



Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas



Sistema para la interacción y control centralizado de dispositivos en aplicaciones web v2.0

Autores: Sandy Campos Rodriguez.

Hector Luis Rodríguez Sánchez

Tutores: Msc. Yudenia Ramírez Mastrapa

Ing. Jeandy Bryan Pineda Oro

La Habana, 14 de junio del 2012

DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores del trabajo titulado:

Sistema para la interacción y control centralizado de dispositivos en aplicaciones web v2.0. Y autorizamos 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: Sandy Campos Rodriguez

Autor: Hector Luis Rodríguez Sánchez

Tutor: Msc. Yudenia Ramírez Mastrapa

Tutor: Ing. Jeandy Bryan Pineda Oro

DATOS DEL CONTACTO**Autores:**

Sandy Campos Rodriguez.

- Correo electrónico: scrodriguez@estudiantes.uci.cu.

Hector Luis Rodríguez Sánchez

- Correo electrónico: hlorodriguez@estudiantes.uci.cu.

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.
- Jefa del Departamento de Soluciones de Software del Centro de Identificación y Seguridad Digital.
- Correo electrónico: yudenia@uci.cu.

Ing. Jeandy Bryan Pineda Oro.

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2010.
- Correo electrónico: jbpineda@uci.cu.

DEDICATORIA

Dedicado a esas personas que son el todo en mi vida: mi mamá, mi papá, mi hermana Sadys, mi sobrinito y mis abuelos.

Sandy

A mi familia.

A mi amigo y profe Ulises que hoy no se encuentra entre nosotros.

Hector Luis

AGRADECIMIENTOS

Ante todo agradecer de manera muy especial a mi familia. Ella es lo más importante en mi vida. Gracias a ella estoy aquí.

A mi compañero de tesis y amigo Hector, porque sin él nada hubiese salido a tiempo.

A mis tutores Yudenia y Jeandy, por la gran ayuda que me han brindado durante todo este tiempo.

A los amigos que han compartido conmigo durante estos 5 años y han estado a mi lado en los buenos y malos momentos.

A todo el equipo de desarrollo del proyecto, por su ayuda incondicional, por brindarme su conocimiento y por compartir conmigo tantas líneas de código.

A todas aquellas personas que de una manera u otra han contribuido a hacer este sueño realidad.

A todos muchas gracias.

Sandy

A mi compañero de tesis Sandy por haberme permitido ser parte del DGM.

A mis padres por darme la vida, alentarme y hacerme ver que todo en la vida es posible.

A mi abuelita Delia por malcriarme y llegar a verme lograr este sueño. A mi abuela Maricela por su constancia, apoyo, aliento, esperanza, perseverancia y un sinfín de cualidades que no pudiera mencionar en este documento y ser ese pilar alrededor del cual nos movemos tus hijos y nietos. A Papi por estar siempre presente, hacerme ver cosas esenciales en la vida y brindarme su ayuda aunque no tuviera ni remota idea de lo que es una Clase, un Objeto y mucho menos un Workflow.

A mis tíos Waldo, Mimi y Tato que son padres para mí, especialmente a mi tía Mimi que siempre me dice la verdad aunque sea dura, me aconseja, me apoya y me consiente; que aunque esté lejos, siempre está a mi lado en las buenas y en las malas.

A mi novia por quererme, apoyarme incondicionalmente, soportar mis caras y resabios y por sus innumerables revisiones y correcciones al documento. A mi familia de Cárdenas por acogerme como un hijo.

A todos los compañeros del proyecto en 24, 202 y finalmente en el laboratorio de UCIMININT: el Kinde, Mavilio, Dayana, Arnaldo, Tutty, Yanet, Blanco, Iván, Dainer y la My. Yudith.

A todos los profes que durante estos 6 (5 + 1) años han contribuido a mi educación, especialmente agradezco a los que han sido como familia: Joel, Geidis, Eilén y Damián. A mis tutores Yudenia y Jeandy por su ayuda y empeño a que este trabajo concluyera exitosamente.

A mis hermanos Orlando y Tony por ser como mi familia y acompañarme en todos mis inventos y pruebas de tecnologías. A los compañeros de travesía Sandy, Libán, Guille y el Yuro.

No se me pueden quedar aquellas personas que me ayudaron, alentaron y me hicieron como su familia cuando estuve en Venezuela: Michel, Aiannis, Yuliet, Mely, la Profe Lourdes, Maceo, Yamilet, Aleida, Eduardo, Dayana, Julio y Elena.

A todos los que han estado durante este tiempo y los que no he mencionado, muchas gracias.

Hector Luis

RESUMEN

La interacción con dispositivos conectados a estaciones locales posibilita entre otras cosas, capturar datos biométricos de las personas para utilizarla con distintos fines, entre ellos el de la identificación y seguridad. Sin embargo, en el caso de las aplicaciones web dependen del uso de 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.

Este documento describe los resultados de la investigación para el desarrollo y la implementación de un Sistema para la interacción y control centralizado de dispositivos en aplicaciones web que facilita la integración de dispositivos de hardware en aplicaciones web del lado del cliente, de manera independiente del navegador y garantiza la seguridad, la administración y el control centralizado de los dispositivos.

Actualmente se encuentra en uso por parte del equipo de desarrollo del Sistema Único de Identificación Nacional y en 98 oficinas de carné de identidad del país, permitiendo la interacción con 17 dispositivos de hardware y llevando el control además de las 547 estaciones de trabajo que componen esta red.

Palabras clave: dispositivos, interacción, control, aplicaciones web.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA	6
1.1 Dispositivos de hardware y sus SDK	6
1.2 Aplicaciones web.....	7
1.3 Las aplicaciones web y los dispositivos de hardware	8
1.4 Análisis de las soluciones existentes.....	10
1.5 Propuesta de solución	15
1.6 Ambiente de desarrollo	16
1.7 Conclusiones parciales	29
CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA	31
2.1 Presentación de la solución.....	31
2.2 Modelo de dominio	32
2.3 Descripción del sistema propuesto.....	34
2.4 Procesos	35
2.5 Requisitos del sistema	37
2.6 Conclusiones parciales	46
CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA	47
3.1 Arquitectura de la solución	47
3.2 Patrones de diseño utilizados.....	51
3.3 Especificación de clases	53
3.4 Servicios del sistema.....	57
3.5 Diseño de los workflows utilizados en el Sistema para el control centralizado de dispositivos	59
3.6 Modelo de datos.....	60
3.7 Conclusiones parciales	62
CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS	63

4.1	Estándares de codificación.....	63
4.2	Tratamiento de errores.....	64
4.3	Diagrama de despliegue.....	65
4.4	Implementación.....	66
4.5	Pruebas.....	73
4.6	Análisis comparativo entre la versión 1.0 y 2.0 del DGM.....	79
4.7	Conclusiones parciales.....	82
CONCLUSIONES.....		83
RECOMENDACIONES.....		84
REFERENCIAS BIBLIOGRÁFICAS.....		85
ANEXOS.....		90
	Anexo 1 Documento de aceptación del producto.....	90
	Anexo 2 Fases de MSF for CMMI.....	92
	Anexo 3 Descripción de los requisitos funcionales.....	93
	Anexo 4 Especificaciones de clases.....	112
	Anexo 5 Servicios del sistema.....	120
	Anexo 6 Descripción de los workflows.....	124
	Anexo 7 Descripción del Modelo de datos.....	130
	Anexo 8 Casos de prueba.....	134

ÍNDICE DE FIGURAS

Fig 1.1 Arquitectura del SAGEM MEMS.....	12
Fig 1.2 Aplicación Web usando el DGM v1.0.	14
Fig 2.1 Abstracción del sistema.	31
Fig 2.2 Modelo de dominio.....	32
Fig 2.3 Vista general del sistema.	34
Fig 2.4 Proceso de extensión del sistema.....	36
Fig 2.5 Proceso de uso del sistema.	37
Fig 3.1 Arquitectura del Servicio local para el manejo de dispositivos.....	47
Fig 3.2 Arquitectura de la aplicación para la gestión centralizada de los dispositivos.....	49
Fig 3.3 Representación conceptual de los servicios.....	51
Fig 3.4 Ejemplo de un servicio del Framework DGMJS.....	51
Fig 3.5 Ejemplo de utilización del patrón Inversión de Control.	52
Fig 3.6 Ejemplo de utilización del patrón Singleton.	53
Fig 3.7 Diagrama de clases del requisito Gestionar agrupaciones lógicas.	54
Fig 3.8 ActivationWf utilizado en la activación y desactivación de las estaciones de trabajo.....	60
Fig 3.9 Modelo de datos del sistema DGM.....	61
Fig 4.1 Diagrama de despliegue.	66
Fig 4.2 Diagrama de componentes del Servicio local para el manejo de dispositivos.	68
Fig 4.3 Diagrama de componentes del Sistema para el control centralizado de los dispositivos.....	69

Fig 4.4 Diagrama de componentes del Framework DGMJS.....	70
Fig 4.5 Diagramación general.. ..	71
Fig 4.6 Interfaz del Servicio Local para el manejo de dispositivos.....	72
Fig 4.7 Interfaz del Sistema para el control centralizado de dispositivos.....	73
Fig 4.8 Resultado de la prueba unitaria al servicio EncoderService.. ..	74
Fig 4.9 Resultado de las no conformidades.	79

ÍNDICE DE TABLAS

Tabla 1.1 Sistemas analizados y sus características..	15
Tabla 2.1 Descripción de roles del sistema.	37
Tabla 2.2 Descripción del RF “Generar petición para la ejecución del servicio”.	40
Tabla 2.3 Descripción del RF “Manejar servicios web”.	41
Tabla 2.4 Descripción del RF “Gestionar agrupaciones lógicas”.	41
Tabla 3.1 Descripción de la interfaz IEncoderWCFService..	54
Tabla 3.2 Descripción del servicio EncoderWCFService.	55
Tabla 3.3 Descripción de la interfaz IEncoderConnector.	55
Tabla 3.4 Descripción de la clase DGMConnectorBase.	56
Tabla 3.5 Descripción de la interfaz IEncoderConnector.	56
Tabla 3.6 Descripción de la clase Entidad.	57
Tabla 3.7 Descripción de la clase Group.	57
Tabla 3.8 Descripción de la clase LogicalGrouping.	57
Tabla 3.9 Descripción de la entidad nAgrupacionLogica.	62
Tabla 4.1 Caso de prueba para la funcionalidad Gestionar estaciones de trabajo	75
Tabla 4.2 Resultados de las pruebas.	76
Tabla 4.3 Análisis entre la versión 1.0 del DGM y la versión 2.0.	79

INTRODUCCIÓN

La informatización de los procesos que se llevan a cabo en una empresa o entidad resulta un elemento clave que ocupa un lugar importante en las agendas de trabajo de las mismas. Entre los procesos que resultan prioritarios automatizar por la importancia que tienen en la sociedad se encuentran los relacionados con la identificación y registro de las personas pues de ellos dependerá el eficiente control de los datos de la población.

Cuba ha dado ya los primeros pasos en el desarrollo de la informática en función de brindar un servicio rápido y eficiente a la población en esta área. Es por ello que el Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas (UCI) de conjunto con el Ministerio del Interior (MININT) desarrolla el Sistema Único de Identificación Nacional (SUIN), un sistema informático capaz de gestionar de manera automatizada los procesos relacionados con la identificación y registro de los ciudadanos, llevados a cabo en las oficinas de la Dirección de Identificación y Registro de la población (DIR).

En el desarrollo de una solución de gran envergadura como es el SUIN, las tecnologías web son apropiadas, pues su uso disminuye el costo de la puesta en marcha y mantenimiento de la misma, eliminando a su vez las restricciones del sistema operativo en los clientes, ya que para su correcta ejecución solo se necesitarían un navegador moderno que cumpla con los estándares de la web.

El SUIN entre sus principales funcionalidades tiene la obtención de fotos, huellas y firmas de las personas para su posterior procesamiento y almacenamiento. Durante el desarrollo de estas funcionalidades se hace necesaria la inclusión de dispositivos de hardware del lado del cliente. Cuando se integran dispositivos de hardware en una aplicación web del lado del cliente se presenta un laberinto de soluciones.

Las aplicaciones web y el afán de sustituir por completo las aplicaciones escritorio, se han encontrado una barrera a la hora de integrar dispositivos de hardware con las aplicaciones en los clientes. Las soluciones 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 exista 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, se identifica como **problema de investigación**: ¿Cómo garantizar una comunicación independiente del navegador entre las aplicaciones web y dispositivos de hardware?

Una vez planteado el problema de investigación se define como **objeto de estudio** de la investigación: interacción de aplicaciones con dispositivos de *hardware*.

Con el objetivo de resolver el problema planteado, se definió como **objetivo general**: Desarrollar un sistema de componentes para la integración de dispositivos de *hardware* en aplicaciones web que garanticen la comunicación e independencia del navegador.

Como **campo de acción** se define la interacción con dispositivos de hardware en la web.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas**:

1. Análisis de los principales conceptos asociados a los procesos de comunicación y control de dispositivos para obtener una base teórica necesaria para el desarrollo de la solución. (Sandy Campos Rodriguez, Hector Luis Rodríguez Sánchez).
2. Revisión bibliográfica acerca de los proveedores de hardware, SDK¹ y soluciones existentes que permiten la comunicación con los dispositivos, para determinar fortalezas e insuficiencias en las mismas y valorar la viabilidad de su uso. (Sandy Campos Rodriguez, Hector Luis Rodríguez Sánchez).

¹ Software Development Kit: Kit de Desarrollo de Software.

² Sistema Administrativo de Identificación Migración y Extranjería de la República Bolivariana de Venezuela.

3. Realización de un estudio que permita definir cuáles son las herramientas informáticas y metodologías a usar para el desarrollo del sistema Device Grid Manager (DGM) v.2.0. (Sandy Campos Rodriguez, Hector Luis Rodríguez Sánchez).
4. Análisis de los mecanismos para el manejo de dispositivos del lado del cliente en una arquitectura cliente – servidor para determinar o no su utilización en la propuesta de solución. (Sandy Campos Rodriguez).
5. Especificación de los requisitos del sistema de componentes para el manejo de dispositivos, con el objetivo de determinar las funcionalidades a implementar. (Sandy Campos Rodriguez).
6. Especificación de los requisitos del sistema para el control centralizado de los dispositivos, con el objetivo de determinar las funcionalidades a implementar. (Hector Luis Rodríguez Sánchez)
7. Diseño de la solución del sistema de componentes para el manejo de dispositivos. (Sandy Campos Rodriguez).
8. Diseño de la solución del sistema para el control centralizado de los dispositivos. (Hector Luis Rodríguez Sánchez).
9. Desarrollo de un servicio que permita el manejo de dispositivos de hardware en la estación de trabajo. (Sandy Campos Rodriguez).
10. Desarrollo de un componente que permita la actualización de los servicios en las estaciones de trabajo. (Sandy Campos Rodriguez).
11. Desarrollo de un componente que garantice mecanismos de autenticación – autorización entre los servicios de forma segura. (Sandy Campos Rodriguez).
12. Desarrollo de un componente para la administración centralizada de los dispositivos de hardware distribuidos en las distintas estaciones de trabajo. (Hector Luis Rodríguez Sánchez).
13. Desarrollo de un componente que permita la gestión de servicios por cada estación de trabajo que compone la red de dispositivos. (Hector Luis Rodríguez Sánchez).
14. Aplicación de las pruebas de software al Servicio local de manejo de dispositivos y al Sistema para el control centralizado de dispositivos en función de validar la propuesta de solución. (Sandy Campos Rodriguez y Hector Luis Rodríguez Sánchez).

Para llevar a cabo las tareas de la investigación se emplearon los siguientes **métodos científicos**.

1. **Histórico – lógico:** En un primer momento de la investigación se desarrolló un estudio del estado del arte de la problemática; así como se analizaron las ventajas y desventajas de las herramientas utilizadas actualmente para el control de dispositivos en aplicaciones web.
2. **Analítico-sintético:** Se empleó para el procesamiento de la bibliografía necesaria para llevar a cabo la investigación.
3. **Modelación:** Permitió la creación de modelos (propuestas, alternativas y estrategias) que visualizan una reproducción simplificada de la realidad.
4. **Entrevista:** Este método se evidenció en la realización de entrevistas a personas responsables de desarrollar toda la integración de otros sistemas como SAIME² con dispositivos de hardware, así como las entrevistas realizadas a los arquitectos de software del SUIN y a los clientes.

Los **resultados esperados** con esta investigación son:

- Un servicio para la captura, procesamiento y transmisión de los datos obtenidos desde los dispositivos de hardware hacia el SUIN, con mecanismos que permitan definir con cuáles dispositivos interactuar, actualizar el servicio y almacenar trazas para la realización de reportes.
- Un componente para el monitoreo, control del estado y actualización del servicio de captura, procesamiento y transmisión de datos.
- Un componente para la integración de aplicaciones web con el Servicio local para el manejo de dispositivos.
- Una aplicación que permita la administración de las estaciones de trabajo y los dispositivos instalados en las mismas, así como la generación de reportes según las necesidades del cliente.

² Sistema Administrativo de Identificación Migración y Extranjería de la República Bolivariana de Venezuela.

El presente documento contiene una introducción, cuatro capítulos, conclusiones generales, recomendaciones, bibliografía y anexos. A continuación se describe el contenido que se abordará en cada capítulo:

- **Capítulo 1. Fundamentación teórica.** Este capítulo cuenta con los conceptos fundamentales para comprender la solución así como el respaldo teórico de la misma. Presenta además el estudio realizado de las tecnologías y herramientas a emplear para el desarrollo de la solución.
- **Capítulo 2. Características del sistema.** Se hace un análisis del dominio actual así como los diagramas realizados para la comprensión del mismo. Se presentan además los requisitos funcionales y no funcionales del sistema conjuntamente con sus descripciones, y 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, así como el diseño de la base de datos y demás restricciones del sistema.
- **Capítulo 4. Implementación y prueba.** Se analizan detalladamente todos los componentes que resultan indispensables para el desarrollo de la solución, los diagramas de clases y las relaciones entre los distintos componentes que conforman la misma. Contiene también una vista de despliegue y las pruebas de software aplicadas al sistema.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Un previo estudio del estado y evolución de la temática analizada representa un elemento clave dentro de cualquier investigación. Es necesario poseer una fuerte base teórica que sirva de respaldo a la investigación así como saber manejar los conceptos fundamentales de la misma. El presente capítulo contiene la fundamentación teórica, que da respaldo a la solución propuesta. Se abordan temas relativos a la comunicación con dispositivos de hardware, enfocándose más bien en el uso de estos desde tecnologías web. Se hace además un estudio de sistemas y aplicaciones ya existentes especializadas en la comunicación, control y administración de dispositivos, con vistas a obtener conclusiones parciales acerca de posibilidades de desarrollo en esta área. Se analiza el ambiente de desarrollo propuesto caracterizando las herramientas y tecnologías que se definieron para ser utilizadas en la confección de la solución.

1.1 *Dispositivos de hardware y sus SDK*

En la actualidad los dispositivos de *hardware* juegan un papel crucial en el desarrollo científico técnico, pues permiten a las estaciones de trabajos y a las aplicaciones que se ejecutan en ellas, integrarse en todas las áreas de la ciencia y la técnica. Existen diversos dispositivos que posibilitan la realización de tareas diariamente, como es el caso de la impresora, escáner, micrófono, cámara digital, dispositivos de almacenamiento, entre otros que se encuentran en hogares y centros de trabajo. Coexisten con ellos otro grupo de dispositivos que son utilizados con frecuencia dentro de las soluciones empresariales y se encargan de automatizar tareas más complejas, ejemplo de estos son los lectores de huellas dactilares, tablas electrónicas para captar firmas, relojes para controlar la entrada y salida de personal, lectores de tarjetas, entre otros. (3) (4)

Existen muchas empresas que fabrican y distribuyen estos equipos, tal es el caso de Hewlett-Packard, Sony, Canon, Gemalto, GreenBit (3), por solo citar algunos ejemplos. Dichas empresas además de proveer el hardware, facilitan un paquete de *drivers*³ que hacen posible la comunicación del dispositivo

³ *Drivers*: fragmento de software capaz de proveer una capa de abstracción entre el sistema y un dispositivo determinado.

con la máquina y un software, comúnmente denominado *SDK*, el cual provee al comprador de un grupo de herramientas, librerías de clases y componentes permitiéndole interactuar desde una perspectiva de desarrollo de software con estos dispositivos.

1.2 Aplicaciones web

Las aplicaciones web son aquellas utilizadas por los usuarios accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. (5)

Dichas aplicaciones han ganado en popularidad debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Es por ello que muchas empresas han tomado como estrategia el acceso al software vía web dado a las ventajas que esto trae consigo. Acorde a Web Trends, expertos en tendencias de las tecnologías web, algunas de estas ventajas son:

- Ahorro de tiempo; se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- Evita problemas de compatibilidad, basta tener un navegador actualizado para poder utilizarlas.
- Son multiplataforma, o sea se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Son independientes del ordenador donde se utilice porque se accede a través de una página web.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web cada vez más ricas e interactivas. (5)

Normalmente estas aplicaciones ofrecen menos funcionalidades que las aplicaciones de escritorio, debido a que las posibilidades del navegador web son más restringidas que las del sistema operativo, es por ello que a veces se hace necesario el uso de otras herramientas que permitan suprimir estas limitantes. (6) (7)

1.3 Las aplicaciones web y los dispositivos de hardware

Las aplicaciones web, cuentan del lado del cliente con código HTML, CSS y JavaScript, estos lenguajes de programación en conjunto conforman un *script*⁴ que es descargado e interpretado por el navegador con el objetivo de mostrar la interfaz visual de la aplicación. Una de las limitaciones que poseen los navegadores es que no cuentan con la capacidad de acceder a los recursos locales de la estación de trabajo, incluyendo además a todos los dispositivos de hardware conectados a la misma, esto está definido de esa manera por cuestiones de seguridad. Actualmente existen tecnologías que permiten la interacción de las aplicaciones web con los recursos de la máquina, compañías como Microsoft cuentan con los ActiveX, Oracle con los Applets y la mayor parte de los navegadores cuentan con sus complementos.

1.3.1 ActiveX

Una de las tecnologías que se puede utilizar para implementar parte de las aplicaciones web en el lado del cliente está basada en el uso de controles ActiveX como los que se utilizan en el desarrollo de aplicaciones para Windows. Los controles ActiveX están construidos sobre COM (*Component Object Model*), el modelo de Microsoft para desarrollo de componentes anterior a la plataforma .NET. a diferencia de JavaScript que es un lenguaje totalmente interpretado, los controles ActiveX se compilan previamente, lo que permite su ejecución más eficiente. (7)

Los ActiveX se pueden incrustar en una página web, donde aparece típicamente como un gráfico interactivo. Los controles ActiveX son distribuidos como binarios ejecutables, y deben ser compilado para cada equipo de destino y el sistema operativo. (7) Existen una serie de controles ActiveX disponibles para el Microsoft Internet Explorer.

⁴ Archivo de procesamiento por lotes.

1.3.2 Applets

Un Java Applet es un subprograma escrito en el lenguaje de programación Java. Este puede correr en un navegador web utilizando la *Java Virtual Machine (JVM)*⁵, o en el AppletViewer de Sun Microsystem. (9)

Entre sus características se puede 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 Applet, proporcionar una forma fácil de ejecutar aplicaciones desde el navegador web. (9)

1.3.3 Extensiones de 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 simplemente 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.mozilla.org (10). Otros navegadores como Opera, Internet Explorer y Safari cuentan con una sólida infraestructura para el desarrollo de extensiones. (11)

1.3.4 Limitantes

El principal problema que surge para usar estas tecnologías está dado por las dependencias que crean. Al usar un ActiveX, se está dependiendo de Internet Explorer y al usar Applets de Java se necesita tener instalada la JVM en la computadora.

Por otra parte los navegadores ofrecen complementos que incrementan sus funcionalidades, pero la gran limitante que tienen es que un complemento desarrollado para un navegador es incompatible con otro. Por lo cual se necesitaría desarrollar un complemento para cada navegador, teniendo en cuenta además que

⁵ Máquina Virtual de Java.

existen diferencias entre las versiones de los dichos navegadores, por lo que habría que implementar un complemento compatible con todas las versiones de un navegador específico o limitar el uso de la extensión a una versión específica del navegador web.

Para interactuar con los dispositivos es necesario además que una instancia del navegador se encuentre en ejecución, es decir que cuando el navegador se cierra se pierde la comunicación con el dispositivo, provocando inestabilidad en el funcionamiento de este.

1.4 Análisis de las soluciones existentes

Actualmente en el mundo existen sistemas que permiten la comunicación con dispositivos de hardware; tal es el caso del sistema Axiom DMS, el Software de Administración y Enrolamiento Centralizado SAGEM MEMS, el Device Grid Manager (DGM) v1.0 empleado actualmente para la interacción con los dispositivos de captura biométrica utilizados por el Sistema Único de Identificación Nacional (SUIN).

Resulta oportuno realizar un análisis de las soluciones mencionadas anteriormente con el propósito de evaluar su comportamiento ante la situación problemática planteada.

1.4.1 Axiom Device Management System (DMS)

Axiom DMS es una aplicación de servidor que controla todo tipo de dispositivos utilizados por los diferentes módulos de Axiom Studio Suite. Se comunica con terminales de control de accesos y asistencia y con dispositivos conectados a las terminales. (14)

La aplicación está soportada por los sistemas operativos Windows XP, Windows Vista y Windows Server 2008 y utiliza una base de datos centralizada sobre Microsoft SQL Server 2000, Microsoft SQL Server 2005 y Microsoft SQL Server 2008.

Permite controlar múltiples terminales que a su vez controlan dispositivos como pueden ser lectores biométricos, lectores de tarjeta, puertas, sirenas, luces, etc. La configuración de los dispositivos es administrada por el servidor Axiom DMS, almacenada en la base de datos y transmitida a las terminales. Permite almacenar en el servidor todos los eventos generados por los clientes.

Algunas de las características del sistema Axiom DMS son:

- Controla un número prácticamente ilimitado de dispositivos.
- Administra la configuración y comportamiento de cada dispositivo y almacena dicha información en una base de datos centralizada.
- 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.
- 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. (13)

Uno de los principales problemas del Axiom DMS es que no permite la interacción con los dispositivos de hardware desde las aplicaciones web, viéndose atado a usar aplicaciones de escritorio.

1.4.2 Software de Administración y Enrolamiento Centralizado (SAGEM MEMS)

SAGEM MEMS es un software que ofrece la posibilidad de enrolar a las personas en una base de datos central a través de una estación de trabajo y luego replicar esta información a los distintos dispositivos conectados a la red. También es posible configurar de forma remota los dispositivos y recuperar centralizadamente la información de auditorías (registros de acceso o fichajes). Ver Fig 1.1.

Entre las características más relevantes del sistema se encuentran:

- Interfaz gráfica de fácil uso.
- Opera en una estación de trabajo cliente 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.
- Arquitectura Cliente/Servidor disponible con versión SQL Server y Oracle.
- Recuperación de trazas y generación de reportes.

- Funcionalidad de enrolamiento de huellas dactilares y registro de información básica del personal.
- Compresión de imagen y verificación de huellas.
- Control de accesos de acuerdo a zonas de tiempo.

Para que el sistema SAGEM MEMS pueda distribuir los patrones de identificación a los terminales y a la vez recuperar la información desde un punto central de administración, se requiere que los distintos terminales cuenten con una conexión a una red de comunicaciones de datos. El dispositivo se integra a la red de comunicaciones a través del protocolo estándar TCP/IP. (15) Muy similar al Axiom DMS presenta la desventaja de no poderse integrar a las aplicaciones web, además de su alto precio en el mercado del software que oscila entre 1 000 y 2 000 libras esterlinas. (16)

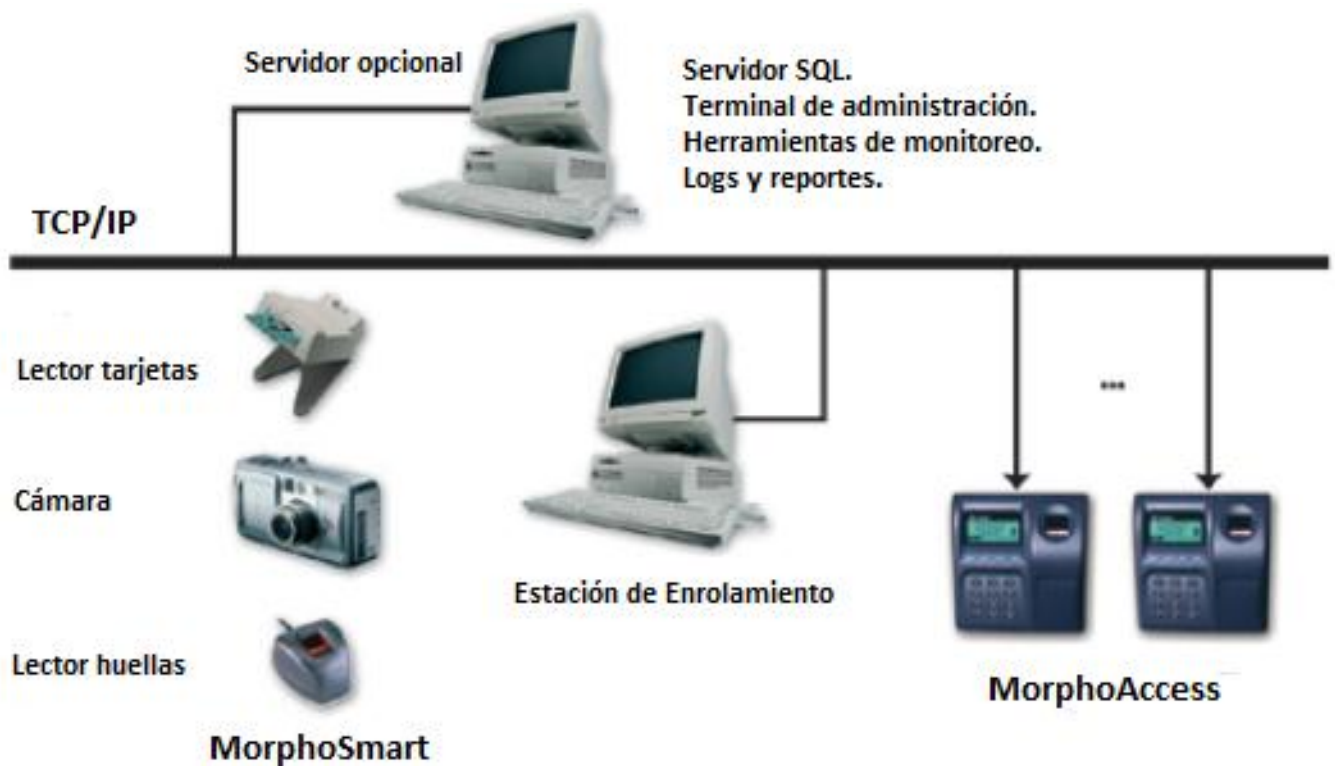


Fig 1.1 Arquitectura del SAGEM MEMS. Fuente: MorphoAccess (TM).

1.4.3 *Device Grid Manager (DGM) v1.0*

El DGM v1.0 es un sistema realizado por el Centro de Identificación y Seguridad Digital de la UCI en colaboración con el Ministerio del Interior de la República de Cuba (MININT). (Ver Fig 1.2) El sistema es usado por el SUIN para la comunicación con los dispositivos de captación biométrica. Fue concebido para interactuar con él, tanto desde aplicaciones de escritorio como desde aplicaciones web y posee un *framework* para integrar el acceso a los dispositivos desde la web extensible y modificable. El DGM v1.0 se basa fundamentalmente en dos aplicaciones, una para la interacción con los dispositivos, la cual se encuentra instalada localmente en las estaciones de trabajo y otra aplicación instalada en un servidor que permite llevar un control centralizado de los dispositivos, aunque esta última aún no está disponible. Entre las características más relevantes del DGM v1.0 se tienen:

- Permite una integración 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.
- Permite la incorporación de nuevos dispositivos en la medida que vayan surgiendo las necesidades para nuevos módulos, además acepta otros equipos con las mismas características que el actual pero de otros fabricantes, de manera que se puedan hacer reemplazos sin hacer cambios en las aplicaciones.
- Es totalmente configurable por lo cual se pueden definir los dispositivos habilitados sin necesidad de recompilar la aplicación aun siendo necesario usar dispositivos de otros fabricantes para realizar las mismas u otras operaciones.
- Permite el acceso a dispositivos desde aplicaciones web a través de librerías que se incluyen en las aplicaciones.
- Contiene complementos que gestionan la autorización para el acceso a los dispositivos. (13) (16)



Fig 1.2 Aplicación Web usando el DGM v1.0. Fuente: Sistema para la integración y el control centralizado de dispositivos en aplicaciones web (13).

Como se ha descrito el DGM v1.0 tiene una serie de características que permiten interactuar con los dispositivos pero aún carece de funcionalidades que posibiliten el control de los elementos de la red. Algunas de las deficiencias que impiden que el DGM v1.0 sea una solución integral en el manejo de dispositivos son:

- No presenta una herramienta que provea una interfaz visual amigable para gestionar la configuración.
- No permite la gestión y el control de los dispositivos que forman parte de la red.
- No presenta un sistema de trazas, por lo cual no es posible la generación y obtención de reportes para determinar el tiempo de uso de los dispositivos y las operaciones realizadas sobre ellos.
- Dependencias del navegador en las aplicaciones web que se comunican con los dispositivos a través del DGM (usa complementos del navegador para llevar a cabo la comunicación).
- Carencia de seguridad en los servicios de intercambio entre el SUIN y el Servicio local para el manejo de dispositivos, puesto que los servicios de comunicación entre los dos sistemas no están

protegidos y por ende las estaciones de trabajo pueden ser suplantadas permitiendo así la explotación de vulnerabilidades por terceras personas.

- No presenta mecanismos que permitan la actualización automática de sus componentes.
- Solo permite la interacción con dispositivos de captación de huellas y firmas, lo cual resulta insuficiente para la realización de todos los procesos de captación biométrica que se llevan a cabo en los procesos de identificación.

1.5 Propuesta de solución

Como se ha visto los sistemas presentados anteriormente poseen cada uno sus ventajas y desventajas sirviendo su análisis como base para un mejor entendimiento de lo que el nuevo sistema deberá realizar. Ver Tabla 1.1.

Tabla 1.1 Sistemas analizados y sus características. Fuente: Elaboración propia.

Sistemas	Control en tiempo real de dispositivos	Reportes de uso	Posibilidades de uso en aplicación web	Extensible y modificable
Axiom DMS	x	x		
SAGEM MEMS	x	x		
DGM v 1.0	x		x	x

Teniendo en cuenta los requerimientos del Sistema Único de Identificación Nacional en cuanto al uso de dispositivos externos, surge la necesidad de crear un *framework* para la interacción con dichos dispositivos de forma sencilla y transparente a la implementación del SUIN. Los *frameworks* son una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Son considerados como una aplicación genérica y configurable a la que se pueden añadir las últimas piezas para construir un sistema concreto. (17)

Las características fundamentales con las que debe cumplir el *framework* de acceso a dispositivos son las siguientes:

Integrable: Se debe integrar de forma fácil y segura a cualquier aplicación o módulo, de manera 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.

Configurable: Debe ser posible a través de configuraciones, preferentemente en el estándar XML, definir los dispositivos habilitados para la captura, de modo que no sea preciso 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.

El estudio previamente realizado demuestra la necesidad de desarrollar una segunda versión del sistema DGM 1.0, ya que es el que más se adapta a las necesidades. Dicha versión debe incluir las funcionalidades que le faltan a la versión 1.0, en aras de hacer del mismo un sistema integral para el manejo y control de dispositivos orientado a la arquitectura web, que provea herramientas para la administración de los dispositivos y su interacción desde aplicaciones web. Utilizando las experiencias de las aplicaciones estudiadas y adicionando nuevas funcionalidades se pretende realizar un nuevo sistema aplicable al SUIN, que permita la interacción con un mayor número de dispositivos, la continua actualización del sistema, la configuración y control centralizado de los dispositivos, la obtención de reportes por parte del usuario y un aumento en la seguridad en el sistema, logrando reducir, con el desarrollo del mismo, el costo por importación al país de un sistema de este tipo.

