

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

**“SISTEMA WEB DE GESTION DE AUDIO STREAMING
ORIENTADO A DISEÑOS ADAPTIVOS
CASO: PRODUCTORA LOS SUPERCLASICOS”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POSTULANTE: OVIDIO ROJAS NINA
TUTOR METODOLOGICO: M. Sc. ALDO RAMIRO VALDEZ ALVARADO
ASESOR: M. Sc. CARLOS MULLISACA CHOQUE**

**LA PAZ – BOLIVIA
2015**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A Dios por estar siempre a mi lado y brindarme en todo momento mucha fortaleza.

A mi madre, Eustaquia Nina Mamani por su infinito amor, sacrificio y ejemplo de vida. El trabajo va dedicado especialmente a ella. La quiero mucho y Dios la bendiga.

A mi padre, Pedro Rojas Yupanqui por haberme educado de manera correcta.

A mi hermanita, Erika Carla Rojas Nina por haberme alentado y apoyado en todo momento.

Ovidio Rojas Nina

AGRADECIMIENTOS

Infinitamente a nuestro creador por estar siempre a mi lado, por acompañarme en todo momento con la luz de su sabiduría y permitirme llegar hasta este punto, como la culminación de una de mis metas.

A nuestra casa superior de estudios por acogerme durante el periodo que me tomo concluir la carrera, siendo fuente de saber y conocimiento en mi formación profesional.

Al motor más grande mi mamita Eustaquia Nina Mamani, gracias por darme la vida por ser tan paciente, por tu comprensión de todos estos años, por tu ayuda incondicional, tus palabras, tus consejos, por el esfuerzo que hiciste de apoyarme siempre. Gracias por todo mama, que Dios siempre te bendiga.

A mi papa Pedro Rojas Yupanqui por ser la fuerza de la familia, por enseñarme y guiarme. Gracias por la educación que me brindaste y el apoyo que siempre tienes.

A mi hermanita Erika Carla Rojas Nina por ser siempre un gran apoyo y motivación sobre todo de hacer lo que más le gusta a uno.

A mi tutor M. Sc. Aldo Ramiro Valdez Alvarado quien me guio a lo largo de todo este tiempo, quien me demostró que el conocimiento y el buen humor hacen que las personas sean profesionales de calidad y sobre todo darle las gracias por dedicarle tiempo para la culminación del proyecto.

A mi asesor M. Sc. Carlos Mullisaca Choque quien me oriento, guio y sugirió en todo momento, gracias por confiar en mi trabajo, dedicarme tiempo con buen ánimo para la colaboración del presente proyecto.

Al gerente de la productora LOS SUPER CLASICOS Henry Larrea Alurralde por haber confiado en mí, en el desarrollo del proyecto y por su predisposición en todo momento.

A todos mis amigos con los que compartí momentos inolvidables en todo este tiempo, por ser parte de recuerdos añorables, gracias por ser parte de mi vida.

RESUMEN

El “Sistema Web de Gestion de Audio Streaming Orientado a Diseños Adaptativos”, Caso productora los SuperClasicos implementa un servidor de streaming para emitir contenido de audio por la organización en tiempo real el mismo se replica en un sistema web y un aplicativo móvil. A su vez el sistema web permite gestionar contenido relacionado con la productora.

El desarrollo del sistema se lo realizo bajo ciertas metodologías, es así que como metodologías de desarrollo se utilizó XP (Programación Extrema) para el sistema web y Mobile-D para el aplicativo móvil. Las mismas ofrecen versatilidad al momento de desarrollar mediante las iteraciones y contar con artefactos que facilitan la elaboración del proyecto.

Para complementar el sistema web, se utilizó WebML (Web Modeling Language) que es un lenguaje de modelado para la especificación de diseños web, estructurando de mejor manera el sistema.

La calidad del sistema se lo realizo bajo el estándar ISO 9126 que evalúa aspectos como funcionalidad, fiabilidad, usabilidad y mantenibilidad entre otros, proporcionando una evaluación tras la implementación del sistema.

El costo beneficio del sistema se lo realizo en primera instancia sobre COCOMO II (Modelo Constructivo de Costes), posteriormente se cálculo el VAN (Valor Actual Neto), TIR (Tasa Interna de Retorno) para conocer el costo, viabilidad y rentabilidad del proyecto respectivamente.

Finalmente se presenta las conclusiones y recomendaciones. Las Conclusiones que reflejaran los objetivos alcanzados y obtenidos con el presente trabajo. Y las recomendaciones a la institución para futuros proyectos.

Palabras Claves: Sistema Web, Servidor de audio streaming, Diseños Adaptativos, Aplicativo Móvil, Metodología XP, Metodología Mobile-D, WebML.

SUMMARY

The "Web Management System Oriented Adaptive Streaming Audio Designs" Case for the super producer implements a streaming server to broadcast audio content by the organization in real time it replicates in a Web system and a mobile application. In turn the web system to manage production related content.

The development of the system would perform under certain methodologies, so that as development methodologies used XP (Extreme Programming) for the Web and Mobile-D system for mobile application. They offer versatility when developing through iterations and have devices that facilitate the development of the project.

To complement the web system, WebML (Web Modeling Language) is a modeling language for specifying web designs, better structured system was used.

The quality system would perform under the ISO 9126 standard that evaluates aspects such as functionality, reliability, usability and maintainability among others, providing an evaluation after the implementation of the system.

The cost benefit of the system is performed in the first instance II COCOMO (Constructive Cost Model), then the NPV (Net Present Value), IRR (Internal Rate of Return) was calculated to meet the cost, viability and profitability of the project respectively .

Finally, conclusions and recommendations are presented. Conclusions reflecting the objectives achieved and obtained with this study. And recommendations to the institution for future projects.

Keywords: System Web, Server streaming audio, Adaptive Designs, Aplicativo Mobile, XP Methodology, Methodology Mobile-D, WebML.

ÍNDICE

CAPÍTULO 1	1
MARCO INTRODUCTORIO.....	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES	2
1.2.1 ANTECEDENTES INSTITUCIONALES.....	2
1.2.1.1 ORIGEN DE LOS SUPER CLASICOS	2
1.2.1.2 LOS SUPER CLASICOS PRODUCTORA.....	3
1.2.1.3 LOS SUPER CLASICOS EVENTOS.....	3
1.2.1.4 SECTORES RADIO Y TELEVISIÓN	3
1.2.1.5 FORTALEZA DE LA EMPRESA LOS SUPER CLASICOS	4
1.2.1.6 MATERIAL DE LOS SUPER CLASICOS	4
1.2.1.7 NOVEDADES PARA MÁS ADELANTE	5
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES	5
1.3 PLANTEAMIENTO DEL PROBLEMA.....	7
1.3.1. PROBLEMA CENTRAL.....	7
1.3.2. PROBLEMAS SECUNDARIOS	7
1.4 DEFINICIÓN DE OBJETIVOS.....	8
1.4.1 OBJETIVO GENERAL	8
1.4.2 OBJETIVOS ESPECÍFICOS	8
1.5 JUSTIFICACIÓN.....	9
1.5.1 JUSTIFICACIÓN ECONÓMICA.....	9
1.5.2 JUSTIFICACIÓN SOCIAL	9
1.5.3 JUSTIFICACIÓN TECNOLÓGICA	10
1.6 ALCANCES Y LÍMITES	10
1.6.1 ALCANCES	10
1.6.2 LÍMITES	11
1.7 APORTES	11
1.7.1 PRÁCTICO	11
1.7.2 TEÓRICO.....	12

1.8 METODOLOGÍA.....	12
1.8.1 METODOLOGÍA DE DESARROLLO	13
1.8.2 TÉCNICA.....	13
1.8.3 HERRAMIENTAS.....	13
CAPÍTULO 2	14
MARCO TEÓRICO	14
2.1 INTRODUCCIÓN.....	14
2.2 INGENIERÍA DEL SOFTWARE.....	14
2.2.1 INGENIERÍA	15
2.2.2 SOFTWARE.....	16
2.2.3 CARACTERÍSTICAS DEL SOFTWARE	16
2.2.4 CICLOS DE VIDA DE UN SOFTWARE.....	17
2.2.4.1 ETAPAS.....	17
2.3 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	19
2.3.1 VENTAJAS DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE... 20	
2.4 METODOLOGÍAS DE DESARROLLO TRADICIONALES.....	20
2.5 METODOLOGÍAS DE DESARROLLO ÁGIL	21
2.6 DIFERENCIAS ENTRE METODOLOGÍAS TRADICIONALES Y ÁGILES	21
2.7 EL POR QUÉ USAR METODOLOGÍAS ÁGILES.....	22
2.8 METODOLOGÍA DE PROGRAMACIÓN EXTREMA (XP).....	22
2.8.1 ELEMENTOS DE LA METODOLOGÍA	22
2.8.1.1 HISTORIAS DE USUARIOS	22
2.8.1.2 ROLES XP	22
2.8.1.3 PROCESO XP	23
2.8.2 CICLO DE VIDA DE XP	24
2.8.2.1 EXPLORACIÓN	24
2.8.2.2 PLANIFICACIÓN DE LA ENTREGA	24
2.8.2.3 ITERACIONES	25
2.8.2.4 PRODUCCIÓN	25
2.8.2.5 MANTENIMIENTO	25

2.8.2.6 MUERTE DEL PROYECTO.....	25
2.8.3 PRACTICAS XP	25
2.8.4 PRINCIPIOS DE XP	28
2.8.5 ACTIVIDADES DE XP	29
2.8.5.1 CODIFICAR.....	29
2.8.5.2 PROBAR	29
2.8.5.3 ESCUCHAR.....	30
2.8.5.4 DISEÑAR.....	30
2.8.6 ARTEFACTOS XP	30
2.8.6.1 HISTORIAS DE USUARIO	30
2.8.6.2 TARJETAS TAREAS	31
2.8.6.3 TARJETAS CRC.....	32
2.8.6.4 PRUEBAS DE ACEPTACIÓN	33
2.9 MODELADO DE DATOS.....	33
2.10 INGENIERÍA WEB	34
2.11 WEBML (WEB MODELING LANGUAGE).....	35
2.11.1 MODELO DE DATOS	36
2.11.2 MODELO DE HIPERTEXTO	38
2.11.3 MODELO DE PRESENTACIÓN.....	39
2.11.4 ELEMENTOS DEL MODELADO DE HIPERTEXTO WebML	40
2.11.4.1 UNIDADES DE DATOS	40
2.11.4.2 UNIDADES DE MULTIDATOS	41
2.11.4.3 UNIDADES DE DESPLAZAMIENTO	41
2.11.4.4 UNIDADES DE FILTRO	42
2.11.4.5 PÁGINAS.....	43
2.12 INGENIERÍA MÓVIL.....	45
2.12.1 INTRODUCCIÓN.....	45
2.12.2 CARACTERÍSTICAS DEL ENTORNO MÓVIL.....	45
2.12.3 CICLO DE VIDA DEL SOFTWARE PARA DISPOSITIVOS MÓVILES	47
2.12.3 METODOLOGÍAS AL DESARROLLO DE APLICACIONES MÓVILES.....	47

2.13 MOBILE-D.....	48
2.13.1 CICLO DE VIDA DE MOBILE-D.....	48
CAPÍTULO 3	51
MARCO APLICATIVO	51
3.1 INTRODUCCIÓN.....	51
3.1.1 INTERRELACIÓN DE METODOLOGÍAS	51
3.2 PLANIFICACIÓN, EXPLORACIÓN E INICIALIZACIÓN	53
3.2.1 ESTABLECIMIENTO DE USUARIOS XP y MOBILE-D	53
3.2.2 REQUERIMIENTOS DEL PRODUCTO XP y MOBILE-D	54
3.2.3 HISTORIAS DE USUARIO XP y MOBILE-D	54
3.2.4 TARJETA DE TAREA XP y MOBILE-D	58
3.2.5 PLAN DE ENTREGAS	70
3.2.6 VELOCIDAD DEL PROYECTO.....	71
3.2.7 REUNIONES	72
3.2.9 METÁFORA DEL SISTEMA	72
3.2.10 ESTABLECIMIENTO DE RECURSOS MOBILE-D	73
3.3 DISEÑO	74
3.3.1 TARJETAS CRC. XP	74
3.3.2 MODELO ESTRUCTURAL WEBML	76
3.3.3 PRIMERA ITERACIÓN XP y WEB ML.....	78
3.3.4 SEGUNDA ITERACIÓN XP y WEB ML	79
3.3.5 TERCERA ITERACIÓN XP y WEB ML	81
3.4 DESARROLLO, PRODUCCIÓN Y ESTABILIZACIÓN.....	83
3.4.1 DESARROLLO XP.....	84
3.4.1.1 PRIMERA ITERACIÓN.....	84
3.4.1.2 SEGUNDA ITERACIÓN.....	85
3.4.1.3 TERCERA ITERACIÓN	87
3.4.2 PRODUCCIÓN MOBILE-D.....	89
3.4.2.1 CUARTA ITERACIÓN PARTE 1.....	89
3.4.2.2 CUARTA ITERACIÓN PARTE 2.....	91

3.4.3 ESTABILIZACIÓN MOBILE-D.....	92
3.5 PRUEBAS	93
3.5.1 PRUEBAS DE ACEPTACIÓN XP y MOBILE-D.....	93
3.5.2 PRUEBAS DE STRESS XP	97
3.5.4 PRUEBAS DE FUNCIONAMIENTO MOBILE-D	99
CAPÍTULO 4	100
CALIDAD Y SEGURIDAD	100
4.1 INTRODUCCIÓN.....	100
4.2 CALIDAD DE SOFTWARE	100
4.3 ISO 9126.....	101
4.3.1 FUNCIONALIDAD	101
4.3.2 FIABILIDAD	104
4.3.3 USABILIDAD.....	105
4.3.4 MANTENIBILIDAD	107
4.3.5 PORTABILIDAD.....	108
4.3.6 EFICIENCIA.....	108
4.4 CALIDAD GLOBAL.....	109
4.5 SEGURIDAD	110
4.5.1 POLÍTICAS DE CONTROL DE ACCESO AL SISTEMA WEB	110
4.5.2 POLÍTICAS DE RESPALDO A LA BASE DE DATOS.....	110
4.5.3 POLÍTICAS DE REGISTROS DE EVENTOS	111
CAPÍTULO 5	112
COSTO/BENEFICIO	112
5.1 INTRODUCCIÓN.....	112
5.2 COCOMO II.....	112
5.3 COSTO DEL SISTEMA	114
5.3.1 COSTO DE DESARROLLO DEL SOFTWARE.....	114
5.3.2 COSTO DE IMPLEMENTACIÓN.....	117
5.3.3 COSTO DE ELABORACIÓN DEL PROYECTO	117
5.3.4 COSTO TOTAL DEL SOFTWARE.....	117

5.4 VALOR ACTUAL NETO.....	118
5.4.1 COSTO / BENEFICIO	119
5.5 TASA INTERNA DE RETORNO.....	119
CAPÍTULO 6	121
CONCLUSIONES Y RECOMENDACIONES	121
6.1 INTRODUCCIÓN.....	121
6.2 CONCLUSIONES.....	121
6.3 RECOMENDACIONES	122

ÍNDICE DE FIGURAS

Figura 2.1. Componentes del software	16
Figura 2.2. Ciclo de vida de XP.....	24
Figura 2.3. Practicas XP	26
Figura 2.4. Ejemplo diagrama "Entidad - Relación"	37
Figura 2.5. Ejemplo diagrama de clases	37
Figura 2.6. Notación para el esquema de estructura.....	38
Figura 2.7. Notación para composición y navegación.....	39
Figura 2.8. Enfoque WebML.....	40
Figura 2.9. Notación para unidades de datos.....	40
Figura 2.10. Notación para unidades multidatos	41
Figura 2.11. Notación para unidades de desplazamiento	42
Figura 2.12. Notación para unidades de filtro	42
Figura 2.13. Notación de texto y grafica en páginas anidadas	43
Figura 2.14. Notación de texto y grafica en páginas alternas.....	44
Figura 2.15. Fases y etapas de Mobile-D	49
Figura 3.1. Diagrama de clases.....	77
Figura 3.2. Diagrama de hipertexto Acceso a recursos al usuario final	78
Figura 3.3. Diagrama de presentación Acceso a recursos al usuario final	79
Figura 3.4. Diagrama de hipertexto Portal de información	80
Figura 3.5. Diagrama de presentación Portal de información	80
Figura 3.6. Diagrama de hipertexto Portal de auspiciadores	81
Figura 3.7. Diagrama de hipertexto Portal de contactos.....	82
Figura 3.8. Diagrama de presentación Portal de auspiciadores	82
Figura 3.9. Diagrama de presentación Portal de contactos.....	83
Figura 3.10. Pantalla Portal de la Productora vista ordenador de escritorio	84
Figura 3.11. Pantalla Portal de la Productora vista teléfono inteligente.....	85
Figura 3.12. Pantalla Portal de información la Productora vista ordenador de escritorio	86
Figura 3.13. Pantalla Portal de Información la Productora vista teléfono inteligente.....	86
Figura 3.14. Pantalla Portal de auspiciadores la Productora vista ordenador de escritorio..	87
Figura 3.15. Pantalla Portal de contactos de la Productora vista ordenador de escritorio....	88
Figura 3.16. Pantalla Portal de Auspiciadores y Contactos vista teléfono inteligente	89
Figura 3.17. Captura de pantalla Liberación Historia de usuario 7 Tarjeta tarea 1.1	90
Figura 3.18. Captura de pantalla Liberación Historia de usuario 7 tarjeta tarea 1.2	92
Figura 3.19. Fase de estabilización Integración de acciones del aplicativo	92
Figura 3.20. Datos transferidos, memoria del sistema y carga de la CPU	97
Figura 3.21. Solicitud de apertura y transferencia de datos.....	98
Figura 3.22. Peticiones de usuario.....	98
Figura 3.23. Capacidad de carga del sistema.....	99

