



**Universidad de las Ciencias Informáticas**

**Junio 2009**



**Sistema para la interacción y control centralizado de  
Dispositivos en aplicaciones web.**

**Autor:** Jeandy Bryan Pineda Oro.

**Tutores:** Msc. Yudenia Ramírez Mastrapa

Msc. Erik de la Vega García

*“Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte.”*

*Leonardo Da Vinci*

# OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

**Título:** Sistema de Personalización de Documentos de Identificación de la República de Cuba.

**Autores:** Jeandy Bryan Pineda Oro

---

**Tutor:**Msc. Yudenia Ramírez Mastrapa

---

**Tutor:** Erik de la Vega García

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo titulado:

Sistema para la interacción y control centralizado de Dispositivos en aplicaciones web. Y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

**Autor:** Jeandy Bryan Pineda Oro

\_\_\_\_\_

**Tutor:**Msc. Yudenia Ramírez Mastrapa

\_\_\_\_\_

**Tutor:** Erik de la Vega García

## DATOS DEL CONTACTO

### Tutores:

Msc. Yudenia Ramírez Mastrapa

- Ingeniera Informática, CUJAE, 2003.
- Categoría docente: Profesora Asistente.
- Máster en Gestión de Proyectos Informáticos.
- Profesora del Departamento de Sistemas Digitales, Facultad 1, Universidad de las Ciencias Informáticas.
- Jefe de departamento de Soluciones de software del Centro de Identificación y Seguridad digital.
- Correo electrónico: yudenia@uci.cu.

MsC. Erik de la Vega García.

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2007.
- Categoría docente: Adiestrado
- Máster en Informática Aplicada.
- Profesor del Departamento de Ingeniería de Software y Práctica Profesional, Facultad 1, Universidad de las Ciencias Informáticas.
- Correo electrónico: edelavega@uci.cu

## AGRADECIMIENTOS

*A la Revolución por ser la responsable de hacer realidad mis sueños creando una universidad de la envergadura de la UCI.*

*A mis abuelos que siempre han confiado en mi trabajo y su confianza hacia mí es inagotable.*

*A mi mamá y mi tío por ser la fuente de inspiración para mí día a día y por ser los responsables de que hoy sea el hombre que soy.*

*A mi novia por ser un ejemplo de perseverancia, responsabilidad y por todo ese amor.*

*A la parte más nueva de mi familia, por acogerme como un hijo y entenderme con mis virtudes y defectos, Ofelia y Belkys.*

*Al padre de esta idea, sin su creatividad hoy no existiera este sistema, Jorge Landrian.*

*A mis tutores por ser un ejemplo profesional y por su compromiso y fe en este trabajo, Yudenia y Erik.*

*Al equipo de desarrollo del proyecto del cual he aprendido mucho tanto en el plano profesional como en el plano personal.*

*Y a todo esos amigos que han compartido con migo tantas líneas de código durante estos cinco años, si a ustedes mismos.*

**DEDICATORIA**

*A esas cinco personas imprescindibles en mi vida: mi abuela, mi abuelo, mi mamá, mi tío y mi novia Linnett.*

---

---

## RESUMEN

El desarrollo continuo de las aplicaciones web y la sustitución sucesiva de las aplicaciones escritorio, se ha encontrado un laberinto de soluciones a la hora de integrar dispositivos de *hardware* con sus aplicaciones en los clientes. Las aplicaciones web se ven restringidas a utilizar las tecnologías web del lado del cliente para la interacción con dispositivos, que se limitan a las bondades del navegador utilizado y no cuentan con mecanismos de acceso a los recursos locales.

El trabajo que se presenta propone el desarrollo de una infraestructura que facilita la integración de dispositivos de *hardware* en aplicaciones web del lado del cliente y garantiza la administración y control centralizado de los mismos, dentro de una red corporativa. Las soluciones propuestas tienen como premisas el uso de estándares, seguridad y adaptabilidad.

Estas se encuentran en uso en el Proyecto Identidad Cuba, para la integración de los dispositivos de *hardware*, tales como impresoras, lectores de huellas, tableta de firma, requeridos para la realización de los procesos que se informatizan en diferentes direcciones del Ministerio del Interior.



---

---

## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b>	<b>1</b>
<b>CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA</b>	<b>6</b>
INTRODUCCIÓN	7
DISPOSITIVOS DE HARDWARE Y SUS SDK	7
TECNOLOGÍAS PARA LA INTERACCIÓN ENTRE EL NAVEGADOR WEB Y LOS RECURSOS LOCALES	7
1.1.1 ACTIVE X	8
1.1.2 APPLETS	9
1.1.3 EXTENSIONES EN LOS NAVEGADORES	9
1.1.3.1 Internet Explorer	9
1.1.3.2 Mozilla Firefox	10
ANÁLISIS DE SOLUCIONES EXISTENTES	10
1.1.4 SISTEMAS PARA EL CONTROL Y ADMINISTRACIÓN DE DISPOSITIVOS	10
TENDENCIAS TECNOLÓGICAS	15
1.1.5 MSF FOR CMMI PROCESS IMPROVEMENT COMO BASE EN EL DESARROLLO DE LA SOLUCIÓN	15
1.1.6 LENGUAJES DE DESARROLLO	16
1.1.7 AMBIENTE DE DESARROLLO	19
1.1.7.1 SharePoint Services como herramienta para gestión documental	19
1.1.7.2 Microsoft Visual Studio Team System 2008	19
1.1.7.3 Microsoft .NET Framework 3.5	20
1.1.7.4 Windows Communication Foundation (WCF)	20
1.1.7.5 ADO.NET Entity Framework	21
1.1.7.6 Sistema Gestor de Base de Datos	21
1.1.7.7 Altova UModel® 2009	22

---

---

<b>CONCLUSIONES</b>	<b>23</b>
<b>2 CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA</b>	<b>24</b>
<b>INTRODUCCIÓN</b>	<b>25</b>
<b>MODELO DE DOMINIO</b>	<b>25</b>
<b>2.1.1 GLOSARIO DE TÉRMINOS DEL MODELO DE DOMINIO</b>	<b>26</b>
<b>DESCRIPCIÓN DEL SISTEMA PROPUESTO</b>	<b>28</b>
<b>PROCESOS</b>	<b>29</b>
<b>2.1.2 PROCESO PARA EXTENDER EL SISTEMA.</b>	<b>30</b>
<b>2.1.3 PROCESO DE USO DEL SISTEMA.</b>	<b>31</b>
<b>ESPECIFICACIÓN DE LOS REQUISITOS FUNCIONALES</b>	<b>31</b>
<b>2.1.4 CATÁLOGO DE REQUISITOS.</b>	<b>31</b>
2.1.4.1 Framework JavaScript	31
2.1.4.2 Complemento para el navegador web	32
2.1.4.3 Servicio local para el manejo de dispositivos.	32
2.1.4.4 Servicio para la obtención de información de la estación de trabajo	32
2.1.4.5 Control centralizado de dispositivos	32
<b>2.1.5 DESCRIPCIÓN REQUISITOS FUNCIONALES</b>	<b>34</b>
2.1.5.1 Framework JavaScript	34
2.1.5.2 Complemento para el navegador web	35
2.1.5.3 Servicio local para el manejo de dispositivos	35
2.1.5.4 Servicio para la obtención de información de la estación de trabajo	37
2.1.5.5 Control centralizado de dispositivos	38
<b>2.1.6 ESPECIFICACIÓN DE REQUISITOS NO FUNCIONALES</b>	<b>40</b>
<b>CONCLUSIONES</b>	<b>42</b>
<b>3 CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA</b>	<b>43</b>

---

---

---

<b>INTRODUCCIÓN</b>	<b>44</b>
<b>ARQUITECTURA DE LA SOLUCIÓN</b>	<b>44</b>
<b>3.1.1 SERVICIO LOCAL PARA EL MANEJO DE DISPOSITIVOS</b>	<b>48</b>
<b>PATRONES DE DISEÑO UTILIZADOS</b>	<b>50</b>
<b>ESPECIFICACIÓN DE CLASES</b>	<b>50</b>
<b>3.1.2 CONTROL CENTRALIZADO DE DISPOSITIVOS</b>	<b>50</b>
3.1.2.1 Diagrama de clases	51
3.1.2.2 Descripción de las clases	52
<b>3.1.3 SERVICIOS DEL SISTEMA</b>	<b>57</b>
<b>MODELO DE DATOS</b>	<b>58</b>
<b>3.1.4 DESCRIPCIÓN DE LAS ENTIDADES FUNDAMENTALES</b>	<b>60</b>
<b>CONCLUSIONES</b>	<b>60</b>
<b>4 CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS</b>	<b>61</b>
<b>INTRODUCCIÓN</b>	<b>62</b>
<b>ESTÁNDARES DE CODIFICACIÓN</b>	<b>62</b>
<b>DIAGRAMA DE DESPLIEGUE</b>	<b>64</b>
<b>4.1.1 DESCRIPCIÓN DEL DIAGRAMA DE DESPLIEGUE</b>	<b>64</b>
<b>DIAGRAMA DE COMPONENTES</b>	<b>65</b>
<b>4.1.2 DESCRIPCIÓN DEL DIAGRAMA DE COMPONENTE</b>	<b>66</b>
<b>4.1.3 MODELO DE PRUEBAS</b>	<b>66</b>
<b>4.1.4 DISEÑO DE CASOS DE PRUEBAS</b>	<b>67</b>
<b>4.1.5 PRUEBAS UNITARIAS (CAJA BLANCA)</b>	<b>70</b>
<b>4.1.6 PRUEBAS DEL SISTEMA</b>	<b>71</b>
4.1.6.1 Iteraciones	72
<b>4.1.7 RESULTADO DE LAS PRUEBAS</b>	<b>74</b>
<b>CONCLUSIONES</b>	<b>76</b>

**CONCLUSIONES** **77**

---

**RECOMENDACIONES** **78**

---

**REFERENCIAS BIBLIOGRÁFICAS** **79**

---

**GLOSARIO** **81**

---

---

---

## ÍNDICE DE FIGURAS

<i>Figura. 1</i> Arquitectura DMS .....	11
<i>Figura. 2</i> Modelo de dominio .....	26
<i>Figura. 3</i> Vista general del sistema.....	28
<i>Figura. 4</i> Proceso de extensión del sistema .....	30
<i>Figura. 5</i> Proceso de uso del sistema .....	31
<i>Figura. 6</i> En Espera .....	35
<i>Figura. 7</i> Desconectado .....	35
<i>Figura. 8</i> Conectado.....	35
<i>Figura. 9</i> Representación conceptual de la estructura del mapeo de los servicios en JavaScript. ....	45
<i>Figura. 10</i> Ejemplo real del mapeo de servicios en el framework JavaScript.....	46
<i>Figura. 11</i> Arquitectura del control centralizado de dispositivos.....	47
<i>Figura. 12</i> Arquitectura del Servicio local para el manejo de dispositivos.....	48
<i>Figura. 13</i> Diagrama de clases del requisito "Gestionar agrupaciones lógicas" .....	51
<i>Figura. 14</i> Diagrama de clases del requisito "Gestionar agrupaciones lógicas" .....	52
<i>Figura. 15</i> Modelo de datos.....	59
<i>Figura. 16</i> Diagrama de despliegue .....	64
<i>Figura. 17</i> Diagrama de componentes Servicio local.....	65
<i>Figura. 18</i> Diagrama de componentes del módulo para Control centralizado de dispositivos.....	66
<i>Figura. 19</i> Código de la prueba unitaria al servicio LogicGroupService .....	71
<i>Figura. 20</i> Resultados de las pruebas .....	75
<i>Figura. 21</i> Resultado de la prueba unitaria del servicio LogicGroupService .....	75
<i>Figura. 22</i> Diagrama de clase del servicio local para el manejo de dispositivos.....	<b>iError! Marcador no definido.</b>
<i>Figura. 23</i> Diagrama de clases del servicio para la obtención de información del hardware de la estación de trabajo .....	<b>iError!</b>

**Marcador no definido.**

## ÍNDICE DE TABLAS

Tabla 1. Sistemas analizados y sus características. ....	15
Tabla 2. Descripción del requisito: Generar petición para la ejecución del servicio. ....	35
Tabla 3. Descripción del requisito: Monitorear de forma visual el estado del servicio local. ....	35
Tabla 4. Descripción del requisito: Permitir el manejo de servicios web ....	36
Tabla 5. Descripción del requisito: Obtener información relativa al hardware de la estación de trabajo.....	38
Tabla 6. Descripción del requisito funcional Permitir "Gestionar agrupaciones físicas" .....	39
Tabla 7. Descripción del requisito funcional Permitir "Gestionar agrupaciones lógicas" .....	40
Tabla 8. Descripción de la clase DGMConnectorBase.....	52
Tabla 9. Descripción de la interfaz ILogicGroupConnector .....	53
Tabla 10 Descripción de la interfaz IPhysicalGroupConnector.....	54
Tabla 11. Descripción de la clase LogicGroupConnector.....	55
Tabla 12 Descripción de la clase PhysicalGroupConnector .....	56
Tabla 13. Descripción del servicio LogicGroupService.....	57
Tabla 14 Descripción del servicio PhisicalGroupService.....	58
Tabla 15. Descripción de la entidad nagrupacionlogica. ....	60
Tabla 16. Descripción de la entidad dagrupacionfisica.....	60
Tabla 17. Diseño de prueba de caja negra del requisito funcional Gestionar agrupación lógica .....	70
Tabla 18. Prueba del sistema resultado de la tercera iteración del requisito funcional "Gestionar agrupaciones lógicas" .....	74
Tabla 19. Descripción del requisito: Manejar respuestas de las solicitudes a través de eventos .....	<b>¡Error! Marcador no definido.</b>
Tabla 20 Descripción del requisito: Permitir la transformación de las peticiones a un formato entendible por el complemento del navegador.....	<b>¡Error! Marcador no definido.</b>
Tabla 21. Descripción del requisito: Configurar llamada de los servicios locales. ....	<b>¡Error! Marcador no definido.</b>
Tabla 22. Descripción del requisito: Permitir la ejecución de servicios web. ....	<b>¡Error! Marcador no definido.</b>
Tabla 23. Descripción del requisito: Aceptar nuevos controladores de dispositivos.....	<b>¡Error! Marcador no definido.</b>
Tabla 24. Descripción del requisito: Permitir manejar dispositivos remotos .....	<b>¡Error! Marcador no definido.</b>
Tabla 25. Descripción del requisito: Aceptar el manejo de dispositivos remotamente .....	<b>¡Error! Marcador no definido.</b>
Tabla 26. Descripción del requisito: Permitir consultar el estado del servicio remotamente .....	<b>¡Error! Marcador no definido.</b>
Tabla 27. Descripción del requisito: Permitir el manejo del servicio remotamente.....	<b>¡Error! Marcador no definido.</b>
Tabla 28. Descripción del requisito: Obtener un identificador partir de la información del Hardware. ....	<b>¡Error! Marcador no definido.</b>

---

---

Tabla 29. Descripción de la clase Core .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 30. Descripción de la clase ServiceManage.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 31. Descripción de la interfaz IAuthorizationComponent .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 32. Descripción de la clase DGMAuthorizationComponent .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 33. Descripción de la clase VirtualFileSystem .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 34. Descripción de la clase ControllerManage.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 35. Descripción de la clase BasicController .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 36. Descripción de la clase BasicServerController .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 37. Descripción de la clase BasicClient.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 38. Descripción de la interfaz IPcInfoClientService.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 39. Descripción de la clase PcInfoClient .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 40. Descripción de la interfaz IPcInfoController .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 41. Descripción de la interfaz IPcInfoController .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 42. Descripción de la clase PhysicalMemory.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 43. Descripción de la clase BaseBoard .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 44. Descripción de la clase Bios.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 45. Descripción de la clase Network .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 46. Descripción de la clase Processor.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 47. Descripción de la clase PcInfo .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 48. Descripción de la clase BaseBoardManage.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 49. Descripción de la clase BiosMange .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 50. Descripción de la clase HardDriveManage.....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 51. Descripción de la clase NetworkManage .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 52. Descripción de la clase PhysicalMemoryManage .....	<b><i>¡Error! Marcador no definido.</i></b>
Tabla 53. Descripción de la clase ProcessorManage .....	<b><i>¡Error! Marcador no definido.</i></b>



# Introducción



## INTRODUCCIÓN

La revolución de internet tiene sus inicios en los años 90, sin embargo, a pesar de que su creación data de décadas anteriores, no es hasta el final del siglo XX que toma popularidad. Su primera fase se ve marcada por la era del punto COM, los sitios web que se publicaban en estos comienzos brindaban la posibilidad a los usuarios de acceder como consumidores pasivos del contenido publicado. La evolución de la web continúa y surge un nuevo concepto el cual trae consigo un gran desarrollo tecnológico, web 2.0. Este nuevo concepto cambia la forma en que los usuarios tradicionales interactuaban con la web y a su vez trae un desarrollo tecnológico que permite la creación de aplicaciones mucho más ricas, elegantes y funcionales. Con estos adelantos muchas empresas comienzan a migrar sus aplicaciones de escritorio a la nueva tendencia de desarrollo orientada a la web, acogándose a todos los beneficios que esta le aporta a sus soluciones.

Al desarrollar soluciones basadas en tecnología web se cuenta con una serie de posibilidades que disminuyen el costo de desarrollo, puesta en marcha y mantenimiento de las aplicaciones. Los usuarios de estas aplicaciones no se ven atados a un sistema operativo o arquitectura de *hardware* en su estación de trabajo, solo se requiere para la correcta ejecución la existencia de un navegador web que cumpla con los estándares establecidos. No es necesaria la presencia física del usuario en alguna localización geográfica específica, ya que este solo necesita una conexión a internet para poder acceder a ella. Al estar todo el código y recursos alojados en un servidor web solo se requiere actualizar una vez, centralizadamente, por lo que el costo de esta tarea comparado con su homóloga en aplicaciones de escritorio es bien bajo.

El desarrollo continuo de las aplicaciones web y el afán de sustituir por completo las aplicaciones escritorio, se ha encontrado un laberinto de soluciones a la hora de integrar dispositivos de *hardware* con sus aplicaciones en los clientes. Las aplicaciones web se ven restringidas a utilizar las tecnologías web del lado del cliente para la interacción con dispositivos. Estas tecnologías están limitadas por su parte a las bondades del navegador web que se esté utilizando, y por razones de seguridad estos no cuentan con un mecanismo de acceso a los recursos locales de forma directa.

Las alternativas con que cuentan los navegadores para la interacción con recursos locales son escasas y no se encuentran estandarizadas, las más formales sólo permitirían el control administrativo de los dispositivos mientras existiera una instancia del navegador funcionando, ya que su ejecución dependería totalmente del navegador web, por lo que no podría ser posible el control total de los dispositivos en cada momento. Las limitantes mencionadas hacen necesario el desarrollo de una solución

que permita la administración en tiempo real de los dispositivos de *hardware* y la interacción con ellos de una forma cómoda y eficiente, que no se vea restringida al uso de un navegador u otro.

Atendiendo la situación planteada y enfocándose en su solución, se ha tomado como problema científico a resolver:

¿Cómo crear un mecanismo para facilitar la comunicación con dispositivos de *hardware* en aplicaciones web del lado del cliente y su control centralizado?

Con el anterior planteamiento del problema científico, el objeto de estudio queda enmarcado en la comunicación y control de dispositivos de *hardware*, quedando como campo de acción, las aplicaciones web que interactúen del lado del cliente con dispositivos de *hardware* y tengan un control centralizado de estos.

En vista a la solución del problema científico planteado, se trazó como objetivo general: Desarrollar un sistema de componentes para la integración de dispositivos de *hardware* en aplicaciones web. Desglosándose este en los siguientes objetivos específicos:

1. Desarrollar el marco teórico de la investigación.
2. Obtener los requisitos funcionales y no funcionales del sistema para la integración de dispositivos.
3. Realizar el modelo de dominio al sistema para la integración de dispositivos.
4. Implementar el sistema de integración de dispositivos.
5. Validar el sistema de integración de dispositivos.

La hipótesis que se ha definido es la siguiente: Si se desarrolla un sistema de componentes para la integración de dispositivos de *hardware* en aplicaciones web, se logrará un mecanismo que facilite la comunicación con dispositivos de *hardware* en aplicaciones web del lado del cliente y su control centralizado.

