

**PROTOTIPO DE APLICACIÓN DE TELEMEDICINA BASADO EN EL  
ESTÁNDAR DVB-MHP PARA LA TELEVISION DIGITAL TERRESTRE  
INTERACTIVA**

**AUTORES:**

LINA MARÍA GARCÉS RODRÍGUEZ

DEIVER ARLEY ROVIRA SÁNCHEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

Bucaramanga 2009

**PROTOTIPO DE APLICACIÓN DE TELEMEDICINA BASADO EN EL  
ESTÁNDAR DVB-MHP PARA LA TELEVISION DIGITAL TERRESTRE  
INTERACTIVA**

**AUTORES:**

LINA MARÍA GARCÉS RODRÍGUEZ

DEIVER ARLEY ROVIRA SÁNCHEZ

Trabajo de grado para obtener el título de Ingeniero de Sistemas

**DIRECTOR:**

Ph.D SERGIO CASTILLO CASTELBLANCO

PROFESOR ESCUELA DE INGENIERIA DE SISTEMAS E  
INFORMATICA - UIS

**CODIRECTOR:**

Msc. VICTOR EDUARDO MARTÍNEZ ABAUNZA

PROFESOR ESCUELA DE INGENIERIA DE SISTEMAS E  
INFORMATICA - UIS

MODALIDAD DE TRABAJO DE INVESTIGACIÓN

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

Bucaramanga 2009

*A todos aquellos que me han formado.*

*Especialmente a mi madre querida quien ha  
sido el motivo especial para esforzarme  
y lograr grandes cosas.*

*A mis profesores, familiares, compañeros y amigos,  
de los cuales he aprendido muchas cosas.*

*A mis hermanos quienes han sido de gran apoyo durante toda mi vida,  
y quienes siempre han creído en mí.*

*A mi novia, quien ha sido mi felicidad durante todo este tiempo.*

*A mi compañera de trabajo con la cual compartí gratos momentos.*

*A mis compañeros universitarios con los que compartí los mejores momentos.*

*A Jesucristo,  
quien ha sido el guía durante toda mi vida y quien hace que esto sea hoy posible.*

**DEIVER ROVIRA SANCHEZ**

*A mis padres y hermanas quienes me han dado su apoyo durante toda mi vida y por los cuales seguiré dando mi mayor esfuerzo.*

*A mi familia quien ha estado presente para ayudarme en momentos difíciles.*

*A mi mejor amiga, Eliana, por su paciencia y comprensión.*

*A mis amigos de la Universidad con los que he pasado momentos inolvidables y me han enseñado el valor del trabajo en equipo.*

*A mi compañero de proyecto y amigo, por su alegría e infinita paciencia.*

*A mis profesores de Colegio y Universidad que me han aportado los conocimientos necesarios para desempeñarme profesionalmente.*

*A mi novio y mejor amigo, Javier, quien ha sido un gran apoyo en muchos momentos de mi vida y me ha brindado su cariño incondicional.*

*Gracias, Totales!*

**LINA MARÍA GARCÉS RODRÍGUEZ**

## AGRADECIMIENTOS

Al Profesor Sergio Castillo Castelblanco, director de proyecto. Quien nos guió y enseñó muchos de sus conocimientos en el área, y quien confió en nosotros para el desarrollo de este proyecto.

Al Profesor Víctor Eduardo Martínez, co-director del proyecto, por la experiencia transmitida en el campo de la Televisión Digital.

Al Ingeniero Javier Noguera, por su aporte de ideas y experiencias vividas en el desarrollo de software.

A los profesores de la Escuela de Ingeniería de Sistemas e Informática de la Universidad por todos los conocimientos transmitidos a lo largo de toda nuestra vida universitaria.

A Johannes Schmid, Gerente general de MIT-xperts, por facilitarnos la versión de prueba del iDesigner Professional, el cual fue utilizado para el desarrollo del prototipo.

Al profesor de la Universidad de Málaga España Dr. David Bueno Vallejo, por creer en nosotros y ayudarnos con conocimientos claves para el desarrollo de nuestro proyecto.

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	1
1. DESCRIPCIÓN DEL PROYECTO .....	2
OBJETIVO GENERAL .....	3
OBJETIVOS ESPECÍFICOS .....	3
METODOLOGÍA.....	5
XP COMO METODOLOGÍA .....	5
Fases de XP: .....	6
Ventajas.....	6
ORGANIZACIÓN DEL LIBRO .....	7
2. MARCO TEÓRICO .....	9
2.1. Transición de la televisión analógica a la televisión digital .....	9
2.2. Televisión Digital Interactiva (TVDi) (Multimedia Home Platforms, 2006) .....	10
2.2.1. Definición .....	10
2.2.2. Aplicaciones .....	10
2.3. Estándares de transmisión de Televisión Digital Terrestre (TDT) 11	
2.3.1. ATSC (Advanced Television System Committee) .....	11
2.3.2. ISDB-T (Integrated Service Digital Broadcasting Terrestrial) 12	
2.3.3. DVB-T (Digital Video Broadcasting Terrestrial) .....	12
2.4. Estándares middleware para televisión digital interactiva.....	13
2.4.1. ACAP .....	13
2.4.2. MHP .....	13
2.4.3. Java TV .....	14
2.4.4. GEM (Globally Executable MHP) (DVB, 2009) .....	18

2.5. MHP (Multimedia Home Platforms) (Multimedia Home Platforms, 2006) .....	19
2.5.1. Introducción.....	19
2.5.2. Arquitectura de MHP .....	22
2.6. Canal de Retorno.....	27
2.7. DSM-CC (Digital Storage Media Command and Control) .....	28
2.8. MPEG-2.....	29
2.9. Aplicaciones MHP.....	31
2.9.1. Aplicaciones Service bounds .....	31
2.9.2. Aplicaciones Unbound.....	31
2.9.3. Aplicaciones Almacenadas .....	31
2.9.4. Aplicaciones Nativas .....	32
2.9.5. Aplicaciones Locales.....	32
2.10. Introducción a los XLETS .....	33
3. ANÁLISIS COMPARATIVO DE LOS ENTORNOS DE DESARROLLO INTEGRADOS EXISTENTES PARA DESARROLLAR APLICACIONES MHP .....	37
3.1. Definición .....	37
3.2. IDE's Existentes para desarrollar aplicaciones para Televisión Digital .....	38
3.2.1. Herramientas de Desarrollo Privadas.....	38
3.2.2. Herramientas de Desarrollo Libres.....	39
3.3. Evaluación de IDE's (Lawrence Pfleeger).....	41
3.3.1. Técnica de Evaluación: Análisis de Características .....	41
3.3.2. Modelo de calidad ISO 9126 .....	42
3.3.3. Diagrama comparativo de los IDE's seleccionados.....	46
3.3.4. Resultado de la Evaluación.....	48
4. ANÁLISIS Y DISEÑO DEL PROTOTIPO.....	48
4.1. Especificación de Requisitos .....	48

Requerimientos funcionales del prototipo mhp .....	48
Requerimientos no funcionales del prototipo mhp .....	50
Requerimientos funcionales de la aplicación web .....	51
Requerimientos no funcionales de la aplicación web .....	51
4.2. Lógica de la Aplicación .....	52
4.3. Casos de Uso .....	52
4.4. Diagramas de Secuencia.....	57
4.4.1. Aplicación Web.....	57
4.4.2. Servidor .....	60
4.4.3. Aplicación Mhp .....	61
4.5. Diagrama de Clases .....	68
4.6. Modelo Entidad Relación .....	69
4.7. Distribución del Proyecto .....	70
5. DESARROLLO DEL PROTOTIPO DE APLICACIÓN MHP .....	76
5.1. Introducción .....	76
5.2. Software Genérico .....	76
5.2.1. Servlet (Ceballos, 2006).....	76
5.2.2. JSP (Java Server Pages).....	77
5.2.3. Postgresql .....	78
5.2.4. Tomcat .....	80
5.2.5. Netbeans.....	80
5.2.6. iDesigner .....	81
5.3. Desarrollo de Aplicaciones MHP con Netbeans y “XletView” ..	82
5.3.1. API’s necesarias para desarrollar aplicaciones MHP .....	82
5.3.1.1. API’s de Núcleo .....	82
5.3.1.2. API’s de Java Media Framework .....	84
5.3.1.3. API’s de Java TV .....	84
5.3.1.4. API’s de DAVIC (Digital Audio Visual Council).....	86

5.3.1.5.	API's de HAVI (Home Audio Video Interoperability).....	87
5.3.1.6.	API's de DVB .....	87
5.3.2.	J2ME: Configuración y Perfiles de Java para las aplicaciones MHP .....	90
5.3.3.	Herramientas Software necesarias .....	93
5.3.4.	Ejemplo "Hola Mundo" con Netbeans y XletView .....	93
5.4.	Desarrollo de Aplicaciones MHP con iDesigner.....	102
5.4.1.	Ejemplo "Hola Mundo" con iDesigner.....	102
5.5.	Prototipo de Aplicación MHP con iDesigner .....	108
6.	PRUEBAS.....	129
7.	CONCLUSIONES Y RECOMENDACIONES.....	136
8.	GLOSARIO .....	138
9.	REFERENCIAS BIBLIOGRÁFICAS .....	142

## LISTA DE FIGURAS

Ilustración 1. Entorno hardware y software de un receptor de tv digital (microsystems, 2009).....	16
Ilustración 2 Contexto mhp. (MARTÍNEZ P, 2008) .....	23
Ilustración 3 Interfaces externas entre mhp y el mundo exterior. (MARTÍNEZ P, 2008).....	23
Ilustración 4 Arquitectura de mhp. (MHP, 2009) .....	25
Ilustración 5. Tasa de datos de algunas tecnologías de canal de retorno .....	28
Ilustración 6 Carrusel de objetos.....	29
Ilustración 7 Diagrama de los estados del ciclo de vida de un xlet. (MHP, 2009).....	34
Ilustración 8 Modelo de calidad segun iso 9126 .....	43
Ilustración 9 Lógica de la aplicación .....	52
Ilustración 10 Caso de uso doctor.....	53
Ilustración 11 Casos de uso servidor y aplicacion mhp .....	54
Ilustración 12 Caso de uso aplicacion mhp.....	55
Ilustración 13 Casos de uso paciente .....	56
Ilustración 14 Secuencia: cargar paciente .....	57
Ilustración 15 Secuencia: crear tratamiento.....	57
Ilustración 16 Secuencia: formular examen .....	58
Ilustración 17 Secuencia: ingresar datos paciente.....	58
Ilustración 18 Secuencia: formular medicamento .....	59
Ilustración 19 Secuencia: insertar orden de examen .....	60
Ilustración 20 Secuencia: buscar recordatorios .....	60
Ilustración 21 Secuencia: crear archivo de recordatorios .....	61
Ilustración 22 Secuencia: consultar archivo de recordatorios .....	61
Ilustración 23 Secuencia aceptar y posponer recordatorio .....	62
Ilustración 24 Secuencia: cancelar cita.....	62
Ilustración 25 Secuencia: cancelar examen.....	63
Ilustración 26 Secuencia: insertar citas.....	63
Ilustración 27 Secuencia: modificar doctor cita.....	64
Ilustración 28 Secuencia: modificar doctor examen.....	64

Ilustración 29 Secuencia: modificar fecha examen .....	65
Ilustración 30 Secuencia: modificar fecha cita .....	65
Ilustración 31 Secuencia: modificar hora cita.....	66
Ilustración 32 Secuencia: modificar hora examen.....	66
Ilustración 33 Secuencia: recordar cita .....	67
Ilustración 34 Secuencia: recordar examen.....	67
Ilustración 35 Secuencia: recordar medicamento .....	67
Ilustración 36 Diagrama de Clases .....	68
Ilustración 37 Modelo Entidad Relación .....	69
Ilustración 38 Distribución del Proyecto en Netbeans.....	70
Ilustración 39 Aplicación Web .....	71
Ilustración 40 Servlets del prototipo .....	72
Ilustración 41 Clases del Proyecto.....	73
Ilustración 42 Librerías.....	73
Ilustración 43 Ejemplo del modelo de Codificación.....	74
Ilustración 44 Distribución del proyecto en iDesigner .....	75
Ilustración 45 Páginas de la aplicación MHP .....	75
Ilustración 46 Funcionamiento de un servlet.....	77
Ilustración 47 Relación entre configuraciones, perfiles y paquetes en jme .....	91
Ilustración 48 Relación entre configuración, Perfil y Paquete en un Decodificador u otro hardware multimedia.....	92
Ilustración 49 Crear nuevo proyecto .....	94
Ilustración 50 Selección de Categoría y tipo de proyecto .....	94
Ilustración 51 Carga de Configuración.....	94
Ilustración 52 Agregar Plataforma.....	95
Ilustración 53 Emulador de Plataforma .....	95
Ilustración 54 Seleccionar carpeta de la Plataforma .....	95
Ilustración 55 Verificación de la Plataforma .....	96
Ilustración 56 Fin de la carga de la Plataforma .....	96
Ilustración 57 Ingreso del nombre del Proyecto.....	96
Ilustración 58 Finalizar creación del proyecto .....	97
Ilustración 59 Agregar paquetes de XletView y JavaTv .....	97
Ilustración 60 Selección de carpeta de los paquetes de XletView y JavaTv .....	97
Ilustración 61 Insertar Xlet .....	98
Ilustración 62 Nombre del Xlet.....	98

Ilustración 63 Métodos del ciclo de vida del Xlet.....	99
Ilustración 64 Interfaz de XletView.....	101
Ilustración 65 Administrador de aplicaciones del XletView .....	101
Ilustración 66 Carpeta del Xlet.....	102
Ilustración 67 Xlet ejecutándose en el emulador.....	102
Ilustración 68 Área de trabajo de iDesigner .....	103
Ilustración 69 Herramienta Text.....	103
Ilustración 70 Editor de texto .....	103
Ilustración 71 Vista del texto .....	103
Ilustración 72 Panel de propiedades.....	104
Ilustración 73 Herramienta rectángulo .....	104
Ilustración 74 Rectángulo como botón.....	105
Ilustración 75 Texto en un botón.....	105
Ilustración 76 Agrupación de elementos .....	106
Ilustración 77 Foco en un botón.....	106
Ilustración 78 Scripts de un botón.....	107
Ilustración 79 Ejecución del emulador .....	107
Ilustración 80 Ejemplo en el emulador .....	107
Ilustración 81 Pantalla principal aplicación MHP.....	108
Ilustración 82 Menú principal de la aplicación MHP.....	109
Ilustración 83 Ingreso de usuarios .....	109
Ilustración 84 Menú medicamento .....	110
Ilustración 85 Iniciar medicamento.....	110
Ilustración 86 Menú de Citas .....	111
Ilustración 87 Ingreso de Citas paso 1 .....	111
Ilustración 88 Ingreso de citas paso 2.....	112
Ilustración 89 Ingreso de citas paso 3.....	112
Ilustración 90 Ingreso de citas paso 4.....	113
Ilustración 91 Ingreso de citas paso 5.....	113
Ilustración 92 Ingreso de citas paso 6.....	114
Ilustración 93 Modificación de citas paso 1.....	114
Ilustración 94 Modificación de citas paso 2.....	115
Ilustración 95 Modificación de citas paso 3.....	115
Ilustración 96 Modificación de citas paso 4.....	116
Ilustración 97 Modificación de citas paso 5.....	116
Ilustración 98 Modificación de citas paso 6.....	117
Ilustración 99 Cancelación de cita paso 1.....	117

Ilustración 100 Cancelación de cita paso 2.....	118
Ilustración 101 Cancelación de cita paso 3.....	118
Ilustración 102 Menú de Exámenes.....	119
Ilustración 103 Ingreso de examen paso 1 .....	119
Ilustración 104 Ingreso de examen paso 2 .....	120
Ilustración 105 Ingreso de examen paso 3 .....	120
Ilustración 106 Ingreso de examen paso 4 .....	121
Ilustración 107 Ingreso de examen paso 5 .....	121
Ilustración 108 Ingreso de examen paso 6 .....	122
Ilustración 109 Modificación de examen paso 1 .....	122
Ilustración 110 Modificación de examen paso 2 .....	123
Ilustración 111 modificación de examen paso 3 .....	123
Ilustración 112 Modificación de examen paso 4 .....	124
Ilustración 113 Modificación de examen paso 5 .....	124
Ilustración 114 Modificación de examen paso 6 .....	125
Ilustración 115 Cancelación de examen paso 1.....	125
Ilustración 116 Cancelación de examen paso 2.....	126
Ilustración 117 Cancelación de examen paso 3.....	126
Ilustración 118 Recordatorio de medicamento.....	127
Ilustración 119 Recordatorio de citas.....	127
Ilustración 120 Recordatorio de examen .....	128
Ilustración 121. EPG o en español guía electrónica de programas. Ejemplo de aplicación interactiva.....	150

## LISTA DE TABLAS

Tabla 1 Cumplimiento de Objetivos específicos .....	4
Tabla 2 Definición de las Características de calidad del modelo ISO 9126.....	43
Tabla 3 Definición de los atributos del modelo de calidad ISO 9126 ....	44
Tabla 4 Análisis de características del modelo de calidad iso 9126 aplicado a los ides seleccionados.....	46
Tabla 5 Análisis de atributos específicos .....	47
Tabla 6 Otros atributos a tener en cuenta.....	47
Tabla 7 API's de Núcleo.....	82
Tabla 8 API's de Java TV.....	84
Tabla 9 API's de DAVIC.....	86
Tabla 10 API's de DVB .....	87
Tabla 11 Código Hola Mundo en Netbeans .....	99
Tabla 12 Prueba caso de uso Reanudar tratamiento.....	130
Tabla 13 Prueba caso de uso Formular medicamento .....	130
Tabla 14 Prueba caso de uso formular exámenes.....	130
Tabla 15 Prueba caso de uso Ingresar datos paciente.....	131
Tabla 16 Prueba caso de uso Ingresar tratamiento .....	131
Tabla 17 Prueba caso de uso Recordar medicamento .....	131
Tabla 18 Prueba caso de uso Recordar cita .....	132
Tabla 19 Prueba caso de uso Recordar examen.....	132
Tabla 20 Prueba caso de uso controlar recordatorio .....	132
Tabla 21 Prueba caso de uso Ingresar citas.....	133
Tabla 22 Prueba caso de uso Modificar citas .....	133
Tabla 23 Prueba caso de uso Cancelar citas .....	133
Tabla 24 Prueba caso de uso Ingresar examen .....	134
Tabla 25 Prueba caso de uso Modificar examen .....	134
Tabla 26 Prueba caso de uso Cancelar examen .....	134
Tabla 27 Prueba caso de uso Aceptar recordatorios .....	135
Tabla 28 Prueba caso de uso Posponer Recordatorios.....	135
Tabla 29. Niveles técnicos de interactividad relacionados con aplicaciones. Tomada de MHP-Guide .....	151
Tabla 30 Cuadros de Casos de Uso .....	153

## LISTA DE ANEXOS

Aplicaciones Interactivas .....	144
Niveles de Interactividad .....	149
Cuadros de Casos de Uso .....	153

# RESUMEN

## TÍTULO:

PROTOTIPO DE APLICACIÓN DE TELEMEDICINA BASADO EN EL ESTÁNDAR DVB-MHP PARA LA TELEVISION DIGITAL TERRESTRE INTERACTIVA\*

## AUTORES:

Garcés Rodríguez, Lina María\*\*  
Rovira Sánchez, Deiver Arley

## PALABRAS CLAVES:

DVB-T(Digital Video Broadcasting - Terrestrial),MHP (Multimedia Home Platform), TDT (Televisión Digital Terrestre), TVDi (Televisión Digital Interactiva), JavaTv, Xlet, Técnica Análisis de Características, Modelo de Calidad ISO 9126.

## DESCRIPCIÓN:

La televisión ha venido evolucionando hasta llegar a lo que se conoce hoy en día como Televisión Digital Interactiva. Colombia en el 2008 ha decidido acoger el estándar europeo para Televisión Digital Terrestre DVB-T, lo que trae consigo un cambio impactante en la población Colombiana, por esto se hace necesario el iniciar a estudiar esta tecnología en toda su extensión en especial en cómo desarrollar servicios y aplicaciones interactivas para TDT.

El presente proyecto pretende ser una guía para los desarrolladores, los cuales necesitan conocer los diferentes tipos de aplicaciones y servicios MHP que se pueden crear, las herramientas software existentes en el mercado que permiten hacerlos realidad, los estándares que rigen la tecnología, cómo está conformado el decodificador o Set-top box (el cual es el elemento principal de esta tecnología) y cómo es la programación de aplicaciones, para esto se elaboró un prototipo software llamado pastillero interactivo como ejemplo básico de desarrollo. Este ejemplo permite mostrar cómo puede ser la comunicación entre el televidente y la aplicación MHP, enviando y recibiendo datos de una base de datos determinada, característica fundamental de la interactividad.

También se presentan los modelos y diagramas que resultaron del análisis de requisitos del pastillero interactivo, un documento de pruebas del prototipo realizadas en el emulador proporcionado por el IDE, unos cuadros comparativos resultado de la evaluación de algunos de los IDE's que se encuentran en el mercado, tanto libres como privativos, y una evaluación basada en la técnica Análisis de Características y en el modelo de Calidad ISO 9126.

---

\* Trabajo de Grado en la Modalidad de Investigación

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Ph.D Sergio Fernando Castillo Castelblanco. Codirector: Msc. Víctor Eduardo Martínez Abaunza.

# SUMMARY

**TITLE:**APPLICATION PROTOTYPE OF TELEMEDICINE BASED ON DVB-MHP STANDARD FOR INTERACTIVE DIGITAL TERRESTRIAL TELEVISION\*

**AUTHORS:**

Garcés Rodríguez, Lina María  
Rovira Sánchez, Deiver Arley\*\*

**KEYWORDS:**DVB-T(Digital Video Broadcasting - Terrestrial),MHP (Multimedia Home Platform), DTT (Digital Terrestrial Television), iDTV (Interactive Digital Television), JavaTv, Xlet, Technical Analysis Features, ISO 9126 Quality Model.

**DESCRIPTION:**

The television has been evolving until getting to what is known today as Interactive Digital Television. Colombia in 2008 has decided to take the European standard for Digital Terrestrial Television (DTT) DVB-T, which brings an important change in the Colombian people, therefore it becomes necessary to beginning to study this technology in all its extension in special in how to develop services and interactive applications for TDT.

The current project pretends to be a guide for the developers, who need to know about the different types of MHP applications and services that can be created, the market for existing software tools that allow them possible, the standards that rule this technology, how the decoder or Set-top box works (which is the main element of this technology) and how is the application development process. For all this it has been developed a software prototype named Interactive Pillbox as a development basic example. This example allows showing the how could be the communication between the viewer and the MHP application, sending and receiving data from a specific database, an essential feature of interactivity.

Also it's presented the models and diagrams as result of the requirement analysis of the Interactive Pillbox, a prototype testing document carried out on the emulator provided by the IDE, some comparative tables result from the evaluation of some of the IDE's on the market, both free and proprietary, and an evaluation based in the Feature Analysis technique in the ISO 9126 Quality Model.

---

\* Graduation paper, Research mode

\*\* Faculty of Physical-Mechanical Engineering. School of Engineering and Computer Science. Director: Ph.D Sergio Fernando Castillo Castelblanco. Codirector: Msc.Víctor Eduardo Martínez Abaunza.

# ESPECIFICACIONES DEL PROYECTO

## TITULO:

PROTOTIPO DE APLICACIÓN DE TELEMEDICINA BASADO EN EL ESTÁNDAR DVB-MHP PARA LA TELEVISION DIGITAL TERRESTRE INTERACTIVA

## DIRECCIÓN:

DIRECTOR: Profesor. SERGIO FERNANDO CASTILLO CASTELBLANCO

CODIRECTOR: Profesor. VICTOR EDUARDO MARTÍNEZ ABAUNZA

## AUTORES:

LINA MARÍA GARCÉS RODRÍGUEZ

DEIVER ARLEY ROVIRA SÁNCHEZ

## ENTIDADES INTERESADAS EN EL PROYECTO:

Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander

Comisión Nacional de Televisión

Ministerio de Telecomunicaciones

Demás entidades relacionadas

## INTRODUCCIÓN

Gracias a la reciente fusión entre la televisión digital y el internet, se dio origen a lo que hoy se conoce como televisión digital interactiva (TVDi), la cual trae consigo gran cantidad de beneficios para los televidentes, dentro de los que se destacan una mejor calidad de sonido, video en alta definición (HD), mayor cantidad de canales, programas interactivos y servicios que le brindan al televidente la posibilidad de interactuar con los programas que acostumbra a ver, dejando atrás el antiguo modelo de televisión donde el usuario participaba tan solo como un receptor. En este sentido, el televidente podrá escoger, que, cuando y como ver lo que realmente desea ver.

Con el nacimiento de este tipo de tecnologías, se verán fortalecidos los sectores de la salud, la educación, el sector científico, tecnológico y el comercio de todo el país, pues se generarán nuevos servicios para toda la población en general.

La TVDi puede ser emitida por diferentes medios, ya sea televisión cableada, por satélite o por antenas terrestres. Actualmente, la televisión que se recibe por vía terrestre todavía es analógica, pero no por mucho tiempo. El próximo año, en el país se iniciarán las obras para implementar la Televisión Digital Terrestre (TDT), la cual reemplazará a la actual televisión analógica.

Es necesario mencionar que la TDT ya ha sido implantada en los países desarrollados y en algunos países de Latinoamérica se están iniciando las obras, por tal motivo Colombia tiene la necesidad de seguir el paso hacia la evolución tecnológica que otros países líderes han adoptado.

De esta manera con este proyecto se aporta a aquellas personas interesadas en incurrir en este mundo de la televisión digital interactiva, una guía de conceptos básicos, un análisis de algunas de las herramientas existentes para desarrollar aplicaciones para este sector, y un ejemplo básico de cómo se deben crear aplicaciones para esta tecnología siguiendo las reglas del estándar DVB-MHP, el cual fue aprobado por Colombia como estándar para la transmisión de Televisión Digital Terrestre Interactiva (TDT) en el país.

Para un enfoque práctico y aprovechando el auge de esta nueva tecnología, se creó un prototipo software que le permitirá a los televidentes controlar la toma de medicamentos, asistencia a citas y verificación de exámenes, utilizando el televisor el cual es uno de los electrodomésticos de mayor uso en los hogares.

En los capítulos siguientes se hace una descripción del problema planteado en este proyecto, su justificación, la descripción de las técnicas y software utilizado para su desarrollo y finalmente las pruebas de funcionamiento del prototipo final.

## 1. DESCRIPCIÓN DEL PROYECTO

La televisión es el medio de comunicación masivo más importante en todo el mundo, se ha visto fortalecido por la llegada de la Televisión Digital (TVD), la cual permite entregar muchos más programas con mayor calidad. Esta nueva forma de ver Televisión se hace más atractiva puesto que da infinitas posibilidades de interacción, y se puede transmitir satelitalmente, por cable y de forma terrestre.

Un tema importante a tratar es la TVDi o Televisión Digital Interactiva, que además de permitir ver canales nacionales e internacionales por medio de un proveedor de servicios, ofrece la integración con otro importante medio masivo de comunicación como lo es el Internet y sus aplicaciones de E-Commerce, E- Business, E-Health, E-Care, las cuales fueron base para el surgimiento de aplicaciones análogas por medio de la TVD, como T-Commerce, T- Business, T-Health, T-Care, respectivamente.

El 2008 fue para Colombia un año de decisiones importantes, y una de ellas fue adoptar el estándar Europeo para la transmisión de Televisión Digital Terrestre, este Estándar conocido como DVB – T fue desarrollado por el consorcio DVB (Digital Video Broadcasting) el cual busca unificar la forma en que se transmite la TVD definiendo una Plataforma independiente del contexto que permite generar aplicaciones para TVDi sin importar el hardware en que se ejecute.

Ahora, los investigadores, desarrolladores y proveedores de servicios de nuestro país, deben empezar a introducirse en esta nueva forma de ver TV, el estudio de los estándares que lo rigen y de las plataformas en que se desarrollan las aplicaciones pueden llevar a que Colombia sea uno de los países líderes en ésta tecnología en Latino América.

Con este proyecto se pretende conocer el Estándar adoptado antes de que se emitan las primeras señales de TDT (Televisión Digital Terrestre) en el país, que empezará en el 2010, y poder adquirir conocimientos que permitan un avance en esta tecnología y un desarrollo interno de los productos que se pueden necesitar en un futuro.

Se ha decidido iniciar este conocimiento desarrollando una aplicación en el campo de T-Care, puesto que esta es una de las ramas más importantes en cualquier sociedad, junto con la educación y el comercio. El prototipo de aplicación llamado **pastillero interactivo**<sup>1</sup> está orientado a aquellos pacientes de avanzada edad, que están en tratamiento continuo y necesitan tener presente la dosis y la hora exacta en que debe ingerir un medicamento determinado, éstas personas por sus características pasan más de cuatro horas al día viendo TV<sup>2</sup> (Ipsos-Napoleon Franco, 2008), lo que permite utilizar este electrodoméstico como una herramienta que sirva de apoyo para el paciente.

## OBJETIVO GENERAL

Desarrollar el prototipo de un “**pastillero interactivo**” basado en el estándar Europeo DVB - MHP (Digital Video Broadcasting - Multimedia Home Platform) para televisión digital.

## OBJETIVOS ESPECÍFICOS

---

<sup>1</sup> Un pastillero interactivo es una aplicación que ofrece al usuario un recordatorio personalizado en la toma de medicamentos, que contiene información relacionada con la hora, dosis, y nombre del medicamento a ingerir.

<sup>2</sup> Datos tomados de “La Gran Encuesta de la Televisión en Colombia” hecha por Ipsos-Napoleón Franco para la Comisión Nacional de Televisión en Marzo del 2008.

Elaborar un cuadro comparativo donde se detallen las características de cada uno de los emuladores para televisión digital disponibles en el mercado, y seleccionar el más adecuado a nuestras necesidades.

Análisis y Diseño de la aplicación llamada “pastillero interactivo”, Diseño de BD y Diseño de la interfaz de usuario de acuerdo a las especificaciones dadas por el estándar DVB-MHP.

Codificación de la aplicación utilizando la API de Java TV del lenguaje JAVA.

Realizar pruebas de funcionamiento del prototipo en el emulador seleccionado.

**TABLA 1 CUMPLIMIENTO DE OBJETIVOS ESPECÍFICOS**

OBJETIVOS ESPECIFICOS A CUMPLIR	DETALLE	PORCENTAJE DE CUMPLIMIENTO
<b>Elaborar un cuadro comparativo donde se detallen las características de cada uno de los emuladores para televisión digital disponibles en el mercado, y seleccionar el más adecuado a nuestras necesidades.</b>	Se eligieron las herramientas disponibles en el mercado para realizar el prototipo. Basados en el Modelo de Calidad ISO 9126 y en el método de Análisis de Características, se seleccionó la herramienta con mayor puntuación.	100%
<b>Análisis y diseño de la aplicación llamada "Pastillero Interactivo", Diseño de BD y diseño de la interfaz de usuario de acuerdo a las especificaciones dada por el estándar DVB - MHP.</b>	Se hizo el análisis de requisitos funcionales y no funcionales tanto de la parte WEB como de la aplicación MHP. Se crearon los diagramas UML de casos de uso, secuencias, de clases y de Entidad Relación para la aplicación. Se diseñó la interfaz de usuario siguiendo el estándar DVB-MHP	100%
<b>Codificación de la aplicación utilizando la API de Java TV del lenguaje Java</b>	Para la programación se utilizó el IDE IDesign, el cual está creado 100% en Java y cumple totalmente con las especificaciones del estándar europeo ya que utiliza la API de Java TV.	100%

<b>Realizar pruebas de funcionamiento del prototipo en el emulador seleccionado</b>	Se hicieron las pruebas de casos de uso para la aplicación MHP y para la aplicación WEB utilizando el emulador seleccionado, enviando y recibiendo datos desde y hasta el servidor de aplicaciones. De forma que se pudiera comprobar un correcto funcionamiento del prototipo.	100%
---	---	------

## METODOLOGÍA

Se eligió la Programación Extrema puesto que es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

XP es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil, la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración, donde el equipo vuelve a evaluar las prioridades del proyecto.

### XP COMO METODOLOGÍA

Básicamente, la programación extrema, busca dos objetivos claramente: hacer un software bien (con calidad) y de la forma más rápida posible. De hecho estos son los objetivos fundamentales de cualquier metodología aplicada al desarrollo de software y a cualquier otra área en general. A pesar de esto, con las metodologías de desarrollo actuales, el 70% de los proyectos fracasan y aproximadamente, también, el 70% de los fallos no son debidos a

cuestiones técnicas, son debidos a cambios en la gestión o problemas de comunicación.<sup>3</sup>

Con estos datos es lógico pensar en que las metodologías actuales no son lo suficientemente buenas, porque una tasa de éxito inferior a una tercera parte del total de proyectos no es algo deseable. Una vez analizado el problema, se puede ver en XP la solución, o al menos un acercamiento. La programación extrema centra su atención en la producción de software con una fuerte arquitectura, intentando sacar productos al mercado rápidamente, con gran calidad y motivando al equipo de trabajo para seguir mejorando esta tendencia. Como metodología, la programación extrema, presenta muchos puntos comunes con el desarrollo incremental, comenzando por el hecho de que el software desarrollado con XP se realiza de forma incremental.

#### FASES DE XP:

- Planificar
- Diseñar
- Codificar
- Probar

*La programación extrema proporciona una serie de ventajas a la hora de afrontar ciertos proyectos, que deben ser tenidas muy en cuenta, porque el uso de la XP en ciertos contextos puede permitir el ahorro de tiempo y recursos.*

#### VENTAJAS

- El resultado son productos usables con mayor **rapidez**.
- El proceso de **integración** es continuo, por lo que el esfuerzo final para la integración es nulo.
- Se consigue integrar todo el trabajo con mayor facilidad.
- Se atienden las **necesidades del usuario** con mayor exactitud. Esto se logra gracias a las continuas versiones que se ofrecen al usuario.
- Los productos son más fiables y **robustos** contra los fallos gracias al diseño de los test de forma previa a la codificación.
- Se obtiene código más **simple** y más fácil de entender, reduciendo el número de errores.

---

<sup>3</sup> Datos extraídos del documento "XP-An Overview"

- Gracias a la filosofía del “pair programming” (programación en parejas), se consigue que los desarrolladores apliquen las **buenas prácticas** que se les ofrecen con la XP.
- Gracias al “refactoring”<sup>4</sup> es **más fácil modificar** los requerimientos del usuario.

## ORGANIZACIÓN DEL LIBRO

El libro está organizado de la siguiente forma:

El capítulo 2 describe los conceptos generales que se tuvieron en cuenta para la realización del proyecto, empezando por explicar el porqué existe la necesidad de pasar de la televisión analógica a la digital y cuáles son las ventajas de hacer esta transición. Este capítulo también contempla en forma general los estándares de transmisión de la Televisión Digital Terrestre (TDT) y de middleware para el decodificador que existen en el mundo. Se explica también el significado de Televisión Digital Interactiva y todos los beneficios que trae consigo, sus niveles de interactividad y el tipo de aplicaciones que se pueden desarrollar para ésta. Este capítulo finaliza explicando lo que es MHP o Multimedia Home Platforms, el Canal de Retorno, el carrusel de datos, el protocolo de codificación MPEG-2 y los Xlets.

El capítulo 3 contiene el análisis comparativo de los entornos de desarrollo integrado existentes para desarrollar aplicaciones MHP, tanto libres como privativos, los cuadros comparativos de la evaluación y su resultado.

En el capítulo 4 se plasma el Análisis y Diseño del prototipo que contiene los requisitos de la aplicación, casos de uso, diagramas de secuencia y clases, y el modelo entidad relación.

El capítulo 5 contiene el software que se utilizó para desarrollar el prototipo, las diferentes APIs necesarias para crear una aplicación MHP, un breve manual de cómo desarrollar una aplicación con software libre

---

<sup>4</sup> En ingeniería del software, el término *refactorización* se usa a menudo para describir la modificación del código fuente sin cambiar su comportamiento, lo que se conoce informalmente por *limpiar el código*

y por último se da a conocer el resultado del prototipo objetivo de este proyecto.

En el capítulo 6 se muestran las pruebas que se realizaron de la aplicación en el emulador.

Por último se dan a conocer las conclusiones y recomendaciones que se tuvieron al desarrollar este proyecto.

## 2. MARCO TEÓRICO

### 2.1. TRANSICIÓN DE LA TELEVISIÓN ANALÓGICA A LA TELEVISIÓN DIGITAL

El mundo de la radiodifusión está cambiando rápidamente gracias al nacimiento de nuevas tecnologías. La introducción de la televisión digital terrestre (TDT) a nuestro país es una realidad y será implantada gracias a la decisión tomada por la Comisión Nacional de Televisión (CNTV), ya que el 28 de agosto del año 2008, tras largas jornadas de incertidumbre decidió adoptar el estándar europeo de televisión digital terrestre (DVB-T).

Esta elección del estándar digital terrestre representa así el tercer momento más importante en la historia de la televisión, tras la llegada de ésta en el año 1954 y posteriormente en el año de 1979 donde se pasó de la tradicional televisión de blanco y negro a una televisión a color.

Esta decisión no fue tarea fácil, ya que desde el año 2006 la CNTV aprobó la implementación de la TDT y se dio a la tarea de buscar el estándar que se adaptara mejor a las necesidades y geografía colombiana.

Con la introducción de la televisión digital terrestre se espera mejorar el uso del espectro electromagnético, ya que permitirá a los canales actuales enviar más de un programa de televisión por el mismo canal dedicado, brindando una mayor oferta de canales al igual que la posibilidad al usuario de ver lo que quiera, donde podrá escoger por ejemplo en el canal 4, entre el canal 4a, 4b, 4c donde cada uno de estos emitirá diferentes tipos de programas para los usuarios.

Otra de las ventajas importantes en la implementación de la TDT en Colombia, es que se podrá observar canales con mejor calidad de imagen y sonido, pues con la digitalización de las señales se alcanzará una calidad de video cercana a la brindada por el DVD y un sonido envolvente 5.1, de la misma forma se podrá ofrecer servicios especiales

como la introducción de subtítulos en diferentes idiomas a los programas favoritos de los televidentes, información extra de las películas o la posibilidad por ejemplo de ver el historial de un jugador de fútbol que acaba de anotar un gol, este tipo de servicios serán posibles si las cadenas radiodifusoras permiten acceder a este tipo de información.

De ahora en adelante el televisor pasará de ser un simple electrodoméstico donde se recibe información a ser un portal interactivo donde el usuario decidirá qué, cómo y cuándo ver. Es por esto que se dice que habrá una revolución en la forma como se ve la televisión en nuestro país.

## 2.2. TELEVISIÓN DIGITAL INTERACTIVA (TVDi) (MULTIMEDIA HOME PLATFORMS, 2006)

### 2.2.1. DEFINICIÓN

La televisión interactiva (TVi) puede ser vista como la evolución de la televisión, donde se fusionan el internet con la televisión digital. TVi mejora la TV tradicional a lo largo de dos dimensiones: permite el uso de señales digitales y contiene interactividad mediante un canal de retorno dándole la posibilidad al usuario de enviar datos ó hacer peticiones de contenido al canal radiodifusor, permitiendo que el usuario pueda interactuar de forma directa con los programas.

