



Universidad de las Ciencias
Informáticas

Facultad 5

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
EN CIENCIAS INFORMÁTICAS

*Herramienta para la gestión de la
programación televisiva en la Dirección de
Comunicación Institucional*

AUTOR:

JUAN JOSÉ PÉREZ HERNÁNDEZ

TUTOR:

ING. YASELF MACHADO TUGORES

Co – TUTOR:

ING. LEONARKIS ACOSTA SUAREZ

LA HABANA, 2016

“AÑO 57 DE LA REVOLUCIÓN”

Declaración de Autoría

Declaración de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2016.

Juan José Pérez Hernández

Autor

Ing. Yaself Machado Tugores

Tutor

Ing. Leonarkis Acosta Suarez

Co-Tutor

Datos de los tutores

Datos de los tutores.

Generales del tutor:

Nombre y apellidos: Yaself Machado Tugores

Especialidad: Ingeniero en Ciencias Informáticas

Síntesis del tutor:

Graduado de Ingeniero en Ciencias Informáticas en el año 2008 se desempeñó por 5 años como líder de proyecto y desarrollador del Proyecto Laboratorios Virtuales de la Facultad 5. Se desempeña como Jefe de Departamento en la Dirección de Comunicación Institucional.

Correo electrónico: ytugores@uci.cu

Teléfono de contacto: (+53) 8358874

Generales del Co – Tutor:

Recién graduado de Ingeniero en Ciencias Informáticas (2015). Ha participado en eventos nacionales y ha publicado varios artículos. Vinculado al Departamento Desarrollo de Componentes del Centro de Entornos Interactivos 3D.

Correo electrónico: leonarkis@uci.cu

Dedicatoria

Dedicatoria.

Este trabajo de diploma va dedicado especialmente a mis abuelos que fueron los que lucharon conmigo hasta el cansancio, en los tiempos buenos y en los malos. Especialmente a mi abuelo Pepín que aunque no lo tengo vivo siempre quiso que fuera ingeniero y más en esta rama.

Dedicarle a mi madre gran parte de este trabajo porque siempre estuvo conmigo aunque la distancia nos haya separado siempre estuvo al tanto de mí, de mis momentos malos en la universidad como en los buenos. A mi padre que también formo parte de mi desarrollo como profesional. A mi familia en general por cuidarme, por criarme, por ser parte de ellos, me siento orgulloso de ustedes, los quiero.

Agradecimientos

Agradecimientos.

Primeramente agradecerle a mi familia por todo el apoyo que me han dado desde que entré a esta universidad. Agradecerles a mis padres por confiar en mí y saber que yo podía llegar al final. A mi hermana Lyen, eres un sol, te debo mucho mi hermana y te lo agradezco, te amo. Agradecerle a mi otra familia que fueron parte de mi desarrollo como profesional y siempre estuvieron ahí cuando más lo necesitaba, estoy hablando de Luis D Fernández, Daysi Fernández, Yindra Ruiz, mis ahijados, etc. Agradecerle también infinitamente a Mercedes Campello, quien fue guía y profesora desde que era chiquitico así, jejeje. A julio Padrón, amigo, quien confió en mí y desde la secundaria me daba mis buenos consejos como estudiante hasta la decisión de optar por la Vocacional y llegar hasta aquí. A mis amigos y hermano de gtno, los quiero muchos, Mayito, Dailet, etc, son muchos.

A Tía Lim que siempre, siempre estuvo conmigo al lado mío. Me cuidó, me trató bien, me enseñó lo que era unidad en la familia, llevarse bien, en fin me trató como a un hijo que nunca tuvo. A Jorge Luis, a ti que me supiste guiar por el buen camino en el poco tiempo que te conozco, me demostraste honestidad, valor de hombre, humanidad, solidaridad y muchas más cosas, nunca olvidaré la frase tuya “qué clase de bobería tiene papo, ahí díoss” no peles más brother, te quiero.

Agradecimientos

A Nauria que estuvo conmigo en los buenos y malos momentos de mi vida, cuando más lo necesitaba ahí estabas tú, gracias, demostrarme un cariño profundo y sincero de amor de verdad. mil gracias BB.

Al piquete loco de 1er año, cuando entre aquí, sé que muchos se han ido y otros se han quedado, pero están en mi mente y en mi corazón. Frank loco, te quiero con la vida.

Agradecerle infinitamente todo el esfuerzo y dedicación que me han dado, mis tutores, el Ing Yaself Machado y el Ing Leonarkis Acosta, los tendré siempre presente.

Un agradecimiento especial para mi tía Loly, mi primo Hector que me ayudaron y apoyaron, buenos momentos pasamos, jajaja, tiempos aquellos.

No quiero dejar atrás aquellas personas que confiaron en mí que me ayudaron, que me dieron su granito de arena en todo este proceso. Rene te amo mi hermano te llevo siempre en mi corazón, eres un brother para mí.

Bueno ahora sí, LOS QUIERO A TODOS.

Resumen

Resumen

La gestión de los programas a transmitir por parte de la televisión universitaria es un proceso que se realiza semanalmente en el departamento de comunicación de la Dirección de Comunicación Institucional. Dicho proceso se hace engorroso, ya que intervienen múltiples variables que pueden provocar errores, tanto humanos como tecnológicos. Este trabajo propone el desarrollo de una herramienta web para la gestión de la programación en los canales universitarios, que minimice el tiempo de ejecución y los errores que puedan surgir en los procesos vinculados a la planificación y control de la programación televisiva.

Para ello se realizó un estudio de las principales tecnologías y las herramientas utilizadas en el mundo para el desarrollo de este tipo de sistemas. Esta investigación se basó en bibliografías actualizadas referentes a este campo, basándose en las buenas prácticas de la programación orientada a objetos (POO), las nuevas tendencias en el diseño de las interfaces, así como la elección de la metodología adecuada para el desarrollo de este trabajo.

Palabras claves: aplicación web, desarrollo de software, gestión, programación televisiva.



ÍNDICE

índice	VII
Índice de Tablas.....	IX
Introducción	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1: Introducción	5
1.2: Conceptos asociados al problema.....	5
1.3: Aplicaciones informáticas vinculadas al campo de acción de este trabajo.....	6
1.4: Metodologías.....	8
1.4.1: Metodología de desarrollo de software. Su selección.....	8
1.5: Tecnologías.....	11
1.6: Bibliotecas para Graficar.....	15
1.7: Marco de Trabajo.....	16
1.8: Entorno de Desarrollo Integrado (IDE).....	17
1.9: Lenguaje de Modelado.....	18
1.10: Herramienta CASE. Visual Paradigm.....	18
1.11: Descripción del proceso	19
Conclusiones parciales del capítulo.....	20
CAPÍTULO 2: MODELACIÓN DEL SISTEMA.....	21
2.1: Requisitos del Sistema.....	21
2.1.1 Requisitos Funcionales.....	21
2.1.2 Requisitos no Funcionales.....	23
2.3 Plan de Iteraciones	35
2.4: Diseño del Sistema.....	36
2.4.1 Tarjetas CRC.....	36
2.5: Arquitectura de Software.....	39
2.5.1 Modelo-Vista-Controlador	40

Índice

2.6 Patrones de Diseño	41
2.6.1 Patrones GRASP	42
Conclusiones Parciales	44
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....	45
3.1 Tareas de Ingeniería	45
3.1 Diseños de Casos de Pruebas	47
3.2 Pruebas de rendimiento.....	49
3.3 Resultados esperados.....	50
CONCLUSIONES GENERALES.....	53
Recomendaciones.....	54
Bibliografía.....	55
Anexos.....	57

ÍNDICE DE TABLAS

Tabla 1: HU1 Autenticar Usuario.....	25
Tabla 2: HU2 Modificar rol de Usuario.	26
Tabla 3: HU3 Eliminar usuario.....	27
Tabla 4: HU7 Insertar Canal.	27
Tabla 5: HU4 Listar usuario.....	28
Tabla 6: HU6 Buscar usuario.....	29
Tabla 7: HU15 Listar Programación.....	29
Tabla 8: HU12 Insertar programación	30
Tabla 9: HU13 Modificar programación.....	31
Tabla 10: HU14 Eliminar programación	31
Tabla 11: HU16 Mostrar datos detallados de cada programación.....	32
Tabla 12: HU17 Mostrar cartelera por día	32
Tabla 13: HU18 Exportar en formato PLS la lista de reproducción dada la fecha.	33
Tabla 14: Estimación de esfuerzos por HU.	34
Tabla 15: Plan de iteraciones.	35
Tabla 16: Plan de entrega.	36
Tabla 17: Tarjeta CRC UsuarioController	38
Tabla 18: Tarjeta CRC CanalesController.	38
Tabla 19: Tarjeta CRC ProgramaciónController	38
Tabla 20: Tarjeta CRC CategoríaController.	38
Tabla 21: Tarjeta CRC ReportesController.....	39
Tabla 22: Tarjeta CRC ArchivosAdicionalesController.	39
Tabla 23: Tarea de Ingeniería 1 de la HU1.	45
Tabla 24: Tarea de Ingeniería 1 de la HU2.	45
Tabla 25: Tarea de Ingeniería 1 de la HU3.	46
Tabla 26: Tarea de Ingeniería 1 de la HU5.	46
Tabla 27: Tarea de Ingeniería1 de la HU6.	46
Tabla 28: Tarea de Ingeniería1 de la HU7.	47
Tabla 29: Mostrar datos del usuario.	47
Tabla 30: Listar usuarios.	47
Tabla 31: Autenticar usuario.	48
Tabla 32: Buscar usuario.....	48
Tabla 33: Insertar canal.	49
Tabla 34: Modificar canal.....	49
Tabla 35: Caso 1.....	50
Tabla 36: Caso 2.....	50
Tabla 37: Anexo 1.1: HU8 Modificar canal.	57
Tabla 38: Anexo 1.2: HU9 Eliminar canal.	57
Tabla 39: Anexo 1.3: HU10 Listar canales.	58
Tabla 40: Anexo 1.4: HU11 Mostrar datos detallados de cada canal.	58
Tabla 41: Anexo 1.5: HU19 Insertar categoría.....	59
Tabla 42: Anexo 1.6: HU20 Modificar categoría.	59
Tabla 43: Anexo 1.7: HU21 Eliminar categoría.....	60
Tabla 44: Anexo 1.8: HU22 Listar categoría.	60

Índice

Tabla 45: Anexo 1.9: HU23 Mostrar datos detallados de cada categoría.	61
Tabla 46: Anexo 1.10: HU24 Listar estadísticas de programación en formato PDF.	62
Tabla 47: Anexo 1.11: HU25 Insertar archivo adicional.	62
Tabla 48: Anexo 1.12: HU26 Modificar archivo adicional.	63
Tabla 49: Anexo 1.13: HU27 Eliminar archivo adicional.	63
Tabla 50: Anexo 1.14: HU28 Listar archivo adicional.	64
Tabla 51: Anexo 1.15: HU29 Mostrar datos detallados de cada archivo adicional.	64
Tabla 52: Anexo 1.16: HU30 Insertar notificación.	65
Tabla 53: Anexo 1.17: HU31 Modificar notificación.	66
Tabla 54: Anexo 1.18: HU32 Eliminar notificación.	66
Tabla 55: Anexo 1.19: HU33 Listar Notificaciones realizadas.	67
Tabla 56: Anexo 1.20: HU34 Mostrar datos detallados de cada notificación.	67
Tabla 57: Anexo 1.21: HU35 Enviar notificación vía correo.	68
Tabla 58: Anexo 1.22: HU36 Buscar cartelera dada la fecha.	68
Tabla 59: Anexo 1.23: HU37 Convertir video en formato mp4.	69
Tabla 60: Anexo 1.24: HU38 Mostrar gráficamente la cantidad de programación por categoría.	69
Tabla 61: Anexo 1.25: HU39 Mostrar alerta de poco espacio en disco.	70
Tabla 62: Anexo 2.1 Tarea de Ingeniería 1 de la HU13.	71
Tabla 63: Anexo 2.1 Tarea de Ingeniería 1 de la HU8.	71
Tabla 64: Anexo 2.1 Tarea de Ingeniería 1 de la HU20.	71

Índice

Índice de Figuras

Figura 1: Arquitectura en la capa Modelo.....	40
Figura 2: Arquitectura capa Vista.....	40
Figura 3: Arquitectura en la capa Controlador.....	41
Figura 4: Patrón MVC en Yii (17).....	41
Figura 5: Diagrama de clases que representa el Patrón Experto.....	42
Figura 6: Diagrama de clases que representa el Patrón Controlador.....	43
Figura 7: Diagrama de clase que representa el Patrón Creador.....	43

INTRODUCCIÓN

Hoy en día el desarrollo acelerado de la tecnología ha permitido grandes avances en el desarrollo económico de un país, así como en la vida en general a través del uso del internet. La industria de software como el mercado del mismo hace que cada día crezca más la productividad para abastecer la demanda que se genera a diario.

El uso de la informática en la actividad diaria del hombre ha permitido que los grandes volúmenes de información que se manejan hayan sido digitalizados y que muchos de los procesos actuales se ejecuten de forma automática.

En el año 2002 es creada por Fidel Castro Ruz (Presidente, en aquel entonces, de la República de Cuba), la Universidad de las Ciencias Informáticas (UCI) con el objetivo de formar profesionales comprometidos a la creación de lo que se le llamaría, la industria de software en Cuba.

Dentro de las infraestructuras creadas en la universidad se encuentra la Dirección de Comunicación Institucional que tiene como principal misión definir y hacer cumplir los procesos comunicativos en la universidad. Además, cuenta con un nodo de transmisiones desde el cual se brinda el servicio de transmisión de la señal televisiva, tanto de los canales nacionales como de los canales de producción propia de la dirección. En estos momentos se mantienen en transmisión dos canales que responden a varios objetivos dentro de la institución, el Canal Académico y el UCItve que se encargan de brindar una programación vinculada a temas docentes y otra al entretenimiento respectivamente.

Es de vital importancia destacar los procesos que se realizan en dicha dirección, pero más bien se centraría en el departamento de comunicación que es el encargado de planificar los programas que se transmitirán en el día o en la semana según se organice.

Para llevar a cabo el proceso de la programación se diseña una lista de programas a transmitir dependiendo del horario del día, por parte del Departamento de Comunicación, luego de aprobada la misma es enviada para que se ejecute el proceso de transmisión. Esta lista llegada al transmisor ya contiene incluidos los archivos de video a transmitir ubicados en una carpeta compartida en el servidor. Sin embargo, debido a que suelen ocurrir errores humanos a la hora de planificar dicha lista, entran en juego algunas deficiencias detectadas que se mencionan a continuación:

Introducción

- La necesidad de recalcular el tiempo de duración de la programación a partir de cada programa que va a salir al aire para poder reajustarse a las horas de transmisión del día.
- Desorganización al elaborar la lista de programas diarios y semanales producto de que los archivos están en diferentes lugares de los servidores e intervienen en el proceso más de una persona.
- A menudo falla la copia de archivos de video por parte del planificador en los servidores para dejar en punta a la hora de transmitirlo.
- Suele perderse algunos listados de programaciones ya realizadas a lo largo del tiempo.
- Es complicado realizar un análisis de los programas transmitidos pues se archivan los listados de programación en más de un fichero de texto y la recolección de esos datos se hace compleja.

Derivado de la situación problemática expuesta anteriormente, se plantea como **problema a resolver**: ¿Cómo facilitar una mejor organización y control en los procesos que se realizan en la gestión de la programación televisiva en la Dirección de Comunicación Institucional?

A partir de este problema se plantea como **objeto de estudio** el proceso de gestión de la programación televisiva, así como se propone como **campo de acción** herramientas web para la gestión de la programación televisiva. El **objetivo general** para darle solución al problema formulado anteriormente es:

Desarrollar una herramienta web para la gestión de la programación televisiva que facilite una mejor organización y control de la programación televisiva en la Dirección de Comunicación Institucional.

De esta manera, y para dar cumplimiento al objetivo trazado se definieron las siguientes tareas a cumplir:

- Elaborar el marco teórico de la investigación.
- Revisar la bibliografía en lo referente al estado del arte del tema, así como las herramientas, estándares y métodos para el desarrollo de aplicaciones de este tipo en la Dirección de Comunicación Institucional.
- Identificar los requisitos de software para determinar las funcionalidades y restricciones que la herramienta deba tener.
- Diseñar la arquitectura del software.
- Implementar las funcionalidades identificadas.
- Ejecutar las pruebas funcionales y de rendimiento a la herramienta.

Introducción

- Valorar los resultados obtenidos a partir de las pruebas realizadas.
- Desplegar la herramienta implementada.

Durante el proceso de investigación se espera como resultado:

- ❖ Aplicación web que permita llevar a cabo la gestión de la programación televisiva en la Dirección de Comunicación Institucional.

Para poder dar cumplimiento a los objetivos trazados se llevó a cabo una investigación en la que se utilizaron algunos de los métodos científicos existentes.

Los métodos utilizados en la investigación se explican a continuación:

Métodos Teóricos:

Analítico-Sintético: Se utilizó para analizar todas las citas, documentos y libros escogidos en soporte digital con el objetivo de lograr un mayor alcance a los procesos que se realizan a la hora de realizar una herramienta de gestión.

Métodos-Empíricos:

Observación: Este método permitió la búsqueda de información precisa acerca de algunos mecanismos utilizados para desarrollar los procesos de gestión en plataformas libres y dar paso a la implementación de aplicaciones de este tipo.

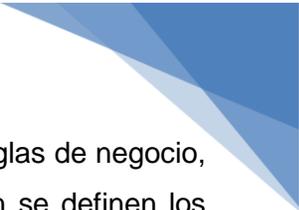
Entrevista: Se realizó una entrevista al personal del departamento de programación para tener una mayor información sobre la situación problemática antes mencionada, así como al director de dicha dirección para conocer más del proceso.

Para un mejor entendimiento de la investigación, se propone la siguiente estructura del documento:

Capítulo 1: Fundamentación Teórica.

Se refiere en este capítulo al estado del arte referente a la investigación. Se especifican los conceptos fundamentales para dar un mejor entendimiento de la situación problemática planteada. Se exponen las tecnologías que se utilizan y su justificación de uso, así como la metodología a emplear.