ÍNDICE DE TABLAS

Tabla 2.1. Metodologías tradicionales vs agiles.....	21
Tabla 2.2. Plantilla "historia de usuario"	31
Tabla 2.3. "Tarjeta de Ingeniería"	31
Tabla 2.4. Modelo tarjeta CRC.....	32
Tabla 2.5. Tarjeta "Casos de prueba"	33
Tabla 3.1. Interrelación de metodologías	51
Tabla 3.2. Establecimiento de usuarios del sistema web.....	53
Tabla 3.3. Establecimiento de usuarios del aplicativo móvil	53
Tabla 3.4. Requerimientos del producto web y móvil.....	54
Tabla 3.5. Historia de usuario Nro. 1	55
Tabla 3.6. Historia de usuario Nro. 2	55
Tabla 3.7. Historia de usuario Nro. 3	56
Tabla 3.8. Historia de usuario Nro. 4	56
Tabla 3.9. Historia de usuario Nro. 5	57
Tabla 3.10. Historia de usuario Nro. 6	57
Tabla 3.11. Historia de usuario Nro. 7	58
Tabla 3.12. Tarjeta de Tarea Implementacion modulo levantar servidor de streaming	58
Tabla 3.13. Tarjeta de Tarea Obtención de la direccion IP del servidor streaming	59
Tabla 3.14. Tarjeta de Tarea Maquetación del sistema web	59
Tabla 3.15. Tarjeta de Tarea Maquetación ordenadores de escritorio.....	60
Tabla 3.16. Tarjeta de Tarea Maquetación laptops	60
Tabla 3.17. Tarjeta de Tarea Maquetación tablets.....	61
Tabla 3.18. Tarjeta de Tarea Maquetación telefonos inteligentes	61
Tabla 3.19. Tarjeta de Tarea Módulo de acceso de recursos al usuario final.....	62
Tabla 3.20. Tarjeta de Tarea Implementación del Diagrama de clases	62
Tabla 3.21. Tarjeta de Tarea Desarrollo de interfaz adaptable.....	63
Tabla 3.22. Tarjeta de Tarea Implementación de cartel de anuncios	63
Tabla 3.23. Tarjeta de Tarea Implementación de redes sociales de la productora	64
Tabla 3.24. Tarjeta de Tarea Implementación del reproductor de audio.....	64
Tabla 3.25. Tarjeta de Tarea Implementación Módulo portal de informacion.....	65
Tabla 3.26. Tarjeta de Tarea Implementación del Diagrama de clases	65
Tabla 3.27. Tarjeta de Tarea Desarrollo de interfaz adaptable.....	66
Tabla 3.28. Tarjeta de Tarea Implementación Módulo portal de auspiciadores	66
Tabla 3.29. Tarjeta de Tarea Implementación del Diagrama de clases	67
Tabla 3.30. Tarjeta de Tarea Desarrollo de interfaz adaptable.....	67
Tabla 3.31. Tarjeta de Tarea Implementación Módulo portal de contactos	68
Tabla 3.32. Tarjeta de Tarea Implementación del Diagrama de clases	68
Tabla 3.33. Tarjeta de Tarea Desarrollo de interfaz adaptable.....	69

Tabla 3.34. Tarjeta de Tarea Diseño de la interfaz de la aplicación movil	69
Tabla 3.35. Tarjeta de Tarea Integración de la dirección IP del servidor streaming.....	70
Tabla 3.36. Plan de entregas	71
Tabla 3.37. Planificación de iteraciones	72
Tabla 3.38. Metáfora de sistema.....	72
Tabla 3.39. Glosario de términos.....	73
Tabla 3.40. Tarjeta C.R.C. Servidor de streaming	74
Tabla 3.41. Tarjeta C.R.C. Aplicativo web	74
Tabla 3.42. Tarjeta C.R.C. Portal productora.....	75
Tabla 3.43. Tarjeta C.R.C. Portal de información	75
Tabla 3.44. Tarjeta C.R.C. Portal auspiciadores	76
Tabla 3.45. Tarjeta C.R.C. Portal contactos	76
Tabla 3.46. Tarjeta C.R.C. Aplicativo móvil.....	76
Tabla 3.47. Planeación cuarta iteración primera parte	90
Tabla 3.48. Planeación cuarta iteración segunda parte.....	91
Tabla 3.49. Verificar la implementación del servidor streaming	93
Tabla 3.50. Verificación del sistema web con diseño adaptativo.....	94
Tabla 3.51. Verificación del portal de la productora.....	94
Tabla 3.52. Verificación del portal de información.....	95
Tabla 3.53. Verificación del portal de auspiciadores	95
Tabla 3.54. Verificación del portal de contactos	96
Tabla 3.55. Verificar la implementación del aplicativo móvil.....	96
Tabla 3.56. Dispositivos móviles utilizados para la prueba de funcionamiento	99
Tabla 4.1. Factor de ponderación para la funcionalidad.....	102
Tabla 4.2. Valores de los puntos función	102
Tabla 4.3. factor de ajuste de valor.....	103
Tabla 4.4. Preguntas facilidad de uso	106
Tabla 4.5. Escala de ponderación para usabilidad.....	106
Tabla 4.6 . Resultados de la encuesta de usabilidad.....	106
Tabla 4.7. Evaluación de mantenibilidad	107
Tabla 4.8. Cuestionario de factor de eficiencia	109
Tabla 4.9. Calidad global.....	109
Tabla 5.1. Coeficientes: a, b, c y d COCOMO II	114
Tabla 5.2. Conversión de puntos función a KLDC	114
Tabla 5.3. Costo de elaboración del proyecto	117
Tabla 5.4. Costo total del software	117
Tabla 5.5. Calculo del VAN	118
Tabla 5.6. Criterio de interpretación del VAN.....	119
Tabla 5.7. Calculo de la tasa interna de retorno	120

CAPÍTULO 1

MARCO INTRODUCTORIO

1.1 INTRODUCCIÓN

Con el paso del tiempo la comunicación auditiva ha ido evolucionando hacia nuevas tecnologías, enfocándonos en los comunicadores radiales, la oportunidad de expansión para va aumentando día a día. Y más aún con el internet que habrá un futuro prometedor.

Las emisoras de radio en frecuencia modulada (FM) y amplitud modulada (AM) ya no tienen mucha cabida para licitar frecuencias y el costo de adquirir una frecuencia es bastante elevado.

Noruega será el primer país del mundo en eliminar la radio FM. [Lacey-Bordeaux, 2015]. Por este motivo los comunicadores y productores radiales optan por otros caminos para hacer comunicación como ser: la radio digital y la radio por internet.

La informática ha desarrollado la tecnología streaming como medio para transmitir audio y video en tiempo real. “La palabra streaming se refiere a una corriente continuada, que fluye sin interrupción”. La tecnología describe un búfer de datos que va almacenando lo que se va descargando en la terminal del usuario para mostrar el material descargado. Esto se contrapone al desarrollo habitual de descargas de archivos, que requiere que el usuario descargue por completo los archivos para poder así acceder a su contenido. El término streaming se aplica a la difusión de audio y video.

En el mundo de los comunicadores y productores radiales, la difusión es cada vez más competitiva, y se buscan nuevas formas de llegar al oyente, no solo de manera local sino proyectarse a nivel mundial. Con el fin de cubrir las exigencias el internet se presenta como un medio que satisfaga esta necesidad a través de la tecnología streaming centrándonos genuinamente en el audio streaming. Recordando que streaming hace referencia a video y audio.

La difusión de audio por internet a través de streaming es uno de los servicios más atractivos actualmente para los comunicadores radiales, abriendo un espacio en la red con mayor y más fácil cobertura a los oyentes en tiempo real. Facilitando la implementación, evitando la licitación del espectro radial y disminuyendo costos en equipos comunicacionales.

Por ser un canal interactivo, altamente flexible y autónomo, contribuye a la descentralización del conocimiento. Permite la posibilidad al comunicador o productor radial utilizar su creatividad y generando su propio contenido.

La empresa “LOS SUPER CLASICOS” encabezado por su gerente Henry Larrea Alurralde personaje de los medios de comunicación, quien a su vez es productor de eventos, conciertos, conductor de programas radiales y televisivos. Actualmente desempeña sus funciones en Radio Monumental 105.1 FM departamento de La Paz - Bolivia con 2 programas radiales el primero: La revista “Cuestión de actitud” de lunes a viernes de 09:00am a 12:30am y los fines de semana con “Los Súper Clásicos” sábados de 11:00am a 14:00pm.

Con el propósito de optimizar el trabajo que realiza la empresa y contribuir al desarrollo tecnológico, se optó por implementar un sistema de gestión de audio streaming para mejorar las condiciones de la organización. El mismo estará integrado por un servidor de audio streaming, sistema web y un aplicativo móvil.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INSTITUCIONALES

1.2.1.1 ORIGEN DE LOS SUPER CLASICOS

El programa Los Súper Clásicos, nace en una charla de amigos un fin de semana en Ciudad Satélite, Roberto Ruiz L. y Henry Larrea Alurralde escuchando música en discos y casets, uno tras otro empezó hablar de los artistas, orígenes, componentes, y se propusieron juntar su material y presentar clásicos de los 70's y 80's para la gente que añoraba esos recuerdos.

El programa se inicia en radio Infinita 102.1 F.M. un 10 de Septiembre de 1990 que se emitía todos los sábados de 9:00 a 13:00. Posteriormente Roberto Ruiz, toma otro rumbo y en 1991 Henry Larrea Alurralde se queda a cargo de la conducción y producción del programa.

1.2.1.2 LOS SUPER CLASICOS PRODUCTORA

Después de varios años al aire en radio y después de armar la propia productora “MAS COMUNICACIÓN” Henry Larrea Alurralde decide llevar a la pantalla chica el programa “Los Súper Clásicos”, ya que la productora cuenta con una buena cantidad de material en videos DVD para presentar a la audiencia, saliendo al aire por la Red P.A.T. un 1 de Mayo del año 2005. Posteriormente el año 2006 pasando a canal 7 Televisión Boliviana manteniendo este programa por tres años al aire. Actualmente se difunde por canal 4 RTP para todo el país.

1.2.1.3 LOS SUPER CLASICOS EVENTOS

Con la experiencia ganada en medios de comunicación, se decidió ingresar a la organización de eventos, como fiestas de clásicos en diferentes departamentos de Bolivia, destacando en Sucre, Potosí, Tarija, Cochabamba, y sobre todo en La Paz, con LA FIESTA DE LOS SUPER CLASICOS. Posteriormente el año 2010 se ingresó a organizar eventos más grandes como la llegada de grupos musicales como SCORPIONS a La Paz - Bolivia el año 2010 en septiembre y la llegada de TWISTED SISTER.

1.2.1.4 SECTORES RADIO Y TELEVISIÓN

Los sectores tienen la aceptación en el público, mencionando los sectores que reciben más comentarios son: LA VERSION EN VIVO, donde se presentan artistas en diferentes escenarios a nivel Internacional, este sector conocemos a los verdaderos músicos, ya que en vivo los artistas desarrollan su verdadera capacidad.

Otro de los sectores que gusta mucho a la gente es MUSICALIZANDO LA HISTORIA, donde se revisa la historia de nuestro país, año tras año y también revisamos la historia de

la música en el mundo, con imágenes que logran captar la atención de la gente, por ejemplo: En 1971 se creó el Partido Socialista bajo la conducción de “Marcelo Quiroga Santa Cruz” y el Movimiento de la Izquierda Revolucionaria que jugaría un importante papel contra la dictadura de “Hugo Banzer Suarez”, y dentro la historia de la música, en 1971 Se juntan a Deep Purple el vocalista Ian Gillan y los extraordinarios solos de guitarra de Ritchie Blackmore, se implementan las canciones largas, muchas de las cuales duraban más del doble, que un tema estándar.

1.2.1.5 FORTALEZA DE LA EMPRESA LOS SUPER CLASICOS

Henry Larrea Alurralde “No se puede hablar de los demás programas”, en los medios sabemos que la gente es la que decide, pero te puedo asegurar que no me considero un pinchador de discos y de videos, al programa trato de darle contenido, historia, anécdotas, que la gente no solamente vea su canción favorita, sino también conozca algo más de su historia, su vida, que relacione las imágenes o algún detalle en el comentario con su vivencia personal.

Es un trabajo de toda la semana para un programa de una hora, creo que en estos tiempos de piratería, la música ya no se vende por sí sola, tienes que ponerle un condimento un plus adicional y eso es lo que hace diferente a LOS SUPER CLASICOS.

1.2.1.6 MATERIAL DE LOS SUPER CLASICOS

Conseguir el material ahora ya no es tan difícil como antes, el trabajo “duro” está en revisar el material, y la gente dirá: ¿ver videos es un trabajo duro?, y al final se llega a la conclusión que amo la música y me pagan por hacer lo que más quiero en la vida, a la hora de hablar del trabajo no todos pueden decir lo mismo, expresa Henry Larrea Alurralde.

Pero son horas y horas de revisar nuestro material y revisar material pirata, además presento especiales muy bien editados con lo mejor de cada artista, este trabajo fue muy bien reconocido por nuestra gente cuando pusimos en escena el especial de Michael Jackson, Journey, Bon Jovi, U2. Guns & Roses, entre otros.

1.2.1.7 NOVEDADES PARA MÁS ADELANTE

Henry Larrea Alurralde, estamos apostando a la participación de la gente, estamos preparando el proyecto del Knock Out en televisión, música dentro de un cuadrilátero imaginario, donde los contendientes son los grupos o solistas y los golpes la música, ¿suena bien no?, y los televidentes son lo más importante, porque recibiremos sugerencias en vivo, para cada uno de los géneros que se enfrenten. También estamos trabajando arduamente para terminar nuestro libro, esta es una primicia, son más de 20 años de guiones y tenemos mucho que compartir con los amantes de la música, será un libro muy interesante

Para cerrar quiero agradecer a la gente por su cariño, la promesa seguir presentando la música que resistió el paso del tiempo, a mi familia por el apoyo, ahora trabajo en familia, mis hijos trabajan en cámaras y edición.

1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES

La Carrera de Informática de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés cuenta con una tesis grado, que contempla parte mínima del tema presentado. Sin embargo la carrera cuenta con el siguiente trabajo realizado:

Título: Modelo para el Diseño de Servicios de Video – Streaming Adaptativo en Redes AD HOC

Autor: Univ. Juan Luis Choque Yujra

Año: 2013

La tesis hace referencia a establecer un modelo de servicios de video streaming capaz de adaptarse de manera automática a las condiciones variables de las redes MANET (Mobile Ad hoc Network). El modelo desarrollado transmite una secuencia de video, utilizando los protocolos RTP (Real Time Protocol) y RTCP (Real Time Control Protocol) sobre UDP como protocolos de transporte para la transmisión de flujos de video sobre redes móviles Ad-Hoc (MANET).

Se muestran algunos trabajos existentes en el exterior del país aplicados al campo. Que en comparación de los locales son mayoritarios.

- Título: Estudio del streaming de audio y video sobre redes heterogéneas

Autor: María Del Carmen Gómez Cruz

Año: 2010

Proporciona una panorámica actual de las herramientas de codificación de audio y video, de las redes de transporte de banda ancha existentes y su tendencia para soportar las cada vez mayores exigencias del mercado en servicios avanzados como: Voz sobre IP, datos de alta capacidad y distribución de televisión de alta definición.

- Título: Implementación de plataforma para streaming de video en tiempo real a partir de tecnologías libres

Autor: Axel Omar Meza Arrecis

Año: 2012

Hace referencia a desarrollar una plataforma para la distribución de audio y video de alta calidad a través de internet, donde cualquier persona puede compartir un evento especial o simple, desde su cámara Web enfocando ese momento y enviarlo en cuestión de décimas de segundo a una o varias personas. Sin importar donde se encuentren, contando con una computadora con conexión de internet y un navegador que soporte contenido flash.