1.6 Ambiente de desarrollo

El ambiente de desarrollo lo constituyen la metodología, las herramientas y tecnologías empleadas para llevar a cabo la realización del software propuesto, las cuales en este caso fueron definidas por el proyecto Identificación, Inmigración y Extranjería de la República de Cuba atendiendo a las ventajas que provee el empleo de las mismas.

1.6.1 Metodología de desarrollo: MSF for CMMI

Microsoft Solution Framework for CMMI es una metodología para el desarrollo de software así como para la planificación, desarrollo y gestión de proyectos tecnológicos. Está centrada en el modelo de procesos y de equipo. Consta de cinco fases: Previsión, Planificación, Desarrollo, Estabilización e Implementación. (Ver Anexo 2) Cada fase encapsula un conjunto de secuencias de actividades de trabajo y concluye con

un punto de control y cada punto de control brinda una oportunidad para autorizar la continuación o suspensión del proyecto. Además la integración armoniosa de MSF en Team System soporta el desarrollo iterativo rápido con el aprendizaje continuo y el refinamiento. Entre otras características se destacan la escalabilidad, ya que se puede organizar equipos pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más y la flexibilidad dado que se puede utilizar en el ambiente de desarrollo de cualquier cliente. (19)

1.6.2 Notación de modelado: BPMN

*Business Process Modeling Notation*⁶ (BPMN), es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (*workflow*⁷).

Esta notación ha sido diseñada para coordinar la secuencia de eventos y los mensajes que fluyen entre los diferentes procesos participantes, haciendo de BPMN un estándar capaz de expresar más patrones que los diagramas de actividad, pues es gráficamente más rico y con menos símbolos fundamentales, lo que facilita su comprensión por parte del personal no experto; además cubre casi totalmente los patrones de *workflow* con lo cual se le supone una gran expresividad a la hora de especificar procesos. “BPMN define un *Business Process Diagram*⁸ (BPD), que se basa en una técnica de grafos de flujo para crear modelos gráficos de operaciones de procesos de negocio. Estos elementos habilitan el fácil desarrollo de diagramas simples (diagramas de flujo) que serán familiares para la mayoría de analistas de negocio”.

“Con BPD se pueden crear modelos de procesos de negocios internos. El modelado de procesos de negocio suele empezar capturando actividades de alto nivel para luego ir bajando de nivel de detalle dentro de diferentes diagramas. Pueden haber múltiples niveles de diagramas, dependiendo de la metodología usada para desarrollar los modelos. BPMN es independiente de cualquier metodología”. (19)

⁶ Notación para el Modelado de Procesos de Negocio.

⁷ Flujos de trabajo.

⁸ *Business Process Diagram*: Diagrama de Procesos de Negocio.

1.6.3 Lenguaje de modelado: UML

El Lenguaje Unificado de Modelado (UML⁹ 2.0) define un conjunto de notaciones y diagramas estandarizados para modelar sistemas orientados a objetos, además de describir la semántica de lo que estos significan.

UML puede ser empleado para modelar distintos tipos de sistemas y se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Además UML tiene varias ventajas, entre las cuales es posible citar:

- Notación estándar para el análisis y diseño de negocios y sistemas informáticos.
- Independencia de la arquitectura o el lenguaje seleccionado para la realización.
- Incorporación de mejores prácticas a nivel internacional.
- Cuenta con un amplio apoyo entre empresas e instituciones. (20).

1.6.4 Herramienta de modelado: Altova UModel 2009

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 pasándolos a diagramas UML y luego afina los diseños. Entre las características más relevantes de esta herramienta se encuentran:

- Soporte para los 14 tipos de diagramas UML y 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.
- Generación de documentación personalizable de proyecto y colaboración entre proyectos.
- Integración con sistemas de control de versiones y estrecha integración con Visual Studio y Eclipse. (23)

⁹ *Unified Modeling Language* (Language Unificado de Modelado).

1.6.5 Herramienta de modelado de la base de datos: Embarcadero ER/Studio 8.0

El Embarcadero ER/Studio 8.0 es una herramienta que permite el modelado de datos. Incluye funcionalidades de sub-modelado en el nivel de atributos y opciones de comparación. Entre las características de esta herramienta se tienen:

- Separación entre modelos lógicos y físicos.
- Soporte a sistemas gestores de bases de datos heterogéneos.
- Arquitectura flexible.
- Publicación de modelos e informes en una gran variedad de formatos incluyendo HTML, RTF, Microsoft Office y ER/Studio Viewer.
- Generación de esquemas XML desde modelos físicos y lógicos.
- Soporte completo al ciclo de vida de la base de datos.
- Ingeniería inversa y directa.
- Generación automatizada de código de bases de datos.
- Modelado dimensional de los modelos físico y lógico.
- Validación de diseño y formación de la integridad referencial. (24)

1.6.6 Herramientas y tecnologías para el desarrollo integrado

Microsoft Visual Studio 2010

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles.

Microsoft Visual Studio 2010 proporciona un entorno integrado de herramientas e infraestructura de servidor que simplifica todo el proceso de desarrollo de aplicaciones. Ofrece resultados empresariales usando procesos productivos, predecibles y personalizables, y aumenta la transparencia y el seguimiento durante el ciclo de vida con análisis detallados. Aumenta la productividad del equipo usando características de colaboración avanzadas y herramientas de pruebas y depuración integradas para detectar y corregir errores de un modo rápido y sencillo.

Incorpora diferentes tecnologías a las aplicaciones que pueden ser creadas con este. Algunas de estas tecnologías es la compatibilidad con aplicaciones desarrolladas en las versiones Compact, 3.0, 3.5 y 4.0

del .NET Framework, Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF), Windows Workflow Foundation, Silverlight, Windows Forms, ASP.NET, Extensible Application Markup Language (XAML) y Language Integrated Query (LINQ).

Incluye además compatibilidad con diferentes aplicaciones y lenguajes entre ellos Visual Basic, Visual C#, Visual C++, Visual F#, JavaScript, aplicaciones para web, Office, Windows, SharePoint y Smart Device. (18)

Microsoft .NET Framework 4.0

Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

.NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes. (18)

Este *framework* contiene muchas características antiguas y nuevas que se acoplan a las de las versiones anteriores .NET Framework 2.0, 3.0 y 3.5, por ejemplo, los conjuntos de características de Windows Workflow Foundation (WWF), Windows Communication Foundation (WCF) y Windows Presentation Foundation (WPF).

Es neutral en cuanto al lenguaje de programación y funciona en base a librerías. Se puede programar utilizando Visual Basic.NET, Visual C++ .NET C# y F#. Esta neutralidad en el lenguaje de programación es posible gracias a la arquitectura del .NET Framework. Es un componente integral de Windows que admite la creación y la ejecución de la siguiente generación de aplicaciones y servicios web XML. (18)

Esta versión brinda nuevas mejoras y características que lo hacen diferente a sus versiones anteriores, aunque posee la capacidad de funcionar en paralelo con ellas:

- Mejoras en Common Language Runtime (CLR), ASP.NET e innovaciones en los lenguajes Visual Basic y C#.
- Entity Framework, donde se permite a los desarrolladores programar con bases de datos relacionales usando objetos .NET y Language Integrated Query (LINQ).
- Servicios de datos de Windows Communication Foundation (WCF), componente que permite crear servicios y aplicaciones que usen protocolo de datos abierto para exponer y usar datos a través de la web.
- Compatibilidad con formularios para nuevas mejoras en la librería AJAX.

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 transacción segura y de confianza, que se integren en diferentes plataformas y que interoperen con las versiones existentes. (25)

El modelo de programación orientada a servicios de WCF está construida sobre Microsoft .NET Framework y simplifica el desarrollo de sistemas conectados. Además unifica un amplio rango de capacidades de sistemas distribuidos en una arquitectura que se puede componer y extender, abarcando transportes, sistemas de seguridad, patrones de mensajería, cifrados, topologías de redes, y modelos de hospedaje. (26)

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.

WCF se ha diseñado para ofrecer un enfoque manejable para la creación de servicios web y clientes de servicios web. Puede ser utilizado en innumerables ocasiones y se indican a continuación algunos escenarios de ejemplo:

- Un servicio seguro para procesar transacciones comerciales.

- Un servicio que proporciona datos actualizados a otras personas, como un informe sobre tráfico u otro servicio de supervisión.
- Una aplicación de panel que sondea los datos de uno o varios servicios y los muestra en una presentación lógica.
- Exponer un flujo de trabajo implementado utilizando WWF como un servicio WCF.
- Una aplicación de Silverlight para sondear un servicio en busca de las fuentes de datos más recientes. (27)

Windows Workflow Foundation (WWF)

Es una plataforma que permite a los usuarios crear un flujo de trabajo en sus aplicaciones. Consiste en un motor de flujos de trabajo y diseñadores para Microsoft Visual Studio que pueden ser utilizados para modelar escenarios simples como sería mostrar controles de interfaz de usuario basados en entradas de usuarios o escenarios complejos como el procesamiento de pedidos y control de inventarios. WWF contiene un modelo de programación extensible y un diseñador para construir actividades que encapsulan la funcionalidad del flujo de trabajo para usuarios finales o para ser reutilizado a través de múltiples proyectos. (28)

Entre sus principales características están:

- Asignar actividades a las personas de forma automática y según cualquier criterio o carga de trabajo.
- Optimizar la colaboración entre personas que comparten actividades.
- Automatizar o controlar el flujo de documentos datos e imágenes.
- Proveer métricas para responsables de áreas, organizadores, gestores de procesos y calidad tanto para efectos de mejora continua como de indicadores de calidad y gestión.

Además con su uso todo desarrollador será capaz de crear aplicaciones con procesos de negocio complejos que soporten conceptos tales como transacciones, concurrencia, compensación, *tracking* y comunicaciones. (29)

AJAX

Asynchronous JavaScript + XML es la unión de una serie de tecnologías para lograr un fin común. Las características de AJAX en ASP.NET permiten crear rápidamente páginas web para que la experiencia del usuario sea más satisfactoria gracias a elementos de la interfaz de usuario más familiares y receptivos. Estas aplicaciones web enriquecidas que tienen muchas ventajas frente a las aplicaciones web basadas completamente en servidor. Las aplicaciones que usan AJAX ofrecen:

- Mayor eficacia, porque las partes importantes del proceso de una página web se realizan en el explorador.
- Elementos de interfaz de usuario familiares, como indicadores de progreso, información sobre herramientas y ventanas emergentes.
- Actualizaciones parciales de la página, que actualizan sólo las partes de la página web que han cambiado.
- Integración de clientes con los servicios de aplicación de ASP.NET para la autenticación de formularios, funciones y perfiles de usuario.
- Un marco que permite personalizar los controles de servidor para incluir funciones de cliente.

Otro gran elemento a su favor es su compatibilidad con los exploradores más populares y utilizados habitualmente, incluidos Microsoft Internet Explorer, Mozilla Firefox, Google Chrome y Apple Safari. (29)

ADO.NET Entity Framework

Es un conjunto de clases que exponen servicios de acceso a datos para el programador de .NET, ofreciendo abundancia de componentes para la creación de aplicaciones de uso compartido de datos distribuidas. Constituye una parte integral de .NET Framework y proporciona acceso a datos relacionales, XML y de aplicaciones. (31)

ADO.NET Entity Framework permite a los desarrolladores de aplicaciones programar en un modelo de datos de entidad conceptual asignado de manera flexible a un esquema de almacenamiento, o sea permite trabajar con datos en forma de objetos y propiedades específicos del dominio, sin tener que pensar en las tablas de las bases de datos subyacentes y en las columnas en las que se almacenan estos datos. Amplía además la capacidad del modelo conceptual utilizado comúnmente en las aplicaciones

empresariales permitiendo a los desarrolladores escribir código que opera en objetos generados desde un Modelo de Datos de Entidad (EDM)¹⁰ en lugar de hacerlo directamente en el Modelo Lógico (relacional). (32)

Entre las ventajas que proporciona ADO.NET Entity Framework están:

- Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un modelo lógico concreto.
- Las asignaciones entre el modelo conceptual y el modelo lógico específico de almacenamiento pueden cambiarse sin tener que cambiar el código de la aplicación.
- Los desarrolladores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a varios esquemas de almacenamiento.
- Se pueden asignar varios modelos de aplicación a un único esquema de almacenamiento.
- La compatibilidad con LINQ proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.
- La expresividad del modelo de datos es más rica que la de una base de datos relacional tradicional, con características como el establecimiento inflexible de tipos con jerarquías de tipos, relaciones navegables y tipos complejos. (32)

Sistema Gestor de Base de Datos

Oracle Database 11g proporciona funcionalidades que garantizan alto rendimiento, alta escalabilidad, fiabilidad y seguridad mediante el uso de plataformas *Grid*¹¹, asegurando a la vez altos niveles de calidad de servicio e incrementos de la flexibilidad de negocio, reduciendo los costes de explotación. Permite además hacer pruebas de cambios en aplicaciones simulando las cargas reales generadas por los usuarios en los entornos de producción.

¹⁰ Por sus siglas en inglés *Entity Data Model*.

¹¹ Infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos que son propiedad y están administrados por diferentes instituciones.

Entre las características más relevantes se tienen:

- El soporte brinda la disponibilidad sobre varias plataformas (Windows, Linux, Unix).
- La fiabilidad posibilita la continua disponibilidad de las aplicaciones y los datos.
- Es escalable al presentar la condición que tienen los sistemas de adaptarse al crecimiento de trabajo de manera fluida y hacerse más grande sin perder calidad en los servicios ofrecidos.
- La seguridad y protección de datos permite compartir la red de recursos de una empresa con la confianza de que la privacidad se mantiene.
- La auto-gestión de Oracle automatiza muchas de las funciones de infraestructura de modo que un solo Administrador puede administrar cientos de servidores. (33) (34)

ASP.NET

Es un modelo de desarrollo web 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 tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el CLR, entre ellos Visual Basic y C#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del CLR, seguridad de tipos, herencia, etc.

ASP.NET incluye:

- Formularios Web con mayor compatibilidad integrada con el enrutamiento de ASP.NET, compatibilidad mejorada con estándares web, actualización de la compatibilidad de los exploradores, nuevas características para los controles de datos y para la administración de los estados de vista.
- Datos dinámicos con compatibilidad con las aplicaciones web existentes, compatibilidad con las relaciones de varios a varios y con la herencia, nuevos atributos y plantillas de campo, y filtrado de datos mejorado.
- Microsoft AJAX con compatibilidad adicional con las aplicaciones AJAX basadas en cliente de Microsoft AJAX Library.
- Visual Web Developer con IntelliSense mejorado para JavaScript, nuevos fragmentos de código de autocompletar para marcado HTML y ASP.NET, y compatibilidad mejorada con CSS.

- Compatibilidad con múltiples versiones con mejora del filtrado de las características que no están disponibles en la versión de destino de .NET Framework. (34)

1.6.7 Lenguajes de desarrollo

C#

Es un lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que carece de elementos heredados innecesariamente en .NET. Por esta razón, se suele decir que es el lenguaje nativo de .NET.

La sintaxis y estructura es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de Visual Basic.

C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de su biblioteca de clases base (BCL en inglés) usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK. (18)

Algunas de las principales características de este lenguaje son:

- Provee el beneficio de un ambiente elegante y unificado.
- El manejo de errores está basado en excepciones.
- Soporta los conceptos de encapsulación, herencia y polimorfismo de la programación orientada a objetos.
- Soporta los modificadores de acceso *private*, *protected*, *public* y agrega un cuarto modificador *internal*.

- Permite la interoperabilidad con APIs al estilo C y DLLs, esta característica para acceder a APIs nativas es llamada Platform Invocation Services (PInvoke).
- No soporta herencia múltiple, solamente el Runtime .NET permite la herencia múltiple en la forma de interfaces, las cuales no pueden contener implementación.

Language Integrated Query (LINQ)

Es una innovación introducida en Visual Studio 2008 y .NET Framework versión 3.5 que elimina la distancia que separa el mundo de los objetos y el mundo de los datos. (37)

“LINQ convierte una consulta en una construcción de lenguaje de primera clase en C# y Visual Basic. Las consultas se escriben para colecciones de objetos fuertemente tipadas, utilizando palabras claves del lenguaje y operadores con los que se está familiarizado.” (37)

Reduce la complejidad para los desarrolladores, y ayuda a fomentar la productividad. “El objetivo de crear LINQ es permitir que todo el código hecho en Visual Studio (incluidas las llamadas a bases de datos, datasets, XMLs) sean también orientados a objetos, facilitando y estandarizando el acceso a dichos objetos.” (38)

JavaScript

Es un lenguaje de scripts, interpretado, multiplataforma y parcialmente orientado a objetos, muy similar a Java. Fue creado por Netscape específicamente para su uso en el desarrollo de sitios web. Actualmente es un estándar mantenido por el ECMA¹². El código JavaScript puede enlazarse o añadirse a las páginas web proporcionando un control total y dinámico sobre ellas. Es un lenguaje scripting seguro y fiable y aunque sus script tienen capacidades limitadas por razones de seguridad permite controlar hasta cierto punto las aplicaciones que lo ejecutan, habitualmente navegadores (38).

¹² ECMA: Organización internacional basada en membresías de estándares para la comunicación y la información.

CSS

Cascading Style Sheets, u Hojas de Estilo en Cascada es la tecnología desarrollada por la World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación. (40)

Es empleada para darle el estilo a los documentos HTML y XML, separando el contenido de estos de la presentación. Estos estilos son los que definen la forma de mostrar los elementos HTML y XML. CSS le permite a los desarrolladores de páginas web controlar el estilo y el formato de varias de estas al mismo tiempo ya que un cambio en el estilo marcado para un elemento afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

XHTML

Lenguaje de Marcado de Hipertexto Extensible (XHTML por sus siglas en inglés), es una versión más estricta y limpia de HTML, que tiene como principal objetivo 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. (39)

XML

Extensible Markup Language: Lenguaje extensible de marcas no es más que un conjunto de reglas que sirven para definir etiquetas para organizar un documento. Podría decirse que XML es un metalenguaje que permite definir otros lenguajes de etiquetas. (40)

XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello. (43)

PL/SQL

Incorporado en Oracle para resolver las grandes limitaciones que existían con los editores SQL, en cuanto al control de las secuencias de ejecución de instrucciones, el uso de variables, la gestión de posibles errores.

PL/SQL es un lenguaje basado en ADA¹³, por lo que incluye todas las características de los lenguajes de tercera generación, permitiendo así manejar las variables, tener una estructura modular (procedimientos y funciones) y controlar las excepciones, además de soporte para la programación orientada a objetos.

Los programas creados con PL/SQL pueden ser almacenados en base de datos como cualquier otro objeto, quedando disponibles para los usuarios. El uso del lenguaje PL/SQL es imprescindible para construir disparadores de bases de datos. (44)

1.7 Conclusiones parciales

De los aspectos tratados en este capítulo se puede concluir que:

- El estudio de las aplicaciones web y los dispositivos de *hardware* permitió demostrar que es posible acceder a los recursos locales de una estación de trabajo desde las aplicaciones web, pero estos mecanismos acarrearán dependencias de los navegadores y las plataformas en las que se desarrollen.
- El análisis de los sistemas homólogos arrojó que existen sistemas que posibilitan la interacción y el control centralizado de los dispositivos de hardware, pero la mayoría carecen de mecanismos que permitan su utilización desde aplicaciones web. En este análisis se incluye el sistema Device Grid Manager (DGM) v1.0, el cual presenta una serie de deficiencias que impiden su uso pleno, de ahí la necesidad de implementar una nueva versión del mismo que posibilite administrar dispositivos de *hardware* en una arquitectura web y permita la interacción con estos.
- La selección de las metodologías, lenguajes y herramientas del ambiente de desarrollo permiten disminuir el número de artefactos que se generan por el uso de una metodología ágil, el empleo de

¹³ADA: Lenguaje de programación orientado a objetos, fuertemente tipado y concurrente.

las tecnologías seleccionadas influyen en el desarrollo de una solución actualizada, ya que los cambios que se pudieran realizar en futuras versiones de estas tecnologías tendrían la menor incidencia posible en el producto.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se describe la solución que se propone para dar respuesta al problema planteado. Se define el Modelo de dominio actual, describiendo además cada una de las entidades que intervienen en este modelo en aras de lograr un mayor entendimiento del mismo. Se explica con detalles la propuesta del sistema, definiendo cada uno de los subsistemas que lo componen así como un diagrama que provee una vista general de este y se presenta una lista y descripción de los requisitos funcionales y no funcionales que el sistema debe cumplir, obteniendo con estos una concepción general de las funcionalidades.

2.1 Presentación de la solución

La solución de software estará conformada por un conjunto de herramientas que posibilitarán la interacción de la aplicación web con los dispositivos de hardware instalados en las estaciones de trabajo, así como una aplicación que permitirá gestionar y controlar todos los recursos (estaciones de trabajos y dispositivos) de manera centralizada. Su principal objetivo será proporcionar un mecanismo para poder usar los dispositivos empleados para los procesos de captación biométrica desde el SUIN y controlar dichos dispositivos. La Fig 2.1 muestra la interacción entre las aplicaciones externas y los dispositivos de hardware conectados a la estación de trabajo.

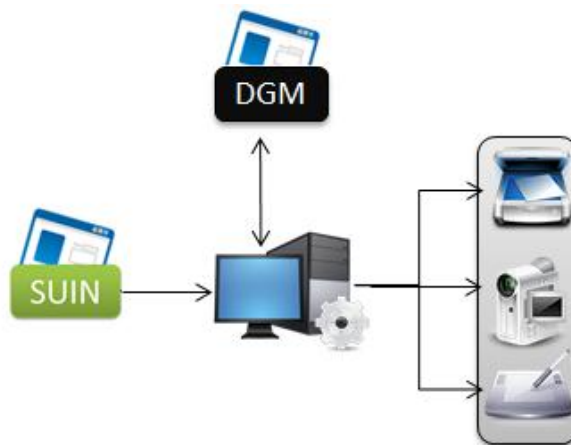


Fig 2.1 Abstracción del sistema. Fuente: Elaboración propia.

2.2 Modelo de dominio

Después de haber hecho el análisis correspondiente no se pudieron identificar claramente los procesos de negocio, sólo elementos conceptuales por lo que se propone un Modelo de dominio. Ver Fig 2.2.

El Modelo de dominio es un artefacto que incorpora comportamientos y datos. Este modelo es la representación visual de las clases conceptuales u objetos del mundo real que reportan interés para la investigación o proyecto. Puede ser empleado en casos donde el negocio no comprende actividades demasiado cambiantes. (45) La determinación de realizar un Modelo de dominio se toma a partir de la falta de claridad en los procesos que se ejecutan como para modelar un negocio, de no tener las fronteras determinadas para este y por no encontrarse claramente definidos quiénes son los responsables de iniciar el negocio y quiénes se benefician del mismo.

Al Modelo de dominio también se le puede conocer como modelo conceptual y emplea el estándar UML para su representación. En su construcción se modela un diagrama de clases compuesto por objetos del dominio o clases conceptuales, relaciones entre estas y los atributos que las componen.

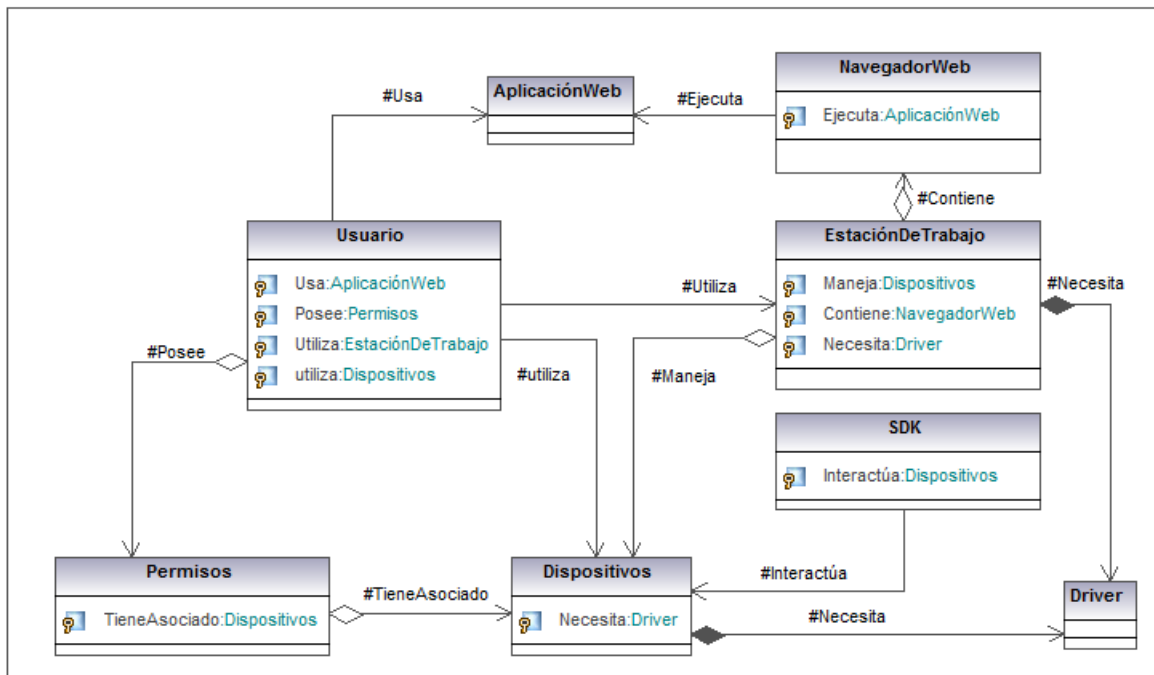


Fig 2.2 Modelo de dominio. Fuente: Elaboración propia.

2.2.1 Glosario de Términos del Modelo de dominio

En este acápite se explicará para un mejor entendimiento cada una de las clases que conforma el Modelo de dominio. (Ver Fig 2.2)

Usuario: Persona encargada de interactuar con la aplicación web y acceder a sus funcionalidades, posee un grupo de permisos, que lo asocian al uso de un grupo de dispositivos.

Estación de trabajo: 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: 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: 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: 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 e imprimir documentos. Estos dispositivos varían en dependencia de su marca, funcionalidades y distribuidor, necesitando *drivers* para poder ser reconocidos en la estación de trabajo a la que se conectan.

Driver: 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: Cuentan con la información acerca de qué usuario puede o no usar un determinado dispositivo.

SDK: 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.

2.3 Descripción del sistema propuesto

El sistema propuesto se compone por tres subsistemas fundamentales: el Servicio local para el manejo de dispositivos, el *framework* JavaScript DGMJS (de aquí en adelante Framework DGMJS) y el subsistema para el control centralizado de dispositivos (de aquí en adelante Sistema para el control centralizado de dispositivos). Estos subsistemas trabajan como un todo, permitiendo la interacción y control de los dispositivos de hardware de una forma cómoda y elegante, basada en un desarrollo estandarizado que facilita la extensión de la aplicación sin grandes esfuerzos por parte de los desarrolladores. En la Fig 2.3 se muestra la vista general del sistema y la interacción entre cada uno de los subsistemas que lo componen.

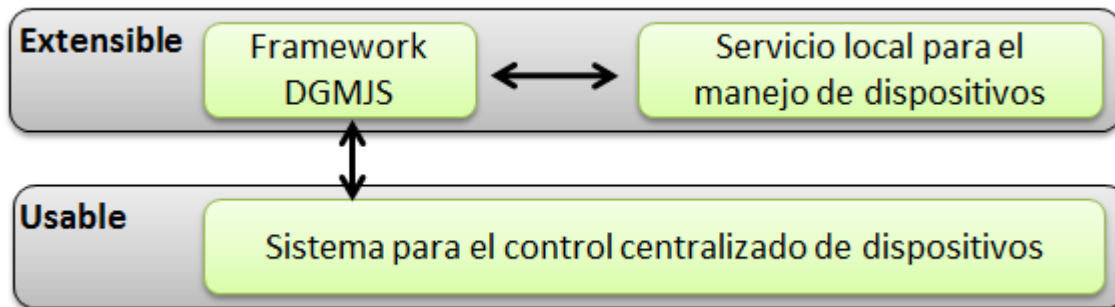


Fig 2.3 Vista general del sistema. Fuente: Elaboración propia.

2.3.1 Descripción del Framework DGMJS

El Framework DGMJS es un componente que cuenta con un grupo de clases que permiten al desarrollador la interacción con servicios publicados en la estación de trabajo donde se esté ejecutando el Servicio local para el manejo de dispositivos. De esta manera se establece una comunicación entre la aplicación web externa y el servicio local, posibilitando así interactuar con los dispositivos. El desarrollo de este *framework* 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.

2.3.2 Descripción del Servicio local para el manejo de dispositivos

Este subsistema cuenta con tres partes fundamentales: el servicio proxy, 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 navegador web que han sido solicitadas por el Framework DGMJS y reenvía estas peticiones al motor de manejo de los servicios para cuando se reciba una respuesta por parte del servicio que maneja el dispositivo, enviársela al navegador web. La segunda parte gestiona las solicitudes llegadas desde el servicio proxy, estas solicitudes traen consigo información la cual conlleva al uso de un cliente determinado para interactuar con los dispositivos. La tercera y última está conformada por el grupo de clientes y controladores, los clientes constituyen la interfaz para el uso de los dispositivos, quienes utilizan los controladores que son los que interactúan directamente con el dispositivo.

2.3.3 Descripción del Sistema para el control centralizado de dispositivos

Este subsistema brinda una interfaz visual en forma de aplicación web que permite al usuario gestionar cada uno de los dispositivos de la red así como ver en tiempo real el estado de estos. Permite además gestionar las localizaciones de los dispositivos y obtener reportes de uso y comportamiento de estos.

2.4 Procesos

El sistema propuesto se caracteriza por el fácil uso y extensión por parte de los desarrolladores, permitiéndose usar en prácticamente cualquier escenario así como integrar otros dispositivos sin la necesidad de emplear grandes esfuerzos por parte de los desarrolladores. En los procesos que se describen a continuación se mostrará la interacción del desarrollador con el sistema en los dos posibles escenarios, aumentando las funcionalidades del sistema para el uso de nuevos dispositivos y en el uso del Framework DGMJS para comunicarse con estos.

2.4.1 Proceso para la extensión del Servicio local para el manejo de dispositivos

En caso que exista la necesidad de agregar un nuevo dispositivo a la red se hace necesario extender el sistema en aras de incluir el dispositivo para su manejo. En este momento el desarrollador instalaría los *drivers* del dispositivo, los cuales deben estar incluidos en un paquete de instalación con el dispositivo o

se pueden encontrar en el sitio oficial del fabricante. Luego crearía una nueva funcionalidad para interactuar con dicho dispositivo, esta funcionalidad la agrega al Servicio local para el manejo de dispositivos y por último actualiza el Framework DGMJS agregándole la configuración necesaria para el uso del dispositivo, quedando completamente actualizado el sistema. En la Fig 2.4 se muestra el proceso de extensión del sistema.

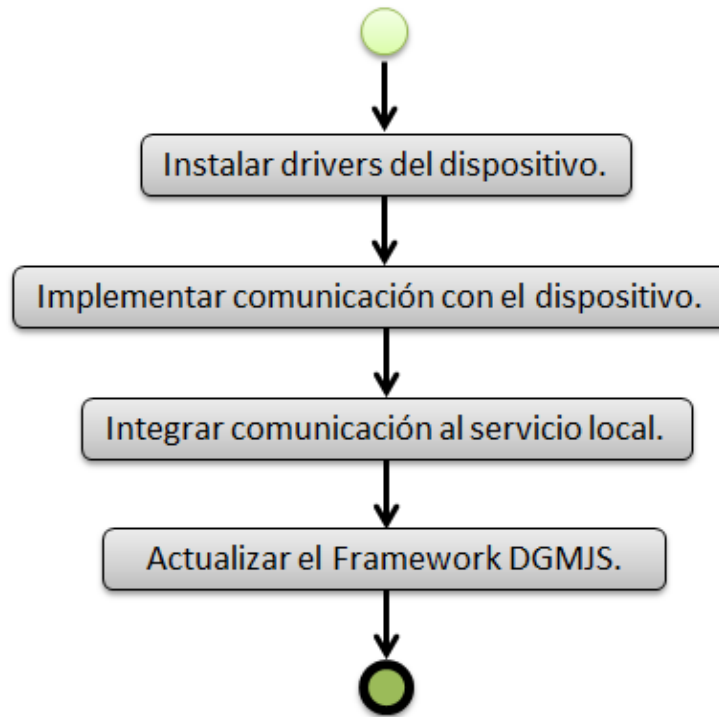


Fig 2.4 Proceso de extensión del sistema. Fuente: Elaboración propia.

2.4.2 Proceso para el uso del sistema

Durante el proceso de uso del sistema el desarrollador solamente debe usar las funcionalidades del Framework DGMJS para interactuar con el dispositivo de hardware que desee desde la aplicación web, ya que el resto de las partes del sistema se encuentran listas para su uso. Ver Fig 2.5.

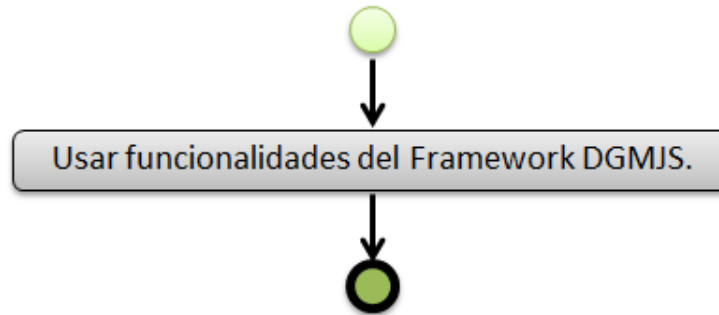


Fig 2.5 Proceso de uso del sistema. Fuente: Elaboración propia.

2.5 Requisitos del sistema

El propósito fundamental del flujo de trabajo de los requisitos dentro de un proceso de desarrollo de software es guiar de manera correcta el desarrollo del sistema. Esto se consigue a partir de una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema (44).

2.5.1 Descripción de los roles del sistema

En la Tabla 2.2 se especifican los roles que intervienen en el sistema y su función en el mismo.

Tabla 2.2 Descripción de roles del sistema. Fuente: Elaboración propia.

Rol	Descripción
Sistema	Realiza todas las operaciones que se ejecutan en el sistema tales como la actualización, generación de trazas, y ejecución de los servicios que se exponen.
Administrador	Es el responsable de ejecutar todas las operaciones de gestión y obtención de reportes en el sistema.
Usuario	Es el responsable de usar los servicios web para la interacción con los dispositivos.
Desarrollador	Es el responsable de desarrollar todas funcionalidades en aras de extender el sistema.

2.5.2 Requisitos funcionales del sistema

Los requisitos funcionales de un software son los que definen qué es lo que el sistema debe hacer. Estos requisitos no son más que las funcionalidades requeridas y restricciones que debe presentar el sistema que son aprobados en mutuo acuerdo con el cliente (45). En el siguiente epígrafe se muestran los

requisitos funcionales del sistema que se propone. Para profundizar en el análisis de los requisitos funcionales del sistema consultar el Anexo 3.

Definición de los Requisitos funcionales del sistema.

Framework DGMJS

- RF 1. Generar petición para la ejecución de un servicio.
- RF 2. Manejar respuestas de las solicitudes a través de eventos.
- RF 3. Transformar las peticiones a un formato entendible.
- RF 4. Configurar llamadas a los servicios locales.

Servicio local para el manejo de dispositivos

- RF 5. Manejar los servicios web.
- RF 6. Aceptar nuevos controladores de dispositivos.
- RF 7. Manejar dispositivos remotos.
 - RF 7.1 Establecer comunicación con un servicio remoto.
- RF 8. Aceptar el manejo de dispositivos remotamente.
- RF 9. Consultar el estado de un servicio remotamente.
- RF 10. Gestionar las peticiones recibidas desde el Framework DGMJS.
- RF 11. Validar el acceso a los servicios web.
- RF 12. Gestionar los servicios web a usar en el Servicio local para el manejo de dispositivos.
- RF 13. Gestionar los dispositivos a usar en el Servicio local para el manejo de dispositivos.
- RF 14. Actualizar el Servicio local para el manejo de dispositivos.
 - RF 14.1 Actualizar los servicios web que se exponen.
 - RF 14.2 Actualizar los controladores de los dispositivos.