Para dar cumplimiento a los objetivos específicos se proponen las siguientes tareas:

1. Investigación de los proveedores de *hardwares* y los SDK<sup>1</sup> que ofrecen para la comunicación con sus dispositivos.

---

<sup>1</sup> Software development kit. Kit de desarrollo de software.

2. Estudio de los mecanismos para el manejo de dispositivos en una arquitectura cliente servidor del lado del cliente.
3. Estudio de las tecnologías para el enriquecimiento de funcionalidades nativas del navegador web.
4. Descripción de los requisitos del sistema de componentes para el manejo de dispositivos.
5. Realización de los prototipos de interfaz al sistema de componentes para el manejo de dispositivos.
6. Diseño de la solución del sistema de componentes para el manejo de dispositivos.
7. Desarrollo de un servicio que permita el manejo de dispositivos de *hardware* en la estación de trabajo.
8. Desarrollo de un componente que permita la comunicación entre el navegador web y el servicio local.
9. Desarrollo de un componente para la administración centralizada de los dispositivos de *hardware* distribuidos en las distintas estaciones de trabajo.
10. Aplicación de las pruebas de software al servicio local de manejo de dispositivos, al componente para la comunicación entre el navegador y el servicio local y al sistema para la administración central de dispositivos de *hardware*.
11. Interpretación del los resultados arrojados en las pruebas aplicadas a cada uno de los componentes.

Para el desarrollo de estas tareas se aplican una serie de métodos científicos, tanto teóricos como empíricos.

**Teóricos:**

Histórico-Lógico: Durante la revisión de la bibliografía utilizada para el estudio de los diferentes estándares y protocolos.

Análisis-Síntesis: Durante la definición de las funcionalidades y caracterización de los mecanismos necesarios para la comunicación con dispositivos de *hardware*.

**Empíricos:**

Entrevistas: Se desarrollaron entrevistas a los responsables de desarrollar toda la integración de otros

---

---

proyectos como SAIME <sup>2</sup> con dispositivos de *hardware* y los especialistas en arquitectura del proyecto Identidad Cuba.

**Los posibles resultados a obtener son:**

La obtención de una infraestructura para el manejo de dispositivos de *hardware* desde el navegador web y su administración que proporcione:

Un *Framework* de fácil extensión y entendimiento orientados a la comunicación entre las aplicaciones web del lado del cliente y el servicio de manejo de dispositivo local.

Un complemento para el navegador web que funcione como pasarela entre las tecnologías web del lado del cliente y el servicio local para el manejo de dispositivo.

Un servicio local para la administración y control de la seguridad y comunicación con los dispositivos de *hardware* locales.

Una aplicación web para la administración de dispositivos de *hardware* conectados a las estaciones de trabajos mediante el servicio local de manejo de dispositivos.

El presente trabajo está conformado por cuatro capítulos, distribuidos de la siguiente forma:

**Capítulo 1. Fundamentación teórica.** Cuenta con un respaldo teórico de los conceptos fundamentales para el entendimiento de la solución. Presenta el estudio realizado de las tecnologías y herramientas para el desarrollo del sistema.

**Capítulo 2. Características del sistema.** Se realiza un análisis del dominio actual y los diagramas realizados para su comprensión. Se presentan los requisitos funcionales y no funcionales del sistema conjuntamente con sus descripciones, además de una vista general de la propuesta.

**Capítulo 3. Análisis y Diseño.** Se muestran los diagramas de clases del diseño para mejorar el entendimiento del sistema, conjuntamente el diseño de la base de datos y demás restricciones del sistema.

**Capítulo 4. Implementación y Prueba.** Se analizan los componentes indispensables para el desarrollo de la solución, diagramas de clases y las relaciones entre los distintos componentes que conforman la aplicación. Se presenta la vista de despliegue y las pruebas de software aplicadas a la solución.

---

<sup>2</sup> Sistema Administrativo de identificación migración y extranjería de la república Venezolana.



# Capítulo I

FUNDAMENTACIÓN TEÓRICA

## Introducción

En el presente capítulo se abordarán temas relativos a la comunicación con dispositivos de *hardware*, enfocándose en el uso de estos dispositivos desde tecnologías web. Se hace un estudio de sistemas y aplicaciones existentes especializadas en la comunicación, control y administración de dispositivos, con vistas a obtener conclusiones parciales acerca de posibilidades de desarrollo en esta área.

## Dispositivos de Hardware y sus SDK

El papel que juegan los dispositivos de *hardware* en las estaciones de trabajo es esencial, dotando tanto a estas como a los programas que se ejecutan en ellas, de características que facilitan la integración en todos los campos de la vida cotidiana. Existen grupos de dispositivos de presencia común en la cotidianidad: impresoras, micrófonos, cámaras digitales, dispositivos de almacenamiento; y otro grupo, menos común, pero utilizado con frecuencia dentro de soluciones empresariales, que se orientan a dar solución a tareas más sofisticadas, ejemplo de estos son los lectores de huellas dactilares, tablas electrónicas para captar firmas, relojes para controlar la entrada y salida de personal (1), especializadas en complejos algoritmos de impresión, lectores de tarjetas, entre otros (2).

Existen empresas que se dedican a la distribución de este tipo de dispositivos, como BundesDruckerei, Sony, Canon, Gemalto, Digital Persona, GreenBit, (1) (3), entre otros, y que junto a su *hardware* distribuyen generalmente un *software* llamado SDK, el cual provee al comprador de un kit de desarrollo que permite interactuar desde una perspectiva de desarrollo de *software* con estos dispositivos, al igual que un conjunto de *drivers*<sup>3</sup> que hacen posible la comunicación del equipo con la máquina.

## Tecnologías para la interacción entre el navegador web y los recursos locales

Las aplicaciones web convencionales, cuentan del lado del cliente con código HTML, CSS y JavaScript, estos lenguajes en conjunto conforman el *script*<sup>4</sup> que es descargado por el navegador con el objetivo de interpretarlo y mostrar la interfaz visual de la aplicación. Por cuestiones de seguridad estos lenguajes no cuentan con la capacidad de accesos a los recursos locales de la estación de trabajo donde se ejecuta el navegador. Algunas tecnologías han sido creadas con el objetivo de suplir estas necesidades, compañías como Microsoft cuentan con sus ActiveX, ORACLE con sus Applets y la gran mayoría de los navegadores cuentan con sus complementos.

---

<sup>3</sup> Programa informático que facilita la comunicación del sistema operativo con un periférico comportándose como una abstracción del *Hardware*.

<sup>4</sup> Archivo de procesamiento por lotes.

### 1.1.1 ActiveX

ActiveX según Microsoft: es un estándar que permite a los componentes de *software* interactuar en un entorno de red, independientemente de los idiomas que se utilizaron para crearlos. La mayoría de los usuarios de World Wide Web (WWW) experimentarán la tecnología de ActiveX en forma de controles ActiveX, documentos ActiveX y secuencias de comandos ActiveX (4).

ActiveX según World Wide Web Consortium: es una tecnología desarrollada por Microsoft Corporation para distribuir *software* a través de Internet. Al igual que los Java applets, un ActiveX "Control" se pueden incrustar en una página web, donde aparece típicamente como un gráfico interactivo. Una serie de controles ActiveX están disponibles para el Microsoft Internet Explorer, incluyendo una marquesina, un generador de sonido de fondo, y un ActiveX control que ejecuta Applets de Java. A diferencia de Java, que es un lenguaje de programación independiente de la plataforma, los controles ActiveX son distribuidas como binarios ejecutables, y debe ser compilado para cada equipo de destino y el sistema operativo. (5)

#### Controles ActiveX

Los controles ActiveX, anteriormente conocidos como controles OLE u OCX, son componentes (u objetos) que pueden insertarse en una página web u otro programa para que pueda reutilizar una funcionalidad empaquetada programada por otra persona. Por ejemplo, los controles ActiveX incluidos con Internet Explorer permiten mejorar las páginas web con sofisticadas características de formato y animación.

Una ventaja clave de los controles ActiveX sobre los programas Java y los *plug-in* Netscape es que los controles ActiveX se pueden utilizar igualmente en programas escritos en muchos lenguajes de programación, incluidos todos los lenguajes de programación y de base de datos de Microsoft (4).

#### Documentos ActiveX

Cuando se está explorando con un explorador web compatible con ActiveX, como Internet Explorer, los documentos ActiveX permiten abrir un programa con sus propias barras de herramientas y menús disponibles. Esto significa que puede abrir archivos que no son de HTML, como archivos de Microsoft Excel o Microsoft Word, con un explorador web compatible con ActiveX (4).

#### Secuencias de comandos ActiveX

Las secuencias de comandos ActiveX son compatibles con los lenguajes de secuencia de comandos más conocidos, incluidos Microsoft Visual Basic Script y JavaScript. Las secuencias de

comandos ActiveX se pueden utilizar para integrar el comportamiento de varios controles ActiveX o programas Java del explorador o servidor web, ya que amplían su funcionalidad (4).

### 1.1.2 Applets

Un *applet Java* es un subprograma escrito en el lenguaje de programación Java. Estos pueden correr en un navegador web utilizando la *Java Virtual Machine* (JVM), o en el AppletViewer de Sun.

Entre sus características se pueden mencionar un esquema de seguridad que permite que los *applets* que se ejecutan en el equipo no tengan acceso a partes sensibles (por ejemplo, no pueden escribir archivos), a menos que se le otorguen los permisos necesarios en el sistema; la desventaja de este enfoque es que la entrega de permisos es engorrosa para el usuario común, y esto juega en contra de uno de los objetivos de los *Java applets*: proporcionar una forma fácil de ejecutar aplicaciones desde el navegador web.

En Java un *applet*, es un programa que puede incrustarse en un documento HTML; es decir en una página web. Cuando un Navegador carga una página web que contiene un *applet*, este se descarga en el navegador web y comienza a ejecutarse. Esto permite crear programas que cualquier usuario puede ejecutar con tan solo cargar la página web en su navegador (6).

### 1.1.3 Extensiones en los navegadores

Las extensiones son pequeñas aplicaciones que se instalan en los navegadores con el objetivo de mejorar y extender las funcionalidades con las que se cuenta en su forma básica. Existen extensiones de diferentes tipos, orientadas a manejar la seguridad en la web, para aumentar la compatibilidad con una determinada red social o tan simplemente como para aumentar la vistosidad de la interfaz visual del navegador. El líder en el campo de las extensiones es el Mozilla Firefox, que cuenta actualmente con aproximadamente 5000 extensiones, que pueden ser descargadas desde la web en *addons.mozillaorg* (7). Otros navegadores que cuentan con una sólida infraestructura para el desarrollo de extensiones son Opera, Internet Explorer y Safari.

#### 1.1.3.1 Internet Explorer

Las extensiones de navegador se introdujeron en la versión 5 del Microsoft Internet Explorer, y permiten a los desarrolladores agregar funciones al navegador y mejorar la interfaz de usuario de una manera que no está directamente relacionada con el contenido visible de páginas web. Esta clase de extensibilidad o *addon* incluyen funcionalidades que aumenta la experiencia de navegación a los usuarios, mediante herramientas que hacen la navegación mucho más cómoda y funcional.



### 1.1.3.2 Mozilla Firefox

Las Extensiones son pequeños *add-ons* (complementos, agregados) que añaden nuevas funcionalidades a las aplicaciones Mozilla, tales como Firefox y Thunderbird. Las extensiones permiten añadir a dichas aplicaciones cualquier elemento, desde un botón para una barra de herramientas hasta características totalmente nuevas. Permiten personalizar completamente la aplicación para ajustarla a las necesidades de cada usuario, sin aumentar de forma significativa el tamaño de la misma.

#### Análisis de soluciones existentes

### 1.1.4 Sistemas para el control y administración de dispositivos

#### Axiom DMS (8)

El sistema *Axiom DMS* es una aplicación de Servidor que controla todo tipo de dispositivos utilizados por los diferentes módulos de la *Suite Axiom Studio*. *Axiom DMS* le permite configurar la comunicación con terminales de registro y con los dispositivos conectados a ellas y cuenta con un mecanismo de consulta y registro de transacciones y accesos a través de dichas terminales.

*Axiom DMS* puede correr bajo sistemas operativos Windows XP, Windows Vista o Windows Server 2008 y utiliza una base de datos centralizada sobre Microsoft SQL Server 2000, Microsoft SQL Server 2005 o Microsoft SQL Server 2008.

El servidor *Axiom DMS* controla múltiples terminales que a su vez controlan dispositivos como pueden ser lectores biométricos, lectores de tarjeta, puertas, sirenas, luces, etc. La configuración de cada dispositivo es administrada por *Axiom DMS*, almacenada en la base de datos y transmitida a las terminales. Todas las transacciones y eventos generados por los dispositivos son recibidos por el servidor *Axiom DMS* y almacenados en la base de datos. A través de la red, clientes remotos pueden conectarse al servidor *Axiom DMS* para administrar o controlar dispositivos y terminales.

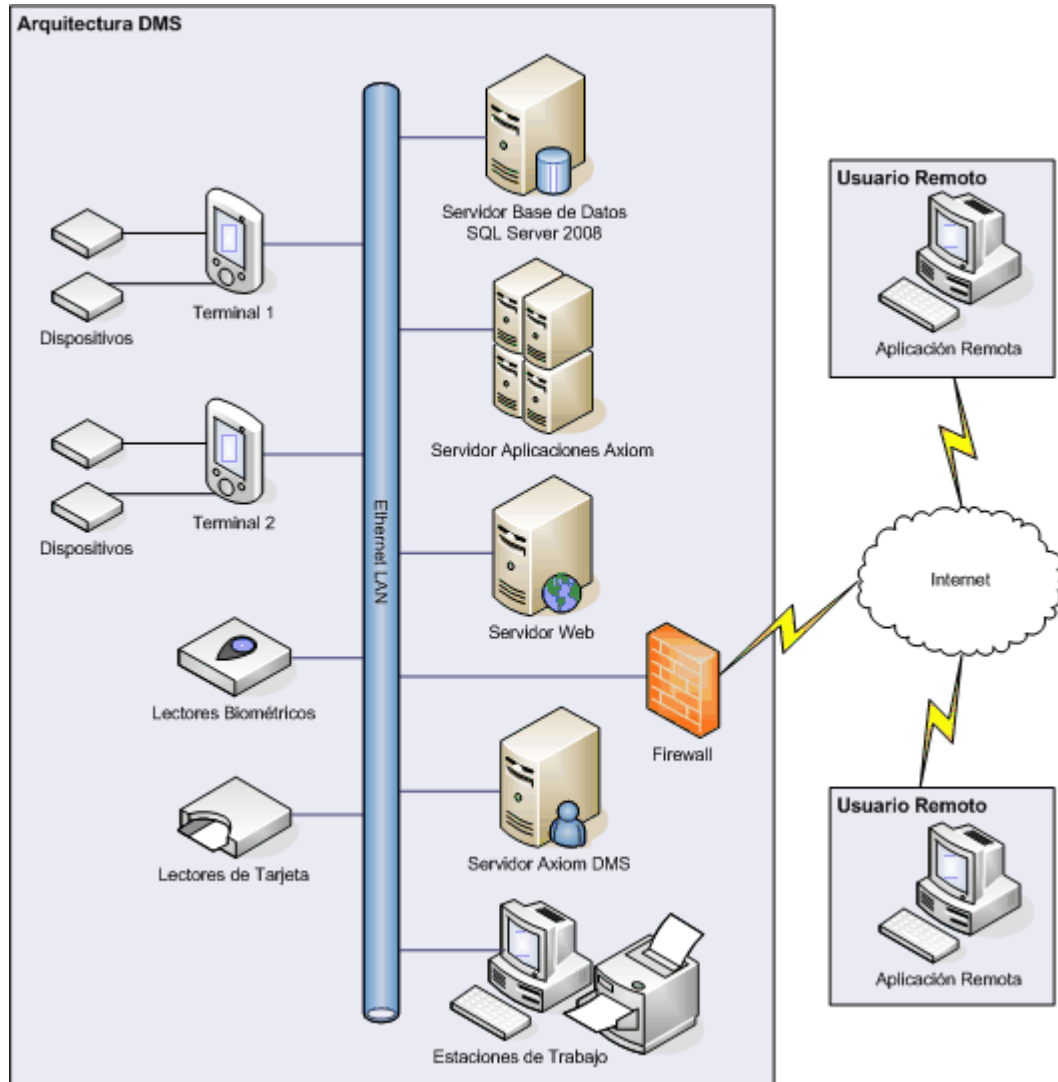


Figura. 1 Arquitectura DMS

Algunas de las características del sistema Axiom DMS son:

- Puede controlar un número prácticamente ilimitado de dispositivos.
- Administra la configuración y comportamiento de cada dispositivo y almacena dicha información en una base de datos centralizada.
- Permite configurar tipos de eventos dependiendo del dispositivo y utilizando diversos parámetros.
- Almacena información de transacciones y eventos generados en los dispositivos en una base de datos centralizada.
- Cuenta con un módulo de diagnóstico en línea que permite ver el estado de los dispositivos de manera remota.
- Administra información biométrica de usuarios y distribuye dicha información a las terminales necesarias.
- Administra listas de accesos y distribuye dicha información a las terminales.
- Cuenta con un módulo de descarga simultánea de archivos e información a terminales.
- Cuenta con un servidor TCP/IP que permite a clientes autorizados monitorear el funcionamiento o interactuar con los dispositivos de manera remota.
- Cuenta con una interfaz gráfica y amigable.

### **SAGEM MEMS - Software de Administración y Enrolamiento Centralizado (9)**

Funcionalidad del Sistema MEMS:

- Interfaz gráfica de fácil uso. Opera en una estación de trabajo basada en PC con Sistema Operativo Windows.
- Base de datos central de huellas dactilares: Microsoft Access, SQL Server, Oracle.
- Consola de configuración, administración en red, y reportes para múltiples terminales MorphoAccess.
- Arquitectura Cliente/Servidor disponible con versión SQL Server y Oracle.
- Perfiles de Usuario: acceso en modalidad de Administrador y Operador.
- Administración de Terminales MorphoAccess
  - Configuración remota.
  - Sincronización de base de datos con el servidor.

- Recuperación de *logs* (registro de auditoría): los accesos de personas son almacenados localmente en un archivo dedicado, el *log* incluye todos los datos para aplicaciones de acceso y control de asistencia.
- Manejo de múltiples grupos de terminales.
- Generación de Reportes.
- Funcionalidad de Enrolamiento de huellas dactilares y registro de información básica del personal
- Capacidad de regular la calidad de la imagen.
- Control de Accesos de acuerdo a zonas horarias.

### Framework para integrar el acceso a dispositivos

Como parte de los procesos que se desarrollan en muchos sistemas y aplicaciones, es necesaria la captura de datos a través de distintos dispositivos externos especializados en cada tipo de captación, quienes, luego de ser analizadas las distintas variantes presentes en el mercado a partir de las características y las funcionalidades que brindan, haciéndolas coincidir con los requerimientos para los que serán destinados, comienza un proceso de asimilación a través del cual se persigue el objetivo de lograr una perfecta integración de sus funcionalidades principales con las aplicaciones del sistema. (10)

Debido a las características de los distintos negocios, al requerir el uso de dispositivos externos, surge la necesidad de crear un *framework* para la interacción con dispositivos, de forma sencilla y transparente a la implementación de los negocios, a través de interfaces que describan su comportamiento. Para el diseño de este *framework* de acceso a dispositivos se debe cumplir un conjunto de características fundamentales, que parten del análisis de las principales necesidades de los sistemas y aplicaciones que lo puedan usar: (10)

**Integrable:** Se debe integrar de forma fácil y segura a cualquier aplicación o módulo, de forma tal que pueda ser usado en los distintos procesos de captación definidos en un sistema.

**Extensible:** Es necesario que el *framework* permita la incorporación de nuevos dispositivos en la medida que vayan surgiendo las necesidades para nuevos módulos, o la aceptación de otros equipos con las mismas características que el actual pero de otros fabricantes, de forma tal que se puedan hacer reemplazos sin hacer cambios en las aplicaciones. (10)

**Componentes Gráficos:** Algunos tipos de dispositivo de captura, requieren de retroalimentación gráfica en la interfaz de la aplicación, para lo que se debe proveer de los componentes gráficos necesarios para su fácil integración, y de esa forma cumplir con los requisitos de crear una interfaz amigable. (10)

**Configurable:** Debe ser posible a través de configuraciones, preferentemente en el estándar XML, definir los dispositivos habilitados para la captura, para que no sea necesario recompilar la aplicación en el caso que sea necesario usar otros dispositivos de otros fabricantes para realizar las mismas u otras operaciones de captura, y además se pueda crear una herramienta en la propia aplicación que sea capaz de gestionar este archivo de configuración de forma sencilla. (10)

### **Análisis**

Los sistemas y *frameworks* analizados en el acápite anterior cuentan con una serie de ventajas y desventajas las que han sido estudiadas, haciendo un aporte importante para sentar las bases del desarrollo de una nueva solución integral para la administración y comunicación con dispositivos de *hardware*.