La televisión interactiva es una colección de servicios que soporta que el suscriptor seleccione el programa o programas de video que desee ver sin importar el día y la hora. Aunque la TVi se puede lograr usando señales análogas, ya hay una infraestructura para los servicios de TVD la cual provee un nuevo nivel de experiencia al usuario.

TVi tiene una larga historia, un emocionante presente, y un prometedor futuro. Para habilitar el presente y el futuro hay múltiples tecnologías y estándares relacionados al transporte y a la presentación de contenido.

### 2.2.2. APLICACIONES

Las aplicaciones de televisión interactiva dejan atrás el antiguo sistema de televisión de puro consumismo, donde solo el usuario recibe información y no puede interactuar o profundizar sobre los temas de interés.

La clasificación de estas aplicaciones puede ser vista en 10.1 Anexo: Aplicaciones Interactivas

Veamos ahora algunos de los estándares más importantes de televisión digital existentes en el mercado mundial.

### 2.3. ESTÁNDARES DE TRANSMISIÓN DE TELEVISIÓN DIGITAL TERRESTRE (TDT)

Como se dijo anteriormente, Colombia está pasando por un proceso de cambio, una evolución de lo que conocemos hoy en día como televisión digital. Actualmente, las personas que pagan una suscripción a un proveedor de servicios, reciben más de cien canales diferentes en sus televisores, y esto se debe a que este servicio es transmitido de forma digital. Algunos de estos proveedores de servicios utilizan diferentes medios de transmisión, ya sea por Cable (cable coaxial, Ej. Une), o de forma Satelital (donde cada suscriptor tiene una antena dirigida dedicada, Ej. Telefónica).

El país necesitaba nivelarse con respecto a Servicios interactivos que se estaban ofreciendo en los países desarrollados, servicios que utilizaban el televisor y los cuales cubrían la totalidad de la población de esos países, dado que el TV es el electrodoméstico de mayor uso en las sociedades. Para tal nivelación Colombia debía elegir la forma de transmitir la TV interactiva teniendo en cuenta aspectos sociales, económicos y geográficos.

De este análisis que se hizo durante aproximadamente tres años, se decidió utilizar la difusión terrestre, la cual permite llegar a cualquier parte del país (de igual forma como lo ha hecho la telefonía celular), lo que no permitía la difusión por cable puesto que la geografía no es apta para su implementación y saldría muy costosa. Otra ventaja de la difusión terrestre es su relativa economía para el televidente (pues solamente se debe adquirir un decodificador o un televisor específico).

Ya elegida la forma de transmisión, se debía escoger el estándar que la iba a regir y a continuación se nombran los que entraron en consideración por la CNTV (Comisión Nacional de Televisión).

#### 2.3.1. ATSC (ADVANCED TELEVISION SYSTEM COMMITTEE)

Es un grupo que se encarga del desarrollo de los estándares de la televisión digital en los Estados Unidos, y en base a estos estándares Canadá, México, Corea del Sur y recientemente Guatemala, Honduras y Costa Rica lo adoptaron. ATSC intenta reemplazar en los Estados Unidos a la NTSC <sup>5</sup>. (NTSC frecuentemente es usado en Estados Unidos, Canadá, Japón y buena cantidad de países de América Latina).

### 2.3.2. ISDB-T (INTEGRATED SERVICE DIGITAL BROADCASTING TERRESTRIAL)

**ISDB-T** o Transmisión Digital de Servicios Integrados es el estándar de televisión digital y radio digital que Japón ha creado para permitir a las estaciones de radio y televisión la conversión a digital y su respectiva transmisión de forma terrestre.

Además de transmisión de audio y video, ISDB también define conexiones de datos con Internet como un **canal de retorno** sobre varios medios y con diferentes protocolos. Esto se usa, por ejemplo, para interfaces interactivas como la transmisión de datos y guías electrónicas de programas.

### 2.3.3. DVB-T (DIGITAL VIDEO BROADCASTING TERRESTRIAL)

Es un estándar técnico creado por el Proyecto DVB que especifica la estructura de la imagen, codificación del canal y modulación del sistema de radiodifusión de televisión digital terrestre. Desde su inicio en 1997 ha sido ampliamente adoptado en el mundo, con más de 40 millones de receptores desarrollados en más de 30 países. Es un sistema flexible que permite entregar un amplio rango de servicios, desde HDTV (Televisión de alta definición) hasta múltiples canales SDTV (Televisión de definición estándar), recepción fija, portátil y móvil.

El flujo de datos transmitido tiene un único destino que es el STB (Set-top box) o decodificador, o en algunos casos el mismo televisor, y es aquí donde se marca la diferencia entre los estándares anteriores. Cada equipo debe tener un middleware que comunique el software con el hardware, y cada estándar anterior tiene sus propias especificaciones

---

<sup>5</sup> **NTSC** (*National Television System Committee*, en español *Comisión Nacional de Sistemas de Televisión*) es un sistema de codificación y transmisión de Televisión a color analógica desarrollado en Estados Unidos en torno a 1940, y que se emplea en la actualidad en la mayor parte de América y Japón, entre otros países

para el middleware independientemente del medio de transmisión, puesto que el mismo software middleware puede funcionar tanto para difusión vía satélite, cable o terrestre.

A continuación se listan las especificaciones middleware de los estándares para televisión digital.

## 2.4. ESTÁNDARES MIDDLEWARE PARA TELEVISIÓN DIGITAL INTERACTIVA.

### 2.4.1. ACAP

ATSC desarrolló el sistema middleware DASE (Digital TV Applications Software Environment), basado en sus propios estándares de Televisión Digital. DASE es la base para la siguiente generación que es conocida como ACAP. Canadá y Korea adoptaron ACAP para sus servicios de transmisión terrestre.

Para TDT, CableLabs<sup>6</sup> y ATSC trabajaron juntos para definir una especificación común basada en GEM (ver 3.2.5), ACAP, que asegura máxima compatibilidad entre los receptores de cable y de señales aéreas.

ACAP tiene como propósito proveer a los televidentes servicios interactivos avanzados y ofrecerles a los proveedores de contenidos, difusores, operadores de tv por cable y por satélite, y a los fabricantes de hardware los aspectos técnicos necesarios para desarrollar productos y servicios interoperables

### 2.4.2. MHP

La televisión digital en sus comienzos enfrentó graves dificultades, puesto que las distintas entidades crearon estándares, especificaciones hardware para STB y especificaciones middleware diferentes, sin tener en cuenta el mercado global, lo que llevó a que el mercado de la televisión digital fuera vertical. DVB fue uno de los organismos líderes en la creación de un estándar middleware independiente del hardware del STB. El objetivo principal del trabajo del grupo DVB era cubrir lo que se creía un objetivo difícil: “estandarizar los elementos de lo que se

---

<sup>6</sup> Cable Television Laboratories, Inc. (CableLabs) es un consorcio de investigación y desarrollo sin ánimo de lucro cuyos miembros son los operadores de cable.

conoce como Home Platform (set-top box, televisor, etc.), que son vistos como la clave para el éxito de las aplicaciones interactivas del dominio de la televisión digital.

En su forma más simple, MHP es una capa middleware o API (Application Programming Interface) que permite que las aplicaciones interactivas o servicios sean accedidos independientemente de la plataforma hardware en la que corren.

Actualmente, el trabajo se ha expandido para cubrir lo que son las redes digitales en casa ( in-home digital networks), PVR(Personal video recorder), aplicaciones móviles, otras tecnologías y su convergencia.

MHP es el estándar que han adoptado mayor cantidad de países alrededor del mundo, por esto se trata con más profundidad en la sección 2.5

### 2.4.3. JAVA TV

#### 2.4.3.1. JAVA TV API

En Marzo de 1998, Sun anuncia la API de JavaTV, y define una plataforma, para aplicaciones de televisión digital, desarrollada totalmente en Java. Parte de este trabajo de estandarización involucró el uso de APIs existentes en las plataformas de Televisión Digital, como Java Media Framework (JMF) para controlar el contenido de audio y video.

Java TV no fue el único estándar bajo desarrollo en esos momentos. Los requerimientos comerciales de MHP habían sido aceptados al mismo tiempo que JavaTV fue anunciado, y poco tiempo después de este anuncio, DVB seleccionó a Java como la base para la plataforma MHP. Muchas de las compañías involucradas en la iniciativa de JavaTV también trabajaron en MHP.

Esta relación cercana entre los dos estándares significa que ambos fueron diseñados para ser complementarios, y muchas de las coincidencias entre los dos son el resultado de que JavaTV sea una plataforma más neutral que MHP. JavaTV toma elementos de MHP y los aplica a todos los mercados de televisión digital.

JavaTV tiene por objeto proveer un conjunto básico de características para todos los receptores. Esto no define una plataforma completa para Televisión digital que ofrezca un completo acceso a cada característica en el receptor, puesto que estándares como MHP fueron diseñados para eso. Sin embargo, JavaTV ofrece un conjunto de APIs ( que proveen elementos comunes ) y un framework en el cual usarlas.

Debido al acuerdo hecho en 1999 entre Sun y DVB para utilizar Java en MHP, la inclusión de JavaTV en MHP fue asegurada para la versión 1.0 de MHP.

API Java TV es un paquete opcional de Java Micro Edition (Java ME) que proporciona control sobre las funcionalidades específicas de los receptores de televisión. Algunas de las características que la API Java TV brinda son: Acceso al sistema de información de base de datos, selección de contenido, control de ejecución multimedia, y acceso a los datos transmitidos por la señal de televisión.

Esta API está dirigida a desarrolladores Java, principalmente a aquellos que crean contenidos de televisión digital interactiva o herramientas para la misma. La mayoría de las funcionalidades que son requeridas para el desarrollo de contenido de TVDi es proporcionada por la plataforma Java sobre el receptor.

Esta API puede ser desplegada en ambientes con los estándares existentes para servicios o programas de información. Es importante que aquellos estándares y el código basado en éstos sean fácilmente integrados dentro del framework de Java TV API. Por lo tanto, la API de Java TV está definida con un alto nivel de abstracción de hardware y de protocolos.

La API de Java TV es una extensión de la plataforma Java desarrollada por Sun Microsystems y los líderes de la industria de televisión digital. Las principales compañías manufactureras de electrónica de consumo han anunciado su apoyo público hacia la adopción de ésta como API estándar para televisión digital en todo el mundo.

Esta API fue diseñada para proporcionar acceso a las funciones exclusivas de los receptores de televisión digital, incluyendo:

- Audio / video
- Acceso condicional
- Acceso a la información de servicio
- Sintonizador de control para cambiar de canal
- Control gráfico en pantalla.

Además, la API de Java TV proporciona una funcionalidad adicional en forma de sincronización de los medios de comunicación y el control del ciclo de vida de la aplicación. La sincronización de los medios de comunicación permite presentar contenido interactivo que se sincroniza con el audio y video de fondo de un programa de televisión mientras que el control del ciclo de vida de la aplicación permite la presentación de contenido dentro de programas de televisión, tales como comerciales.

En general, la API de Java TV y la plataforma Java permite a los desarrolladores crear contenidos verdaderamente convincentes.

### Entorno del API

La Ilustración 1. Entorno hardware y software de un receptor de tv digital (microsystems, 2009) muestra el entorno Hardware y Software para la API de Java TV cuando se aplica en un receptor de televisión digital.

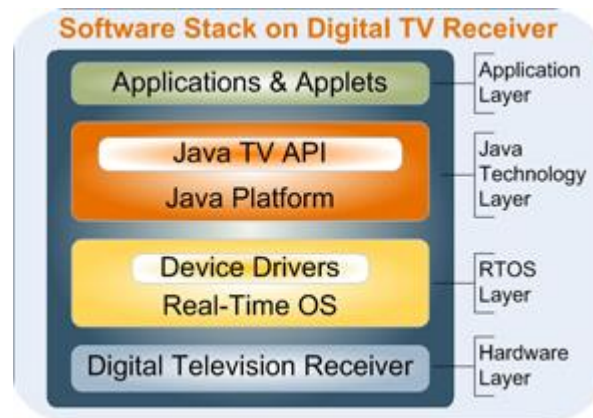


ILUSTRACIÓN 1. ENTORNO HARDWARE Y SOFTWARE DE UN RECEPTOR DE TV DIGITAL (MICROSYSTEMS, 2009)

#### 2.4.3.1.1. ENTORNO SOFTWARE

El entorno software consiste en la plataforma Java y la API de Java TV, que se ejecuta generalmente sobre un *sistema operativo de tiempo real*. (RTOS)

En el nivel más alto del entorno de software, una aplicación desarrollada para un receptor de televisión puede usar la API de Java TV y las librerías de clase en la plataforma Java subyacente. Esas librerías permiten a los desarrolladores de contenidos para televisión ofrecer a los usuarios contenidos de televisión interactiva, tales como, video por demanda, guías electrónicas de programación (EPGs) y múltiples ángulos de cámara por ejemplo en eventos deportivos.

En el nivel más bajo, el RTOS y las librerías de los dispositivos relacionados controlan el hardware a través de una colección de controladores de los dispositivos. El RTOS provee al sistema el soporte necesario para implementar la máquina virtual de Java (JVM) y las librerías de clases que componen la plataforma Java. La API de Java TV encapsula la funcionalidad expuesta por el sistema de librerías que controlan el hardware, por esto los desarrolladores pueden enfocarse en sus aplicaciones más que en detalles específicos de los dispositivos a utilizar.

#### 2.4.3.1.2. ENTORNO HARDWARE

La API Java TV opera dentro del contexto del hardware de un receptor de televisión. La característica definida de un receptor de televisión, digital o análogo, es la existencia de un medio transmisor y una tubería de datos. Esta tubería normalmente tiene ciertas características comúnmente entendidas, o subsistemas, como un radio y un demultiplexor. Ésta API proporciona una capa de abstracción que permite al programador de la aplicación tomar ventaja del hardware de televisión subyacente mientras ignora los detalles específicos del dispositivo.

#### Beneficios para el Cliente

La API de Java TV es dirigida a atender las necesidades de los fabricantes, operadores del sistema, y proveedores de contenido

buscando estándares para entregar contenido interactivo seguro a los consumidores por medio de los set-top-boxes de cable, receptores satelitales y televisión digital:

Los Fabricantes se beneficiarán por entregar tecnología Java habilitada para televisores y set-top-boxes al mercado que brinda un contenido completo de televisión interactiva a sus consumidores. El objetivo de Java TV es permitir a los fabricantes introducir nuevos productos en el mercado que pueden evolucionar para soportar nuevas formas de contenido interactivo en la medida que estén disponibles.

Los Operadores de Sistemas se verán beneficiados por la habilidad de desarrollar sus servicios de valor añadido para un conjunto diverso de dispositivos de tecnología Java habilitada en sus redes, independientemente de los procesadores o sistemas operativos en los dispositivos. La independencia de la plataforma, permite a los proveedores del servicio de televisión reducir el tiempo de mercadeo y costos al evitar la necesidad de portar sus servicios y aplicaciones a nuevas plataformas. Esto también puede permitirles la libertad de desplegar soluciones de set-top-boxes competitivas dentro de sus redes. Finalmente, ellos se beneficiarán de la compatibilidad con contenido de televisión interactiva existente que puede ser desplegado en sus sistemas.

Los creadores de contenido tendrán ventaja debido a la adopción de un estándar universal que amplía el mercado de oportunidades para televisión interactiva, y les permite desarrollar contenido de una vez y desplegarlo secuencialmente en un gran rango de receptores de televisión digital. Además, los desarrolladores de contenido tendrán la habilidad de emplear un verdadero lenguaje orientado a objetos, permitiéndoles rehusar los componentes de software existentes para entregar sus servicios interactivos en el mercado con rapidez y de manera rentable

#### 2.4.4. GEM (GLOBALLY EXECUTABLE MHP) (DVB, 2009)

Es la especificación abierta de middleware del proyecto DVB. Permite el despliegue de aplicaciones interactivas sobre la radiodifusión y redes de banda ancha, así como para paquetes de media, tales como los discos

Blu-ray. Está basado en Java y ofrece una plataforma, independiente del ambiente de ejecución de la aplicación, usada para crear contenido interactivo para los diferentes dispositivos y mercados. La plataforma está construida usando Java Micro Edition (JavaME) con APIs adicionales, las cuales ofrecen al TV funcionalidades específicas.

GEM no está restringido a una emisión de señales específica sino que define una abstracción de los conceptos comunes de varios sistemas de TV. GEM ha sido adoptado por un número de organizaciones incluidas CableLabs, ATSC, ARIB<sup>7</sup> y la Asociación de discos Blu-ray. GEM es el estándar middleware para televisión interactiva que recomienda el ITU-T<sup>8</sup>.

GEM es un framework que permite a las diferentes organizaciones crear armonía en las especificaciones técnicas, como la selección de un único motor de ejecución y (de ser posible) un conjunto común de APIs. El objetivo es que todas las aplicaciones y contenidos puedan ser interoperables para todas las plataformas basadas en GEM.

## 2.5.MHP (MULTIMEDIA HOME PLATFORMS) (MULTIMEDIA HOME PLATFORMS, 2006)

### 2.5.1. INTRODUCCIÓN

#### 2.5.1.1. EL PROYECTO DVB

Alrededor de 1990, las tecnologías de codificación de video, canal de la modulación y la integración de chips – después de décadas de investigación – llegan a un estado en que se indica que la radiodifusión de televisión digital podría ser comercialmente realizable en un futuro no lejano. En 1991, el “European Launching Group” fue establecido para comenzar a coordinar, junto a todos los interesados, la tecnología básica necesaria para la introducción de la Televisión Digital. Desde la fundación oficial del proyecto DVB creado por éste grupo en 1993, esta iniciativa ha incluido todos los avances técnicos en el ámbito de la televisión digital. Éste ha desarrollado especificaciones para los

---

<sup>7</sup> Association of Radio Industries and Businesses, es una organización de estandarización en Japón. (ARIB, 2009)

<sup>8</sup>El Sector de Estandarización de Telecomunicaciones (ITU-T) coordina los estándares de las Telecomunicaciones en nombre de la Unión Internacional de Telecomunicaciones (ITU). (ITU, 2009)

sistemas de televisión digital, los cuales han sido introducidos en organismos de normalización internacionales tales como la ETSI (Instituto Europeo de Estándares de Telecomunicaciones). Los estándares DVB/ETSI han sido adoptados a lo largo del mundo y han proveído las bases de cientos de redes cableadas, satelitales y terrestres y los mercados relacionados a esas redes.

Dentro de Europa, el éxito de DVB ha aportado un grado de interoperabilidad nunca antes visto en la historia. DVB cuenta actualmente con unas 250 empresas miembros de todo el mundo que abarcan todos los aspectos técnicos y de servicios relacionados con la televisión digital.

#### 2.5.1.2. LA NECESIDAD DE MHP COMO UNA API ESTÁNDAR ABIERTA

En la primera fase de estandarización, DVB se concentra en los fundamentos como la codificación para transmisión por cable, satélite y terrestre, servicios de información, transmisión de teletexto y los mecanismos básicos para la encriptación. Esto fue importante para lograr un comienzo exitoso para el mercado de la televisión digital alrededor de 1995.

Sin embargo, surgieron discusiones en algunos sectores sobre los sistemas de las APIs. Quedó claro que el potencial de la televisión digital podría ser considerablemente reforzado por la interactividad, pero al mismo tiempo aparecieron diferentes soluciones API en el mercado las cuales fueron incompatibles por ser propietarias. Ambos aspectos de esas soluciones, ser propietarias e incompatibles, causó muchas preocupaciones a la mayoría de los desarrolladores y sólo en algunos mercados orientados verticalmente, se introdujeron diferentes APIs. En esos mercados usualmente un operador de televisión de pago decide que API usar y financia los correspondientes decodificadores. Otros proveedores de servicio fueron bloqueados por hacer su contenido visible a esos decodificadores. En consecuencia, no se pudo llevar a cabo una ampliación del mercado.

Por lo tanto, en 1997, el proyecto DVB empezó a extender el ámbito de su excelente trabajo desde televisión digital pura a televisión digital interactiva y a desarrollar una sola API estándar.

Según la política de la Unión Europea, *“La interoperabilidad de los servicios interactivos de televisión y de los equipos de televisión digital, a nivel del consumidor, deberían estimularse para asegurar un flujo de información libre, diversidad cultural y pluralismo en los medios. Es deseable que los consumidores tengan la posibilidad de recibir, independientemente del modo de transmisión, todos los servicios de televisión digital interactiva, teniendo en cuenta la neutralidad tecnológica, los avances tecnológicos futuros, la necesidad de promover la adopción de la televisión digital, y el estado de competencia en los mercados de servicios de televisión digital. Los operadores de la plataforma de televisión digital interactiva deberían esforzarse por implementar una interfaz de programación de aplicaciones (API) abierta, que se ajustase a las normas o especificaciones adoptadas por un Consejo Europeo de normas de organización.”* (Recital 31 of the Directive 2002/21/EC)

Esta cita del marco regulador de la Unión Europea para las redes de comunicaciones electrónicas y servicios, pone en manifiesto la voluntad política de abrir los mercados y la normalización de las API en el ámbito de la televisión digital interactiva.

En consecuencia, la comisión de la Unión Europea acoge con satisfacción la introducción de MHP como norma europea para la iTV.

*MHP ó Multimedia Home Platform*, es un sistema intermediario abierto, diseñado por el proyecto DVB y estandarizado por la ETSI, el cual define una plataforma común para las aplicaciones interactivas de la televisión digital, independiente tanto del proveedor de servicios interactivos como del receptor de televisión utilizado. De este modo, MHP favorece la creación de un mercado horizontal donde aplicaciones, red de transmisión y terminales MHP pueden ser suministrados por proveedores o fabricantes independientes

#### *Actividades de MHP en DVB*

Poco después de los esfuerzos por estandarizar un API en DVB queda claro que, tecnológicamente, el nuevo estándar no podía estar basado en alguno de las APIs propietarias que existían en el mercado en ese entonces. La decisión fue hacer uso de la tecnología Java como base

de las nuevas APIs. Así, cada fabricante de productos compatibles con MHP dispone de un mercado más amplio sin depender de un organismo de radiodifusión televisiva, y los proveedores de servicios pueden desarrollar aplicaciones interactivas sin depender de un fabricante en particular, el único requisito que deben cumplir es ser compatible con MHP.

*La fuerza de esta normalización es su enfoque abierto.*

La primera versión de este estándar (DVB-MHP 1.0) se dispuso en Julio del 2000, después de hacer una demostración pública al aire libre sobre aplicaciones y decodificadores MHP en agosto de 1999, la cual tuvo lugar en la Internationale Funkausstellung IFA (Exhibición Internacional de Radiodifusión) en Berlin. Esta primera versión del estándar MHP tuvo una serie de deficiencias, ciertos aspectos técnicos no habían sido cubiertos totalmente. En consecuencia, se creó la siguiente versión de la norma (MHP 1.0.1). Ésta fue publicada por la ETSI, en octubre de 2001 y divulgada por la Comisión Europea como estándar Europeo el 31 de diciembre de 2002 (DO 2002 / C 331/04). En junio de 2002, se introdujo MHP 1.0.2. Esto, a su vez, fue seguida por MHP versión 1.0.3.

#### 2.5.2. ARQUITECTURA DE MHP

En su nivel más simple, MHP está ubicado en el contexto mostrado en la Ilustración 2 contexto mhp. (MARTÍNEZ P, 2008). El software MHP tiene acceso a *streams*<sup>9</sup> de audio/video y datos, pudiendo almacenar cierta cantidad de información. La plataforma puede ser capaz de redirigir *streams* hacia un sumidero o almacén. La plataforma recibe las entradas por medio de los dispositivos de acceso del usuario y envía las comunicaciones a través de una pantalla (u otros dispositivos tales como altavoces) para entregarlas al espectador. La plataforma también podría tener acceso a comunicaciones con entidades remotas

---

<sup>9</sup> Hace referencia al “flujo de contenido continuo y unidireccional”

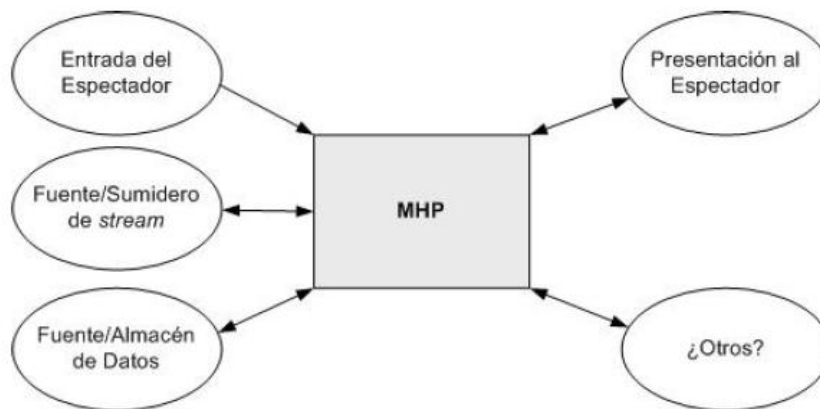


ILUSTRACIÓN 2 CONTEXTO MHP. (MARTÍNEZ P, 2008)

En el diagrama de la Ilustración 3 Interfaces externas entre mhp y el mundo exterior. (MARTÍNEZ P, 2008) se muestra un posible conjunto de interfaces externas entre un MHP y el mundo exterior.

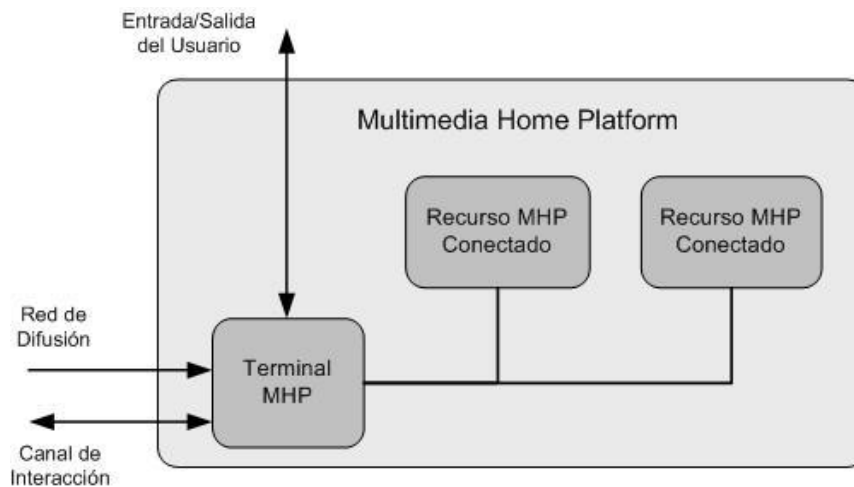


ILUSTRACIÓN 3 INTERFACES EXTERNAS ENTRE MHP Y EL MUNDO EXTERIOR. (MARTÍNEZ P, 2008)

Los recursos de MHP, accesibles desde una aplicación, pueden estar contenidos en una serie de entidades físicas conectadas al terminal MHP. Se llama “entidad” a cada una de las piezas independientes de hardware que forman parte de un cúmulo local (múltiple) de elementos que, como un todo, conforman una MHP. Una entidad es, por ejemplo, un decodificador, un videocasete digital o un módulo de acceso

condicional. Una entidad contiene una serie de recursos y, a su vez, cada recurso provee cierta cantidad de funciones.

Las especificaciones MHP definen cuatro perfiles para diferentes clases de funcionalidades. MHP 1.0 especifica solo el primer y segundo perfil. El tercer perfil fue introducido con MHP 1.1 y el cuarto con MHP 1.2

- Enhanced Broadcast Profile (Perfil 1): La versión simple de un ambiente MHP soporta éste perfil. Es encaminado por el STB (Set top box) sin un canal de retorno en un área de bajo costo. Este perfil permite el desarrollo de aplicaciones interactivas locales. Dado que carece de un canal de retorno, las aplicaciones pueden ser descargadas solo del flujo de contenido del radiodifusor – MPEG -2 Transport Stream-. Las aplicaciones típicas basadas en este perfil son guías de programas electrónicos, tickers de noticias o aplicaciones de teletexto.
- Interactive Broadcast Perfil (Perfil 2): Incluye las funciones del perfil 1, este perfil incluye soporte para un canal de retorno normalizado. Basado en un canal de retorno, una interacción entre la audiencia y los radiodifusores puede ser posible. Esto posibilita el soporte para aplicaciones como el televoting, T-commerce o pay-per-view (PPV). Otra ventaja del canal de retorno es que las aplicaciones MHP 1.1 pueden también ser descargadas por medio de una conexión a internet.
- Internet Access Profile (Perfil 3): En este perfil, el perfil 2 es extendido con el soporte para aplicaciones de internet. Solo se especifican APIs para acceso a diferentes servicios de Internet más que para servicios concretos. Para el uso de este perfil, los típicos servicios punto a punto como email y WWW pueden ser combinados con el mundo
- IPTV Profile (Perfil 4): Es el mayor perfil. Integra DVB-IPTV en la plataforma MHP. DVB-IPTV es formado por una colección de varias especificaciones para el desarrollo de DTV usando IP.

La siguiente figura presenta una vista de la arquitectura de software de MHP.



ILUSTRACIÓN 4 ARQUITECTURA DE MHP. (MHP, 2009)

A continuación se dará una explicación de las capas que conforman la Arquitectura MHP.

#### *Capa de Recursos (Resources)*

Este representa las diferentes plataformas de hardware de set-top-boxes. Así como también diferentes componentes como la CPU, memoria e interfaz de red, los componentes de la televisión como la interfaz DVB y el modulo del decodificador MPEG-2 también residen en esta capa.

#### *Capa del Sistema Software (System software)*

Las aplicaciones no están relacionadas directamente con los recursos. La capa de Sistema Software brinda una vista abstracta de los recursos. Esta capa intermedia aísla la aplicación del hardware, permitiendo la portabilidad de la aplicación.

*Application Manager:* Ésta capa incluye una aplicación de gestión de la función, que es responsable de la gestión del ciclo de vida de todas las aplicaciones, incluyendo las interoperables.

#### *Capa de Aplicación*

Como se muestra en la figura 4, MHP es capaz de manejar múltiples aplicaciones en una JVM(Java Virtual Machine), sobre MHP API.

Además de las aplicaciones normales también los plugins<sup>10</sup> pueden ser implementados por MHP, los cuales son usados para extender la funcionalidad de la plataforma. Hay dos categorías de aplicaciones y plugins en MHP: las interoperables y las no interoperables. Las aplicaciones y plugins interoperables pueden ser usadas por todos los tipos de receptores MHP, sobre MHP API. Éstas aplicaciones DVB-J<sup>11</sup> han sido especificadas en Java TV. El modelo de aplicación de las aplicaciones DVB-J sigue el modelo de los Xlets de Java TV. En contraste a las aplicaciones normales de Java, los Xlets tienen un concepto similar a los applets de Java. La implementación de aplicaciones y plugins específicos no son interoperables. El código nativo o las APIs especiales de Java, que no están disponibles en MHP, son comúnmente usados. Por esta razón, tales aplicaciones y plugins pierden la habilidad de correr en cualquier tipo de receptor MHP.

Además de DVB-J, hay otro método para construir aplicaciones MPH interoperables. El lenguaje aclaratorio DVB-HTML<sup>12</sup> para aplicaciones TVi ha sido incluido desde MHP 1.1. Los conceptos básicos de DVB-HTML como el ciclo de vida de las aplicaciones ya ha sido introducido desde MHP 1.0 pero no formalizado en un lenguaje específico. El framework básico de DVB-HTML está basado en una selección de módulos XHTML 1.0. Para la estructura, el nivel dos de CSS<sup>13</sup> es usado. ECMAScript<sup>14</sup> y DOM<sup>15</sup> nivel 2 soportan formularios, la

---

<sup>10</sup> Aplicaciones que se relacionan con otras para aportarles una función nueva y generalmente muy específica. Estas aplicaciones adicionales son ejecutadas por las aplicaciones principales e interactúan por medio de la API.

<sup>11</sup> Plataforma de DVB-MHP basada en la Máquina Virtual de Java (JVM). Define un conjunto de APIs genéricas, situadas entre las aplicaciones y el sistema de software, para proporcional a las distintas aplicaciones acceso a los recursos disponibles en el receptor.

<sup>12</sup> Digital Video Broadcasting – HyperText Markup Language. Estándar que permite el acceso a los contenidos de Internet.

<sup>13</sup> Cascading Style Sheets. Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

<sup>14</sup> Lenguaje de programación basado en JavaScript. Define un lenguaje de tipos dinámicos ligeramente inspirado en Java y lenguajes del estilo C. Soporta algunas características de la Programación Orientada a Objetos.

<sup>15</sup> Document Object Model. Interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

base de los aspectos dinámicos de las aplicaciones DVB-HTML. La principal razón para la introducción de DVB-HTML es que muchas compañías son especializadas en HTML y rechazaban la idea de adoptar otro lenguaje, y que Java no es la mejor opción para crear aplicaciones de presentación dirigidas. DVB-HTML está disponible para cada uno de los tres perfiles de MHP pero es más importante para el Internet Access Profile.

MHP está siendo usado en todo el mundo. Especialmente en Europa, MHP es la plataforma de TVi dominante. Los más grandes seguidores en Europa son Italia, Finlandia, Alemania entre otros. Globalmente, MHP gana mucha importancia respecto a los demás estándares de TVi, dado que estos se basan en especificaciones MHP.

## 2.6. CANAL DE RETORNO

El canal de retorno (conocido como: *Back Channel*, *Backchannel*, *Return Channel*, *Return Link*, *Reverse Channel*, *Back Channel Network Connection*), es el camino físico por donde el televidente puede enviar información o peticiones al operador de red, siendo parte integral de la Televisión Digital Interactiva.

Es un camino de comunicación de doble vía entre el televidente o usuario final y el proveedor de contenido interactivo, esto es lo contrario al canal de emisión, el cual es un camino con una vía de comunicación desde el operador de red, red de televisión, hacia el televidente, lo que permite una experiencia de televisión linear.

El canal de retorno puede ofrecer gran cantidad de opciones, las cuales incluyen módems estándar y líneas telefónicas, cable módems, conexiones satelitales, entre otros.

El canal de retorno puede variar dependiendo del costo del receptor y del tipo de red a la que él esté conectado. La Ilustración 5 muestra algunas de las tecnologías de canal de retorno disponibles.

	<i>PSTN</i>	<i>ISDN</i>	<i>GSM</i>	<i>SMS</i>	<i>GPRS</i>	<i>UMTS</i>	<i>DSL</i>	<i>Cable</i>	<i>SATMODE</i>
Downstream (Kbits/s)	56	64-128	14	14	171	2048	256-52000	512-10000	30000
Upstream (Kbits/s)	56	64-128	9.20	160 bits/ packet	171	384	64-3400	64-128	1-64

**ILUSTRACIÓN 5. TASA DE DATOS DE ALGUNAS TECNOLOGÍAS DE CANAL DE RETORNO**

## 2.7.DSM-CC (DIGITAL STORAGE MEDIA COMMAND AND CONTROL)

DSM-CC es utilizado en la transmisión de aplicaciones y datos MHP como también flujos de eventos. Dado que un sistema de emisión es de un solo camino, el cliente no tiene cómo decirle al emisor cuales datos desea. Por esto los datos son organizados en carruseles. Por ejemplo, un archivo es emitido después de otro y entonces el primero es emitido de nuevo inmediatamente, etc.

Esto significa que el cliente puede tener que esperar una vuelta completa para acceder a un archivo específico al menos que el archivo sea repetido varias veces durante una vuelta. La duración del tiempo de acceso para un archivo es definido por el ancho de banda del carrusel y el tamaño de los datos sobre éste.

Las aplicaciones y datos MHP son emitidas en carruseles de objetos. DSM-CC también soporta carruseles de objetos que son la forma sencilla del carrusel.

El carrusel de Objetos DSM-CC añade al carrusel de datos el concepto de archivos, directorios y flujos de datos, proporcionando una funcionalidad cercana a un sistema de archivos estándar.

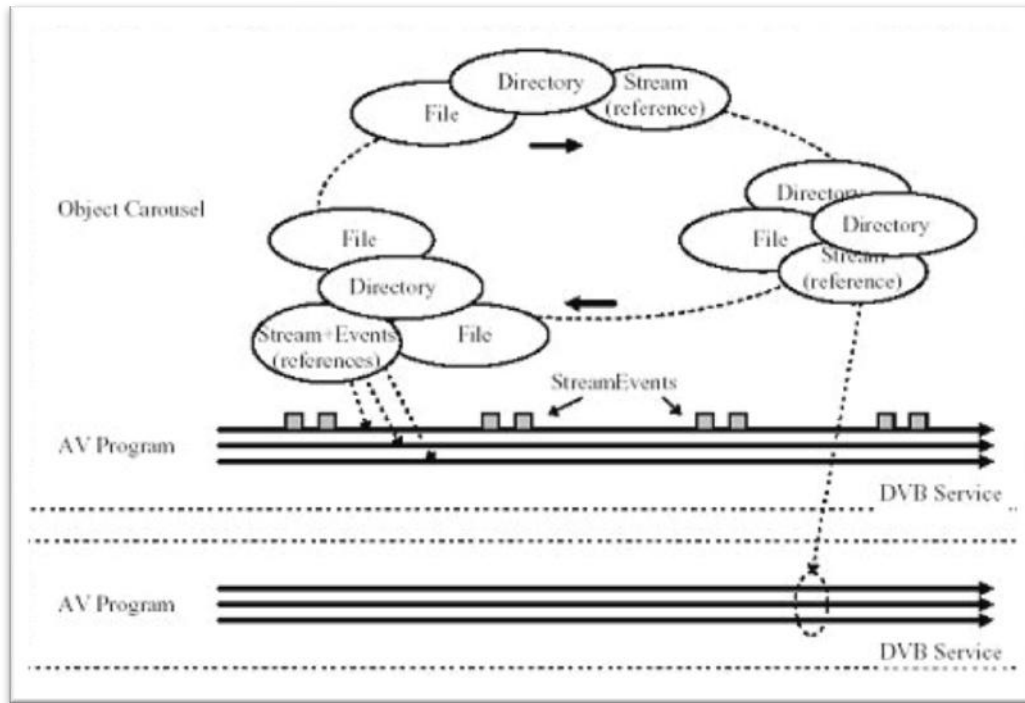


ILUSTRACIÓN 6 CARRUSEL DE OBJETOS

## 2.8.MPEG-2

Es un estándar para codificar video y audio. Describe una combinación de métodos para comprimir video y audio los cuales permiten el almacenamiento y transmisión.

MPEG-2 es ampliamente usado como el formato para señales de televisión digital que se transmiten de forma terrestre, cableada o satelital. También especifica el formato de las películas y otros programas que se distribuyen en DVD o discos similares. Como tal, las estaciones de televisión, los receptores, y otros equipos son diseñados para este estándar.

MPEG-2 define dos distintos, pero relacionados, formatos de contenidos. Uno de ellos es el *Transport Stream o Flujo de Transporte*, diseñado para multiplexar varios programas, y es el que utiliza DVB.

También define el formato *Program Stream* o *Flujo de Programa*, creado para multiplexar un solo programa, utilizado por el DVD.

MPEG-2 no especifica el tipo de multiplexación utilizar, se puede usar ya sea TDM (Multiplexación por División de Tiempo) o la estadística. La TDM asigna un espacio de tiempo concreto y constante a cada componente del flujo de programa. En la multiplexación estadística los diferentes programas se intercambian información del ancho de banda requerido para la transmisión. Así, un programa que necesite mucho ancho de banda puede beneficiarse de uno que tenga espacio libre y que, de otra manera, se desaprovecharía utilizando bits de relleno o *stuffings bits*. Las dos multiplexaciones son en el dominio del tiempo.