Sistema para el control centralizado de dispositivos

- RF 15. Gestionar los servicios y dispositivos de las estaciones de trabajo.
- RF 16. Gestionar agrupaciones lógicas.
 - RF 16.1 Adicionar una agrupación lógica.
 - Datos de la agrupación lógica:

a) Nombre.

b) Tipo.

RF 16.2 Eliminar una agrupación lógica dado su identificador.

RF 16.3 Modificar una agrupación lógica dado su identificador.

Datos de la agrupación lógica:

a) Nombre.

b) Tipo.

RF 16.4 Mostrar los datos de una agrupación lógica dado su identificador.

Datos de la agrupación lógica:

a) Nombre.

b) Tipo.

RF 17. Gestionar agrupaciones físicas.

RF 17.1 Adicionar una agrupación física.

Datos de la agrupación física:

a) Nombre.

b) Es contenedor.

c) Agrupación física padre.

RF 17.2 Eliminar una agrupación física dado su identificador.

RF 17.3 Modificar una agrupación física dado su identificador.

Datos de la agrupación física:

a) Nombre.

b) Es contenedor.

c) Agrupación física padre.

RF 17.4 Mostrar los datos de una agrupación física dado su identificador.

Datos de la agrupación física:

a) Nombre.

b) Es contenedor.

c) Agrupación física padre.

RF 18. Gestionar estaciones de trabajo.

RF 19. Establecer dispositivos en mantenimiento.

RF 19.1 Mostrar dispositivos que necesitan mantenimiento.

RF 19.2 Poner dispositivos en mantenimiento.

RF 20. Generar reportes.

RF 20.1 Generar reportes del estado de los servicios.

RF 20.2 Generar reportes de las estaciones de trabajo.

Descripción de los Requisitos funcionales del sistema.

A continuación se muestra las tablas 2.3, 2.4 y 2.5 en las cuales se describe el RF 1 “Generar petición para la ejecución del servicio”, el RF 5 Manejar servicios web y el RF Gestionar agrupaciones lógicas respectivamente. Las descripciones de los restantes requerimientos se encuentran en el Anexo3.

RF 1. Generar petición para la ejecución de un servicio.

Tabla 2.3 Descripción del RF “Generar petición para la ejecución del servicio”. Fuente: Elaboración propia.

Propósito	Conformar una petición con todos los datos necesarios para solicitar una llamada a un servicio web.	
Roles	Sistema.	
Precondiciones	Debe haber sido solicitada la ejecución de un servicio web.	
Entidades tratadas	Entidad	Atributos
	ServiceConfig	<ul style="list-style-type: none"> • serviceUrl • serviceParameter
	Petition	<ul style="list-style-type: none"> • id • infoServiceCall • event
Descripción	<ol style="list-style-type: none"> 1. Se crea una nueva petición de llamada a un servicio web con los siguientes datos: <ol style="list-style-type: none"> a) Identificador de la petición. b) Dirección del servicio web. c) Evento que se lanzará al obtener la respuesta. 2. Se coloca la nueva solicitud en la cola de solicitudes. 	
Validaciones	1. La URL del servicio web debe estar bien formada.	
Postcondiciones	1. Se actualizó la lista de solicitudes.	
Prototipo	NP	

RF 5. Manejar los servicios web.

Tabla 2.4 Descripción del RF “Manejar servicios web”. Fuente: Elaboración propia.

Propósito	Manejar los servicios web.	
Roles	Sistema.	
Precondiciones	Debe existir el fichero de configuración del sistema.	
Entidades tratadas	Entidad	Atributos
	Services	<ul style="list-style-type: none"> ServiceHost
Descripción	<ol style="list-style-type: none"> Obtener de la configuración del servicio local y extraer de cada servicio la siguiente información: <ol style="list-style-type: none"> Nombre del ensamblado del cliente. Nombre del ensamblado del controlador. Si está activo o no el servicio. Conformar los servicios usando los clientes y los controladores respectivamente de cada uno de los servicios activos. Iniciar cada uno de los servicios conformados. Detener cada uno de los servicios conformados. Detener los servicios tras un tiempo de inactividad. 	
Validaciones	<ol style="list-style-type: none"> Los nombres de los ensamblados que conforman el servicio deben coincidir con ficheros físicos. 	
Postcondiciones	<ol style="list-style-type: none"> Deben estar disponibles los servicios web. 	
Prototipo	NP	

RF 16. Gestionar Agrupaciones lógicas.

Tabla 2.5 Descripción del RF “Gestionar agrupaciones lógicas”. Fuente: Elaboración propia.

Propósito	Gestionar agrupaciones lógicas.	
Roles	Administrador.	
Precondiciones	El administrador debe haber adquirido los permisos necesarios.	
Entidades tratadas	Entidad	Atributos
	Agrupación lógica.	<ul style="list-style-type: none"> Identificador. Descripción. Tipo de agrupación.

Descripción	<ol style="list-style-type: none"> 1. Mostrar listado de agrupaciones lógicas existentes. 2. Mostrar la opción “Adicionar” para crear una nueva agrupación lógica. <ol style="list-style-type: none"> 2.1 Llenar los siguientes datos, si se selecciona la opción “Adicionar”: <ol style="list-style-type: none"> a) Nombre. b) Tipo de agrupación lógica. 2.2 Mostrar las opciones “Aceptar” y “Cancelar”. <ol style="list-style-type: none"> 2.2.1 Guardar la nueva agrupación lógica, si se selecciona la opción “Adicionar”. 2.2.2 Mostrar el listado de agrupaciones lógicas existentes, si se selecciona la opción “Cancelar”. 3. Mostrar la opción “Modificar” para editar una agrupación lógica. <ol style="list-style-type: none"> 3.1 Mostrar los datos de la agrupación lógica en los siguientes campos: <ol style="list-style-type: none"> a) Nombre. b) Tipo de agrupación lógica. 3.2 Mostrar las opciones de “Editar” y “Cancelar”. <ol style="list-style-type: none"> 3.2.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Editar”. <ol style="list-style-type: none"> 3.2.1.1 Guardar los cambios en la agrupación lógica, si se selecciona “Si”. 3.2.1.2 Cancelar la acción si se selecciona “No”. 4. Mostrar la opción “Eliminar” para eliminar una agrupación lógica. <ol style="list-style-type: none"> 4.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Aceptar”. <ol style="list-style-type: none"> 4.1.1 Eliminar la agrupación lógica seleccionada, si se selecciona “Si”. 4.1.2 Mostrar el listado de agrupaciones lógicas, si se selecciona “No”. 5. Mostrar la opción “Cancelar”. <ol style="list-style-type: none"> 5.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Cancelar”. <ol style="list-style-type: none"> 5.1.1 Ir a la página principal si se selecciona “Si”. 5.1.2 Cerrar la alerta si se selecciona “No”.
Validaciones	<ol style="list-style-type: none"> 1. Comprobar que se haya seleccionado una agrupación lógica para poder realizar las acciones “Modificar” o “Eliminar”. 2. Los campos “Nombre” y “Tipo de agrupación lógica” en las opciones “Adicionar” y “Modificar” no pueden estar vacíos. 3. Al “Eliminar” una agrupación comprobar que ésta no contenga dispositivos asociados.
Postcondiciones	Se crearon, modificaron o eliminaron las agrupaciones lógicas.

Prototipo

Gestionar agrupación lógica

Listado de agrupaciones lógicas

	Agrupación lógica	Tipo de agrupación
1	DATYS	Oficina
2	UCI	Oficina
3	Colón1	Oficina
4	Matanzas	Oficina
5	Matanzas 2	Oficina
6	Cárdenas	Oficina
7	Jagüey Grande	Oficina
8	Jovellanos	Oficina
9	Varadero	Oficina
10	Unión de Reyes	Oficina

Página 1 de 20 Mostrando 1 - 10 de 197

Mostrar listado de agrupaciones lógicas

Adicionar agrupación lógica

Nombre Tipo de agrupación lógica

Adicionar agrupación lógica

Modificar agrupación lógica

Nombre Tipo de agrupación lógica

Modificar agrupación lógica

2.5.3 Requisitos no funcionales del sistema

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener y resultan fundamentales en el logro de una solución acorde a las necesidades del cliente. Estos hacen al producto atractivo, usable, rápido y confiable. (47) 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. Existen diferentes categorías para clasificar los requisitos no funcionales, requisitos de software, hardware, restricciones del diseño, implementación, apariencia,

usabilidad, seguridad y soporte son algunas de ellas. A continuación se presenta el conjunto de requisitos no funcionales del sistema.

Requisitos de apariencia

- RnF 1. El sistema debe contar con una interfaz accesible e intuitiva para controlar y mostrar el estado de los servicios al usuario así como para realizar las configuraciones necesarias.
- RnF 2. Los elementos del sistema contendrán claro y bien estructurados los datos, y al mismo tiempo permitirán la interpretación correcta e inequívoca de la información.

Requisitos de usabilidad

- RnF 3. Facilidad de uso de parte del usuario.
- RnF 4. Facilidad de uso de parte de los desarrolladores de software.
- RnF 5. El sistema debe ser fácil de extender por parte de los desarrolladores.

Requisitos de soporte

- RnF 6. El sistema debe ser fácil de implantar en distintos ambientes.
- RnF 7. El sistema debe poseer documentos técnicos.
- RnF 8. El sistema debe poseer el Manual de Usuario.
- RnF 9. El sistema debe poseer el Manual de Instalación.

Requisitos de software

- RnF 10. Sistema Operativo: Windows XP SP3 o superior.
- RnF 11. .NET Framework v4.0 o superior.
- RnF 12. Drivers y SDK de los dispositivos a usar.
- RnF 13. Internet Information Server 7.5.
- RnF 14. ODTwithODAC112012.
- RnF 15. DotConnect para Oracle 6.0.
- RnF 16. Oracle 11g r2.
- RnF 17. Suse Enterprise Edition 10.0.

Requisitos de seguridad

- RnF 18. Garantizar un fuerte tratamiento de excepciones.
- RnF 19. Aplicarle seguridad a los servicios WCF contra posibles ataques.
- RnF 20. Garantizar el control de acceso a las aplicaciones.
- RnF 21. Autenticación segura a partir del hardware.

Requisitos de diseño e implementación

- RnF 22. Utilizar las tecnologías que brinda el .NET Framework sobre la programación orientada a objetos.
- RnF 23. El sistema debe implementarse usando el lenguaje C#.
- RnF 24. El sistema gestor de bases de datos, será Oracle 11g.
- RnF 25. El sistema debe desarrollarse usando el IDE Visual Studio Team System 2010.
- RnF 26. Se utilizará el Microsoft Visual Studio Team Foundation Server 2010 como control de código fuente.
- RnF 27. El paquete de servicios debe codificarse siguiendo los estilos de código definidos por el proyecto. (46)

Requisitos de hardware

- RnF 28. Las estaciones de trabajo cliente deben tener como mínimo 128 Mb de RAM y procesador Pentium IV o superior.
- RnF 29. Las estaciones de trabajo cliente que sean utilizadas para la captura biométrica deben tener como mínimo 512 Mb de RAM.
- RnF 30. El servidor para la instalación del Sistema para el control centralizado de dispositivos debe tener como mínimo 4 Gb de RAM, 20 GB de disco duro disponible para el almacenamiento de la Base de datos y procesador superior 2.40 Ghz.

2.6 Conclusiones parciales

De la realización de cada una de las actividades en el presente capítulo se puede concluir que:

- Del modelo de dominio realizado y la obtención de los principales conceptos y sus relaciones se definieron los elementos fundamentales para el desarrollo de los subsistemas componentes: el Framework DGMJS, el Servicio local para el manejo de dispositivos y el Sistema para el control centralizado de los dispositivos.
- La obtención de 20 requisitos funcionales a partir de la definición del sistema que se propone, permitió identificar las operaciones que podrán ser realizadas y la determinación de 30 requisitos no funcionales del sistema resultan fundamentales en el logro de una solución acorde a las necesidades del cliente.

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

El análisis y el diseño del software son etapas importantes en su ciclo de desarrollo, toda aplicación debe estar soportada por un adecuado análisis y una correcta selección de su arquitectura. El presente capítulo detalla cada uno de los puntos fundamentales dentro del diseño como la arquitectura del sistema, la selección de los patrones de diseño, la especificación de las clases y el modelado de la base de datos.

3.1 Arquitectura de la solución

La arquitectura es la descripción de la organización y estructura de un sistema. Varios niveles de arquitectura intervienen en la creación de sistemas de software, desde la arquitectura física del hardware hasta la arquitectura lógica de un sistema de aplicaciones. (49)

La solución propuesta en cuanto a arquitectura se refiere se divide en dos partes fundamentales, primeramente la arquitectura del Servicio local para el manejo de dispositivos que controla los dispositivos de las estaciones de trabajo y en segundo lugar la arquitectura de la aplicación que permite la gestión centralizada de dichos dispositivos.

La arquitectura del Servicio local para manejo de dispositivos está formada por varias capas que se describen en la Fig 3.1.

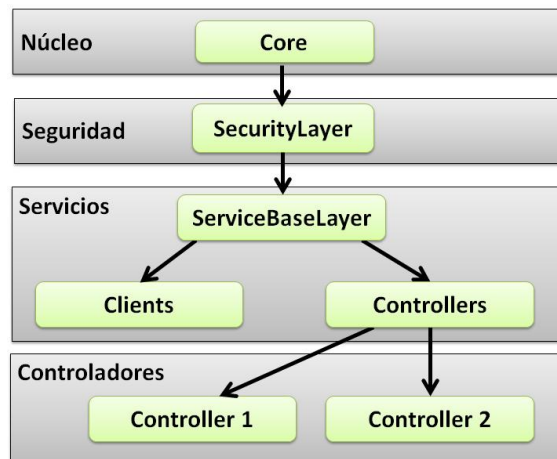


Fig 3.1 Arquitectura del Servicio local para el manejo de dispositivos. Fuente: Elaboración propia.

Núcleo: Esta capa controla toda la lógica del Servicio local para el manejo de dispositivos y dirige el funcionamiento de las demás capas para lograr su objetivo. Instancia cada uno los servicios web que se exponen en el Servicio local para el manejo de dispositivos.

Seguridad: Contiene toda las herramientas que garantizan la seguridad de la aplicación, autoriza las peticiones a los servicios mediante el sistema que permite la gestión centralizada de los dispositivos o por simples reglas definidas en el Servicio local para el manejo de dispositivos. 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 ejecutas por los servicios que corren en el núcleo.

Servicios: Esta capa contiene todos los servicios web dentro del Servicio local para el manejo de dispositivos. Está compuesta por el componente ServiceBase que expone los comportamientos comunes para todos los servicios y por los Clients y los Controllers de cada uno de los servicios. Los Clients se encargan de exponer los métodos necesarios para la utilización de los servicios web y los Controllers definen las operaciones que se realizarán con los dispositivos.

Controladores: Depende de la capa Servicios, es la encargada de implementar la comunicación con los dispositivos.

Por otra parte el Sistema para el control centralizado de dispositivos presenta una arquitectura en N-capas, la misma está estructurada de la siguiente forma: (Ver Fig 3.2).

La **Capa de Presentación** es la encargada de comunicar al usuario con la aplicación, está compuesta por las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Se encuentra representada por el proyecto web de la aplicación y tiene interacción directa con la capas de Servicios y Procesos y con la de Negocio. En ella también se incluye el componente UIToolsBox que posibilita la inclusión de componentes en las páginas que conforman la vista con el objetivo de agilizar el desarrollo de la aplicación y garantizar una interfaz de usuario uniforme.

La **Capa de Servicios y Procesos** contiene la lógica de los procesos de negocio representados por *workflows* (WF), los servicios de intercambio, en la que se incluyen los servicios de tiempo real y los

servicios de negocio, encargados de la interacción con los *workflows* definidos; estos elementos se encuentran vinculados directamente con el *framework* Bison.¹⁴

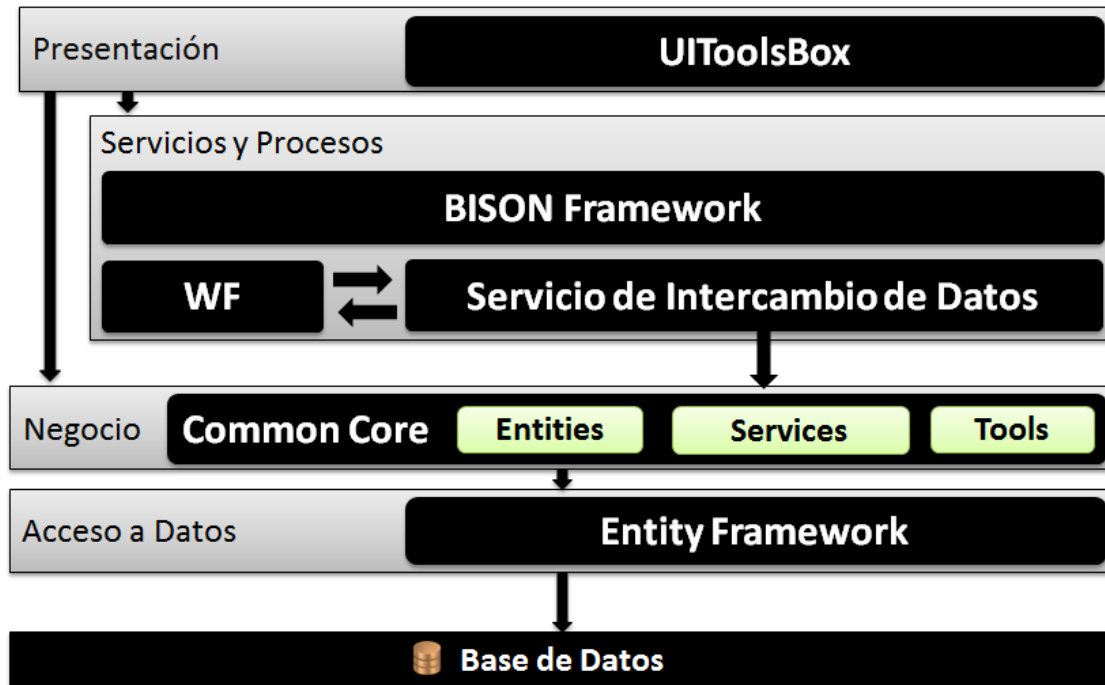


Fig 3.2 Arquitectura de la aplicación para la gestión centralizada de los dispositivos.
Fuente: Documento de la arquitectura del SUIN

La **Capa de Negocio** recoge todos los servicios necesarios para dar solución a los requisitos de negocio que no pueden ser satisfechos por el *workflow*. Los servicios se encuentran definidos según el contexto en el que se desenvuelven y tienen la responsabilidad de manejar todas las operaciones sobre una entidad de negocio en específico, así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. Por cada entidad de negocio se crea un controlador y una interfaz que debe ser implementada por el acceso a dato que le dará soporte.

¹⁴ Es un *framework* para la orquestación de procesos de negocio con Windows Workflow Foundation. Su principal objetivo es proporcionar un componente que permita gestionar las instancias de *workflows*.

La **Capa de Acceso a Datos** está directamente relacionada con los servicios definidos en el negocio. Para establecer esta relación hace uso de las interfaces de conectores que define la capa de negocio. De esta manera es posible realizar cambios en esta capa sin que se vean afectadas las demás capas. Su principal función es realizar una implementación de las interfaces definidas en la capa de negocio y al mismo tiempo trabajar directamente con las fuentes de datos establecidas.

La **Capa de Base de Datos** contiene una pequeña parte de la lógica de las funcionalidades dentro de la aplicación. Por la importancia que posee el desarrollo de procedimientos y vistas dentro del negocio, se decidió separar la base de datos en una nueva capa. La única capa encargada de interactuar con esta es la de Acceso a Datos. (50)

Dentro de la propuesta de solución luego de establecer las arquitecturas del Servicio local para el manejo de dispositivos y de la aplicación que permite la gestión centralizada de estos, se incluyó el Framework DGMJS para lograr la interrelación entre los dos sistemas. El Framework DGMJS 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 respuestas.

El Framework DGMJS está basado en el mismo estilo que utiliza JQuery¹⁵ para la ejecución de llamadas AJAX, facilitando el aprendizaje y la utilización de este. El Framework DGMJS está conformado por dos ficheros, uno contiene toda la lógica de la implementación y el otro se utiliza para representar los métodos y URL de los servicios a ejecutar (Ver Fig 3.3). Además se emplearon alias para facilitar las llamadas a los servicios y métodos. La representación de estos alias se hace mediante un objeto JSON (*JavaScript Object Notation*) el cual forma una estructura jerárquica en forma de árbol la cual queda de la siguiente manera:

¹⁵ JQuery: Es una librería JavaScript que simplifica la interacción entre un documento HTML y JavaScript.

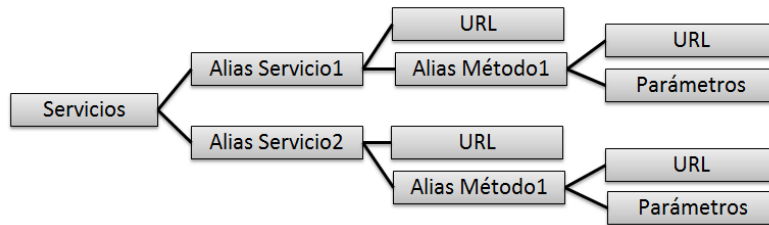


Fig 3.3 Representación conceptual de los servicios. Fuente: Elaboración propia.

La siguiente figura muestra un ejemplo real de cómo quedaría conformado el objeto JSON ya representado en el lenguaje JavaScript usando el mapeo de los servicios.

```

Printer: {
  url: 'http://localhost:4002/PrinterService.Client/',

  Connect: {
    url: 'Connect',
    parameter: []
  },
  Print: {
    url: 'Print',
    parameter: []
  },
  Disconnect: {
    url: 'Disconnect',
    parameter: []
  },
  Status: {
    url: 'Status',
    parameter: []
  },
  PrintDocumentTemplate: {
    url: 'PrintDocumentTemplate',
    parameter: ['instanceId']
  }
},

```

Fig 3.4 Ejemplo de un servicio del Framework DGMJS. Fuente: Elaboración propia.

3.2 Patrones de diseño utilizados

Al iniciar el desarrollo de un sistema de software se hace necesario identificar problemas comunes en el diseño del producto para aplicar soluciones existentes que se apliquen a estos. Por este motivo uno de los

aspectos a tener en cuenta en la creación de una aplicación son los patrones de diseño de software que puedan ser utilizados.

Un patrón de diseño es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. Se entiende también por patrón de diseño a un proyecto o estructura de implementación que logra una finalidad determinada, una manera más práctica de describir ciertos aspectos de la organización de un programa. (49)

En la implementación del sistema se aplicaron los siguientes patrones:

Subclase: propone heredar miembros por defecto de una superclase, las clases hijas tienen los atributos y métodos del padre pero tienen sus propios atributos y funcionalidades. Este patrón se pone de manifiesto en clases como: Group como padre y LogicalGroup como hija. (51)

Inversión de control: La inversión de control es un patrón de diseño usado para desacoplar las clases evitando de esa manera las dependencias fuertes entre ellas. Este patrón es implementado mediante la inyección de dependencias en un componente cuando este es construido. Estas dependencias son comúnmente proveídas como interfaces. (49) Un ejemplo de la utilización de este patrón es durante la creación de la instancia del controlador de un servicio. (Ver Fig 3.5) El campo `_camaraController` es inicializado en el constructor de la clase `CameraClient` con el valor del campo `_controller` definido en la clase padre `BasicClient`.

```
public class CameraClient : BasicClient, ICameraClientService
{
    private readonly ICameraController _camaraController;

    public CameraClient()
    {
        _camaraController = (ICameraController) _controller;
    }
}
```

Fig 3.5 Ejemplo de utilización del patrón Inversión de Control. Fuente: Elaboración propia.

Singleton: Asegura que solo se cree una instancia de la clase y ofrece un punto global de acceso a esta instancia. El uso de este patrón permite que una clase pueda ser instanciada solo una vez y acceder luego a la instancia. (50) Como ejemplo de este patrón se tiene la clase `DGMNotification`. (Ver Fig 3.6)


```
private static DGMNotification _instance;
private static ILogNotification _notifier;

private DGMNotification()
{
    _notifier = new LogsNotification();
}

public static DGMNotification Notifications()
{
    return _instance ?? (_instance = new DGMNotification());
}
```

Fig 3.6 Ejemplo de utilización del patrón Singleton. Fuente: Elaboración propia.

3.3 Especificación de clases

Todo desarrollo de software está marcado por la creación y diseño de clases. Pressman en su libro "*Software Engineering. A Practitioner's Approach*", define como clase una "descripción generalizada (por ejemplo, una plantilla, un patrón o un prototipo) que describe una colección de objetos similares". En "*The Unified Modeling Language. Reference manual.*" el concepto de clase se establece como "descriptor de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y comportamientos". (50) (20)

En el diseño del sistema se expusieron distintos tipos de clases dependiendo de las capas de la arquitectura en las que fueron definidas, entre estos tipos de clases se encuentran: clases conectoras, controladoras, entidades, además de los servicios, interfaces, argumentos y los *workflows*. A continuación se muestra un ejemplo del diagrama de clases para el requisito Gestionar agrupaciones lógicas descrito en el Capítulo II y posteriormente la descripción de cada una de sus clases. (Ver Fig 3.7, Tablas 3.1 - 3.8). El resto de los diagramas y descripciones de las clases se encuentra en el Anexo 4.

3.3.1 Diagrama de clases del requisito Gestionar agrupaciones lógicas

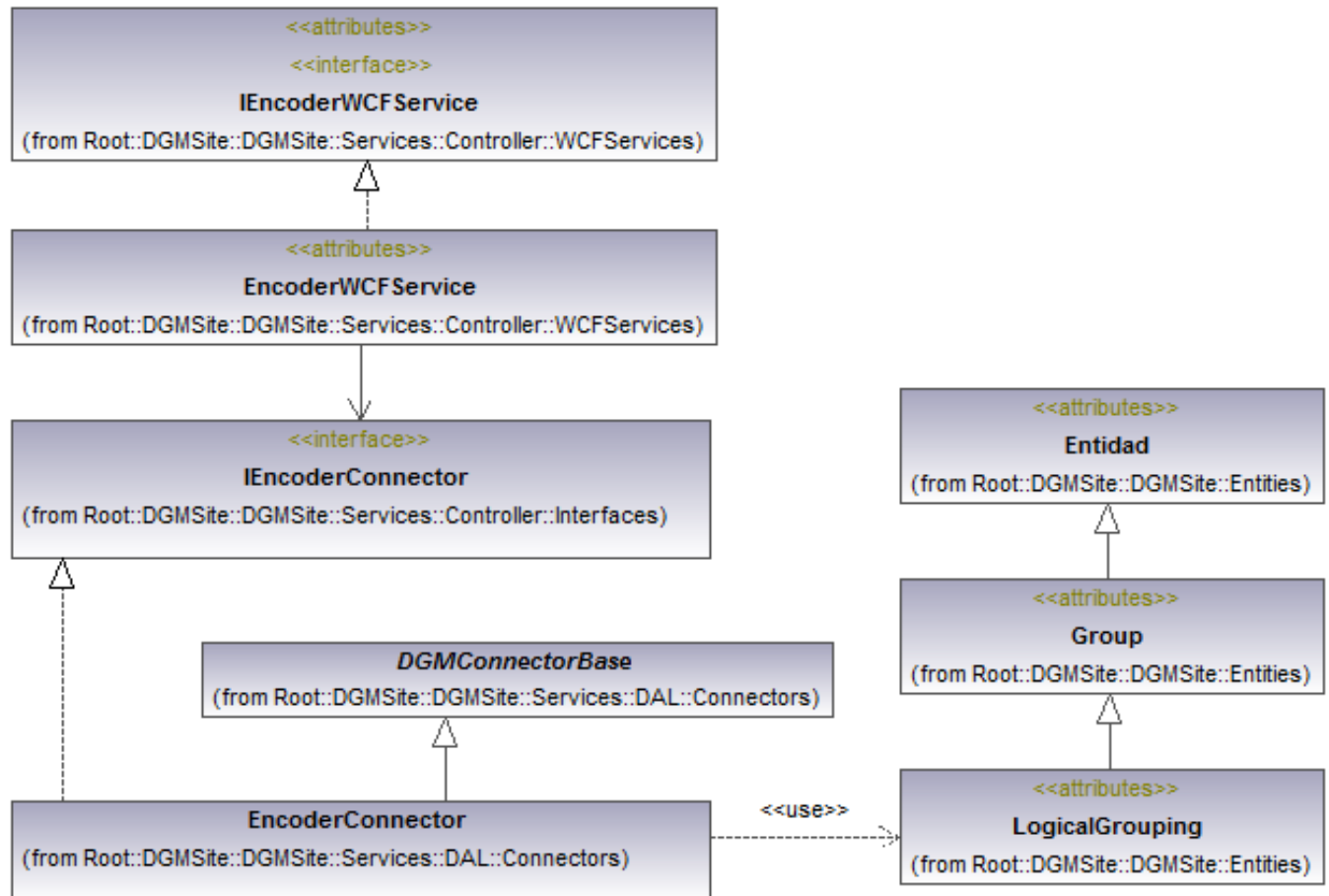


Fig 3.7 Diagrama de clases del requisito Gestionar agrupaciones lógicas. Fuente: Elaboración propia.

A continuación se describen las clases que se muestran en el diagrama de clases de la Fig 3.7.

Tabla 3.1 Descripción de la interfaz IEncoderWCFService. Fuente: Elaboración propia.

Nombre	IEncoderWCFService		
Descripción	Es la interfaz que define los comportamientos para el servicio de nomencladores.		
Visibilidad	Método	Argumentos	Descripción
Público	List<NTIPOAGRUPACIONLOGICA> GetTypesLogicalGrouping		Obtiene los tipos de agrupaciones lógicas.
Público	Hashtable GetTypeLogicalGroupingById	idType	Obtiene el tipo de agrupación lógica dado el id y devuelve un

			hashtable.
Público	List<Hashtable> LoadLogicGrouping		Obtiene todas las agrupaciones lógicas y las devuelve en un hashtable.
Público	InsertLogicalGrouping	logicalGrouping	Inserta una nueva agrupación lógica.
Público	bool UpdateLogicalGrouping	logicalGrouping	Actualiza una agrupación lógica.
Público	bool DeleteLogicalGrouping	logicalGrouping	Elimina una agrupación lógica.

Tabla 3.2 Descripción del servicio EncoderWCFService. Fuente: Elaboración propia.

Nombre	EncoderWCFService		
Descripción	Es el servicio de nomencladores que implementa los comportamientos definidos en la interfaz IEncoderWCFService.		
Visibilidad	Método	Argumentos	Descripción
Público	List<NTIPOAGRUPACIONLOGICA> GetTypesLogicalGrouping		Obtiene los tipos de agrupaciones lógicas.
Público	Hashtable GetTypeLogicalGroupingById	idType	Obtiene el tipo de agrupación lógica dado el id y devuelve un hashtable.
Público	List<Hashtable> LoadLogicGrouping		Obtiene todas las agrupaciones lógicas y las devuelve en un hashtable.
Público	InsertLogicalGrouping	logicalGrouping	Inserta una nueva agrupación lógica.
Público	bool UpdateLogicalGrouping	logicalGrouping	Actualiza una agrupación lógica.
Público	bool DeleteLogicalGrouping	logicalGrouping	Elimina una agrupación lógica.

Tabla 3.3 Descripción de la interfaz IEncoderConnector. Fuente: Elaboración propia.

Nombre	IEncoderConnector		
Descripción	Es la interfaz que define los comportamientos para el servicio de nomencladores en la capa de acceso a datos.		
Visibilidad	Método	Argumentos	Descripción
Público	List<NTIPOAGRUPACIONLOGICA> GetTypesLogicalGrouping		Obtiene los tipos de agrupaciones lógicas.
Público	Hashtable GetTypeLogicalGroupingById	idType	Obtiene el tipo de agrupación lógica dado el id y devuelve un hashtable.
Público	List<Hashtable> LoadLogicGrouping		Obtiene todas las agrupaciones lógicas y las devuelve en un hashtable.

Público	InsertLogicalGrouping	logicalGrouping	Inserta una nueva agrupación lógica.
Público	bool UpdateLogicalGrouping	logicalGrouping	Actualiza una agrupación lógica.
Público	bool DeleteLogicalGrouping	logicalGrouping	Elimina una agrupación lógica.

Tabla 3.4 Descripción de la clase DGMConnectorBase. Fuente: Elaboración propia.

Nombre	DGMConnectorBase		
Descripción	Es la clase base para todos los conectores a la base de datos, cuenta con un grupo de funcionalidades que facilitan la conversión de 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.		
Visibilidad	Método	Argumentos	Descripción
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 3.5 Descripción de la interfaz IEncoderConnector. Fuente: Elaboración propia.

Nombre	EncoderConnector		
Descripción	Es la clase que implementa los comportamientos definidos por la interfaz IEncoderConnector en la capa de acceso a datos.		
Visibilidad	Método	Argumentos	Descripción
Público	List<NTIPOAGRUPACIONLOGICA> GetTypesLogicalGrouping		Obtiene los tipos de agrupaciones lógicas.
Público	Hashtable GetTypeLogicalGroupingById	idType	Obtiene el tipo de agrupación lógica dado el id y devuelve un hashtable.
Público	List<Hashtable> LoadLogicGrouping		Obtiene todas las agrupaciones lógicas y las devuelve en un hashtable.
Público	InsertLogicalGrouping	logicalGrouping	Inserta una nueva agrupación lógica.
Público	bool UpdateLogicalGrouping	logicalGrouping	Actualiza una agrupación lógica.
Público	bool DeleteLogicalGrouping	logicalGrouping	Elimina una agrupación lógica.

Tabla 3.6 Descripción de la clase Entidad. Fuente: Elaboración propia.

Nombre	Entidad	
Descripción	Es la clase que define los atributos base del resto de las entidades del negocio.	
Visibilidad	Atributo	Descripción
Público	Id	Identificador de la entidad.
Público	Description	Nombre o descripción de la entidad.
Público	IsActive	Campo lógico que define si el registro está activo.

Tabla 3.7 Descripción de la clase Group. Fuente: Elaboración propia.

Nombre	Group	
Descripción	Es la clase base que agrupa las agrupaciones físicas y lógicas del negocio.	
Visibilidad	Atributo	Descripción.

Tabla 3.8 Descripción de la clase LogicalGrouping. Fuente: Elaboración propia.

Nombre	LogicalGrouping	
Descripción	Es clase que define los atributos de una agrupación lógica dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	IdLogicalGroupType	Identificador del tipo de agrupación lógica.

3.4 Servicios del sistema

Los Servicios Web son una tecnología basada en estándares e independiente de la plataforma. Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer, como el modo de interactuar con ella. Realizan una tarea concreta, la cual puede corresponder a un proceso de negocio muy sencillo o pueden acoplarse dentro de una aplicación completa que proporcione servicios de alto nivel, con un grado de complejidad muy superior. (53)

Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse y recombinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento en el seno de una organización. (53)

A continuación se mencionan los servicios utilizados en cada una de las aplicaciones que conforman el Sistema para la interacción y control centralizado de dispositivos en aplicaciones web. La descripción de estos servicios se encuentra en el Anexo 5.

3.4.1 Descripción de los servicios web utilizados por el Servicio local para el manejo de dispositivos y el Sistema para el control centralizado de dispositivos

EncoderWCFService: Servicio encargado de realizar las acciones sobre los nomencladores.

ProblemReportWCFService: Su función es recibir los reportes de los problemas generados enviados por el Servicio local para el manejo de dispositivos en cada una de las estaciones de trabajo.

PcInformationWCFService: Este servicio recibe la información de una estación de trabajo para luego devolverla a la aplicación que la solicite. Es usado para transmitir de forma segura la información de una estación de trabajo.