- Título: Plataforma de video streaming para la red social comunitaria de CENDITEL

Autor: Víctor F. Terán H

Año: 2010

Institución: CENDITEL (Centro Nacional de Desarrollo e Investigación de Tecnologías Libres – Estado Venezolano)

Realiza el desarrollo de un software que permita prestar a las comunidades un servicio local de streaming incentivando la participación ciudadana. A través de una plataforma que maneje audio y video en formatos libres dentro de una red comunitaria conocida como CENDITEL y extendiéndose a la sociedad. Por otro lado se fomenta al uso de la tecnología con desarrollo de software libre.

1.3 PLANTEAMIENTO DEL PROBLEMA

Durante los últimos años la tecnología y la comunicación ha ido evolucionando en varios aspectos, siendo cada vez más común que las personas tienen acceso a estas tecnologías.

Sumando a estos aspectos es imprescindible tener un ordenador en casa y más aún el auge de los dispositivos móviles. La evolución ha modificado profundamente las relaciones entre las personas dirigiéndolas hacia una sociedad interconectada donde el eje principal es la comunicación.

Sin embargo, no es suficiente comunicarse de manera tradicional, sino incursionar en las nuevas tecnologías, proyectándose al exterior y no solamente centrándose en el interior del país.

1.3.1. PROBLEMA CENTRAL

De acuerdo a lo descrito, dentro de la empresa se plantea el problema central:

¿De qué manera la productora radial “Los Súper Clásicos” puede transmitir audio aplicando nuevas tecnologías de streaming y orientándolas a diseños adaptativos, mejorando la forma tradicional de hacer comunicación?

1.3.2. PROBLEMAS SECUNDARIOS

Al mismo tiempo se encontraron los siguientes problemas secundarios con sus respectivas causas y efectos:

- La difusión de la productora radial se realiza de manera local, perdiendo clientes en el exterior y en el resto del país.
- El programa radial se promociona en internet a través una red social, disminuyendo la llegada a los oyentes por medio de un portal web dinámico y adaptativo.
- La empresa “Los Súper Clásicos” promueve actualmente sus actividades y eventos por medio del programa radial, perdiendo difusión en los clientes en la red.
- La empresa promueve sus auspiciadores y publicidades por medio del programa radial de manera local, descartando su llegada al resto del país y el mundo por la red de internet.
- Si bien el programa radial y “Los Súper Clásicos” se escucha en teléfonos inteligentes que cuentan con un radio FM, se descarta a los nuevos dispositivos móviles que carecen de un sintonizador..

1.4 DEFINICIÓN DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e implementar un sistema web de gestión de audio streaming para la empresa “Los Súper Clásicos” orientada a diseños adaptativos, transmitiendo a través de un servidor de audio streaming por la red, recepcionando el contenido de forma paralela en la Web y en los dispositivos móviles.

1.4.2 OBJETIVOS ESPECÍFICOS

- Implementar un servicio a un servidor de audio streaming contemplando requerimientos físicos y lógicos de la empresa.
- Desarrollar un sistema web dinámico e interactivo que permita gestionar los contenidos de la productora.
- Contemplar un diseño adaptativo, visualizando el sistema web de forma elegante en los diferentes dispositivos.

- Desarrollar un aplicativo móvil que complemente al sistema web de gestión de audio streaming.
- Verificar el desempeño del sistema con la adecuada información en la web y el contenido de audio streaming en el aplicativo móvil.

1.5 JUSTIFICACIÓN

1.5.1 JUSTIFICACIÓN ECONÓMICA

Hace algunos años existía la dificultad de poder trabajar con la tecnología streaming debido a la escasez del ancho de banda en la red y equipos con considerable precio en algunos casos con recursos técnicos limitados. Pero ahora con la disminución de costos en cuanto se refiere a: ordenadores personales, laptop y el auge de los dispositivos móviles. Los mismos cuentan con mayor capacidad de: almacenamiento, procesamiento, pantallas de alta resolución, comunicación vía Bluetooth. Reducción en la conexión de internet para ordenadores y teléfonos móviles destacándose servicios de 3G, 3.5G y ahora con la llegada tímida del 4G.

Para la empresa “Los Súper Clásicos” el desarrollo e implementación del sistema web de gestión con tecnología streaming generara mayores beneficios económicos realzando el nombre de la organización.

Con una marca establecida en el interior y en el exterior país, los comunicadores y productores radiales tienen como mayor ingreso económico por medio de la publicidad o también conocidos como auspiciadores, sin dejar de lado los contratos que tiene la productora tanto a nivel radial como televisivo y la organización de eventos musicales como los grandes referentes de la productora.

1.5.2 JUSTIFICACIÓN SOCIAL

Con el desarrollo del proyecto se beneficiara principalmente la sociedad seguidora del contenido auditivo ofreciéndole un material minuciosamente seleccionado, el mismo se podrá oír en tiempo real con la tecnología streaming y en cualquier lugar.

Estará al alcance del hogar a través de un ordenador, en la mano a través de un dispositivo móvil, por medio de un portal web diferente con un diseño adaptativo.

La transmisión del contenido será sin interrupciones ni cortes para los oyentes evitando caídas de señal e interferencia de emisoras clandestinas.

1.5.3 JUSTIFICACIÓN TECNOLÓGICA

Con el avance del tiempo con la tecnología es posible desarrollar e implementar sistemas que permiten la adquisición de contenido multimedia a través de internet para la web y móviles como es el servicio streaming. De manera general esta tecnología permite aligerar la descarga y ejecución de contenido multimedia con la facilidad de reproducir sin tener que guardar la información en la memoria.

El proyecto se destaca de manera relevante por la necesidad de orientar a los esquemas actuales del servicio streaming hacia sitios web y dispositivos móviles. Realizando la evaluación y selección del servidor de streaming observando los criterios que cobran mayor importancia en cuanto a la distribución de contenido de audio, para luego desarrollar un sistema que preste el servicio streaming como interfaz de conexión entre los emisores y receptores.

1.6 ALCANCES Y LÍMITES

El sistema de gestión a realizarse dentro la empresa “Los Súper Clásicos” ayudara directamente a la parte gerencial de la misma. De ahí se puede deducir los límites y alcances.

1.6.1 ALCANCES

El presente trabajo se centrara exclusivamente en la implementación de un servicio a un servidor de streaming de audio como emisor que contemple la comunicación a través de la red y recepcionar el contenido de manera conjunta tanto para el sistema web como los dispositivos móviles.

El sistema desarrollado e implementado en la empresa “Los Súper Clásicos” contará con los siguientes módulos:

- Manejo y control del material de audio streaming.- El modulo será operado por el comunicador Henry Larrea Alurralde, quien programara el contenido auditivo para llegar a la audiencia de mejor manera con un repertorio diferente.
- Subir y descargar material de la productora.- El modulo será operado por Henry Larrea Alurralde quien subirá contenido exclusivo al sistema, los oyentes podrán descargar los archivos del sistema web.
- Actualización e información del sitio Web.- El módulo describe el constante cambio de contenido de la página ofreciendo novedades concernientes a la empresa e interactuando con la gente. Proponiendo material valioso en cuanto se refiere al oyente. Sera controlado y seleccionado por el gerente general de la empresa.

1.6.2 LÍMITES

El sistema tiene por limitaciones:

- Cabe resaltar que el sistema de gestión solo se enfoca al material de audio streaming, dejando para otro tema de investigación al video streaming. Haciendo conocer que la palabra streaming tiene un amplio campo de desarrollo.
- El proyecto estará orientado de manera exclusiva a los sitios Web como las aplicaciones móviles esto en cuanto se refiere al material y al contenido de audio streaming.

1.7 APORTES

1.7.1 PRÁCTICO

El sistema de manera práctica aportara conocimiento en:

- La relación e interacción del sistema de gestión de material de audio entre servidor de streaming, el sistema web y la aplicación móvil.

- Características de como reproducir audio streaming en aplicativos webs y móviles.
- Protocolos de comunicación en la transmisión de audio streaming.
- Arquitectura de un servidor de streaming de audio.
- La implementación de un servicio a un servidor de audio streaming tomando en cuenta los requerimientos tanto físico como lógico de la empresa y eligiendo las mejores opciones para su mejor funcionamiento.

1.7.2 TEÓRICO

El proyecto en la parte teórica aportara conocimiento en:

- Uso de la metodología de desarrollo ágil Programación Extrema - XP la cual es sencilla de entender y brinda un enfoque de desarrollo de software.
- Uso de la metodología ágil MOBILE-D para el desarrollo de software de la aplicación móvil.
- Para el diseño del sitio web se hará uso de WebML (Web Modeling Lenguaje) como metodología para una notación visual.

1.8 METODOLOGÍA

Para el desarrollo de la presente investigación se procederá con el método científico que conduce a una serie de pasos para descubrir las propiedades del objeto de estudio y al final de la investigación llegue a la búsqueda constante de resultados óptimos basados en un desarrollo técnico.

La investigación de desarrollo que corresponde al presente proyecto, está en su fase de inicio es de tipo exploratoria y en su fase final de tipo descriptiva.

- Investigación exploratoria.- Es el diseño de la investigación que tiene por objetivo principal profundizar y comprender el problema.
- Investigación descriptiva.- Es el tipo de investigación concluyente que tiene como objetivo principal: la descripción, el saber el por qué y para que se está realizando.

1.8.1 METODOLOGÍA DE DESARROLLO

Las metodologías de desarrollo software serán: Programación Extrema - XP y Mobile-D.

XP es un proceso que se aplica en equipos con muy pocos programadores quienes llevan reducidos procesos en paralelo. Consiste en planificar, diseñar, desarrollar y hacer pruebas lo más rápido posible, hasta en casos se recomienda saltar la documentación y los procedimientos tradicionales. Se fundamenta en la capacidad del equipo para comunicarse entre sí y las ganas de aprender de los errores propios inherentes en un programador. La gran ventaja de XP es su increíble capacidad de respuesta ante imprevistos, ideal para equipos extremadamente pequeños que se centran en un solo cliente.

Mobile-D es una metodología cuyo enfoque y características la hacen especialmente apta para el mercado de dispositivos móviles, donde los requerimientos cambian constantemente y el software se requiere en el momento justo. Al centrarse en grupos de trabajo pequeño y debido a su rápida velocidad de desarrollo.

1.8.2 TÉCNICA

Para el desarrollo del presente proyecto se usaran las siguientes plataformas:

- Sistema Operativo: Windows 8
- Sistema Operativo: Google Android

1.8.3 HERRAMIENTAS

El desarrollo del trabajo utilizara dos lenguajes de programación:

1ro.- Un lenguaje de desarrollo orientado a las aplicaciones Web Dinámicas, como es PHP y de fácil acceso como software libre.

2do.- Java es un lenguaje de programación orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana, la misma será usada para desarrollar la aplicación móvil.

CAPÍTULO 2

MARCO TEÓRICO

2.1 INTRODUCCIÓN

El presente capítulo tiene como finalidad aclarar, describir conceptos y definiciones que serán utilizados en el desarrollo de los diferentes capítulos del proyecto.

Los mismos ayudarán a quienes lean este proyecto a aclarar muchos de los conceptos que se necesitan y se requieren para leer este proyecto.

2.2 INGENIERÍA DEL SOFTWARE

IEEE (Corresponde a las siglas de Institute of Electrical and Electronics Engineers, es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías) define a la ingeniería del Software como: “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software” [Hans Van Vliet, 2002]. La ingeniería de software es una disciplina que concierne a todos los aspectos de la producción del software.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería del software. Los objetivos de la ingeniería del software son:

- Mejorar la calidad de los productos de software.
- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

La ingeniería de software se relaciona con varias disciplinas como: gerencia, matemáticas, gestión de proyectos, gestión de calidad, ingeniería de sistemas y ergonomía del software. Abarca un conjunto de áreas del conocimiento que son la base fundamental para el desarrollo de un proyecto software, estas áreas son: requerimientos, diseño, construcción, pruebas, mantenimiento, gestión de la configuración, gestión de la ingeniería, procesos de ingeniería, herramientas y métodos de la ingeniería, calidad del software. A continuación se detallan las más importantes: [Hans Van Vliet, 2002].

- a) **Requerimientos de software.**- Se define como una exigencia de debe ser cumplida para dar solución a un problema del mundo real. Contiene áreas como: especificación de requerimientos, análisis, validación, clasificación, negociación, entre otros.
- b) **Diseño del software.**- Es el proceso de definir la arquitectura, componentes. Interfaces y otras características relativas al sistema como tal. Fundamentos, claves en el diseño. Estructura y calidad son algunas de las áreas secundarias que comprende el diseño de software.
- c) **Construcción del software.**- Es la creación detallada de un sistema software a través de la combinación de codificación, verificación, pruebas de unidad, pruebas de integración y depuración.
- d) **Pruebas del software.**- Consiste en la verificación dinámica del comportamiento de un software ante un conjunto limitado de casos de prueba.
- e) **Mantenimiento del software.**- Las actividades de mantenimiento comienzan teóricamente cuando el producto final es liberado, pero en la práctica empieza desde etapas mucho más tempranas, debido a los cambios en la necesidad del usuario a la que la aplicación debe adaptarse.

2.2.1 INGENIERÍA

La ingeniería es el estudio y la aplicación de las distintas ramas de la tecnología. El profesional en este ámbito recibe el nombre de ingeniero.

La ingeniería también supone la aplicación de la inventiva y del ingenio para desarrollar una cierta actividad. Esto, por supuesto, no implica que no se utilice el método científico para llevar a cabo los planes. [Bauer, 1968].

2.2.2 SOFTWARE

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computacional. [IEEE, 1993].

El software se puede definir como el conjunto de tres componentes:

- Programas.- (Instrucciones), este componente proporciona la funcionalidad deseada y el rendimiento cuando se ejecute.
- Datos.- Este componente incluye los datos necesarios para manejar, aprobar los programas y las estructuras requeridas para mantener y manipular estos datos.
- Documentación.- Este componente describe la operación y el uso del programa.



Figura 2.1. Componentes del software
Fuente: INTECO, 2009

2.2.3 CARACTERISTICAS DEL SOFTWARE

El software es esencialmente un conjunto de instrucciones (programas) que proporcionan la funcionalidad requerida, los datos relacionados y documentos. Por lo tanto, el software es un elemento lógico y se diferencia hardware, un elemento físico, en sus características.

El software no se estropea, pero se deteriora. Durante su vida, el software sufre cambios (mantenimiento). Conforme se hacen los cambios, es bastante probable que se introduzcan nuevos defectos, lo que hace que el software se vaya deteriorando debido a los cambios.

El hardware usa componentes estándar con funciones e interfaces bien definidas. El uso de estos componentes ayuda a evitar reinventar la rueda. La fase de diseño en el ciclo de un producto hardware implica seleccionar los componentes disponibles más adecuados y decidir el enfoque para montarlos. Los componentes de hardware estándar son útiles porque conducen a:

- Reducir el coste y el tiempo de lanzamiento al mercado.
- Buena calidad
- Ingeniería rápida
- Fácil mantenimiento
- Fácil mejora

2.2.4 CICLOS DE VIDA DE UN SOFTWARE

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea la inicia hasta que el software es retirado o remplazado.

Entre las funciones que debe tener un ciclo de vida se puede destacar.

- Determinar el orden de las fases del proceso de software.
- Establecer los criterios de transmisión para pasar de una fase a la siguiente.
- Definir las entradas y salidas de cada fase.
- Describir los estados por los que pasa el producto.
- Describir las actividades a realizar para transformar el producto.

2.2.4.1 ETAPAS

La ingeniería de software requiere llevar a cabo numerosas tareas, dentro de etapas como las siguientes:

a) Análisis de requisitos

Extraer los requisitos de un producto software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere

habilidad y experiencia en la ingeniería del software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento de especificación de requisitos.

b) Especificación

Es la tarea de describir detalladamente el software a ser desarrollado, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas.

c) Diseño y arquitecturas

Se refiere a determinar cómo funcionara el software de forma general sin entrar en detalles. Se definen los casos de uso para cubrir las funciones que realizara el sistema, y se transformaran las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercado a la programación orientada a objetos.

d) Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería del software, pero no necesariamente es la que demanda mayor trabajo ni la más complicada. La complejidad y la duración de esta etapa se relaciona íntimamente al o los lenguajes de programación utilizados, así como al diseño previamente realizado.

La programación es el proceso de diseñar, escribir, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación.

e) Prueba

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica que las pruebas sean efectuadas por alguien distinto al desarrollador que la programo.

f) Mantenimiento

Mantener y mejorar el software para solventar errores descubiertos y tratar con nuevos requisitos. El mantenimiento puede ser de cuatro tipos: perfectivo (mejorar la calidad interna de los sistemas), evolutivo (incorporaciones, modificaciones y eliminaciones necesarias en un producto para cubrir la expansión o cambio en las necesidades del usuario), adaptativo (modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo: cambios de configuración del hardware, software de base, gestores de base de datos, comunicaciones) y correctivo (corrección de errores).