Los dos formatos de contenido definidos por MPEG-2, *Transport Stream* y *Program Stream* utilizan paquetes para la transmisión o almacenamiento de datos. Un **paquete** es una colección de bits de un mismo tipo (video, audio, datos).

Los paquetes pueden ser de tamaño fijo o variable, en función de la aplicación para la que estén destinados. Los de tamaño variable son utilizados para aplicaciones en las que el medio esté libre de errores: DVD, CD-ROM, Disco Duro. La longitud de dichos paquetes puede llegar a ser muy grande, minimizando el espacio reservado para cabeceras y demás mecanismos de señalización los cuales reducen la eficiencia del sistema. Los paquetes de tamaño fijo normalmente son pequeños y se utilizan para aplicaciones en las que el medio es susceptible de introducir errores (por ejemplo en la transmisión). Al tener paquetes de longitud fija y pequeña se simplifica muchísimo la corrección de errores en los mismos, además de reducir el efecto de dichos errores sobre el sistema.

Los paquetes son la llave a la flexibilidad y a la extensibilidad de los sistemas MPEG-2. Un sistema basado en paquetes es flexible, ya que permite una multiplexación sencilla basada en el tiempo y al mismo tiempo la optimización de recursos a través de una asignación dinámica basada en la prioridad de los diferentes paquetes.

Un sistema basado en paquetes es fácilmente ampliable, por lo que se garantiza la extensibilidad del mismo. Dicha extensibilidad se consigue

haciendo que el receptor descarte aquellos paquetes que no es capaz de decodificar. De esa forma, la ampliación de las prestaciones del sistema no afecta a los receptores antiguos, ya que simplemente descartan lo que desconocen y siguen siendo capaces de decodificar el resto.

## 2.9. APLICACIONES MHP

OCAP y MHP soportan aplicaciones escritas en lenguaje Java o HTML, aunque las más usadas son las primeras. Las aplicaciones para receptores MHP hechas en Java son llamadas DVB-J, y para OCAP, OCAP-J, igualmente las aplicaciones HTML reciben el nombre de DVB-HTML y OCAP-HTML. La misma aplicación puede ser de tipo DVB-J o OCAP-J puesto que se permite ejecutar en los dos estándares.

MHP y OCAP soportan aplicaciones descargadas de flujos MPEG emitidas. MHP 1.1 añade soporte para cargar aplicaciones sobre conexiones IP, usando HTTP u otros protocolos, OCAP también las soporta.

Desde la perspectiva del desarrollador todas las aplicaciones pueden ser idénticas, pero existen diferencias importantes entre ellas. A continuación se van a nombrar los tipos de aplicaciones que se pueden ejecutar en un receptor:

### 2.9.1. APLICACIONES SERVICE BOUNDS

Son descargadas cada vez que se necesiten ejecutar, y pueden ser iniciadas por el usuario o por el receptor. Son asociadas a un canal específico o evento.

### 2.9.2. APLICACIONES UNBOUND

Son descargadas y no son asociadas a un servicio. Son usadas típicamente por los operadores de red para ofrecer servicios con valor añadido, un ejemplo muy común es un EPG.

A diferencia de las aplicaciones Service-bounds, las aplicaciones Unbound no son parte de MHP y no pueden ser usadas en sistemas MHP. Las aplicaciones Unbound solo se pueden cargar a través de una conexión IP de dos vías, ejemplo un cable modem.

### 2.9.3. APLICACIONES ALMACENADAS

Se crearon para evitarle al usuario la descarga continua de aplicaciones. Una aplicación almacenada es una aplicación MHP transmitida que puede ser almacenada en el terminal MHP para ser usada después. Esto puede ser muy útil si la aplicación se carga más rápido localmente que de lo que carga desde el canal de transmisión. Una aplicación asociada con un servicio de radiodifusión puede ser iniciada solo cuando este está señalado en la AIT (Tabla de Información de Aplicaciones, usada para señalar y controlar las aplicaciones MHP de un servicio) de un servicio de radiodifusión, incluso si este se encuentra almacenado en el terminal MHP. Además de las aplicaciones almacenadas relacionadas a servicios específicos, también se pueden desarrollar aplicaciones independientes, como, por ejemplo, un juego.

#### 2.9.3.1. RESTRICCIONES DE LAS APLICACIONES ALMACENADAS

Las aplicaciones almacenadas relacionadas con servicios de radiodifusión se comportan como cualquier otra aplicación de radiodifusión, no hay diferencias, a pesar de que las aplicaciones almacenadas se cargan más rápido que las aplicaciones desde el canal de transmisión.

Por tanto, las aplicaciones almacenadas autónomas no deberían lidiar con información relacionada con el actual servicio.

Una aplicación almacenada consiste en objetos descargados desde el canal de transmisión. Estos objetos descargados pueden ser de tipo directorio o archivos. Por lo tanto cualquier otro tipo de objetos no podrán ser guardados en la terminal MHP.

Si no hay espacio para almacenar una aplicación, cualquier aplicación con mínima prioridad debe ser removida. Todas las aplicaciones almacenadas deben ser removidas cuando el receptor cambia de red.

Las aplicaciones almacenadas no son utilizadas en los sistemas MHP, mientras que en el sistema OCAP si funcionan y se usan.

#### 2.9.4. APLICACIONES NATIVAS

Soportadas por OCAP, son aplicaciones escritas en un lenguaje diferente a Java o HTML y están incorporadas en el receptor.

#### 2.9.5. APLICACIONES LOCALES

De forma diferente como funciona en aplicaciones MHP, una aplicación local es almacenada en una memoria no-volátil del terminal MHP por parte del fabricante. Consecuentemente, una aplicación local no tiene que ser tratada como una aplicación MHP. Ejemplos claros de estos son: el navegador, herramientas para la configuración de la terminal MHP, algunos juegos o el administrador de Xlets.

Debido a que las aplicaciones locales son propias de cada fabricante, no se define que complementos deben ser incluidos en una terminal MHP. De esta manera las aplicaciones MHP transmitidas no son capaces de interactuar con las aplicaciones locales. Podría existir una comunicación indirecta cuando las aplicaciones accedan a funciones básicas como la selección de servicios, la cual es también manejada por el navegador de aplicaciones del terminal MHP y por tanto, podría influir en el comportamiento del navegador.

## 2.10. INTRODUCCIÓN A LOS XLETS

El modelo de las aplicaciones normales de Java hace varias hipótesis acerca del entorno, que no son compatibles con un producto de consumo como lo es el STB. En particular asumen que solo una aplicación está ejecutándose en la máquina virtual de Java (JVM) y cuando la aplicación termina, la máquina también lo hace. En un PC esto no es un problema, pero esto causa dolores de cabeza cuando se trabaja en un sistema donde no se pueden tener esas suposiciones. El modelo normal del ciclo de vida también asume que una aplicación se carga, se ejecuta de inmediato y se termina, lo que no es cierto en un sistema de consumo.

El modelo de ciclo de vida de los Java applets en la Web es más apropiado: El browser carga un applet en una JVM, lo inicializa y lo ejecuta. Si una página contiene dos applets, ellos pueden correr los dos en la misma JVM sin crear interferencia entre ellos. Cuando un applet termina, es removida de la JVM sin afectar a los demás que están corriendo en ese instante.

Dado que el modelo del applet todavía tiene muchas funcionalidades atadas a la Web, o no es apropiado para todos los casos, se generalizó un modelo de ciclo de vida más adecuado para los sistemas de

consumo. El resultado “Xlets”. Esto constituye las bases para todos los sistemas basados en JavaTV, incluyendo MHP y OCAP. Como los applets los xlets permiten una interfaz externa ( Application Manager en el caso de MHP) para controlar el ciclo de vida de una aplicación y proveerle un camino de comunicación con su entorno.

La ejecución de un Xlet puede ser pausada y reanudada, puesto que en un ambiente como el receptor de televisión digital varias aplicaciones pueden ser ejecutadas al mismo tiempo, y por las restricciones del Hardware sólo una aplicación puede ser vista por el usuario. Las aplicaciones que no son usadas en un momento dado, deben ser pausadas para mantener los recursos libres para las próximas aplicaciones a usarse.

Dada la importancia de la fiabilidad y la robustez en sistemas de televisión digital, un Xlet tiene más restricciones de seguridad impuestas en él que un applet.

Los Xlets tienen un ciclo de vida bien definido con estados de “ No carga, Carga, Pausa, Activación y Destrucción”.

Una aplicación puede tener un estado “Inválido” cuando no puede correr el servicio actual debido al “garbage collected”<sup>16</sup>.

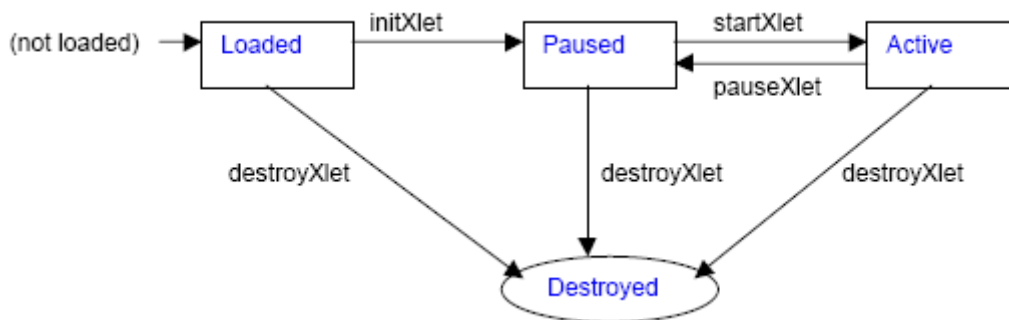


ILUSTRACIÓN 7 DIAGRAMA DE LOS ESTADOS DEL CICLO DE VIDA DE UN XLET. (MHP, 2009)

<sup>16</sup> Un recolector de basura (en inglés, garbage collector) es un mecanismo implícito de gestión de memoria implementado en algunos lenguajes de programación de tipo interpretado o semi-interpretado.

**Loaded:** La instancia de aplicación de DVB-J es cargada pero no inicializada.

**Paused:** La instancia de aplicación minimiza su uso de recurso.

**Active:** La instancia de aplicación está trabajando y proporcionando el servicio.

**Destroyed:** La instancia de aplicación ha liberado todos sus recursos y terminado.

La razón de tener estos diferentes estados de aplicación puede encontrarse en dos restricciones MHP inherentes:

- Las aplicaciones han de ejecutarse en entornos de bajo rendimiento: En promedio en los Terminales MHP los recursos del sistema pueden ser usados por una aplicación hasta un nivel en donde ninguna otra aplicación puede empezar. En lugar de una destrucción continua de la primera aplicación, es posible liberar los recursos escasos y ejecutarse en un modo pausado. Por lo tanto, la multitarea básica es posible utilizando los recursos del terminal eficientemente.
- Las aplicaciones no deben colisionar entre sí: Para mantener el sistema terminal limpio, es necesario que las aplicaciones se liberen después de ser terminadas. Esto se da al ingresar al estado **destroyer** antes de cambiar al estado **not loaded** de nuevo. Esto también permite la introducción de una instancia superior de gestión para controlar todas las aplicaciones por medio de una interface bien definida. Cada estado tiene tareas típicas que son realizadas cuando la aplicación entra al estado.

Muchos Xlets son permitidos por el administrador "Xlet" (Xlet Manager), el cual es una aplicación residente. El mecanismo de comunicación interno de "Xlet Manager" permite a los Xlets la comunicación y cooperación entre sí. Una sola aplicación es permitida por su propia instancia VM (Virtual Machine).

La clase inicial de las aplicaciones MHP debe implementar la interface `javax.xlet.Xlet` y todos los métodos definidos, los cuales se usarán por el Xlet manager para indicar el cambio de estado de las aplicaciones

como se muestra en la fig. 5. Cuando el manejador de aplicaciones de un terminal MHP es invitado a iniciar una aplicación, llamará primero `initXlet(XletContext ctx)` y después `startXlet()`. El método `pauseXlet()` enviará el Xlet al fondo y `startXlet()` puede ser usado de nuevo para reaundar el Xlet.

Para finalizar la aplicación MHP, el Xlet manager llama `destroyXlet(boolean unconditional)`, retorna el Xlet en estado **not loaded**. El switch booleano se usa para decirle al Xlet si es obligado a terminar o si tiene la opción de negar la solicitud de terminación.

Si un Xlet quiere cambiar el estado actual del ciclo de vida de la aplicación por sí mismo, debe informarle al Xlet manager acerca del cambio. La señalización del cambio de estado al Xlet manager se realiza mediante la invocación de métodos según `XletContext`, el cual lo proporciona el Xlet por la invocación de `initXlet(XletContext ctx)`. Hay tres métodos para informarle al Xlet manager acerca del cambio:

- **Void resumeRequest():** Si el Xlet quiere cambiar su estado de **paused** a **active**, tiene que invocar el método `resumeRequest()` de su objeto `XletContext`. Pero tenga en cuenta que el Xlet manager no necesita configurar el Xlets en estado activo.
- **void notifyPaused():** Tan pronto como el Xlet entra al estado **paused** por sí mismo, debe usar éste método para señalar el estado **paused** a el Xlet manager. Esto no es necesario, si el Xlet manager invoca el método `pauseXlet()`.
- **Void notifyDestroyed():** Cuando el Xlet entra al estado **destroyed** por sí mismo sin la iniciativa del Xlet manager (invocación de `destroyXlet(boolean unconditional)`), entonces el Xlet manager es informado sobre la autodestrucción de la aplicación MHP con la llamada al método `notifyDestroyed()` a `XletContent`.

El objeto `XletContent` da acceso a la configuración de entorno del Xlet con el método `getXletProperty(java.lang.String key)`. Similarmente para el método `getProperty(java.lang.String key)` del objeto del Sistema de java, una cadena predefinida es usada para acceder a los ajustes. Actualmente existen las siguientes propiedades accesibles:

- `dvb.org.id`": El identificador de la organización del Xlet al cual XletContent pertenece, como se señala en la emisión. El valor es retornado como un identificador de la organización codificado como String.
- `dvb.app.id`": Señala el identificador de la aplicación del Xlet. El valor retornado es el identificador de la aplicación codificado como String.
- `dvb.caller.parameters`": Provee parámetros cuando el Xlet fue iniciado por un AppProxy. Los parámetros son retornados como String[]. Si los parámetros no están disponibles, entonces String[] tiene longitud cero.
- `XletContext.ARGS`: Parámetros señalados en la emisión. Estos parámetros son retornados como String[]. Si no hay parámetros disponibles, entonces String[] tiene longitud cero.

### 3. ANÁLISIS COMPARATIVO DE LOS ENTORNOS DE DESARROLLO INTEGRADOS EXISTENTES PARA DESARROLLAR APLICACIONES MHP

#### 3.1. DEFINICIÓN

Un Entorno de Desarrollo Integrado o en inglés (*Integrated Development Environment - IDE*), es un programa compuesto por un conjunto de herramientas que sirven de ayuda al desarrollador de aplicaciones a hacer su trabajo más fácil.

Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, puede utilizarse para varios.

Un IDE consiste en:

- Un editor de texto
- Un compilador
- Un depurador

- Un intérprete<sup>17</sup>
- Herramientas de automatización
- Posibilidad de ofrecer un sistema de control de versiones
- Factibilidad para ayudar en la construcción de interfaces gráficas de usuarios (GUI)

### 3.2.IDE'S EXISTENTES PARA DESARROLLAR APLICACIONES PARA TELEVISIÓN DIGITAL<sup>18</sup>

Actualmente existe gran cantidad de IDEs para desarrollar aplicaciones interactivas para Televisión Digital, y entre tanta variedad se destacan dos líneas por las cuales las organizaciones y desarrolladores independientes dedicados a esta tecnología pueden decidirse. Por un lado se tiene los IDEs propietarios, desarrollados por empresas privadas, y por otro los IDEs libres, elaborados por comunidades de software libre. En cada una de éstas líneas existen diferentes opciones, y cada opción presenta ventajas y desventajas a la hora de utilizarlas, y uno de los objetivos de este proyecto es dar a conocer algunas de ellas.

En primera instancia, se ilustran las ventajas y desventajas de usar ya sea herramientas privadas o libres para desarrollar aplicaciones MHP para televisión digital interactiva.

#### 3.2.1. HERRAMIENTAS DE DESARROLLO PRIVADAS

##### 3.2.1.1. VENTAJAS

- Facilitan el desarrollo de aplicaciones
- Incluyen en una librería las clases necesarias para compilar la aplicación MHP, dado que la especificación MHP solo contempla un subconjunto de clases Java.
- Permite un desarrollo más rápido y sencillo
- Permite una adopción de la tecnología sencilla.

---

<sup>17</sup> Programa informático capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel. Los intérpretes se diferencian de los compiladores en que mientras estos traducen un programa desde su descripción en un lenguaje de programación al código máquina del sistema destino, los primeros (los intérpretes) sólo realizan la traducción a medida que sea necesario, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.

<sup>18</sup> Basado en el artículo: TDT: herramientas para desarrollo MHP. (Sanz de Acedo, 2008)

Permite comprender fácilmente:

- Las capas en las que se divide la presentación de una aplicación.
- La creación de páginas y las propiedades que se le pueden asignar.
- Las propiedades de navegación entre páginas.
- Se cuenta con un emulador para PC<sup>19</sup>
- La presentación final es mejor
- No se precisan conocimientos avanzados en Java

Se diferencian claramente las tres etapas del desarrollo: presentación, propiedades y contenidos

#### 3.2.1.2. DESVENTAJAS

- No permiten optimizar el código generado por el IDE
- El costo por adquirir su licencia es elevado, en promedio de € 2.500 por equipo
- No permite profundizar en el código de bajo nivel
- No se tiene control sobre el código generado

#### 3.2.2. HERRAMIENTAS DE DESARROLLO LIBRES

##### 3.2.2.1. VENTAJAS

- Se requiere solamente de un editor de texto con un compilador de Java, con el cual se pueden lograr aplicaciones más eficientes
- Permite optimizar los escasos recursos con los que se cuentan para este tipo de aplicaciones.
- Se precisan conocimientos en Java.
- Se profundiza en el desarrollo de aplicaciones MHP.
- Se tiene total control sobre el código de la aplicación y por tanto sobre la calidad y optimización del mismo.

##### 3.2.2.2. DESVENTAJAS

- Se necesitan conocimientos en programación Java avanzados.

---

<sup>19</sup> Permite hacer las pruebas de diseño y funcionamiento, emulando las características de un decodificador MHP en el PC en cualquier etapa del desarrollo.

- Se requiere que el mismo desarrollador incluya este conjunto de clases o librería, para poder crear aplicaciones MHP que cumplan con el estándar.
- Por el momento, no cumplen todas las especificaciones de estándar
- Las herramientas libres no han sido probadas tan exhaustivamente como las soluciones comerciales, debido al tamaño del mercado y al número de diferentes combinaciones posibles de equipamiento, parámetros y streams.
- No es recomendable el uso exclusivo de estas para la creación de aplicaciones en producción.
- No presentan funcionalidades ni ayudas para el desarrollo específico de aplicaciones en MHP.
- No se cuenta con un entorno muy amigable.
- Se desarrollará de forma más lenta.
- La presentación final puede no ser satisfactoria si no se le dedica tiempo.

Cuando se habla de entornos de desarrollo propietarios<sup>20</sup> para desarrollar aplicaciones para televisión digital interactiva (TVDi), los expertos piensan en aquellas soluciones que se destacan por su flexibilidad, simplicidad y versatilidad, entre estas se destacan<sup>21</sup>:

**Alticomposer** (Alticast) (de la empresa Coreana Alticast)

**IDesigner** (Xperts) (de la empresa Alemana Mit - Xperts)

**JAMEAuthor** (Jame)

**Cardinal Studio - ICAREUS iTV SUITE** (Icareus)

**OSMOSYS SDK** (Osmosys)(de la empresa Suiza OSMOSYS)

---

<sup>20</sup> Llamados Authoring Tools.

<sup>21</sup> Si desea ver una lista más amplia de todas las herramientas propietarias refiérase a la guía MHP (Multimedia Home Platforms, 2006) página 201.

Todas estas son posibles alternativas para aquellas personas y compañías que pretendan incurrir en el desarrollo de aplicaciones y servicios para la TVDi.

Para los que se decidan por una alternativa libre, Existen entornos de desarrollo integrado como Netbeans y Eclipse, de la misma manera se cuenta con el Kit de desarrollo Java, **JDK**. En cuanto a las posibilidades de emulación, están disponibles de forma gratuita emuladores libres como **XletView** (XletView) y **OpenMHP**. Son una buena forma de empezar teniendo poco presupuesto.

Como se puede observar se tiene una amplia variedad a la hora de escoger el IDE con el cual desarrollar para televisión interactiva, por lo mismo, se ha decidido escoger de cada línea varias herramientas para evaluarlas y de acuerdo a esa evaluación seleccionar cuál utilizar para el desarrollo del “pastillero interactivo”. En la siguiente sección se detalla cómo se va a realizar esta evaluación.

### 3.3.EVALUACIÓN DE IDE'S (LAWRENCE PFLEEGER)

Para la evaluación de un producto software existen muchos tipos de estudios, técnicas y modelos. Ésta evaluación se basó en el modelo ISO 9126 y en la técnica de análisis de características, que son descritos a continuación.

#### 3.3.1. TÉCNICA DE EVALUACIÓN: ANÁLISIS DE CARACTERÍSTICAS

Es un tipo de valoración sencillo, que se utiliza para tasar y categorizar los atributos de varios productos a fin de facilitar la decisión acerca de qué herramienta de diseño adquirir.

Pasos:

- Seleccionar los atributos claves que deberá tener la herramienta.
- Identificar las herramientas posibles.
- Para cada una de ellas se evalúan éstos criterios desde 1 (no satisface) a 5 (satisface completamente).
- A cada criterio se le da un nivel de importancia desde 1 ( poco o nada importante) a 5 ( muy importante ).

- Se multiplica el valor de la importancia por el puntaje del criterio para cada criterio, y después se suma.
- Sobre la base de estos puntajes, se selecciona la herramienta que más puntaje obtuvo.

### 3.3.2. MODELO DE CALIDAD ISO<sup>22</sup> 9126

Es un modelo útil para articular lo que se valora más en el software que se usa.

Es un modelo jerárquico con seis atributos principales que contribuyen a la calidad. En la Ilustración 8 Modelo de calidad según ISO 9126 se muestra la jerarquía y en la Tabla 3 Definición de los atributos del modelo de calidad ISO 9126 se definen los atributos principales.

La norma recomienda medir las características presentadas a la derecha en forma directa, pero no da detalles acerca de cómo debe hacerse la medida.

La jerarquía ISO 9126 es estricta: cada característica de la derecha sólo está relacionada exactamente con un atributo de la sección izquierda del modelo. Las características a la derecha se relacionan con la visión del usuario del software<sup>23</sup>.

---

<sup>22</sup> International Standardization Organization

<sup>23</sup> En esta parte del proyecto, los autores son los usuarios del software



ILUSTRACIÓN 8 MODELO DE CALIDAD SEGUN ISO 9126

TABLA 2 DEFINICIÓN DE LAS CARACTERÍSTICAS DE CALIDAD DEL MODELO ISO 9126

Característica de calidad	Definición
<b>Funcionalidad</b>	En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.
<b>Confiabilidad</b>	Un conjunto de atributos que inciden sobre la aptitud del software para mantener su nivel de desempeño bajo las condiciones establecidas y por el período de tiempo establecido.
<b>Facilidad de uso</b>	Un conjunto de atributos que determinan el esfuerzo necesario para el uso y la valoración individual de tal uso por parte de un conjunto de usuarios declarados o implicados.
<b>Eficiencia</b>	Un conjunto de atributos que afectan la relación entre el desempeño del software y la cantidad de recursos utilizados bajo condiciones establecidas.

<b>Portabilidad</b>	Un conjunto de atributos que incide sobre la aptitud del software para ser transferido de un ambiente a otro (lo que incluye ambiente organizacional, de hardware o de software).
---------------------	---

**TABLA 3 DEFINICION DE LOS ATRIBUTOS DEL MODELO DE CALIDAD ISO 9126**

<b>Atributo de calidad</b>	<b>Definición</b>
<b>Adecuación</b>	Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición
<b>Exactitud</b>	Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
<b>Interoperación</b>	Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
<b>Seguridad</b>	Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.
<b>Madurez</b>	La capacidad del producto de software para evitar fallas como resultado de errores en el software.
<b>Tolerancia a defectos</b>	Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
<b>Facilidad de recuperación</b>	Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo
<b>Facilidad de comprensión</b>	Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
<b>Facilidad de aprender</b>	Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación
<b>Facilidad de operar</b>	Agrupar los conceptos que evalúan la operación y el control del sistema.
<b>Comportamiento en el tiempo</b>	Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
<b>Comportamiento de los recursos</b>	Atributos del software relativos a la cantidad de recursos usados y la

	duración de su uso en la realización de sus funciones.
<b>Adaptabilidad</b>	Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
<b>Facilidad de instalación</b>	Es el esfuerzo necesario para instalar el software en un ambiente determinado.
<b>Conformidad</b>	La capacidad del software para adherirse a estándares o convenciones relacionados a la portabilidad.
<b>Facilidad de reemplazo</b>	Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

Para la valoración de los IDEs se tuvieron en cuenta otros atributos, que le permitirán a las personas interesadas en adquirir una herramienta de desarrollo decidir cuál es la que mejor se adecúa a sus necesidades, presupuesto, hardware, expectativas, etc. (Pressman), estas son:

- Auto documentación: El grado en el que el IDE proporciona documentación significativa. (Por ejemplo: autocompletar cuando se está programando).
- Estética: Qué tan agradable es a la vista el software para el usuario.
- Escalabilidad: Capacidad de ir aumentando el número de plugins para el IDE. Permite crear las herramientas necesarias para la interactividad (menús, gráficas, textos, etc.) que se van sumando al IDE.
- Conectividad: Referente a la comunicación con bases de datos, XML, y demás herramientas que permitan utilizar el canal de retorno.
- Integración: Capacidad de reconocer diversos lenguajes, XML, SQL, etc.
- Comunidad: Personas usuarias de la herramienta que se reúnen para aclarar dudas, hacer recomendaciones y aprender más sobre la herramienta.
- Tipo de Licencia del IDE y precio.
- Software adicional que requiere para funcionar
- Sistemas Operativos bajo los cuales se ejecuta

- Niveles de interactividad de las aplicaciones desarrolladas permitidas por el IDE.
- GUI: Contiene una Interfáz Gráfica de Usuario y para cada uno de sus componentes un panel de propiedades.
- Tipo de software que se puede desarrollar: Se evalúa si el IDE permite hacer sistemas de información completos o aplicaciones sencillas.

### 3.3.3. DIAGRAMA COMPARATIVO DE LOS IDE'S SELECCIONADOS

Los IDEs seleccionados fueron iDesigner, Jame Author, Icareus TV Suite, Netbeans con el emulador XletView. Se escogieron estas herramientas puesto que las empresas, en el caso de las herramientas propietarias, tienen a disposición de los desarrolladores versiones Demos que pueden ser solicitadas en las respectivas páginas Web y luego ser descargadas. Las licencias de los Demos tienen duración entre 1 y 3 meses.

A continuación se muestran los cuadros comparativos que permitieron analizar cuál herramienta escoger para desarrollar el prototipo.

**TABLA 4 ANÁLISIS DE CARACTERÍSTICAS DEL MODELO DE CALIDAD ISO 9126 APLICADO A LOS IDEs SELECCIONADOS**

Característica	Atributo	iDesigner	Jame Author	Icareus TV Suite	Netbeans	Importancia
<b>Funcionalidad</b>	Adecuación	5	4	4	3	3
	Exactitud	5	5	4	3	3
	Interoperación	1	1	4	5	5
	Seguridad	-	-	-	-	-
<b>Confiabilidad</b>	Madurez	3	2	2	3	5
	Tolerancia a defectos	3	2	2	4	5
	Facilidad de recuperación	3	3	4	4	5
<b>Facilidad de uso</b>	Facilidad de comprensión	5	4	5	2	5
	Facilidad de aprender	4	5	5	2	5
	Facilidad de operar	4	5	5	2	5
<b>Eficiencia</b>	Comportamiento en el tiempo	5	5	4	3	5
	Comportamiento de los recursos	4	5	4	4	5
<b>Portabilidad</b>	Adaptabilidad	5	3	3	5	5
	Facilidad de instalación	5	5	5	4	3
	Conformidad	5	4	2	3	5
	Facilidad de reemplazo	1	1	1	5	5

<b>Total</b>	<b>260</b>	242	244	240
--------------	------------	-----	-----	-----

**TABLA 5 ANÁLISIS DE ATRIBUTOS ESPECÍFICOS**

<b>Atributo</b>	<b>iDesigner</b>	<b>JameAuthor</b>	<b>Icareus TV Suite</b>	<b>Netbeans</b>	<b>Importancia</b>
<b>Auto documentación</b>	1	1	1	1	4
<b>Estética</b>	5	5	4	2	3
<b>Escalabilidad</b>	5	2	5	5	5
<b>Conectividad</b>	5	1	1	5	5
<b>Comunidad</b>	3	1	3	3	5
<b>GUI</b>	5	5	5	1	5
<b>Total</b>	<b>109</b>	64	86	80	

**TABLA 6 OTROS ATRIBUTOS A TENER EN CUENTA**

<b>Atributo</b>	<b>iDesigner</b>	<b>JameAuthor</b>	<b>Icareus TV Suite</b>	<b>Netbeans</b>
<b>Licencia</b>	Propietaria	Propietaria	Propietaria	Propietaria
<b>Precio</b>	€ 3.700 <sup>24</sup>	€ 5.000		Gratis
<b>Licencia Emulador</b>	Propietaria	Propietaria	Propietaria	Libre
<b>Software adicional</b>	JRE 1.4.2 o superior	JRE 1.4.2_13	JRE, JMF	JRE, JMF, J2ME, MHP, HAVI, DVB.
<b>S.O que lo soporta</b>	MAC OS, Linux, Windows 2000 en adelante	Windows 2000/XP	Windows XP	MAC OS, Linux, Windows 2000 en adelante
<b>Niveles de interactividad de las aplicaciones desarrolladas</b>	Unidireccional Solo emisión Bidireccional	Unidireccional al Solo emisión	Unidireccional Sólo emisión	Unidireccional Solo emisión Bidireccional
<b>Tipo de software que se puede desarrollar</b>	Todo tipo de aplicaciones	Noticias, portales de información	Noticias, portales de información.	Todo tipo de aplicaciones

<sup>24</sup> Para usos educativos se obtiene 50% de descuento en esta licencia

### 3.3.4. RESULTADO DE LA EVALUACIÓN

En la tablaTabla 4 se puede apreciar el resultado de aplicar el Análisis de Características, como Técnica de Evaluación, sobre los atributos definidos por el Modelo de Calidad ISO 9126. El puntaje de importancia que se usó se basó en el hecho de que se necesitaba un IDE que permitiera realizar el prototipo de forma eficiente, es decir, que la programación no fuera una tarea extenuante, que el producto final se rigiera por el estándar DVB-MHP, y que permitiera realizar un prototipo de acuerdo a los requisitos que se plantean en la sección 4.1.

Algo muy importante que se estaba buscando en un IDE, era el poder realizar aplicaciones bidireccionales, por esta razón Icareus y Jame Author no fueron seleccionados.

Netbeans con el emulador XletView permite realizar aplicaciones bidireccionales que no cumplen completamente con el estándar DVB-MHP, además de que la codificación es extensa y monótona, por estas razones no fue seleccionada esta herramienta.

La herramienta software que tuvo mejores resultados fue iDesigner<sup>25</sup>, el cual brinda las características necesarias para cumplir con los objetivos del presente proyecto.

## 4. ANÁLISIS Y DISEÑO DEL PROTOTIPO

### 4.1.ESPECIFICACIÓN DE REQUISITOS

#### REQUERIMIENTOS FUNCIONALES DEL PROTOTIPO MHP

---

<sup>25</sup> El prototipo se desarrolló en una versión de prueba con licencia de 6 meses, la cual puede ser solicitada a través de la página web : <http://www.mitxperts.com/products/idesigner/demo.php>

#### REQUERIMIENTOS DE USUARIO

Debido a que en un hogar varias personas usan el TV, la aplicación debe distinguir a que persona o paciente va destinado el recordatorio.

Un paciente necesita el recordatorio no solo del medicamento a ingerir, sino también, de las citas de control y los exámenes.

Un paciente puede ser atendido por varios doctores (un médico general, un especialista, un terapeuta, etc.), dependiendo de la enfermedad que padezca.

La aplicación puede ser deshabilitada por el usuario en cualquier momento, debido a que puede causar molestia a la hora de ver tv.

La aplicación debe controlar si el paciente ingiere el medicamento, o si va ya sea al examen o la cita, esto depende del recordatorio.

El paciente puede estar en diferentes tratamientos al mismo tiempo, por esto es necesario que para cada medicamento, se lleve un control.

Los medicamentos pueden ser en tableta, inyecciones, jarabes, gotas, etc. Lo que requiere que la aplicación controle el tipo de medicamento y la forma de suministrarse.

La aplicación puede ser consultada a cualquier hora, ya sea para, agregar evento, editar, consultar o eliminar.

#### REQUERIMIENTOS DEL SISTEMA

Se debe manejar una tabla de usuarios, los cuales serán asociados a los recordatorios.

Se deben almacenar las citas, exámenes y medicamentos para cada usuario de la aplicación.

Se debe crear una tabla de doctores y otra para el tipo de profesional (médico general, especialista, terapeuta, etc).

Debe existir un plugin que le permita al televidente mediante los botones del control remoto, minimizar la aplicación.

Los recordatorios se controlan mediante el botón "Aceptar Recordatorio", que significa que el usuario realizó el recordatorio, de

otra forma existirá un botón “Aplazar” el cual permite volver a ejecutar el recordatorio minutos después.

Cada usuario puede estar asociado a uno o más medicamentos, por esto se debe crear una tabla de medicamentos.

Se debe crear una tabla con el tipo de medicamento y otra con la forma de suministro.

Se deben crear formularios que permitan crear, modificar, consultar y eliminar ya sean citas, exámenes o medicamentos.

La aplicación debe controlar el tiempo entre toma de medicamento y ejecutar un “plugin” que indique que medicamento ingerir (almacenar en pila).

La aplicación se ejecuta, esté o no activa por el usuario.

Para las citas y los exámenes, se debe ejecutar el “plugin” el día antes del evento y debe decir las recomendaciones (por ejemplo: llevar exámenes anteriores, ir en ayunas, etc.)

Las recomendaciones son ingresadas por el doctor.

La aplicación debe capturar la hora de la primera dosis del medicamento y tener en cuenta el tiempo que este dure.

Cuando un paciente termine el tratamiento, poner nota de felicitación si lo cumplió totalmente.

La aplicación debe conectarse a una DB por medio de la red.

Cuando se prenda el tv, o se active la aplicación se deben mostrar los recordatorios vencidos.

#### REQUERIMIENTOS NO FUNCIONALES DEL PROTOTIPO MHP

##### REQUERIMIENTOS DE USUARIO

Debe ser fácil de usar.

Debe ser navegable mediante los botones del control remoto

No impedir que el televidente siga viendo su programa de televisión.

Los mensajes se puedan leer desde una distancia de 2 m.

Debe ser intuitiva.

#### REQUERIMIENTOS DEL SISTEMA

La interfaz de usuario debe basarse en el estándar mhp.

La aplicación MHP debe ser desarrollada en Java.

#### REQUERIMIENTOS FUNCIONALES DE LA APLICACIÓN WEB

##### REQUERIMIENTOS DE USUARIO

Debe permitir ingresar los datos del paciente (el cual se convertirá en televidente) y la fórmula médica.

Dar opción al doctor de reanudar tratamiento.

##### REQUERIMIENTOS DEL SISTEMA

Para reanudar un tratamiento, se deben almacenar los tratamientos anteriores, para que el doctor no deba ingresar todos los datos desde el principio.

Se debe crear un formulario donde le permita al doctor mediante el ingreso de número de cédula del paciente, ingresar la fórmula médica, la cual debe ser almacenada en una base de datos.

El doctor no puede asignarle citas ni exámenes al paciente. Estas asignaciones se realizan vía telefónica, y el paciente por medio de la aplicación MHP debe ingresar los recordatorios asociados a citas y exámenes.

La aplicación web se debe enlazar a la misma base de datos de la aplicación MHP.

#### REQUERIMIENTOS NO FUNCIONALES DE LA APLICACIÓN WEB

##### REQUERIMIENTOS DE USUARIO

Debe ser intuitiva

Debe tener menús que le permitan al doctor minimizar el uso del teclado.

## REQUERIMIENTOS DEL SISTEMA

Esta aplicación debe ser orientada a la Web.

### 4.2.LÓGICA DE LA APLICACIÓN



ILUSTRACIÓN 9 LÓGICA DE LA APLICACIÓN

La aplicación MHP localizada en el decodificador o STB, es la que se encarga de comunicarse con las aplicaciones del lado del servidor, en este caso Servlets, y muestra en la pantalla del televisor las interfaces correspondientes.

