

UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

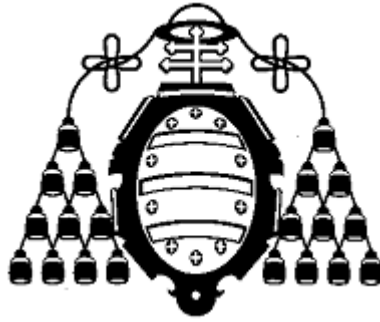
MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

**FLUMUSIC: EMISIÓN DE VIDEOS MUSICALES BAJO DEMANDA A
TRAVÉS DE SMART TV**



**RUBÉN MENES HEVIA
JUNIO 2013**



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

**FLUMUSIC: EMISIÓN DE VIDEOS MUSICALES BAJO DEMANDA A
TRAVÉS DE SMART TV**



**RUBÉN MENES HEVIA
JUNIO 2013**

**IZERTIS
TUTOR: JOSÉ IGNACIO PINTADO CIMADEVILLA
COTUTOR: FRANCISCO JOSÉ SUÁREZ ALONSO**

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN	14
2	ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS).....	15
2.1	Establecimiento del alcance del sistema.....	15
2.2	Estudio de la situación actual.....	15
2.3	Definición de los requisitos del sistema.....	17
2.3.1	Requisitos funcionales.....	17
2.3.2	Requisitos No Funcionales	19
2.4	Estudio de las alternativas de la solución.....	20
2.4.2	Valoración de las alternativas	25
2.5	Solución	31
2.6	Planificación y presupuesto.....	31
2.6.1	Planificación.....	31
2.6.2	Identificación de las tareas a realizar.....	32
2.6.3	Estimación temporal de las tareas a realizar	33
2.6.4	Presupuestos parciales	36
2.6.5	Presupuesto total.....	37
3	ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI)	38
3.1	Especificaciones del sistema.....	38
3.1.1	Límites del sistema	38
3.1.2	Diagrama de casos de uso general	38
3.1.3	División en subsistemas	39
3.2	Especificación de subsistemas.....	40
3.2.1	Subsistema principal.....	40
3.2.2	Subsistema server	41
3.2.3	Subsistema player	43
3.2.4	Subsistema audio	46
3.3	Modelo de datos del sistema.....	48
3.3.1	Modelo gráfico	48
3.3.2	Catálogo de etiquetas.....	49
3.3.3	Descripción de atributos	49
3.4	Modelo de eventos del sistema	50

3.4.1	Diagramas de interacción	51
3.5	Interfaces de usuario (pantallas)	60
3.5.1	Interfaces de usuario	60
3.5.2	Relación entre pantallas y escenarios	60
3.5.3	Pantallas.....	62
3.5.4	Relación entre pantallas y eventos	62
4	DISEÑO DEL SISTEMA DE INFORMACIÓN	63
4.1	Diseño de la arquitectura del sistema	63
4.1.1	Entorno tecnológico de implantación	63
4.1.2	Entorno tecnológico de desarrollo	64
4.1.3	Especificación de estándares y normas de diseño y construcción	65
4.1.4	Especificación de subsistemas de diseño.....	69
4.2	Diseño detallado de los subsistemas de diseño.....	70
4.2.1	Subsistema principal (main.js).....	70
4.2.2	Subsistema display (display.js).....	73
4.2.3	Subsistema data (data.js)	74
4.2.4	Subsistema server (server.js)	75
4.2.5	Subsistema audio (audio.js).....	76
4.2.6	Subsistema player (player.js).....	76
4.4	Ficheros auxiliares	79
4.4.1	Plugin de video	79
5	PRUEBAS	88
5.1	Descripción general.....	88
5.2	Batería de pruebas V1.1	88
5.2.1	Requerimientos del sistema.....	88
5.2.2	CASOS DE PRUEBAS	89
5.2.3	LISTA DE CHEQUEO.....	138
6	MANUAL DEL PROGRAMADOR.....	140
6.1	Entorno de desarrollo Samsung SDK.....	140
6.1.1	Requisitos hardware	140
6.1.2	Requisitos software	141
6.2	Framework de Samsung	141
6.2.1	Introducción	141
6.2.2	Composición de una escena.....	142

6.2.3	Estructura de la aplicación.....	143
6.3	Módulos de la aplicación	145
6.4	Sistema de votación de videoclips	157
6.4.1	Especificación del sistema.....	157
6.4.2	Implementación	158
6.6	Anexo	159
6.6.1	Index.html	159
6.6.2	Scene1.html.....	160
6.6.3	Scene1.css	161
6.6.4	Main.jss.....	169
6.6.5	Player.jss	174
6.6.6	Audio.jss	179
6.6.7	Server.jss.....	179
6.6.8	Data.jss.....	182
6.6.9	Display.jss.....	184
7	MANUAL DE USUARIO	190
7.1	Instalación	190
7.1.1	Registro de la cuenta.....	191
7.1.2	Cómo crear un Smart TV ID	191
7.1.3	Samsung Apps	192
7.1.4	Instalación de FluMusic	193
7.2	Utilización de la aplicación	193
7.2.1	Menú de reproducción	195
8	MANUAL TÉCNICO DEL SISTEMA.....	198
8.1	Smart Tv.....	198
8.2	Samsung Smart Hub	199
8.3	Formatos de vídeo	199
8.3.1	Conceptos generales.....	199
8.3.2	Formatos y códecs de vídeo.....	200
8.4	Transmisión de vídeo por internet.....	203
8.4.1	Vídeo bajo demanda.....	203
8.4.2	Streaming	203
8.5	Tecnologías web	206
8.5.1	PHP	206

8.5.2	MySQL.....	206
9	POSIBLES MEJORAS	208
10	CONCLUSIONES.....	209
11	FUENTES DE DOCUMENTACIÓN.....	210
11.1	Comunidad Samsung.....	210
11.2	Manuales	210
11.3	Códigos de ejemplo	210
11.4	Servicios multimedia	210
11.5	Otros	211
12	ANEXO 1. PLAN DE PRUEBAS.....	212
12.1	Introducción	212
12.1.1	Propósito	212
12.1.2	Alcance.....	212
12.1.3	Definiciones, acrónimos y abreviaturas	214
12.1.4	Referencias	214
12.1.5	Vista general	214
12.2	Estrategia de pruebas.....	216
12.2.1	Ciclo de pruebas.....	216
12.3	Tipos de pruebas	221
12.3.1	Pruebas unitarias.....	221
12.3.2	Pruebas del sistema	222
12.3.3	Pruebas de regresión	223
12.3.4	Pruebas funcionales	223
12.3.5	Pruebas de usabilidad.....	224
12.3.6	Pruebas de aceptación.....	225
12.4	Matriz de tipificación de pruebas.....	226
12.5	Técnicas de ejecución de pruebas.....	226
12.6	Recursos humanos	228
12.7	Recurso del sistema.....	229
12.7.1	Configuración del entorno de pruebas.....	229
12.8	Criterios de inicio de ejecución	230
12.9	Criterios de evaluación.....	230
12.10	Criterios de terminación	233
12.11	Criterios de suspensión.....	233

13	ANEXO 2. DOCUMENTACIÓN PRUEBAS	234
13.1.1	Release notes	234
13.1.2	Casos de pruebas	235
13.1.3	Lista de chequeo	243

ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales	19
Tabla 2: Requisitos funcionales	20
Tabla 3: Requisitos funcionales	22
Tabla 4: Características Sony Internet Tv	22
Tabla 5: Características Lg Smart Tv	23
Tabla 6: Versiones Samsung Sdk	24
Tabla 7: Características Navegación Samsung Smart Tv	26
Tabla 8: Características Navegación LG Smart Tv	27
Tabla 9: Características Navegación Philips Smart Tv	28
Tabla 10: Características PhoneGap	30
Tabla 11: Estimación Temporal de Tareas	33
Tabla 12: Desglose Temporal de Tareas.....	34
Tabla 13: Presupuesto de ejecución material	36
Tabla 14: Presupuesto de recursos software	36
Tabla 15: Presupuesto de recursos hardware.....	37
Tabla 16: Presupuesto de implantación.....	37
Tabla 17: Presupuesto Total	37
Tabla 18: Escenario “Iniciar aplicación”	40
Tabla 19: Escenario “Cerrar aplicación”	41
Tabla 20: Escenario “Obtener ranking videoclips”	42
Tabla 21: Escenario “Obtener información videoclips”	42
Tabla 22: Escenario “Votar videoclips”	42
Tabla 23: Escenario “Obtener puntuación videoclips”	43
Tabla 24: Escenario “Reproducir videoclips”	44
Tabla 25: Escenario “Pausar videoclips”	44
Tabla 26: Escenario “Detener videoclips”	44
Tabla 27: Escenario “Rebobinado de Avance videoclips”	45
Tabla 28: Escenario “Rebobinado de Retroceso videoclips”	45
Tabla 29: Escenario “Modo pantalla completa”	45
Tabla 30: Escenario “Salir de pantalla completa”	45
Tabla 31: Escenario “Aumentar volumen”	46
Tabla 32: Escenario “Disminuir volumen”	47
Tabla 33: Escenario “Silenciar sonido”	47

Tabla 34: Escenario “Activar sonido”	47
Tabla 35: Catálogo de etiquetas	49
Tabla 36: Descripción de atributos.....	50
Tabla 37: Modelo de eventos del sistema	50
Tabla 38: Relación entre pantallas y escenarios.....	61
Tabla 39: Relación entre pantallas y eventos.....	62
Tabla 40: Especificación física de los nodos	64
Tabla 41: Especificación lógica de los nodos	64
Tabla 42: Entorno Tecnológico. Hardware	65
Tabla 43: Métodos “Subsistema principal”	72
Tabla 44: Métodos “Subsistema Display”	74
Tabla 45: Métodos “Subsistema Data”	75
Tabla 46: Métodos “Subsistema Server”	76
Tabla 47: Métodos “Subsistema Audio”	76
Tabla 48: Métodos “Subsistema Player”	78
Tabla 49: Plugin de Vídeo “Función SetDisplayArea”	80
Tabla 50: Plugin de Vídeo “Función SetInitialBuffer”	80
Tabla 51: Plugin de Vídeo “Función SetPendingBuffer”	80
Tabla 52: Plugin de Vídeo “Función Play”	81
Tabla 53: Plugin de Vídeo “Evento OnCurrentPLayTime”	81
Tabla 54: Plugin de Vídeo “Evento OnBufferingStart”	81
Tabla 55: Plugin de Vídeo “Evento OnBuffertingProgress”	82
Tabla 56: Plugin de Vídeo “Evento OnBufferingComplete”	82
Tabla 57: Plugin de Vídeo “Función Pause”	82
Tabla 58: Plugin de Vídeo “Función Stop”	83
Tabla 59: Plugin de Vídeo “Función Resume”	83
Tabla 60: Plugin de Vídeo “Función JumpForward”	83
Tabla 61: Plugin de Vídeo “Función JumpBackward”	84
Tabla 62: Plugin de Vídeo “Función GetDuration”	84
Tabla 63: Plugin de Audio “Función SetVolumenWithKey”	85
Tabla 64: Plugin de Audio “Función GetVolume”	85
Tabla 65: Plugin de Audio “Función SetSystemMute”	85
Tabla 66: Plugin TVMW “Función GetProfile”	86
Tabla 67: Plugin TVMW “Función SetProfile”	86

Tabla 68: Plugin Network “Función GetMAC”	87
Tabla 69: Batería de pruebas. requerimientos del sistema	88
Tabla 70: Caso de prueba 1-1.1	90
Tabla 71: Caso de prueba 2-1.1	91
Tabla 72: Caso de prueba 1-1.2	92
Tabla 73: Caso de prueba 1-2.1	94
Tabla 74: Caso de prueba 2-2.1	95
Tabla 75: Caso de prueba 1-2.3	97
Tabla 76: Caso de prueba 1-2.4	98
Tabla 77: Caso de prueba 1-2.5	99
Tabla 78: Caso de prueba 2-2.5	101
Tabla 79: Caso de prueba 3-2.5	102
Tabla 80: Caso de prueba 1-3.1	103
Tabla 81: Caso de prueba 2-3.1	104
Tabla 82: Caso de prueba 3-3.1	105
Tabla 83: Caso de prueba 4-3.1	107
Tabla 84: Caso de prueba 1-3.2	108
Tabla 85: Caso de prueba 2-3.2	109
Tabla 86: Caso de prueba 1-3.3	110
Tabla 87: Caso de prueba 2-3.3	112
Tabla 88: Caso de prueba 1-3.4	113
Tabla 89: Caso de prueba 1-3.5	114
Tabla 90: Caso de prueba 1-3.6	115
Tabla 91: Caso de prueba 2-3.6	116
Tabla 92: Caso de prueba 1-3.7	117
Tabla 93: Caso de prueba 1-4.1	119
Tabla 94: Caso de prueba 1-4.2	120
Tabla 95: Caso de prueba 1-4.3	121
Tabla 96: Caso de prueba 1-4.4	122
Tabla 97: Prueba de usabilidad.....	126
Tabla 98: Caso de Prueba Técnico. Pt-1.....	127
Tabla 99: Caso de Prueba Técnico. Pt-2.....	128
Tabla 100: Caso de Prueba Técnico. Pt-3.....	129
Tabla 101: Caso de Prueba Funcionales. 1.1	129

Tabla 102: Caso de Prueba Funcionales. 1.2	130
Tabla 103: Caso de Prueba Funcionales. 2.1	130
Tabla 104: Caso de Prueba Funcionales. 2.2	131
Tabla 105: Caso de Prueba Funcionales. 2.3	131
Tabla 106: Caso de Prueba Funcionales. 2.4	132
Tabla 107: Caso de Prueba Funcionales. 2.5	132
Tabla 108: Caso de Prueba Funcionales. 3.1	133
Tabla 109: Caso de Prueba Funcionales. 3.2	133
Tabla 110: Caso de Prueba Funcionales. 3.3	134
Tabla 111: Caso de Prueba Funcionales. 3.4	134
Tabla 112: Caso de Prueba Funcionales. 3.5	135
Tabla 113: Caso de Prueba Funcionales. 3.6	135
Tabla 114: Caso de Prueba Funcionales. 3.7	136
Tabla 115: Caso de Prueba Funcionales. 4.1	136
Tabla 116: Caso de Prueba Funcionales. 4.2	137
Tabla 117: Caso de Prueba Funcionales. 4.3	137
Tabla 118: Caso de Prueba Funcionales. 4.4	137
Tabla 119: Requerimientos no funcionales VS Casos de prueba técnicos	138
Tabla 120: Lista de chequeo	139
Tabla 121: Parámetros Streaming Adaptativo	151

ÍNDICE DE FIGURAS

Figura 1: Ventas Smart TV	16
Figura 2: Ventas mundiales 2012.....	20
Figura 3: Cuota de mercado 2012.....	21
Figura 4: Diagrama Samsung Smart TV	24
Figura 5: Diseño de APPs para samsung	31
Figura 6: Diagrama GANTT	35
Figura 7: Diagrama Caso de Uso General.....	38
Figura 8: División en Subsistemas.....	39
Figura 9: Diseño Caso de Uso “Principal”	40
Figura 10: Diseño de Caso de Uso: “Server”	41
Figura 11. Diseño de Caso de Uso: “Player”	43
Figura 12: Diseño Caso de Uso “Audio”	46
Figura 13. Sistema de votación: Modelo Gráfico	49
Figura 14. Diagrama de Interacción: “Iniciar Aplicación”	51
Figura 15: Diagrama de Interacción “Cerrar Aplicación”.....	52
Figura 16: Diagrama de Interacción “Obtener Ranking Videoclips”	52
Figura 17: Diagrama de Interacción “Obtener Información Videoclips”	53
Figura 18: Diagrama de Interacción “Votar Videoclips”	53
Figura 20: Diagrama de Interacción “Reproducir Videoclip”	54
Figura 21: Diagrama de Interacción “Pausar Videoclip”	55
Figura 22: Diagrama de Interacción “Detener Videoclip”	55
Figura 24: Diagrama de Interacción “Rebobinado de Retroceso”	56
Figura 25: Diagrama de Interacción “Aumentar Volumen”	57
Figura 26: Diagrama de Interacción “Disminuir Volumen”	57
Figura 27: Diagrama de Interacción “Silenciar Sonido”	58
Figura 28: Diagrama de Interacción “Activar Sonido”	58
Figura 29: Diagrama de Interacción “Modo de Pantalla Completa”	59
Figura 30: Diagrama de Interacción “Salir de Pantalla Completa”	59
Figura 31: Arquetipo de Pantalla de Aplicación	62
Figura 32: Diagrama Entorno de Implantación	63

Figura 33: Diagrama Modelo-Vista-Controlador	67
Figura 34: Estructura de Directorio App	67
Figura 35: Secuencia Cargador de Samsung.....	68
Figura 36: Diagrama De Paquetes	69
Figura 37: Herramientos de Programación Samsung SDK	140
Figura 38: Framework Samsung	142
Figura 39: Composición de Escena.....	142
Figura 40: Estructura de Directorios Proyecto.....	143
Figura 41: Cargador de Samsung	144
Figura 42: Módulos Aplicación	145
Figura 43: Fichero “config.xml”	146
Figura 44: Estructura Pantalla Aplicación.....	148
Figura 45: Entorno de votación: Especificación del sistema.....	157
Figura 46: Pantalla de Visualización	190
Figura 47: Registro Cuenta	191
Figura 48: Pantalla almacén Samsung APP.....	192
Figura 49: Ejemplo de pantalla de instalación	193
Figura 50: Pantalla Aplicación	194
Figura 51: Sección Información Videoclip	195
Figura 52: Barra de Progreso	195
Figura 53: Barra de Progreso “Reproduciendo”	195
Figura 54: Barra de Progreso “Pausa”.....	196
Figura 55: Selector de videoclip	196
Figura 56: Televisor Samsung.....	198
Figura 57: Convergencia Samsung	199
Figura 58: Lógica de Los Códecs.....	202
Figura 59: Streaming Adaptatvo.....	205

1 INTRODUCCIÓN

Desde que en 1926 fuese presentado el primer televisor comercial de la historia hasta la actualidad, la televisión ha experimentado innumerables cambios. Hoy en día ya no se demanda una programación encorsetada, sino una programación a la carta con contenidos bajo demanda, que convierten a los usuarios en administradores de su propio entretenimiento.

Todo ello sumado a la incipiente televisión interactiva (SMART TV), propiciada por la irrupción y consolidación de los terminales móviles, que contagiada por el éxito de los smartphones, pretende convertir al televisor en el centro del entretenimiento multimedia del hogar. Visionar contenidos bajo demanda, ejecutar aplicaciones en un televisor, desplazará a los ordenadores convencionales poco a poco de este sector del entretenimiento. Y todo bajo el prisma de una flamante convergencia tecnológica, cuya principal máxima es alcanzar la completa interoperabilidad entre dispositivos electrónicos.

Es por ello que se antoja una necesidad, y a la vez una interesante oportunidad de mercado, el desarrollo de aplicaciones para este nuevo sector que, en la actualidad, todavía está por explotar.

2 ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS)

2.1 Establecimiento del alcance del sistema

El proyecto propone el desarrollo de una aplicación para una plataforma de Smart TV, orientada hacia la generación de una radio interactiva con emisión de videoclips la cual publicará una lista de éxitos en función de los votos de sus oyentes.

Los oyentes podrán acceder a la aplicación a través del SmartTV, ver el estado de la lista de éxitos y visualizar los videoclips de las canciones más votadas.

La aplicación necesita de la fusión de las tecnologías más actuales como HTML5, CSS3, audio y vídeo bajo demanda y en directo, jQuery, a desarrollar bajo el nuevo entorno SDK 4 de SAMSUNG SMART TV.

2.2 Estudio de la situación actual

En la actualidad se ha producido una extraordinaria tendencia hacia la nueva generación de televisiones, que tiene por reclamo una hibridación entre la televisión y los dispositivos móviles. La televisión se está convirtiendo en un nuevo medio de comunicación interactivo y organizado por la propia comunidad usuaria, y las mejores marcas de televisión ya se están adaptando a las nuevas demandas de los consumidores.

En la actualidad los principales fabricantes de televisores inteligentes disponen de los siguientes productos:

- **SAMSUNG.** SAMSUNG SMART TV
- **SONY.** SONY INTERNET TV
- **LG.** LG SMART TV
- **APPLE.** APPLE TV

Los televisores Smart suponen ya el 36% de las ventas en el mercado europeo (2,1 millones de unidades), el 32% en el chino y hasta el 46% en el japonés. Las ventas de estos televisores conectados también son altas en los mercados emergentes.

**FIGURA 1: VENTAS SMART TV**

Esto combinado con la escasez de un heterogéneo elenco de aplicaciones, y promovido por la sobresaliente oportunidad de negocio, induce a la innovación y generación de aplicaciones en un sector tan pujante.

2.3 Definición de los requisitos del sistema

2.3.1 Requisitos funcionales

Identificador	Descripción	Prioridad
RF1. Listado de éxitos musicales		
RF1.1	La aplicación debe ser capaz de recuperar listados musicales de un servidor.	ALTA
RF1.2	La aplicación debe ser capaz de ordenar el listado en función de las votaciones recibidas.	ALTA
Requisitos sobre la estructura de los datos		
RF.1.2.1	Los elementos musicales tendrán los siguientes elementos: <ol style="list-style-type: none"> 1 Título 2 Link 3 Descripción 	ALTA
RF1.3	La aplicación debe mostrar en pantalla el listado recibido.	ALTA
Requisitos sobre la estructura de los datos		
RF 1.3.1	La información a mostrar debe contener los siguientes items: <ol style="list-style-type: none"> 1 Título 2 Link 3 Descripción 	ALTA
RF 1.4	La aplicación debe permitir reproducir cada video recibido del servidor en streaming.	ALTA
RF 2	Reproductor de vídeo	

RF 2.1	El reproductor de video debe disponer de la funcionalidad básica de un reproductor de video: play, pause, fast-forward, rewind.	ALTA
RF 2.2	El reproductor de vídeo debe admitir una reproducción en streaming de cada videoclip (con extensión mp4), del que recibirá un enlace.	ALTA
RF 2.3	El reproductor debe mostrar una imagen de previsualización cuando se ponga el foco sobre un elemento de la lista.	MEDIA
RF 2.4	El reproductor debe disponer de un buffer lo suficientemente grande para evitar interrupciones de video.	MEDIA
RF 3	Votación de los videoclips	
RF 3.1	El sistema debe permitir la votación de las canciones.	BAJA
Requisitos sobre la estructura de los datos		
RF 3.1.1	Se contemplarán las siguientes categorías para puntuar los videoclips: <ul style="list-style-type: none"> • 1 estrella • 2 estrellas • 3 estrellas • 4 estrellas 	BAJA
RF3.1.2	Se debe enviar identificador del televisor desde el cual se realiza la votación.	BAJA
RF 3.1.3	Se debe enviar	BAJA

	identificador de la versión de la lista que se vota.	
RF 3.2	Se visualizará el promedio de votos para cada uno de los videoclips.	BAJA

TABLA 1: REQUISITOS FUNCIONALES**2.3.2 Requisitos No Funcionales**

Identificador	Descripción	Prioridad
RNF1	Eficiencia La aplicación debe ser eficiente minimizando la espera del usuario en la carga y tratamiento de los archivos. Ha de optimizar el uso de los recursos del televisor.	MEDIA
RNF2	Fiabilidad La aplicación ha de ser fiable, evitando todo tipo de excepciones e integrando sólo elementos suficientemente probados.	ALTA
RNF3	Usabilidad Se tendrá en cuenta la comodidad del usuario realizando un entorno amigable.	ALTA
RNF4	Tiempos de respuesta Para los accesos y descarga de vídeos, el tiempo de respuesta no debe superar los 7 segundos en la carga, con una conexión a internet media (de 2 Mb).	MEDIA

RNF5	Mantenimiento y nuevas necesidades La aplicación debe de facilitar en la medida de lo posible la realización de futuros cambios o nuevas funcionalidades.	MEDIA
-------------	---	-------

TABLA 2: REQUISITOS FUNCIONALES

2.4 Estudio de las alternativas de la solución

A continuación se analizarán teniendo en cuenta dos factores: volumen de ventas, madurez del entorno de desarrollo, y especificaciones funcionales del producto.

2.4.1.1 Volumen de ventas

Marcas más vendidas a nivel mundial (excluidas marcas fuera del estudio)

1. **Samsung**
2. **LG**
3. **SONY**
4. **APPLE**

Aunque el escenario se encuentra un tanto difuminado, se puede constatar que Samsung lidera con holgura las ventas de televisores, en el mercado mundial y por extensión, en el español, con una cuota de mercado a nivel mundial superior al 25%.

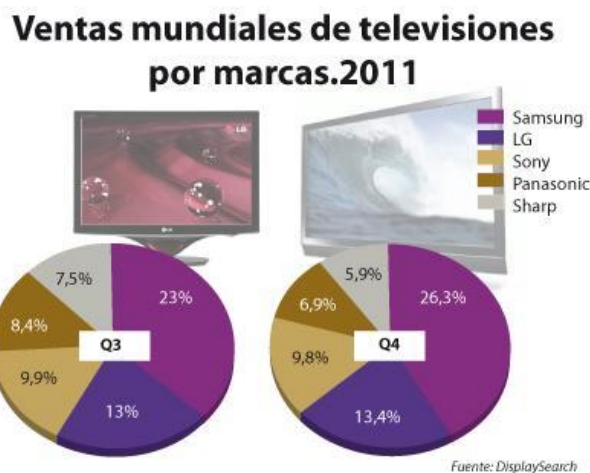


FIGURA 2: VENTAS MUNDIALES 2012

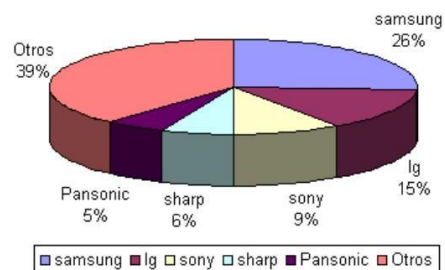


FIGURA 3: CUOTA DE MERCADO 2012

2.4.1.2 Especificaciones funcionales de los televisores SAMSUNG. SAMSUNG SMART TV

Tecnología	Necesidades	Soportado	
HTML5	Tags genéricos	Alto a partir de la versión 2.5 - 2011+	
CSS3	Modificadores genéricos	Alto a partir de la versión 2.5 - 2011+	
DOM		Completo	
Flash	SWF	Actualizado a la versión 11.1	
	Streaming	RTMP/RTMPe	
DRM		No soportado	
	Streaming	HTTP	Soportado
		HTTPS	Soportado
		MMS*2	Soportado
		RTP/RTSP	Soportado (2012+ – SDK 3.0+)
VOD	Adaptative Streaming	HAS (xml metadata)	No soportado
		Mpeg-Dash (xml metadata)	Soportado
		HLS (m3u8 metadata)	Soportado
		HLS audio	Soportado
Live Streaming (Adaptive Streaming)	Adaptative Streaming	HAS LS	No Soportado
		HLS LS	Soportado

	WidevineLS	No soportado
--	------------	--------------

TABLA 3: REQUISITOS FUNCIONALES

En la tabla se puede observar que dispone de todos los requisitos exigidos para la implementación de la aplicación propuesta. Dispone de motor de navegación web que soporta jQuery, reproducción de contenidos bajo demanda, entre otros.

SONY. SONY INTERNET TV

Tecnología	Necesidades	Soportado	
HTML5	Tags genéricos	Completo	
CSS3	Modificadores genéricos	Completo	
DOM		Completo	
Flash	SWF	Soportado	
	Streaming	Soportado	
DRM		No soportado	
	Streaming	HTTP	Soportado
		HTTPS	Soportado
		MMS*2	Desconocido
		RTP/RTSP	Desconocido
VOD	Adaptative Streaming	HAS (xml metadata)	No soportado
		Mpeg-Dash (xml metadata)	Soportado
		HLS (m3u8 metadata)	Soportado
		HLS audio	Soportado
Live Streaming (Adaptive Streaming)	Adaptative Streaming	HAS LS	No Soportado
		HLS LS	Soportado
		WidevineLS	No soportado

TABLA 4: CARACTERÍSTICAS SONY INTERNET TV

La alianza de Google con Sony Corp. ha brindado la posibilidad de integrar el sistema operativo Android para satisfacer todas las necesidades exigidas en un desarrollo de estas características.

LG. LG SMART TV

Tecnología	Necesidades	Soportado
HTML5	Tags genéricos	Completa
	Vídeo	Alta. Elementos básicos, completa.
	Audio	Alta. Elementos básicos, completa.
CSS3	No especifica las características que soporta, simplemente que lo hace parcialmente.	
	DOM	Completo
Formatos de vídeo soportados	<ul style="list-style-type: none"> • MPEG 4 • OGG 	
Interacción con el usuario	No especificado	
Formatos de streaming soportados	Flash	<ul style="list-style-type: none"> • SW Codec On2 , Sorenson - Low quality video (because of SW Codec limitation) • H.264 (with AAC audio only)
	VOD	HTTP/HTTPS Supported HLS Supported Widevine Supported Smooth Streaming Not Supported (*Diff_NC3.0)
	Live Streaming (Adaptive Streaming)	Linear or live broadcasting : HLS Supported Widevine Supported Smooth Streaming Not Supported (*Diff_NC3.0)

TABLA 5: CARACTERÍSTICAS LG SMART TV

En línea con los anteriores, cubre todos los requisitos técnicos requeridos para el desarrollo del proyecto.

APPLE. APPLE TV

El principal escollo de este sistema es el lenguaje de programación requerido para el desarrollo de sus aplicaciones. Aunque con alternativas como Velneo se podría programar para posteriormente desplegar en diferentes sistemas. No obstante, la necesidad de formarse en su herramienta y el coste de la licencia para proyectos empresariales abocan a descartar la opción.

También cabe recordar que la distribución e instalación de una APP en un dispositivo de Apple requiere de la autorización de desarrollador -previo pago a Apple, lo cual supone un sobre coste innecesario para este proyecto.

2.4.1.3 Madurez del entorno de desarrollo

SAMSUNG SMART TV

Samsung dispone de su propio entorno de desarrollo Samsung SDK integrado en Eclipse. Este cuenta con un recorrido de más de 3 años, lo cual exhibe una evolución con razonables garantías de madurez.

Versiones SDK SAMSUNG	2010	2011	2012	2013
	TV/AV SDK 1.5	TV/AV SDK 2.5	TV/AV SDK 3.0	TV/AV SDK 4.0

TABLA 6: VERSIONES SAMSUNG SDK



FIGURA 4: DIAGRAMA SAMSUNG SMART TV

SONY INTERNET TV

Gracias a la alianza con google, el desarrollo a través de Android mediante su entorno, no sólo permite una gran integración entre dispositivos android, sino que también facilita la implementación de apps y garantiza un entorno de programación con gran madurez, estabilidad y documentación.

LG SMART TV

LG dispone de un entorno SDK al igual que Samsung, con menor tiempo de existencia (enero 2012) pero con más versiones, la última data de febrero de 2013. Dispone de documentación y foro de desarrolladores, aunque la documentación y frameworks escasean con respecto a su competidor Samsung.

APPLE TV

Descartada por la política de desarrollo de aplicaciones de Apple, y por el coste de su cuota anual imprescindible para volcar aplicaciones a la plataforma itunes, que es la única vía para instalar la aplicación en el dispositivo de Apple de forma lícita.

2.4.2 Valoración de las alternativas

2.4.2.1 Solución Web

La aplicación web se plantea como la solución más genérica de todas las estudiadas, pero necesita de que la gran mayoría de las soluciones de Smart TV existentes en el mercado dispongan de un navegador web que soporte las tecnologías de desarrollo web actuales. Los estudios están basados en función de los siguientes parámetros:

- HTML5
 - En especial en tags de audio y vídeo
- CSS3
- Javascript
- Websocket
- Vídeo bajo demanda / streaming
 - Protocolos aceptados

2.4.2.1.1 Motores de navegación web

Samsung SmartTV

La solución aportada por Samsung es una de los más completos y extendidos, lo que nos permitiría llegar a un mayor número de público, y además a un coste bastante ajustado. Su sistema está basado en Android, sistema operativo de código abierto, posee una gran comunidad de desarrolladores y reduce los costes de desarrollo.

Tecnología	Necesidades	Soportado
HTML5	Tags genéricos	Alto a partir de la versión 2.5 - 2011+
	Vídeo	A partir de la versión 2.5 – 2011+ excluida el tag “track”
	Audio	Soportado a partir de la versión 2.5 – 2011+
	Canvas	Completa
CSS3	2D Transformaciones	Prácticamente completa, tan solo excluye las transformaciones de elementos inline
	3D Transformaciones	Prácticamente completa, excluyendo la propiedad <i>scale Z</i>
	Selectores	Casi completa, excluye: <ul style="list-style-type: none"> • :enabled on a fieldset element • :indeterminate and input type=radio • :nth-child selector and CSS comments
DOM		Completo
Formatos de vídeo soportados	Formatos de vídeo genéricos <ul style="list-style-type: none"> • MPEG 4 • OGG Formato de vídeo opcional: <ul style="list-style-type: none"> • WEBM 	<ul style="list-style-type: none"> • MPEG 4 • OGG
	<ul style="list-style-type: none"> • Atributos ocultos • Scroll • Edición de contenido • Drag & drop • Deshacer 	<ul style="list-style-type: none"> • Atributos ocultos • Scroll • Edición de contenido • Drag & drop
Formatos de streaming soportados	Flash	
	VOD	Streaming
		Adaptive Streaming
	Live Streaming (Adaptive Streaming)	
RTMP/RTMPe en todas las versiones HTTP, HTTPS, MMS*2, RTP/RTSP en todas las versiones <ul style="list-style-type: none"> • Mpeg-Dash 3.0+ • HLS(m3u8 metadata) 2.5+ NO soportado <ul style="list-style-type: none"> • HAS • HLS audio • HLS LS 2.5 + 		

TABLA 7: CARACTERÍSTICAS NAVEGACIÓN SAMSUNG SMART TV

LG SmartTV

Tecnología	Necesidades	Soportado
HTML5	Tags genéricos	Alta
	Vídeo	Completa
	Audio	Completa
CSS3	No especifica las características que soporta, simplemente que lo hace parcialmente.	
	DOM	Completo
Formatos de vídeo soportados	<ul style="list-style-type: none"> • MPEG 4 • OGG 	
	Interacción con el usuario	No especificado
Formatos de streaming soportados	Flash	Soportado
	VOD	HTTP/HTTPS Supported HLS Supported Widevine Supported Smooth Streaming Not Supported (*Diff_NC3.0)
	Live Streaming (Adaptive Streaming)	Linear or live broadcasting : HLS Supported Widevine Supported Smooth Streaming Not Supported (*Diff_NC3.0)

TABLA 8: CARACTERÍSTICAS NAVEGACIÓN LG SMART TV

Sony Internet TV / Google TV

El navegador de Sony Internet TV es Google Chrome completo con todas sus funcionalidades. Por lo que todas nuestras necesidades están cubiertas.

Philips Smart TV

Tecnología	Necesidades	Soportado
HTML5	Tags genéricos	Alta
	Vídeo	Completa
	Audio	Completa
	Canvas	Completa
CSS3	2D Transformaciones	Prácticamente completa, tan solo excluye las transformaciones de elementos inline
	3D Transformaciones	Prácticamente completa, excluyendo la propiedad <i>scale Z</i>

	Selectores	Casi completa, excluye: <ul style="list-style-type: none"> • :enabled on a fieldset element • :indeterminate and input type=radio • :nth-child selector and CSS comments
DOM		Completo
Formatos de vídeo soportados	<ul style="list-style-type: none"> • MPEG 4 	
Interacción con el usuario	No especificado	
	Flash	Soportado
Formatos de streaming soportados	VOD	No especifica formatos, tan solo que soporta el protocolo
	Live Streaming (Adaptive Streaming)	No especificado

TABLA 9: CARACTERÍSTICAS NAVEGACIÓN PHILIPS SMART TV

Apple TV

La solución de Apple está orientada única y exclusivamente a sus dispositivos. No tiene navegador web, con lo cual, queda descartado en nuestro desarrollo web.

2.4.2.2 Conclusiones

Existe un gran abanico de posibilidades a la hora de decidir plataforma para el desarrollo de una aplicación para televisión. Una vez definidas las necesidades a cubrir en el proyecto, el resultado obtenido, en función del tanto por ciento soportado, público y repercusión actual y futuro, por las plataformas fue el siguiente.

2.4.2.2.1 Samsung Smart TV

El SmartTV desarrollado por Samsung cubre el 100% de las necesidades tecnológicas y soporta un gran formato de protocolos de streaming, lo que nos da mayor flexibilidad a la hora de elegir un servidor de vídeo bajo demanda. Además, la marca coreana es actualmente la líder de ventas de televisores Smart tv en España, lo que la coloca como la mejor puntuada en este sentido.

2.4.2.2.2 Sony Internet TV / Google TV

El SmartTV impulsado por Google y apoyado por Sony cuenta con la gran ventaja de que está desarrollado sobre su sistema operativo Android. Puntuando la parte tecnológica obtiene la misma nota que Samsung, ya que soporta todo lo necesario. Donde obtiene peor nota es en el público al que podríamos llegar y la repercusión que tendría la aplicación, ya que Sony actualmente tiene una cuota de mercado inferior a Samsung o LG. Por tanto, queda como la tercera mejor puntuada.

2.4.2.2.3 LG

La solución impulsada por LG está basada en Linux, y permite tanto desarrollo de aplicaciones como desarrollo para su navegador web, el cual cumple con todos los requisitos fijados por la aplicación. Además, en cuanto a número de ventas está por encima de Sony, por lo que queda como la segunda mejor valorada.

2.4.2.2.4 Apple TV

En la actualidad, Apple TV no permite la instalación de aplicación y tampoco cuenta con un navegador web, por lo que queda descartada.

2.4.2.3 SOLUCIÓN NATIVA

En esta solución se plantea el desarrollo de una aplicación para cada uno de los sistemas operativos más difundidos en dispositivos móviles. Estos son Android, iOS y Windows 8.

Pero, atendiendo al tiempo de desarrollo y el conocimiento de los recursos disponibles de cara al primer incremento, la solución nativa se tendría que centrar en un sistema operativo, concretamente Android. El desarrollo sobre iOS sería más costoso al ser necesario invertir en el entorno de desarrollo y en la formación del equipo de desarrollo en Object C.

En cuanto a Windows 8 aún se encuentra bastante por detrás en difusión de Android e iOS. La desventaja lógica es que se perderían de golpe todos clientes potenciales que no dispongan de un dispositivo Android.

Como ventajas, desarrollar nativamente para un dispositivo siempre es mucho más potente que hacerlo de forma genérica, donde se pierden todas las funciones nativas, como por ejemplo, recoger eventos de otras aplicaciones, avisos en segundo plano, etc. que para la aplicación que vamos a desarrollar sería bastante interesante tener.

SOLUCIÓN GENÉRICA PHONEGAP – APACHE CORDOVA

PhoneGap es un framework creado para el desarrollo de aplicaciones móviles que permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML 5 y CSS3.

Las aplicaciones resultantes son híbridas, es decir, que no son aplicaciones nativas al operativo donde van a correr, pero tampoco son aplicaciones web ya que el empaquetado está preparado para el despliegue en el sistema operativo.

Listado de características de cada sistema operativo accesibles con PhoneGap:

Característica	iPhone / iPhone 3G	iPhone 3Gs +	Android 1.0 – 4.2	Windows Phone
Acelerómetro	Si	Si	Si	Si
Cámara - Fotos	Si	Si	Si	Si
Brújula	No	Si	Si	Si
Contactos	Si	Si	Si	Si
Sistema de ficheros	Si	Si	Si	Si
Geo localización	Si	Si	Si	Si
Reproductores de Audio y Vídeo	Si	Si	Si	Si
Red	Si	Si	Si	Si
Alertas visuales	Si	Si	Si	Si
Sonido	Si	Si	Si	Si
Vibración	Si	Si	Si	Si
Almacenamiento	Si	Si	Si	Si
Lector de Barcode	Si	Si	Si	No

TABLA 10: CARACTERÍSTICAS PHONEGAP

2.4.2.4 Conclusiones

Como podemos ver en la tabla anterior PhoneGap nos permite acceder a un buen número de recursos del dispositivo móvil, pero siempre y cuando la aplicación esté corriendo en primer plano.

La ventaja más clara es que el desarrollo es genérico para los sistemas operativos más difundidos en dispositivos móviles, Android, iOS, Windows Phone, Blackberry, etc. y una curva de aprendizaje suave.

En la API actual para el stream de vídeo y audio aún no está soportado por los actuales navegadores de iOS y Android, por lo que si se quiere emitir audio y vídeo en directo desde un dispositivo móvil la solución no puede ser desarrollada con PhoneGap. Esto puede suponer un problema si se desea implementar sucesivas mejoras.

Por otra parte, el fabricante Samsung no dispone de soporte para Android todavía en sus televisores. Sólo se puede desarrollar en su entorno de programación.

2.5 Solución

Atendiendo a la cuota de mercado actual de venta de televisores, en la que Samsung predomina con cierta distancia respecto a sus competidores, a las limitaciones de algunos frameworks comerciales que no están optimizados o desarrollados para la reproducción de video bajo demanda, y a los conocimientos previos en las tecnologías web, se opta por un desarrollo nativo para la plataforma SDK 4.0 de Samsung.

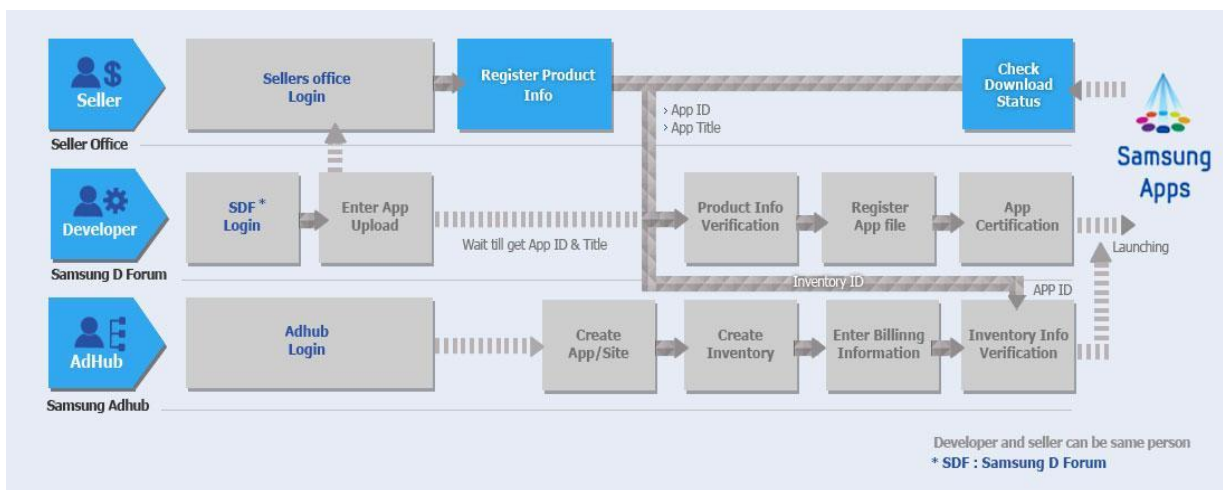


FIGURA 5: DISEÑO DE APPS PARA SAMSUNG

2.6 Planificación y presupuesto

2.6.1 Planificación

Las tareas en las que se dividirá el Proyecto son las siguientes:

- Estudio de viabilidad (EVS).
- Análisis del sistema (ASI).
- Diseño (DSI).
- Implementación (CSI).
- Pruebas (PRU).
- Documentación.
- Implantación.
- Formación técnica.

2.6.2 Identificación de las tareas a realizar

Estudio de viabilidad

Esta tarea implica comprobar que es viable llevar a cabo el desarrollo del proyecto. Se establecerá un catálogo de requisitos junto con el cliente, así en posteriores etapas del proyecto se tendrá en cuenta este catálogo de forma que la nueva aplicación cubra sus necesidades. Se realizará una prospección tecnológica para conocer el estado del arte de la técnica.

Análisis del sistema

Durante esta fase se obtendrá una especificación del sistema que se pretende desarrollar. Una vez hecho esto se realizará una división en subsistemas, reflejando la composición de cada uno de ellos. Asimismo se creará un esbozo del mapa de pantallas del sistema final.

Diseño

El diseño es el siguiente paso al análisis del sistema. Consiste en obtener una descripción más detallada de los componentes del sistema, utilizando toda la información obtenida en los pasos anteriores, de forma que se satisfagan todos los requisitos iniciales.

Implementación

La fase de implementación consiste en el desarrollo real de la aplicación por medio de la codificación en los lenguajes de programación elegidos de los distintos componentes del sistema.

Pruebas

Tras la implementación de los distintos componentes, se procederá a realizar pruebas, estableciendo previamente el correspondiente Plan de Pruebas.

Documentación

Preciso al comienzo y paralelamente a la realización de las demás tareas, se irá construyendo la documentación de las mismas.

El objetivo principal de la documentación es facilitar el futuro mantenimiento del sistema, de forma que una lectura de la misma, transmita la suficiente información como para poder llevar a cabo una modificación o ampliación de la aplicación de forma adecuada y coherente.

Implantación

Esta tarea consiste en incorporar de forma definitiva la aplicación al catálogo de aplicaciones de la tienda de apps de SAMSUNG.