Capítulo 2: Modelación del Sistema.



Introducción

Se analiza de forma detallada una descripción de la solución planteada, sus reglas de negocio, requisitos funcionales y no funcionales que debe cumplir la solución. También se definen los patrones de diseño a cumplir, así como la arquitectura a utilizar a lo largo del desarrollo para dar paso a la solución de la propuesta realizada.

Capítulo 3: Implementación y Prueba.

En este capítulo se aborda la implementación de la herramienta, las pruebas realizadas a la misma, así como los diagramas de despliegue y de componente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1: Introducción

En este capítulo se pretende abordar sobre los conceptos y aspectos generales a tratar sobre la gestión de la programación televisiva en la Dirección de Comunicación Institucional, una descripción del estado del arte del tema a tratar, así como las herramientas, lenguajes, tecnologías y metodología del desarrollo del software a utilizar, con el objetivo de seleccionar la mejor opción para el desarrollo.

1.2: Conceptos asociados al problema.

Las herramientas informáticas cumplen hoy en día una destacada labor en cuanto a resolver miles de necesidades que se presentan a diario en la vida de cada ser humano, dichas necesidades están encaminadas a que se pueda lograr una mejor eficiencia y productividad en los trabajos que se realizan. Para ello es de destacar que existen conceptos y definiciones asociados a los problemas que los informáticos deben de resolver, en este caso refiriéndose a una herramienta de gestión de los programas de televisión.

La Gestión se puede asociar a cómo llevar adelante una iniciativa o un proyecto, es la parte administrativa y organizacional de una entidad, en términos generales *“es la capacidad de una institución u organización para definir, alcanzar y evaluar sus propósitos administrando los recursos disponibles”* (1). Sin embargo, es válido aclarar que la gestión de un proceso es punto fundamental para la creación de herramientas informáticas, ya que estas son hechas para solucionar o facilitar los procesos en un área determinada de una empresa u organismo, o para mejorar el trabajo al personal que lo desee. Es aquí donde cumplen un papel excepcional las herramientas informáticas donde son utilizadas con el fin de facilitar la realización de una tarea mecánica que requiere de una aplicación correcta. En el ámbito de la informática estas herramientas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillas (2). En esta solución se propone el factor de gestión para facilitar mejor el trabajo en el departamento de comunicación de la Dirección de Comunicación Institucional.

Por otra parte, la programación televisiva es un término importante a tocar en la realización de este trabajo. La misma no es más que una lista ordenada de programas a transmitir en una jornada dada, creada por el departamento de programación del canal que se esté transmitiendo. Es también, *“la declaración previa de lo que emitirá la televisión en un determinado momento”* (3). Este término se ha extendido por todo el mundo y es normalmente utilizado para dirigir la labor de organizar y agrupar esas emisiones, no sólo de comunicar esa

Capítulo 1: Fundamentación Teórica

organización. Pero además, también se designa con este término al resultado final de lo que realmente se emite, es decir, la emisión (4).

1.3: Aplicaciones informáticas vinculadas al campo de acción de este trabajo.

El desarrollo de aplicaciones web para la gestión de la programación de un canal de televisión ha propiciado la disminución de la carga del trabajo manual, contribuyendo también a aumentar la eficiencia en el trabajo del personal del departamento de comunicación.

Se realizó un estudio de las principales herramientas existentes para determinar las características y funcionalidades que debe cumplir la aplicación a realizar. A continuación se mencionan las ventajas y limitantes que poseen estas herramientas.

- ❖ **VSNCREATV** : Es una aplicación para la planificación de la programación y gestión televisiva.
 - Permite realizar la gestión de los procesos que se realizan en los departamentos de publicidad, producción y emisión para planificar la transmisión de materiales.
 - Posee la característica de trabajar desde cualquier ordenador siempre y cuando este se encuentre instalado en el mismo (5).

Es un software ideal para el trabajo con la gestión de la programación televisiva, pero al ser desarrollado en una plataforma Desktop (plataforma de escritorio a nivel de ventanas) lo hace vulnerable a virus que el ordenador cliente pudiera tener. Al ser un software privado presenta las desventajas propias de este tipo de sistemas en cuanto a problemas de pagos de licencia y activaciones. Es por estas razones que no es ideal para utilizarlo como medio en la Dirección de Comunicación Institucional.

Al realizar el estudio previo de esta herramienta permitió tener la idea esencial de realizar un software para la gestión de la programación televisiva, teniendo en cuenta la secuencia de las operaciones a realizar para la confección de la cartelera y de la programación diaria.

- ❖ **inTV**: Es un Sistema de Gestión Televisiva que entre sus principales características se destacan:
 - Controlar los contratos de materiales fílmicos, desde un contacto inicial hasta su recepción y posterior liquidación.
 - Mantener una base de datos de las películas o series contratadas.
 - Realizar procesos automatizados como la planificación y ejecución de la continuidad diaria.
 - Permitir el registro, seguimiento y control de órdenes de publicidad.
 - Contiene un sistema de reportes para llevar a cabo comerciales No Transmitidos.

Capítulo 1: Fundamentación Teórica

- Ventas por programa, por línea de producto, por cliente, etc.
- Plataforma privada por lo que su uso es limitado. (6)

El uso de esta herramienta facilita la gestión de la programación televisiva en un canal, pero más bien se encarga de controlar los contratos de los materiales que los clientes o las productoras gestionan y facilita el proceso de venta y publicidad de materiales fílmicos. La misma posee la limitante de ser privada y desarrollada en plataforma Desktop. Además, no realiza la conversión de videos en diversos formatos, lo cual es importante a la hora de crear la programación televisiva en el departamento de programación.

Esta herramienta permitió elaborar una secuencia lógica de la programación, así como vincular la lista de programas a transmitir en un solo fichero. Todo lo antes mencionado demuestra que es un software ideal para utilizarlo en la confección de la programación en el departamento de comunicación de la Dirección de Comunicación Institucional, pero el hecho de ser privada limita su uso.

- ❖ **Virtual Power Video Server:** Es un completo sistema de administración de materiales audiovisuales, que permite la emisión de contenido de video.
 - Posee un potente planificador de emisiones, para programar días y horas exactas para la reproducción de contenido audiovisual.
 - Es un software de automatización de televisión.
 - Soluciona todo el proceso de continuidad de una estación o canal de tv.
 - Funciona sobre equipos standard.
 - Posee un potente programador, basado en reglas configuradas por el usuario, que permite generar relojes de programación para cada hora, cada día de la semana, generar la parrilla de programación siempre será muy fácil.
 - Es desarrollado para ejecutarse en plataformas Desktop.
 - Es software privativo, depende de actualizaciones constantes de su licencia. (7)

Esta potente herramienta administradora de materiales audiovisuales cumple con muchos requisitos para poder ser utilizada para la gestión de la programación televisiva en el canal, como lo es en el proceso de la producción audiovisual, transmite material programado, pero el hecho de que sea privada y de no mantener un flujo de trabajo como el que se realiza en el departamento de comunicación de la Dirección de Comunicación Institucional deja de cumplir con las expectativas trazadas para ser utilizada.

Las funcionalidades de esta aplicación permitieron desarrollar el generador de ficheros en formato PLS (extensión para las listas de reproducción) el cual es una funcionalidad fundamental para facilitar el trabajo a la hora de transmitir la programación.

Capítulo 1: Fundamentación Teórica

Después del análisis realizado de las diferentes herramientas se puede concluir que todas parten de la planificación de una programación televisiva garantizando una correcta elaboración de la lista de programas a transmitir, pero el hecho de que sean privadas y sus funcionalidades no estén acorde a los procesos que se realizan en la Dirección de Comunicación Institucional se decide desarrollar una aplicación que cumpla con dichos requerimientos.

1.4: Metodologías.

Al concluir el análisis de los sistemas existentes, la continuidad de este trabajo es la selección de la metodología y un análisis de las tecnologías usadas en el mismo. *“Probablemente la informática sea la herramienta más poderosa que el hombre haya tenido jamás en sus manos, y actualmente interviene de forma directa, o indirecta en las actividades diarias del ser humano”.* (8)

El presente trabajo se desarrollará sobre el Sistema Operativo Linux, por políticas migratorias de la Universidad, y es por ello que se centra en tecnologías y herramientas de software libre.

En los epígrafes siguientes se identifican algunas herramientas y tecnologías actuales de las cuales se describe su utilización en el desarrollo de la aplicación, así como la arquitectura escogida en este caso.

1.4.1: Metodología de desarrollo de software. Su selección.

La gran variedad de herramientas y metodologías capaces de brindar soluciones eficaces presentan un gran dilema a los desarrolladores de hoy, en la forma de emplear la metodología y las herramientas adecuadas, ya que las mismas determinan la calidad que tendrá un producto determinado, pues definen qué papel debe ejercer cada miembro dentro de un equipo de desarrollo, las actividades que se deben de cumplir, los artefactos a generarse en cada tarea, así como el registro de cada detalle de la información a generar.

Las metodologías y técnicas existentes para el desarrollo de productos de software, son analizadas y definidas en la arquitectura de cada proyecto. Se conocen dos tipos definidos de metodologías, las robustas o tradicionales y las ágiles. La primera está pensada para usarse en proyectos de gran escala y con muchos integrantes, por lo que están diseñadas para soportar la gran complejidad técnica de estos, en los que se dificulta la sincronización y comunicación entre sus miembros. Aunque las metodologías tradicionales pueden aplicarse a proyectos pequeños la gran cantidad de roles, tareas, artefactos y documentación que generan no la hacen adecuada para este tipo de empresa. En cambio las metodologías ágiles presentan diferencias substanciales con respecto a la anterior, pues se enfocan menos en la generación de artefactos y documentos y más en el código que se genera (8). Además ofrecen

Capítulo 1: Fundamentación Teórica

una buena solución para proyectos variables donde los requisitos no se conocen con exactitud.

Estas características de las metodologías ágiles la hacen un candidato perfecto para el desarrollo que se propone, ya que se cuenta con un proyecto pequeño (1 persona) y un tiempo de desarrollo corto. Sobre estas últimas se han desarrollado varios ejemplos con características similares, de los cuales tres de los más importantes fueron objeto de estudio con el fin de seleccionar la metodología que guiará el proceso de desarrollo de la aplicación propuesta. A continuación se muestra dicho análisis:

- **SCRUM** : Es una metodología desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle (desarrolladores de la metodología SCRUM), la cual define un marco para la gestión de proyectos y goza de gran popularidad debido a características como su utilización en proyectos con un rápido cambio de requisitos, el desarrollo de reuniones a lo largo del ciclo de vida del proyecto, así como un desarrollo basado en iteraciones, con una duración de 30 días cada una. Cada una de estas iteraciones se entiende como un incremento ejecutable que se muestra al cliente. Scrum se valora para desarrollarse en entornos complejos, donde los requisitos son cambiables consecutivamente y poco definidos, de tal manera que a la hora de analizarlo no cumple para ser escogido como metodología de desarrollo para este trabajo, ya que los requisitos están bien definidos y no sufrirán cambios a corto plazo. (9)
- **ASD**: Es una metodología impulsada por Jim Highsmith (ingeniero en software y autor del libro “Software Development Methodology”) cuyas principales características son ser iterativa, orientada a los componentes de software, tolerante a los cambios y de desarrollo ágil. Tiene un ciclo de vida dividido en tres fases principales: especulación, colaboración y aprendizaje. En la primera fase se inicia el proyecto y se planifican las características del software, en la segunda fase se desarrollan las características y finalmente en la tercera fase se revisa su calidad, y se entrega al cliente. Se puede destacar que es adaptable a sistemas complejos, pero se recomienda su utilización en proyectos medianos o pequeños sin ninguna complejidad por medio. (8)
- **XP**: Disciplina de desarrollo de software creada por Kent Beck (ingeniero en software y creador de la metodología), centrada principalmente en la simplicidad, la comunicación y el reciclado continuo de código, preocupándose por el aprendizaje de los desarrolladores y propiciando la existencia de un buen clima de trabajo. Sus objetivos son muy simples pues *“están centrados en la satisfacción del cliente y potenciar al máximo el trabajo en equipo”* (10). Además es una metodología que presenta seis fases principales en su ciclo de vida muy bien definidas, Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento, Muerte del

Capítulo 1: Fundamentación Teórica

Proyecto, además cuenta con una gran bibliografía, proyectos y expertos que la desarrollan. Es la metodología más destacada en los procesos ágiles de desarrollo de software, pues se hace más énfasis en la adaptabilidad que en la previsibilidad y se promueve el trabajo en equipos, aspecto que se considera fundamental para el aprendizaje de los desarrolladores del mismo (10).

Entre las principales características de la metodología XP se encuentran (10):

- Permite introducir nuevos requisitos o cambiar los anteriores ágilmente.
- Adecuada para proyectos pequeños y medianos.
- Adecuada para proyectos con alto riesgo.
- Su ciclo de vida es iterativo e incremental.
- Cada iteración dura entre una y tres semanas.
- No se genera demasiada documentación acerca del diseño o la planificación.

Al ser XP una metodología que lleva a la culminación de un producto con mayor rapidez, donde el cliente forma parte del equipo de trabajo y es la que mejor se ajusta a las necesidades de este proyecto en cuanto a recursos técnicos, humanos y tiempo de desarrollo, permite dar paso a resultados ventajosos e importantes para desarrollar proyectos de pequeña envergadura, un producto de alta calidad y en el mínimo tiempo disponible. Por todo lo antes expuesto hacen de XP la metodología escogida para desarrollar el software propuesto.

Siendo los roles a desempeñar los siguientes: (10)

Administrador (programador): Su misión es administrar el ambiente de programación. Realiza las actividades de Configurar el Ambiente de Programación.

Programador: Su misión es ser responsable de implementar el código para dar soporte a los requisitos del cliente. Realiza las actividades de Definir el Estimado de la Tarea, Estimado del Requisito del Cliente, entre otros.

Probador: Su misión es ayudar al cliente a definir y escribir pruebas de aceptación para los requisitos. Realiza las actividades de Correr las Pruebas del cliente, Automatizar las Pruebas del cliente y Configurar el Ambiente del Probador.

Cliente: Su misión es ser responsable de definir cuál es el producto correcto a construir, determinar las características del mismo y asegurarse de que el producto realmente satisface sus requerimientos. Realiza las actividades de ajustar las iteraciones de las actividades, definir la iteración, definir la visión del documento, escribir los requisitos, entre otros, generando los artefactos de Requisitos, Prueba del Cliente, Plan de Iteraciones, Plan de Entrega, entre otros.

Capítulo 1: Fundamentación Teórica

Encargado de Seguimiento: Su misión es medir y comunicar el progreso del proyecto. Realiza las actividades de seguimiento del progreso de la iteración y seguimiento del progreso del release del producto.

Entrenador (coaching): Su misión es ayudar a mantener la disciplina y aprendizaje del equipo. Realiza las actividades de Explicar Proceso, Mejorar las habilidades del Equipo, Resolver Conflictos, entre otros.

Luego de seleccionar la metodología se procede al análisis de las herramientas necesarias para apoyar el diseño e implementación de la aplicación, teniendo en cuenta las características del entorno donde se desplegará la aplicación.

1.5: Tecnologías.

A continuación se realizará un análisis de las tecnologías que responden al tipo de aplicación que se desea desarrollar.

- **PHP 5.6.15**

PHP es un lenguaje interpretado, ampliamente usado, diseñado especialmente para desarrollo web (Ciberaula, 2010). Además, es un lenguaje de programación de estilo clásico con variables, sentencias condicionales, bucles, funciones, etc. Es un lenguaje multiplataforma. Presenta gran capacidad de conexión con la mayoría de los manejadores de base de datos y destaca su conectividad con MySQL. Posee gran capacidad de expandir su potencial utilizando la enorme cantidad de módulos llamados extensiones (Álvarez, 2009). Puede leer y manipular datos desde diversas fuentes. Permite las técnicas de programación orientada a objetos, así como crear los formularios para la web (Marcías, 2007).

Esta versión de PHP posee una serie de nuevas características como son:

- Expresiones en constantes. Previamente a PHP 5.6, era imposible asignar a una constante un valor que no fuera estático. A partir de ahora, se podrán usar expresiones.
- Uso de funciones Variádicas. Ahora se pueden implementar usando el operador (...), en lugar de depender de `func_get_args()`.
- Desempaquetar argumentos. El operador (...) también sirve para desempaquetar argumentos en Arrays y objetos Traversable.
- Nuevo operador de exponenciación. Mediante el nuevo operador ** y su asignación corta ** = podemos realizar exponenciación, algo que antes dependía de la función pow.

Capítulo 1: Fundamentación Teórica

- Ampliación del operador use. El operador use ahora permitirá importar funciones y constantes mediante use const y use function (antes solo se podían importar clases).
- Los cierres ahora soportan \$this.
- Se han mejorado los mensajes de error de análisis y las advertencias de argumentos incompatibles.

Por sus características y el buen desempeño que presenta a la hora de utilizarse en la creación de aplicaciones web, es seleccionado para desarrollar este trabajo, ya que es fácil de manipular y abunda su presencia en las mayorías de los servidores en el mundo, incluyendo donde se implementa la herramienta a desarrollar.

▪ PostgreSQL

PostgreSQL es un excelente sistema gestor de bases de datos relacional, su código se encuentra bajo licencia BSB (*Berkeley Software Distribution*), la cual lo hace totalmente libre. Es ejecutado en varios sistemas operativos como son: Linux, Windows y hasta Mac OS. Este gestor de base de datos es adaptable a cualquier necesidad que el cliente posea y presenta una variada documentación en cualquier idioma. Posee también características avanzadas como el trabajo con llaves foráneas, vistas, integridad transaccional, control de concurrencia y disparadores. Por sus características PostgreSQL se hace una de las más potentes del mercado actual (11).

▪ MySQL 5.5.32

MySQL es un sistema de gestión de base de datos relacional capaz de distribuir una gran cantidad de datos. Es soportado por muchos lenguajes de programación que se usan en la actualidad y utiliza el lenguaje de consulta estructurado SQL (12).