### 4.3.CASOS DE USO

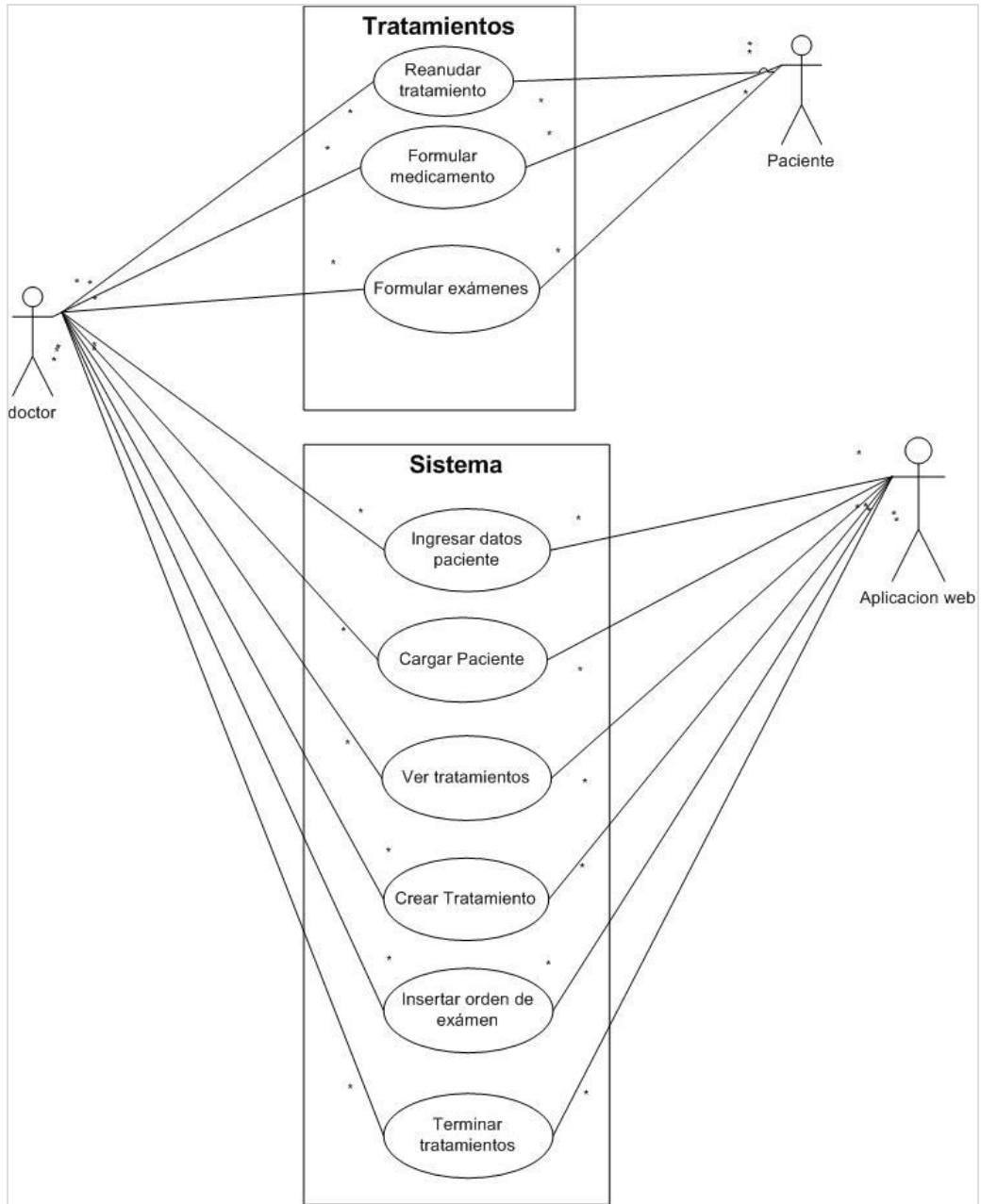
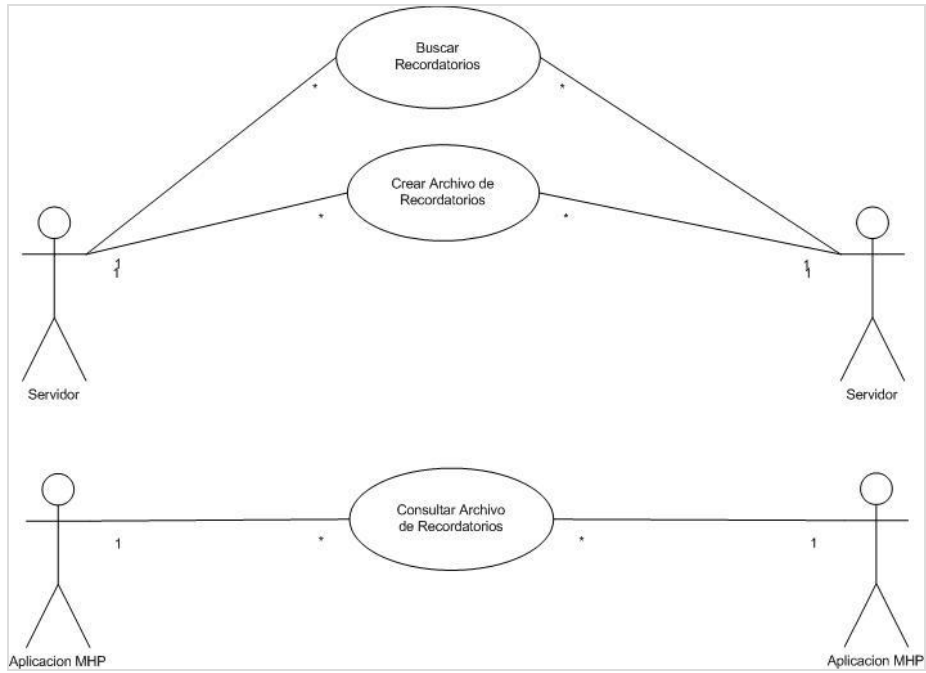


ILUSTRACIÓN 10 CASO DE USO DOCTOR



**ILUSTRACIÓN 11 CASOS DE USO SERVIDOR Y APLICACION MHP**

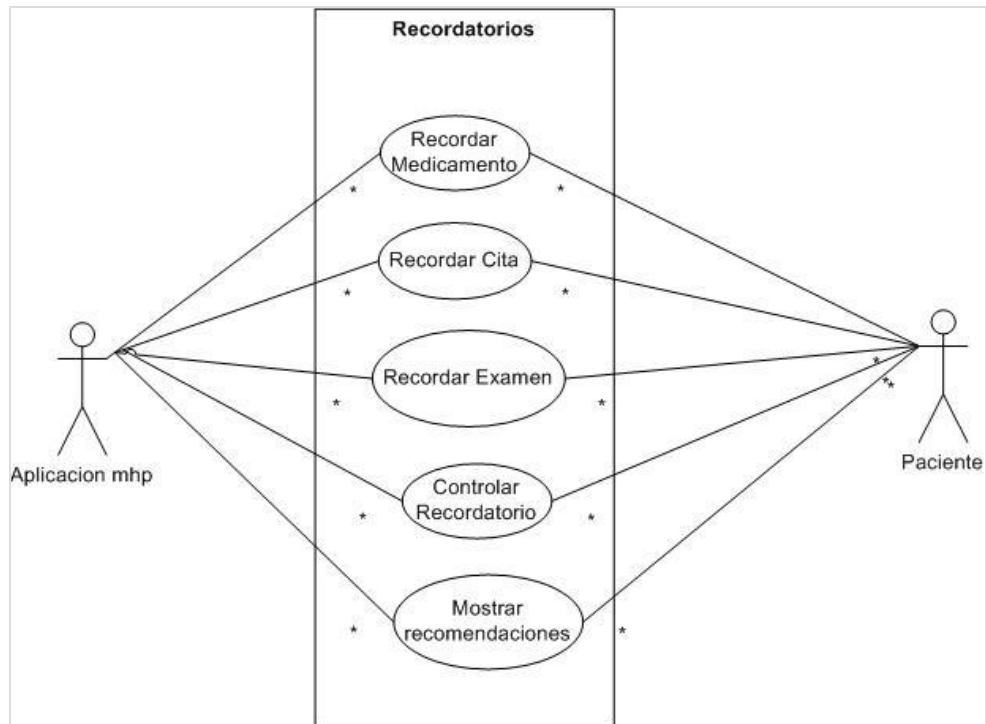


ILUSTRACIÓN 12 CASO DE USO APLICACION MHP

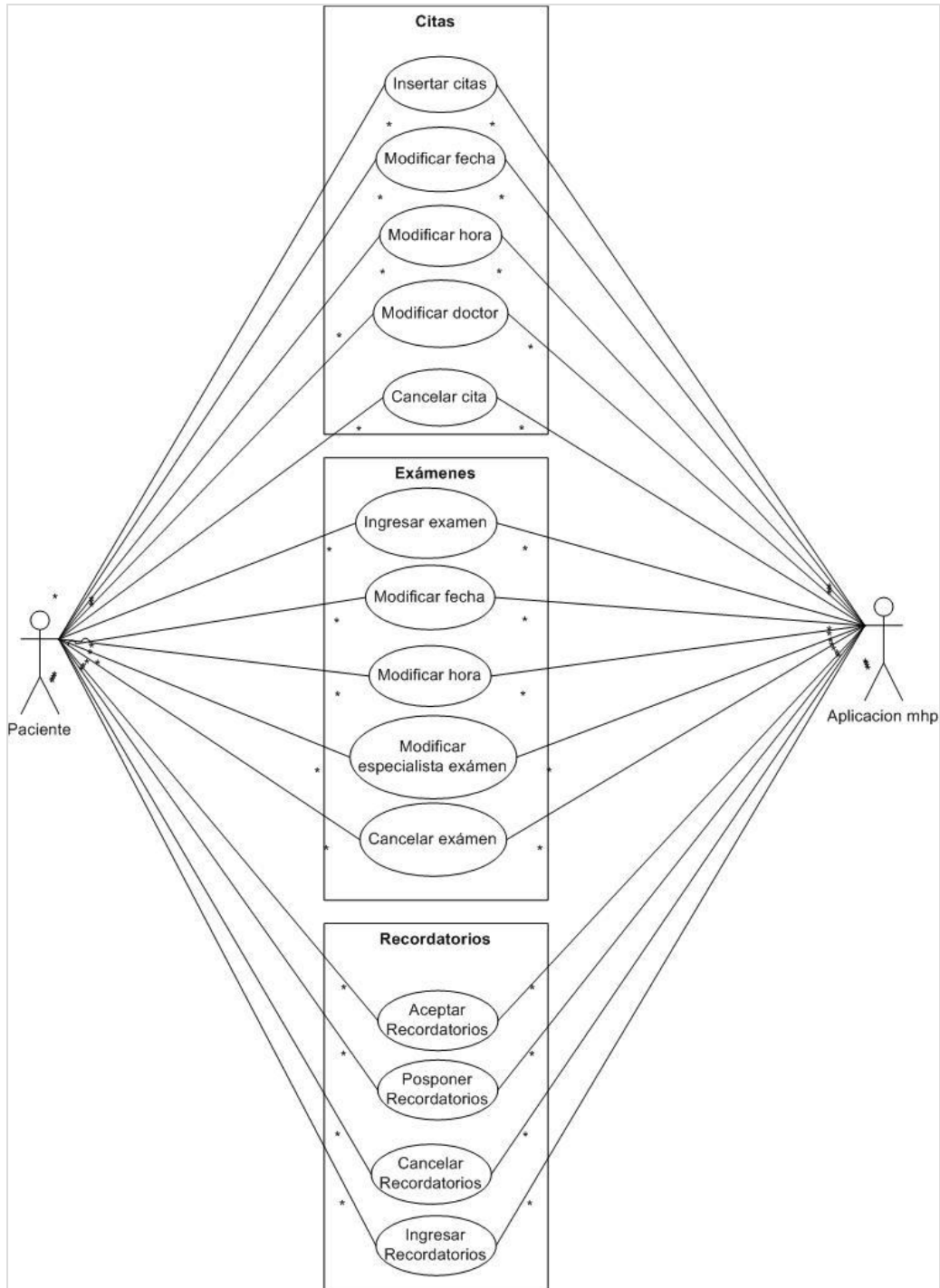


ILUSTRACIÓN 13 CASOS DE USO PACIENTE

Las especificaciones de cada caso de uso pueden verse en el Anexo 10.3

#### 4.4. DIAGRAMAS DE SECUENCIA

##### 4.4.1. APLICACIÓN WEB

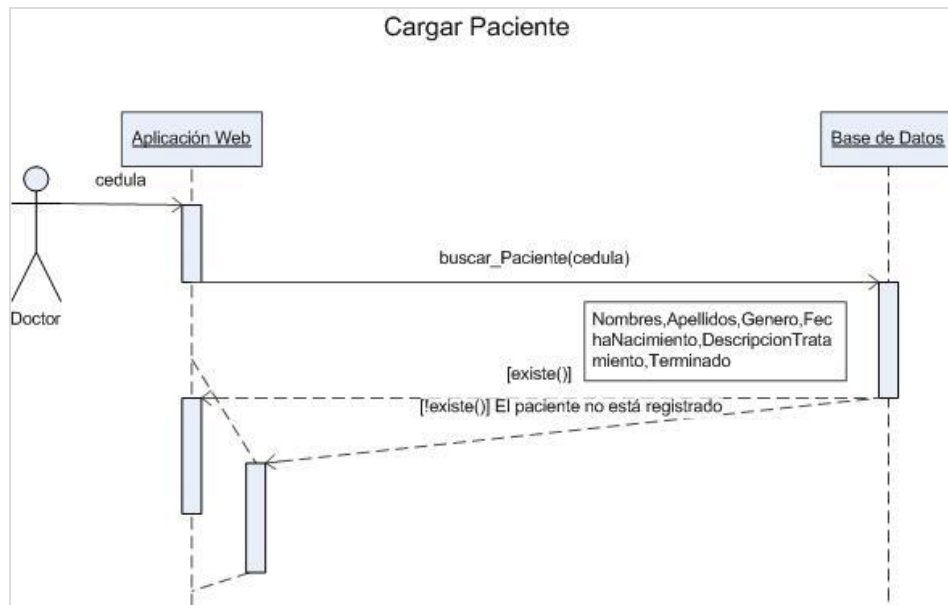


ILUSTRACIÓN 14 SECUENCIA: CARGAR PACIENTE

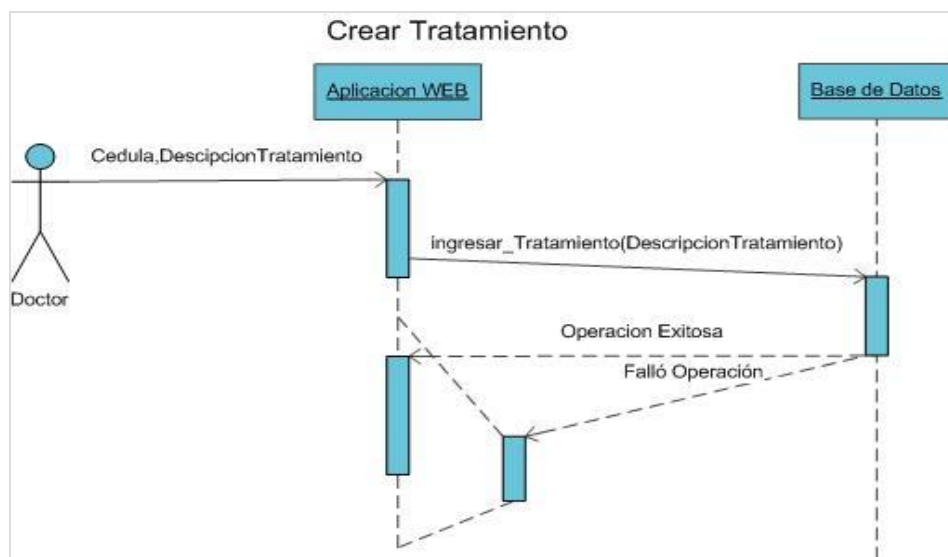


ILUSTRACIÓN 15 SECUENCIA: CREAR TRATAMIENTO

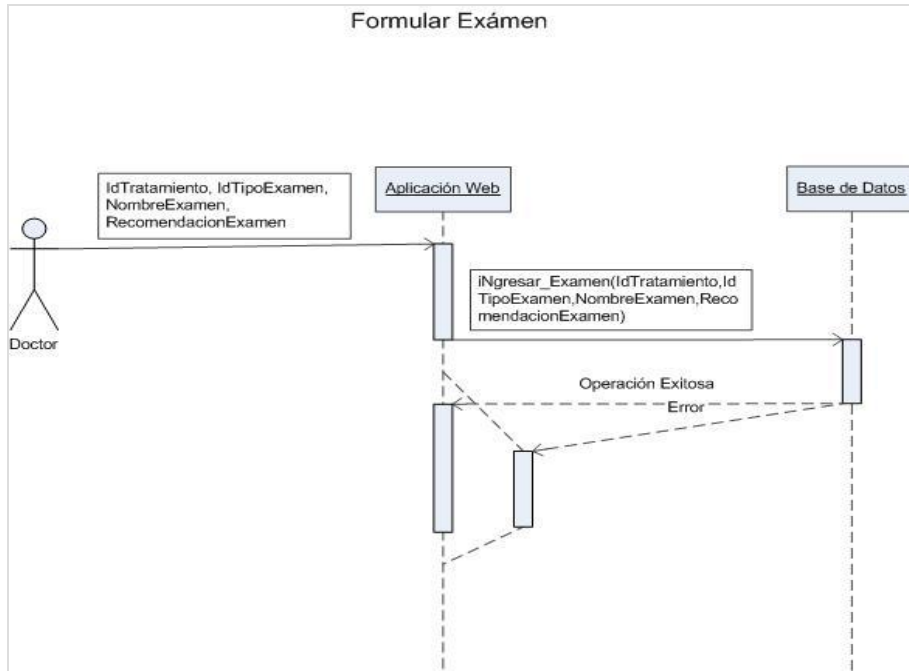


ILUSTRACIÓN 16 SECUENCIA: FORMULAR EXAMEN

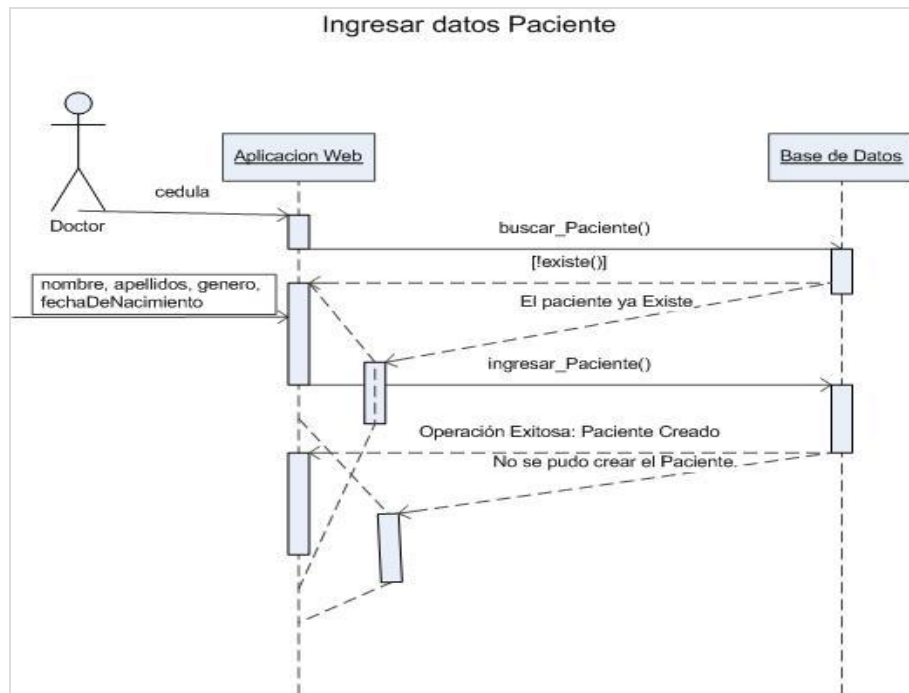
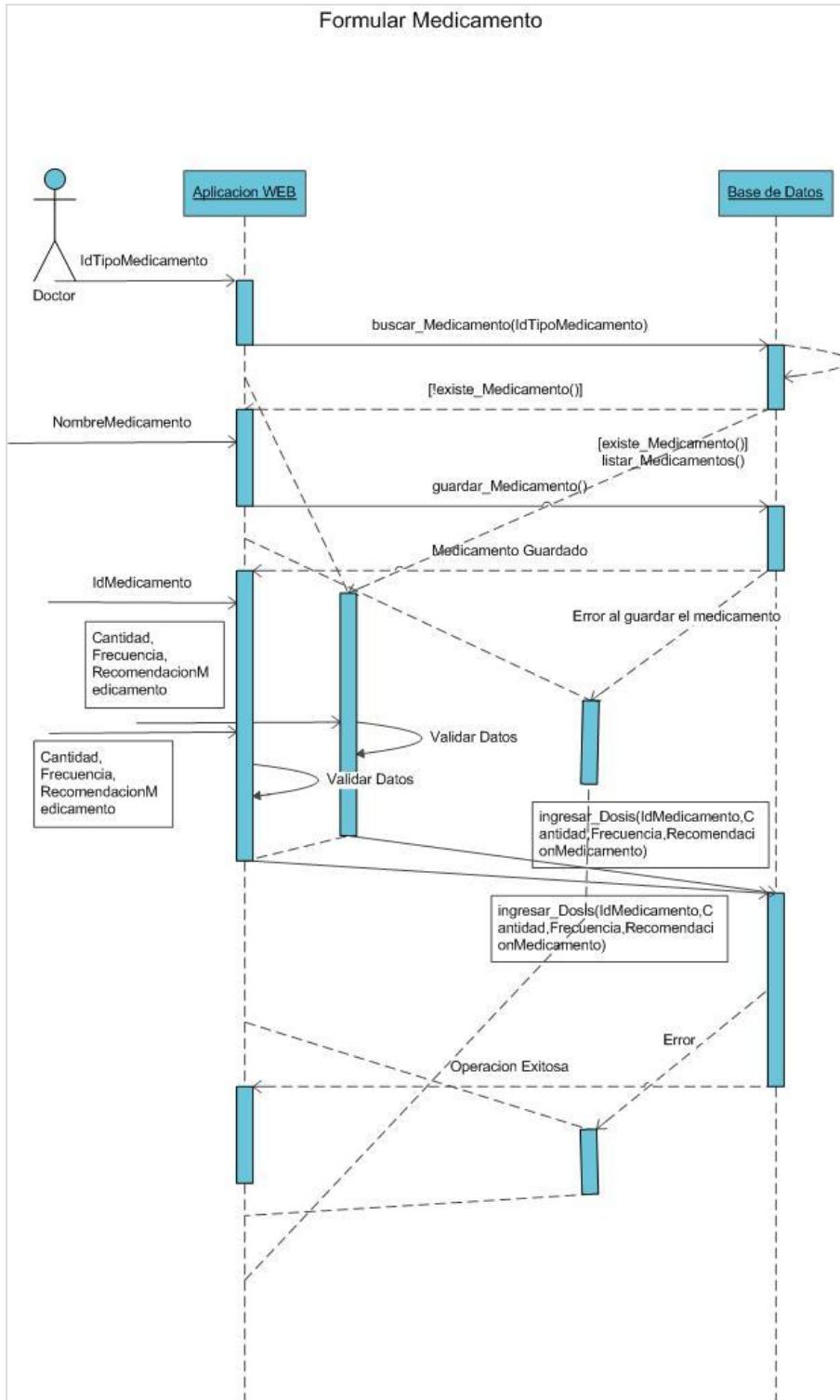
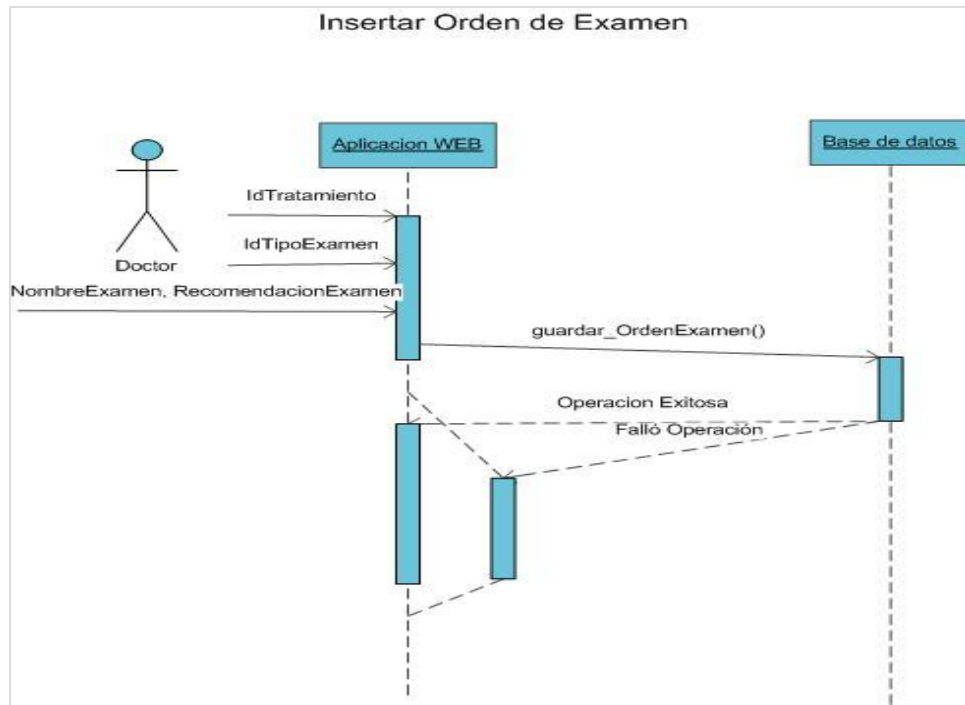


ILUSTRACIÓN 17 SECUENCIA: INGRESAR DATOS PACIENTE

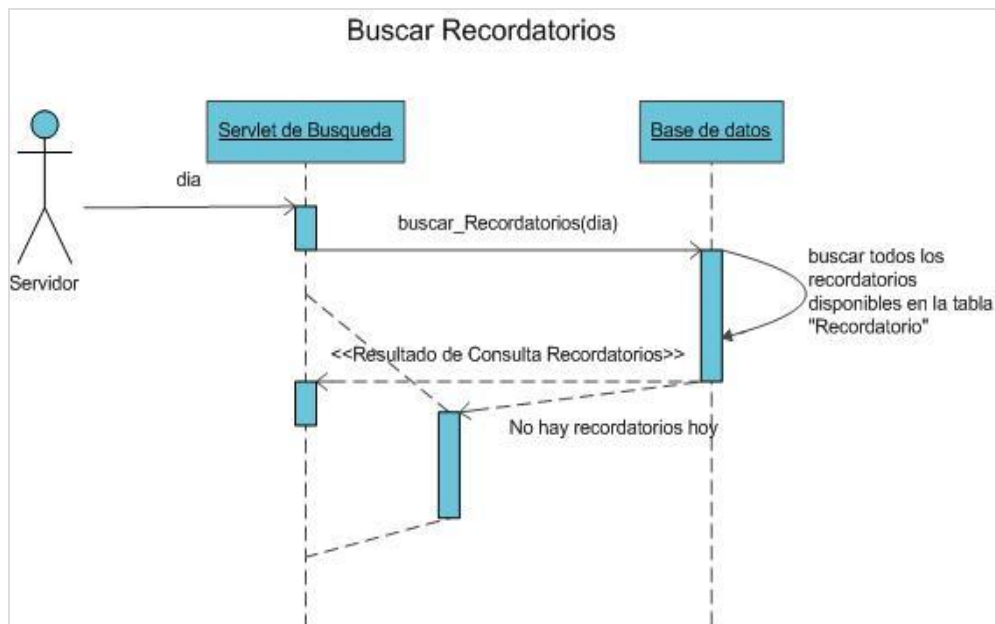


**ILUSTRACIÓN 18 SECUENCIA: FORMULAR MEDICAMENTO**



**ILUSTRACIÓN 19 SECUENCIA: INSERTAR ORDEN DE EXAMEN**

#### 4.4.2. SERVIDOR



**ILUSTRACIÓN 20 SECUENCIA: BUSCAR RECORDATORIOS**

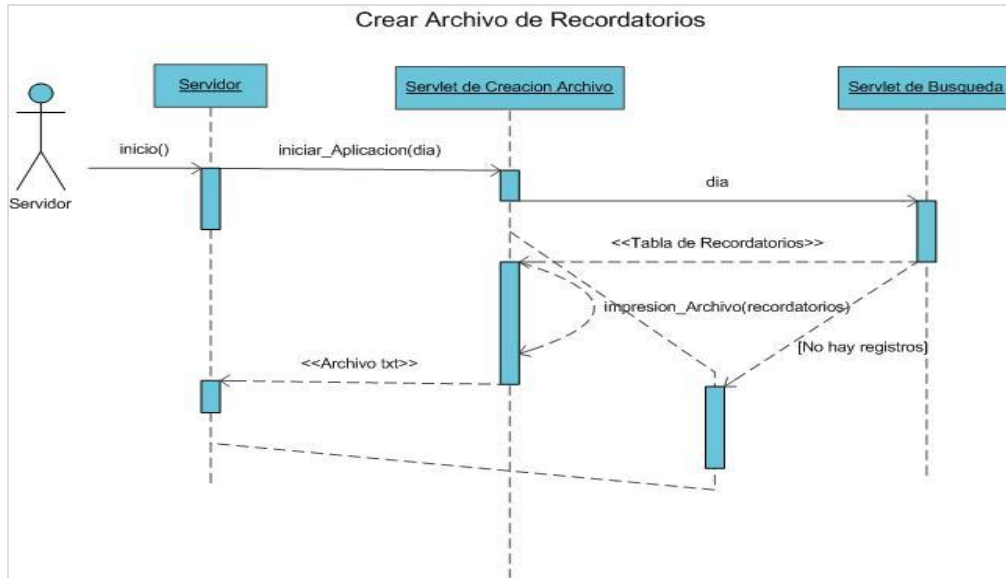


ILUSTRACIÓN 21 SECUENCIA: CREAR ARCHIVO DE RECORDATORIOS

#### 4.4.3. APLICACIÓN MHP

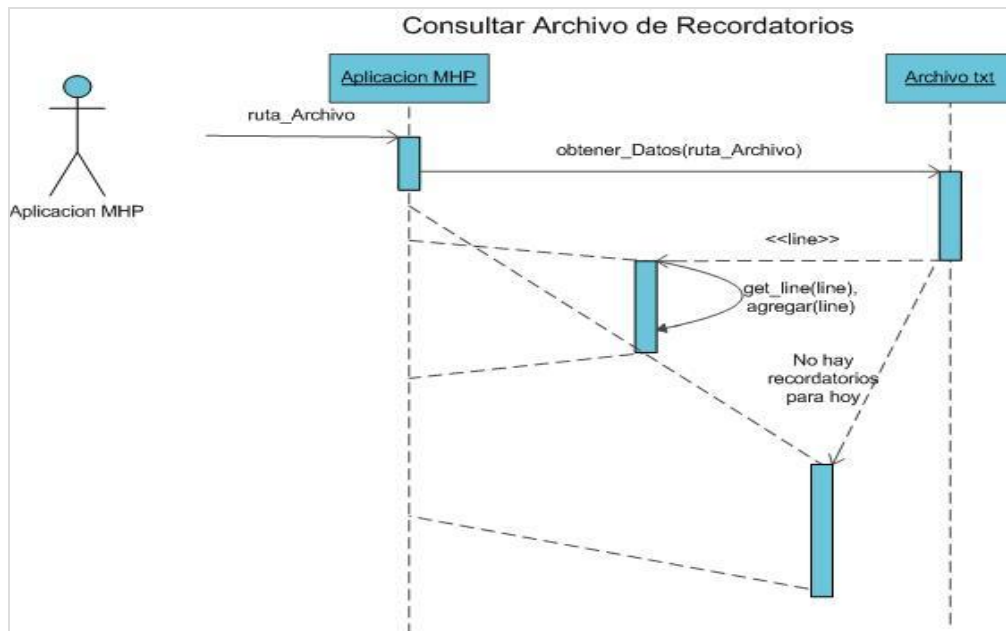
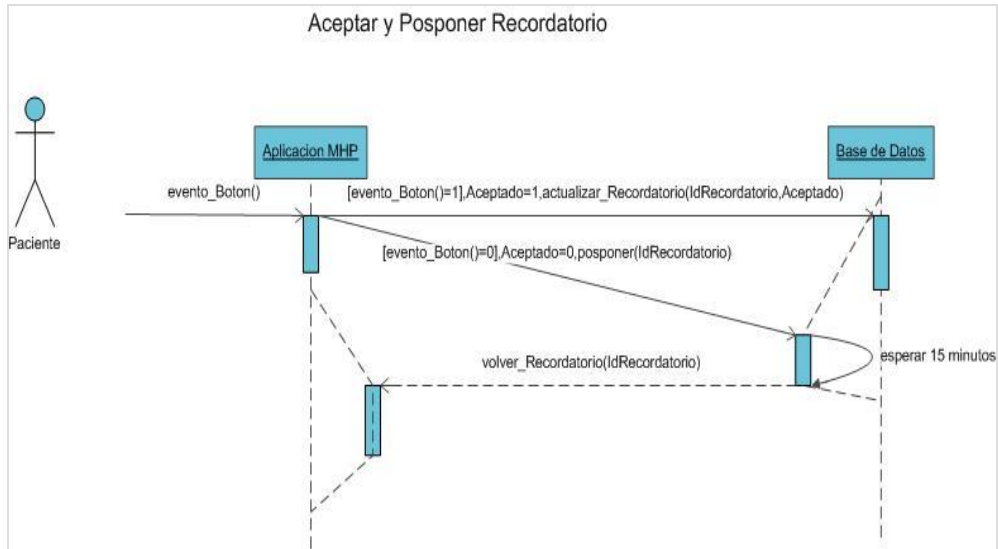
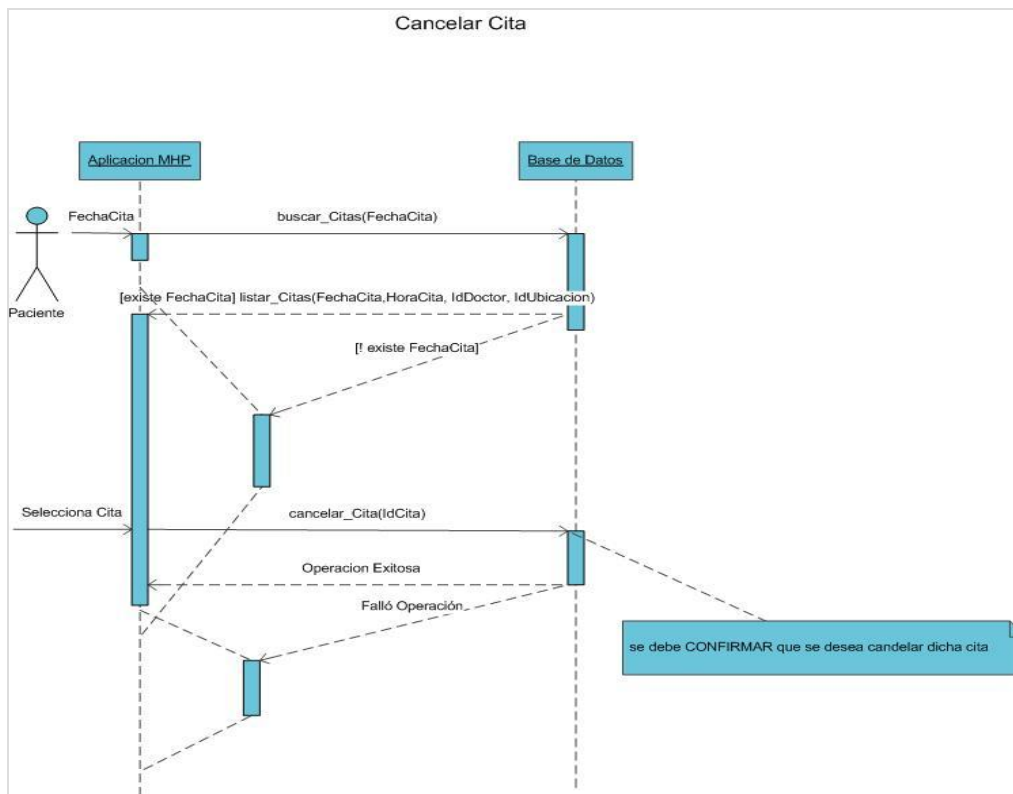


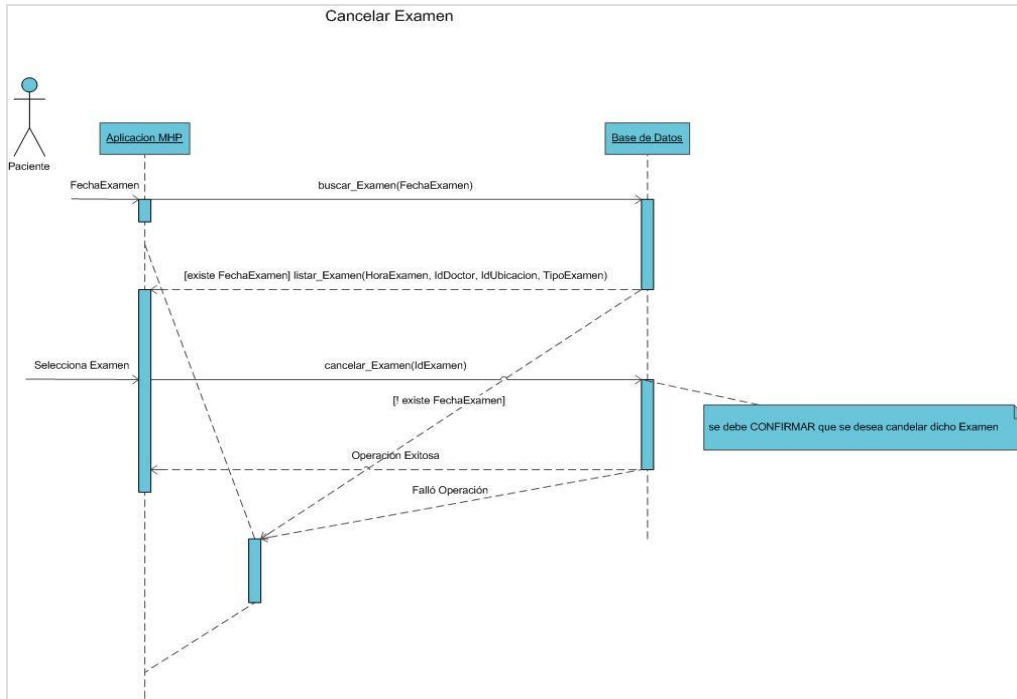
ILUSTRACIÓN 22 SECUENCIA: CONSULTAR ARCHIVO DE RECORDATORIOS



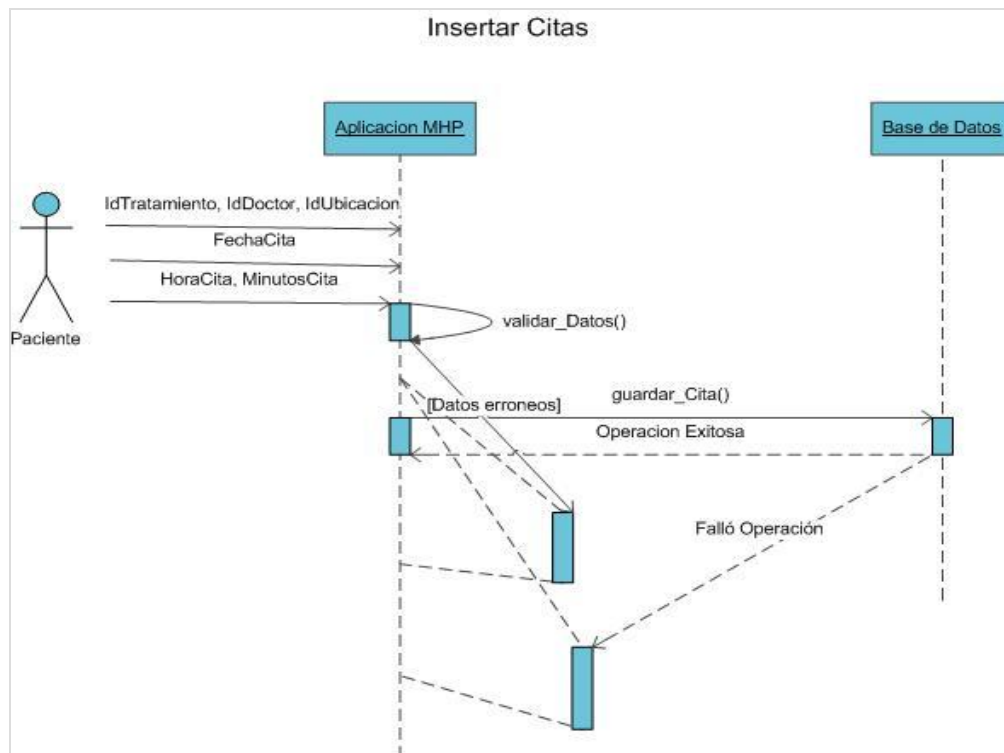
**ILUSTRACIÓN 23 SECUENCIA ACEPTAR Y POSPONER RECORDATORIO**



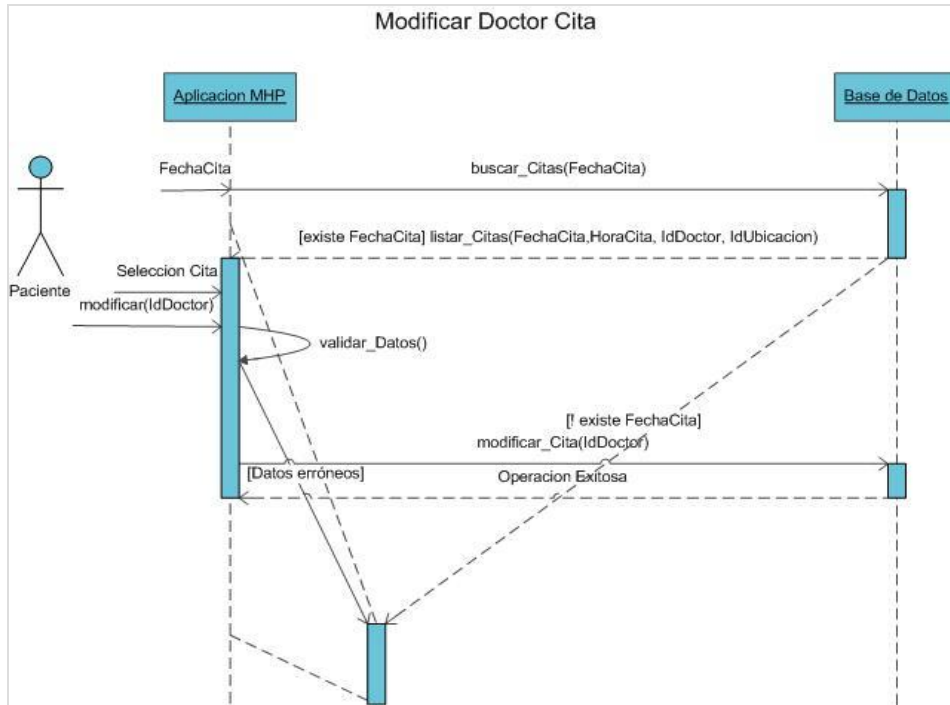
**ILUSTRACIÓN 24 SECUENCIA: CANCELAR CITA**