2.6.3 Estimación temporal de las tareas a realizar

Tarea	Duración (horas)	Perfil
Estudio de viabilidad (EVS)	30	ANALISTA
Análisis del sistema (ASI)	56	ANALISTA
Diseño (DSI)	40	ANALISTA
Diseño de Pruebas	8	ANALISTA
Implementación (CSI)	232	PROGRAMADOR
Pruebas (PRU)	17	TESTEADOR
Integración módulos	22	PROGRAMADOR
Implantación	8	OPERADOR
Documentación	--	TODOS
TOTAL	413	

TABLA 11: ESTIMACIÓN TEMPORAL DE TAREAS

Nombre de tarea	Duración	Comienzo	Fin
Proyecto	54 días	lun 04/03/13	jue 16/05/13
Análisis	16,75 días	lun 04/03/13	mar 26/03/13
Estudio de viabilidad (EVS)	30 horas	lun 04/03/13	jue 07/03/13
Análisis del sistema (ASI)	56 horas	jue 07/03/13	lun 18/03/13
Diseño (DSI)	40 horas	lun 18/03/13	lun 25/03/13
Diseño de pruebas	8 horas	lun 25/03/13	mar 26/03/13
Adquisiciones previas al desarrollo	2 días	lun 04/03/13	mar 05/03/13
Desarrollo	34,13 días	mar 26/03/13	lun 13/05/13
Subsistema principal	9,5 días	mar 26/03/13	mar 09/04/13
Modulo principal	64 horas	mar 26/03/13	vie 05/04/13
Prueba I	2 horas	vie 05/04/13	vie 05/04/13
Corrección de pruebas I	6 horas	lun 08/04/13	lun 08/04/13
Pruebas II	2 horas	lun 08/04/13	lun 08/04/13
Corrección de problemas II	2 horas	mar 09/04/13	mar 09/04/13
Subsistema de datos	1,25 días	mar 09/04/13	mié 10/04/13
Desarrollo módulo data	8 horas	mar 09/04/13	mié 10/04/13

Pruebas	1 hora	mié 10/04/13	mié 10/04/13
Corrección de problemas	1 hora	mié 10/04/13	mié 10/04/13
Subsistema server	2,63 días	mié 10/04/13	lun 15/04/13
Desarrollo módulo server	16 horas	mié 10/04/13	vie 12/04/13
Pruebas	4 horas	vie 12/04/13	vie 12/04/13
Corrección de pruebas	1 hora	lun 15/04/13	lun 15/04/13
Subsistema de reproducción	6,63 días	lun 15/04/13	mar 23/04/13
Desarrollo módulo reproducción	46 horas	lun 15/04/13	lun 22/04/13
Pruebas I	3 horas	lun 22/04/13	mar 23/04/13
Corrección de problemas I	2 horas	mar 23/04/13	mar 23/04/13
Pruebas II	1 hora	mar 23/04/13	mar 23/04/13
Corrección de problemas II	1 hora	mar 23/04/13	mar 23/04/13
Subsistema de audio	1,25 días	mar 23/04/13	mié 24/04/13
Desarrollo módulo audio	8 horas	mar 23/04/13	mié 24/04/13
Pruebas I	1 hora	mié 24/04/13	mié 24/04/13
Corrección de problemas I	1 hora	mié 24/04/13	mié 24/04/13
Subsistema visual	9 días	jue 25/04/13	mar 07/05/13
Desarrollo subsistema	62 horas	jue 25/04/13	lun 06/05/13
Pruebas	2 horas	lun 06/05/13	lun 06/05/13
Corrección de pruebas I	8 horas	mar 07/05/13	mar 07/05/13
Integración	3,88 días	mié 08/05/13	lun 13/05/13
Integración de subsistemas	22 horas	mié 08/05/13	vie 10/05/13
Pruebas I	2 horas	vie 10/05/13	vie 10/05/13
Corrección de problemas I	4 horas	lun 13/05/13	lun 13/05/13
Pruebas II	1 hora	lun 13/05/13	lun 13/05/13
Corrección de problemas II	2 horas	lun 13/05/13	lun 13/05/13
Manual de usuario	0,5 días	mar 14/05/13	mié 15/05/13
Pruebas de aceptación	8 horas	lun 13/05/13	mar 14/05/13
Implantación	3 horas	mié 15/05/13	mié 15/05/13
Documentación	54 días	lun 04/03/13	jue 16/05/13
Cierre de proyecto	1 hora	mié 15/05/13	mié 15/05/13

TABLA 12: DESGLOSE TEMPORAL DE TAREAS

La documentación es una tarea compartida entre todos los perfiles implicados en el proyecto. Por ello sus horas están desglosadas por perfil.

El siguiente diagrama representa de forma atemporal la duración de las tareas, así como las que pueden realizarse simultáneamente. Cada división representa una jornada estándar de trabajo de ocho horas.

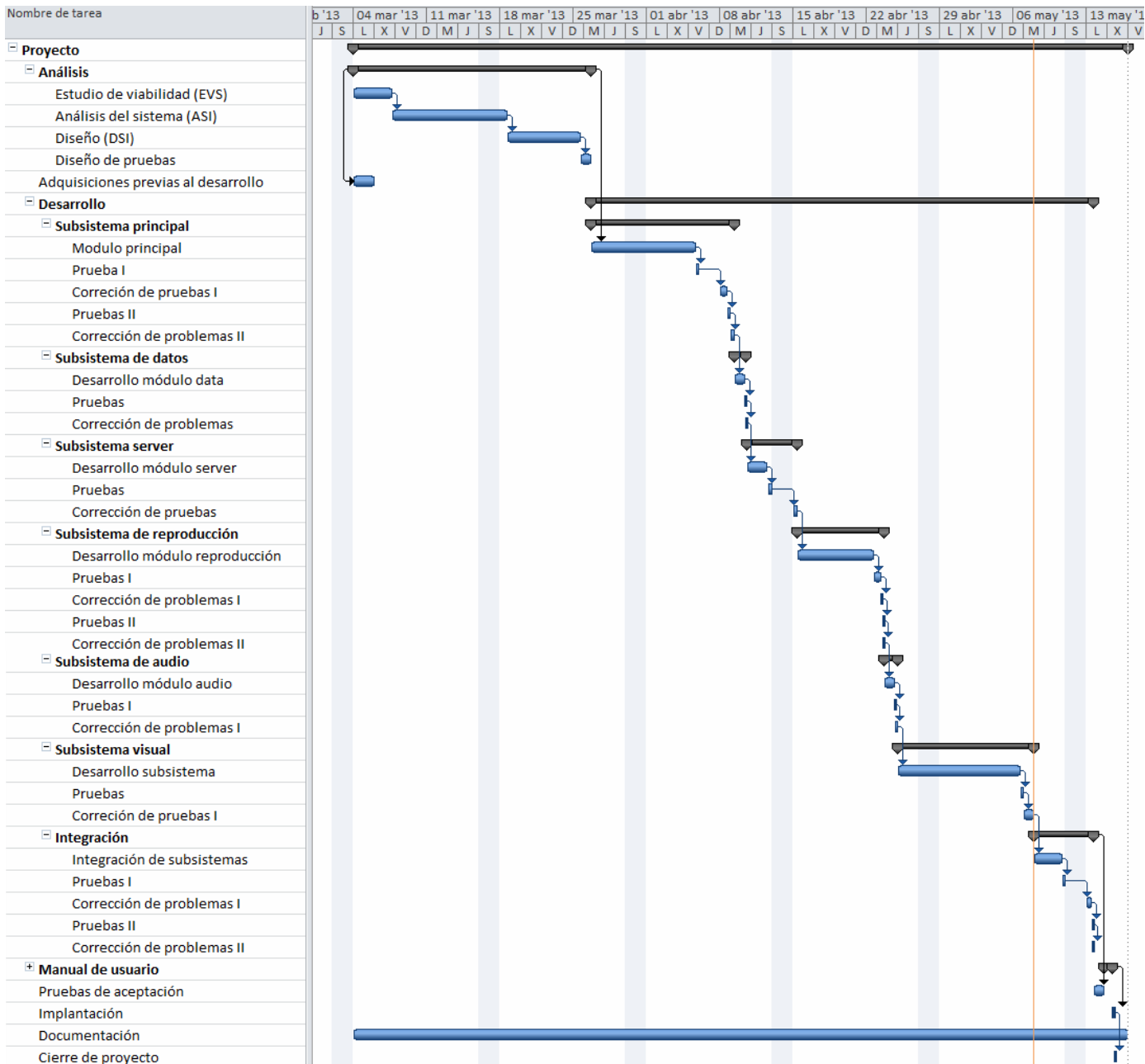


FIGURA 6: DIAGRAMA GANTT

2.6.4 Presupuestos parciales

2.6.4.1 Presupuesto de ejecución material

Id	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
C1-101	Estudio de viabilidad	30	Hora	40 €	1.200 €
C1-102	Análisis y Diseño	104	Hora	40 €	4.160 €
C1-103	Implementación	232	Hora	25 €	5.800 €
C1-104	Pruebas	17	Hora	20 €	340 €
C1-105	Integración	22	Hora	25 €	550 €
C1-106	Implantación	8	Hora	20 €	160 €
					12.210 €

TABLA 13: PRESUPUESTO DE EJECUCIÓN MATERIAL

2.6.4.2 Presupuesto de recursos software

Id	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
C2-101	Sistema Operativo Windows 7	1	Unidad	0 €	87 €
C2-102	Microsoft Visio	1	Unidad	0 €	300 €
C2-103	Microsoft Project	1	Unidad	0 €	1.168 €
C2-104	XAMPP 1.7.4	1	Unidad	0 €	0 €
C2-105	Samsung SDK 4.0	1	Unidad	0 €	0 €
C2-106	HTML5	1	Unidad	0 €	0 €
C2-107	CSS3	1	Unidad	0 €	0 €
C2-108	jQuery	1	Unidad	0 €	0 €
C2-109	Streaming de Audio y vídeo	1	Unidad	0 €	0 €
					1.555 €

TABLA 14: PRESUPUESTO DE RECURSOS SOFTWARE

2.6.4.3 Presupuesto de recursos hardware

Id	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
C3-101	Ordenador personal	1	Unidad	0 €	0 €
C3-102	Televisor Samsung UE32ES6100	1	Unidad	483 €	483 €
					483 €

TABLA 15: PRESUPUESTO DE RECURSOS HARDWARE**2.6.4.4 Presupuestos de implantación**

Id	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
C4-101	Tienda de aplicaciones de Samsung	3	Hora	20 €	60 €
					60 €

TABLA 16: PRESUPUESTO DE IMPLANTACIÓN**2.6.5 Presupuesto total**

Id	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
D-101	Ejecución material	1	Unidad	0 €	12.210 €
D-102	Recursos software	1	Unidad	0 €	1.555 €
D-103	Recursos hardware	1	Unidad	0 €	483 €
					14.248 €

TABLA 17: PRESUPUESTO TOTAL

3 ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI)

3.1 Especificaciones del sistema

3.1.1 Límites del sistema

En este punto se analizará la creación de una aplicación para SMART TV que cubra los requisitos del sistema anteriormente expuestos.

3.1.2 Diagrama de casos de uso general

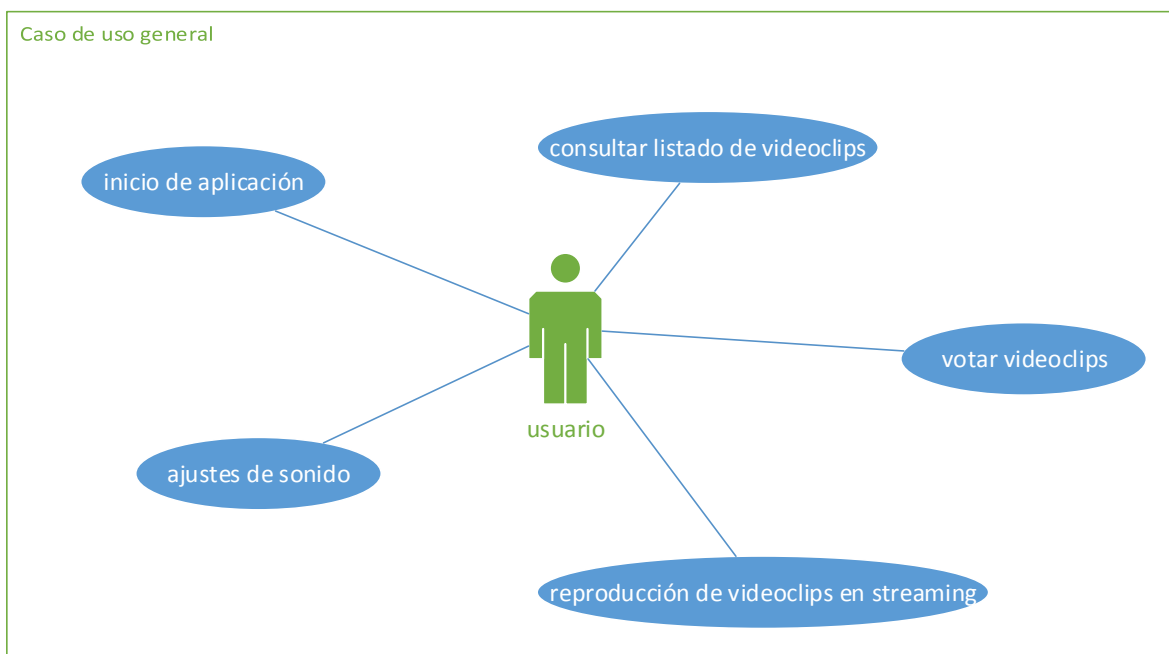


FIGURA 7: DIAGRAMA CASO DE USO GENERAL

3.1.2.1 Descripción de los actores

A continuación se enumeran los distintos tipos de usuarios que interactuarán con el sistema:

- **Usuario:** cualquier usuario que instale la aplicación en un televisor SAMSUNG SMART TV.

3.1.3 División en subsistemas

- **Subsistema principal:** Responsable de coordinación de las funcionalidades de la aplicación.
- **Subsistema data:** Se ocupa de almacenar los datos recibidos por la aplicación.
- **Subsistema server:** Responsable de interactuar con otros servidores, intercambiando información: listado de videoclips (recepción) y votación de los videoclips (transmisión).
- **Subsistema audio:** Controla el nivel de volumen de audio.
- **Subsistema player:** Responsable de la reproducción de los videoclips.

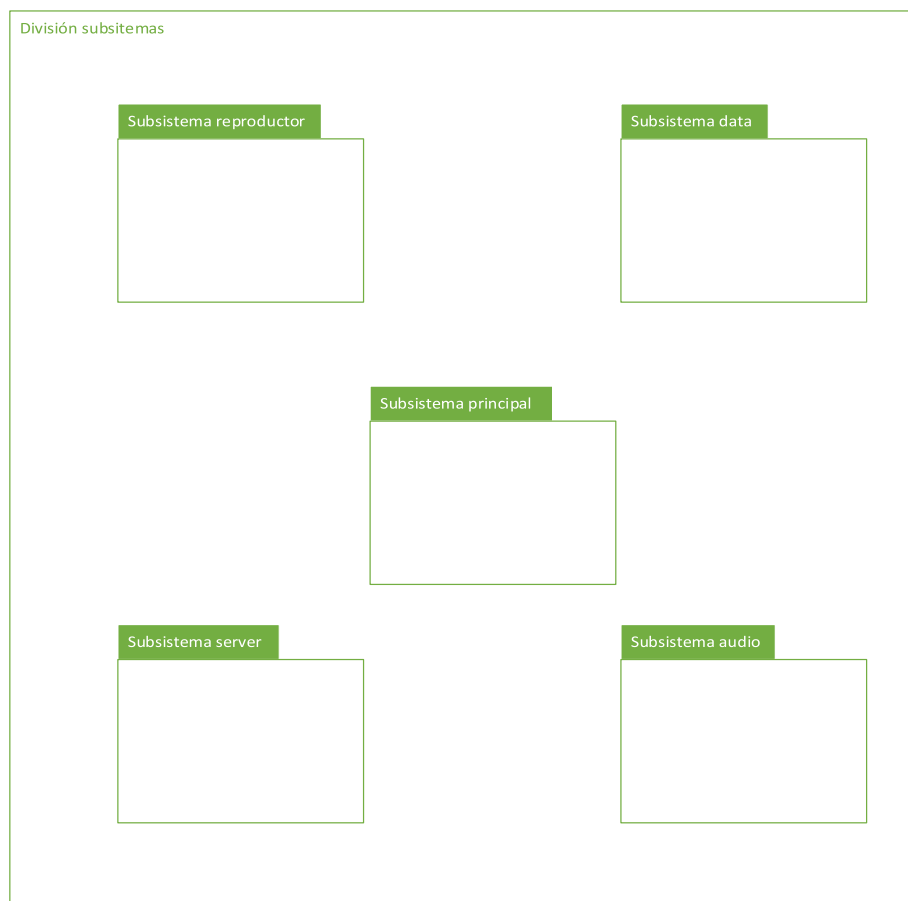


FIGURA 8: DIVISIÓN EN SUBSISTEMAS

3.2 Especificación de subsistemas

3.2.1 Subsistema principal

3.2.1.1 Diagrama de casos de uso

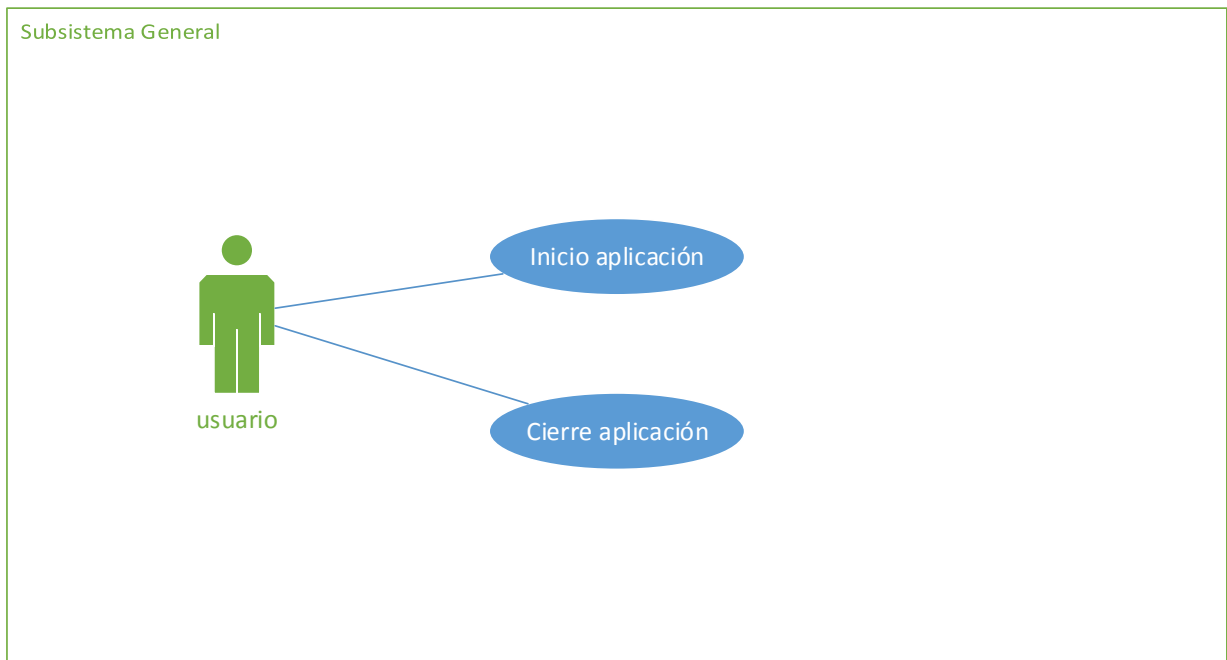


FIGURA 9: DISEÑO CASO DE USO “PRINCIPAL”

3.2.1.2 Detalles de los casos de uso

ESCENARIO: Iniciar aplicación
Numeración: 1.1
Precondiciones: -
Postcondiciones: La aplicación habrá cargado todos los elementos necesarios para su correcta ejecución.
Quién lo comienza: El usuario
Quién lo finaliza: El sistema operativo
Excepciones: -
Descripción: La aplicación carga todos los elementos necesarios y muestra su pantalla principal con una lista de los éxitos recibidos.

TABLA 18: ESCENARIO “INICIAR APLICACIÓN”

ESCENARIO: Cerrar aplicación
Numeración: 1.2
Precondiciones: -
Postcondiciones: La aplicación mata todos los procesos creados por la aplicación, libera variables y espacio de memoria.
Quién lo comienza: El usuario
Quién lo finaliza: El sistema operativo
Excepciones: -
Descripción: La aplicación se cierra liberando al sistema.

TABLA 19: ESCENARIO “CERRAR APLICACIÓN”

3.2.2 Subsistema server

3.2.2.1 Diagrama de casos de uso

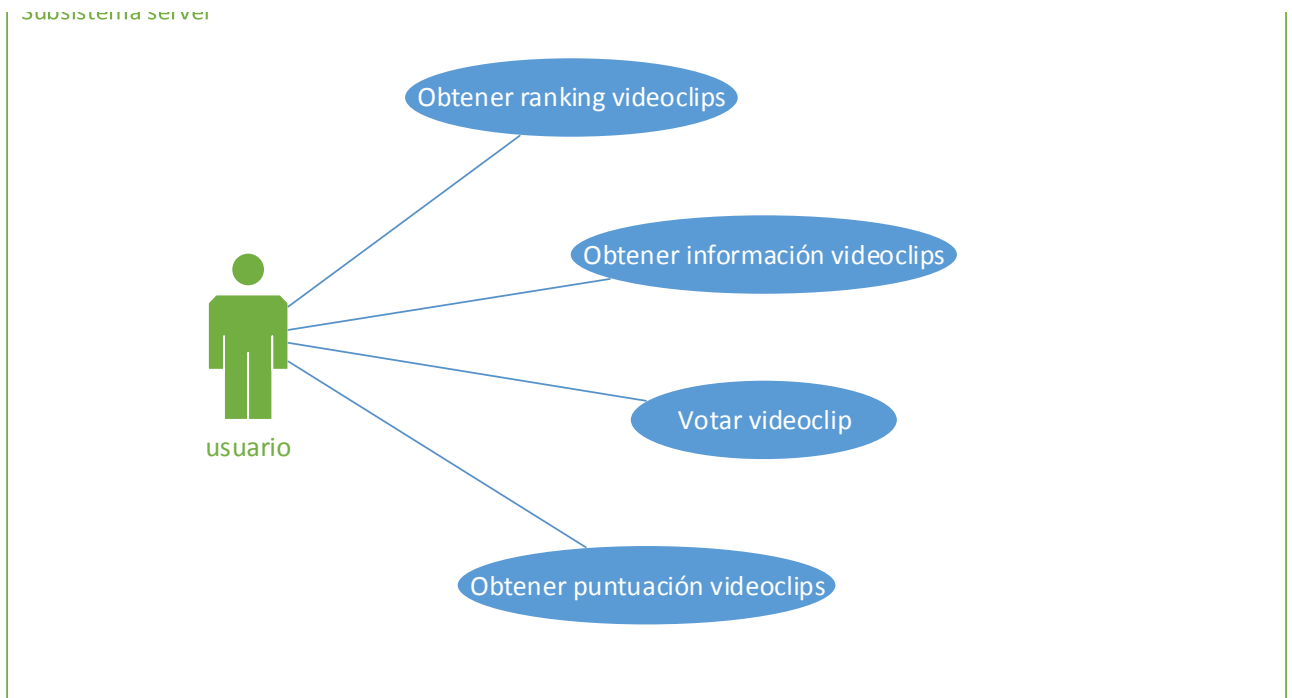


FIGURA 10: DISEÑO DE CASO DE USO: “SERVER”

3.2.2.2 Detalles de los casos de uso

ESCENARIO: Obtener ranking de videoclips
Numeración: 2.1
Precondiciones: -
Postcondiciones: Se obtiene listado de los videoclips
Quién lo comienza: La aplicación
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se parsea el XML para obtener el listado de videoclips

TABLA 20: ESCENARIO “OBTENER RANKING VIDEOCLIPS”

ESCENARIO: Obtener información videoclip
Numeración: 2.2
Precondiciones: -
Postcondiciones: Se obtiene información de los videoclips
Quién lo comienza: La aplicación
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se parsea el XML para obtener la información de cada videoclip

TABLA 21: ESCENARIO “OBTENER INFORMACIÓN VIDEOCLIPS”

ESCENARIO: Votar videoclip
Numeración: 2.3
Precondiciones: Videoclip no votado previamente
Postcondiciones: Se envía la puntuación asignada por el usuario.
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: El usuario califica el videoclip con una puntuación y esta será enviada a un servidor.

TABLA 22: ESCENARIO “VOTAR VIDEOCLIPS”

ESCENARIO: Obtener puntuación videoclip
Numeración: 2.4
Precondiciones: -
Postcondiciones: Se muestra promedio de las puntuaciones asignadas.
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se recibe el valor promedio de las puntuaciones recibidas y se muestra para cada videoclip en el listado de videos.

TABLA 23: ESCENARIO “OBTENER PUNTUACIÓN VIDEOCLIPS”

3.2.3 Subsistema player

3.2.3.1 Diagrama de casos de uso

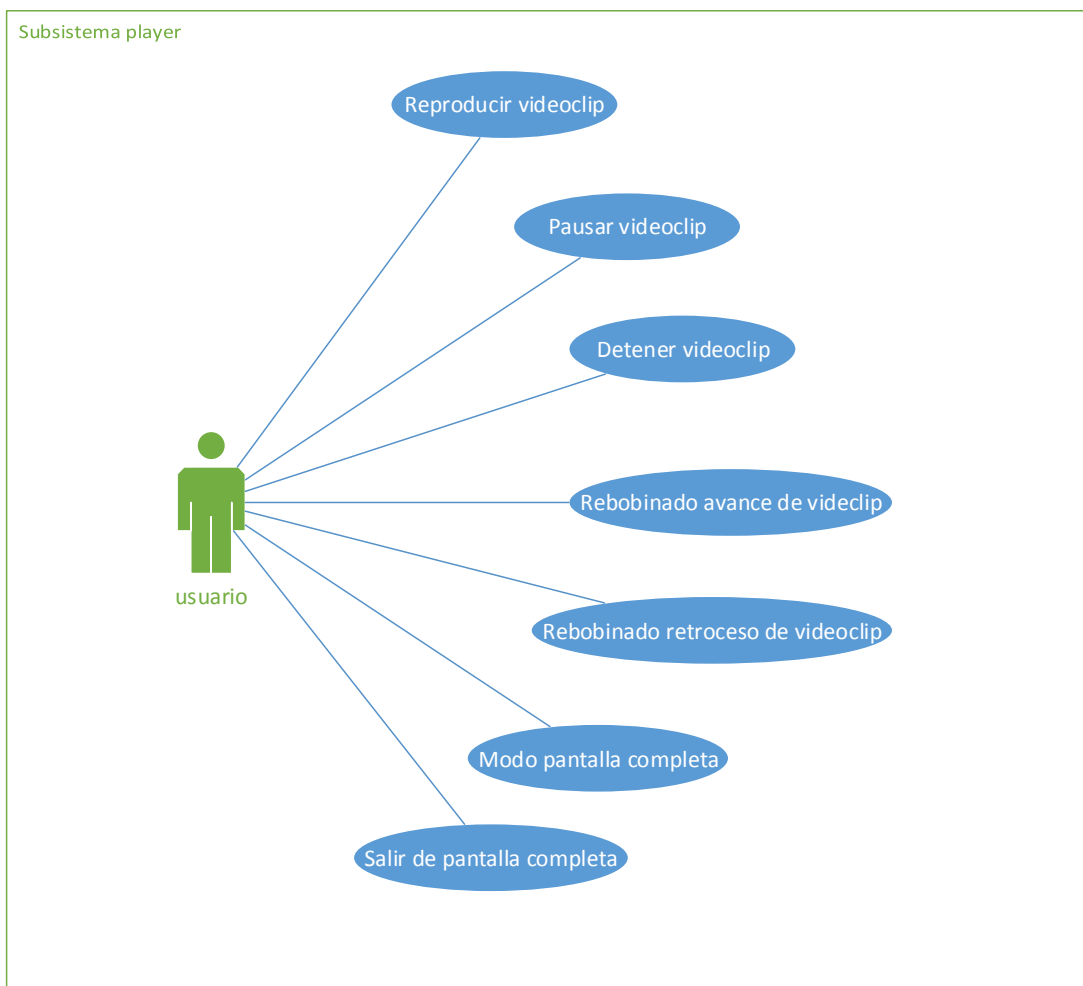


FIGURA 11. DISEÑO DE CASO DE USO: “PLAYER”

3.2.3.2 Detalles de los casos de uso

ESCENARIO: Reproducir videoclip
Numeración: 3.1
Precondiciones: -
Postcondiciones: Se reproduce un videoclip de la lista
Quién lo comienza: El usuario
Quién lo finaliza: El usuario o la aplicación
Excepciones: -
Descripción: Se reproduce un videoclip de la lista

TABLA 24: ESCENARIO “REPRODUCIR VIDEOCLIPS”

ESCENARIO: Pausar videoclip
Numeración: 3.2
Precondiciones: -
Postcondiciones: Se pausa la reproducción del videoclip
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se pausa la reproducción del videoclip

TABLA 25: ESCENARIO “PAUSAR VIDEOCLIPS”

ESCENARIO: Detener videoclip
Numeración: 3.3
Precondiciones: -
Postcondiciones: Se detiene la reproducción del videoclip
Quién lo comienza: El usuario o la aplicación
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se detiene la reproducción del videoclip

TABLA 26: ESCENARIO “DETENER VIDEOCLIPS”

ESCENARIO: Rebobinado de avance de videoclip
Numeración: 3.4
Precondiciones: -
Postcondiciones: Se rebobina el video hacia adelante
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se avanza el video de 5 segundos en 5 segundos.

TABLA 27: ESCENARIO “REBOBINADO DE AVANCE VIDEOCLIPS”

ESCENARIO: Rebobinado de retroceso de videoclip
Numeración: 3.5
Precondiciones: -
Postcondiciones: Se rebobina el video hacia atrás
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se retrocede el video de 5 segundos en 5 segundos.

TABLA 28: ESCENARIO “REBOBINADO DE RETROCESO VIDEOCLIPS”

ESCENARIO: Modo pantalla completa
Numeración: 3.6
Precondiciones: No encontrarse en modo de pantalla completa
Postcondiciones: Se presenta la reproducción en pantalla completa
Quién lo comienza: El usuario
Quién lo finaliza: El usuario
Excepciones: -
Descripción: Se presenta la reproducción en pantalla completa

TABLA 29: ESCENARIO “MODO PANTALLA COMPLETA”

ESCENARIO: Salir de pantalla completa
Numeración: 3.7
Precondiciones: Modo de pantalla completa
Postcondiciones: Se retorna a la visualización normal de la aplicación
Quién lo comienza: El usuario
Quién lo finaliza: El usuario
Excepciones: -
Descripción: Se retorna a la visualización normal de la aplicación desde el modo de pantalla completa

TABLA 30: ESCENARIO “SALIR DE PANTALLA COMPLETA”

3.2.4 Subsistema audio

3.2.4.1 Diagrama de casos de uso

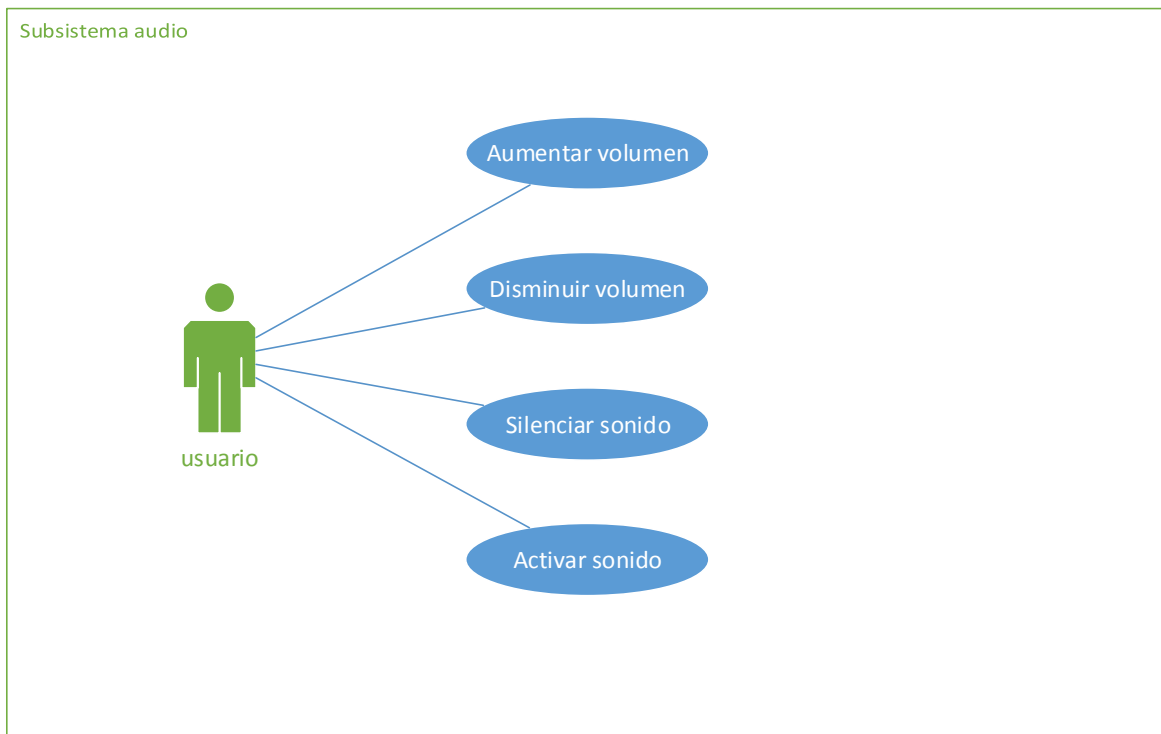


FIGURA 12: DISEÑO CASO DE USO “AUDIO”

3.2.4.2 Detalles de los casos de uso

ESCENARIO: Aumentar volumen
Numeración: 4.1
Precondiciones: -
Postcondiciones: Se incrementa el nivel de sonido en una unidad
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se incrementa el nivel de sonido en una unidad

TABLA 31: ESCENARIO “AUMENTAR VOLUMEN”

ESCENARIO: Disminuir volumen
Numeración: 4.2
Precondiciones: -
Postcondiciones: Se decrementa el nivel de sonido en una unidad
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se decrementa el nivel de sonido en una unidad

TABLA 32: ESCENARIO “DISMINUIR VOLUMEN”

ESCENARIO: Silenciar sonido
Numeración: 4.3
Precondiciones: Aplicación no silenciada
Postcondiciones: Se silencia el sonido de la aplicación
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se silencia el sonido de la aplicación

TABLA 33: ESCENARIO “SILENCIAR SONIDO”

ESCENARIO: Activar sonido
Numeración: 4.4
Precondiciones: Sonido no activo
Postcondiciones: Se reactiva el sonido de la aplicación
Quién lo comienza: El usuario
Quién lo finaliza: La aplicación
Excepciones: -
Descripción: Se reactiva el sonido de la aplicación después de haber sido silenciada.

TABLA 34: ESCENARIO “ACTIVAR SONIDO”

3.3 Modelo de datos del sistema

A continuación se presenta el modelo de datos del sistema, que en el caso de este proyecto, sólo consta de un fichero, al no disponer de base de datos.

3.3.1 Modelo gráfico

El modelo gráfico del sistema se ha dividido en dos secciones: la que corresponde a la aplicación y la relacionada con el servidor externo en el cual residirán la información acerca de las votaciones de los usuarios.

3.3.1.1 Aplicación

El sistema exclusivamente importa un fichero XML con toda la información necesaria de la aplicación que resultará almacenada en la clase Data.js.

El lenguaje XML se trata de un lenguaje fuertemente tipado, lo cual exige que todas sus etiquetas se encuentren presentes. La estructura del fichero *videoList.xml* es la siguiente:

```
<item>
  <title> </title>
  <link> </link>
  <description> </description>
  <type> </type>
  <cover> </cover>
  <logo> </logo>
  <score> </score>
</item>
```

3.3.1.2 Sistema de votación de videoclips

Para el sistema de votación se necesita incorporar un servidor con una base de datos que responda al siguiente modelo:

La estructura de la base de datos, se compone de una clave primaria compuesta, ya que de este modo se evita registrar sucesivos votos de un mismo usuario para el mismo videoclip.

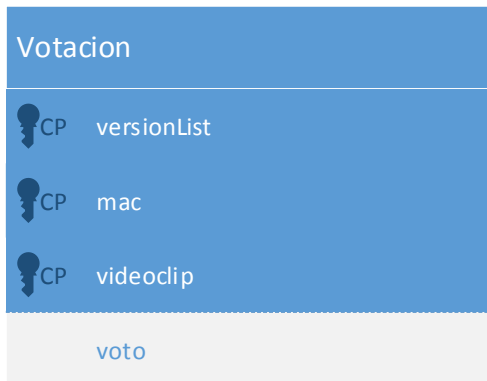


FIGURA 13. SISTEMA DE VOTACIÓN: MODELO GRÁFICO

3.3.2 Catálogo de etiquetas

Nombre	Descripción	Etiqueta
Title	Contiene el título del videoclip.	<title> </title>
Link	Contiene la url en la que se encuentra disponible el servidor desde donde se extraerá el flujo de vídeo.	<link> </link>
Description	Almacena la información del vídeo que se quiera aportar.	<description> </description>
Type	Indica el tipo de video: en directo o bajo demanda.	<type> </type>
Cover	Contiene una url con una imagen que actuará de presentación del videoclip.	<cover> </cover>
Logo	Contiene una url con un logo del videoclip.	<logo> </logo>
Score	Contiene un promedio de las puntuaciones del videoclip emitidas por los usuarios.	<score> </score>

TABLA 35: CATÁLOGO DE ETIQUETAS

3.3.3 Descripción de atributos

Nombre	Descripción	Tipo	Tamaño
versionList	Número de versión de la lista de éxitos	Decimal	4

Nombre	Descripción	Tipo	Tamaño
mac	Dirección MAC del televisor de Samsung	Varchar	12
videoclip	Id del videoclip votado	Varchar	40
voto	Puntuación asignada por el usuario	int	2

TABLA 36: DESCRIPCIÓN DE ATRIBUTOS

3.4 Modelo de eventos del sistema

ID	EVENTO	CASO DE USO
E1	El usuario inicia la aplicación.	Iniciar aplicación
E2	El usuario cierra la aplicación.	Cerrar aplicación
E3	El usuario selecciona un videoclip.	Obtener ranking videoclip
E4	El usuario visualiza la información del videoclip.	Obtener información videoclip
E5	El usuario vota el videoclip.	Votar videoclip
E6	El usuario visualiza el promedio de la puntuación del videoclip emitida por los usuarios.	Obtener puntuación de videoclip.
E7	El usuario ordena la reproducción del video.	Reproducir videoclip
E8	El usuario ordena el pausado del video.	Pausar videoclip
E9	El usuario ordena la detención del video.	Detener videoclip
E10	El usuario ordena el rebobinado hacia delante del video.	Rebobinado de avance de videoclip
E11	El usuario ordena el rebobinado hacia atrás del video.	Rebobinado hacia atrás de videoclip
E12	El usuario sube el volumen	Aumentar volumen
E13	El usuario baja el volumen	Disminuir volumen
E14	El usuario silencia la aplicación	Silenciar sonido
E15	El usuario activa el sonido de la aplicación	Activar sonido
E16	El usuario ordena conmutar a pantalla completa	Modo pantalla completa
E17	El usuario ordena la salida de la pantalla completa.	Salir de pantalla completa

TABLA 37: MODELO DE EVENTOS DEL SISTEMA

3.4.1 Diagramas de interacción

3.4.1.1 Iniciar aplicación

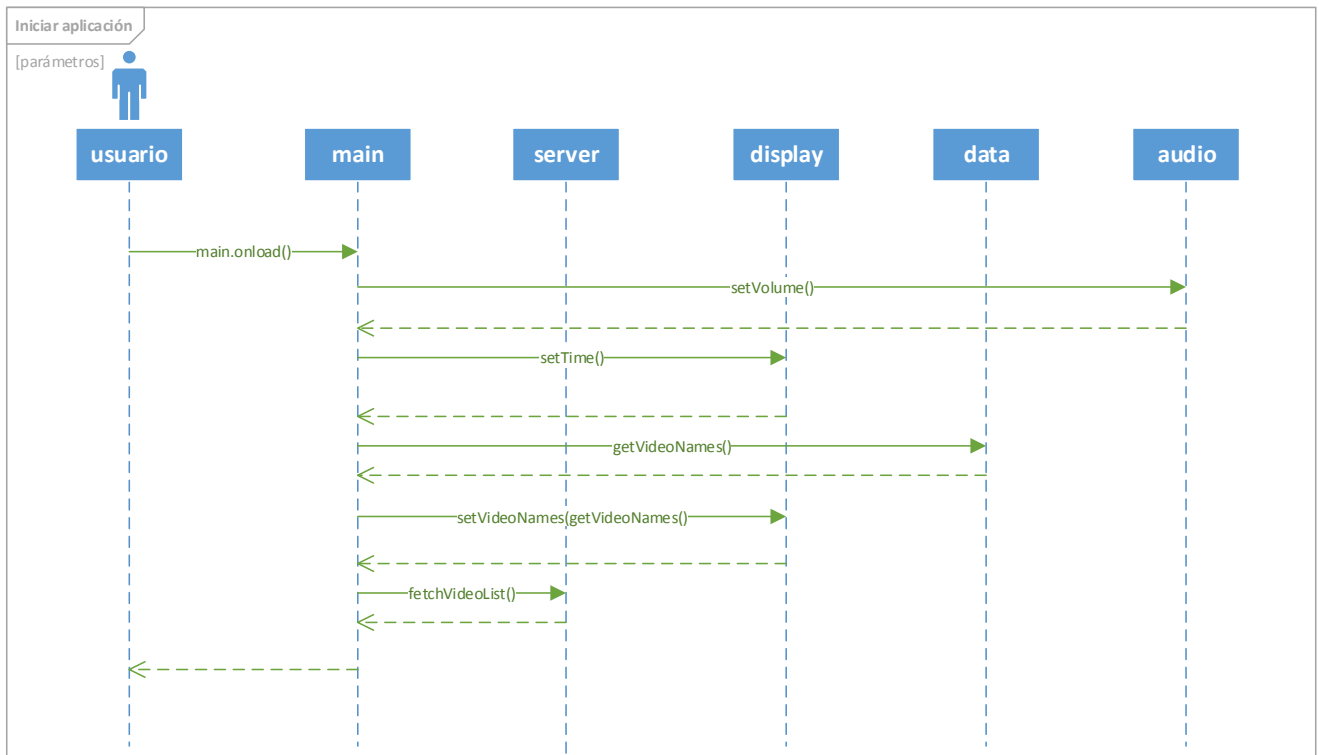


FIGURA 14. DIAGRAMA DE INTERACCIÓN: “INICIAR APLICACIÓN”

3.4.1.2 Cerrar aplicación

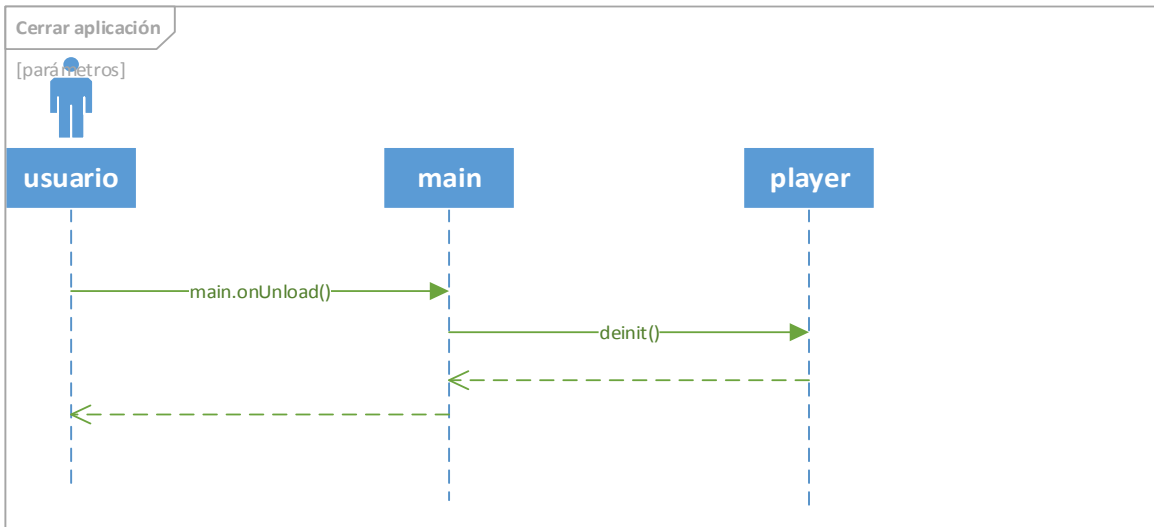


FIGURA 15: DIAGRAMA DE INTERACCIÓN “CERRAR APLICACIÓN”