MySQL en su diseño, su principal objetivo es la velocidad, tanto al conectarse al servidor como al responder a alguna consulta, consume muy pocos recursos tanto de Unidad de Proceso Central (CPU) como de memoria aleatoria por lo que tiene mayor rendimiento. Tiene mayores utilidades de administración, copia de seguridad, recuperación de errores y no suele perder información ni corromper datos.

Valoración de su uso en el trabajo.

- Veloz en la accesibilidad a los datos introducidos en las distintas tablas independientes que forman las bases de datos de este lenguaje.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios.

Capítulo 1: Fundamentación Teórica

- Facilidad de configuración e instalación.
- Es software libre, por tanto, se puede obtener y trabajar sin preocupaciones, utiliza la licencia GPL (Licencia de código abierto) para definir que se puede o no hacer.
- Es multiplataforma.

Por su gran aceptabilidad en el mercado, el buen manejo de la información y las características antes mencionadas es seleccionado para utilizarse en el desarrollo de este trabajo.

▪ **JavaScript 1.8.5**

JavaScript es un lenguaje de programación que permite crear interactividad a las páginas web. Es ideal para agregar ciertas funciones rápidas a una página web. Es un lenguaje muy potente de alto nivel. Es cierto que no se puede hacer nada directamente al nivel de la máquina, pero es capaz de trabajar con muchas propiedades de los exploradores web, páginas web y, a veces, con el propio sistema donde se ejecuta el explorador. No necesita una fase de compilación como C y el explorador no ha de cargar ninguna máquina virtual para ejecutar el código (Peterson, 2014). Solo hay que crear el código y cargarlo. También se trabaja con propiedades como las funciones del constructor o la estratificación en jerarquías. Con lo que hay más opciones para utilizar el código (13).

Se selecciona este lenguaje por su sencillez y su compatibilidad con la mayoría de los navegadores. Además tiene la ventaja que mediante él se puede realizar gran parte de las funciones del cliente, lo que reduce la carga en el servidor.

▪ **JQuery 2.0.2**

JQuery es una biblioteca de JavaScript, que puede ser usada en proyectos libres y privativos, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código (Bibeault, 2013). Logra simplificar la manera de interactuar con los documentos HTML (lenguaje de marcas de hipertexto). Permite manejar eventos, desarrollar animaciones, y agregar interacción con otras tecnologías a páginas web. Entre las ventajas de utilizar esta librería se encuentran la reducción y reutilización de código, código más conciso y de fácil lectura, desarrollo de componentes dinámicos que enriquecen las aplicaciones, fácil manejo de eventos y animaciones.

Por lo ante expuesto, se decide usar esta biblioteca de JavaScript para la implementación del proyecto, dando paso a la utilización de sus funcionalidades, las cuales le dan un aspecto más atractivo y funcional al software.

Capítulo 1: Fundamentación Teórica

▪ HTML 5

HTML es un lenguaje de marcas hipertextuales que está diseñado para estructurar textos para generar páginas web. El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo (Harris, 2010). Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, vídeo, etc.

Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y es un metalenguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones (13).

“HTML 5 es una colección de estándares para el diseño y desarrollo de páginas web” (14). Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella. Permite una mayor interacción de las páginas web con contenidos de reproducción digital (videos, audio, juegos, entre otros) y facilidad a la hora de maquetar una página web.

Entre sus nuevas características se tiene que:

- Incorpora etiquetas (canvas 2D y 3D, audio, video) con codecs (codificador - decodificador) para mostrar los contenidos multimedia.
- Etiquetas para manejar grandes conjuntos de datos. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios. Nuevos tipos de datos y facilidades para validar el contenido sin JavaScript.

Son muchas las razones para utilizar este lenguaje como parte de la aplicación por su gran importancia, ya que es el encargado de visualizar toda la estructura de la aplicación.

▪ Hojas de Estilo en Cascada versión 3 (CSS 3).

Las Hojas de Estilos en Cascada (CSS) describe una serie de estilos y personalizaciones que usted le puede atribuir a su página web, da la forma, tamaño y color de cómo se va a mostrar un documento en la pantalla, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de estilos le ofrece al desarrollador el control total sobre los formatos de sus documentos.

“CSS se utiliza para dar estilo a documentos HTML y XML y separa el contenido de la presentación” (15).

Capítulo 1: Fundamentación Teórica

Algunos navegadores no soportan esta tecnología, sólo algunos como Netscape versiones de la 4 en adelante y los de Microsoft a partir de la versión 3. Es claro destacar que con el avance de estas tecnologías se ha llegado a un consenso de poder utilizar los estándares posibles para que no ocurran desastres como, por ejemplo, la página no carga las imágenes, no tomó completo el formato que tenía original, etc (15).

Para la propuesta de solución se usará el CSS3 ya que a diferencia de versiones anteriores, este permite:

- Nuevas propiedades para crear diseños de texto más atractivos.
- Bordes formados por imágenes con border-image, y sombras con box-shadow, entre otras opciones.
- La posibilidad de cambiar el tamaño y posición de una figura con la utilización de diversos métodos y la propiedad transform.
- Transiciones que permiten cambiar el estilo de un objeto de manera progresiva.

1.6: Bibliotecas para Graficar.

EjsChart

Es un sistema para mostrar datos en la web en forma de gráfica. Desarrollado totalmente con JavaScript. Proporciona las herramientas que se necesitan para publicar sus datos de forma rápida y en una variedad de formatos, no los publica en tiempo real, lo que hay que recargar la página de su navegador para mostrarse los nuevos datos. Posee variedades de control de evento, tales como, el control del ratón, alfombrillas de eventos, seguimiento y acontecimientos claves, zoom, desplazamiento, y retículos. Todos estos aumentan la interactividad y la experiencia del usuario en la web de gráficos a un nuevo nivel. Tiene como desventaja que es lenta pues hay un alto consumo por parte del JavaScript (16).

Highcharts 4.1.9

Highcharts es una biblioteca de gráficos desarrollada en JavaScript que ofrece la manera de agregar gráficos a la aplicación web que estás desarrollando. Funciona en todos los navegadores y se adapta correctamente a todos los dispositivos. Su funcionamiento se basa solo en las tecnologías de navegadores nativos y no requiere componentes secundarios para su funcionamiento (17).

Highcharts también tiene como característica que en ninguna de las licencias, gratuita o no, se le permite descargar el código fuente y hacer sus propias ediciones, se puede utilizar en cualquier aplicación siempre y cuando tengan un destino no comercial. Pero en esta versión highcharts la liberó gratuitamente, pues fue la única versión de esta biblioteca libre para cualquier uso que se le pudiera dar.

Capítulo 1: Fundamentación Teórica

Entre sus principales características se encuentra la fácil manipulación de las gráficas, ya sea de barra, pastel, y las variadas animaciones que presentan las mismas, al mostrar en tiempo real los datos obtenidos en una de sus variadas representaciones. Posee una simple configuración de su sintaxis lo que posibilita facilidad al desarrollador poder utilizarla. A través de una API completa, puedes añadir, eliminar y modificar series y puntos de ejes en cualquier momento después de creación de los gráficos (17).

Por todo lo antes mencionado de esta librería (HightChart) se puede llegar a la conclusión de que es elegida para ser utilizada en este trabajo, ya que posee la liberación de su licencia conservando el código fuente de la misma.

1.7: Marco de Trabajo

Symfony

Es un marco de trabajo diseñado para optimizar el desarrollo de aplicaciones web. Cuenta con muchas herramientas y clases orientadas a disminuir el tiempo de desarrollo de una aplicación web compleja. Symfony está desarrollada con PHP 5 y ha sido probada en numerosos proyectos reales aquí en nuestra universidad (18).

Yii 1.1.14

Yii es un framework para PHP de alto rendimiento, basado en componentes web para desarrollar aplicaciones de gran envergadura. Permite desarrollar una máxima reusabilidad en la programación web y puede acelerar significativamente el proceso de desarrollo de un software.

Es un framework genérico para programar en la web que puede ser utilizado para desarrollar virtualmente cualquier tipo de aplicaciones, es liviano y está equipado con las soluciones más sofisticadas. Está especialmente diseñado para trabajar con aplicaciones web de tráfico alto, como portales, foros, sistemas de administración de contenidos (CMS) y sistemas de comercio electrónico (19).

Yii implementa el patrón de diseño Modelo Vista Controlador (Model-View-Controller, MVC) el cual es utilizado ampliamente en la programación web.

Este patrón tiene como objetivo separar la lógica del negocio de las interfaces de usuario para que los desarrolladores puedan modificar cada parte más fácilmente sin afectar a la otra (19).

Algunas de las principales características del framework son:

- Fácil de instalación.
- Liviano de correr y equipado con soluciones de cacheo sofisticadas.
- Permite la reutilización.
- Ha sido probado en Apache HTTP server en Windows y en Linux. Pero también deberá correr en otros servidores web o plataformas que soporten PHP 5.

Capítulo 1: Fundamentación Teórica

- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.
- Posee entradas de formularios y validación.
- También incorpora soporte de autenticación vía role-based access control (RBAC) jerárquico.

Se elige Yii como marco de trabajo del lado del servidor dado que se destaca por encima de otros framework y cumple con los requisitos previos para el desarrollo de una aplicación de gestión y planificación, utilizando PHP.

Bootstrap 2.3

Bootstrap es el framework de Twitter que permite crear interfaces web con CSS y JavaScript, este simplifica la creación de diseños web combinando CSS con JavaScript. Es esencial para la maquetación de un sitio web brindando la posibilidad de utilizar plantillas predefinidas y adaptables al gusto del desarrollador.

Algunas características que ofrece el framework: (20)

- Es un framework compatible con la mayoría de los navegadores web.
- Las aplicaciones están disponibles para todos los dispositivos y navegadores.
- Proporciona uniformidad a la hora de navegar, ya que posee un conjunto de hojas de estilo para todos los componentes de HTML.
- Proveen elementos adicionales de interfaz de usuario como diálogos, tooltips y carruseles, etc.
- Es cómodo su trabajo y rápido su desarrollo.

Se elige Bootstrap como marco de trabajo del lado del cliente, dado que se destaca su desarrollo ágil utilizando HTML5, CCS3, JavaScript.

1.8: Entorno de Desarrollo Integrado (IDE).

NetBeans es una aplicación gratuita de licencia libre y código abierto creada con el propósito de hacer programas, es multilenguaje y posee un sinnúmero de herramientas que nos facilitan el trabajo. Tiene soporte para crear interfaces gráficas de forma visual, crear aplicaciones para móviles y desarrollar aplicaciones web con elevado contenido gráfico (21).

Por estas razones y por las diferentes vistas de los datos y herramientas útiles para la creación de aplicaciones es seleccionada para el desarrollo del trabajo.

Capítulo 1: Fundamentación Teórica

1.9: Lenguaje de Modelado.

Utilizar un lenguaje de modelado nos permite verificar efectivamente el funcionamiento adecuado que el modelo debe poseer, pues existen menos medios de verificación, lo cual hace al modelo más sencillo que la propia programación.

El Lenguaje Unificado de Modelado (UML) no es un lenguaje de programación, se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información que contiene un sistema de software orientado a objetos (22).

Por lo que su propósito consiste en elaborar los artefactos de un sistema a través de las distintas etapas de su ciclo de vida, principalmente durante el análisis y el diseño del mismo. Es, sin duda, genial para desarrollar diagramas en proyectos realizados con la metodología XP.

Algunas características de UML son: (23)

- Mayor rigor en la especificación, construyéndose así modelos precisos, no ambiguos y completos.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa.
- Modela estructuras complejas.
- Permite realizar una verificación y validación del modelo realizado.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Permite visualizar de forma gráfica un sistema de forma tal que otro lo puede entender.

1.10: Herramienta CASE. Visual Paradigm

Las herramientas CASE representan una forma que permite modelar los procesos de negocios de las empresas y desarrollar los sistemas de información gerenciales. Son una aplicación de métodos y técnicas a través de los cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, procedimientos y sus respectivas documentaciones (24).

Con la aplicación de las herramientas CASE se puede:

- Permitir al administrador de un proyecto informático, llevar adelante el proyecto de forma eficaz y eficiente.
- Mejora la productividad en el desarrollo y mantenimiento del software.
- Aumenta la calidad del software.
- Mejora el tiempo y costo de desarrollo de los sistemas informáticos.

Capítulo 1: Fundamentación Teórica

- Mejora la planificación de un proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestiona todas las fases de desarrollo de un software.
- Facilita el uso de distintas metodologías de software.

Visual Paradigm

Visual Paradigm es una herramienta UML profesional con características gráficas muy cómodas que soporta el ciclo de vida completo del desarrollo de software (18): análisis y diseño orientado a objetos (OO), construcción, pruebas y despliegue.

Esta herramienta es de gran ayuda para la construcción de una aplicación con alta calidad y a un menor costo. La misma está diseñada para una amplia gama de usuarios, incluidos los ingenieros de software, analistas de sistemas y analistas de negocio, o para cualquiera que esté interesado en la construcción de forma fiable a gran escala de sistemas de software, utilizando un enfoque orientado a objetos (25).

Se caracteriza por ser software libre, ser multiplataforma (Windows y Linux), ser de diseño centrado en casos de uso y enfocado al negocio lo que genera un software de mayor calidad, ser fácil de instalar y actualizar, así como que es de licencia gratuita.

Debido a todo lo antes mencionado, se puede concluir que Visual Paradigm se escogió por su integración con varios IDE's de desarrollo como Eclipse y NetBeans, además de su robustez, usabilidad, accesibilidad, sencillez y portabilidad.

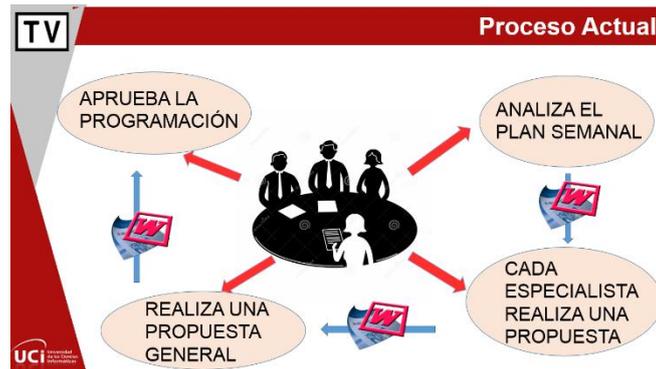
La herramienta escogida tras una amplia labor investigativa fue Visual Paradigm, ya que es una de las herramientas CASE más completas que existe en el mercado actualmente.

1.11: Descripción del proceso

En el departamento de comunicación se realiza la confección de la programación televisiva, en este proceso intervienen un grupo de especialistas que tienen la responsabilidad de realizar una propuesta de cada una de las secciones que atienden, el jefe de grupo es el encargado de verificar la correcta elaboración de cada propuesta y confeccionar la programación general para su transmisión.

Capítulo 1: Fundamentación Teórica

A continuación se describe este proceso detalladamente:



1. Se analiza el plan semanal que consiste en definir el tema con que va a estar relacionada la cartelera de la semana, teniendo en cuenta las efemérides y fechas conmemorativas.
2. La programación está estructurada por secciones y cada especialista del grupo de programación es responsable de definir los programas a transmitir en cada una de las secciones en función del tema seleccionado.
3. Luego de tener definidos los programas el jefe de grupo es el encargado de unificar en una propuesta de cartelera la programación para su posterior aprobación.
 - 3.1 Si la misma es aprobada se archiva en la carpeta correspondiente en el servidor para su transmisión.
 - 3.2 En caso de no ser aprobada es reevaluada y en muchas ocasiones comienza el proceso descrito anteriormente desde el principio.

Conclusiones parciales del capítulo.

Enfocando este trabajo a las necesidades que posee la Dirección de Comunicación Institucional de la Universidad, a partir de la evidente existencia de un problema, se lleva a cabo la realización de un previo estudio de las principales herramientas y tecnologías más usadas en el mundo en la realización de una aplicación web. El estudio realizado del marco teórico fundamenta la necesidad de la utilización de la herramienta informática propuesta, además de las tecnologías y herramientas que se utilizarán. De igual manera la investigación sobre los principales conceptos que se manejan, permite lograr una mejor comprensión del negocio y la comunicación con el cliente. Todo lo planteado anteriormente sienta las bases para el futuro desarrollo de la aplicación.

CAPÍTULO 2: MODELACIÓN DEL SISTEMA

Introducción

Este capítulo estará dirigido a describir la propuesta de solución de la herramienta a desarrollar. Se profundizará en la descripción de los requisitos funcionales y no funcionales, así como en las fases de Exploración, Planificación e Iteración de la metodología escogida, de donde se da paso a la confección de las Historias de Usuario (HU). Se describirán las funcionalidades de la herramienta, además se especificará el plan de iteraciones y el plan de entrega. De igual manera se detallarán y analizarán los patrones de diseño a utilizar.

Atendiendo al estudio realizado en el capítulo anterior se define la siguiente **propuesta de solución**:

ProgramTV será una aplicación web que propone desde la creación de la programación televisiva hasta la generación de la cartelera, será capaz de gestionar todos los procesos que se realizan a la hora de crear la programación televisiva. La generación de ficheros en formato PDF es otra de las funcionalidades principales, además, brindará la posibilidad de mostrar estadísticas de las programaciones realizadas. También propone la creación de los canales televisivos para luego vincularlos con la programación que se realice para cada uno de ellos. Tendrá incluido un convertidor de video a formato mp4, permitiendo tener la lista de reproducción en un formato único y evitar errores de codecs en las computadoras destinadas a la transmisión. El generador gráfico para mostrar estadísticas por categorías de programaciones realizadas, permitirá al programador encargado de confeccionar la programación televisiva, una serie de datos cuantitativos que le permitirán realizar un análisis detallado y lograr así una programación variada de acuerdo al plan propuesto.

2.1: Requisitos del Sistema

Un requisito es una función detallada y formal de un sistema. Es una declaración abstracta de un alto nivel de un servicio que debe de brindar el sistema. Son las exigencias que quedan plasmada en un documento formal o contrato (26).

En el sub-epígrafe 2.2.1 se describe detalladamente mediante Historias de Usuario (HU) como lo establece la metodología a utilizar.

2.1.1 Requisitos Funcionales

RF1 Autenticar usuario (usuario, contraseña).

RF2 Modificar rol de Usuario (rol).

RF3 Eliminar usuario.