2.3 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

El desarrollo de software no es una tarea fácil, prueba de ello de que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas, notaciones que se usaran. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos. Una posible mejora es incluir en los procesos de desarrollo más actividades, artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

2.3.1 VENTAJAS DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Son muchas las ventajas que puede aportar el uso de una metodología. A continuación se van exponer algunas de ellas, clasificados desde distintos puntos de vista.

Desde el punto de vista de gestión:

- Facilitar la tarea de planificación.
- Facilitar la tarea del control y seguimiento de un proyecto.
- Mejorar la relación costo/beneficio.
- Facilitar la comunicación efectiva entre usuarios y desarrolladores.

Desde el punto de vista de los ingenieros del software:

- Ayudar a la comprensión del problema.
- Optimizar el conjunto y cada una de las fases del proceso de desarrollo.
- Facilitar el mantenimiento del producto final.
- Permitir la reutilización del producto final.

Desde el punto de vista del cliente o usuario:

- Garantía de un determinado nivel de calidad en el producto final.
- Confianza en los plazos de tiempo fijados en la definición del proyecto.
- Definir el ciclo de vida que más se adecue a las condiciones y características del desarrollo.

2.4 METODOLOGÍAS DE DESARROLLO TRADICIONALES

Las metodologías tradicionales son denominadas, a veces, de forma peyorativa, como metodologías pesadas.

Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto definido todo esto en la fase inicial del desarrollo del proyecto.

La característica más importante dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. Las metodologías tradicionales se focalizan en la documentación en la documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, entre otros.)

2.5 METODOLOGÍAS DE DESARROLLO ÁGIL

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan.

2.6 DIFERENCIAS ENTRE METODOLOGÍAS TRADICIONALES Y ÁGILES

La tabla 2.1 muestra las principales diferencias entre las metodologías tradicionales y ágiles

METODOLOGÍAS TRADICIONALES	METODOLOGÍAS ÁGILES
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurística provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Expuestas externamente.	Impuestas internamente por el equipo.
Procesos mucho más controlados con numerosas políticas y normas.	Procesos menos controlados con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños menos de 10 integrantes y trabajando en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

Tabla 2.1. Metodologías tradicionales vs ágiles
Fuente: INTECO, 2009

2.7 EL POR QUE USAR METODOLOGÍAS ÁGILES

Tomando las ideas de la tabla 2.1 se puede concluir que las metodologías tradicionales presentan deficiencias a la hora de abordar proyectos:

- Costosas fases previas de especificación de requisitos, análisis y diseño. La corrección durante el desarrollo de errores introducidos en estas fases será costosa, perdiendo flexibilidad antes los cambios.
- El proceso de desarrollo esta encorsetado por documentos firmados.
- El desarrollo es más lento, es difícil para los desarrolladores entender un sistema complejo en su globalidad.

2.8 METODOLOGÍA DE PROGRAMACIÓN EXTREMA (XP)

La programación extrema (XP) es un enfoque de la ingeniería del software formulado por Kent Beck. Es el más conocido de los procesos ágiles de desarrollo de software. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.

2.8.1 ELEMENTOS DE LA METODOLOGÍA

2.8.1.1 HISTORIAS DE USUARIOS

Es la técnica utilizada para especificar los requisitos del software, se trata de tarjetas de papel en la cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos pueden ser funcionales o no funcionales.

Las historias de usuario se descomponen en tarjetas tareas que se asignan a los desarrolladores para ser implementadas durante una o varas iteraciones.

2.8.1.2 ROLES XP

Los roles de acuerdo con la propuesta de Beck son:

- Programador: Escribe las pruebas unitarias y produce el código del sistema.

- Cliente: Escribe las historias de usuario y las pruebas de aceptación para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuales se implementan en cada iteración centrándose en apoyar mayor valor al negocio.
- Encargado de pruebas: Ayuda al cliente a escribir las pruebas funcionales, ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para las pruebas.
- Encargado de seguimiento: Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- Entrenador: Este rol es el responsable del proceso global, provee guías al equipo de forma que se apliquen las practicas XP y se siga el proceso correctamente.
- Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- Gestor: Es el que coordina entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas.

2.8.1.3 PROCESO XP

El ciclo de desarrollo consiste de manera general, en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El programador construye ese valor.
4. Vuelve al paso 1.
5. En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden.

No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse de que el sistema tenga el mayor valor de negocio posible.

2.8.2 CICLO DE VIDA DE XP

En la figura 2.2 muestra el desarrollo del ciclo de vida ideal de XP consiste en: exploración, planificación de la entrega, iteraciones, producción, mantenimiento y muerte del proyecto.

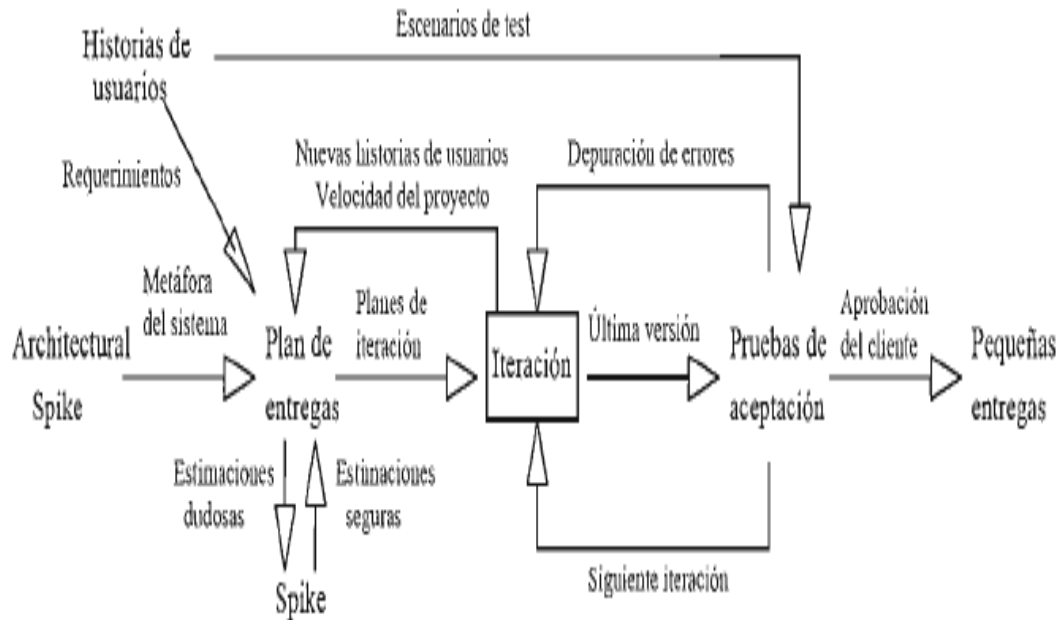


Figura 2.2. Ciclo de vida de XP
Fuente: ONESS, 2005

2.8.2.1 EXPLORACIÓN

En esta fase, los clientes plantean de manera general las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiarizara con las herramientas, tecnologías y prácticas que utilizaran en el proyecto.

2.8.2.2 PLANIFICACIÓN DE LA ENTREGA

En esta fase el cliente establece la prioridad de cada historia de usuario, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses, durando pocos días.

2.8.2.3 ITERACIONES

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide que historias se implementaran en cada iteración para maximizar el valor de negocio.

2.8.2.4 PRODUCCIÓN

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

2.8.2.5 MANTENIMIENTO

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que se desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

2.8.2.6 MUERTE DEL PROYECTO

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura.

2.8.3 PRACTICAS XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del coste del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes practicas:

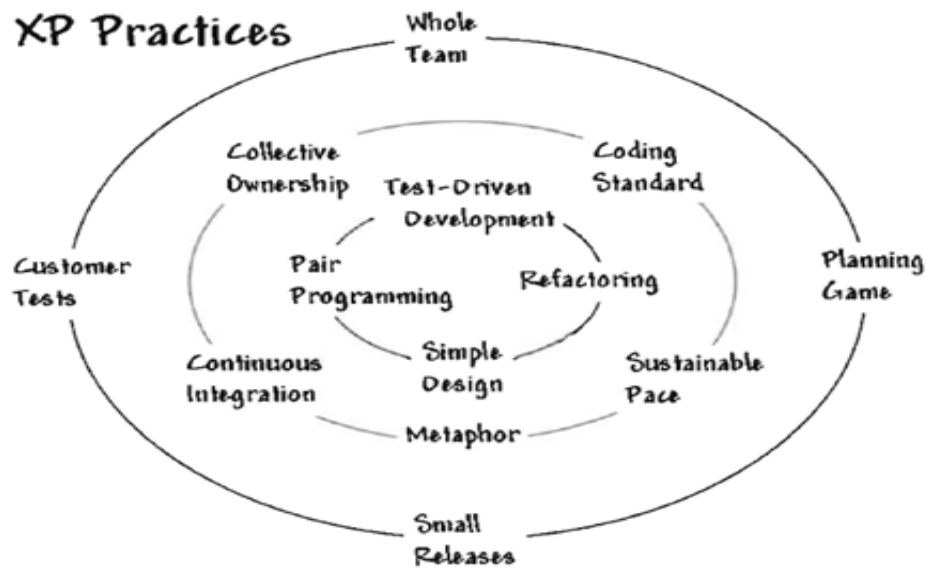


Figura 2.3. Practicas XP
Fuente: ONESS, 2005

- El juego de la planificación.- Hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- Entregas pequeñas.- Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de 3 meses.
- Metáfora.- El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe como debería funcionar el sistema (conjunto de nombre que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- Diseño simple.- Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

- Pruebas.- La producción de código está dirigida por las pruebas unitarias. Estas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- Refactorización.- Es una actividad constante de reestructuración del código con el objetivo de evitar duplicación del código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- Programación en parejas.- Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menos tasa de errores, mejor diseño y mayor satisfacción de los programadores).
- Propiedad colectiva del código.- Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- Integración continua.- Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- 40 horas por semana.- Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva el trabajo.
- Cliente in-situ.- El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Este es uno de los principales factores de éxito del proyecto XP.
El cliente conduce constantemente el trabajo hacia lo que aportara mayor valor al negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- Estándares de programación.- La programación extrema enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener código legible.

2.8.4 PRINCIPIOS DE XP

Los principios básicos de la programación extrema son: simplicidad, comunicación, retroalimentación y coraje.

- **Simplicidad.**- La simplicidad es la base de la programación extrema, se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hacen que la complejidad aumente exponencialmente.

Para mantener la simplicidad es necesaria la refactorización del código, esta es la manera de mantener el código simple a medida que crece.

- **Comunicación.**- La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor mientras más simple sea. Si el código es complejo hay que esforzarse para hacerlo legible.

El código autodocumentado es más fiable que los comentarios ya que estos últimos pronto quedan desfasados con el código a medida que es modificado. Debe comentarse solo aquello que no va a variar, por ejemplo: el objetivo de una clase o la funcionalidad de un método.

- **Retroalimentación.**- Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real.

Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en los que es más importante.

- **Coraje o valentía.**- Para los gerentes la programación en parejas puede ser difícil de aceptar, parece como si la productividad se fuese a reducir a la mitad ya que solo la mitad de los programadores está escribiendo código.

Hay que ser valiente para confiar en que la programación por parejas beneficiara la calidad del código sin repercutir negativamente en la productividad.

2.8.5 ACTIVIDADES DE XP

2.8.5.1 CODIFICAR

Los defensores de XP argumentan que el único producto realmente importante del proceso de desarrollo de sistemas es el código (un concepto al que dan una definición más amplia que la que puedan dar otros). Sin el código no se tiene nada. La codificación puede ser dibujar diagramas que generaran código, hacer scripts de sistemas basados en web o codificar un programa que ha de ser compilado.

La codificación también puede usarse para entender la solución más apropiada. Por ejemplo, XP recomendaría que si nos enfrentamos con varias alternativas para un problema de programación, uno debiera simplemente codificar todas las soluciones y determinar con pruebas automatizadas que solución es la más adecuada. La codificación puede ayudar también a comunicar pensamientos sobre problemas de programación.

2.8.5.2 PROBAR

Nadie puede estar seguro de algo si no lo ha probado. Las pruebas no es una necesidad primaria percibida por el cliente. Mucho software se libera sin pruebas adecuadas sin embargo funcionan. En el desarrollo de software, XP dice que esto significa que uno no puede estar seguro de que una función funciona si no la prueba. Esto sugiere definir, de lo que uno puede no estar seguro:

- No puedes estar seguro de si lo que has codificado es lo que querías significar. Para probar esta incertidumbre, XP usa pruebas unitarias. Son pruebas automatizadas que prueban el código. El programador intentara escribir todas las pruebas en las que pueda pensar que puedan cargarse el código que está escribiendo; si todas las pruebas se ejecutan satisfactoriamente entonces el código está completo.
- No puedes estar seguro de si lo que querías significar era lo que deberías. Para probar esta incertidumbre, XP usa pruebas de aceptación basadas en los requisitos dados por el cliente.

2.8.5.3 ESCUCCHAR

Los programadores no saben necesariamente todo sobre el lado del negocio del sistema bajo desarrollo. La función del sistema está determinada por el lado del negocio.

Para que los programadores encuentren cual debe ser la funcionalidad del sistema, deben escuchar al negocio. Tienen que escuchar las necesidades de los clientes.

2.8.5.4 DISEÑAR

Desde el punto de vista de la simplicidad, uno podría decir que el desarrollo de sistemas no necesita más que codificar, probar y escuchar. Si estas actividades se desarrollan bien, el resultado debería ser un sistema que funcionase.

2.8.6 ARTEFACTOS XP

A continuación describimos los artefactos de XP, entre los que se encuentran: historias de usuario, tarjetas tareas, tarjetas CRC y prueba de aceptación.

2.8.6.1 HISTORIAS DE USUARIO

Representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada características principal del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, remplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

Estas deben proporcionar solo el detalle suficiente como para poder hacer razonable la estimación de cuánto tiempo requiere la implementación de la historia, difiere de los casos de uso porque son escritos por el cliente, no por los programadores, empleando terminología del cliente. “Las historias de usuario son más amigables que los casos de uso formales”

La tabla 2.2 muestra el modelo de una historia usuario contemplando sus diferentes características.

Historia de Usuario	
Número:	Nombre Historia de Usuario:
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

Tabla 2.2. Plantilla "historia de usuario"
Fuente: Echeverry, 2007

2.8.6.2 TARJETAS TAREAS

Las tarjetas de tareas de ingeniería nos permiten presentar el o los objetivos que una historia de usuario nos brindara. Puede haber una o más tareas de ingeniería por cada historia de usuario.

TAREA DE INGENIERÍA	
Número:	Historia de Usuario (Nro. Y Nombre):
Nombre Tarea:	
Tipo de Tarea : Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

Tabla 2.3. "Tarjeta de Ingeniería"
Fuente: Beck, 2000

2.8.6.3 TARJETAS CRC

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y colaboradores.

Nombre de la clase	
Responsabilidades	Colaboradores

Tabla 2.4. Modelo tarjeta CRC
Fuente: Ambler, 2002

Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

En la práctica conviene tener pequeñas tarjetas de cartón, que se llenaran y que son mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas.

Para encontrar las clases debemos pensar que cosas interactúan con el sistema (en nuestro caso el usuario), y que cosas son parte del sistema, así como las pantallas útiles a la aplicación (un despliegue de datos, una entrada de parámetros y pantalla general, entre otros).

Una vez que las clases principales han sido encontradas se procede a buscar los atributos y las responsabilidades, para esto se puede formular la pregunta ¿Qué sabe la clase? y ¿Qué hace la clase? Finalmente se buscan los colaboradores dentro de la lista de clases que se tenga.

2.8.6.4 PRUEBAS DE ACEPTACIÓN

Estas tarjetas se utilizan para ver la aceptación de la implementación de una historia de usuario por parte del cliente.

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

Tabla 2.5. Tarjeta "Casos de prueba"
Fuente: Beck, 2000

2.9 MODELADO DE DATOS

En informática, un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, un modelo de datos permite describir las estructuras de datos de base (el tipo de los datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregando, borrando, modificación y recuperación de los datos base).

En un enfoque más amplio, un modelo de datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí.

2.10 INGENIERÍA WEB

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

La ingeniería web es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas web de alta calidad. [Murugesan & Deshp & Hansen & Ginige, 2001].

El desarrollo de aplicaciones web posee determinadas características que lo hacen diferente del software tradicional y los sistemas de información. Es así que, una de las características que se debe cumplir para la ingeniería web, es que sea iterativo e incremental. En respuesta a la continua evolución de las aplicaciones web, así como el corto tiempo en el que normalmente se requiere que sean implementadas.

El proceso de ingeniería web está compuesto por las siguientes fases: [Pressman, 2004].