3.4.1.3 Obtener ranking videoclips

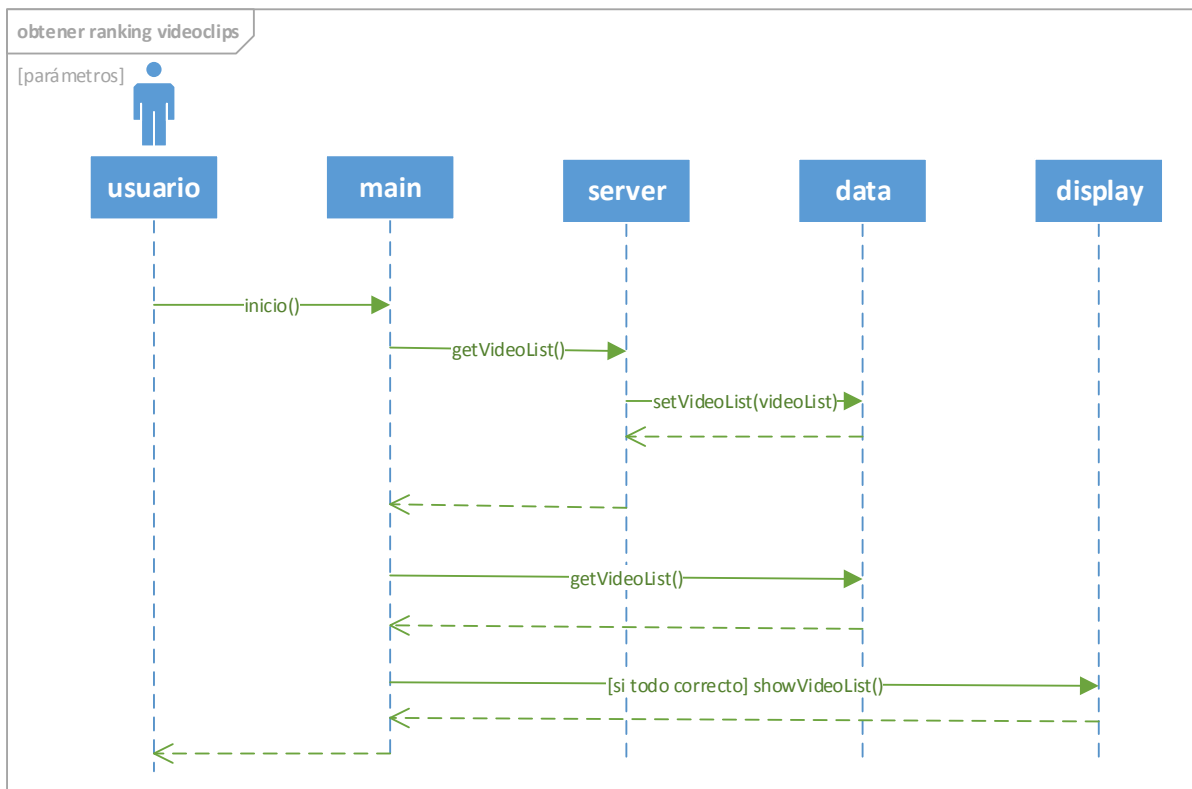


FIGURA 16: DIAGRAMA DE INTERACCIÓN “OBTENER RANKING VIDEOLCIPS”

3.4.1.4 Obtener información videoclips

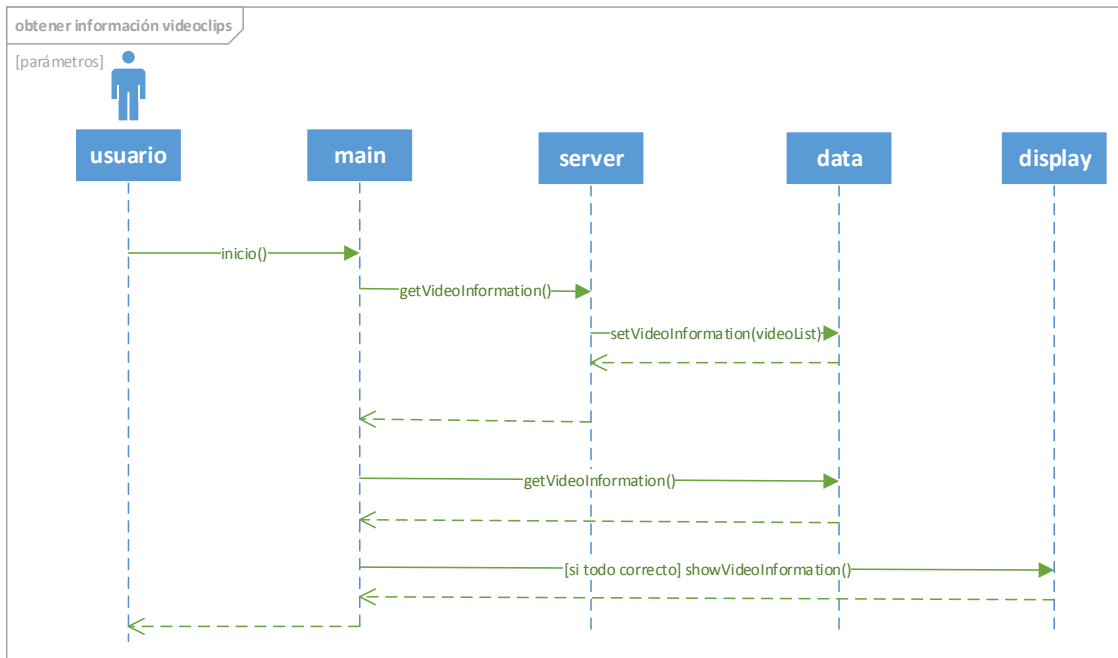


FIGURA 17: DIAGRAMA DE INTERACCIÓN “OBTENER INFORMACIÓN VIDEOCLIPS”

3.4.1.5 Votar videoclips

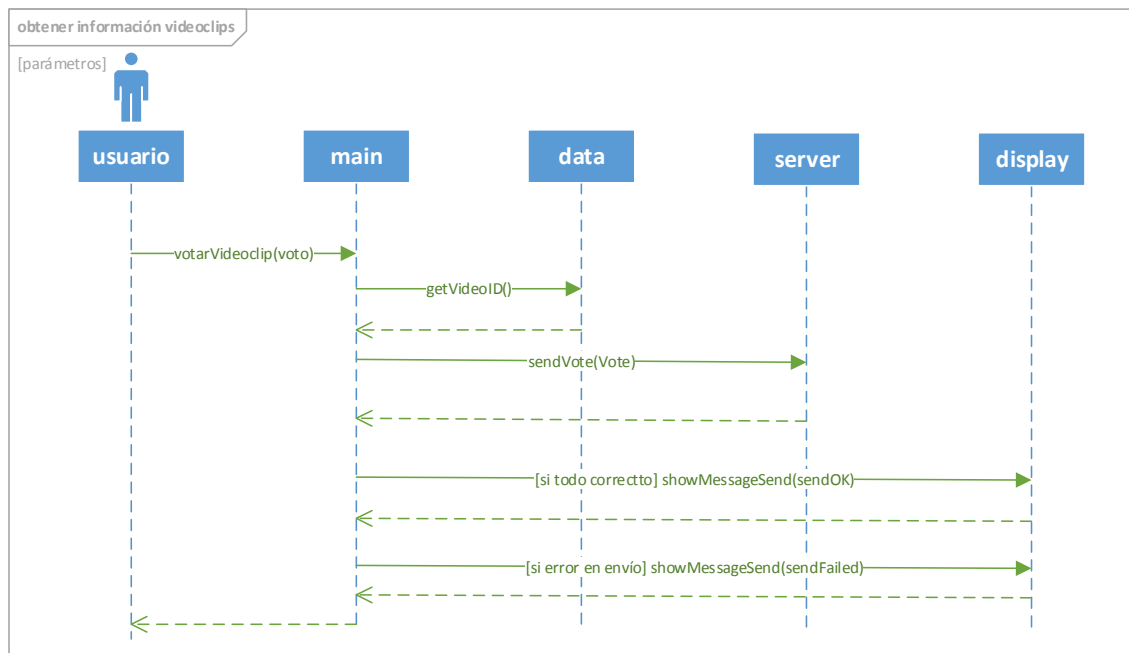


FIGURA 18: DIAGRAMA DE INTERACCIÓN “VOTAR VIDEOCLIPS”

3.4.1.6 Obtener puntuación de videoclip

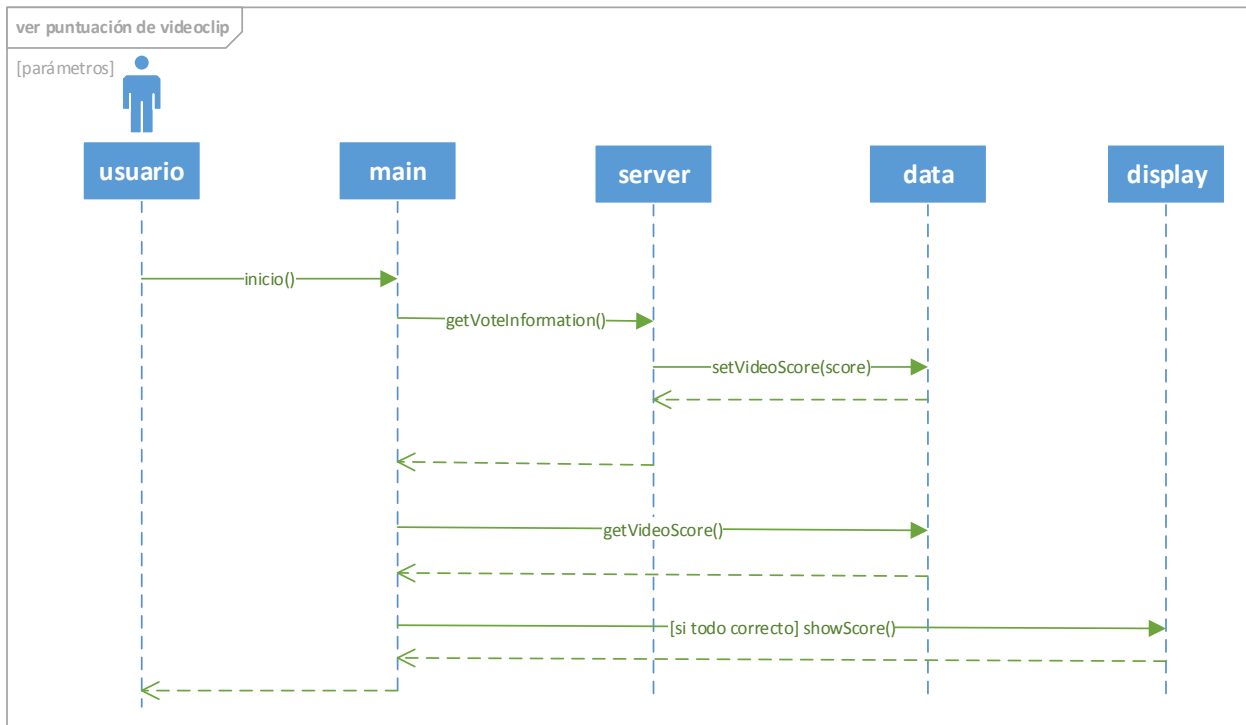


FIGURA 19: DIAGRAMA DE INTERACCIÓN “OBTENER PUNTUACIÓN DE VIDEOCLIP”

3.4.1.7 Reproducir videoclip

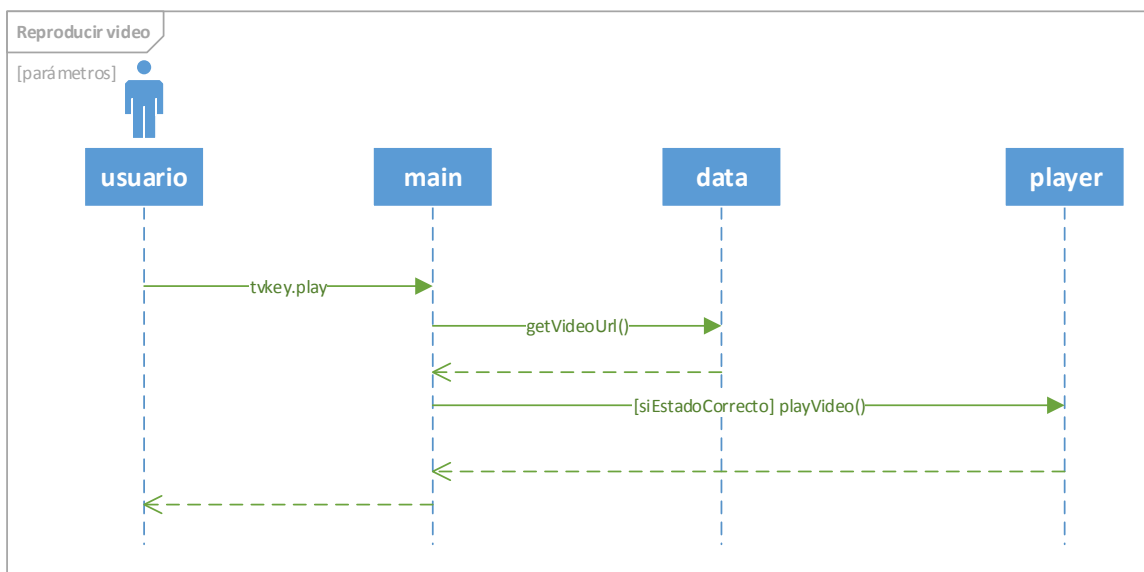


FIGURA 20: DIAGRAMA DE INTERACCIÓN “REPRODUCIR VIDEOCLIP”

3.4.1.8 Pausar videoclip

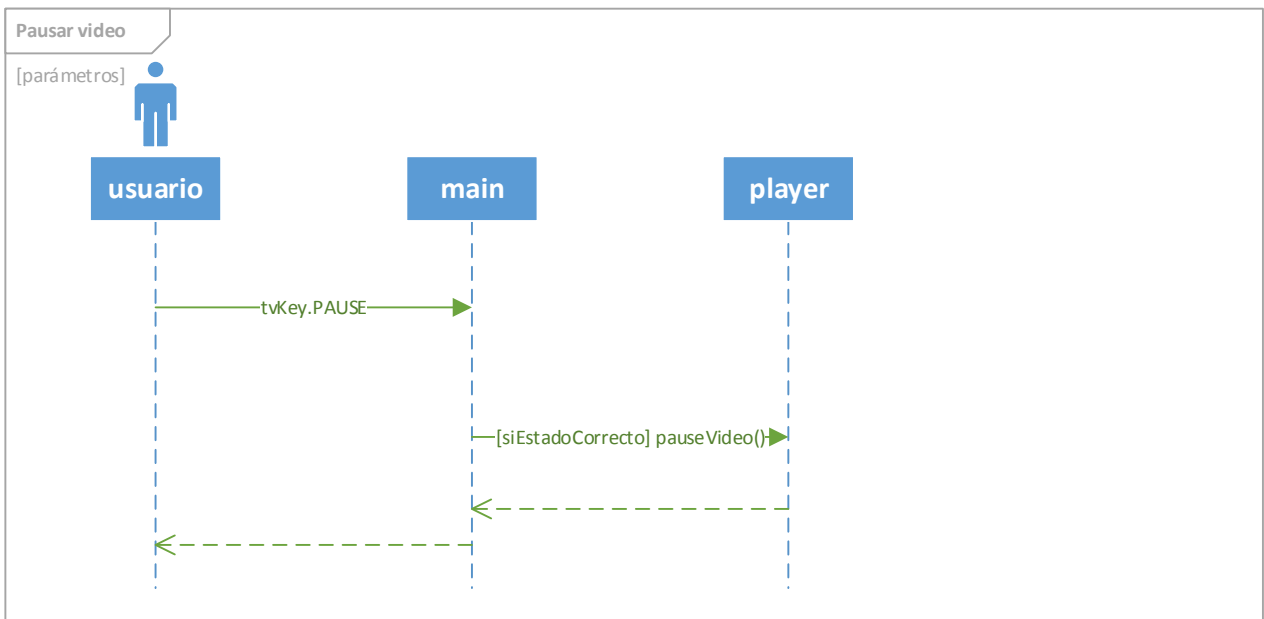


FIGURA 21: DIAGRAMA DE INTERACCIÓN "PAUSAR VIDEOCLIP"

3.4.1.9 Detener videoclip

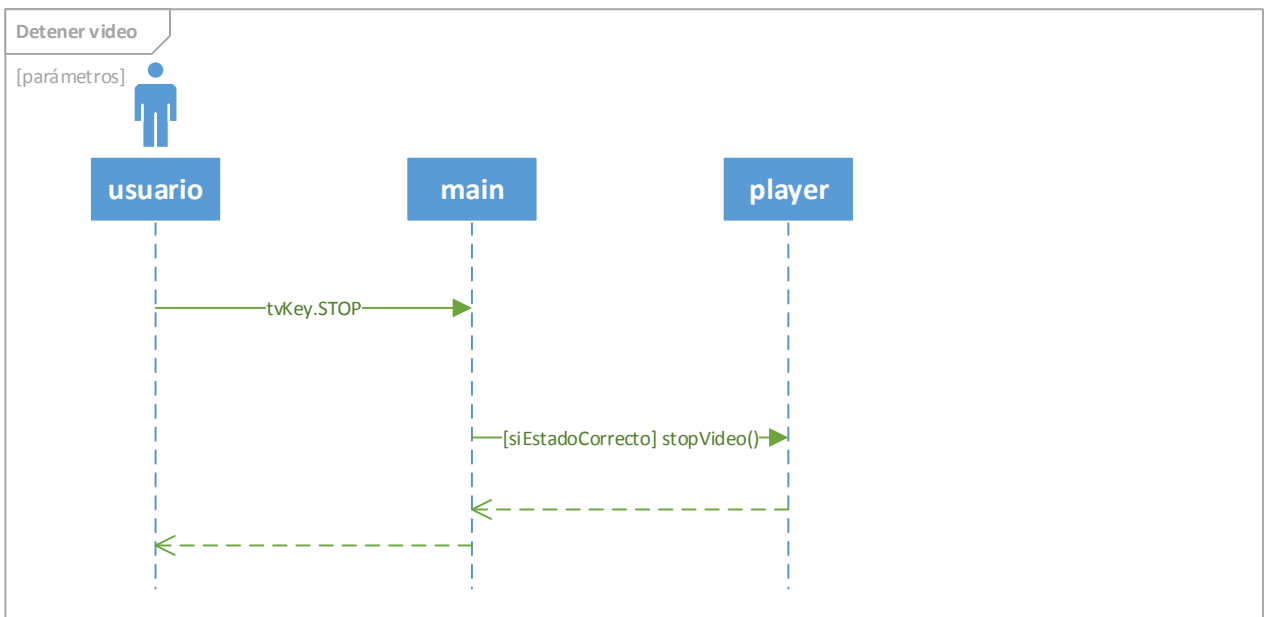


FIGURA 22: DIAGRAMA DE INTERACCIÓN "DETENER VIDEOCLIP"

3.4.1.10 Rebobinado de avance de videoclip

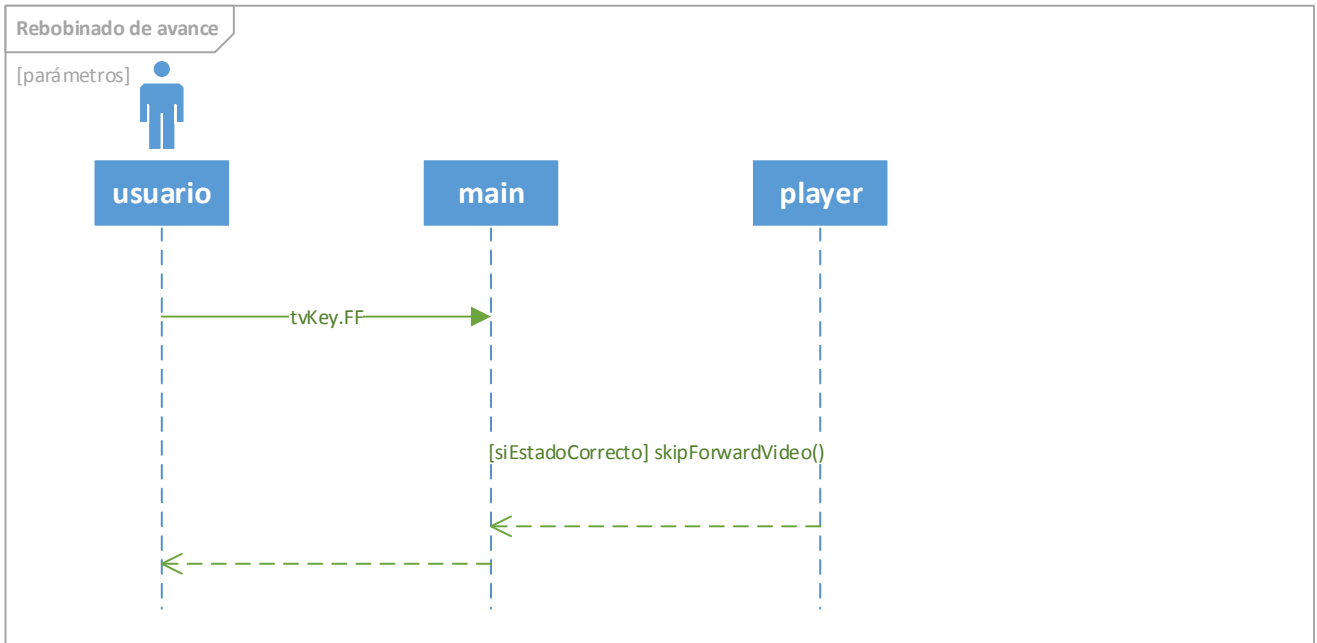


FIGURA 23: DIAGRAMA DE INTERACCIÓN “REBOBINADO DE AVANCE”

3.4.1.11 Rebobinado de retroceso de videoclip

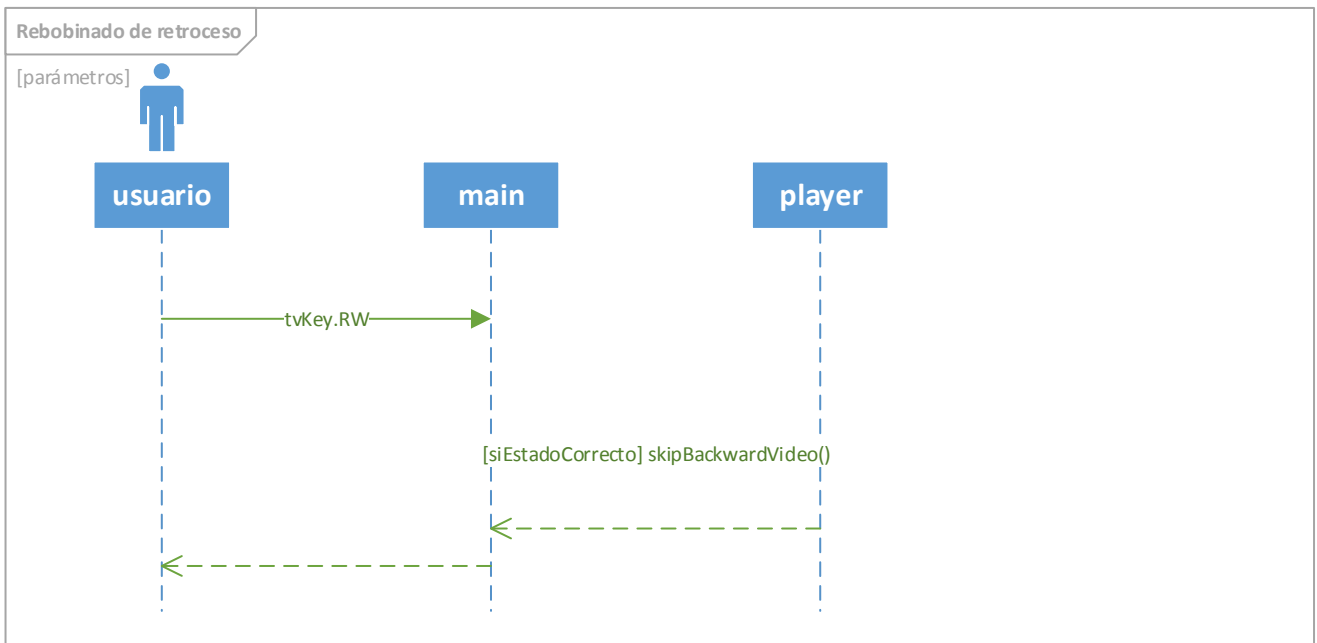


FIGURA 24: DIAGRAMA DE INTERACCIÓN “REBOBINADO DE RETROCESO”

3.4.1.12 Aumentar volumen

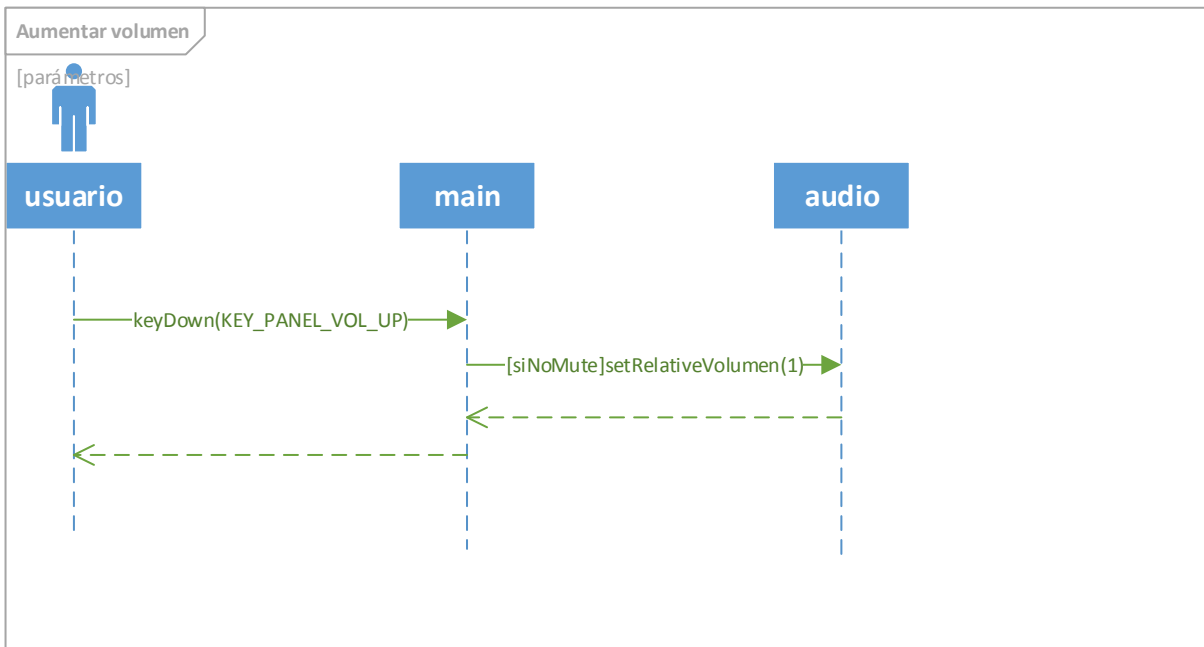


FIGURA 25: DIAGRAMA DE INTERACCIÓN “AUMENTAR VOLUMEN”

3.4.1.13 Disminuir volumen

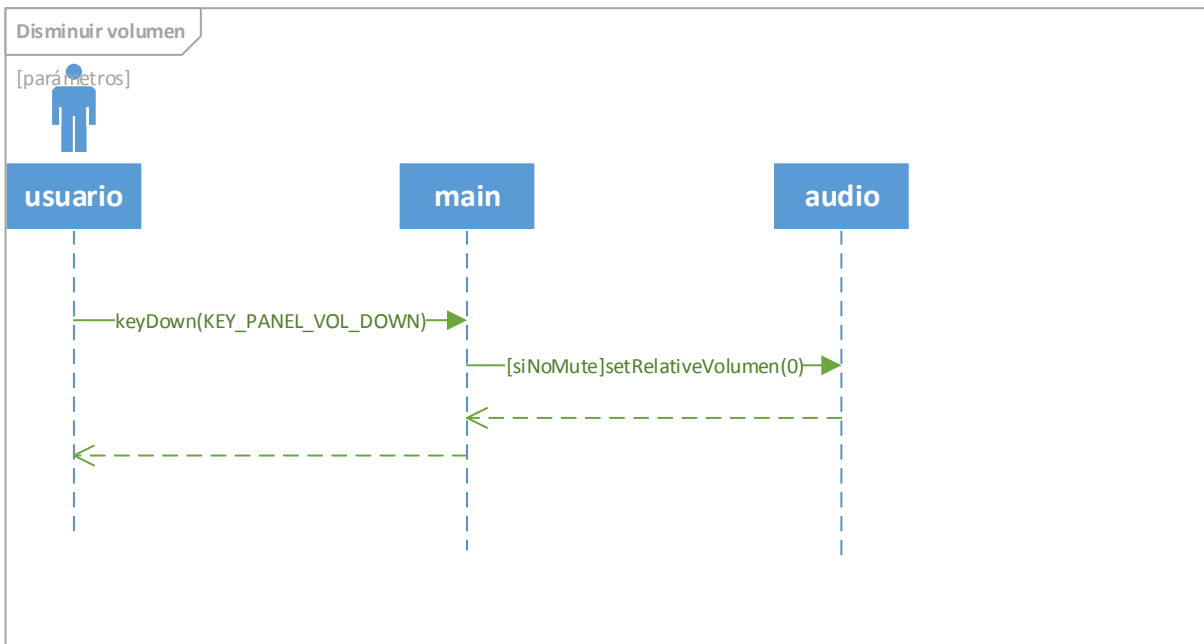


FIGURA 26: DIAGRAMA DE INTERACCIÓN “DISMINUIR VOLUMEN”

3.4.1.14 Silenciar sonido

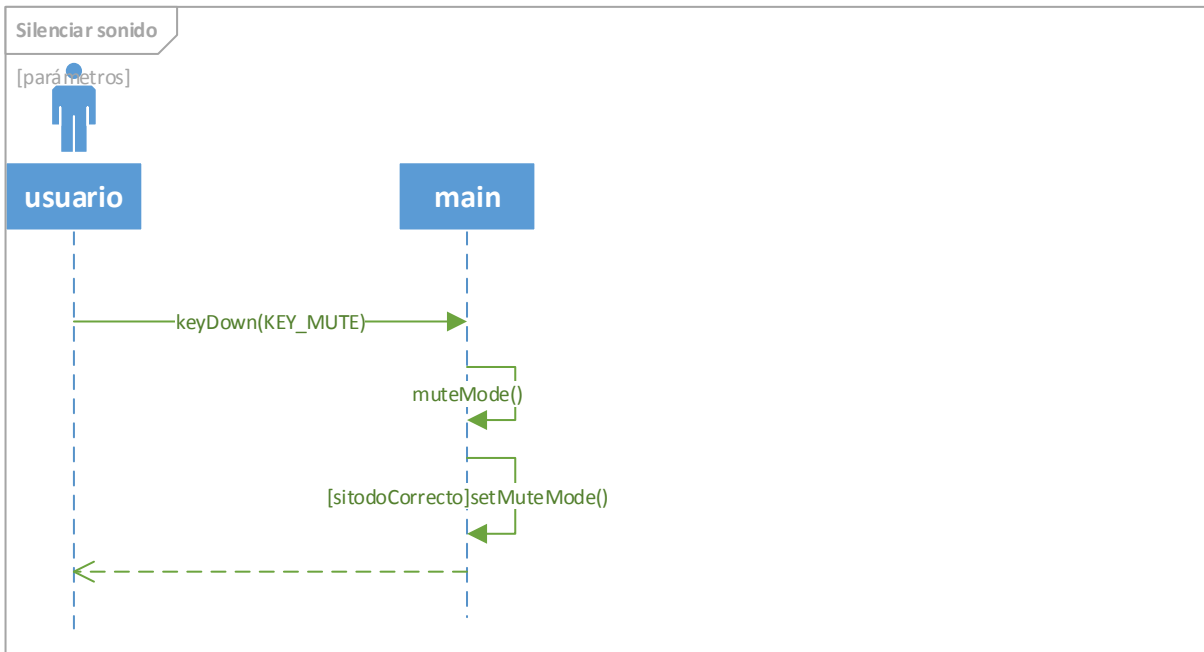


FIGURA 27: DIAGRAMA DE INTERACCIÓN “SILENCIAR SONIDO”

3.4.1.15 Activar sonido

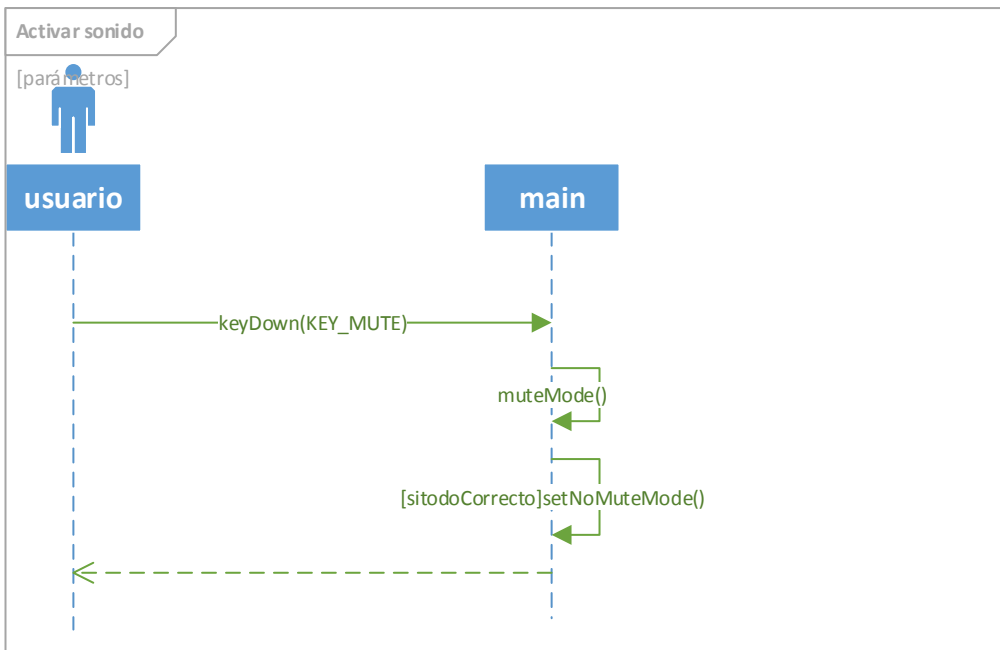


FIGURA 28: DIAGRAMA DE INTERACCIÓN “ACTIVAR SONIDO”

3.4.1.16 Modo de pantalla completa

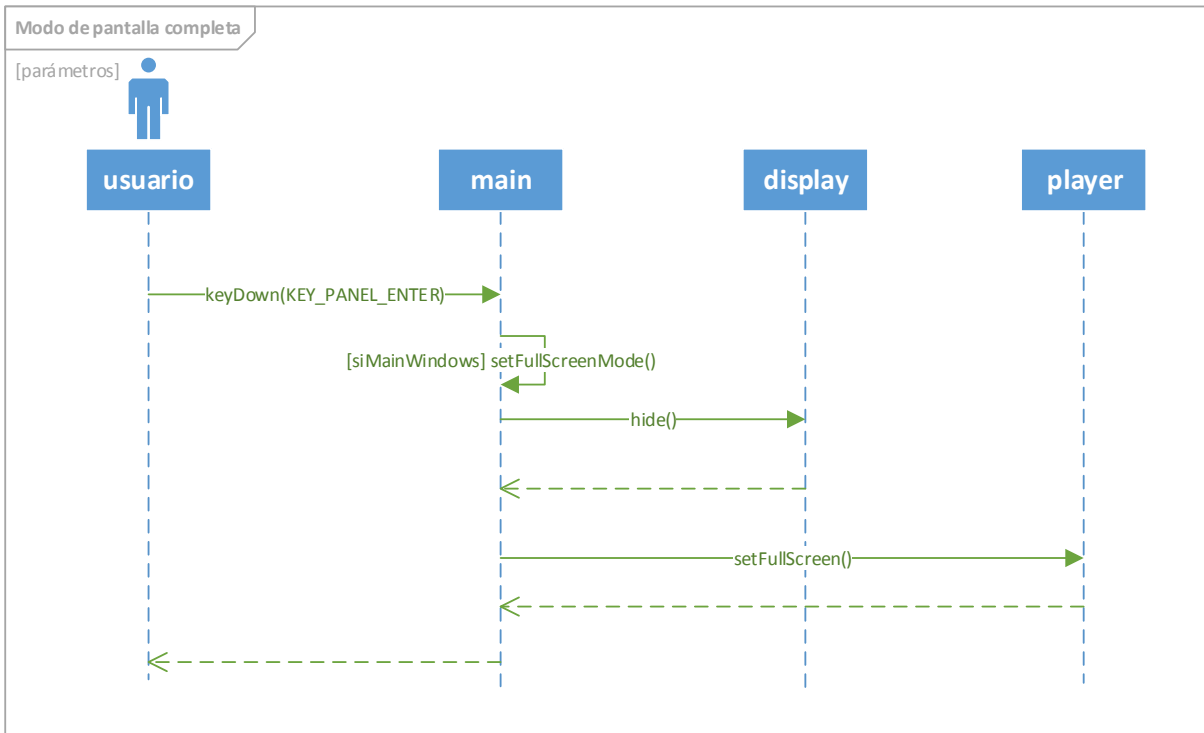


FIGURA 29: DIAGRAMA DE INTERACCIÓN “MODO DE PANTALLA COMPLETA”

3.4.1.17 Salir de pantalla completa

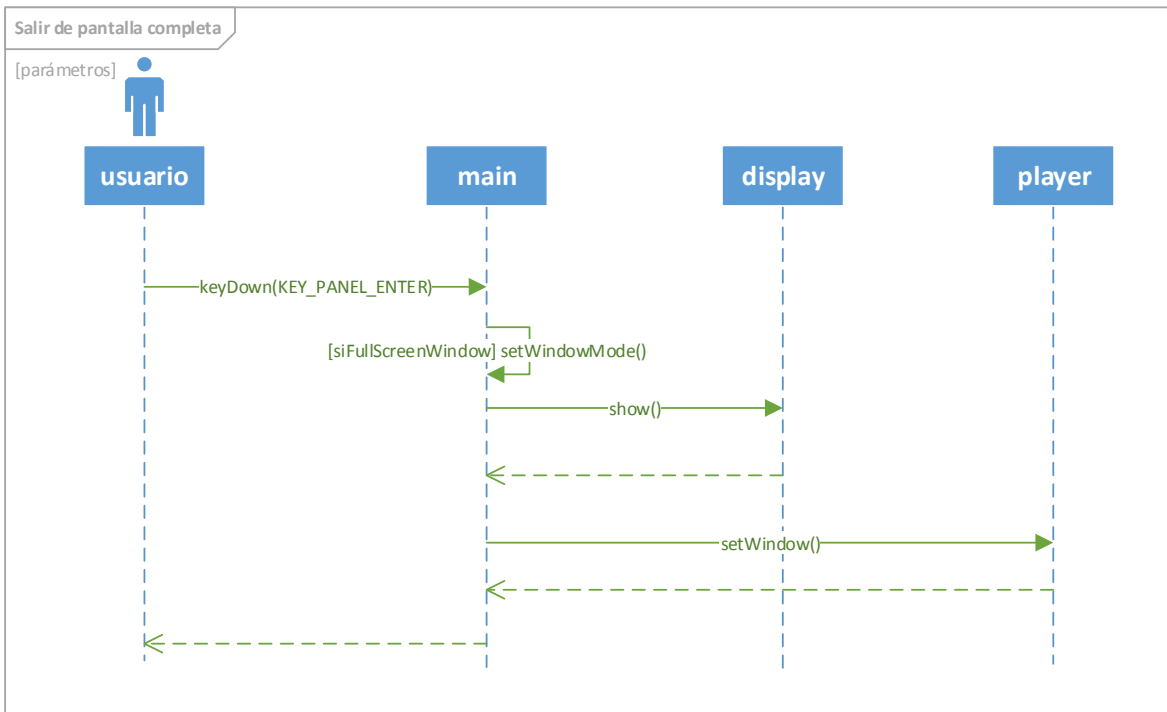


FIGURA 30: DIAGRAMA DE INTERACCIÓN “SALIR DE PANTALLA COMPLETA”

3.5 Interfaces de usuario (pantallas)

3.5.1 Interfaces de usuario

1. Cada pantalla estará contenida dentro del **main**, que dispondrá de varios marcos: *logo, lefthalf, righthalf* y *navi*.
 - a. **Logo**. Para el logo de la aplicación.
 - b. **LeftHalf**. Para el menú de videoclips.
 - i. Videolist. Contiene el listado de videoclips.
 - ii. Previous. Contiene el botón para seleccionar videoclip anterior.
 - iii. Next. Contiene el botón para seleccionar videoclip posterior.
 - c. **RightHalf**. Para la interfaz de reproducción.
 - i. ProgressBar. Dispone la barra de progreso.
 - ii. TimeInfo. Dispone el tiempo de reproducción en formato numérico.
 - iii. Cover. Contenedor que solapa al reproductor de vídeo. Muestra la portada del videoclip seleccionado.
 - d. **Navi**. Para la información de navegación.
 - i. HelpNavi. Contiene las imágenes de los botones de control del mando.

3.5.2 Relación entre pantallas y escenarios

Nodo	Descripción
Principal	<p>Escenario 1.1: “Iniciar aplicación”</p> <p>Escenario 1.2: “Cerrar aplicación”</p> <p>Escenario 2.1: “Obtener ranking videoclip”</p> <p>Escenario 2.2: “Obtener información videoclip”</p> <p>Escenario 2.3: “Votar videoclip”</p> <p>Escenario 2.4: “Ver puntuación videoclip”</p>

Nodo	Descripción
	<p>Escenario 3.1: “Reproducir videoclip”</p> <p>Escenario 3.2: “Pausar videoclip”</p> <p>Escenario 3.3: “Detener videoclip”</p> <p>Escenario 3.4: “Rebobinado de avance rápido”</p> <p>Escenario 3.5: “Rebobinado de retroceso rápido”</p> <p>Escenario 3.6: “Modo pantalla completa”</p> <p>Escenario 3.7: “Salir de pantalla completa”</p> <p>Escenario 4.1: “Aumentar volumen”</p> <p>Escenario 4.2: “Disminuir volumen”</p> <p>Escenario 4.3: “Silenciar sonido”</p> <p>Escenario 4.4: “Activar sonido”</p>

TABLA 38: RELACIÓN ENTRE PANTALLAS Y ESCENARIOS

3.5.3 Pantallas

3.5.3.1 Arquetipo de pantalla principal

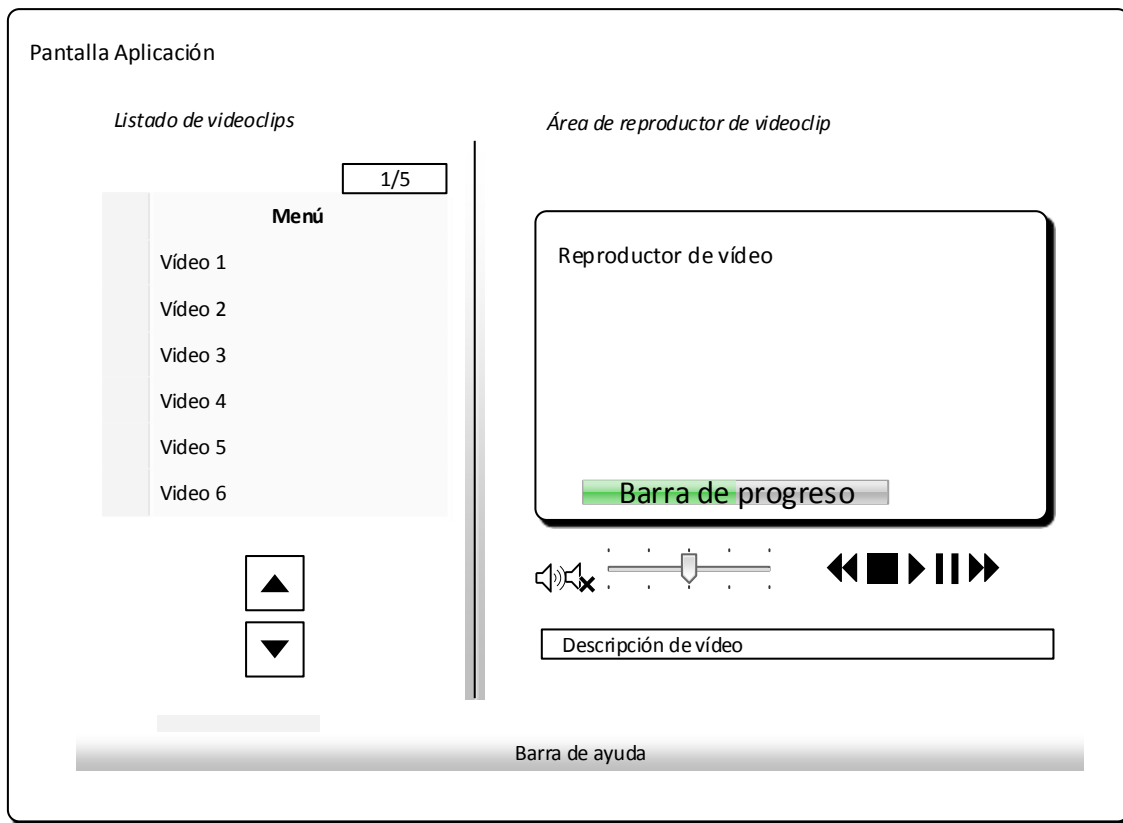


FIGURA 31: ARQUETIPO DE PANTALLA DE APLICACIÓN

3.5.4 Relación entre pantallas y eventos

Identificador	EV1	EV2	EV3	EV4	EV5	EV6	EV7	EV8	EV9	EV10	EV11	EV12	EV13	EV14	EV15	EV16	EV17	EV18
Principal	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

TABLA 39: RELACIÓN ENTRE PANTALLAS Y EVENTOS

4 DISEÑO DEL SISTEMA DE INFORMACIÓN

4.1 Diseño de la arquitectura del sistema

4.1.1 Entorno tecnológico de implantación

4.1.1.1 Equipo físico

El equipo se compone de un servidor con un módulo de PHP y una base de datos MySQL incorporada.