El sistema Axiom DMS mantiene un control en tiempo real de todos los dispositivos distribuidos bajo su control en una red basada en el sistema Operativo Windows, cuentan con módulos para la interacción biométrica y sistema de actualización de sus terminales. Este sistema cuenta con una licencia propietaria y con un alto costo monetario, además de estar altamente arraigado a los estándares del Sistema Operativo Windows, y no cuenta con ningún módulo para la interacción con dispositivos desde aplicaciones web en sus clientes.

SAGEM MEMS es una muy completa plataforma biométrica para el enrolamiento y control de acceso. Cuenta con módulos para el control y manejo de cada uno de sus dispositivos, generando reportes basados en acciones, fecha y tiempo. Es capaz de administrar remotamente cada uno de los terminales bajo su control. Este sistema posee aplicaciones de escritorio para su uso, por lo que es imposible insertarlo en una arquitectura web, al no contar con ninguna función para la interacción con dispositivos desde aplicaciones web, además de poseer un alto costo de adquisición y una pobre gestión de roles. Axiom DMS cuenta con ciertas características las cuales se tomarán en cuenta en esta investigación, como son el concepto de administración centralizada de los clientes, la generación de reportes y su módulo para el diagnóstico de los dispositivos bajo su control.

El análisis del **Framework para integrar el acceso a dispositivos** muestra una herramienta con un alto grado de extensibilidad, que no cuenta con un módulo de administración de dispositivos, pero sí con una arquitectura bien definida y perfilada la cual puede ser una base para el diseño de la interacción de

dispositivos de cualquier sistema en el caso de no poder acoplar directamente este *framework*<sup>5</sup> por razones de tecnologías o arquitectura.

Sistemas	Control en tiempo real de dispositivos.	Reportes de uso.	Posibilidades de uso en aplicación web.	Extensible y modificable.
<b>Axiom DMS</b>	x	x		
<b>SAGEM MEMS</b>	x	x		
<b>Framework para integrar el acceso a dispositivos.</b>			x	x

Tabla 1. Sistemas analizados y sus características.

El estudio realizado ha llevado a la conclusión de la necesidad de desarrollar un sistema integral para el manejo y control de dispositivos orientado a la arquitectura web, que provea tanto de herramientas para la administración de los mismos como para la interacción desde aplicaciones web. Con la unión de conceptos manejados por las aplicaciones estudiadas anteriormente y la introducción de nuevas funcionalidades y características en un nuevo sistema, se lograría el desarrollo de un producto aplicable a un gran número de soluciones, y que sería perfectamente comercializable y reduciría las importaciones de un sistema para el control de dispositivos al país.

### Tendencias tecnológicas

#### 1.1.5 MSF for CMMI Process Improvement como base en el desarrollo de la solución

Con la creación del producto *Visual Studio Team System* se han producido a su vez dos metodologías para guiar o mejorar de forma más organizada el proceso de desarrollo de *software* en los proyectos productivos como lo son *MSF for Agile Software Development* para el trabajo en entornos que emplean metodologías ágiles, y *MSF<sup>6</sup> for CMMI Process Improvement* para el trabajo en entornos con el modelo CMMI.

<sup>5</sup> La palabra inglesa **framework** define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

<sup>6</sup> Microsoft Solutions *Framework* por sus siglas en Inglés

La metodología por tener la ventaja de estar integrada con el *Visual Studio Team System* brinda la posibilidad de contar con una serie de plantillas y guías adaptadas y orientadas a los roles definidos en cada una (11).

La integración armoniosa de MSF en *Team System* soporta el desarrollo iterativo rápido con el aprendizaje continuo y el refinamiento. Las iteraciones pequeñas le permiten reducir el margen de error en sus estimaciones y proporcionar información rápida acerca de la exactitud de sus planes de proyecto ya que cada iteración debería formar parte estable de todo el sistema. (11).

MSF para Mejora de Procesos CMMI define cinco fases para el ciclo de vida del proyecto que encapsulan conjuntos de *Workstreams*<sup>7</sup> y actividades. Cada fase concluye con un punto de control. Cada punto de control brinda una oportunidad para autorizar el trabajo en el proyecto o para cancelar o suspender el proyecto. El objetivo dentro de una fase es proporcionar las respuestas a cuestiones de dirección y también las respuestas con los datos del proyecto transparente, obtenida a través del día a día las operaciones de la organización de ingeniería de *software*. (11)

### 1.1.6 Lenguajes de Desarrollo

#### ASP.Net

ASP.NET es un modelo de desarrollo web unificado que incluye los servicios necesarios para crear aplicaciones web empresariales con el código mínimo. ASP.NET forma parte de *.NET Framework* y al codificar las aplicaciones ASP.NET tiene acceso a las clases en *.NET Framework*. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el *Common Language Runtime* (CLR), entre ellos *Microsoft Visual Basic*, *C#*, *JScript.NET* y *J#*. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del *Common Language Runtime*, seguridad de tipos, herencia, entre otros (12).

---

<sup>7</sup> Flujo de trabajo.

ASP.NET incluye:

- Marco de trabajo de página y controles
- Compilador de ASP.NET
- Infraestructura de seguridad
- Funciones de administración de estado
- Configuración de la aplicación
- Supervisión de estado y características de rendimiento
- Capacidad de depuración
- Marco de trabajo de servicios web XML
- Entorno de *host* extensible y administración del ciclo de vida de las aplicaciones
- Entorno de diseñador extensible

## C#

C# es un lenguaje de programación que se ha diseñado para generar diversas aplicaciones que se ejecutan en *.NET Framework*. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C (13).

## LINQ

*Language-Integrated Query* (LINQ) es un conjunto de características en *Visual Studio 2008* que agrega eficaces capacidades de consulta a la sintaxis de los lenguajes C# y *Visual Basic*. LINQ incluye patrones estándar y de fácil aprendizaje para consultar y actualizar datos, y su tecnología se puede extender para utilizar potencialmente cualquier tipo de almacén de datos. *Visual Studio 2008* incluye ensamblados de proveedores para LINQ que habilitan el uso de LINQ con colecciones de *.NET Framework*, bases de datos de *SQL Server*, conjuntos de datos de *ADO.NET* y documentos XML. (14)



## JavaScript

JavaScript es un lenguaje de *scripts*, interpretado, multiplataforma y parcialmente orientado a objetos. Fue creado por Netscape específicamente para su uso en el desarrollo de sitios web. Actualmente es un estándar mantenido por el ECMA<sup>8</sup> (15).

El código JavaScript puede enlazarse o añadirse a las páginas web proporcionando un control total y dinámico sobre ellas. Además, también permite controlar hasta cierto punto las aplicaciones que lo ejecutan, habitualmente navegadores (16).

## XML

*Extensible Markup Language* es un formato sencillo, formado por texto flexible derivado de SGML (ISO 8879). Originalmente diseñado para satisfacer los desafíos de las publicaciones electrónicas a gran escala, XML está siendo usado cada vez más para el intercambio de una amplia variedad de datos en la web y otros entornos. (17)

## XUL

XUL es el lenguaje XML para interfaces de usuario de Mozilla. Permite crear potentes aplicaciones multiplataforma que pueden ejecutarse con conexión a Internet o sin ella. Estas aplicaciones son fácilmente personalizables con texto alternativo, gráficos, y diseños por lo que pueden ser fácilmente instalados o traducidos para diversos *softwares* (16).

## XHTML

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML, XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. (18)

## CSS

Hojas de Estilo en Cascada (*Cascading Style Sheets*), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. (19)

---

<sup>8</sup> Organización internacional basada en membresías de estándares para la comunicación y la información.

## 1.1.7 Ambiente de desarrollo

### 1.1.7.1 SharePoint Services como herramienta para gestión documental

*Microsoft Windows SharePoint Services 3.0* es una tecnología versátil, para aumentar la eficiencia de los procesos de negocio y mejorar la productividad del equipo, proporciona una plataforma base para construir aplicaciones basadas en web de negocio que puede flexionar y escalar fácilmente para satisfacer las necesidades cambiantes y crecientes del negocio.

#### Características

- Proporciona un espacio de trabajo único para los equipos, para coordinar los horarios, organizar documentos, y participar en las discusiones dentro de la organización y en la extranet.
- Fácil gestión de documentos, y contribuye a garantizar la integridad con funciones mejoradas, incluida la opción de exigir comprobación de documentos antes de la edición, la posibilidad de ver las revisiones pasadas y restaurar las versiones anteriores, y la capacidad de establecer documentos de seguridad específicos.
- Proporcionar foros creativos para una lluvia de ideas, la creación de bases de conocimiento, o simplemente la recopilación de información de una forma fácil de editar el formato con nuevas plantillas de aplicación de los *blogs* (también conocidos como *weblogs*) y los *wikis* (sitios web que pueden ser rápidamente editados por los miembros del equipo-no conocimientos técnicos especiales requeridos) (20).

### 1.1.7.2 Microsoft Visual Studio Team System 2008

Microsoft Visual Studio Team System 2008 es una plataforma para herramientas del ciclo de vida del desarrollo de software extensible, integrado y productivo que ayuda a los equipos de desarrollo de software mediante la mejora de las comunicaciones y la colaboración durante todo el proceso de desarrollo. Contiene un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y hace más sencilla la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de *.NET Framework*, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios web XML (21).

### 1.1.7.3 Microsoft .NET Framework 3.5

*Microsoft .NET Framework 3.5* contiene nuevas características que se agregan de forma incremental a *.NET Framework 2.0* y *3.0* e incluye *.NET Framework 2.0 Service Pack 1* y *.NET Framework 3.0 Service Pack 1*. Este agrega de forma incremental las nuevas características de *.NET Framework 3.0*. Por ejemplo, los conjuntos de características de *Windows Workflow Foundation (WF)*, *Windows Communication Foundation (WCF)*, *Windows Presentation Foundation (WPF)* y *Windows CardSpace*. Además, *.NET Framework 3.5* contiene una serie de características nuevas en distintas áreas tecnológicas que se han agregado como nuevos ensamblados para evitar cambios destacados. Algunas de estas características son:

- Integración total de LINQ (*Language Integrated Query*) y del reconocimiento de los datos. Esta nueva característica le permitirá escribir código en idiomas habilitados para LINQ para filtrar, enumerar y crear proyecciones de varios tipos de datos SQL, colecciones, XML y conjuntos de datos usando la misma sintaxis.
- ASP.NET AJAX le permite crear experiencias web más eficaces, más interactivas y con un gran índice de personalización que funcionan con los exploradores más usados.
- Nueva compatibilidad con el protocolo web para generar servicios WCF, como por ejemplo AJAX, JSON, REST, POX, RSS, ATOM y distintos estándares WS nuevos.
- Compatibilidad total con las herramientas de Visual Studio 2008 para WF, WCF y WPF, incluida la nueva tecnología de servicios habilitados para flujos de trabajo.
- Nuevas clases en la biblioteca de clases base (BCL) de *.NET Framework 3.5* que tratan numerosas solicitudes de cliente comunes (22).

### 1.1.7.4 Windows Communication Foundation (WCF)

*Windows Communication Foundation (WCF)* es el modelo de programación unificado de Microsoft para generar aplicaciones orientadas a servicios. Permite a los programadores generar soluciones con transacciones seguras y de confianza, que se integren en diferentes plataformas y que interoperen con las inversiones existentes.

WCF simplifica el desarrollo de aplicaciones conectadas a través de un nuevo modelo de programación orientado a servicios. WCF admite muchos estilos de desarrollo de aplicaciones distribuidas proporcionando una arquitectura superpuesta. En su base, la arquitectura de canal de WCF proporciona primitivos asíncronos de paso de aprobación de mensajes sin tipo. Generados sobre esta base están las

funciones de protocolos para un intercambio de datos de transacción seguro y fiable, así como una amplia variedad de opciones de codificación y transporte.

#### **1.1.7.5 ADO.NET Entity Framework**

ADO.NET *Entity Framework* está diseñado para permitir a los programadores crear aplicaciones de acceso a datos programando con un modelo de la aplicación conceptual en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y mantenimiento que se necesita para las aplicaciones orientadas a datos. Las aplicaciones de *Entity Framework* ofrecen las siguientes ventajas:

Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones. Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento. Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación. Los programadores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes. Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento. La compatibilidad con Language-Integrated Query proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

#### **1.1.7.6 Sistema Gestor de Base de Datos**

Oracle Database 11g R2 ofrece un rendimiento líder en la industria, escalabilidad, seguridad y fiabilidad en la elección de la opción clúster o de un solo servidor con Windows, Linux o UNIX. También cuenta con funciones globales para administrar fácilmente el procesamiento de las transacciones más exigentes, inteligencia empresarial, y aplicaciones de gestión de contenidos. Este cuenta con una amplia gama de opciones que lo sitúan en el número uno dentro de las bases de datos que requieran cumplir con altos niveles de seguridad, rendimiento y disponibilidad.

Las características relacionadas con la capacidad de administración en Oracle Database 11g están diseñadas para ayudar a las empresas a administrar fácilmente los *grids* de infraestructura y cumplir exitosamente con las expectativas del nivel de servicio al usuario. Las nuevas características y mejoras en la capacidad de administración ayudan a aumentar la productividad de los DBA y reducir los costos de

administración, mientras aumenta el desempeño, la escalabilidad y la seguridad de sus aplicaciones comerciales durante las 24 horas.

Oracle protege los sistemas de todas las causas comunes relacionadas con el tiempo de baja planificado y no planificado, con inclusión de las fallas en el sitio, en el almacenamiento y en el servidor, así como de una de las causas más importantes: el error humano. En Oracle Database 11g se incluyen además elementos que favorecen la administración eficientemente de la carga de información durante todo el ciclo de vida de las bases de datos al optimizar los recursos de almacenamiento de acuerdo con la demanda.

Brinda una plataforma segura y escalable para el acceso confiable y rápido a todo tipo de información utilizando interfaces estándares como XML, *spatial*, multimedia, imágenes médicas, y tecnologías semánticas, que representa un área de gran crecimiento para muchas empresas.

#### **1.1.7.7 Altova UModel® 2009**

Altova UModel® 2009 es un punto de partida para el desarrollo de software, permite el diseño visual de modelos de aplicaciones en UML y genera código Java, C#, o Visual Basic .NET, así como documentación del proyecto. Realiza ingeniería inversa de programas existentes pasando los diagramas UML, luego afina los diseños y completa con la generación de código. Entre sus características presenta:

- Soporte para los 14 tipos de diagramas UML
- Modelado de esquemas XML en diagramas UML
- Diagramas de proceso de negocio (BPMN)
- Generación de código fuente en lenguajes Java, C#, y VB.NET
- Ingeniería inversa de código fuente y ficheros binarios Java, C# y VB.NET
- Sincronizado de modelo y código a través de ingeniería de ida y vuelta
- Crea diagramas de secuencia desde el código fuente de la ingeniería inversa
- Generación de documentación personalizable de proyecto
- Compartir subproyectos para colaboración o reutilización
- Capas de diagramas con visibilidad selectiva
- Hyperlinks entre diagramas, documentos, o páginas web
- Soporte para intercambio de modelos XMI 2.1
- Integración con sistemas de control de versiones
- API extendida para permitir manipulaciones externas

- Estrecha integración con Visual Studio y Eclipse

### **Conclusiones**

El desarrollo de este capítulo brinda un estudio de las tecnologías que posibilitan la interacción con dispositivos de *hardware* vinculados a una estación de trabajo desde el navegador web, haciendo uso de la tecnología web del lado del cliente. Se detallaron tecnologías y estándares que facilitan el desarrollo, modelación y control adecuado de la nueva solución. Se abordan las distintas tendencias, y se conforma una base clara con la retroalimentación de proyectos y soluciones semejantes existentes, los cuales cuentan con un alto grado de confiabilidad.

Se aborda la necesidad de la implementación de un nuevo sistema que cuenta con la posibilidad de administrar dispositivos de *hardware* en una arquitectura web y que permita la interacción con estos. Es importante aclarar que la implementación de este sistema facilitaría la migración y desarrollo de aplicaciones que necesiten el uso de dispositivos de *hardware* en su negocio sobre una arquitectura web.



# Capítulo II

CARÁCTERÍSTICAS DEL SISTEMA

## **Introducción**

En este capítulo se desarrolla el Modelo de Dominio actual, dejando claramente definido para su entendimiento una descripción por cada una de las entidades que interviene en este modelo. Se detalla la propuesta del sistema, haciendo una descripción de los módulos que lo componen y un diagrama que provee una vista general de este. Se presenta una lista de los requisitos que el sistema tiene que cumplir, obteniendo con estos una concepción general de las funcionalidades.

## **Modelo de Dominio**

El modelo de dominio es la representación visual de las clases conceptuales u objetos del mundo real que reportan interés para esta investigación o proyecto. Este modelo al que también se le puede conocer como modelo conceptual, utiliza el estándar UML para su representación. Para su construcción se modela un diagrama de clase compuesto por objetos del dominio o clases conceptuales, relaciones entre estas clases y los atributos que las componen.

La determinación de realizar un modelo de dominio se toma a partir de no tenerse una claridad suficiente en los procesos como para modelar un negocio, no tener fronteras claramente determinadas para este o no existir una claridad de quiénes son los que inician el negocio y quiénes se benefician con este.



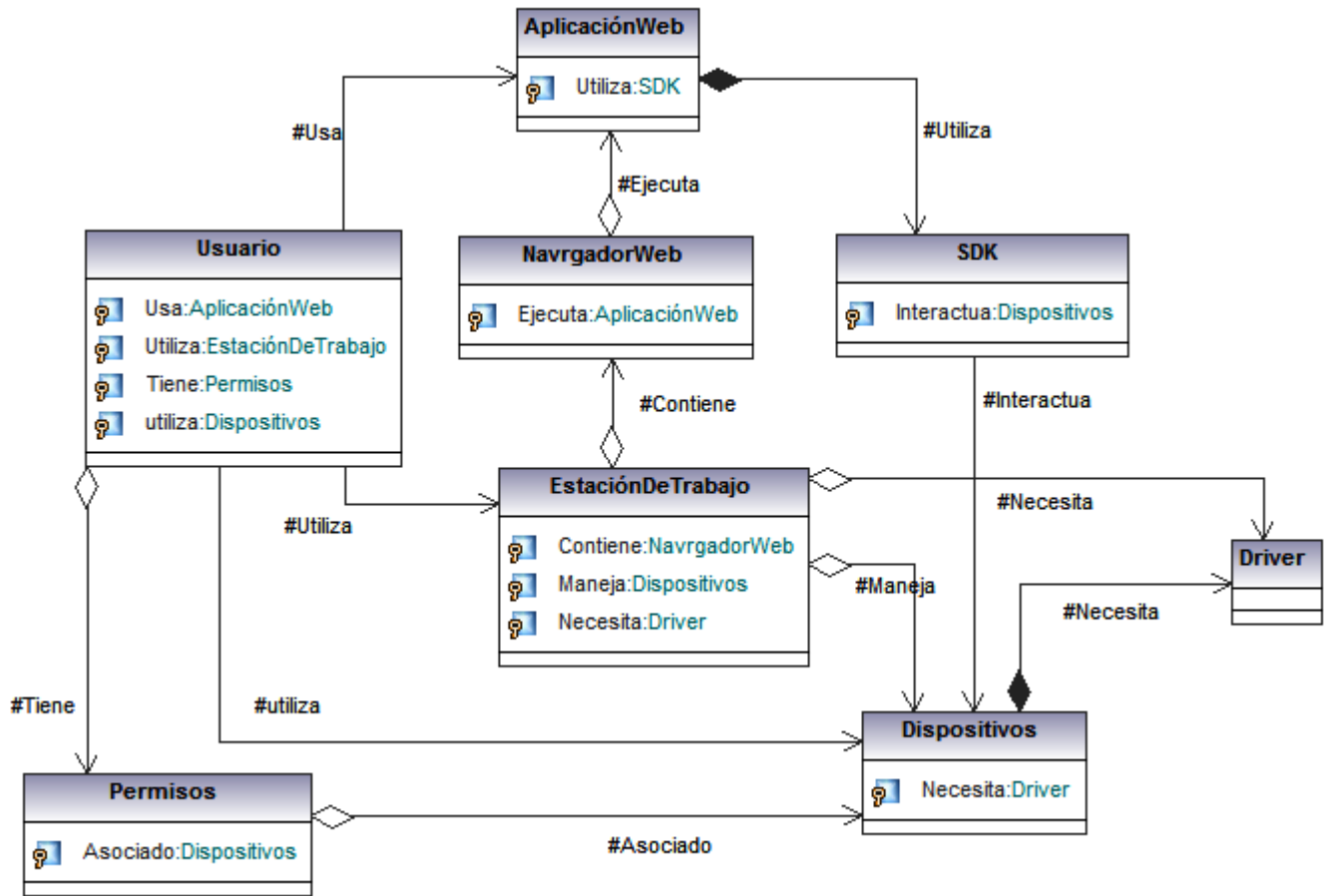


Figura. 2 Modelo de dominio

### 2.1.1 Glosario de Términos del Modelo de Dominio

En este acápite se dará una explicación para el mejor entendimiento cada una de las clases que conforma el modelo conceptual.