Capítulo 2: Modelación del Sistema

RF4 Listar usuarios

RF5 Mostrar datos del usuario.

RF6 Buscar usuario (solapín, nombre, apellidos, rol).

RF7 Insertar canal (nombre, hora apertura, hora cierre, estado).

RF8 Modificar canal (nombre, hora apertura, hora cierre, estado).

RF9 Eliminar canal.

RF10 Listar canales.

RF11 Mostrar datos detallados de cada canal.

RF12 Insertar programación (día, programa, título, duración, sinopsis, canal, estado).

RF13 Modificar programación (día, programa, título, duración, sinopsis, canal, estado).

RF14 Eliminar programación.

RF15 Listar programación.

RF16 Mostrar datos detallados de cada programación.

RF17 Mostrar cartelera por día.

RF18 Exportar en formato PLS (Archivo de lista de reproducción) la lista de reproducción de una fecha dada.

RF19 Insertar categoría (nombre).

RF20 Modificar categoría (nombre).

RF21 Eliminar categoría.

RF22 Listar categorías.

RF23 Mostrar datos detallados de cada categoría.

RF24 Listar estadística en formato pdf de programación.

RF25 Insertar Archivo Adicional (nombre, video).

RF26 Modificar Archivo Adicional (nombre, video).

RF27 Eliminar Archivo Adicional.

RF28 Listar Archivo Adicional.

Capítulo 2: Modelación del Sistema

RF29 Mostrar dato detallado de cada Archivo Adicional.

RF30 Insertar notificación (nombre, descripción).

RF31 Modificar notificación (nombre, descripción, estado).

RF32 Eliminar notificación.

RF33 Listar notificación realizado.

RF34 Mostrar datos detallados de cada notificación.

RF35 Enviar notificación vía email.

RF36 Buscar cartelera dada la fecha.

RF37 Convertir videos en formato mp4.

RF38 Mostrar gráficamente la cantidad de programación por categoría

RF39 Mostrar alerta de poco espacio en el disco.

2.1.2 Requisitos no Funcionales.

Los requisitos no funcionales son aquellas cualidades y atributos de calidad que el producto debe de tener. A continuación se exponen las mismas.

Usabilidad

- La aplicación se rige por el diseño de la estrategia marcaría XABAL creada en la universidad, acorde con los intereses del cliente.
- Debe presentar una navegabilidad sencilla, en la cual los usuarios puedan distinguir fácilmente los botones y enlaces de cada página, de forma tal que los usuarios puedan aprender su funcionamiento en menos tiempo posible.
- Los enlaces y botones deben de estar asociados con las operaciones que realizan.

Confiabilidad

- La información en la herramienta está protegida de acceso no autorizado.
- El usuario debe de autenticarse con sus datos para acceder a la herramienta.
- La aplicación debe mostrar un mensaje de error de autenticación en caso que ocurra.

Restricciones del Diseño

- Lenguaje de programación PHP integrado con el IDE de desarrollo NetBeans.
- Se debe de realizar validaciones de las entradas de los datos para que el usuario cometa el mínimo error posible, mostrando un mensaje de alerta.

Capítulo 2: Modelación del Sistema

- Hacer uso del estándar de codificación que propone Yii.

Integridad

- Se debe a la corrección de los datos en la base de datos, garantizando la calidad de los mismos.
- Debe garantizar que una vez eliminado o actualizado los datos, se hayan modificados los asociados a él en la base de datos.

Rendimiento

- La aplicación debe ejecutarse con el mínimo de recursos para el cual fue prevista.
- El tiempo de respuesta de las peticiones no debe exceder los 37 segundos.

Requisitos de Software

-Para el cliente:

- Utilizar navegador (Firefox 35 en adelante, Google Chrome 38 en adelante).
- Sistema Operativo multiplataforma.
- Tener instalado media player classic.

-Para el servidor:

- Requiere de Apache 2 o Nginx 1.8, como servidores web.
- Sistema Operativo Linux o Windows.
- Servidor de Base de Datos MySQL.
- Servidor con módulo PHP 5.6 disponible e instalar ffmpeg.

Requisitos de Hardware

-Para el cliente:

- Se requiere una pc de 512 MB de RAM como mínimo, 1.8 GHz de velocidad del procesador, tarjeta de red incluida LAN 100Mbps, disco duro de 40 GB.

-Para el servidor:

- Capacidad de disco duro de 3 TB, con un mínimo de 4 GB de memoria RAM, y un procesador como mínimo Core i5 a 3.0 GHz de velocidad, tarjeta de red 100Mbps LAN como mínimo.

Seguridad

- El sistema deberá garantizar el acceso controlado a la información.
- La autenticación de usuarios debe realizarse utilizando certificado SSL.

Capítulo 2: Modelación del Sistema

Interfaz

- El diseño no puede contener animaciones exageradas que afecten el rendimiento de la aplicación.
- Debe de interactuar lo mejor posible con el usuario, sin perder al mismo durante su navegabilidad.

2.2: Exploración

Es la fase donde se identifica las Historias de Usuario (HU), pues son las formas en que se especifican los requisitos del sistema.

En esta etapa los clientes definen sus necesidades a través de las HU y los desarrolladores se familiarizan con las herramientas y tecnologías que se utilizarán en la realización del proyecto. Dependiendo del tamaño del proyecto y la experiencia que tengan los programadores con las tecnologías esta etapa toma de pocas semanas a pocos meses.

2.2.1: Historias de Usuario

Las HU son consideradas una técnica que se utiliza en la metodología XP para especificar los requisitos de software. Tienen la misma finalidad que los casos de usos (CU), pero con la diferencia de que contienen algunas líneas escritas por el cliente en un lenguaje no técnico. Están elaboradas en forma de tabla, la cual describe las características que la aplicación debe tener. Las HU están desglosadas por tareas de programación para que los programadores puedan implementar durante una iteración.

Como resultado de la fase Exploración, en las siguientes tablas se encuentran detalladas algunas HU por su importancia definida por el cliente, de un total de 39.

Tabla 1: HU1 Autenticar Usuario.

Historia de Usuario	
Número: 1	Usuarios: Todos
Nombre de Historia de Usuario: Autenticar Usuario	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.2 semanas
Descripción: La herramienta debe permitir la autenticación de usuario para acceder al entorno de trabajo del mismo mediante usuario y contraseña. Se debe poner en práctica el Protocolo Ligero de Acceso a Directorios (LDAP) de la universidad, para la verificación de usuarios en el dominio. Automáticamente al usuario autenticado se le asigna el rol de usuario.	

Capítulo 2: Modelación del Sistema

Prototipo:

Tabla 2: HU2 Modificar rol de Usuario.

Historia de Usuario	
Número: 2	Usuarios: Administrador, Programador
Nombre de Historia de Usuario: Modificar rol de Usuario	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir modificar el rol del usuario. Para ello se accede a la Administración de Usuarios localizada en el menú principal, luego se da clic en la imagen que aparece con el nombre de Administrar Usuarios y aparecerá la lista de los usuarios insertados en la herramienta, después se da clic al botón Editar correspondiente al usuario que se quiera modificar su rol y se selecciona el nuevo rol y se da clic al botón cambiar rol.</p>	
Prototipo:	

Capítulo 2: Modelación del Sistema

Tabla 3: HU3 Eliminar usuario.

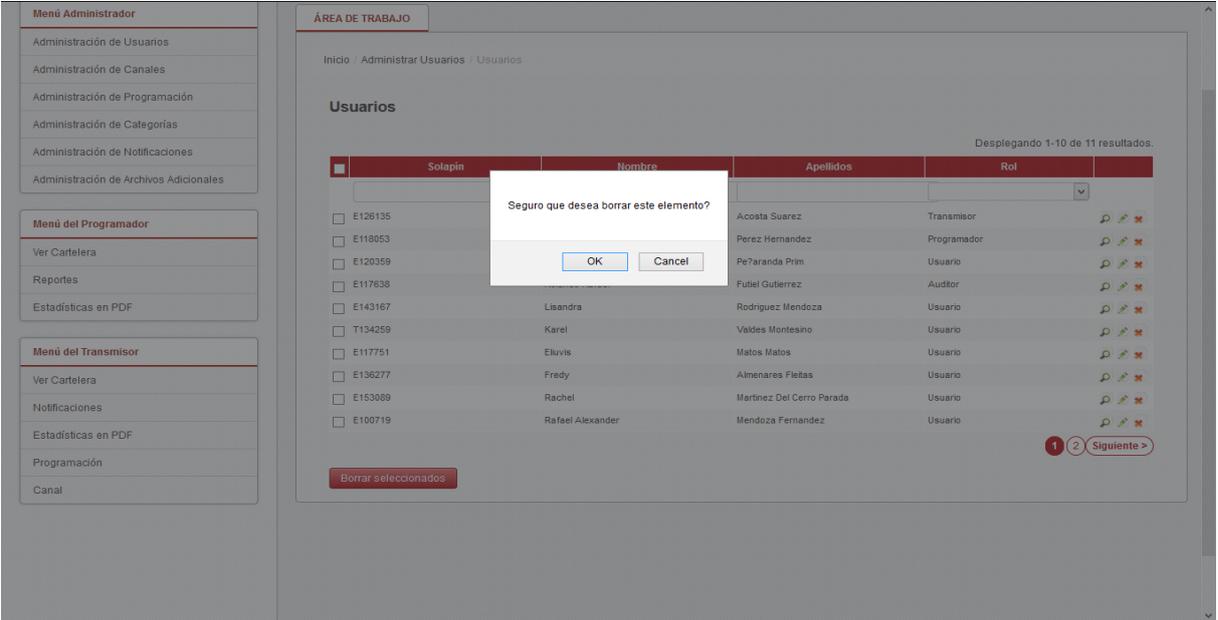
Historia de Usuario	
Número: 3	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Eliminar usuario.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir la eliminación de uno o varios usuarios. Para ello se listan todos los usuarios en la vista de administración, se selecciona uno o varios usuarios que se quieran eliminar, dando clic en el botón eliminar y luego dar aceptar en el mensaje que aparece (¿Seguro que desea borrar este elemento?) para eliminarlo.</p>	
Prototipo	
	

Tabla 4: HU7 Insertar Canal.

Historia de Usuario	
Número: 7	Usuarios: Administrador, Programador
Nombre de Historia de Usuario: Inserta Canal	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.3 semanas
<p>Descripción: La herramienta debe permitir insertar un nuevo canal a la lista de canales. Para ello debe de ir a Administración de canales, dar clic en el botón agregar un canal (+) e introducir los datos correspondientes. Una vez creado el canal se muestra un cartel diciendo (“Canal Creado Satisfactoriamente”).</p>	

Capítulo 2: Modelación del Sistema

Prototipo:

AREA DE TRABAJO

Inicio / Canales / Crear Canal

Crear un Canal

Nombre *

Hora Apertura *

Hora Cierre *

Estado * Sin Transmitir ▾

Tabla 5: HU4 Listar usuario.

Historia de Usuario	
Número: 4	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Listar usuario.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
Descripción: La herramienta debe permitir listar todos los usuarios. Para ello se presiona la opción Administrar usuarios y de esa forma se listan en la parte derecha de la página.	

Prototipo

Administración de Canales

Administración de Programación

Administración de Categorías

Administración de Notificaciones

Administración de Archivos Adicionales

Menú del Programador

Ver Cartelera

Reportes

Estadísticas en PDF

Menú del Transmisor

Ver Cartelera

Notificaciones

Estadísticas en PDF

Programación

Canal

Usuarios

Desplegando 1-10 de 11 resultados.

Solapín	Nombre	Apellidos	Rol		
<input type="checkbox"/>	E126135	Leonaris	Acosta Suarez	Transmisor	
<input type="checkbox"/>	E118053	Juan Jose	Perez Hernandez	Programador	
<input type="checkbox"/>	E120359	Arnaldo Enrique	Peñaranda Pim	Usuario	
<input type="checkbox"/>	E117638	Rolando Rafael	Futiel Gutierrez	Auditor	
<input type="checkbox"/>	E143167	Lisandra	Rodriguez Mendoza	Usuario	
<input type="checkbox"/>	T134259	Karel	Valdes Montesino	Usuario	
<input type="checkbox"/>	E117751	Elvis	Matos Matos	Usuario	
<input type="checkbox"/>	E136277	Fredy	Almenares Fletas	Usuario	
<input type="checkbox"/>	E153089	Rachel	Martinez Del Cerro Parada	Usuario	
<input type="checkbox"/>	E100719	Rafael Alexander	Mendoza Fernandez	Usuario	

1 < > Siguiente >

Borrar seleccionados

Universidad de las Ciencias Informáticas. Facultad 5. © 2016

Capítulo 2: Modelación del Sistema

Tabla 6: HU6 Buscar usuario.

Historia de Usuario																																																							
Número: 6	Usuario: Administrador, Programador																																																						
Nombre de Historia de Usuario: Buscar usuario.																																																							
Prioridad en Negocio: Alta	Iteración Asignada: 1																																																						
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana																																																						
<p>Descripción: La herramienta debe permitir buscar a cualquier usuario que se encuentre registrado. Para ello se debe de ir a la vista de Listado de usuarios y teniendo en cuenta los siguientes criterios de búsqueda: el solapín, nombre, apellidos y rol, se buscan los usuarios.</p>																																																							
Prototipo																																																							
 <p>Desplegando 1-10 de 11 resultados.</p> <table border="1"> <thead> <tr> <th>Solapín</th> <th>Nombre</th> <th>Apellidos</th> <th>Rol</th> </tr> </thead> <tbody> <tr><td><input type="checkbox"/></td><td>E126135</td><td>Leonarkis</td><td>Acosta Suarez</td><td>Transmisor</td></tr> <tr><td><input type="checkbox"/></td><td>E118053</td><td>Juan Jose</td><td>Perez Hernandez</td><td>Programador</td></tr> <tr><td><input type="checkbox"/></td><td>E120359</td><td>Arnaldo Enrique</td><td>Pe?aranda Prim</td><td>Usuario</td></tr> <tr><td><input type="checkbox"/></td><td>E117638</td><td>Rolando Rafael</td><td>Futiel Gutierrez</td><td>Auditor</td></tr> <tr><td><input type="checkbox"/></td><td>E143167</td><td>Lisandra</td><td>Rodriguez Mendoza</td><td>Usuario</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>T134259</td><td>Karel</td><td>Valdes Montesino</td><td>Usuario</td></tr> <tr><td><input type="checkbox"/></td><td>E117751</td><td>Eliuvis</td><td>Matos Matos</td><td>Usuario</td></tr> <tr><td><input type="checkbox"/></td><td>E136277</td><td>Fredy</td><td>Almenares Fleitas</td><td>Usuario</td></tr> <tr><td><input type="checkbox"/></td><td>E153089</td><td>Rachel</td><td>Martinez Del Cerro Parada</td><td>Usuario</td></tr> <tr><td><input type="checkbox"/></td><td>E100719</td><td>Rafael Alexander</td><td>Mendoza Fernandez</td><td>Usuario</td></tr> </tbody> </table> <p>1 2 Siguiente ></p>		Solapín	Nombre	Apellidos	Rol	<input type="checkbox"/>	E126135	Leonarkis	Acosta Suarez	Transmisor	<input type="checkbox"/>	E118053	Juan Jose	Perez Hernandez	Programador	<input type="checkbox"/>	E120359	Arnaldo Enrique	Pe?aranda Prim	Usuario	<input type="checkbox"/>	E117638	Rolando Rafael	Futiel Gutierrez	Auditor	<input type="checkbox"/>	E143167	Lisandra	Rodriguez Mendoza	Usuario	<input checked="" type="checkbox"/>	T134259	Karel	Valdes Montesino	Usuario	<input type="checkbox"/>	E117751	Eliuvis	Matos Matos	Usuario	<input type="checkbox"/>	E136277	Fredy	Almenares Fleitas	Usuario	<input type="checkbox"/>	E153089	Rachel	Martinez Del Cerro Parada	Usuario	<input type="checkbox"/>	E100719	Rafael Alexander	Mendoza Fernandez	Usuario
Solapín	Nombre	Apellidos	Rol																																																				
<input type="checkbox"/>	E126135	Leonarkis	Acosta Suarez	Transmisor																																																			
<input type="checkbox"/>	E118053	Juan Jose	Perez Hernandez	Programador																																																			
<input type="checkbox"/>	E120359	Arnaldo Enrique	Pe?aranda Prim	Usuario																																																			
<input type="checkbox"/>	E117638	Rolando Rafael	Futiel Gutierrez	Auditor																																																			
<input type="checkbox"/>	E143167	Lisandra	Rodriguez Mendoza	Usuario																																																			
<input checked="" type="checkbox"/>	T134259	Karel	Valdes Montesino	Usuario																																																			
<input type="checkbox"/>	E117751	Eliuvis	Matos Matos	Usuario																																																			
<input type="checkbox"/>	E136277	Fredy	Almenares Fleitas	Usuario																																																			
<input type="checkbox"/>	E153089	Rachel	Martinez Del Cerro Parada	Usuario																																																			
<input type="checkbox"/>	E100719	Rafael Alexander	Mendoza Fernandez	Usuario																																																			

Tabla 7: HU15 Listar Programación.