NotificationWCFService: Es el encargado de recibir las notificaciones generadas por el Servicio local para el manejo de dispositivos en las estaciones de trabajos y hacerlas persistir en la base de datos.

UpdaterWCFService: El servicio de actualizaciones le permite al Servicio local para el manejo de dispositivos en las estaciones de trabajo obtener las actualizaciones del sistema.

LocationWCFService: Es el encargado de devolver la ubicación física de una o varias oficinas y de las estaciones de trabajo.

WorkstationWCFService: Es el responsable de realizar las acciones sobre las entidades relacionadas con las estaciones de trabajo.

3.4.2 Descripción de los servicios de negocio utilizados por el Sistema para el control centralizado de dispositivos

Los servicios que a continuación se describen pertenecen a la capa de negocio y garantizan el intercambio de los datos en la aplicación, asegurando la independencia entre las distintas capas de la arquitectura y aumentando la extensibilidad de la misma.

ActivationRService: Servicio del Runtime en el cual se definen los eventos que son invocados para navegar por las distintas actividades de los *workflows* y las responsabilidades correspondientes a la activación y desactivación de las estaciones de trabajo así como el envío de las entidades y las acciones que se realizarán sobre estas (Adicionar, Modificar y Eliminar).

ManageEntityEM: Servicio del negocio que es invocado desde los *workflows* que componen la aplicación para realizar las acciones sobre las entidades de la base de datos invocando a los servicios del sistema mencionados en el epígrafe anterior.

La descripción de estos servicios se encuentra en el Anexo 5.

3.5 Diseño de los workflows utilizados en el Sistema para el control centralizado de dispositivos

Los flujos de trabajo (*workflows*) describen en mayor parte el proceso de negocio que desarrolla la aplicación. El Sistema para el control centralizado de dispositivo contiene dos *workflows* que controlan el negocio de la aplicación. Uno de los *workflows*, “ActivationWf” es el que maneja el proceso de activación y desactivación de las estaciones de trabajo dentro de la red de dispositivos que componen el DGM. El otro flujo de trabajo “ManageEntitiesWf” es utilizado para gestionar los datos de las entidades que componen el sistema. En la Fig 3.8 se muestra el diseño del *workflow* “ActivationWf” que modela las funcionalidades de activación y desactivación de las estaciones de trabajo, en el Anexo 6 se muestran en detalle los *workflows* de la aplicación.

Los *workflows* están integrados por distintos componentes y actividades que posibilitan modelar los procesos de negocio de la manera más cercana posible a la realidad. En la realización de estos se utilizaron componentes incluidos en el Microsoft .NET Framework 3.5 y otros componentes que se incluyen en el Bison Framework desarrollado por el Proyecto Identificación, Inmigración y Extranjería de la República de Cuba; las especificaciones de los componentes del Bison Framework se encuentran descritos en el Documento de Arquitectura del proyecto antes mencionado. (50)

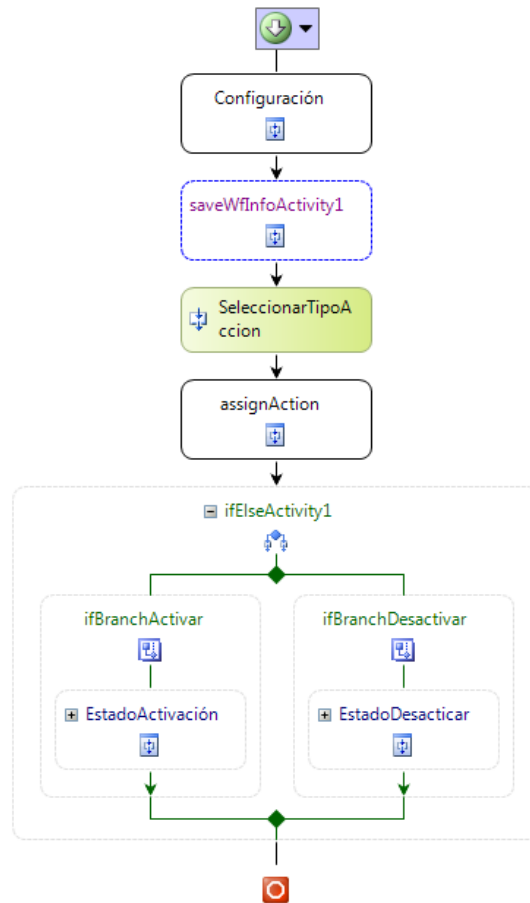


Fig 3.8 ActivationWf utilizado en la activación y desactivación de las estaciones de trabajo. Fuente: Elaboración propia.

3.6 Modelo de datos

Un modelo de datos permite describir las estructuras de datos de la base (el tipo de los datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base). En un enfoque más amplio, un modelo de datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí. (54)

En la Fig 3.9, se muestra el modelo de datos del sistema. Este diagrama está compuesto por las entidades que permiten almacenar los datos de las agrupaciones lógicas y físicas, las entidades que

almacenan los datos de los distintos dispositivos que interactúan con el sistema, los servicios y las entidades que representan los nomencladores del sistema.

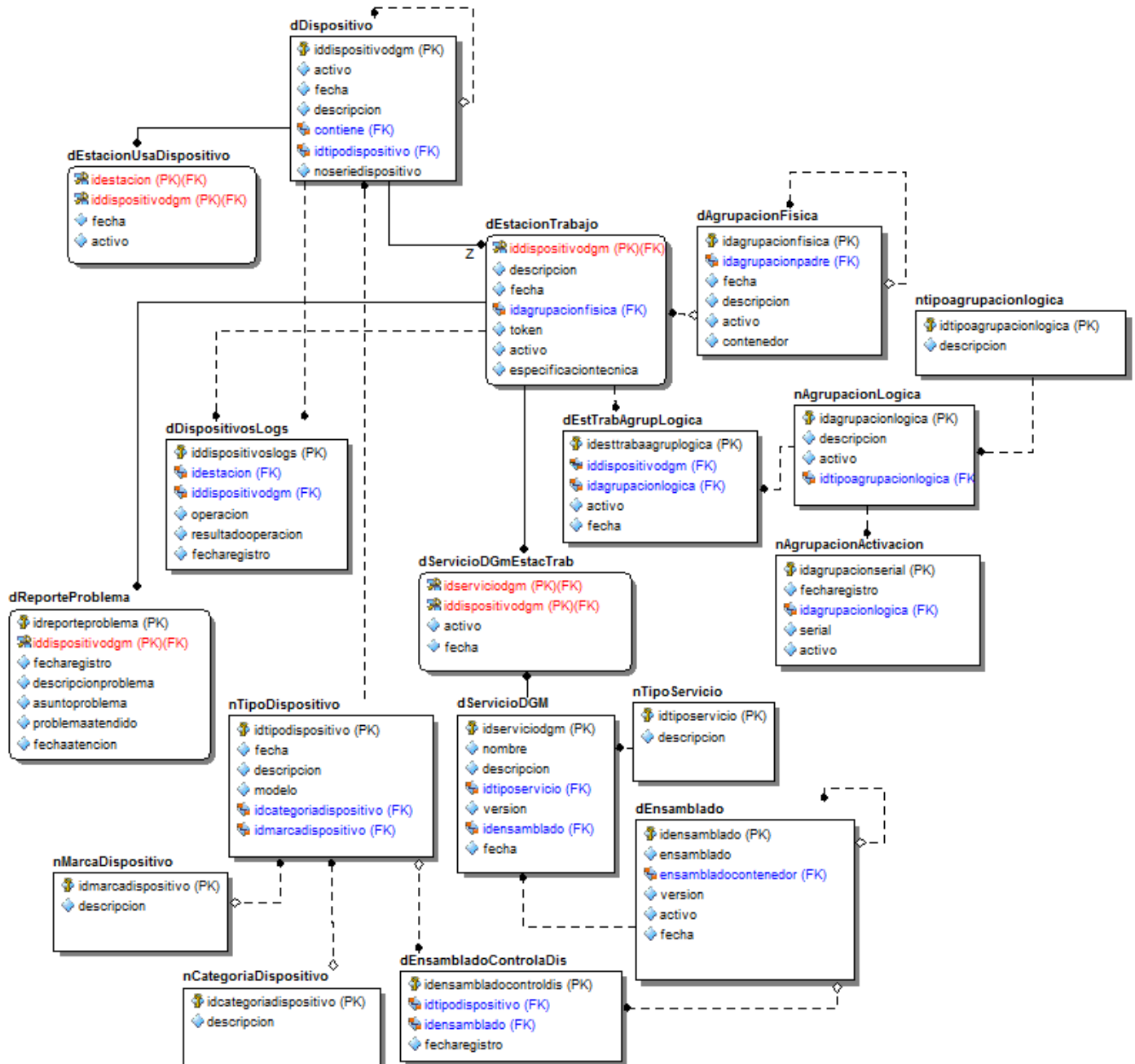


Fig 3.9 Modelo de datos del sistema DGM. Fuente: Elaboración propia.

3.6.1 Descripción de las entidades fundamentales

A continuación, en la Tabla 3.9 se describe la entidad nAgrupacionLogica, presente en el modelo de datos. (Ver Fig 3.9) El resto de las entidades se encuentran descritas en el Anexo 7.

Tabla 3.9 Descripción de la entidad nAgrupacionLogica. Fuente: Elaboración propia.

Entidad	nAgrupacionLogica		
Descripción	Entidad que representa una agrupación lógica.		
Atributos	Tipo de Dato	Nulo	Definición
idagrupacionlogica	NUMERIC(8, 0)	N	El identificador de la agrupación lógica.
descripcion	VARCHAR(80)	N	La descripción de la agrupación lógica.
activo	BIT	N	Si la agrupación lógica está activa o no.
idtipoagrupacionlogica	NUMERIC(8, 0)	N	Dominio del tipo de dato numérico para las secuencias que se generan como identificadores de los nomencladores.

3.7 Conclusiones parciales

El análisis y diseño del sistema es un punto de partida hacia la implementación del producto, los elementos tratados en este capítulo permitieron concluir que:

- La definición de la arquitectura y los patrones de diseño a utilizar en la realización de la propuesta de solución, propician la uniformidad entre los componentes de software que se obtendrán además de una guía en el desarrollo del software.
- Con la definición de los servicios de la aplicación se garantiza un mecanismo de interacción entre los componentes de la arquitectura del sistema para el control centralizado de dispositivos y también entre las aplicaciones externas y el servicio local.
- La correcta definición del modelo de datos garantiza la persistencia de la información que se procesa, con estos fines se obtuvieron 18 entidades que conforman la base de datos.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

Las actividades de implementación y pruebas abarcan gran parte de las acciones más importantes para el desarrollo de un producto de software. Primeramente la implementación en sí, con todo el proceso de formación de la solución, que incluye el establecimiento de los estándares de codificación, el tratamiento de los errores, la elaboración de los componentes y sus relaciones, el desarrollo de cada una de las interfaces según las necesidades del cliente, hasta el diseño del futuro despliegue del sistema. En segundo lugar las pruebas encargadas de obtener todos los errores cometidos durante el desarrollo del producto. El presente capítulo refleja cada una de las etapas de las mencionadas actividades.

4.1 Estándares de codificación

Los estándares de codificación son generalmente utilizados para garantizar la uniformidad de los elementos que se diseñan e implementan cuando se desarrolla un software. Su uso facilita la comprensión y el entendimiento entre los integrantes del equipo de desarrollo y mejora la interpretación del código de programación. A continuación se reflejan los estándares definidos para la creación del Device Grid Manager (DGM).

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 sensitive*.
- No se deberá crear dos espacios de nombres que se diferencien sólo en el uso de las mayúsculas.
- No se deberá crear funciones con nombres de parámetros que se diferencien sólo en el uso de la mayúscula.
- No se deberá crear espacios de nombres con nombres de clases que se diferencien sólo en el uso de las mayúsculas.
- No se deberán crear clases con propiedades que se diferencien sólo en el uso de las mayúsculas.
- No se deberán crear clases con métodos que se diferencien sólo en el uso de las mayúsculas.

Evitando confusión de nombre y tipo:

Se utilizan nombres que describen a sus identificadores en vez de nombres que describen el tipo de identificador. (46)

4.2 Tratamiento de errores

Para garantizar una mayor integridad y confiabilidad de la información que se procesa en el sistema se realizan diferentes acciones, las cuales se mencionan a continuación:

- Implementación de clases para el lanzamiento y la captura de excepciones.
- Validación de los campos de los formularios pertenecientes a las interfaces relacionadas con el Sistema para el control centralizado de dispositivos, haciendo uso de los componentes ASP.NET `RegularExpressionValidator` y `RequiredFieldValidator` para definir las expresiones regulares y detectar los campos vacíos respectivamente.
- Los mensajes de error mostrados en el sistema deben estar escritos en un lenguaje comprensible para los usuarios.
- Luego de introducir información errónea o dejar campos vacíos en los formularios se mostrará un mensaje y se resaltaré el campo indicando el error.

4.3 Diagrama de despliegue

El diagrama de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software. (45) Para la propuesta de solución se estableció el diagrama de despliegue representado en la Fig 4.1, el cual está compuesto por dos servidores, el primero de ellos será el Servidor de Aplicaciones y contendrá el Sistema para el control centralizado de dispositivos. Este servidor tendrá instalado Internet Information Service 7.5, Microsoft .NET Framework 4.0, Oracle Client y DevartOraDirect¹⁶. El segundo servidor contendrá la base de datos del sistema y deberá poseer Oracle 11g. La comunicación entre dichos servidores será a través de TNS¹⁷.

El diagrama representa además dos estaciones de trabajo que serán nodos clientes en los cuales estará desplegado el Servicio local para el manejo de dispositivos y tendrán comunicación vía USB con los dispositivos de hardware conectados a ellas. Los nodos clientes tendrán comunicación con el Servidor de Aplicaciones a través del protocolo HTTPS. Entre los nodos clientes se establece una comunicación usando el protocolo REST¹⁸, en dichos nodos deberá estar instalado el Microsoft .NET Framework 4.0.

¹⁶ DevartOraDirect, también conocido como OraDirect .NET es un ORM que posibilita a desarrolladores de aplicaciones .NET realizar aplicaciones que utilicen bases de datos Oracle.

¹⁷ TNS: Nombre por el que se conocen las instancias de una base de datos de Oracle en una red.

¹⁸ *Representational State Transfer*: técnica de arquitectura de software para sistemas hipermedia distribuidos como la *World Wide Web*.

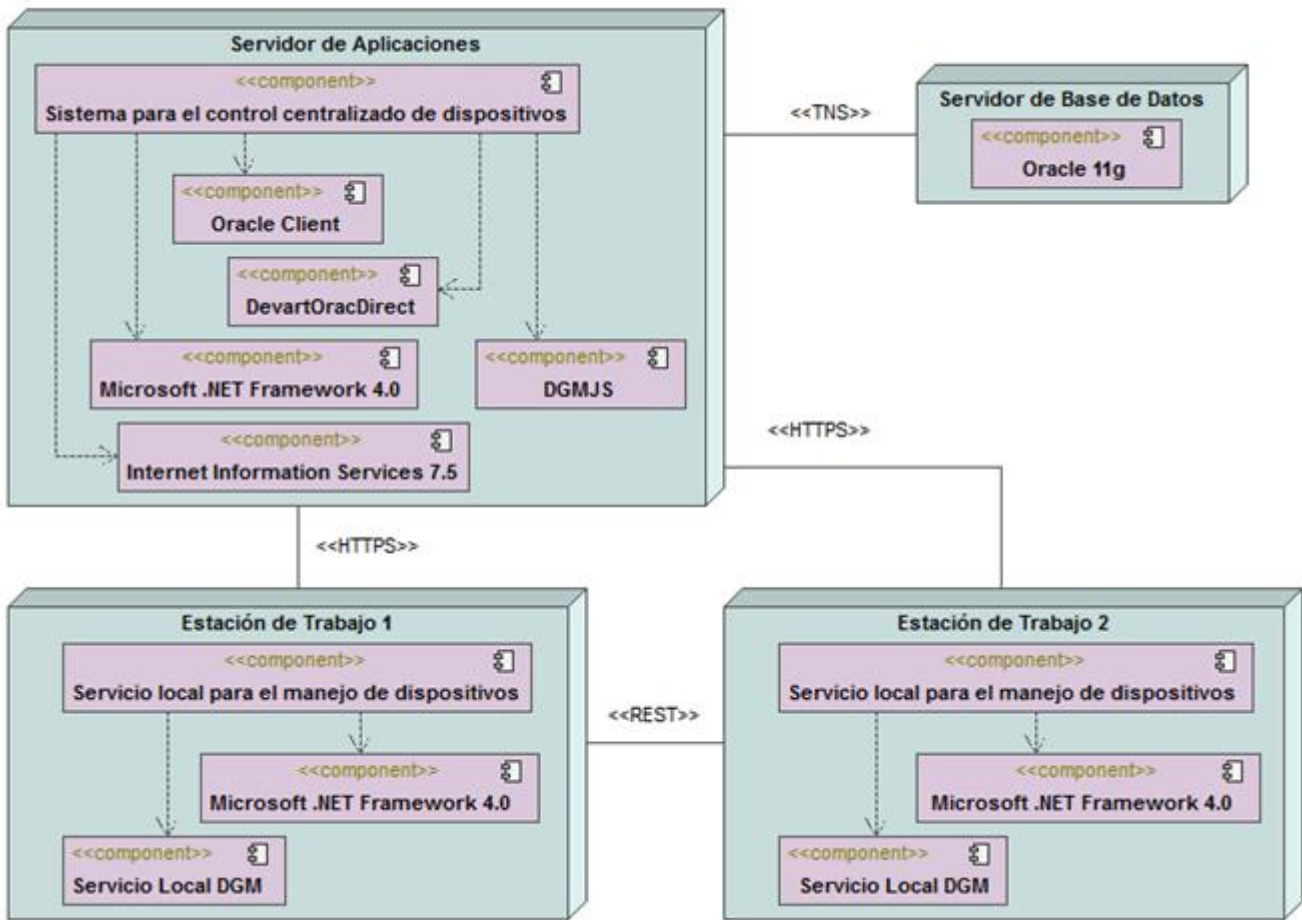


Fig 4.1 Diagrama de despliegue. Fuente: Elaboración propia.

4.4 Implementación

Durante la implementación se desarrolla el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Uno de sus objetivos fundamentales es lograr implementar las clases y subsistemas. En particular, las clases se implementan como componentes de ficheros que contienen código fuente. (45)

Para la implementación de la propuesta de solución se tuvo en cuenta el desarrollo de dos aplicaciones fundamentales, el Servicio local para el manejo de dispositivos y el Sistema para el control centralizado de dispositivos. De ellas se obtuvo la implementación de cada una de sus clases y servicios, así como las

interfaces correspondientes que mostraron las necesidades del cliente. Además se desarrolló un componente denominado Framework DGMJS que permitió la comunicación de los componentes implementados en dichas aplicaciones.

En los epígrafes que se muestran a continuación se exponen cada uno de los diagramas de componentes que respaldan la propuesta de solución y un ejemplo de las interfaces obtenidas en la implementación de la misma.

4.4.1 Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo como son las clases en el modelo de diseño. El diagrama de componentes describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Especifica además cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (45)

Servicio local para el manejo de dispositivos

La Fig 4.2 representa el diagrama de componentes del Servicio local para el manejo de dispositivos. En él se muestra el componente ServiceCore que usa dos componentes fundamentales, el ServiceBase y el SecurityLayer. El ServiceBase es el que contiene las clases que permiten crear los servicios a partir de la información del cliente y el controlador del servicio. Por otra parte el SecurityLayer contiene las clases y operaciones necesarias para validar cada una de las peticiones hechas a los servicios permitiendo o denegando el acceso según la configuración establecida.

Se muestra además el componente CameraService.Client usado por el ServiceBase que representa el cliente de un servicio, en este caso el servicio de las cámaras. El mismo contiene las clases y define los métodos que contendrá el servicio. El ServiceBase usa además el CameraService.Controller que es el componente encargado de definir las operaciones que el resto de los controladores del servicio deberán implementar para llevar a cabo la interacción con el dispositivo como es el caso del componente CameraService.Controller.WebCamController que implementa la comunicación con una Webcam.

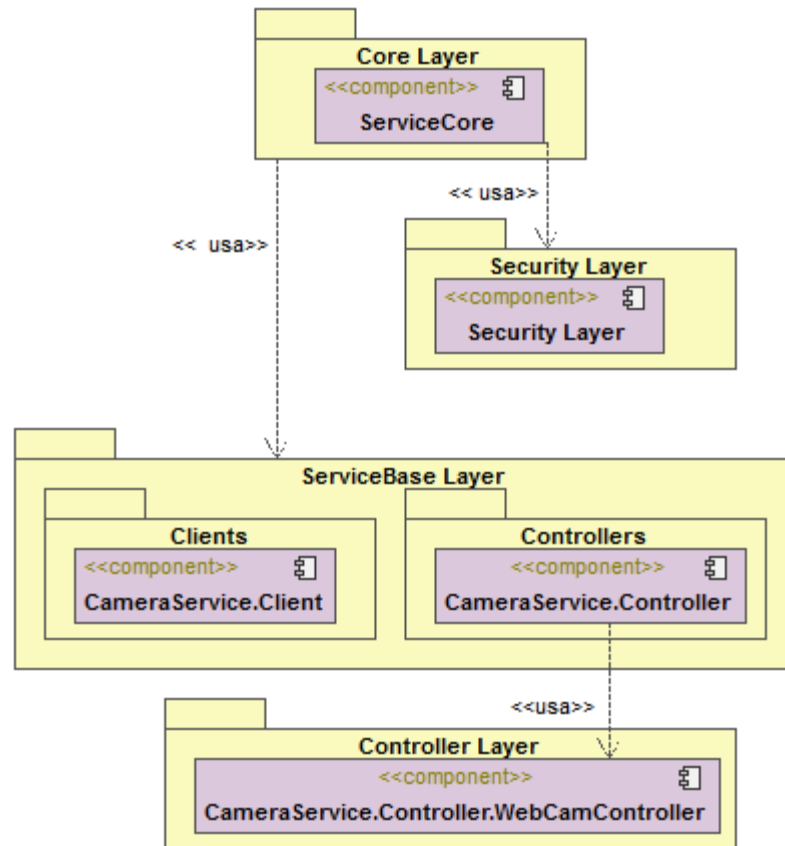


Fig 4.2 Diagrama de componentes del Servicio local para el manejo de dispositivos. Fuente: Elaboración propia.

Sistema para el control centralizado de dispositivos

Los componentes del Sistema para el control centralizado de dispositivos fueron desarrollados cumpliendo con la arquitectura propuesta por el proyecto Identificación, Inmigración y Extranjería de la República de Cuba. La Fig 4.3 muestra el diagrama de componentes de este sistema. En la capa de presentación se encuentra el componente DGMAApp que representa la Aplicación Web que incluye las clases: JavaScript, CCS y las interfaces de usuario, estas últimas separadas por módulos. Se encuentra además la referencia al componente UIToolBox, librería que permite la inclusión de componentes en las páginas que conforman la vista con el objetivo de agilizar el desarrollo de la aplicación y garantizar una interfaz de usuario uniforme. En este ensamblado se encuentran también la implementación de los servicios web para la interacción con JavaScript que son utilizados en la capa de presentación, esta capa se comunica con la

capa de Procesos y Servicios; en esta última capa se encuentran los componentes Common.Services.Client, CommonActivities, CommonTools, CommonEntities y el Bison Framework, encargado de la gestión y administración de los procesos y servicios de intercambio.

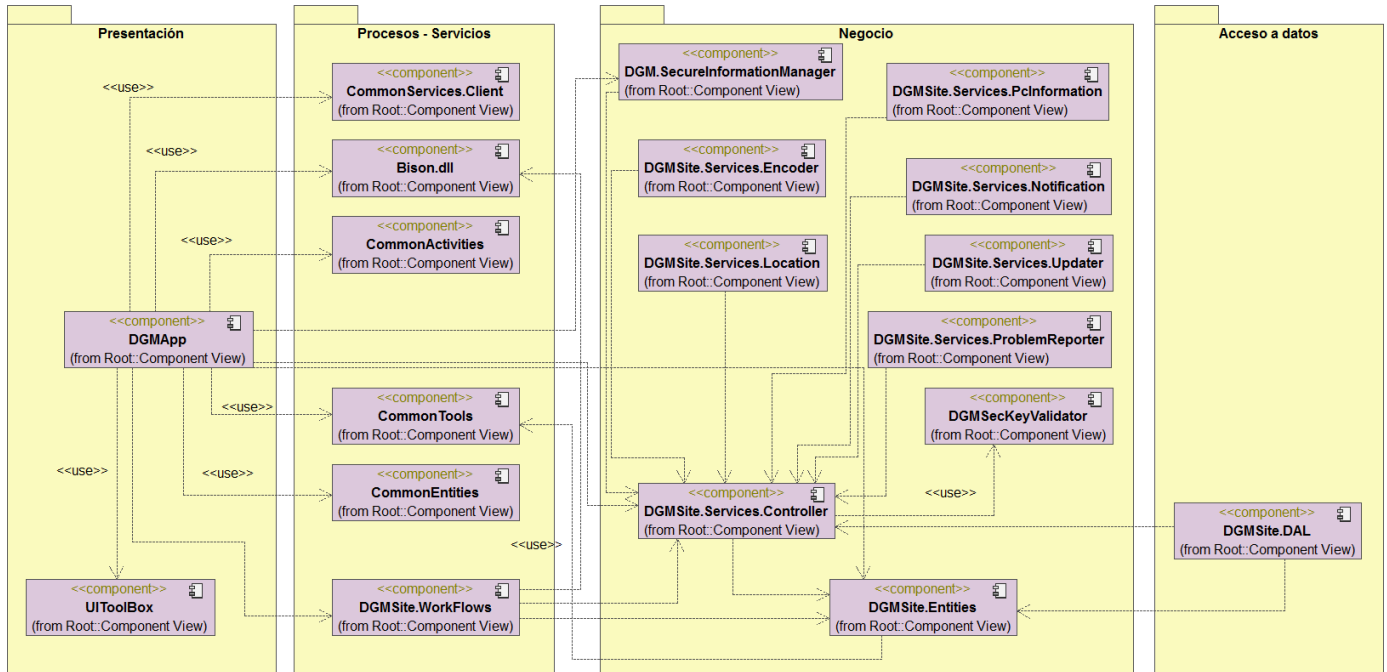


Fig 4.3 Diagrama de componentes del Sistema para el control centralizado de los dispositivos. Fuente: Elaboración propia.

El componente DGMSite.Workflows contiene el diseño y la implementación de los *workflows* que permiten la modelación y ejecución de los procesos de gestión de nomencladores y activación de estaciones de trabajo del Sistema para el control centralizado de dispositivos. Aquí también se encuentran las referencias a los servicios de WCF y los servicios del Runtime que permiten la interacción e intercambio de datos con los procesos.

Los componentes encargados de exponer los servicios y las responsabilidades que posibilitan el funcionamiento del Servicio local para el manejo de dispositivos son los siguientes: DGMSite.Services.Location, DGMSite.Services.Notification, DGMSite.Services.PcInformation, DGMSite.Services.ProblemReporter y DGMSite.Services.Updater.

El ensamblado DGMSite.Entities encapsula las clases entidades que funcionan como contenedoras de la información que fluye a través de las capas de Presentación, Procesos y Servicios y Negocio, tiene como principal objetivo describir el modelo de objetos. Para el acceso a datos se implementaron una serie de servicios que se encargaran de operar con la información persistente y enviarla a las clases entidades, los componentes responsables son DGMSite.Services, DGMSite.Connector.Interfaces y DGMSite.DAL; el primero que se menciona es el que expone los servicios de gestión de nomencladores, la activación de estaciones de trabajo, el reporte de errores entre otros; el DGMSite.Connector.Interfaces declara las interfaces que definen los comportamientos del acceso a datos que luego son implementadas en el componente DGMSite.DAL. El componente DGMSecurityValidator es usado en el proceso de autenticación para consumir algunos servicios, se encarga de validar la llave que se genera a partir del identificador de una estación de trabajo. El componente DGMSecureInformationManager es utilizado para subir la información del hardware de las estaciones de trabajo de forma segura al servidor de aplicaciones.

Framework DGMJS

La Fig 4.4 muestra el diagrama de componentes para el Framework DGMJS. El diagrama solo presenta un componente denominado DGMJS el cual contiene las dos clases que posibilitan la comunicación del Sistema para el control centralizado de dispositivos con el Servicio local para el manejo de dispositivos.

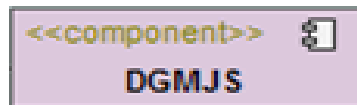


Fig 4.4 Diagrama de componentes del Framework DGMJS. Fuente: Elaboración propia.

4.4.2 Pautas del diseño del Sistema para el control centralizado de los dispositivos

La apariencia de un producto es un elemento importante a tener en cuenta cuando se desarrolla una aplicación. Los elementos visuales de la aplicación deben transmitir la identidad del cliente y la del producto además de mantener una estructura uniforme que permita al usuario interactuar de forma intuitiva y sin dificultad con la aplicación.

En este epígrafe se describen las principales pautas de diseño definidas para los sistemas del Proyecto Identificación, Inmigración y Extranjería de la República de Cuba, entre los que se encuentra el Sistema para el control centralizado de dispositivos.

Diagramación general

En la Fig 4.5 se muestra la estructura que debe tener el sistema con sus principales áreas para mostrar los distintos tipos de contenido. A continuación se describen cada una de ellas teniendo en cuenta su función y las características en cuanto a diseño que deben poseer:

1. **Cabezal:** Muestra la identidad visual del sistema y el nombre oficial de la aplicación. **Dimensiones:** Largo: 1005 px; Altura: 63 px.
2. **Área de notificación e íconos complementarios:** Zona donde se ubica el nombre completo del usuario autenticado en el sistema, así como los botones para mostrar la ayuda y cerrar la aplicación. **Dimensiones:** Largo: 1005 px; Altura: 30 px.
3. **Menú lateral:** Se ubican las funcionalidades de la aplicación agrupadas por módulos. **Dimensiones de los botones del menú:** Largo: 205 px; Altura: 30 px.
4. **Área de trabajo:** Se muestra el contenido de cada página de la aplicación. **Dimensiones:** Largo: 800 px; Altura: automática.
5. **Pie de página:** Delimitador del final de la página. **Dimensiones:** Largo: 1005 px; Altura: 30 px.

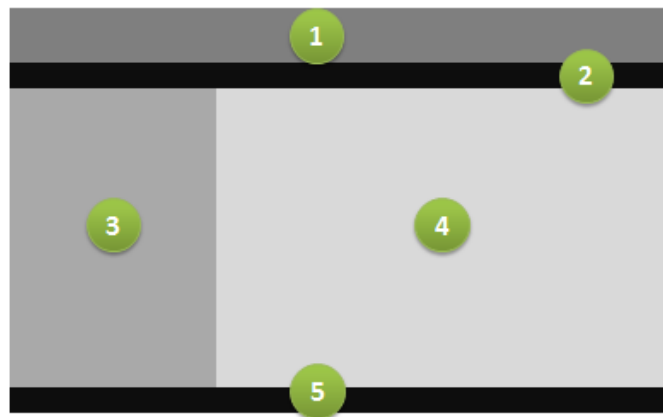


Fig 4.5 Diagramación general. Fuente: Manual de pautas para el Sistema Único de Identificación Nacional (SUIN).

Tipografía

Se utilizará la tipografía Tahoma en todas sus variantes, la utilización de cada una de ellas se pautan en cada una de las secciones en particular. En las entradas de datos y dentro de los cuadros de texto se usará la Tipografía Arial en su variante normal. (56)

El resto de las especificaciones se encuentran en el documento Manual de pautas para el Sistema Único de Identificación Nacional del Proyecto Identificación, Inmigración y Extranjería de la República de Cuba.

A continuación se muestra la Fig 4.6 y la Fig 4.7 con las interfaces obtenidas en la nueva versión funcional del Sistema DGM.

Interfaz del Servicio Local para el manejo de dispositivos

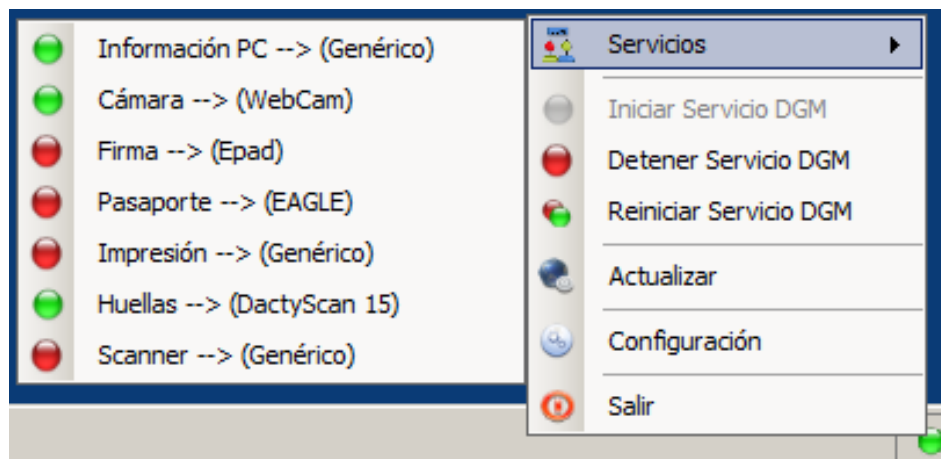


Fig 4.6 Interfaz del Servicio Local para el manejo de dispositivos. Fuente: Elaboración propia.

Interfaz del Sistema para el control centralizado de dispositivos



The screenshot shows the 'Activar estación de trabajo' (Activate workstation) form in the DGM Device Grid Manager. The interface includes a sidebar with navigation options: 'Gestionar', 'Estaciones de trabajo', and 'Reportes'. The main form is titled 'Activar estación de trabajo' and contains three sections: 'Activación' with a 'Serial de Activación' field; 'Dispositivo' with 'Nombre' and 'Tipo de dispositivo' fields; and 'Estación de trabajo' with a 'Ubicación física' field containing 'Cuba'. At the bottom, there are three buttons: 'Especificaciones técnicas', 'Activar', and 'Cancelar'.

Fig 4.7 Interfaz del Sistema para el control centralizado de dispositivos. Fuente: Elaboración propia.

4.5 Pruebas

Las pruebas de software según la IEEE Estándar Glossary of Software Engineering Terminology, se definen como: “El proceso de operar un sistema o un componente bajo condiciones especificadas, observando o registrando los resultados, y de hacer una evaluación de un cierto aspecto del sistema o del componente.” (57). El objetivo de las pruebas es la detección de defectos en el software, las pruebas muestran la presencia de errores pero nunca la ausencia de ellos.

Existen diferentes tipos de pruebas, cada una de las cuales está enfocada según los resultados que se pretende obtener con su realización, ejemplo de ellas son las pruebas de integración, unitarias, de sistema, aceptación entre otras. La propuesta de solución está evaluada mediante las pruebas unitarias y de sistema, selección realizada por el equipo de calidad del proyecto Identificación, Inmigración y Extranjería de la República de Cuba.

4.5.1 Pruebas unitarias

Las pruebas unitarias sirven para utilizar otro código fuente llamando directamente a los métodos de una clase, pasando los parámetros apropiados. Su código y sus atributos residen en un archivo de código fuente en un proyecto de prueba del mismo lenguaje. La prueba se orienta al código específico que se desea probar. (58)

Con el objetivo de validar que las clases y componentes que conforman el sistema cumplen las especificaciones y tienen el comportamiento esperado, se emplearon pruebas unitarias para validar la propuesta de solución. En la Fig 4.8 se muestra la prueba que se realizó a la funcionalidad “Adicionar agrupación lógica” del servicio EncoderService y el resultado arrojado por la misma.

