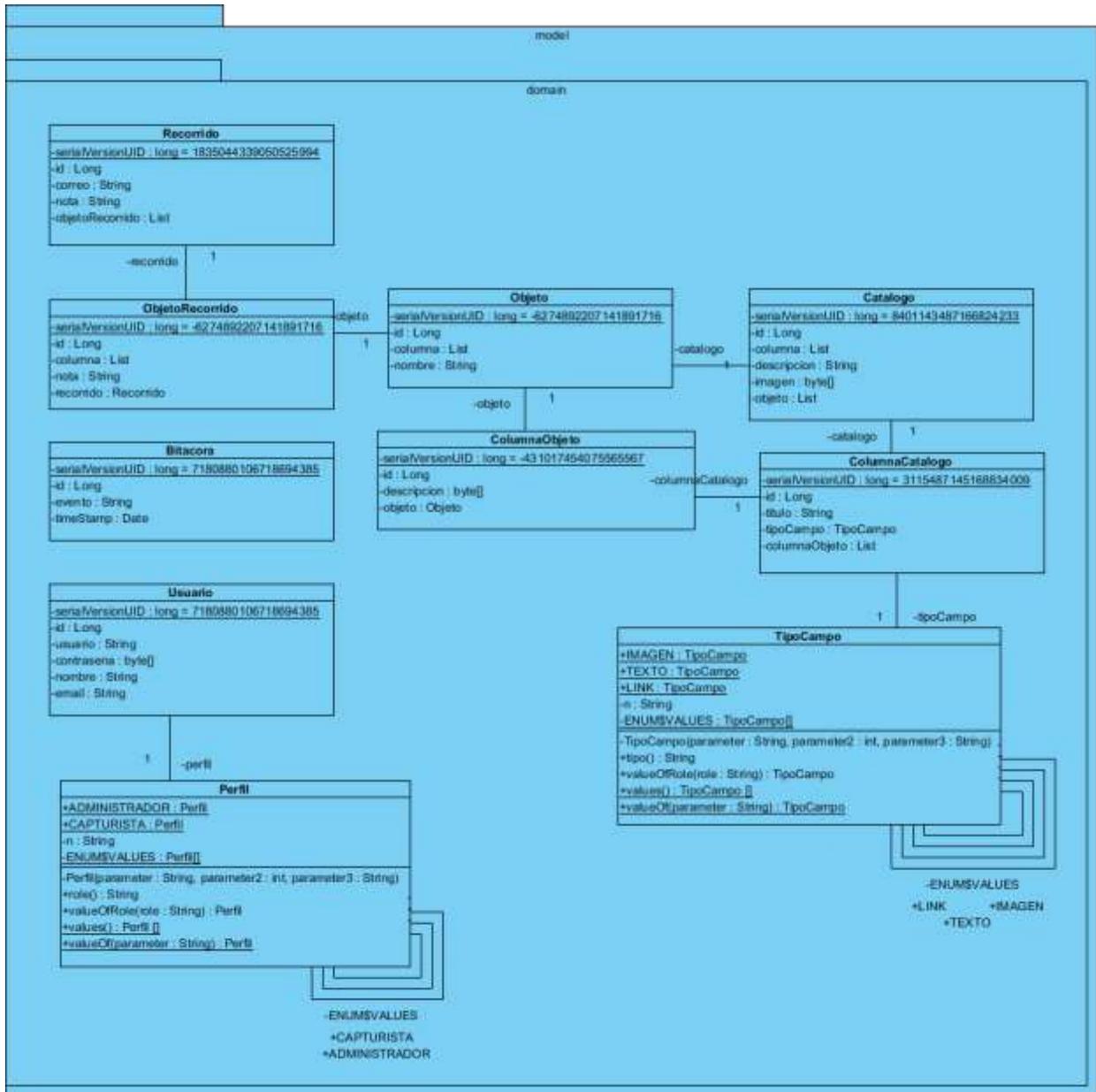


Figura 57 Diagrama de clases Model - domain

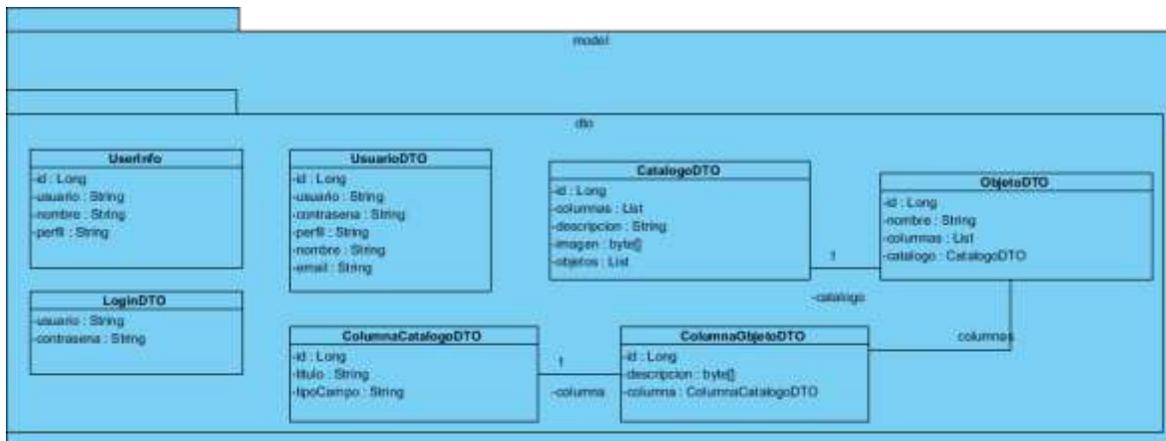


Figura 58 Diagrama de clases Model - dto

#### 4.4.4 Diagrama de clases Exceptions

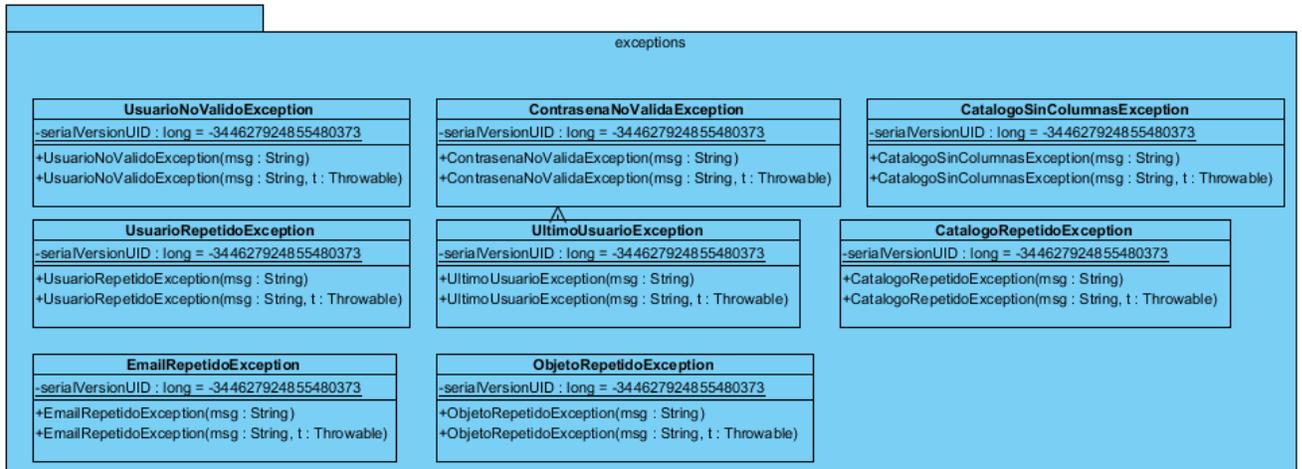


Figura 59 Diagrama de clases Exceptions

Esta última capa es la encargada de las validaciones y prevenciones de errores del sistema.

## CAPÍTULO 5 DESARROLLO

Esta sección trata de describir la parte funcional más importante del sistema, mostrando como se desarrolló y el flujo de estos.

### 5.1 Sitio web

En esta sección se describirá la manera en cómo fue implementado el sistema web, habiendo hecho uso del framework de Spring principalmente y combinándolo con otros como *Shiro*, *Hibernate*, JPA y JTA para la mejor funcionalidad del mismo.

Además, se hizo uso del framework OSGi para modularizar el sistema y así no depender de que todas las capas funcionen correctamente. Es muy importante mencionar que la implementación de todas las interfaces del sistema se hicieron con la biblioteca de *Blueprint*, donde en un archivo de configuración de XML se asigna la *interface* y la clase que la implementará. Por ejemplo, para implementar la interface de *UsuarioDao* se hizo uso de las siguientes líneas:

```
<service ref="usuarioDao" interface="mx.ipn.pisinc.model.dao.UsuarioDao"/>
<bean id="usuarioDao" class="mx.ipn.pisinc.model.jpa.UsuarioDaoJpa">
    <tx:transaction method="*" value="Required" />
    <jpa:context property="entityManager" unitname="pisincUnit" />
</bean>
```

#### 5.1.1 Model

Primeramente cabe destacar que todas las tablas se mapearon con *Hibernate/JPA* y *JTA*, este último nos ayuda a no depender de un script SQL para instalar el sistema en cualquier equipo, basta con ejecutar el sistema para que la base de datos se genere a partir de las clases.

##### 5.1.1.1 Persistencia

El archivo de configuración de persistencia, que se encuentra ubicado en la capa *Model* dentro de *META-INF*, se describe a grandes rasgos a continuación.

En la siguiente línea se hace llamada del pool de conexiones configurado previamente en el servidor de aplicaciones:

```
<jta-data-source>jdbc/pisinc</jta-data-source>
```

Todas las tablas de la base de datos se mapean con el siguiente bloque de código:

```
<class>mx.ipn.pisinc.model.domain.Catalogo</class>
<class>mx.ipn.pisinc.model.domain.Objeto</class>
<class>mx.ipn.pisinc.model.domain.Usuario</class>
<class>mx.ipn.pisinc.model.domain.ColumnaCatalogo</class>
```

```
<class>mx.ipn.pisinc.model.domain.ColumnaObjeto</class>
<class>mx.ipn.pisinc.model.domain.Recorrido</class>
<class>mx.ipn.pisinc.model.domain.ObjetoRecorrido</class>
<class>mx.ipn.pisinc.model.domain.Bitacora</class>
```

La siguiente línea nos ayuda a que si en alguna transacción en la base de datos falla, todos los cambios se reviertan automáticamente:

```
<property name="javax.persistence.validation.mode" value="callback, ddl"/>
```

Por último, para hacer el correcto uso de JTA y poder aplicar la funcionalidad mencionada anteriormente se hace uso de la siguiente línea de código:

```
<property name="hibernate.transaction.jta.platform" value="com.justcloud.aries.transaction.AriesPlatform"/>
```

### 5.1.1.2 Domain

Cada entidad perteneciente a las tablas de la base de datos, se describe con anotaciones. Todas se inician con la anotación *Entity* que nos indica que es una tabla y prosigue con *NamedQueries* que describe las consultas especiales que se harán en el sistema, por ejemplo:

```
@Entity
@NamedQueries({
    @NamedQuery(name = "Catalogo.findByDescripcion", query = "SELECT c FROM Catalogo c WHERE c.descripcion = :descripcion")
})
```

Además todas las entidades se generaron con un identificador de la siguiente forma:

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Override
```

### 5.1.1.3 Modelo

Para cada entidad se creó una *interface DAO* con su respectiva implementación para hacer uso de transacciones hacia la base de datos. Por ejemplo, para obtener todos los catálogos con una descripción se hace uso de las siguientes líneas de código:

```
public List<Catalogo> getCatalogoRepeat(String descripcion) {
    return entityManager.createNamedQuery("Catalogo.findByDescripcion",Catalogo.class)
        .setParameter("descripcion", descripcion)
        .getResultList();
}
```

Donde en *createNamedQuery*, se llama a la sentencia SQL creada en la entidad dada asignándoles atributos y/o parámetros.

### 5.1.2 Service

En esta capa se describen los servicios que interactuarán entre la vista del sistema y la capa de *Model*, que hace transacciones hacia la base de datos. Así mismo, se controla mediante la capa de *Exceptions*, los errores que puedan existir en el sistema.

Además, dentro hay un paquete llamado DTO que lo único que hace es transportar los datos de la base de datos hacia la vista del sistema.

#### 5.1.2.1 DTO

Estas clases son simples clases que contienen las mismas variables de las entidades pero sin anotaciones, con sus respectivos *getters* y *setters*.

#### 5.1.2.2 Servicios

Todos los servicios se implementan de forma parecida a los *Model* de la capa *Domain*, es decir hay una *interface* para cada servicio y a través de un archivo de configuración *Blueprint*, se implementan en otra clase.

Estos servicios hacen uso de las interfaces de *Domain* para acceder a los datos de la base de datos, por ejemplo para la consulta de todos los catálogos del sistema se ocupó el siguiente método:

```
public List<CatalogoDTO> getCatalogos() {
    logger.debug("Consultando Catalogos...");
    List<Catalogo> catalogo = catalogoDao.getAll();
    List<CatalogoDTO> listaDTO = new ArrayList<CatalogoDTO>();
    for(Catalogo cat : catalogo){
        CatalogoDTO dto = mapper.map(cat, CatalogoDTO.class);
        dto.setColumnas(new ArrayList<ColumnaCatalogoDTO>());
        for(ColumnaCatalogo columna : cat.getColumna()){
            ColumnaCatalogoDTO columnaDto = new ColumnaCatalogoDTO();
            columnaDto.setId(columna.getId());
            columnaDto.setTitulo(columna.getTitulo());
            columnaDto.setTipoCampo(columna.getTipoCampo().name());
            dto.getColumnas().add(columnaDto);
        }
        dto.setObjetos(new ArrayList<ObjetoDTO>());
        for(Objeto objeto : cat.getObjeto()){
            ObjetoDTO objetoDto = mapper.map(objeto, ObjetoDTO.class);
            dto.getObjetos().add(objetoDto);
        }
        listaDTO.add(dto);
    }
    return listaDTO;
}
```

Como se puede observar, la variable *catalogoDao* tiene un método llamado *getAll*, éste nos da todos los catálogos del sistema y es un método autoimplementado por JTA, que es otra de las razones por la que lo utilizamos, ya que nos genera algunos métodos por defecto y así evitarnos hacer código extra. Posteriormente todos los catálogos se pasan a una lista de DTO que es la que se enviará a la capa de Web.

### 5.1.3 Web

Esta capa es la de más alto nivel y en ella se describen las vistas del sistema mediante archivos FTL, que son jsp pero implementados en el framework de *FreeMarkers*.

*FreeMarkers* nos ayudó a definir las vistas del sistema en un archivo llamado *decorator*, y así evitarnos estar implementando el estilo de la página en todos los FTL que hagamos. Todo esto se configuró con lo siguiente:

```
<decorators defaultdir="/decorators">
  <excludes>
    <pattern>* .json*</pattern>
    <pattern>/decorators</pattern>
    <pattern>/public/javascripts/generated/validationLegends.js</pattern>
  </excludes>
  <decorator name="master" page="main.ftl">
    <pattern>/*</pattern>
  </decorator>
</decorators>
```

Ahí se puede ver, cual es el archivo que contiene la plantilla del sistema (main.ftl) y que archivos excluirémos de la plantilla.

También, es importante mencionar que *FreeMarkers* nos ayuda a separar el código java de la vista implementando etiquetas tales como: *<#IF>*, *<#ELSE>*, etc. y accediendo a las variables mediante *#{nombre\_variable}*. Por ejemplo, en el sistema para identificar si un objeto se le agregará un texto o una imagen, se utilizó lo siguiente:

```
<#if columna.columna.tipoCampo == "TEXTO" >
  <textarea rows="3" class="span6 required" id='columnas[#{columna_index}].descripcion'
name='columnas[#{columna_index}].descripcion'>${util.bytesToString(columna.descripcion)}</textarea>

  <#elseif columna.columna.tipoCampo == "LINK" >
  <input class="span6 required url" type='text' id='columnas[#{columna_index}].descripcion'
name='columnas[#{columna_index}].descripcion' value='${util.bytesToString(columna.descripcion)}' />
  <#else>
    <span class="ie_show">
      <input type="file" name="columnas[#{columna_index}].descripcion"
id="columnas[#{columna_index}].descripcion" onchange='${"#input-
file${columna_index}").val(reemplazarFakePath($(this).val());" />
    </span>
  <span class="ie_hide">
```

```

        <div class="input-append">
            <input id="input-file${columna_index}" class="input-large" type="text"
onclick="$('input[id=columnas\\[${columna_index}\\]\\.descripcion]').click();" style="cursor: pointer;' readonly>
                <a class="btn"
onclick="$('input[id=columnas\\[${columna_index}\\]\\.descripcion]').click();"><i class="icon-picture"></i></a>
            </div>
        </span>
        <a class="imagen" href="{rc.contextPath}/capturista/imagenObjeto/${objeto.id}/${columna.id}"></a>
    </#if>

```

Además, es la única capa que no hace uso de *Blueprint*, sino que utiliza un bloque de código nativo de Spring para hacer las inyecciones de los servicios y poderlos utilizar en cualquier parte del sistema; todo esto se hace con el bloque de código siguiente:

```

<osgi:reference id="usuarioService" interface="mx.ipn.pisinc.service.UsuarioService" timeout="10000" />
<osgi:reference id="catalogoService" interface="mx.ipn.pisinc.service.CatalogoService" timeout="10000" />
<osgi:reference id="objetoService" interface="mx.ipn.pisinc.service.ObjetoService" timeout="10000" />
<osgi:reference id="securityService" interface="mx.ipn.pisinc.service.SecurityService" timeout="10000" />

```

En las líneas anteriores, se hace llamada de los servicios y se le da un tiempo de espera máximo.

Un punto importante a mencionar, es que se usó una biblioteca de *JavaScript* llamada *HandleBars* para hacer dinámico el registro de catálogos, para usarla primero se tuvo que crear un template de la siguiente forma:

```

<script id="item_catalogo" type="text/x-handlebars-template">
    <tr id="fila{{id}}">
        <td>
            <input type="text" class="required span4" id="columnas[{{id}}].titulo"
name="columnas[{{id}}].titulo" maxlength="50" />
        </td>
        <td>
            <select id="columnas[{{id}}].tipoCampo" name="columnas[{{id}}].tipoCampo"
class="span4 required">
                <option value="">Seleccione un Tipo de Campo</option>
                <#list tiposcampos?keys as tipo>
                <option value="{{tipo}}">{{tiposcampos[tipo]}</option>
                </#list>
            </select>
        </td>
        <td style="text-align:center;">
            <button type="button" class="btn-danger btn" id="botonFila{{id}}">
onclick="remove_item({{id}});"><i class="icon-remove"></i></button>
        </td>
    </tr>
</script>

```

Y posteriormente, se implementó en el siguiente script:

```
<script>
  var item_num=1;
  template = Handlebars.compile($("#item_catalogo").html());
  function add_catalogo(){
    $("#catalogosTable").append(template({id:item_num}));
    item_num++;
  }
  function remove_item(id){
    $("#fila"+id).remove();
    if( id < item_num-1){
      for( var i=id; i<item_num; i++){
        $("#fila"+(i+1)).attr("id","fila"+i);

        $("#columnas\\"+(i+1)+"\\".tipoCampo").attr("id","columnas\\"+(i)+"\\".tipoCampo");
        $("#columnas\\"+(i+1)+"\\".titulo").attr("id","columnas\\"+(i)+"\\".titulo");

        $("#columnas\\"+(i)+"\\".tipoCampo").attr("name","columnas\\"+(i)+"\\".tipoCampo");
        $("#columnas\\"+(i)+"\\".titulo").attr("name","columnas\\"+(i)+"\\".titulo");
        $("#botonFila"+(i+1)).attr("onclick","remove_item("+i+")");
        $("#botonFila"+(i+1)).attr("id","botonFila"+i);
      }
    }
    item_num--;
  }
</script>
```

### 5.1.3.1 Shiro

En esta misma capa, se implementó el framework de *Shiro* para darle seguridad a todo el sitio web, esto se llevó a cabo en 2 partes: una configuración XML y funciones en una clase de java.

La primera parte se desarrolló en el archivo spring-filter.xml, agregando las siguientes líneas de código:

```
<bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">
  <property name="securityManager" ref="securityManager" />
  <property name="loginUrl" value="/"/>
  <property name="filterChainDefinitions">
    <value>
      /* = anon
      /assets/* = anon
      /images/* = anon
      /administrador/* = authc,perms[vista:administrador],perms[vista:capturista]
      /capturista/* = authc,perms[vista:capturista]
      /usuario/* = authc,perms[vista:capturista]
      /errores/* = authc,perms[vista:capturista]
    </value>
  </property>
</bean>
```

```

                </value>
            </property>
        </bean>
        <bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">
            <property name="realm" ref="pisincRealm" />
        </bean>
        <bean id="pisincRealm" class="mx.ipn.pisinc.web.shiro.ShiroRealmImpl">
            <property name="securityService" ref="securityService" />
            <property name="rolesPermissions">
                <map>
                    <entry key="ADMINISTRADOR">
                        <list>
                            <value>vista:administrador</value>
                            <value>vista:capturista</value>
                        </list>
                    </entry>
                    <entry key="CAPTURISTA">
                        <list>
                            <value>vista:capturista</value>
                        </list>
                    </entry>
                </map>
            </property>
        </bean>

```

En la segunda parte, se creó la clase *ShiroRealmImpl* para describir la seguridad del sitio, esta parte es la más complicada ya que se maneja información muy sensible y si hay un solo error, no funcionará ninguna capa del sistema.

El método más importante y el que lleva a cabo el inicio de sesión se describe a continuación:

```

public AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principals) {
    logger.debug("Verificando autorización");
    Set<String> roles = new HashSet<String>();
    Set<Permission> permissions = new HashSet<Permission>();
    @SuppressWarnings("unchecked")
    List<UserInfo> principalsList = principals.asList();
    if (principalsList.isEmpty()) {
        logger.warn("Lista Principales Vacía");
        throw new AuthorizationException("Lista Principales Vacía");
    }
    // Agregando los permisos
    for (UserInfo userPrincipal : principalsList) {
        logger.debug("Loading stuff for principal {}", userPrincipal.getUsuario());
        UsuarioDTO usuarioDTO = securityService.getUsuarioByUsuario(userPrincipal.getUsuario());
        String role = usuarioDTO.getPerfil();
        logger.debug("Rol de " + userPrincipal.getUsuario() + " es " + role);

        if(role.equals("ADMINISTRADOR"))

```

```

        roles.add("CAPTURISTA");
    roles.add(role);

    for (String r : roles) {
        if (rolesPermissions.containsKey(r)) {
            permissions.addAll(convert(rolesPermissions.get(r),
                new Converter<String, Permission>() {

                    @Override
                    public Permission convert(String from) {
                        return new WildcardPermission(from);
                    }

                }
            ));
        }
    }

    SimpleAuthorizationInfo info = new SimpleAuthorizationInfo(roles);
    info.setRoles(roles); // fill in roles
    info.setObjectPermissions(permissions);

    return info;
}

```

### 5.1.3.2 Controller

Una parte importante de las vistas en Spring son las clases llamadas *Controller*, las cuales nos describen que nombre le daremos a nuestros archivos FTL en la web y que archivo es el que se ejecutará mediante la anotación *RequestMapping*, además de que podemos mandarle variables a través de la clase *ModelMap* y recibir parámetros con la anotación *PathVariable*.

Por ejemplo, en *AdministradorController*, se describen todas las pantallas a las que tendrá acceso el usuario administrador, uno de las pantallas se muestra a continuación:

```

@RequestMapping(value = "/home")
public String home(ModelMap model) {
    logger.debug("Entrando a la pantalla principal del administrador...");
    UserInfo userInfo = (UserInfo) SecurityUtils.getSubject().getPrincipal();
    model.addAttribute("nombre_usuario", userInfo.getNombre());
    return "/administrador/home";
}

```

Primero se le da el nombre a la página con *model.addAttribute* se agrega una variable en el archivo “/administrador/home.ftl”.

Otro método donde se puede ver los datos recibidos de un formulario es el siguiente:

```

@RequestMapping(value = "/usuario/{idUsuario}")

```

```

public String consultarUsuario(ModelMap model,
    @PathVariable("idUsuario") Long idUsuario) {
    logger.debug("Entrando a la consulta del usuario con ID: "+idUsuario);
    model.addAttribute("usuario", usuarioService.getUsuarioById(idUsuario));
    model.addAttribute("tipousuarios", usuarioService.getTiposUsuarios());
    return "/administrador/modificarusuario";
}

```

Principalmente, en la anotación *RequestMapping* se puede observar un texto con el siguiente formato {nombre\_variable}, esto sirve para que con la anotación *PathVariable* se pueda recoger y asignar en una variable de java.

### 5.1.4 Funciones

El sistema web tiene 3 funciones principales: creación de catálogos, creación de objetos y generación de códigos QR.

#### 5.1.4.1 Creación de catálogos

En esta función se trabaja en todas las capas del sistema, la primera capa es la de *Model*, donde se tiene la entidad mapeada con Hibernate dividiéndose en 2 tablas: *Catalogo* y *ColumnaCatalogo*. En la capa intermedia de *Service* se recogen los datos de la capa de las vistas y se hace la transacción con las entidades previamente descritas. En las vistas se hizo uso de la biblioteca *HandleBars* para permitir que fuera dinámico el registro. En la siguiente imagen se puede ver el flujo del proceso de un registro de un catálogo:

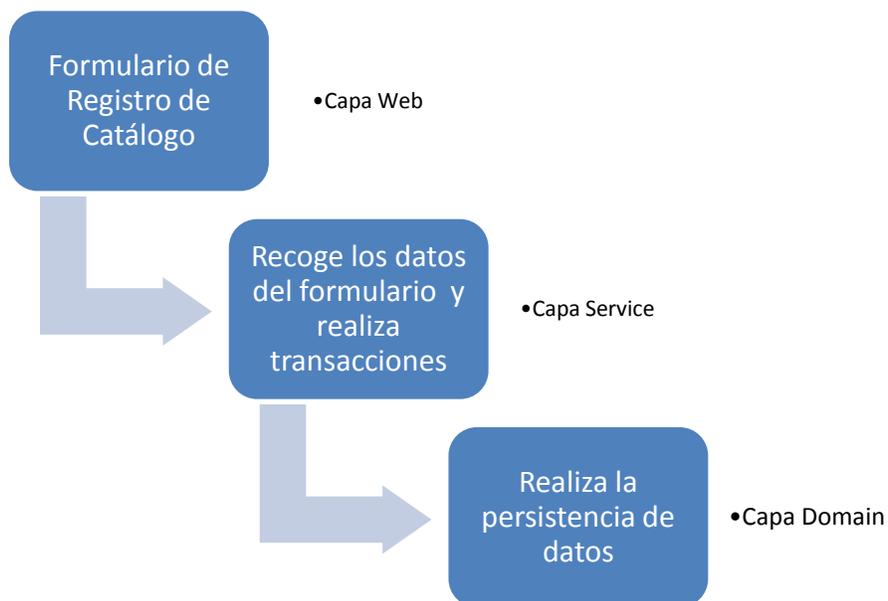


Figura 60 Flujo de Registro de Catálogos

### 5.1.4.2 Creación de objetos

En esta función igualmente se trabaja en todas las capas del sistema, la primera capa es la de *Model*, donde se tiene la entidad mapeada con *Hibernate* dividiéndose en 2 tablas: *Objeto* y *ColumnaObjeto*. En la capa intermedia de *Service* se recogen los datos de la capa de las vistas y se hace la transacción con las entidades previamente descritas. En las vistas se hizo uso de *Ajax* para cargar dinámicamente todas las columnas del catálogo seleccionado. En la siguiente imagen se puede ver el flujo del proceso de un registro de un objeto:

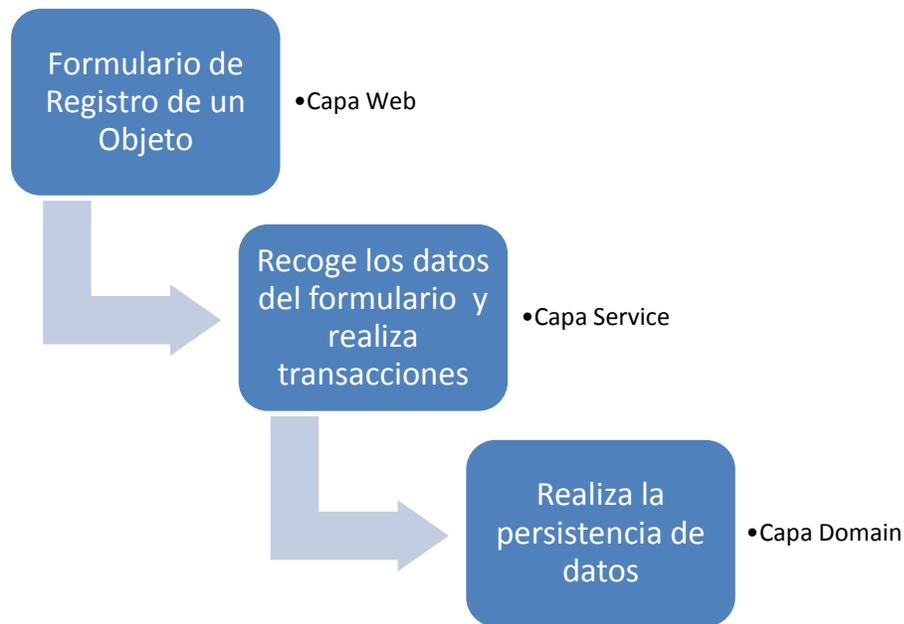


Figura 60 Flujo de Registro de Objetos

### 5.1.4.3 Generación de códigos QR

Esta función es la más relevante de la página web ya que se genera el código QR para que los dispositivos móviles puedan cargar la información de los objetos del sitio de interés cultural. La generación de realiza con la biblioteca de *Google* llamada *ZXing*. En la capa de *Service* es donde se lleva a cabo el proceso y la vista es la que le manda el identificador del objeto.

Con el siguiente código se genera el código QR principalmente, aunque se realizaron otros métodos para binarizar la imagen o invertir los colores:

```
BitMatrix bm;  
Writer writer = new QRCodeWriter();  
byte[] imageInByte = null;
```

```

try {
    bm = writer.encode(datos, BarcodeFormat.QR_CODE, ancho, alto);
    // Crear un buffer para escribir la imagen
    BufferedImage image = new BufferedImage(ancho, alto,
        BufferedImage.TYPE_INT_RGB);

    // Iterar sobre la matriz para dibujar los pixeles
    for (int y = 0; y < ancho; y++) {
        for (int x = 0; x < alto; x++) {
            int grayValue = (bm.get(x, y) ? 1 : 0) & 0xff;
            image.setRGB(x, y, (grayValue == 0 ? 0 : 0xFFFFFF));
        }
    }

    // Invertimos colores para binarizar correctamente
    image = invertirColores(image, ancho, alto);

    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    ImageIO.write(image, "gif", baos);
    baos.flush();
    imageInByte = baos.toByteArray();
    baos.close();
} catch (Exception e) {}

```

En la siguiente imagen se observa el flujo de la generación del código QR:

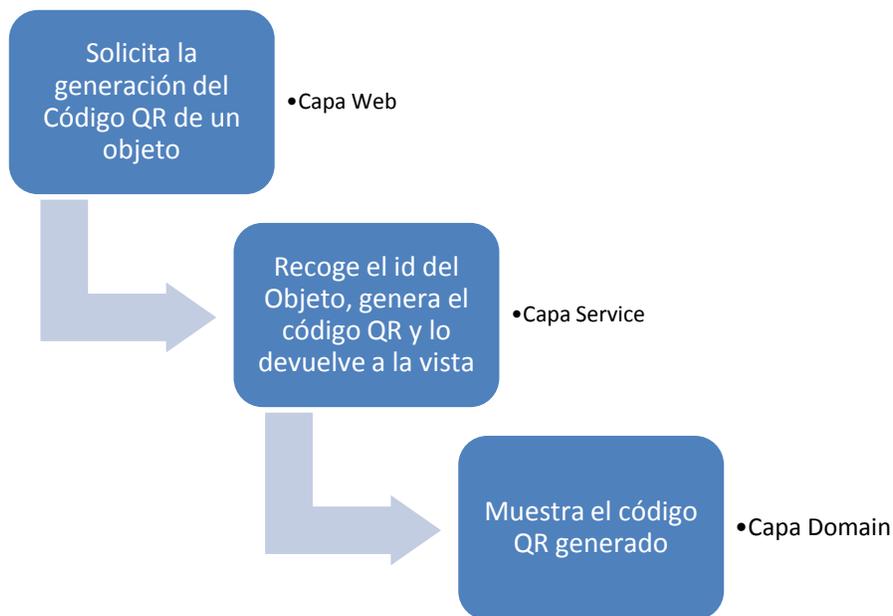


Figura 61 Flujo de generación de código QR

## 5.2 Dispositivo móvil

Ahora se describirá el funcionamiento del sistema en los dispositivos móviles, dividiéndose en 2 módulos principales: carga de información en las vistas y comunicación con el servicio web.

Se hizo uso del framework de *PhoneGap*, cuyas principales herramientas a utilizar son JavaScript, CSS3 y HTML5.

### 5.2.1 Vistas

Para que el sistema en el dispositivo móvil funcione de manera adecuada, se tienen dos funciones que son las más importantes en el desarrollo para el correcto funcionamiento de la aplicación, las cuales son:

- *mostrar*
- *mostrar\_editar*

En las cuales se describe el siguiente pseudo-código:

Función *mostrar* (recibe como parámetro la “*página*” donde pondrá las obras que han sido escaneadas)

```
{
```

```
    Se declara la variable $obras = lo que encuentre en la página con la clase “obras” y quitara ese elemento del html
```

```
    For(i=0; i<tamaño; i++) // donde el tamaño cambiara de acuerdo al número de obras escaneadas
```

```
    {
```

```
        $temp = El clon de la variable $obras //para mayor facilidad de manejo se crea una variable temporal
```

```
        For(k=0; k<tamaño; k++) //ahora buscaremos dentro de cada elemento del JSON
```

```
        {
```

```
            //el JSON es un objeto que se creó con la finalidad de ir almacenando
```

```
            //todos los escaneos, llamado pisinc.recorrido_virtual
```

```
            If (el índice de algún escaneo coincide con la i)
```

```
                Buscará dentro de cada uno de sus elementos y se los asignara al div o etiqueta correspondiente para ser mostrados en la pantalla de ver recorrido
```

Al final agregara ese div con atributos *name* y *id*, correspondientes a *k* y *idR* respectivamente

```
}  
}  
}
```

La principal diferencia entre las funciones de *mostrar* y *mostrar\_editar* es que en esta última se agregan iconos referentes a una flecha o a un icono de lista, para que el visitante tenga la opción de entrar a editar su obra escaneada, o modificar el orden de la lista de obras que tiene en pantalla, respectivamente.

También es importante mencionar que dentro de la aplicación no solo existe la variable *pisinc.recorrido\_virtual*, sino que también existe *pisinc.recorrido\_original* y *pisinc.recorrido\_obra*. La segunda nos ayuda a siempre recordar como fue el recorrido por cómo se vayan dando las circunstancias y para posteriores usos como el que se quite algún dato en editar la información, este puede ayudarnos a recuperarla. La información que se va a ir editando será guardada en *pisinc.recorrido\_virtual* ya que es el que se muestra siempre en las vistas del sistema, y el ultimo es el que nos ayuda para identificar qué información ha solicitado el visitante se le envié al correo electrónico que proporciono durante su registro al sistema.

De manera general el cuerpo del HTML del sistema en el dispositivo móvil, funciona de la siguiente manera, auxiliándose del JavaScript *mobileFramework*:

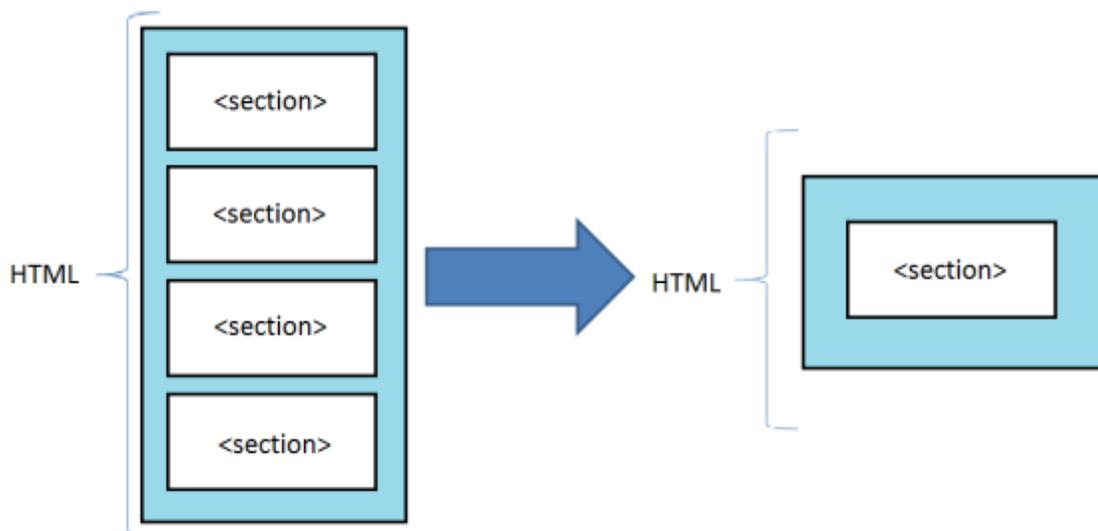


Figura 62 Cuerpo html del Phonegap

Una vez que el HTML ya emulado en *PhoneGap* ha identificado todos sus componentes y todo el cuerpo (imagen del lado izquierdo), y ha cargado las funciones con las que va a trabajar cada etiqueta `<section>`, esperara a que con *mobileFramework*, se le indique la sección con la que iniciara, en un “*init*”, y cada que el visitante presione algun icono del menú de opciones que se le presentaran, el *mobileFramework* hará una llamada al método de “*replace*”. Tanto “*init*” como “*replace*” lo que hacen es guardar la referencia del cuerpo completo del HTML y solo montar a la aplicación la “pantalla” o la “sección” correspondiente a la que se haya presionado, esto con la finalidad de ahorrarle recursos al móvil al momento de que se requiera buscar algún elemento dentro del HTML y los tiempos de respuesta se vean mejorados, ya que en lugar de buscar en todo el HTML, (que podría representar un problema, ya que podrían existir id’s o clases repetidas en más de una ocasión en el HTML), solo realiza la búsqueda en la sección que compete para la pantalla que se está mostrando (imagen del lado derecho).

## 5.2.2 Comunicación con el servicio web

Las funciones implementadas para la comunicación con el servidor, tienen como base el cliente SOAP empleado para el consumo del servicio web montado. Este cliente se encuentra implementado en el archivo *soapclient21.js*, de la carpeta *assets/www/js* de la aplicación móvil.

El cliente está construido en JavaScript y depende principalmente de la interfaz *XMLHttpRequest*, la cual nos permite realizar peticiones a servidores web, ya sea mediante HTTP o HTTPS. Así mismo, depende del analizador de XML presente en los navegadores web, en este caso, del *WebView* manejado por *PhoneGap*.

Las funciones principales de este cliente se encuentran descritas a continuación.

SOAPClient.\_soapresult2object

SOAPClient.\_node2object

Convierte las respuestas XML del servidor a su equivalente en objetos, para poder ser manipulados mediante JavaScript.

SOAPClient.\_sendSoapRequest

SOAPClient.\_onSendSoapRequest

Funciones encargadas del envío y la recepción de la petición al servicio web. Estas funciones tienen un comportamiento asíncrono, por lo que son dependientes de una función *callback*, en el caso de que se obtenga un resultado válido.

SOAPClient.\_loadWsdL

SOAPClient.\_onLoadWsdL

Funciones encargadas de obtener el archivo WSDL del servidor para el manejo correcto de las operaciones (llamadas a funciones, lectura/escritura de parámetros, etc.) del cliente.

SOAPClient.invoke  
SOAPClientParameters

Función y clase principal empleada para la llamada al servicio web con los parámetros indicados. La función *invoke* es asíncrona, por lo que por lo general deberá de pasarse una función *callback* para recibir el resultado de las operaciones llamadas.

La función *invoke* está definida de la siguiente manera.