- **Planteamiento y formulación.-** Identificamos los objetivos de nuestra aplicación, y delimitamos el alcance de la primera iteración.
- **Planificación.-** Una vez planteado el problema, podremos estimar costos, riesgos y esfuerzo durante el desarrollo. Recordemos que en la planeación iterativa solamente se detalla la iteración actual, y las iteraciones subsecuentes sólo se plantean de forma general.
- **Análisis.-** Durante esta etapa establecemos los requerimientos técnicos, gráficos, y de contenido, que incorporaremos en la iteración.
- **Ingeniería.-** La actividad de ingeniería incorpora dos grupos de tareas que se realizan en paralelo: el diseño del contenido y la producción, se enfocan en el diseño, producción y adquisición del contenido de texto, gráfico y video que se vayan a integrar en la aplicación. Estas tareas son realizadas por personal no

técnico. Por otro lado, están el diseño arquitectónico, de navegación e interfaz, el cual lidia con los aspectos técnicos.

- **Generación de páginas y pruebas.-** Se prueba que el contenido dinámico se genere correctamente, utilizando las plantillas, interfaces y contenidos diseñados en la fase de ingeniería. Posteriormente se realizan las pruebas pertinentes, que dependerán del tipo de aplicación y requerimientos no funcionales (pruebas de desempeño, entre otros).
- **Evaluación del cliente.-** Al final de cada iteración se debe realizar una evaluación con el cliente, para validar el avance y determinar los cambios o mejoras en caso de ser necesarios, que se aplicarán en las siguientes iteraciones.

Las aplicaciones Web tienen particularidades, lo que hace que se puedan plantear modelos específicos o la forma de realizar el proceso de modelado para ser más precisos y tener más ventajas. En la actualidad se propusieron muchos tipos de los modelos entre los cuales podemos indicar a los más reconocidos: UWE, WebML y OODHM.

2.11 WEBML (WEB MODELING LANGUAGE)

La metodología de modelado WebML apunta a proveer un acercamiento en la estructura del diseño intensivo de datos que se manejan en un sitio Web.

Un conjunto de modelos integrados ayudan a los diseñadores en la producción del sitio de alta calidad ya que todas las facetas del diseño deben ser y son dirigidas.

Los principales objetivos del proceso de diseño WebML son:

- Expresar la estructura de una aplicación Web con un alto nivel de descripción que pueda ser usado para consultas, evolución y mantenimiento futuro de la aplicación.
- Provee múltiples vistas del mismo contenido.
- Descompone la información en páginas, navegación y presentación, que puedan ser definidas de manera independiente.

- Permite almacenar la meta-data recolectada durante el diseño para emplearlos durante toda la vida de la aplicación para generar páginas Web de forma dinámica.
- Sitios Web que manejen gran cantidad de datos haciendo uso de interfaces dirigidas al público en general.

El desarrollo de aplicaciones Web con WebML consiste en diferentes fases que deben ser aplicados de manera iterativa e incremental. El proceso involucra varios ciclos, cada uno de los cuales produce un prototipo o versión parcial de la aplicación, lo que permite realizar evaluaciones y pruebas desde las fases iniciales de desarrollo.

El proceso de desarrollo comienza con el Modelo de Datos del sistema en el que mediante algún lenguaje de modelado como UML (WebML no exige ninguno en concreto), se representa la estructura estática del mismo. Tras esto, se realiza el modelo hipertexto donde se describen uno o más hipertextos que pueden ser publicados en el sitio Web. Cada uno de estos hipertextos define una vista del sitio. La descripción de los hipertextos se realiza mediante dos modelos: el modelo de composición, que define las páginas que componen el sistema y el modelo de navegación que describe como se podrá navegar a través de ellas. Y por último, se describe el modelo de presentación que define la apariencia física de las páginas. [Ceri, 2000].

Para el desarrollo de sistemas Web, los modelos en la metodología de diseño WebML son:

- Modelo de datos
- Modelo de hipertexto
- Modelo de presentación

2.11.1 MODELO DE DATOS

El modelo de datos de WebML (modelo de estructura + modelo de derivación) es una adaptación conveniente de los modelos conceptuales de diseño de datos que se emplea en otras disciplinas como diseño de base de datos, ingeniería de software y representación del conocimiento.

El modelo de datos de WebML es compatible con el modelo de datos entidad-relación (ver figura 2.4) usado en el diseño conceptual de bases de datos, también es compatible con los diagramas de clases UML (ver figura 2.5) empleados en el modelado orientado a objetos.

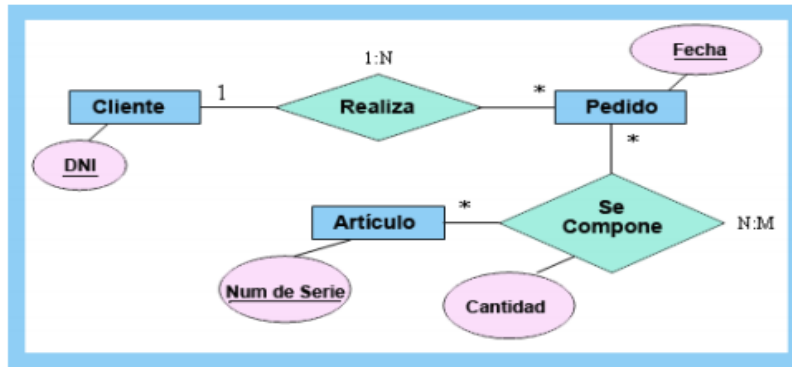


Figura 2.4. Ejemplo diagrama "Entidad - Relación"
Fuente: Korth, 2002

Para tener un mejor control del modelado del diagrama entidad-relación se obtendrá este diagrama que es una manera más visual para poder observar cómo está relacionada cada tabla de la base de datos.

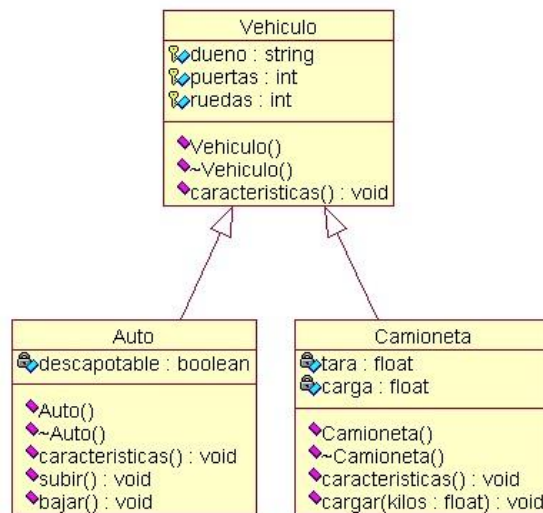


Figura 2.5. Ejemplo diagrama de clases
Fuente: Korth, 2002

El elemento fundamental del modelo de datos son las entidades, definidas como contenedores de elementos de datos y sus relaciones definidas como las conexiones semánticas entre entidades. Las entidades tienen propiedades, llamadas atributos, con un tipo asociado. Las entidades pueden ser organizadas de manera jerárquica y sus relaciones pueden restringirse por medio de la cardinalidad. [Ceri, 2000]

2.11.2 MODELO DE HIPERTEXTO

Para tener más claro el concepto del modelo de hipertexto previamente se define (ver figura 2.6) un ejemplo de notación para el esquema de estructura.

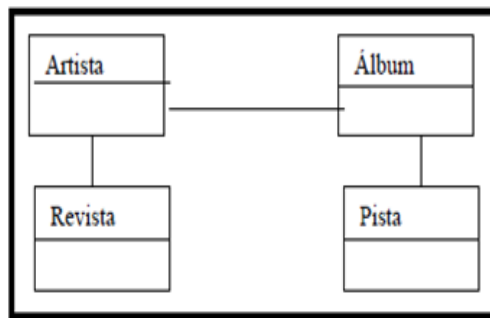


Figura 2.6. Notación para el esquema de estructura
Fuente: Ceri, 2000

El modelo de hipertexto (navegación + composición) especifica cómo se compone el sitio y la navegación (ver fig. 2.7). La composición del sitio describe las páginas que forman parte del hipertexto y las unidades de contenido que constituyen cada página. Las páginas de un sitio Web son catalogas como contenedores de información que es enviada al usuario.

Las unidades son elementos atómicos de contenido empleados para publicar información descrita en el modelo de datos. WebML contiene tipos de unidades predefinidas para desarrollar páginas Web: data, multi-data, índice (variantes jerárquicas y selección múltiple), entry y scroller. Cada unidad está asociada a una unidad subyacente, de la cual se obtiene el contenido. La especificación de una entidad subyacente determina el tipo de objeto del cual se deriva el contenido de la unidad.

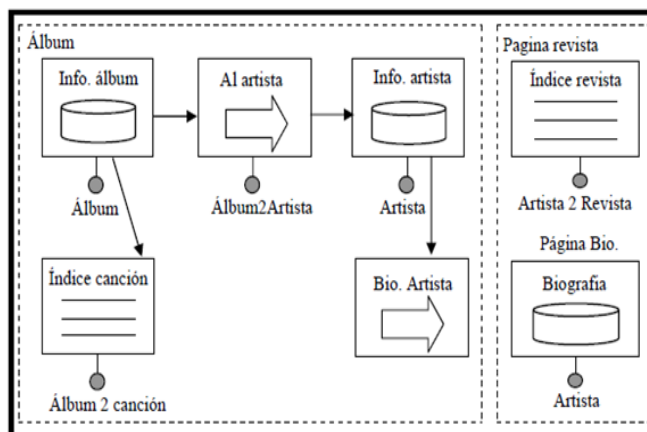


Figura 2.7. Notación para composición y navegación
 Fuente: Ceri, 2000

La navegación del sitio se realiza a través de enlaces, los cuales se definen entre unidades que se encuentran en una misma página, en diferentes páginas o entre páginas completas. La información transportada a través de los enlaces se conoce como contexto de navegación o simplemente contexto. Los enlaces que transportan información contextual se denominan enlaces contextuales mientras que los que no transportan información son conocidos como enlaces no contextuales. La información contextual generalmente es necesaria para asegurar las operaciones de computación de unidades. [Ceri, 2000].

2.11.3 MODELO DE PRESENTACIÓN

Define como lucirá la vista del sitio. WebML incluye un modelo simple de presentación que permite colocar contenidos dinámicos en la página además de aplicar estilos distintos para cada uno. Cada diagrama muestra un pantalla de cómo se verá el sistema, este diagrama la idea que se plasmara a lo largo del desarrollo.

Para poder entender de mejor manera la utilidad de los modelos dentro del desarrollo la siguiente figura (ver fig. 2.8) muestra la estructura de un sitio web modelado haciendo uso de WebML. Una observación muy importante es el hecho de que WebML no es el mejor enfoque para sitios Web estáticos [Ceri, 2000].

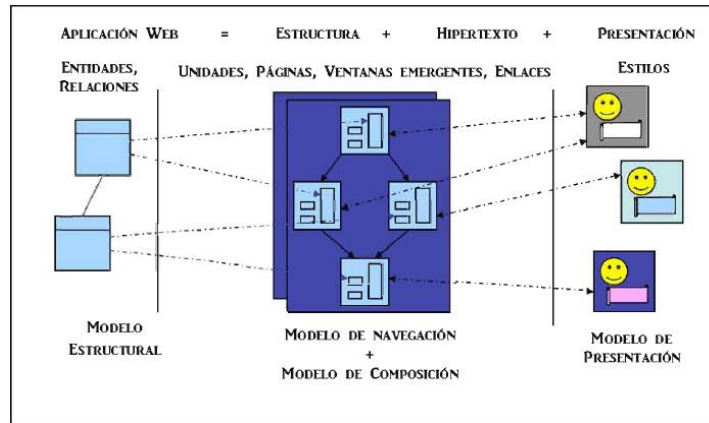


Figura 2.8. Enfoque WebML
Fuente: Pastor, 2008

2.11.4 ELEMENTOS DEL MODELADO DE HIPERTEXTO WebML

2.11.4.1 UNIDADES DE DATOS

Las unidades de datos son definidas para seleccionar información que proveen significado a un concepto dado de un esquema de estructura. Para ello se requiere la indicación del concepto al cual se refiere la unidad por medio del elemento “DataUnit” y su contenido se expresa por medio de “Element” de la forma en que se codifica y muestra la figura 2.9.

```
<DATAUNIT id="Tipo.Artista" entity="Artist">
  <INCLUDE attribute="nombre" />
  <INCLUDE attribute="apellido" />
  <INCLUDE attribute="foto" />
</DATAUNIT>
```

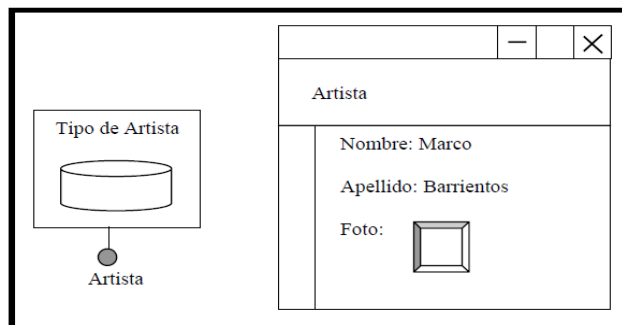


Figura 2.9. Notación para unidades de datos
Fuente: Peña & Gutiérrez, 2004

2.11.4.2 UNIDADES DE MULTIDATOS

Representan múltiples instancias de una entidad al repetir la presentación de varias unidades idénticas, tiene 2 partes:

El contenedor “MultidataUnit” que incluye las instancias a ser desplegadas, las cuales pueden referirse a una entidad, relación o atributo.

La unidad de datos usados en la representación de cada instancia, tal como muestra la figura 2.10.

```
<MULTIDATAUNIT id="MultiAlbumUnidad" entity="Artista">  
  <DATAUNIT id="TipoArtista" entity="Artista">  
    <INCLUDEALL/>  
  </DATAUNIT>  
</MULTIDATAUNIT>
```

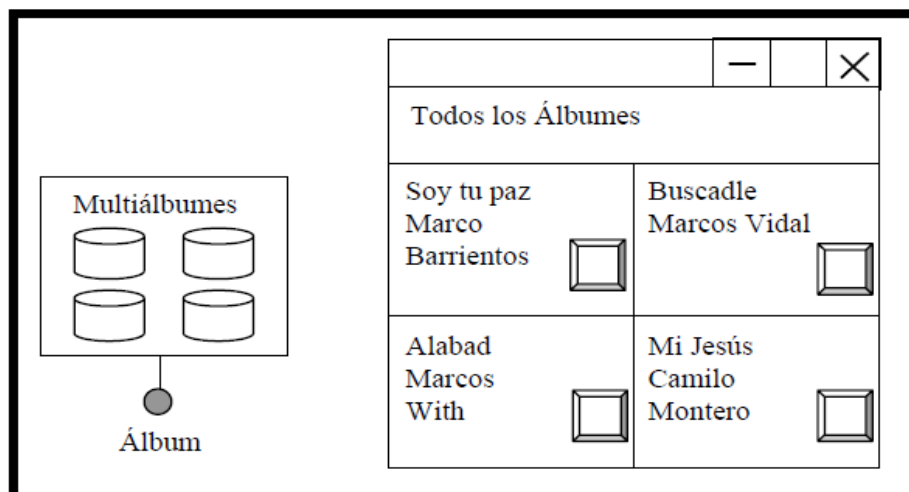


Figura 2.10. Notación para unidades multidadatos
Fuente: Peña & Gutiérrez, 2004

2.11.4.3 UNIDADES DE DESPLAZAMIENTO

Conocidas como las unidades de recorrimiento ofrecen medios para desplazarse a través de los objetos en un contenedor como son las instancias de una entidad o los objetos asociados a otro vía una relación. Para ello utiliza “ScrollerUnit” tal como muestra la figura 2.11.


```
<SCROLLERUNIT id="ÁlbumRecorrido" entity="Álbum" first="yes"
last="yes" previous="yes" next="yes" />
```

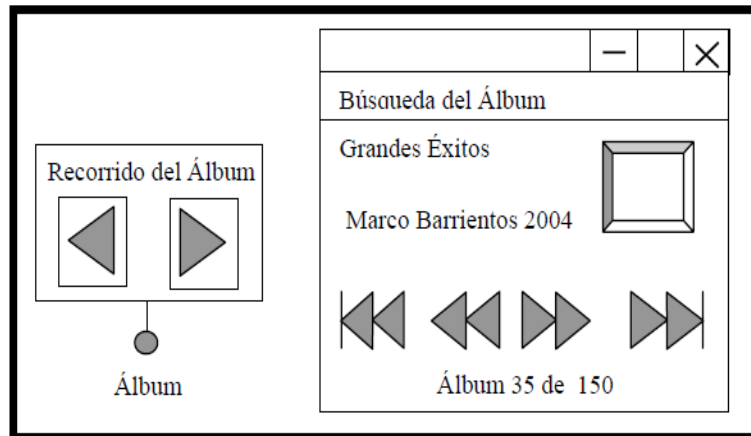


Figura 2.11. Notación para unidades de desplazamiento
Fuente: Peña & Gutiérrez, 2004