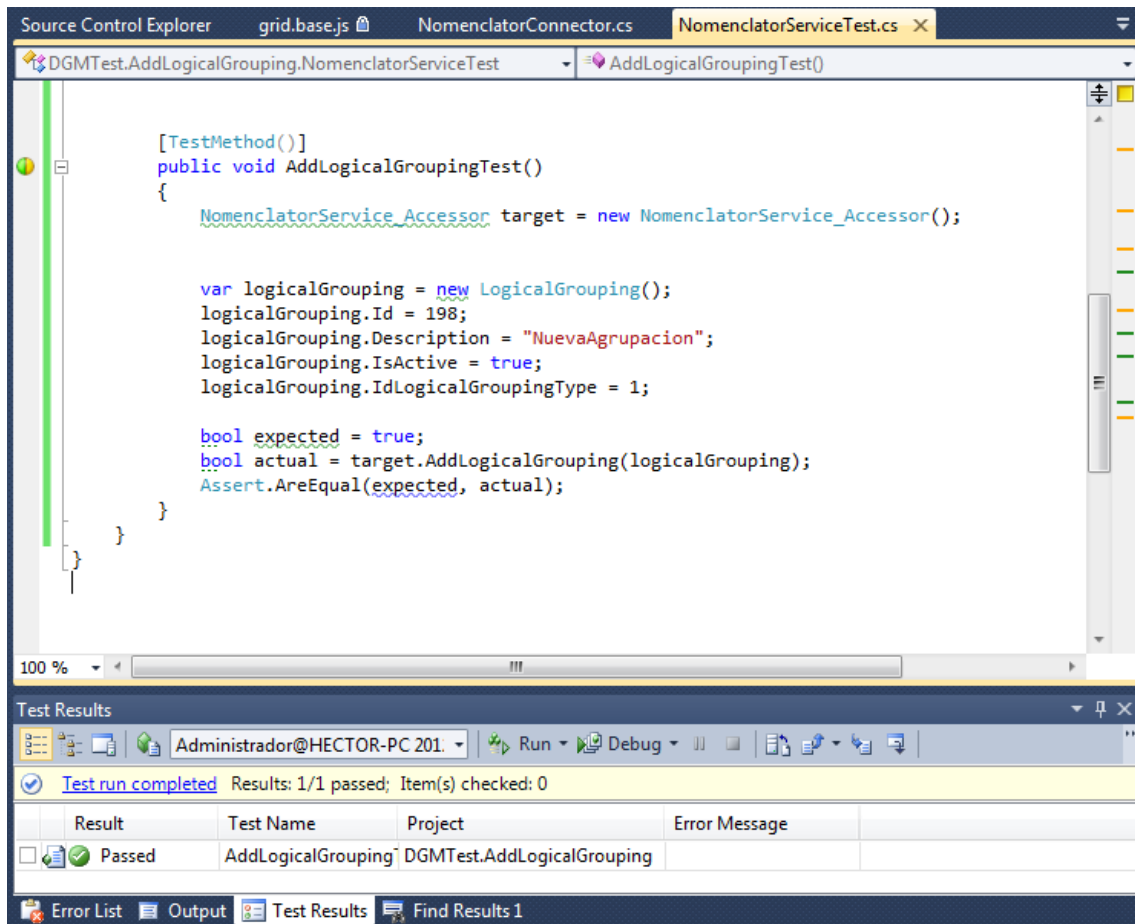


Fig 4.8 Resultado de la prueba unitaria al servicio EncoderService. Fuente: Elaboración propia.

4.5.2 Pruebas de sistema

En las pruebas de sistema los procesos soportados por la aplicación se cumplen completamente, es decir, los procesos fluyen desde su inicio hasta el final. (57) Dentro de las pruebas de sistema se encuentran las pruebas de caja negra, que se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (59)

4.5.3 Diseño de casos de prueba

Un caso de prueba determina una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. Lo que se prueba puede venir dado por un requisito o colección de requisitos del sistema cuya implementación justifica una prueba que es posible realizar. (45) A continuación en la Tabla 4.1, se muestra el diseño de caso de prueba de la funcionalidad “Gestionar estaciones de trabajo” utilizando la técnica de caja negra, el resto de los casos de pruebas se encuentran en el Anexo 8.

Tabla 4.1 Caso de prueba para la funcionalidad Gestionar estaciones de trabajo. Fuente: Elaboración propia.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
1. Gestionar estaciones de trabajo.	1.1 Activar estación de trabajo correctamente.	El sistema activa la estación de trabajo.	<ol style="list-style-type: none"> Estaciones de trabajo Activar estación Adicionar Aceptar
	1.2 Activar estación de trabajo con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> Estaciones de trabajo Activar estación Adicionar Aceptar
	1.3 Activar estación de trabajo con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> Estaciones de trabajo Activar estación Adicionar Aceptar
	1.4 Mostrar las especificaciones técnicas de la estación de	El sistema muestra un mensaje con las especificaciones técnicas de la estación	<ol style="list-style-type: none"> Estaciones de trabajo Activar estación Especificaciones técnicas Aceptar

	trabajo	de trabajo.	
	1.5 Activar estación de trabajo.	El sistema redirecciona a la página principal.	<ol style="list-style-type: none"> Estaciones de trabajo Activar estación Cancelar
	1.6 Desactivar estación correctamente.	El sistema desactiva la estación correctamente.	<ol style="list-style-type: none"> Estaciones de trabajo Desactivar estación Desactivar
	1.7 Desactivar estación sin haber seleccionado una oficina.	El sistema muestra un mensaje indicando que debe seleccionar una oficina.	<ol style="list-style-type: none"> Estaciones de trabajo Desactivar estación Desactivar Aceptar
	1.8 Desactivar estación sin haber seleccionado una estación de trabajo.	El sistema muestra un mensaje indicando que debe seleccionar una estación de trabajo.	<ol style="list-style-type: none"> Estaciones de trabajo Desactivar estación Desactivar Aceptar

Tabla 4.2 Resultados de las pruebas. Fuente: Elaboración propia.

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
1.1	Clases Válidas	Serial de Activación: "8796941e7ee2483a8df340f04b2c455f" Nombre: "SUIN-PC1" Tipo de dispositivo: "Hp Pavilion dm4 (No hay descripción)" Ubicación física: "Colón"	Se activa una estación de trabajo.	Se activa una estación de trabajo.	
1.3	Clases Inválidas	Serial de Activación: "8796941e7ee2483a8df340f04b2c455f" Nombre: "SUIN-PC1" Tipo de dispositivo: "" Ubicación física: "Colón"	Se muestra un mensaje que indica que el campo está vacío.	Se muestra un mensaje que indica que el campo está vacío.	

1.6	Clases Válidas	Agrupación lógica: "Colón" Estación de trabajo: "Hector-PC"	Se desactiva una estación de trabajo.	Se desactiva una estación de trabajo.	
1.7	Clases Inválidas	Agrupación lógica: "Colón" Estación de trabajo: ""	Se muestra un mensaje indicando que debe seleccionar una estación de trabajo.	Se muestra un mensaje indicando que debe seleccionar una estación de trabajo.	

Para lograr que todas las no conformidades obtenidas en las pruebas realizadas se mitiguen es necesario efectuar varias iteraciones de pruebas. En cada iteración se debe entregar al desarrollador un resumen de todas las no conformidades para ser erradicadas por el mismo. En la Tabla 4.3 se muestran los resultados relacionados con la primera iteración de pruebas realizadas. Los resultados de las pruebas de la segunda iteración se encuentran en la Tabla 69 del Anexo 8.

Tabla 4.3 Resultado de las pruebas de la primera iteración. Fuente: Elaboración propia.

No	No conformidad	Etapas de detección del error	Importancia
1	Al adicionar una agrupación lógica el campo "Nombre" no permite introducir números.	Pruebas	Media
2	Al modificar una agrupación lógica el campo "Nombre" no permite introducir números.	Pruebas	Media
3	Al modificar una agrupación física el sistema no valida que los campos no estén vacíos.	Pruebas	Media
4	Al adicionar un dispositivo el campo "Descripción" no permite introducir números.	Pruebas	Media
5	Al adicionar un dispositivo el sistema no valida que el campo "Tipo de dispositivo" no esté vacío.	Pruebas	Media
6	Al adicionar un tipo de dispositivo el sistema no valida que los campos "Categoría" y "Marca" no estén vacíos.	Pruebas	Media
7	Al modificar un tipo de dispositivo el sistema no valida que los campos "Categoría" y "Marca" no estén vacíos.	Pruebas	Media
8	Al modificar un tipo de dispositivo el sistema no muestra un mensaje indicando el resultado de la acción.	Pruebas	Media

4.5.4 Resultados de las pruebas

Durante la ejecución de tres iteraciones de pruebas se detectaron un conjunto de diez no conformidades, las mismas fueron mitigadas. (Ver Fig 4.8) En la primera iteración se realizó una versión funcional del software que fue probado en el entorno de desarrollo del Proyecto Identificación, Inmigración y Extranjería de la República de Cuba con el objetivo de probar el sistema y corregir cualquier deficiencia antes de ser desplegado en un ambiente real.

En dicha iteración se encontraron un total de 8 no conformidades, que en todos los casos fueron problemas de validación de datos relacionados con la entrada de caracteres no válidos en campos de texto de las interfaces del Sistema para el control centralizado de dispositivos. Luego de solucionar las dificultades detectadas se realizó una segunda iteración generando una versión funcional del producto que fue probado junto con el cliente en la Dirección de Identificación y Registro (DIR) perteneciente al MININT y con CALISOFT, entidad perteneciente a la UCI que brinda servicios para la realización de revisiones y pruebas de calidad a productos de software.

En esa iteración se comprobó que no quedaban errores de la primera, pero se encontraron 2 nuevas no conformidades relacionadas con el componente de verificación de rostro y los controladores de los dispositivos de captación de huellas que fueron solucionadas. En la última iteración no se identificaron no conformidades; la misma se realizó de manera conjunta entre el cliente, CALISOFT y el equipo de desarrollo. Además de probarse en el ambiente de desarrollo del proyecto y la DIR, el sistema fue desplegado en las oficinas de Carné de Identidad y Registro de la Población de todas las provincias desde Pinar del Río hasta Sancti Spíritus.

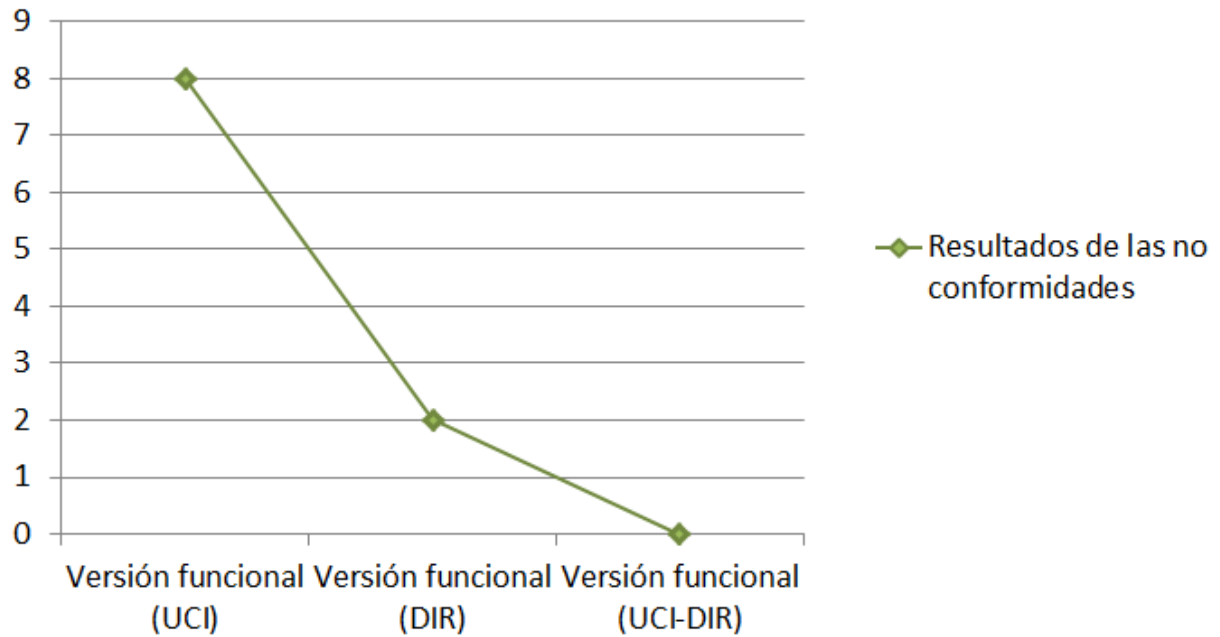


Fig 4.8 Resultado de las no conformidades. Fuente: Elaboración propia.

4.6 Análisis comparativo entre la versión 1.0 y 2.0 del DGM

En la siguiente tabla se hace un análisis de las principales características de las versiones 1.0 y 2.0 del DGM. La Tabla 4.4 muestra la comparación entre las distintas versiones del DGM.

Tabla 4.4 Análisis entre la versión 1.0 del DGM y la versión 2.0. Fuente: Elaboración propia.

Aspecto	Elemento	DGM v1.0	DGM v2.0
Sistema de administración centralizado de dispositivos		✗	✓
Interacción con dispositivos	Lectores de huellas dactilares	✓	✓
	Cámaras	✗	✓
	Lectores de pasaporte	✗	✓
	Escáneres	✗	✓
	Impresoras	✗	✓

	Lectores de firma	✓	✓
	Controlador compartido ¹⁹	✓	✓
Configuración	Configuración de la aplicación	✗	✓
	Actualización manual	✓	✓
	Actualización automática	✗	✓
Comunicación	Independiente del navegador	✗	✓
Seguridad	Validación de peticiones	✓	✓
	Sistema de trazas	✗	✓
	Autenticación segura a partir del hardware	✗	✓
Servicios	Información del hardware del ordenador	✓	✓
	Notificación de problemas	✗	✓
Reportes	Tiempo de uso de los dispositivos	✗	✓
	Estaciones de trabajo activas por agrupación lógica	✗	✓
Grado de despliegue	Equipo desarrollo	✓	✓
	Cliente	✗	✓
Arquitectura	x86	✓	✓
	x64	✗	✓
Sistema operativo	Windows	✓	✓
	Linux	✗	✗

En la tabla anterior se aprecian las mejoras realizadas al DGM en la versión 2.0, respecto a la primera versión, de manera que se puede considerar que se ha obtenido un sistema notablemente mejorado, y que se ha dado solución al problema de la investigación, teniendo este en general los siguientes elementos:

¹⁹ Controlador Compartido: Funcionalidad que permite usar un dispositivo que se encuentra en una estación de trabajo remota.

1. Se obtuvo un producto genérico capaz de controlar e interactuar con múltiples dispositivos de hardware conectados a estaciones de trabajo; obteniendo resultados de capturas hechas con cámaras, lectores de huellas, escáneres y otros tipos de dispositivos además de realizar impresiones; garantizando la realización de todos los procesos de captación biométrica que se llevan a cabo en las oficinas de identificación.
2. La comunicación entre el SUIN y el Servicio Local para el manejo de dispositivos se establece de manera independiente del navegador a través de un modelo de comunicación orientado a servicios.
3. Permite el control de todos los recursos que conforman la red, realizando un monitoreo continuo a los servicios y a los dispositivos, entre los que se encuentran computadoras u otros periféricos y la obtención de reportes con el fin de supervisar las operaciones que se realizan con dichos dispositivos.
4. Se obtuvo un componente para complementar el sistema de seguridad dentro de la red de dispositivos y la comunicación que se establece entre el SUIN y el Servicio Local para el manejo de dispositivos.
5. Permite generar reportes de tiempo de uso de los dispositivos, lo que posibilita planificar y realizar mantenimientos a los dispositivos conectados a la red.
6. El Sistema para la administración centralizada de los dispositivos almacena datos específicos del hardware de las estaciones de trabajo conectadas a la red, posibilitando detectar irregularidades que puedan ser realizadas sobre los componentes de las mismas, lo que incrementa la seguridad en la red.
7. Cuenta con un mecanismo de actualización automática que reduce considerablemente el tiempo de respuesta a dificultades que puedan surgir o la inclusión de nuevas funcionalidades o dispositivos.

Como se ha mencionado anteriormente el sistema se encuentra desplegado en el entorno de desarrollo del SUIN y en 98 oficinas de carné de identidad del país, teniendo una buena aceptación por parte del cliente (Ver Anexo 1). Es importante destacar además que han sido capacitados los funcionarios de la Dirección de Identificación y Registros en 5 iteraciones de cursos de capacitación.

Las aplicaciones obtenidas pueden ser integradas de manera sencilla, con una baja curva de aprendizaje por parte de los desarrolladores en cualquier sistema que necesite la interacción con dispositivos de hardware.

4.7 Conclusiones parciales

En la etapa de implementación y pruebas del sistema, después de realizar las actividades y los artefactos correspondientes se puede concluir:

- En la vista de despliegue en un ambiente real del sistema, se indican los elementos fundamentales y los protocolos de comunicación que se establecen entre ellos.
- Durante la etapa final del desarrollo del sistema se realizaron las pruebas a los componentes de software con el objetivo de validar el producto; como resultado de la última iteración no se detectaron no conformidades lo que permitió la entrega de un producto de calidad al cliente.
- Estableciendo una comparación entre la versión 1.0 y 2.0 del DGM se pudieron comprobar las mejoras entre una versión y otra, la versión 2.0 cuenta con 15 funcionalidades más, respecto a la primera versión y se realizaron mejoras a 9 funcionalidades que existían anteriormente.

CONCLUSIONES

La investigación realizada permite a los autores concluir que:

- El análisis de las tecnologías y los mecanismos para acceder a recursos locales de una estación de trabajo desde la web confirmó que es posible el manejo de los dispositivos, pero en general presentan dependencias del navegador o de otras aplicaciones.
- El estudio de sistemas homólogos a nivel mundial demostró que estas herramientas, aunque poseen características requeridas por el cliente son propietarias o no cumplen con las especificaciones del proyecto, en el caso del DGM 1.0 se determinó tomarlo como base para el desarrollo, aunque tiene deficiencias que impiden que sea una solución integral para el manejo de dispositivos al no ser independiente del navegador ni tener una seguridad adecuada.
- La obtención de un Framework JavaScript para el acceso a servicios fuera del marco del dominio de la aplicación web que esté en ejecución, garantiza la independencia del navegador logrando un mecanismo de comunicación desde las aplicaciones web del lado del cliente totalmente independiente y estandarizado.
- Las nuevas aplicaciones obtenidas enriquecen notablemente las funcionalidades respecto a la versión anterior del sistema a partir de la posibilidad realizar configuraciones y actualizaciones, así como la obtención de reportes para el control del uso de los dispositivos, y pueden ser integradas a sistemas que requieran la interacción con dispositivos de hardware de manera rápida y eficaz.
- La realización de tres iteraciones de pruebas durante el proceso de desarrollo, permitió obtener una versión estable final del producto, que ha sido integrada al Sistema Único de Identificación Nacional y desplegada en 98 oficinas en el territorio nacional.

RECOMENDACIONES

Los autores proponen las siguientes recomendaciones:

- Integrar el DGM v 2.0 y con el Sistema de Contabilidad para las TICs del MININT, en aras de mantener un mayor control de los dispositivos que conforman la red.
- Agregar funcionalidades al Servicio local para el manejo de dispositivos que permitan brindar soporte a las estaciones de trabajo de forma remota y reportar posibles problemas desde el cliente.
- Valorar la migración de la aplicación a otras plataformas para su posible uso en sistemas operativos libres.

REFERENCIAS BIBLIOGRÁFICAS

1. Bundesdruckerei. *Bundesdruckerei*. [Online] marzo 1, 2010. <http://www.bundesdruckerei.de/>.
2. SUN. *SUN*. [Online] Marzo 3, 2010. <http://java.sun.com/javacard/>.
3. **About.com**. WebTrends. [Online] The New York Times. [Cited: enero 10, 2012.] http://webtrends.about.com/od/webapplications/a/web_application.htm.
4. **Mora, Sergio Luján**. *Programación de aplicaciones web: historia, principios básicos y clientes web*. s.l. : Editorial Club Universitario, 2002.
5. —. *Programación en Internet: Clientes Web*. s.l. : Editorial Club Universitario, 2001.
6. **Berzal, Fernando, Cortijo, Francisco José and Cubero, Juan Carlos**. *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. 2005. ISBN: 84-609-4245-7.
7. **World Wide Web Consortium**. World Wide Web Consortium. *World Wide Web Consortium*. [Online] W3C, mayo 9, 2010. [Cited: diciembre 2011, 16.] <http://www.w3.org>.
8. **Kamthan, Pankaj**. Internet Related Technologies. *Internet Related Technologies*. [Online] 3 7, 1999. [Cited: enero 10, 2011.] <http://www.irt.org/articles/js151/>.
9. **Mozilla Foundation**. Mozilla Addons. *Mozilla Addons*. [Online] Mozilla Foundation, marzo 16, 2010. [Cited: enero 2012, 20.] <https://addons.mozilla.org/>.
10. **Orgera, Scott**. About.com. [Online] The New York Times. [Cited: noviembre 15, 2011.] <http://browsers.about.com/od/addonsplugi2/u/browseraddons.htm>.
11. **Axiompass, LLC**. axiompass. *axiompass*. [Online] Axiompass, LLC, 2010. [Cited: Diciembre 12, 2011.] <http://www.axiompass.com/Solutions/Access.aspx>.
12. **Oro, Ing. Jeandy Bryan Pineda**. *Sistema para la interaccion y control centralizado de dispositivos en aplicaciones web*. Centro de Identificación y Seguridad Digital. Departamento de Soluciones Integrales., Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n., 2009. Tesis de Grado.

13. **MorphoAccess Chile.** MorphoAccess (TM). *MorphoAccess (TM)*. [Online] MorphoAccess Chile, 2011. [Cited: Diciembre 12, 2011.] <http://www.morphoaccess.cl/SAGEMMorphoAccess/tabid/179/language/en-US/Default.aspx>.
14. **Romsey Associates.** Eynetwatch. [Online] 2011. [Cited: abril 10, 2012.] <http://www.eyenetwatch.com/Products/165-sagem-mems-enterprise-fingerprint-software.aspx>.
15. *SISTEMA PARA LA INTERACCIÓN Y CONTROL CENTRALIZADO DE DISPOSITIVOS EN APLICACIONES WEB.* **García, Msc. Jorge Landrian, et al., et al.** La Habana : s.n., 2011, Vol. X Seminario Iberoamericano de Seguridad en las Tecnologías.
16. **Microsoft.** MSDN Library . *MSDN Library*. [Online] <http://www.msdn.microsoft.com/en-us..>
17. **Guckenheimer , Sam and Pérez, Juan J.** *Software Engineering with Microsoft Visual Studio Team System (Microsoft .NET Development Series)*. s.l. : Addison-Wesley Professional, 2006. ISBN: 0321278720.
18. **Object Management Group.** *Business Process Modeling Notation, V1.1*. 2008. OMG Available Specification. formal/2008-01-17.
19. **Booch, Grady , Rumbaugh, James and Jacobson, Ivar .** *El lenguaje unificado de modelado*. s.l. : Addison Wesley, 2006.
20. **Altova.** Altova. [Online] 2011. [Cited: diciembre 2011, 6.] <http://www.altova.com/umodel.html>.
21. Danysoft. [En línea] [Citado el: 20 de Enero de 2011.] <http://www.codegear-shop.com/Embarcadero-ER/Studio/es>.
22. MSDN Local. [En línea] [Citado el: 19 de Enero de 2011.] 17. ms-help://MS.MSDNQTR.v90.es/wcf_con/html/fd327ade-0260-4c40-adbe-b74645ba3277.htm.
23. Danysoft. [En línea] [Citado el: 19 de Enero de 2011.] <http://www.danysoft.com/bol/visualstudio2008.htm>.

24. **Microsoft.** MSDN. [En línea] [Citado el: 20 de Enero de 2011.] <http://msdn.microsoft.com/es-es/library/ms731082.aspx>.
25. **Corporation, Microsoft.** MSDN. [Online] [Cited: noviembre 5, 2010.] <http://msdn.microsoft.com/>.
26. **Olivarez, José Luis.** *Windows Workflow Foundation: Manejo de procesos en los nuevos productos de Microsoft.* 2005.
27. **McPeak, Jeremy, Zakas, Nicholas C and Fawcett, Joe.** *Professional Ajax.* Indianapolis : Wiley Publishing, Inc, 2006. ISBN: 978-0-471-77778-6.
28. **Microsoft.** Ayuda Online Local. [En línea] [Citado el: 17 de Enero de 2011.] ms-help://MS.MSDNQTR.v90.es/wd_adonet/html/5b96ed06-9759-4966-a797-a1d5f6ee50ca.htm.
29. —. MSDN Local. [En línea] [Citado el: 17 de Enero de 2011.] ms-help://MS.MSDNQTR.v90.es/wd_adonet/html/c57089e9-1b88-4c6e-8210-8cbd8f7f572a.htm.
30. **Asktheoracle.** Asktheoracle. [Online] 2011. [Cited: diciembre 6, 2011.] <http://www.asktheoracle.net/oracle-11g-new-features.html>.
31. Oracle. [Online] 2011. [Cited: diciembre 7, 2011.] <http://www.oracle.com/es/products/database/index.html>.
32. **McDonald, Matew and Szpuzta, Mario.** *Pro ASP.NET 3.5 in C# 2008.* Berkley : APRESS, 2008. ISBN: 978-1-4302-1568-4.
33. **Microsoft Corporation.** MSDN. *Visual Studio 2010, Introduction to LINQ.* [Online] 2010. [Cited: Julio 21, 2010.] <http://msdn.microsoft.com/es-es/library/bb397926.aspx>.
34. **Granados, Jimmy.** NeuronalTraining.NET. The LINQ Project. *.NET Language Integrated Query.* [Online] 2006. [Cited: Septiembre 20, 2011.] <http://www.neuronaltraining.net>.
35. **Flanagan, David.** *JavaScript: The Definitive Guide, Fifth Edition.* Fifth . s.l. : O'Reilly Media Inc, 2006. ISBN: 978-0-596-10199-2.

36. **Barcia, Diego.** Maestros de la Web. *¿Qué es CSS?* [Online] [Cited: Julio 21, 2010.] <http://www.maestrosdelweb.com/editorial/introcssl/>.
37. **Castro, Elizabeth.** *HTML con XHTML y CSS.* Madrid : Anaya Multimedia, 2007. p. 656. ISBN 9788441521834.
38. **Dykes, Lucinda and Tittel, Ed.** *XML for Dummies 4th Edition.* Indianapolis : Wiley Publishing, Inc, 2005. ISBN: 978-0-7645-8845-7.
39. **Alvarez, Miguel Angel.** Desarrollo Web. *Que es XML.* [Online] Junio 13, 2001. [Cited: Julio 21, 2010.] <http://www.desarrolloweb.com/articulos/449.php>.
40. **ORACLE.** ORACLE. [Online] [Cited: diciembre 8, 2011.] http://docs.oracle.com/cd/B19306_01/appdev.102/b14261/overview.htm#i9141.
41. **Dahan, Udi.** MSDN Magazine. *MSDN.* [Online] marzo 9, 2011. <http://msdn.microsoft.com/es-es/magazine/ee236415.aspx>.
42. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000. ISBN: 84-7829036-2.
43. **www.MiTecnologico.com.** MiTecnologico. *MiTecnologico.* [Online] 2012. [Cited: 5 14, 2012.] <http://www.MiTecnologico.com>.
44. **Vega, Erick de la.** *Estándares de Codificación para el Proyecto Identificación, Inmigración y Extranjería de la República de Cuba.* Departamento de Soluciones Integrales, Universidad de las Ciencias Informáticas. La Habana : s.n., 2009.
45. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* La Habana : Felix Varela, 2004.
46. **Zambrano, Reynier Blanco.** *Documento de Arquitectura de Software. Proyecto Identificación, Inmigración y Extranjería.* La Habana : UCI, 2009.

47. **Jackson, Daniel and Devadas, Srinivas.** Open Course Ware. [Online] Massachusetts Institute of Technology, 2001. [Cited: abril 2, 2012.] <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>.
48. **Blé Jurado, Carlos.** *Diseño Agil con TDD*. Primera Edición. Madrid : Safe Creative, 2010. ISBN: 978-1-4452-6471-4.
49. **Shalloway, Alan and Trott, James.** *Designs Patterns Explained*. s.l. : Addison Wesley, 2004. ISBN-13: 978-0321247148.
50. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico. 5ta Edición*. s.l. : McGraw-hil Interamericana, 2001. p. 958 . ISBN: 8448132149.
51. **Líderes del grupo de Tecnología SOA de Accenture.** *Arquitectura Orientada a Servicios*. s.l. : Centro de Alto Rendimiento de Accenture (CAR), 2008.
52. Definicion. [Online] [Cited: abril 9, 2012.] <http://definicion.de/modelo-de-datos/>.
53. **Proyecto Identificación, Inmigración y Extranjería de la República de Cuba.** *Manual de Normas. Sistema Único de Identificación Nacional*. Proyecto Identificación, Inmigración y Extranjería de la República de Cuba. La Habana : s.n., 2009.
54. Escuela de Ingeniería de Sistemas y Computación. *Escuela de Ingeniería de Sistemas y Computación*. [Online] [Cited: abril 10, 2012.] <http://eisc.univalle.edu.co/cursos/web/material/750073/1/GSQA-IntroduccionPruebasSoftware.ppt>.
55. **Microsoft.** MSDN. [Online] Microsoft. [Cited: abril 9, 2012.] msdn.microsoft.com/es-es/library/ms182516%28v=vs.90%29.aspx.
56. *Estándar IEEE para Unidad de Testeo de Software*. **IEEE/ANSI**. New York : s.n., 1986.

ANEXOS**Anexo 1 Documento de aceptación del producto**

Acta de Aceptación



Proyecto: Modernización del sistema de Identificación, Inmigración y Extranjería de Cuba (Identidad Cuba).

Producto: Sistema Único de Identificación Nacional de la Población de la República de Cuba (SUIN).

Categoría de las pruebas: Revisión a la aplicación.

Fecha de conciliación: 22 de Mayo de 2012

Observaciones del proceso:

Durante el proceso de pruebas de aceptación se detectaron un conjunto de No Conformidades (NC) y Pedidos de Cambio (PC) que fueron registrados adecuadamente en los correspondientes informes diarios de NC y PC, con sus respectivas observaciones.

Teniendo en cuenta que fue validada por el cliente la corrección de la mayoría de las No Conformidades y gestionados los Pedidos de Cambios y valorando que las No Conformidades pendientes de solución no se consideran críticas por parte del cliente, para la etapa de despliegue del sistema, se ha tomado el acuerdo de Aceptar la aplicación Sistema Único de Identificación Nacional de la Población de la República de Cuba (SUIN) en su versión 1.0 y el documento "Manual de Usuario" en su versión 1.0, todos con fecha 21 de Mayo de 2012.

El conjunto de incidencias pendientes, de las que fueron detectadas durante el proceso de pruebas de aceptación han sido especificadas en los documentos Informe Final de No Conformidades (NC) e Informe Final de Pedidos de Cambio (PC), con sus respectivas observaciones.

Acta de Aceptación



Para que conste la Aceptación de los resultados de las pruebas y por tanto la Aceptación de los entregables especificados, dando fe del acuerdo, se extiende la presente **Acta** en tres (3) ejemplares, rubricados por los principales **Representantes de las Partes**.

Coronel Juan Manuel López Acevedo
Jefe de Dirección de Identificación y
Registros

Tte Coronel Magnolia Soto Bernal
Jefe de Departamento CIRP

MsC. Talyché Capote García
Representante Centro Nacional de
Calidad de Software
(CALISOFT)

MsC. Yudenia Ramírez Mastrapa
Jefa del proyecto



Anexo 2 Fases de MSF for CMMI

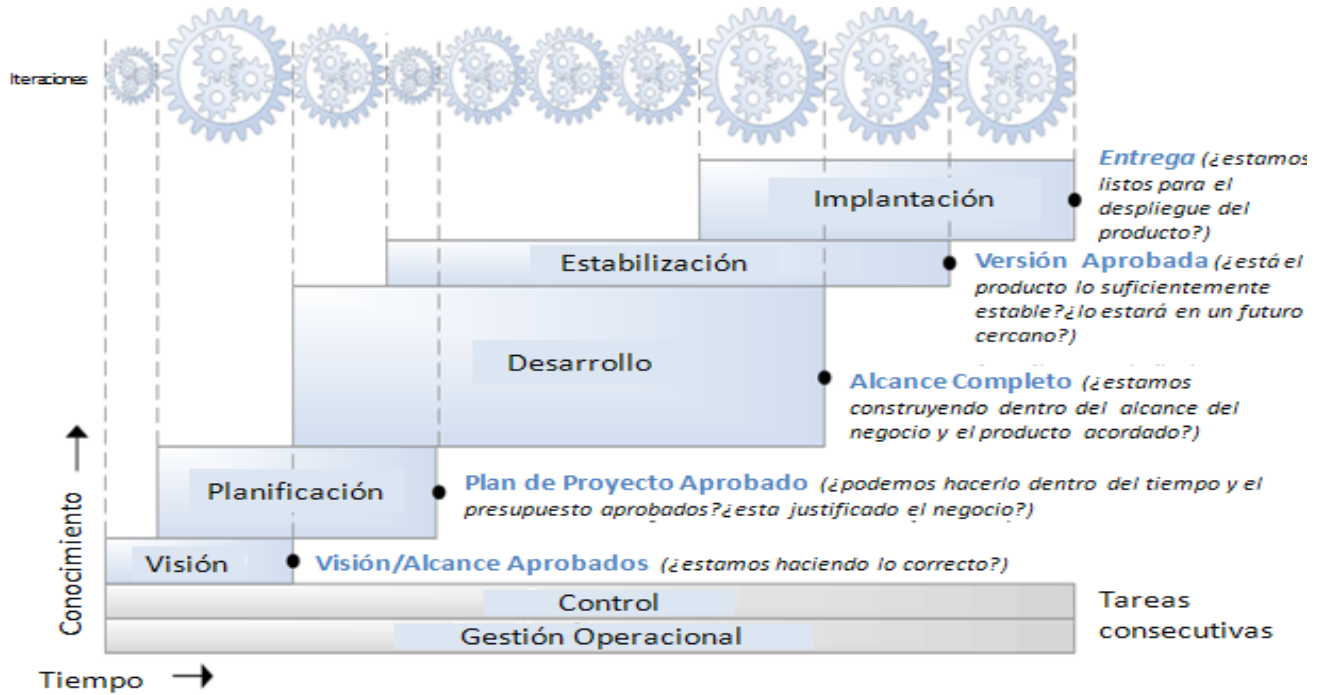


Figura 1 Fases de MSF for CMMI. Fuente: MSDN.

Anexo 3 Descripción de los requisitos funcionales

RF 2. Manejar respuestas de las solicitudes a través de eventos.

Tabla 1 Descripción del RF “Manejar respuestas de las solicitudes a través de eventos”. Fuente: Elaboración propia.

Propósito	Lanzar el evento asociado a una solicitud al recibir la respuesta.	
Roles	Sistema.	
Precondiciones	1. Debe existir al menos una solicitud en la cola de solicitudes.	
Entidades tratadas	Entidad	Atributos
	Petición	<ul style="list-style-type: none"> • id • infoServiceCall • event
Descripción	<ol style="list-style-type: none"> 1. Tomar una petición de la cola de peticiones 2. Buscar su respuesta en la lista de respuestas. 3. Si existe una respuesta para la solicitud: <ol style="list-style-type: none"> 3.1. Lanzar el evento suscrito a la solicitud con los valores de la respuesta. 3.2. Eliminar la solicitud de la cola de solicitudes. 3.3. Eliminar la respuesta de la lista de respuesta. 4. Si no existe, se pasa a analizar otra solicitud. 	
Validaciones	1. El identificador de la respuesta debe coincidir con el identificador de la solicitud.	
Postcondiciones	1. Se actualizó la cola de solicitudes y la lista de respuestas.	
Prototipo	NP	

RF 3. Transformar las peticiones a un formato entendible.

Tabla 2 Descripción del RF “Transformar las peticiones a un formato entendible”. Fuente: Elaboración propia.