#### Usuario:

Es la persona encargada de interactuar con la aplicación web y acceder a sus funcionalidades, esta persona tiene un grupo de permisos, que lo asocian al uso de un grupo de dispositivos.

### **Estación de trabajo:**

Es la computadora destinada a ser usada por los usuarios para interactuar con las aplicaciones web mediante el navegador instalado en la misma, puede poseer un grupo de dispositivos conectados a ella.

### **Navegador web:**

Es una aplicación que se encuentra instalada en la estación de trabajo que permite la ejecución de una aplicación web, esta aplicación puede variar en dependencia de las preferencias del usuario.

### **Aplicación web:**

La aplicación web es el contenido que se ejecuta dentro del navegador web, que está destinado para la interacción con el usuario a través de las funcionalidades que brindan, y que en determinados casos requieren el uso de dispositivos.

### **Dispositivos:**

Los dispositivos son accesorios de *hardware* que se conectan a la estación de trabajo para dotarla de nuevas funcionalidades, que pueden ser utilizadas por aplicaciones, como pueden ser, capturar huellas, capturar fotos, imprimir documentos. Estos dispositivos varían en dependencia de su marca, funcionalidades, distribuidor, necesitando *drivers* para poder ser reconocidos en la estación de trabajo a la que se conectan.

### **Driver:**

Es un paquete de software que proveen los fabricantes, incluido junto con los dispositivos, y que se requiere para establecer comunicación con los dispositivos en función de su correcto funcionamiento por los sistemas operativos en las estaciones de trabajos.

### **Permisos:**

Los permisos cuentan con la información acerca de qué usuario puede o no usar un determinado dispositivo.

**SDK:**

Es un conjunto de herramientas para la comunicación con determinado dispositivo, que provee el fabricante para una cómoda integración de su producto con las aplicaciones de software en general.

**Descripción del sistema propuesto**

El sistema que se propone está compuesto por cuatro módulos, estos trabajan como un todo, integradamente, permitiendo la interacción y control de los dispositivos de *hardware* de una forma cómoda y elegante, basada en un estándar de desarrollo que facilita la extensión de la aplicación sin grandes esfuerzos por parte de los desarrolladores.

**Vista general del sistema**

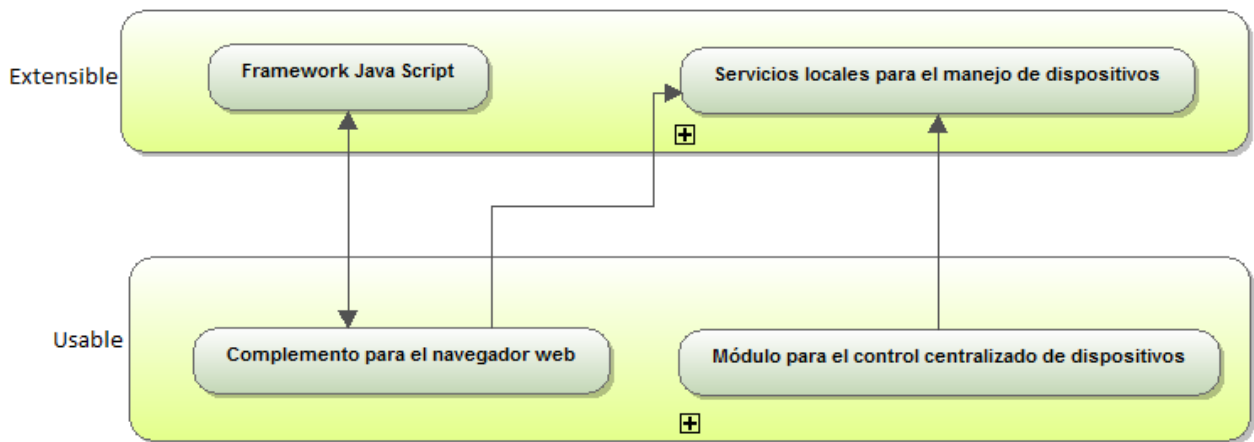


Figura. 3 Vista general del sistema

**Framework JavaScript:**

Este componente cuenta con un grupo de clases que facilitan al desarrollador web la interacción con servicios publicados en la estación de trabajo donde se esté ejecutando la aplicación. El desarrollo de este es similar a la implementación de AJAX, una tecnología bien conocida y estandarizada, la cual es de conocimiento de un gran porcentaje de programadores web, por lo que el desarrollador no necesita asimilar un alto grado de conocimiento haciendo la curva de aprendizaje bien baja.

### **Complemento para el navegador web:**

El *Framework* descrito anteriormente, implementado totalmente en el lenguaje JavaScript, por sí solo no puede acceder a los servicios que se encuentran publicados en la estación de trabajo. El complemento es el encargado de establecer la comunicación del *Framework* JavaScript con los servicios locales, funcionando este como un intermediario que es capaz de obtener peticiones desde el *Framework*, interpretarlas, conformar y ejecutar llamadas a los servicios web, el cual en caso de la devolución de datos de parte del servicio deben tomarlos y enviarlos al *Framework* JavaScript.

### **Servicio local para el manejo de dispositivo**

Este servicio cuenta con dos partes fundamentales: el motor de manejo para los servicios asociados a los dispositivos y el grupo de clientes y controladores encargados de interactuar directamente con un dispositivo de *hardware*. La primera parte gestiona las peticiones que provienen desde el complemento del navegador que han sido solicitadas por el *Framework* JavaScript, estas solicitudes traen consigo información la cual conlleva al uso de un cliente determinado para interactuar con los dispositivos. La segunda parte es el grupo de clientes y controladores, los clientes constituyen la interfaz para el uso de los dispositivos, quienes utilizan los controladores para su manejo.

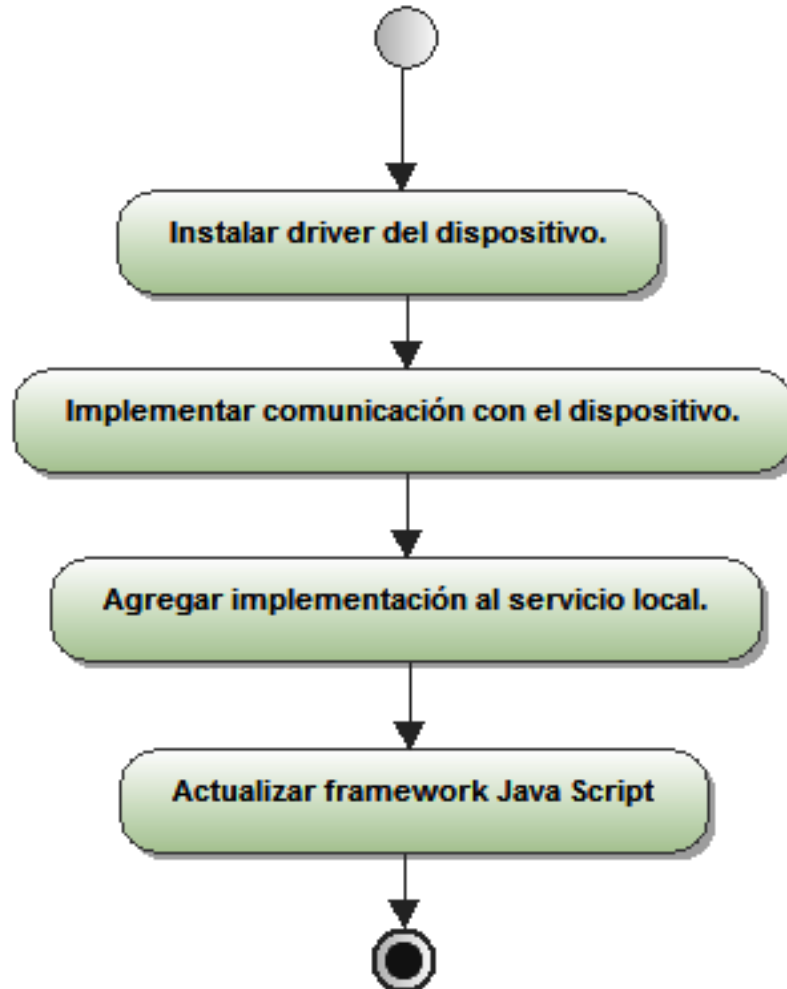
### **Módulo para el control centralizado de dispositivos**

Este módulo brinda una interfaz visual que muestra en tiempo real el estado de los distintos dispositivos que se encuentran distribuidos en las estaciones de trabajos, permitiendo su administración y mostrar reportes de comportamiento.

### **Procesos**

En los siguientes procesos se mostrará la interacción del desarrollador con el sistema en los dos posibles escenarios, aumentando las funcionalidades de sistema para el uso de nuevos dispositivos y en el uso del *Framework* para comunicarse con los dispositivos.

**2.1.2 Proceso para extender el sistema.**



*Figura. 4 Proceso de extensión del sistema*

**Descripción.**

El sistema necesitaría ser extendido a la hora de agregar un nuevo dispositivo para su manejo. El desarrollador instalaría los *driver* del dispositivo, luego crearía una nueva funcionalidad para interactuar con este, esta funcionalidad la agrega al servicio local para el manejo de dispositivos y por último actualiza el *Framework* JavaScript agregándole la configuración necesaria para el uso del dispositivo desde este, quedando completamente actualizado el sistema.

### 2.1.3 Proceso de uso del sistema.

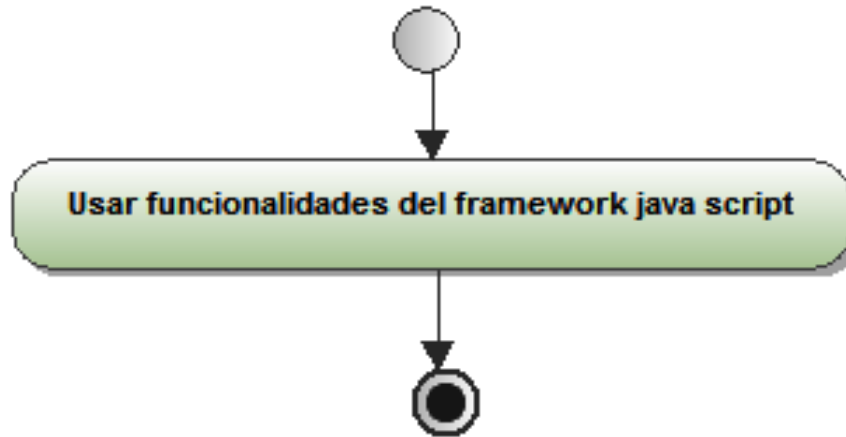


Figura. 5 Proceso de uso del sistema

#### Descripción.

El proceso de uso del sistema es bien sencillo, el desarrollador solo debe usar las funcionalidades del *Framework JavaScript* para interactuar con el dispositivo de *hardware* que desee, ya que todas las demás partes del sistema se encuentran listas para su uso.

#### Especificación de los requisitos funcionales

La *IEEE Standard Glossary of Software Engineering Terminology* define un requerimiento como Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo (23).

### 2.1.4 Catálogo de requisitos.

#### 2.1.4.1 Framework JavaScript

- RF 1. Generar petición para la ejecución del servicio.
- RF 2. Manejar respuestas de las solicitudes a través de eventos.
- RF 3. Brindar funciones de código para la ejecución de los servicios locales.
- RF 4. Permitir la transformación de las peticiones a un formato entendible por el complemento del navegador.
- RF 5. Transformar las respuestas del complemento del navegador al formato de las respuestas de solicitudes.
- RF 6. Configurar llamada de los servicios locales.

#### **2.1.4.2 Complemento para el navegador web**

RF 1. Monitorear de forma visual el estado del servicio local.

RF 2. Permitir la ejecución de servicios web.

RF 2.1 Manejar solicitudes de ejecución de servicios web.

RF 2.2 Ejecutar un servicio web basado en la información contenida en la solicitud.

RF 2.3 Transformar las respuestas de un servicio web al formato de respuesta de una solicitud.

#### **2.1.4.3 Servicio local para el manejo de dispositivos.**

RF 1. Permitir el manejo de servicios web.

RF 1.1 Ejecutar servicios en demanda.

RF 1.2 Detener servicios luego de un tiempo inactivo.

RF 2. Aceptar nuevos controladores de dispositivos.

RF 3. Permitir manejar dispositivos remotos.

RF 3.1 Establecer comunicación con servicio remoto.

RF 3.2 Establecer comunicación con controlador remoto.

RF 4. Aceptar el manejo de dispositivos remotamente.

RF 5. Permitir consultar el estado del servicio remotamente.

RF 6. Permitir el manejo del servicio remotamente.

#### **2.1.4.4 Servicio para la obtención de información de la estación de trabajo**

RF 1. Obtener información relativa al *hardware* de la estación de trabajo.

RF 1.1 De cada una de las tarjetas de red de la estación de trabajo.

RF 1.2 De la placa base de la estación de trabajo

RF 1.3 De los procesadores de la estación de trabajo

RF 1.4 De los discos duros de la estación de trabajo

RF 1.5 De la BIOS de la estación de trabajo.

RF 2. Obtener un identificador partir de la información del *Hardware*.

#### **2.1.4.5 Control centralizado de dispositivos**

RF 1. Permitir mostrar el estado actual de los dispositivos de las estaciones de trabajo.

RF 2. Permitir gestionar Agrupaciones lógicas.

RF 2.1 Permitir adicionar una agrupación lógica.

- a) Identificador de la agrupación lógica.
  - b) Nombre de la agrupación lógica.
  - c) Identificador de la agrupación lógica padre.
- RF 2.2 Permitir eliminar una agrupación lógica dado el identificador de la agrupación.
- RF 2.3 Permitir modificar una agrupación lógica dado un id.
- a) Nombre de la agrupación lógica.
  - b) Identificador de la agrupación lógica padre.
- RF 2.4 Permitir mostrar los datos de una agrupación lógica dado su identificador.
- a) Nombre de la agrupación lógica.
  - b) Nombre de la agrupación lógica padre.
- RF 3. Permitir gestionar agrupaciones físicas.
- RF 3.1 Permitir adicionar una agrupación física
- a) Identificador de la agrupación física.
  - b) Nombre de la agrupación física.
  - c) Identificador de la agrupación física padre.
- RF 3.2 Permitir eliminar una agrupación física dado el identificador de la agrupación.
- RF 3.3 Permitir modificar una agrupación física dado un id.
- a) Nombre de la agrupación física.
  - b) Identificador de la agrupación física padre.
- RF 3.4 Permitir mostrar los datos de una agrupación física dado su identificador.
- a) Nombre de la agrupación física.
  - b) Nombre de la agrupación física padre.
- RF 4. Permitir Gestionar Estaciones de trabajo.
- RF 5. Permitir validar la veracidad de una sesión.
- RF 6. Gestionar mantenimiento de los dispositivos.
- RF 6.1 Definir el tiempo necesario de mantenimiento por tipo de dispositivo.
- RF 7. Permitir el envío de alertas por correo de los dispositivos que necesitan mantenimiento.
- RF 8. Permitir habilitar los servicios de manejos de dispositivos remotamente.
- RF 9. Permitir deshabilitar los servicios de manejos de dispositivos remotamente.
- RF 10. Registrar mantenimiento finalizado con los siguientes datos:
- a) Código de mantenimiento del dispositivo
- RF 11. Mostrar reportes



RF 11.1 Mostrar reportes del estado de los servicios.

- a) Listado de los servicios.
- b) Estado
  - (a) Habilitado
  - (b) Deshabilitado

RF 11.2 Mostrar reportes del comportamiento de los dispositivos con los siguientes datos:

- a) Hora de uso del dispositivo
- b) Tipo de dispositivo.
- c) Si fue satisfactorio su uso.

RF 11.3 Mostrar reportes de dispositivos que necesitan mantenimiento con los siguientes datos:

- a) Estación de trabajo
- b) Descripción del dispositivo
- c) Código de mantenimiento del dispositivo

## 2.1.5 Descripción Requisitos funcionales

### 2.1.5.1 Framework JavaScript

RF 1. Generar petición para la ejecución del servicio.

<b>Propósito</b>	Conformar una petición con todos los datos necesarios para solicitar una llamada a un servicio web.	
<b>Roles</b>	Sistema.	
<b>Precondiciones</b>	Debe haber sido solicitada la ejecución de un servicio web.	
<b>Entidades tratadas</b>	Entidad	Atributos
	ServiceConfig	<ul style="list-style-type: none"> <li>• serviceUrl</li> <li>• serviceParameter</li> </ul>
	Petición	<ul style="list-style-type: none"> <li>• id</li> <li>• infoServiceCall</li> <li>• event</li> </ul>
<b>Descripción</b>	<p>1.1. El sistema crea una nueva petición de llamada a un servicio web con los siguientes datos:</p> <p>2.1.5.1.1 Identificador de la petición.</p> <p>2.1.5.1.2 Dirección del servicio web.</p>	

	2.1.5.1.3 Evento que se lanzará al obtener la respuesta. 1.2. Se coloca la nueva solicitud en la cola de solicitudes.
<b>Validaciones</b>	La URL del servicio web debe estar bien formada.
<b>Postcondiciones</b>	Se actualizó la lista de solicitudes.
<b>Prototipo</b>	-----

Tabla 2. Descripción del requisito: Generar petición para la ejecución del servicio.

### 2.1.5.2 Complemento para el navegador web

RF 1. Monitorear de forma visual el estado del servicio local.

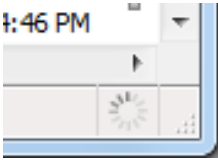

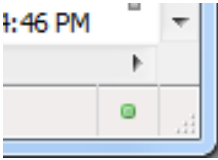
<b>Propósito</b>	Muestra el estado de la disponibilidad del servicio local.	
<b>Roles</b>	Sistema.	
<b>Precondiciones</b>	El complemento debe estar instalado en el navegador.	
<b>Entidades tratadas</b>	Entidad	Atributos
<b>Descripción</b>	1.1 Ejecuta una llamada al servicio. 1.1.1 Mostrar icono de espera 1.2 Si recibe respuesta antes de 30 segundos mostrar el icono que representa conectado. 1.3 Si no responde luego de los 30 segundos muestra el icono desconectado. 1.4 Se repite el proceso cada 40 segundos.	
<b>Validaciones</b>	La respuesta debe ser antes de los 30 segundos.	
<b>Postcondiciones</b>	Queda actualizado el icono representativo de la disponibilidad del servicio local.	
<b>Prototipo</b>	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>Figura. 6 En Espera</p> </div> <div style="text-align: center;">  <p>Figura. 7 Desconectado</p> </div> <div style="text-align: center;">  <p>Figura. 8 Conectado</p> </div> </div>	

Tabla 3. Descripción del requisito: Monitorear de forma visual el estado del servicio local.

### 2.1.5.3 Servicio local para el manejo de dispositivos

RF 1. Permitir el manejo de servicios web.