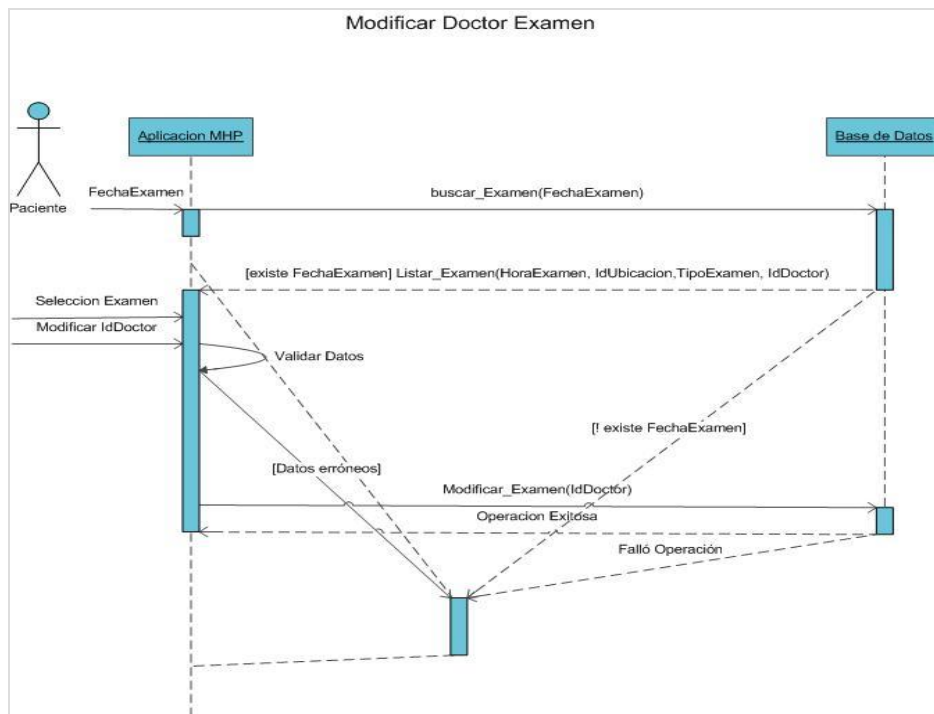
**ILUSTRACIÓN 25 SECUENCIA: CANCELAR EXAMEN**



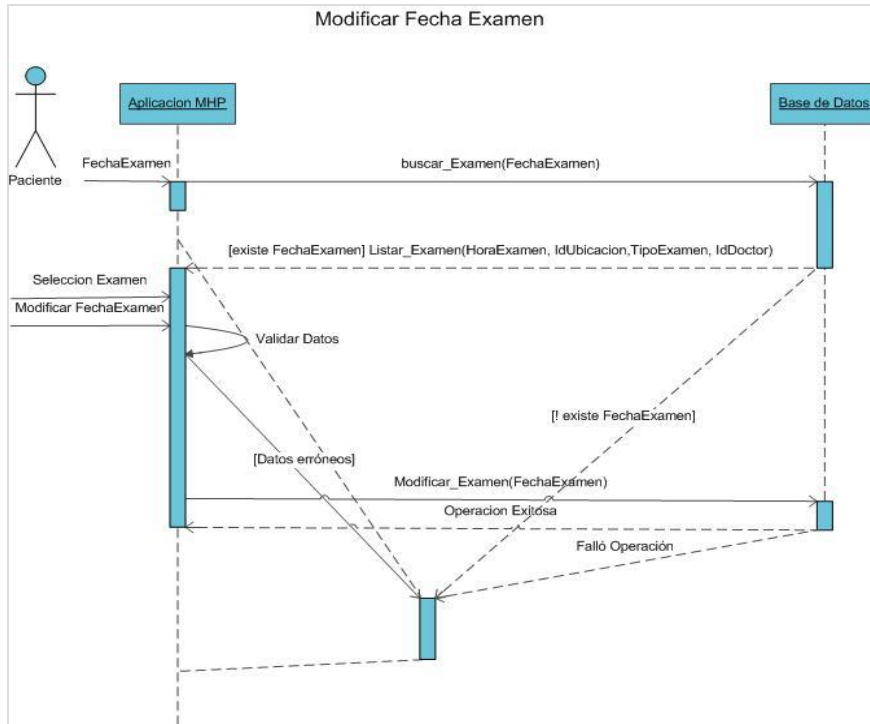
**ILUSTRACIÓN 26 SECUENCIA: INSERTAR CITAS**



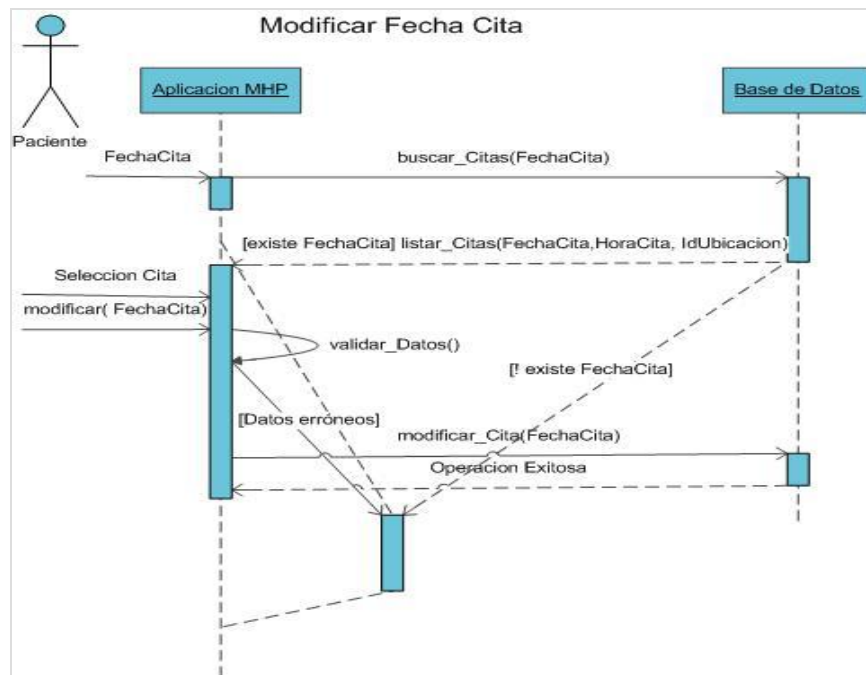
**ILUSTRACIÓN 27 SECUENCIA: MODIFICAR DOCTOR CITA**



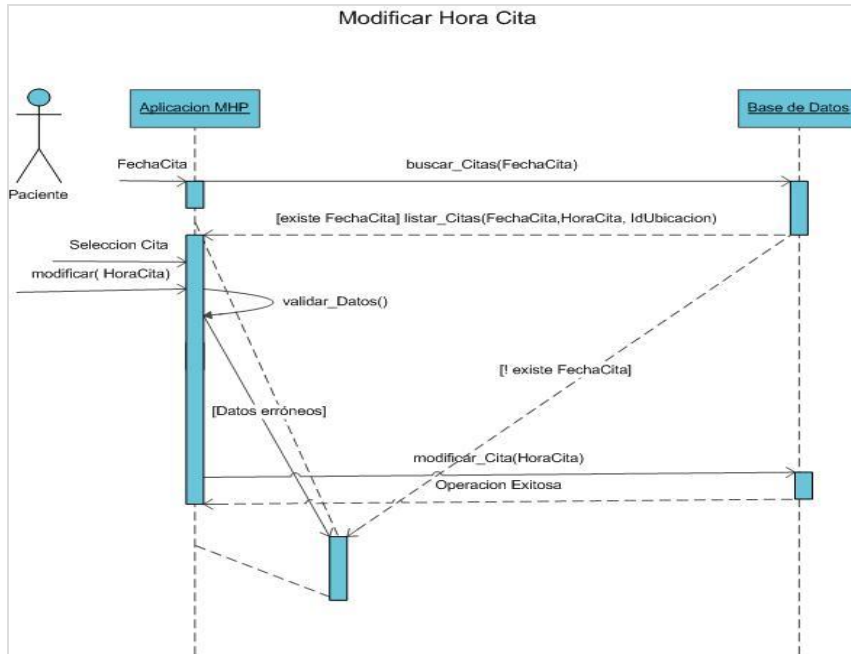
**ILUSTRACIÓN 28 SECUENCIA: MODIFICAR DOCTOR EXAMEN**



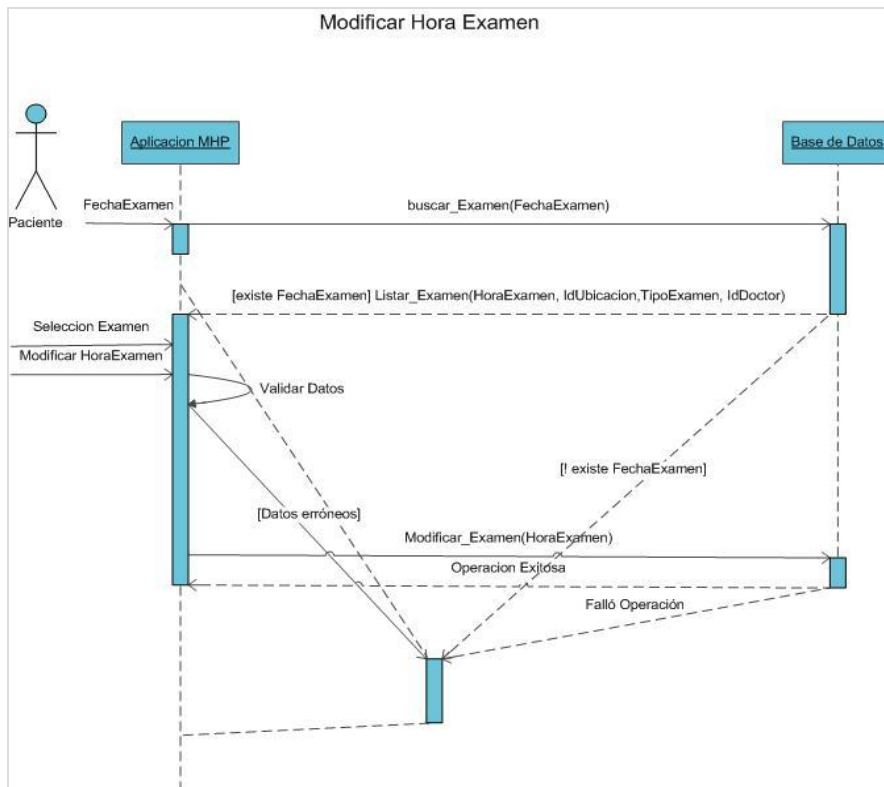
**ILUSTRACIÓN 29 SECUENCIA: MODIFICAR FECHA EXAMEN**



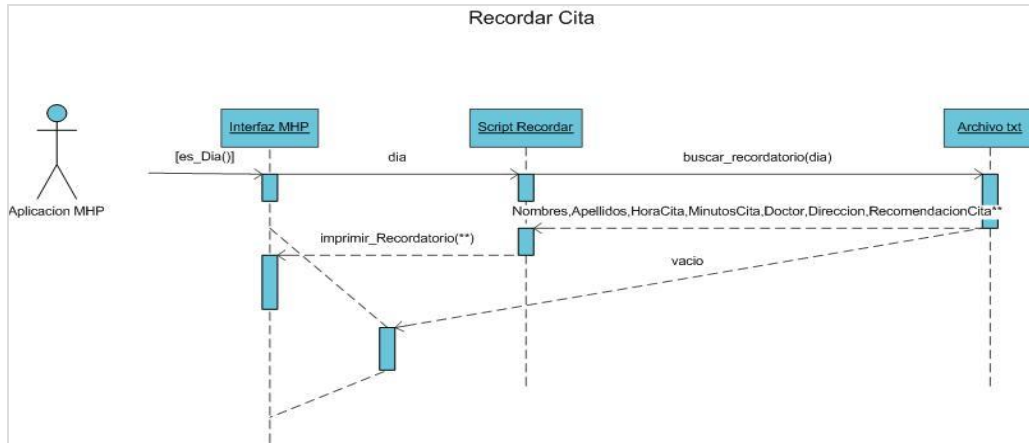
**ILUSTRACIÓN 30 SECUENCIA: MODIFICAR FECHA CITA**



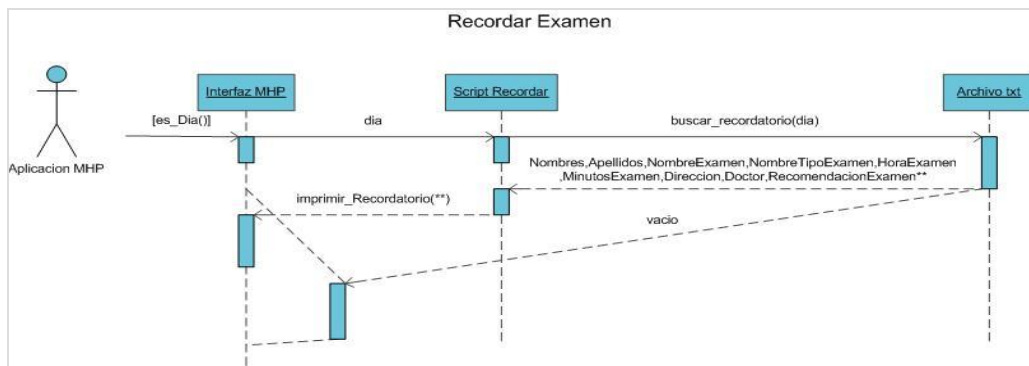
**ILUSTRACIÓN 31 SECUENCIA: MODIFICAR HORA CITA**



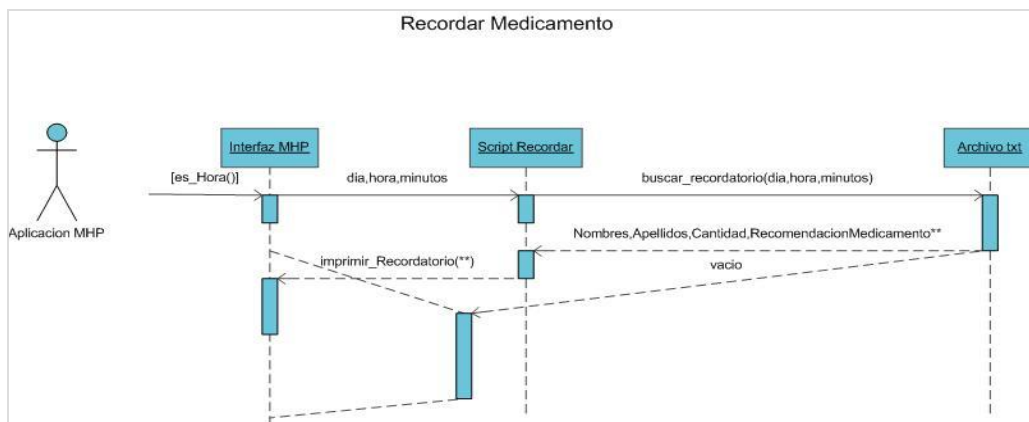
**ILUSTRACIÓN 32 SECUENCIA: MODIFICAR HORA EXAMEN**



**ILUSTRACIÓN 33 SECUENCIA: RECORDAR CITA**



**ILUSTRACIÓN 34 SECUENCIA: RECORDAR EXAMEN**



**ILUSTRACIÓN 35 SECUENCIA: RECORDAR MEDICAMENTO**

## 4.5. DIAGRAMA DE CLASES

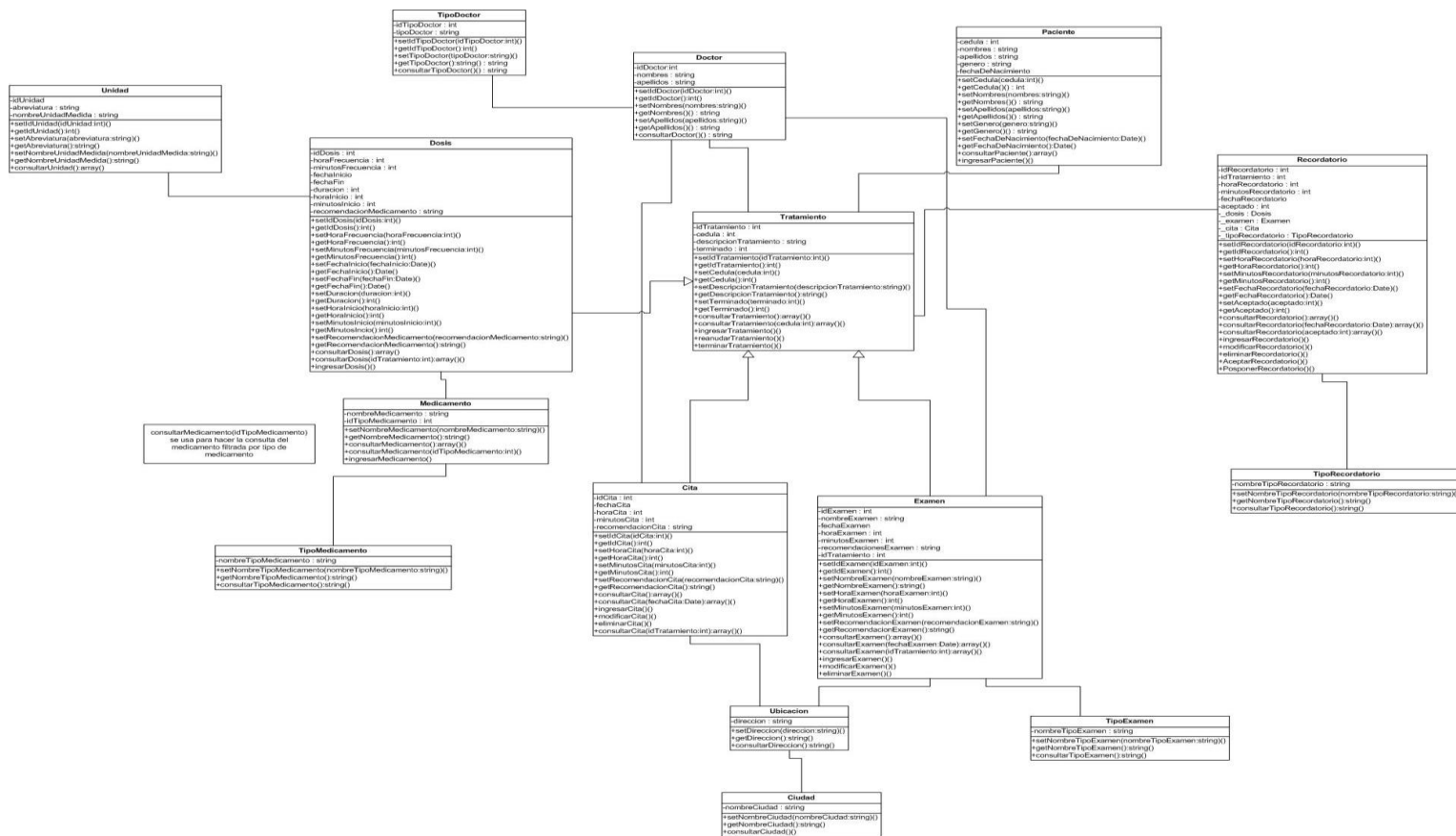


ILUSTRACIÓN 36 DIAGRAMA DE CLASES

## 4.6. MODELO ENTIDAD RELACIÓN

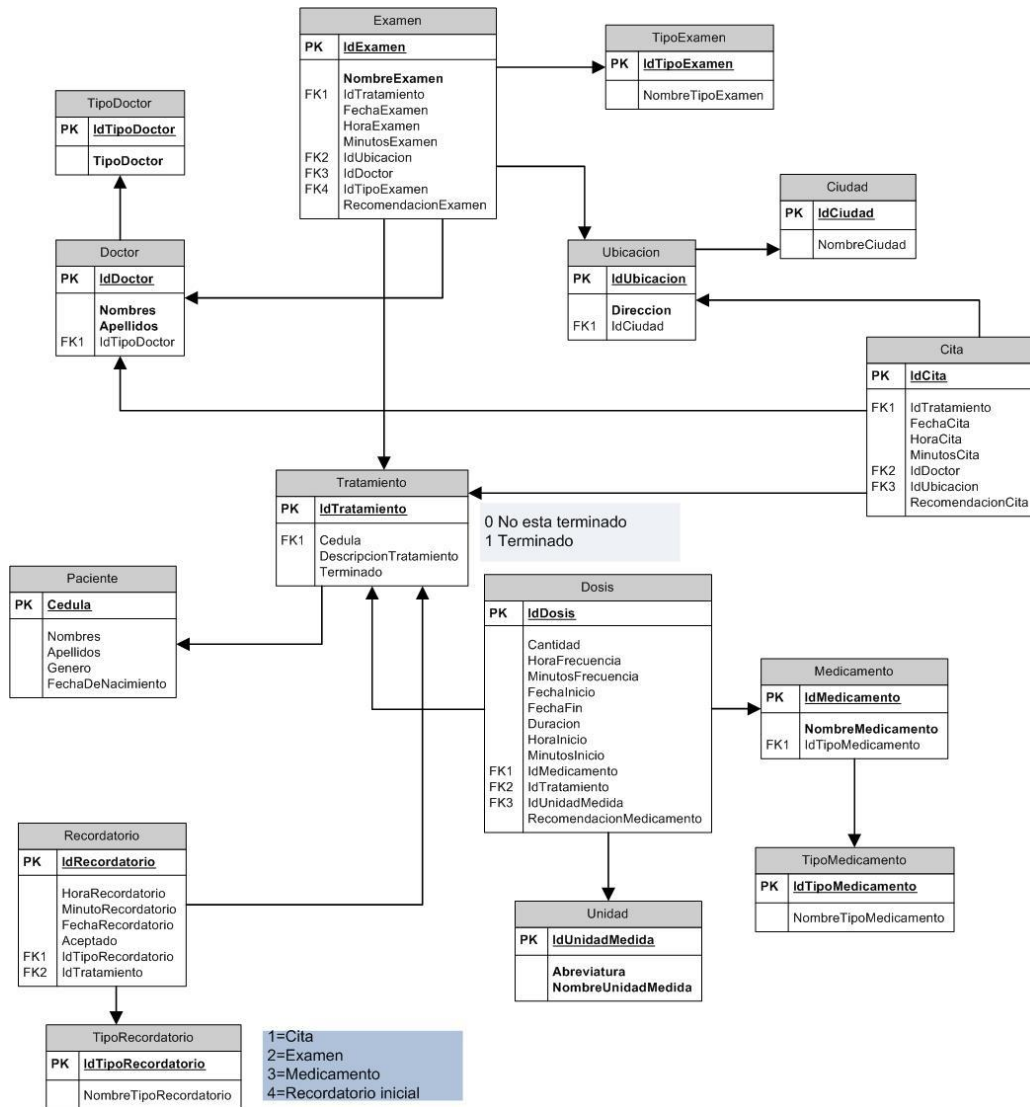


ILUSTRACIÓN 37 MODELO ENTIDAD RELACION

#### 4.7.DISTRIBUCIÓN DEL PROYECTO

Como se comentó en la sección 3.3.4, el prototipo se realizó utilizando iDesigner, el cual permitió codificar la aplicación MHP para luego ser emulada en el iDesigner Player. Una característica muy importante del prototipo es la conexión a una base de datos, para poder acceder a estos datos, la aplicación MHP debe conectarse mediante Servlets al motor de base de datos (en este caso es PostgreSQL). También se desarrolló una pequeña aplicación Web para uso del doctor, tanto los servlets como ésta aplicación se desarrollaron utilizando Netbeans.

La Ilustración 38 Distribución del Proyecto en Netbeans muestra cómo es la distribución del proyecto<sup>26</sup> llamado “prototipo” en Netbeans

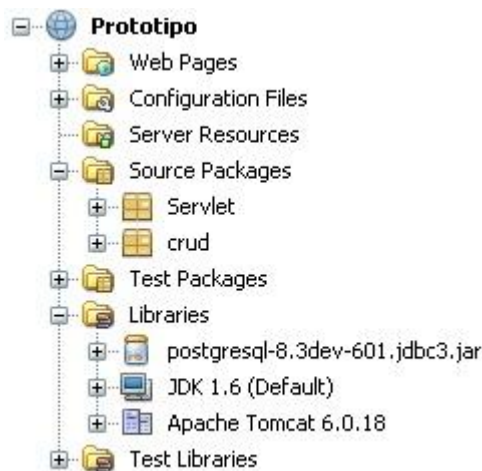
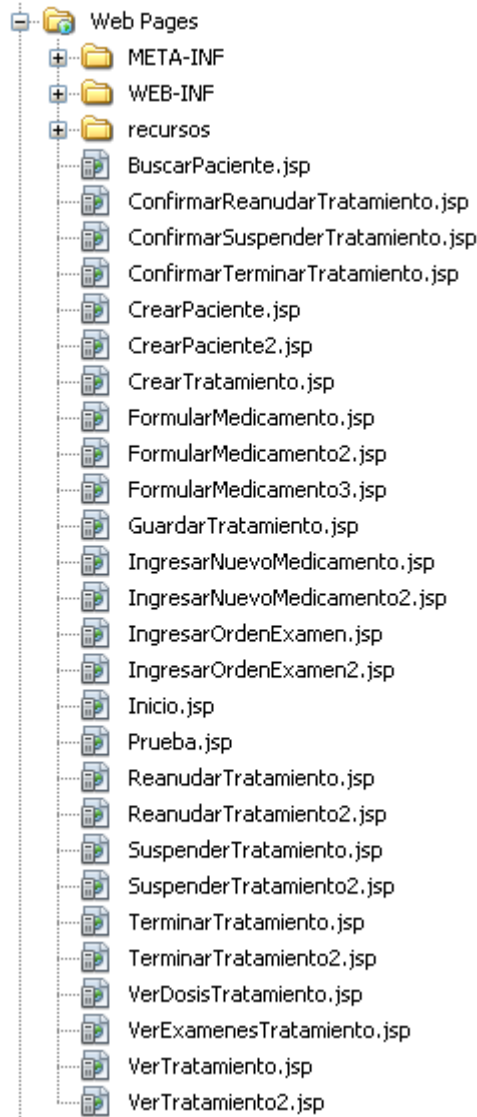


ILUSTRACIÓN 38 DISTRIBUCIÓN DEL PROYECTO EN NETBEANS

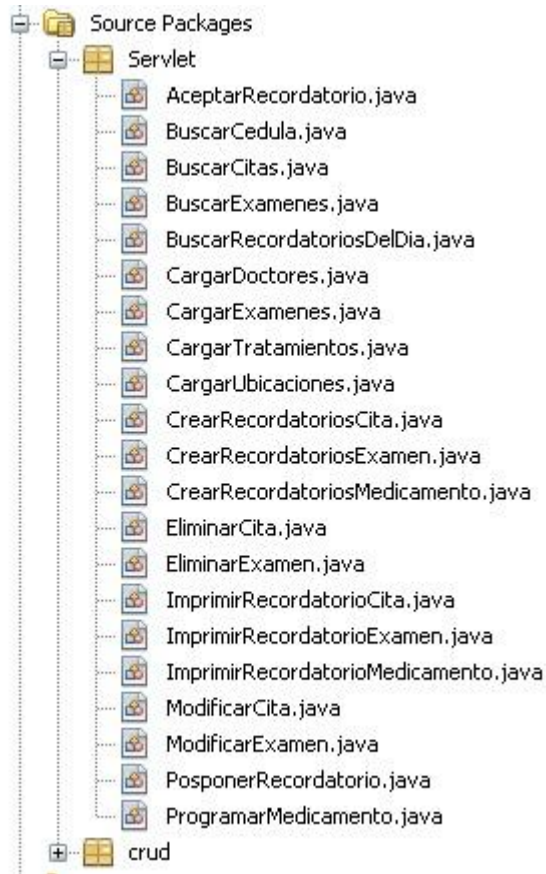
La carpeta Web Pages contiene las páginas JSP que conforman la interfaz de usuario para la aplicación que utilizará el doctor, las cuales se presentan en la Ilustración 39 Aplicación Web .

<sup>26</sup> El proyecto es del tipo Java Web Application.



**ILUSTRACIÓN 39 APLICACIÓN WEB**

Los Servlets que sirven de comunicación entre la base de datos y la aplicación MHP se encuentran en la carpeta “Source Packages” y conforman el paquete denominado “Servlets”, su estructura se muestra en la Ilustración 40.



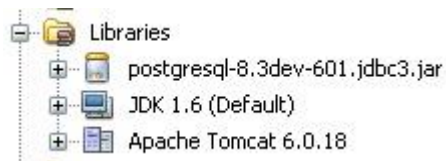
**ILUSTRACIÓN 40 SERVLETS DEL PROTOTIPO**

Dentro de la carpeta “Source Packages” se encuentra el paquete denominado “crud” que contiene las clases del proyecto. Cada clase está asociada a una tabla de la base de datos y contienen métodos para Crear, Leer, Modificar, Eliminar registros de la tabla específica. Esto se puede observar mejor en la Ilustración 41.



**ILUSTRACIÓN 41 CLASES DEL PROYECTO**

Para poder conectarse a la base de datos se necesitan las librerías que se muestran en la Ilustración 42.



**ILUSTRACIÓN 42 LIBRERIAS**

Las librerías “JDK” y “Apache Tomcat” se agregan al crear la aplicación. La librería “postgresql” se debe añadir manualmente.

En la Ilustración 43 se puede observar el estándar de codificación adoptado en el desarrollo del prototipo.

Los nombres de las clases inician con mayúscula (ej. La clase “Cita”).

Las variables inician en minúscula y si son nombres compuestos la siguiente palabra inicia en mayúscula (ej. Variable “idCita”).

Los métodos inician en minúscula y la siguiente palabra en mayúscula (ej. El método “getIdCita”)

```
public class Cita {  
  
    private Integer idCita;  
    private Integer idTratamiento;  
    private Date fechaCita;  
    private Integer horaCita;  
    private Integer minutosCita;  
    private Integer idDoctor;  
    private Integer idUbicacion;  
    private String recomendacionCita;  
    private String mensaje;  
    private String nombrecompleto;  
    private String dir;  
    private Integer hora;  
    private String array;  
  
    public Cita() {...}  
  
    public Date getFechaCita() {...}  
  
    public void setFechaCita(Date fechaCita) {...}  
  
    public Integer getHoraCita() {...}  
  
    public void setHoraCita(Integer horaCita) {...}  
  
    public Integer getIdDoctor() {...}  
  
    public void setIdDoctor(Integer idDoctor) {...}
```

ILUSTRACIÓN 43 EJEMPLO DEL MODELO DE CODIFICACIÓN

En iDesigner, el proyecto se divide por páginas las cuales tienen sus propiedades específicas. La Ilustración 44 muestra los componentes de una aplicación MHP. En la carpeta “pages” se guardan todas las interfaces de la aplicación ( ver Ilustración 45 ), y la carpeta “Templates” contiene los elementos que son comunes para todas las páginas.



ILUSTRACIÓN 44 DISTRIBUCIÓN DEL PROYECTO EN IDESIGNER



ILUSTRACIÓN 45 PÁGINAS DE LA APLICACIÓN MHP

## 5. DESARROLLO DEL PROTOTIPO DE APLICACIÓN MHP

### 5.1. INTRODUCCIÓN

En este capítulo se pretende dar a conocer el proceso de desarrollo del prototipo objetivo del proyecto. Se inicia con un listado de las herramientas software utilizadas en este proceso y se continúa con un breve manual de cómo hacer aplicaciones MHP utilizando tanto herramientas libres (Netbeans y XletView), como la herramienta privada que se escogió para desarrollar el prototipo (iDesigner). También se muestra el resultado del desarrollo del Pastillero Interactivo ejecutándose en el emulador seleccionado.

### 5.2. SOFTWARE GENÉRICO

#### 5.2.1. SERVLET (CEBALLOS, 2006)

Un servlet es un programa que se ejecuta en el contenedor Web de un servidor de aplicaciones. Los clientes lo pueden invocar utilizando el protocolo HTTP. Comparativamente, lo mismo que un applet es cargado y ejecutado por un explorador, un servlet es cargado y ejecutado por un contenedor Web.

La Ilustración 46 funcionamiento de un servlet indica que un servlet acepta peticiones de un cliente, procesa la información relativa a la petición realizada, y le devuelve a éste los resultados que podrán ser mostrados mediante applets, páginas HTML, etc.

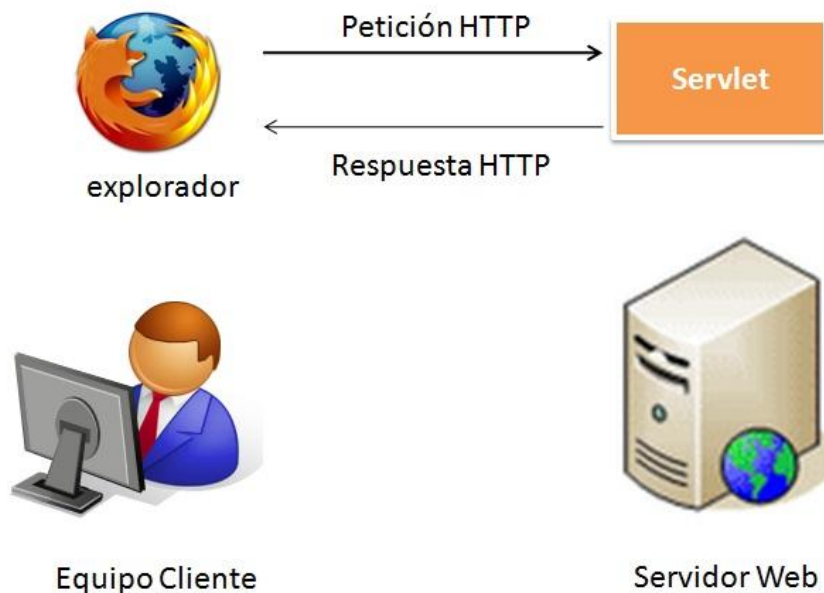


ILUSTRACIÓN 46 FUNCIONAMIENTO DE UN SERVLET

Sin embargo, ésta no es la única función de un servlet; puede realizar también otras tareas como comunicarse con otro servlet para ayudarlo en su trabajo, o bien facilitar el acceso a bases de datos.

Java proporciona el soporte necesario para escribir servlets a través del paquete **javax.servlet**.

#### 5.2.2. JSP (JAVA SERVER PAGES)

Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La

principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web

### 5.2.3. POSTGRESQL

**PostgreSQL** es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD<sup>27</sup>.

Como muchos otros proyectos open source, el desarrollo de **PostgreSQL** no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (**PostgreSQL Global Development Group**).

Algunas de sus principales características son, entre otras:

#### Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

#### Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).

---

<sup>27</sup> La **licencia BSD** es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

### Otras características

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Disparadores (*triggers*): Un disparador o *trigger* se define en una acción específica basada en algo ocuriente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:
  - El nombre del disparador o *trigger*
  - El momento en que el disparador debe arrancar
  - El evento del disparador deberá activarse sobre...
  - La tabla donde el disparador se activará
  - La frecuencia de la ejecución
  - La función que podría ser llamada
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

### Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Algunos de los lenguajes que se pueden usar son los siguientes:

- Un lenguaje propio llamado [PL/PgSQL (similar al PL/SQL de Oracle).
- C.
- C++.

- Java PL/Java web.
- PL/Perl.
- pI PHP.
- PL/Python.
- PL/Ruby.
- PL/sh.
- PL/Tcl.
- PL/Scheme.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser manejados igual a una fila retornada por una consulta.

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados".

#### 5.2.4. TOMCAT

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

#### 5.2.5. NETBEANS

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)

#### 5.2.6. IDESIGNER

Permite crear rápidamente aplicaciones MHP complejas, de la misma forma como las herramientas profesionales para crear aplicaciones y servicios de manera eficiente, está enfocado hacia los radiodifusores, diseñadores y creadores de contenidos.

IDesigner viene empaquetado con un número de componentes -listos para usar- que pueden ser útiles para muchas aplicaciones.

Para crear aplicaciones unidireccionales y de sólo emisión con iDesigner, no es necesario conocimientos avanzados de programación en Java. iDesigner cuenta con plugins, los cuales pueden ser accedidos, puesto que son de código abierto y están documentados. El

conjunto de plugins se denomina “plugin del API iDesigner”. Cualquier desarrollador puede extender la funcionalidad de esta herramienta puesto que permite la adición de plugins desarrollados para aplicaciones específicas.

#### Características Avanzadas

- Soporte para pantalla 16:9
- Generación automática de I-Frame
- Posibilidad de compartir componentes entre las diferentes capas dentro de la aplicación así como a través de aplicaciones
- Permite crear servicios interactivos rápidamente

### 5.3. DESARROLLO DE APLICACIONES MHP CON NETBEANS Y “XLETVIEW”

#### 5.3.1. API’S NECESARIAS PARA DESARROLLAR APLICACIONES MHP<sup>28</sup>

MHP utiliza un número de estándares ya existentes y también define APIs adicionales específicas. A continuación se nombran las principales APIs de Java utilizadas en MHP, que deben ser tenidas en cuenta a la hora de empezar a desarrollar aplicaciones para este estándar.