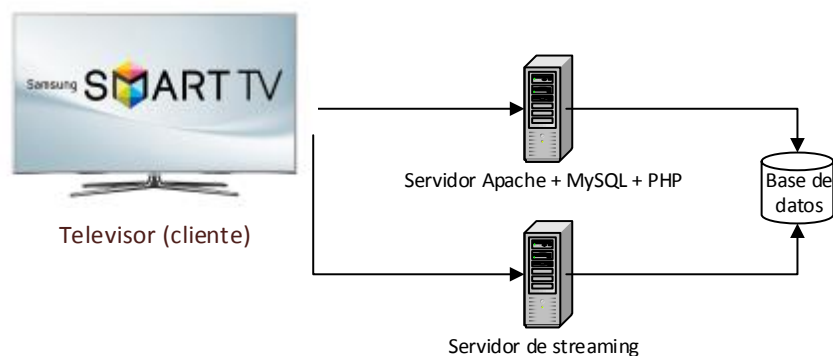


FIGURA 32: DIAGRAMA ENTORNO DE IMPLANTACIÓN

Especificación física de los nodos:

Nodo	Descripción	Características hardware
Servidores Apache + Servidor MySQL + Servidor PHP	Servidor Base de datos, Servidor web y Servidor de PHP	-
Servidor Streaming	Servidor para emisión de flujo de vídeo.	-

Nodo	Descripción	Características hardware
Televisor (cliente)	Este nodo representa al equipo en el que reside la aplicación, y realiza la conexión con los servidores.	-

TABLA 40: ESPECIFICACIÓN FÍSICA DE LOS NODOS

4.1.1.2 Equipo lógico

Nodo	Descripción	Características hardware
Servidores Apache + Servidor MySQL + Servidor PHP	Servidor Base de datos, Servidor web y Servidor de PHP	-
Servidor Streaming	Servidor para emisión de flujo de vídeo.	-
Televisor (cliente)	Este nodo representa al equipo en el que reside la aplicación, y realiza la conexión con los servidores.	-

TABLA 41: ESPECIFICACIÓN LÓGICA DE LOS NODOS

4.1.2 Entorno tecnológico de desarrollo

4.1.2.1 Hardware

Nodo	Descripción	Características hardware
Servidor	Servidor BD y de Streaming	Procesador: Intel Core 2 DUO E6550 Processor2x4MB Cache, 2,33GHz, 2327MHz FSB

Nodo	Descripción	Características hardware
		Memoria: 3GB 1200MHz Tarjeta de Video: Intel GMA 3100 Disco Duro: 2 dd c/u 300GB, 10K RPM Serial-Attach SCSI 3Gbps 3,5 in HotPlug HardDrive Red: WAN Miniport
Equipo de trabajo	Este nodo representa al equipo que contiene el entorno de programación en Eclipse.	Procesador: Intel Core 2 DUO E6550 Processor2x4MB Cache, 2,33GHz, 2327MHz FSB Memoria: 3GB 1200MHz Tarjeta de Video: Intel GMA 3100 Disco Duro: 2 dd c/u 300GB, 10K RPM Serial-Attach SCSI 3Gbps 3,5 in HotPlug HardDrive Red: WAN Miniport

TABLA 42: ENTORNO TECNOLÓGICO. HARDWARE

4.1.2.2 Software

El software empleado para el desarrollo de la aplicación Web es el siguiente:

- SO del equipo utilizado para el desarrollo: Microsoft Windows 7.
- Servidor Web: Apache 2.2
- Procesador de textos para la documentación: Microsoft Office Word 2010
- Entorno de desarrollo: Eclipse con SAMSUNG SDK 4.0
- Editor de textos para los archivos fuente: Notepad++
- Software para la gestión de la Base de Datos: phpMyAdmin 2.11.6
- Herramientas CASE: Microsoft Office Visio 2013
- Herramienta de administración de proyectos: Microsoft Office Project 2010

4.1.3 Especificación de estándares y normas de diseño y construcción

En esta apartado se definen los **estándares técnicos** y de nomenclatura, **normas** y **recomendaciones**, que generalmente están relacionados con la adopción o diseño de una arquitectura o infraestructura tecnológica concreta, y que pueden condicionar el diseño o la construcción del sistema de información.

Dentro del apartado **Selección de la solución** se eligió un widget para televisión Samsung que se organiza mediante el patrón vista-controlador.

4.1.3.1 XML e interoperabilidad

Los ficheros XML ofrecen innumerables ventajas:

- **Lenguaje extensible**, lo que posibilita añadir todo tipo de información.
- **Muy legible** para desarrolladores y usuario.
- **Simplifica el transporte** de información.
- Simplifica los **cambios de plataforma**.
- **Facilidad de implantar, programar** y aplicar a diferentes sistemas.
- **Aplicaciones web**.

La principal virtud de incorporar lenguaje XML en la transmisión de datos radica en la posibilidad de poder recibir la información de diferentes sistemas, tales como **Web Services**, Servidores XML y demás plataformas.

Por una parte los mensajes que intercambia la aplicación, al ser un estándar, permiten interactuar con sistemas tales como PHP o Java. Por otro parte, la transmisión de vídeos puede provenir de cualquier servidor que transmita multimedia con cualquier extensión de las admitidas por la SmartTV de Samsung.

Es esa **interoperabilidad** entre diferentes sistemas la cual concede libertad al proveedor de contenidos, de disponer la plataforma óptima para los servidores, sin preocuparse por conflictos de integración.

4.1.3.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una **solución** a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser **reusable**, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

En la Aplicación se pueden distinguir los siguientes patrones de diseño:

- **Patrón MVC (Modelo-Vista-Controlador):**
 - El Modelo que representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
 - La Vista que transforma el modelo en una página Web que permite al usuario interactuar con ella.
 - El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

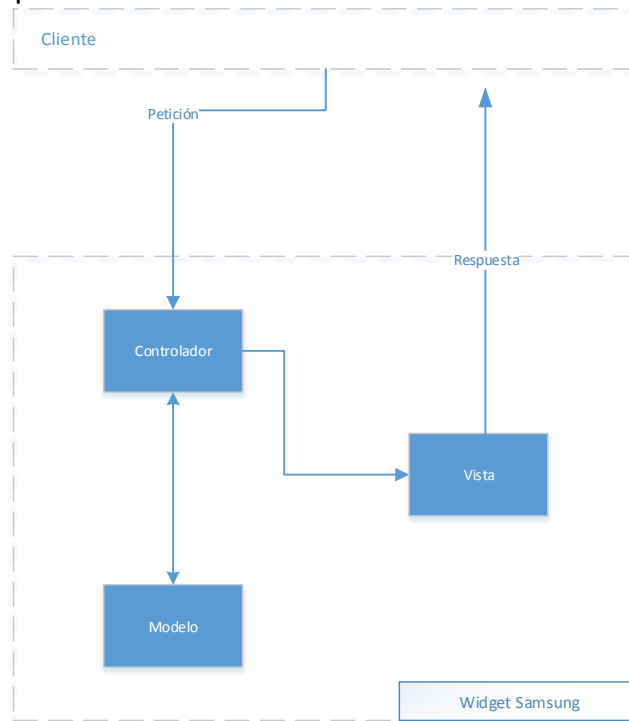


FIGURA 33: DIAGRAMA MODELO-VISTA-CONTROLADOR

La aplicación se compone de varias escenas en **HTML** con un fichero **CSS** asociado a cada escena. Todos los componentes de la aplicación se encuentran según el patrón **modelo-vista-controlador** anteriormente mencionado, y son implementados en el lenguaje de programación **JavaScript**.



FIGURA 34: ESTRUCTURA DE DIRECTORIO APP

En el inicio de la aplicación, el sistema operativo invoca a un *loader* para realizar el proceso de carga de los distintos componentes segmentados de la aplicación: html, estilo y componentes funcionales. Este es un gráfico de la secuencia:

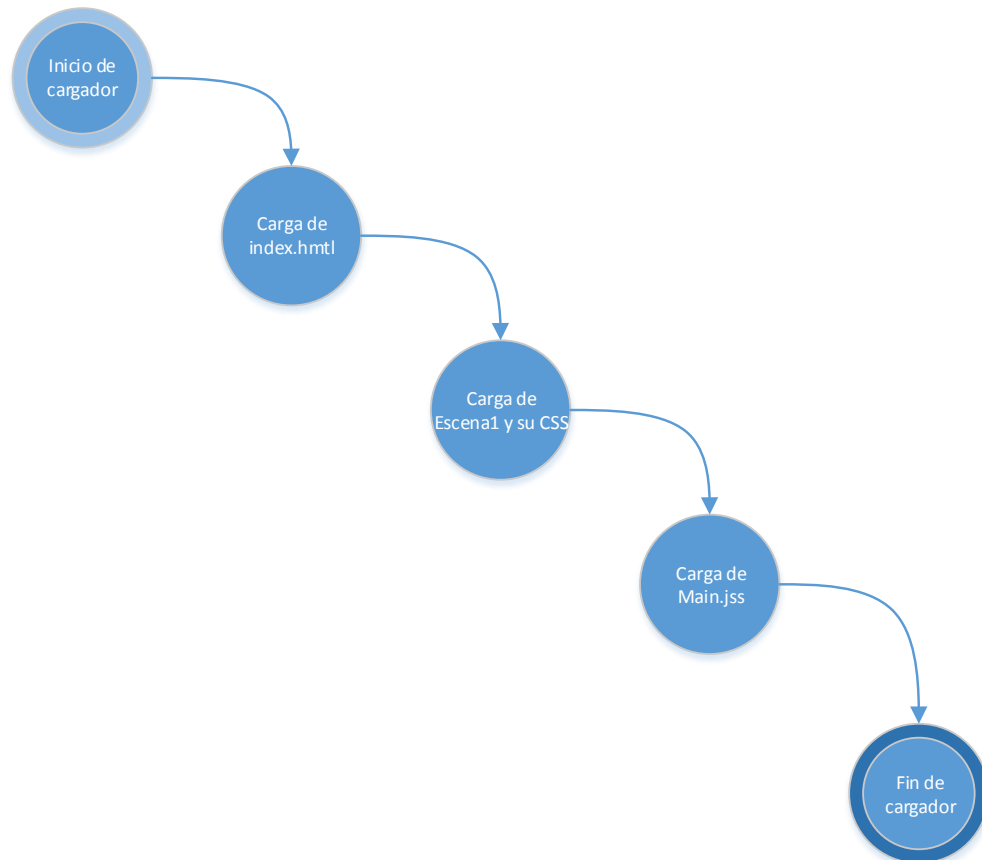


FIGURA 35: SECUENCIA CARGADOR DE SAMSUNG

Especificación de estándares

Un **estándar** es una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

En la Aplicación Web se pueden distinguir los siguientes estándares:

- **Estándares Web recomendados por el W3C:** La misión del W3C es: " *Guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web*".

En la aplicación se utilizan las siguientes tecnologías que se encuentran especificadas en los estándares:

- **HTML5**
- **XML**
- **CSS3**

4.1.4 Especificación de subsistemas de diseño

4.1.4.1 Diagrama de paquetes

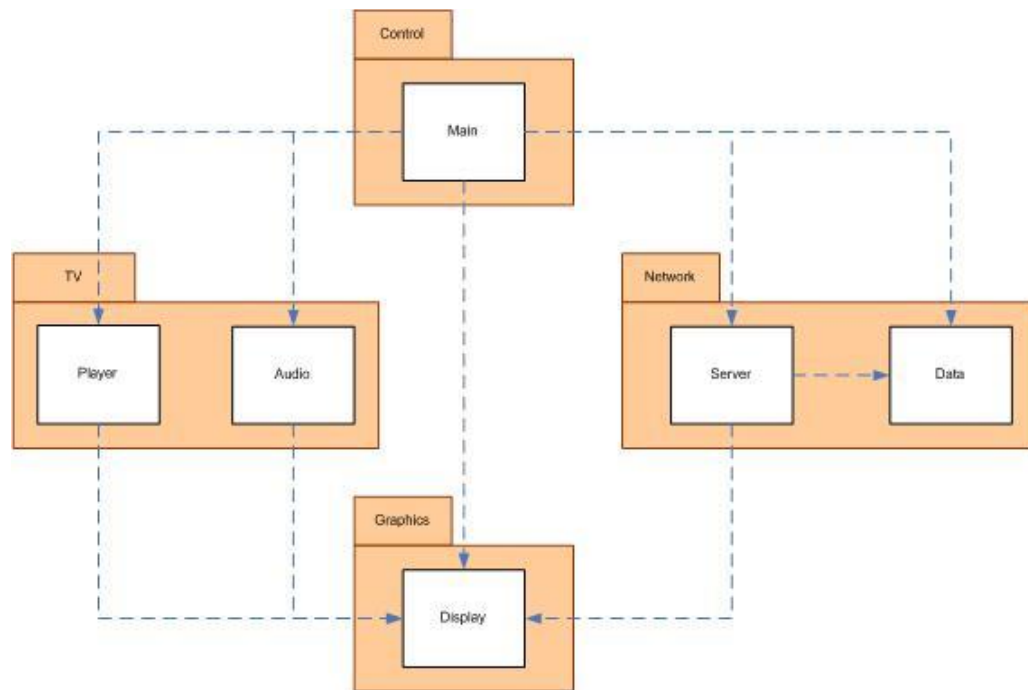


FIGURA 36: DIAGRAMA DE PAQUETES

4.1.4.2 Relación de subsistemas específicos

- **Subsistema main:** Responsable de coordinación de las funcionalidades de la aplicación.
- **Subsistema display:** Responsable para la visualización de los gráficos y la información de texto que utilizan HTML dinámico.
- **Subsistema data:** Se ocupa del almacenamiento de datos de vídeo dentro de la aplicación.
- **Subsistema server:** Maneja la recuperación de la fuente RSS desde el servidor de datos utilizando AJAX.
- **Subsistema bitrate:** Responsable de mostrar el contenido actual bitrates gráfico de barras en modo de pantalla completa.
- **Subsistema audio:** Controla el nivel de volumen de audio usando un plugin.
- **Subsistema player:** Controla la reproducción de audio y vídeo desde el servidor de contenido usando un plugin.

4.2 Diseño detallado de los subsistemas de diseño

4.2.1 Subsistema principal (main.js)

El Subsistema *Principal* actúa de controlador, se encarga de procesar las interacciones del usuario y lanzar todos aquellos componentes necesarios.

La capa del controlador consta de:

- **Un controlador frontal:** Carga los elementos de la aplicación y determina la acción a ejecutarse.
- **Acciones:** Las acciones contienen la lógica de la aplicación, pueden acceder a los datos. Son las diferentes funciones que componen al controlador.
- **Componentes:** Las componentes son parecidos a las acciones pero no necesitan ser invocados por el controlador frontal y las variables que definen no son utilizadas por las plantillas directamente, sino por los elementos parciales que se incluyen en éstas.

Métodos	Descripción
onLoad()	<ol style="list-style-type: none"> 1. Carga cuatro subsistemas: player, audio, display y server. 2. Declara una función para recuperar el modo de visión normal cuando finaliza un video a pantalla completa. 3. Declara una función mostrar en pantalla la lista de éxitos recibida. 4. Carga el fichero XML (Server.fetchVideoList) 5. Carga administrador de aplicaciones del televisor. (<i>Dispone de un método que notifica al administrador de aplicaciones de la partida de una aplicación, un método que registra y libera las teclas de los usuarios desean utilizar, y así sucesivamente.</i>)
onUnload()	Detiene el plugin del reproductor invocando a la función <i>player.deinit()</i>

Métodos	Descripción
updateCurrentVideo()	<p>Actualiza el listado de videoclips, los muestra en pantalla y actualiza algunos elementos:</p> <ul style="list-style-type: none"> • Url • Posición • Descripción del videoclip • Tipo de video: “on-Demand” o “live” • Lanza carátula del videoclip • Carga las puntuaciones de los videos.
enableKeys()	<p>Pone el foco sobre un elemento de javascript onClick() para, en caso de pulsar un botón del mando a distancia, invocar a una función de tratamiento.</p>
keyDown()	<p>Trata cualquier interrupción provocada por presionar un botón del mando a distancia. Los distintos tipos de pulsaciones reconocidas son:</p> <ul style="list-style-type: none"> • Return • Panel_return • Play • Stop • Pause • Key_FF • Key_RW • Key_Vol_Up • Key_Vol_Down • Key_Panel_Vol_Down • Key_Down • Key_Up • Key_Mute <p>En caso de presionar un botón no reconocido reporta el error a un log.</p>
handlePlayKey()	<p>Evalúa el estado del reproductor, si está pausado, continua la reproducción; si está parado, inicia la reproducción.</p>
handlePauseKey()	<p>Evalúa el estado del reproductor, si está reproduciendo, pausa la reproducción. De lo contrario, no hace nada.</p>
selectNextVideo()	<p>Selecciona el video siguiente de la lista y</p>

Métodos	Descripción
	detiene la reproducción del vídeo si se diese el caso.
selectPreviousVideo()	Selecciona el video anterior de la lista y detiene la reproducción del vídeo si se diese el caso.
setFullScreenMode()	<ol style="list-style-type: none"> 1. Evalúa el estado de la ventana de reproducción. 2. Amplia la vista del reproductor a pantalla completa.
setWindowMode()	<ol style="list-style-type: none"> 1. Evalúa el estado de la ventana de reproducción. 2. Recupera la vista del reproductor al tamaño normal
toggleMode()	<ol style="list-style-type: none"> 1. Si la reproducción está en pausa la reanuda. 2. Además, si la conmuta la el tamaño de la ventana de reproducción de vista normal, a pantalla completa o viceversa.
setMuteMode()	<ol style="list-style-type: none"> 1. Silencia el volumen. 2. Modifica el elemento gráfico del volumen del reproductor para indicar el nuevo estado.
noMuteMode()	<ol style="list-style-type: none"> 1. Activa el volumen. 2. Modifica el elemento gráfico del volumen del reproductor para indicar el nuevo estado.
muteMode()	Evalúa el estado del sistema, y llama a la función <i>setMuteMode()</i> , <i>noMuteMode()</i> , según proceda.

TABLA 43: MÉTODOS “SUBSISTEMA PRINCIPAL”

4.2.2 Subsistema display (display.js)

El subsistema visual tiene por cometido encargarse de todas aquellas modificaciones visuales dinámicas generadas como consecuencia del uso de la aplicación.

Métodos	Descripción
Init()	<ol style="list-style-type: none"> 1. Localiza a una variable dinámicamente una sección del <i>html</i> que será modificada posteriormente. 2. Devuelve éxito o error a una variable llamada <i>success</i>
setTotalTime(time)	Modifica una variable que contiene el tiempo total del videoclip.
setTime(time)	<p>Modifica el contador del tiempo en la visualización del video con el valor de <i>time</i>.</p> <p>Previamente se convierte valor de la variable en segundos al siguiente formato de tiempo (<i>HH:MM:ss</i>)</p>
status(status)	Pone el foco sobre un elemento de javascript
setVolume(level)	Modifica el elemento gráfico de la barra de sonido del reproductor y su valor numérico.
setVideoList(nameList)	Modifica el DOM de la página para mostrar el listado de videos contenidos en la variable <i>nameList</i>
setVideoListPosition(position, move)	Reordena el listado de videoclips según se ordene subir o bajar con el mando a distancia.
setDescription(description)	Muestra la descripción del video.
coverLaunch(url)	Lanza un visor de imágenes con la carátula del videoclip.

Métodos	Descripción
coverShow()	Muestra el visor de imágenes.
coverHide()	Oculto el visor de imágenes.
hide()	Oculto el contenido de la etiqueta <i>main</i> del <i>html</i> para visualizar en pantalla completa.
show()	Muestra el contenido de la etiqueta <i>main</i> para mostrar la vista normal de reproducción de video.
showVoteError()	Muestra un mensaje lanzando un popup indicando el envío exitoso del voto.
showVoteSend()	Muestra un mensaje lanzando un popup indicando el error en el envío del voto.
showScore(score)	Devuelve una etiqueta html de imagen (), asociada a una imagen de cuatro disponibles. La imagen corresponderá con el valor de la puntuación recibida como parámetro.

TABLA 44: MÉTODOS “SUBSISTEMA DISPLAY”

4.2.3 Subsistema data (data.js)

Este subsistema almacena y refresca los datos de la aplicación.

Métodos	Descripción
setVideoNames(names)	Almacena en un array los nombre de los distintos videoclips.
setVideoUrl (urls)	Almacena en un array las urls de los diferentes videoclips.
setVideoDescription(descriptions)	Almacena en un array las descripciones

Métodos	Descripción
	de los diferentes videoclips.
getVideoUrl(index)	Obtiene la url de un videoclip. <i>Index</i> contiene la posición del videoclip.
getVideoCount()	Devuelve un array con las urls de los videoclips.
getVideoNames()	Devuelve un array con los nombres de los videoclips.
getVideoDescription()	Devuelve un array con las descripciones de los videoclips.

TABLA 45: MÉTODOS “SUBSISTEMA DATA”

4.2.4 Subsistema server (server.js)

Contiene todos los métodos para procesar una petición a un servidor XML y obtener todos los datos relativos al listado de éxitos, *parsearlos* y actualizar la base de datos del sistema.

Métodos	Descripción
Init()	Inicializa un buffer para una futura petición <i>XMLHttpRequest</i> .
fetchVideoList()	Obtiene el XML mediante una <i>XMLHttpRequest</i> .
createVideoList()	Parsea el XML obtenido y genera el listado de videoclips.
vote(versionList, name, vote, mac)	Envía a un servidor determinado una petición <i>XMLHttpRequest</i> con un objeto de tipo POST con los datos de la votación del usuario. Datos: <ul style="list-style-type: none"> • versionList: Versión de la lista votada. • name: identificador del videoclip en

Métodos	Descripción
	la lista. <ul style="list-style-type: none"> • vote: voto emitido por el usuario. • mac: identificador del televisor que envía el voto.

TABLA 46: MÉTODOS “SUBSISTEMA SERVER”

4.2.5 Subsistema audio (audio.js)

Modifica el volumen de la aplicación invocando a funciones dispuestas por el sistema.

Métodos	Descripción
Init()	Captura a una variable el plugin de audio.
setRelativeVolumen(delta)	Modifica el volumen del plugin y la barra de volumen.
getVolume(delta)	Obtiene el valor numérico del volumen.

TABLA 47: MÉTODOS “SUBSISTEMA AUDIO”

4.2.6 Subsistema player (player.js)

Contiene todas las funciones para controlar la reproducción de los videoclips con las garantías de calidad del servicio.

Métodos	Descripción
Init()	<ol style="list-style-type: none"> 1. Captura a una variable el plugin de video. 2. Asigna el estado del plugin como <i>stop</i>. 3. Muestra en pantalla la ventana de video. 4. Declara funciones para mostrar el tiempo de reproducción, y las acciones cuando se complete el buffer de video.
delnit()	Se restaura la configuración original y se liberan los recursos al cerrar la aplicación.

Métodos	Descripción
setWindow()	Se modifica tamaño de la ventana de reproducción a tamaño normal.
setFullscreen()	Se modifica tamaño de la ventana de reproducción a pantalla completa.
setVideoURL(url)	Se fija la url para posterior visionado del video.
playVideo()	<p>Evalúa si hay url asignada</p> <ol style="list-style-type: none"> 1. En caso de existir url: <ol style="list-style-type: none"> a. Modifica estado a <i>playing</i>. b. Oculta el visor de imágenes con la carátula. c. Ajusta el marco de pantalla del plugin de video. d. Inicializa los buffers de video. e. Invoca la función play del plugin. f. Desactiva el audio silenciado del plugin de audio. 2. En caso de no existir url: <ol style="list-style-type: none"> a. Lanza alerta.
pauseVideo()	<ol style="list-style-type: none"> 1. Se modifica la variable de estado. 2. Se modifican los parámetros visuales necesarios. 3. Se pausa la reproducción.
stopVideo()	<ol style="list-style-type: none"> 1. Se verifica estado del video 2. Si el video está en un estado distinto de <i>stop</i>: <ol style="list-style-type: none"> a. La variable estado pasa a ser <i>stop</i> b. Se modifican los parámetros visuales necesarios. c. Se detiene la reproducción. d. Se lanza el visor de imágenes con la carátula del video. 3. Si es vídeo está en estado <i>stop</i> se lanza una alerta.
resumeVideo()	<ol style="list-style-type: none"> 1. Se verifica estado del video 2. La variable estado pasa a ser <i>play</i>. 3. Se modifican los parámetros visuales necesarios. 4. Se inicia la reproducción.

Métodos	Descripción
skipForwardVideo()	<ol style="list-style-type: none"> 1. Se modifica el estado 2. Se saltan 5 segundos hacia delante.
skipBackwardVideo()	<ol style="list-style-type: none"> 1. Se modifica el estado 2. Se saltan 5 segundos hacia delante.
getState()	Se devuelve el estado del reproductor.
onBufferingStart()	<ol style="list-style-type: none"> 1. Se visualiza el estado de “<i>Buffering</i>”. 2. Se atenúan los botones de rebobinado.
onBufferingComplete()	Se muestra el porcentaje cargado en el buffering.
onBufferingProgress()	<ol style="list-style-type: none"> 1. Se muestra el estado de <i>play</i>. 2. Se muestran los botones de rebobinado.
setCurTime()	Se modifica el tiempo de reproducción.
setTotalTime()	Se modifica el tiempo total de reproducción.
onServerError()	Se lanza una alerta por error del servidor.
OnNetworkDisconnected()	Se lanza una alerta por error desconexión de la red.
getBandwidth(bandwidth)	Se lanza una alerta para depuración indicando el valor del ancho de banda.
onDecoderReady()	Se lanza una alerta para depuración indicando inicio de decodificado.
onRenderError()	Se lanza una alerta indicando error en renderizado.
stopPlayer()	Se lanza una alerta para depuración, indicando stop de la reproducción.
setTottalBuffer()	Ajusta tamaño máximo del buffer.
setCurBuffer()	Ajusta tamaño del buffer mínimo.

TABLA 48: MÉTODOS “SUBSISTEMA PLAYER”

4.4 Ficheros auxiliares

A continuación se muestran los diferentes **módulos** que se han integrado en este proyecto en formato plugin, procedentes de la biblioteca de SAMSUNG SMART TV que se han importado como objetos en el [index.html](#):

4.4.1 Plugin de video

La aplicación importa un plugin de video desarrollado por Samsung que contiene toda la lógica de **reproducción**.

El plugin dispone de la siguiente API:

<i>Function SetDisplayArea</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	SetDisplayArea(Number x, Number y, Number width, Number height)
Return Value	Returns true if it succeeds, else returns false.
Parameter	<p>X</p> <p>Number</p> <p>Specifies the initial x-coordinate of display area in TV screen coordinates. This x value must be smaller than the width of the TV screen, 960.</p> <p>Y</p> <p>Number</p> <p>Specifies the initial y-coordinate of display area in TV screen coordinates. This y value must be smaller than the height of the TV screen, 540.</p> <p>Width</p> <p>Number</p> <p>The width of the display area. This value must be smaller than width of the TV screen.</p> <p>Height</p> <p>Number</p> <p>The height of crop area from source Image. This value must be smaller than height of source image.</p>

TABLA 49: PLUGIN DE VÍDEO “FUNCIÓN SETDISPLAYAREA”

<i>Function SetInitialBuffer</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	SetInitialBuffer(Number byte)
Return Value	true on success, else false

TABLA 50: PLUGIN DE VÍDEO “FUNCIÓN SETINITIALBUFFER”

<i>Function SetPendingBuffer</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	SetPendingBuffer(Number byte)
Return Value	true on success, else false
Parameter	byte Number Specifies the buffer size in bytes which media player goes out from buffering. This API is optional and media player has a default value.

TABLA 51: PLUGIN DE VÍDEO “FUNCIÓN SETPENDINGBUFFER”

<i>Function Play</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	Play(String url)

<i>Function Play</i>	
Return Value	true on success, else false
Parameter	url String Specifies the URL of content

TABLA 52: PLUGIN DE VÍDEO “FUNCIÓN PLAY”

<i>Event OnCurrentPlayTime</i>	
Version	PLAYER-0001
Parameter	milli-sec Number

TABLA 53: PLUGIN DE VÍDEO “EVENTO ONCURRENTPLAYTIME”

<i>Event OnBufferingStart</i>	
Version	PLAYER-0001
Example	<pre> Player.OnBufferingStart=OnBufferingStart; function OnBufferingStart() { //Drawing buffering image. } </pre>

TABLA 54: PLUGIN DE VÍDEO “EVENTO ONBUFFERINGSTART”

<i>Event OnBufferingProgress</i>	
Version	PLAYER-0001
Parameter	percent

<i>Event OnBufferingProgress</i>	
	Number
Remarks	The minimum value of parameter is 0 and the maximum is 100.

TABLA 55: PLUGIN DE VÍDEO “EVENTO ONBUFFERTINGPROGRESS”

<i>Event OnBufferingComplete</i>	
Version	PLAYER-0001
Example	<pre> Player.OnBufferingComplete=OnBufferingComplete; function OnBufferingComplete() { //unload buffering image. } </pre>

TABLA 56: PLUGIN DE VÍDEO “EVENTO ONBUFFERINGCOMPLETE”

<i>Function Pause</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	Pause()
Return Value	true on success, else false

TABLA 57: PLUGIN DE VÍDEO “FUNCIÓN PAUSE”

<i>Function Stop</i>	
Version	Support from PLAYER-0001
Syntax	Stop()

<i>Function Stop</i>	
Return Value	true on success, else false

TABLA 58: PLUGIN DE VÍDEO “FUNCIÓN STOP”

<i>Function Resume</i>	
Version	Support from PLAYER-0001
Syntax	Resume()
Return Value	true on success, else false

TABLA 59: PLUGIN DE VÍDEO “FUNCIÓN RESUME”

<i>Function JumpForward</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	JumpForward(Number offset)
Return Value	true on success, else false
Parameter	offset Number Relative time offset from current time in seconds

TABLA 60: PLUGIN DE VÍDEO “FUNCIÓN JUMPFORWARD”

<i>Function JumpBackward</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	JumpBackward(Number offset)

<i>Function JumpBackward</i>	
Return Value	true on success, else false
Parameter	offset Number Relative time offset from current time in seconds

TABLA 61: PLUGIN DE VÍDEO “FUNCIÓN JUMPBACKWARD”

<i>Function GetDuration</i>	
Version	Support from PLAYER-0001
Security Type	Multimedia
Syntax	GetDuration()
Return Value	The total playing time on success, else -1

TABLA 62: PLUGIN DE VÍDEO “FUNCIÓN GETDURATION”

4.4.1.1 Plugin de audio

El plugin de audio concentra las funciones de **audio**, que en este caso corresponden al silenciado, aumento y disminución del volumen de la aplicación.

<i>SetVolumeWithKey (Function)</i>	
Version	Support from AUDIO-0002
Security Type	Multimedia
Syntax	SetVolumeWithKey (key)
Return Value	PLR_TRUE on success, else a negative number
Parameter	key

SetVolumeWithKey (Function)	
	<ul style="list-style-type: none"> ▪ PL_AUDIO_VOLUME_KEY ▪ PL_AUDIO_VOLUME_KEY_UP : To increase volume level ▪ PL_AUDIO_VOLUME_KEY_DOWN : To decrease volume level

TABLA 63: PLUGIN DE AUDIO “FUNCIÓN SETVOLUMENWITHKEY”

GetVolume (Function)	
Version	Support from AUDIO-0001
Security Type	Multimedia
Syntax	GetVolume ()
Return Value	A value between 0 and 100 as a volume level on success, else a negative number

TABLA 64: PLUGIN DE AUDIO “FUNCIÓN GETVOLUME”

SetSystemMute (Function)	
Version	Support from AUDIO-0005
Security Type	Multimedia
Syntax	SetSystemMute (systemMute)
Return Value	PLR_TRUE on success, else a negative number
Parameter	systemMute <ul style="list-style-type: none"> ▪ PLBOOL ▪ PLR_TRUE: To set state as SystemMute ▪ PLR_FALSE: To set state as non-SystemMute

TABLA 65: PLUGIN DE AUDIO “FUNCIÓN SETSYSTEMMUTE”

4.4.1.2 Plugin TVMW

Responsable de obtener el **perfil de usuario** con su idioma, país, y llave de registro. Esto se incluye para posteriores ampliaciones.

<i>FunctionGetProfile</i>	
Version	Support from TVMW-0001
Security Type	System
Syntax	GetProfile(<i>PL_TVMW_PRfid</i> profileID)
Return Value	If success, returns value of profile ID\Otherwise, returns error code.
Parameter	profileID <i>PL_TVMW_PRfid</i> Profile id to get profile information

TABLA 66: PLUGIN TVMW “FUNCIÓN GETPROFILE”

<i>FunctionSetProfile</i>	
Version	Support from TVMW-0001
Security Type	System
Syntax	SetProfile(<i>PL_TVMW_PRfid</i> profileID , STRING value)
Return Value	If success, returns PLR_TRUE. Otherwise, returns error code.
Parameter	profileID <i>PL_TVMW_PRfid</i> Designated field to be set value STRING Value to be set

TABLA 67: PLUGIN TVMW “FUNCIÓN SETPROFILE”

4.4.1.3 Plugin NETWORK

Se trata de un plugin que devuelve información acerca del sistema de red del televisor. En este proyecto su cometido es obtener la dirección MAC del televisor, que a todos los efectos se convierte en un identificador fiable, cuya finalidad será evitar votaciones masivas de un mismo videoclip.

<i>Function GetMAC</i>	
Version	Support from NETWORK-0001
Syntax	GetMAC (Number interfaceType)
Return Value	MAC address string on success, else a null string on error
Parameter	<p>interfaceType (<i>Number</i>) This function accepts call with a single parameter, or without a parameter</p> <p>The types of interface (wired/wireless)</p> <p>1 = wired, 0 = wireless</p> <p>If called without a parameter, it returns the result for a wired network</p>

TABLA 68: PLUGIN NETWORK “FUNCIÓN GETMAC”

5 PRUEBAS

5.1 Descripción general

Los casos de uso seleccionados, para efectuar las pruebas del sistema, se han elegido teniendo en cuenta su importancia respecto de las distintas funcionalidades del sistema, así como de la complejidad de los mismos. En este sentido se ha hecho hincapié en aquellos casos de uso relacionados con:

Reproducción de videoclips.
Lectura de lista de videoclips.
Selección de videoclips.

Todos ellos relacionados con las funcionalidades de mayor interés, que conforman el grueso de la aplicación.

La verificación del artefacto software se desarrollará conforme al plan de pruebas diseñado, el cual dispone un marco de referencia para la generación de las pruebas y su posterior validación. [Plan de pruebas](#)

A continuación se presenta el formato que se utilizará como “Release Notes”, el cual deberá acompañar cada una de las versiones entregadas para pruebas.

5.2 Batería de pruebas V1.1

Este documento contiene el resultado de la aplicación del plan de pruebas para la primera versión de la aplicación.

5.2.1 Requerimientos del sistema

COMPONENTE	REQUERIMIENTO
HARDWARE	<ul style="list-style-type: none"> Procesador de 2 núcleos con 2 GHz de frecuencia y 2 GB de memoria RAM.
SISTEMA OPERATIVO	<ul style="list-style-type: none"> Windows 7 versión 32 o 64 bits
SOFTWARE BASE	<ul style="list-style-type: none"> Entorno SDK de Samsung con emulador de televisor.

TABLA 69: BATERÍA DE PRUEBAS. REQUERIMIENTOS DEL SISTEMA

5.2.2 CASOS DE PRUEBAS

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 1	1-1.1		VERSIÓN DE EJECUCIÓN		1	
			FECHA EJECUCIÓN		09/04/2013	
CASO DE USO:	1.1		MODULO DEL SISTEMA		Main	
Descripción del caso de prueba:	Inicio correcto de la aplicación					
1. CASO DE PRUEBA						
a. Precondiciones						
Ninguna						
b. Pasos de la prueba						
Lanzamiento de la aplicación.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Load OK	X		Load OK
c. Post condiciones						
Aplicación iniciada.						
2. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones			Probador			

<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 70: CASO DE PRUEBA 1-1.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 2	2-1.1		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	09/04/2013		
CASO DE USO:	1.1		MODULO DEL SISTEMA	Main		
Descripción del caso de prueba:	Inicio correcto de la aplicación después de cierre					
3. CASO DE PRUEBA						
d. Precondiciones						
Aplicación cerrada previamente.						
e. Pasos de la prueba						
Lanzamiento de la aplicación. Cierre de la aplicación. Lanzamiento de la aplicación.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
-	-	Correcto	Load OK	X		Load OK
f. Post condiciones						
Aplicación iniciada.						

4. RESULTADOS DE LA PRUEBA	
Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso
Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 71: CASO DE PRUEBA 2-1.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	1-1.2	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	09/04/2013
CASO DE USO:	1.2	MODULO DEL SISTEMA	Main
Descripción del caso de prueba:	Cierre correcto de la aplicación		
5. CASO DE PRUEBA			
g. Precondiciones			
Aplicación lanzada			
h. Pasos de la prueba			
Verificar correcta restauración de contexto del sistema y cierre de la aplicación.			

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Unload OK	X		Unload OK
i. Post condiciones						
Aplicación cerrada y contexto restaurado del sistema.						
6. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
Ninguno						<input checked="" type="checkbox"/> Paso
Observaciones				Probador		
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
				Firma: Nombre: Fecha:		

TABLA 72: CASO DE PRUEBA 1-1.2

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	1-2.1	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	09/04/2013
CASO DE USO:	2.1	MODULO DEL SISTEMA	Server
Descripción del caso de prueba:	Obtención de ranking de videoclips		
7. CASO DE PRUEBA			
j. Precondiciones			

Fichero XML con listado de videoclips disponible en su ubicación.

k. Pasos de la prueba

Lanzar el método CreateVideoList

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
title	Video oso	Correcto	Video oso	X		-
link	http://127.0.0.1/video.mp4	Correcto	http://127.0.0.1/video.mp4	X		-
description	Video test	Correcto	Video test	X		-
type	onDemand	Correcto	onDemand	X		-
cover	http://cdn.arstechnica.net/wp-content/uploads/2012/10/06_PLace_2073_1_mis.jpg	Correcto	http://cdn.arstechnica.net/wp-content/uploads/2012/10/06_PLace_2073_1_mis.jpg	X		-

l. Post condiciones

Array con el listado de videoclips

8. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso

Observaciones	Probador	
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>		
	Firma: Nombre: Fecha:	

TABLA 73: CASO DE PRUEBA 1-2.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 2	2-2.1		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	09/04/2013		
CASO DE USO:	2.1		MODULO DEL SISTEMA	Server		
Descripción del caso de prueba:	Comprobación de estabilidad de la aplicación con xml incorrecto.					
9. CASO DE PRUEBA						
m. Precondiciones						
Fichero XML con listado de videoclips disponible en su ubicación con errores de etiquetado.						
n. Pasos de la prueba						
Lanzar el método CreateVideoList						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
tittle	Etiqueta xml incorrect a	Incorrect a	No mostrar listado	X		No mostrar listado

o. Post condiciones	
No mostrar listado sin cierre inesperado de la aplicación	
10.RESULTADOS DE LA PRUEBA	
Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso
Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 74: CASO DE PRUEBA 2-2.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	1-2.3	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	15/05/2013
CASO DE USO:	2.3	MODULO DEL SISTEMA	Principal
Descripción del caso de prueba:	Visualización carátula de videoclip		
11.CASO DE PRUEBA			
p. Precondiciones			
Fichero XML con listado de carátulas disponible en su ubicación.			
q. Pasos de la prueba			
Se selecciona hacia abajo los sucesivos elementos de la lista.			

[Una vez concluida la operación anterior] Se seleccionan los elementos en orden ascendente. Se comprueba que la información de cada videoclip es correcta.

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
cover	http://www.los40.com/dmz/especial/conciertos-40/img/lagarto-amarillo.jpg?20130410	Correcto	Mostrar carátula	X		Carátula mostrada
cover	http://cdn.arstechnica.net/wp-content/uploads/2012/10/06_PLace_2073_1_mis.jpg	Correcto	Mostrar carátula	X		Carátula mostrada
cover	-	Incorrecto	Mensaje de error	X		Mensaje de error

r. Post condiciones

Se visualiza carátula del videoclip.

12.RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto
En algunas ocasiones si se navega entre los diferentes videoclips, la carátula no resulta mostrada.	<input checked="" type="checkbox"/> Paso

Observaciones	Probador	
Se prevee que los problemas de la visualización de la carátula son responsabilidad del emulador.		
	Firma:	
	Nombre:	
	Fecha:	

TABLA 75: CASO DE PRUEBA 1-2.3

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 1	1-2.4		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	13/05/2013		
CASO DE USO:	2.4		MODULO DEL SISTEMA	Principal		
Descripción del caso de prueba:	Consulta votos de videoclips.					
13.CASO DE PRUEBA						
s. Precondiciones						
Servidor de votaciones activo.						
t. Pasos de la prueba						
Se inicia la aplicación.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
Video1.vote	4 estrellas	Correcto	4 estrellas	X		4 estrellas
Video2.vote	3 estrellas	Correcto	3 estrellas	X		3 estrellas
Video3.vote	2 estrellas	Correcto	2 estrellas	X		2 estrellas

Video4.vote	1 estrella	Correcto	1 estrella	X		1 estrella
Video5.vote	Sin votos	Correcto	Sin votos	X		Sin votos
u. Post condiciones						
Se cargan todas las votaciones asignadas a los videoclips.						
14.RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
-						<input checked="" type="checkbox"/> Paso
Observaciones				Probador		
El elemento visual de las votaciones se encuentre ligeramente desplazado a la derecha, quedando fuera del área de selección del cursor cuando se selecciona el vídeo.				Firma: Nombre: Fecha:		

TABLA 76: CASO DE PRUEBA 1-2.4

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	1-2.5	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	13/05/2013
CASO DE USO:	2.5	MODULO DEL SISTEMA	Principal
Descripción del caso de prueba:	Votación de vídeo		
15.CASO DE PRUEBA			
v. Precondiciones			
Servidor de recepción de votaciones activo.			
w. Pasos de la prueba			
Se inicia la aplicación. Se selecciona un vídeo. Se presiona uno de los cuatro botones de colores para votación (azul, amarillo, verde, rojo).			

** Se repite la operación para cada color.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
Tv.key	Red	Correcto	Votación enviada con éxito	X		Votación enviada con éxito
Tv.key	Green	Correcto	Votación enviada con éxito	X		Votación enviada con éxito
Tv.key	Yellow	Correcto	Votación enviada con éxito	X		Votación enviada con éxito
Tv.key	Blue	Correcto	Votación enviada con éxito	X		Votación enviada con éxito
x. Post condiciones						
El voto es enviado al servidor asignado.						
16.RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones			Probador			
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
			Firma: Nombre: Fecha:			

TABLA 77: CASO DE PRUEBA 1-2.5

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

CASO DE PRUEBA No. 1	2-2.5		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	13/05/2013		
CASO DE USO:	2.5		MODULO DEL SISTEMA	Principal		
Descripción del caso de prueba:	Votación de vídeo					
17.CASO DE PRUEBA						
y. Precondiciones						
Servidor de recepción de votaciones inactivo						
z. Pasos de la prueba						
Se inicia la aplicación. Se selecciona un vídeo. Se presiona el botón de votación rojo.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
Tv.key	Red	Correcto	Se ha producido un error en el envío de votación	X		Se ha producido un error en el envío de votación
aa. Post condiciones						
Se detecta el error y se lanza mensaje.						
18.RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones			Probador			

<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 78: CASO DE PRUEBA 2-2.5

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 1	3-2.5		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	13/05/2013		
CASO DE USO:	2.3		MODULO DEL SISTEMA	Principal		
Descripción del caso de prueba:	Votación de vídeo					
19. CASO DE PRUEBA						
bb. Precondiciones						
Servidor de recepción de votaciones activo.						
cc. Pasos de la prueba						
Se reproduce video. Se presiona botón amarillo de votación.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
Tv.key	Yellow	Correcto	Votación enviada con éxito	X		Votación enviada con éxito
dd. Post condiciones						
Se envía votación del videoclip al servidor asignado.						
20. RESULTADOS DE LA PRUEBA						

Defectos y desviaciones		Veredicto
Se detecta que al lanzarse el PopUp que indica el éxito de la operación, y presionarse la tecla "OK" para cerrar el PopUp, la ventana de reproducción resulta aumentada.		<input checked="" type="checkbox"/> Paso
Observaciones	Probador	
Parece que al lanzarse el PopUp, la aplicación no pone el foco del mando sobre el elemento.		
	Firma: Nombre: Fecha:	

TABLA 79: CASO DE PRUEBA 3-2.5

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	1-3.1	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	09/04/2013
CASO DE USO:	3.1	MODULO DEL SISTEMA	Player
Descripción del caso de prueba:	Reproducción de videoclip		
21. CASO DE PRUEBA			
ee. Precondiciones			
Video disponible en un directorio local Url del video asignada.			
ff. Pasos de la prueba			
Se lanza el método playVideo()			
DATOS DE ENTRADA	RESPUESTA	COINCI	RESPUESTA DEL

			ESPERADA DE LA APLICACIÓN	DE		SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Reproduce video	X		Reproduce video
gg. Post condiciones						
Se reproduce el video con la url previamente asignada.						
22. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
Ninguno						<input checked="" type="checkbox"/> Paso
Observaciones				Probador		
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
				Firma: Nombre: Fecha:		

TABLA 80: CASO DE PRUEBA 1-3.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	2-3.1	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	11/04/2013
CASO DE USO:	3.1	MODULO DEL SISTEMA	Player
Descripción del caso de prueba:	Reproducción de videoclip por streaming		
23. CASO DE PRUEBA			
hh. Precondiciones			
Video bajo demanda con extensión .mp4 en un servidor de streaming online.			
ii. Pasos de la prueba			

Se ordena reproducción de vídeo. Se detiene el vídeo. Se ordena reproducción de vídeo.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Reproduce video	X		Reproduce video
jj. Post condiciones						
Se reproduce el video con la url previamente asignada.						
24.RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
Ninguno						<input checked="" type="checkbox"/> Paso
Observaciones				Probador		
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
				Firma: Nombre: Fecha:		

TABLA 81: CASO DE PRUEBA 2-3.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 1	3-3.1	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	11/04/2013
CASO DE USO:	3.1	MODULO DEL SISTEMA	Player
Descripción del caso de prueba:	Reproducción de videoclip en directo		

25.CASO DE PRUEBA						
kk. Precondiciones						
Streaming de video con formato HLS de un servidor disponible.						
ll. Pasos de la prueba						
Se ordena reproducción de vídeo. Se detiene el vídeo. Se ordena reproducción de vídeo.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Reproduce video	X		Reproduce video
mm. Post condiciones						
Se reproduce el video con la url previamente asignada.						
26.RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
<p>La barra de progreso sale del marco al cumplirse el minuto de reproducción.</p> <p>El contador del tiempo marca un final de vídeo no congruente, al ser video en directo.</p>					<input checked="" type="checkbox"/> Falló	
Observaciones			Probador			
Se insta a subsanar el defecto de reproducción.						
			Firma:			
			Nombre:			
			Fecha:			

TABLA 82: CASO DE PRUEBA 3-3.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

CASO DE PRUEBA No. 1	4-3.1		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	19/04/2013		
CASO DE USO:	3.1		MODULO DEL SISTEMA	Player		
Descripción del caso de prueba:	Reproducción de videoclips con diferentes formatos					
27.CASO DE PRUEBA						
nn.Precondiciones						
Videos disponibles en modo local con formatos: mp4, avi, mkv, 3gp, ts, mpg.						
oo.Pasos de la prueba						
Se ordena reproducción de vídeo. Se detiene el vídeo. Se ordena reproducción de vídeo.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
url (video)	http://127.0.0.1/video.mp4	Correcto	Reproduce video	X		Reproduce video
url (video)	http://127.0.0.1/video.avi	Correcto	Reproduce video	X		Reproduce video
url (video)	http://127.0.0.1/video.mkv	Correcto	Reproduce video		X	No reproduce video
url (video)	http://127.0.0.1/video.3gp	Correcto	Reproduce video	X		Reproduce video
url (video)	http://127.0.0.1/video.ts	Correcto	Reproduce video	X		Reproduce video
url (video)	http://127.0.0.1/video.mpg	Correcto	Reproduce video	X		Reproduce video
pp. Post condiciones						
Se reproduce el video con la url previamente asignada.						
28.RESULTADOS DE LA PRUEBA						

Defectos y desviaciones		Veredicto
El formato mkv no es reproducido por el plugin de video.		<input checked="" type="checkbox"/> Paso
Observaciones	Probador	
Los formatos recomendado por Samsung: MP4 y AVI; se ejecutan sin ninguna desviación.		
	Firma: Nombre: Fecha:	

TABLA 83: CASO DE PRUEBA 4-3.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 6	1-3.2		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	09/04/2013		
CASO DE USO:	3.2		MODULO DEL SISTEMA	Player		
Descripción del caso de prueba:	Pausar videoclip					
29. CASO DE PRUEBA						
qq. Precondiciones						
Video en reproducción.						
rr. Pasos de la prueba						
Se pausa el video mediante el método player.Pause()						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
-	-	Correcto	Pausa video	X		Pausa video
ss. Post condiciones						

Se pausa el video.	
30.RESULTADOS DE LA PRUEBA	
Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso
Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 84: CASO DE PRUEBA 1-3.2

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 6	2-3.2	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	19/04/2013
CASO DE USO:	3.2	MODULO DEL SISTEMA	Player
Descripción del caso de prueba:	Pausar videoclip		
31.CASO DE PRUEBA			
tt. Precondiciones			
Videoclip en reproducción.			
uu. Pasos de la prueba			
Se pausa el videoclip en reproducción. Se amplía a modo de pantalla completa. Se pausa vídeo.			

DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Videoclip pausado	X		Videoclip pausado
vv. Post condiciones						
Se pausa el video.						
32. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
<p>Al ampliar a modo de pantalla completa el videoclip en modo PAUSA, este se reproduce pese a seguir en el estado PAUSA.</p> <p>Sólo se puede restablecer presionando el botón de REPRODUCIR para recuperar la funcionalidad.</p>						<input checked="" type="checkbox"/> Falló
Observaciones				Probador		
Revisar funcionalidades de <i>Pausar Video</i> y <i>Pantalla Completa</i> .				Firma: Nombre: Fecha:		

TABLA 85: CASO DE PRUEBA 2-3.2

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 7	1-3.3	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	09/04/2013

CASO DE USO:	3.3	MODULO DEL SISTEMA	Player
Descripción del caso de prueba:	Detener videoclip		
33. CASO DE PRUEBA			
ww. Precondiciones			
Video en reproducción			
xx. Pasos de la prueba			
Se detiene la reproducción del vídeo.			
DATOS DE ENTRADA			RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO	
			RESPUESTA ESPERADA DE LA APLICACIÓN
			COINCIDE
			SI
			NO
-	-	Correcto	Detención de vídeo.
			X
yy. Post condiciones			
Se detiene el vídeo.			
34. RESULTADOS DE LA PRUEBA			
Defectos y desviaciones			Veredicto
Ninguno			<input checked="" type="checkbox"/> Paso
Observaciones		Probador	
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>			
		Firma:	
		Nombre:	
		Fecha:	

TABLA 86: CASO DE PRUEBA 1-3.3

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 7	2-3.3		VERSIÓN DE EJECUCIÓN		1	
			FECHA EJECUCIÓN		09/04/2013	
CASO DE USO:	3.3		MODULO DEL SISTEMA		Player	
Descripción del caso de prueba:	Detener videoclip pausado					
35. CASO DE PRUEBA						
zz. Precondiciones						
Video en pausa						
aaa. Pasos de la prueba						
Se detiene la reproducción del vídeo.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Detención de vídeo.	X		Detención de vídeo.
bbb. Post condiciones						
Se detiene el vídeo.						
36. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones			Probador			
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						

	Firma: Nombre: Fecha:
--	-----------------------------

TABLA 87: CASO DE PRUEBA 2-3.3

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 8	1-3.4		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	09/04/2013		
CASO DE USO:	3.4		MODULO DEL SISTEMA	Player		
Descripción del caso de prueba:	Rebobinado de avance rápido de videoclip					
37. CASO DE PRUEBA						
ccc. Precondiciones						
Video en reproducción.						
ddd. Pasos de la prueba						
Se avanza la reproducción 5 segundos.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
-	-	Correcto	Avance de 5 segundos	X		Avance de 5 segundos
eee. Post condiciones						
Se avanza el vídeo 5 segundos.						
38. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	

Observaciones	Probador	
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>		
	Firma:	
	Nombre:	
	Fecha:	

TABLA 88: CASO DE PRUEBA 1-3.4

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 9	1-3.5		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	09/04/2013		
CASO DE USO:	3.5		MODULO DEL SISTEMA	Player		
Descripción del caso de prueba:	Rebobinado de retroceso rápido de videoclip					
39. CASO DE PRUEBA						
fff. Precondiciones						
Video en reproducción.						
ggg. Pasos de la prueba						
Se retrocede la reproducción 5 segundos.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
-	-	Correcto	Retroceso de 5 segundos	X		Retroceso de 5 segundos
hhh. Post condiciones						
Se retrocede el vídeo 5 segundos.						
40. RESULTADOS DE LA PRUEBA						

Defectos y desviaciones		Veredicto
Ninguno		<input checked="" type="checkbox"/> Paso
Observaciones	Probador	
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>		
	Firma: Nombre: Fecha:	

TABLA 89: CASO DE PRUEBA 1-3.5

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 10	1-3.6	VERSIÓN DE EJECUCIÓN		1		
		FECHA EJECUCIÓN		10/04/2013		
CASO DE USO:	3.6	MODULO DEL SISTEMA		Player		
Descripción del caso de prueba:	Modo pantalla completa					
41. CASO DE PRUEBA						
iii. Precondiciones						
Video en reproducción.						
jjj. Pasos de la prueba						
Se invoca la función que modifica el área del reproductor.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Visualización a	X		Visualización a pantalla

			pantalla completa			completa
kkk. Post condiciones						
Se visualiza el video a pantalla completa.						
42.RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones				Probador		
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
				Firma: Nombre: Fecha:		

TABLA 90: CASO DE PRUEBA 1-3.6

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 10	2-3.6	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	12/04/2013
CASO DE USO:	3.6	MODULO DEL SISTEMA	Player
Descripción del caso de prueba:	Modo pantalla completa con reproducción en directo.		
43.CASO DE PRUEBA			
III. Precondiciones			
Video para reproducción en directo disponible.			
mmm. Pasos de la prueba			
Se reproduce vídeo en directo.			

Se invoca la función que modifica el área del reproductor.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Visualización a pantalla completa	X		Visualización a pantalla completa
nnn. Post condiciones						
Se visualiza el video a pantalla completa.						
44. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
En algunos casos se produce una excepción en el emulador del televisor. Sin embargo, siguiendo los mismos pasos, en otros no se produce.						<input checked="" type="checkbox"/> Paso
Observaciones				Probador		
Se sospecha posible fallo del emulador. Se sugiere probar en televisor.						
				Firma: Nombre: Fecha:		

TABLA 91: CASO DE PRUEBA 2-3.6

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No. 11	1-3.7	VERSIÓN DE EJECUCIÓN	1
		FECHA EJECUCIÓN	10/04/2013
CASO DE USO:	3.7	MODULO DEL SISTEMA	Player

Descripción del caso de prueba:		Salir de pantalla completa				
45. CASO DE PRUEBA						
ooo. Precondiciones						
Video en reproducción.						
ppp. Pasos de la prueba						
Se invoca la función que modifica el área del reproductor.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
-	-	Correcto	Visualización en formato reducido	X		Visualización en formato reducido
qqq. Post condiciones						
Se visualiza el video en formato reducido.						
46. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones				Probador		
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
				Firma: Nombre: Fecha:		

TABLA 92: CASO DE PRUEBA 1-3.7

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA

CASO DE PRUEBA No. 12	1-4.1		VERSIÓN DE EJECUCIÓN		1	
			FECHA EJECUCIÓN		10/04/2013	
CASO DE USO:	4.1		MODULO DEL SISTEMA		Audio	
Descripción del caso de prueba:	Aumentar volumen					
47. CASO DE PRUEBA						
rrr. Precondiciones						
sss. Pasos de la prueba						
Se invoca a la función setVolumenWithKey(int) del plugin de audio.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
Entero	0	Correcto	Aumentar volumen una unidad	X		Aumentar volumen una unidad
Entero	2	Incorrecto	Ignorar petición	X		Ignorar petición
ttt. Post condiciones						
Aumento del volumen.						
48. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones					Veredicto	
Ninguno					<input checked="" type="checkbox"/> Paso	
Observaciones				Probador		
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						

	Firma: Nombre: Fecha:
--	-----------------------------

TABLA 93: CASO DE PRUEBA 1-4.1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 13	1-4.2		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	10/04/2013		
CASO DE USO:	4.2		MODULO DEL SISTEMA	Audio		
Descripción del caso de prueba:	Disminuir volumen					
49. CASO DE PRUEBA						
uuu. Precondiciones						
vvv. Pasos de la prueba						
Se invoca a la función setVolumenWithKey(int) del plugin de audio.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
Entero	0	Correcto	Disminuir volumen una unidad	X		Disminuir volumen una unidad
Entero	2	Incorrecto	Ignorar petición	X		Ignorar petición
www. Post condiciones						
Disminución del volumen.						
50. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
Ninguno						<input checked="" type="checkbox"/> Paso

Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 94: CASO DE PRUEBA 1-4.2

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 14	1-4.3		VERSIÓN DE EJECUCIÓN	1		
			FECHA EJECUCIÓN	10/04/2013		
CASO DE USO:	4.3		MODULO DEL SISTEMA	Audio		
Descripción del caso de prueba:	Silenciar sonido					
51.CASO DE PRUEBA						
xxx. Precondiciones						
Sonido activo						
yyy. Pasos de la prueba						
Se invoca a la función setSystemMute(bool) del plugin de audio.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENA RIO		SI	NO	
Booleano	1	Correcto	Silenciar sonido	X		Silenciar sonido
zzz. Post condiciones						
Sonido silenciado						
52.RESULTADOS DE LA PRUEBA						

Defectos y desviaciones		Veredicto
Ninguno		<input checked="" type="checkbox"/> Paso
Observaciones	Probador	
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>		
	Firma: Nombre: Fecha:	

TABLA 95: CASO DE PRUEBA 1-4.3

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No. 15	1-4.4	VERSIÓN DE EJECUCIÓN		1		
		FECHA EJECUCIÓN		10/04/2013		
CASO DE USO:	4.4	MODULO DEL SISTEMA		Audio		
Descripción del caso de prueba:	Activar sonido					
53.CASO DE PRUEBA						
aaa. Precondiciones						
Sonido silenciado.						
bbb. Pasos de la prueba						
Se invoca a la función setSystemMute(bool) del plugin de audio.						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
Booleano	0	Correcto	Activar sonido	X		Activar sonido

cccc. Post condiciones	
Sonido activo	
54.RESULTADOS DE LA PRUEBA	
Defectos y desviaciones	Veredicto
Ninguno	<input checked="" type="checkbox"/> Paso
Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 96: CASO DE PRUEBA 1-4.4

5.2.2.1 ENCUESTA PARA PRUEBAS DE USABILIDAD

Las pruebas de usabilidad se guiaran por la siguiente estructura de encuesta:

PREGUNTA	CRITERIOS DE EVALUACIÓN
1. ¿Hay términos en idiomas diferentes mezclados?	<p>1 = Se encuentran en todo el sistema</p> <p>2 = Se encuentra en algunas partes del sistema.</p> <p>3 = No se encuentran en ninguna parte del sistema.</p>
2. ¿Es simple el vocabulario utilizado?	<p>1 = El vocabulario es demasiado técnico.</p> <p>2 = El vocabulario presenta algunas dificultades de comprensión.</p>

	<p>3 = El vocabulario es completamente comprensible.</p>
3. ¿Se proporciona tiempo suficiente para realizar las entradas por teclado?	<p>1 = El tiempo es muy limitado.</p> <p>2 = El tiempo es limitado para algunas funcionalidades.</p> <p>3 = El tiempo es completamente suficiente.</p>
4. ¿Hay algún tipo de asistencia para los usuarios que hacen uso del sistema por primera vez?	<p>1 = No existe ninguna ayuda.</p> <p>2 = Se encuentra ayuda en algunas partes.</p> <p>3 = Existen ayudas en todo el sistema.</p>
3. ¿El sistema es fácil de operar para alguien que no recibió capacitación en su operación?	<p>1 = El sistema es de difícil comprensión.</p> <p>2 = El sistema es fácil de operar en algunas de sus funcionalidades.</p> <p>3 = El sistema es completamente fácil de operar.</p>
6. ¿Se entienden la interfaz y su contenido?	<p>1 = No se entiende su interfaz.</p> <p>2 = La interfaz se entiende en algunas partes.</p> <p>3 = La interfaz es completamente entendible.</p>
7. ¿Resulta fácil identificar un objeto o una acción?	<p>1 = Es difícil identificar los objetos o acciones.</p> <p>2 = Se pueden identificar los objetos y acciones en algunas partes del sistema.</p> <p>3 = Todos los objetos y acciones son fácilmente identificables.</p>
8. ¿Resulta fácil entender el resultado de una acción?	<p>1 = Los resultados de las acciones no son entendibles.</p>

	<p>2 = Los resultados de las acciones son entendibles en algunas partes o la mayor parte del sistema.</p> <p>3 = Todos los resultados de las acciones son entendibles.</p>
9. ¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?	<p>1 = La interfaz es difícil de usar.</p> <p>2 = La interfaz es difícil de usar en algunas partes del sistema.</p> <p>3 = La interfaz es completamente sencilla de usar.</p>
10. ¿Son apropiados los mensajes presentado por el sistema?	<p>1 = Los mensajes no son apropiados.</p> <p>2 = Los mensajes son apropiados en algunas partes del sistema.</p> <p>3 = Todos los mensajes son apropiados y fáciles de comprender.</p>
11. ¿Actúa el sistema en la prevención de errores?	<p>1 = El sistema no previene errores del usuario.</p> <p>2 = El sistema previene algunos o la mayoría de los errores del usuario.</p> <p>3 = El sistema previene cualquier error que pueda cometer el usuario.</p>
12. ¿El sistema informa claramente sobre los errores presentados?	<p>1 = El sistema no informa de manera adecuada sobre los errores cometidos.</p> <p>2 = El sistema informa de manera adecuada algunos o la mayoría de los errores cometidos por el usuario.</p> <p>3 = El sistema informa de forma adecuada todos los errores cometidos por el usuario.</p>

13. ¿Se utiliza mensajes y textos descriptivos?	<p>1 = Los mensajes de texto no son descriptivos.</p> <p>2 = La mayoría de los textos son descriptivos o fáciles de interpretar</p> <p>3 = Todos los textos son descriptivos o fáciles de interpretar.</p>
14. ¿Permite una cómoda navegación dentro del producto y una fácil salida de éste?	<p>1 = La navegación no es sencilla.</p> <p>2 = La navegación presenta algunas dificultades.</p> <p>3 = La navegación es sencilla.</p>
15. ¿Se permite al usuario personalizar la interfaz?	<p>1 = La interfaz no es personalizable.</p> <p>2 = La interfaz es personalizable con algunas restricciones.</p> <p>3 = La interfaz es completamente personalizable.</p>
16. ¿Se proporciona información visual de dónde está el usuario, qué está haciendo y qué puede hacer a continuación?	<p>1 = No se presenta ninguna información visual ni otro tipo de ayuda.</p> <p>2 = Presenta ayudas en algunas partes del sistema.</p> <p>3 = Las ayudas son apropiadas y están distribuidas a lo largo del sistema.</p>
17. ¿Se presenta al usuario la información que sólo necesita?	<p>1 = La información presentada es más de la que necesita y tiende a ser confusa.</p> <p>2 = En algunas partes se presenta mayor información a la necesaria.</p> <p>3 = La información es estrictamente la necesaria según el perfil.</p>

18. ¿Se proporciona una experiencia de uso agradable?	<p>1 = La aplicación responde con lentitud a las pulsaciones.</p> <p>2 = Algunas funcionalidades se ejecutan con lentitud.</p> <p>3 = La navegación resulta muy fluida.</p>
---	--

TABLA 97: PRUEBA DE USABILIDAD

5.2.2.2 CASOS DE PRUEBA TÉCNICOS

A continuación se presentan la sucesión de pruebas técnicas realizadas al sistema. Estas pruebas son ejecutadas una vez se disponen de una versión liberada del producto.

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
Tipo de Prueba:	Carga	Código de la prueba	PT-1
Descripción de la prueba:	Carga archivo XML con 20 elementos.		
Versión de Ejecución	1	Fecha de Ejecución	16/05/2013
1. Prerrequisitos de la prueba			
Fichero XML disponible y coherente.			
2. Insumos de la prueba			
<Lista de Insumos necesarios para ejecutar la prueba>			
3. Lista de chequeo de la prueba			
Pasos a Seguir	Prueba satisfactoria		Observaciones
	SI	NO	
Iniciar aplicación	X		
Recorrer listado de videoclips	X		
Reproducir videoclip	X		
4. Resultados de la prueba			

Defectos y desviaciones		Veredicto
<Lista de defectos o desviaciones encontrados por el analista o usuario al ejecutar la prueba>		<input checked="" type="checkbox"/> Paso
Observaciones		Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>		
		Firma: Nombre: Fecha:

TABLA 98: CASO DE PRUEBA TÉCNICO. PT-1

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
Tipo de Prueba:	Volumen	Código de la prueba	PT-2
Descripción de la prueba:	Carga archivo XML con 50 elementos.		
Versión de Ejecución	2	Fecha de Ejecución	25/05/2013
1. Prerrequisitos de la prueba			
Fichero XML disponible y coherente.			
2. Insumos de la prueba			
Fichero XML con 50 videoclips y su información correspondiente.			
3. Lista de chequeo de la prueba			
Pasos a Seguir	Prueba satisfactoria		Observaciones
	SI	NO	
Iniciar aplicación	X		
Recorrer listado de videoclips	X		
Reproducir videoclip	X		
4. Resultados de la prueba			
Defectos y desviaciones		Veredicto	
<Lista de defectos o desviaciones encontrados por el analista		<input checked="" type="checkbox"/> Paso	

o usuario al ejecutar la prueba>	
Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 99: CASO DE PRUEBA TÉCNICO. PT-2

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
Tipo de Prueba:	Carga	Código de la prueba	PT-3
Descripción de la prueba:	Carga videooclips		
Versión de Ejecución	1	Fecha de Ejecución	16/05/2013
1. Prerrequisitos de la prueba			
2. Insumos de la prueba			
Fichero XML con videos de alta calidad y duración superior a 2 minutos.			
3. Lista de chequeo de la prueba			
Pasos a Seguir	Prueba satisfactoria		Observaciones
	SI	NO	
Iniciar aplicación	X		
Recorrer listado de videooclips	X		
Reproducir videooclips	X		
4. Resultados de la prueba			
Defectos y desviaciones			Veredicto
<Lista de defectos o desviaciones encontrados por el analista o usuario al ejecutar la prueba>			<input checked="" type="checkbox"/> Paso

Observaciones	Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>	
	Firma: Nombre: Fecha:

TABLA 100: CASO DE PRUEBA TÉCNICO. PT-3

5.2.2.3 MATRIZ CASOS DE USO VS CASOS DE PRUEBA FUNCIONALES

A continuación se presenta el formato de matriz de trazabilidad que se llevara para asegurar que sean probados todos los aspectos definidos dentro de los casos de uso.

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
1.1	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	1-1.1 2-1.1
	3. Reglas de Negocios	
	<Lista de casos de prueba>	-
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	-
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-1.1 2-1.1

TABLA 101: CASO DE PRUEBA FUNCIONALES. 1.1

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
1.2	1. Datos Entrada	

	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	
	<Lista de casos de prueba>	1-1.2 2-1.2
	3. Reglas de Negocios	
	<Lista de casos de prueba>	-
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-1.2 2-1.2

TABLA 102: CASO DE PRUEBA FUNCIONALES. 1.2

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
2.1	1. Datos Entrada	
	Obligatoriedad	1-2.1
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	
	<Lista de casos de prueba>	1-2.1 2-2.1
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-2.1
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	2-2.1
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
6. Flujo Básico	1-1.2	

TABLA 103: CASO DE PRUEBA FUNCIONALES. 2.1

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
2.2	1. Datos Entrada	
	Obligatoriedad	1-2.2
	Longitud	-
	Tipo de Dato	-

	2. Reglas de Negocios Relacionadas con datos de entrada	
	<Lista de casos de prueba>	1-2.2 2-2.2
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-2.2
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	2-2.2
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-1.2

TABLA 104: CASO DE PRUEBA FUNCIONALES. 2.2

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
2.3	1. Datos Entrada	
	Obligatoriedad	1-2.3
	Longitud	-
	Tipo de Dato	1-2.3
	2. Reglas de Negocios Relacionadas con datos de entrada	
	<Lista de casos de prueba>	1-2.3
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-2.3
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	1-2.3
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-2.3

TABLA 105: CASO DE PRUEBA FUNCIONALES. 2.3

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
2.4	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-

	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-2.4
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
6. Flujo Básico	1-2.4	

TABLA 106: CASO DE PRUEBA FUNCIONALES. 2.4

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
2.5	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-2.5 2-2.5 3-2.5
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	2-2.5
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
6. Flujo Básico	1-2.5 3-2.5	

TABLA 107: CASO DE PRUEBA FUNCIONALES. 2.5

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.1	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	1-3.1 2-3.1 3-3.1
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	1-3.1

		2-3.1 3-3.1
	3. Reglas de Negocios <Lista de casos de prueba>	1-3.1 2-3.1 3-3.1
	4. Flujos de Excepción <Lista de flujos de excepción>	-
	5. Flujos Alternos <Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-3.1 2-3.1 3-3.1

TABLA 108: CASO DE PRUEBA FUNCIONALES. 3.1

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.2	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios <Lista de casos de prueba>	1-3.2 2-3.2
	4. Flujos de Excepción <Lista de flujos de excepción>	-
	5. Flujos Alternos <Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-3.2 2-3.2

TABLA 109: CASO DE PRUEBA FUNCIONALES. 3.2

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.3	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de	-

	entrada	
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-3.3 2-3.3
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	2-3.3
	6. Flujo Básico	1-3.3

TABLA 110: CASO DE PRUEBA FUNCIONALES. 3.3

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.4	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-3.4
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
6. Flujo Básico	1-3.4	

TABLA 111: CASO DE PRUEBA FUNCIONALES. 3.4

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.5	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	

	<Lista de casos de prueba>	1-3.5
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-3.5

TABLA 112: CASO DE PRUEBA FUNCIONALES. 3.5

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.6	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	1-3.6 2-3.6
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-3.6 2-3.6
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-3.6 2-3.6

TABLA 113: CASO DE PRUEBA FUNCIONALES. 3.6

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
3.7	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-3.7
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-

	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-3.7

TABLA 114: CASO DE PRUEBA FUNCIONALES. 3.7

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
4.1	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-4.1
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
6. Flujo Básico	1-4.1	

TABLA 115: CASO DE PRUEBA FUNCIONALES. 4.1

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
4.2	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-4.2
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
6. Flujo Básico	1-4.2	

TABLA 116: CASO DE PRUEBA FUNCIONALES. 4.2

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
4.3	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-4.3
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-4.3

TABLA 117: CASO DE PRUEBA FUNCIONALES. 4.3

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
4.4	1. Datos Entrada	
	Obligatoriedad	-
	Longitud	-
	Tipo de Dato	-
	2. Reglas de Negocios Relacionadas con datos de entrada	-
	<Lista de casos de prueba>	-
	3. Reglas de Negocios	
	<Lista de casos de prueba>	1-4.4
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	-
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	-
	6. Flujo Básico	1-4.4

TABLA 118: CASO DE PRUEBA FUNCIONALES. 4.4

5.2.2.4 MATRIZ REQUERIMIENTOS NO FUNCIONALES VS CASOS DE PRUEBA TÉCNICOS

A continuación se presenta el formato de matriz de trazabilidad que se llevara para asegurar que sean probados todos los aspectos técnicos de la solución; en esta matriz se registrará cada caso de prueba técnico y requerimiento no funcional que será verificado. Esta matriz será diligenciada en la medida que las pruebas técnicas sean diseñadas.

CÓDIGO DE LA PRUEBA TÉCNICA	REQUERIMIENTO NO FUNCIONAL VERIFICADO	OBSERVACIONES
PT-1	RNF1	-
PT-2	RNF2	-
PT-3	RNF4	-

TABLA 119: REQUERIMIENTOS NO FUNCIONALES VS CASOS DE PRUEBA TÉCNICOS

5.2.3 LISTA DE CHEQUEO

A continuación se presenta el formato que se utilizará para lista de chequeo de las pruebas ejecutadas

TIPO DE PRUEBA	Versión de Ejecución	Fecha de Ejecución	EJECUTADA	CUMPLE	NO CUMPLE	Observaciones
Funcional	1-1.1		SÍ	X		
Funcional	2-1.1		SÍ	X		
Funcional	1-1.2		SÍ	X		
Funcional	2-1.2		SÍ	X		
Funcional	1-2.1		SÍ	X		
Funcional	2-2.1		SÍ	X		
Funcional	1-2.2		SÍ	X		
Funcional	2-2.2		SÍ	X		
Funcional	1-2.3		SÍ	X		
Funcional	1-2.4		SÍ	X		
Funcional	1-2.5		SÍ	X		
Funcional	2-2.5		SÍ	X		
Funcional	3-2.5		SÍ	X		
Funcional	1-3.1		SÍ	X		
Funcional	2-3.1		SÍ		X	
Funcional	3-3.1		SÍ	X		
Funcional	1-3.2		SÍ	X		
Funcional	2-3.2		SÍ		X	
Funcional	1-3.3		SÍ	X		
Funcional	2-3.3		SÍ	X		
Funcional	1-3.4		SÍ	X		

Funcional	1-3.5		SÍ	X		
Funcional	1-3.6		SÍ	X		
Funcional	2-3.6		SÍ	X		
Funcional	1-3.7		SÍ	X		
Funcional	1-4.1		SÍ	X		
Funcional	1-4.2		SÍ	X		
Funcional	1-4.3		SÍ	X		
Funcional	1-4.4		SÍ	X		
Técnica	PT-1		SI	X		
Técnica	PT-2		SI	X		
Técnica	PT-3		SI	X		

TABLA 120: LISTA DE CHEQUEO

6 MANUAL DEL PROGRAMADOR

El manual del programador proporciona toda la información necesaria para el desarrollo y la modificación de la aplicación. Se abordarán los siguientes puntos:

- Entorno SDK
- Framework de Samsung
- Módulos de la aplicación

6.1 Entorno de desarrollo Samsung SDK

Samsung utiliza el editor *Eclipse* como **entorno de programación**, el cual incorpora un plugin para importar y crear proyectos basados en la “framework” de Samsung. El entorno se complementa con la instalación de las tres versiones del emulador de *Samsung Smart TV*.

Todo ello, proporciona las herramientas necesarias para poder **desarrollar, depurar y testear** las aplicaciones. -Aunque la inestabilidad del emulador, en ocasiones pueda complicar el proceso de desarrollo.

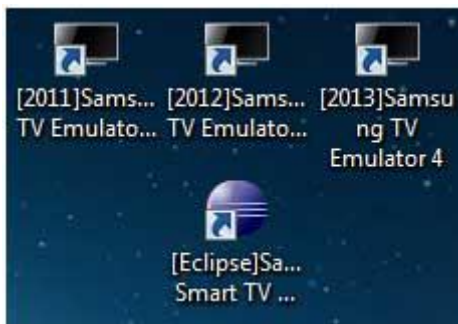


FIGURA 37: HERRAMIENTOS DE PROGRAMACIÓN SAMSUNG SDK

6.1.1 Requisitos hardware

- Procesador: Dual Core 1.5GHz / Single Core 3GHz o superior.
- RAM: 1 GB o superior.
- Sistema operativo (32bit): Windows XP Service Pack 2 o superior, Windows 7
- Resolución de pantalla: 1280 x 1024 o superior.
- Disco duro: 5GB o más

6.1.2 Requisitos software

- .NET Framework 2.0
- Visual C++ Redistributable Package 2005, 2010
- DirectX End-User Runtime
- DirectX End-User Runtime Web Installer Download URL
- <http://www.microsoft.com/download/en/details.aspx?id=35>
- Java Standard Edition(Java SE) 1.6 or higher
- Install Java Runtime Environment (JRE) :
- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

NOTA: Es fundamental instalar previamente *Microsoft Visual C++ 2010 Redistributable Package* en su versión x86. La versión de 64 bits no sirve puesto que la aplicación está implementada para 32 bits.

6.2 Framework de Samsung

6.2.1 Introducción

Este *framework* se utiliza para crear aplicaciones basadas en web para Samsung Smart TV. Esto permite a los desarrolladores concentrarse en la lógica de negocios y servicios. La **framework** proporciona **APIs** envoltorio de alto nivel que simplifican el desarrollo de aplicaciones. Entre otros, se pueden encontrar:

- Conmutación entre diferentes escenas de entrada
- Mostrar y ocultar cada DIVs
- Los controladores de eventos utilizando la funcionalidad DTV-especificado

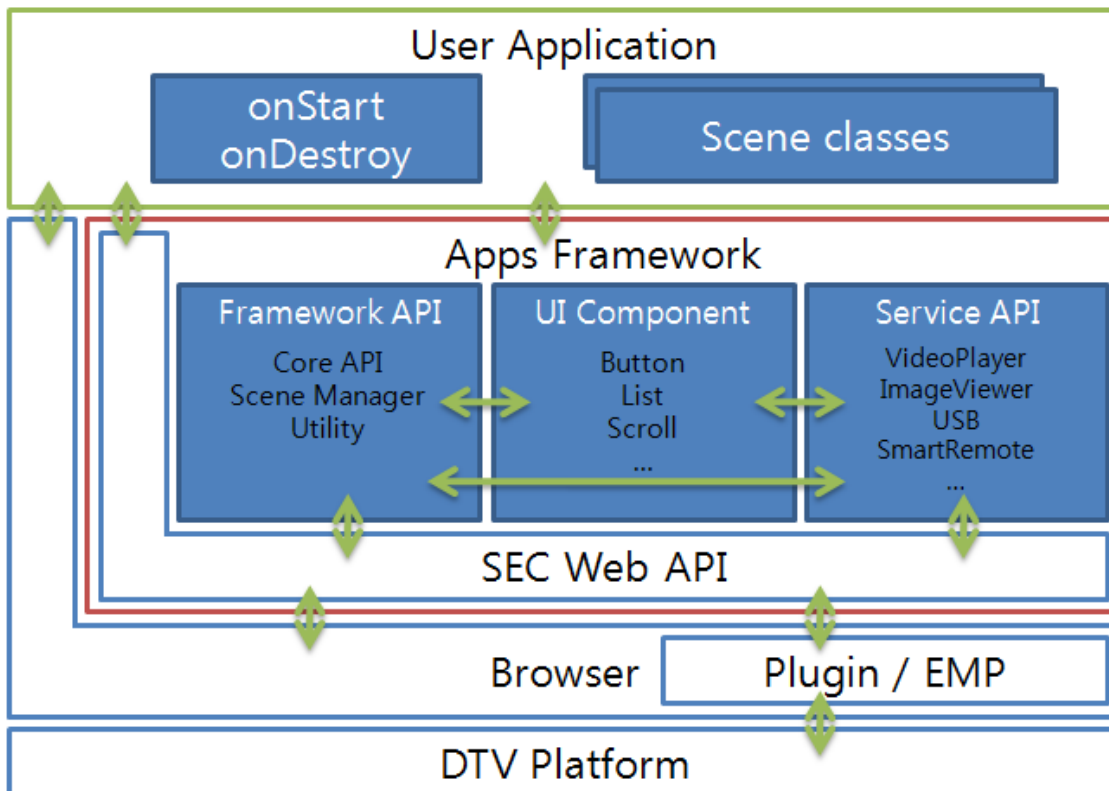


FIGURA 38: FRAMEWORK SAMSUNG

6.2.2 Composición de una escena

El *framework* de Samsung utiliza escenas para la **creación** de las diferentes **pantallas**. Siendo cada pantalla una combinación de un fichero *html*, un fichero *css*, y un fichero *javascript*. Todos ellos soportan tanto el diseño como la funcionalidad, la última proporcionada por el javascript.

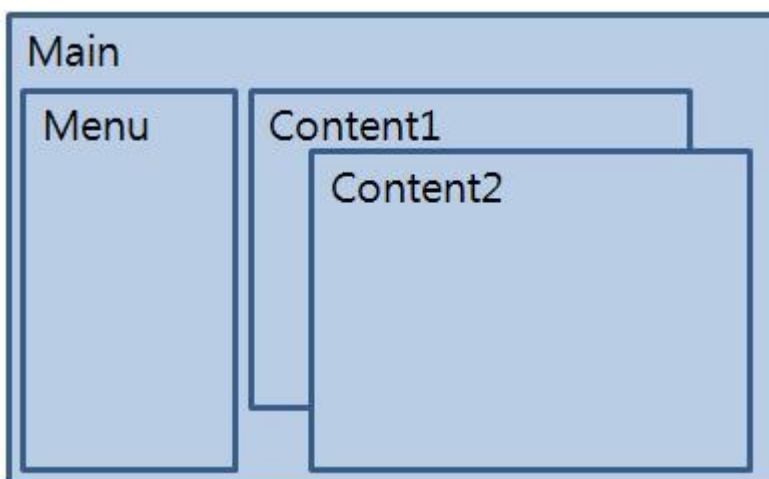


FIGURA 39: COMPOSICIÓN DE ESCENA

6.2.3 Estructura de la aplicación

La estructura mínima de una aplicación consta de los siguientes directorios:

- **App**
 - Html: contiene los ficheros html.
 - Scenes. proporciona la funcionalidad de la de cada escena mediante código javascript.
 - Stylesheets. alberga los estilos CSS correspondientes a cada escena.
- **Icon**
- **Images.** Las imágenes usadas por la aplicación.
- **Layout.** Almacena un fichero por cada escena. En cada uno permite modificar la disposición visual de los elementos que componen la escena. De forma automática se inserta el código los tres directorios que componen la carpeta *app* (html, scenes y stylesheets).

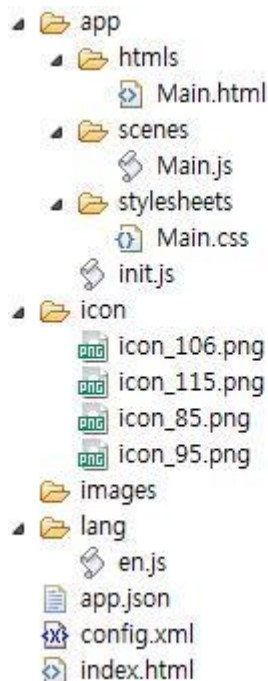


FIGURA 40: ESTRUCTURA DE DIRECTORIOS PROYECTO

Index.html

El directorio raíz contiene los archivos index.html. Sirve como punto de acceso de la aplicación.

config.xml

Un archivo XML en la raíz de la estructura de la aplicación que contiene información sobre la configuración de la aplicación, ejecución de la aplicación, las actualizaciones, la configuración del entorno de funcionamiento, etc. Dependiendo de la información, el gestor de aplicaciones controla la versión de la aplicación, establece el entorno en el que se ejecuta la aplicación, y crea y gestiona las cuentas de usuario. El archivo config.xml debe estar ubicado en el directorio en el que está instalada la aplicación, y deberá contener las etiquetas que figuran en la siguiente tabla.

En el inicio de la aplicación, el sistema operativo invoca a un *loader* para realizar el proceso de carga de los distintos componentes segmentados de la aplicación: *html*, estilo (CSS) y componentes funcionales (javascript). Este es un gráfico de la secuencia:

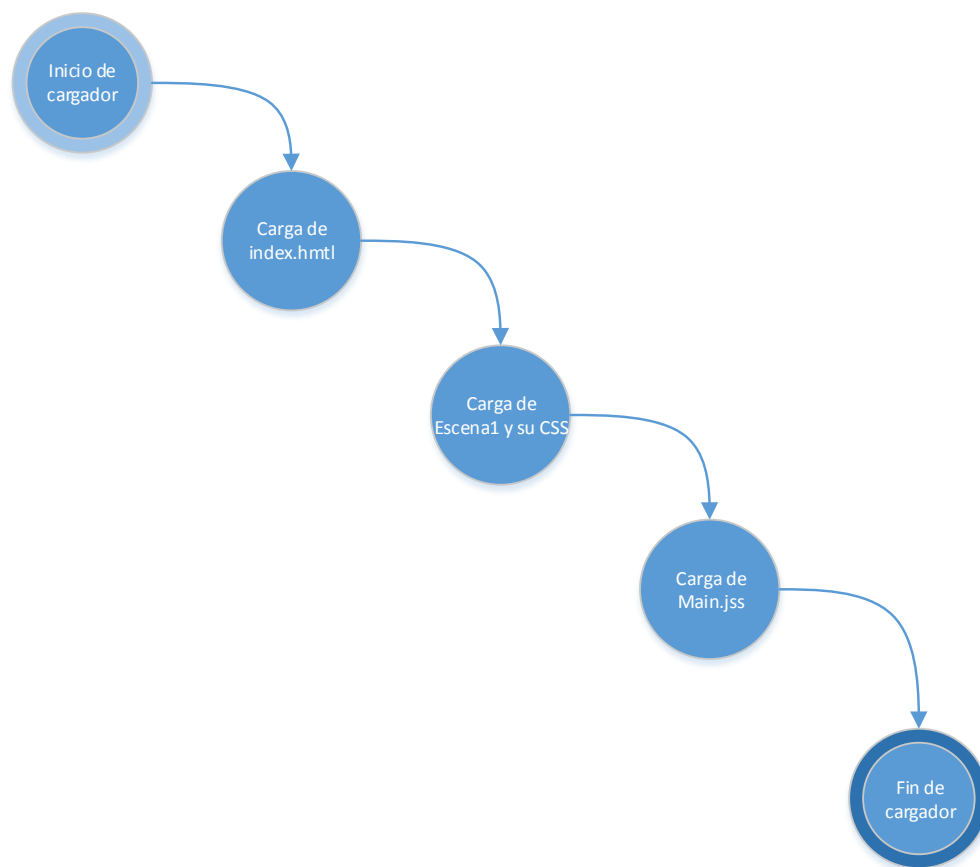


FIGURA 41: CARGADOR DE SAMSUNG

6.3 Módulos de la aplicación

En este apartado se detallan los diferentes archivos y módulos que componen la aplicación desarrollada.

Se recomienda la lectura del apartado 4 de esta memoria. En él se encontrará información sobre la estructura modular, funciones, y demás información de interés.

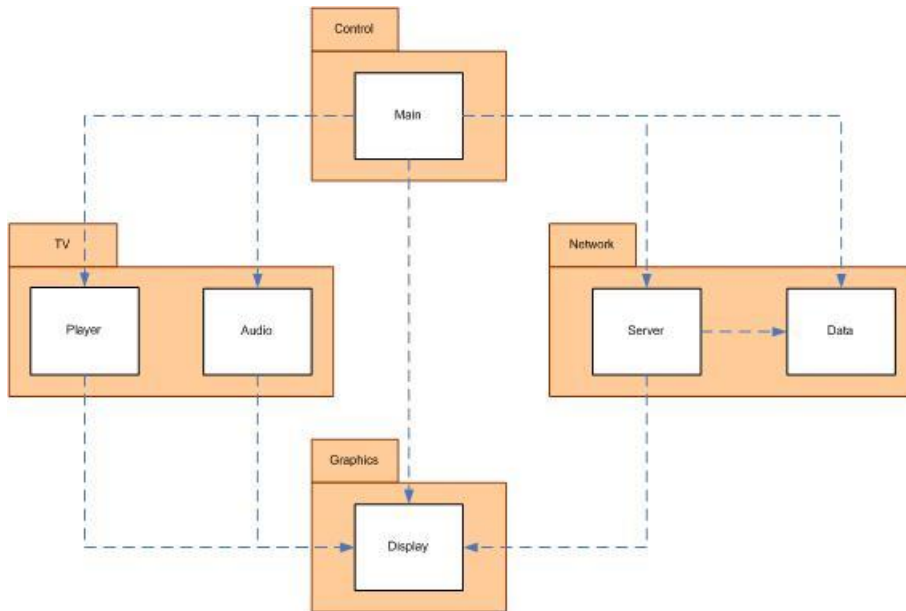


FIGURA 42: MÓDULOS APLICACIÓN

La estructura de la aplicación dispone un módulo controlador, que invoca los diferentes componentes.

A continuación se muestra el fichero config.xml correspondiente al proyecto:

▲ [e] widget	
[e] previewjs	Flumotion
[e] preicon	
[e] cpname	
[e] cplogo	
[e] cpauthjs	
[e] ver	
[e] mgrver	
[e] fullwidget	y
[e] srcctl	n
[e] ticker	n
[e] childlock	n
[e] audiomute	n
[e] videomute	n
[e] dcont	y
[e] type	user
[e] widgetname	Flumotion: Radio Top
[e] description	
[e] width	960
[e] height	540
▲ [e] author	
[e] name	Rubén Menes Hevia
[e] email	
[e] link	
[e] organization	Universidad de Oviedo

FIGURA 43: FICHERO “CONFIG.XML”

Index.html

Por una parte tenemos el *index.html*, el cual contiene el <head> de la página en el que se disponen todos los scripts y diferentes cargadores de la framework de Samsung.

Debe utilizar un módulo común, una biblioteca que contiene las funciones esenciales prestados por el Gestor de aplicaciones. Para ello se agrega el componente *widget.js*. A su vez el *loader.js*, mencionado en el subapartado anterior, es el responsable de inicializar todos los componentes necesarios para la correcta ejecución de una aplicación con escenas,

```
<script type="text/javascript" language="javascript"
src="$MANAGER_WIDGET/Common/af/2.0.0/loader.js"></script>
<script type='text/javascript' language='javascript'
src='Common/API/Widget.js'></script>
<script type='text/javascript' language='javascript'
src='Common/API/TVKeyValue.js'></script>
```

A continuación se cargan los diferentes **módulos** de la aplicación: **main**, **server**, **data**, **player** y **display**.

```
<script language="javascript" type="text/javascript"
src="Javascript/Main.js"></script>
<script language="javascript" type="text/javascript"
src="Javascript/Server.js"></script>
<script language="javascript" type="text/javascript"
src="Javascript/Data.js"></script>
<script language="javascript" type="text/javascript"
src="Javascript/Player.js"></script>
<script language="javascript" type="text/javascript"
src="Javascript/Display.js"></script>
<script language="javascript" type="text/javascript"
src="Javascript/Audio.js"></script>
```

Con las etiquetas **object**, se cargan los tres plugins que utiliza la aplicación: **audio**, **video** y **tmw** (para mostrar propiedades de usuario).

```
<object id="pluginPlayer" border=0 classid="clsid:SAMSUNG-INFOLINK-
PLAYER"></object>
<object id="pluginAudio" border=0 classid="clsid:SAMSUNG-INFOLINK-
AUDIO"></object>
<object id="pluginTVMW" border=0 classid="clsid:SAMSUNG-INFOLINK-
TVMW"></object>
```

Scene1.html

El fichero **scene1.html** contiene la escena de la aplicación. La aplicación podría mostrar diferentes escenas en sucesivas ampliaciones.

Scene1.css

El fichero **scene1.css** contiene la definición de los estilos *html*. Todos los estilos se encuentran asociados a la escena 1.

En la siguiente imagen se indican, a modo orientativo, las posiciones de algunas de las secciones del div.