2.11.4.4 UNIDADES DE FILTRO

Las unidades de filtro facilitan la selección de objetos de un contenedor a través del suministro de un valor de búsqueda asociado a los atributos de la unidad por medio de la palabra reservada "FilterUnit", se muestra en la figura 2.12 su estructura.

```
<FILTERUNIT id="ÁlbumFiltro" entity="Álbum" />
</SEARCHATTRIBUTE name="título" predicate="like">
</SEARCHATTRIBUTE name="año" predicate="like">
</FILTERUNIT>
```

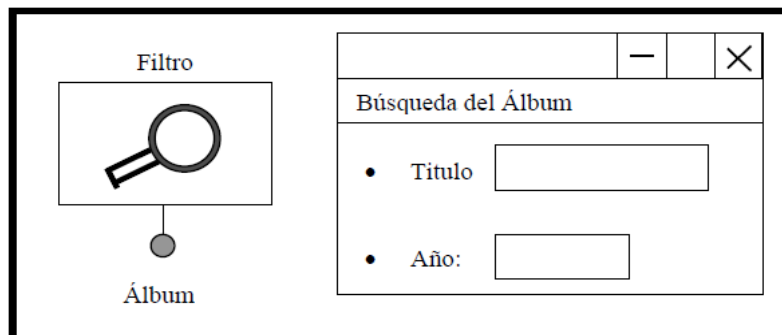


Figura 2.12. Notación para unidades de filtro
Fuente: Peña & Gutiérrez, 2004

2.11.4.5 PÁGINAS

Las páginas se organizan internamente en unidades o recursivamente en otras páginas, también puede haber páginas anidadas, y algunas que pueden considerarse subpáginas de otras para representar muchas estructuras de páginas complejas cuya organización de unidades se establece por medio del elemento “page” y la “alternative” para especificar subpáginas alternativas tal como se muestra a continuación con un ejemplo de páginas anidadas en la figura 2.13

```
<PAGE id="outermost" >  
  <PAGE id="leftmost" ><UNIT id="Índice pasado" /><UNIT id="Índice este año"/>  
</PAGE>  
<PAGE id="rightmost" ><UNIT id="Información Álbum" /> </PAGE>  
</PAGE>
```

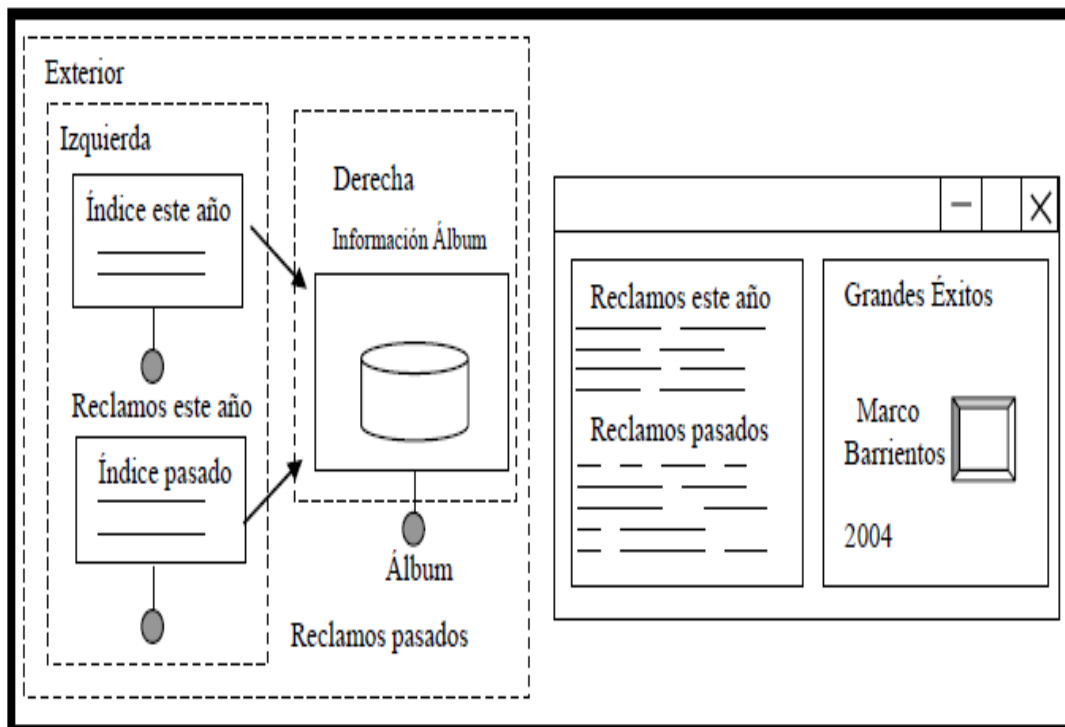


Figura 2.13. Notación de texto y grafica en páginas anidadas
Fuente: Peña & Gutiérrez, 2004

Otro ejemplo de navegación alterna se muestra en la figura 2.14 antecedido por su notación correspondiente.

```

<PAGE id="outermost" > <PAGE id="leftmost" >
  <UNIT id="Índice de Artista" /> <UNIT id="Índice del Álbum"/> </PAGE>
<PAGE id="rightmost1" >
  <ALTERNATIVE>
    <PAGE id="rightmost1"> <UNIT id="Info. Artista"/> </PAGE>
    <PAGE id="rightmost2"> <UNIT id="Info. Álbum"/> </PAGE>
  </ALTERNATIVE>
</PAGE> </PAGE>

```

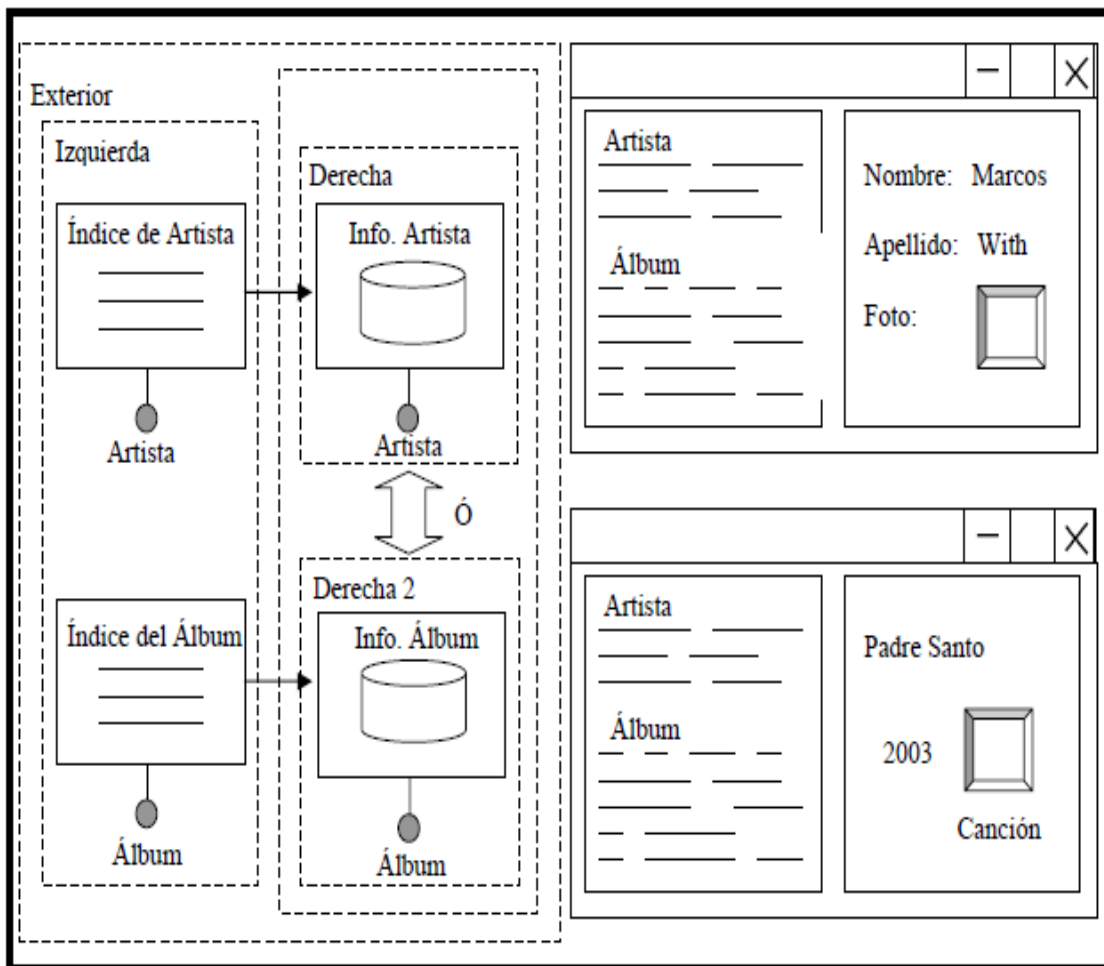


Figura 2.14. Notación de texto y grafica en páginas alternas
Fuente: Peña & Gutiérrez, 2004

2.12 INGENIERÍA MÓVIL

2.12.1 INTRODUCCIÓN

El desarrollo de software requiere de un proceso planeado y estandarizado si se quiere generar productos de alta calidad, tanto en su documentación como en su aplicativo final. Para ello existen varios métodos de ingeniería de software que establecen un conjunto de entregables con el ánimo de dar una trazabilidad al producto, y asegurar que pueda ser interpretado, actualizado y adaptado fácilmente por los usuarios finales. [Vargas, 2011].

Las metodologías de desarrollo software difieren del software móvil, las mismas fueron concebidas para el desarrollo de aplicaciones cliente-servidor, sin tener muy en cuenta el avance de los diferentes ambientes de desarrollo móvil, los cuales requieren un sinnúmero de particularidades que hacen insuficientes las etapas de ciclo del vida del software para responder a las necesidades de los interesados. [Zapata & Vargas, 2009].

2.12.2 CARACTERÍSTICAS DEL ENTORNO MÓVIL

El desarrollo de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos también difieran de las del software clásico. Esto es porque el software móvil tiene que satisfacer una serie de requerimientos y condicionantes especiales que lo hace más complejo:

- **Canal radio:** consideraciones tales como la disponibilidad, las desconexiones, la variabilidad del ancho de banda, la heterogeneidad de redes o los riesgos de seguridad han de tenerse especialmente en cuenta en este entorno de comunicaciones móviles.
- **Movilidad:** aquí influyen consideraciones como la migración de direcciones, alta latencia debido a cambio de estación base o la gestión de la información dependiente de localización. Sobre esta última, de hecho, se pueden implementar un sinnúmero de aplicaciones, pero la información de contexto asociada resulta muchas veces incompleta y varía frecuentemente.

- **Portabilidad:** la característica portabilidad de los dispositivos terminales implica una serie de limitaciones físicas directamente relacionadas con el factor de forma de los mismos, como el tamaño de las pantallas (algo que ha variado sustancialmente con la popularización de las pantallas táctiles), o del teclado, limitando también el número de teclas y su disposición.
- **Fragmentación de la industria:** la existencia de una considerable variedad de estándares, protocolos y tecnologías de red diferentes añaden complejidad al escenario del desarrollo móvil.
- **Capacidades limitadas de los terminales:** aquí incluimos factores como la baja potencia de cálculo o gráfica, los riesgos en la integridad de datos, las interfaces de usuario poco funcionales en muchos aspectos, la baja capacidad de almacenamiento, la duración de las baterías o la dificultad para el uso de periféricos en movilidad. Factores todos que, por otro lado, están evolucionando en la dirección de la convergencia de los ultra portátiles con los dispositivos inteligentes constituyendo cada vez menos un elemento diferencial.
- **Diseño:** desde el punto de vista del desarrollo, el diseño multitarea y la interrupción de tareas es clave para el éxito de las aplicaciones de escritorio; pero la oportunidad y frecuencia de éstas es mucho mayor que en el software tradicional, debido al entorno móvil que manejan, complicándose todavía más debido a la limitación de estos dispositivos.
- **Usabilidad:** las necesidades específicas de amplios y variados grupos de usuarios, combinados con la diversidad de plataformas tecnológicas y dispositivos, hacen que el diseño para todos se convierta en un requisito que genera una complejidad creciente difícil de acotar.
- **Time-to-market:** en un sector con un dinamismo propio, dentro de una industria en pleno cambio, los requisitos que se imponen en términos de tiempo de lanzamiento son muy estrictos y añaden no poca dificultad en la gestión de los procesos de desarrollo.

El diseño de sistemas de software móvil es, por tanto, bastante más complejo que el tradicional visto en los otros proyectos de desarrollo, forzando a los investigadores a reconsiderar el uso de las metodologías actuales de desarrollo de software. Como se ha visto, el uso de metodologías ágiles es el medio más apropiado para el desarrollo de tecnología en móviles, aunque las características especiales de los terminales y de las redes de telefonía móvil demandan algunos ajustes sobre las actuales metodologías ágiles. [Vargas, 2011].

2.12.3 CICLO DE VIDA DEL SOFTWARE PARA DISPOSITIVOS MÓVILES

Cuando se piensa desarrollar una aplicación para un dispositivo móvil en cualquiera de las plataformas y para cualquier entorno es de vital importancia reconocer y establecer condiciones que garanticen la pertinencia, la calidad, la seguridad, la eficiencia y el rendimiento del programa que se desea construir y utilizar por medio del dispositivo móvil [Fling, 2009].

Por tal razón es de suma importancia seguir en forma clara las etapas generales del ciclo de vida del software (definición y análisis de requisitos, diseño, desarrollo, pruebas, y mantenimiento), pero teniendo muy presentes las grandes diferencias que existen entre el desarrollo de una aplicación para ejecutar en un PC de escritorio y el de una aplicación para ejecutar en un dispositivo móvil.

Cada etapa debe ir enfocada a establecer muy claramente qué es lo que se busca en una pieza de software para un dispositivo móvil, que garantice la movilidad, el fácil uso y el aprovechamiento de los limitados recursos de memoria.

2.12.3 METODOLOGÍAS AL DESARROLLO DE APLICACIONES MÓVILES

En el mundo del desarrollo de software existen muchos métodos de desarrollo, cada uno con sus puntos fuertes y sus puntos débiles. En el caso del desarrollo de aplicaciones móviles sucede lo mismo, y cuando planteas qué método elegir debes saber escoger en función de tus necesidades. [Ramírez, 2013].

Algunos de los métodos más conocidos son los siguientes:

- Modelo waterfall
- Desarrollo rápido de aplicaciones
- Desarrollo ágil (cualquiera de sus variantes)
- Mobile-D

2.13 MOBILE-D

Una metodología de desarrollo nueva, especialmente diseñada para el desarrollo de aplicaciones móviles, recibe el nombre de Mobile-D y es propuesta por Pekka Abrahamsson y su equipo del VTT (Valtion Teknillinen Tutkimuskeskus, en inglés Technical Research Centre of Finland) en Finlandia que lideran una corriente muy importante de desarrollo ágil. [Pekka, 2007].

El método se basa en prácticas ágiles como Extreme Programming y crystal; las prácticas asociadas a Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software. [Abrahamsson & Hanhineva & Hulkko & Ihme & Jääliñoja & Korkala & Koskela & Kyllönen & Salo].

Mobile-D también está orientada al desarrollo de aplicaciones como: seguridad, financieras, logística y de simulación. La misma es utilizada por un equipo de no más de diez desarrolladores, trabajando en conjunto para suministrar un producto listo en un plazo máximo de diez semanas.

2.13.1 CICLO DE VIDA DE MOBILE-D

Siguiendo el enfoque de Mobile-D se compone de 5 fases como se observa en la figura 2.15 exploración, inicialización, producción, estabilización y pruebas del sistema. Cada una de estas fases tiene asociado etapas, tareas y prácticas.