```
SOAPClient.invoke = function(url, method, parameters, async, callback, callbackError){  
    this.callbackError = callbackError;  
    if(async){  
        SOAPClient._loadWSDL(url, method, parameters, async, callback);  
    }else{  
        return SOAPClient._loadWSDL(url, method, parameters, async, callback);  
    }  
}
```

Podemos ver que la función recibe la url del servicio web, el método a ser llamado, los parámetros pertinentes para el método especificado, así como de si se tratará de una llamada asíncrona y por último las funciones *callback* de éxito y error en la operación del servicio web.

La función *callbackError*, es ejecutada internamente por el cliente soap para determinar errores en la comunicación con el servicio web.

Esta función debe ser definida dentro del conjunto de las funciones que tendrán como base el cliente soap.

SOAPClientParameters.\_serialize

Función que se emplea en el llamado a *invoke*, para la serialización de los datos del objeto a su equivalente en XML, de acuerdo a la descripción proporcionada por el archivo WSDL.

Dentro de las funciones empleadas para manipular la información obtenida del servicio web, se desarrollaron clases para el manejo de la base del cliente. Ya que la mayoría de ellas presentan un patrón, se decidió colocar esta funcionalidad dentro del cliente soap descrita arriba (como es el caso de la función *invoke*), creando así funciones más específicas para el desarrollo de la aplicación.

Estas funciones fueron separadas a su vez en dos clases principales.

Las clases y funciones a continuación descritas se encuentran definidas en el archivo *pisinc-logica.js* dentro de la carpeta *assets/www/js*.

```
function Sesion(id){ this.id = id; }
```

La clase sesión es la encargada de mantener toda la información respecto a la sesión del usuario en el sitio. Estos datos abarcan, la id de identificación con el servidor, en caso de querer recuperar los datos, el contador de objetos, para el número de objetos descargados y el arreglo de las ids de los objetos que ya se tienen en la cache.

```
Sesion.agregarObjeto = function(obj);
```

Función para añadir un objeto a la lista de objetos escaneados, así de cómo actualizar las variables de sesión para marcar el objeto como recibido.

```
RemoteServiceDao
```

Esta clase contiene todos los métodos necesarios para la manipulación de la información provista por el servicio web. Esta clase es la encargada de almacenar datos como son la url compuesta por la ip, el puerto y la dirección del archivo WSDL del servicio web. Las funciones principales se describen a continuación.

A continuación se pasan a describir los métodos más importantes.

```
RemoteServiceDao.getObjeto = function(idObjeto);
```

Función para obtener un objeto con todos sus atributos.

Parámetros:

- idObjeto: Id del objeto a obtener del servidor.
- callback: Función a llamar para retornar el objeto descargado.

Se presenta la estructura de la función getObjeto.

```
RemoteServiceDao.getObjeto = function(idObjeto){  
    // Colocando los parámetros a obtener del WS.  
    var pl = new SOAPClientParameters();  
    pl.add("idObjeto", idObjeto);  
  
    // Petición del objeto.  
    SOAPClient.invoke(this.url, "getObjeto", pl, true,  
        ServiceHandler.objectHandler);  
}
```

Podemos ver que el parámetro pasado a la función es la id del objeto, esta id es obtenida mediante la lectura del código QR.

Dentro de la función se llenan los parámetros pl con dicha id, y por último se llama a la función invoke, con los parámetros correspondientes. Dentro de ellos observamos a la función ServiceHandler.objectHandler que es la encargada de recibir los datos del objeto y procesarlos.

Las demás funciones presentadas presentan la misma construcción.

RemoteServiceDao.iniciarRecorrido = **function**(correo, callback);

Función para registrar un nuevo recorrido en el servidor.

Parámetros:

- correo: Correo electrónico del usuario para registrar un nuevo recorrido.
  - NOTA: este es el correo empleado para enviar los datos del recorrido final.
- *callback*: Función empleada para recibir la Id de recorrido y procesarla/almacenarla.

RemoteServiceDao.enviarRecorrido = **function**(idRecorrido);

Función para que el servicio web envíe el recorrido al usuario.

Parámetros: Ninguno

RemoteServiceDao.recuperarRecorrido = **function**(idRecorrido);

Recupera el recorrido de un usuario de una sesión que aún se encuentre abierta.

Parámetros:

- idRecorrido: id del recorrido que aún es válida en el servidor.

ServiceHandler

Clase con los métodos necesarios para manipular las respuestas obtenidas del web service, siendo estas los objetos descargados o las sesiones que se crean para los usuarios.

ServiceHandler.guardarIdRecorrido = function(id);

Recibe la Id de recorrido y la almacena para comenzar el escaneo de los objetos.

Parámetros:

- Id: Respuesta del servicio web.

ServiceHandler.objectHandler = function(response){

Asignar los valores principales del objeto.

Recibir el objeto y formatearlo a uno de la forma:

objeto : {

  tituloCol1 : descripcion1

  tituloCol2 : descripcion2

  tituloCol3 : descripción3

```
tituloCol4 : descripción4
  ...
}
```

```
ServiceHandler.procesarRecorrido = function(response);
```

Función empleada para el procesado del recorrido que fue recuperado después de que la sesión con una id válida perdiera la comunicación con el servidor.

```
function ObjectBuilder({});
Clase: ObjectBuilder
```

Construye una instancia de Objeto, mediante la cabecera y las columnas del mismo.

```
ObjectBuilder.buildObjeto = function(data);
```

Construye un objeto a partir de los datos del WS.

```
function updateInterfazRecorrido(objeto)
```

Muestra el objeto en la interfaz.

```
function b64_to_utf8( str )
```

Convierte los datos de binario a ASCII para cadenas en Unicode.

```
function updateInterfazRecorrido(objeto);
```

Muestra el objeto en la interfaz.

### 5.2.3 Escaneo del código QR