- logo
- leftHalf
 - videoList
- rightHalf
 - Description
- navi

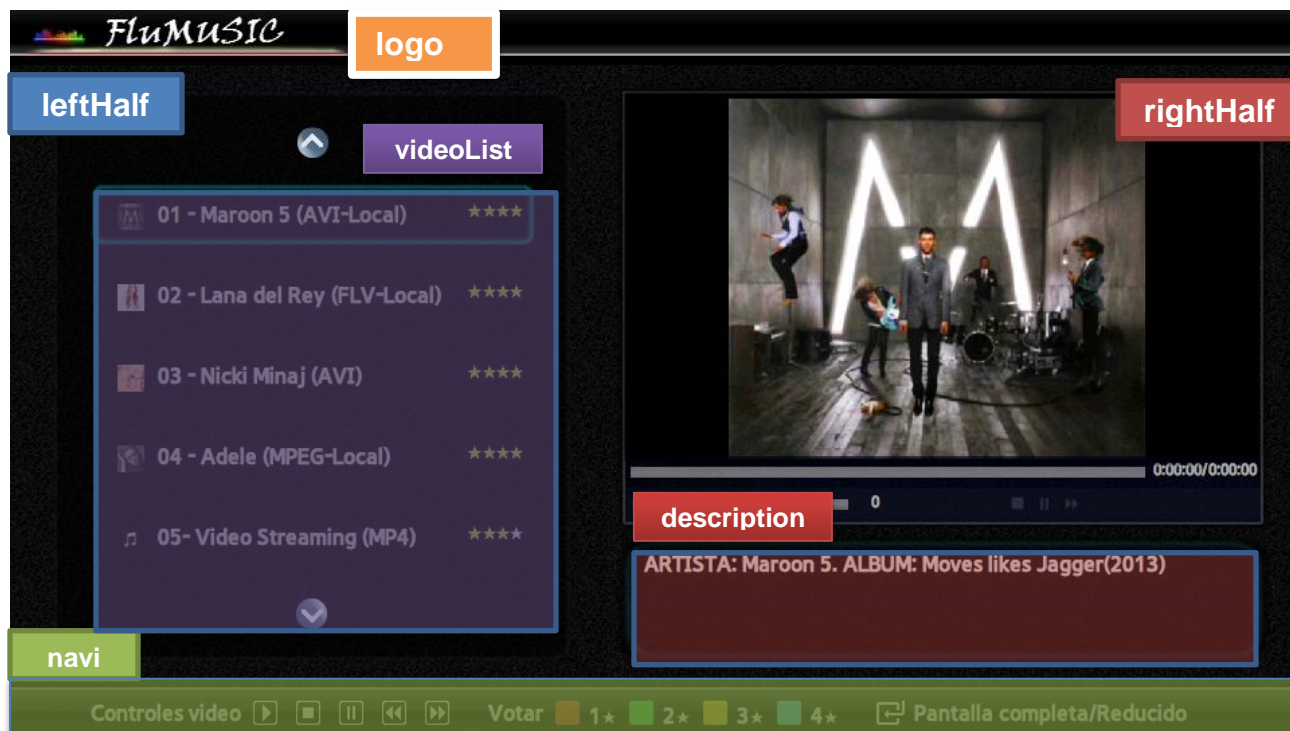


FIGURA 44: ESTRUCTURA PANTALLA APLICACIÓN

Main.jss

Este módulo actúa de controlador (véase apartado 4.2) detectando las pulsaciones mediante eventos, para posteriormente invocar las funciones necesarias de los diferentes módulos.

Para que la aplicación responda a las pulsaciones del teclado, se debe invocar el método `keyDown()`. En este método se recibe un evento con el código de la tecla presionada.

```

Main.keyDown = function()
{
    var keyCode = event.keyCode;
    alert("Key pressed: " + keyCode);

    switch(keyCode)
    {
        case tvKey.KEY_RETURN:
        case tvKey.KEY_PANEL_RETURN:
            alert("RETURN");
            Player.stopVideo();
            widgetAPI.sendReturnEvent();
            break;
    }
}

```

(...)

El módulo define también dos métodos necesarios en el framework de Samsung: `onLoad()`, y `onUnload()`. Su función iniciar el controlador y los componentes necesarios así como de salvar el contexto del sistema operativo. Este será restablecido al cierra de la aplicación, cuando resultará invocada el método `onUnload`.

Player.jss

Este módulo captura el plugin incorporado como `<object>` en el `index.html`, para posteriormente ejecutar las funciones de su API.

```
this.plugin = document.getElementById("pluginPlayer");
```

También define en su método de inicio los manejadores de eventos.

```
this.plugin.OnCurrentPlayTime = 'Player.setCurTime';
this.plugin.OnStreamInfoReady = 'Player.setTotalTime';
this.plugin.OnBufferingStart = 'Player.onBufferingStart';
this.plugin.OnBufferingProgress = 'Player.onBufferingProgress';
this.plugin.OnBufferingComplete = 'Player.onBufferingComplete';
```

El plugin de vídeo permite el **streaming adaptativo** (capacidad conmutación entre diferentes calidades de vídeo en función del estado de la red), por lo que acepta paso de parámetros para definir requisitos de la reproducción.

Los parámetros se pasan en la misma **url**, concatenados con la **barra vertical** "|". La sintaxis sería al siguiente:

URL syntax CONTENT_URL | BITRATES | UPTIMER | STARTBITRATE | COMPONENT

Parámetro	Descripción	Atributo
CONTENT_URL	La URL de tipo HTTP OIPF adaptativa de streaming adaptativo.	Obligatorio
BITRATES	Especifica el bitrate del video con diferentes modalidades.	Opcional
	Sintaxis BITRATES= BRGROUP[GSEP BRGROUP] BRGROUP = GSTART BITRATELIST GEND BITRATELIST= xx: yy: zz or xx~zz	

Parámetro		Descripción		Atributo
		<pre>GSTART : '(' //Inicio de grupo GEND : ')' //Fin de grupo GSEP : NORMSWITCH DONLYSWITCH NORMSWITCH : ':' //Conmutación normal DONLYSWITCH : '>' //Conmutación solo a menor bitrate.</pre> <ul style="list-style-type: none"> • Si no se especifica velocidades, entonces todo los bitrates descritos en MPD se utilizan. • Se permite siempre conmutar dentro del mismo grupo de bitrates. • Con el modo DONLYSWITCH sólo se permite la conmutación a menor bitrate. 		
UPTIMER	Especifica el umbral del tiempo de conmutación inicial UP que se aplica a la conmutación entre las corrientes en diferentes grupos de bitrates.			Opcional
	sintaxis	<pre>UPTIMER=xx</pre> <p>UPTIMER no tiene sentido si hay un grupo de bits. UPTIMER se aumenta dinámicamente si hay inestabilidad en el ancho de banda de la red. UPTIMER no se aplica sobre el cambio dentro del mismo grupo de bitrate. Si no se especifica UPTIMER, a continuación, se aplica UPTIMER = 0 por defecto.</p>		
STARTBITRATE	Especifica el bitrate inicial			Opcional
	syntax	<pre>STARTBITRATE=100000</pre>		
ADMODE	Specifies the AD force play mode when a seek (jump) operation is requested.			Opcional
	syntax	<pre>ADMODE=FIRST LAST</pre>		
	FIRST	<p>Seeks the first AD among ADs between the current position and the target jump position. Then force-plays the first AD and keeps the main content play from the AD insertion time.</p>		

Parámetro		Descripción		Atributo
		LAST (default)	Seeks the last AD among ADs between the current position and the target jump position then force-plays the last AD and starts playing from the target seek position.	
	COMPONENT	Especifica el tipo de streaming adaptativo: estándar o definido por Apple.		Obligatorio
		syntax	COMPONENT=HAS o COMPONENT =HLS	
		HAS	OIPF Type (HTTP Adaptive Streaming)	
		HLS	Apple Type (HTTP Live Streaming)	

TABLA 121: PARÁMETROS STREAMING ADAPTATIVO

Audio.jss

Encargado de modificar el volumen de la aplicación. Para ello utiliza los métodos definidos por el plugin. Al igual que con el módulo *player.jss*, al integrarse en la aplicación como *object*, el plugin debe capturarse de la siguiente forma para ser utilizado.

```
this.plugin = document.getElementById("pluginAudio");
```

Server.jss

Define el inicio se declara una variable que contendrá la *url* de la que se cargará el fichero XML del que se recibe el listado de videos. En la fase de producción, este XML se encuentra en un directorio propio de la aplicación. Sin embargo, el módulo está diseñado para obtener el XML de un servidor de internet, tanto de un Web Service como cualquier otro tipo de servicio que exporte un fichero XML.

```
url : "XML/videoList.xml"
```

Para *parsear* el fichero XML se obtiene de internet mediante una función para luego utilizar la tecnología *AJAX* y poder obtener los diferentes elementos.

```
this.XHRObj = new XMLHttpRequest();
```

Se declaran las variables de los campos a extraer y se obtiene el campo "item" del fichero XML.

```
var items = xmlElement.getElementsByTagName("item");

var videoNames = [ ];
var videoURLs = [ ];
var videoDescriptions = [ ];
var videoCovers = [ ];
var videoLogos = [ ];
var videoType = [ ];
```

Con esta sentencia se obtiene un campo del XML:

```
var titleElement = items[index].getElementsByTagName("title")[0];
```

Con esta se lee el primer campo:

```
videoNames[index] = titleElement.firstChild.data;
```

Finalmente se almacena invocando a una función del módulo *Data*:

```
Data.setVideoNames(videoNames);
```

Display.jss

Responsable de modificar todos los elementos visuales dinámicos, tales como el selector de los vídeos, la barra de progreso de la reproducción y el visor de imágenes.

setTime

En la implementación se dispone de una función que modifica el contador de la reproducción, una vez recibido la información del plugin de vídeo. Esta se actualiza en el transcurso de la reproducción. Si el vídeo es en directo, no se visualiza el tiempo de duración, pues este no resulta fiable.

```
Display.setTime = function(time)
{
    var timePercent = (100 * time) / this.totalTime;
    var timeElement = document.getElementById("timeInfo");
    var timeHTML = "";
    var timeHour = 0; var timeMinute = 0; var timeSecond = 0;
    var totalTimeHour = 0; var totalTimeMinute = 0; var totalTimeSecond =
0;
```



```
document.getElementById("progressBar").style.width = timePercent +
"%";

if(Player.state == Player.PLAYING)
{
    totalTimeHour = Math.floor(this.totalTime/360000);
    timeHour = Math.floor(time/360000);

    totalTimeMinute = Math.floor((this.totalTime%360000)/60000);
    timeMinute = Math.floor((time%360000)/60000);

    totalTimeSecond = Math.floor((this.totalTime%60000)/1000);
    timeSecond = Math.floor((time%60000)/1000);

    timeHTML = timeHour + ":";

    if(timeMinute == 0)
        timeHTML += "00:";
    else if(timeMinute <10)
        timeHTML += "0" + timeMinute + ":";
    else
        timeHTML += timeMinute + ":";

    if(timeSecond == 0)
        timeHTML += "00/";
    else if(timeSecond <10)
        timeHTML += "0" + timeSecond;
    else
        timeHTML += timeSecond;

    if (this.videoType!="live"){
        timeHTML += "/" + totalTimeHour + ":";

        if(totalTimeMinute == 0)
            timeHTML += "00:";
        else if(totalTimeMinute <10)
            timeHTML += "0" + totalTimeMinute + ":";
        else
            timeHTML += totalTimeMinute + ":";

        if(totalTimeSecond == 0)
            timeHTML += "00";
        else if(totalTimeSecond <10)
            timeHTML += "0" + totalTimeSecond;
        else
            timeHTML += totalTimeSecond;
    }
}
else
    timeHTML = "0:00:00/0:00:00";

widgetAPI.putInnerHTML(timeElement, timeHTML);
}
```

setVideoListPosition

Otra parte del grueso de la implementación de este módulo, la ocupa la función *setVideoListPosition*, a la que se le pasan dos parámetros. El listado con los nombres de los diferentes videoclips, y la *url* de los logos que los acompañan.

```
Display.setVideoListPosition = function(position, move)
```

Esta función tiene por cometido desplazar los elementos del listado de videoclips, como resultado de la navegación del menú.

Para ello utiliza dos variables como límites laterales de la lista de elementos:

```
this.FIRSTIDX;  
this.LASTIDX;
```

Las dos primeras sentencias actualizan la numeración de elementos del extremo superior derecho del menú.

```
listHTML = (position + 1) + " / " + Data.getVideoCount();  
widgetAPI.putInnerHTML(document.getElementById("videoCount"),  
listHTML);
```

El siguiente paso coloca el selector en la posición indicada como parámetro, si el número de videos no supera el límite de 5. En ese caso no habría desplazamiento de los elementos.

```
if(Data.getVideoCount() < 5)  
{  
    for (var i = 0; i < Data.getVideoCount(); i++)  
    {  
        if(i == position)  
            this.videoList[i].style.backgroundImage=  
"url (Images/listBox/selector.png)";  
        else  
            this.videoList[i].style.backgroundImage= "url (none)";  
    }  
}
```

Esta condición evalúa si el selector está en el extremo de la lista o no, debido a que el movimiento puede requerir unciamente el desplazamiento del selector o de todos los elementos de la lista.

```
else if((this.currentWindow!=this.LASTIDX && move==Main.DOWN) ||  
(this.currentWindow!=this.FIRSTIDX && move==Main.UP))
```

coverLaunch

También dispone de un método para lanzar un visor de imágenes que se superpone al reproductor de vídeo.

```
Display.coverLaunch = function(url)
{
    sf.service.ImageViewer.draw({
        url: url,    // image URL
        width: '1600', // image width
        height: '1067' // image height
    });
}
```

showVoteSend

Muestra un popup indicando que el voto se ha enviado correctamente.

```
Display.showVoteSend = function()
{
    $('#vecPopup_ok_cancel_5i7n3x04jjazr').sfPopup({ text: "Tu voto ha
    sido enviado", buttons: "OK"});
    $('#vecPopup_ok_cancel_5i7n3x04jjazr').sfPopup('show');
    $('#vecPopup_ok_cancel_5i7n3x04jjazr').sfPopup('focus');
}
```

showVoteError

Muestra un popup indicando que se ha producido un error en la transmisión del voto emitido.

```
Display.showVoteSend = function()
{
    $('#vecPopup_ok_cancel_5i7n3x04jjazr').sfPopup({ text: "Error en el
    envío", buttons: "OK"});
    $('#vecPopup_ok_cancel_5i7n3x04jjazr').sfPopup('show');
    $('#vecPopup_ok_cancel_5i7n3x04jjazr').sfPopup('focus');
}
```

showScore

Inserta una imagen de entre cuatro distinta, la cuales son seleccionadas en función de la puntuación recibida como parámetro. Su propósito es devolver un array con la imagen que se añadirá a la variable *html* que contiene todo el contenido a mostrar en la *ListBox* (lista que contiene los videoclips). Dependiendo de la puntuación se mostrará una imagen con más o menos estrellas coloreadas.

```
Display.showScore = function(score)
{
    switch(score) {
        case "0":
            html = '';
            break;
        case "1":
            html = '';
            break;
        case "2":
            html = '';
            break;
        case "3":
            html = '';
            break;
        case "4":
            html = '';
            break;
        default:
            html = '';
            break;
    }
return html;
}
```

6.4 Sistema de votación de videoclips

Para acompañar la implementación del componente de votación de los videoclips, y poder verificar su correcto funcionamiento, se puso en marcha un servidor local con una base de datos MySQL, con el propósito de recibir los votos emitidos por el televisor y proceder a su registro.

6.4.1 Especificación del sistema

Las votaciones se envían desde el televisor mediante la tecnología **AJAX** (explicada en apartados anteriores), la cual envía información al servidor que se haya asignado en la aplicación. El servidor recibirá la petición **POST** y ejecutará un script en PHP. Realizará la consulta **SQL** para registrar el voto recibido y finalizará retornando un mensaje de vuelta al televisor indicando el estado el éxito de la operación.

En la siguiente imagen se puede observar el modelo de funcionamiento del componente:

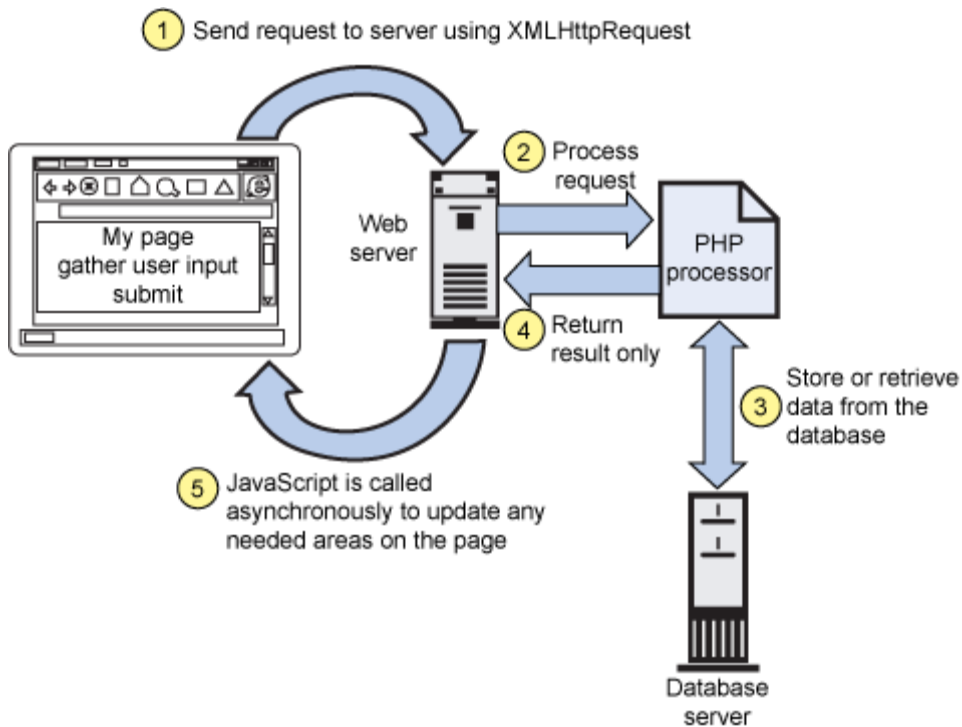


FIGURA 45: ENTORNO DE VOTACIÓN: ESPECIFICACIÓN DEL SISTEMA

6.4.2 Implementación

Como se ha mencionado, el servidor ejecuta un script implementado en lenguaje PHP una vez recibe la petición POST por parte del televisor.

El script captura la información enviada, la inserta en la base de datos y cierra la conexión. La implementación tiene el siguiente código:

```
<?php
// if data are received via POST, with index of 'test'
if (isset($_POST['versionList']) && isset($_POST['name']) &&
isset($_POST['vote']) && isset($_POST['mac'])) {
    $versionList = $_POST['versionList']; // get version
    $name = $_POST['name']; // get name
    $vote = $_POST['vote']; // get vote
    $mac = $_POST['mac']; // get mac

    #Conectamos con MySQL
    $conexion = mysql_connect("localhost","root","")
    or die ("Fallo en el establecimiento de la conexión");

    #Seleccionamos la base de datos a utilizar
    mysql_select_db("votacion")
    or die("Error en la selección de la base de datos");

    $sql="insert into votacion(versionList, mac, tema, voto) values
('$versionList','$mac', '$name', '$vote)";
    $resultado=mysql_query($sql);

    #Cerramos la conexión con la base de datos
    mysql_close($conexion);
}
?>
```

6.6 Anexo

6.6.1 Index.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/div/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Flumotion: Lista de Éxitos</title>

    <!-- Common widget API -->
    <script type="text/javascript" language="javascript" src="$MANAGER_WIDGET/Common/af/2.0.0/Loader.js"></script>
    <script type='text/javascript' language='javascript' src='Common/API/Widget.js'></script>
    <script type='text/javascript' language='javascript' src='Common/API/TVKeyValue.js'></script>

    <!-- Widget code -->
    <script language="javascript" type="text/javascript" src="Javascript/Main.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Server.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Data.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Player.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Display.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Audio.js"></script>
    <script language="javascript" type="text/javascript" src="Javascript/Bitrate.js"></script>

    <!-- Style sheets -->
    <link rel="stylesheet" href="CSS/Main.css" type="text/css">

    <!-- Plugins -->
    <object id="pluginPlayer" border=0 classid="clsid:SAMSUNG-INFOLINK-PLAYER"></object>
    <object id="pluginAudio" border=0 classid="clsid:SAMSUNG-INFOLINK-AUDIO"></object>
    <object id="pluginTV" border=0 classid="clsid:SAMSUNG-INFOLINK-TV"></object>

```

6.6.2 Scene1.html

```

<div id="main">
  <div id="Logo">Flumotion: Lista de Éxitos</div>
  <div id="rightHalfBackOne"></div>
  <div id="rightHalfBackTwo"></div>
  <div id="rightHalfBackThree"></div>
  <div id="rightHalfBackFour"></div>
  <div id="rightHalf">
    <div id="videoBox_top"></div>
    <div id="videoBox_left"></div>
    <div id="videoBox_right"></div>
    <div id="videoBox_bottom">
      <div id="time">
        <div id="progressBarBG">
          <div id="progressBar"></div>
        </div>
        <div id="timeInfo"></div>
      </div>
      <div id="volume">
        <div id="volumeIcon"></div>
        <div id="volumeBarBG">
          <div id="volumeBar"></div>
        </div>
        <div id="volumeInfo"></div>
      </div>
      <div id="videoControl">
        <div id="rewind"></div>
        <div id="play"></div>
        <div id="stop"></div>
        <div id="pause"></div>
        <div id="forward"></div>
      </div>
      <div id="status"></div>
    </div>
    <div id="description_top"></div>
    <div id="description_bottom"></div>
    <div id="description"></div>
  </div>
  <div id="LeftHalfback"></div>
  <div id="LeftHalf">
    <div id="videoList" class="style_videoList">
      <div id="video0"></div>
      <div id="video1"></div>
      <div id="video2"></div>
      <div id="video3"></div>
      <div id="video4"></div>
      <div id="videoCount"></div>
    </div>
    <div id="previous"></div>
    <div id="next"></div>
  </div>
  <div id="navi">
    <div id="help_navi">
      </img>
      </img>
      </img>
      </img>
      </img> &nbsp;<a class="style_navi">Controles video<
      </img> &nbsp;<a class="style_navi">Pantalla completa/<
      </img> &nbsp;<a class="style_navi">Regresar</a>
    </div>
  </div>
</div>

```


6.6.3 Scene1.css

```
#video2
{
    position: absolute;
    left: 30px; top: 190px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

#video3
{
    position: absolute;
    left: 30px; top: 250px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

#video4
{
    position: absolute;
    left: 30px; top: 310px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

#videoCount
{
    position: absolute;
    left: 330px; top: 30px;
    width: 30px; height: 20px;

    text-align: left;
    font-size: 12px;
}

#previous
{
    position: absolute;
    left: 210px; top: 80px;
    width: 30px; height: 30px;
    background-image: url('../Images/ListBox/previous.png');
    opacity: 0.2;
}

#next
{
    position: absolute;
    left: 210px; top: 430px;
    width: 30px; height: 30px;
    background-image: url('../Images/ListBox/next.png');
    opacity: 0.2;
}
```

```
/* Left Half */
#leftHalfback
{
    position: absolute;
    left: 0px; top: 37px;
    width: 420px; height: 480px;
    background: url('../images/background_texture.png') repeat;
}

#leftHalf
{
    position: absolute;
    left: 0px; top: 0px;
    width: 420px; height: 540px;
}

#videoList
{
    position: absolute;
    left: 30px; top: 55px;
    width: 390px; height: 430px;
    background-image: url('../Images/listBox/listBox.png');
    opacity: 0.6;
}

#video0
{
    position: absolute;
    left: 30px; top: 70px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}

.logo
{
    position: relative;
    left: 0px; top: 5px;
    width: 20px; height: 20px;

    padding-right: 10px;
    padding-top: 0px;
}

#video1
{
    position: absolute;
    left: 30px; top: 130px;
    width: 330px; height: 47px;

    background-repeat: no-repeat;
    padding-left: 20px;
    padding-top: 10px;
}
```

```
#pause
{
  position: absolute;
  left: 60px; top: 0px;
  width: 20px; height: 20px;
  background-image: url('../..//images/control/kpause.png');
  opacity: 0.2;
}

#forward
{
  position: absolute;
  left: 80px; top: 0px;
  width: 20px; height: 20px;
  background-image: url('../..//images/control/kfwd.png');
  opacity: 0.2;
}

#status
{
  position: absolute;
  left: 390px; top: 28px;
  width: 98px; height: 16px;

  text-align: left;
  font-size: 12px;
}

#description_top
{
  position: absolute;
  left: 30px; top: 390px;
  width: 488px; height: 50px;
  background-image: url('../..//images/descriptionBox/description_top.png');
  opacity: 0.6;
}

#description_bottom
{
  position: absolute;
  left: 30px; top: 440px;
  width: 488px; height: 50px;
  background-image: url('../..//images/descriptionBox/description_bottom.png');
  opacity: 0.6;
}

#description
{
  position: absolute;
  left: 52px; top: 405px;
  width: 458px; height: 80px;

  text-align: left;
  font-size: 16px;
  overflow: hidden;
  text-overflow : ellipsis;
}
```

```
#volumeBarBG
{
    position: absolute;
    left: 22px; top: 8px;
    width: 144px; height: 8px;
    background-image: url('../..//images/videoBox/volumeBarBG.png');
}

#volumeBar
{
    position: absolute;
    left: 0px; top: 0px;
    width: 0%; height: 8px;
    background-image: url('images/videoBox/volumeBar.png');
}

#volumeInfo
{
    position: absolute;
    left: 168px; top: 4px;
    width: 32px; height: 16px;

    text-align: left;
    font-size: 12px;
}

#videoControl
{
    position: absolute;
    left: 250px; top: 25px;
    width: 100px; height: 20px;
}

#rewind
{
    position: absolute;
    left: 0px; top: 0px;
    width: 20px; height: 20px;
    background-image: url('images/control/krew.png');
    opacity: 0.2;
}

#play
{
    position: absolute;
    left: 20px; top: 0px;
    width: 20px; height: 20px;
    background-image: url('images/control/kplay.png');
    opacity: 1.0;
}

#stop
{
    position: absolute;
    left: 40px; top: 0px;
    width: 20px; height: 20px;
    background-image: url('../..//images/control/kstop.png');
    opacity: 0.2;
}
```

```
#videoBox_bottom
{
    position: absolute;
    left: 30px; top: 333px;
    width: 488px; height: 56px;
    background-image: url('../..//images/videoBox/bottom.png');
    z-index: 100;
    opacity: 0.7;
}

#time
{
    position: absolute;
    left: 8px; top: 0px;
    width: 472px; height: 24px;
}

#progressBarBG
{
    position: absolute;
    left: 4px; top: 8px;
    width: 383px; height: 8px;
    background-image: url('../..//images/videoBox/progressBarBG.png');
}

#progressBar
{
    position: absolute;
    left: 0px; top: 0px;
    width: 0%; height: 8px;
    background-image: url('../..//images/videoBox/progressBar.png');
}

#timeInfo
{
    position: absolute;
    left: 393px; top: 4px;
    width: 60px; height: 16px;

    text-align: center;
    font-size: 12px;
}

#volume
{
    position: absolute;
    left: 8px; top: 24px;
    width: 472px; height: 24px;
}

#volumeIcon
{
    position: absolute;
    left: 4px; top: 4px;
    width: 17px; height: 16px;
    background-image: url('../..//images/videoBox/volume.png');
}
```

```
#rightHalfBackThree
{
    position: absolute;
    left:930px; top: 37px;
    width: 34px; height: 478px;
    background: url('../..//images/background_texture.png') repeat;
}

#rightHalfBackFour
{
    position: absolute;
    left:445px; top: 333px;
    width: 488px; height: 180px;
    background: url('../..//images/background_texture.png') repeat;
}

#rightHalf
{
    position: absolute;
    left:420px; top: -5px;
    width: 540px; height: 540px;
    z-index: 100;
}

#videoBox_top
{
    position:absolute;
    left:30px; top:55px;
    width:488px; height:8px;
    background-image:url('../..//images/videoBox/top.png');
    z-index: 100;
    opacity: 0.6;
}

#videoBox_left
{
    position:absolute;
    left:30px; top:55px;
    width:8px; height:278px;
    background-image:url('../..//images/videoBox/left.png');
    z-index: 100;
    opacity: 0.6;
}

#videoBox_right
{
    position:absolute;
    left:510px; top:55px;
    width:8px; height:278px;
    background-image:url('../..//images/videoBox/right.png');
    z-index: 100;
    opacity: 0.6;
}
```

```
*
{
    padding: 0;
    margin: 0;
    border: 0;
    color:#FFFFFF;
    background-color: transparent;
}

#SceneScene1 {
    padding: 0;
    margin: 0px;
    border: 0;
    top: -15px;
    position: relative;
    color:#FFFFFF;
    background-color: transparent;
    width: 960px; height: 520px;
}

#main
{
    position: absolute;
    left: 0px; top: 0px;
    width: 960px; height: 540px;
}

#logo
{
    position: absolute;
    top: -5px;
    height: 35px;
    width: 100%;
    background-image: url('../..//images/BgLogo.png');
    background-repeat:repeat-x;
    font-size:25px;
    padding-top:5px;
    padding-left:20px;
}

/* Right Half */
#rightHalfBackOne
{
    position: absolute;
    left:420px; top: 37px;
    width: 540px; height: 18px;
    background: url('../..//images/background_texture.png') repeat;
}

#rightHalfBackTwo
{
    position: absolute;
    left:420px; top: 37px;
    width: 37px; height: 478px;
    background: url('../..//images/background_texture.png') repeat;
}
```

```
#navi
{
    position: absolute;
    top: 500px;
    height: 35px;
    width: 100%;
    background-image: url('../..//Images/BgNavigator.png');
    background-repeat: repeat-x;
    opacity: 0.9;
}

#help_navi
{
    position: absolute;
    left: 80px; top: 0px;
    width: 785px; height: 28px;
}

.style_navi {
    bottom: 4px;
    height: 20px;
    font-size: 20px;
}

.style_bitrate
{
    position: absolute;
    left: 65px; top: 313px;
    width: 830px; height: 220px;
    vertical-align: bottom;
    z-index: 20;
    opacity: 0.5;
    border: none;
    background-image: url('../..//images/bitrate/bitrateBG4.png');
}

.style_bitratebar
{
    position: absolute;
    left: 1px; top: 1px;
    width: 1px; height: 1px;
    opacity: 0.5;
    border: none;
    background-image: url('../..//images/bitrate/blue128.png');
}

.style_bandwidthbar
{
    position: absolute;
    left: 1px; top: 1px;
    width: 1px; height: 1px;
    opacity: 0.5;
    border: none;
    background-image: url('../..//images/bitrate/red128.png');
}
```


6.6.4 Main.jss

```

var widgetAPI = new Common.API.Widget();
var tvKey = new Common.API.TVKeyValue();

var Main =
{
    selectedVideo : 0,
    mode : 0,
    mute : 0,

    UP : 0,
    DOWN : 1,

    WINDOW : 0,
    FULLSCREEN : 1,

    NMUTE : 0,
    YMUTE : 1
}

Main.onLoad = function()
{
    if ( Player.init() && Audio.init() && Display.init() && Server.init() )
    {
        Display.setVolume( Audio.getVolume() );
        document.getElementById("volumeBar").style.backgroundImage =
"url (Images/videoBox/volumeBar.png)";
        Display.setTime(0);

        Player.stopCallback = function()
        {
            /* Return to windowed mode when video is stopped
            (by choice or when it reaches the end) */
            Main.setWindowMode();
        }

        // Start retrieving data from server
        Server.dataReceivedCallback = function()
        {
            /* Use video information when it has arrived */
            Display.setVideoList( Data.getVideoNames(), Data.getVideoLogos()
);
            Main.updateCurrentVideo();
        }
        Server.fetchVideoList(); /* Request video information from server */

        // Enable key event processing
        this.enableKeys();

        widgetAPI.sendReadyEvent();
    }
    else
    {
        alert("Failed to initialise");
    }
}

```

```
    }  
  
}  
  
Main.onUnload = function()  
{  
    Player.deinit();  
}  
  
Main.updateCurrentVideo = function(move)  
{  
    Player.setVideoURL( Data.getVideoURL(this.selectedVideo) );  
  
    Display.setVideoListPosition(this.selectedVideo, move);  
  
    Display.setDescription( Data.getVideoDescription(this.selectedVideo));  
  
    Display.setVideoType( Data.getVideoType(this.selectedVideo));  
  
    Display.coverLaunch( Data.getVideoCover(this.selectedVideo));  
}  
  
Main.enableKeys = function()  
{  
    document.getElementById("anchor").focus();  
}  
  
Main.keyDown = function()  
{  
    var keyCode = event.keyCode;  
    alert("Key pressed: " + keyCode);  
  
    switch(keyCode)  
    {  
        case tvKey.KEY_RETURN:  
        case tvKey.KEY_PANEL_RETURN:  
            alert("RETURN");  
            Player.stopVideo();  
            widgetAPI.sendReturnEvent();  
            break;  
            break;  
  
        case tvKey.KEY_PLAY:  
            alert("PLAY");  
            this.handlePlayKey();  
            //Bitrate.init();  
            var playerplugin = document.getElementById("pluginPlayer");  
            break;  
  
        case tvKey.KEY_STOP:  
            alert("STOP");  
            Player.stopVideo();  
            Bitrate.stopMonitor();  
            Bitrate.deinit();  
            break;
```

```
case tvKey.KEY_PAUSE:  
    alert("PAUSE");  
    this.handlePauseKey();  
    break;
```

```
case tvKey.KEY_FF:  
    alert("FF");  
    if(Player.getState() != Player.PAUSED)  
        Player.skipForwardVideo();  
    break;
```

```
case tvKey.KEY_RW:  
    alert("RW");  
    if(Player.getState() != Player.PAUSED)  
        Player.skipBackwardVideo();  
    break;
```

```
case tvKey.KEY_VOL_UP:  
case tvKey.KEY_PANEL_VOL_UP:  
    alert("VOL UP");  
    if(this.mute == 0)  
        Audio.setRelativeVolume(0);  
    break;
```

```
case tvKey.KEY_VOL_DOWN:  
case tvKey.KEY_PANEL_VOL_DOWN:  
    alert("VOL DOWN");  
    if(this.mute == 0)  
        Audio.setRelativeVolume(1);  
    break;
```

```
case tvKey.KEY_DOWN:  
    alert("DOWN");  
    this.selectNextVideo(this.DOWN);  
    break;
```

```
case tvKey.KEY_UP:  
    alert("UP");  
    this.selectPreviousVideo(this.UP);  
    break;
```

```
case tvKey.KEY_ENTER:  
case tvKey.KEY_PANEL_ENTER:  
    alert("ENTER");  
    this.toggleMode();  
    break;
```

```
case tvKey.KEY_MUTE:  
    alert("MUTE");  
    this.muteMode();  
    break;
```

```
default:  
    alert("Unhandled key");  
    break;
```

```
    }  
}
```

```
Main.handlePlayKey = function()
{
    switch ( Player.getState() )
    {
        case Player.STOPPED:
            Player.playVideo();
            break;

        case Player.PAUSED:
            Player.resumeVideo();
            break;

        case Player.PREV:
            Player.stopVideo();
            Player.playVideo();

        default:
            alert("Ignoring play key, not in correct state");
            break;
    }
}

Main.handlePauseKey = function()
{
    switch ( Player.getState() )
    {
        case Player.PLAYING:
            Player.pauseVideo();
            break;

        default:
            alert("Ignoring pause key, not in correct state");
            break;
    }
}

Main.selectNextVideo = function(down)
{
    if(Player.getState() != Player.STOP) Player.stopVideo();

    this.selectedVideo = (this.selectedVideo + 1) % Data.getVideoCount();

    this.updateCurrentVideo(down);
}

Main.selectPreviousVideo = function(up)
{
    if(Player.getState() != Player.STOP) Player.stopVideo();

    if (--this.selectedVideo < 0)
    {
        this.selectedVideo += Data.getVideoCount();
    }
}
```

```
        this.updateCurrentVideo(up);
    }

Main.setFullScreenMode = function()
{
    if (this.mode !== this.FULLSCREEN)
    {
        Display.hide();

        Player.setFullscreen();

        this.mode = this.FULLSCREEN;
    }
}

Main.setWindowMode = function()
{
    if (this.mode !== this.WINDOW)
    {
        Display.show();

        Player.setWindow();

        this.mode = this.WINDOW;
    }
}

Main.toggleMode = function()
{
    state = Player.getState();

    switch (this.mode)
    {
        case this.WINDOW:
            if (state==Player.PLAYING || state==Player.PAUSED)
this.setFullScreenMode();
            break;

        case this.FULLSCREEN:
            this.setWindowMode();
            break;

        default:
            alert("ERROR: unexpected mode in toggleMode");
            break;
    }
}

Main.setMuteMode = function()
{
    if (this.mute !== this.YMUTE)
    {
        var volumeElement = document.getElementById("volumeInfo");
        Audio.plugin.SetSystemMute(true);
        document.getElementById("volumeBar").style.backgroundImage =
"url (Images/videoBox/muteBar.png) ";
    }
}
```

```

        document.getElementById("volumeIcon").style.backgroundImage =
"url (Images/videoBox/mute.png) ";
        widgetAPI.putInnerHTML(volumeElement, "MUTE");
        this.mute = this.YMUTE;
    }
}

Main.noMuteMode = function()
{
    if (this.mute != this.NMUTE)
    {
        Audio.plugin.SetSystemMute(false);
        document.getElementById("volumeBar").style.backgroundImage =
"url (Images/videoBox/volumeBar.png) ";
        document.getElementById("volumeIcon").style.backgroundImage =
"url (Images/videoBox/volume.png) ";
        Display.setVolume( Audio.getVolume() );
        this.mute = this.NMUTE;
    }
}

Main.muteMode = function()
{
    switch (this.mute)
    {
        case this.NMUTE:
            this.setMuteMode();
            break;

        case this.YMUTE:
            this.noMuteMode();
            break;

        default:
            alert("ERROR: unexpected mode in muteMode");
            break;
    }
}

```

6.6.5 Player.jss

```

var Player =
{
    plugin : null,
    state : -1,
    skipState : -1,
    stopCallback : null, /* Callback function to be set by client */
    originalSource : null,

    STOPPED : 0,
    PLAYING : 1,

```

```

    PAUSED : 2,
    FORWARD : 3,
    REWIND : 4
}

Player.init = function()
{
    var success = true;

    this.state = this.STOPPED;

    this.plugin = document.getElementById("pluginPlayer");

    if (!this.plugin)
    {
        success = false;
    }
    else
    {
        var mwPlugin = document.getElementById("pluginTVMW");

        if (!mwPlugin)
        {
            success = false;
        }
        else
        {
            /* Save current TV Source */
            this.originalSource = mwPlugin.GetSource();

            /* Set TV source to media player plugin */
            mwPlugin.SetMediaSource();
        }
    }

    this.setWindow();

    this.plugin.OnCurrentPlayTime = 'Player.setCurTime';
    this.plugin.OnStreamInfoReady = 'Player.setTotalTime';
    this.plugin.OnBufferingStart = 'Player.onBufferingStart';
    this.plugin.OnBufferingProgress = 'Player.onBufferingProgress';
    this.plugin.OnBufferingComplete = 'Player.onBufferingComplete';

    return success;
}

Player.deinit = function()
{
    var mwPlugin = document.getElementById("pluginTVMW");

    if (mwPlugin && (this.originalSource != null) )
    {
        /* Restore original TV source before closing the widget */
        mwPlugin.SetSource(this.originalSource);
        alert("Restore source to " + this.originalSource);
    }
}

```

```
Player.setWindow = function()
{
    this.plugin.SetDisplayArea(458, 58, 472, 270);
}

Player.setFullscreen = function()
{
    this.plugin.SetDisplayArea(0, 0, 960, 540);
}

Player.setVideoURL = function(url)
{
    this.url = url;
    alert("URL = " + this.url);
}

Player.playVideo = function()
{
    if (this.url == null)
    {
        alert("No videos to play");
    }
    else
    {
        this.state = this.PLAYING;
        document.getElementById("play").style.opacity = '0.2';
        document.getElementById("stop").style.opacity = '1.0';
        document.getElementById("pause").style.opacity = '1.0';
        document.getElementById("forward").style.opacity = '1.0';
        document.getElementById("rewind").style.opacity = '1.0';
        Display.status("Play");
        this.setWindow();

        this.plugin.SetInitialBuffer(640*1024);
        this.plugin.SetPendingBuffer(640*1024);

        this.plugin.Play( this.url );
        Audio.plugin.SetSystemMute(false);
    }
}

Player.pauseVideo = function()
{
    this.state = this.PAUSED;
    document.getElementById("play").style.opacity = '1.0';
    document.getElementById("stop").style.opacity = '1.0';
    document.getElementById("pause").style.opacity = '0.2';
    document.getElementById("forward").style.opacity = '0.2';
    document.getElementById("rewind").style.opacity = '0.2';
    Display.status("Pause");
    this.plugin.Pause();
}

Player.stopVideo = function()
{
    if (this.state != this.STOPPED)
```



```

    {
        this.state = this.STOPPED;
        document.getElementById("play").style.opacity = '1.0';
        document.getElementById("stop").style.opacity = '0.2';
        document.getElementById("pause").style.opacity = '0.2';
        document.getElementById("forward").style.opacity = '0.2';
        document.getElementById("rewind").style.opacity = '0.2';
        Display.status("Stop");
        this.plugin.Stop();
        Display.setTime(0);

        if (this.stopCallback)
        {
            this.stopCallback();
        }
        else
        {
            alert("Ignoring stop request, not in correct state");
        }
    }
}

```

```
Player.resumeVideo = function()
```

```

{
    this.state = this.PLAYING;
    document.getElementById("play").style.opacity = '0.2';
    document.getElementById("stop").style.opacity = '1.0';
    document.getElementById("pause").style.opacity = '1.0';
    document.getElementById("forward").style.opacity = '1.0';
    document.getElementById("rewind").style.opacity = '1.0';
    Display.status("Play");
    this.plugin.Resume();
}

```

```
Player.skipForwardVideo = function()
```

```

{
    this.skipState = this.FORWARD;
    this.plugin.JumpForward(5);
}

```

```
Player.skipBackwardVideo = function()
```

```

{
    this.skipState = this.REWIND;
    this.plugin.JumpBackward(5);
}

```

```
Player.getState = function()
```

```

{
    return this.state;
}

```

```
// Global functions called directly by the player
```

```
Player.onBufferingStart = function()
```

```

{
    Display.status("Buffering...");
    switch(this.skipState)

```

```
{
    case this.FORWARD:
        document.getElementById("forward").style.opacity = '0.2';
        break;

    case this.REWIND:
        document.getElementById("rewind").style.opacity = '0.2';
        break;
}

Player.onBufferingProgress = function(percent)
{
    Display.status("Buffering:" + percent + "%");
}

Player.onBufferingComplete = function()
{
    Display.status("Play");
    switch(this.skipState)
    {
        case this.FORWARD:
            document.getElementById("forward").style.opacity = '1.0';
            break;

        case this.REWIND:
            document.getElementById("rewind").style.opacity = '1.0';
            break;
    }
}

Player.setCurTime = function(time)
{
    Display.setTime(time);
}

Player.setTotalTime = function()
{
    Display.setTotalTime(Player.plugin.GetDuration());
}

onServerError = function()
{
    Display.status("Error de servidor");
}

OnNetworkDisconnected = function()
{
    Display.status("Error de conexión");
}

getBandwidth = function(bandwidth) { alert("getBandwidth " + bandwidth); }

onDecoderReady = function() { alert("onDecoderReady"); }

onRenderError = function() { alert("onRenderError"); }
```

```
stopPlayer = function()
{
    Player.stopVideo();
}

setTottalBuffer = function(buffer) { alert("setTottalBuffer " + buffer); }

setCurBuffer = function(buffer) { alert("setCurBuffer " + buffer); }
```

6.6.6 Audio.jss

```
var Audio =
{
    plugin : null
}

Audio.init = function()
{
    var success = true;

    this.plugin = document.getElementById("pluginAudio");

    if (!this.plugin)
    {
        success = false;
    }

    return success;
}

Audio.setRelativeVolume = function(delta)
{
    this.plugin.SetVolumeWithKey(delta);
    Display.setVolume( this.getVolume() );
}

Audio.getVolume = function()
{
    return this.plugin.GetVolume();
}
```

6.6.7 Server.jss

```
var Server =
{
    /* Callback function to be set by client */
    dataReceivedCallback : null,

    XHRObj : null,
    url : "XML/videoList.xml"
```

```
}

Server.init = function()
{
    var success = true;

    if (this.XHRObj)
    {
        this.XHRObj.destroy(); // Save memory
        this.XHRObj = null;
    }

    return success;
}

Server.fetchVideoList = function()
{
    if (this.XHRObj == null)
    {
        this.XHRObj = new XMLHttpRequest();
    }

    if (this.XHRObj)
    {
        this.XHRObj.onreadystatechange = function()
        {
            if (Server.XHRObj.readyState == 4)
            {
                Server.createVideoList();
            }
        }

        this.XHRObj.open("GET", this.url, true);
        this.XHRObj.send(null);
    }
    else
    {
        alert("Failed to create XHR");
    }
}

Server.createVideoList = function()
{
    if (this.XHRObj.status != 200)
    {
        Display.status("XML Server Error " + this.XHRObj.status);
    }
    else
    {
        var xmlElement = this.XHRObj.responseXML.documentElement;

        if (!xmlElement)
        {
            alert("Failed to get valid XML");
        }
        else
        {

```

```

// Parse RSS
// Get all "item" elements
var items = xmlElement.getElementsByTagName("item");

var videoNames = [ ];
var videoURLs = [ ];
var videoDescriptions = [ ];
var videoCovers = [ ];
var videoLogos = [ ];
var videoType = [ ];

for (var index = 0; index < items.length; index++)
{
    var titleElement =
items[index].getElementsByTagName("title")[0];
    var descriptionElement =
items[index].getElementsByTagName("description")[0];
    var linkElement = items[index].getElementsByTagName("link")[0];
    var typeElement = items[index].getElementsByTagName("type")[0];
    var coverElement =
items[index].getElementsByTagName("cover")[0];
    var logoElement = items[index].getElementsByTagName("logo")[0];

    if (titleElement && descriptionElement && linkElement &&
coverElement && logoElement)
    {
        videoNames[index] = titleElement.firstChild.data;
        videoURLs[index] = linkElement.firstChild.data;
        videoDescriptions[index] =
descriptionElement.firstChild.data;
        videoType[index] = typeElement.firstChild.data;
        videoCovers[index] = coverElement.firstChild.data;
        videoLogos[index] = logoElement.firstChild.data;
    }
}