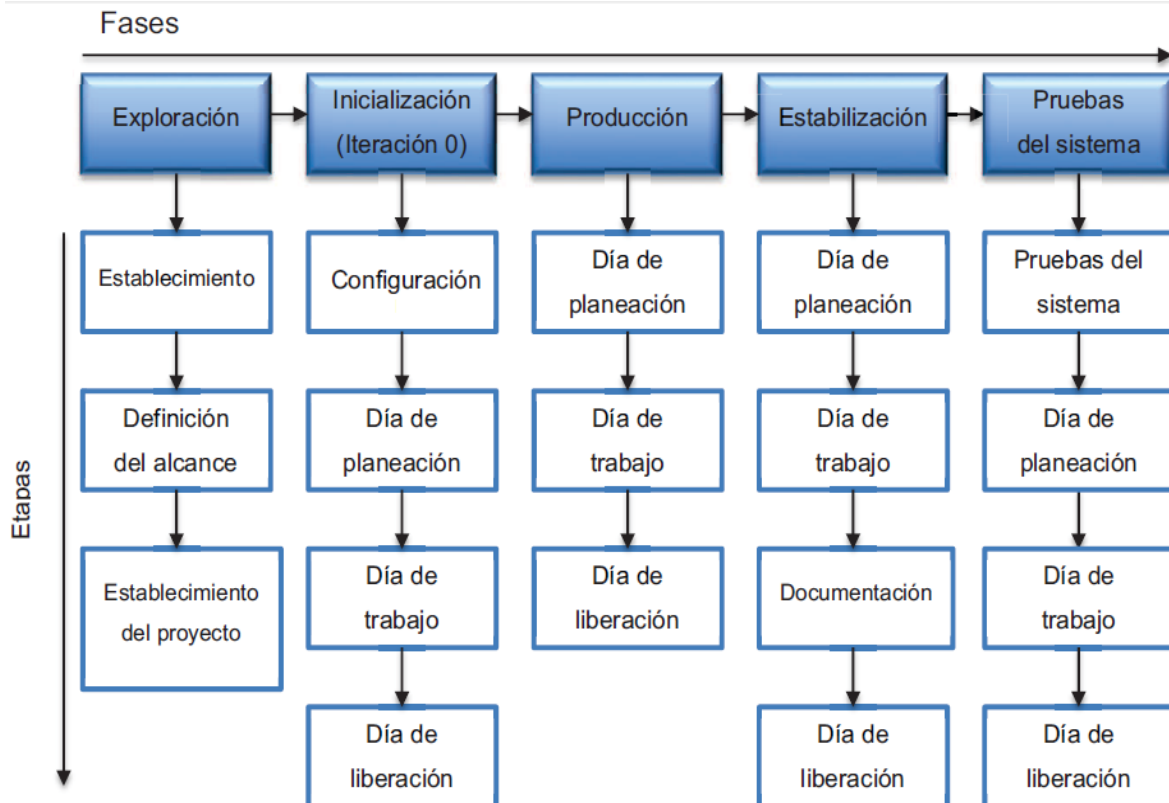


Figura 2.15. Fases y etapas de Mobile-D
Fuente: VTT and K2 Quality Partners, 2015

Exploración.- El equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento actores, definición del alcance y el establecimiento de proyectos.

Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos.

Inicialización.- Los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico como los recursos físicos, tecnológicos y de comunicaciones (incluyendo el entrenamiento del equipo de desarrollo).

Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

Producción.- Se repite la programación de tres días (planificación, trabajo, liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano.

Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

Estabilización.- Se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos.

En esta fase, los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de “producción”, aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

Prueba.- Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

CAPÍTULO 3

MARCO APLICATIVO

3.1 INTRODUCCIÓN

El capítulo dará a conocer las fases de la metodología XP y MOBILE-D desarrollando el sistema web y el aplicativo móvil respectivamente. Mostrado prácticas, procesos y principios. Entendiéndolo mejor de forma interactiva por medio de tablas y figuras.

3.1.1 INTERRELACIÓN DE METODOLOGÍAS

La apertura de Mobile-D permite refactorizar sus fases en la programación extrema, recordando que la metodología móvil tiene sus bases en XP. Llevando así el desarrollo del sistema de manera complementaria.

La tabla 3.1 muestra la interrelación de metodologías implementadas en el sistema.

FASES		ARTEFACTOS XP	ARTEFACTOS MOBILE-D	MODELOS WEB ML
MOBILE-D	XP			
Exploración Inicialización	Planificación	<ul style="list-style-type: none"> • Historias de usuario • Tarjetas tarea • Plan de entregas (iteraciones) • Cronograma de iteraciones • Metáfora del sistema 	<ul style="list-style-type: none"> • Establecimiento de usuarios • Requerimientos del producto 	
	Diseño	<ul style="list-style-type: none"> • Tarjetas CRC 		<ul style="list-style-type: none"> • Modelo estructural • Modelo hipertexto • Modelo presentación
Producción Estabilización	Desarrollo	<ul style="list-style-type: none"> • Captura de pantallas 	<ul style="list-style-type: none"> • Planificación de iteración • Diagrama de estabilización 	
Prueba	Prueba	<ul style="list-style-type: none"> • Pruebas de aceptación y stress 	<ul style="list-style-type: none"> • Pruebas de funcionamiento 	

Tabla 3.1. Interrelación de metodologías
Fuente: Elaboración propia

A continuación se describen las etapas de numeración de la interrelación metodologías para una mejor comprensión del capítulo, detallando cada una de las fases y procesos de la refactorización.

Etapa 1 – PLANIFICACIÓN, EXPLORACIÓN E INICIALIZACIÓN

En la primera etapa refactorizamos las fases de exploración e inicialización de Mobile-D en la fase de planificación de XP. Implementando de manera conjunta en ambas metodologías los artefactos y procesos de las mismas.

A continuación se describen el desarrollo de las mismas en el orden aplicado: establecimiento de usuarios involucrados, requerimientos del producto, historias de usuario, tarjetas tarea, plan de entregas, cronograma de iteraciones y la metáfora del sistema.

Etapa 2 – DISEÑO

En la segunda etapa nos enfocamos exclusivamente a la metodología XP y el lenguaje de modelado WebML.

Inicialmente capturamos el diseño y el entorno del sistema web por medio de las tarjetas CRC provenientes de la programación extrema, con las cuales recabaremos información que nos permitirá establecer el lenguaje del modelado web a través de los modelos de: estructura o datos, hipertexto y presentación.

Etapa 3 – DESARROLLO, PRODUCCIÓN Y ESTABILIZACIÓN

La tercera etapa la dividiremos en 2 partes: la primera orientada a la programación extrema y la segunda a Mobile-D:

En la primera parte se muestra el desarrollo del sistema web por medio de las capturas de pantalla, cumpliendo así con las respectivas iteraciones planteadas en las etapas anteriores.

En la segunda parte se detalla la: fase de producción, la fase estabilización de la metodología Mobile-D implementando las iteraciones y procesos correspondientes al aplicativo móvil.

Etapa 4 - PRUEBA

En la última etapa se implementan las pruebas de aceptación tanto para la metodología XP como Mobile-D para verificar el funcionamiento adecuado. Además se realizan pruebas de stress para el sistema web y pruebas de funcionamiento para el aplicativo móvil.

3.2 PLANIFICACIÓN, EXPLORACIÓN E INICIALIZACIÓN

3.2.1 ESTABLECIMIENTO DE USUARIOS XP y MOBILE-D

El establecimiento de usuarios nos permite tener un mejor contexto de cómo va interactuar el sistema web y el aplicativo móvil con su entorno. Así más adelante se recabara las historias de usuario de manera clara entre el cliente y el desarrollador.

La tabla 3.2 muestra los usuarios involucrados en el sistema web y describiendo sus características.

Usuario	Descripción
Usuario del sistema web	Persona que accede al sistema web por medio de una dirección de internet. Una vez dentro, visita el contenido y los recursos que ofrece la productora a su vez puede escuchar el contenido de audio.
Administrador	Persona que gestiona el sistema y genera contenido de audio por medio del servidor streaming.

Tabla 3.2. Establecimiento de usuarios del sistema web
Fuente: Elaboración propia

La tabla 3.3 muestra los usuarios involucrados en el aplicativo móvil y describiendo sus características.

Usuario	Descripción
Usuario del aplicativo móvil	Persona que descarga e instalada la aplicación en su dispositivo móvil. Y escucha el contenido de audio.
Administrador	Persona que genera contenido de audio por medio del servidor streaming.

Tabla 3.3. Establecimiento de usuarios del aplicativo móvil
Fuente: Elaboración propia

3.2.2 REQUERIMIENTOS DEL PRODUCTO XP y MOBILE-D

La tabla 3.4 muestra los requerimientos para la construcción del sistema web y el aplicativo móvil recabando por parte del cliente, detallando sus particularidades correspondientes.

Requerimientos sistema web	Requerimientos aplicativo móvil
<ul style="list-style-type: none">• Acceso al sistema web por medio de internet a través del dominio de la productora.• Resaltar la insignia de la productora en la página de inicio.• Resaltar las actividades de la productora.• Acceso a redes sociales y publicaciones de la productora en las mismas.• Sistema de diseño adaptable a distintos dispositivos.• Reproductor de audio en sistema web.	<ul style="list-style-type: none">• Descargar la aplicación del sistema web o una tienda de aplicaciones gratuita.• Manipulación sencilla, sin complicaciones solo los elementos necesarios.• Identificación institucional en la aplicación.• La aplicación permita escuchar el contenido de audio.

Tabla 3.4. Requerimientos del producto web y móvil
Fuente: Elaboración propia

3.2.3 HISTORIAS DE USUARIO XP y MOBILE-D

A continuación se muestran las historias de usuario recolectadas en la productora “LOS SUPER CLASICOS”, las historias fueron recabadas en distintas reuniones llevadas a cabo con el gerente de la unidad.

Con las historias de usuarios presentadas a continuación se podrá conocer cómo están organizados los módulos del sistema web y las características que tendrá el aplicativo móvil a desarrollar detallando sus diferentes procesos. De acuerdo a las entrevistas realizadas se pudieron detectar sus necesidades plasmadas en el sistema.

HISTORIA DE USUARIO Nro. 1 (ver tabla 3.5), responde a la necesidad de poder transmitir contenido de audio a través del servidor de streaming, la cual será escuchada por medio del sistema web y la aplicación móvil.

HISTORIA DE USUARIO	
Numero: 1	Nombre de Historia de Usuario: Implementar un servicio de un servidor de audio streaming
Usuario: Gerente de la productora	Prioridad en Negocio: Alta (Alta/Media/Baja)
Descripción: El servidor es el encargado de transmitir el contenido de audio a las terminales (aplicación web y móvil) usando la tecnología streaming. Que permite a los oyentes no tener que esperar minutos hasta que se baje el audio completamente, sino escuchar en tiempo real.	
Observaciones: El servidor de audio streaming tendrá que transmitir de manera continua y sin interrupciones, ya que es pilar fundamental del sistema.	

Tabla 3.5. Historia de usuario Nro. 1
Fuente: Elaboración propia

HISTORIA DE USUARIO Nro. 2 (ver tabla 3.6), responde a la necesidad de tener un sistema web que pueda ser visualizado en dispositivos móviles y equipos portátiles ya que la adquisición de estas terminales va en aumento masivamente.

HISTORIA DE USUARIO	
Numero: 2	Nombre de Historia de Usuario: Sistema web con diseño adaptativo
Usuario: Gerente de la productora	Prioridad en Negocio: Alta (Alta/Media/Baja)
Descripción: Una vez implementado el servidor de audio streaming, se tendrá que escuchar por medio del sistema web y visto por medio de un diseño adaptativo, es decir que será visualizado en diferentes terminales (tabletas, teléfonos inteligentes, laptops) de manera agradable al usuario final.	
Observaciones:	

Tabla 3.6. Historia de usuario Nro. 2
Fuente: Elaboración propia

HISTORIA DE USUARIO Nro. 3 (ver tabla 3.7), responde a la necesidad de tener información y contenido de la productora, pero como recurso más destacado el portal alojara el reproductor de audio que recibirá la emisión del servidor de streaming.

HISTORIA DE USUARIO	
Numero: 3	Nombre de Historia de Usuario: Portal de la productora
Usuario: Gerente de la productora	Prioridad en Negocio: Alta (Alta/Media/Baja)
Descripción: Es el inicio del sistema, contendrá variedad de servicios y recursos. Lo más destacado permitirá escuchar el contenido de enviado por el servidor a través de un reproductor de audio.	
Observaciones:	

Tabla 3.7. Historia de usuario Nro. 3
Fuente: Elaboración propia

HISTORIA DE USUARIO Nro. 4 (ver tabla 3.8), responde a la necesidad de exponer la información y servicios que ofrece la productora al usuario final refiriéndose a los oyentes y seguidores.

HISTORIA DE USUARIO	
Numero: 4	Nombre de Historia de Usuario: Portal de información de la productora
Usuario: Gerente de la productora	Prioridad en Negocio: Alta (Alta/Media/Baja)
Descripción: Exhibe información y servicios importantes de la productora, que ayuda de gran manera al usuario final.	
Observaciones:	

Tabla 3.8. Historia de usuario Nro. 4
Fuente: Elaboración propia

HISTORIA DE USUARIO Nro. 5 (ver tabla 3.9), responde a la necesidad de exponer los auspicios con los que cuenta la productora y de esa manera percibir ingresos económicos a la empresa.

HISTORIA DE USUARIO	
Numero: 5	Nombre de Historia de Usuario: Portal de auspiciadores
Usuario: Gerente de la productora	Prioridad en Negocio: Alta (Alta/Media/Baja)
Descripción: El portal de auspiciadores visualiza los anuncios con los cuales cuenta la productora al público, cuyo objetivo es promover los negocios de los servicios publicitarios.	
Observaciones:	

Tabla 3.9. Historia de usuario Nro. 5
Fuente: Elaboración propia

HISTORIA DE USUARIO Nro. 6 (ver tabla 3.10), responde a la necesidad poder dar a conocer la dirección de donde se encuentra ubicada la productora, si se requiere algún servicio.

HISTORIA DE USUARIO	
Numero: 6	Nombre de Historia de Usuario: Portal de contactos
Usuario: Gerente de la productora	Prioridad en Negocio: Media (Alta/Media/Baja)
Descripción: Visualiza la ubicación e información de la productora para el público si requiere algún servicio que brinda la productora.	
Observaciones:	

Tabla 3.10. Historia de usuario Nro. 6
Fuente: Elaboración propia

HISTORIA DE USUARIO Nro. 7 (ver tabla 3.11), responde a la necesidad de tener al alcance una aplicación móvil, ya que la adquisición de teléfonos inteligentes va aumentando y se quiere mayor llegada al público en general.

HISTORIA DE USUARIO	
Numero: 7	Nombre de Historia de Usuario: Aplicativo móvil
Usuario: Gerente de la productora	Prioridad en Negocio: Media (Alta/Media/Baja)
Descripción: La aplicación móvil permite escuchar el contenido de audio enviado desde el servidor de streaming el mismo es operado por el administrador.	
Observaciones: El servidor audio streaming debe transmitir de manera continua y sin interrupciones para el buen funcionamiento de la aplicación.	

Tabla 3.11. Historia de usuario Nro. 7
Fuente: Elaboración propia

3.2.4 TARJETA DE TAREA XP y MOBILE-D

A continuación se describen las tarjetas de tarea que corresponden a cada una de las historias de usuarios descritas anteriormente.

TARJETA DE TAREA	
Numero de Tarea: 1.1	Historia de Usuario: 1. Implementar un servicio de un servidor de audio streaming
Nombre Tarea: Levantar el modulo servidor de audio streaming	
Tipo de Tarea: Implementación del servidor de streaming	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Realizar el módulo de levantar el servidor de audio streaming implica realizar suscripciones, instalaciones y una serie de configuraciones.	

Tabla 3.12. Tarjeta de Tarea Implementacion modulo levantar servidor de streaming
Fuente: Elaboración propia

La tarjeta de tarea 1.1 (ver tabla 3.12) describe el trabajo de realizar el módulo de configurar e instalar el servidor de streaming para emitir el contenido de audio a las diferentes terminales.

TARJETA DE TAREA	
Numero de Tarea: 1.2	Historia de Usuario: 1. Implementar un servicio de un servidor de audio streaming
Nombre Tarea: Obtención de la dirección IP y el puerto del servidor streaming	
Tipo de Tarea: Obtención	
Programador Responsable: Ovidio Rojas Nina	
Descripción: La principal característica es obtener la dirección IP y el puerto del servidor de audio streaming que nos permitirá escuchar la transmisión tanto en el sistema web como en la aplicación móvil.	

Tabla 3.13. Tarjeta de Tarea Obtención de la dirección IP del servidor streaming
Fuente: Elaboración propia

La tarjeta de tarea 1.2 (ver tabla 3.13) describe la tarea de rescatar la dirección IP y el número de puerto del servidor de audio streaming para enlazar posteriormente al sistema web y al aplicativo móvil.

TARJETA DE TAREA	
Numero de Tarea: 2.1	Historia de Usuario: 2. Sistema web con diseño adaptativo
Nombre Tarea: Modulo de maquetación del sistema web	
Tipo de Tarea: Diseño	
Programador Responsable: Ovidio Rojas Nina	
Descripción: La maquetación representa la ubicación de los elementos del sistema web de cómo se adaptaran visualmente en las diferentes terminales manteniendo la estructura de diseño.	

Tabla 3.14. Tarjeta de Tarea Maquetación del sistema web
Fuente: Elaboración propia

La tarjeta de tarea 2.1 (ver tabla 3.14) describe visualmente cómo se verán los elementos del sistema en las terminales y como se van adaptando sin modificar su estructura de diseño.

TARJETA DE TAREA	
Numero de Tarea: 2.2	Historia de Usuario: 2. Sistema web con diseño adaptativo
Nombre Tarea: Maquetación ordenadores de escritorio	
Tipo de Tarea: Diseño	
Programador Responsable: Ovidio Rojas Nina	
Descripción: La maquetación para ordenadores de escritorio representa la ubicación de los elementos del sistema web observándose de manera agradable y respetando la estructura.	

Tabla 3.15. Tarjeta de Tarea Maquetación ordenadores de escritorio
Fuente: Elaboración propia

La tarjeta de tarea 2.2 (ver tabla 3.15) describe la maquetación de diseño para ordenadores de escritorio contiendo los elementos del sistema web viéndose de manera elegante y manteniendo la estructura.