```
var scanCode = function()
```

Esta función es la encargada de mandar a llamar al código nativo para la apertura de la cámara y la carga del código QR.

El código nativo empleado, es el desarrollado por el proyecto *ZXing* con su correspondiente estructura en *JavaScript*.

## 5.3 Vistas del sistema

En esta sección se muestran las vistas que componen el sistema indicando a que parte funcional del sistema corresponden, hablando tanto del Sitio Web como del Dispositivo Móvil.

### 5.3.1 Sitio web

Todas las vistas del sitio web utilizan clases llamadas *Controller*, hay uno para cada tipo de usuario además de 3 adicionales que nos ayudarán a mantener la seguridad del sitio.

#### 5.3.1.1 Pantalla de inicio de sesión

A continuación se muestra la pantalla correspondiente al inicio de sesión que tendrá el sistema web.

- ✓ Corresponde al caso de uso de Inicio de Sesión del Sitio Web.
- ✓ Principalmente hace uso de las clases *SecurityService* y *UsuarioService*, donde hace transacciones a la base de datos en la entidad de Usuario a través de la clase *UsuarioJPA*. Implementa el uso del framework de seguridad *Shiro* para el acceso a los datos.



Figura 63 Pantalla de Inicio de Sesión

### 5.3.1.2 Pantalla de bienvenida del administrador

Una vez ingresados los datos solicitados en la pantalla de inicio de sesión, y siempre y cuando los datos sean válidos, el sistema mostrara la siguiente pantalla. (La siguiente figura muestra la pantalla de bienvenida al administrador del sistema).

Inicio Administrador Capturista Jesús Norberto Sosa Rivera

## Bienvenido Jesús Norberto Sosa Rivera

ESCOM - Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-6000 Ext.: 46188. E-mail: direccion\_escom@ipn.mx

Desarrollado por la Escuela Superior de Cómputo ... TT 2012-A035

Figura 64 Pantalla de bienvenida del administrador

### 5.3.1.3 Pantalla de bienvenida del capturista

Si el inicio de sesión corresponde al rol de capturista, la figura que aparece a continuación, corresponde a la pantalla de bienvenida al capturista del sistema.

- ✓ Utiliza la clase CatalogoService quien a su vez hace transacciones a la base de datos en la entidad de Catalogo a través de la clase CatalogoJPA.

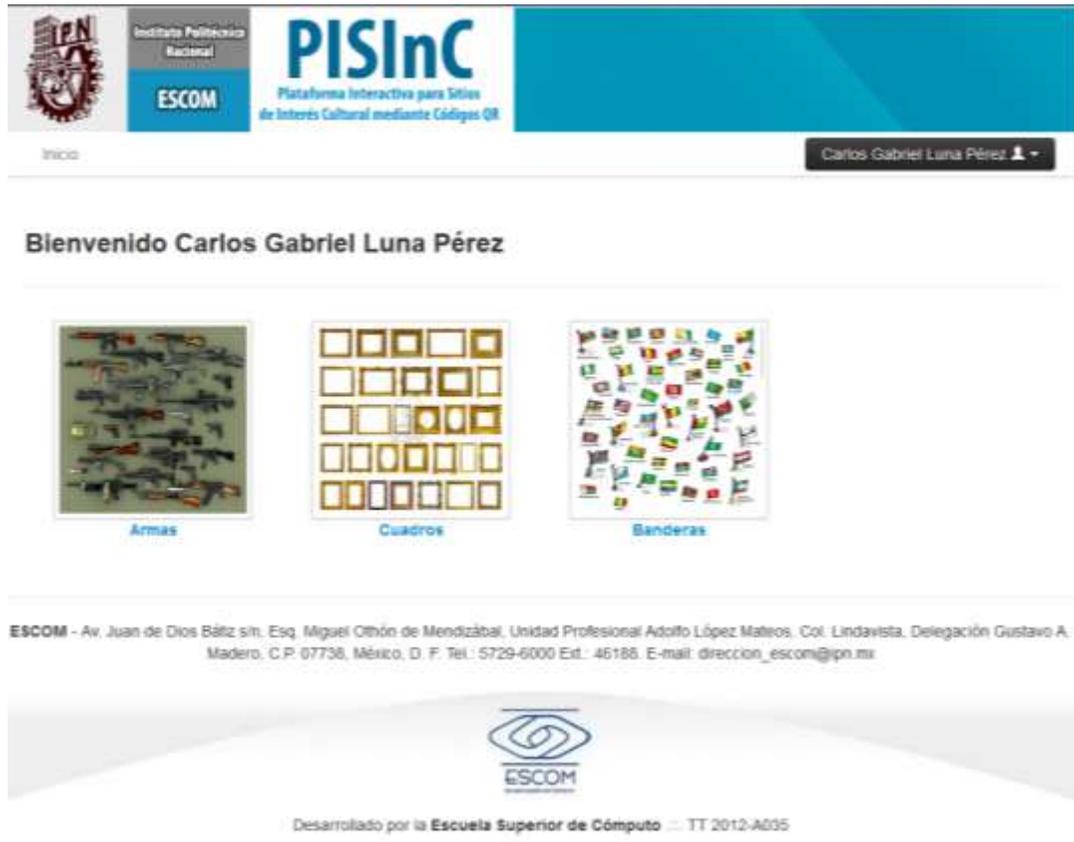


Figura 65 Pantalla de bienvenida del capturista

### 5.3.1.4 Pantalla de gestión de usuarios

Si el rol de inicio de sesión fue de administrador, este podrá tener acceso a los usuarios que tenga registrados el sistema, para que pueda gestionarlos. La figura que aparece a continuación corresponde a la gestión de esos usuarios.

- ✓ Corresponde al caso de uso de Gestión de Usuarios.
- ✓ Hace uso de la clase UsuarioService, donde hace transacciones a la base de datos en la entidad de Usuario a través de la clase UsuarioJPA.

Inicio Administrador Capturista Jesús Norberto Sosa Rivera

## Usuarios

Mostrar 10 registros Buscar:

ID	Nombre de Usuario	Tipo de Usuario	Acciones
1	admin	ADMINISTRADOR	▼
2	captura	CAPTURISTA	▼

Mostrando desde 1 hasta 2 de 2 registros

Registrar Nuevo Usuario

← Anterior Siguiente →

Modificar  
Eliminar

ESCOM - Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-6000 Ext.: 46188. E-mail: direccion\_escom@ipn.mx

ESCOM  
Escuela Superior de Cómputo

Desarrollado por la Escuela Superior de Cómputo ... TT 2012-A035

Figura 66 Pantalla de gestión de usuarios

### 5.3.1.5 Pantalla de registro de usuarios

Para el caso en que el administrador desee dar de alta nuevos usuarios, la figura siguiente muestra la pantalla correspondiente para esta acción.

- ✓ Corresponde al caso de uso de Registrar Usuario.
- ✓ Hace uso de la clase UsuarioService, donde hace transacciones a la base de datos en la entidad de Usuario a través de la clase UsuarioJPA.

The screenshot displays the 'Registrar Usuario' (Register User) form within the PISInC application. The header includes the IPN logo, 'Instituto Politécnico Nacional', 'ESCOM', and 'PISInC Plataforma Interactiva para Sitios de Interés Cultural mediante Códigos QR'. The user 'Jesús Norberto Sosa Rivera' is logged in. The form contains the following fields:

- Nombre:
- E-Mail:
- Usuario:
- Contraseña:
- Tipo de Usuario:

Buttons for 'Cancelar' (red) and 'Registrar' (green) are located at the bottom of the form. Below the form, contact information for ESCOM is provided: 'ESCOM - Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-6000 Ext.: 46188. E-mail: direccion\_escom@ipn.mx'. The ESCOM logo and the text 'Desarrollado por la Escuela Superior de Cómputo :::: TT 2012-A035' are at the bottom.

Figura 67 Pantalla de registro de usuarios

### 5.3.1.6 Pantalla de modificación de usuarios

Existe la posibilidad de que el administrador del sitio web desee modificar los roles de los usuarios registrados en el sistema, o bien algún otro dato. La siguiente figura muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Modificar Usuario.
- ✓ Hace uso de la clase UsuarioService, donde hace transacciones a la base de datos en la entidad de Usuario a través de la clase UsuarioJPA.

**Modificar Usuario**

Datos del Usuario

Nombre: Carlos Gabriel Luna Pérez

E-Mail: captura@correo.com

Usuario: captura

Contraseña Actual: captura

Nueva Contraseña:

Repita la Contraseña:

Tipo de Usuario: Capturista

Cancelar Modificar

ESCOM - Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-6000 Ext.: 46188. E-mail: direccion\_escom@ipn.mx

ESCOM

Desarrollado por la Escuela Superior de Cómputo ...: TT 2012-A035

Figura 68 Pantalla de modificación de usuarios

### 5.3.1.7 Pantalla de gestión de catálogo

El administrador del sitio web será el único encargado de gestionar los catálogos que estén registrados en el sistema. La siguiente figura muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Gestión de Catálogos.
- ✓ Hace uso de la clase CatalogoService, donde hace transacciones a la base de datos en la entidad de Catalogo a través de la clase CatalogoJPA.

Mostrar 10 registros

Buscar:

ID	Descripción	Acciones
5	Armas	<input type="button" value="v"/>
6	Cuadros	<input type="button" value="v"/>
7	Banderas	<input type="button" value="v"/>

Mostrando desde 1 hasta 3 de 3 registros

Modificar  
Eliminar

ESCOM - Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-6000 Ext.: 46168. E-mail: direccion\_escom@ipn.mx

Figura 69 Pantalla de gestión de catálogo

### 5.3.1.8 Pantalla de crear catálogo

Dentro de la gestión de los catálogos, el administrador del sitio web, podrá dar de alta nuevos catálogos. La siguiente figura muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Crear Catálogo.
- ✓ Hace uso de la clase CatalogoService, donde hace transacciones a la base de datos en la entidad de Catalogo a través de la clase CatalogoJPA.



#### Crear Catálogo

Datos del Catálogo

Descripción:

Imagen:  

Nombre del Campo	Tipo
<input type="text"/>	Imagen
<input type="text"/>	Seleccione un Tipo de Campo
	Imagen
	Link
	Texto

ESCOM - Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-8000 Ext.: 46188. E-mail: direccion\_escom@ipn.mx



Desarrollado por la Escuela Superior de Cómputo :::: TT 2012-A035

Figura 70 Pantalla de crear catálogo

### 5.3.1.9 Pantalla de modificación de catálogo

Existe la posibilidad de que el administrador del sitio web desee modificar los campos de los catálogos registrados en el sistema, o bien algún otro dato. La siguiente figura muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Modificar Catálogo.
- ✓ Hace uso de la clase CatalogoService, donde hace transacciones a la base de datos en la entidad de Catalogo a través de la clase CatalogoJPA.

Inicio Administrador\* Capturista Jesús Norberto Sosa Rivera

Tenga en cuenta que si elimina una columna, se eliminarán los datos pertenecientes a esa columna en todos los objetos relacionados al catálogo.

### Modificar Catálogo

Datos del Catálogo

Descripción: Armas

Imagen:

Nombre del Campo	Tipo	
Imagen	IMAGEN	X
Calibre	TEXTO	X
Año	TEXTO	X
Lugar de Origen	TEXTO	X
Dimensiones	TEXTO	X
Material	TEXTO	X

Agregar

Cancelar Modificar

Figura 71 Pantalla de modificación de catálogo

### 5.3.1.10 Pantalla de gestión de objetos (Administrador)

De manera análoga a la gestión de catálogos se cuenta con una pantalla para la gestión de los objetos que se registren en el sistema. La figura siguiente muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Gestión de Objetos.
- ✓ Hace uso de la clase ObjetoService, donde hace transacciones a la base de datos en la entidad de Objeto a través de la clase ObjetoJPA.

**Objetos**

Mostrar 10 registros      Buscar:

ID	Nombre del Objeto	Catálogo	Acciones
13	Sub Ametralladora Thompson	Armas	▼
14	Carabina Máuser	Armas	▼
15	Trabuco	Armas	▼
16	Pistola de Chispa	Armas	▼
17	General Pedro José Zarzosa de Oviedo	Cuadros	▼
18	Guadalupe Victoria	Cuadros	▼
19	Agustín Cosme Damián de Iturbide y Arámburu	Cuadros	▼
20	Arcabuz de Serpentín	Armas	▼

Mostrando desde 1 hasta 8 de 8 registros

[Registrar Nuevo Objeto](#)      [← Anterior](#)

- Generar Código QR
- Modificar
- Eliminar

ESCOM - Av. Juan de Dios Bátis s/n, Esq. Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Col. Lindavista, Delegación Gustavo A. Madero, C.P. 07738, México, D. F. Tel.: 5729-6000 Ext.: 48188, E-mail: direccion\_escom@ipn.mx

  
Desarrollado por la Escuela Superior de Cómputo :::: TT 2012-A035

Figura 72 Pantalla de gestión de objetos (Administrador)

### 5.3.1.1 Pantalla de gestión de objetos (Capturista)

El capturista tiene una forma más intuitiva para gestionar los objetos del sistema, esta vista será compatible con cualquier dispositivo móvil o Tablet.

- ✓ Corresponde al caso de uso de Gestión de Objetos.
- ✓ Hace uso de la clase ObjetoService, donde hace transacciones a la base de datos en la entidad de Objeto a través de la clase ObjetoJPA.

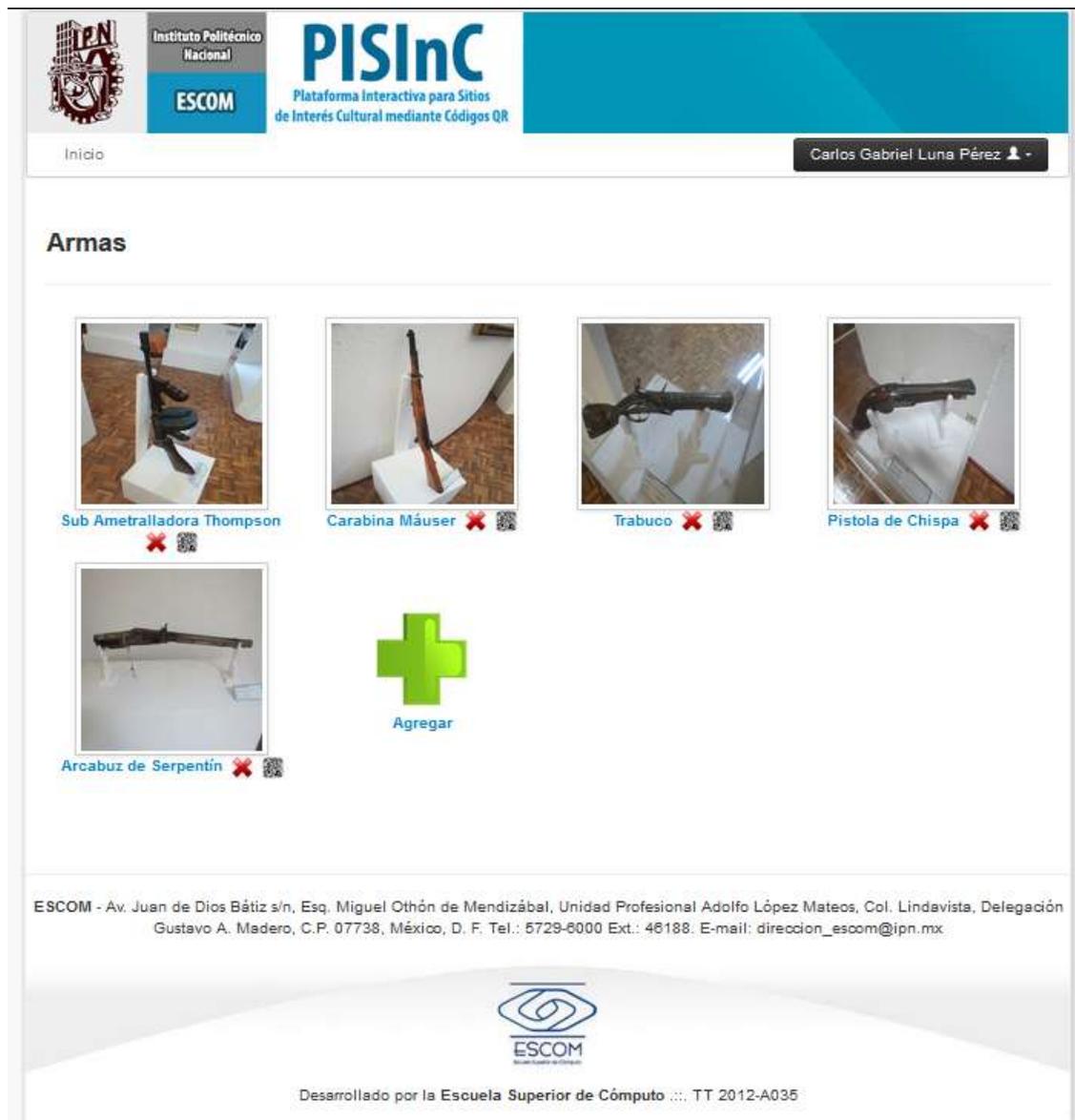


Figura 73 Pantalla de gestión de objetos (Capturista)

### 5.3.1.12 Pantalla de registrar objeto

Dentro de la gestión de los objetos, existe la opción de dar de alta nuevos objetos. Para lo cual, la figura siguiente ejemplifica la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Registrar Objeto.
- ✓ Hace uso de la clase ObjetoService, donde hace transacciones a la base de datos en la entidad de Objeto a través de la clase ObjetoJPA.

The screenshot shows the 'Registrar Objeto' page. The header features the IPN logo, 'Instituto Politécnico Nacional', 'ESCOM', and 'PISInC' with the subtitle 'Plataforma Interactiva para Sitios de Interés Cultural mediante Códigos QR'. A user profile for 'Carlos Gabriel Luna Pérez' is in the top right. The main content area is titled 'Registrar Objeto' and contains a form with the following fields:

- Catalogo: Armas
- Nombre: [Empty text box]
- Imagen: [Image upload button]
- Calibre: [Empty text box]
- Año: [Empty text box]
- Lugar de Origen: [Empty text box]
- Dimensiones: [Empty text box]
- Material: [Empty text box]

At the bottom of the form are two buttons: 'Cancelar' (red) and 'Registrar' (green).

Figura 74 Pantalla de registrar objeto

### 5.3.1.13 Pantalla de modificación de objeto

Existe la posibilidad de que el administrador del sitio web desee modificar los campos de los objetos registrados en el sistema, o bien algún otro dato. La siguiente figura muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Modificar Objeto.
- ✓ Hace uso de la clase ObjetoService, donde hace transacciones a la base de datos en la entidad de Objeto a través de la clase ObjetoJPA.

The screenshot displays the 'Modificar Objeto' (Modify Object) page of the PISInC system. The header includes the logos for IPN, Instituto Politécnico Nacional, ESCOM, and PISInC (Plataforma Interactiva para Sitios de Interés Cultural mediante Códigos QR). A user profile for 'Carlos Gabriel Luna Pérez' is visible in the top right. The main content area is titled 'Modificar Objeto' and contains a form with the following fields:

- Datos del Objeto:**
  - Catálogo:
  - Nombre:
  - Imagen:  with a small image thumbnail.
  - Calibre:
  - Año:
  - Lugar de Origen:
  - Dimensiones:
  - Material:

At the bottom of the form are two buttons: 'Cancelar' (red) and 'Modificar' (green).

Figura 75 Pantalla de modificación de objeto

### 5.3.1.14 Pantalla de generar código QR

Una vez que se tenga capturado un objeto dentro del sistema, se cuenta con la opción de generar el código QR correspondiente al objeto en cuestión. La figura mostrada a continuación, ejemplifica la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Generar código QR.
- ✓ Hace uso de la clase ObjetoService, donde hace transacciones a la base de datos en la entidad de Objeto a través de la clase ObjetoJPA e implementa la biblioteca ZXing para la generación del código QR.

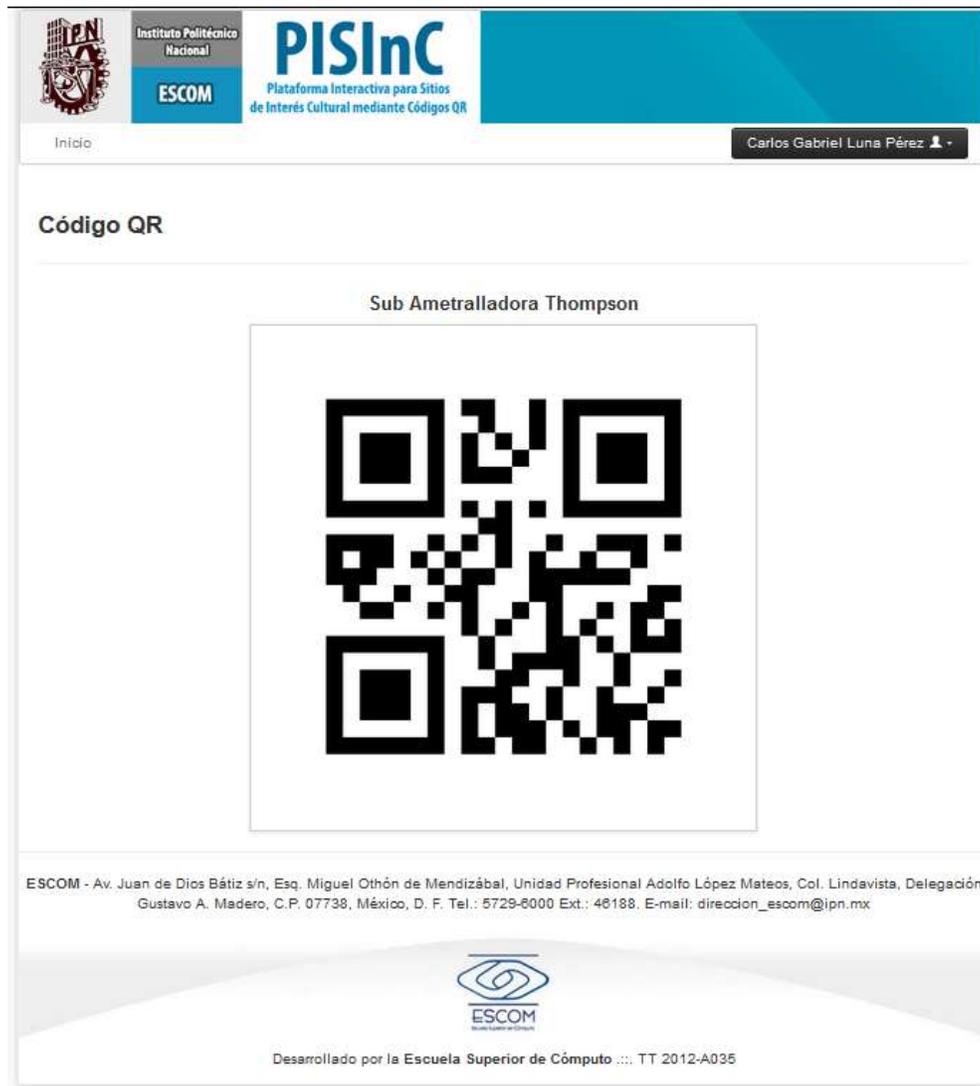


Figura 76 Pantalla de generar código QR

### 5.3.2 Dispositivo móvil

En el presente capítulo se detalla la información correspondiente a las vistas que se desarrollaron para los dispositivos móviles.

#### 5.3.2.1 Pantalla de bienvenida

En el momento en el que se ejecute la aplicación en el dispositivo móvil, se le muestra al visitante del sitio de interés cultural un pequeño mensaje de bienvenida a la plataforma. El código correspondiente se encuentra en index.html en la sección del tutorial.



Figura 77 Pantalla de bienvenida

### 5.3.2.2 Pantalla de registro de correo electrónico

Se le proporcionara al visitante la opción de que al final de su recorrido por el sitio de interés cultural, pueda enviarse la información recabada a lo largo del mismo. Por lo tanto como se muestra en la siguiente figura, se le solicita al visitante ingrese una dirección de correo electrónico para que pueda recibir la información que le resultado de interés. El código correspondiente se encuentra de igual manera en la sección de tutorial, del archivo index.html.



Figura 78 Pantalla de registro de correo

### 5.3.2.3 Tutorial de uso

Para que de alguna manera el visitante comience a familiarizarse con la aplicación, se ha agregado un pequeño tutorial sobre que significan cada uno de los iconos que se encuentran en la aplicación, así como su funcionalidad. Estas pantallas se pueden ubicar en la sección del tutorial en el archivo index.html.



Figura 79 Icono de la cámara



Figura 80 Icono del recorrido



Figura 81 Icono de editar



Figura 82 Icono de enviar



Figura 83 Icono de ajustes

### 5.3.2.4 Pantalla de inicio de sesión

La figura que se muestra a continuación corresponde al inicio de sesión que tiene el visitante una vez que ya tenga registrado su correo electrónico en la aplicación. Para posteriores cambios a la dirección de correo que se ha proporcionado, se deberá acceder a la sección de ajustes y consultar el apartado correspondiente mencionado en los próximos subcapítulos.

- ✓ Corresponde al caso de uso de iniciar sesión.



Figura 84 Pantalla de inicio de sesión

### 5.3.2.5 Pantalla del escáner

La figura que se muestra a continuación corresponde a la pantalla de escaneo de los códigos QR, que hace uso de la cámara fotográfica con la que debe contar el dispositivo móvil. Esta pantalla aparece cada vez que en la aplicación del dispositivo móvil se le dé clic al icono de la cámara. El código correspondiente a esta pantalla se encuentra en `pisinc.js`, una vez que el dispositivo cargue los archivos y se encuentre listo para lanzar la aplicación.

- ✓ Corresponde al caso de uso de Escanear código QR.



Figura 85 Pantalla del escáner

### 5.3.2.6 Pantalla del editor

Una vez que ya se haya escaneado un código QR, que corresponda a alguna de las exposiciones disponibles en el sitio de interés cultural, el dispositivo muestra en pantalla una lista de las exposiciones que ya se hayan escaneado previamente, con el fin de dejarle ver al visitante las exposiciones que ha visitado. En la sección de editar, código ubicado en `pisinc.js` una vez que haya hecho el cambio de pantalla, se muestra toda la información referente al objeto del cual se haya extraído la información, para que el visitante pueda modificar la información mostrada de acuerdo a sus necesidades. La figura que aparece a continuación muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Modificar Datos en el Editor.
- ✓ Corresponde al caso de uso de Modificar Datos.



Figura 86 Pantalla del recorrido



Figura 87 Pantalla del editor

### 5.3.2.7 Pantalla de la lista de objetos

El visitante tiene la oportunidad que dentro de la aplicación del dispositivo, se irá almacenando (en modo de lista) el recorrido, cada que el visitante vaya escaneando los códigos QR disponibles en las exposiciones. La figura que se muestra a continuación ejemplifica la pantalla correspondiente.



Figura 88 Pantalla de la lista de objetos

### 5.3.2.8 Pantalla de enviar correo

Una vez que el visitante ya no desee escanear más objetos dentro del sitio de interés, tiene la opción de enviarse la información por correo electrónico, para esto deberá verificar que el correo que se le muestra en la pantalla es el adecuado para recibir esta información y de ser así solo deberá presionar el botón de enviar. La figura siguiente muestra la pantalla correspondiente.

- ✓ Corresponde al caso de uso de Enviar información al Correo.



Figura 89 Pantalla de envío de información al correo electrónico

Una vez que se envía la información al correo electrónico, los datos que se tengan en el dispositivo móvil sobre el recorrido y diversas variables que se manejan, se limpian, y se cierra la sesión y se muestra la pantalla de iniciar sesión.

- ✓ Corresponde al caso de uso de Cerrar sesión.

### 5.3.2.9 Pantalla de ajustes

Se muestra una lista de los ajustes que el visitante le podrá hacer a la aplicación durante su uso, como lo es el cambio de correo electrónico, en el caso de que haya ingresado un correo que no considere el adecuado para recibir la información que encontró en el sitio de interés cultural, o bien para volver a ver el tutorial que se mostró al inicio en caso de que tenga dudas sobre el uso de la aplicación con respecto a sus iconos.



Figura 90 Pantalla de principal de ajustes



Figura 91 Pantalla de cambio de correo



Figura 92 Pantalla de ajustes (Ver nuevamente el Tutorial) y Pantalla de configuraciones avanzadas

## CAPÍTULO 6 PRUEBAS Y RESULTADOS

En esta sección se presentan las diversas pruebas, reglas y normas a las que el sistema está sujeto para verificar el correcto desarrollo y obtención de los resultados esperados en la implantación del mismo.

Un punto importante para conocer el buen desempeño de nuestra aplicación es el conjunto de parámetro al que ésta debe ajustarse. A cada uno de los módulos se le deberán aplicar distintos tipos de mediciones, principalmente al módulo móvil, ya que deberá de cumplir con los estándares que han venido originándose en torno a las aplicaciones móviles, siendo así que los usuarios son cada vez más exigentes en el diseño que las aplicaciones deben de tener, así como en su fácil uso.

Aunque no con las mismas restricciones, el módulo web, deberá de cumplir con una serie de normas para su fácil aceptación y comprensión por parte de los usuarios de esta parte del sistema, para así ser puesta en uso en el menor tiempo posible.

### 6.1 Métricas

Previo a las pruebas, el sistema deberá ajustarse a ciertas métricas que describiremos en esta sección. Las principales a tomar en cuenta son las descritas a continuación.

#### 6.1.1 Desarrollo y aplicación de cuestionarios

Los cuestionarios serán nuestra principal fuente de retroalimentación en el uso de la aplicación móvil. Estos se basan en una serie de reglas conocidas para el diseño de interfaces que involucran la interacción del humano con la computadora (*Human Computer Interaction*, HCI), mejor conocidas como las 8 reglas de oro del diseño de interfaces[24]. Con estos cuestionarios se espera cumplir con lo descrito en estas reglas, así como con las sugerencias y puntos de vista de los usuarios finales (visitantes al sitio de interés cultural).

#### 6.1.2 Aplicación de las 8 reglas del diseño de interfaces gráficas

Este conjunto de principios fueron propuestos derivados de manera heurística de la experiencia y aplicación en distintos sistemas interactivos, esto luego de ser adecuadamente refinados, ampliados e interpretados. Tomamos en cuenta que, para mejorar la usabilidad de una aplicación es importante tener una interfaz bien diseñada.

Las reglas son las siguientes:

1. **Conseguir consistencia:** En situaciones similares se requiere de una secuencia consistente de acciones; como es el caso de la terminología en las indicaciones, menús y pantallas de ayuda, así como comandos consistentes deben de ser empleados a lo largo del uso de la aplicación.

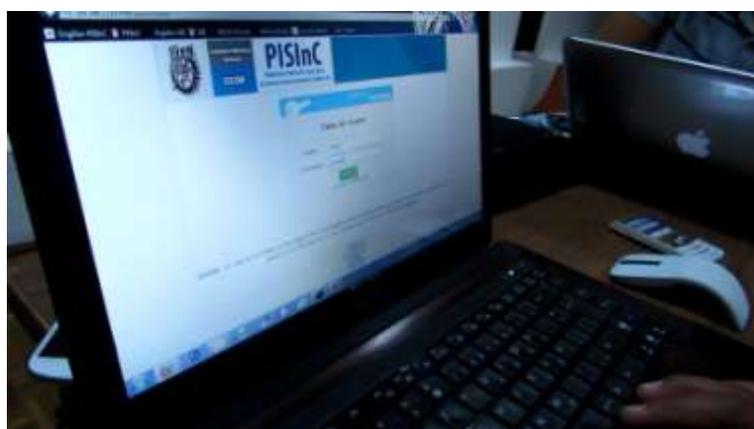
2. **Permitir a los usuarios frecuentes el uso de atajos:** Mientras la frecuencia de uso se incrementa, el usuario prefiere reducir el número de pasos necesarios para realizar acciones e incrementar así su ritmo en el desarrollo de la app. Algunos ejemplos de atajos son teclas de funciones, comandos ocultos, macros, abreviaciones; los cuales son de gran ayuda a usuarios expertos.
3. **Ofrecer retroalimentación informativa:** Para cada acción realizada, debe haber una retroalimentación, Para acciones frecuentes pero de menor importancia, la respuesta puede ser modesta, mientras que para acciones poco frecuentes e importantes, la respuesta debe ser más sustancial.