Propósito	Convertir los datos de una solicitud en el formato entendible por el Framework DGMJS.	
Roles	Sistema.	
Precondiciones		
Entidades tratadas	Entidad	Atributos
	Petición	<ul style="list-style-type: none"> • id • infoServiceCall • event

Descripción	<ol style="list-style-type: none"> 1. Extraer los siguientes datos de una petición: <ol style="list-style-type: none"> a Id b infoServiceCall 2. Formatear los datos extraídos en el formato entendible por el Framework DGMJS. El formato es el siguiente: <code>{[id solicitud]}/{[URL del servicio]}</code>
Validaciones	<ol style="list-style-type: none"> 1. El campo id e infoServiceCall de la entidad Petición no pueden estar vacíos.
Postcondiciones	<ol style="list-style-type: none"> 1. Se formatearon los datos de la petición.
Prototipo	NP

RF 4. Configurar llamada de los servicios locales.

Tabla 3 Descripción del RF “Configurar llamada a los servicios locales”. Fuente: Elaboración propia.

Propósito	Configurar el Framework DGMJS para hacer llamadas a los servicios locales.	
Roles	Desarrollador.	
Precondiciones		
Entidades tratadas	Entidad	Atributos
	ServiceMapping	<ul style="list-style-type: none"> • serviceURL • serviceMethod <ol style="list-style-type: none"> a. Url b. Parameter c. Method
Descripción	<ol style="list-style-type: none"> 1. Crear un objeto que representa un servicio con la información. <ol style="list-style-type: none"> a serviceURL. 2. Adicionar al objeto que representa el servicio los métodos con la siguiente información: <ol style="list-style-type: none"> a Url. b Parameter. c Method. 3. Adicionar el objeto que representa el servicio a los servicios de la entidad ServiceMapping. 	
Validaciones	<ol style="list-style-type: none"> 1. Cada uno de los datos llenados en el servicio debe coincidir con la Url, los métodos y los parámetros de los métodos respectivamente. 	
Postcondiciones	<ol style="list-style-type: none"> 1. Se actualizó la lista de configuraciones de llamadas a servicios web. 	
Prototipo	NP	

RF 6. Aceptar nuevos controladores de dispositivos

Tabla 4 Descripción del RF “Aceptar nuevos controladores de dispositivos”. Fuente: Elaboración propia.

Propósito	Actualizar un controlador existente.	
Roles	Sistema	
Precondiciones	1. Debe existir un cliente que lo utilice.	
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> 1. Revisar la configuración del sistema. 2. Obtener la configuración del servicio. 3. Sí existe cambio en el controlador. <ol style="list-style-type: none"> 3.1. Se debe detener el servicio. 4. Conformar un nuevo servicio con el controlador nuevo. 5. Iniciar el servicio. 	
Validaciones	1. Los nombres de los ensamblados que conforman el servicio deben coincidir con ficheros físicos.	
Postcondiciones	1. El controlador del servicio web quedó disponible.	
Prototipo	NP	

RF 7. Manejar dispositivos remotos.

Tabla 5 Descripción del RF “Manejar dispositivos remotos”. Fuente: Elaboración propia.

Propósito	Manejar un dispositivo ubicado en una estación de trabajo remota.	
Roles	Sistema.	
Precondiciones	<ol style="list-style-type: none"> 1. El servicio debe estar iniciado. 2. El controlador debe estar iniciado como un servicio en la estación de trabajo remota. 3. Se debe contar con permisos para conectarse al controlador remoto. 	
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> 1. Recibir una petición al servicio. 2. Ejecutar el método correspondiente en el controlador 3. Establece comunicación con el controlador remoto. 4. Si la conexión es satisfactoria. <ol style="list-style-type: none"> 4.1. Ejecutar el método del controlador remoto. 4.2. Generar traza de uso del dispositivo. 4.3. Retornar la información al servicio. 5. Si la conexión no es satisfactoria. <ol style="list-style-type: none"> 5.1. Mostrar mensaje de error. 	

Validaciones	1. La dirección del controlador remoto debe ser una URL valida.
Postcondiciones	1. Queda retornado un valor como respuesta a la petición recibida al servicio.
Prototipo	NP

RF 8. Aceptar el manejo de dispositivos remotamente.

Tabla 6 Descripción del RF "Aceptar el manejo de dispositivos remotamente". Fuente: Elaboración propia.

Propósito	Interactuar con un dispositivo remotamente.	
Roles	Sistema	
Precondiciones	1. El controlador local del dispositivo debe estar expuesto como un servicio.	
Entidades tratadas	Entidad	Atributos
Descripción	1. Recibir una petición de manejo de dispositivo. 2. Verificar que el dispositivo no se encuentre en uso. 3. Si se encuentra en uso. 3.1. Rechazar la solicitud. 4. Si no se encuentra en uso. 4.1. Ejecutar el método correspondiente a la petición. 4.2. Devolver el valor retornado a la estación de trabajo remoto.	
Validaciones		
Postcondiciones	Se interactuó con el dispositivo remoto.	
Prototipo	NP	

RF 9. Consultar el estado del servicio remotamente.

Tabla 7 Descripción del RF "Consultar el estado del servicio remotamente". Fuente: Elaboración propia.

Propósito	Ofrecer información sobre el funcionamiento del servicio remotamente.	
Roles	Sistema	
Precondiciones	1. El servicio de administración debe estar iniciado.	
Entidades tratadas	Entidad	Atributos
Descripción	1. Recibir una petición al servicio de administración. 2. Retornar el estado del servicio con la siguiente información: a Iniciado. b Detenido. c No existe.	

	d Controlador activo.
Validaciones	
Postcondiciones	Se obtuvo información del servicio.
Prototipo	NP

RF 10. Gestionar las peticiones recibidas desde el Framework JavaScript.

Tabla 8 Descripción del RF “Gestionar las peticiones recibidas desde el Framework DGMJS”. Fuente: Elaboración propia.

Propósito	Recibir las peticiones hechas por el Framework DGMJS a los servicios locales y devolver la respuesta de la petición.	
Roles	Sistema.	
Precondiciones	1. El servicio de proxy debe estar iniciado.	
Entidades tratadas	Entidad	Atributos
Descripción	1. Recibir una petición en el servicio proxy desde el framework DGMJS. 2. Validar que la petición provenga del Framework DGMJS. 3. Si la petición es válida. 3.1. Ejecutar la petición al servicio web correspondiente de acuerdo a la información proveniente del Framework DGMJS. 3.2. Retornar la respuesta al Framework DGMJS.	
Validaciones		
Postcondiciones	Se retorna una respuesta al Framework DGMJS proveniente del servicio web.	
Prototipo	NP	

RF 11. Validar el acceso a un servicio web.

Tabla 9 Descripción del RF “Validar el acceso a un servicio web”. Fuente: Elaboración propia.

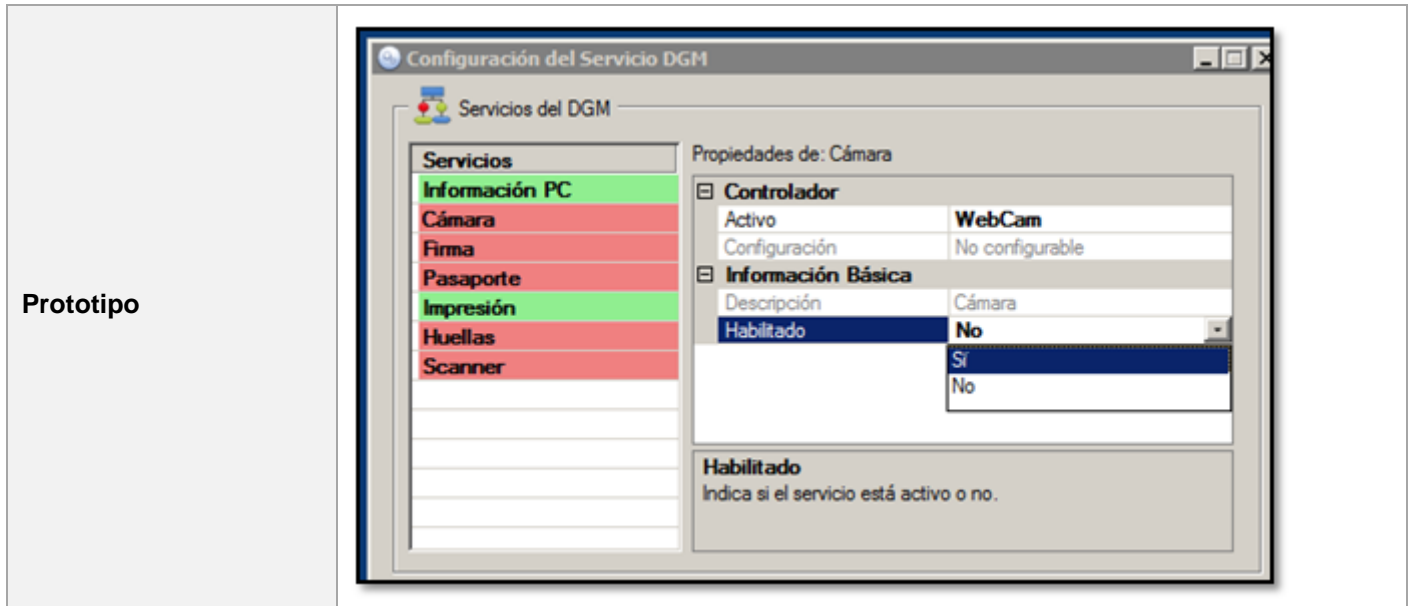
Propósito	Validar el acceso a un servicio web expuesto en el Servicio local para el manejo de dispositivos.	
Roles	Sistema	
Precondiciones	1. El servicio web debe estar iniciado en el Servicio local para el manejo de dispositivos.	
Entidades tratadas	Entidad	Atributos
Descripción	1. Recibir una petición al servicio web.	

	<ol style="list-style-type: none"> 2. Extraer la siguiente información de la petición: <ol style="list-style-type: none"> a Vía de acceso. b Usuario c Dispositivo. 3. Validar que la vía de acceso esté permitida y que el usuario tiene permisos sobre el dispositivo. 4. Si el acceso es permitido. <ol style="list-style-type: none"> 4.1. Procesar la petición. 4.2. Retornar la respuesta de la petición.
Validaciones	
Postcondiciones	Se concede o deniega el acceso al servicio web.
Prototipo	NP

RF 12. Gestionar los servicios web a usar en el Servicio local para el manejo de dispositivos.

Tabla 10 Descripción del RF "Gestionar los servicios web a usar en el Servicio local". Fuente: Elaboración propia.

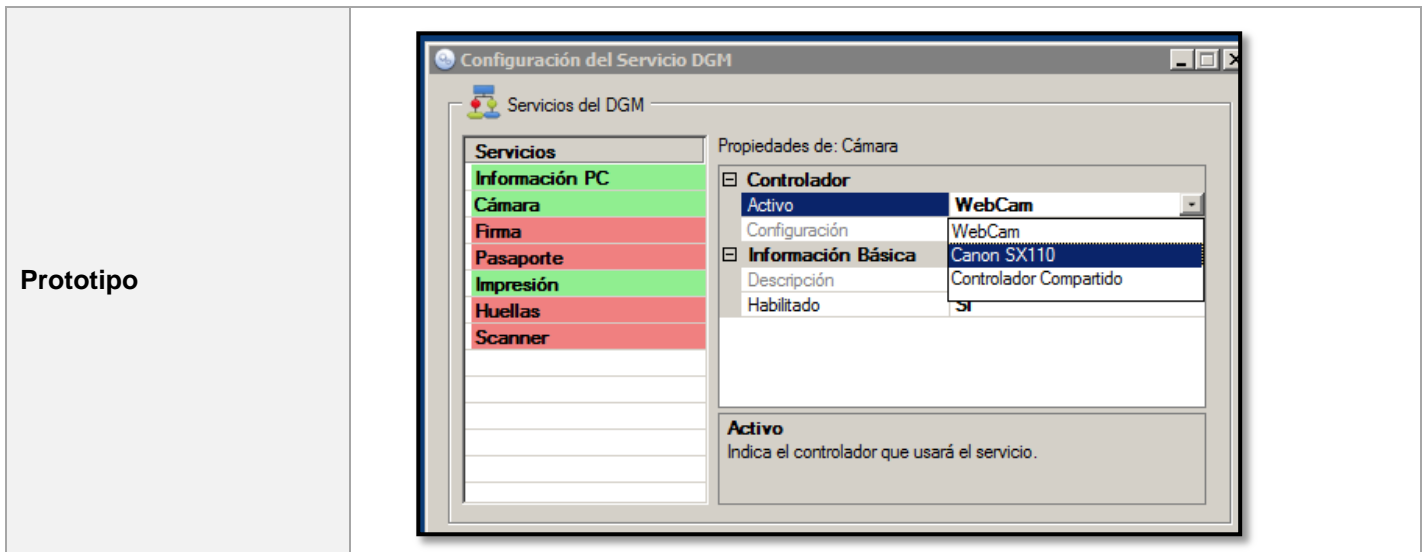
Propósito	Poder configurar que servicios web estarán activos y cuáles no.	
Roles	Administrador	
Precondiciones		
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el servicio que se desea gestionar. 2. Definir si el servicio estará habilitado o no con la siguiente información: <ol style="list-style-type: none"> a Sí. b No. Cambiarle el estado al servicio seleccionado.	
Validaciones		
Postcondiciones	El servicio especificado cambia de estado.	



RF 13. Gestionar los dispositivos a usar en el Servicio local para el manejo de dispositivos.

Tabla 11 Descripción del RF “Gestionar los dispositivos a usar en el Servicio local”. Fuente: Elaboración propia.

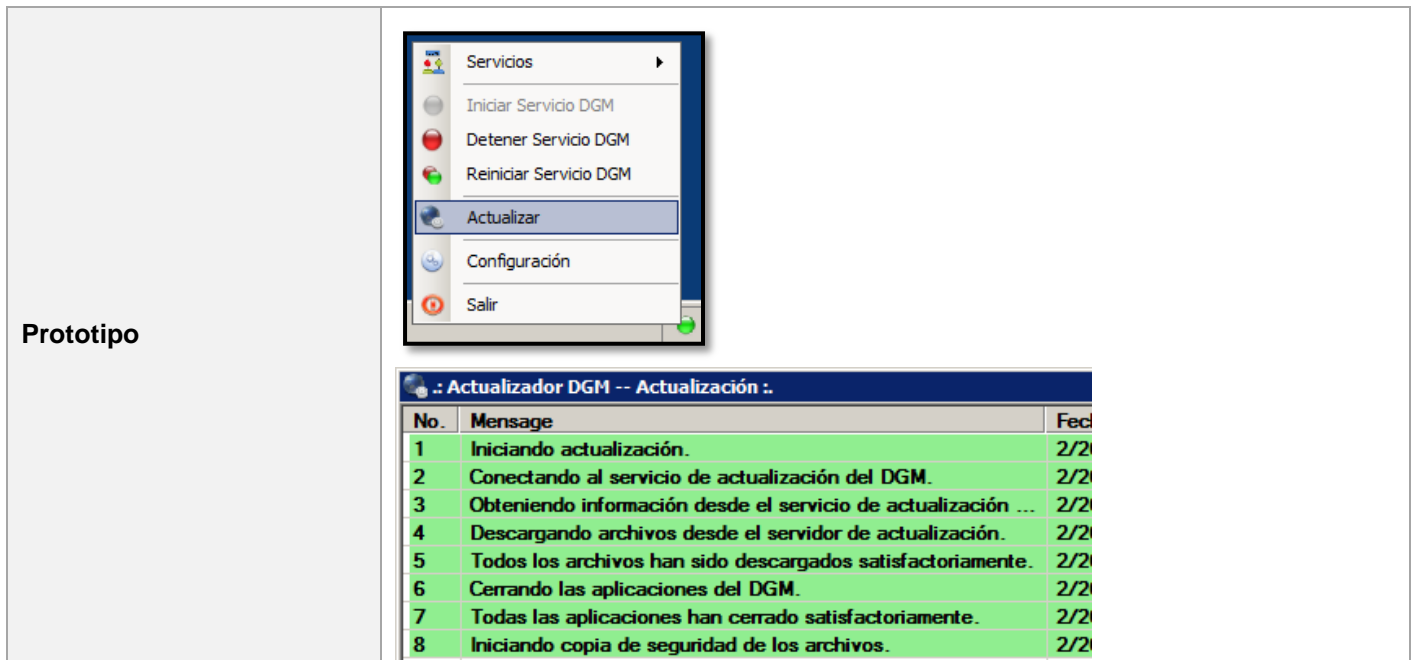
Propósito	Poder configurar que dispositivo usar para cada servicio web.	
Roles	Administrador y Sistema.	
Precondiciones		
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el servicio al cual se desea cambiar el dispositivo. 2. Seleccionar el dispositivo que se desea usar para el servicio. 3. Cambiar el dispositivo del servicio. 	
Validaciones		
Postcondiciones	Cambia el dispositivo para el servicio especificado.	



RF 14. Actualizar el Servicio local para el manejo de dispositivos.

Tabla 12 Descripción del RF “Actualizar el Servicio local para el manejo de dispositivos”. Fuente: Elaboración propia.

Propósito	Mantener el Servicio local para el manejo de dispositivos, todos los servicios web, y todos los controladores y sus dependencias actualizadas.	
Roles	Administrador y Sistema.	
Precondiciones	1. El servicio de actualización y la aplicación de actualización deben estar en ejecución y correctamente configuradas.	
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> 1. Se selecciona la opción “Actualizar” en el Panel de Control del DGM. 2. Conectar el actualizador al servicio de actualización. 3. Descargar la información necesaria para llevar a cabo el proceso de actualización. 4. Detener todos los servicios web. 5. Descargar todos los ficheros necesarios para la actualización. 6. Reemplazar todos los ficheros antiguos. Iniciar todos los servicios web ya actualizados.	
Validaciones		
Postcondiciones	El Servicio local para el manejo de dispositivos con todos los servicios web se encuentran actualizados.	



RF 15. Gestionar los servicios y dispositivos de las estaciones de trabajo.

Tabla 13 Descripción del RF “Gestionar los servicios y dispositivos de las estaciones de trabajo”. Fuente:
Elaboración propia.

Propósito	Permitir al administrador del sistema gestionar el estado de los servicios web y los dispositivos en las estaciones de trabajo.	
Roles	Administrador.	
Precondiciones	1. El administrador debe haber adquirido los permisos necesarios para poder gestionar los servicios necesarios de las estaciones de trabajo.	
Entidades tratadas	Entidad	Atributos
	Estación de trabajo	
Descripción	1. Mostrar listado de agrupaciones lógicas existentes. <ul style="list-style-type: none"> 1.1 Mostrar un mensaje indicando que no existen agrupaciones lógicas en caso que no exista ninguna. 2. Seleccionar agrupación lógica para obtener las estaciones de trabajo asociadas a ella. <ul style="list-style-type: none"> 2.1 Se selecciona la estación de trabajo a la cual se le desea gestionar los servicios y los dispositivos. <ul style="list-style-type: none"> 2.1.1 Mostrar los servicios y dispositivos asociados a la estación de trabajo. 2.1.2 Mostrar la opción “Adicionar” para añadir nuevos servicios y dispositivos asociados a la estación de trabajo. 2.1.3 Llenar los siguientes datos si se selecciona la opción “Adicionar”: <ul style="list-style-type: none"> a Nombre del servicio. 	

	<ul style="list-style-type: none"> b Cliente del servicio. c Controladores del servicio. d Ensamblados del servicio e Si está habilitado o no. f Dispositivo a usar. <p>2.1.4 Mostrar las opciones “Aceptar” y “Cancelar”.</p> <ul style="list-style-type: none"> 2.1.4.1 Guardar los cambios, si se selecciona la opción “Aceptar”. 2.1.4.2 Mostrar el listado de servicios y dispositivos asociados a la estación de trabajo seleccionada, si se selecciona la opción “Cancelar”. <p>2.1.5 Mostrar la opción “Modificar” para editar los servicios y dispositivos asociados a la estación de trabajo.</p> <p>2.1.6 Editar si se desea los campos que se listan a continuación:</p> <ul style="list-style-type: none"> a Cliente del servicio. b Controladores del servicio. c Ensamblados del servicio d Si está habilitado o no. e Dispositivo a usar. <p>2.1.7 Mostrar las opciones “Aceptar” y “Cancelar”.</p> <ul style="list-style-type: none"> 2.1.7.1 Guardar los cambios, si se selecciona la opción “Aceptar”. 2.1.7.2 Mostrar el listado de servicios y dispositivos asociados a la estación de trabajo seleccionada, si se selecciona la opción “Cancelar”. <p>2.1.8 Mostrar la opción “Eliminar” para borrar los servicios y dispositivos asociados a la estación de trabajo.</p> <p>2.1.9 Seleccionar los servicios y dispositivos asociados a la estación de trabajo que se deseen eliminar.</p> <ul style="list-style-type: none"> 2.1.9.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Aceptar”. <ul style="list-style-type: none"> a Eliminar la agrupación física seleccionada, si se selecciona “Si”. b Mostrar el listado de agrupaciones físicas, si se selecciona “No”.
Validaciones	Debe haberse seleccionado una estación de trabajo para poder realizar las acciones “Adicionar”, “Modificar” y “Eliminar” sobre los servicios.
Postcondiciones	Se adicionó, modificó o eliminó el servicio web y los dispositivos para la estación de trabajo especificada.

Prototipo

Gestionar servicios

Servicios

	Nombre	Descripción	Version	Tipo de servicio	Ensamblado
1	Video	Otro mas	2.9.9.0	Servicio Auxiliar	AForge.Video.dll
2	Este servicio	esta es la nueva des	6.0.9.3	Servicio Auxiliar	CameraService.Car
3	Validador	Valida las respuesta	1.0.0.0	Servicio de Dispositivos	DGMSecurityValidat

Página 1 de 1 10 Mostrando 1 - 3 de 3

Adicionar
Modificar
Eliminar
Cancelar

Gestionar servicios

Gestionar dispositivos

Dispositivos

	Descripción	Categoría	Marca	Modelo	Serie
1	DigitalPerson CISED	Lector de Huellas	Greenbit	U.are.U 4000B Reader	
2	Scanner Dacty Scan 15 MININT	Lector de Huellas	Greenbit	Dacty ID 15	
3	Lector de Firmas CISED	Lector de Firmas	Interlink	EpadInk	
4	Dacty Scan 40 MININT	Lector de Firmas	Greenbit	Dacty Scan 40	
5	Scanner Identidad	Escaner	Hp	ScanJet 4770	345JH4HK
6	DigitalPerson MININT	Lector de Huellas	Greenbit	U.are.U 4000B Reader	67736202

Página 1 de 1 10 Mostrando 1 - 6 de 6

Adicionar
Modificar
Eliminar
Cancelar

Gestionar dispositivos

RF 17. Gestionar agrupaciones físicas.

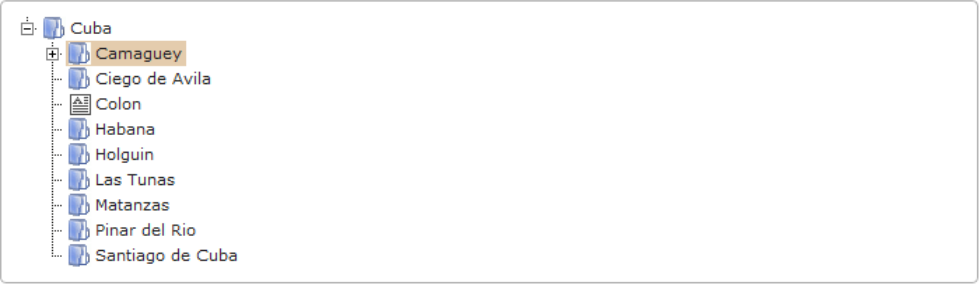
Tabla 14 Descripción del RF "Gestionar agrupación física". Fuente: Elaboración propia.

Propósito	Gestionar agrupaciones físicas.	
Roles	Administrador.	
Precondiciones	El administrador debe haber adquirido los permisos necesarios.	
Entidades	Entidad	Atributos

tratadas	Agrupación física	<ul style="list-style-type: none"> • Identificador. • Id de la agrupación padre. • Descripción. • Es contenedor.
Descripción	<ol style="list-style-type: none"> 1. Mostrar listado de agrupaciones físicas existentes. <ol style="list-style-type: none"> 1.1 Mostrar un mensaje indicando que no existen agrupaciones físicas en caso que no exista ninguna. 2. Seleccionar una agrupación física. 3. Mostrar la opción “Adicionar” para crear una nueva agrupación física. <ol style="list-style-type: none"> 3.1 Llenar los siguientes datos, si se selecciona la opción “Adicionar”: <ol style="list-style-type: none"> a Agrupación física padre. b Nombre. c Es contenedor. 3.2 Mostrar las opciones “Aceptar” y “Cancelar”. <ol style="list-style-type: none"> 3.2.1 Guardar la nueva agrupación física, si se selecciona la opción “Adicionar”. 3.2.2 Mostrar el listado de agrupaciones físicas existentes, si se selecciona la opción “Cancelar”. 4. Mostrar la opción “Modificar” para editar una agrupación física. <ol style="list-style-type: none"> 4.1 Mostrar los datos de la agrupación física en los siguientes campos: <ol style="list-style-type: none"> a Nombre b Es contenedor 4.2 Mostrar las opciones de “Aceptar” y “Cancelar”. <ol style="list-style-type: none"> 4.2.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Aceptar”. <ol style="list-style-type: none"> 4.2.1.1 Guardar los cambios en la agrupación física, si se selecciona “Si”. 4.2.1.2 Cancelar la acción si se selecciona “No”. 5. Mostrar la opción “Eliminar” para eliminar una agrupación física. <ol style="list-style-type: none"> 5.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Aceptar”. <ol style="list-style-type: none"> 5.1.1 Eliminar la agrupación física seleccionada, si se selecciona “Si”. 5.1.2 Mostrar el listado de agrupaciones físicas, si se selecciona “No”. 6. Mostrar la opción “Cancelar”. <ol style="list-style-type: none"> 6.1 Mostrar un mensaje de confirmación con las opciones “Si” y “No”, si se selecciona la opción “Cancelar”. <ol style="list-style-type: none"> 6.1.1 Ir a la página principal se selecciona “Si”. 6.1.2 Cerrar la alerta si se selecciona “No”. 	
Validaciones	<ol style="list-style-type: none"> 1. Comprobar que se haya seleccionado una agrupación física para poder realizar las acciones de “Adicionar”, “Modificar” o “Eliminar”. 2. Los campos “Nombre” en las opciones “Adicionar” y “Modificar” no pueden estar vacíos. 3. Al “Eliminar” una agrupación comprobar que ésta no contenga dispositivos asociados o contenga elementos hijos asociados a otros dispositivos. 	
Postcondiciones	Se crearon, modificaron o eliminaron las agrupaciones físicas.	

Prototipo

Gestionar agrupación física



Mostrar agrupaciones físicas.

Adicionar agrupación física

Nombre

Es contenedor

Adicionar agrupación física.

Modificar agrupación física

Nombre

Es contenedor

Modificar agrupación física.

RF 18. Gestionar Estaciones de trabajo.

Tabla 15 Descripción del RF “Gestionar Estaciones de trabajo”. Fuente: Elaboración propia.

Propósito	Permitir activar, desactivar, modificar y mostrar las estaciones de trabajo	
Roles	Administrador, Usuario	
Precondiciones	El administrador debe haber adquirido los permisos necesarios para realizar las operación excepto para activar una estación de trabajo que puede ser realizado por el Usuario.	
Entidades tratadas	Entidad	Atributos
Descripción	1. Mostrar la interfaz de activar estación de trabajo, se selecciona la opción “Activar estación” 1.1 Especificar los campos siguientes: a Serial de activación. b Nombre.	

	<ul style="list-style-type: none"> c Tipo de dispositivo. d Ubicación física. <p>1.2 Mostrar las opciones “Especificaciones técnicas”, “Aceptar” y “Cancelar”.</p> <p>1.2.1 Mostrar un cuadro de diálogo con las especificaciones técnicas de la estación de trabajo que se está activando, si se selecciona la opción “Especificaciones técnicas”.</p> <p>1.2.2 Mostrar un mensaje con el nombre de la estación de trabajo, la agrupación física, la agrupación lógica y las opciones “Si” y “No” pidiendo la confirmación para activar la estación de trabajo, si se selecciona la opción “Aceptar”.</p> <p>1.2.2.1 Guardar los datos de la estación de trabajo si se selecciona “Si”.</p> <p>1.2.2.2 Mostrar la interfaz de usuario para agregar una nueva estación de trabajo si se selecciona “No”.</p> <p>1.2.3 Ir a la página principal si se selecciona la opción “Cancelar”.</p> <p>2. Mostrar la interfaz de desactivar estación de trabajo si se selecciona la opción “Desactivar estación”.</p> <p>2.1 Mostrar un listado con las agrupaciones lógicas. De cada agrupación se mostrarán los campos:</p> <ul style="list-style-type: none"> a Nombre. b Tipo de agrupación. <p>2.2 Mostrar un listado con las estaciones de trabajo pertenecientes a una agrupación lógica, si se selecciona alguna de las que se muestran en el listado anterior. Cada estación de trabajo deberá mostrar los valores de los campos:</p> <ul style="list-style-type: none"> a Nombre. b Marca. c Modelo. <p>2.3 Mostrar las opciones “Aceptar” y “Cancelar”.</p> <p>2.4 Mostrar un mensaje de confirmación para eliminar la estación de trabajo seleccionada, con las opciones “Si” y “No”, si se selecciona la opción “Aceptar”.</p> <p>2.4.1 Desactivar la estación de trabajo si se selecciona la opción “Si”.</p> <p>2.4.2 Mostrar la interfaz para desactivar una estación de trabajo nuevamente, si se selecciona “No”.</p>
Validaciones	<ol style="list-style-type: none"> 1. Comprobar que el serial que se introdujo esté activo y no se esté usando por ninguna estación de trabajo. 2. Comprobar que se haya seleccionado una agrupación física al activar la estación de trabajo. 3. Comprobar que los campos “Nombre” y “Tipo de dispositivo” no estén vacíos. 4. En la desactivación comprobar que se haya seleccionado una agrupación lógica y luego una estación de trabajo para poder desactivar la misma.
Postcondiciones	<ol style="list-style-type: none"> 1. Se guardan los datos de una nueva estación de trabajo si se selecciona la opción activar. 2. Se elimina una estación de trabajo determinada si se selecciona la opción desactivar.

Prototipo

Activar estación de trabajo

Activación

Serial de Activación:

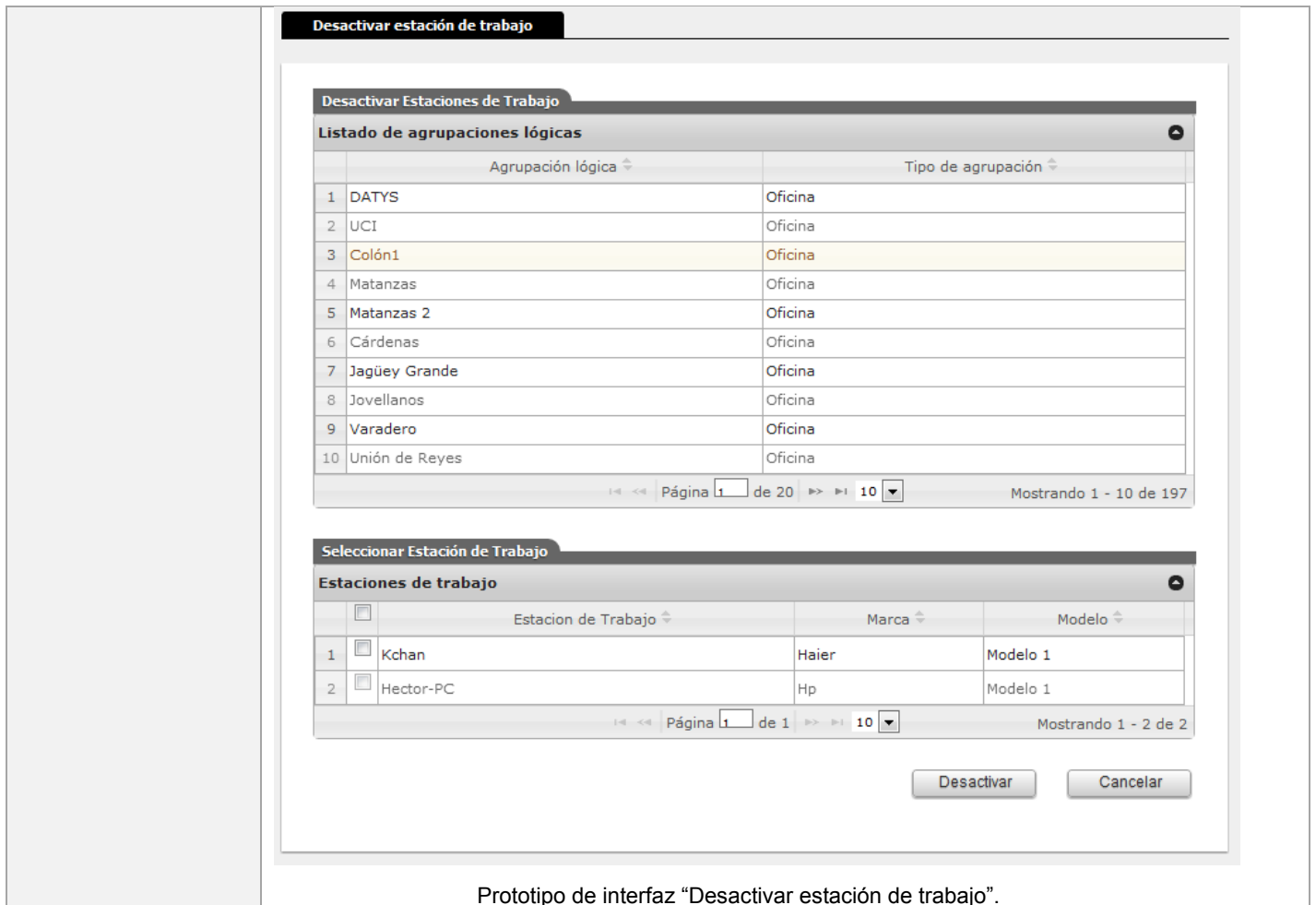
Dispositivo

Nombre: Tipo de dispositivo:

Estación de trabajo

Ubicación física:

Prototipo de interfaz "Activar estación de trabajo".



Prototipo de interfaz "Desactivar estación de trabajo".

RF 19. Gestionar mantenimiento de los dispositivos.

Tabla 16 Descripción del RF "Gestionar mantenimiento de los dispositivos". Fuente: Elaboración propia.


Propósito	Poder definir cada cuanto tiempo se le debe dar mantenimiento a los dispositivos y obtener cuales dispositivos necesitan mantenimiento.	
Roles	Administrador.	
Precondiciones	El administrador debe haber adquirido los permisos necesarios.	
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> Mostrar un campo para que el usuario seleccione el tiempo de uso de los dispositivos. Mostrar la opción "Obtener dispositivos". 	

	<ol style="list-style-type: none"> 2.1 Mostrar un listado con los dispositivos que tienen más del tiempo de uso del especificado por el usuario. 2.2 Mostrar para cada dispositivo los campos siguientes: <ol style="list-style-type: none"> a Descripción. b No de serie. c Tipo de dispositivo. d Estación. e Ubicación. f Tiempo de uso. 3. Mostrar las opciones "Poner en modo mantenimiento". <ol style="list-style-type: none"> 3.1 Mostrar un mensaje de confirmación si se selecciona la opción "Poner en modo mantenimiento", con las opciones "Sí" y "No". 3.2 Poner los dispositivos en modo mantenimiento si se selecciona la opción "Sí". 3.3 Ocultar la alerta si se selecciona "No". 4. Mostrar la opción "Cancelar". <ol style="list-style-type: none"> 4.1 Ir a la página principal si se selecciona "Cancelar".
Validaciones	Debe haberse seleccionado al menos un dispositivo para poner en mantenimiento cuando se seleccione la opción "Poner en modo mantenimiento".
Postcondiciones	Se ponen en mantenimiento los dispositivos seleccionados.
Prototipo	NP

RF 20. Mostrar reportes.

Tabla 17 Descripción del RF "Mostrar reportes". Fuente: Elaboración propia.