<b>Propósito</b>	Permitir la ejecución y detección de los servicios.	
<b>Roles</b>	Sistema	
<b>Precondiciones</b>	Debe existir el fichero de configuración del sistema.	
<b>Entidades tratadas</b>	Entidad	Atributos
	Services	ServiceHost
<b>Descripción</b>	<p>1.1. Leer de la configuración del sistema la configuración de cada uno de los servicios para obtener los siguientes datos:</p> <ul style="list-style-type: none"> <li>a) Nombre del ensamblado del cliente.</li> <li>b) Nombre del ensamblado del controlador.</li> <li>c) Si está activo o no el servicio.</li> </ul> <p>1.2. Conformar los servicios usando los clientes y los controladores respectivamente de cada uno de los servicios activos.</p> <p>1.3. Iniciar cada uno de los servicios conformados.</p> <p>1.4. Detener cada uno de los servicios conformados.</p>	
<b>Validaciones</b>	Los nombres de los ensamblados que conforman el servicio deben coincidir con ficheros físicos.	
<b>Postcondiciones</b>	Deben estar disponibles los servicios web.	
<b>Prototipo</b>		

Tabla 4. Descripción del requisito: Permitir el manejo de servicios web

**2.1.5.4 Servicio para la obtención de información de la estación de trabajo**

RF 1. Obtener información relativa al *hardware* de la estación de trabajo.

<b>Propósito</b>	Obtener información acerca del <i>hardware</i> de la estación de trabajo.	
<b>Roles</b>	Sistema	
<b>Precondiciones</b>		
<b>Entidades tratadas</b>	Entidad	Atributos
	BaseBoard	<ul style="list-style-type: none"> <li>• Manufacturer</li> <li>• Model</li> <li>• Version</li> <li>• SerialNumber</li> </ul>
	Bios	<ul style="list-style-type: none"> <li>• Version</li> <li>• SerialNumber</li> <li>• Manufacturer</li> </ul>
	HardDrive	<ul style="list-style-type: none"> <li>• Model</li> <li>• Type</li> <li>• SerialNo</li> </ul>
	Network	<ul style="list-style-type: none"> <li>• Mac</li> <li>• Domain</li> <li>• Description</li> <li>• HostName</li> </ul>
	Processor	<ul style="list-style-type: none"> <li>• Name</li> <li>• ProcessorId</li> <li>• Manufacturer</li> <li>• Speed</li> </ul>
	PhysicalMemory	<ul style="list-style-type: none"> <li>•</li> </ul>
<b>Descripción</b>	<p>1.1. Obtener información de la placa base de la estación de trabajo.</p> <ol style="list-style-type: none"> <li>a) Manufacturer</li> <li>b) Model</li> <li>c) Version</li> <li>d) SerialNumber</li> </ol> <p>1.2. Obtener información de la BIOS de la estación de trabajo.</p> <ol style="list-style-type: none"> <li>a) Version</li> <li>b) SerialNumber</li> <li>c) Manufacturer</li> </ol> <p>1.3. Obtener un listado de los discos duros de la estación de trabajo, con los siguientes datos:</p>	

	<p>a) Model b) Type c) SerialNo</p> <p>1.4. Obtener un listado de las tarjetas de red de la estación de trabajo, con los siguientes datos: a) Mac b) Domain c) Description d) HostName</p> <p>1.5. Obtener un listado de los procesadores de la estación de trabajo, con los siguientes datos: a) Name b) ProcessorId c) Manufacturer d) Speed</p> <p>1.6. Obtener un listado de las tarjetas de memoria de la estación de trabajo, con los siguientes datos:</p>
<b>Validaciones</b>	
<b>Postcondiciones</b>	Se obtuvieron los datos del <i>hardware</i> de las estaciones de trabajo.
<b>Prototipo</b>	-----

Tabla 5. Descripción del requisito: Obtener información relativa al hardware de la estación de trabajo

### 2.1.5.5 Control centralizado de dispositivos

RF 1. Permitir gestionar agrupaciones físicas.

<b>Propósito</b>	Gestionar nuevas agrupaciones físicas	
<b>Roles</b>	Administrador	
<b>Precondiciones</b>	El administrador debe haber adquirido los permisos necesarios.	
<b>Entidades tratadas</b>	Entidad	Atributos
	Agrupación física	<ul style="list-style-type: none"> <li>• Identificador</li> <li>• Id de la agrupación padre</li> <li>• Descripción</li> </ul>

<b>Descripción</b>	<p>1.1 Si se desea adicionar una agrupación física padre.              1.1.1 Seleccionar la opción "Adicionar".                  1.1.1.1 Mostrar una nueva pantalla con los siguientes datos:                      (a) Descripción</p> <p>1.2 Si no se desea adicionar una agrupación física como padre              1.2.1 Seleccionar la agrupación padre a la cual pertenecerá.              1.2.2 Seleccionar la opción "Adicionar".                  1.2.2.1 Mostrar una nueva pantalla con los siguientes datos:                      (a) Muestra el nombre del padre                      (b) Permitir introducir la descripción.</p> <p>1.3 Si se desea modificar una agrupación.              1.3.1 Seleccionar una agrupación.              1.3.2 Seleccionar la opción "modificar".                  1.3.2.1 Mostrar una nueva ventana con los siguientes datos.                      (a) Permitir modificar el id                      (b) Permitir modificar la descripción.</p> <p>1.4 Si se desea eliminar una agrupación.              1.4.1 Seleccionar la agrupación a eliminar.              1.4.2 Seleccionar la opción "Eliminar".                  1.4.2.1 Mostrar una nueva ventana con los datos de la agrupación a elimina:                      (a) Identificador.                      (b) Descripción.                  1.4.2.2 Seleccionar la opción "Aceptar" para eliminar la agrupación.                  1.4.2.3 Seleccionar la opción "Cancelar" para anular la operación.</p>
<b>Validaciones</b>	
<b>Postcondiciones</b>	Se crearon, modificaron o eliminaron las ubicaciones físicas.
<b>Prototipo</b>	-----

Tabla 6. Descripción del requisito funcional Permitir "Gestionar agrupaciones físicas"

RF 12. Permitir gestionar Agrupaciones lógicas.

<b>Propósito</b>	Gestionar nuevas agrupaciones lógicas	
<b>Roles</b>	Administrador	
<b>Precondiciones</b>	El administrador debe haber adquirido los permisos necesarios.	
<b>Entidades tratadas</b>	Entidad	Atributos
	Agrupación física	<ul style="list-style-type: none"> <li>• Identificador</li> <li>• Id de la agrupación padre</li> </ul>

	• Descripción
<b>Descripción</b>	<p>1.5 Si se desea adicionar una agrupación lógica padre.</p> <p>1.5.1 Seleccionar la opción "Adicionar".</p> <p>1.5.1.1 Mostrar una nueva pantalla con los siguientes datos:</p> <p>(a) Descripción</p> <p>1.6 Si no se desea adicionar una agrupación lógica como padre</p> <p>1.6.1 Seleccionar la agrupación padre a la cual pertenecerá.</p> <p>1.6.2 Seleccionar la opción "Adicionar".</p> <p>1.6.2.1 Mostrar una nueva pantalla con los siguientes datos:</p> <p>(a) Muestra el nombre del padre</p> <p>(b) Permitir introducir la descripción.</p> <p>1.7 Si se desea modificar una agrupación.</p> <p>1.7.1 Seleccionar una agrupación.</p> <p>1.7.2 Seleccionar la opción "modificar".</p> <p>1.7.2.1 Mostrar una nueva ventana con los siguientes datos.</p> <p>(a) Permitir modificar el id</p> <p>(b) Permitir modificar la descripción.</p> <p>1.8 Si se desea eliminar una agrupación.</p> <p>1.8.1 Seleccionar la agrupación a eliminar.</p> <p>1.8.2 Seleccionar la opción "Eliminar".</p> <p>1.8.2.1 Mostrar una nueva ventana con los datos de la agrupación a eliminar:</p> <p>(a) Identificador.</p> <p>(b) Descripción.</p> <p>1.8.2.2 Seleccionar la opción "Aceptar" para eliminar la agrupación.</p> <p>1.8.2.3 Seleccionar la opción "Cancelar" para anular la operación.</p>
<b>Validaciones</b>	
<b>Postcondiciones</b>	Se crearon, modificaron o eliminaron las ubicaciones lógicas.
<b>Prototipo</b>	-----

Tabla 7. Descripción del requisito funcional Permitir "Gestionar agrupaciones lógica".

### 2.1.6 Especificación de requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estos requisitos hacen al producto atractivo, usable, rápido o confiable. Estos están vinculados a los requisitos funcionales, una vez que se conozca lo que el sistema debe hacer es posible determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

#### Requisitos de Usabilidad

- Facilidad de uso de parte de los desarrolladores de software.
- El sistema debe ser fácil de extender por parte de los desarrolladores.

### **Requisito de Rendimiento**

- El sistema debe garantizar la mayor velocidad posible en el procesamiento de la información.
- Optimizar el flujo de información que se trasmite por la red, utilizando eficientemente el canal.
- El sistema deberá hacer uso racional de los recursos del navegador web, evitando la sobrecarga de este.
- Hacer uso racional de los recursos de *hardware* en las estaciones de trabajo.

### **Requisitos de Fiabilidad**

- Los servicios deben ser tolerantes al fallo, capaces de recuperarse ante errores.
- Los dispositivos deben poder reiniciarse y recuperarse en caso de existir errores en su funcionamiento.

### **Requisitos de Soporte**

- Debe ser fácil de implantar en distintos ambientes sin necesidad de realizar grandes esfuerzos.



### **Requisitos de Software**

- Sistema Operativo: Windows XP o superior.
- .NET Framework v3.5 SP1 o superior.

### **Conclusiones**

En este capítulo se desarrolló la propuesta de solución del sistema, teniendo en cuenta el modelado de dominio. Se muestra un listado de todos los requisitos del sistema tanto funcional como no funcional y su descripción completa. Se presentaron dos procesos donde se define el uso del sistema de parte del desarrollador, mostrando la sencillez del flujo.



# Capítulo III

ANÁLISIS Y DISEÑO DEL SISTEMA

### **Introducción**

El objetivo fundamental del presente capítulo es mostrar detalladamente los elementos utilizados en el proceso de desarrollo de cada uno de los componentes del sistema. Realizar una descripción detallada de la arquitectura, por la cual se rige el desarrollo haciendo énfasis en los puntos cruciales de esta, y la presentación del modelo de datos a utilizar con su descripción correspondiente.

### **Arquitectura de la solución**

La arquitectura de la solución cuenta de dos módulos arquitectónicamente significativos, el sitio para el control administrativo del sistema y el servicio de manejo de dispositivos que se encuentra alojado en cada estación de trabajo. Estos dos módulos poseen características puntuales de complejidad, que los convierten en los pilares para la definición de la arquitectura, como por ejemplo, la garantía de la seguridad y extensibilidad que deben proveer.

El sistema de administración debe ser una ampliación web flexible, la cual no contenga ataduras entre sus capas y sea fácil de extender. Para el desarrollo de la presentación se utiliza el Modelo Vista Controlador como patrón arquitectónico, debido a las facilidades y desacoplamiento que brinda este patrón. El acceso a datos o capa de implementación de la persistencia está separado totalmente de todo el negocio del sistema permitiendo así el uso de diferentes fuentes de datos sin necesidades de grandes esfuerzos.

El servicio para manejo de los dispositivos cuenta con varias capas, entre las cuales se destaca la capa de seguridad. Esta capa cuenta con un interceptor de solicitudes http el cual se encarga de evaluar la procedencia de las peticiones y el nivel de permiso de su ejecución antes de que sean ejecutadas por los servicios que corren en su núcleo. Realiza verificaciones de cada uno de los ensamblados que lo compone y su autenticidad, evitando una sustitución malintencionada de componentes de la aplicación.

## Framework JavaScript

El *Framework JavaScript* es el primer eslabón de la cadena que hace posible la interacción con los dispositivos que se encuentran acoplados a la estación de trabajo. Este implementa la lógica necesaria para poder ejecutar los servicios que se encuentran fuera del dominio de la aplicación, su funcionamiento se limita a la conformación de los paquetes que representan las llamadas a los servicios web, y el monitoreo de la recepción de respuesta. El *framework* está basado en el mismo estilo que utiliza la conocida librería de JavaScript JQuery para la ejecución de llamadas AJAX, facilitando el aprendizaje y la utilización de este. Dos ficheros conforman a este *framework*, uno contiene toda la lógica de su implementación y el otro se utiliza para representar los métodos y URL de los servicios a ejecutar. Para evitar utilizar directamente estas largas cadenas de texto, las cuales son sensibles a la introducción de errores, se sustituyen por alias que son representados dentro de este fichero, estos son más amigables de tratar dentro de nuestra aplicación.

La representación de estos alias se hace mediante un objeto JSON el cual forma una estructura jerárquica en forma de arbol la cual queda de la siguiente manera:

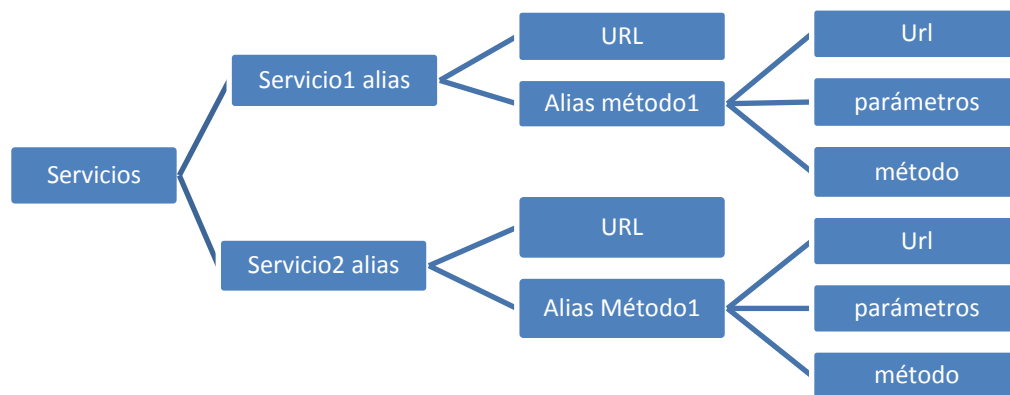


Figura. 9 Representación conceptual de la estructura del mapeo de los servicios en JavaScript.

La siguiente imagen muestra un ejemplo real de cómo quedaría conformado el objeto JSON ya representado en el lenguaje JavaScript en un ambiente real.

```

Service = {
  pcInfo: {
    url: 'http://localhost:8608/rest/PcInfoClient/',

    Mac: {
      url: 'GetNetworks',
      parameter: '?value=[]&otrovalue=[]',
      method: 'GET'
    },
    Image: {
      url: 'GetImage',
      parameter: '',
      method: 'GET'
    },
    Procesador: {
      url: 'getProcesor',
      parameter: ['si','no'],
      method : 'POST'
    },
    test: {
      url: 'TestList',
      parameter: ['nombre'],
      method: 'POST'
    },
    PcId: {
      url: 'GetPcId',
      parameter: [],
      method: 'GET'
    }
  }
}

```

Figura. 10 Ejemplo real del mapeo de servicios en el framework JavaScript

### Complemento para el navegador web

Como apoyo al *framework* JavaScript se encuentra un complemento instalado en el navegador web, este tiene la responsabilidad de ejecutar realmente las llamadas a los servicios web. El complemento toma las peticiones realizadas por el *framework*, las interpreta y las transforma en una petición http. La respuesta que se obtiene como resultado de la llama al servicio web la formatea y la coloca en la cola de respuesta para que el *framework* pueda acceder a ella. El trabajo de este complemento se resume en un *proxy* de llamadas http.

### Control centralizado de dispositivos

Este sistema cuenta con un grupo de funcionalidades administrativas, que permiten controlar de forma centralizada todos los dispositivos de la organización, conociendo su estado y localización en cualquier momento.

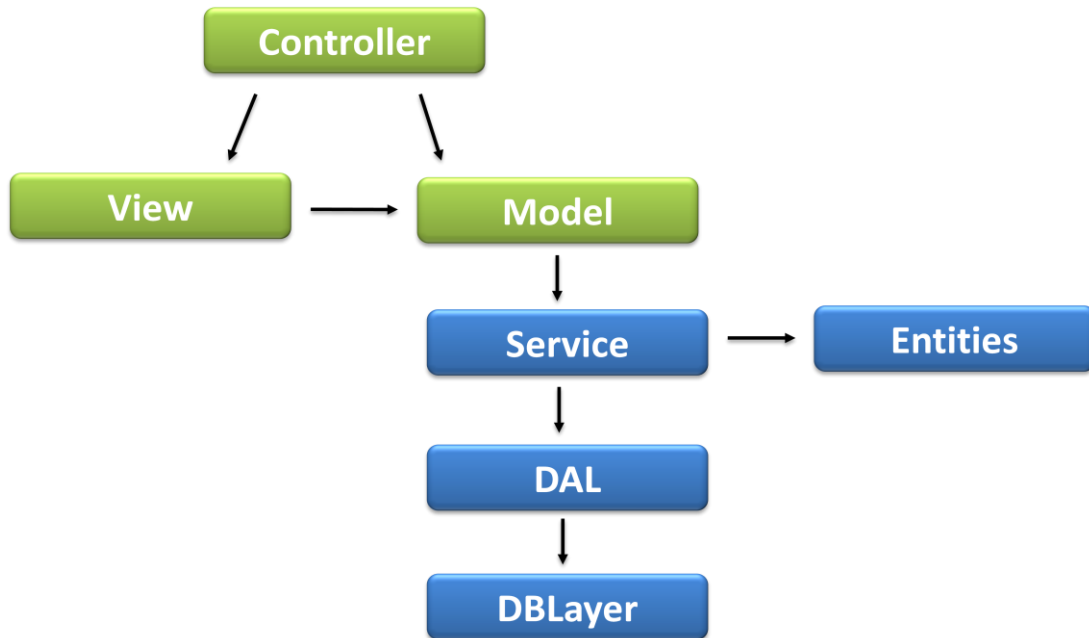


Figura. 11 Arquitectura del control centralizado de dispositivos

**Controller, View, Model:** Son las encargadas de manejar todo lo relacionado con la presentación, captura e inyección de datos en esta. Los tres componentes que la forman responden al patrón arquitectónico Modelo Vista Controlador.

**Services:** Los servicios son los encargados de exponer los métodos para manejar la lógica de la aplicación desde la presentación, en este caso la capa que interactúa directamente con ella es *Model*.

**Entities:** Se encuentran en esta capa todas las clases que representan las entidades que intervienen en el negocio de la aplicación. Esta capa puede ser accedida desde la capa de servicio hasta la presentación, permitiendo que todo el flujo de datos dentro de la aplicación se encuentre encapsulado en objetos.

**DAL (Capa de acceso a datos):** La capa de acceso a datos permite la persistencia de los objetos de negocio, esta capa se utiliza con el objetivo de estandarizar el acceso a datos, y eliminar el acoplamiento de los componentes de negocio al gestor de base de datos. De esta forma, se puede minimizar el impacto

de posibles modificaciones en el modelo de datos. Es la única capacitada para interactuar con la capa *Data Base Layer*.

**DBLayer (Base de datos):** Por la importancia que posee el desarrollo de procedimientos y vistas dentro del negocio, se ha decidido separar la base de datos en una nueva capa, donde se concentra una pequeña parte de la lógica de funcionalidades dentro de la aplicación. La única capa encargada de interactuar con esta es DAL.