Data.setVideoNames(videoNames);
Data.setVideoURLs(videoURLs);
Data.setVideoDescriptions(videoDescriptions);
Data.setVideoCovers(videoCovers);
Data.setVideoType(videoType);
Data.setVideoLogos(videoLogos);

if (this.dataReceivedCallback)
{
    this.dataReceivedCallback(); /* Notify all data is received
and stored */
}
}
}
}
}

```

6.6.8 Data.jss

```
var Data =
{
  videoNames : [ ],
  videoURLs : [ ],
  videoDescriptions : [ ],
  videoCovers : [ ],
  videoLogos : [ ],
  videoType : [ ]
}

Data.setVideoNames = function(list)
{
  this.videoNames = list;
}

Data.setVideoURLs = function(list)
{
  this.videoURLs = list;
}

Data.setVideoLogos = function(list)
{
  this.videoLogos = list;
}

Data.setVideoDescriptions = function(list)
{
  this.videoDescriptions = list;
}

Data.setVideoCovers = function(list)
{
  this.videoCovers = list;
}

Data.setVideoType = function(list)
{
  this.videoType = list;
}

Data.setSelectedVideo = function(list)
{
  this.selectedVideo = list;
}

Data.getVideoURL = function(index)
{
  var url = this.videoURLs[index];

  if (url) // Check for undefined entry (outside of valid array)
  {
    return url;
  }
}
```

```
    else
    {
        return null;
    }
}

Data.getVideoLogos = function()
{
    return this.videoLogos;
}

Data.getVideoCount = function()
{
    return this.videoURLs.length;
}

Data.getVideoNames = function()
{
    return this.videoNames;
}

Data.getVideoDescription = function(index)
{
    var description = this.videoDescriptions[index];

    if (description) // Check for undefined entry (outside of valid array)
    {
        return description;
    }
    else
    {
        return "No description";
    }
}

Data.getVideoType = function(index)
{
    var vtype = this.videoType[index];
    return vtype;
}

Data.getVideoCover = function(index)
{
    var cover = this.videoCovers[index];

    if (cover) // Check for undefined entry (outside of valid array)
    {
        return cover;
    }
    else
    {
        return "";
    }
}
```

6.6.9 Display.jss

```
var Display =
{
  statusDiv : null,
  FIRSTIDX : 0,
  LASTIDX : 4,
  currentWindow : 0,

  SELECTOR : 0,
  LIST : 1,

  videoList : new Array(),
  videoType : "",
}

Display.init = function()
{
  var success = true;

  this.statusDiv = document.getElementById("status");

  if (!this.statusDiv)
  {
    success = false;
  }

  return success;
}

Display.setTotalTime = function(total)
{
  this.totalTime = total;
}

Display.setVideoType = function(type)
{
  this.videoType = type;
}

Display.setTime = function(time)
{
  var timePercent = (100 * time) / this.totalTime;
  var timeElement = document.getElementById("timeInfo");
  var timeHTML = "";
  var timeHour = 0; var timeMinute = 0; var timeSecond = 0;
  var totalTimeHour = 0; var totalTimeMinute = 0; var totalTimeSecond = 0;

  document.getElementById("progressBar").style.width = timePercent + "%";

  if(Player.state == Player.PLAYING)
  {
    totalTimeHour = Math.floor(this.totalTime/360000);
  }
}
```



```

    timeHour = Math.floor(time/360000);

    totalTimeMinute = Math.floor((this.totalTime%360000)/60000);
    timeMinute = Math.floor((time%360000)/60000);

    totalTimeSecond = Math.floor((this.totalTime%60000)/1000);
    timeSecond = Math.floor((time%60000)/1000);

    timeHTML = timeHour + ":";

    if(timeMinute == 0)
        timeHTML += "00:";
    else if(timeMinute <10)
        timeHTML += "0" + timeMinute + ":";
    else
        timeHTML += timeMinute + ":";

    if(timeSecond == 0)
        timeHTML += "00/";
    else if(timeSecond <10)
        timeHTML += "0" + timeSecond;
    else
        timeHTML += timeSecond;

    if (this.videoType!="live"){
        timeHTML += "/" + totalTimeHour + ":";

        if(totalTimeMinute == 0)
            timeHTML += "00:";
        else if(totalTimeMinute <10)
            timeHTML += "0" + totalTimeMinute + ":";
        else
            timeHTML += totalTimeMinute + ":";

        if(totalTimeSecond == 0)
            timeHTML += "00";
        else if(totalTimeSecond <10)
            timeHTML += "0" + totalTimeSecond;
        else
            timeHTML += totalTimeSecond;
    }
}
else
    timeHTML = "0:00:00/0:00:00";

widgetAPI.putInnerHTML(timeElement, timeHTML);
}

Display.status = function(status)
{
    alert(status);
    widgetAPI.putInnerHTML(this.statusDiv, status);
}

Display.setVolume = function(level)
{

```



```

    else if((this.currentWindow!=this.LASTIDX && move==Main.DOWN) ||
(this.currentWindow!=this.FIRSTIDX && move==Main.UP))
    {
        if(move == Main.DOWN)
            this.currentWindow ++;
        else
            this.currentWindow --;

        for (var i = 0; i <= this.LASTIDX; i++)
        {
            if(i == this.currentWindow)
                this.videoList[i].style.backgroundImage=
"url (Images/listBox/selector.png)";
            else
                this.videoList[i].style.backgroundImage= "url (none)";
        }
    }
    else if(this.currentWindow == this.LASTIDX && move == Main.DOWN)
    {
        if(position == this.FIRSTIDX)
        {
            this.currentWindow = this.FIRSTIDX;

            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = '';
                listHTML += Data.videoNames[i] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);

                if(i == this.currentWindow)
                    this.videoList[i].style.backgroundImage=
"url (Images/listBox/selector.png)";
                else
                    this.videoList[i].style.backgroundImage= "url (none)";
            }
        }
        else
        {
            for(i = 0; i <= this.LASTIDX; i++)
            {
                listHTML = '';
                listHTML += Data.videoNames[i + position - this.currentWindow] ;
                widgetAPI.putInnerHTML(this.videoList[i], listHTML);
            }
        }
    }
    else if(this.currentWindow == this.FIRSTIDX && move == Main.UP)
    {
        if(position == Data.getVideoCount()-1)
        {
            this.currentWindow = this.LASTIDX;

            for(i = 0; i <= this.LASTIDX; i++)
            {

```

```

        listHTML = '';
        listHTML += Data.videoNames[i + position - this.currentWindow] ;
        widgetAPI.putInnerHTML(this.videoList[i], listHTML);

        if(i == this.currentWindow)
            this.videoList[i].style.backgroundImage=
"url (Images/listBox/selector.png)";
        else
            this.videoList[i].style.backgroundImage= "url (none)";
    }
}
else
{
    for(i = 0; i <= this.LASTIDX; i++)
    {
        listHTML = '';
        listHTML += Data.videoNames[i + position] ;
        widgetAPI.putInnerHTML(this.videoList[i], listHTML);
    }
}
}
}

Display.setDescription = function(description)
{
    var descriptionElement = document.getElementById("description");

    widgetAPI.putInnerHTML(descriptionElement, description);
}

Display.coverShow = function()
{
    Display.coverLaunch( Data.getVideoCover(this.selectedVideo));
}

Display.coverHide = function()
{
    sf.service.ImageViewer.hide();
}

Display.coverLaunch = function(url)
{
    sf.service.ImageViewer.draw({
        url: url, // image URL
        width: '1600', // image width
        height: '1067' // image height
    });
}

Display.hide = function()
{
    document.getElementById("main").style.display="none";
}

```