Historia de Usuario	
Número:15	Usuarios: Administrador, Programador, Transmisor
Nombre de Historia de Usuario: Listar Programación.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir listar la programación. Para ello se debe de ir a Administración de Programación localizada en el menú izquierdo de la página principal y aparecerán las programaciones realizadas. En caso de ser el rol de Transmisor se debe de ir a Programación ubicada en el menú izquierdo y dar clic.</p>	

Capítulo 2: Modelación del Sistema

Prototipo:

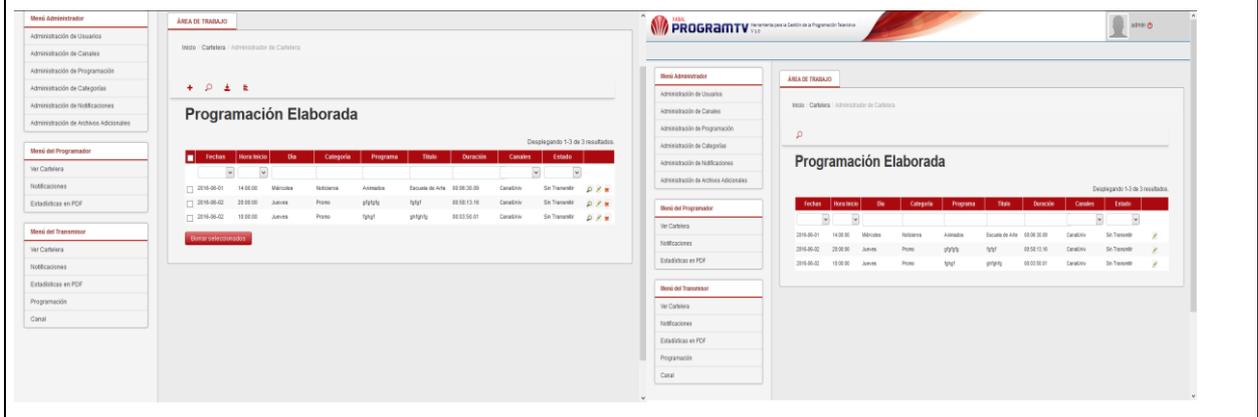


Tabla 8: HU12 Insertar programación

Historia de Usuario	
Número: 12	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Insertar programación.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.2 semanas
<p>Descripción: La herramienta debe permitir adicionar una programación. Para ello se debe ir a la opción Administración de programación ubicada en el menú izquierdo de la página, luego dar un clic en el botón agregar un programa (+), seguidamente saldrá una vista para introducir la información en los campos correspondientes, luego dar clic en el botón Crear. Si los datos introducidos son los correctos aparecerá un cartel mostrando (“Se ha creado la programación correctamente.”).</p>	

Prototipo



Capítulo 2: Modelación del Sistema

Tabla 9: HU13 Modificar programación.

Historia de Usuario	
Número: 13	Usuario: Administrador, Programador y Transmisor
Nombre de Historia de Usuario: Modificar programación.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir modificar una programación. Para ello dar en la opción Editar de la programación específica para ser modificada. Aparecerá una vista con los datos a modificar de dicha programación escogida, se le modifican los datos correspondientes. Luego dar clic en el botón Actualizar y aparecerá un cartel con el nombre (" La programación se ha modificado correctamente."). En caso de que el rol sea Transmisor, debe aparecer la lista de programación seguida de la opción Editar en cada programación, luego aparecerá una vista con solo acceso a modificar el Estado de esa programación. A continuación una vista de modificar un programación.</p>	
<p>Prototipo</p> 	

Tabla 10: HU14 Eliminar programación

Historia de Usuario	
Número: 14	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Eliminar programación.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir la eliminación de uno o varias programaciones. Para ello se listan todas las programaciones en la vista de administración, se selecciona una o varias programaciones que se quieren eliminar, dando clic en el botón eliminar y luego dar aceptar en el mensaje que aparece (¿Seguro que desea borrar este elemento?) para eliminarlo.</p>	

Capítulo 2: Modelación del Sistema

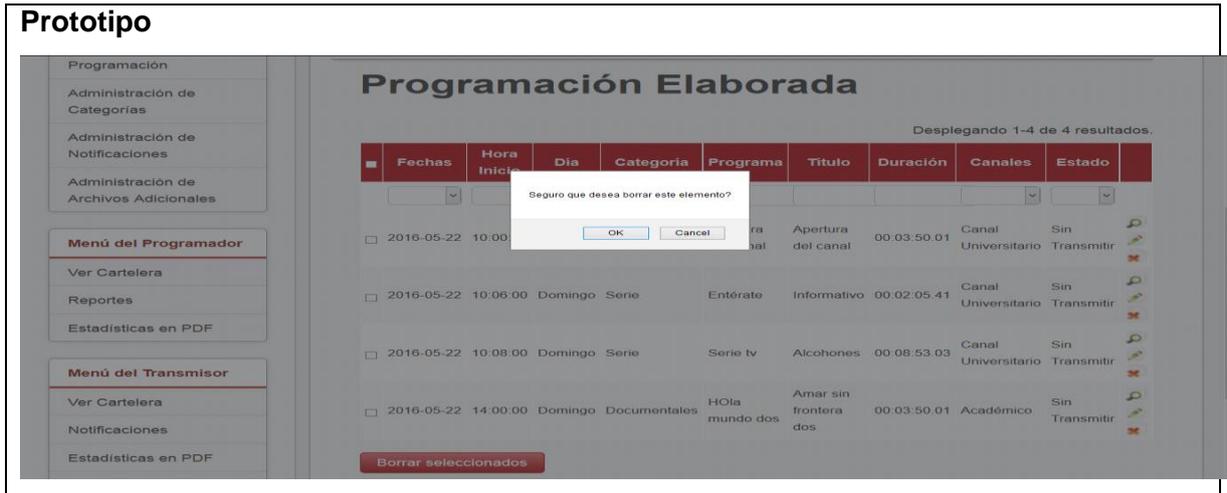


Tabla 11: HU16 Mostrar datos detallados de cada programación.

Historia de Usuario																			
Número: 16	Usuario: Administrador, Programador																		
Nombre de Historia de Usuario: Mostrar datos detallados de cada programación.																			
Prioridad en Negocio: Alta	Iteración Asignada: 1																		
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana																		
<p>Descripción: La herramienta debe permitir ver los datos de cada programación seleccionada. Para ello se presiona la opción Ver de una programación que aparece en la vista de administración de programación. En la nueva pestaña que se muestra encontrará todos los datos de la programación seleccionada.</p>																			
Prototipo																			
<table border="1"> <tbody> <tr> <td>Día</td> <td>Domingo</td> </tr> <tr> <td>Hora Inicio</td> <td>10:00:00</td> </tr> <tr> <td>Categoría</td> <td>Apertura</td> </tr> <tr> <td>Programa</td> <td>Apertura del canal</td> </tr> <tr> <td>Título</td> <td>Apertura del canal</td> </tr> <tr> <td>Duración</td> <td>00:03:50.01</td> </tr> <tr> <td>Canal</td> <td>Canal Universitario</td> </tr> <tr> <td>Sinopsis</td> <td>Programa para dar paso al inicio del canal.</td> </tr> <tr> <td>Estado</td> <td>Sin Transmitir</td> </tr> </tbody> </table>		Día	Domingo	Hora Inicio	10:00:00	Categoría	Apertura	Programa	Apertura del canal	Título	Apertura del canal	Duración	00:03:50.01	Canal	Canal Universitario	Sinopsis	Programa para dar paso al inicio del canal.	Estado	Sin Transmitir
Día	Domingo																		
Hora Inicio	10:00:00																		
Categoría	Apertura																		
Programa	Apertura del canal																		
Título	Apertura del canal																		
Duración	00:03:50.01																		
Canal	Canal Universitario																		
Sinopsis	Programa para dar paso al inicio del canal.																		
Estado	Sin Transmitir																		

Tabla 12: HU17 Mostrar cartelera por día.

Historia de Usuario	
Número: 17	Usuario: Todos
Nombre de Historia de Usuario: Mostrar cartelera por día.	
Prioridad en Negocio: Alta	Iteración Asignada: 3

Capítulo 2: Modelación del Sistema

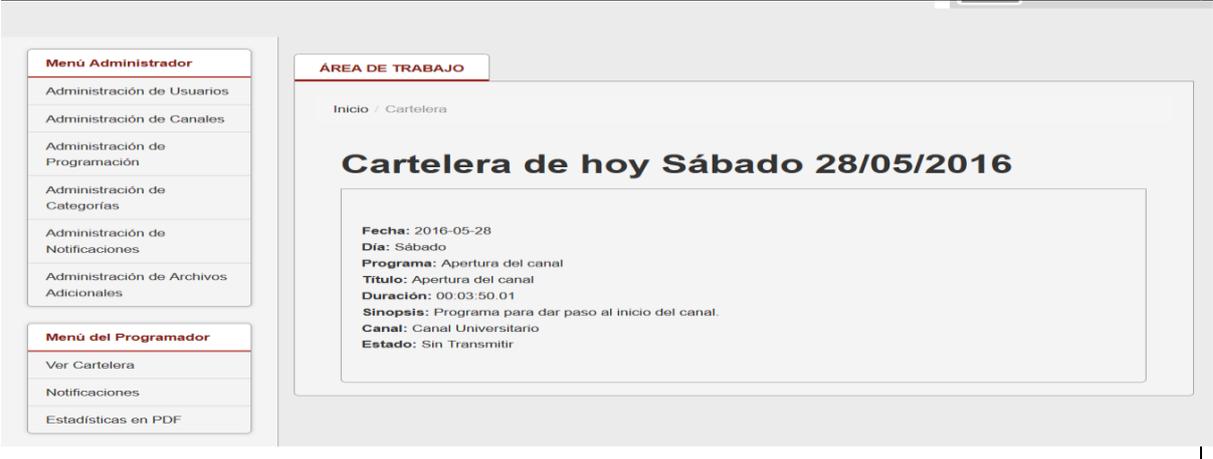
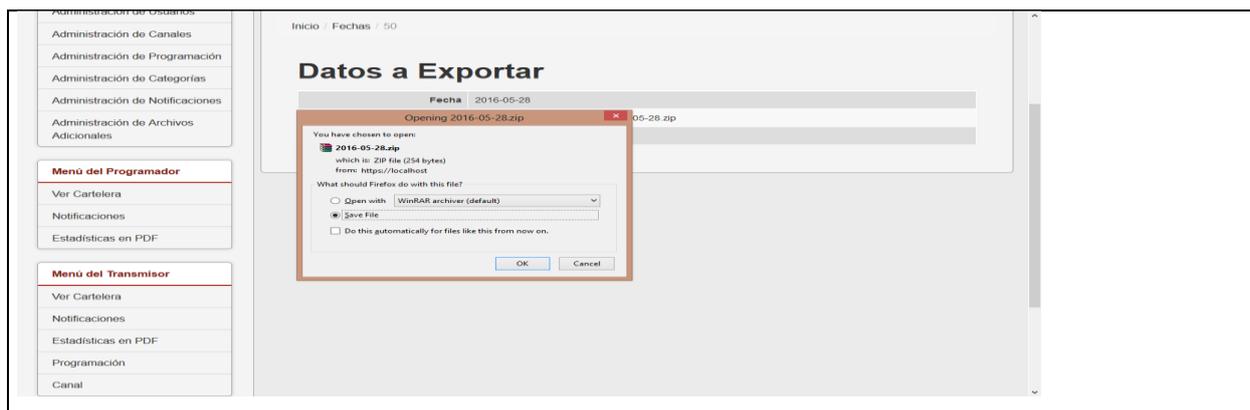
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir mostrar la cartelera diaria. Para ello el usuario se autentica en la aplicación y si es satisfactoria la autenticación aparecerá la vista de la cartelera del día actual. De otra manera se accedería dando clic en la opción Ver Cartelera del menú izquierdo y aparecerá la cartelera del día actual en la parte derecha de la página.</p>	
<p>Prototipo</p> 	

Tabla 13: HU18 Exportar en formato PLS la lista de reproducción dada la fecha.

Historia de Usuario	
Número: 18	Usuario: Administrador, Programador, Transmisor
Nombre de Historia de Usuario: Exportar en formato PLS la lista de reproducción dada la fecha.	
Prioridad en Negocio: Media	Iteración Asignada: 1
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.3 semanas
<p>Descripción: La herramienta debe permitir exportar la lista de reproducción en formato PLS para ser reproducida con el reproductor Media Player Clásico. Para ello se debe ir a la Administración de Programación, dar clic en el botón exportar fichero a PLS, dar clic en el botón generar lista PLS, seleccionar la fecha, el canal y la sección de transmisión correspondiente, luego dar clic en generar archivo PLS y aparecerá la descarga del archivo Winrar donde está incluido en su interior el fichero de la lista.</p>	
<p>Prototipo</p>	

Capítulo 2: Modelación del Sistema



2.2.2 Estimación de esfuerzos por historia de usuario.

Para la realización de la solución propuesta se realizó la estimación de esfuerzo la cual se presenta en la siguiente tabla.

Tabla 14: Estimación de esfuerzos por HU.

No.	Historia de usuario	Estimación(semanas)
1	Autenticar Usuario	0.2
2	Modificar rol de Usuario	0.1
3	Eliminar Usuario	0.1
4	Listar Usuarios	0.1
5	Mostrar Datos detallados de cada Usuario	0.1
6	Buscar Usuario	0.1
7	Inserta Canal	0.3
8	Modifica Canal	0.1
9	Elimina Canal	0.1
10	Listar Canales	0.2
11	Mostrar datos detallados de cada Canal	0.1
12	Inserta Programación	0.2
13	Modifica Programación	0.1
14	Elimina Programación	0.1
15	Listar Programación	0.1
16	Mostrar datos detallados de cada Programación	0.1
17	Mostrar Cartelera por día	0.1
18	Exportar en formato PLS la lista de reproducción	0.3
19	Inserta Categoría	0.2
20	Modifica Categoría	0.1

Capítulo 2: Modelación del Sistema

21	Elimina Categoría	0.1
22	Listar Categorías	0.1
23	Mostrar datos detallados de cada Categoría	0.1
24	Generar Estadística de Programación mediante PDF	0.1
25	Inserta Archivos Adicionales	0.2
26	Modifica Archivos Adicionales	0.1
27	Elimina Archivos Adicionales	0.1
28	Listar Archivos Adicionales	0.1
29	Mostrar datos detallados de cada Archivos Adicional	0.1
30	Inserta Notificación	0.2
31	Modifica Notificación	0.1
32	Elimina Notificación	0.1
33	Lista Notificación realizada	0.1
34	Mostrar datos detallados de cada Notificación	0.1
35	Envía notificación vía email	0.2
36	Buscar Cartelera dada la fecha	0.2
37	Convierte videos en formato mp4	0.4
38	Mostrar gráficamente la cantidad de programación por categoría	0.2
39	Mostrar alerta de poco espacio en el disco.	0.1

2.3 Plan de Iteraciones

Luego de conocer las HU y determinar la estimación del esfuerzo por cada una de ellas, se comienza a realizar el plan de iteraciones. La metodología XP se divide en iteraciones relativamente cortas. Las HU con mayor importancia por ser funcionalidades indispensables para la aplicación, deben ser implementadas en las primeras iteraciones.

A continuación se muestra el plan de iteraciones divididas 3 iteraciones, así como la descripción y el total de semanas que durarán cada una de estas.

Tabla 15: Plan de iteraciones.

Iteración	Descripción	Orden de las HU a implementar	Duración Total
1	Las HU que poseen una prioridad alta se implementan en la primera iteración, logrando así	1-24	0.24 semanas

Capítulo 2: Modelación del Sistema

	cubrir las necesidades del cliente. Además son fundamentales para lograr un buen funcionamiento y desarrollo de la aplicación.		
2	Las HU que poseen una prioridad media se implementan en la segunda iteración, ya complementan a las HU anteriores, pero estas son de menor importancia.	25-30	0.5 semanas
3	Las HU que poseen una prioridad baja se implementan en la tercera iteración, pues estas son menos importantes que las anteriores dándole la posibilidad de no realizar funciones primarias.	31-39	0.15 semanas

2.3.1 Plan de entrega

En el plan de entrega se agrupan las HU de manera ordenada con el objetivo de conformar una entrega, la cual tiene como objetivo dar a conocer la cantidad de HU por iteraciones y su fecha de entrega. A continuación se muestra una tabla con el plan de entrega para cada iteración.

Tabla 16: Plan de entrega.

Iteraciones	Iteración #1	Iteración #2	Iteración #3
Cantidad de HU	24	5	8
Fecha de entrega	15/3/2016	25/3/2016	5/4/2016

2.4: Diseño del Sistema

Para el diseño de aplicaciones, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando UML, sino que utiliza otras técnicas como lo son: las tarjetas CRC (clase, responsabilidad, colaboración), como una extensión informal a UML (10).

2.4.1 Tarjetas CRC.

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son una herramienta de diseño, las cuales se dividen en tres secciones que contienen la información del nombre de las clases,

Capítulo 2: Modelación del Sistema

sus responsabilidades y sus colaboraciones. Estas normalmente son usadas cuando se determina primero las clases que se necesitan, su iteración y después la implementación de la solución (27). La estructura de una tarjeta CRC está compuesta por el nombre de la clase, a la izquierda todas las responsabilidades, y a la derecha las clases implicadas en cada responsabilidad. En fin, las tarjetas CRC contribuyen al diseño a todo el equipo del proyecto.

A continuación se presentan algunas de las tarjetas CRC de la herramienta. Las otras se pueden consultar en el Anexo 2.

Capítulo 2: Modelación del Sistema

Tabla 17: Tarjeta CRC UsuarioController

Tarjeta CRC	
Clase: UsuarioController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> -Crear usuario. -Modificar rol de usuario -Eliminar usuario. -Listar usuario. -Mostrar en una vista detallada los datos de un usuario. 	Usuario

Tabla 18: Tarjeta CRC CanalesController.

Tarjeta CRC	
Clase: CanalesController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> -Crear canal. -Modificar canal -Eliminar canal. -Listar canal. -Mostrar en una vista detallada los datos de un canal. 	Canales

Tabla 19: Tarjeta CRC ProgramaciónController

Tarjeta CRC	
Clase: ProgramaciónController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> -Crear programación de un día. -Modificar programación e un día en específico. -Eliminar programación dada una fecha. -Listar programación. -Mostrar en una vista detallada los datos de una programación. 	Programación Canales Fechas

Tabla 20: Tarjeta CRC CategoríaController.

Capítulo 2: Modelación del Sistema

Tarjeta CRC	
Clase: CategoríaController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> -Crear categoría. -Modificar categoría. -Eliminar categoría. -Listar categoría. -Mostrar en una vista detallada los datos de una categoría. 	<ul style="list-style-type: none"> Categoría Programación

Tabla 21: Tarjeta CRC ReportesController.

Tarjeta CRC	
Clase: ReportesController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> -Crear reportes. -Modificar reportes. -Eliminar reportes. -Listar reportes. -Mostrar en una vista detallada los datos de un reporte. 	<ul style="list-style-type: none"> Reportes

Tabla 22: Tarjeta CRC ArchivosAdicionalesController.