### 3.1.1 Servicio local para el manejo de dispositivos

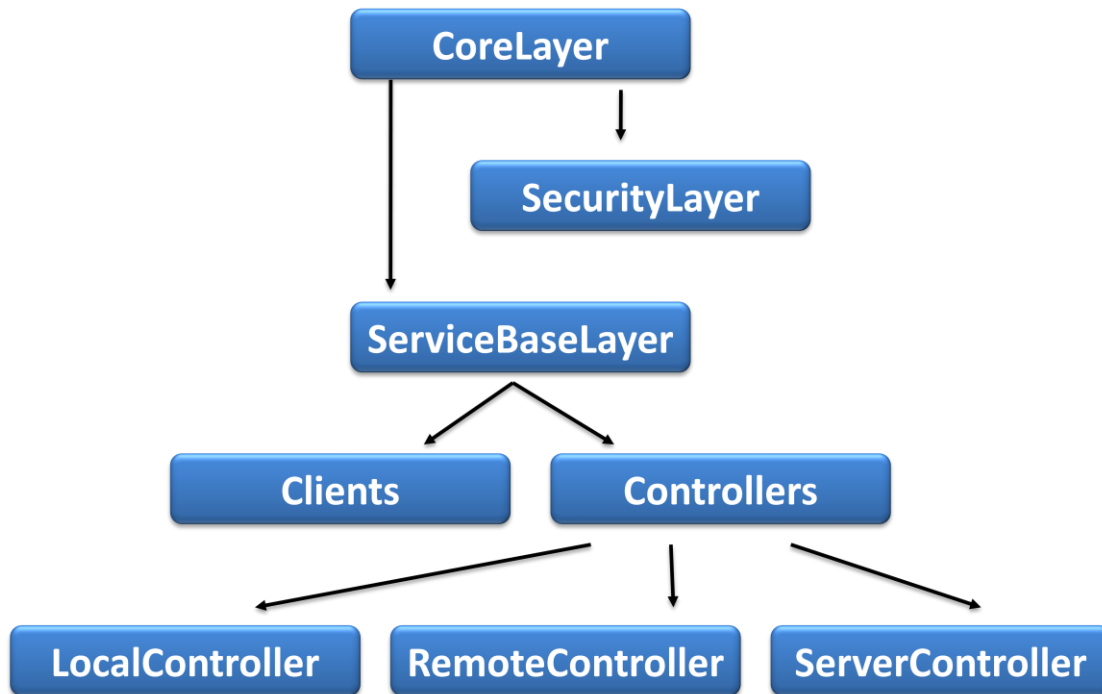


Figura. 12 Arquitectura del Servicio local para el manejo de dispositivos

**CoreLayer:** Esta capa es la encargada de manejar toda la lógica de la aplicación, orquesta el funcionamiento de las demás capas para lograr su objetivo, además de contar con el control de los servicios que se exponen en la aplicación y exponer un servicio web para la administración remota de la aplicación.

**SecurityLayer:** Es la capa que contiene todas las herramientas para garantizar la seguridad en la aplicación. Va desde las validaciones de los ensamblados y reconocimiento de integridad en la aplicación

hasta la autorización de las peticiones a los servicios, mediante el sistema centralizado de control de dispositivos o por simples reglas definidas en la aplicación.

### **Validación de peticiones http:**

El flujo de ejecución de una petición comienza en el código JavaScript de la página web. Este realiza una solicitud de ejecución de un determinado servicio, la solicitud es procesada por el *framework* JavaScript, quien le da el formato necesario para poder ser asimilada por el *addon*. En este proceso de conformación de la solicitud se incluye el *token* extraído de la sección activa de la aplicación web que reside dentro de la *cookie* emitida por la aplicación.

Una vez que la petición se encuentra en poder del *addons* es analizada y se pasa a ejecutar la petición http correspondiente a la llamada al servicio. Se le agrega una cabecera http que tiene como clave *token* y como valor el *token* de la sesión. El servicio que reside en la estación de trabajo cuenta con un mecanismo de intersección de las peticiones, quien extrae el *token* de sección de la cabecera de la petición y pasa a validar la vigencia de la misma utilizando el servicio de autenticación que brinda el sistema centralizado de control de dispositivos. Cabe destacar que este *token* viaja encriptado en todo momento mediante el uso del algoritmo Triple DES.

**ServiceBaseLayer:** Esta capa cuenta con las clases que manejan la conformación de los servicios, que no son más que la unión de los Clientes y los Controladores, gestiona la carga dinámica de los ensamblados, además de contener las clases base para el desarrollo de los controladores y los clientes.

**Clients:** Los clientes son la interfaz de los controladores, estos manejan interfaces gráficas en caso de necesitarlo, cuentan con los contratos para exponer los servicios además de brindar las interfaces que necesitan los controladores para acoplarse a ellos.

**Controller:** Los controladores son los responsables de la comunicación directa con los dispositivos. Existen tres tipos de controladores LocalController, este es el más común, su función es realizar la comunicación directa con un dispositivo conectado localmente; RemoteController está basado en la comunicación con un controlador remoto el cual permite el manejo de dispositivos a distancia; y por último se tiene el ServerController el cual tiene la función de exponer las funcionalidades del controlador como servicios web, para que puedan ser accedidas remotamente.



### Patrones de diseño utilizados

**Encapsulación:** propone esconder algunos componentes, permitiendo sólo accesos estilizados al objeto. Se hace uso de este patrón en casi todas las clases que componen al sistema, permitiendo que estas solo posean como elementos públicos aquellos que son exclusivamente necesarios.

**Subclase:** propone heredar miembros por defecto de una superclase, seleccionando la implementación correcta a través de resoluciones sobre qué implementación debe ser ejecutada. Se puede encontrar este patrón con más fuerza en las entidades de negocio que por su conceptualización las funciones y la información que almacenan pueden estar diferenciadas en cierta medida.

**Excepciones:** propone introducir estructuras de lenguaje para arrojar e interceptar excepciones. Se identificaron los diferentes tipos de errores a tratar dentro del sistema creando clases que permitan identificar cada tipo de error en el momento de ejecución.

**Fábrica:** provee de una interfaz para crear familias de objetos relacionados o dependientes sin especificar los tipos concretos de clases. Su uso se encuentra centrado en la creación los conectores correspondientes al acceso a datos que se esté utilizando, así como en la obtención de los servicios a utilizar.

**Singleton:** se asegura que solo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esta instancia. El uso de este patrón permite que los servicios puedan ser creados solo una vez.

**Grasp:** el uso de este patrón está totalmente ligado a cada componente desarrollado en el sistema, donde cada uno de ellos posee solo las funcionalidades acorde a las particularidades que lo caracterizan.

### Especificación de clases

#### 3.1.2 Control centralizado de dispositivos

Este sistema está formado por un grupo de clases servicios, conectores los que hacen posible su correcto funcionamiento.

3.1.2.1 Diagrama de clases

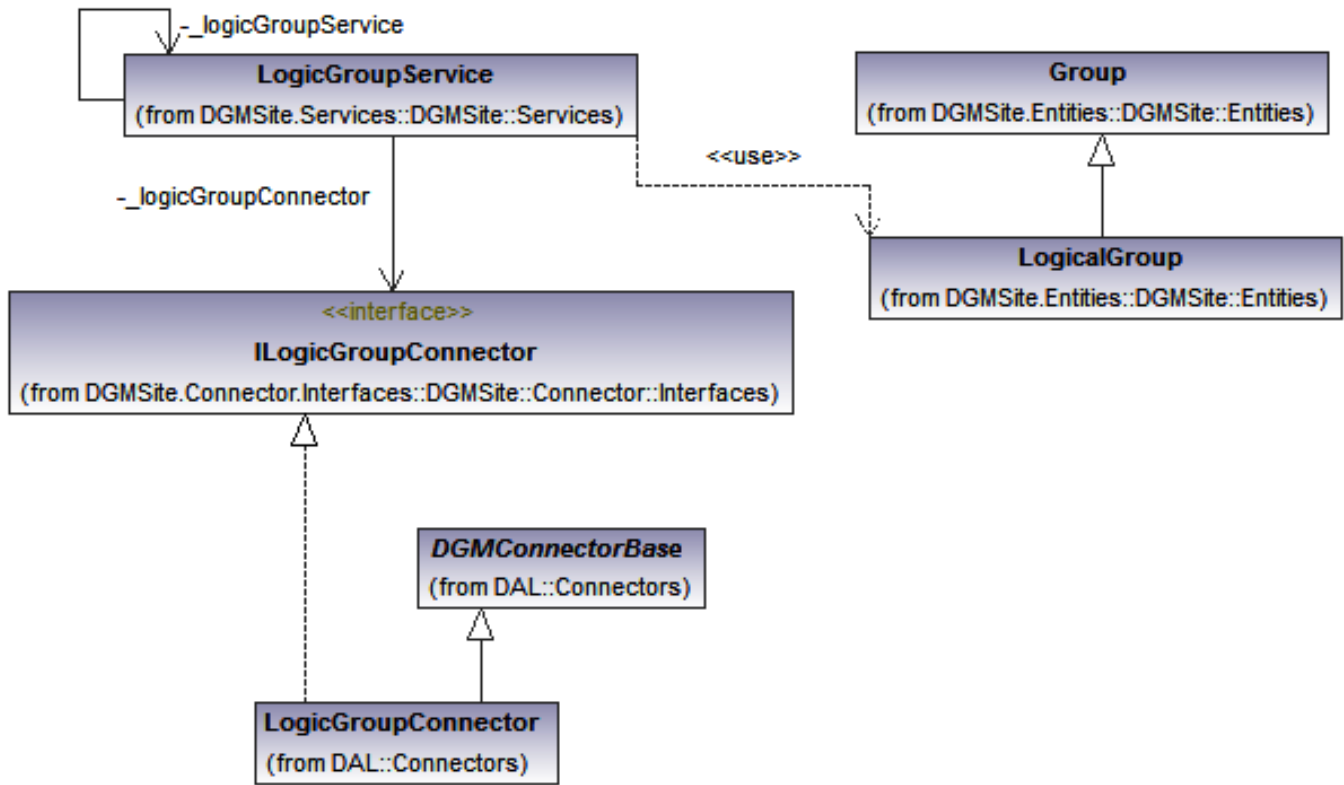


Figura. 13 Diagrama de clases del requisito "Gestionar agrupaciones lógicas"

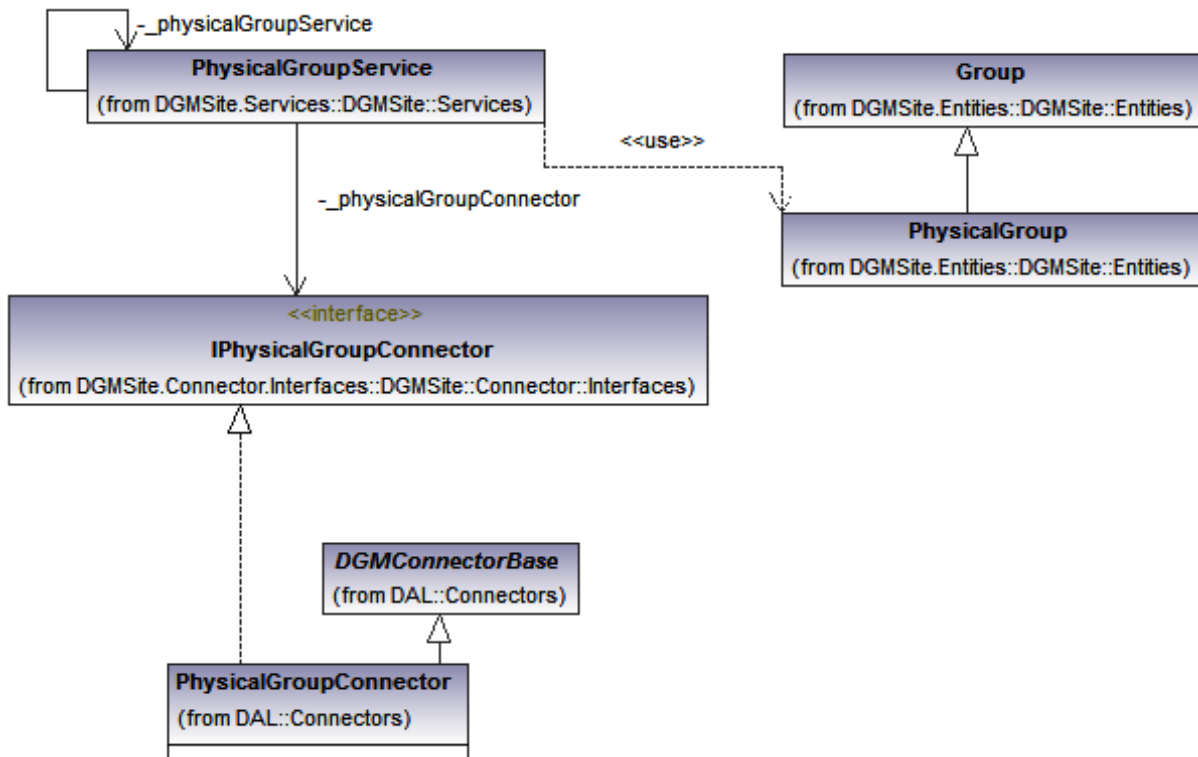


Figura. 14 Diagrama de clases del requisito "Gestionar agrupaciones lógicas"

### 3.1.2.2 Descripción de las clases

<b>Nombre</b>	DGMConnectorBase		
<b>Descripción</b>	Es la clase base para todos los conectores a la base de datos, cuenta con un grupo de funcionalidades que facilitan convertir las entidades de acceso a datos a una colección estándar, fácil de manejar y que independiza la capa de acceso a datos del resto de la aplicación.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción.</b>
Público	List<Hashtable> ConvertToHashTableList	Object	Transforma un objeto en una lista de hashtable.
Privado	Hashtable FromAnonymousObject	Object	Transforma un tipo de objeto anónimo a un hashtable.
Privado	Hashtable FromPrimitiveObject	Object	Transforma un objeto primitivo a un hashtable.
Protegido	Hashtable FromEntityObject	Object	Transforma una entidad de acceso a datos a hashtable.
Privado	bool IsScalar	Object	Verifica si es un tipo de dato básico.

Tabla 8. Descripción de la clase DGMConnectorBase

<b>Nombre</b>	ILogicGroupConnector		
<b>Descripción</b>	Es la interfaz que define las necesidades que deben ser implementadas por un controlador para la persistencia de los grupos lógicos.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción.</b>
Público	AddLogicGroup	description, idParentLogicGroup	Adiciona un grupo lógico en la base de datos.
Público	UpdateParentLogicGroup	idLogicGroup, idParentLogicGroup	Actualiza el padre de un grupo lógico.
Público	DeleteLogicGroup	idLogicGroup	Elimina un grupo lógico.
Público	List<Hashtable> LoadLogicGroupById	idLogicGroup	Obtiene los datos de un grupo lógico dado el identificador.
Público	List<Hashtable> LoadLogicGroupsChildrenById	idLogicGroup	Obtiene un listado de los grupos hijos.
Público	List<Hashtable> LoadLogicGroupsRoot		Retorna un listado con los grupos que no tienen padres.
Público	List<Hashtable> LoadLogicGroupParentById	idLogicGroup	Retorna el grupo lógico padre.
Público	bool IsFather	idLogicGroup	Indica si un grupo tiene hijos.

Tabla 9. Descripción de la interfaz ILogicGroupConnector

<b>Nombre</b>	IPhysicalGroupConnector		
<b>Descripción</b>	Es la interfaz que define las necesidades que deben ser implementadas por un controlador para la persistencia de los grupos físicos.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción</b>
Público	AddPhysicalGroup	description, idParentPhysicalGrou, active	Adiciona un grupo físico en la base de datos.
Público	UpdatePhysicalGroupDescription	idPhysicalGroup, description	Actualiza los datos de un grupo físico.
Público	UpdateParentPhysicalGroup	idPhysicalGroup, idParentPhysicalGroup	Actualiza el padre de un grupo físico.
Público	DeletePhysicalGroup	idPhysicalGroup	Elimina un grupo físico dado su id.
Público	List<Hashtable> LoadPhysicalGroupByld	idPhysicalGroup	Retorna los datos de un grupo físico dado su id.
Público	List<Hashtable> LoadPhysicalGroupsChildrenByld	idPhysicalGroup	Retorna los grupos físicos hijos dado el id del padre.
Público	List<Hashtable> LoadPhysicalGroupsRoot		Retorna un listado con los grupos que no tienen padres.
Público	List<Hashtable> LoadPhysicalGroupParentByld	idPhysicalGroup	Retorna los datos del grupo padre dado el id del hijo.
Público	bool IsFather	idPhysicalGroup	Retorna si un grupo físico es padre.

Tabla 10 Descripción de la interfaz IPhysicalGroupConnector

<b>Nombre</b>	LogicGroupConnector		
<b>Descripción</b>	Adiciona, actualiza, obtiene y elimina un grupo lógico en la base de datos.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción</b>
Público	AddLogicGroup	Description, idParentLogicGroup	Adiciona un grupo lógico en la base de datos.
Público	UpdateParentLogicGroup	idLogicGroup, idParentLogicGroup	Actualiza el padre de un grupo lógico.
Público	UpdateLogicGroupDescription	idLogicGroup, description	Actualiza la descripción de un grupo lógico.
Público	DeleteLogicGroup	idLogicGroup	Elimina un grupo lógico.
Público	List<Hashtable> LoadLogicGroupById	idLogicGroup	Obtiene los datos de un grupo lógico dado el identificador.
Público	List<Hashtable> LoadLogicGroupsChildrenById	idLogicGroup	Obtiene un listado de los grupos hijos.
Público	List<Hashtable> LoadLogicGroupsRoot		Retorna un listado con los grupos que no tienen padres.
Público	List<System.Collections.Hashtable> LoadLogicGroupParentById	idLogicGroup	Retorna el grupo lógico padre.
Público	bool IsFather	idLogicGroup	Indica si un grupo tiene hijos.

Tabla 11. Descripción de la clase LogicGroupConnector

<b>Nombre</b>	PhysicalGroupConnector		
<b>Descripción</b>	Adiciona, actualiza, obtiene y elimina un grupo físico en la base de datos.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción</b>
Público	AddPhysicalGroup	description, idParentPhysicalGrou, active	Adiciona un grupo físico en la base de datos.
Público	UpdatePhysicalGroupDescription	idPhysicalGroup, description	Actualiza los datos de un grupo físico.
Público	UpdateParentPhysicalGroup	idPhysicalGroup, idParentPhysicalGroup	Actualiza el padre de un grupo físico.
Público	DeletePhysicalGroup	idPhysicalGroup	Elimina un grupo físico dado su id.
Público	List<Hashtable> LoadPhysicalGroupByld	idPhysicalGroup	Retorna los datos de un grupo físico dado su id.
Público	List<Hashtable> LoadPhysicalGroupsChildrenByld	idPhysicalGroup	Retorna los grupos físicos hijos dado el id del padre.
Público	List<Hashtable> LoadPhysicalGroupsRoot		Retorna un listado con los grupos que no tienen padres.
Público	List<Hashtable> LoadPhysicalGroupParentByld	idPhysicalGroup	Retorna los datos del grupo padre dado el id del hijo.
Público	bool IsFather	idPhysicalGroup	Retorna si un grupo físico es padre.

Tabla 12 Descripción de la clase PhysicalGroupConnector

### 3.1.3 Servicios del sistema

<b>Nombre</b>	LogicGroupService		
<b>Descripción</b>	Es el encargado de gestionar los grupos lógicos a nivel de objetos con la utilización de un conector para manejar la persistencia de estos objetos.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción.</b>
Privado	LogicGroupService		Crea una instancia.
Público	LogicGroupService LogicGroupServiceIntance		Crea una instancia en caso de o existir y luego la retorna.
Público	List<LogicalGroup> LogicGroups		Retorna todos los grupos lógicos que no tiene padres.
Público	List<LogicalGroup> LogicGroups	logicGroupFather	Retorna la lista de grupos lógicos hijos dado un grupo lógico.
Público	SaveLogicalGroup	logicalGroup, logicalGroupFather	Persiste un grupo lógico en la base de datos.
Público	UpdateLogicalGroup	logicalGroup	Actualiza la información de un grupo lógico.
Público	UpdateLogicalGroupFather	logicalGroup, logicalGroupFather	Actualiza el padre de un grupo lógico.
Público	DeleteLogicalGroup	logicalGroup	Elimina un grupo lógico
Privado	LogicalGroup HashtableToLogicGroup	hashtable	Convierte un hashtable a un grupo lógico.

Tabla 13. Descripción del servicio LogicGroupService

<b>Nombre</b>	PhysicalGroupService		
<b>Descripción</b>	Es el encargado de gestionar los grupos físicos a nivel de objetos con la utilización de un conector para manejar la persistencia de estos objetos.		
<b>Visibilidad</b>	<b>Método</b>	<b>Argumentos</b>	<b>Descripción.</b>
Privado	PhysicalGroupService		Crea una instancia.
Público	PhysicalGroupService PhysicalGroupServiceIntance		Crea una instancia en caso de no existir y luego la retorna.
Público	List<PhysicalGroup> PhysicalGroupsChildren	physicalGroup	Retorna los grupos físicos hijos dado el padre.
Público	SavePhysicalGroup	physicalGroup, physicalGroupFather	Persiste un grupo físico dado el grupo padre.
Público	PhysicalGroup	physicalGroup	Retorna un grupo físico.