##### 5.3.1.1. API’S DE NÚCLEO

TABLA 7 API’S DE NÚCLEO

API	Descripción
<b>java.awt</b>	Construye aplicaciones con interfaces de usuario gráficas. Se clasifica de dos formas: <ul style="list-style-type: none"> <li>• Creación de la interfaz</li> <li>• Definición del comportamiento</li> </ul>
<b>java.awt.event</b>	Proporciona interfaces y clases para tratar con diferentes tipos de eventos lanzados por

<sup>28</sup> (Multimedia Home Platforms, 2006) Página 206.

	componentes AWT.
<b>java.awt.image</b>	Este paquete contiene clases e interfaces que permiten crear nuevas imágenes desde cero en vez de cargarlas desde un archivo y modificar las imágenes ya existentes.
<b>java.beans</b>	Contiene clases relacionadas al desarrollo con Java Beans <sup>29</sup> .
<b>java.io</b>	Proporciona el sistema de entrada y salida a través del flujo de datos, la serialización y los archivos del sistema.
<b>java.lang</b>	Proporciona las clases fundamentales del diseño del lenguaje de programación Java.
<b>java.lang.reflect</b>	Proporciona clases e interfaces para obtener información reflexiva sobre clases y objetos
<b>java.net</b>	Proporciona las clases para implementar aplicaciones de red.
<b>java.security</b>	Proporciona las clases e interfaces para el marco de seguridad.
<b>java.security.cert</b>	Proporciona clases e interfaces para el análisis y el manejo de certificados.
<b>java.util</b>	Contiene los marcos de colecciones, clases de colección de legado, modelo de evento, facilidades para fecha y tiempo, internacionalización, y diversas clases de utilidades (generador de números randómicos, arreglo de bits).
<b>java.util.zip</b>	Proporciona clases para la lectura y escritura para los formatos de archivo estándar ZIP y GZIP.

---

<sup>29</sup> Modelos de componentes para la construcción de aplicaciones en Java. Se usan para encapsular varios objetos en un único objeto (llamado bean), para hacer uso de un solo objeto en lugar de varios más simples.

### 5.3.1.2. API'S DE JAVA MEDIA FRAMEWORK

Estas APIs son utilizadas para controlar cómo los contenidos de audio y video son decodificados y mostrados. Si bien esto se usa principalmente para controlar lo que se transmite en forma de vídeo (y cómo se presenta), las aplicaciones pueden usar JMF para reproducir archivos de audio o mostrar un tipo especial de formato de imagen.

JMF 1.0 fue originalmente diseñado para controlar flujo de audio y video desde un archivo de disco local o desde un servidor de archivos de video sobre una red. Para los flujos de emisión, JMF fue extendido por DAVIC con su paquete **org.davic.media**. EL JMF 1.0 utilizado para televisión interactiva contiene los siguientes paquetes:

- java.media
- java.media.protocol

### 5.3.1.3. API'S DE JAVA TV

TABLA 8 API'S DE JAVA TV

API	Descripción
<b>javax.tv.graphics</b>	Añade algunos soportes a AWT para cuestiones específicas de los Televisores como la mezcla video-gráficos y descubrir el contenedor principal de la aplicación.
<b>java.tv.locator</b>	Proporciona los mecanismos para referencias como URL de transmisión de servicios y clips multimedia.
<b>javax.tv.media</b>	Añade soporte para funcionalidades específicas del TV a JMF
<b>javax.tv.media.protocol</b>	Agrega soporte para protocolos de emisión de flujos a JMF
<b>javax.tv.net</b>	Proporciona mecanismos para el acceso a datagramas IP contenidos en una emisión de

	flujo.
<b>javax.tv.service</b>	Conceptos de alto nivel para describir servicios de TV digital. Este paquete también proporciona el mecanismo básico para consultar la información del servicio del servicio de radiodifusión.
<b>javax.tv.service.guide</b>	Soporte para aplicaciones del tipo EPG (Electronic Program Guide), incluyendo conceptos para describir eventos individuales, cronograma de programas e información de rating.
<b>javax.tv.service.navigation</b>	Soporte para servicios <sup>30</sup> navegables. Esto incluye soporte para listas de servicios favoritos, componentes dentro de los servicios y tipos de servicios.
<b>javax.tv.service.selection</b>	Conceptos que definen cómo los servicios serán presentados al usuario, y cómo estos nuevos servicios van a ser seleccionados. También describe los conceptos para presentar varios servicios a la vez.
<b>javax.tv.service.transport</b>	Conceptos que describen los mecanismos de transporte usados por un servicio de TV digital, como un flujo de transporte, redes de difusión y ramas.
<b>Javax.tv.util</b>	Clases de utilidades para aplicaciones JavaTV, incluyendo la administración de timers y eventos timer.
<b>Javax.tv.xlet</b>	Modelo de ciclo de vida de las aplicaciones y clases de soporte.

---

<sup>30</sup> Aquí se define un canal de televisión como un servicio.

#### 5.3.1.4. API'S DE DAVIC (DIGITAL AUDIO VISUAL COUNCIL)

DAVIC fue una organización creada para promover las aplicaciones audiovisuales mediante estándares abiertos. Se basó en MHEG y agregó una API de Java ofreciendo mejor control de un receptor para una aplicación. Muchos participantes en DAVIC se unieron a DVB para su trabajo en MHP.

Hoy en día, en MHP, hay varios paquetes originarios del trabajo en DAVIC. Estos paquetes le dan a una aplicación mayor control en la selección y presentación del audio y el video. DAVIC también proporciona un marco de notificación de escasos recursos, el cual es utilizado de forma consistente en MHP.

Los paquetes de DAVIC que utilizan las aplicaciones MHP son:

TABLA 9 API'S DE DAVIC

API	Descripción
<b>org.davic.media</b>	Incluye extensiones de JMF relevantes para los contenidos de TV. Lo más importante es el control de selección del idioma para el audio y los subtítulos.
<b>Org.davic.mpeg</b>	Representa conceptos del sistema MPEG a través del flujo de transporte, los servicios y las clases del Flujo Elemental.
<b>Org.davic.mpeg.dvb</b>	Contiene versiones extendidas de las clases con información adicional específica de DVB.
<b>Org.davic.mpeg.sections</b>	Proporciona la habilidad de filtrar secciones MPEG específicas de un flujo de transporte.
<b>Org.davic.net</b>	Trata con referencias de contenido por medio de la clase Locator.
<b>Org.davic.net.dvb</b>	Esta clase es más específica para DVB.
<b>Org.davic.net.tuning</b>	Maneja la sintonización explícitamente entre flujos de

	transporte. Sin embargo esto no es necesario cuando los servicios se conmutan.
<b>Org.davic.resources</b>	Contiene el marco de notificación de recursos adoptado en MHP. Cada recurso escaso en un dispositivo implementa en sí el régimen de notificación, pero debe estar basado en las interfaces encontradas en este paquete.

#### 5.3.1.5. API'S DE HAVI (HOME AUDIO VIDEO INTEROPERABILITY)<sup>31</sup>

HAVI ha definido un extenso conjunto de paquetes basados en Java para dispositivos electrónicos de consumo. Una parte de HAVi que ha sido adoptado por los estándares que soportan Java, incluyen paquetes de interfaz de usuario (UI). Este conjunto de paquetes requieren de un subconjunto de paquetes del estándar java.awt para una definición más completa de HAVi. La interfaz de usuario de HAVi contiene los siguientes paquetes:

- Org.havi.ui
- Org.havi.ui.event

#### 5.3.1.6. API'S DE DVB

TABLA 10 API'S DE DVB

API	Descripción
<b>org.dvb.application</b>	Proporciona acceso a las listas de aplicaciones, las cuales están disponibles en su contexto y la habilidad de ejecutar esas aplicaciones
<b>Org.dvb.dsmcc</b>	Prevé ampliar el acceso a los archivos

<sup>31</sup> La especificación HAVI define un conjunto de APIs y middleware capaz de detectar automáticamente dispositivos en la red, coordina las funcionalidades de varios dispositivos, instala aplicaciones y software para interfaz de usuario en cada artefacto, y asegura la interoperabilidad entre múltiples marcas de dispositivos. (HAVI)

	transportados en la emisión del flujo. Esta API también proporciona acceso a objetos como flujos y eventos de flujo, y descriptores de evento de flujo y descriptores NPT. Una de las clases más importantes es DSMCObject. Instancias de este objeto representa objeto en un carrusel DSM-CC <sup>32</sup> . Los DSMCObject pueden ser usados directamente o utilizando métodos java.io.File.
<b>Org.dvb.event</b>	Proporciona acceso a eventos de entrada de usuario antes de que sean procesados a través del mecanismo de eventos del paquete java.awt. Una clase que implementa UserEventListener puede recibir eventos de entrada aunque la aplicación no tenga el foco de entrada de usuario.
<b>Org.dvb.io</b>	Proporciona soporte para la comunicación entre aplicaciones y una extensión del paquete java.io para acceder a archivos que se mantienen en almacenamiento persistente.
<b>Org.dvb.io.ixc.IxcRegistry</b>	Clase que permite obtener una referencia a un objeto remoto que reside en un Xlet diferente pero que se encuentra en el mismo terminal MHP.
<b>Org.dvb.io.persistent</b>	Proporciona acceso y atributos a archivos en almacenamiento persistente.
<b>Org.dvb.lang</b>	Proporciona aquellas características relacionadas con la plataforma central que no se encuentran en el paquete java.lang. La clase DVBCClassLoader es usada para cargar clases y recursos desde una ruta de búsqueda de URLs que refiere a los lugares donde los archivos de clases Java pueden estar almacenados.

<sup>32</sup> DSM-CC (Digital Storage Media Command and Control). Formato para la transmisión de datos e información de control en una sección privada de MPEG-2.

<b>Org.dvb.media</b>	Proporciona extensiones específicas DVB para JMF. Los Players <sup>33</sup> que están presentando video en un HVideoComponent para controlar cómo el video es recortado usan la interface org.dvb.media.VideoPresentationControl. El org.dvb.media.BackgroundVideoPresentationControl solamente se usa con video presentado en plano de video. La interface puede ser usada para poner el tamaño, la posición, y cortar una región. Org.dvb.media.VideoFormatControl proporciona una forma para transformar la señal entrante de acuerdo al formato definido en el DFC (Decoder Format Conversion).
<b>Org.dvb.net</b>	Proporciona características generales de las redes que no se encuentran en otros paquetes, también las extensiones para el API de acceso condicional de DAVIC, administración de sesión para conexiones IP bidireccionales las cuales tienen sesiones basadas desde el punto de vista de una aplicación y extensiones del API de ajuste de DAVIC.
<b>Org.dvb.si</b>	Proporciona acceso a los servicios de información DVB (DVB SI).
<b>Org.dvb.test</b>	Permite probar las aplicaciones registrando mensajes durante su ejecución e indica su condición de terminación de manera independiente de la plataforma.
<b>Org.dvb.ui</b>	Proporciona funcionalidad gráfica ampliada.
<b>Org.dvb.user</b>	Proporciona acceso a configuraciones y preferencias realizadas por el usuario final.

Varias APIs pueden ser utilizadas para la misma característica. A continuación se presentan las principales características y las APIs disponibles para cada una.

---

<sup>33</sup> Elemento importante en JMF. Es responsable de decodificar y ejecutar una pieza multimedia. Cualquier Player en un sistema JMF tiene subclases de la clase javax.media.Player.

Control de Gráficas:

AWT

HAVI

DVB UI

Acceso a Sistemas de Información:

JavaTV

DVB SI

Selección de Servicio (Canal):

JavaTV Service Selection

JMF

Canal de Retorno:

Org.dvb.net.rc

Control de Audio/Video:

JMF

DAVIC

Administración de recursos por una aplicación:

Org.davic.resource

API para el control del sintonizador:

DAVIC

5.3.2. J2ME: CONFIGURACIÓN Y PERFILES DE JAVA PARA LAS  
APLICACIONES MHP

La funcionalidad de un dispositivo específico definida en J2ME está basada en los siguientes conceptos:

5.3.2.1. CONFIGURACIÓN

Define la JVM, el lenguaje básico que soporta, y las clases fundamentales de una amplia gama de dispositivos, como lo son todas las terminales móviles wireless o todos los dispositivos stb (decodificadores). Un dispositivo específico debe encontrarse en los requerimientos de al menos una configuración, CLDC o CDC, para calificarlo como un dispositivo JME.

#### 5.3.2.2. PERFIL

Los perfiles tienen APIs esenciales para construir aplicaciones fundamentales para una familia de dispositivos. El perfil más conocido es el MIDP, el cual le da poder a los teléfonos móviles y se sitúa sobre CLDC. Igualmente importante es el Foundation Profile (FP), el cual es análogo al MIDP para ofrecer clases de apoyo en la aplicación. A diferencia de los dispositivos basados en CLDC, en los dispositivos basados en CDC típicamente se incluyen ya sea el Personal Basic Profile(PBP) o el Personal Profile(PP) o ambos, para proporcionarle a las aplicaciones soporte para interfaces de usuario.

#### 5.3.2.3. PAQUETE

Un paquete, es un objeto o grupo de objetos en un contenedor común. Tan importante como son las plataformas y los perfiles de la modularidad de la plataforma JME, los paquetes son posiblemente la llave del continuo éxito de JavaME.

La Ilustración 47 muestra cómo estas abstracciones se unen para definir las características software de un dispositivo.



ILUSTRACIÓN 47 RELACIÓN ENTRE CONFIGURACIONES, PERFILES Y PAQUETES EN JME

#### 5.3.2.4. CONFIGURACIÓN Y PERFILES PARA APLICACIONES MHP

##### 5.3.2.4.1. CDC (CONNECTED DEVICE CONFIGURATION)

La configuración CDC es para aquellos dispositivos que tienen mayores capacidades como los PDAs high-end, set-top boxes o decodificadores. Esta configuración es la que se va a utilizar para el desarrollo de las aplicaciones interactivas para TDT.

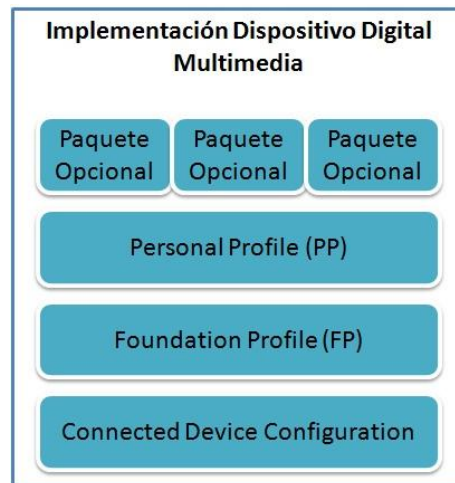


ILUSTRACIÓN 48 RELACIÓN ENTRE CONFIGURACIÓN, PERFIL Y PAQUETE EN UN DECODIFICADOR U OTRO HARDWARE MULTIMEDIA

Los perfiles asociados a esta configuración son los siguientes:

- PP (PERSONAL PROFILE)

Soporta dispositivos con restricciones de recurso con herramientas de interfaz de usuario (GUI) basadas en AWT. Proporciona un ambiente de aplicación J2ME completo para productos de consumo y dispositivos embebidos. Las características principales del PP son:

- Compatibilidad completa AWT
- Soporte para el modelo de programación de aplicaciones "applet".
- Una ruta de migración para tecnologías heredadas como PersonalJava
- Todas las APIs incluidas en el PBP

- Personal Profile es adecuado para productos con diseño de escenarios que requieren compatibilidad completa con AWT y soporte para applet.
- PBP (PERSONAL BASICS PROFILE)

Proporciona dos jerarquías de clases para las aplicaciones: el modelo applet y una nueva jerarquía para dispositivos multimedia que define el modelo de programación Xlet. PBP define un subconjunto de AWT para desarrollo GUI, incluyendo soporte para `java.awt.Window`, `java.awt.Frame`, `java.awt.Component`, `java.awt.Container`. No define controles tradicionales AWT, incluyendo botones, listas, y otros, dado que pueden tener conexiones a componentes importantes de la máquina virtual. Sin embargo, se pueden crear componentes propios o importar un paquete sobre una plataforma específica que proporcione los componentes que se necesiten. También incluye clases que soporta y facilita la comunicación entre dos Xlets que se ejecutan en un mismo dispositivo.

### 5.3.3. HERRAMIENTAS SOFTWARE NECESARIAS

Para poder desarrollar una aplicación MHP con Netbeans y XletView se necesita incluir lo siguiente:

- Tener instalada la Configuración CDC<sup>34</sup>.
- Descargar el paquete de JavaTV en (Microsystems, JavaTv)
- Descargar el paquete de XletView disponible en (sourceforge)
- Demás herramientas de java para poder ejecutar Netbeans.

### 5.3.4. EJEMPLO “HOLA MUNDO” CON NETBEANS Y XLETVIEW

A continuación, se dará una introducción, por medio de un ejemplo básico, de cómo se pueden desarrollar aplicaciones MHP con las dos herramientas anteriormente mencionadas.

1. Iniciar Netbeans y crear un nuevo proyecto.

---

<sup>34</sup> Para realizar el ejemplo se utilizó la versión 1.0, disponible en (Microsystems, CDC 1.0)

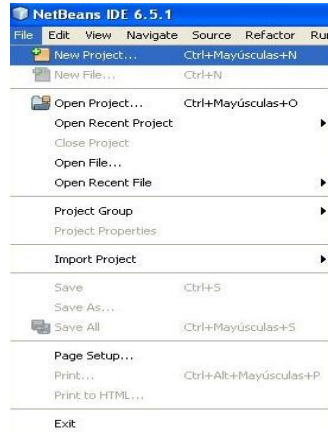


ILUSTRACIÓN 49 CREAR NUEVO PROYECTO

2. Se selecciona la categoría Java ME y el tipo de proyecto CDC.

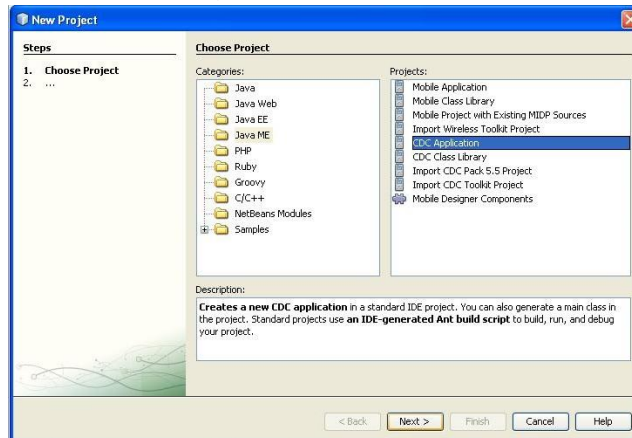


ILUSTRACIÓN 50 SELECCION DE CATEGORÍA Y TIPO DE PROYECTO

3. Se carga la configuración instalada previamente.

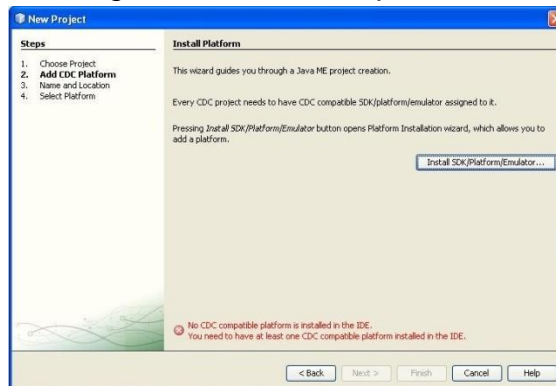


ILUSTRACIÓN 51 CARGA DE CONFIGURACIÓN

#### 4. Seleccionar “Agregar Plataforma”

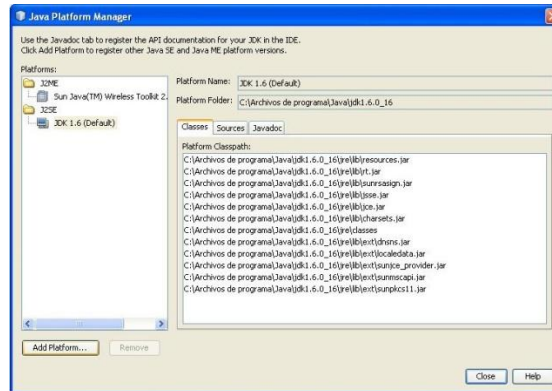


ILUSTRACIÓN 52 AGREGAR PLATAFORMA

#### 5. Escoger el emulador para el tipo de plataforma seleccionada.

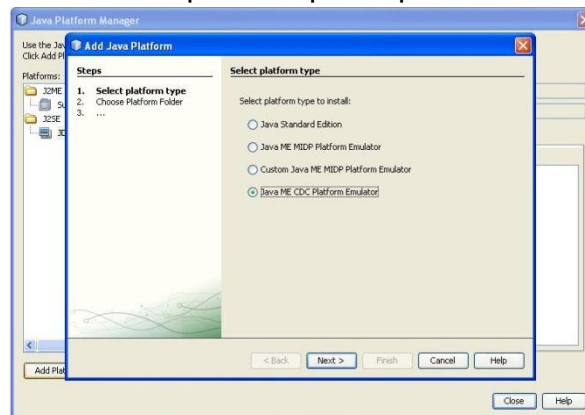


ILUSTRACIÓN 53 EMULADOR DE PLATAFORMA

#### 6. Se selecciona la carpeta de la plataforma.

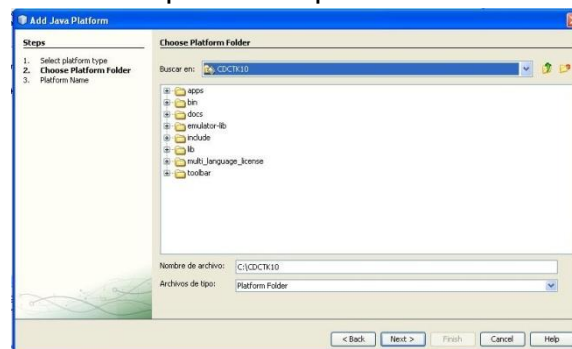


ILUSTRACIÓN 54 SELECCIONAR CARPETA DE LA PLATAFORMA

#### 7. Verificar si la plataforma cargada es la deseada

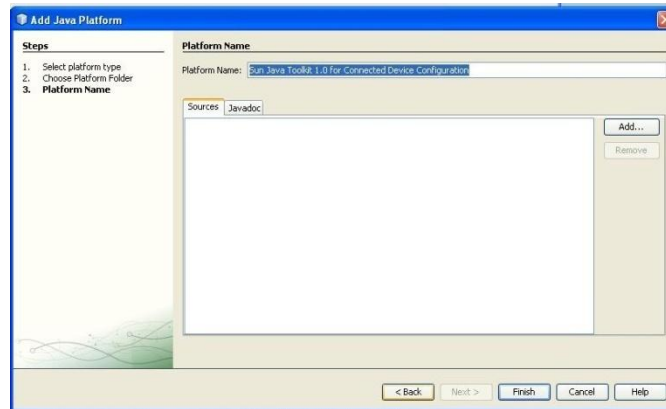


ILUSTRACIÓN 55 VERIFICACIÓN DE LA PLATAFORMA

8. Se observa que la plataforma ha sido cargada y se cierra el asistente de instalación de la plataforma CDC.

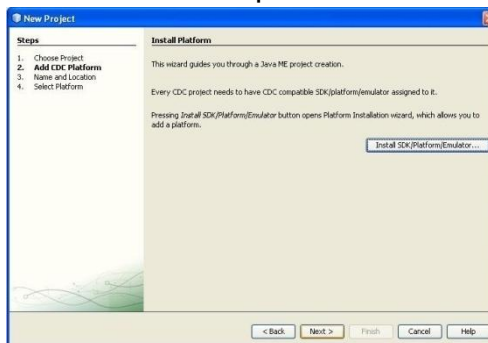


ILUSTRACIÓN 56 FIN DE LA CARGA DE LA PLATAFORMA

9. Se ingresa el nombre del proyecto

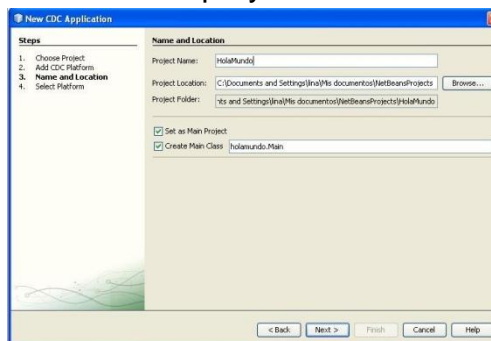


ILUSTRACIÓN 57 INGRESO DEL NOMBRE DEL PROYECTO

10. Finalizar

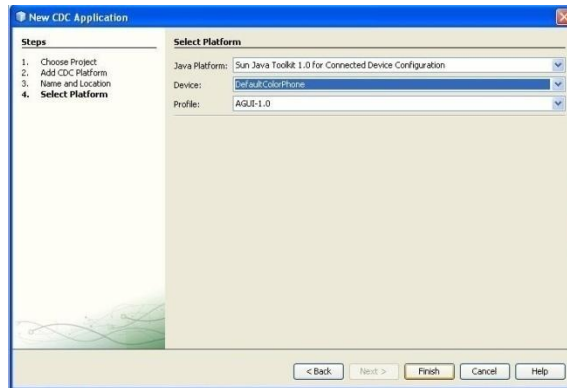


ILUSTRACIÓN 58 FINALIZAR CREACIÓN DEL PROYECTO

11. Ahora, se van a agregar los paquetes que contienen todas las librerías de XletView y JavaTv.



ILUSTRACIÓN 59 AGREGAR PAQUETES DE XLETVIEW Y JAVATV

12. Se selecciona la carpeta donde se descargaron los paquetes y se agregan al proyecto.



ILUSTRACIÓN 60 SELECCION DE CARPETA DE LOS PAQUETES DE XLETVIEW Y JAVATV

13. Crear el Xlet donde se escribirá el código de ejemplo.

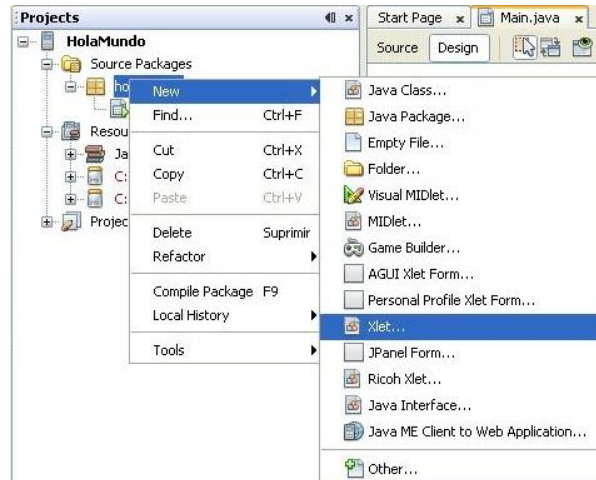


ILUSTRACIÓN 61 INSERTAR XLET

14. Ingresar el nombre del Xlet y los demás datos necesarios.  
Finalizar

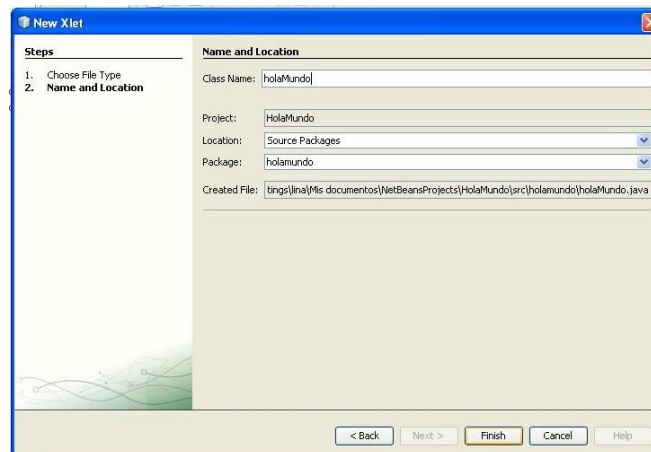


ILUSTRACIÓN 62 NOMBRE DEL XLET

15. Se pueden ver los métodos del ciclo de vida del Xlet que se mostró en la Ilustración 7.

```

package holamundo;
import javax.tv.xlet.*;
/**
 * @author lina
 */
public class holaMundo implements javax.tv.xlet.Xlet {
    /**...*/
    public holaMundo() {...}

    /**...*/
    public void initXlet(javax.tv.xlet.XletContext context)
        throws javax.tv.xlet.XletStateChangeException {
        // TODO implement initialization
    }

    /**...*/
    public void startXlet() throws javax.tv.xlet.XletStateChangeException {...}

    /**...*/
    public void pauseXlet() {...}

    /**...*/
    public void destroyXlet(boolean unconditional)
        throws javax.tv.xlet.XletStateChangeException {...}
}

```

ILUSTRACIÓN 63 MÉTODOS DEL CICLO DE VIDA DEL XLET

16. Digitar el siguiente código:

TABLA 11 CÓDIGO HOLA MUNDO EN NETBEANS

```

import javax.tv.xlet.XletContext;
import javax.tv.xlet.XletStateChangeException;
import org.havi.ui.*;
public class holaMundo implements javax.tv.xlet.Xlet{
    private boolean inicializado; //Se utiliza para indicar que no se ha
    ingresado al método Start
    private XletContext contexto;//Variable que permite guardar el
    contexto
    private HScene escena;//Variable que referencia la escena
    public holaMundo() {
    }
    public void destroyXlet(boolean unconditional) throws
    XletStateChangeException {
        if(escena != null){ // Se destruye la escena en caso de estar
        creada
            escena.setVisible(false);//Se deja de mostrar la escena
            escena.dispose();//Liberacion de recursos
            escena = null;//Con esto, se evita que el Xlet ingrese a otro
            metodo destroy
        }
    }
    public void initXlet(XletContext ctx) throws

```

```

XletStateChangeException {
    inicializado = true;//El Xlet ha sido inicializado
    contexto = ctx;//Se almacena el contexto
}

public void pauseXlet() {
}
//El método startXlet puede ser invocado varias veces
public void startXlet() throws XletStateChangeException {
    if(inicializado == false){
        return;// el Xlet no se ha inicializado, debe invocar initXlet
    }
    inicializado = false;//Si ya se inicio, se pone inicializado en falso
    //para que pueda volver a entrar a initXlet cuando se reanude el
Xlet
    //Aquí iría el código de la aplicación

    //*****
    //1.- Conseguir la referencia a la capa gráfica(a través de la
pantalla)
    HScreen pantalla = HScreen.getDefaultHScreen();//referencia a
la pantalla por defecto
    HGraphicsDevice          capaGrafica          =
pantalla.getDefaultHGraphicsDevice();
    //ref a la capa grafica a traves de la pantalla
    //*****
    //2.- Conseguir un HScene a pantalla completa, tenemos definida
la varble escena global
    HSceneFactory factoria = HSceneFactory.getInstance();
    escena= factoria.getFullScreenScene(capaGrafica);
    //Se pasa a la escena la capaGrafica
    escena.setBackgroundMode(HScene.BACKGROUND_FILL);
    //*****
    //3.- Se añaden los elementos q forman la interfaz de usuario
HStaticText texto = new HStaticText("HOLA MUNDO! \nEjemplo
con XletView y Netbeans",
    50, 100, 600, 100,
        new java.awt.Font("Tiresias", java.awt.Font.BOLD,
28), java.awt.Color.WHITE,
        java.awt.Color.BLUE, new
HDefaultTextLayoutManager());//El layoutManager es un gestor

    //Se agrega el componente al contenedor
    escena.add(texto);

    //4.- Se muestra la interfaz de usuario
    escena.show();
}
}

```

17. Se compila el Xlet y en la carpeta donde se descargó el paquete del XletView, se descomprime y se ejecuta el archivo .jar el cual muestra la interfaz del emulador.

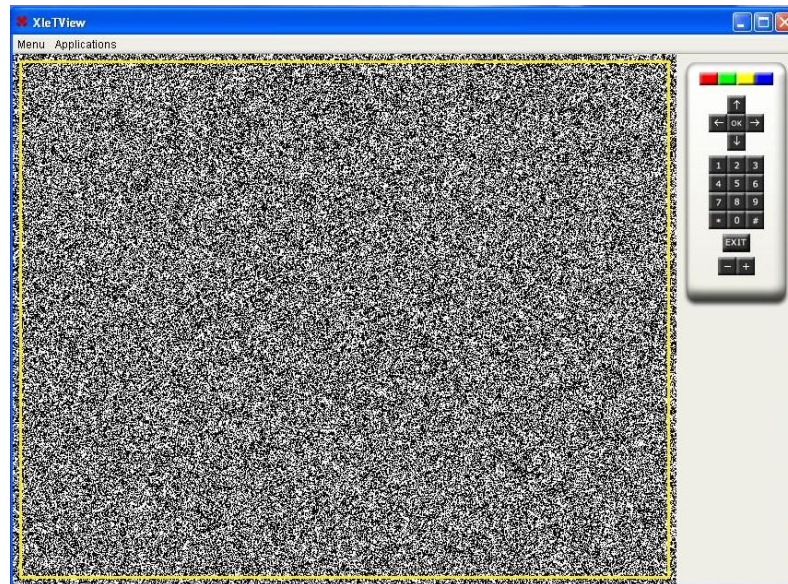


ILUSTRACIÓN 64 INTERFAZ DE XLETVIEW

18. En el XletView, dirigirse a Applications-Manage Applications

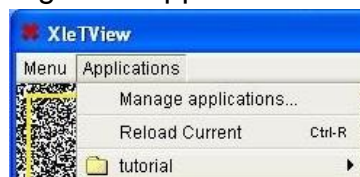
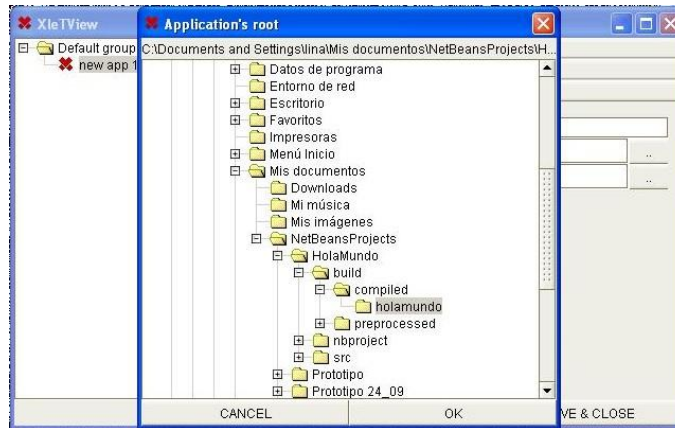


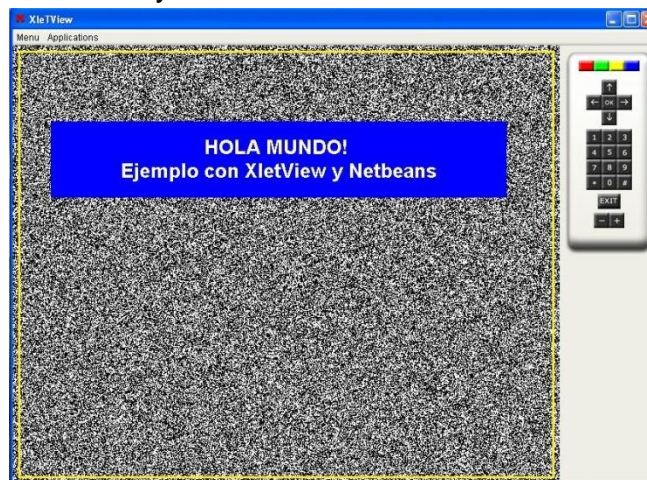
ILUSTRACIÓN 65 ADMINISTRADOR DE APLICACIONES DEL XLETVIEW

19. Aquí se selecciona la carpeta donde está el Xlet; Ingresar el nombre de la aplicación, la ruta donde está el archivo que se compiló en el numeral 17 y seleccionar el archivo compilado.



**ILUSTRACIÓN 66 CARPETA DEL XLET**

20. Finalmente, ir a Applications, escoger el nombre de la aplicación ingresada en 19 y observar el Xlet emulado.



**ILUSTRACIÓN 67 XLET EJECUTÁNDOSE EN EL EMULADOR**

## 5.4. DESARROLLO DE APLICACIONES MHP CON IDESIGNER

### 5.4.1. EJEMPLO “HOLA MUNDO” CON IDESIGNER

1. Iniciar el iDesinger.

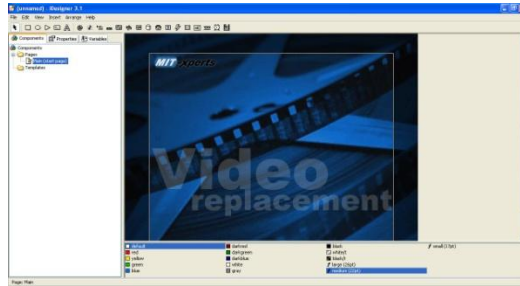


ILUSTRACIÓN 68 ÁREA DE TRABAJO DE IDESIGNER

2. Seleccionar la herramienta de Texto.



ILUSTRACIÓN 69 HERRAMIENTA TEXT

3. Ingresar el Texto que desea mostrar. Click en OK

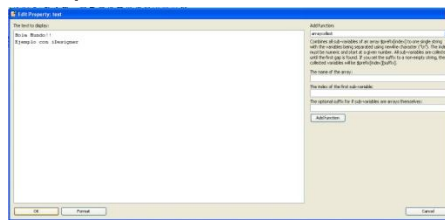


ILUSTRACIÓN 70 EDITOR DE TEXTO

4. El texto se selecciona y se le ajustan las propiedades.



ILUSTRACIÓN 71 VISTA DEL TEXTO

5. Seleccionar el color de fondo para el texto. Esto se hace seleccionando la pestaña Properties en la parte superior derecha

de la pantalla - sección Colors - background, y escoger el color de preferencia<sup>35</sup>.

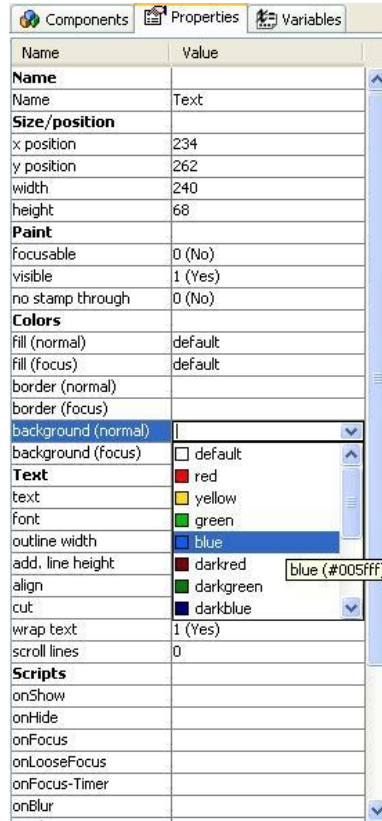


ILUSTRACIÓN 72 PANEL DE PROPIEDADES

6. Un botón puede ser creado mediante círculos, rectángulos, triángulos, etc. En este caso ir al menú principal, escoger la herramienta rectángulo



ILUSTRACIÓN 73 HERRAMIENTA RECTÁNGULO

7. Dibujar un rectángulo en el área de trabajo

<sup>35</sup> Los colores que se muestran son los predeterminados para las aplicaciones MHP. También existe la opción de crear colores personalizados.