Tarjeta CRC	
Clase: ArchivosAdicionalesController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> -Crea un archivo adicional -Modifica un archivo adicional -Elimina un archivo adicional -Lista los archivos adicionales -Muestra una vista detallada de cada archivo adicional 	<ul style="list-style-type: none"> ArchivosAdicionales

2.5: Arquitectura de Software.

De acuerdo al termino arquitectura de software definido por el Instituto de Ingeniera de Software (Software Engineering Institute, SEI), se puede decir que, “La *Arquitectura de Software* son las estructuras de un sistema, compuestas de elementos con propiedades

Capítulo 2: Modelación del Sistema

visibles de forma externa y las relaciones que existen entre ellos.” (28), es la organización fundamental de un sistema para orientar el trabajo, su diseño y evolución.

Como propuesta para el desarrollo de la herramienta se decide escoger el patrón Modelo-Vista-Controlador (MVC), el cual permite dividir la aplicación en tres componentes distintos: el modelo, la vista y el controlador, que los cambios ocurridos en una de estas etapas no afecten a las demás.

2.5.1 Modelo-Vista-Controlador

Yii por su parte, está basado en este modelo, permitiendo un mejor manejo de los datos y está basado en tres niveles fundamentales.

Modelo: Representa la estructura de los datos en la aplicación, es decir, su lógica de negocio. Normalmente las clases del modelo permiten crear, actualizar y eliminar datos de la base de datos de forma rápida y sencilla, también permite llevar el control detallado de las vistas de la aplicación como tal.

A continuación, se muestra la estructura del componente modelo.

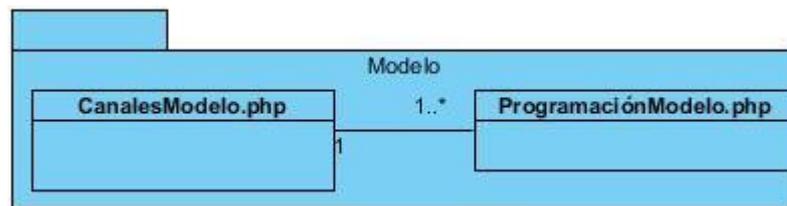


Figura 1: Arquitectura del componente Modelo.

Vista: Es la encargada de visualizar los datos representados por el Modelo. Una vista es la representación de la información que va a ver el usuario. Una vista es normalmente una página web. A continuación, se muestra una estructura del componente vista.

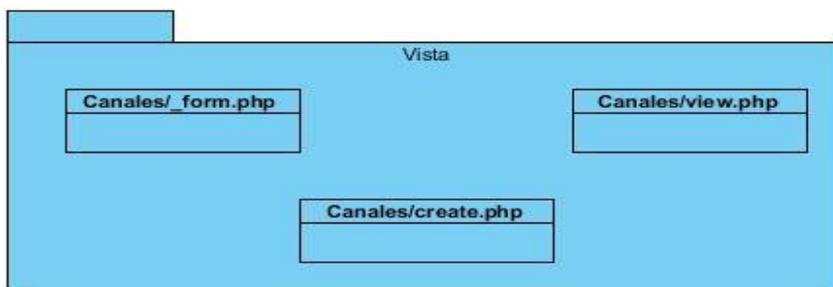


Figura 2: Arquitectura del componente Vista.

Controlador: Es el componente intermedio entre el modelo y la vista. Es el encargado de llamar al modelo para obtener algunos datos y enviárselos a la vista correspondiente. Es el encargado de implementar la validación de los datos y tiene la tarea de recibir los datos entrados por el usuario y procesarlos.

Capítulo 2: Modelación del Sistema

A continuación, se muestra una estructura en el componente controlador.

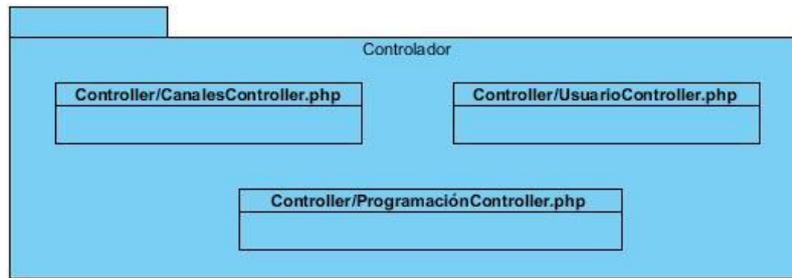


Figura 3: Arquitectura en el componente Controlador.

Yii por su parte le realizó al patrón algunas modificaciones para poder adaptarlo a su funcionamiento sin perder la lógica. Incorporó un componente intermedio llamado widgets (**Figura 4**) entre el modelo, la vista y el controlador para posibilitar el manejo de información, este componente es utilizado tanto para la entrada de datos, como para mostrar los mismos. También incorporó un componente de aplicaciones, el cual es el encargado de conectar el controlador con los restantes componentes que utiliza Yii, en caso de que se requiera su utilización, esto facilitará el trabajo para el desarrollador, cuando necesite incorporar funcionalidades que esos componentes de aplicaciones contienen.

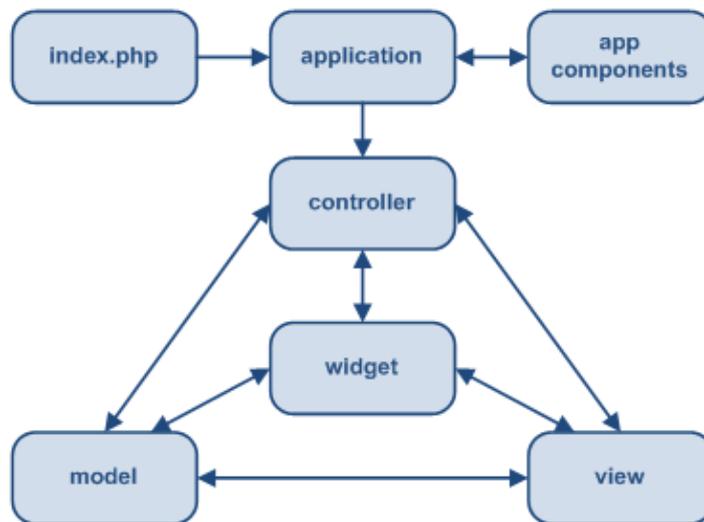


Figura 4: Patrón MVC en Yii (17).

2.6 Patrones de Diseño

Los patrones de diseños son una estrategia para dar soluciones a problemas típicos y recurrentes que se presentan durante el proceso de diseño de un software. A gran escala brindan una solución probada y documentada a problemas de desarrollo de un software. Cada patrón de diseño se centraliza en un problema, describiendo cuando aplicarlo, teniendo en

Capítulo 2: Modelación del Sistema

cuenta algunas restricciones de diseño y su uso. Emplean un conjunto de buenas prácticas simplificando así el trabajo, y facilitan la comprensión y aprendizaje de personas ajenas al sistema al formalizar un vocabulario común (29).

2.6.1 Patrones GRASP

Los patrones de diseño GRASP, por sus siglas en inglés (General Responsibility Assignment Software Patterns), son patrones creados para la Asignación General de Responsabilidades. Estos describen la asignación de responsabilidades a objetos en este caso (30).

Se muestra a continuación los utilizados en el desarrollo de la herramienta.

Experto: Este patrón es el encargado de asignar un responsable a la clase que cuenta con la información necesaria para cumplir con una tarea determinada. El patrón experto permitió una correcta distribución de las clases para tener un mejor manejo de la información, ya que en las vistas está definido todo lo que el usuario va a ver, en los modelos las validaciones y los controladores todo el desarrollo de las funcionalidades. Por ejemplo, la clase UsuarioController funciona como experta en el control de la información para llevar a cabo el trabajo con los usuarios, en la misma se crean, modifican, eliminar, se listan usuarios.

La siguiente figura muestra un ejemplo del patrón experto en la herramienta.

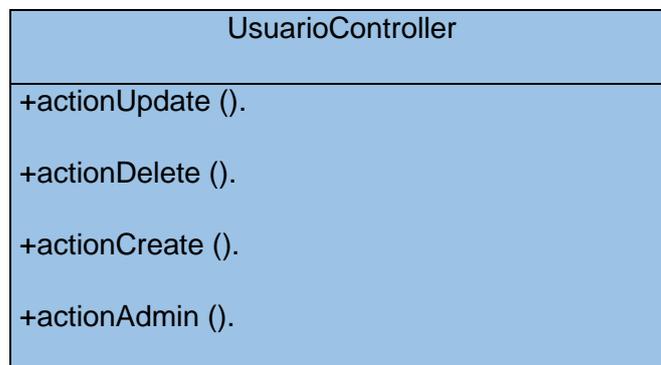


Figura 5: Diagrama de clases que representa el Patrón Experto.

Controlador: Se encarga de manejar los mensajes de los eventos de un sistema a una clase. En la programación orientada a objetos cuando se recibe un evento externo hay que elegir los controladores que se encarguen específicamente de procesar este evento (30).

A continuación se muestra en patrón controlador en la herramienta.

Capítulo 2: Modelación del Sistema

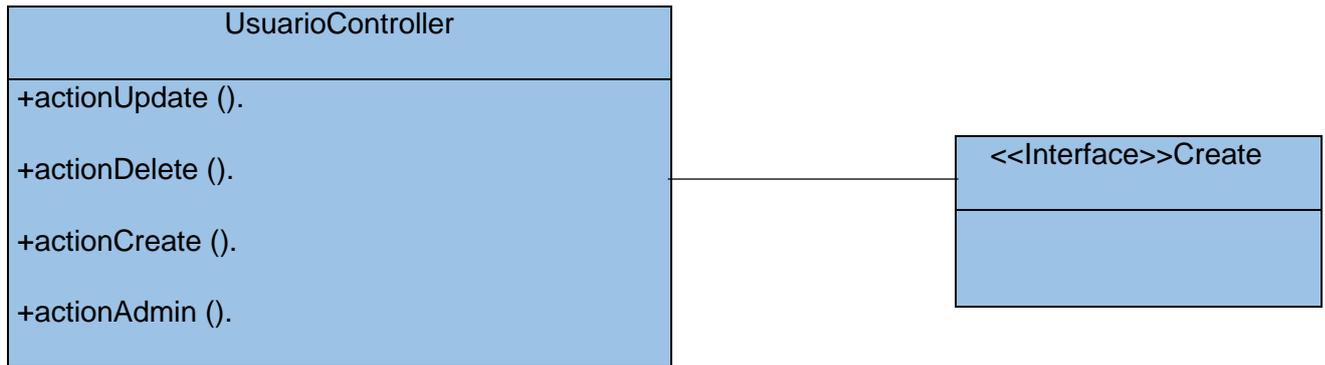


Figura 6: Diagrama de clases que representa el Patrón Controlador.

Creador: El patrón creador ayuda a identificar quien debe ser el responsable de la creación de un objeto o clase. El nuevo objeto deberá contener la información necesaria para realizar la creación, se debe manejar o almacenar varias instancias de la clase y agregarla a la misma.

A continuación se muestra el patrón creador en la herramienta.



Figura 7: Diagrama de clase que representa el Patrón Creador.

Bajo Acoplamiento: El bajo acoplamiento es un patrón que estimula la asignación de responsabilidades, fomenta el aumento de la reutilización y la eliminación de las redundancias. Soporta el diseño de clases más independientes, que reduce el impacto de los cambios. No se aísla de otros patrones como el Experto o el de Alta Cohesión, que seguidamente de este estaremos hablando, sino que se incluye como principio de diseño que influye en la decisión de asignar responsabilidades. Es necesario tener un bajo acoplamiento cuando las clases son muy genéricas y con grandes probabilidades de reutilización. Para alcanzar un bajo acoplamiento las clases controladoras heredan únicamente de la clase Controller.

Alta Cohesión: La alta cohesión es saber cuan relacionadas y enfocadas están las relaciones en una clase. Este patrón expresa que se debe de asignar responsabilidades a una misma área de la aplicación y que no tenga mucha complejidad. Es de utilidad tener clases con alta cohesión ya que hace más fácil el mantenimiento de la misma (30).

La herramienta para la planificación de la programación en la Dirección de Comunicación Institucional sigue los principios de la alta cohesión ya que Yii brinda la posibilidad de no hacer

Capítulo 2: Modelación del Sistema

uso de las clases saturadas de métodos y organizar mejor las responsabilidades con una alta cohesión.

Conclusiones Parciales

El trabajo descrito en el presente capítulo permitió el desarrollo, tanto de los requisitos como del diseño de la aplicación, lo que constituye la base para la posterior implementación de la misma. Las tres iteraciones y 39 HU identificadas permiten describir los aspectos a tener en cuenta en el desarrollo de la solución. La definición de las tarjetas CRC aportan la descripción de las responsabilidades de las clases y sus colaboraciones. Los patrones de diseño escogidos permiten rapidez, flexibilidad y eficiencia a la hora de diseñar la aplicación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se especifican las tareas de ingeniería, así como las fases de implementación y prueba, con el objetivo de verificar si se cumplieron las tareas de ingeniería planificadas para el desarrollo de la aplicación. Además se muestra el diagrama de despliegue del sistema a implementar.

3.1 Tareas de Ingeniería

Las Tareas de Ingeniería (TI) son un conjunto de actividades que dan cumplimiento a cada HU. Se describen de forma tal que se entienda lo que el sistema tiene que hacer y así facilitar su construcción. Estas tareas pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

En las Tablas 23 –28 se muestran las TI correspondientes a la HU Gestionar Usuario de la primera iteración, las restantes pueden ser consultadas en el documento “Tareas de Ingeniería” en el anexo 2:

Tabla 23: Tarea de Ingeniería 1 de la HU1.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 1
Nombre de Tarea: Autenticar usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2 semanas
Fecha Inicio: 2/3/2016	Fecha Fin: 4/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda autenticarse.	

Tabla 24: Tarea de Ingeniería 1 de la HU2.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 2
Nombre de Tarea: Modificar rol de usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2 semanas
Fecha Inicio: 4/3/2016	Fecha Fin: 5/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario autorizado pueda cambiar el rol correspondiente a un usuario registrado en la aplicación.	

Capítulo 3: Implementación y Pruebas

Tabla 25: Tarea de Ingeniería 1 de la HU3.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 3
Nombre de Tarea: Eliminar usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1 semana
Fecha Inicio: 7/3/2016	Fecha Fin: 8/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario autorizado pueda eliminar un usuario registrado en la aplicación.	

Tabla 26: Tarea de Ingeniería 1 de la HU5.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 5
Nombre de Tarea: Mostrar datos de un usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2 semanas
Fecha Inicio: 10/3/2016	Fecha Fin: 12/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario autorizado pueda tener acceso a ver los datos específicos de cada usuario registrado en la aplicación.	

Tabla 27: Tarea de Ingeniería 1 de la HU6.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 6
Nombre de Tarea: Buscar usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1 semanas
Fecha Inicio: 15/3/2016	Fecha Fin: 16/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario autorizado pueda buscar a un usuario registrado.	

Capítulo 3: Implementación y Pruebas

Tabla 28: Tarea de Ingeniería1 de la HU7.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 7
Nombre de Tarea: Insertar Canal	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3 semanas
Fecha Inicio: 16/3/2016	Fecha Fin: 19/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario autorizado pueda insertar un canal en la aplicación.	

3.1 Diseños de Casos de Pruebas

Las pruebas de aceptación son realizadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. Es una de las alternativas que propone la metodología XP para validar las soluciones. Estas pruebas también son conocidas como prueba de caja negra o pruebas funcionales. Además son destinadas a evaluar si se obtuvo la aplicación requerida y su correcto funcionamiento.

A continuación se muestran algunas pruebas de aceptación realizadas a la solución propuesta y con los resultados obtenidos.

Tabla 29: Mostrar datos del usuario.

Caso de Prueba de Aceptación	
Código:	HU: 5
Nombre: Mostrar datos del usuario.	
Descripción: La herramienta debe permitir visualizar los datos completos de un usuario en específico.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación.	
Pasos de Ejecución: Dar clic en el botón Ver del usuario correspondiente.	
Resultado Esperado: Se muestra una página donde están los datos de ese usuario.	
Resultados Obtenidos: Se muestra los datos detallados de un usuario en específico.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 30: Listar usuarios.

Caso de Prueba de Aceptación	
Código:	HU: 4
Nombre: Listar usuarios.	

Capítulo 3: Implementación y Pruebas

Descripción: La herramienta debe permitir listar los usuarios registrados.
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación.
Pasos de Ejecución: Seleccionar la opción Administración de Usuario ubicada en el menú izquierdo, luego seleccionar el botón Administrar usuario.
Resultado Esperado: Los usuarios debe ser listado en la aplicación.
Resultados Obtenidos: Se muestra los usuarios registrados en la aplicación.
Evaluación de la Prueba: Satisfactoria.

Tabla 31: Autenticar usuario.

Caso de Prueba de Aceptación	
Código:	HU: 1
Nombre: Autenticar usuario.	
Descripción: La herramienta debe permitir la autenticación de un usuario para acceder al entorno de trabajo de la misma haciendo uso de usuario y contraseña.	
Condiciones de Ejecución: La página de autenticación de la herramienta debe estar abierta.	
Pasos de Ejecución: Introducir los datos en los campos (usuario y contraseña). Luego acceder mediante un clic en el botón Acceder.	
Resultado Esperado: El usuario debe ser autenticado en el sistema.	
Resultados Obtenidos: El usuario autenticado satisfactoriamente en el sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 32: Buscar usuario.

Caso de Prueba de Aceptación	
Código:	HU: 6
Nombre: Buscar usuario.	
Descripción: La herramienta debe permitir buscar los usuarios registrados.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación.	
Pasos de Ejecución: Seleccionar la opción Administración de Usuario ubicada en el menú izquierdo, luego seleccionar el botón Administrar usuario, introducir el nombre de usuario a buscar.	
Resultado Esperado: Los usuarios debe ser listado en la aplicación.	
Resultados Obtenidos: Se muestra un listado con uno o varios usuarios registrados en la aplicación perteneciente al criterio de búsqueda realizado.	
Evaluación de la Prueba: Satisfactoria.	

Capítulo 3: Implementación y Pruebas

Tabla 33: Insertar canal.