```
}  
  
Display.show = function()  
{  
    document.getElementById("main").style.display="block";  
}
```

7 MANUAL DE USUARIO

En este capítulo se mostrará el modo de utilización de esta aplicación interactiva, incluyendo la fase de instalación pese a no encontrarse en el almacén de aplicaciones antes de la entrega de este proyecto.

7.1 Instalación

Para realizar la instalación de la aplicación se hace necesaria la conexión al menú del televisor denominado Smart Hub. Este menú proporciona los servicios y contenidos de entretenimiento directamente desde la web al televisor.

Pantalla de visualización

Puede buscar diferentes tipos de contenidos para utilizarlos en su televisor. Escriba una palabra clave para buscar diferentes aplicaciones multimedia o información en línea para utilizarlas en su televisor.

Pantalla del televisor: Muestra la pantalla de la fuente de entrada actual.

Información: Muestra noticias, nuevas aplicaciones y anuncios que le ofrece la introducción de productos de Samsung.

Salir: Cierra **SMART HUB**.

Servicio de aplicación: Puede probar diferentes servicios proporcionados por Samsung.

Aplicaciones recomendadas: Muestra el servicio recomendado por Samsung. No puede añadir ni borrar ningún servicio en esta lista.

Lista de aplicaciones instaladas: Muestra las aplicaciones instaladas en el televisor.

Controles:

- **A** Rojo (**Ini. ses.**): Para registrarse en **SMART HUB**.
- **B** Verde (**Ordenar por**): Para clasificar las aplicaciones por categorías.
- **C** Amarillo (**Modo edic.**): Puede editar las aplicaciones (como mover, crear carpetas).
- **D** Azul (**Config.**): Editar y configurar aplicaciones y configurar SMART HUB.

Los botones de colores pueden funcionar de forma diferente, según la aplicación.

FIGURA 46: PANTALLA DE VISUALIZACIÓN


Configure la red antes de utilizar SMART HUB. Para obtener más información, consulte configuración de red.

Las fuentes no admitidas del contenido del proveedor no se mostrarán correctamente.

Puede haber respuestas lentas e interrupciones según el estado de la red.

Según la región, quizás sólo se admita el idioma inglés en los servicios de la aplicación.

7.1.1 Registro de la cuenta

En una aplicación con varias cuentas, seleccione  para acceder a ellas.







Para utilizar más satisfactoriamente una aplicación, debe registrarse e iniciar la sesión en su cuenta.

Para ver las instrucciones sobre cómo crear una cuenta, consulte “Cómo crear un Smart TV ID”.

La imagen mostrada puede variar según el modelo.




FIGURA 47: REGISTRO CUENTA

1. Pulse el botón rojo en la página de inicio de SMART HUB.
2. Seleccione  con los botones  /  /  / . Aparece la lista de las cuentas. Si desea crear una cuenta, seleccione Crear cuenta y aparecerá la ventana OSD para crear la cuenta.
3. Seleccione el Smart TV ID deseado y pulse el botón ENTER .
4. Tras seleccionar el Smart TV ID, escriba la Contraseña mediante el mando a distancia. Una vez registrada la cuenta, el Smart TV ID se mostrará en la pantalla.

7.1.2 Cómo crear un Smart TV ID

Antes de crear un Smart TV ID, debe descargar la aplicación en Samsung Apps.

1. Pulse el botón rojo. Aparece la pantalla de inicio de sesión.
2. Seleccione Crear cuenta. Se muestra la pantalla Crear cuenta.
3. Pulse el botón ENTER  para introducir un nuevo Smart TV ID. Aparecerá la pantalla del teclado.
4. Mediante la pantalla del teclado, introduzca un Smart TV ID.

Se debe crear un Smart TV ID con el formato “ejemplo@ejemplo.com”.

5. Mediante al mando a distancia, introduzca la Contraseña.
 - La contraseña debe tener 6~12 caracteres.
 - Mediante el botón PRE-CH se pueden borrar los caracteres escritos.
 - Mediante los botones ◀◀(REW) o ▶▶(FF), se puede seleccionar el icono que se desea introducir en la pantalla del teclado de iconos.
6. Se ha creado la cuenta. Se puede utilizar los servicios proporcionados por todos los usuarios del TV Samsung en un único inicio de sesión gracias a la vinculación de la cuenta de los usuarios al ID del TV. Si desea registrar un ID del sitio del servicio, seleccione Sí. En la pantalla aparecerá la lista de los sitios de servicio.
7. Seleccione el registro del sitio de servicio que desee y pulse el botón ENTER ↵.
8. Mediante los botones numéricos, escriba el ID del sitio de la aplicación y la contraseña. A continuación pulse el botón ENTER ↵.
9. Registro satisfactorio. Si ha añadido otro sitio de servicio, seleccione Sí, pulse el botón ENTER ↵ y vaya al paso 5.
10. Una vez completado el proceso, seleccione Aceptar. A continuación pulse el botón ENTER ↵.

7.1.3 Samsung Apps

Samsung Apps es un almacén para descargar aplicaciones que se pueden usar en equipos de TV / AV de Samsung.

Los usuarios pueden disfrutar de una gran variedad de contenidos como vídeo, música, fotos, juegos, información útil, etc., mediante las aplicaciones que se pueden descargar

Uso de los botones de colores con Samsung Apps.

- **A Rojo** (Ini. ses.): Para iniciar el servicio de Internet.
- **B Verde** (Vista de lista / Vista en miniatura): Para cambiar el modo de visualización.
- **↶ Volver**: Vuelve al menú anterior.

Las categorías disponibles son las siguientes:

- **Novedades**: Muestra las aplicaciones registradas más recientes.
- **Vídeo**: Muestra varios medios de vídeo como películas, programas de televisión y vídeos de corta duración.
- **Juego**: Muestra varios juegos, como el sudoku o el ajedrez.

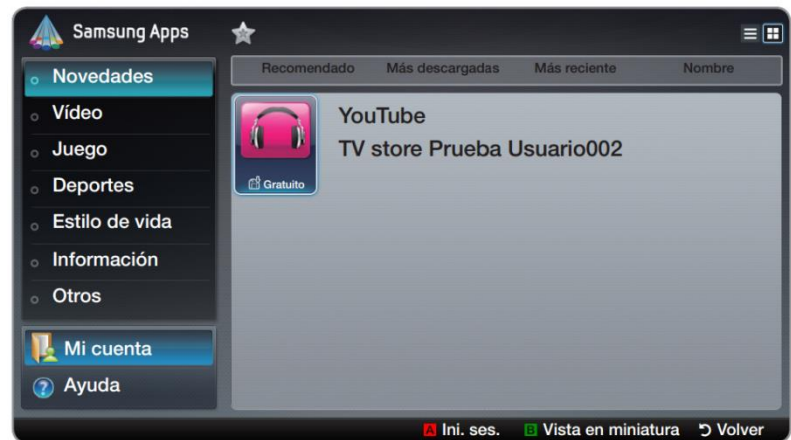


FIGURA 48: PANTALLA ALMACÉN SAMSUNG APP

- **Deportes:** Muestra varios medios deportivos como información sobre partidos, imágenes y vídeos de corta duración.
- **Estilo de vida:** Incluye varios servicios de medios de estilo de vida, como música, herramientas de gestión de fotos personales y redes sociales como Facebook y Twitter.
- **Información:** Incluye varios servicios de información como noticias, la bolsa y el tiempo.
- **Otros:** Incluye otros servicios diversos.
- **Mi cuenta:** Muestra la lista de aplicaciones y el balance de caja de la aplicación.
- **Ayuda:** Si tiene preguntas sobre Samsung Apps, consulte esta sección en primer lugar.

7.1.4 Instalación de FluMusic

Para proceder a la instalación de FluMusic (una vez esta se encuentre en el *store* de aplicaciones), se deben realizar los siguientes pasos:

1. Acceder a Samsung APP
2. Presionar Botón **A** para iniciar servicio.
3. Acceder a la categoría *Estilo de Vida*.
4. Localizar la aplicación
5. Presionar sobre el icono de la aplicación.
6. Pulsar sobre la opción *Download now*.



FIGURA 49: EJEMPLO DE PANTALLA DE INSTALACIÓN

7.2 Utilización de la aplicación

La aplicación presenta un menú lateral izquierdo **A** con los diferentes videoclips ordenados. En el lateral derecho **B** se encuentra el reproductor en miniatura que presenta una imagen por cada videoclip.

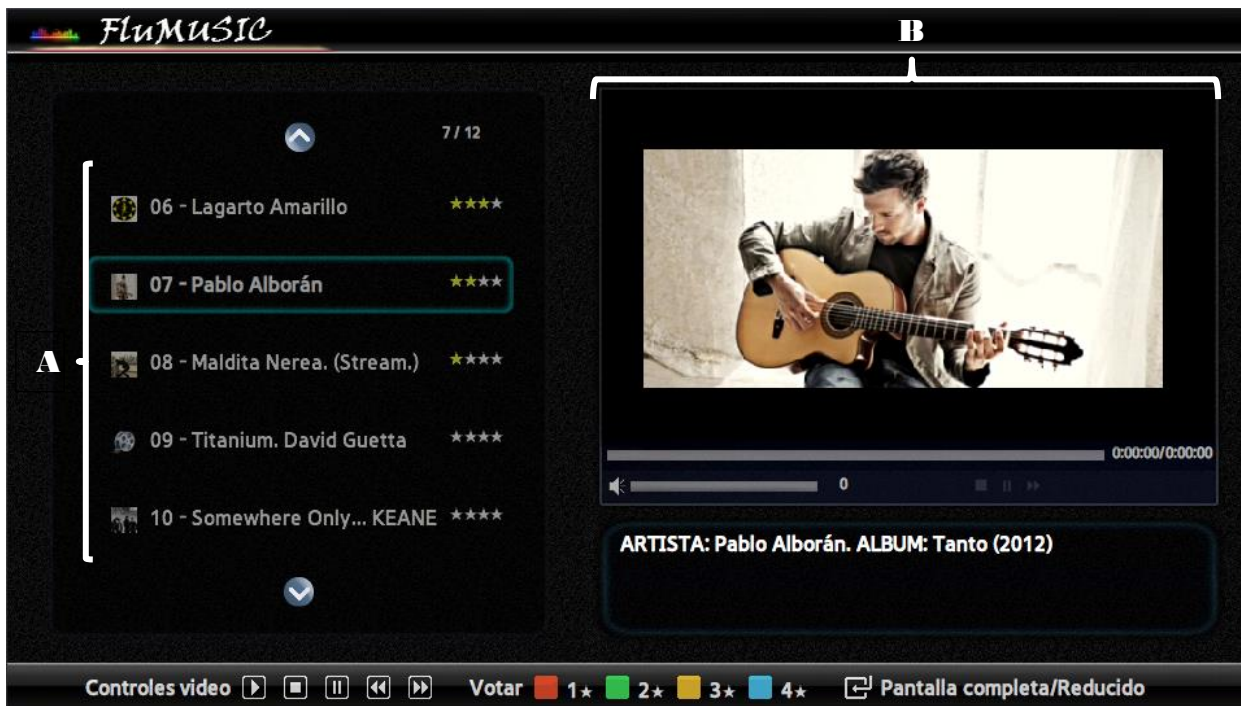


FIGURA 50: PANTALLA APLICACIÓN

SELECCIÓN DE VIDEOCLIP

Las teclas ▲ / ▼ permiten seleccionar los diferentes elementos a reproducir, visualizar su descripción y su imagen de presentación.

**FIGURA 51: SECCIÓN INFORMACIÓN VIDEOCLIP**

7.2.1 Menú de reproducción

REPRODUCCIÓN

Para iniciar la reproducción del video pulse  (reproducir).

El menú de reproducción consta de una barra de progreso de la reproducción del videoclip, un contador, una barra de control de volumen y elementos de control de reproducción.

**FIGURA 52: BARRA DE PROGRESO**

Los elementos del menú de reproducción se resaltan cuando se inicia la reproducción de un videoclip.

**FIGURA 53: BARRA DE PROGRESO “REPRODUCIENDO”**

Si la reproducción es de un canal de televisión en lugar de un videoclip, no se mostrará el tiempo de finalización.

STOP

Presionando el siguiente botón  , usted puede detener la reproducción de vídeo.

PAUSA




Si presiona la tecla  la reproducción pasa a modo **PAUSA**. La barra de estado muestra el nuevo estado en pantalla.



FIGURA 54: BARRA DE PROGRESO “PAUSA”



AVANCE Y RETROCESO

Para avanzar o retroceder la reproducción 5 segundos pulse los botones  (REW) y  (FF). Esta acción sólo se permite en modo *reproducción*.

PANTALLA COMPLETA

Si desea activar la visualización a **pantalla completa** o volver a **modo normal** presione el botón **ENTER** .

VOTAR VIDEOCLIP

Si desea votar un videoclip, debe primero seleccionar el videoclip con el selector. Para ello utilice las teclas  / .

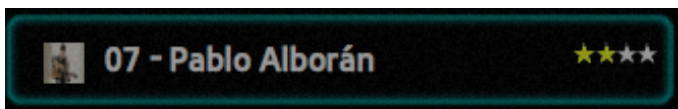


FIGURA 55: SELECTOR DE VIDEOCLIP

Una vez seleccionado presione una de las teclas de colores del mando a distancia. A continuación se muestra los códigos de colores:

Color	Puntuación
	1 estrella
	2 estrellas
	3 estrellas
	4 estrellas

Una vez pulsado el botón, el voto se transmite, y se muestra un mensaje de confirmación del envío, o código de error en su defecto.

8 MANUAL TÉCNICO DEL SISTEMA

En este documento se describirá todo lo relacionado con la parte técnica: tecnologías utilizadas, sistema de implantación,... y demás conceptos técnicos importantes imprescindibles para comprender el alcance del proyecto.

8.1 Smart Tv

Smart TV es una denominación que las compañías de televisores han dado a sus modelos conectados y más avanzados. En ellos, además de posibilidad de reproducir contenido desde diferentes fuentes, coincide una **conexión a Internet** que puede ser vía WiFi y la posibilidad de instalar o **usar aplicaciones** diseñadas específicamente para un televisor.

Esos **televisores Smart TV** también facilitan la comunicación, el acceso a las redes sociales y suelen integrar un buscador para localizar contenido entre las múltiples fuentes de las que disponen. En los modelos de 2012 se han introducido también el **control por voz y gestos**.



FIGURA 56: TELEVISOR SAMSUNG

8.2 Samsung Smart Hub

Es el punto de entrada a esta galería de servicios que incluyen navegación libre, aplicaciones, acceso a redes sociales e interoperatividad entre dispositivos vía inalámbrica.

Smart Hub nos ofrece un menú bastante completo en el cual todo gira en torno al buscador. Una serie de opciones permiten activar y desactivar filtros para delimitar las búsquedas. El sistema permite buscar en todas las fuentes de vídeo locales u online, en redes sociales, en el catálogo de Samsung Movies o en una herramienta llamada **Your Video**

Todas estas aplicaciones requieren casi necesariamente un teclado. En lugar de inventar teclados específicos, Samsung ha optado por una maniobra más inteligente. Lanzar una **aplicación Smart TV compatible con todos los móviles Android**.

Con esta aplicación instalada en un teléfono o tablet con WiFi podemos sincronizar el



FIGURA 57: CONVERGENCIA SAMSUNG

terminal con el televisor y convertirlo en un **mando a distancia táctil** con un extenso abanico de opciones. La aplicación no se queda ahí. También podemos ver los contenidos de la TV en el móvil, ver otro canal distinto al que tenemos en el televisor, y utilizar el teclado en pantalla como método para buscar en Smart TV o escribir actualizaciones de estado.

8.3 Formatos de vídeo

8.3.1 Conceptos generales

Live: Consiste en distribuir un flujo que parte de una emisión en directo. Para lograrlo se ha de codificar la señal original en un formato adecuado para la transmisión a través de Internet. Flumotion ofrece una solución end-to-end, que va desde la adquisición de la señal hasta el usuario final.

On Demand: Se trata de emitir a través de Internet un archivo de audio o vídeo y no una señal de directo. Estos archivos se almacenan en un servidor o una Content Delivery Network (CDN), como es el caso de Flumotion, y se sirven en su formato. Para cambiar de formato se han de transcodificar.

Bitrate: Define el número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Es la velocidad de transferencia de datos: kilobits por segundo (kbps), Megabits por segundo (Mbps)...

HD: El HD video o vídeo de alta definición tiene una resolución de 720 o 1080 líneas (número de líneas verticales con que se forma la imagen). En Internet, HD tiende a estar codificado con bitrates superiores a 1 Mbps y a utilizar formato 16:9 o panorámico.

SD: La definición estándar transmite imágenes de alrededor de 500 líneas. Formatos típicos son los de 576 y 480 líneas. En Internet, la definición estándar suele tener bitrates a partir de 500 kbps.

LD: La baja definición transmite imágenes con una resolución inferior a la definición estándar.

Formato / Container: Hace referencia al formato del fichero. Sus especificaciones afectan a la forma en que se almacenan los datos no en cómo se codifican, que depende del códec. En archivos de audio y vídeo, cada container es compatible con unos códecs determinados. Windows Media Video (.wmv), Flash Video (.flv), QuickTime (.mov) son formatos multimedia.

Códec: Abreviatura de 'coder-decoder'. Permiten codificar un flujo o una señal para su transmisión y descifrarlo para su reproducción en un formato apropiado. Son necesarios para la emisión de vídeo y audio en Internet.

8.3.2 Formatos y códecs de vídeo

WMV: Windows Media Video es el códec de Microsoft. Funciona con el códec de audio WMA en flujos con el formato del mismo nombre (WMV). Se reproduce con el Windows Media Player, entre otros reproductores.

MP4: Es un formato compatible con diferentes reproductores (Flash, Silverlight o QuickTime) y dispositivos móviles (iPhone, Android). Entre otros, trabaja con los códecs H.264 y AAC; su extensión es .mp4.

AVI (Audio Video Interleaved = Audio y Video Intercalado): Es el formato estándar para almacenar video digital.

- Admite distintos códecs de compresión como CinePak, Intel Indeo 5, DV, etc. Los códecs con más capacidad de compresión y una calidad aceptable son DivX y XviD.
- El formato AVI puede ser visualizado con la mayoría de reproductores: Windows Media, QuickTime, etc. siempre y cuando se encuentren instalados en el equipo los adecuados códecs para cada tipo de reproductor.
- Es ideal para guardar videos originales que han sido capturados de la cámara digital (codificados con DV).
- No es recomendable publicarlos en Internet en este formato por su enorme peso.
- Los códecs CinePak, Intel Indeo, DV, etc. no ofrecen una gran compresión. Los códecs DivX y XviD por el contrario consiguen una óptima compresión aunque se suelen destinar sobre todo a la codificación de películas de larga duración.

8.3.2.1 ¿Cómo funcionan los códecs?

Los **formatos de video** basan su eficacia en los **codecs**, unos algoritmos matemáticos que comprimen y descomprimen la señal de video digital. A diferencia de otras técnicas visuales como la fotografía, el video tiene el reto de transmitir, como mínimo, **25 imágenes en un segundo y por un caudal determinado** (canal de transmisión). Este canal puede tener unas características y limitaciones concretas, por lo que la señal de video deberá adecuarse para poder “pasar” por ese canal y llegar a su destino.

Lo que llamamos “canal” puede ser un cable, una frecuencia Wi-Fi, una frecuencia UHF o cualquier otro instrumento que sirva para transmitir datos de un lugar a otro. Tras esta explicación, entenderás que **la señal de video ocupa muchísimo volumen**, y por tanto tenemos que recurrir a un mecanismo que nos optimice todo ese volumen de información y nos lo encapsule en un formato con el que podamos manipular y transmitir todo eso. Esa sería la explicación simplificada y no-técnica de lo que es un formato de video, qué contiene en su interior y para qué sirve.

Ese mecanismo al que nos referimos es la **compresión de información**. Todo en video se comprime, excepto en casos muy concretos y muy avanzados, y uno de los estándares

más utilizados tradicionalmente por la industria es el **MPEG**. En él se basan codecs y formatos tan conocidos como el **DivX**, **H.264**, **la TDT**, **la tv por satélite**, **cable e IPTV**, **los DVD's**... Las primeras técnicas de compresión que se utilizaron partían de la fórmula “**6xVerde**” (ejemplo): si en un archivo de video había que escribir la información de seis píxeles como: Verde-Verde-Verde-Verde-Verde-Verde, resultaba mucho más útil escribir **6xVerde**, ya que se ahorra un considerable volumen de información. Optimizamos.

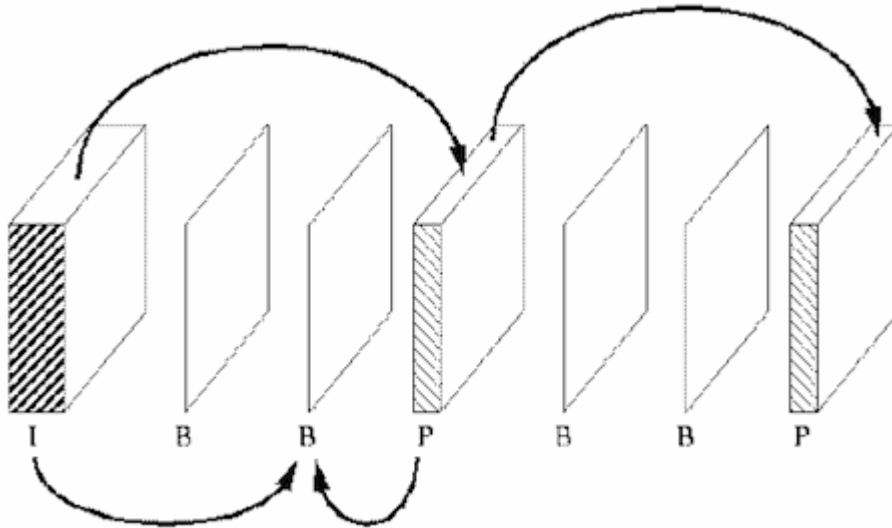


FIGURA 58: LÓGICA DE LOS CÓDECS

De una forma sencilla, **MPEG es un algoritmo de codificación y descodificación que elimina la información redundante**, no perceptible por el ojo humano hasta cierto límite, en favor de una reducción del espacio que ocupa esa información de video, lo que nos permite manipularla y transmitirla. MPEG utiliza **tres técnicas** para eliminar información redundante:

- La **redundancia temporal**, que actúa sobre un píxel o número de píxeles que se repiten constantemente en una secuencia de imágenes, fotograma a fotograma. Reduce esa información al mínimo.
- La **redundancia espacial**, que actúa sobre la similitud entre un píxel y sus vecinos adyacentes. Si son muy parecidos a la percepción que pueda tener el ojo humano, se reducen a una información común. Por ejemplo, el gris claro, y el gris un poco más claro se reducirían ambos a “gris claro simple”.
- La **redundancia estadística**, no tiene pérdidas (a diferencia de las dos anteriores), y basa su acción en que en una transmisión de video existen palabras (agrupaciones de bits) que se repiten constantemente. Por tanto, sólo se envía una

de las palabras que se repiten. Es una fórmula parecida a la que explicábamos antes: $6xVerde$, en este caso, $0110x010010101001100$, donde 0110 correspondería al número 6 en sistema decimal, por ejemplo.

Todo este caudal de información codificada la recoge un receptor, y en base a un codec, **aplica el algoritmo de descompresión** de la señal de video en base a técnicas predictivas muy avanzadas y obtenemos el resultado inicial, con pérdidas mínimas que apenas puede apreciar nuestro ojo humano.

8.4 Transmisión de vídeo por internet

8.4.1 Vídeo bajo demanda

La televisión a la carta o vídeo bajo demanda, *en inglés on demand (voD)*, es un sistema que posibilita el acceso a contenidos multimedia a elección del usuario. Se puede visualizar en tiempo real o mediante descarga.

Los servicios de descarga VoD son posibles en casas con conexión vía cable (óptico o coaxial) o bien ADSL. VoD utiliza protocolos en tiempo real, como por ejemplo RTP (Real Time Protocol) sobre UDP (User Datagram Protocol) con el protocolo de control RTCP (Real Time Control Protocol).

8.4.2 Streaming

El **streaming** es la distribución de multimedia a través de una red de computadoras de manera que el usuario consume el producto al mismo tiempo que se descarga. La palabra streaming se refiere a que se trata de una corriente continua (sin interrupción). Este tipo de tecnología funciona mediante un búfer de datos que va almacenando lo que se va descargando para luego mostrarse al usuario. Esto se contrapone al mecanismo de descarga de archivos, que requiere que el usuario descargue los archivos por completo para poder acceder a los Archivos.

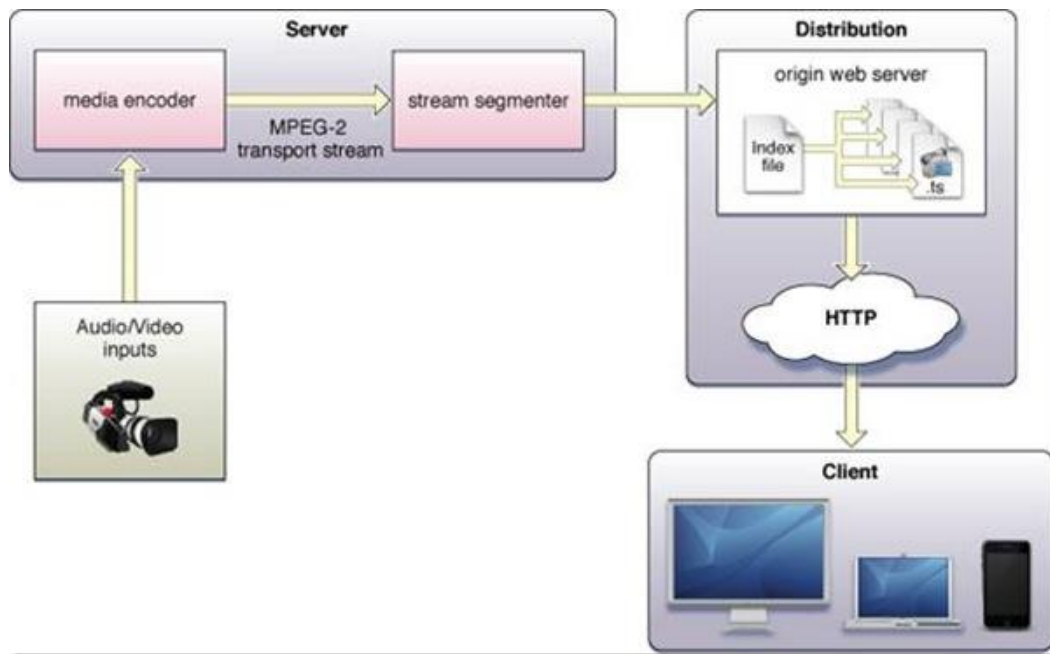
8.4.2.1 Protocolos ligeros

UDP y RTSP (los protocolos empleados por algunas tecnologías de "streaming") hacen que las entregas de paquetes de datos desde el servidor a quien reproduce el archivo se hagan con una velocidad mucho mayor que la que se obtiene por TCP y HTTP. Esta

eficiencia es alcanzada por una modalidad que favorece el flujo continuo de paquetes de datos. Cuando TCP y HTTP sufren un error de transmisión, siguen intentando transmitir los paquetes de datos perdidos hasta conseguir una confirmación de que la información llegó en su totalidad. Sin embargo, UDP continúa mandando los datos sin tomar en cuenta interrupciones, ya que en una aplicación multimedia estas pérdidas son casi imperceptibles.

8.4.2.2 Streaming adaptativo (HAS)

El streaming adaptativo, *en inglés (Adaptive bitrate streaming)*, es una técnica usada en la transmisión de video que consigue mejorar la eficiencia de la transmisión modificando el bitrate en función del estado de la red y de la CPU del dispositivo. Si la red se encuentra muy congestionada el bitrate puede disminuir para mejorar la fluidez de la descarga. Con esto conseguimos recibir un flujo de datos constante, sin cortes en la retransmisión. Muchos canales de televisión online han empezado a implantar esta técnica.



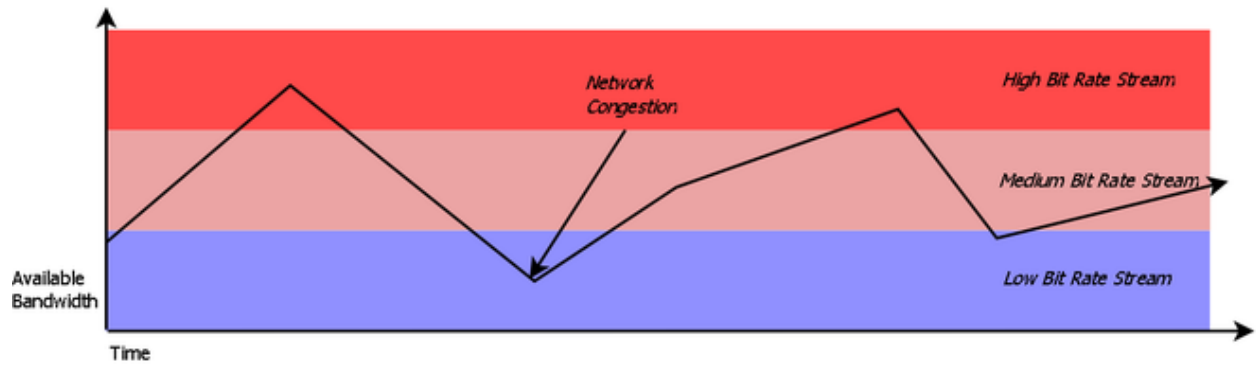


FIGURA 59: STREAMING ADAPTATIVO

En una configuración típica, existen 4 fases:

- **Preparación:** un encoder hardware toma el audio y video de entrada, lo codifica como vídeo H.264 y audio AAC, y lo devuelve en un archivo MPEG-2 Transport Stream, que luego se divide en una serie de pequeños ficheros multimedia de unos 10 segundos de duración (llamados chunks o segmentos) por parte de un segmenter software. Estos archivos se colocan en un servidor web normal. El segmentador también crea y mantiene un archivo de índice que contiene una lista de los archivos en los que se ha partido el vídeo original. La dirección URL del archivo de índice se publica en el servidor web, con una extensión .M3U8. Este proceso de codificación y segmentación se repite para cada una de las calidades que deseen emitir del vídeo (lo que es la base del Adaptive Streaming).
- **Distribución:** la URL del fichero index es accedida por los clientes, los cuales entonces piden los ficheros indexados en secuencia. Los ficheros son servidos por un servidor web tradicional, como Apache o IIS.
- **DRM** (opcional): su función limitar el acceso a contenidos por derechos de autor.
- **El software de cliente** es el responsable de determinar los ficheros o chunks apropiados a pedir por HTTP, descargarlos y reensamblarlos para poder presentarlos al usuario en un stream de vídeo continuo. Si en un momento dado las condiciones del cliente varían, es el responsable de pedir chunks de calidad más baja y regular así la calidad del stream.

8.5 Tecnologías web

8.5.1 PHP



Es un lenguaje de programación de uso general de código del lado del servidor. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje *PHP* se encuentra instalado en más de **20 millones de sitios web** y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante dominio con

otros nuevos lenguajes no tan poderosos desde agosto de **2005**.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como *C* y *Perl*, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de **contenido dinámico** sin tener que aprender todo un nuevo grupo de funciones.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el **intérprete** de PHP. Éste procesa el **script** solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

8.5.2 MySQL

MySQL es un sistema de gestión de bases de datos **relacional, multihilo y multiusuario**. Desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de **Oracle Corporation** desde abril de 2009.



Por un lado se ofrece bajo la **GNU GPL** para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Es muy frecuente encontrarlo en aplicaciones web junto con Apache. Su rendimiento es muy bueno en aplicaciones de baja-media concurrencia.

9 POSIBLES MEJORAS

A pesar de que en su versión de producción, la aplicación desarrollada incorpora requisitos adicionales no previstos en su inicio: ej. reproducción de vídeo en directo por streaming adaptativo, votación de los videoclips... El producto dispone de la funcionalidad de una aplicación de este tipo. Aunque no por ello se deben descartar nuevas ampliaciones.

Esta aplicación proporciona el marco adecuado para generar nuevas escenas, de modo que se permitiría de forma sencilla, generar nuevas pantallas donde se mostrasen listas musicales con diferentes estilos musicales, o éxitos de diferentes países. Todo ello requeriría de la mínima implementación, al poder reaprovecharse el grueso de la aplicación, y sólo se requeriría la incorporación de ligeras modificaciones, replicando la única escena creada hasta el momento.

También se podría incorporar un modo de reproducción continua y aleatoria, así como toda la lógica para la creación de listas de reproducción.

En el aspecto más propiamente de diseño, cabe valorar un rediseño del estilo de la aplicación, adaptándolo a un aspecto más corporativo, si ello resultase necesario.

10CONCLUSIONES

La ejecución de este proyecto ha reafirmado las grandes expectativas que nos brinda la recientemente aterrizada SMART TV, en tanto a potencial como a prestaciones. La oportunidad de un sector en alza, que apuesta por una convergencia tecnológica propicia el diseño de nuevas aplicaciones.

Las tecnologías incorporadas por la Samsung Smart TV disponen de un gran bagaje, lo cual, no sólo facilita la labor de análisis y programación, sino que además requieren de una curva de aprendizaje mínima con la consiguiente disminución de tiempos.

La innovación que supone el desarrollo de aplicaciones en esta plataforma, constata que el entorno de desarrollo –principalmente el emulador de la televisión- todavía carece de la suficiente madurez para evitar riesgos y retardos excesivos en el proceso de implementación. Las constantes excepciones lanzadas por el emulador con Apps verificadas, la inestabilidad del mismo, y su mejorable rendimiento, dificultan notablemente la labor de desarrollo. Todo ello comporta una incertidumbre que puede llegar a disuadir de la apuesta por proyectos en esta plataforma.

Sin embargo, cabe destacar que la evolución experimentada por la misma, insta a apostar de manera clara por aplicaciones de bajo-medio desempeño, siempre y cuando el rendimiento no se convierta en un elemento crítico.

La Smart TV es un elemento con una creciente demanda, y un corto recorrido. Lo que en la estrategia innovadora reza proclamando que: anticiparse en un sector, es mañana liderarlo.

11 FUENTES DE DOCUMENTACIÓN

11.1 Comunidad Samsung

[**Samsung Developers**] Página oficial de desarrolladores de Samsung Smart Tv.
<http://developer.samsung.com/smarttv>

[**Samsung Forum**] Foro oficial de Samsung para desarrolladores.
<http://www.samsungdforum.com/>

11.2 Manuales

[**w3schools.com**] Manual de programación para HTML 5.
<http://www.w3schools.com/html/default.asp>

[**w3schools.com**] Manual de programación para JavaScript.
<http://www.w3schools.com/js/>

[**JavaScript ya**] Tutorial para programación en javascript.
<http://www.javascriptya.com.ar/>

11.3 Códigos de ejemplo

[**Code Google**] Repositorio con ejemplo de código para Samsung Smart Tv.
<https://code.google.com/p/recommender-jsj2/source/browse/trunk/RecommenderClient/?r=123>

[**7tech.co.in**] Página con código combinado de Ajax y Php incorporado en el sistema de votación.
<http://www.7tech.co.in/php/how-to-make-a-jquery-ajax-post-request>

11.4 Servicios multimedia

[**Flumotion**] Página oficial para pruebas de la aplicación servidor de vídeos *Flumotion*.
<http://www.flumotion.com/demosite/>

[**desarrolloWeb**] Página web con información acerca de servicios XML.
<http://www.desarrolloweb.com/articulos/1545.php>

11.5 Otros

[Wikipedia] Enciclopedia gratuita

<http://es.wikipedia.org>

[Xampp] Página para descarga de servidor local para pruebas de la aplicación. Incluye módulos de apache, mySQL y php.

<http://sourceforge.net/projects/xampp/>

12ANEXO 1. PLAN DE PRUEBAS

12.1 Introducción

El contenido de este documento de plan de pruebas hace parte integral de la metodología de pruebas que se encuentra fundamentado en estándares de calidad que no solo permiten el seguimiento y correcciones a tiempo del software sino que además se encuentra definido por etapas, facilitando el seguimiento y control de los procesos del proyecto en desarrollo

12.1.1 Propósito

Este plan de pruebas tiene como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas; incluye responsabilidades de cada una de las tareas, los recursos y los prerequisites que deben ser considerados en el esfuerzo de cada una de las pruebas, permitiendo garantizar el cumplimiento de los requerimientos planteados en el marco del desarrollo del proyecto denominado "...".

12.1.2 Alcance

Este documento de PLAN DE PRUEBAS DETALLADO, se convierte en una guía para desarrollar de una forma organizada las diferentes actividades que se realizarán en el proceso del plan de pruebas en el desarrollo del proyecto denominado "...".

La metodología de pruebas y este documento de plan de pruebas permitirán, evaluar aspectos como: la lógica estructural, la funcionalidad, el rendimiento, y la integración de los diferentes módulos.

TIPO DE PRUEBA	DEFINICIONES	FASE DE RUP
UNITARIAS	Unitarias: Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.	ELABORACIÓN
SISTEMA: <ul style="list-style-type: none"> • Sistema • Carga 	Sistema: Estas pruebas buscan diferencias entre la solución desarrollada y los requerimientos, con el fin de identificar errores que se puedan generar entre la especificación funcional y el diseño del	

<ul style="list-style-type: none"> • Volumen • Interfaz de Usuario • Rendimiento 	<p>sistema.</p> <p>Carga: Valida aquellos volúmenes de datos máximos especificados en los requerimientos no Funcionales</p> <p>Volumen: Esta prueba somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software</p> <p>Interfaz de usuario: Permite verificar que la navegación a través de los elementos que se están probando, reflejen las funciones del negocio y los requerimientos funcionales.</p> <p>Rendimiento: Permite validar si la aplicación cumple los criterios de tiempos de respuesta establecidos.</p>	
<p>FUNCIONALES</p>	<p>Funcional: La prueba funcional es un proceso para procurar encontrar discrepancias entre el programa y la especificación funcional.</p> <p>Caja Negra: Estas pruebas permiten obtener conjuntos de condiciones de entrada que ejecutan todos los requisitos funcionales de un programa.</p> <p>Usabilidad: Esta prueba permite encontrar problemas de factores humanos, o usabilidad.</p>	<p>CONSTRUCCIÓN</p>
<p>ACEPTACIÓN</p>	<p>Es la prueba final basada en las especificaciones del usuario o basada en el uso del programa por el usuario final luego de un periodo de tiempo</p>	<p>TRANSICIÓN</p>
<p>REGRESIÓN</p>	<p>En esta prueba se valida que el sistema mantenga su correcta funcionalidad debido a la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas.</p>	

12.1.3 Definiciones, acrónimos y abreviaturas

- **El plan de prueba:** describe todos los métodos que se utilizarán para verificar que el software satisface la especificación del producto y las necesidades del cliente. Incluye los objetivos de calidad, necesidades de recursos, cronograma, asignaciones, métodos, etc.
- **Casos de prueba:** lista los ítems específicos que serán probados y describe los pasos detallados que serán seguidos para verificar el software.
- **Reporte de pruebas:** describen los problemas encontrados al ejecutar los casos de prueba.
- **Herramientas de pruebas y automatización:** documentación de las herramientas empleadas en el proceso de pruebas.
- **Métricas, estadísticas y resúmenes:** indican como ha sido el progreso del proceso de prueba.

12.1.4 Referecias

Algunos documentos del proyecto son base fundamental de este documento inicial de plan de pruebas, que a continuación se relacionan

- Metodología de Pruebas.
- Documento de Especificación de Requerimientos.
- Documento de Arquitectura General Detallada

12.1.5 Vista general

Descripción resumida de contenido de cada una de las secciones que siguen, y explicación de la forma en que está organizado el presente documento.

Ejecución del Plan de Pruebas

Estrategia de Pruebas	Recursos de Pruebas	Evaluación de Pruebas Ejecutados	Documentación (Anexos)
<ul style="list-style-type: none"> • Técnicas • Herramientas • Tipos de Pruebas • Entregables de Pruebas 	<ul style="list-style-type: none"> • Humano • Del Sistema • Herramienta de Reporte y Control de Incidencias • Administración de versiones 	<ul style="list-style-type: none"> • Evaluación de Resultados • Análisis de Incidencias • Descripción de Documentos de Pruebas 	<ul style="list-style-type: none"> • Release Note • Set de Pruebas • Listas de Chequeo • Informe de Pruebas • Procedimientos de Incidencias

Estrategia de Pruebas:

En este capítulo se presenta una perspectiva general de la estrategia que se va a seguir para analizar, diseñar, implementar y ejecutar las pruebas del proyecto. Así mismo se definirá qué tipos de pruebas se van a realizar y cómo se ejecutarán.

Recursos del Plan de Pruebas:

Este capítulo identifica los recursos humanos y no humanos (hardware, software, herramientas de soporte, configuración de entorno de pruebas, entre otros), necesarios para desarrollar el proceso del plan de pruebas de la solución.

Evaluación de Pruebas Ejecutadas:

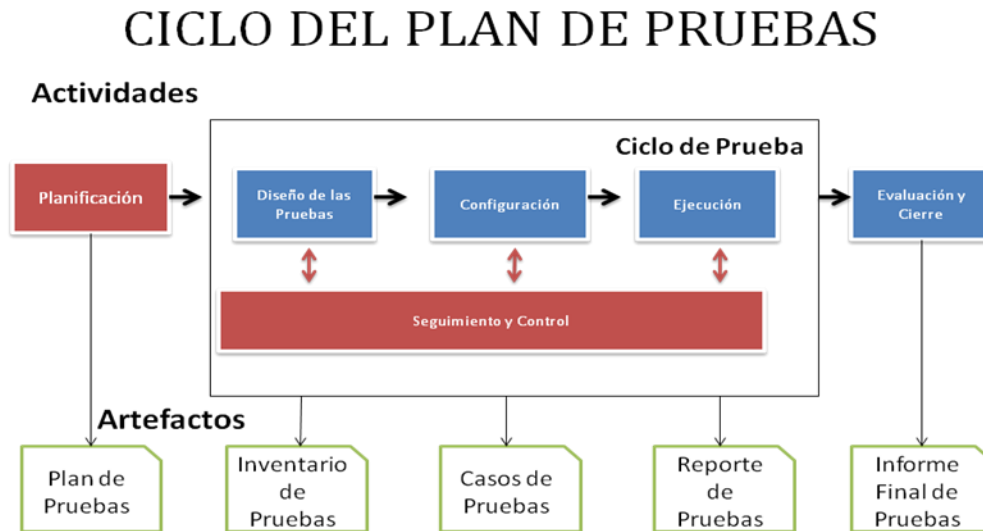
En este capítulo se describe de los métodos de evaluación de las pruebas ejecutadas, de tal forma que permitirá evaluar los grados de aceptación de las pruebas.

Anexos:

En este capítulo se describen los documentos anexos que se utilizarán para la especificación y la documentación de la ejecución de las pruebas.

12.2 Estrategia de pruebas

La estrategia del proceso del plan de pruebas se implementará de acuerdo al esquema de macro-actividades que se presenta en la siguiente gráfica:



12.2.1 Ciclo de pruebas

El ciclo de pruebas comprende seis actividades las cuales deberán ser desarrolladas de la siguiente manera:

12.2.1.1 Planificación

Para el desarrollo de la solución, se considera de gran importancia la ejecución del plan de pruebas, haciéndose necesario la planificación de las mismas, lo que en consecuencia hace necesario tener claro los siguientes planteamientos:

- Se planifican pruebas personalizando los estándares.
- Se definen niveles de pruebas a aplicar.
- Se especifican las técnicas a utilizar.
- Se establece el tiempo para la ejecución de cada una de las pruebas.
- Valoración sobre posible uso de herramientas.
- Criterios de aceptación.
- Recursos involucrados.

En la definición del plan de pruebas, se valorará:

- El alcance de la aplicación.
- La complejidad de sus procesos.

- Plataforma/s en las que se debe probar.
- Conocimientos y formación de quienes ejecutarán las pruebas.
- Normativas legales aplicables.
- Otros recursos involucrados.

Se tendrá en cuenta que:

- Las pruebas estarán presentes a lo largo de todo el ciclo de vida del desarrollo, de la solución.
- Siempre hay errores.
- Probar exhaustivamente el software es imposible.
- No es recomendable que el programador pruebe sus propios programas.
- Se puede disponer de herramientas.
- Se debe considerar la importancia de actualización del plan de pruebas con el fin de reflejar los cambios que se produzcan en los requisitos y/o proceso de desarrollo del producto.

Resultado de la planificación:

- Cronograma detallado de la ejecución de las pruebas; donde se especifica qué prueba se realiza, cuánto tiempo se estima para su ejecución, recursos a utilizar (humanos y tecnológicos); este cronograma se encuentra dentro del cronograma general del proyecto y específicamente en la fase desarrollo (ver plan de construcción)
- Formatos a utilizar para el diseño de las pruebas.
- Formatos a utilizar para el registro y análisis de los resultados de las pruebas.
- Herramientas a utilizar para la gestión de incidencias.
- Procedimientos para el control de cambios.
- Herramientas a utilizar para la ejecución de las pruebas.

12.2.1.2 Diseños de las pruebas

Para el diseño de las pruebas, se tendrán en cuenta aspectos que permitirán encontrar defectos en el periodo de desarrollo del software, la realización de pruebas propias de verificación y validación de datos, según se aclara en los siguientes ítems:

A. **Alcance:** El alcance de las pruebas estará dado por el marco del Sistema de Notificación en Línea, que se encuentra en desarrollo, ésta compuesta (Información tomada de los términos de referencia y del documento de Arquitectura General Detallada) por:

- Modelo Conceptual.
- Descripción de Procesos.

- Vista de Casos de Uso.
- Vista Lógica.
- Diseño de las clases y su organización en paquetes y subsistemas.
- Vista de Datos.
- Vista de Implementación.
- Requerimientos no Funcionales.
- Prototipos del sistema

B. Inventario de las Pruebas: En esta sección se especifica el inventario de las pruebas, el cual permitirá:

- Definir y asignar prioridades como; alta, media o baja.
- Establecer un orden de trabajo.
- Decidir qué casos entrarían en una regresión y cuáles no con mayor facilidad.
- Recortar alcance en forma rápida y ordenada.
- Se estima el tiempo en probar cada funcionalidad.
- Evaluar aspectos técnicos del sistema.

C. Resultado de la ejecución de las Pruebas: En este punto se resaltan las entradas fundamentales que son la partida para la ejecución del plan de pruebas.

- Inventario de pruebas priorizado.
- Estimación de esfuerzo de cada funcionalidad.
- Plan de desarrollo del producto.
- Plazos previstos para el proyecto.

Inventario de Pruebas	
Funcionalidad	Prioridad
RF1.1	Alta
RF1.2	Alta
RF1.3	Alta
RF1.4	Alta
RF2	Alta
RF2.1	Alta
RF2.2	Alta
RF2.3	Alta
RF2.4	Alta

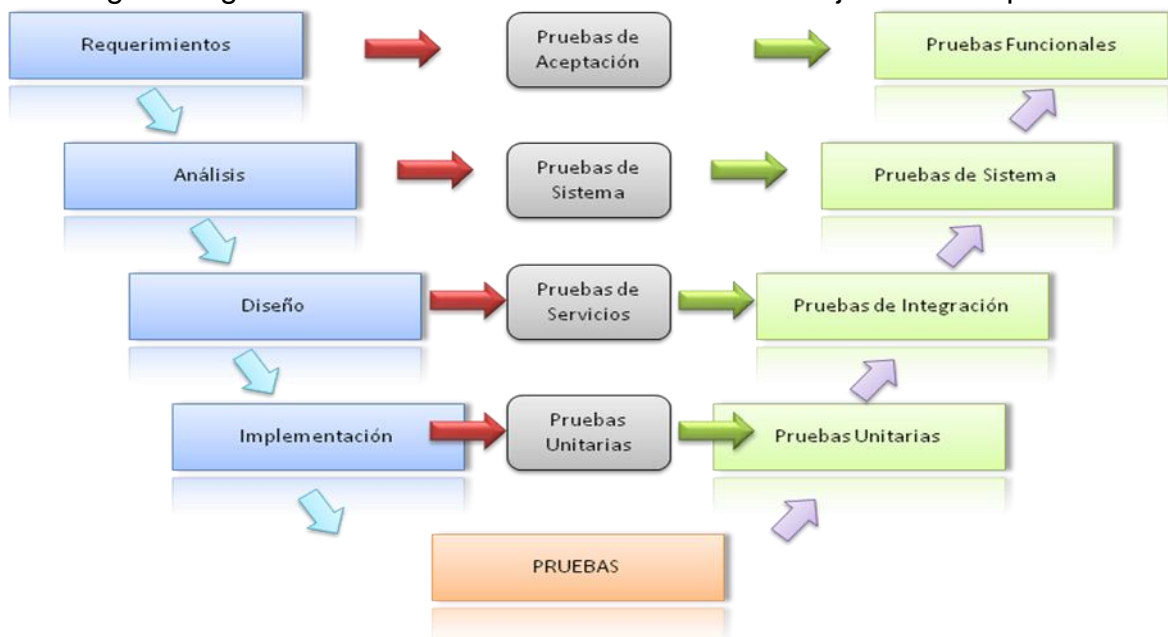
D. **Ciclo de la Prueba:** Las actividades de la prueba se realizarán para una determinada versión del producto, sobre la cual se ejecutan las pruebas y se reportan los incidentes encontrados. Para cada versión del producto se realizan alguna o todas las tareas asociadas a las pruebas.

Plan de Desarrollo		
Versión	Fecha	Funcion.
Versión 1	Fecha 1	1,2,3,4
Versión 2	Fecha 2	5,6,7
Versión 3	Fecha 3	8,9,10

El proceso de planificación se ajusta al comenzar cada ciclo debido a posibles:

- Atrasos de desarrollo
- Modificaciones en los requerimientos iniciales
- Cambios en el alcance del producto
- Calidad del producto

En el siguiente gráfico se muestra el modelo estándar de ejecución de pruebas:



Como se observa, representa un modelo de pruebas en V, a diferencia de los modelos clásicos, extiende las pruebas a lo largo de todo el ciclo de vida del software.

Mientras se realizan las fases de requerimientos, análisis, diseño e implementación se van diseñando las pruebas del mismo nivel. Al llegar a la etapa de pruebas se inicia la ejecución de lo diseñado desde las pruebas unitarias hasta las pruebas funcionales.

Para cada una de las pruebas se realizará el siguiente procedimiento:



Aquí se tendrán en cuenta las siguientes especificaciones:

- Elementos del sistema, es decir; los módulos y características de la solución que se van a probar.
- Se listarán las especificaciones de cada entrada requerida para ejecutar el caso; incluyendo la sincronizaciones entre cada una de estas.
- Especificaciones de todas las salidas y las características requeridas como el tiempo y la respuesta para los elementos que se van a probar. Estas especificaciones se harán utilizando los formatos establecidos en el numeral **Error! Reference source not found.** de este plan de pruebas.
- Necesidades del entorno del proceso de ejecución del hardware, software y recurso humano.
- Requisitos especiales de procedimiento o restricciones especiales en los procedimientos para ejecutar este caso.

12.2.1.3 Evaluación y cierre

Para la evaluación y cierre de las pruebas se presentará el informe de pruebas donde se documentará el resultado de cada una de las diferentes pruebas ejecutadas. El contenido de este informe estará compuesto de la manera descrita en la “Propuesta de esquema y contenido del Informe de pruebas”; esto ya que el informe de pruebas es un entregable independiente.

12.2.1.4 Seguimiento y control

Para el seguimiento y control de las pruebas se llevarán a cabo comités técnicos de seguimiento periódico semanal donde se evalúen los siguientes temas.

- Avance de las pruebas según cronograma
- Estado o resultado de las pruebas ejecutadas

- Seguimiento a las incidencias reportadas según la ejecución de pruebas.
- Se presentará plan de contingencia para aquellas incidencias de mayor impacto que sean de riesgo para el proyecto.

12.3 Tipos de pruebas

Las pruebas que se realizarán serán aquellas que fueron señaladas como tipos de pruebas en el numeral 8.3 de la metodología de pruebas; en este capítulo solo serán mencionados a manera general los tipos de pruebas.

El objetivo principal de la ejecución de las pruebas esta dado a:

- Descubrir tantos errores como sea posible.
- Notificar acerca de los riesgos percibidos del proyecto
- Identificar falencias funcionales de la aplicación, enmarcadas en grados de usabilidad ya definidos.
- Evaluar la calidad del producto y señalar un indicador de aceptación del mismo.
- Evaluar la calidad técnica del producto y resolver las falencias identificadas en las pruebas de tipo técnico.
- Cumplir con los requerimientos específicos del cliente, en cuanto a la ejecución de las pruebas.

12.3.1 Pruebas unitarias

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.

Es una Prueba técnica que permitirá:

- Verificar que los módulos del sistema estén libres de errores.
- Que todos los caminos lógicos principales deben ejecutarse correctamente en cada módulo de la aplicación.
- Todas las transacciones deben ser probados.
- Todos los tipos de registro de entrada válidos deben ser procesados
- Todos los tipos de registro de entrada inválidos deben ser procesados correctamente
- Códigos de vuelta no nulos.
- Excepciones a tratamiento normal.
- Todas las salidas válidas son procesadas.
- Rasgos de Control son probados y documentados.

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Validar las piezas individuales del software como una unidad independiente.
Estrategia:	<ul style="list-style-type: none"> • Se efectúan para los servicios del negocio y para la lógica en capa Web que tengan complejidad alta. • Generar casos de pruebas necesarios que permitan identificar: <ul style="list-style-type: none"> ○ Que al menos cada sentencia o instrucción del programa se ejecute al menos una vez correctamente. ○ Que cada condición tenga por lo menos una vez un resultado positivo y/o negativo. ○ Que cada bucle del sistema se pueda probar considerando: - ignorar el bucle, pasar una vez, pasar n veces.
Herramienta requeridas: Observaciones	<p>La prueba se realizará por Módulo entendiéndose por tal:</p> <ul style="list-style-type: none"> • Bloque básico de programa • Implementa función independiente y simple • Puede probarse por separado.

12.3.2 Pruebas del sistema

Las pruebas de sistema buscan diferencias entre la solución desarrollada y los requerimientos, enfocándose en la identificación de los errores que se puedan generar entre la especificación funcional y el diseño del sistema, así como, el negocio objeto de la aplicación.

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Validar los requerimientos no funcionales de cada proyecto.
Estrategia:	<ul style="list-style-type: none"> • Realizar pruebas de rendimiento básico. Consiste en probar la aplicación simulando la carga esperada en el entorno de producción. • Realizar las pruebas de concurrencia: verificar el comportamiento de la aplicación en condiciones de sobrecarga de usuarios, que supone permitirá identificar potenciales problemas de rendimiento o cuellos de botella, antes de su pase a producción. • Realizar pruebas de requerimientos no funcionales: Consiste en probar la aplicación con cada uno de los requerimientos no funcionales establecidos en el

	proyecto.
Herramienta requeridas:	<ul style="list-style-type: none"> Identificar posibles cuellos de botella o problemas de rendimiento.
Observaciones:	-

12.3.3 Pruebas de regresión

En esta prueba se valida que el sistema mantenga su correcta funcionalidad después de la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas.

Objetivo de la Prueba:	Validar que el sistema siga funcionando perfectamente después de que las acciones correctivas sean aplicadas.
Estrategia:	<ul style="list-style-type: none"> Repetir las pruebas (unitarias, de integración, funcionales y de carga) que se hicieron antes de corregir defectos o de añadir nuevas funcionalidades, para comprobar que las modificaciones no provocan errores donde antes no los había.
Herramienta requeridas:	<ul style="list-style-type: none"> Utilizar las mismas herramientas usadas para las pruebas según sea el caso.
Observaciones	Los responsables de las Pruebas de Regresión se establecen dependiendo del momento en el que se realicen las modificaciones.

12.3.4 Pruebas funcionales

La prueba funcional es un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:

- Los procesos y reglas de negocio establecidas,
- Que se cumplan los requerimientos funcionales establecidos

En esta prueba se validan los Casos de Uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso.

Objetivo de la Prueba:	Se asegura el trabajo apropiado de los requisitos funcionales, Incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
Estrategia :	<ul style="list-style-type: none"> • Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente: <ul style="list-style-type: none"> ○ Los resultados esperados ocurren cuando se usan datos válidos. ○ Se despliegan mensajes de error cuando se usan datos inválidos. ○ Cada regla de negocio es propiamente aplicada. ○ Realizar set de pruebas de los requerimientos mínimos para el adecuado funcionamiento de la aplicación
Herramientas Requeridas:	-
Observaciones:	-

12.3.5 Pruebas de usabilidad

Las pruebas de usabilidad son una forma de medir que tan bien puede una persona usar un objeto hecho por el hombre, como puede ser una página web, una interfaz de usuario, un documento o un dispositivo.

Las pruebas de usabilidad consisten en seleccionar a un grupo de usuarios de una aplicación y solicitarles que lleven a cabo las tareas para las cuales fue diseñada, en tanto el equipo de diseño, desarrollo y otros involucrados toman nota de la interacción, particularmente de los errores y dificultades con las que se encuentran los usuarios.

No es necesario que se trate de una aplicación completamente terminada, pudiendo tratarse de un prototipo

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Validar el grado de usabilidad empírico del sistema. • El grado de usabilidad se medirá en tres aspectos clave: <ul style="list-style-type: none"> ○ Facilidad de aprendizaje: facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. ○ Flexibilidad: relativa a la variedad de posibilidades con las que el usuario y el
------------------------	---

Estrategia :	<p>sistema pueden intercambiar información.</p> <ul style="list-style-type: none"> ○ Robustez: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. <ul style="list-style-type: none"> • Se usarán cuatro métricas principales para medir la usabilidad del sistema <ul style="list-style-type: none"> ○ Exactitud: Número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados. ○ Tiempo requerido para concluir la actividad. ○ Recuerdo: Qué tanto recuerda el usuario después de un periodo sin usar la aplicación. ○ Respuesta emocional: Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera). • Estas métricas será implementadas para cada uno de los aspectos clave señalados en el objetivo de la prueba. • La forma de evaluación será mediante el uso de encuestas; donde cada pregunta evaluará un aspecto clave de usabilidad y aportará valor a una o varias métricas dentro del aspecto clave evaluado. • Las encuestas se realizarán a los usuarios utilizando los prototipos del sistema; para así poder realizar cambios de forma temprana al diseño de la capa de presentación.
Herramientas Requeridas:	<ul style="list-style-type: none"> • Encuesta • Prototipos del sistema.
Observaciones:	-

12.3.6 Pruebas de aceptación

El objetivo de las pruebas de aceptación es validar que la solución desarrollada cumpla con el funcionamiento esperado y permitir al usuario de dicho sistema determine su aceptación, desde el punto de vista de su funcionalidad y de su rendimiento. Estas pruebas son realizadas por el cliente, donde comprueba que el sistema cumple con lo definido y se obtiene la conformidad del cliente. Esta prueba se realiza mediante el proceso de validación de caja negra.

Estas pruebas corresponden a la ejecución de las siguientes pruebas por parte de los usuarios funcionales o cliente:

- Pruebas Funcionales.
- Pruebas de Usabilidad.
- Pruebas de Configuración

12.4 Matriz de tipificación de pruebas

TIPO DE PRUEBAS	TIPO DE PRUEBA
Pruebas Unitarias	Manuales
Pruebas de Sistema	Manuales
Pruebas de Integración	Manuales
Pruebas de Regresión	Manuales
Pruebas Funcionales	Manuales
Pruebas de Usabilidad	Manuales
Pruebas de Aceptación	Manuales

12.5 Técnicas de ejecución de pruebas

TIPO DE PRUEBAS	TECNICA DE EJECUCIÓN	HERRAMIENTAS A UTILIZAR
Pruebas Unitarias	<p>Las pruebas unitarias normalmente involucra los siguientes pasos:</p> <ol style="list-style-type: none"> Crear los casos de prueba unitarios. Ejecución de los casos de prueba. Reporte de los defectos encontrados según las pruebas. Corrección de la incidencia. <p>Repetición de la prueba.</p>	<ul style="list-style-type: none"> Entorno SDK
Pruebas de Sistema	<p>Las pruebas del sistema tal como están concebidas para el proyecto involucra los siguientes pasos:</p> <ol style="list-style-type: none"> Selección de los casos de uso para los que se probará Carga, volumen, estrés, concurrencia. Estos casos de uso corresponderán al 10% del total de casos de uso. Ejecución de las pruebas de Carga, volumen, estrés, 	<ul style="list-style-type: none"> Televisor Samsung Smart TV

TIPO DE PRUEBAS	TECNICA DE EJECUCIÓN	HERRAMIENTAS A UTILIZAR
	<p>conurrencia mediante la herramienta seleccionada.</p> <p>c. Recopilación de resultados.</p> <p>d. Reporte de los defectos encontrados según las pruebas.</p> <p>e. Corrección de la incidencia.</p> <p>f. Repetición de la prueba.</p>	
Pruebas de Regresión	<p>Las pruebas de regresión tal como están concebidas para el proyecto de notificaciones electrónicas involucra los siguientes pasos:</p> <p>a. Repetición de las pruebas funcionales y de sistema que estén involucradas en los cambios realizados al sistema.</p> <p>b. Recopilación de resultados.</p> <p>c. Reporte de los defectos encontrados según las pruebas.</p> <p>d. Corrección de la incidencia.</p> <p>e. Repetición de la prueba.</p>	<ul style="list-style-type: none"> • Casos de Prueba
Pruebas Funcionales	<p>Las pruebas funcionales normalmente involucra los siguientes pasos:</p> <p>a. Crear los casos de prueba mediante el formato establecido para ellos.</p> <p>b. Ejecución de los casos de prueba con forme las funcionalidades van siendo liberadas para pruebas.</p> <p>c. Reporte de los defectos encontrados según las pruebas.</p> <p>d. Corrección de la incidencia.</p> <p>e. Repetición de la prueba.</p>	<ul style="list-style-type: none"> • Casos de Prueba
Pruebas de Usabilidad	<p>Las pruebas de usabilidad tal como están concebidas involucra los siguientes pasos:</p> <p>a. Elaboración de prototipos.</p> <p>b. Elaboración de encuestas de usabilidad.</p> <p>c. Evaluación de prototipos mediante la aplicación de</p>	<ul style="list-style-type: none"> • Prototipos • Encuesta de usabilidad

TIPO DE PRUEBAS	TECNICA DE EJECUCIÓN	HERRAMIENTAS A UTILIZAR
	encuestas de usabilidad a usuarios comunes. d. Recopilación de datos de la encuesta. e. Compilación y análisis de resultados de encuestas de usabilidad.	
Pruebas de Aceptación	Las pruebas de aceptación normalmente involucra los siguientes pasos: a. Ejecución de una muestra de las pruebas funcionales, de carga. b. Reporte de los defectos encontrados según las pruebas. c. Corrección de la incidencia. d. Repetición de la prueba.	<ul style="list-style-type: none"> • Casos de prueba Funcionales. • Listas de Chequeo.

12.6 Recursos humanos

El recurso humano que debe estar disponible para la ejecución de las pruebas varía de acuerdo al tipo de prueba. En el siguiente cuadro se especifica el tipo de perfil necesario por tipo de prueba.

Los perfiles mencionados no necesariamente corresponden a los enunciados en la metodología de pruebas, ya que allí se mencionan perfiles de apoyo al proceso de pruebas y aquí solo se mencionarán los perfiles que van a ejecutar las pruebas o que intervienen directamente en la prueba.

TIPO DE PRUEBAS	PERFIL DEL RECURSO HUMANO
Pruebas Unitarias	<ul style="list-style-type: none"> • Ingeniero Desarrollador. • Analista de Pruebas.
Pruebas de Sistema	<ul style="list-style-type: none"> • Ingeniero Desarrollador. • Analista de Pruebas.
Pruebas de Integración	<ul style="list-style-type: none"> • Ingeniero Desarrollador. • Analista de Pruebas
Pruebas de Regresión	<ul style="list-style-type: none"> • Ingeniero Desarrollador. • Analista de Pruebas
Pruebas Funcionales	<ul style="list-style-type: none"> • Analista de Pruebas
Pruebas de Usabilidad	<ul style="list-style-type: none"> • Analista de Pruebas • Usuario Funcional.
Pruebas de Aceptación	<ul style="list-style-type: none"> • Analista de Pruebas. • Usuario Funcional.

12.7 Recurso del sistema

A continuación se describen las características de la infraestructura del ambiente de pruebas.

DESCRIPCION	FUNCIONALIDAD	CANTIDAD
Servidor	Servidor web Apache	4
Estaciones de Trabajo	Estación para ejecución de emulador de Samsung SMART TV	4
Software: Instalado y configurado	Samsung SMART SDK 4.0	1
Herramientas de pruebas de sistemas	Consola JavaScript de Google Chrome integrada en SDK de SAMSUNG.	1
Dropbox	Compartición de documentos	1

12.7.1 Configuración del entorno de pruebas

El ambiente de pruebas estará implementado en las instalaciones de la UT y su configuración deberá ser la siguiente.

COMPONENTE	CONFIGURACIÓN	SOFTWARE INSTALADO Y CONFIGURADO	CANTIDAD
Servidor 1	Procesador: Intel Core 2 DUO E6550 Processor2x4MB Cache, 2,33GHz, 2327MHz FSB Memoria: 3GB 1200MHz Tarjeta de Video: Intel GMA 3100 Disco Duro: 2 dd c/u 300GB, 10K RPM Serial-Attach SCSI 3Gbps 3,5 in HotPlug HardDrive Red: WAN Miniport	Windows 7	1
Estaciones de Trabajo	Procesador: Intel Core 2 DUO E6550 Processor2x4MB Cache, 2,33GHz, 2327MHz FSB Memoria: 3GB 1200MHz Tarjeta de Video: Intel GMA 3100 Disco Duro: 2 dd c/u 300GB, 10K RPM Serial-Attach SCSI 3Gbps 3,5 in HotPlug HardDrive Red: WAN Miniport	Windows 7	1

12.8 Criterios de inicio de ejecución

A continuación se señalan las condiciones mínimas que se deben presentar para iniciar la ejecución de las pruebas:

- Se poseen los set de pruebas aprobados con escenarios claros.
- El entorno de pruebas es el adecuado para el tipo de pruebas a iniciar.
- Todos los artefactos requeridos se encuentran disponibles.
- Se recibió la Versión del Software para pruebas con su correspondiente Release Note y Lista de Chequeo cuando esta aplique.
- Todos los recursos humanos y técnicos necesarios se encuentran disponibles.

12.9 Criterios de evaluación

Los criterios de evaluación estarán dados de forma independiente para cada tipo de pruebas; el siguiente cuadro muestra los criterios de evaluación generales de las pruebas ejecutadas.

TIPO DE PRUEBAS	CRITERIOS DE EVALUACION
Pruebas Unitarias	<ul style="list-style-type: none"> • Detectar errores en la ejecución de las pruebas. • El 90% de las pruebas realizadas deben ser exitosas.
Pruebas de Sistema	<ul style="list-style-type: none"> • Detectar errores en la ejecución de las pruebas • Tener en cuenta todos los escenarios posibles. • El 90% de las pruebas realizadas deben ser exitosas.
Pruebas de Regresión	<ul style="list-style-type: none"> • Para realizar esta prueba se debe tomar como base los criterios de aceptación de las pruebas que se volverán a realizar.
Pruebas Funcionales	<ul style="list-style-type: none"> • El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. • Encontrar fallas al ejecutar los diferentes casos de pruebas. • La aplicación cumple con los requerimientos funcionales especificados en la fase de análisis • La aplicación cumple con los requerimientos mínimos para el funcionamiento

Pruebas de Usabilidad	<ul style="list-style-type: none"> • El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. • Se deben incluir los datos de entrada válidos y esperados como no validos e inesperados Encontrar los errores al ejecutar los diferentes casos de pruebas. • La aplicación debe cumplir con los requerimientos funcionales especificados en la fase de análisis. • La aplicación debe cumplir con los requerimientos mínimos para el funcionamiento.
Pruebas de Aceptación	<ul style="list-style-type: none"> • Para realizar esta prueba se debe tomar como base los criterios de aceptación de las pruebas que se volverán a realizar.

Para cada una de las pruebas se tendrá en cuenta:

- **Pruebas Unitarias:** Las pruebas unitarias se evalúan por medio de la siguiente tabla o lista de chequeo.

Elemento a Revisar	SI	NO	No Aplica	Observaciones
¿Se realizaron las Pruebas Unitarias con alguna herramienta especializada?				
¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema?				
¿Existe constancia de la realización de las pruebas mencionadas?				
¿El funcionamiento de la prueba unitaria respeta el diseño establecido?				
¿Existe un manejo de errores adecuado?				
¿Se cumplió con la estrategia de ejecución de la prueba?				

- **Pruebas del Sistema:** El resultado de las pruebas del sistema se verá reflejado en el siguiente informe:

Caso de Uso	<Identificador del Caso de uso>	Descripción del escenario	<Número total de casos de prueba ejecutados de acuerdo al escenario>
Número de pruebas exitosas	<Del total de pruebas ejecutadas, cuantas pruebas fueron exitosas>	Número de pruebas Fallidas	<Del total de pruebas ejecutadas, cuantas pruebas fueron fallidas>
Tiempo máximo de ejecución de una prueba	<Tiempo máximo que duró en ejecución una Prueba>	Tiempo Promedio de ejecución de las pruebas	<Tiempo promedio de ejecución de las pruebas>
Número de peticiones exitosas	<Número de peticiones http exitosas>	Número de Peticiones Fallidas	<Número de peticiones http fallidas>
Número de Errores	<Numero de errores ocurridos durante las pruebas>	Tipo de errores	<Descripción del tipo de errores presentados>
% de Utilización del Procesador	<Porcentaje de consumo de utilización de CPU durante la ejecución de la prueba>	Cantidad de Memoria utilizada	<Cantidad de MB de memoria utilizada en la prueba>
Promedio de bytes enviados	<Promedio de bytes enviados>	Promedio de bytes recibidos	<Promedio de bytes recibidos>

- **Pruebas de Regresión:** El resultado de las pruebas de regresión se verá reflejado de acuerdo a los tipos de pruebas seleccionados.
- **Pruebas Funcionales:** El resultado de las pruebas funcionales se verá reflejado de acuerdo al formato de set de pruebas, ver anexos.
- **Pruebas de interfaz de Usuario:** El resultado de las pruebas de interfaz de usuario se verá reflejado en el siguiente informe o lista de chequeo

Elemento a Revisar	SI	NO	No Aplica	Observaciones
¿Se realizaron las pruebas de interfaz de usuario con alguna herramienta especializada?				
¿Cuál fue el porcentaje de cobertura de la prueba con relación al sistema total?				

¿Existe constancia de la ejecución de las pruebas?				
¿Qué páginas se cubrió con la prueba?				
¿Se estableció un criterio para la ejecución de las pruebas? ¿Cuál?				
¿Se cumplió la estrategia de ejecución de la prueba?				

- **Pruebas de Aceptación:** El resultado de las pruebas de aceptación se verá reflejado de acuerdo a los tipos de pruebas seleccionados.

12.10 Criterios de terminación

A continuación se señalan los criterios de terminación de las pruebas a ejecutar.

- Se ejecutaron todas las pruebas del sistema.
- Todas las pruebas se ejecutaron de acuerdo a los criterios de evaluación.
- Los incidentes encontrados en las pruebas fueron corregidos y probados.

12.11 Criterios de suspensión

Los criterios de suspensión impiden la iniciación y/o continuación de las pruebas ante cualquier situación de imprevisto que hace que la ejecución de las pruebas no logre grados satisfactorios de probabilidad de éxito.

- Después de la instalación y configuración del sistema, se evidencia problemas o situaciones anormales en cualquiera de sus componentes.
- Después de la instalación y configuración del sistema, se evidencia que el ambiente de pruebas no es lo suficientemente estable para la ejecución de las pruebas.
- Discrepancia entre la documentación (Set de Pruebas, Casos de Uso) y el sistema.

13ANEXO 2. DOCUMENTACIÓN PRUEBAS

A continuación se listarán los anexos al plan de pruebas, que básicamente corresponden a todos los documentos, formatos o plantillas que se utilizarán en la especificación, ejecución y documentación de resultados de las pruebas.

13.1.1 Release notes

A continuación se presenta el formato que se utilizará como release notes, el cual deberá acompañar cada una de las versiones entregadas para pruebas.

1. Presentación

- a. Identificador del Release: <Numero de Release>
- b. Descripción del producto:

2. Requerimientos de Hardware, Sistema Operativo y Software Base.

Se deben especificar los requerimientos de Hardware, Sistema Operativo y Software Base que el ambiente de pruebas debe tener instalado y configurado antes de iniciar el proceso de instalación del sistema.

COMPONENTE	REQUERIMIENTO
HARDWARE	•
SISTEMA OPERATIVO	•
SOFTWARE BASE	•

3. Requerimientos del Sistema.

Aquí se incluyen los requerimientos de instalación y configuración del sistema.

4. Características Nuevas (opcional).

Se describen las características nuevas que tiene el release entregado

5. Características Obsoletas (opcional).

Se describen las características obsoletas con relación a una versión anterior del Release.

6. Características Eliminadas (opcional).

Se describen las características eliminadas con relación a una versión anterior del Release.

13.1.2 Casos de pruebas

13.1.2.1 Formato casos de pruebas funcionales

A continuación se presenta el formato que se utilizará como Set de Pruebas funcionales.

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA						
CASO DE PRUEBA No.	<Número del caso de prueba constituido por SP-[número del caso de uso]-[Numero del caso de prueba]>			VERSIÓN DE EJECUCIÓN		<Versión diligenciado por el analista de pruebas en el momento de ejecutarla. Este número se incrementa de 1 en 1>
				FECHA EJECUCIÓN		<Fecha de ejecución diligenciado por el analista de pruebas>
CASO DE USO:	<Identificación del caso de uso objeto de la prueba>			MODULO DEL SISTEMA		<Nombre del módulo al que corresponde el caso de uso objeto de la prueba>
Descripción del caso de prueba:	<Descripción de lo que se pretende probar en el caso de prueba>					
55. CASO DE PRUEBA						
dddd. Precondiciones						
<Lista de precondiciones que deben cumplirse para realizar la prueba>						
eeee. Pasos de la prueba						
<Pasos secuenciales que deben ser ejecutados por el analista de pruebas o usuario, ante el sistema para ejecutar la prueba>						
DATOS DE ENTRADA			RESPUESTA ESPERADA DE LA APLICACIÓN	COINCI DE		RESPUESTA DEL SISTEMA
CAMPO	VALOR	TIPO ESCENARIO		SI	NO	
<Descripción del dato de entrada>	<Valor que debe ser	<Tipo de escenario que	<Respuesta que se espera de la aplicación>			<Respuesta que se obtuvo de la aplicación en el momento de la

	suministrado en la prueba para el dato de entrada>	pretende probarse : Correcto/ Incorrecto>				ejecución de la prueba>
fff. Post condiciones						
<Lista de postcondiciones que deben cumplirse después de realizar la prueba>						
56. RESULTADOS DE LA PRUEBA						
Defectos y desviaciones						Veredicto
<Lista de defectos o desviaciones encontrados por el analista o usuario al ejecutar la prueba>						<input type="checkbox"/> Paso <input checked="" type="checkbox"/> Falló
Observaciones					Probador	
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>						
					Firma: Nombre: Fecha:	

13.1.2.2 Lista de chequeos casos de prueba funcionales

Con el fin de garantizar que los casos de prueba contemplen el 100% de los escenarios a probar para cada caso de uso; en su construcción deberá tenerse en cuenta la siguiente lista de chequeo.

Cada conjunto de casos de prueba para cada caso de uso deberá contemplar:

ELEMENTO DEL CASO DE USO	CASO DE PRUEBA
Datos de entrada	Verificar que los datos de entrada

	cumplan con: <ul style="list-style-type: none"> • Obligatoriedad • Tipo de datos • Longitud • Estructura
Reglas de Negocio	Validar reglas de negocio que afecten los datos de entrada (Dependencia de datos).
	Validar reglas de negocio que afecten los flujos.
Flujos Alternos	Verificar la ejecución de todos los flujos alternos.
Flujos de Excepción	Verificar la ejecución de todos los flujos de Excepción.
Flujo Básico	Verificar la ejecución del flujo básico.
Generalidades:	Los casos de prueba deben especificar exactamente rutas, nombres de archivos, valores para los datos de entrada. Para asegurar que las rutas y nombres de archivos se cumplan; deberá instalarse una árbol de carpetas predefinido en la estación donde se ejecutará la prueba.

13.1.2.3 Encuesta para pruebas de usabilidad

Las pruebas de usabilidad se guiarán por la siguiente estructura de encuesta:

PREGUNTA	CRITERIOS DE EVALUACIÓN
1. ¿Hay términos en idiomas diferentes mezclados?	1 = Se encuentran en todo el sistema 2 = Se encuentra en algunas partes del sistema. 3 = No se encuentran en ninguna parte del sistema.
2. ¿Es simple el vocabulario utilizado?	1 = El vocabulario es demasiado técnico. 2 = El vocabulario presenta algunas dificultades de comprensión. 3 = El vocabulario es

	completamente comprensible.
3. ¿Se proporciona tiempo suficiente para realizar las entradas por teclado?	1 = El tiempo es muy limitado. 2 = El tiempo es limitado para algunas funcionalidades. 3 = El tiempo es completamente suficiente.
4. ¿Hay algún tipo de asistencia para los usuarios que hacen uso del sistema por primera vez?	1 = No existe ninguna ayuda. 2 = Se encuentra ayuda en algunas partes. 3 = Existen ayudas en todo el sistema.
3. ¿El sistema es fácil de operar para alguien que no recibió capacitación en su operación?	1 = El sistema es de difícil comprensión. 2 = El sistema es fácil de operar en algunas de sus funcionalidades. 3 = El sistema es completamente fácil de operar.
6. ¿Se entienden la interfaz y su contenido?	1 = No se entiende su interfaz. 2 = La interfaz se entiende en algunas partes. 3 = La interfaz es completamente entendible.
7. ¿Resulta fácil identificar un objeto o una acción?	1 = Es difícil identificar los objetos o acciones. 2 = Se pueden identificar los objetos y acciones en algunas partes del sistema. 3 = Todos los objetos y acciones son fácilmente identificables.
8. ¿Resulta fácil entender el resultado de una acción?	1 = Los resultados de las acciones no son entendibles. 2 = Los resultados de las acciones son entendibles en algunas partes o la mayor parte del sistema.

	<p>3 = Todos los resultados de las acciones son entendibles.</p>
9. ¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?	<p>1 = La interfaz es difícil de usar.</p> <p>2 = La interfaz es difícil de usar en algunas partes del sistema.</p> <p>3 = La interfaz es completamente sencilla de usar.</p>
10. ¿Son apropiados los mensajes presentado por el sistema?	<p>1 = Los mensajes non son apropiados.</p> <p>2 = Los mensajes son apropiados en algunas partes del sistema.</p> <p>3 = Todos los mensajes son apropiados y fáciles de comprender.</p>
11. ¿Actúa el sistema en la prevención de errores?	<p>1 = El sistema no previene errores del usuario.</p> <p>2 = El sistema previene algunos o la mayoría de los errores del usuario.</p> <p>3 = El sistema previene cualquier error que pueda cometer el usuario.</p>
12. ¿El sistema informa claramente sobre los errores presentados?	<p>1 = El sistema no informa de manera adecuada sobre los errores cometidos.</p> <p>2 = El sistema informa de manera adecuada algunos o la mayoría de los errores cometidos por el usuario.</p> <p>3 = El sistema informa de forma adecuada todos los errores cometidos por el usuario.</p>
13. ¿Se utiliza mensajes y textos descriptivos?	<p>1 = Los mensajes de texto no son descriptivos.</p> <p>2 = La mayoría de los textos son descriptivos o fáciles de interpretar</p> <p>3 = Todos los textos son descriptivos o fáciles de</p>

	interpretar.
14. ¿Permite una cómoda navegación dentro del producto y una fácil salida de éste?	<p>1 = La navegación no es sencilla.</p> <p>2 = La navegación presenta algunas dificultades.</p> <p>3 = La navegación es sencilla, requiere de pocos vínculos para acceder a las funcionalidades del sistema.</p>
13. ¿Se permite al usuario personalizar la interfaz?	<p>1 = La interfaz no es personalizable.</p> <p>2 = La interfaz es personalizable con algunas restricciones.</p> <p>3 = La interfaz es completamente personalizable.</p>
16. ¿Se proporciona información visual de dónde está el usuario, qué está haciendo y qué puede hacer a continuación?	<p>1 = No se presenta ninguna información visual ni otro tipo de ayuda.</p> <p>2 = Presenta ayudas en algunas partes del sistema.</p> <p>3 = Las ayudas son apropiadas y están distribuidas a lo largo del sistema.</p>
17. ¿Existe atajos del teclado bien hechos?	<p>1 = No existen atajos por teclado.</p> <p>2 = Existen algunos atajos por teclado.</p> <p>3 = Todas las opciones presentan atajos por teclado.</p>
18. ¿Se presenta al usuario la información que sólo necesita?	<p>1 = La información presentada es más de la que necesita y tiende a ser confusa.</p> <p>2 = En algunas partes se presenta mayor información a la necesaria.</p> <p>3 = La información es estrictamente la necesaria según el perfil.</p>

13.1.2.4 Formato de casos de pruebas técnicas

A continuación se presenta el formato que se utilizará para documentar las pruebas técnicas; estas pruebas serán documentadas conforme avance el desarrollo de la solución y se tengan versiones liberadas sobre las que se aplicarán estas pruebas.

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
Tipo de Prueba:	<Descripción del tipo de prueba: Carga, Volumen, Estrés, ETC>	Código de la prueba	<Codificación de la prueba>
Descripción de la prueba:	<Descripción del objetivo de la prueba>		
Versión de Ejecución	<Versión o iteración de ejecución de la prueba>	Fecha de Ejecución	<Fecha de ejecución en formato AAAA/MM/DD diligenciado por el analista de pruebas al momento de su ejecución>
1. Prerrequisitos de la prueba			
<Lista de los prerrequisitos a tener en cuenta antes de ejecutar la prueba>			
2. Insumos de la prueba			
<Lista de Insumos necesarios para ejecutar la prueba>			
3. Lista de chequeo de la prueba			
Pasos a Seguir	Prueba satisfactoria		Observaciones
	SI	NO	
<Pasos numerados en orden lógico para la ejecución de la prueba>			
4. Resultados de la prueba			
Defectos y desviaciones			Veredicto
<Lista de defectos o desviaciones encontrados por el analista o usuario al ejecutar la prueba>			<input type="checkbox"/> Paso <input type="checkbox"/> Falló
Observaciones			Probador
<Observaciones generales del analista o usuario sobre la ejecución de la prueba>			
			Firma:

	Nombre: Fecha:
--	-------------------

13.1.2.5 Matriz Casos de Uso VS Casos de Prueba Funcionales

A continuación se presenta el formato de matriz de trazabilidad que se llevara para asegurar que sean probados todos los aspectos definidos dentro de los casos de uso.

Caso de Uso	Aspecto a Evaluar	Caso de Prueba
<Identificación del caso de uso>	1. Datos Entrada	
	Obligatoriedad	<Identificación del caso de prueba que evalúa Obligatoriedad>
	Longitud	<Identificación del caso de prueba que evalúa Longitud>
	Tipo de Dato	<Identificación del caso de prueba que evalúa Tipo de dato>
	2. Reglas de Negocios Relacionadas con datos de entrada	
	<Lista de casos de prueba>	<Identificación del caso de prueba que evalúa la regla de negocio>
	3. Reglas de Negocios	
	<Lista de casos de prueba>	<Identificación del caso de prueba que evalúa la regla de negocio>
	4. Flujos de Excepción	
	<Lista de flujos de excepción>	<Identificación del caso de prueba que evalúa los flujos de excepción>
	5. Flujos Alternos	
	<Lista de casos de flujos alternos>	<Identificación del caso de prueba que evalúa los flujos alternos.>
6. Flujo Básico	<Identificación del caso de prueba que evalúa el flujos básico.>	

13.1.2.6 Matriz Requerimientos No Funcionales VS Casos de Prueba Técnicos

A continuación se presenta el formato de matriz de trazabilidad que se llevara para asegurar que sean probados todos los aspectos técnicos de la solución; en esta matriz se registrará cada caso de prueba técnico y requerimiento no funcional que será verificado. Esta matriz será diligenciada en la medida que las pruebas técnicas sean diseñadas.