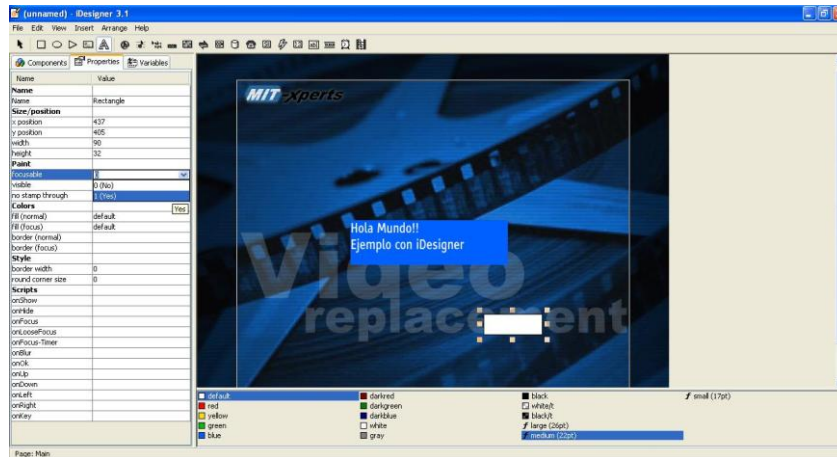


ILUSTRACIÓN 74 RECTÁNGULO COMO BOTÓN

8. Para ponerle texto al botón, se utiliza la herramienta Texto, se selecciona el área donde va a ser ingresado el texto y en el panel Properties ir al campo *text* en la sección *Text*. Aquí se ingresa el texto que tendrá el botón. El tamaño, color y fuente pueden ser cambiados utilizando el menú localizado en la parte inferior del área de trabajo.

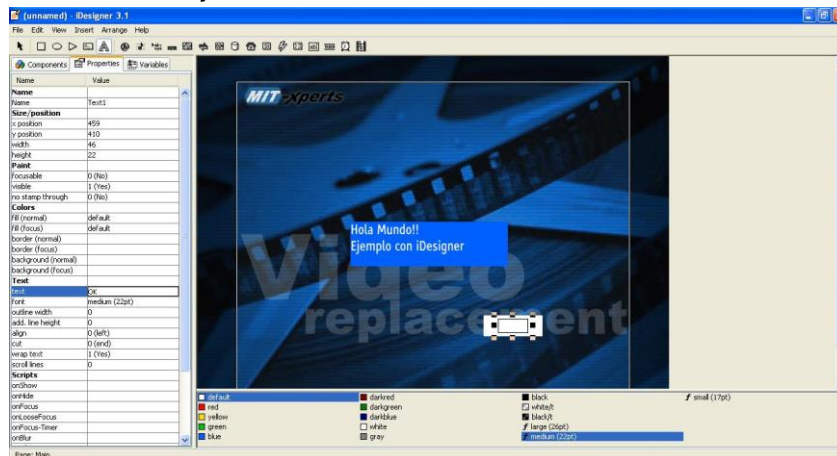


ILUSTRACIÓN 75 TEXTO EN UN BOTÓN

9. Seleccionar el botón junto con su texto, click derecho, Group, para agrupar los dos elementos.

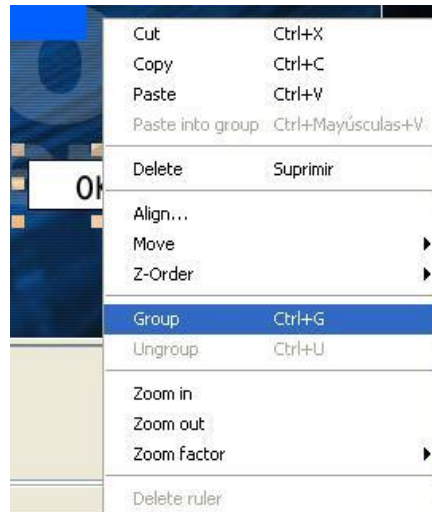


ILUSTRACIÓN 76 AGRUPACIÓN DE ELEMENTOS

10. Para saber cuándo un botón está seleccionado existe la propiedad *focusable* la cual se debe poner en 1 (verdadero), al igual que la propiedad *visible*. Otra característica que deben tener los botones seleccionables es el cambio de color cuando el foco está en ellos. Para ponerles color, ir a la propiedad *fill(focus)* y escoger el color, en este caso es verde.

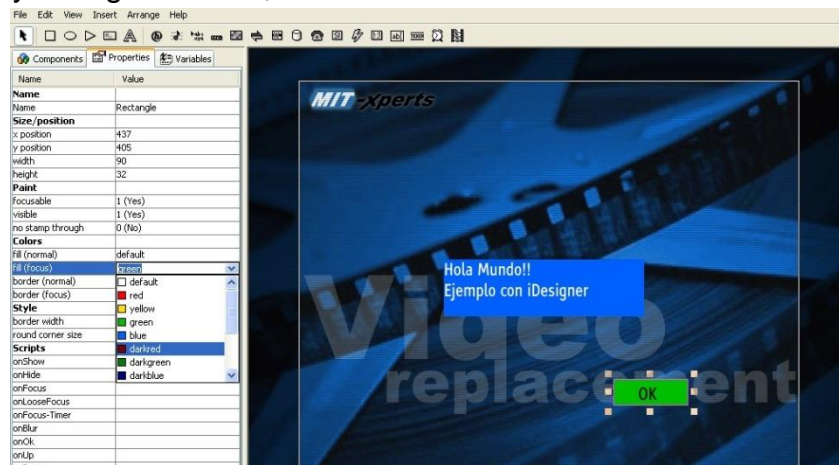


ILUSTRACIÓN 77 FOCO EN UN BOTÓN

11. Los botones tienen *scripts* relacionados con ellos como se muestran a continuación. La acción de estos Scripts dependen de la funcionalidad del botón en la aplicación.

Scripts	
onShow	
onHide	
onFocus	
onLooseFocus	
onFocus-Timer	
onBlur	
onOk	
onUp	
onDown	
onLeft	
onRight	
onKey	

ILUSTRACIÓN 78 SCRIPTS DE UN BOTÓN

12. Por último, se ejecuta el emulador que viene con iDesigner. Ir a *File – Play (test application)*

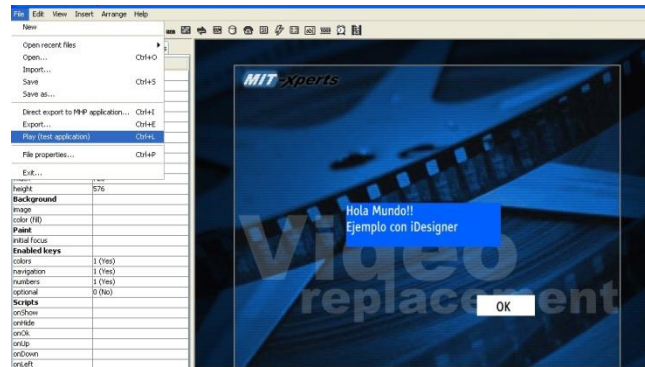


ILUSTRACIÓN 79 EJECUCIÓN DEL EMULADOR

13. Aquí se ve el ejemplo corriendo sobre el emulador.

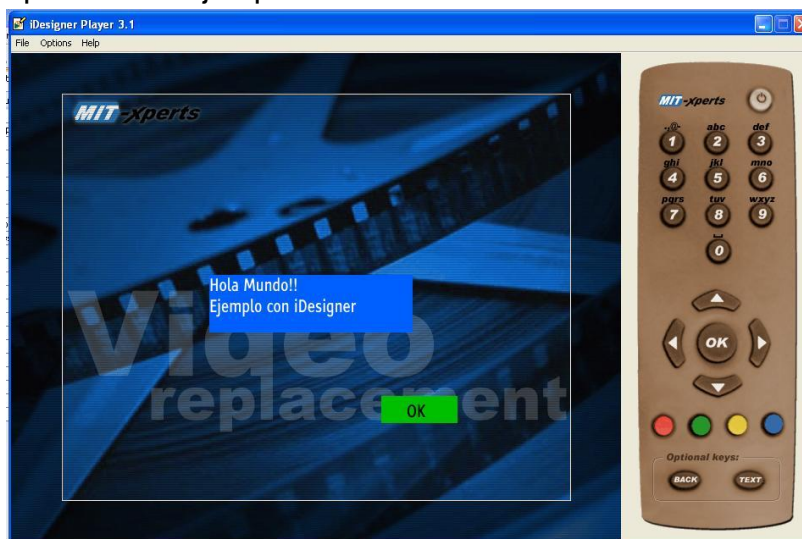


ILUSTRACIÓN 80 EJEMPLO EN EL EMULADOR

## 5.5. PROTOTIPO DE APLICACIÓN MHP CON IDESIGNER

Esta sección está enfocada a dar a conocer el resultado del desarrollo del prototipo de aplicación MHP llamado “Pastillero Interactivo”, utilizando iDesigner.

El prototipo se ejecuta en el emulador proporcionado por este IDE, denominado “iDesigner Player”.

La aplicación puede ser habilitada o deshabilitada, pulsando el botón OK del control cuando el foco esté en la imagen que se encuentra en la parte inferior izquierda de la pantalla del televisor<sup>36</sup>.



ILUSTRACIÓN 81 PANTALLA PRINCIPAL APLICACION MHP

Al dar OK sobre ésta imagen, se muestra el menú principal de navegación.

<sup>36</sup> Cabe aclarar que la imagen de fondo simula el programa de televisión que se esté viendo en ese momento.



ILUSTRACIÓN 82 MENÚ PRINCIPAL DE LA APLICACIÓN MHP

Para ingresar a los menús de Medicamento, Cita y Examen es necesario que el usuario ingrese su número de cédula.



ILUSTRACIÓN 83 INGRESO DE USUARIOS

Al situar el foco en “Medicamento” y dar OK, se muestra el menú de Medicamento, donde se pueden iniciar los medicamentos que el doctor previamente le ha formulado a ese usuario o paciente.



ILUSTRACIÓN 84 MENÚ MEDICAMENTO

Al pulsar el botón Verde, puede acceder a la interfaz de iniciar medicamento, donde muestra los medicamentos que están pendientes y le da la opción de empezar los recordatorios correspondientes.



ILUSTRACIÓN 85 INICIAR MEDICAMENTO

En el menú principal, al situar el foco en “Cita” y dar OK, se muestra el menú de Cita, donde se pueden ingresar citas, modificarlas y cancelarlas. La opción se escoge presionando el botón con el color correspondiente en cada caso.



ILUSTRACIÓN 86 MENÚ DE CITAS

Al presionar el botón Verde, se ejecuta la interfaz de Ingreso de Citas.



ILUSTRACIÓN 87 INGRESO DE CITAS PASO 1



ILUSTRACIÓN 88 INGRESO DE CITAS PASO 2



ILUSTRACIÓN 89 INGRESO DE CITAS PASO 3



ILUSTRACIÓN 90 INGRESO DE CITAS PASO 4



ILUSTRACIÓN 91 INGRESO DE CITAS PASO 5

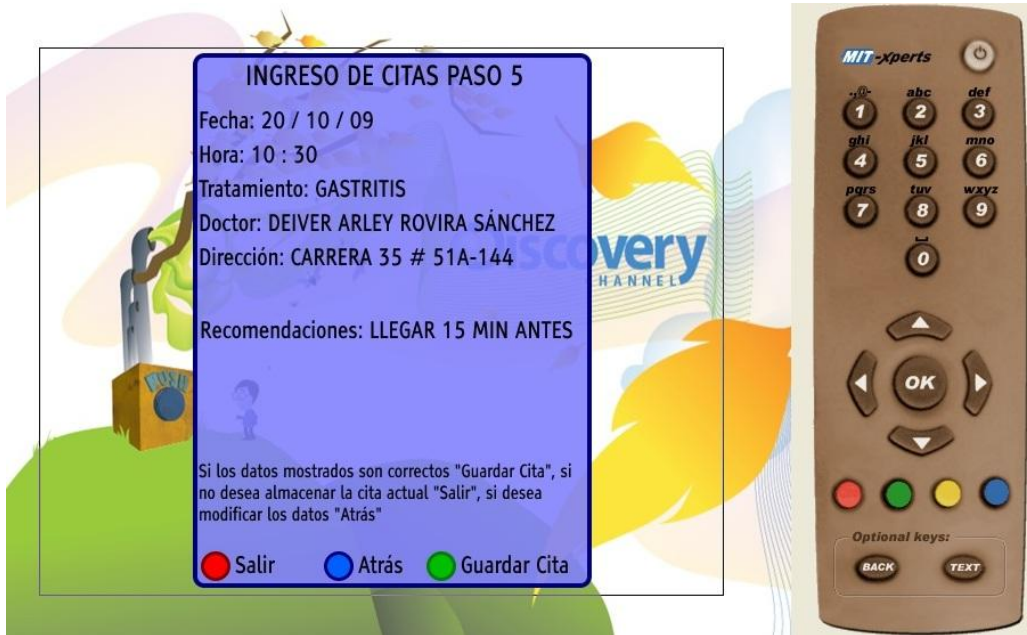


ILUSTRACIÓN 92 INGRESO DE CITAS PASO 6

La opción “Modificar Cita” (botón azul) del menú Cita, ejecuta las siguientes interfaces a la hora de modificar una cita ya existente.



ILUSTRACIÓN 93 MODIFICACIÓN DE CITAS PASO 1

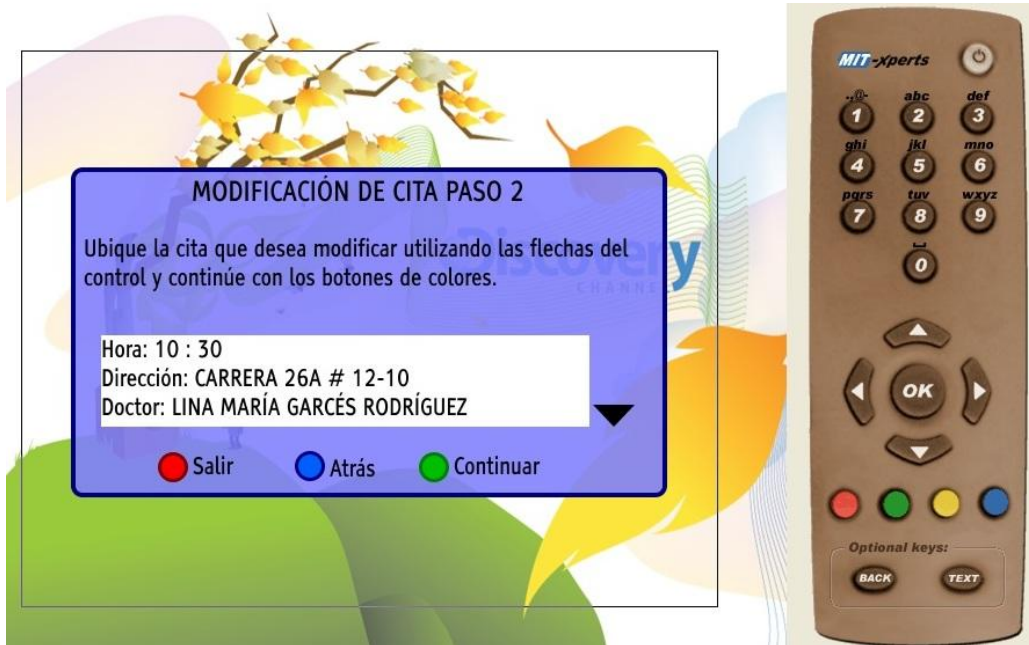


ILUSTRACIÓN 94 MODIFICACIÓN DE CITAS PASO 2



ILUSTRACIÓN 95 MODIFICACIÓN DE CITAS PASO 3



ILUSTRACIÓN 96 MODIFICACIÓN DE CITAS PASO 4



ILUSTRACIÓN 97 MODIFICACION DE CITAS PASO 5

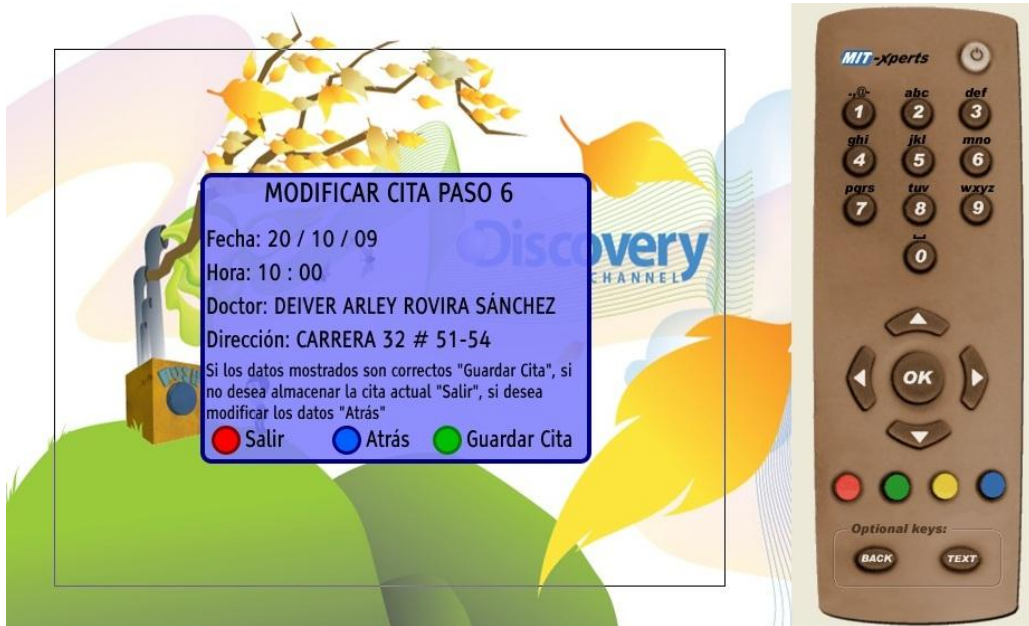


ILUSTRACIÓN 98 MODIFICACION DE CITAS PASO 6

La opción "Cancelar" (botón amarillo) del menú Cita, ejecuta las siguientes interfaces a la hora de cancelar una cita ya existente.



ILUSTRACIÓN 99 CANCELACIÓN DE CITA PASO 1

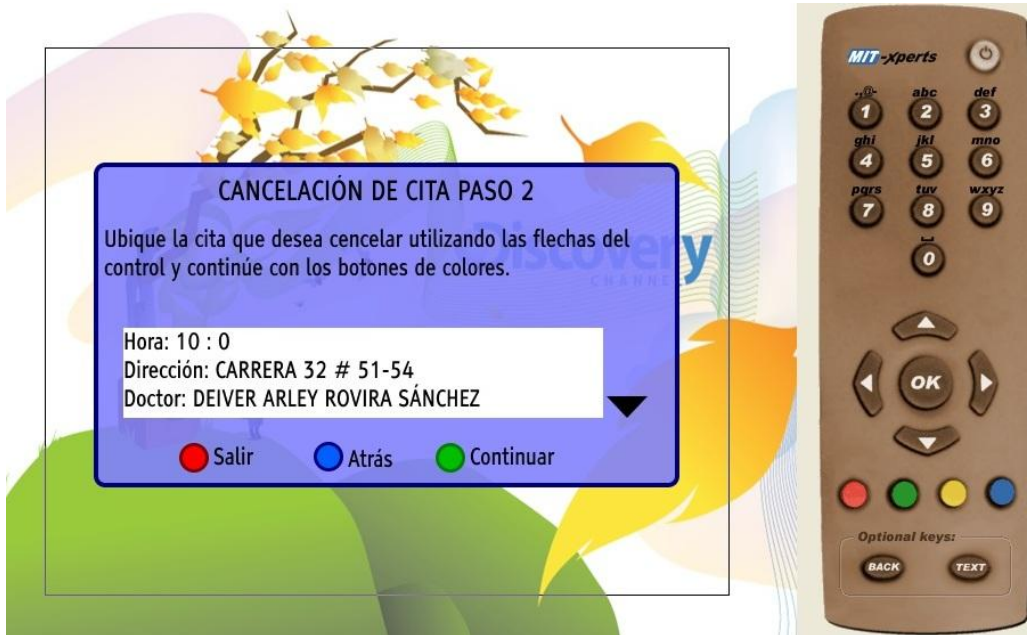


ILUSTRACIÓN 100 CANCELACIÓN DE CITA PASO 2

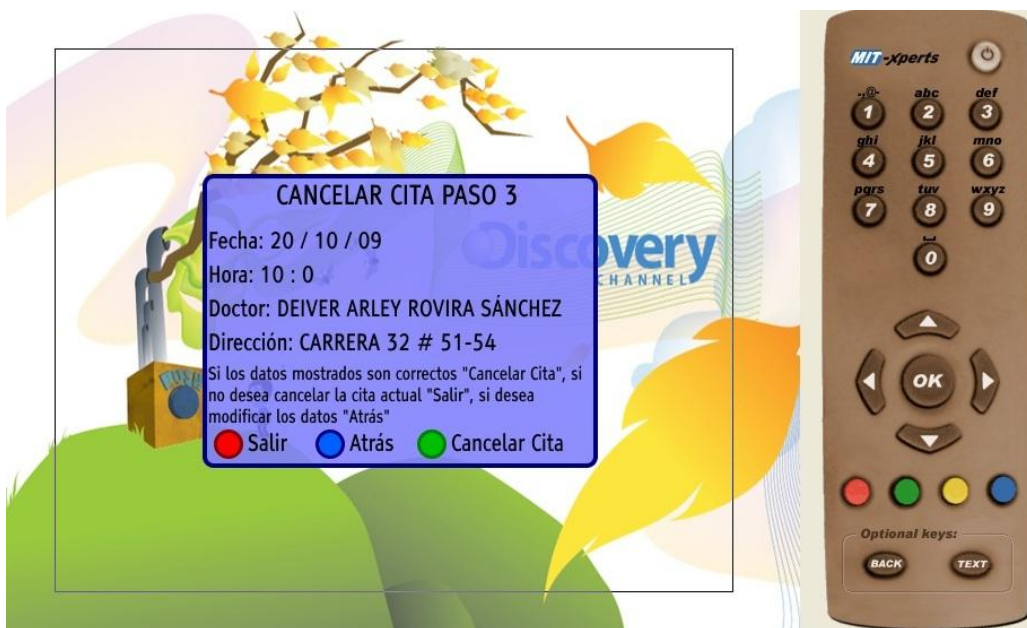


ILUSTRACIÓN 101 CANCELACIÓN DE CITA PASO 3

En el menú principal, al situar el foco en "Examen" y dar OK, se muestra el menú de Exámenes, donde se pueden ingresar exámenes,

modificarlos y cancelarlos. La opción se escoge presionando el botón con el color correspondiente en cada caso.



ILUSTRACIÓN 102 MENÚ DE EXÁMENES

Al presionar el botón Verde, se ejecuta la interfaz de Ingreso de Exámenes.

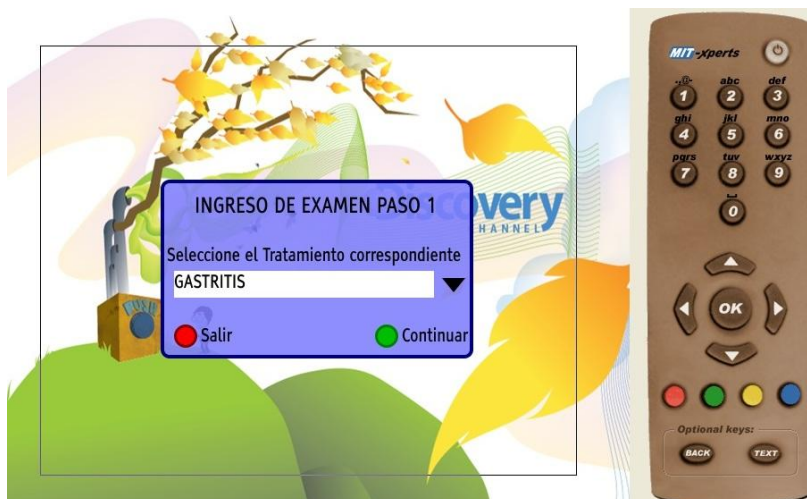


ILUSTRACIÓN 103 INGRESO DE EXAMEN PASO 1

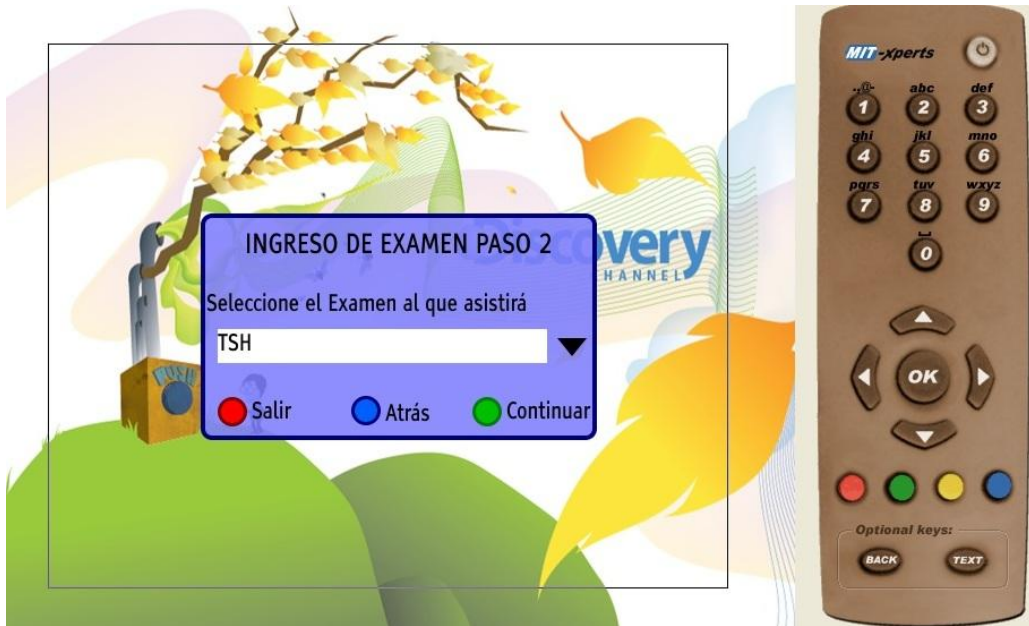


ILUSTRACIÓN 104 INGRESO DE EXAMEN PASO 2



ILUSTRACIÓN 105 INGRESO DE EXAMEN PASO 3



ILUSTRACIÓN 106 INGRESO DE EXAMEN PASO 4



ILUSTRACIÓN 107 INGRESO DE EXAMEN PASO 5

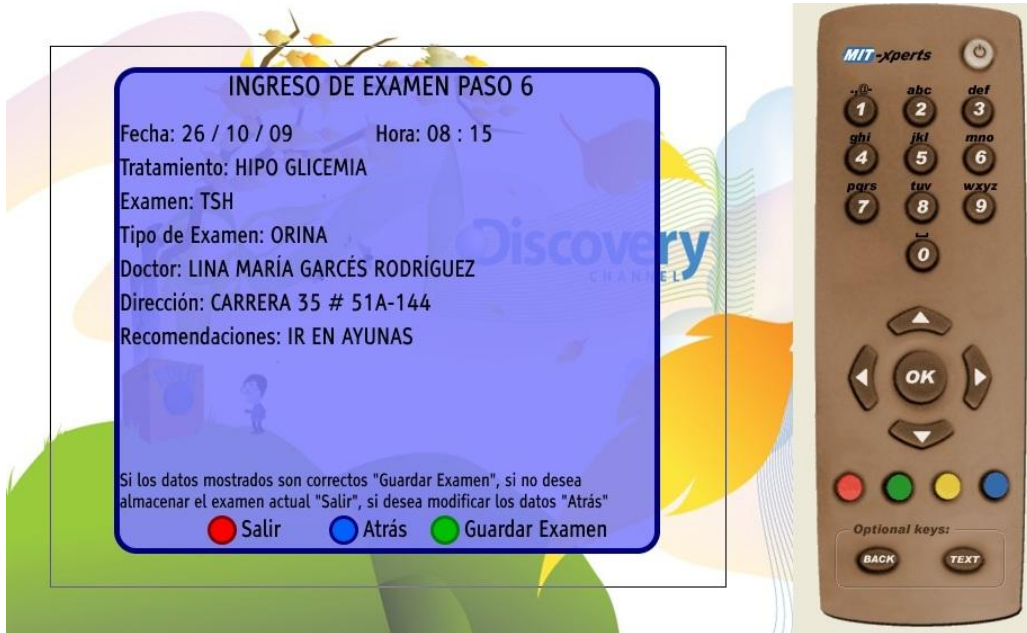


ILUSTRACIÓN 108 INGRESO DE EXAMEN PASO 6

La opción “Modificar Examen” (botón azul) del menú Exámenes, ejecuta las siguientes interfaces a la hora de modificar un examen ya existente.



ILUSTRACIÓN 109 MODIFICACIÓN DE EXAMEN PASO 1

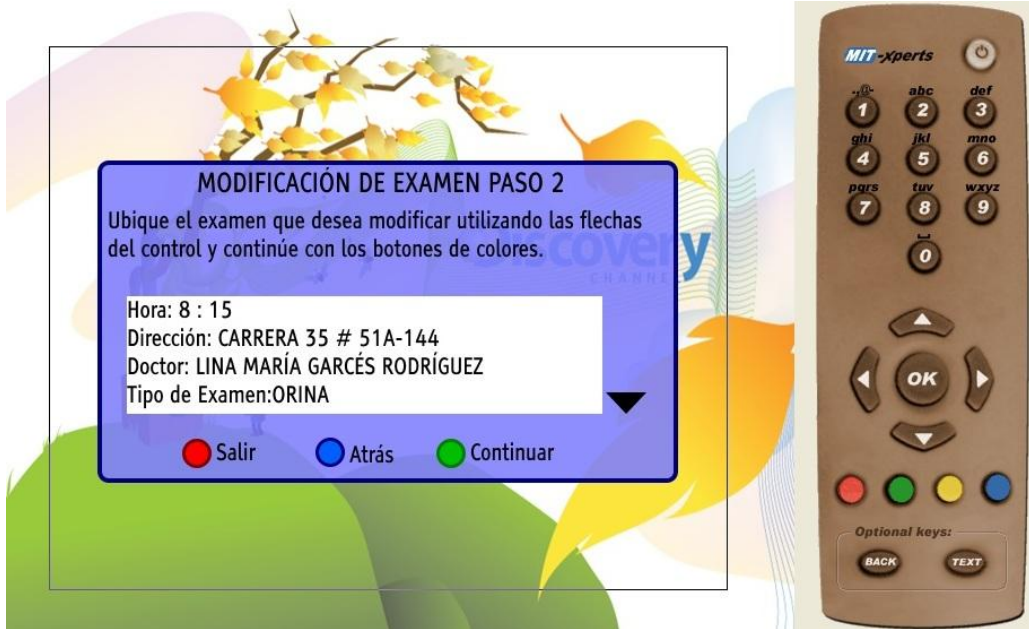


ILUSTRACIÓN 110 MODIFICACIÓN DE EXAMEN PASO 2



ILUSTRACIÓN 111 MODIFICACIÓN DE EXAMEN PASO 3

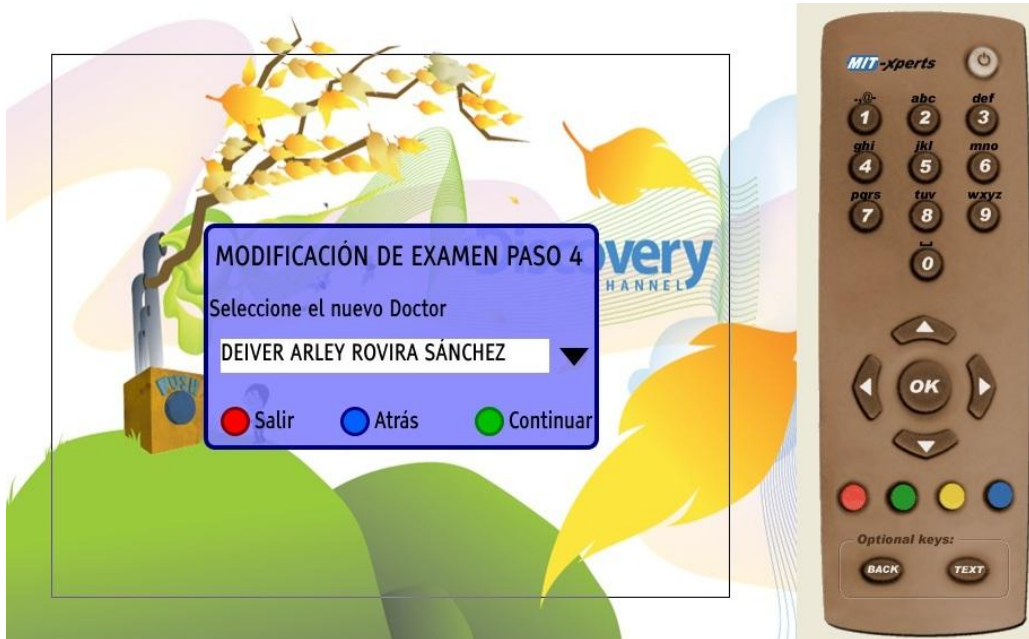


ILUSTRACIÓN 112 MODIFICACIÓN DE EXAMEN PASO 4



ILUSTRACIÓN 113 MODIFICACION DE EXAMEN PASO 5

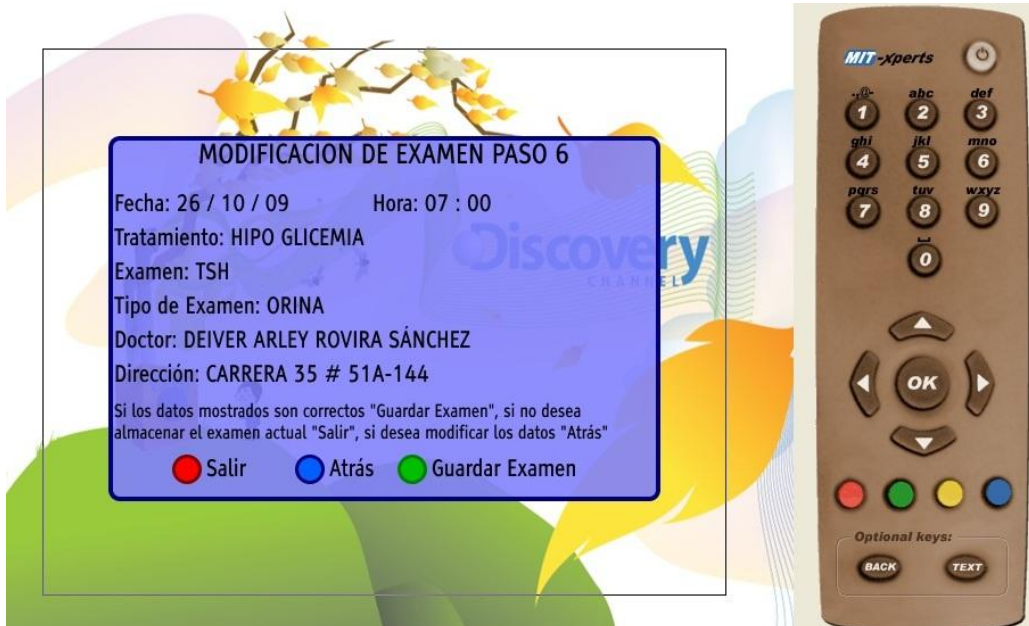


ILUSTRACIÓN 114 MODIFICACIÓN DE EXAMEN PASO 6

La opción “Cancelar Examen” (botón amarillo) del menú Exámenes, ejecuta las siguientes interfaces a la hora de cancelar un examen ya existente.



ILUSTRACIÓN 115 CANCELACIÓN DE EXAMEN PASO 1

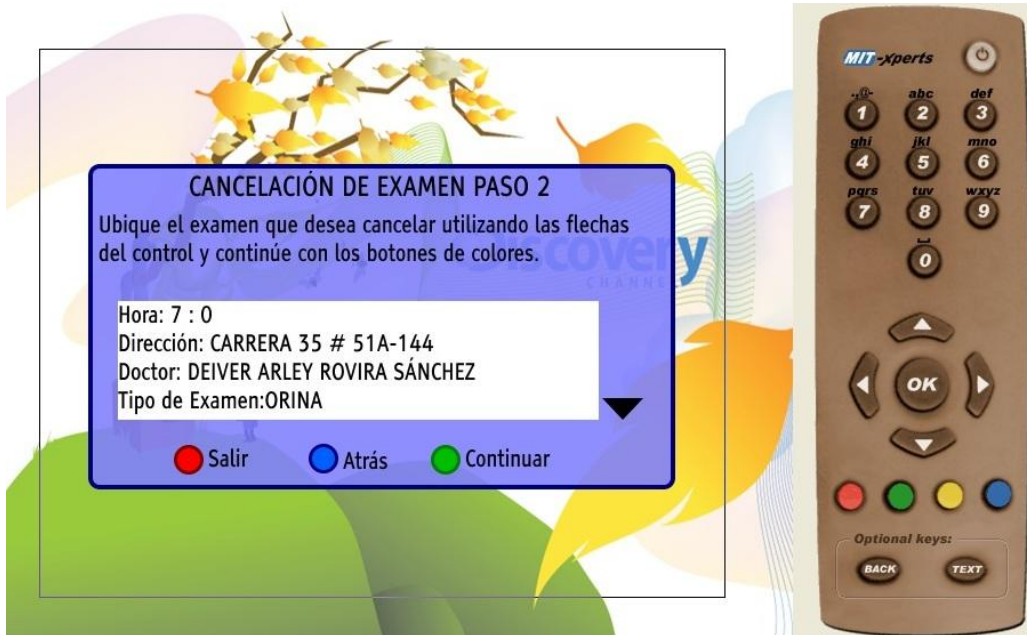


ILUSTRACIÓN 116 CANCELACIÓN DE EXAMEN PASO 2

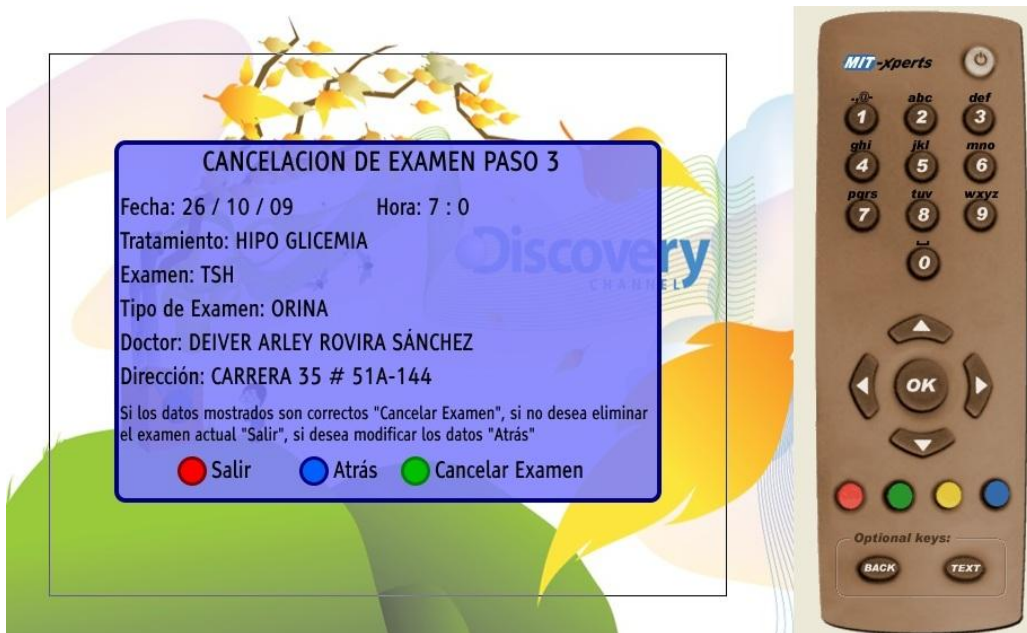


ILUSTRACIÓN 117 CANCELACIÓN DE EXAMEN PASO 3

Los recordatorios son ejecutados dependiendo de la hora. Hay tres tipos de recordatorios, los cuales son mostrados a continuación.