Caso de Prueba de Aceptación	
Código:	HU: 7
Nombre: Insertar canal.	
Descripción: La herramienta debe permitir insertar un canal.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación.	
Pasos de Ejecución: Seleccionar la opción Administración de Canales ubicada en el menú izquierdo, luego seleccionar el botón Agregar un Canal.	
Resultado Esperado: La aplicación debe almacenar en la base de datos los valores introducidos del canal, y mostrar un mensaje de satisfacción al usuario.	
Resultados Obtenidos: Los datos del canal se almacenaron en la base de datos y se mostró un mensaje de satisfacción al usuario	
Evaluación de la Prueba: Satisfactoria.	

Tabla 34: Modificar canal.

Caso de Prueba de Aceptación	
Código:	HU: 8
Nombre: Modificar canal.	
Descripción: La herramienta debe permitir modificar el canal seleccionado.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación.	
Pasos de Ejecución: Seleccionar la opción Administración de Canales ubicada en el menú izquierdo, luego seleccionar el botón editar correspondiente a un canal.	
Resultado Esperado: La aplicación debe almacenar en la base de datos los valores modificados de un canal, y mostrar un mensaje de satisfacción al usuario.	
Resultados Obtenidos: Los datos del canal se almacenaron en la base de datos y se mostró un mensaje de satisfacción al usuario.	
Evaluación de la Prueba: Satisfactoria.	

3.2 Pruebas de rendimiento

Las pruebas de rendimiento del software son centradas fundamentalmente en determinar la velocidad con la que el sistema bajo pruebas realiza una tarea en específica. Estas pruebas ayudan a detectar los cuellos de botellas de la aplicación, antes de que los usuarios sufran de un mal rendimiento, con la condición de pérdida económica o frustración de los usuarios que operan en la aplicación (31).

Capítulo 3: Implementación y Pruebas

En las siguientes tablas se muestran dos pruebas de rendimiento realizada a la aplicación con la herramienta JMeter, donde en un primer caso se realizó para 1 usuario y en un segundo caso se realizó para las 5 personas que trabajan en el departamento de comunicación.

Tabla 35: Caso 1

Nombre de la muestra	# de muestras	Min	Max	%Error	Rendimiento	Kb/s
/sms/	1	16	16	0.00%	44.5/sec	1648
/sms/cartelera/create	1	10	10	0.00%	99.0/sec	2115.1
Total	2	10	16	0.00%	61.0/sec	1824.2

Tabla 36: Caso 2

Nombre de la muestra	# de muestras	Min	Max	%Error	Rendimiento	Kb/s
/sms/	5	16	70	0.00%	27.5/sec	100
/sms/cartelera/create	5	10	86	0.00%	32.0/sec	110
Total	10	10	86	0.00%	59.5/sec	210

Donde:

de muestras: Cantidad de usuarios

Min: El mínimo tiempo transcurrido por URL.

Max: El máximo tiempo transcurrido por URL.

%error: La tasa de errores.

Rendimiento: Rendimiento medio en base a peticiones por segundo.

Kb/s: Velocidad de respuestas.

3.3 Resultados esperados

La herramienta propuesta le da solución al problema planteado inicialmente, ya que ofrece solución a las deficiencias existentes en la Dirección de Comunicación Institucional. Además al poder llevar el control organizado de los listados de programación se disminuye en tiempo y se facilita la creación de la cartelera automáticamente.

Capítulo 3: Implementación y Pruebas

A la herramienta se le aplicaron pruebas de aceptación y de rendimiento logrando, en esta última, resultados satisfactorios y las de aceptación fueron desarrolladas en tres iteraciones. En la primera iteración se obtuvieron 15 pruebas satisfactorias y 10 no conformidades, en una segunda iteración se le dieron solución a las no conformidades encontradas anteriormente y se hallaron 4 no conformidades más, y en una tercera iteración los 39 casos de prueba de aceptación realizados resultaron satisfactorios. A continuación se muestran estos resultados en una gráfica.

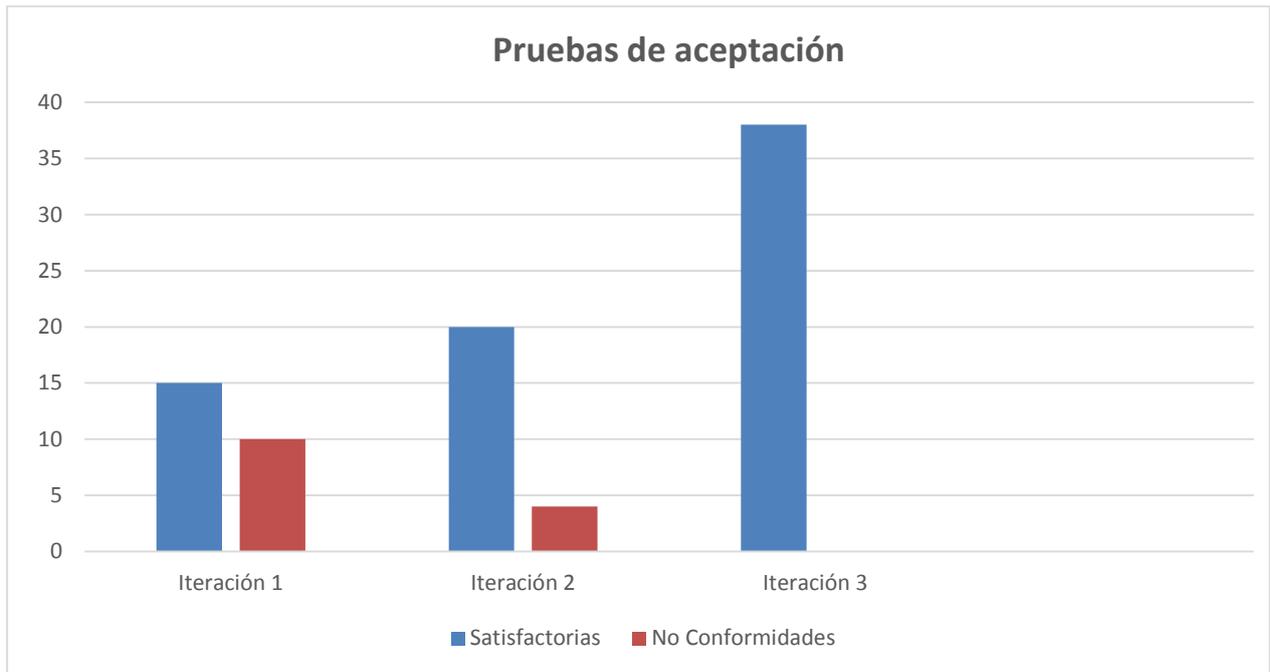


Figura 8: Resultados de las pruebas realizadas.

3.3 Diagrama de despliegue.

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Es utilizado para describir la vista de despliegue estática de una aplicación. Describe también la topología del sistema, es decir la estructura de los elementos de hardware y software que ejecuta cada uno de ellos. El diagrama de despliegue muestra la configuración de los elementos de hardware (nodos) y cómo los elementos y artefactos del software se trazan en esos nodos. Se representan con un conjunto de nodos y sus relaciones (32).

A continuación se muestra el diagrama de despliegue de la herramienta a implementar.

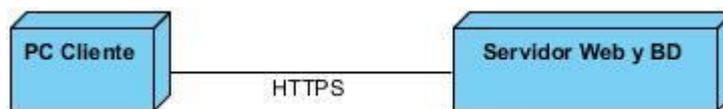


Figura 9: Diagrama de despliegue.

Capítulo 3: Implementación y Pruebas

Conclusiones Parciales

Con la finalización de la última fase de la metodología se cuenta con una aplicación que da solución a todas las funcionalidades descritas en las historias de usuario. Las tareas de ingeniería permitieron agilizar el proceso y evitar una sobrecarga de trabajo. A través de las pruebas de aceptación se validó el correcto funcionamiento del sistema, y se comprobó la acertada implementación de las historias de usuario definidas según las necesidades del cliente.

CONCLUSIONES GENERALES

El estudio de los fundamentos teóricos y del estado del arte relativo en la problemática identificada permitió fundamentar la necesidad del desarrollo de una herramienta que facilitara la gestión de la programación televisiva en el departamento de programación de la DIRCOM. La identificación de los requisitos y patrones sentaron las bases para la implementación de la herramienta web. Los resultados positivos de las pruebas ejecutadas permitieron asegurar un despliegue exitoso de la herramienta implementada. Y el uso de la aplicación en el departamento redundó en una mejor organización y control de los procesos que allí intervienen.

RECOMENDACIONES

Como parte del proceso de investigación y desarrollo de la aplicación, surgieron ideas que se deben tener en cuenta para el futuro mejoramiento de la solución propuesta, estas se mencionan a continuación:

- Poder utilizar la técnica de programación multi-hilos para ejecutar en segundo plano la conversión de videos de larga duración.
- Desarrollar un módulo que nos permita eliminar materiales audiovisuales no visto durante un tiempo determinado.

BIBLIOGRAFÍA

1. Sánchez, Maikel García. *Sistema automatizado para la gestión de las evaluaciones de desempeño de los profesores de la facultad 10 de la Universidad de las Ciencias Informáticas*. 2010.
2. informáticas, Herramientas. <http://www.mastermagazine.info/termino/5234.php#ixzz2qwSpRx1E>. [En línea]
3. Escalonilla, Gloria Gómez. *La programación televisiva en España. Estudio de las parrillas de programación televisiva*. Madrid : s.n., 1998.
4. Varela, Manuel. *La televisión: el espacio vacío de la crítica. Imagofagia*. 2015.
5. VSNCREATV. VSNCREATV. [En línea] <https://www.vsn-tv.com/es/soluciones-broadcast/automatizacion-de-tv/gestion-de-tv/>.
6. Televisiva, Sistema de Gestión. inTV. [En línea] <http://inTV.com>.
7. Server, Virtual Power Video. Virtual Power Video Server. [En línea] <http://www.impactovirtual.com/VirtualPowerVideo/>.
8. Périssé, M.C. *Una Metodología Simplificada*. 2012.
9. Schwaber K, Beedle M, Martin R.C. *Agile Software Development with SCRUM*. 2001.
10. Cadavid, A. N. *Metodologías ágiles para el desarrollo de software. Prospectiva*. 2013.
11. Milani, A. *PostgreSQL-Guía do Programador*. s.l. : Novatec Editora, 2008.
12. Gilfillan, Ian. *MySQL*. s.l. : Anaya Multimedia, 2003. 978-84-415-1558-1 84-415-1558-1.
13. Valbuena, Aponte y María, Angela. *Guía comparativa de Frameworks para los lenguajes HTML 5, CSS y JavaScript para el desarrollo de aplicaciones Web*. 2014.
14. HTML5: ¿Qué es? Y ¿Cómo usarlo? [En línea] <http://www.trazos-web.com/2010/02/01/html5-que-es-y-como-usarlo/>.
15. Guía breve de CSS. [En línea] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>, artículo[¿Para qué sirve CSS?].
16. ejschart. [En línea] <http://www.ejschart.com/documentation.php>.
17. Highcharts. [En línea] <http://www.highcharts.com/products/highcharts>.
18. Juan Eladio, Sánchez Rosas. Comparativa de frameworks en PHP: CakePHP, Symfony y Zend Framework. [En línea] 02 de 09 de 2007. <http://tuxpuc.pucp.edu.pe/articulo/comparativa-de-frameworks-en-php-cakephp-symfony-y-zend-framework>.
19. FRAMEWORK, Y. *Download Yii Framework*. 2011.
20. EcuRed. Bootstrap. [En línea] <http://www.ecured.cu/Bootstrap>.
21. NetBeans IDE. [En línea] 2015. <https://netbeans.org/features>.
22. DocumentObjectModel, Acrónimo de.

Bibliografía

23. CARACTERISTICAS Y FUNCIONES DEL UML. [En línea]
<http://luismanuelarandingua.blogspot.com/2009/07/introduccion-es-un-lenguaje-grafico.html>.
24. Pressman, Roger S. *Ingeniería de Software*. 2005. 0-07-285318-2.
25. Serie Científica de la Universidad de las Ciencias Informáticas. [En línea] 2012.
26. Requisitos de Software. [En línea] http://www.ecured.cu/Requisitos_de_Software.
27. tarjetas CRC. [En línea] <http://lsi.ugr.es/~mvega/docis/crc.pdf>.
28. González, Aleksander. *Método de evaluación de arquitecturas de software basadas en componentes*. s.l. : MECABIC, 2005.
29. Campo, Gustavo Damián. *Patrones de Diseño, Refactorización y Antipatrones, ventajas y Desventajas de su Utilización en el Software Orientado a Objetos*. 2009.
30. Larman, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. s.l. : Prentice Hall, 2006. 8420534382, 9788420534381.
31. Globe. Pruebas de rendimiento. [En línea] <http://www.globetesting.com/pruebas-de-rendimiento/>.
32. Diagrama de Despliegue. [En línea] 2015. <http://www.diadspg.blogspot.com>.

ANEXOS

Anexo 1 Historias de Usuario

Tabla 37: Anexo 1.1: HU8 Modificar canal.

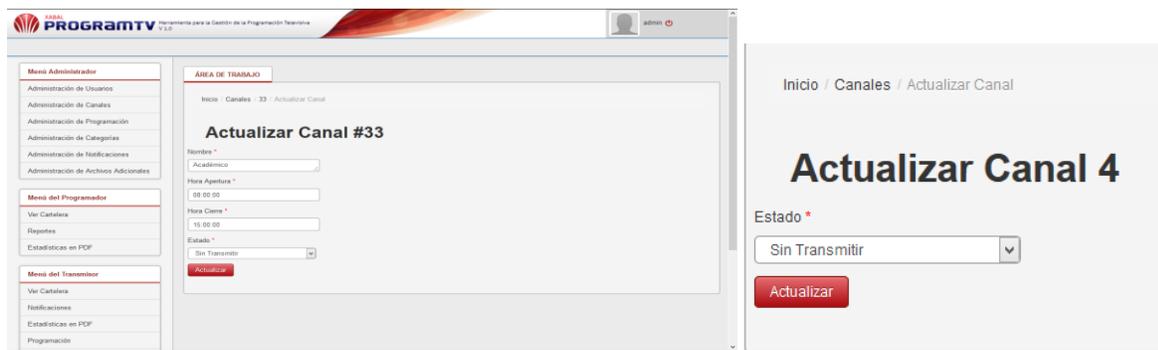
Historia de Usuario	
Número: 8	Usuario: Administrador, Programador y Transmisor
Nombre de Historia de Usuario: Modificar canal.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir actualizar un canal. Para ello dar clic en la opción Editar del canal específico para ser modificado. El usuario con el rol Transmisor solo va a poder actualizar el Estado de la transmisión y el usuario con rol de Administrador o Programador va a modificar todos los valores. A continuación una vista de modificar un canal.</p>	
Prototipo:	<div style="display: flex; justify-content: space-around;"> Rol de Programador o Administrador Rol de Transmisor </div>
 <p>The screenshot shows two views of the 'Actualizar Canal' interface. The left view is for 'Actualizar Canal #33' and includes a form with fields for 'Nombre *' (Académico), 'Hora Apertura *' (08:00:00), 'Hora Cierre *' (15:00:00), and 'Estado *' (Sin Transmitir). The right view is for 'Actualizar Canal 4' and shows a dropdown menu for 'Estado *' set to 'Sin Transmitir' and an 'Actualizar' button.</p>	

Tabla 38: Anexo 1.2: HU9 Eliminar canal.

Historia de Usuario	
Número: 9	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Eliminar canal.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir la eliminación de uno o varios canales. Para ello se listan todos los canales en la vista de administración, se selecciona uno o varios canales que se quieran eliminar, dando clic en el botón eliminar y luego dar aceptar en el mensaje que aparece (¿Seguro que desea borrar este elemento?) para eliminarlo.</p>	

Prototipo

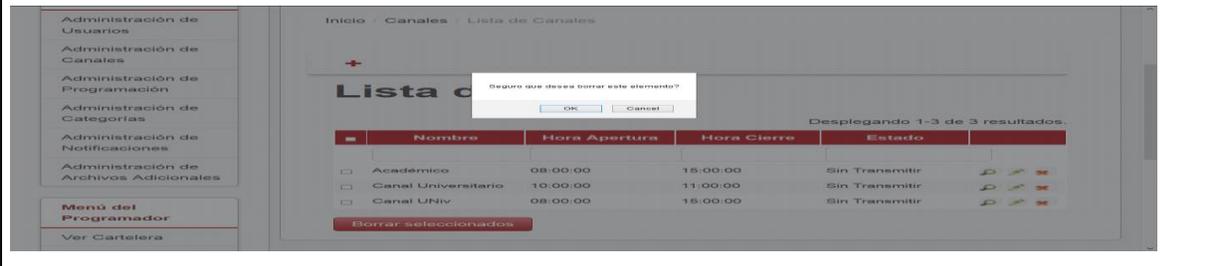


Tabla 39: Anexo 1.3: HU10 Listar canales.

Historia de Usuario	
Número: 10	Usuario: Administrador, Programador y Transmisor
Nombre de Historia de Usuario: Listar canales.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.2 semanas
<p>Descripción: La herramienta debe permitir listar todos los canales. Para ello se presiona la opción Administración de canales ubicada en el menú izquierdo y de esa forma se listan todos los canales en la parte derecha de la página.</p>	

Prototipo



Tabla 40: Anexo 1.4: HU11 Mostrar datos detallados de cada canal.

Historia de Usuario	
Número: 11	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Mostrar datos detallados de cada canal.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir ver los datos del canal seleccionado. Para ello se presiona la opción ver que aparece en la vista de administración de canales. En la nueva pestaña que se muestra encontrará todos los datos del canal.</p>	

Prototipo



Tabla 41: Anexo 1.5: HU19 Insertar categoría.

Historia de Usuario	
Número: 19	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Insertar categoría.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.2 semanas
<p>Descripción: La herramienta debe permitir adicionar una categoría. Para ello se le da clic en la opción Administrar categoría del menú izquierdo, aparecerá un listado de categoría, luego se debe dar clic en el botón (+) Agregar una categoría y seguidamente llenar el campo correspondiente, luego dar clic en el botón Crear. Seguidamente aparecerá un cartel mostrando (“Categoría Creada Satisfactoriamente”).</p>	
Prototipo	

Tabla 42: Anexo 1.6: HU20 Modificar categoría.