4. **Diseñar diálogos que produzcan un cierre:** Las secuencias de acciones deben estar organizadas en grupos con un comienzo, punto intermedio y fin. La retroalimentación informativa al completar un grupo de acciones le da a los usuarios la sensación de lograr un resultado, indicando que todo está preparado para la siguiente acción.
5. **Ofrecer un manejo de errores simple:** Cuanto más sea posible, diseñar el sistema de tal manera que el usuario no pueda producir errores serios. Si un error se comete, el sistema debería ser capaz de detectar el error y ofrecer mecanismos comprensibles y simples de manejar el error.
6. **Permitir el revertir las acciones de manera sencilla:** Esta característica libera ansiedad, dado que el usuario sabe que los errores se pueden deshacer; de esta manera alienta a la exploración de opciones poco familiares. La reversibilidad puede estar dada por una sola acción, una entrada de datos, o todo un grupo completo de acciones.
7. **Soporte a la sensación de control:** Los usuarios experimentados tienen la necesidad de tener cierto control sobre el sistema y que a su vez, el sistema responde a sus acciones. El diseño del sistema se debe realizar con la idea de que los usuarios son los iniciadores de las acciones, en vez de los que responden a ellas.
8. **Reducir el uso de la memoria a corto plazo:** La limitación del procesamiento de información por las personas, en el caso de la memoria a corto plazo, requiere que lo que se muestra se mantenga simple, que las múltiples páginas en una pantalla se consoliden, la frecuencia de movimiento en las ventanas se reduzca, y que se disponga de suficiente tiempo para el entrenamiento de la mente a códigos, mnemónicos, y secuencias de acciones.

## 6.2 Pruebas

Para las pruebas del sistema, se implementó en el Museo de Caballería, que fue el caso de estudio fijado y el cual será la principal fuente que le dará uso. Dentro de los que se abarcan el uso de la interfaz web por parte de los administrativos, así como del uso de los móviles por parte de los visitantes al sitio de interés.



**Figura 93 Encargado del museo usando el sistema móvil**



**Figura 94 Encargado del museo usando el sistema web**

En el sistema web se hicieron pruebas de almacenamiento, es decir, a probar su capacidad para poder mantener toda la información que pudiera llegar a requerir el sitio de interés sin mayores problemas.

Por otra parte el servidor encargado de mantener el servicio web fue sometido a múltiples conexiones con lo cual se analizará su carga de trabajo y su velocidad de respuesta.

La parte móvil fue probada en conjunto con el servidor, ya que hay casos específicos en las que se requiere de una buena recuperación de los datos que haya originado el usuario, en caso de que surjan problemas en la conexión del móvil y el servidor.

Por otro lado, la parte móvil fue probada con su capacidad para manejar los datos obtenidos desde el servicio web, así como la capacidad de edición de los mismos desde el móvil y poder mantener dicha ajustes para cada uno de los móviles que se van conectando al servidor.

De igual manera se realizaron pruebas en el tiempo de respuesta, pues ya que empleamos un framework para el desarrollo de ambas partes, esto podría incrementar los tiempos de respuesta. Es con estas pruebas que se podrían mejorar las características de la aplicación móvil y la respuesta del servicio web.



Figura 95 Equipo de desarrollo durante las pruebas

### 6.3 Resultado de las Pruebas

Como resultado de las pruebas tenemos las siguientes tablas, las cuales nos indican los tiempos de respuesta y la capacidad de los datos. En esta parte no tenemos alguna métrica ya que consideramos que es difícil evaluar qué tiempo debe ser el correcto para funcionar, ya que depende de muchos factores.

Esta primera tabla corresponde a la parte de tiempos de respuesta y fueron tomados con un reloj, por lo que no son muy exactos:

Prueba	Tiempo de Respuesta
<b>Conexión al Servicio Web desde una computadora</b>	2 seg.
<b>Conexión al Servicio Web desde el dispositivo móvil</b>	14 seg.
<b>Carga de Página Web desde una computadora</b>	2 seg.
<b>Carga de Catálogos del Sistema</b>	1 seg.
<b>Carga de Objetos del Sistema</b>	12 seg.
<b>Registro de Catálogo</b>	2 seg.
<b>Registro de Objeto</b>	2 seg.

Tabla 67 Tiempos de respuesta

Consideramos que los tiempos registrados son adecuados para el funcionamiento del sistema ya que aunque notamos que la conexión del dispositivo móvil al servicio web se tardó bastante, es algo solucionable con un amplificador de red en el museo.

## 6.4 Cuestionarios de Calidad

Con el fin de saber si nuestro sistema cumple o no con lo propuesto anteriormente, se hicieron cuestionarios a los visitantes del Museo de Caballería que implementaron nuestro sistema.

El cuestionario se aplicó después de que el usuario utilizó el sistema en sus dispositivos móviles a través de la herramienta web *SurveyMonkey* (en este enlace se puede ver la encuesta: <http://www.surveymonkey.com/s/XY2CVCG>); se realizaron 10 sencillas preguntas donde se puede evaluar de 1 a 10 y un apartado para su opinión personal

Las preguntas de los cuestionarios se promediaron entre todas y se utilizó la siguiente métrica para evaluar el sistema:

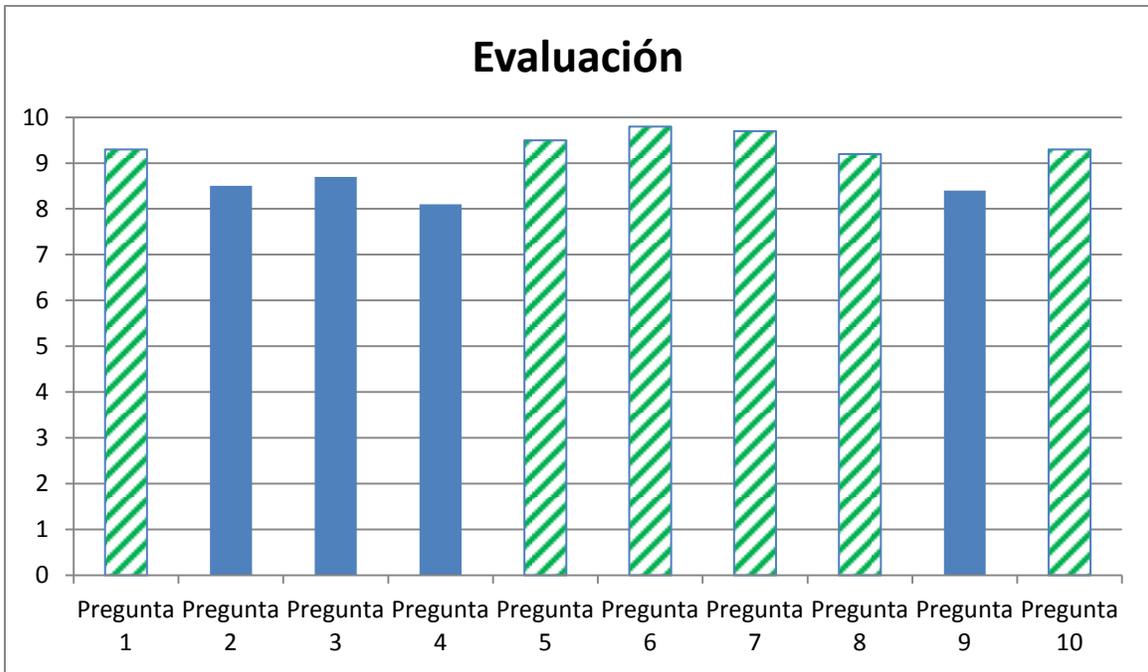
Puntos	Calificación del Sistema
1.0-5.9	Mala
6.0-7.9	Regular
8.0-9.0	Bueno
9.1-10.0	Excelente

Tabla 68 Métrica de evaluación de cuestionarios

Los cuestionarios fueron aplicados a un total de 30 personas, los resultados promedio se reflejan en la siguiente tabla:

Pregunta	Promedio	Nota
1. ¿El sistema es fácil de usar?	9.3	Excelente
2. ¿Qué tanto le es útil el sistema?	8.5	Bueno
3. ¿Cómo considera la velocidad de respuesta del sistema?	8.7	Bueno
4. ¿Qué tanto utilizaría el sistema en un museo?	8.1	Bueno
5. ¿El sistema es agradable a la vista?	9.5	Excelente
6. ¿Tiene todo lo necesario para funcionar adecuadamente?	9.8	Excelente
7. ¿Le gustó el resultado de su recorrido en formato PDF?	9.7	Excelente
8. ¿Qué tanto recomendaría el sistema?	9.2	Excelente
9. ¿Qué tanto cree que el sistema promueva las visitas?	8.4	Bueno
10. En una calificación del 1 al 10, que nota le pondría al sistema.	9.3	Excelente

Tabla 69 Resultado de métrica de cuestionarios



**Figura 96 Gráfica de resultado de la métrica de cuestionarios**

Como podemos percatarnos de la tabla y la gráfica anterior, en general es un sistema fácil de usar y cumple con los objetivos planteados, de acuerdo en la opinión de los visitantes del museo. Las barras con líneas diagonales resaltan en donde se tuvieron los resultados esperados.

# CAPÍTULO 7 CONCLUSIONES Y TRABAJO A FUTURO

## 7.1 Conclusiones

Una vez concluido el desarrollo e implantación del sistema planteado en este documento, retomamos los objetivos propuestos para resaltar las conclusiones que hemos obtenido.

Se desarrolló un sistema compuesto por dos módulos principalmente, el primero de ellos es el módulo web, el cual es capaz de administrar la información correspondiente al sitio de interés, siendo esta en su mayoría muestras museográficas pertenecientes al acervo del Museo de Caballería de la Secretaría de la Defensa Nacional, el sistema se encarga de la generación de los códigos QR, los cuales son colocados por los encargados del museo en las piezas correspondientes. El segundo módulo consta de una aplicación móvil con soporte a las plataformas Android e iOS, el cual es capaz de leer los códigos QR presentes en las piezas designadas de la exposición.

Lo anterior fue realizado mediante el uso del *framework PhoneGap*. Este *framework* fue elegido con el propósito de crear una aplicación multiplataforma.

Si bien el *framework* nos permitió realizar lo anterior, existieron inconvenientes en su utilización, ya que *PhoneGap* está en constante desarrollo, presenta errores en distintos dispositivos de prueba, principalmente con la implementación de la interfaz de usuario. Este inconveniente tuvo una solución parcial teniendo que rediseñar la interfaz mediante las hojas de estilo.

A su vez, fueron desarrollados algoritmos para la interacción con el servicio web, ya que al ser este *framework* dependiente de tecnologías web se debió implementar un cliente en esos lenguajes para su correcto funcionamiento.

La manipulación de la información de los datos del museo en el dispositivo móvil se realiza sin problemas con base en el diseño propuesto, ya que el usuario final es libre de elegir los datos a su conveniencia.

Por último, implementando el sistema en el sitio de interés cultural, pudimos darnos cuenta de que el sistema dependerá del hardware de red presente en el mismo, principalmente en los puntos de acceso a la red local que los dispositivos tienen dentro del museo. En el caso de interés planteado, se encontraron problemas con la red, pues no se abarca más allá de la mitad de las instalaciones, siendo así que se requiera de equipo adicional para completar el correcto funcionamiento del sistema.

A partir de los resultados obtenidos al aplicar los cuestionarios de calidad, podemos decir que el uso del sistema fue bien recibido por los administrativos y visitantes al sitio de interés, dando sugerencias o esperando su implementación en otros sitios que podrían visitar.

## 7.2 Trabajo a futuro

Las ventajas de la implementación del sistema mediante un servicio web, dan la posibilidad de que se tenga acceso a la información únicamente creando o modificando nuevos clientes para el servicio, es decir, que la información perteneciente al museo, puede ser presentada al usuario por medio de nuevas interfaces o formas de acceso a ellas. En el caso planteado en este trabajo, se trató de dispositivos móviles, pero bien podría tratarse de otro tipo de sistemas propios del museo o externos, que tomen la información disponible mediante el servicio y la muestren de maneras completamente distintas a las presentes en este trabajo, mejorando o haciendo más atractivo el recorrido del visitante al sitio de interés cultural.

Dentro de esta propuesta, igualmente incluimos la sugerencia de emplear el desarrollo nativo de las aplicaciones, ya que se presentan varios problemas al emplear *frameworks* como lo fue en el caso de *PhoneGap*.

Visto el interés prestado por parte de las autoridades del sitio de interés cultural, podemos darnos cuenta que este tipo de trabajos de innovación tecnológica, presentan una gran aceptación, tanto por parte de las autoridades, como de los mismos visitantes al sitio. El sistema puede extenderse, e incluso siendo más ambiciosos, puede generarse una red que conecte los museos de una localidad, en la que se tenga acceso a la información presente en todos ellos, de manera libre, como se ha podido ver en otros países y los ejemplos descritos en este trabajo.