Propósito	Obtener reportes del estado de los dispositivos que componen la red.	
Roles	Administrador	
Precondiciones	El administrador debe haber adquirido los permisos necesarios.	
Entidades tratadas	Entidad	Atributos
Descripción	<ol style="list-style-type: none"> 1. Seleccionar la opción Reportes. <ol style="list-style-type: none"> 1.1. Seleccionar la opción "Activación", si se desea ver el reporte de las estaciones de trabajo activas. <ol style="list-style-type: none"> 1.1.1. Mostrar la opción "Ubicación lógica". 1.1.2. Seleccionar una ubicación lógica. 1.1.3. Mostrar un listado de las estaciones de trabajo que están activas dependiendo de la ubicación lógica seleccionada. <ol style="list-style-type: none"> 1.1.3.1. De cada estación de trabajo mostrar los campos: <ol style="list-style-type: none"> a Nombre. 	

	<ul style="list-style-type: none"> b Ubicación. c Marca. d Fecha de modificación. e Activo. <p>1.2. Seleccionar la opción "Seriales", si se desea ver los listados de los seriales activos por ubicación lógica.</p> <p>1.2.1. Mostrar los campos:</p> <ul style="list-style-type: none"> a Ubicación lógica. b Cantidad. <p>1.2.2. Mostrar un listado de los seriales activos por ubicaciones lógicas, si se selecciona una ubicación lógica.</p> <p>1.2.3. Mostrar la opción "Generar".</p> <p>1.2.3.1. Si se selecciona la opción "Generar", crear la cantidad de nuevos seriales especificados en el campo "Cantidad" y asociarlos a la ubicación lógica seleccionada.</p> <p>1.2.3.2. Mostrar el listado de los seriales asociados a la ubicación lógica seleccionada.</p>
Validaciones	Comprobar que esté seleccionada una ubicación lógica y especificada una cantidad si se pulsa la opción "Generar".
Postcondiciones	Se obtuvo un reporte que muestra el estado de los dispositivos que componen la red.
Prototipo	 <p>Prototipo de interfaz "Reporte activación".</p>

Reporte de los seriales

Generación de seriales

Ubicación lógica: Cantidad:

Listado de los seriales de activación

	Serial de activación ↕
1	5b9aa5fd29a24e229db091be91e6f91e
2	24b467e1eb814b0d91db2e57877a37cc
3	d030613eded642578bf273f40f65475f
4	cef62cd3febf41d8b3a7c5ea3b94ffca
5	181d0a5f54394701b08d5ab223b085e9
6	e6de5cf0194b45118de1645ddbe66ad0
7	b1b9a3edf94849e3856d6d269c8f0bab

« << | Página 1 de 1 | >> » 10 ▼ Mostrando 1 - 7 de 7

Prototipo de interfaz "Reporte de los seriales".

Anexo 4 Especificaciones de clases

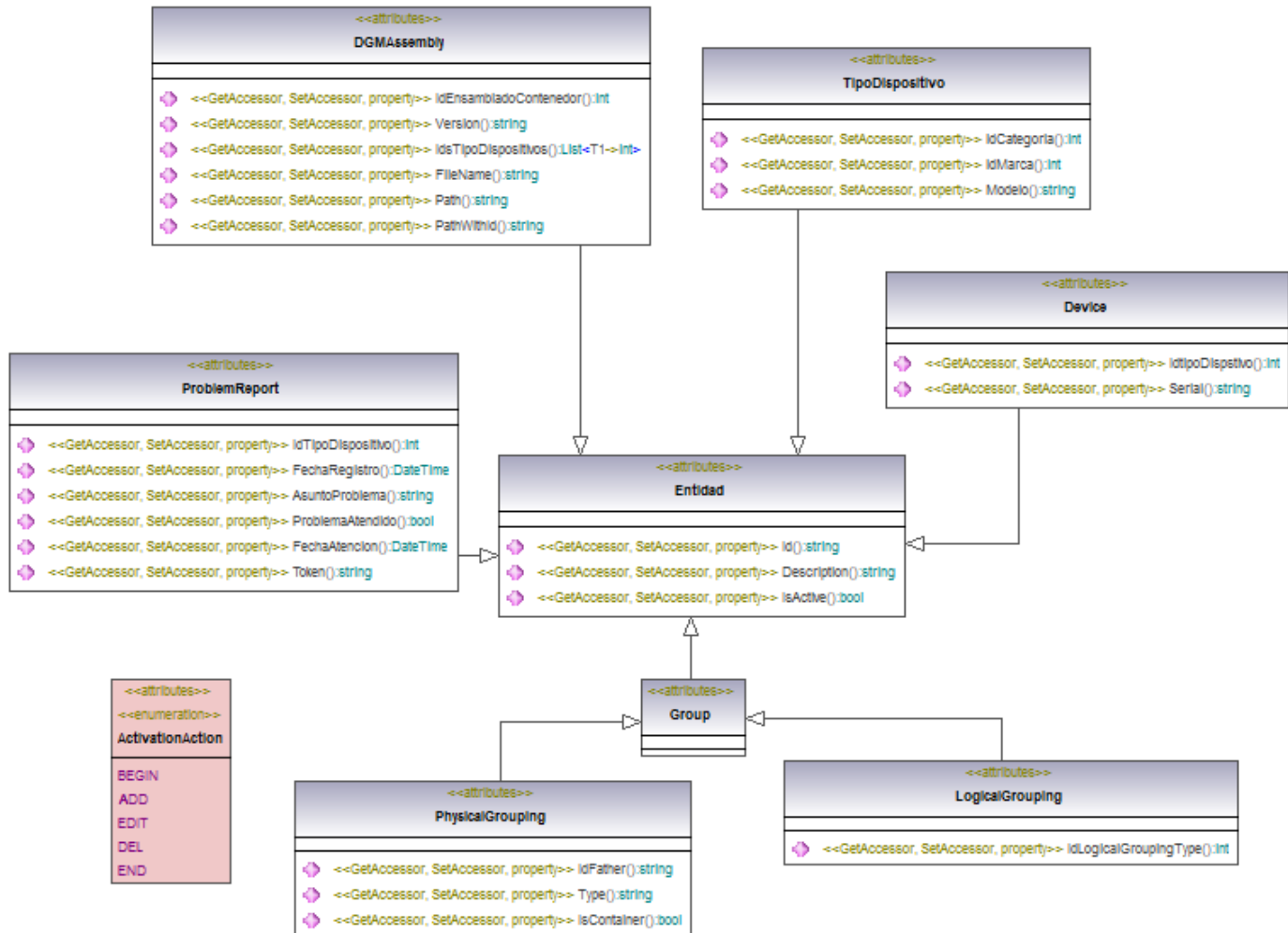


Figura 2. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Parte I. Fuente: Elaboración propia



Figura 3 Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Parte II. Fuente: Elaboración propia.

Tabla 18. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase DGMAssembly. Fuente: Elaboración propia.

Nombre	DGMAssembly	
Descripción	Es la clase que define los atributos de un ensamblado dentro del negocio.	
Visibilidad	Atributo	Descripción
Público	IdEnsambladoContenedor	Identificador del ensamblado que contiene a este ensamblado.
Público	Version	Versión del ensamblado.
Público	idsTipoDispositivos	Lista con los identificadores de los dispositivos que el ensamblado controla.
Público	FileName	Nombre del fichero del ensamblado
Público	Path	Ruta del fichero del ensamblado
Público	PathWithId	Ruta e identificador del fichero del ensamblado

Tabla 19. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase TipoDispositivo. Fuente: Elaboración propia.

Nombre	TipoDispositivo	
Descripción	Es la clase que define los atributos de un tipo de dispositivo dentro del negocio.	
Visibilidad	Atributo	Descripción
Público	IdCategoria	Identificador de la categoría a la cual pertenece el tipo de dispositivo.
Público	Marca	Marca del dispositivo.
Público	Modelo	Modelo del dispositivo.

Tabla 20. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase Device. Fuente: Elaboración propia.

Nombre	Device	
Descripción	Es la clase que define los atributos de un dispositivo dentro del negocio.	
Visibilidad	Atributo	Descripción
Público	IdTipoDispositivo	Identificador del tipo de dispositivo.
Público	Serial	Número de serie del dispositivo.

Tabla 21. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase Entidad. Fuente: Elaboración propia.

Nombre	Entidad	
Descripción	Es la clase que define los atributos base del resto de las entidades del negocio.	
Visibilidad	Atributo	Descripción
Público	Id	Identificador de la entidad.
Público	Description	Nombre o descripción de la entidad.
Público	IsActive	Campo lógico que define si el registro está activo.

Tabla 22. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase Group. Fuente: Elaboración propia.

Nombre	Group	
Descripción	Es la clase base que agrupa las agrupaciones físicas y lógicas del negocio.	
Visibilidad	Atributo	Descripción.

Tabla 23. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase LogicalGroup. Fuente: Elaboración propia.

Nombre	LogicalGrouping	
Descripción	Es la clase que define los atributos de una agrupación lógica dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	IdLogicalGroupType	Identificador del tipo de agrupación lógica.

Tabla 24. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase PhysicalGroup. Fuente: Elaboración propia.

Nombre	PhysicalGrouping	
Descripción	Es la clase que define los atributos de una agrupación física dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	idFather	Identificador de la agrupación física que contiene esta agrupación física.
Público	Type	Tipo de agrupación física.
Público	IsContainer	Indica si esta entidad física contiene otras entidades.

Tabla 25. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase ProblemReportParser. Fuente: Elaboración propia.

Nombre	ProblemReportParser	
Descripción	Es la clase que define los atributos del reporte de un problema dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	Topic	Contiene el asunto del problema.
Público	Description	Contiene la descripción del problema.

Tabla 26. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase Workstation. Fuente: Elaboración propia.

Nombre	Workstation	
Descripción	Es la clase que define los atributos de una estación de trabajo dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	id	Identificador de la estación de trabajo.
Público	idPhysicalGroup	Identificador de la agrupación física en la cual se encuentra.
Público	idLogicalGroup	Identificador de la agrupación lógica en la cual se encuentra
Público	idDeviceType	Identificador del tipo de dispositivo que es.
Público	Name	Nombre de la estación de trabajo.

Público	Especification	Especificaciones de la estación de trabajo.
Público	Token	Token de la estación de trabajo.

Tabla 27. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase UpdaterServiceConfiguration. Fuente: Elaboración propia.

Nombre	UpdaterServiceConfiguration	
Descripción	Es la clase que define los atributos de la configuración del servicio de actualización.	
Visibilidad	Atributo	Descripción.
Público	ServerBaseAddress	Contiene la dirección base del servidor de actualizaciones.

Tabla 28. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase InformationSection. Fuente: Elaboración propia.

Nombre	InformationSection	
Descripción	Es la clase que define los atributos de una sección de información dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	Id	Identificador de la sección.
Público	TimeOut	Tiempo de expiración de la sección en el servidor.
Público	XmlInformation	Contiene la información en formato XML.

Tabla 29. Diagrama de clases del negocio. Sistema para el control centralizado de dispositivos. Descripción de la clase ServerSideUpdateFile. Fuente: Elaboración propia.

Nombre	ServerSideUpdateFile	
Descripción	Es la clase que define los atributos de un fichero de actualización en el servidor dentro del negocio.	
Visibilidad	Atributo	Descripción.
Público	FileName	Nombre del fichero.
Público	ServerPath	Ruta del fichero en el servidor.
Público	LocalPath	Ruta del fichero en la estación de trabajo.
Público	Hash	Información del hash del fichero.
Público	User	Usuario para poder descargar el fichero.
Público	Pass	Contraseña para poder descargar el fichero.

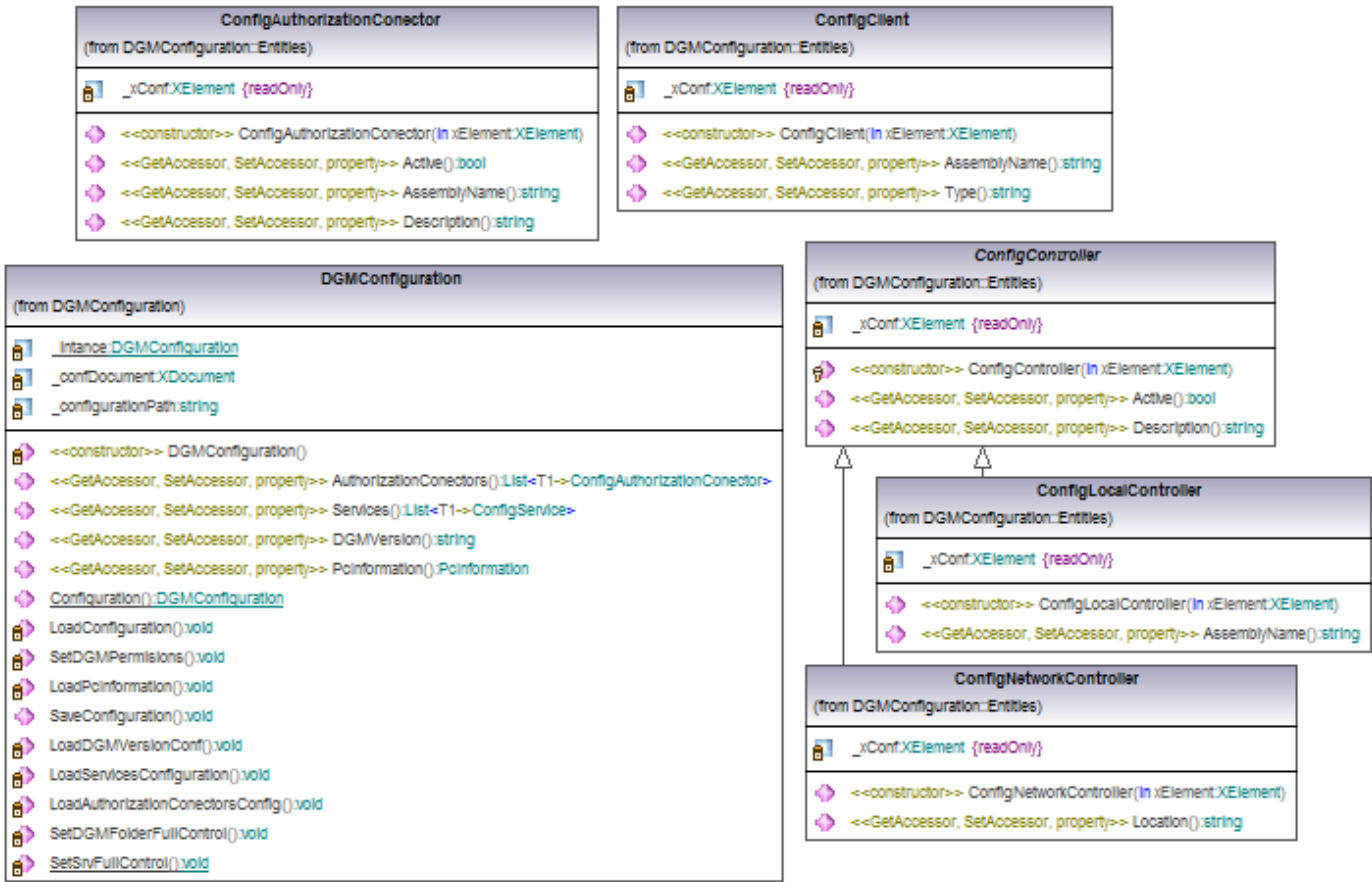


Figura 4 Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos. Fuente: Elaboración propia.

Tabla 30. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos. Descripción de la clase ConfigAuthorizationConector. Fuente: Elaboración propia.

Nombre	ConfigAuthorizationConector	
Descripción	Es la clase que define los atributos de la configuración de un conector de autorización.	
Visibilidad	Atributo	Descripción.
Privado	_xConfElement	Contiene un xml con la configuración.
Público	Active	Indica si el conector está activo o no.
Público	AssemblyName	Nombre del ensamblado del conector.
Público	Description	Descripción del conector de autorización.

Tabla 31. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos. Descripción de la clase ConfigClient. Fuente: Elaboración propia.

Nombre	ConfigClient
--------	--------------

Descripción	Es la clase que define los atributos de la configuración de un cliente.	
Visibilidad	Atributo	Descripción.
Privado	_xConfElement	Contiene un xml con la configuración.
Público	Active	Indica si el cliente está activo o no.
Público	AssemblyName	Nombre del ensamblado del cliente.

Tabla 32. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos.

Descripción de la clase ConfigController. Fuente: Elaboración propia.

Nombre	ConfigController	
Descripción	Es la clase que define los atributos de la configuración de un controlador.	
Visibilidad	Atributo	Descripción.
Privado	_xConfElement	Contiene un xml con la configuración.
Público	Active	Indica si el controlador está activo o no.
Público	Description	Descripción del controlador.

Tabla 33. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos.

Descripción de la clase ConfigLocalController. Fuente: Elaboración propia.

Nombre	ConfigLocalController	
Descripción	Es la clase que define los atributos de la configuración de un controlador local.	
Visibilidad	Atributo	Descripción.
Privado	_xConfElement	Contiene un xml con la configuración.
Público	AssemblyName	Nombre del ensamblado del controlador local.

Tabla 34. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos.

Descripción de la clase ConfigNetworkController. Fuente: Elaboración propia.

Nombre	ConfigNetworkController	
Descripción	Es la clase que define los atributos de la configuración de un controlador de red.	
Visibilidad	Atributo	Descripción.
Privado	_xConfElement	Contiene un xml con la configuración.
Público	Location	Ubicación donde se encuentra el controlador de red.

Tabla 35. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos.

Descripción de la clase DGMConfiguration. Fuente: Elaboración propia.

Nombre	DGMConfiguration	
Descripción	Es la clase que define los atributos de la configuración del Servicio local del DGM.	
Visibilidad	Atributo	Descripción.
Privado	_instance	Contiene un instancia de la configuración.
Privado	_confDocument	Contiene la configuración en un XML.

Privado	_confPath	Contiene la ruta del archivo de configuración
Público	AuthorizationConectors	Contiene una lista con la configuración de todos los conectores de autorización.
Público	Services	Contiene una lista con la configuración de todos los servicios.
Público	DGMVersion	Contiene la versión del DGM
Público	PcInformation	Contiene la información de la estación de trabajo

Tabla 36. Diagrama de clases del módulo de configuración. Servicio local para el manejo de dispositivos.
Descripción de la clase DGMConfiguration. Fuente: Elaboración propia.

Nombre	DGMConfiguration		
Descripción	Es la clase encargada de obtener toda la configuración del sistema y establecerla.		
Visibilidad	Método	Argumentos	Descripción
Público	LoadConfiguration		Carga la configuración desde el archivo.
Público	SetDGMPermissions		Establece permisos a la aplicación DGM.
Público	LoadPcInformation		Carga la información de la estación de trabajo.
Público	SaveConfiguration		Guarda la configuración en un archivo.
Público	LoadDGMVersionConf		Carga la versión del sistema.
Público	LoadServicesConfiguration		Carga la configuración de los servicios.
Público	LoadAuthorizationConectorsConfig		Carga la configuración de los conectores de autorización
Público	SetDGMFolderFullControl		Define permisos a la carpeta de la aplicación
Público	SetSrvFullControl		Establece permisos al servicio DGM.

Anexo 5 Servicios del sistema

Tabla 37 Descripción del servicio EncoderWCFService. Fuente: Elaboración propia.

Nombre	EncoderWCFService		
Descripción	Servicio encargado de realizar las acciones sobre los nomencladores.		
Visibilidad	Método	Argumentos	Descripción
Público	GetTIPOAGRUPACIONLOGICA		Devuelve el nomenclador TIPOAGRUPACIONLOGICA

Tabla 38 Descripción del servicio ProblemReportWCFService. Fuente: Elaboración propia.

Nombre	ProblemReportWCFService		
Descripción	Su función es recibir los reportes de los problemas generados enviados por el Servicio local para el manejo de dispositivos en cada una de las estaciones de trabajo.		
Visibilidad	Método	Argumentos	Descripción
Público	ReportProblem	pclid, tokenKey, xmlReport	Persiste un reporte de un problema en la base de datos a partir de la información del reporte contenida en el argumento xmlReport.

Tabla 39 Descripción del servicio PcInformationWCFService. Fuente: Elaboración propia.

Nombre	PcInformationWCFService		
Descripción	Este servicio recibe la información de una estación de trabajo para luego devolverla al servidor en el momento que se solicite. Este servicio es usado para transmitir de forma segura la información de una estación de trabajo.		
Visibilidad	Método	Argumentos	Descripción
Publico	PcInformationWCFService		Crea una instancia del servicio.
Público	IntroduceDataToServer	token, key, sessionId, xmlInformation	Introduce la información de una estación de trabajo contenida en el argumento en xmlInformation en un contenedor de sesiones para que la misma pueda ser extraída luego por el servidor.
Público	ExtractDataFromServer	sessionId	Extrae la información que contenga como id un valor igual al argumento.
Privado	AddSection	session	Inserta la sección de información pasada en el argumento en el contenedor de sesiones.
Privado	DecreaseTimeOut		Decrementa el tiempo de expiración de la sección para pasado un tiempo si esta no es extraída entonces expire.

Tabla 40 Descripción del servicio NotificationWCFService. Fuente: Elaboración propia.

Nombre	NotificationWCFService		
Descripción	Es el encargado de recibir las notificaciones generadas por el Servicio local para el manejo de dispositivos en las estaciones de trabajos y hacerlas persistir en la base de datos.		
Visibilidad	Método	Argumentos	Descripción
Publico	Notify	token,key, log	Persiste una notificación contenida dentro del argumento <i>log</i> en la base de datos.

Tabla 41 Descripción del servicio UpdaterWCFService. Fuente: Elaboración propia.

Nombre	UpdaterWCFService		
Descripción	El servicio de actualizaciones le permite al Servicio local para el manejo de dispositivos en las estaciones de trabajo obtener las actualizaciones del sistema.		
Visibilidad	Método	Argumentos	Descripción
Publico	GetUpdateFiles	token, key	Genera la lista de ficheros que el Servicio local debe descargar para actualizarse.
Público	GetConfig	token, key	Genera la configuración que el Servicio local debe descargar para actualizarse.

Tabla 42 Descripción del servicio LocationWCFService. Fuente: Elaboración propia.

Nombre	LocationWCFService		
Descripción	Es el encargado de devolver la ubicación física de una o varias oficinas y de las estaciones de trabajo.		
Visibilidad	Método	Argumentos	Descripción
Publico	LoadOfficeByToken	token	Devuelve una agrupación lógica u oficina a partir del <i>token</i> de una estación de trabajo contenida en ella.
Público	LoadAllOffices		Devuelve todas las agrupaciones lógicas u oficinas.
Privado	LoadWorkStationByToken	token	Devuelve una estación de trabajo a partir del <i>token</i> de la misma.

Tabla 43 Descripción del servicio LocationWCFService. Fuente: Elaboración propia.

Nombre	WorkstationWCFService		
Descripción	Es el encargado de realizar las acciones sobre las entidades relacionadas con las estaciones de trabajo.		
Visibilidad	Método	Argumentos	Descripción
Público	EnabledWorkStation	workStation, serial	Activa las estaciones de trabajo.

Público	DisableWorkstation	idWorkStation	Desactiva las estaciones de trabajo y elimina las asociaciones correspondientes entre las entidades relacionadas con las estaciones de trabajo.
---------	--------------------	---------------	---

Tabla 44 Descripción del servicio del Runtime ActivationRService. Fuente: Elaboración propia.

Nombre	ActivationRService		
Descripción	Servicio del Runtime en el cual se definen las responsabilidades correspondientes a la activación y desactivación de las estaciones de trabajo así como el envío de las entidades y las acciones que se realizarán sobre estas (Adicionar, Modificar y Eliminar).		
Visibilidad	Método	Argumentos	Descripción
Público	ActionTypeEvent	instanceld, action	Evento que permite enviar el tipo de acción a realizar.
Público	ActionTypeEvent	instanceld, navInfo	Evento que permite enviar una entidad que contiene el tipo de acción a realizar y la entidad.
Público	CallSerialEvent	instanceld, serial	Envía el serial de la pc en el proceso de activación de la estación de trabajo para comprobar si es válido.
Público	CallWorkstationEvent	instanceld, workStation	Envía la estación de trabajo en el proceso de activación.
Público	EnabledWorkStation	workStation, serial	Activa las estaciones de trabajo.
Público	DisableWorkstation	idWorkStation	Desactiva las estaciones de trabajo y elimina las asociaciones correspondientes entre las entidades relacionadas con las estaciones de trabajo.
Público	CheckSerial	serial	Comprueba si el serial de activación que se envía corresponde a una agrupación lógica.

Tabla 45 Descripción del servicio del Bussines ManageEntityEM. Fuente: Elaboración propia.

Nombre	ManageEntityEM		
Descripción	Servicio del negocio en el cual se definen las acciones que se realizarán sobre cada una de las entidades.		
Visibilidad	Método	Argumentos	Descripción
Público	AddLogicalGrouping	plogicalGrouping	Adiciona una agrupación lógica:

Público	EditLogicalGrouping	plogicalGrouping	Modifica una agrupación lógica:
Público	DeleteLogicalGrouping	plogicalGrouping	Elimina una agrupación lógica:
Público	AddPhisicalGrouping	pPhisicalGrouping	Adiciona una agrupación física.
Público	EditPhisicalGrouping	pPhisicalGrouping	Modifica una agrupación física.
Público	DeletePhisicalGrouping	pPhisicalGrouping	Elimina una agrupación física.
Público	AddDeviceType	pDeviceType	Adiciona un tipo de dispositivo.
Público	EditDeviceType	pDeviceType	Modifica un tipo de dispositivo.
Público	DeleteDeviceType	pDeviceType	Elimina un tipo de dispositivo.
Público	AddDevice	device	Adiciona un dispositivo.
Público	EditDevice	device	Modifica un dispositivo.
Público	AddAssembly	assembly	Adiciona un ensamblado.
Público	EditAssembly	assembly	Modifica un ensamblado.
Público	DeleteAssembly	assembly	Elimina un ensamblado.

Anexo 6 Descripción de los workflows

Workflow "ActivationWf"

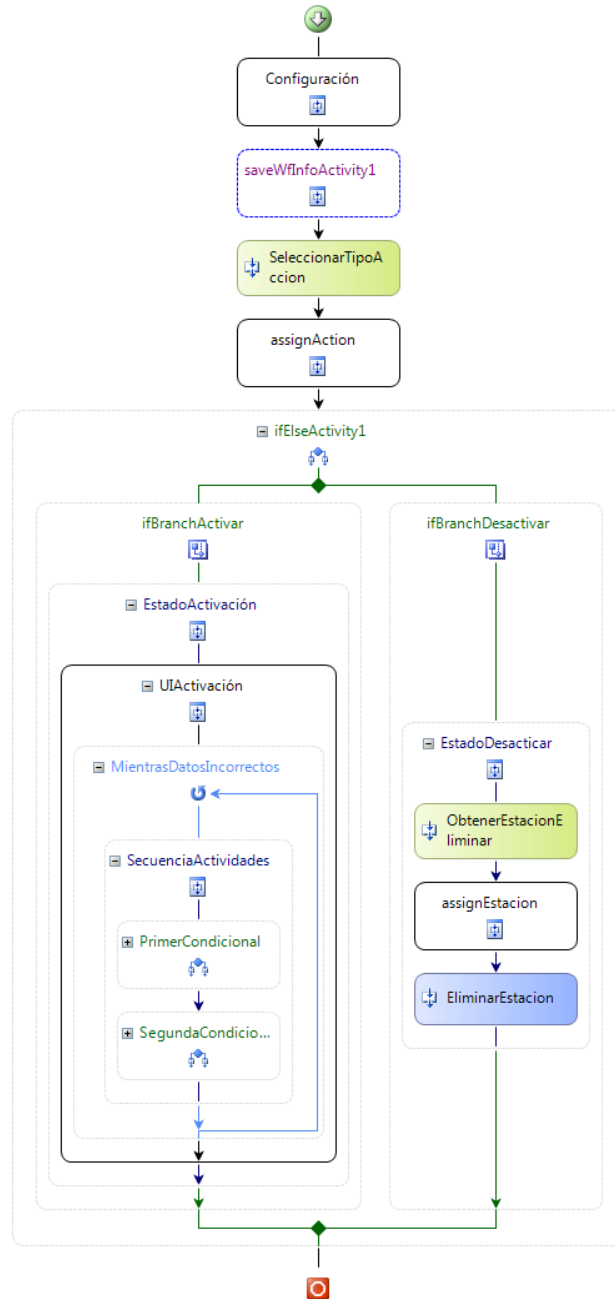


Figura 5 Vista detallada del workflow "ActivationWf". Fuente: Elaboración propia.

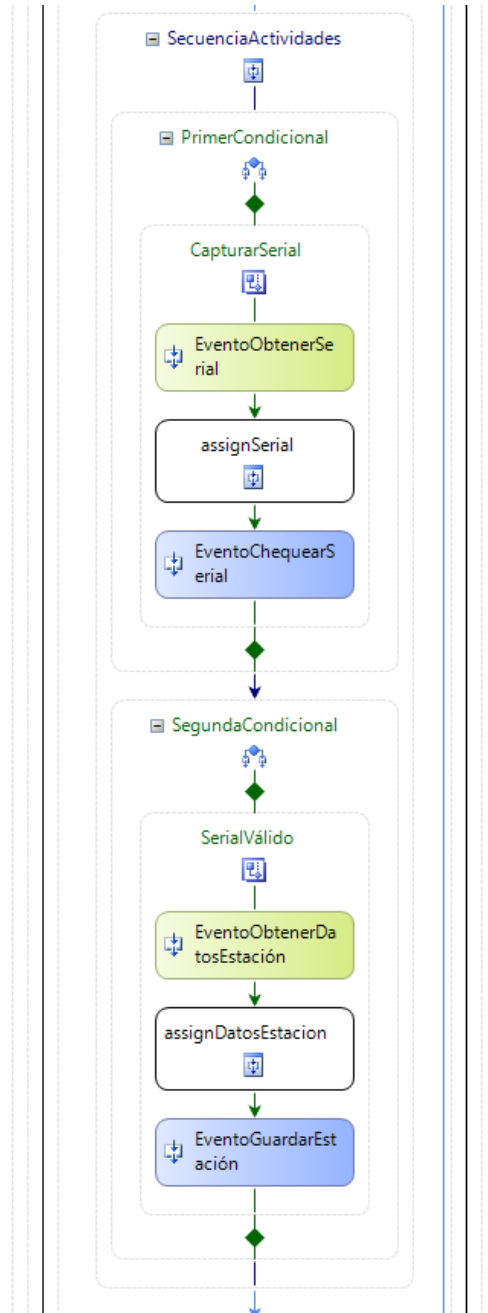


Figura 6 Vista detallada del workflow "ActivationWf". Fuente: Elaboración propia.

Workflow "ManageEntitiesWf"

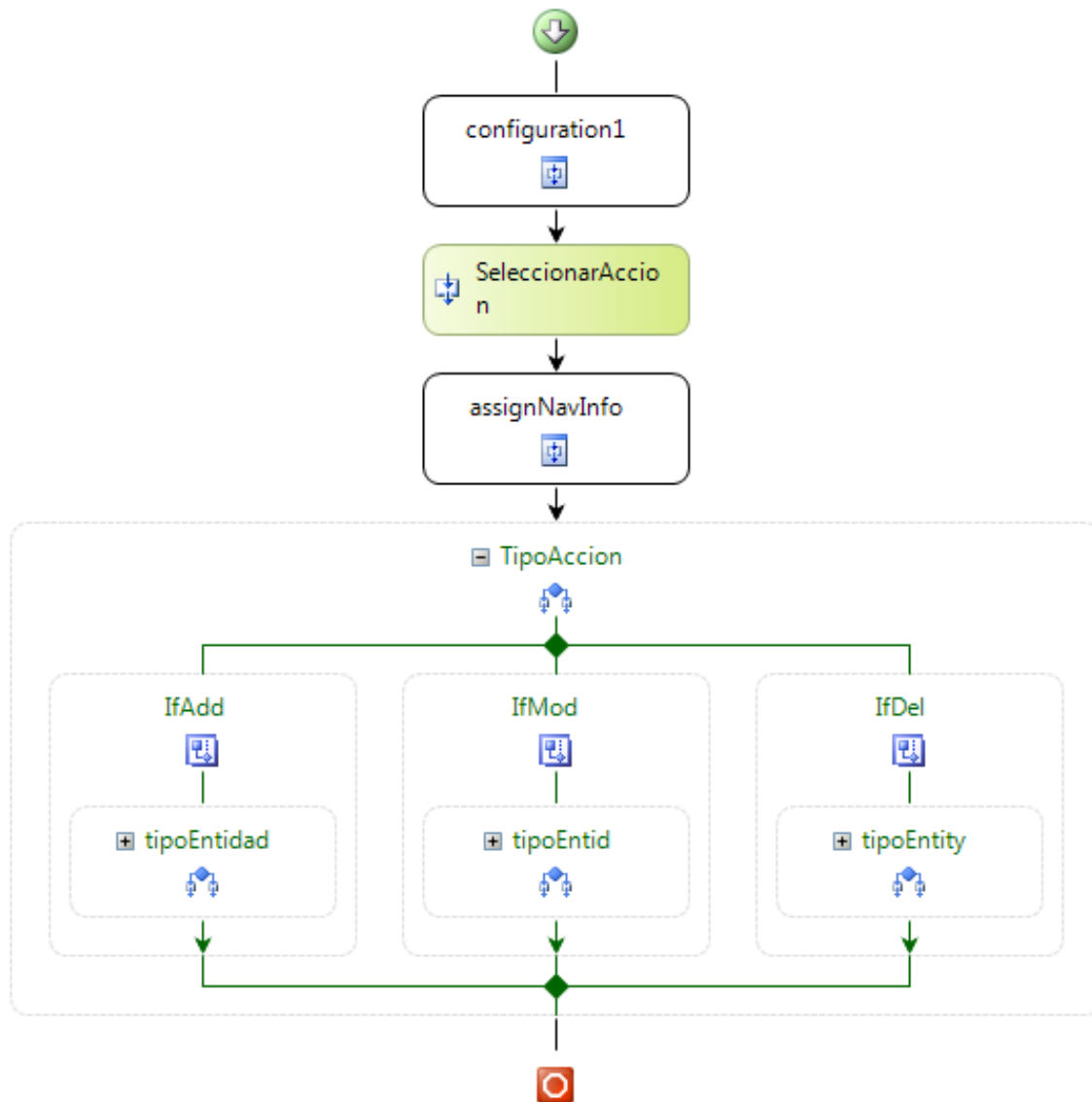


Figura 7 Vista detallada del workflow "ManageEntitiesWf". Fuente: Elaboración propia.