Público	UpdatePhysicalGroup	physicalGroup	Actualiza los datos de un grupo físico.
Público	UpdatePhysicalGroupFather	physicalGroup, physicalGroupFather	Cambial el grupo padre de un grupo físico.
Público	DeletePhysicalGroup	physicalGroup	Elimina un grupo físico.
Privado	HashtableToPhisicalGropup	hashtable	Transforma un hastable en un grupo físico.

*Tabla 14 Descripción del servicio PhysicalGroupService*

### **Modelo de datos**

El siguiente modelo de datos está basado en las necesidades de persistencia del sistema, su principal objetivo es representar relaciones coherentes y optimizadas entre cada una de las entidades de la base de datos, logrando así un uso eficiente del espacio de almacenamiento, evitando la persistencia de datos redundadas y agilizando la ejecución de consultas para la recuperación de información.

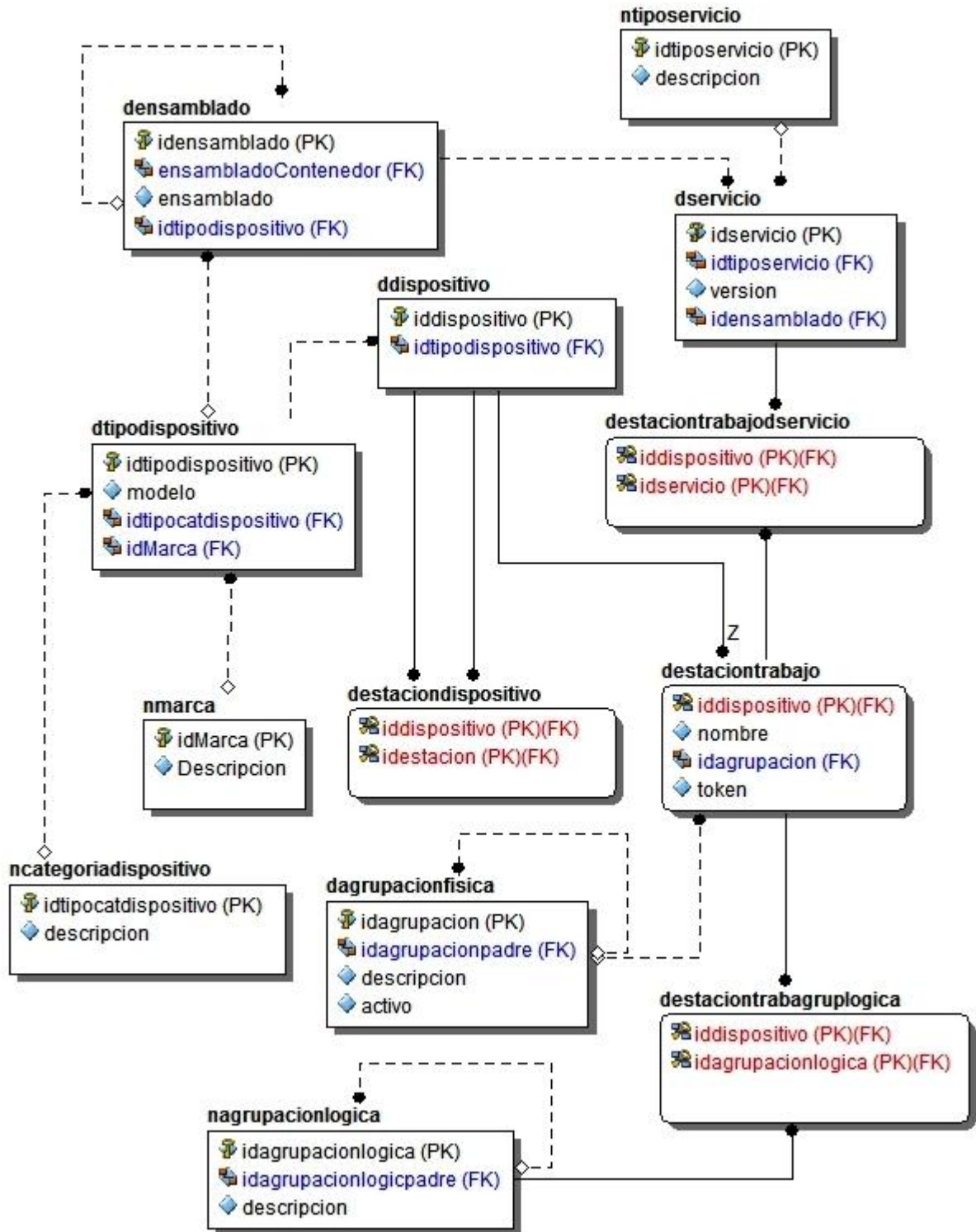


Figura. 15 Modelo de datos

### 3.1.4 Descripción de las entidades fundamentales

<b>Entidad</b>	nagrupacionlogica		
<b>Descripción</b>	Representa de forma recursiva el árbol de dependencia que conforma la estructura de las entidades.		
<b>Atributos</b>	<b>Tipo de Dato</b>	<b>Nulo</b>	<b>Definición</b>
idagrupaciónlogica	RAW(16)	no	Representa el identificador único de la agrupación lógica, mediante un GUID.
ldagrupacionpadre	RAW(16)	no	Representa el identificador único de la agrupación lógica padre, mediante un GUID.
descripcion	Nvarchar2(50)	no	Descripción del grupo lógico.

*Tabla 15. Descripción de la entidad nagrupacionlogica.*

<b>Entidad</b>	dagrupacionfisica		
<b>Descripción</b>	Representa de forma recursiva el árbol de dependencia que conforma la estructura de las entidades físicas.		
<b>Atributos</b>	<b>Tipo de Dato</b>	<b>Nulo</b>	<b>Definición</b>
idagrupacio	RAW(16)	no	Representa el identificador único de la agrupación física, mediante un GUID.
ldagrupacionpadre	RAW(16)	no	Representa el identificador único de la agrupación física padre, mediante un GUID.
descripcion	Nvarchar2(50)	no	Descripción del grupo físico.
activo	NUMBER	no	Indica si se encuentra activa la agrupación.

*Tabla 16. Descripción de la entidad dagrupacionfisica.*

### Conclusiones

Se definió y documentó la arquitectura de cada uno de los módulos del sistema, la cual está basada en el desacoplamiento de las capas que la conforman y a la reutilización de estas. Se realizó el análisis y diseño de la solución definiéndose las clases y servicios necesarios, así como la relación entre ellos, para darle solución a todas las funcionalidades descritas en el catálogo de requisitos. Se diseñó el modelo de datos, especificándose atributos, sus relaciones y descripción de sus entidades para el correcto almacenamiento de los datos del sistema.



# Capítulo IV

IMPLEMENTACIÓN Y PRUEBAS

## Introducción

En el presente capítulo se hace alusión al estándar de código seleccionado para su uso en la implementación de la solución, se muestra el diagrama de despliegue con cada uno de los nodos necesarios para la implantación de la solución, además de los diagramas de componentes que la conforman al igual que toda la etapa de pruebas aplicadas al software tanto las pruebas unitarias como las pruebas de sistemas, siendo estas un eslabón fundamental para la calidad del producto..

## Estándares de codificación

Un patrón de nomenclatura coherente es uno de los elementos más importantes de la previsibilidad y el descubrimiento de una biblioteca de clases administradas. El uso generalizado y la comprensión de estas directrices de denominación deberían eliminar muchas de las preguntas más comunes.

## Estilos para la capitalización

Se podrán utilizar los siguientes tres convenios para la capitalización de los identificadores:

### Pascal

La primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Puede utilizar los identificadores de Pascal case en caso de tres o más caracteres. Por ejemplo:

```
BackColor
```

### Camello

La primera letra en el identificador está en minúscula y la primera letra de cada subsiguiente palabra concatenada es mayúscula. Por ejemplo:

```
backColor
```

### Mayúscula

Todas las letras en el identificador se capitalizan. Esta convención se utilizará sólo para los identificadores que constan de dos o menos letras. Por ejemplo:

```
System.IO
```

```
System.Web.UI
```

### **Sensibilidad a mayúsculas**

- No se deberá utilizar nombres o identificadores que requieran ser *case sensitivity*.
- No se deberá crear dos *namespaces* que se diferencien solo en el uso de las mayúsculas.
- No crear funciones con nombres de parámetros que se diferencian solo en el uso de la mayúscula.
- No se deberá crear *namespaces* con nombres de clases que se diferencien solo en el uso de las mayúsculas.
- No crear clases con propiedades que se diferencien solo en el uso de las mayúsculas.
- No crear clases con métodos que se diferencien solo en el uso de las mayúsculas.

### **Evitando confusión de nombre y tipo**

Utiliza nombres que describan a sus identificadores en vez de nombres que describen el tipo de identificador.

**Diagrama de despliegue**

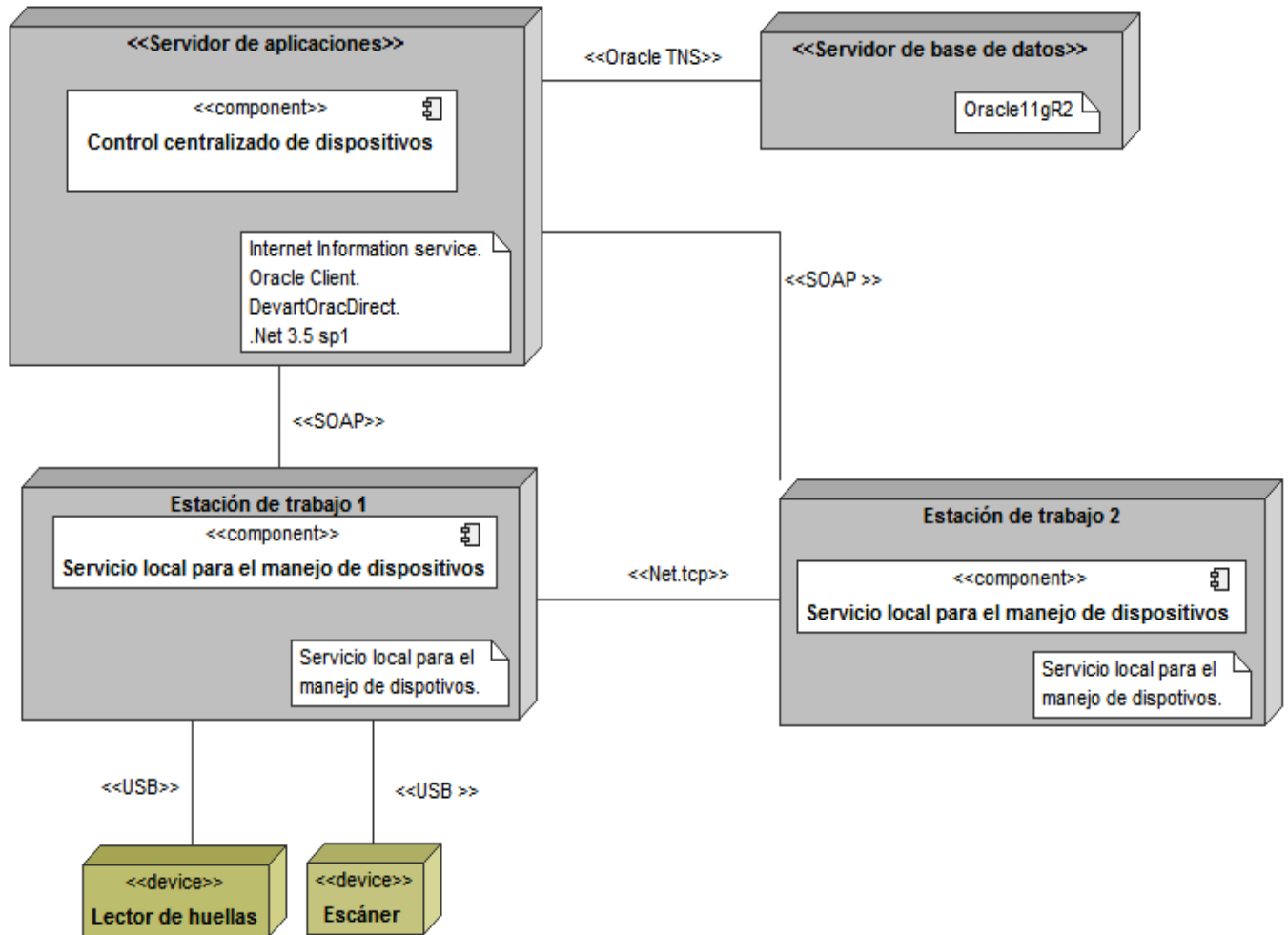


Figura. 16 Diagrama de despliegue

**4.1.1 Descripción del diagrama de despliegue**

El diagrama de despliegue está conformado por dos servidores, un servidor de base de datos el cual está soportado sobre Oracle11g y un servidor de aplicaciones que contiene en su núcleo *Internet Information Server 7*, en el diagrama se representan dos nodos clientes los cuales utilizan el protocolo SOAP<sup>9</sup> para la

<sup>9</sup> (*Simple Object Access Protocol*) Protocolo para el intercambio de mensajes utilizando el formato XML.

comunicación con el servidor de aplicaciones, entre los nodos clientes utilizan el protocolos REST<sup>10</sup> para compartir entre ellos los dispositivos los cuales se conectan a uno mediante USB<sup>11</sup>.

### Diagrama de componentes

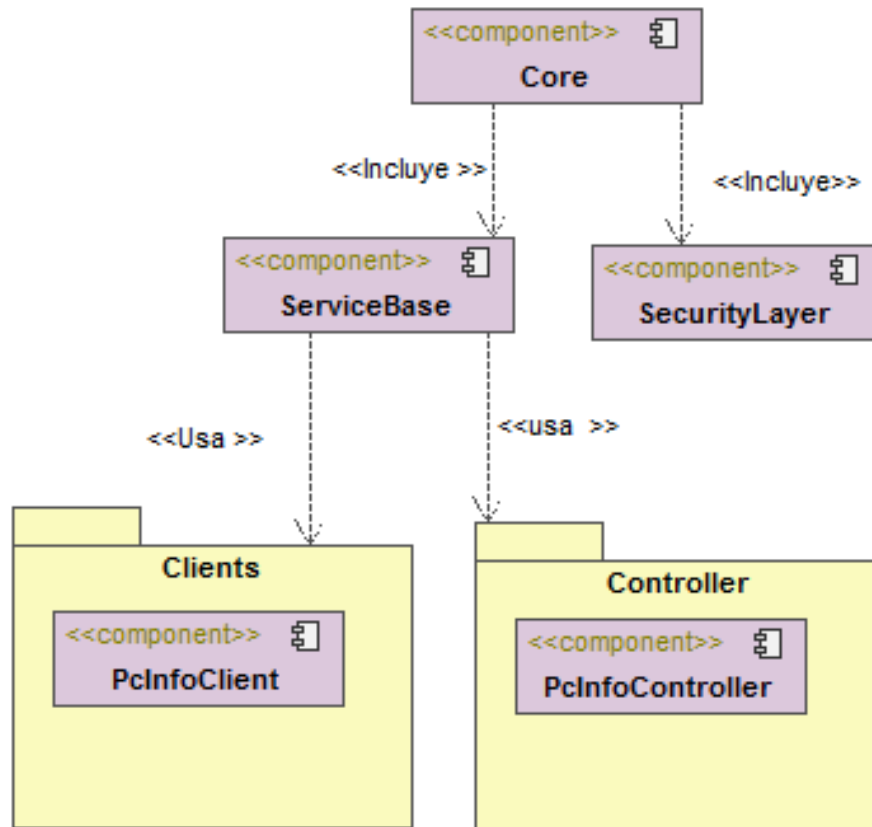


Figura. 17 Diagrama de componentes Servicio local

<sup>10</sup> (*Representational State Transfer*) técnica de arquitectura software para sistemas hipermedia distribuidos como la *World Wide Web*.

<sup>11</sup> (*Universal Serial Bus*) Estándar de la industria para conectar dispositivos externos a diferentes plataformas como puede ser una computadora.



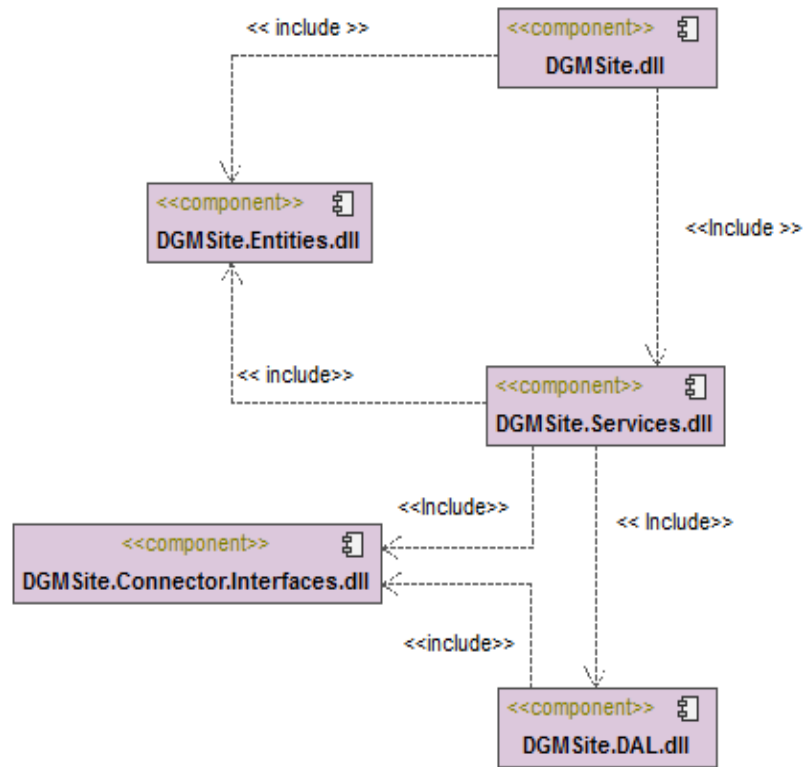


Figura. 18 Diagrama de componentes del módulo para Control centralizado de dispositivos

#### 4.1.2 Descripción del diagrama de componente

En la figura 18 se muestra el diagrama de componentes que representa al módulo para el control centralizado de dispositivos, está conformado por cinco dll las cuales son: GGMSite.dll donde se encuentra toda la lógica de la presentación del sitio; DGMSite.Entities.dll, cuenta con todas las entidades del negocio; GGMSite.Connector.Interface.dll contiene todas las interfaces las cuales deben implementar el DAL; DGMSite.Service.dll contiene las implementaciones de los servicios, quienes hacen uso de DGMSite.DAL.Dll siendo esta la responsable de implementar toda la lógica de acceso a datos.

#### 4.1.3 Modelo de pruebas

Un modelo de pruebas implica la operación o aplicación del mismo a través de condiciones controladas y la consiguiente evaluación de la informática. Las condiciones controladas deben incluir tanto situaciones normales como anormales. El objetivo del modelo de pruebas es encontrar un error para determinar situaciones donde algo pasa cuando no debe de pasar y viceversa.

#### 4.1.4 Diseño de casos de pruebas

Consiste en la confección de los distintos casos de prueba según la técnica o técnicas identificadas previamente. El diseño de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al ejecutar dicho caso para detectar un posible fallo en el programa. Los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Cada técnica de pruebas proporciona unos criterios distintos para generar estos casos o datos de prueba. (24)

#### CPR 1: Gestionar agrupaciones físicas

##### Condiciones de Ejecución

El administrador debe haber adquirido los permisos necesarios.