Historia de Usuario	
Número: 20	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Modificar categoría.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana

Descripción: La herramienta debe permitir modificar una categoría. Para ello dar en la opción Editar de la categoría específica para ser modificada. Aparecerá una vista con el dato a modificar de dicha categoría escogida, se le modifica el dato correspondiente. Luego dar clic en el botón Actualizar e inmediatamente aparecerá un cartel mostrando (“Categoría Modificada Satisfactoriamente”). A continuación una vista de modificar una categoría.

Prototipo



Tabla 43: Anexo 1.7: HU21 Eliminar categoría.

Historia de Usuario	
Número: 21	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Eliminar categoría.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir la eliminación de una o varias categorías. Para ello se listan todas las categorías en la vista de administración, se selecciona una o varias categorías que se quieran eliminar, dando clic en el botón eliminar y luego dar aceptar en el mensaje que aparece (¿Seguro que desea borrar este elemento?) para eliminarlo.</p>	
Prototipo	

Tabla 44: Anexo 1.8: HU22 Listar categoría.

Historia de Usuario	
Número: 22	Usuario: Administrador, Programador

Nombre de Historia de Usuario: Listar categoría.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
Descripción: La herramienta debe permitir listar todas las categorías. Para ello se presiona la opción Administración de categoría y de esa forma se listan en la parte derecha de la página.	
Prototipo	
	

Tabla 45: Anexo 1.9: HU23 Mostrar datos detallados de cada categoría.

Historia de Usuario	
Número: 23	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Mostrar datos detallados de cada categoría.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
Descripción: La herramienta debe permitir ver los datos de cada categoría seleccionada. Para ello en la categoría seleccionada se presiona la opción Ver, que aparece en la vista de Administración de Categoría. En la nueva pestaña que se muestra encontrará todos los datos de la categoría seleccionada.	
Prototipo	
	

Anexos

Tabla 46: Anexo 1.10: HU24 Listar estadísticas de programación en formato PDF.

Historia de Usuario	
Número: 24	Usuario: Administrador, Programador y Transmisor
Nombre de Historia de Usuario: Listar estadísticas de programación en formato PDF.	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir visualizar en formato PDF una estadística de programación realizada hasta el momento. Para ello se debe de dar clic en la opción Estadísticas en PDF del menú izquierdo y aparecerán las estadísticas en una tabla dibujada en formato PDF.</p>	
<p>Prototipo</p> 	

Tabla 47: Anexo 1.11: HU25 Insertar archivo adicional.

Historia de Usuario	
Número: 25	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Insertar archivo adicional.	
Prioridad en Negocio: Media	Iteración Asignada: 2
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.1 semanas
<p>Descripción: La herramienta debe permitir adicionar un archivo adicional. Para ello se le da clic en la opción Administración de Archivos Adicionales del menú izquierdo, aparecerá una vista con todos los archivos adicionales, luego se debe dar clic en el botón Administración donde aparecerá la administración de los archivos adicionales, dar clic en el botón agregar (+) un archivo y llenar los campos correspondientes y por último accionar el botón crear. Aparecerá un cartel con el mensaje (“Se ha creado un archivo adicional correctamente.”)</p>	

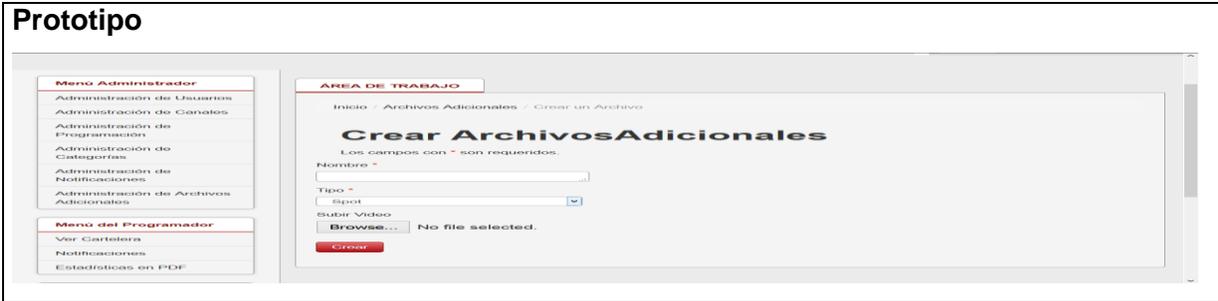


Tabla 48: Anexo 1.12: HU26 Modificar archivo adicional.

Historia de Usuario	
Número: 26	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Modificar archivo adicional.	
Prioridad en Negocio: Media	Iteración Asignada: 2
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir modificar un archivo adicional. Para ello se le da clic en el botón Administración de Administración de Archivos Adicionales donde aparecerá la lista de archivos adicionales, dar clic en el botón Editar correspondiente al archivo y modificar los campos necesarios, por último accionar el botón Actualizar. Aparecerá un cartel con el mensaje (“Se ha actualizado un archivo adicional correctamente.”)</p>	
Prototipo	

Tabla 49: Anexo 1.13: HU27 Eliminar archivo adicional.

Historia de Usuario	
Número: 27	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Eliminar archivo adicional.	
Prioridad en Negocio: Media	Iteración Asignada: 2
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana

Anexos

Descripción: La herramienta debe permitir la eliminación de uno o varios archivos adicionales. Para ello se listan todos los archivos en la vista de administración de archivos adicionales, se selecciona uno o varios archivos que se quieran eliminar, dando clic en el botón eliminar y luego dar aceptar en el mensaje que aparece (¿Seguro que desea borrar este elemento?) para eliminarlo.

Prototipo

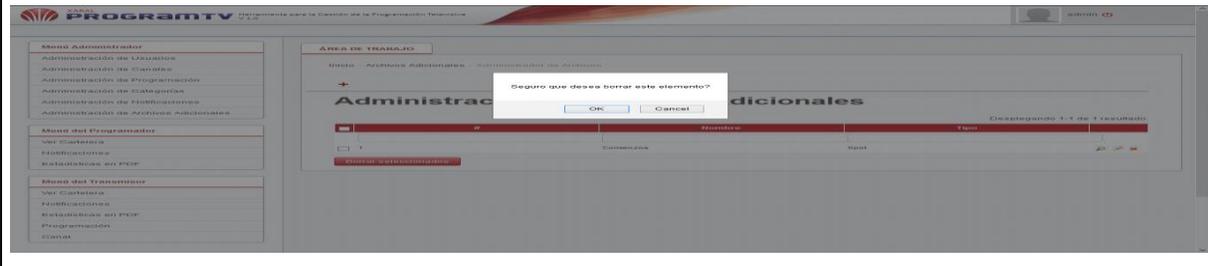


Tabla 50: Anexo 1.14: HU28 Listar archivo adicional.

Historia de Usuario	
Número: 28	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Listar archivo adicional.	
Prioridad en Negocio: Media	Iteración Asignada: 2
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
Descripción: La herramienta debe permitir listar todos los archivos adicionales. Para ello se da clic en la opción Administración de Archivos Adicionales y de esa forma se listan en la parte derecha de la página todos los archivos.	
Prototipo	

Tabla 51: Anexo 1.15: HU29 Mostrar datos detallados de cada archivo adicional.

Historia de Usuario	
Número: 29	Usuario: Administrador, Programador.
Nombre de Historia de Usuario: Mostrar datos detallados de cada archivo adicional.	

Anexos

Prioridad en Negocio: Media	Iteración Asignada: 2
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir ver los datos detallados de cada archivo adicional seleccionado. Para ello se presiona la opción Ver que aparece en la vista de administración de archivo adicional. En la nueva pestaña que se muestra encontrará todos los datos del archivo seleccionado.</p>	
<p>Prototipo</p> 	

Tabla 52: Anexo 1.16: HU30 Insertar notificación.

Historia de Usuario	
Número: 30	Usuario: Administrador, Transmisor, Usuario
Nombre de Historia de Usuario: Insertar notificación.	
Prioridad en Negocio: Media	Iteración Asignada: 2
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir adicionar una notificación. Para ello se le da clic en la opción Notificaciones del menú izquierdo, luego se debe accionar el botón Agregar y seguidamente llenar los campo correspondientes, luego dar clic en el botón Crear y aparecerá un cartel informando (“Notificación Realizada Satisfactoriamente, se le ha enviado un correo al programador”).</p>	
<p>Prototipo</p> 	

Tabla 53: Anexo 1.17: HU31 Modificar notificación.

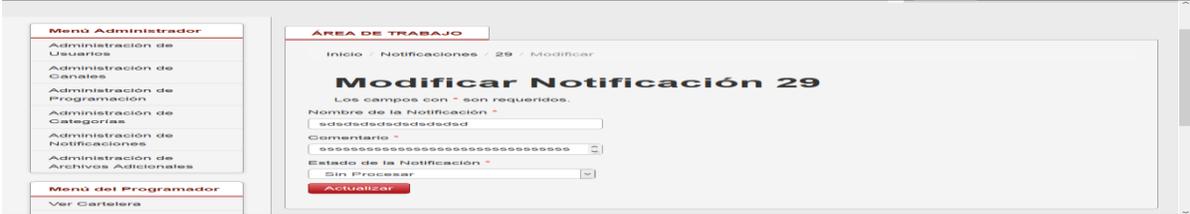
Historia de Usuario	
Número: 31	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Modificar notificación.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir modificar una notificación. Para ello se le da en la opción Editar de la notificación específica para ser modificada. Aparecerá una vista con los datos a modificar de dicha categoría escogida, se le modifican los datos correspondientes. Luego dar clic en el botón Actualizar y aparecerá un cartel mostrando (“Notificación Modificada Satisfactoriamente”).</p>	
<p>Prototipo</p> 	

Tabla 54: Anexo 1.18: HU32 Eliminar notificación.

Historia de Usuario	
Número: 32	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Eliminar notificaciones.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir la eliminación de una o varias notificaciones. Para ello se listan todas las notificaciones en la vista de Administración de Notificación, se se selecciona una o varias notificaciones que se quieran eliminar, dando clic en el botón eliminar y luego dar aceptar en el mensaje que aparece (¿Seguro que desea borrar este elemento?) para eliminarlo.</p>	

Prototipo



Tabla 55: Anexo 1.19: HU33 Listar Notificaciones realizadas.

Historia de Usuario	
Número: 33	Usuario: Administrador, Transmisor, Usuario
Nombre de Historia de Usuario: Listar notificaciones realizadas.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
Descripción: La herramienta debe permitir listar todas las notificaciones. Para ello se presiona la opción Administración de Notificaciones y de esa forma se listan en la parte derecha de la página. Los usuarios que no posean el rol de Programador ni de Administrador presionan la opción Notificaciones en el menú izquierdo, luego se muestran las notificaciones en la parte derecha de la página.	

Prototipo



Tabla 56: Anexo 1.20: HU34 Mostrar datos detallados de cada notificación.

Historia de Usuario	
Número: 34	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Mostrar datos detallados de cada categoría.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.1 semana
Descripción: La herramienta debe permitir ver los datos de cada notificación seleccionada. Para ello se presiona la opción Ver que aparece en la vista de Administración de	

Anexos

Notificaciones. En la nueva pestaña que se muestra encontrará todos los datos de la notificación seleccionada.

Prototipo



Tabla 57: Anexo 1.21: HU35 Enviar notificación vía correo.

Historia de Usuario	
Número: 35	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Enviar notificación vía correo.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.2 semanas
Descripción: La herramienta debe permitir enviar un correo electrónico al usuario con rol Programador con la descripción de la notificación y el usuario de quien la realizó. Para ello se crea la notificación y luego de ser insertada es enviado automáticamente el correo.	
Prototipo	
<p>ProgramTV. Herramienta para la Planificación Televisiva</p> <p>Programador Mediante este email se le notifica que se ha realizado un notificación en la aplicación. A continuación una breve descripción: Posible programadÃn repetida en el canal 22 Notificación realizado poradmin</p> <p>PROGRAMTV</p>	

Tabla 58: Anexo 1.22: HU36 Buscar cartelera dada la fecha.

Historia de Usuario	
Número: 36	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Buscar cartelera dada la fecha.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Medio	Puntos Estimados: 0.2 semanas
Descripción: La herramienta debe permitir buscar la cartelera dada una fecha en específica. Para ello el usuario debe de acceder a la opción Administrar programación en el menú	

Anexos

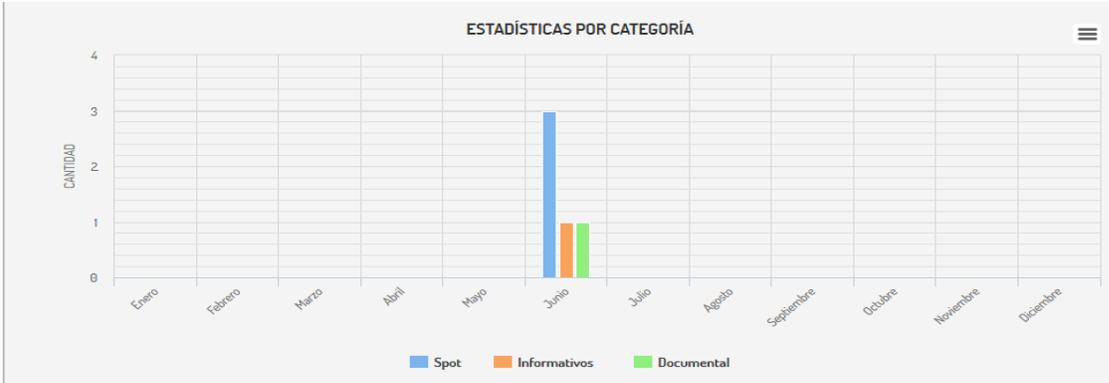
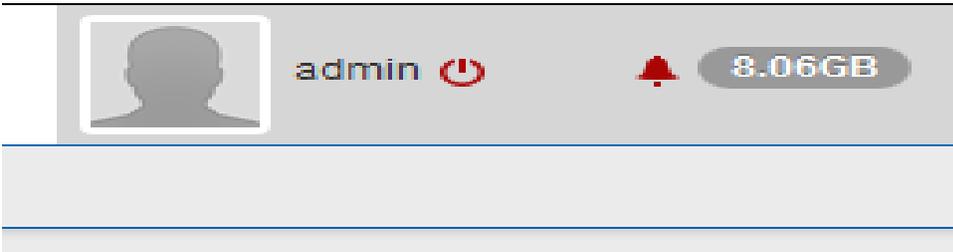
Prioridad en Negocio: Bajo	Iteración Asignada: 3																																																				
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.1 semana																																																				
<p>Descripción: La herramienta debe permitir mostrar en forma de gráfica la cantidad de programación realizada por categoría. Para ello se debe de ir a Administración de programación y dar clic en el botón Gráfica de Estadísticas localizado en la parte superior de la vista, luego se mostrará dicha gráfica.</p>																																																					
<p>Prototipo</p>  <table border="1"> <caption>ESTADÍSTICAS POR CATEGORÍA</caption> <thead> <tr> <th>Mes</th> <th>Spot</th> <th>Informativos</th> <th>Documental</th> </tr> </thead> <tbody> <tr> <td>Enero</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Febrero</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Marzo</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Abril</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Mayo</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Junio</td> <td>3</td> <td>1</td> <td>1</td> </tr> <tr> <td>Julio</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Agosto</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Septiembre</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Octubre</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Noviembre</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Diciembre</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		Mes	Spot	Informativos	Documental	Enero	0	0	0	Febrero	0	0	0	Marzo	0	0	0	Abril	0	0	0	Mayo	0	0	0	Junio	3	1	1	Julio	0	0	0	Agosto	0	0	0	Septiembre	0	0	0	Octubre	0	0	0	Noviembre	0	0	0	Diciembre	0	0	0
Mes	Spot	Informativos	Documental																																																		
Enero	0	0	0																																																		
Febrero	0	0	0																																																		
Marzo	0	0	0																																																		
Abril	0	0	0																																																		
Mayo	0	0	0																																																		
Junio	3	1	1																																																		
Julio	0	0	0																																																		
Agosto	0	0	0																																																		
Septiembre	0	0	0																																																		
Octubre	0	0	0																																																		
Noviembre	0	0	0																																																		
Diciembre	0	0	0																																																		

Tabla 61: Anexo 1.25: HU39 Mostrar alerta de poco espacio en disco

Historia de Usuario	
Número: 39	Usuario: Administrador, Programador
Nombre de Historia de Usuario: Mostrar alerta de poco espacio en el disco.	
Prioridad en Negocio: Bajo	Iteración Asignada: 3
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.1 semana
<p>Descripción: La herramienta debe permitir mostrar una alerta y seguidamente el número de gigas correspondiente al espacio libre que posee el disco duro de la pc. Para ello cuando el disco presente una capacidad libre de 10 GB automáticamente aparecerá la alerta visible en la parte superior izquierda de la vista.</p>	
<p>Prototipo</p> 	

Anexo 2 Tareas de Ingeniería

Tabla 62: Anexo 2.1 Tarea de Ingeniería 1 de la HU13.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 13
Nombre de Tarea: Modificar programación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1 semana
Fecha Inicio: 8/3/2016	Fecha Fin: 9/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario con rol de programador o administrador pueda modificar una programación.	

Tabla 63: Anexo 2.1 Tarea de Ingeniería 1 de la HU8.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 8
Nombre de Tarea: Modificar canal.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1 semana
Fecha Inicio: 23/3/2016	Fecha Fin: 24/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario con rol de programador o administrador pueda modificar un canal.	

Tabla 64: Anexo 2.1 Tarea de Ingeniería 1 de la HU20.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 20
Nombre de Tarea: Modificar categoría.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1 semana
Fecha Inicio: 29/3/2016	Fecha Fin: 30/3/2016
Programador Responsable: Juan J Pérez.	
Descripción: Se implementan las funcionalidades necesarias para que el usuario con rol de programador o administrador pueda modificar un canal.	

Resultados de la evaluación

Anexo 3 Resultados de la evaluación

Anexo 3.1: Reconocimiento en la Jornada Científica Estudiantil.