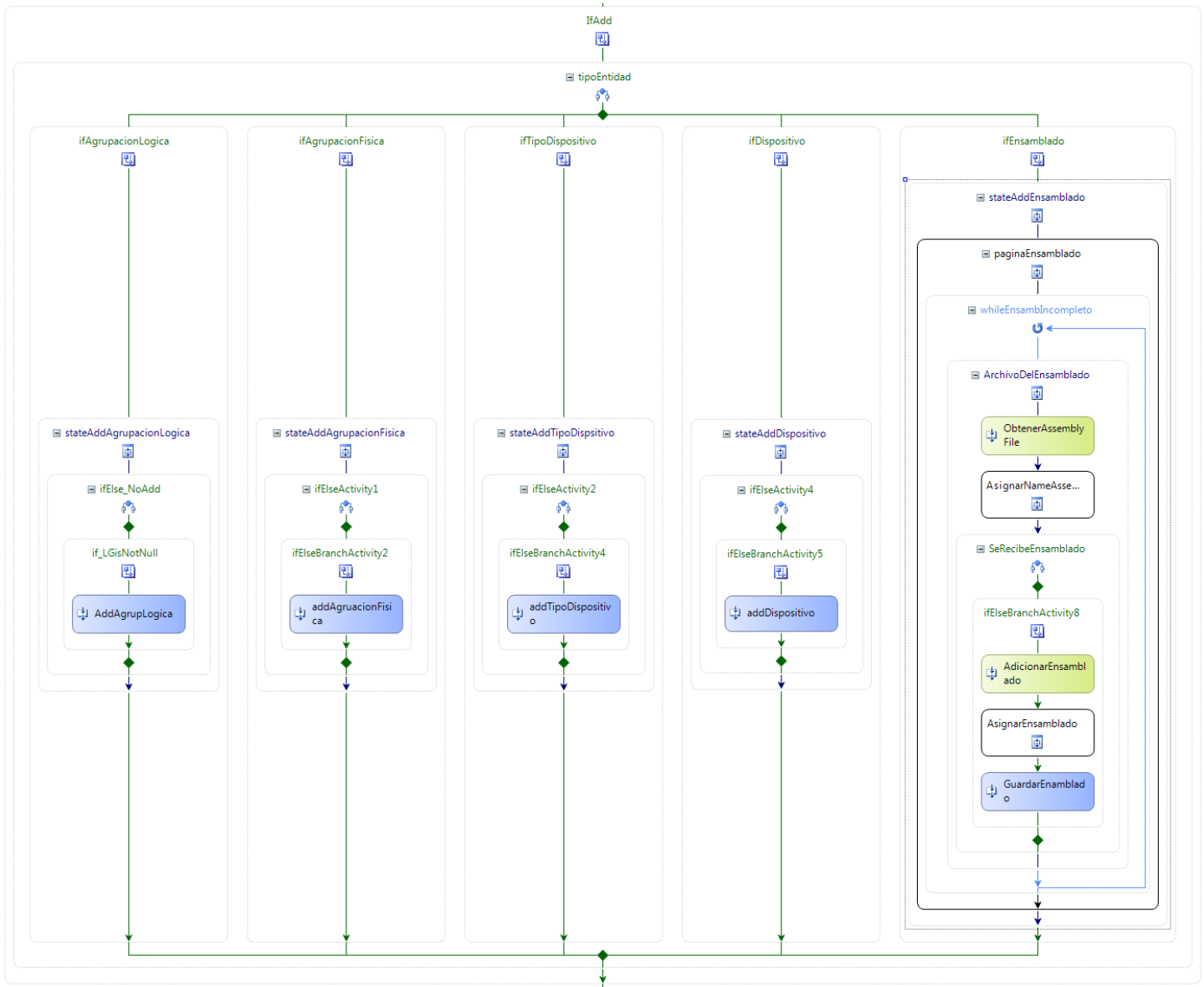


Figura 8 Vista detallada del workflow "ManageEntitiesWf". Insertar entidad. Fuente: Elaboración propia.

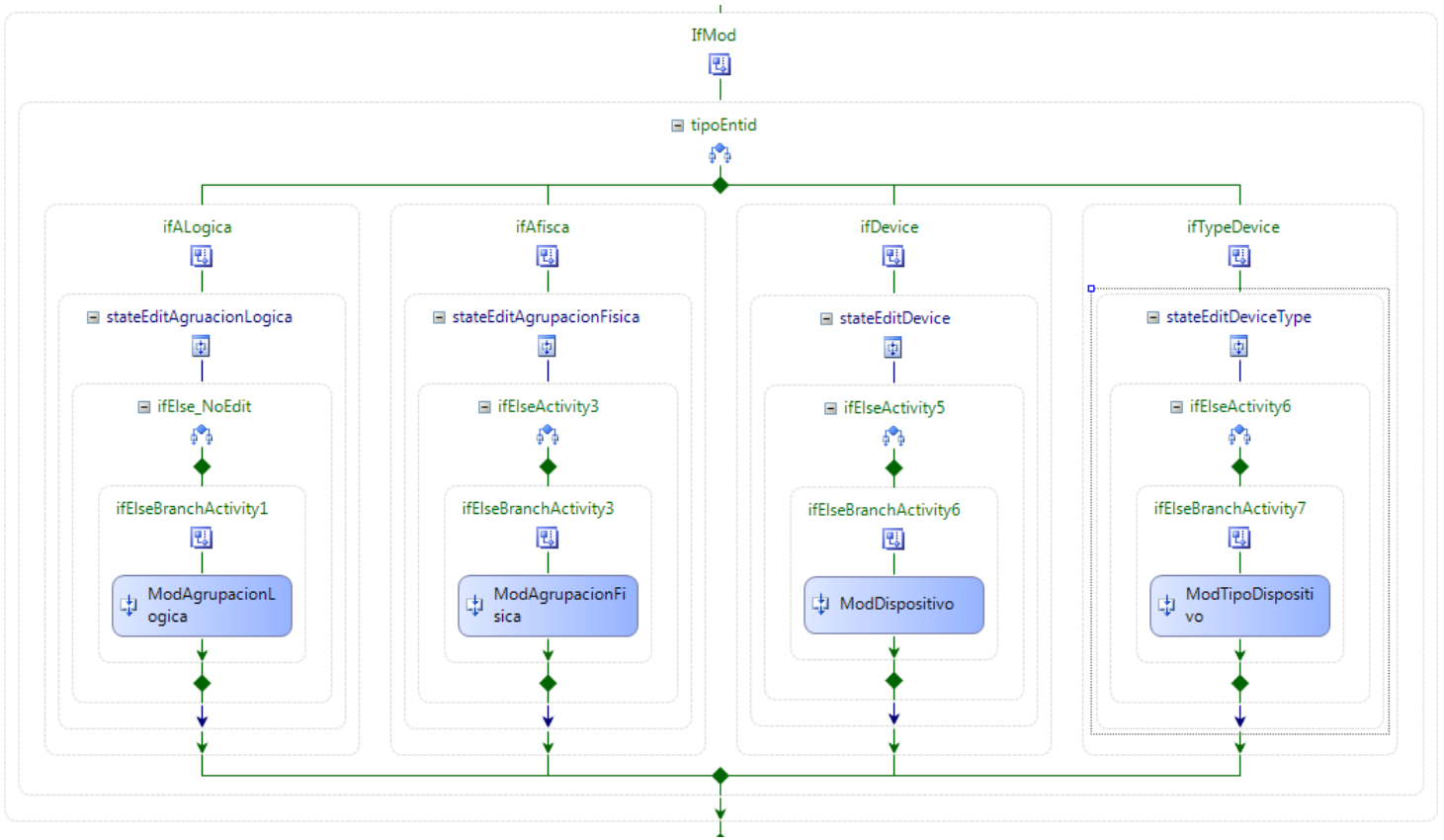


Figura 9 Vista detallada del workflow "ManageEntitiesWf". Modificar entidad. Fuente: Elaboración propia.

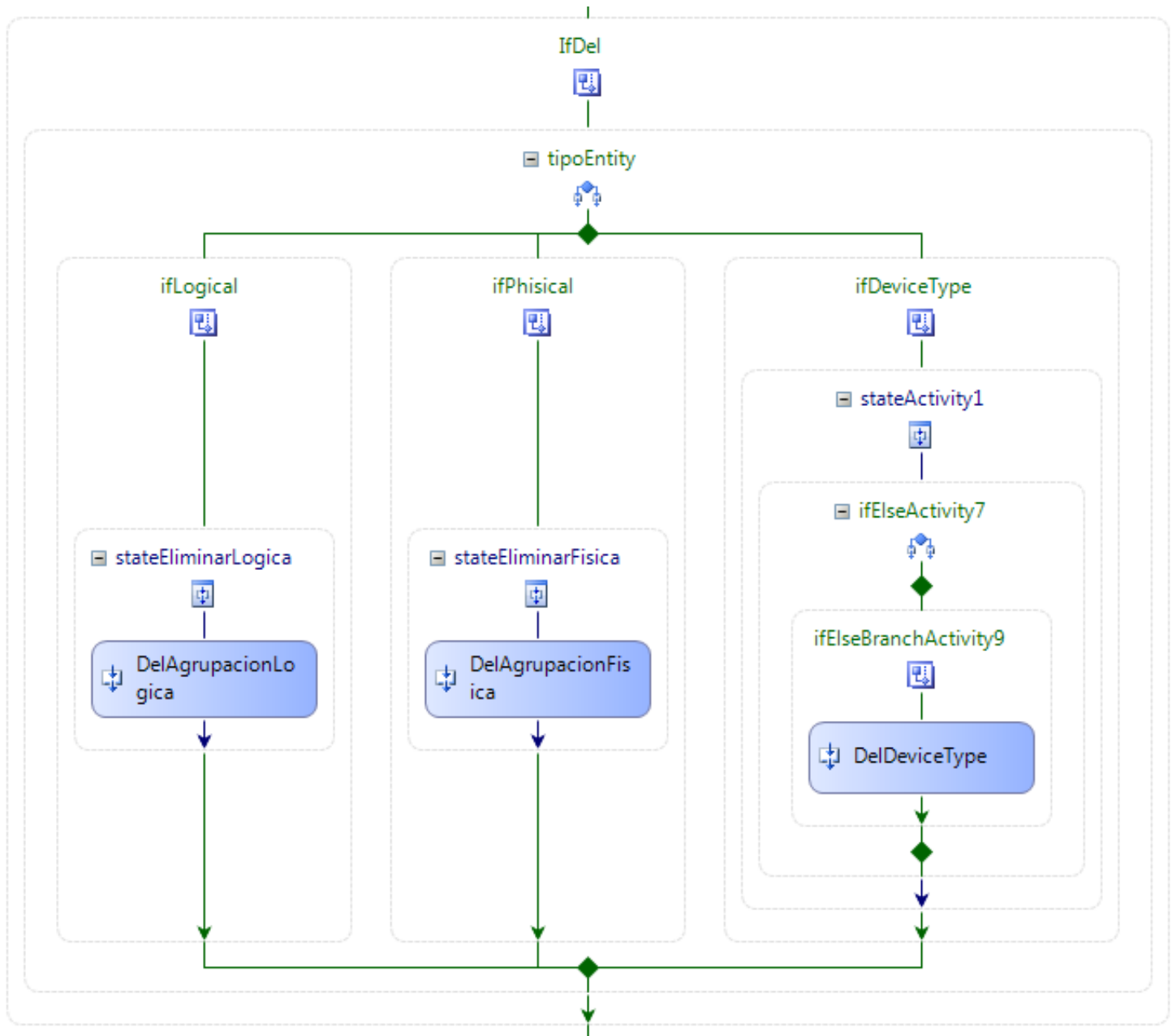


Figura 10 Vista detallada del workflow "ManageEntitiesWf". Eliminar entidad. Fuente: Elaboración propia.

Anexo 7 Descripción del Modelo de datos

Tabla 46 Descripción de la entidad dAgrupacionFisica. Fuente: Elaboración propia.

Entidad	dAgrupacionFisica		
Descripción	Entidad que representa una agrupación física.		
Atributos	Tipo de Dato	Nulo	Definición
idagrupacionfisica	UNIQUEID	N	El identificador de la agrupación física.
idagrupacionpadre	UNIQUEID	S	La asociación de que esta agrupación física tiene como padre la agrupación física especificada.
fecha	DATE	N	La fecha en que se dio de alta la agrupación física.
descripcion	VARCHAR(80)	N	La descripción de la agrupación física.
activo	BIT	N	Si la agrupación física está activa o no.
contenedor	BIT	N	Atributo que representa si la agrupación física es contenedor o no.

Tabla 47 Descripción de la entidad dDispositivo. Fuente: Elaboración propia.

Entidad	dDispositivo		
Descripción	Entidad que representa un dispositivo de hardware.		
Atributos	Tipo de Dato	Nulo	Definición
iddispositivodgm	UNIQUEID	N	El identificador del dispositivo de hardware.
activo	BIT	N	Si el dispositivo está activo o no.
fecha	DATE	N	La fecha en que se dio de alta el dispositivo.
descripcion	VARCHAR(80)	N	La descripción del dispositivo.
contiene	UNIQUEID	S	La asociación de que un dispositivo puede contener otro dispositivo.
idtipodispositivo	NUMERIC(8, 0)	N	El identificador del tipo de dispositivo.
noseriedispositivo	VARCHAR(80)	N	Indica el número de serie o inventario del dispositivo.

Tabla 48 Descripción de la entidad dDispositivo. Fuente: Elaboración propia.

Entidad	dDispositivosLogs		
Descripción	Entidad que representa un log generado por un dispositivo en una estación de trabajo.		
Atributos	Tipo de Dato	Nulo	Definición
iddispositivoslogs	UNIQUEID	N	El identificador del log generado por un dispositivo
idestacion	UNIQUEID	N	El identificador de la estación de trabajo (dispositivo).
iddispositivodgm	UNIQUEID	N	El identificador del dispositivo de hardware.
operacion	VARCHAR(80)	N	Indica la operación que generó el log.
resultadooperacion	BIT	N	Indica si la operación se realizó con éxito o no.
fecharegistro	DATE	N	Fecha en la que se generó el log.
iddispositivoslogs	UNIQUEID	N	El identificador del log generado por un dispositivo

Tabla 49 Descripción de la entidad dEnsamblado. Fuente: Elaboración propia.

Entidad	dEnsamblado		
Descripción	Entidad que representa un ensamblado.		
Atributos	Tipo de Dato	Nulo	Definición
idensamblado	NUMERIC(8, 0)	N	El identificador del ensamblado.
ensamblado	VARCHAR(255)	N	El nombre del ensamblado.
ensambladocontenedor	NUMERIC(8, 0)	S	Representa la asociación de que este ensamblado es contenedor de otro ensamblado.
version	VARCHAR(20)	N	La versión del ensamblado.
activo	BIT	N	Si el ensamblado está activo o no.
fecha	DATE	N	La fecha en que se dio de alta el ensamblado.

Tabla 50 Descripción de la entidad dEnsambladoControlaDis. Fuente: Elaboración propia.

Entidad	dEnsambladoControlaDis		
Descripción	Entidad que representa la relación entre las tablas dEnsamblado y nTipoDispositivo. Donde un ensamblado controla varios dispositivos.		
Atributos	Tipo de Dato	Nulo	Definición
idensambladocontroladis	UNIQUEID	N	Identificador único de la relación entre los ensamblados y los tipo de dispositivos.
idtipodispositivo	NUMERIC(8, 0)	S	El identificador del tipo de dispositivo.
idensamblado	NUMERIC(8, 0)	S	El identificador del ensamblado.
fecharegistro	DATE	N	Dominio del tipo de dato date para las fechas (día, mes y año)

Tabla 51 Descripción de la entidad dEstacionTrabajo. Fuente: Elaboración propia.

Entidad	dEstacionTrabajo		
Descripción	Entidad que representa un estación de trabajo como un caso particular de dispositivo		
Atributos	Tipo de Dato	Nulo	Definición
iddispositivodgm	UNIQUEID	N	El identificador de la estación de trabajo (dispositivo).
descripcion	VARCHAR(80)	N	La descripción de la estación de trabajo.
fecha	DATE	N	La fecha en que se dio de alta la estación de trabajo.
idagrupacionfisica	UNIQUEID	S	El identificador de la agrupación física a la cual pertenece.
token	VARCHAR(250)	N	El token que identifica la estación de trabajo.
activo	BIT	N	Si la estación de trabajo está activa o no.
especificaciontecnic	LONG VARCHAR	N	Las especificaciones técnicas de la estación de trabajo.

Tabla 52 Descripción de la entidad dEstacionUsaDispositivo. Fuente: Elaboración propia.

Entidad	dEstacionUsaDispositivo		
Descripción	Entidad que representa la asociación de que una estación de trabajo (que a la vez es un dispositivo) usa otro dispositivo.		

Atributos	Tipo de Dato	Nulo	Definición
idestacion	UNIQUEID	N	El identificador de la estación de trabajo.
iddispositivodgm	UNIQUEID	N	El identificador del dispositivo usado.
fecha	DATE	N	La fecha en que se crea la asociación.
activo	BIT	N	Si esta activa o no la relación de uso.

Tabla 53 Descripción de la entidad dEstTrabAgrupLogica. Fuente: Elaboración propia.

Entidad	dEstTrabAgrupLogica		
Descripción	Entidad que representa una activación entre una estación de trabajo (dispositivo) y una agrupación lógica.		
Atributos	Tipo de Dato	Nulo	Definición
idesttrabaagruplogica	UNIQUEID	N	El identificador de la activación.
iddispositivodgm	UNIQUEID	N	El identificador de la estación de trabajo (dispositivo).
idagrupacionlogica	NUMERIC(8, 0)	N	El identificador de la agrupación lógica.
activo	BIT	N	Si la activación esta activa.
fecha	DATE	N	La fecha en que se hizo la activación.

Tabla 54 Descripción de la entidad dReporteProblema. Fuente: Elaboración propia.

Entidad	dReporteProblema		
Descripción	Entidad que representa un reporte de un problema encontrado en el sistema.		
Atributos	Tipo de Dato	Nulo	Definición
idreporteproblema	UNIQUEID	N	Identificador único de los problemas que surgen en el DGM.
iddispositivodgm	UNIQUEID	N	El identificador de la estación de trabajo (dispositivo).
fecharegistro	DATE	N	Fecha de registro del problema.
descripcionproblema	VARCHAR(3000)	S	Indica la descripción del problema en cuestión.
asuntoproblema	VARCHAR(3000)	S	Indica el asunto del problema.
problemaatendido	BIT	N	Indica si el problema fue atendido.
fechaatencion	DATE	S	Dominio del tipo de dato date para las fechas (día, mes y año).

Tabla 55 Descripción de la entidad dServicioDGM. Fuente: Elaboración propia.

Entidad	dServicioDGM		
Descripción	Entidad que representa un servicio.		
Atributos	Tipo de Dato	Nulo	Definición
idserviciodgm	NUMERIC(8, 0)	N	El identificador del servicio.
nombre	VARCHAR(80)	N	El nombre del servicio.
descripcion	VARCHAR(80)	N	La descripción del servicio.
idtiposervicio	NUMERIC(8, 0)	N	El identificador de un tipo de servicio.
version	VARCHAR(20)	N	La versión del servicio.
idensamblado	NUMERIC(8, 0)	N	El identificador del ensamblado asociado a ese servicio.

fecha	DATE	N	La fecha en que se dio de alta el servicio.
-------	------	---	---

Tabla 56 Descripción de la entidad dServicioDGmEstacTrab. Fuente: Elaboración propia.

Entidad	dServicioDGmEstacTrab		
Descripción	Entidad que representa que un servicio está asociado a una estación de trabajo (dispositivo).		
Atributos	Tipo de Dato	Nulo	Definición
idserviciodgm	NUMERIC(8, 0)	N	El identificador del servicio.
iddispositivodgm	UNIQUEID	N	El identificador de la estación de trabajo (dispositivo).
activo	BIT	N	Si esta activa o no la asociación.
fecha	DATE	N	La fecha en que se realizó la asociación.

Tabla 57 Descripción de la entidad nAgrupacionActivacion. Fuente: Elaboración propia.

Entidad	nAgrupacionActivacion		
Descripción	Entidad que representa un serial necesario para realizar una activación en una agrupación lógica.		
Atributos	Tipo de Dato	Nulo	Definición
idagrupacionserial	UNIQUEID	N	Identificador del serial necesario para la activación
fecharegistro	DATE	N	Fecha en que se generó el serial
idagrupacionlogica	NUMERIC(8, 0)	N	Dominio del tipo de dato numérico para las secuencias que se generan como identificadores de los nomencladores.
serial	VARCHAR(32)	N	Contiene el serial de la activación
activo	BIT	N	Si está active o no.

Tabla 58 Descripción de la entidad nCategoriaDispositivo. Fuente: Elaboración propia.

Entidad	nCategoriaDispositivo		
Descripción	Entidad que representa una categoría para un dispositivo.		
Atributos	Tipo de Dato	Nulo	Definición
idcategoriadispositivo	NUMERIC(8, 0)	N	El identificador de una categoría de dispositivo
Descripción	VARCHAR(80)	N	La descripción de la categoría de dispositivo.

Tabla 59 Descripción de la entidad nMarcaDispositivo. Fuente: Elaboración propia.

Entidad	nMarcaDispositivo		
Descripción	Entidad que representa una marca para un dispositivo.		
Atributos	Tipo de Dato	Nulo	Definición
idmarcadispositivo	NUMERIC(8, 0)	N	El identificador de la marca de dispositivo.
Descripción	VARCHAR(80)	N	La descripción de la marca de dispositivo.

Tabla 60 Descripción de la entidad nTipoAgrupacionLogica. Fuente: Elaboración propia.

Entidad	nTipoAgrupacionLogica		
Descripción	Entidad que representa los tipos de agrupación lógica.		
Atributos	Tipo de Dato	Nulo	Definición
idtipoagrupacionlogica	NUMERIC(8, 0)	N	Dominio del tipo de dato numérico para las secuencias que se generan como identificadores de los nomencladores.
descripcion	VARCHAR(80)	N	Descripción del nomenclador.

Tabla 61 Descripción de la entidad nTipoDispositivo. Fuente: Elaboración propia.

Entidad	nTipoDispositivo		
Descripción	Entidad que representa un tipo de dispositivo.		
Atributos	Tipo de Dato	Nulo	Definición
idtipodispositivo	NUMERIC(8, 0)	N	El identificador del tipo de dispositivo.
fecha	DATE	S	La fecha en que se da de alta al tipo de dispositivo
Descripción	VARCHAR(80)	S	La descripción del tipo de dispositivo.
modelo	VARCHAR(80)	N	El modelo del dispositivo.
idcategoriadispositivo	NUMERIC(8, 0)	S	El identificador de la categoría de dispositivo.
idmarcadispositivo	NUMERIC(8, 0)	S	La marca del dispositivo

Tabla 62 Descripción de la entidad nTipoServicio. Fuente: Elaboración propia.

Entidad	nTipoServicio		
Descripción	Entidad que representa el tipo de un servicio.		
Atributos	Tipo de Dato	Nulo	Definición
idtiposervicio	NUMERIC(8, 0)	N	El identificador de un tipo de servicio.
Descripción	VARCHAR(80)	N	La descripción de un tipo de servicio.

Anexo 8 Casos de prueba

Tabla 63 Caso de prueba para la funcionalidad Gestionar agrupaciones lógicas. Fuente: Elaboración propia.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
2. Gestionar agrupaciones lógicas.	2.1 Agregar agrupación lógica correctamente.	El sistema agrega la agrupación lógica.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Adicionar 4. Aceptar

2.2	Agregar agrupación lógica con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Adicionar 4. Aceptar
2.3	Agregar agrupación lógica con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Adicionar 4. Aceptar
2.4	Modificar agrupación lógica correctamente.	El sistema modifica la agrupación lógica.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Modificar 4. Aceptar 5. Si
2.5	Modificar agrupación lógica con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Modificar 4. Aceptar
2.6	Modificar agrupación lógica con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Modificar 4. Aceptar
2.7	Modificar agrupación lógica sin haber seleccionado la agrupación a modificar.	El sistema muestra un mensaje indicando que debe seleccionar una agrupación.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Modificar 4. Aceptar
2.8	Eliminar agrupación lógica correctamente.	El sistema elimina la agrupación lógica.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Eliminar 4. Aceptar 5. Si
2.9	Eliminar agrupación lógica sin haber seleccionado la agrupación a eliminar.	El sistema muestra un mensaje indicando que debe seleccionar una agrupación.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Eliminar 4. Aceptar
2.10	Cancelar la gestión de agrupaciones lógicas	El sistema redirecciona a la página principal.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación lógica 3. Cancelar

Tabla 64 Caso de prueba para la funcionalidad Gestionar agrupaciones físicas. Fuente: Elaboración propia.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
3. Gestionar agrupaciones físicas.	3.1 Agregar agrupación física correctamente.	El sistema agrega la agrupación física.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Adicionar 4. Aceptar
	3.2 Agregar agrupación física con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Adicionar 4. Aceptar
	3.3 Agregar agrupación física con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Adicionar 4. Aceptar
	3.4 Modificar agrupación física correctamente.	El sistema modifica la agrupación física.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Modificar 4. Aceptar 5. Si
	3.5 Modificar agrupación física con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Modificar 4. Aceptar
	3.6 Modificar agrupación física con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Modificar 4. Aceptar
	3.7 Modificar agrupación física sin haber seleccionado la agrupación a modificar.	El sistema muestra un mensaje indicando que debe seleccionar una agrupación.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Modificar 4. Aceptar
	3.8 Eliminar agrupación física correctamente.	El sistema elimina la agrupación física.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Eliminar 4. Aceptar 5. Si

	3.9 Eliminar agrupación física sin haber seleccionado la agrupación a eliminar.	El sistema muestra un mensaje indicando que debe seleccionar una agrupación.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Eliminar 4. Aceptar
	3.10 Cancelar la gestión de agrupaciones físicas.	El sistema redirecciona a la página principal.	<ol style="list-style-type: none"> 1. Gestionar 2. Agrupación física 3. Cancelar

Tabla 65 Caso de prueba para la funcionalidad Gestionar tipos de dispositivos. Fuente: Elaboración propia.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
4. Gestionar tipos de dispositivos.	4.1 Agregar tipo de dispositivo correctamente.	El sistema agrega el tipo de dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Adicionar 4. Aceptar
	4.2 Agregar tipo de dispositivo con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Adicionar 4. Aceptar
	4.3 Agregar tipo de dispositivo con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Adicionar 4. Aceptar
	4.4 Modificar tipo de dispositivo correctamente.	El sistema modifica la el tipo de dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Modificar 4. Aceptar 5. Si
	4.5 Modificar tipo de dispositivo con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Modificar 4. Aceptar
	4.6 Modificar tipo de dispositivo con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Modificar 4. Aceptar
	4.7 Modificar tipo de dispositivo sin haber seleccionado el	El sistema muestra un mensaje indicando que debe seleccionar un tipo de dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Modificar 4. Aceptar

	tipo de dispositivo a modificar.		
	4.8 Eliminar tipo de dispositivo correctamente.	El sistema elimina tipo de dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Eliminar 4. Aceptar 5. Si
	4.9 Eliminar tipo de dispositivo sin haber seleccionado tipo de dispositivo a eliminar.	El sistema muestra un mensaje indicando que debe seleccionar un tipo de dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Eliminar 4. Aceptar
	4.10 Cancelar la gestión de tipos de dispositivos.	El sistema redirecciona a la página principal.	<ol style="list-style-type: none"> 1. Gestionar 2. Tipo de dispositivo 3. Cancelar

Tabla 66 Caso de prueba para la funcionalidad Gestionar dispositivos. Fuente: Elaboración propia.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
5. Gestionar dispositivos.	5.1 Agregar dispositivo correctamente.	El sistema agrega el dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Adicionar 4. Aceptar
	5.2 Agregar dispositivo con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Adicionar 4. Aceptar
	5.3 Agregar dispositivo con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Adicionar 4. Aceptar
	5.4 Modificar dispositivo correctamente.	El sistema modifica la el dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Modificar 4. Aceptar 5. Si

	5.5 Modificar dispositivo con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Modificar 4. Aceptar
	5.6 Modificar dispositivo con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Modificar 4. Aceptar
	5.7 Modificar dispositivo sin haber seleccionado el dispositivo a modificar.	El sistema muestra un mensaje indicando que debe seleccionar un dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Modificar 4. Aceptar
	5.8 Eliminar dispositivo correctamente.	El sistema elimina dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Eliminar 4. Aceptar 5. Si
	5.9 Eliminar dispositivo sin haber seleccionado dispositivo a eliminar.	El sistema muestra un mensaje indicando que debe seleccionar un dispositivo.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Eliminar 4. Aceptar
	5.10 Cancelar la gestión de tipos de dispositivos.	El sistema redirecciona a la página principal.	<ol style="list-style-type: none"> 1. Gestionar 2. Dispositivo 3. Cancelar

Tabla 67 Caso de prueba para la funcionalidad Gestionar ensamblados. Fuente: Elaboración propia.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
6. Gestionar ensamblados.	6.1 Agregar ensamblado correctamente.	El sistema agrega el ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Agregar 4. Examinar 5. Subir 6. Aceptar
	6.2 Agregar ensamblado sin haber subido el ensamblado al servidor.	El sistema muestra un mensaje de alerta indicando que no ha subido un ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Agregar 4. Aceptar

6.3	Agregar ensamblado con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Agregar 4. Examinar 5. Subir 6. Aceptar
6.4	Agregar ensamblado sin haber seleccionado el tipo de dispositivo asociado.	El sistema muestra un mensaje de alerta indicando debe seleccionar un tipo de dispositivo para asociarlo al ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Agregar 4. Examinar 5. Subir 6. Aceptar
6.5	Modificar ensamblado correctamente.	El sistema modifica el ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Modificar 4. Aceptar
6.6	Modificar ensamblado con campos incorrectos.	El sistema muestra un mensaje de alerta indicando que existe un campo incorrecto.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Modificar 4. Aceptar
6.7	Modificar ensamblado con campos vacíos.	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Modificar 4. Aceptar
6.8	Modificar ensamblado sin haber seleccionado el ensamblado a modificar.	El sistema muestra un mensaje indicando que debe seleccionar un ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Modificar 4. Aceptar
6.9	Eliminar ensamblado correctamente.	El sistema elimina el ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Eliminar 4. Aceptar 5. Si
6.10	Eliminar ensamblado sin haber seleccionado el ensamblado a eliminar.	El sistema muestra un mensaje indicando que debe seleccionar un ensamblado.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Eliminar 4. Aceptar
6.11	Cancelar la gestión de ensamblado.	El sistema redirecciona a la página principal.	<ol style="list-style-type: none"> 1. Gestionar 2. Ensamblado 3. Cancelar

Tabla 68 Iteraciones. Fuente: Elaboración propia.

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
2.1	Clases Válidas	Nombre: "Camagüey 2" Tipo de agrupación lógica: "Oficina"	Se agrega una nueva agrupación lógica.	Se agrega una nueva agrupación lógica.	
2.3	Clases Inválidas	Nombre: "" Tipo de agrupación lógica: "Oficina"	Se muestra un mensaje que indica que el campo está vacío.	Se muestra un mensaje que indica que el campo está vacío.	
2.4	Clases Válidas	Nombre: "Camagüey 2" Tipo de agrupación lógica: "Impresión"	Se modifica la agrupación lógica.	Se modifica la agrupación lógica	
2.8	Clases Válidas	Agrupación lógica: "Camaguey 2"	Se elimina la agrupación lógica.	Se elimina la agrupación lógica.	
3.1	Clases Válidas	Nombre: "Esmeralda" Contenedor: "SI"	Se agrega una nueva agrupación física.	Se agrega una nueva agrupación física.	
3.3		Nombre: "" Contenedor: ""	Se muestra un mensaje indicando que debe especificar el nombre de la agrupación física.	Se muestra un mensaje indicando que debe especificar el nombre de la agrupación física.	
3.4	Clases Válidas	Nombre: "Esmeralda" Contenedor: "NO"	Se muestra un mensaje indicando que la modificación ha sido satisfactoria.	Se muestra un mensaje indicando que la modificación ha sido satisfactoria.	
3.5	Clases Inválidas	Nombre: "" Contenedor: "NO"	El sistema muestra un mensaje de alerta indicando que existe un campo vacío	El sistema muestra un mensaje de alerta indicando que existe un campo vacío.	

3.8		Agrupación física: “Esmeralda”	El sistema muestra un mensaje indicando que se eliminó correctamente la agrupación física.	El sistema muestra un mensaje indicando que se eliminó correctamente la agrupación física.	
3.9	Clases Inválidas	Agrupación física: “”	El sistema muestra un mensaje indicando que debe seleccionarse la agrupación física a eliminar.	El sistema muestra un mensaje indicando que debe seleccionarse la agrupación física a eliminar.	
4.1	Clases Válidas	Descripción: “No hay descripción” Categoría: “PC” Marca: “HP” Modelo: “Pavilion dm4”	El sistema muestra un mensaje indicando que se ha adicionado el tipo de dispositivo correctamente.	El sistema muestra un mensaje indicando que se ha adicionado el tipo de dispositivo correctamente.	
4.3	Clases Inválidas	Descripción: “No hay descripción” Categoría: “” Marca: “HP” Modelo: “Pavilion dm4”	El sistema muestra un mensaje indicando que hay campos vacíos.	El sistema muestra un mensaje indicando que hay campos vacíos.	
4.4	Clases Válidas	Descripción: “Nueva descripción” Categoría: “PC” Marca: “HP” Modelo: “Pavilion dm4”	El sistema muestra un mensaje indicando que se ha modificado el tipo de dispositivo correctamente.	El sistema muestra un mensaje indicando que se ha modificado el tipo de dispositivo correctamente.	
4.7	Clases Inválidas	Tipo de dispositivo: “”	El sistema muestra un mensaje indicando que debe seleccionarse un tipo de dispositivo.	El sistema muestra un mensaje indicando que debe seleccionarse un tipo de dispositivo.	

4.8	Clases Válidas	Tipo de dispositivo: "Haier Core i7"	El sistema elimina tipo de dispositivo.	El sistema elimina tipo de dispositivo.	
4.9	Clases Inválidas	Tipo de dispositivo: ""	El sistema muestra un mensaje indicando que debe seleccionar un tipo de dispositivo.	El sistema muestra un mensaje indicando que debe seleccionar un tipo de dispositivo.	
5.1	Clases Válidas	Descripción: "Digital Person MININT" No Serie: "67736202" Tipo de dispositivo: "Lector de Huellas GreenBit ..."	El sistema agrega el dispositivo.	El sistema agrega el dispositivo.	
5.3	Clases Inválidas	Descripción: "Digital Person MININT" No Serie: "67736202" Tipo de dispositivo: ""	El sistema muestra un mensaje de alerta indicando que se hay un campo vacío.	El sistema muestra un mensaje de alerta indicando que se hay un campo vacío.	
5.4	Clases Válidas	Descripción: "Digital Person MININT" No Serie: "67736203" Tipo de dispositivo: "Lector de Huellas GreenBit ..."	Se modifica correctamente el dispositivo.	Se modifica correctamente el dispositivo.	
5.6	Clases Inválidas	Descripción: "" No Serie: "67736203" Tipo de dispositivo: "Lector de Huellas GreenBit ..."	El sistema muestra un mensaje de alerta indicando que se hay un campo vacío.	El sistema muestra un mensaje de alerta indicando que se hay un campo vacío.	
5.8	Clases Válidas	Dispositivo: "Digital Person MININT" No Serie: "67736202" Tipo de	Se elimina correctamente el dispositivo.	Se elimina correctamente el dispositivo.	

		dispositivo: "Lector de Huellas GreenBit ..."			
5.9	Clases Inválidas	Dispositivo: ""	El sistema muestra un mensaje indicando que debe seleccionar un dispositivo.	El sistema muestra un mensaje indicando que debe seleccionar un dispositivo.	
6.1	Clases Válidas	Ensamblado: "[Fichero]" Tipo de dispositivo que controla: "Scanner ScanJet 4770"	El sistema agrega el ensamblado.	El sistema agrega el ensamblado.	
6.2	Clases Inválidas	Ensamblado: Tipo de dispositivo que controla:	El sistema muestra un mensaje de alerta indicando que no ha subido un ensamblado.	El sistema muestra un mensaje de alerta indicando que no ha subido un ensamblado.	
6.4	Clases Inválidas	Ensamblado: "[Fichero]" Tipo de dispositivo que controla:	El sistema muestra un mensaje de alerta indicando que no ha seleccionado el tipo de dispositivo asociado al ensamblado.	El sistema muestra un mensaje de alerta indicando que no ha seleccionado el tipo de dispositivo asociado al ensamblado.	
6.9	Clases Válidas	Ensamblado: "ScanJet 4770.dll" Tipo de dispositivo que controla: "Scanner ScanJet 4770"	El sistema elimina el ensamblado seleccionado.	El sistema elimina el ensamblado seleccionado.	
6.10	Clases Inválidas	Ensamblado: ""	El sistema muestra un mensaje indicando que debe seleccionar un ensamblado para poder	El sistema muestra un mensaje indicando que debe seleccionar un ensamblado para poder	

			eliminarlo.	eliminarlo.	
--	--	--	-------------	-------------	--

Tabla 69 Resultado de las pruebas de la segunda iteración. Fuente: Elaboración propia.

Iteración 2					
Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección del error	Importancia
	1	Los resultados arrojados durante el proceso de verificación de la calidad del rostro no son los correctos.		Pruebas	Alta
	2	Cuando se realiza el proceso de captura de huellas con el dispositivo DactyScan 40 el tiempo de respuesta es más alto de lo esperado.		Pruebas	Alta