ILUSTRACIÓN 118 RECORDATORIO DE MEDICAMENTO



ILUSTRACIÓN 119 RECORDATORIO DE CITAS



ILUSTRACIÓN 120 RECORDATORIO DE EXAMEN

## 6. PRUEBAS

Las pruebas son el proceso de encontrar diferencias entre el comportamiento esperado (requerido), especificados por los modelos del sistema, y el comportamiento observado (esperado) por el sistema. (Bruegge & H. Dutoit, 2002)

Existen diferentes tipos de pruebas, los cuales se mencionan a continuación.

- Pruebas Unitarias: Encuentran diferencias entre el modelo del diseño de objetos y sus componentes correspondientes
- Pruebas Estructurales: Encuentran diferencias entre el modelo de diseño del sistema y un subconjunto de subsistemas integrados.
- Pruebas de Desempeño: Encuentran diferencias entre los requerimientos no funcionales y el desempeño del sistema.
- Pruebas Funcionales: Encuentran diferencias entre el modelo de caso de uso y el sistema.
- Pruebas de Interfaz de usuario: Verifica la interacción de los usuarios con el software. Se tiene que asegurar que la interfaz suministre un apropiado acceso y navegación a través de las funciones.
- Pruebas de seguridad: Permite restringir la entrada a algunas funciones específicas del sistema a los diferentes usuarios adscritos al mismo.

Las pruebas que se realizaron se basaron en las denominadas Pruebas Funcionales, y el propósito para cada prueba fue que se cumpliera con los casos de uso mencionados en la sección 4.3 y en el Anexo 10.3.

TABLA 12 PRUEBA CASO DE USO REANUDAR TRATAMIENTO

Caso de uso	Propósito de la prueba	Pre-requisitos	Conclusiones
<b>Reanudar Tratamiento</b>	Reanudar los tratamientos suspendidos	Deben haber tratamientos suspendidos	El tratamiento se reanudó y el doctor ya puede asignarle Medicamentos y Exámenes al paciente
	Guardar el tratamiento reanudado		El paciente puede ingresar las citas que correspondan a ese tratamiento

TABLA 13 PRUEBA CASO DE USO FORMULAR MEDICAMENTO

Caso de uso	Propósito de la prueba	Pre-requisitos	Conclusiones
<b>Formular Medicamento</b>	Agregar un medicamento a un tratamiento	El paciente debe tener tratamientos activos	El medicamento se agregó al tratamiento y el paciente desde su televisor debe programar la toma de sus medicamentos

TABLA 14 PRUEBA CASO DE USO FORMULAR EXÁMENES

Caso de uso	Propósito de la prueba	Pre-requisitos	Conclusiones
<b>Formular exámenes</b>	Ingresar una orden de examen a un paciente	El paciente debe tener tratamientos activos	El doctor ingresó la orden del Examen y el paciente debe ingresar la dirección, hora y fecha del examen

**TABLA 15 PRUEBA CASO DE USO INGRESAR DATOS PACIENTE**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Ingresar Datos Paciente</b>	Validar el ingreso de los datos del paciente.		El paciente se insertó correctamente en la Base de Datos y está habilitado para que el doctor le asigne un tratamiento
	El paciente es registrado correctamente		

**TABLA 16 PRUEBA CASO DE USO INGRESAR TRATAMIENTO**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Ingresar Tratamiento</b>	Validar el ingreso de los datos del tratamiento	El paciente debe estar registrado	El tratamiento se le agrega al paciente y el doctor ya puede Formular medicamentos y exámenes para el tratamiento
	El tratamiento es registrado correctamente		

**TABLA 17 PRUEBA CASO DE USO RECORDAR MEDICAMENTO**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Recordar Medicamento</b>	La aplicación MHP ejecuta el recordatorio a la hora exacta	el paciente debe tener asociado un medicamento	Se le recuerda tomar el medicamento al paciente y este

		El paciente debe haber iniciado el medicamento para que el recordatorio se ejecute	puede posponerlo por 15 minutos o aceptarlo
--	--	--	---

**TABLA 18 PRUEBA CASO DE USO RECORDAR CITA**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Recordar Cita</b>	La aplicación MHP ejecuta el recordatorio 12 horas antes de la cita	el paciente debe agregar los datos de la cita para que se ejecute el recordatorio	Se le recuerda la cita al paciente 12 horas antes y este puede posponerla por 15 minutos hasta aceptarla ó posponerla inmediatamente

**TABLA 19 PRUEBA CASO DE USO RECORDAR EXAMEN**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Recordar Examen</b>	La aplicación MHP ejecuta el recordatorio 12 horas antes de la cita	El doctor debe ordenar el examen el paciente debe agregar los datos del examen para que se ejecute el recordatorio	Se le recuerda el examen al paciente 12 horas antes y este puede posponerla por 15 minutos hasta aceptarla ó posponerla inmediatamente

**TABLA 20 PRUEBA CASO DE USO CONTROLAR RECORDATORIO**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
--------------------	-------------------------------	-----------------------	---------------------

<b>Controlar Recordatorio</b>	la aplicación MHP modifica la hora del recordatorio si se pospone	Deben existir recordatorios	El recordatorio se ejecuta 15 minutos después si se pospone
	la aplicación MHP modifica el campo Aceptado si el recordatorio es aceptado		El recordatorio no se vuelve a mostrar si este es aceptado

**TABLA 21 PRUEBA CASO DE USO INGRESAR CITAS**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Ingresar Citas</b>	Validar el ingreso de los datos de la cita	El paciente debe tener la fecha, hora, nombre del doctor y ubicación de la cita	La cita se ingresó correctamente
	Ingresar la cita en la tabla Citas y Recordatorio		La cita será recordada 12 horas antes de la hora real

**TABLA 22 PRUEBA CASO DE USO MODIFICAR CITAS**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Modificar Citas</b>	Validar la hora y fecha de la cita	El paciente debe tener citas pendientes por recordar	La cita se modificó correctamente en la base de datos
	Modificar la hora, ubicación, doctor y fecha de la cita		

**TABLA 23 PRUEBA CASO DE USO CANCELAR CITAS**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Cancelar Citas</b>	eliminar la cita de los recordatorios	El paciente debe tener citas	La cita se canceló correctamente

	cancelar la cita para el paciente	pendientes por recordar	La cita se eliminó de los recordatorios
--	-----------------------------------	-------------------------	---

**TABLA 24 PRUEBA CASO DE USO INGRESAR EXAMEN**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Ingresar Examen</b>	Validar el ingreso de los datos del examen	El paciente debe tener la orden de examen dada por el doctor	El examen se ingresó correctamente
	Ingresar el examen en la tabla Examen y Recordatorio		El examen será recordado 12 horas antes de la hora real
	Ingresar hora, ubicación, fecha del examen		

**TABLA 25 PRUEBA CASO DE USO MODIFICAR EXAMEN**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Modificar Examen</b>	Validar el ingreso de los datos del examen	El paciente debe tener exámenes pendientes por recordar	El examen se modificó correctamente en la base de datos
	Modificar la hora, ubicación, doctor y fecha de la cita		El examen se mostrará 12 horas antes de la hora real

**TABLA 26 PRUEBA CASO DE USO CANCELAR EXAMEN**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Cancelar Examen</b>	El recordatorio del examen será cancelado	El paciente debe tener citas pendientes por recordar	El examen se canceló correctamente
	El examen será		El examen se

	eliminado de la base de datos		eliminó de los recordatorios
--	-------------------------------	--	------------------------------

**TABLA 27 PRUEBA CASO DE USO ACEPTAR RECORDATORIOS**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Aceptar Recordatorios</b>	Modificar el valor del campo Aceptado	La aplicación MHP debe ejecutar un recordatorio, sea de Medicamentos, Citas o Exámenes	Se modifica el valor del campo Aceptado y el recordatorio no se volverá a mostrar
	Dejar de mostrar los recordatorios		

**TABLA 28 PRUEBA CASO DE USO POSPONER RECORDATORIOS**

<b>Caso de uso</b>	<b>Propósito de la prueba</b>	<b>Pre-requisitos</b>	<b>Conclusiones</b>
<b>Posponer Recordatorios</b>	modificar la tabla de recordatorios	La aplicación MHP debe ejecutar un recordatorio, sea de Medicamentos, Citas o Exámenes	El recordatorio será mostrado 15 minutos después

## 7. CONCLUSIONES Y RECOMENDACIONES

El haber escogido el Entorno de Desarrollo Integrado *iDesigner* para desarrollar y probar el prototipo se basó en las siguientes consideraciones:

- Las alternativas libres no cumplen con todas las especificaciones para las aplicaciones interactivas que se definen en el estándar MHP. Netbeans es un IDE poderoso y por medio de las librerías de JavaTV permitía desarrollar el prototipo. El tiempo de desarrollo invertido utilizando esta herramienta habría sobrepasado el estipulado, además de la dificultad para realizar pruebas en el emulador XletView debido a las adaptaciones a las cuales debía someterse para cubrir las necesidades del proyecto, constituyeron las principales razones por las cuales no se optó por esta herramienta
- El uso del método de análisis de características y del modelo ISO 9126 arrojó como resultado que la mejor elección entre los IDEs encontrados era *iDesigner*, debido a un mayor puntaje en las características consideradas importantes.

El utilizar *iDesigner* permitió desarrollar el prototipo de una manera rápida, puesto que contiene herramientas que ayudan en el diseño de la interfaz de usuario, aspecto importante que deben tener las aplicaciones interactivas para televisión digital.

La metodología XP es una buena opción para el desarrollo de aplicaciones interactivas para TVDi, en especial para el prototipo mostrado en este proyecto, por varias razones.

- Está planeada para usarse en proyectos de exploración como lo son los proyectos basados en nuevas tecnologías, en este caso la televisión digital interactiva bajo el estándar de difusión DVB-T del cual no se tenía previo conocimiento.

- Es adecuada para proyectos dónde los resultados finales son vagamente conocidos desde el inicio del proyecto o hay dudas técnicas o de interfaz de usuario al inicio, que se resuelven mientras va avanzando el proyecto.
- Se acomoda perfectamente en equipos de tamaño pequeño.
- Se debe mantener una comunicación continua y directa entre los participantes.
- Posee una característica importante que es la denominada Pair Programming donde dos personas son las encargadas de desarrollar una parte del proyecto, en este caso fue todo el prototipo por su tamaño, y trabajan conjuntamente para hallar errores, fallas y encontrar la mejor manera de realizar las actividades.

Con la finalización de este trabajo se da un paso más en el estudio de la televisión digital interactiva terrestre en el país pues quedan plasmadas aquí las bases necesarias para comprender como funciona esta tecnología, el alcance y el tipo de aplicaciones que puede brindar a toda la sociedad y lo que se espera de esta en el futuro. Este proyecto forma parte de una línea de investigación nueva en la escuela de Ingeniería de Sistemas e informática de la Universidad siendo el segundo proyecto de investigación en el área y pretende incentivar a sus estudiantes a continuar el estudio tanto de la TDT como de otra tecnología que está llegando al país, IPTV.

Para desarrollos futuros se recomienda adquirir una licencia estudiantil ya sea de Osmosys SDK o de iDesigner debido a que las herramientas libres no cumplen con todas las especificaciones del estándar MHP. También sería conveniente contar con un decodificador MHP para hacer las pruebas reales de las aplicaciones.

## 8. GLOSARIO

### A

ACAP : Advanced Common Application Platform: Especificación middleware basada en GEM

ADSL : Asymmetric Digital Subscriber Line. Permite transmisiones rápidas de datos sobre líneas telefónicas de cobre.

AIT: Application Information Table. Tabla definida por MHP para proveer información básica sobre las aplicaciones MHP.

API: Application Programming Interface. Clases de Software a través de las cuales las aplicaciones operan sobre funciones específicas contenidas en MHP

ATSC :Advanced Television Systems Committee. Elabora los estándares de televisión en Estados Unidos.

### C

CDC : Connected Device Configuration

CLDC : Connected Limited Device Configuration

### D

DAVIC : Digital Audio Video Council: Asociación a nivel mundial que promueven la interoperabilidad entre equipos audiovisuales de todo tipo concentrándose en las interfaces estandarizadas.

DSM-CC :Digital Storage Media Command and Control: Formato para transmisión de datos y control de información en una sección privada MPEG-2.

DTT : Digital Terrestrial Television

DTV : Digital Television

DVB-J: DVB-J (DVB-Java) aplicaciones que son escritas en Java utilizando el conjunto de APIs de MHP

DVB-S : Digital Video Broadcasting – Satellite

DVB-T : Digital Video Broadcasting – Terrestrial

## **E**

EPG : Electronic Program Guide

ESG : Electronic Service Guide

ETSI : European Telecommunications Standards Institute

## **G**

GEM : Global Executable Multimedia Home Platform.

GPRS : General Packet Radio Service

GSM : Global System for Mobile Communications

GUI : Graphical User Interface

## **H**

HAVi : Home Audio Video Interoperability: permite diferentes tipos de entretenimiento en los hogares y diferentes dispositivos de comunicación para que trabajen juntos

HDTV : High Definition Television

## **I**

IP: Internet Protocol:

IPTV : Internet Protocol Television. Televisión transmitida vía redes IP.

ISO : International Organization for Standardization

ITV : Interactive Television.

## **J**

J2ME : Java 2 Micro Edition

J2SE : Java 2 Platform Standard Edition

JavaTV : It is a Java-based API for iTV applications development.

JMF : Java Media Framework

JVM : Java Virtual Machine

## **M**

MHP : Multimedia Home Platform

MPEG-2: Técnica de compresión de Audio y video

MUX : Multiplexor

## **O**

OC (Object Carousel): Estructura de datos que contiene aplicaciones, archivos multimedia, que son transmitidos en un flujo de transporte. En el context de MHP se utilizan los carruseles de objeto DSM-CC

OCAP :Open Cable Application Platform

OFDM : Orthogonal frequency-division multiplexing

## **P**

PersonalJAE : Personal Java Application Environment

## **S**

SDK: Software Development Kit

STB : Set-top Box. Terminal MHP conectado a un televisor

## **T**

T-Care: Cubre los servicios de comunicación que conectan a los pacientes y al personal de la salud en forma bidireccional.

T-Commerce : Cubre las aplicaciones comerciales y transaccionales que son desarrolladas y se ejecutan en la pantalla del TV.

T-Health: Cubre todos los servicios de información relacionados con la salud y el bienestar y son desarrollados para ejecutarse en un TV.

T-Learning : Cubre las aplicaciones de aprendizaje.

## **X**

Xlet : Interface usada para controlar la ejecución del ciclo de vida de las aplicaciones en MHP

## 9. REFERENCIAS BIBLIOGRÁFICAS

- Alticast. (s.f.). *Alticast*. Obtenido de <http://www.alticast.com/main.html>
- ARIB. (2009). *ARIB Association of Radio Industries and Businesses*. Recuperado el 2009, de <http://www.arib.or.jp/english/>
- Bruegge, B., & H. Dutoit, A. (2002). *ingeniería del Software orientada a objetos*. Pearson.
- Ceballos, F. J. (2006). *Java 2. Curso de programación. Tercera Edición*. México DF.: Alfa Omega.
- DVB. (2009). *DVB*. Recuperado el 2009, de [http://www.dvb.org/technology/fact\\_sheets/DVB-GEM-Fact-Sheet.0409.pdf](http://www.dvb.org/technology/fact_sheets/DVB-GEM-Fact-Sheet.0409.pdf)
- HAVI. (s.f.). *HAVI*. Obtenido de HAVI: <http://www.havi.org/index.asp>
- Icareus. (s.f.). *Icareus*. Obtenido de <http://www.icareus.com/web/guest/technologies>
- Ipsos-Napoleon Franco. (2008). *La Gran Encuesta de la Television en Colombia*. Obtenido de [http://www.cntv.org.co/cntv\\_bop/estudios/ipsos\\_rural.pdf](http://www.cntv.org.co/cntv_bop/estudios/ipsos_rural.pdf)
- ITU. (2009). *International Telecommunication Union*. Recuperado el 2009, de <http://www.itu.int/>
- Jame. (s.f.). *Jame*. Obtenido de <http://www.jame.tv/>
- Lawrence Pfleeger, S. *Ingeniería del Software Teoría y Práctica*. En S. Lawrence Pfleeger, *Ingeniería del Software Teoría y Práctica*. Prentice Hall.
- López Peral, I., & Bueno Vallejo, D. (2009). *IMPLEMENTACIÓN DEL JUEGO 50X15 PARA RECEPTORES DE TELEVISIÓN DIGITAL TERRESTRE (TDT) CON MHP*. Málaga España: Universidad de Málaga.

MARTÍNEZ P, A. (2008). MHP: Una oportunidad para la convergencia. 9.

MHP. (2009). *mhp*. Obtenido de <http://www.mhp.org>

Microsystems, S. (s.f.). *CDC 1.0*. Obtenido de [http://java.sun.com/products/cdctoolkit/download-1\\_0.html](http://java.sun.com/products/cdctoolkit/download-1_0.html)

microsystems, S. (2009). *Java TV API*. Recuperado el 2009, de <http://java.sun.com/javame/technology/javatv/overview.html>

Microsystems, S. (s.f.). *JavaTv*. Obtenido de <http://java.sun.com/javame/technology/javatv/>

Multimedia Home Platforms. (2006). *MHP-Guide*.

Osmosys. (s.f.). *Osmosys Tv*. Obtenido de <http://www.osmosys.tv/>

Pressman, R. S. *Ingeniería del Software un enfoque práctico*. Mc Graw Hill.

Sanz de Acedo, R. (2008). *Centros de Excelencia Software- Gobierno de Navarra*. Recuperado el 2009, de <http://www.cesnavarra.net/cesdigital/Lists/Noticias%20CESDigital/DispFormCES.aspx?List=5ec0dfc7-7911-470b-8b6b-71ba72783fdd&ID=34>

sourceforge. (s.f.). *XletView*. Obtenido de <http://sourceforge.net/projects/xletview/files/XleTView/xletview-0.3.6/>

Steven Morris, A. S.-C. (2005). *Interactive TV Standards*. Focal Press.

XletView. (s.f.). *XletView*. Obtenido de <http://www.xletview.org/>

Xperts, M. (s.f.). *MIT Xperts*. Obtenido de <http://www.mit-xperts.com/>

## 10. ANEXOS

### 10.1. APLICACIONES INTERACTIVAS

Las Aplicaciones Interactivas se pueden clasificar en diferentes tipos de aplicaciones como las que acompañan a los tradicionales programas de televisión, por ejemplo:

- Aquellas que se utilizan antes de la emisión de un programa, las cuales pueden servir como instrumento para atraer a los televidentes con antelación a la promoción de dicho programa, como la selección de una película.
- Durante una transmisión, estas permiten a los usuarios una participación activa, como concursos o votaciones en tiempo real ó una encuesta en un noticiero, de la misma forma proveen información adicional que no puede ser cubierta de manera completa por el programa de televisión mismo, por ejemplo en emisiones de grandes eventos como votaciones presidenciales, juegos olímpicos o simplemente programas de entretenimiento.
- Después de un programa, servicios adicionales podrían ofrecer más información relacionada con el programa, así como sitios de interés o de interacción como chat con televidentes del mismo programa. Estos servicios pueden ser tratados por los consumidores independientemente de la franja horaria del programa original.

El otro gran grupo de aplicaciones tienen que ver con las aplicaciones independientes del programa, estas ofrecen servicios de información general, comunicación, entretenimiento, vídeo por demanda (VoD), o servicios T-commerce y, por último, la televisión basada en T-Government, T-Learning ó T-Health.

Es posible crear nuevas aplicaciones combinando algunos tipos de estas, por ejemplo, un digitext, el cual podría ser un servicio de noticias o T-learning combinado con T-Chat, etc. Por lo tanto, las explicaciones

y las clasificaciones sólo sirven para describir los tipos de aplicaciones y sus conceptos *generales*.

#### *Servicios de Información.*

Entre este grupo se destaca la guía electrónica de programas ( **Electronic Program Guide** ). Es una aplicación común que debería estar en todos los países y en todas las cajas decodificadoras, la EPG lista los canales y los programas disponibles en estos canales, brindándole al usuario un cronograma completo de los programas que se van a emitir durante toda una semana.

Los datos de la programación son usualmente obtenidos de sistemas de información (SI) desde las entidades radiodifusoras, por lo tanto le permiten al usuario conocer cuál programa está al aire y cuál será el próximo en emitirse, etc.

#### *Servicios de Noticias/ Servicios de Eventos*

Hay varios tipos de Servicios de Noticias, la mayoría de estos son portales de anuncios o listados de las noticias actuales, algunos con sofisticada categorización y otros con estructuras muy simples, el rango de las aplicaciones en las noticias va desde la superposición de una fina banda (generalmente en la parte inferior de nuestro televisor) anunciando las noticias o mensajes de los televidentes hasta portales de información en grandes eventos como en campeonatos mundiales o en los juegos olímpicos.

#### *Pronóstico del Tiempo*

Los servicios orientados hacia el pronóstico del tiempo son usualmente de tipo “solo de difusión”. La interactividad reside principalmente en la posibilidad de que los televidentes pueden escoger la vista detallada de una determinada región del país para un cierto día.

#### *Servicios de Tráfico*

Similarmente, los servicios de tráfico pueden ofrecer a los usuarios la posibilidad de conocer la información detallada de una región

determinada en cierto momento, esto con el fin de conocer las vías que están muy congestionadas o que simplemente se encuentran fuera de servicio debido a un derrumbe por catástrofes naturales, situación muy común en nuestro país. Otro tipo de información pertinente al tráfico podría incluir también los horarios de funcionamiento para el sistema de transporte público.

### *Texto Digital/Teletexto*

La televisión interactiva ofrece la oportunidad de entregar todo tipo de información adicional relacionada con el programa de televisión de la misma forma como se hace en el internet.

Actualmente las paginas basadas en texto, los prestadores de servicios los organismos de radiodifusión, tienen la posibilidad de mezclar el texto con imágenes, audio y video en portales interactivos, que enriquecen la experiencia del televidente, brindando por ejemplo la posibilidad al usuario de escoger el idioma en el cual desea ver los subtítulos de su programa favorito.

### *Servicios de Comunicación*

Diversas compañías en el mercado ofrecen el servicio de correo electrónico basado en MHP a sus clientes. El servicio de correo electrónico y el chat son claramente aplicaciones de tipo bi-direccional ya que abarca la comunicación entre los usuarios finales, esto aclarando que sería necesario el uso de un canal de retorno para conectar el servidor de correos a internet.

### *Servicios de Entretenimiento*

Los juegos interactivos para la televisión abarcan los tipos de aplicaciones desde “solo de difusión” hasta lo que es “bi-direccional”, ya que comprenden gran cantidad de estos.

Entre los que se conocen como de “solo difusión”, se pueden clasificar aquellos juegos básicos como el tetris o el Black Jack, actualmente existe una versión del tetris para MHP llamado *Quadra*, el cual solo necesita pocos botones en el control de mando para controlarlo y no es necesario una respuesta por parte del proveedor del servicio; debido a

que estos juegos son independientes de los programas emitidos en televisión, generalmente son programados por gente ajena a estas empresas, por lo tanto deben ser descargados de algún portal web de televisión digital donde se desarrollen aplicaciones soportadas por la iTV.

Otro tipo de aplicaciones que los organismos de radiodifusión suelen desarrollar son aquellas en las que se necesita de la participación del televidente en la votación para un concurso determinado, donde claramente se ve la interacción bi-direccional con el usuario, ya que se tiene en cuenta la decisión del televidente.

#### *Servicio de Video por Demanda*

iTV también puede funcionar como una interface para servicios de Video por Demanda (VOD), el cual a través de un portal el usuario puede escoger entre un abanico amplio de películas o programas aquel que más le interese, actualmente este tipo de servicios es utilizado en algunos hoteles europeos, donde el usuario demanda o solicita la transmisión de una película en su habitación y posteriormente el cargo es agregado a su cuenta de hotel.

Es por esto que se dice que la película está siempre disponible para la hora en la que el usuario la desee ver, de esta forma se mejora el servicio de PPV (Pague Por Ver) actual.

#### *T-Commerce*

Análoga al término E-commerce, T-commerce cubre todo aquello relacionado con aplicaciones comerciales y transacciones que son terminadas y realizadas en la pantalla del televisor.

La televisión interactiva ofrece un gran número de posibilidades para el T-commerce, desde la publicidad interactiva hasta la realización real de compras y transacciones valiéndose de un televisor.

Un gran número de aplicaciones para T-Commerce han sido desarrolladas, la iTV abre un mundo de opciones muy grande para el desarrollo de este tipo de aplicaciones, entre las que más se destacan están las tradicionales compras por internet (eBay), donde el usuario

tiene la posibilidad de realizar un seguimiento a las subastas de eBay, al igual de tener la posibilidad de ser informado en el momento de que el estado de una subasta cambie.

#### *Publicidad Interactiva*

También hay gran cantidad de aplicaciones para publicidad interactiva, donde más allá de suministrar solo la información de los productos, se crea una serie de menús interactivos donde el usuario podrá navegar y conocer un poco más de dichos productos.

#### *T-Government*

Esta se refiere principalmente a todos los tipos de servicios de información e idealmente a la comunicación y acciones necesarias con la autoridad pertinente en un área específica, la cual es desarrollada e implementada para la televisión.

Como un instrumento interactivo para una sociedad democrática, la iTV permite al televidente formar parte en la decisión a tomar en una encuesta por parte de un programa específico, ya sea democrático, educativo o de entretenimiento.

#### *T-Learning*

Para este tipo de aplicaciones la iTV brinda grandes ventajas, ya que hace posible la interacción del televidente con el programa que está viendo, brindándole la posibilidad de escoger lo que ve y lo que escucha, por ejemplo, sería posible para el televidente observar un programa en español y que la interacción con este sea en un idioma distinto, ésta sería una técnica muy interesante para la sociedad estudiantil interesada en dominar un segundo idioma.

Existen diferentes tipos de aplicaciones en este campo, no solo para el estudio de nuevos idiomas sino para el aprendizaje en diferentes sectores como negocios e incluso temas científicos, donde el usuario podrá interactuar con el programa de televisión de forma directa.

#### *T-Health / T-Care*

El término T-Health comprende lo relacionado con información sobre salud, temas médicos y bienestar, mientras que T-Care se extiende más hacia lo que es la conexión de los pacientes con instituciones de salud. Generalmente este tipo de aplicaciones se basa en T-Mail, y T-Chat ya que se necesita una conexión bi-direccional para la comunicación entre paciente-especialista.

Una de las principales aplicaciones en este tema tiene que ver con los pacientes con enfermedades crónicas o en edades avanzadas donde el médico les brindará una serie de contenidos educativos, así como también oportunos recordatorios para la ingesta de sus medicamentos -tema clave del proyecto- con el fin de que aumente su bienestar.

### *Business TV*

Más que un tipo de aplicación, éste se podría tratar como un escenario o una combinación de los anteriores tipos de aplicaciones. Sin embargo, la idea de combinar los diferentes tipos de aplicaciones tales como el T-Commerce, T-Communication, T-Government, depende directamente de la creatividad de los desarrolladores para crear aplicaciones que favorezcan el desarrollo de un negocio.

Las organizaciones pueden utilizar este tipo de aplicaciones interactivas para proporcionar todo tipo de servicios de información y comunicación a sus empleados (al estilo de una intranet) y, en menor medida, a sus socios comerciales y clientes.

## 10.2. NIVELES DE INTERACTIVIDAD

Normalmente, la interactividad se da gracias a listas de menús y selección de vínculos (Ilustración 121. EPG o en español guía electrónica de programas. Ejemplo de aplicación interactiva). Para los anuncios, la interactividad puede significar la búsqueda de productos o servicios y la posibilidad de adquirirlos en línea, como por ejemplo la realización de transacciones financieras. Para las Guías Electrónicas de Programas (EPG) la interactividad significa buscar en la guía. Para Video sobre Demanda (VOD) la interactividad puede ser la selección de video y el control de su reproducción. En los juegos la interactividad significa jugarlo y posiblemente simular la competencia entre

televidentes agregando competidores. En la programación de noticias, la interactividad significa la búsqueda de noticias en un sitio similar como se hace en internet. Para los programas de deportes, la interactividad es la habilidad de obtener estadísticas, y la reproducción de partes del partido desde varios ángulos.

8.05pm Mon 17 digital

guide **TV GUIDE LISTINGS**

	Today	8.00 pm	8.30 pm	9.00 pm
101 BBC ONE	Eastenders	Lakesiders	News	
102 BBC TWO	Wildlife ...	Travel Show	Have I Got ...	
106 Sky One	Star Trek Voyager		Sliders	▶
109 UK Gold	Comedy Alternative		Casualty	▶
112 Living	◀ Murder Call		A Woman ...	▶
115 Disney	◀ Danny	The Wonder Years		▶
118 Granada Plus	Return of Sherlock Holmes		Classic Co ...	▶
121 Challenge TV	The Crystal Maze		Prize Time	
124 Bravo	The A-Team		Real Stories	
127 Paramount	Grace ...	Caroline ...	Spin City	

■ Page Up   
 ■ Page Down   
 ■ +24 Hours   
 ■ -24 Hours

Choose title and press **SELECT**

ILUSTRACIÓN 121. EPG O EN ESPAÑOL GUIA ELECTRÓNICA DE PROGRAMAS. EJEMPLO DE APLICACIÓN INTERACTIVA

En principio, los receptores de TVD e iTV son similares a los receptores análogos tradicionales, puesto que ellos reciben la señal de video y los televidentes seleccionan a través de un Control Remoto, y obtienen el video en una pantalla. Los receptores de TVD difieren de los tradicionales en que el video y el audio son entregados a través del transporte MPEG. Los receptores de televisión interactiva mejoran aún más a los receptores de TVD con la capacidad de procesar datos entregados simultáneamente con video y audio, los cuales contienen aplicaciones. Estos datos pueden ser entregados por medio de transporte MPEG o IP. Sin embargo, las principales técnicas de partida para los receptores de iTV más allá de los receptores digitales y análogos no interactivos es la necesidad de un componente software middleware como también un poderoso sistema operativo. La

complejidad derivada de estas nuevas capas de software se cuenta abrumadora.

En general, la interactividad en ITV puede ser categorizada en tres niveles básicos:

- **Solamente Emisión:** Las aplicaciones de emisión, como un teletexto digital y una grabación de un video personal (PVR), obtienen los datos requeridos de un flujo de emisión. Ellos soportan solamente interactividad local donde el consumidor puede interactuar localmente con los datos almacenados en el dispositivo final
- **Interactividad unidireccional:** Las aplicaciones interactivas unidireccionales, como las votaciones en línea y respuestas de anuncios, permiten a los usuarios proporcionar datos en una sola dirección, sin respuesta directa de un servidor de retorno.
- **Interactividad Bidireccional:** Las aplicaciones interactivas bidireccionales, como el email, buscadores web, y juegos online, permiten a los usuarios adquirir datos de fuentes externas al flujo de emisión, como un servidor de retorno.

La siguiente tabla muestra cómo estos niveles técnicos de interactividad se relacionan con ciertos tipos de aplicaciones:

**TABLA 29. NIVELES TÉCNICOS DE INTERACTIVIDAD RELACIONADOS CON APLICACIONES. TOMADA DE MHP-GUIDE**

Tipo	Aplicación	Niveles de interactividad		
		Solo emisión	Interactividad unidireccional	Interactividad Bidireccional
<b>Información</b>	EPG	<input checked="" type="checkbox"/>		
	Noticias	<input checked="" type="checkbox"/>		
	Teletexto/Digitexto	<input checked="" type="checkbox"/>		
<b>Comunicación</b>	T-Mail			<input checked="" type="checkbox"/>
	T-Chat			<input checked="" type="checkbox"/>
<b>Entretenimiento</b>	T-Games	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Video por demanda (VoD)		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Apuestas			<input checked="" type="checkbox"/>
<b>T-Commerce</b>	T-Banking			<input checked="" type="checkbox"/>
	Aplicaciones promocionales		<input checked="" type="checkbox"/>	
	Anuncios			<input checked="" type="checkbox"/>

	interactivos			
<b>T-Goverment</b>	Portales de información	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>T-Learning</b>	Juegos educativos	<input checked="" type="checkbox"/>		
	Profesionales en la enseñanza	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>T-Health</b>	Portales informativos	<input checked="" type="checkbox"/>		
	T-Care			<input checked="" type="checkbox"/>

### 10.3. CUADROS DE CASOS DE USO

TABLA 30 CUADROS DE CASOS DE USO

Caso de Uso	Actor que inicia	Condiciones previas	Pasos en el escenario	Condiciones posteriores	Actor que se beneficia
Reanudar Tratamiento	Doctor	El paciente debe tener tratamientos en estado suspendido	Ingresar cedula paciente	El doctor puede asociarle medicamentos y exámenes al paciente.	Paciente
			Menu principal - Reanudar Tratamiento		
			Seleccione el tratamiento		
			Reanudar	El paciente puede ingresar las citas que correspondan a ese tratamiento	
			Verificar		
Formular medicamento	Doctor	El paciente debe tener tratamientos activos	Ingresar cedula paciente	El paciente desde su televisor debe programar la toma de sus medicamentos	Paciente
			Menu principal-Ver tratamientos		
			Seleccionar Tratamiento		
			Dosis del Tratamiento		
			Formular medicamentos		
			Seleccionar el Tipo de Medicamento		
			Seleccionar el Nombre del Medicamento		
			Ingresar Cantidad, Unidad, Frecuencia, Duracion y Recomendacion		
Click Formular medicamento					
Formular exámenes	Doctor	El paciente debe tener tratamientos activos	Ingresar cedula paciente	El paciente desde su televisor debe ingresar la fecha, hora, lugar y doctor del examen	Paciente
			Menu principal- Ingresar Orden de Examen		
			Ingresar Nombre del tratamiento		

			Ingresar el Tipo de examen		
			Ingresar el nombre del examen		
			Ingresar las recomendaciones		
			Aceptar		

Ingresar datos paciente	Doctor		Ingresar el número de cédula	El doctor puede comenzar a tratar al paciente	Aplicación web
			Ingresar el nombre		
			Ingresar los apellidos		
			Ingresar el género		
			Ingresar la fecha de nacimiento		
			Aceptar		
Ingresar Tratamiento	Doctor	El paciente debe estar registrado	Ingresar cedula paciente	El doctor puede asociarle medicamentos y exámenes al paciente relacionados con el tratamiento creado	Aplicación Web
			Menu principal - Crear Tratamiento		
			Ingresar descripción/nombre del tratamiento		
			Guardar		
Recordar medicamento	Aplicación MHP	El paciente debe tener asociado un medicamento para poder ser recordado	La aplicación MHP ejecuta el recordatorio del medicamento si la hora es la indicada.	El paciente puede posponer el recordatorio para que sea recordado en 15 minutos, o puede aceptar el recordatorio	Paciente
		El paciente debe haber iniciado el medicamento para que los recordatorios se puedan ejecutar			
Recordar cita	Aplicación	El paciente debe	La aplicación MHP ejecuta el recordatorio del	El paciente puede	Paciente

	MHP	ingresar los datos de la cita para que le sea recordada	medicamento 12 horas antes de la cita.	posponer el recordatorio para que sea recordado en 15 minutos, o puede aceptar el recordatorio	
Recordar exámen	Aplicación MHP	El paciente debe haber iniciado los datos del exámen que previamente hubiera sido asignado por el médico	La aplicación MHP ejecuta el recordatorio del medicamento 12 horas antes del exámen	El paciente puede posponer el recordatorio para que sea recordado en 15 minutos, o puede aceptar el recordatorio	Paciente
Controlar Recordatorio	Aplicación MHP	Deben existir recordatorios	La aplicación MHP modifica el campo Aceptado si el recordatorio es aceptado, o cambia la hora del recordatorio si el recordatorio se pospuso.	El recordatorio se ejecutará 15 minutos después si se pospone o no se vuelve a ejecutar más si se acepta	Paciente
Ingresar citas	Paciente	El paciente debe tener la fecha, hora, doctor y dirección de la cita para poderla ingresar	OK (inicio) Ingresar cedula Menú Citas Ingresar Citas Ingresar Fecha y Hora de la cita Seleccionar el tratamiento Seleccionar el doctor Seleccionar la dirección Ingresar las recomendaciones >Ingresar Cita	La cita le será recordada con 12 horas de anticipación, puede ser modificada y cancelada	Aplicación MHP
Modificar citas	Paciente	El paciente debe tener una cita pendiente por recordar	OK (inicio) Ingresar cedula Menú Citas	La cita le será recordada 12 horas antes de la fecha y hora modificadas.	Aplicación MHP

			Modificar Cita		
			Ingresar la fecha de la cita		
			>Continuar		
			Seleccionar la Cita		
			>Continuar		
			Ingresar las modificaciones		
			>Modificar Cita		

Cancelar cita	Paciente	El paciente debe tener una cita pendiente por recordar	OK (inicio)	El recordatorio será cancelado y la Cita eliminada de la base de datos	Aplicación MHP
			Ingresar cedula		
			Menú Citas		
			Cancelar Cita		
			Ingresar la fecha de la cita		
			>Continuar		
			>Cancelar cita		
Ingresar examen	Paciente	El paciente debe tener una orden de exámen ingresada por el médico	OK (inicio)	El exámen será recordado 12 horas antes y puede ser modificado y cancelado	Aplicación MHP
			Ingresar cedula		
			Menú Examen		
			Ingresar Examen		
			Seleccionar el tratamiento		
			>Continuar		
			>Continuar		

			Ingresar la fecha y hora del examen		
			>Continuar		
			Seleccione el Doctor		
			>Continuar		
			Seleccione la Dirección		
			>Ingresar Examen		
Modificar Examen	Paciente	El paciente debe tener un examen pendiente para recordar	OK (inicio)	El examen será recordado 12 horas antes de la fecha y hora modificadas.	Aplicación MHP
			Ingresar cedula		
			Menú Examen		
			Modificar Examen		
			Ingresar la fecha del examen		
			>Continuar		
			Seleccionar el examen		
			>Continuar		
Ingresar las modificaciones	El recordatorio será cancelado y el examen eliminado de la base de datos	Aplicación MHP			
>Modificar Cita					
OK (inicio)					
Ingresar cedula					
Menú Examen					
Cancelar Examen					
Ingresar la fecha del examen					
>Continuar					
Seleccionar el examen					
>Cancelar examen					

Aceptar recordatorios	Paciente	La aplicación MHP debe ejecutar un recordatorio ya sea de Medicamento, cita o examen	>Aceptar Recordatorio	La aplicación modifica el valor del campo Aceptado, para que no se siga ejecutando este recordatorio	Aplicación MHP
Posponer recordatorios	Paciente	La aplicación MHP debe ejecutar un recordatorio ya sea de Medicamento, cita o examen	>Posponer Recordatorio	La aplicación modifica la hora del recordatorio, sumándole 15 minutos para volverlo a ejecutar	Aplicación MHP