Flujo General del requisito

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
1. Gestionar nuevas agrupaciones físicas	1.1 Gestionar nuevas agrupaciones físicas	Gestionar nuevas agrupaciones físicas.	<p>1.1 Si se desea adicionar una agrupación física padre.</p> <p>1.1.1 Seleccionar la opción "Adicionar".</p> <p>1.1.1.1 Mostrar una nueva pantalla con los siguientes datos:</p> <p>(a) Descripción</p> <p>1.1.1.2 Seleccionar la opción "Aceptar" si desea que la agrupación sea creada.</p> <p>1.1.1.3 Seleccionar la opción "Cancelar" si desea cancelar la operación.</p> <p>1.2 Si no se desea adicionar una agrupación física como padre</p> <p>1.2.1 Seleccionar la agrupación padre a la cual pertenecerá.</p> <p>1.2.2 Seleccionar la opción "Adicionar".</p>

			<p>1.2.2.1 Mostrar una nueva pantalla con los siguientes datos:</p> <ul style="list-style-type: none"> <li>(a) Muestra el nombre del padre.</li> <li>(b) Permitir introducir la descripción.</li> </ul> <p>1.2.2.2 Seleccionar la opción "Aceptar" si desea que la agrupación sea creada.</p> <p>1.2.2.3 Seleccionar la opción "Cancelar" si desea cancelar la operación.</p> <p>1.3 Si se desea modificar una agrupación.</p> <p>1.3.1 Seleccionar una agrupación.</p> <p>1.3.2 Seleccionar la opción "modificar".</p> <p>1.3.2.1 Mostrar una nueva ventana con los siguientes datos.</p> <ul style="list-style-type: none"> <li>(a) Permitir modificar el padre</li> <li>(b) Permitir modificar la descripción.</li> </ul> <p>1.3.2.2 Seleccionar la opción "Aceptar" si desea que la agrupación sea modificada.</p> <p>1.3.2.3 Seleccionar la opción "Cancelar" si desea cancelar la operación.</p> <p>1.4 Si se desea eliminar una agrupación.</p> <p>1.4.1 Seleccionar la agrupación a eliminar.</p> <p>1.4.2 Seleccionar la opción "Eliminar".</p> <p>1.4.2.1 Mostrar una nueva ventana con los datos de la agrupación a elimina:</p> <ul style="list-style-type: none"> <li>(a) Identificador.</li> </ul>
--	--	--	--

			<p>(b) Descripción.</p> <p>1.4.2.2 Seleccionar la opción "Aceptar" para eliminar la agrupación.</p> <p>1.4.2.3 Seleccionar la opción "Cancelar" para anular la operación.</p>
1.1.1.2) Opción "Aceptar"	Validar los datos de entrada.	Permite mostrar un mensaje de error cuando la descripción no es introducida.	1) Mostrar mensaje de error "Debe de introducir la descripción".
1.1.1.3) Opción "Cancelar"	Validar el retorno a la página principal	Permite la redirección de la página.	1) Ir a la página principal.
1.2.2.2) Opción "Aceptar"	Validar los datos de entrada.	Permite mostrar un mensaje de error cuando la descripción no es introducida.	1) Mostrar mensaje de error "Debe de introducir la descripción".

1.2.2.3) Opción "Cancelar"	Validar el retorno a la página principal	Permite la redirección de la página.	1) Ir a la página principal.
1.3.2.2 Opción "Aceptar"	Validar los datos de entrada.	Permite mostrar un mensaje de error cuando la descripción no es introducida.	1) Mostrar mensaje de error "Debe de introducir la descripción".
1.3.2.3 Opción "Cancelar"	Validar el retorno a la página principal	Permite la redirección de la página.	1) Ir a la página principal.
1.4.2 Opción "Aceptar"	Validar los datos seleccionados	Permite eliminar la agrupación seleccionada	1) Mostrar mensaje de error "Debe seleccionar la agrupación".

Tabla 17. Diseño de prueba de caja negra del requisito funcional Gestionar agrupación lógica

#### 4.1.5 Pruebas unitarias (Caja blanca)

Es la prueba enfocada a probar los elementos más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están

cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca.

```
/// <summary>
///A test for LogicGroups
///</summary>
[TestMethod()]
public void LogicGroupsTest()
{
    var target = new LogicGroupService_Accessor();

    var logicGroupFather = new LogicalGroup { Id = "37" };

    var expected = new List<LogicalGroup>
        {
            new LogicalGroup
            {
                Id = "39"
            },
            new LogicalGroup
            {
                Id = "38"
            },
        };

    List<LogicalGroup> actual = target.LogicGroups(logicGroupFather);

    for (int i = 0; i < expected.Count; i++)
    {
        Assert.AreEqual(expected[i], actual[i]);
    }
}
```

Figura. 19 Código de la prueba unitaria al servicio LogicGroupService

#### 4.1.6 Pruebas del sistema

Verifica el sistema completo o su aplicación como tal. Se toma el punto de vista del usuario final y los casos de uso de prueba ejecutan acciones típicas del usuario. (25)

4.1.6.1 Iteraciones

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
1.1.1.2)	<b>Clases Válidas</b>	Números y letras	Agrupación adicionada	Se adicionó correctamente la prueba	-
	<b>Clases Inválidas</b>	Valor nulo	Mensaje de error "Debe introducir la descripción."	Se mostró correctamente el mensaje.	-
1.1.1.3)	<b>Clases Válidas</b>	Retornar a la página principal	Redirección la página principal	Retornó a la página principal.	
	<b>Clases Inválidas</b>	Quedarse en la misma página.	Quedarse en la misma página.	No redireccionó a la página principal	
1.2.2.2)	<b>Clases Válidas</b>	Números y letras	Agrupación adicionada	Se adicionó correctamente la agrupación.	

	<b>Clases Inválidas</b>	Valor nulo	Mensaje de error "Debe introducir la descripción."	Se mostró satisfactoriamente el mensaje.	
1.2.2.3)	<b>Clases Válidas</b>	Retornar a la página principal	Redirección la página principal	Retornó a la página principal.	
	<b>Clases Inválidas</b>	Quedarse en la misma página.	Quedarse en la misma página.	No redireccionó a la página principal	
1.3.2.2)	<b>Clases Válidas</b>	Números y letras	Agrupación modificada	La modificación se realizó con éxito.	
	<b>Clases Inválidas</b>	Valor nulo	Mensaje de error "Debe introducir la descripción."	Se mostró satisfactoriamente el mensaje.	



1.3.2.3	<b>Clases Válidas</b>	Retornar a la página principal	Redirección la página principal	Retornó a la página principal.	
	<b>Clases Inválidas</b>	Quedarse en la misma página.	Quedarse en la misma página.	No redireccionó a la página principal	
1.4.2 )	<b>Clases Válidas</b>	Selección de al menos un valor	Agrupación seleccionada.	Se seleccionó correctamente el elemento.	
	<b>Clases Inválidas</b>	Valor nulo	Mostrar un mensaje de error "Debe seleccionar una agrupación".	Se mostró satisfactoriamente el mensaje.	

Tabla 18. Prueba del sistema resultado de la tercera iteración del requisito funcional "Gestionar agrupaciones lógicas"

#### 4.1.7 Resultado de las pruebas

El resultado de las pruebas es una parte primordial en el desarrollo de los sistemas porque brinda la posibilidad de ver que tan bien está el desarrollo con respecto al cumplimiento de los requerimientos funcionales especificados. El sistema propuesto fue probado durante tres iteraciones detectándose en la primera iteración 16 no conformidades, 7 en la segunda y culminando con una exitosa prueba en la tercera iteración como se muestra en la **¡Error! No se encuentra el origen de la referencia.** resueltas todas las anteriores no conformidades dándole al sistema un nivel de seguridad mayor.

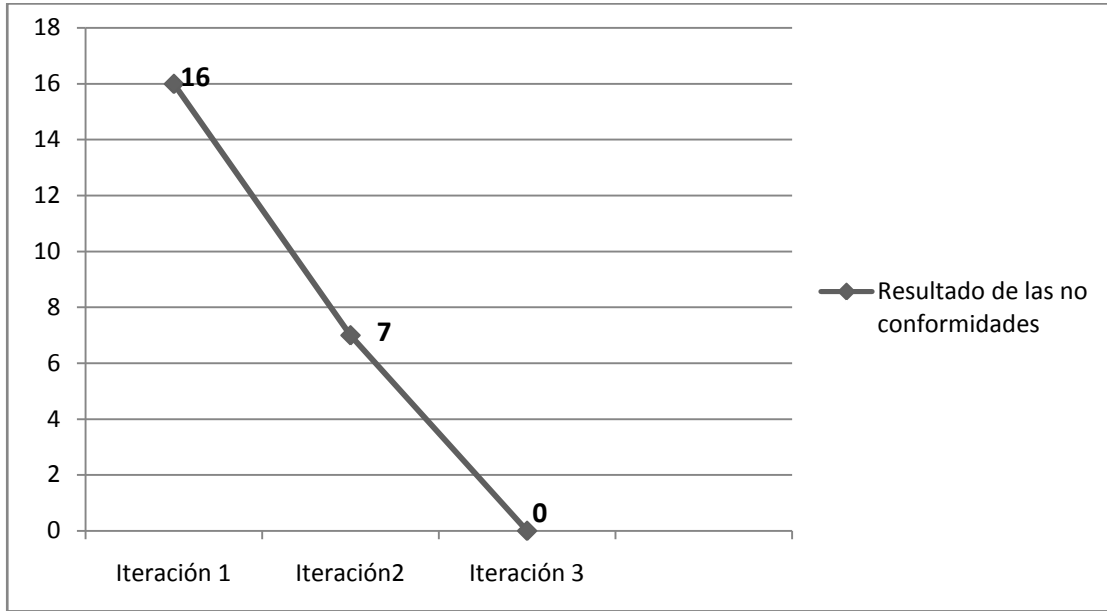


Figura. 20 Resultados de las pruebas

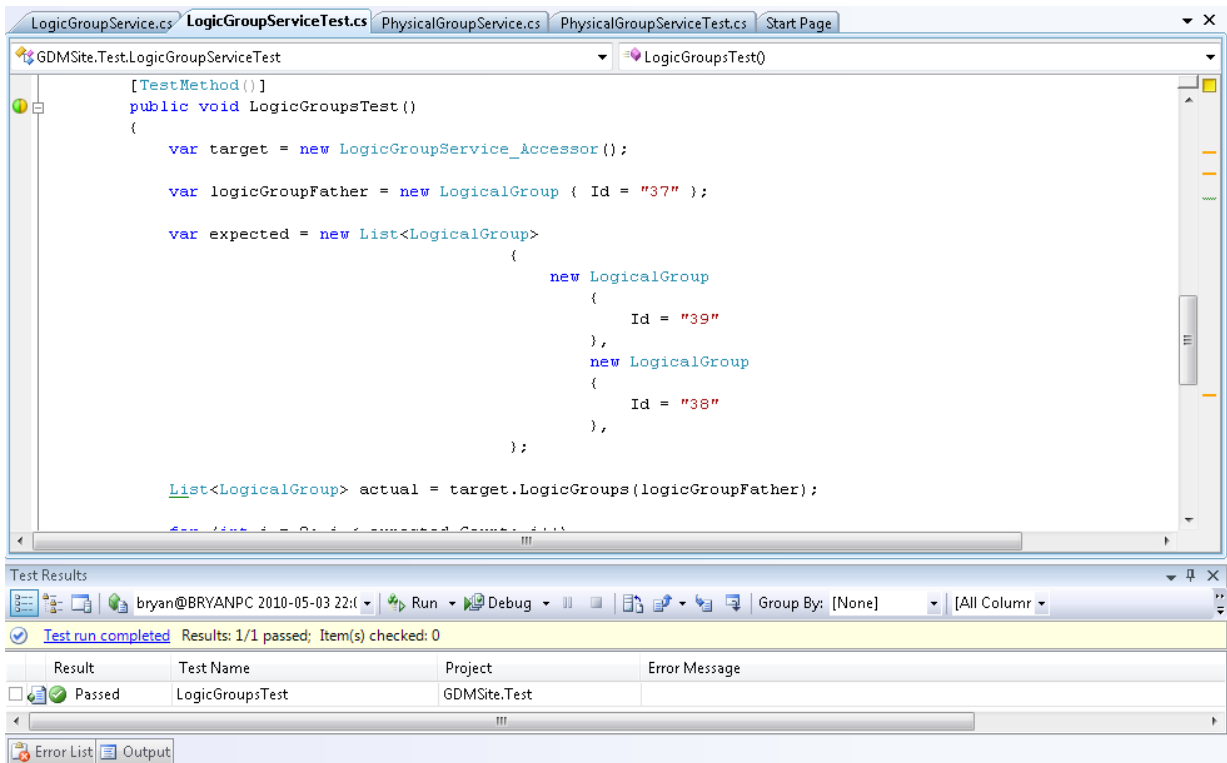


Figura. 21 Resultado de la prueba unitaria del servicio LogicGroupService

### **Conclusiones**

Se describió el estándar de código a utilizar en el desarrollo de la aplicación, para facilitar el entendimiento y la claridad del código fuente de esta. Se desarrolla una primera versión del sistema para el manejo de dispositivos centralizados. Se muestran las interfaces del sistema, las cuales permiten la interacción con el sistema para la administración de dispositivos centralizado. Se realizan las pruebas a la aplicación basándose en los diseños de casos de prueba definidos, donde se muestra la calidad de las especificaciones del cliente y se garantiza la calidad del software.

---

---

## CONCLUSIONES

- Se estudiaron las tecnologías que posibilitan la interacción con dispositivos de *hardware* vinculados a una estación de trabajo desde el navegador web, haciendo uso de la tecnología web del lado del cliente.
- Se detallaron tecnologías y estándares que facilitan el desarrollo, modelación y control adecuado de la solución, abordándose distintas tendencias, para conformar una base clara con la retroalimentación de proyectos y soluciones semejantes existentes, los cuales cuentan con un alto grado de confiabilidad.
- Se seleccionaron las herramientas apropiadas para el desarrollo y control del proyecto.
- Se desarrolló la propuesta de solución del sistema, teniendo en cuenta el modelado de dominio y se obtuvieron los requisitos del sistema, tanto funcionales como no funcionales.
- Se definió y documentó la arquitectura de cada uno de los módulos del sistema, la cual está basada en el desacoplamiento de las capas que la conforman y a la reutilización de estas.
- Se realizó el análisis y diseño de la solución definiéndose las clases y servicios necesarios, así como la relación entre ellos, para darle solución a todas las funcionalidades.
- Se diseñó el modelo de datos, especificándose atributos, sus relaciones y descripción de sus entidades para el correcto almacenamiento de los datos.
- Se desarrolló una primera versión del sistema: un *Framework* JavaScript para manejo de dispositivos, un *addon* para establecer la comunicación entre el navegador, servicios de manejo de dispositivos, un servicio para el manejo de dispositivos de *Hardware* con las funciones para la obtención de información de las estaciones de trabajo y las funcionalidades básicas del módulo web para el manejo centralizado de los dispositivos (DGM).
- Se realizaron las pruebas a la aplicación, tanto de caja negra como de caja blanca, basándose en los diseños de casos de prueba definidos y las funcionalidades del Visual Studio para las pruebas unitarias respectivamente, garantizándose un nivel adecuado de calidad del software.

## RECOMENDACIONES

- Continuar con la implementación de complementos para extender la compatibilidad del sistema con los navegadores web de uso más frecuente, tales como Internet Explorer, Chrome, Opera, entre otros.
- Concluir la integración de los módulos desarrollados en el Proyecto Identidad Cuba, y extenderlo a otros proyectos del Centro de Identificación y Seguridad Digital de la UCI.
- Implementar los módulos para gestión del mantenimiento de los dispositivos y brindar funciones dentro del sistema centralizado para el control de dispositivos que faciliten la detección de posibles robos en la red corporativa.
- Analizar otras variantes existentes para garantizar la seguridad dentro de las plataformas y tecnologías utilizadas, en función de enriquecer la solución actual.
- Continuar la implementación de la administración web, para satisfacer todos los requisitos establecidos.
- Realizar un mayor número de pruebas basadas en ambientes reales para el análisis de la eficiencia, con el objetivo de optimizar las soluciones brindadas.

---

---

**REFERENCIAS BIBLIOGRÁFICAS**

1. Bundesdruckerei. *Bundesdruckerei*. [En línea] 1 de marzo de 2010. <http://www.bundesdruckerei.de/>.
2. SUN. *SUN*. [En línea] 3 de Marzo de 2010. <http://java.sun.com/javacard/>.
3. Digital Persona. *Digital Persona*. [En línea] 20 de marzo de 2010. <http://www.digitalpersona.com/>.
4. Support Microsoft. [En línea] [Citado el: 19 de febrero de 2010.] <http://support.microsoft.com/>.
5. World Wide Web Consortium. *World Wide Web Consortium*. [En línea] 9 de mayo de 2010. <http://www.w3.org>.
6. SUN. [En línea] [Citado el: 19 de febrero de 2010.] <http://java.sun.com>.
7. Mozilla Addons. *Mozilla Addons*. [En línea] 16 de marzo de 2010. <https://addons.mozilla.org/>.
8. axiompas. *axiompas*. [En línea] 5 de marzo de 2010. <http://www.axiompas.com/AxiomDMS.aspx>.
9. sagem. *sagem*. [En línea] 15 de marzo de 2010. <http://www.sagem.com/index.php?id=774&L=8>.
10. *Framework para integrar el acceso a dispositivos*. **Campo, Adonis C. Legón**. Ciudad de la Habana : s.n., 2007.
11. **UCI**. *Documentos Proyecto Identidad Cuba*. 2009.
12. MSDN. [En línea] <http://msdn.microsoft.com/es-es/library/4w3ex9c2.aspx>.
13. msdn. [En línea] <http://msdn.microsoft.com/en-us/library/aa753587%28VS.85%29.aspx>.
14. MSDN. [En línea] noviembre de 2007. [http://msdn.microsoft.com/es-es/library/bb397926\(VS.90\).aspx](http://msdn.microsoft.com/es-es/library/bb397926(VS.90).aspx).
15. ECMA. *ECMA*. [En línea] 3 de abril de 2010. <http://www.ecma-international.org/>.
16. Mozilla developer center. [En línea] [Citado el: 2 de febrero de 2010.] <https://developer.mozilla.org/>.
17. W3e. [En línea] 11 de Mayo de 2010. <http://www.w3.org/XML/>.
18. W3e. [En línea] 11 de mayo de 2010. <http://www.w3c.es/divulgacion/guiasbreves/XHTML>.

19. W3e. [En línea] 5 de Mayo de 2010. <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
20. Microsoft TechNet. Microsoft TechNet. [En línea] [Citado el: 24 de febrero de 2010.] <http://technet.microsoft.com/en-us/windowsserver/sharepoint/bb684453.aspx..>
21. MSDN. [En línea] [Citado el: 19 de febrero de 2010.] <http://msdn.microsoft.com/>.
22. Microsoft. [En línea] [Citado el: 19 de febrero de 2010.] <http://www.microsoft.com/downloads/details.aspx?familyid=333325FD-AE52-4E35-B531-508D977D32A6&displaylang=es>.
23. IEE. [En línea] 1990 . <http://ieeexplore.ieee.org/>.
24. **Vegas, Sira, Juristo, Natalia y Moreno, Ana M.** Técnicas de Evaluación de Software. [En línea] [http://is.ls.fi.upm.es/docencia/erdsi/Documentacion\\_Evaluacion\\_7.pdf](http://is.ls.fi.upm.es/docencia/erdsi/Documentacion_Evaluacion_7.pdf).
25. Ingeniería de software orientada a objetos con Java e Internet. [aut. libro] A WEITZENFELD. 2006.
26. *Framework para integrar el acceso a dispositivos.* **Campo, Adonis Cesar Legón.**

---



---

**GLOSARIO**
**A****API**

Conjunto de funciones residentes en sistema operativo en que permiten que una aplicación se ejecute., 22

**B****BPMN**

Notación para el Modelado de Procesos de Negocio, 22

**C****CMMI**

Modelo Integrado de Madurez de la Capacidad, 15

**D****Diagrama de componentes**

Esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo., 65

**diagrama de despliegue**

Disposición física de los distintos nodos que componen un sistema, 62

**E****ECMA**

Organización internacional basada en membrecías de estándares para la comunicación y la información., 18

**F****Framework**

Conjunto de clases modeladas de forma general para resolver problemas relacionados en un contexto específico., 28

**L****LINQ**

Es un conjunto de características que agregan capacidades de consulta a los lenguajes C# y Visual Basic., 17

**M****Modelo de pruebas**

Implica la operación o aplicación del mismo a través de condiciones controladas y la consiguiente evaluación de la informática., 66

**modelos conceptuales**

Representan la realidad a un alto nivel de abstracción., 21

**MSF**

Microsoft Solutions Framework por sus siglas en Inglés., 15

**R****requisitos no funcionales**

Propiedades o cualidades que el producto debe tener., 40

**W****Windows Communication Foundation**

Es un conjunto de tecnologías .NET para construir y poner en marcha sistemas conectados., 20

**Workstreams**

Secuencia de actividades de trabajo., 16

**X****XML**

Metalenguaje extensible de etiquetas., 18