Sin duda la innovación tecnológica es de gran aceptación dentro de los sitios de interés cultura, por lo que este y trabajos similares tienen grandes posibilidades de ser implementados y aceptados, trayendo grandes beneficios en estos sitios.

# ANEXOS

## I Diccionario de Datos

### I.I Tablas

Tabla	Hijos	Padres	Columnas
<b>bitacora</b>		3	3
<b>catalogo</b>	3		3
<b>columnacatalogo</b>	1	1	4
<b>columnaobjeto</b>	1	2	4
<b>objeto</b>	3	1	3
<b>objetorecorrido</b>	1	2	4
<b>objetorecorrido_columnaobjeto</b>		2	2
<b>recorrido</b>	1		3
<b>usuario</b>	1		6
TOTAL: 9 Tablas			<b>32</b>

Tabla 70 Tablas del Sistema

- bitacora: Se guardará el registro de cada acción dentro del sistema web.
- catalogo: Se guardarán los catálogos del sistema.
- columnacatalogo: Cada registro corresponde a una descripción de algún catálogo registrado.
- columnaobjeto: Cada registro corresponde a la información correspondiente a una columna del catálogo de algún objeto registrado.
- objeto: Se guardarán los objetos del sistema.
- objetorecorrido: Se guardará cada objeto escaneado desde el sistema móvil de algún recorrido.
- objetorecorrido\_columnaobjeto: Se guardará la relación de las columnas, de algún objeto escaneado, seleccionadas por el usuario del sistema móvil.
- recorrido: Se guardará el registro de cada recorrido iniciado desde el sistema móvil.
- usuario: Se guardarán los usuarios del sistema web.

### I.II bitacora

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>id</b>	bigint	19		√			
<b>evento</b>	longtext	2147483647	√		null		
<b>timestamp</b>	datetime	19	√		null		
<b>usuario_id</b>	bigint	19	√		null		▪usuario
<b>objeto_id</b>	bigint	19	√		null		▪objeto
<b>catalogo_id</b>	bigint	19	√		null		▪catalogo

Tabla 71 Tabla bitacora

- id: Índice de la tabla.
- evento: Acción que se realizó.
- timestamp: Fecha cuando se realizó la acción.

- usuario\_id: Índice del usuario al que corresponde.
- objeto\_id: Índice del objeto al que corresponde.
- catalogo\_id: Índice del catálogo al que corresponde.

### I.II.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
id	Primary key	Asc	PRIMARY

Tabla 72 Indexes bitacora

### I.III catalogo

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
id	bigint	19		√		▪columnacatalogo ▪objeto	
descripcion	varchar	50	√		null		
imagen	longblob	2147483647	√		null		

Tabla 73 Tabla catalogo

- id: Índice de la tabla.
- descripcion: Nombre del catálogo.
- imagen: Imagen en bytes del catálogo.

### I.III.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
id	Primary key	Asc	PRIMARY

Tabla 74 Indexes catalogo

### I.IV columnacatalogo

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
id	bigint	19		√		▪columnaobjeto	
tipocampo	int	10	√		null		
titulo	varchar	80					
catalogo_id	bigint	19	√		null		▪catalogo

Tabla 75 Tabla columnacatalogo

- id: Índice de la tabla.
- tipocampo: Tipo de campo de la columna.
- titulo: Nombre de la columna.
- catalogo\_id: Índice del catálogo al que corresponde.

### I.IV.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
id	Primary key	Asc	PRIMARY

<b>catalogo_id</b>	Performance	Asc	FK28925C41FB8691F
--------------------	-------------	-----	-------------------

Tabla 76 Indexes columnacatalogo

### I.V columnaobjeto

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>id</b>	bigint	19		√		▪objetorecorrido_c olumnaobjeto	
<b>descripción</b>	longblob	2147483647	√		null		
<b>columnacatalogo_id</b>	bigint	19	√		null		▪columnacatalogo
<b>objeto_id</b>	bigint	19	√		null		▪objeto

Tabla 77 Tabla columnaobjeto

- id: Índice de la tabla.
- descripción: Información de la columna del objeto.
- columnacatalogo\_id: Índice de la columna del catálogo al que corresponde.
- objeto\_id: Índice del objeto al que corresponde.

### I.V.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
<b>id</b>	Primary key	Asc	PRIMARY
<b>columnacatalogo_id</b>	Performance	Asc	FK93E414D434854B75
<b>objeto_id</b>	Performance	Asc	FK93E414D4AE59397F

Tabla 78 Indexes columnaobjeto

### I.VI objeto

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>id</b>	bigint	19		√		▪columnaobjeto ▪objetorecorrido	
<b>nombre</b>	varchar	50					
<b>catalogo_id</b>	bigint	19	√		null		▪catalogo

Tabla 79 Tabla objeto

- id: Índice de la tabla.
- nombre: Nombre del objeto.
- catalogo\_id: Índice del catálogo al que corresponde.

### I.VI.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
<b>id</b>	Primary key	Asc	PRIMARY
<b>catalogo_id</b>	Performance	Asc	FK8C659169FB8691F
<b>nombre</b>	Must be unique	Asc	nombre

Tabla 80 Indexes objeto

## I.VII objetorecorrido

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>id</b>	bigint	19		√		▪objetorecorrido_columnaobjeto	
<b>nota</b>	longtext	2147483647	√		null		
<b>objeto_id</b>	bigint	19	√		null		▪objeto
<b>recorrido_id</b>	bigint	19	√		null		▪recorrido

Tabla 81 Tabla objetorecorrido

- id: Índice de la tabla.
- nota: Información extra añadida por el usuario del objeto escaneado.
- objeto\_id: Índice del objeto al que corresponde.
- recorrido\_id: Índice del recorrido al que corresponde.

### I.VII.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
<b>id</b>	Primary key	Asc	PRIMARY
<b>recorrido_id</b>	Performance	Asc	FK131DEE8C86746315
<b>objeto_id</b>	Performance	Asc	FK131DEE8CAE59397F

Tabla 82 Indexes objetorecorrido

## I.VIII objetorecorrido\_columnaobjeto

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>ObjetoRecorrido_id</b>	bigint	19					▪objetorecorrido
<b>columna_id</b>	bigint	19					▪columnaobjeto

Tabla 83 Tabla objetorecorrido\_columnaobjeto

- ObjetoRecorrido\_id: Índice del objeto del recorrido al que corresponde.
- columna\_id: Índice de la columna del objeto al que corresponde.

### I.VIII.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
<b>columna_id</b>	Must be unique	Asc	columna_id
<b>columna_id</b>	Performance	Asc	FK62557061352EA2FE
<b>ObjetoRecorrido_id</b>	Performance	Asc	FK62557061C0C47755

Tabla 84 Indexes objetorecorrido\_columnaobjeto

## I.IX recorrido

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>id</b>	bigint	19		√		▪objetorecorrido	

<b>correo</b>	varchar	80				
<b>nota</b>	longtext	2147483647	√		null	

Tabla 85 Tabla recorrido

- id: Índice de la tabla.
- correo: Correo a donde se enviará la información recabada.
- nota: Información extra añadida por el usuario del recorrido en general.

### I.IX.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
<b>id</b>	Primary Key	Asc	PRIMARY
<b>correo</b>	Must be unique	Asc	correo

Tabla 86 Indexes recorrido

### I.IX usuario

Columna	Tipo	Tamaño	Nulls	Auto	Default	Hijos	Padres
<b>id</b>	bigint	19		√			
<b>contrasena</b>	longblob	2147483647	√		null		
<b>email</b>	varchar	50					
<b>nombre</b>	varchar	50	√		null		
<b>perfil</b>	int	10	√		null		
<b>usuario</b>	varchar	15					

Tabla 87 Tabla usuario

- id: Índice de la tabla.
- contrasena: Contraseña del usuario encriptada.
- email: Correo del usuario.
- nombre: Nombre real del usuario.
- perfil: Tipo de usuario.
- usuario: Nombre de usuario para acceder al sistema.

### I.X.I Indexes

Columna(s)	Tipo	Orden	Nombre de Restricción
<b>id</b>	Primary Key	Asc	PRIMARY
<b>email</b>	Must be unique	Asc	email
<b>usuario</b>	Must be unique	Asc	usuario

Tabla 88 Indexes recorrido

## REFERENCIAS BIBLIOGRÁFICAS

[1] Ponencia Intercom, 2009, Torreon Coahuila, "Museando Ando". [Online]. Disponible:  
[http://www.museando-ando.com.mx/ponencia\\_esp.html](http://www.museando-ando.com.mx/ponencia_esp.html)

[2] Dag Svanæs, "Part I – Introduction - The interactive dimension" in *Understanding Interactivity*, 2000.

[3] "A Brief History of Human Computer Interaction Technology". [Online]. Disponible:  
<http://www.cs.cmu.edu/~amulet/papers/uihistory.tr.html>

[4] Encuesta nacional de prácticas y consumo culturales [2010]

[5] "Museum Definition". [Online]. Disponible:  
<http://icom.museum/the-vision/museum-definition/>

[6] "Museografía". [Online]. Disponible:  
<http://museo.itm.edu.co/index.php/areas-de-trabajo/museografia>

[7] Witker, 2008. La museología digital y el museo mexicano. Herramientas museológicas digitales.

[8] Ing. M<sup>a</sup> Soledad Gómez Vílchez, "QR Codes en Museos" 2010. [Online]. Disponible:  
<http://mediamusea.files.wordpress.com/2010/10/qr-code-en-museos.pdf>

[9] Directora Lavert Proyectos Culturales s.l., "La cultura app en los museos", [Online]. Disponible:  
<http://www.lavert.net/Blog/La-cultura-app-en-los-museos>

[10] IDE, University of Karlskrona/Ronneby, "The V Model". [Online]. Disponible:  
[http://www.bucanac.com/documents/The\\_V-Model.pdf](http://www.bucanac.com/documents/The_V-Model.pdf)

[11] "Suscripciones a teléfonos celulares móviles. Serie Anual." [Online], Disponible:  
<http://siemt.cft.gob.mx/SIEM/home.php?seccion=informacion&contenido=10&titulo=Glosario%20de%20t%C3%A9rminos#!prettyPhoto/45/>

[12] "Crecen smartphones en México" [Online], Disponible:  
<http://www.eluniversal.com.mx/articulos/71278.html>

[13] "Android Anatomy and Physiology" [Online], Disponible:  
<http://androidteam.googlecode.com/files/Anatomy-Physiology-of-an-Android.pdf>

[14] David Ehringer, "Design Overview" in *The Dalvik Virtual Machine Architecture*, March 2010, p. 3

[15] "Portable Operating System Interface" [Online], Disponible:  
<http://pubs.opengroup.org/onlinepubs/9699919799/>

[16] "Adobe Reintroduces PhoneGap, Expanding Mobile App Options". [Online]. Disponible:  
<http://ecommercepros.org/blog/wievblog.php?id=150>

[17] "Phonegap". [Online]. Disponible:  
<http://www.phonegap.com>

[18] "IDE de Programación". [Online]. Disponible:

---

[http://www.ecured.cu/index.php/IDE\\_de\\_Programaci%C3%B3n](http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n)

[19] “¿Qué es un Sistema Gestor de Base de Datos?”. [Online]. Disponible:  
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>

[20] G. Alonso, *Web Services: Concepts, Architectures and Applications*. Springer, 2004.

[21] B. V. Kumar, *Web Services*. Tata McGraw-Hill Education, 2004.

[22] “Types of Web Services - The Java EE 6 Tutorial” . [Online]. Disponible:  
<http://docs.oracle.com/javaee/6/tutorial/doc/giqsx.html>

[23] Kenneth E. Kendall and Julie E. Kendall, *Análisis y diseño de sistemas*, Sexta Edición, México: Pearson Educación, 2005.

[24] "Shneiderman's Eight Golden Rules of Interface Design" [Online]. Disponible:  
<http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html>