TARJETA DE TAREA	
Numero de Tarea: 2.3	Historia de Usuario: 2. Sistema web con diseño adaptativo
Nombre Tarea: Maquetación laptops	
Tipo de Tarea: Diseño	
Programador Responsable: Ovidio Rojas Nina	
Descripción: La maquetación para laptops representa la ubicación de los elementos del sistema web observándose de manera agradable y respetando la estructura.	

Tabla 3.16. Tarjeta de Tarea Maquetación laptops
Fuente: Elaboración propia

La tarjeta de tarea 2.3 (ver tabla 3.16) describe la maquetación de diseño para laptops contiendo los elementos del sistema web viéndose de manera elegante y manteniendo la estructura.

TARJETA DE TAREA	
Numero de Tarea: 2.4	Historia de Usuario: 2. Sistema web con diseño adaptativo
Nombre Tarea: Maquetación tablets	
Tipo de Tarea: Diseño	
Programador Responsable: Ovidio Rojas Nina	
Descripción: La maquetación para tablets representa la ubicación de los elementos del sistema web observándose de manera agradable y respetando la estructura.	

Tabla 3.17. Tarjeta de Tarea Maquetación tablets
Fuente: Elaboración propia

La tarjeta de tarea 2.4 (ver tabla 3.17) describe la maquetación de diseño para tablets contiendo los elementos del sistema web viéndose de manera elegante y manteniendo la estructura.

TARJETA DE TAREA	
Numero de Tarea: 2.5	Historia de Usuario: 2. Sistema web con diseño adaptativo
Nombre Tarea: Maquetación teléfonos inteligentes	
Tipo de Tarea: Diseño	
Programador Responsable: Ovidio Rojas Nina	
Descripción: La maquetación para teléfonos inteligentes representa la ubicación de los elementos del sistema web observándose de manera agradable y respetando la estructura.	

Tabla 3.18. Tarjeta de Tarea Maquetación telefonos inteligentes
Fuente: Elaboración propia

La tarjeta de tarea 2.5 (ver tabla 3.18) describe la maquetación de diseño para teléfonos inteligentes contiendo los elementos del sistema web viéndose de manera elegante y manteniendo la estructura.

TARJETA DE TAREA	
Numero de Tarea: 3.1	Historia de Usuario: 3. Portal de la productora
Nombre Tarea: Desarrollar módulo de acceso de recursos al usuario final	
Tipo de Tarea: Desarrollo	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Realizar el módulo de acceso de recursos para el usuario final, representa poder manipular por el público en general los contenidos que ofrece la productora como ser: logo empresarial, anuncio empresarial, redes sociales, reproductor de audio, entre los más resaltantes.	

Tabla 3.19. Tarjeta de Tarea Módulo de acceso de recursos al usuario final
Fuente: Elaboración propia

La tarjeta de tarea 3.1 (ver tabla 3.19) describe los recursos que ofrece la productora para el usuario final, accediendo de manera libre y fácil a los contenidos.

TARJETA DE TAREA	
Numero de Tarea: 3.2	Historia de Usuario: 3. Portal de la productora
Nombre Tarea: Implementación de la maquetación del portal	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementara las clases necesarias sobre el acceso de recursos al usuario final de esta manera los contenidos estarán mejor ordenados.	

Tabla 3.20. Tarjeta de Tarea Implementación del Diagrama de clases
Fuente: Elaboración propia

La tarjeta de tarea 3.2 (ver tabla 3.20) describe como se implementara la maquetación que correspondan al módulo de acceso de recursos al usuario final y sus respectivos detalles.

TARJETA DE TAREA	
Numero de Tarea: 3.3	Historia de Usuario: 3. Portal de la productora
Nombre Tarea: Desarrollo de Interfaz adaptable	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se desarrollara una interfaz adaptable a diferentes terminales sin alterar la estructura del módulo de acceso al usuario final.	

Tabla 3.21. Tarjeta de Tarea Desarrollo de interfaz adaptable
Fuente: Elaboración propia

La tarjeta de tarea 3.3 (ver tabla 3.21) describe la interfaz del módulo de acceso al usuario final, resaltando que el mismo tendrá un diseño adaptativo a diferentes terminales.

TARJETA DE TAREA	
Numero de Tarea: 3.4	Historia de Usuario: 3. Portal de la productora
Nombre Tarea: Implementación de cartel de anuncios	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementara un cartel de anuncios donde la productora publique contenido relacionado a las actividades que realiza la organización y al usuario final.	

Tabla 3.22. Tarjeta de Tarea Implementación de cartel de anuncios
Fuente: Elaboración propia

La tarjeta de tarea 3.4 (ver tabla 3.22) describe la implementación de carteles de anuncios, que actualiza sobre las actividades que realiza la productora siendo los mismos dinámicos.

TARJETA DE TAREA	
Numero de Tarea: 3.5	Historia de Usuario: 3. Portal de la productora
Nombre Tarea: Implementación de redes sociales	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementaran fragmentos de redes sociales como Facebook y Twitter de la productora, para tener un mayor alcance y comunicación al público en general.	

Tabla 3.23. Tarjeta de Tarea Implementación de redes sociales de la productora
Fuente: Elaboración propia

La tarjeta de tarea 3.5 (ver tabla 3.23) describe la implementación de fragmentos de redes sociales con las que cuenta la productora, la mismas permiten tener mayor relación entre la organización y el público en general. Destacando que las mismas van en aumento.

TARJETA DE TAREA	
Numero de Tarea: 3.6	Historia de Usuario: 3. Portal de la productora
Nombre Tarea: Implementación del reproductor de audio streaming	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementara un reproductor de audio que se enlaza del servidor de streaming previamente configurado e instalado.	

Tabla 3.24. Tarjeta de Tarea Implementación del reproductor de audio
Fuente: Elaboración propia

La tarjeta de tarea 3.6 (ver tabla 3.24) describe la implementación de un reproductor de audio que se conecta al servidor de streaming a través de la dirección IP previamente rescatada.

TARJETA DE TAREA	
Numero de Tarea: 4.1	Historia de Usuario: 4. Portal de información de la productora
Nombre Tarea: Desarrollar el módulo de portal de información	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Realizar el módulo de portal de información de la productora da a conocer en profundidad la historia, actividades y eventos que realiza la organización.	

Tabla 3.25. Tarjeta de Tarea Implementación Módulo portal de informacion
Fuente: Elaboración propia

La tarjeta de tarea 4.1 (ver tabla 3.25) describe la implementación del módulo de portal de información que actualiza sobre actividades que realiza la productora para el público en general.

TARJETA DE TAREA	
Numero de Tarea: 4.2	Historia de Usuario: 4. Portal de información de la productora
Nombre Tarea: Implementación de la maquetación del portal	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementara las clases necesarias para el portal de información de la productora para que se pueda tener el control adecuado.	

Tabla 3.26. Tarjeta de Tarea Implementación del Diagrama de clases
Fuente: Elaboración propia

La tarjeta de tarea 4.2 (ver tabla 3.26) describe la implementación de la maquetación que corresponden al módulo de portal de información de la productora así también como sus respectivos detalles.

TARJETA DE TAREA	
Numero de Tarea: 4.3	Historia de Usuario: 4. Portal de información de la productora
Nombre Tarea: Desarrollo de Interfaz adaptable	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se desarrollara una interfaz adaptable a diferentes terminales sin alterar la estructura del módulo portal de información de la productora.	

Tabla 3.27. Tarjeta de Tarea Desarrollo de interfaz adaptable
Fuente: Elaboración propia

La tarjeta de tarea 4.3 (ver tabla 3.27) describe la interfaz del módulo portal de información de la productora, resaltando que la mismo tendrá un diseño adaptativo a diferentes terminales.

TARJETA DE TAREA	
Numero de Tarea: 5.1	Historia de Usuario: 5. Portal de auspiciadores
Nombre Tarea: Desarrollar el módulo de portal de auspiciadores	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Realizar el módulo de auspiciadores representa las empresas con las cuales cuenta la productora para percibir ingresos económicos, puestos al público en general para su consumo.	

Tabla 3.28. Tarjeta de Tarea Implementación Módulo portal de auspiciadores
Fuente: Elaboración propia

La tarjeta de tarea 5.1 (ver tabla 3.28) describe la implementación del módulo de portal de auspiciadores, que exhibe a las empresas con las que cuenta la productora las mismas ofrecen sus servicios al público en general.

TARJETA DE TAREA	
Numero de Tarea: 5.2	Historia de Usuario: 5. Portal de auspiciadores
Nombre Tarea: Implementación de la maquetación del portal	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementara las clases necesarias para el portal de auspiciadores de la productora para que se pueda tener el control adecuado.	

Tabla 3.29. Tarjeta de Tarea Implementación del Diagrama de clases
Fuente: Elaboración propia

La tarjeta de tarea 5.2 (ver tabla 3.29) describe la maquetación que corresponden al módulo de portal de auspiciadores de la productora así también como sus detalles.

TARJETA DE TAREA	
Numero de Tarea: 5.3	Historia de Usuario: 5. Portal de auspiciadores
Nombre Tarea: Desarrollo de Interfaz adaptable	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se desarrollara una interfaz adaptable a diferentes terminales sin alterar la estructura del módulo portal de auspiciadores de la productora.	

Tabla 3.30. Tarjeta de Tarea Desarrollo de interfaz adaptable
Fuente: Elaboración propia

La tarjeta de tarea 5.3 (ver tabla 3.30) describe la interfaz del módulo portal de auspiciadores de la productora, resaltando que la misma tendrá un diseño adaptativo a diferentes terminales.

TARJETA DE TAREA	
Numero de Tarea: 6.1	Historia de Usuario: 6. Portal de contactos
Nombre Tarea: Desarrollar el módulo de portal de contactos	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Realizar el módulo de contactos permite conocer dirección sobre la productora si se requiere algún servicio que ofrece la misma por parte del público en general	

Tabla 3.31. Tarjeta de Tarea Implementación Módulo portal de contactos
Fuente: Elaboración propia

La tarjeta de tarea 6.1 (ver tabla 3.31) describe la implementación del módulo de portal de contactos que visualiza la dirección de la productora para el público en general si este requiere algún servicio que brinda la empresa.

TARJETA DE TAREA	
Numero de Tarea: 6.2	Historia de Usuario: 6. Portal de contactos
Nombre Tarea: Implementación de la maquetación del portal	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se implementara las clases necesarias para el portal de contactos de la productora para que se pueda tener el control adecuado.	

Tabla 3.32. Tarjeta de Tarea Implementación del Diagrama de clases
Fuente: Elaboración propia

La tarjeta de tarea 6.2 (ver tabla 3.32) describe la implementación de la maquetación que corresponden al módulo de portal de contactos de la productora así también como sus respectivos detalles.

TARJETA DE TAREA	
Numero de Tarea: 6.3	Historia de Usuario: 6. Portal de contactos
Nombre Tarea: Desarrollo de Interfaz adaptable	
Tipo de Tarea: Mejora	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se desarrollara una interfaz adaptable a diferentes terminales sin alterar la estructura del módulo portal de contactos de la productora.	

Tabla 3.33. Tarjeta de Tarea Desarrollo de interfaz adaptable

Fuente: Elaboración propia

La tarjeta de tarea 6.3 (ver tabla 3.33) describe la interfaz del módulo portal de contactos de la productora, resaltando que la mismo tendrá un diseño adaptativo a diferentes terminales.

TARJETA DE TAREA	
Numero de Tarea: 7.1	Historia de Usuario: 7. Aplicativo móvil
Nombre Tarea: Diseño de la interfaz	
Tipo de Tarea: Desarrollo	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se desarrolla el diseño de la interfaz de la aplicación móvil, considerando la identificación de la institución.	

Tabla 3.34. Tarjeta de Tarea Diseño de la interfaz de la aplicación movil

Fuente: Elaboración propia

La tarjeta de tarea 7.1 (ver tabla 3.34) describe el trabajo de realizar el diseño de la interfaz de la aplicación móvil considerando la insignia de la productora.

TARJETA DE TAREA	
Numero de Tarea: 7.2	Historia de Usuario: 1 Aplicativo móvil
Nombre Tarea: Integración de la dirección IP del servidor streaming	
Tipo de Tarea: Desarrollo	
Programador Responsable: Ovidio Rojas Nina	
Descripción: Se integra la dirección IP del servidor de audio streaming la que permite enlazar al sistema la aplicación móvil.	

Tabla 3.35. Tarjeta de Tarea Integración de la dirección IP del servidor streaming
Fuente: Elaboración propia

La tarjeta de tarea 7.2 (ver tabla 3.35) describe el trabajo de realizar la integración de la dirección IP del servidor de streaming a la aplicación móvil.

3.2.5 PLAN DE ENTREGAS

En el plan de entregas se dará a conocer el número de iteraciones que tendrá el desarrollo del proyecto en el cual una o varias historias de usuarios son agrupadas para ser producidas de forma que no exista una sobre carga de trabajo y cumplir con los tiempos marcados con comodidad.

También consideramos el objetivo de cada iteración y la duración que tendrá la misma para la implementación del sistema.

Se tomó en cuenta todas las observaciones y sugerencias que realizo el gerente de la productora para establecer el plan de entregas (ver tabla 3.36) en coordinación con el programador así trabajando en equipo y con la satisfacción de ambas partes. Cumpliendo así con una de las practicas fundamentales de la metodología XP

HISTORIA DE USUARIO	ITERACION ASIGNADA	OBJETIVO	DURACIÓN
1. Implementar un servicio de un servidor de audio streaming 2. Sistema web con diseño adaptativo 3. Portal de la productora	Primera	Implementar el servidor audio streaming para el sistema web y la aplicación móvil. Diseñar el sistema web adaptable a diferentes terminales. Y para finalizar desarrollar el portal de la productora	3 Semana
4. Portal de información de la productora	Segunda	Desarrollar el portal de la productora y la información de la misma.	1 Semana
5. Portal de auspiciadores 6. Portal de contactos	Tercera	Desarrollar el portal de auspiciadores y contactos de la productora.	2 Semana
7. Aplicativo móvil	Cuarta	Desarrollar la aplicación móvil	2 Semanas

Tabla 3.36. Plan de entregas
Fuente: Elaboración propia

3.2.6 VELOCIDAD DEL PROYECTO

Como se puede observar en la Tabla 3.36 contaremos con un total de 4 iteraciones, es así que dividiremos el tiempo que tenemos para desarrollar el sistema, que son 8 semanas, en tiempos de iteraciones relativamente iguales. A fin de no acelerar ni ralentizar el proyecto y que el mismo se desarrolle de una manera prácticamente continua y sin mucha presión.

Por otro lado también podemos promediar la velocidad del proyecto utilizando:

$$\text{Velocidad del Proyecto} = \text{Numero de Historias de Usuario} / \text{Numero de Iteraciones}$$

$$\text{Velocidad del Proyecto} = 7 / 4$$

$$\text{Velocidad del Proyecto} = 1.75$$

Este indicador se puede interpretar de la siguiente manera: En cada 4 semanas que se trabaje en el proyecto se avanzara 3 historias de usuario más la mitad de la siguiente.

3.2.7 REUNIONES

Las reuniones con el equipo de trabajo se las realizara una vez a la semana, más específicamente cada martes, esto con el fin de mostrar avances en cuanto al sistema.

Esto al final se vuelve en una retroalimentación directa hacia el desarrollo del sistema. De esta manera aseguraremos que el cliente se sienta parte del proyecto, así como su entera satisfacción.

3.2.8 PLANIFICACIÓN DE ITERACIONES

La planificación de las iteraciones ayudara tanto al cliente como al programador para saber de una forma general la duración de cada iteración y de sus respectivas presentaciones. Ayudará al programador a una mejor administración de su tiempo.

ITERACIÓN	SEPTIEMBRE				OCTUBRE			
	Semanas				Semanas			
	1ra	2da	3ra	4ta	1ra	2da	3ra	4ta
1								
2								
3								
4								

Tabla 3.37. Planificación de iteraciones
Fuente: Elaboración propia

3.2.9 METÁFORA DEL SISTEMA

La metáfora del sistema nos ayudara a la fácil comunicación entre el cliente y el equipo de programación. En si el objetivo de la misma es el de poder llegar a obtener un lenguaje común para ambas partes.

METÁFORA DEL SISTEMA
Toda productora gira alrededor de un productor radial y/o televisivo, locutor, auspiciador, cliente y oyentes. Estos componentes se relacionan de acuerdo a la actividad que se realicen.

Tabla 3.38. Metáfora de sistema
Fuente: Elaboración propia

A partir de la metáfora del sistema y del intercambio verbal de ideas con el cliente se llega a generar un glosario de términos. En este definiremos términos que ayudara a ambas partes para una comunicación más sencilla y fluida.

GLOSARIO	
Término	Descripción
Productor radial	Un productor de radio tiene un papel clave en la creación del contenido de un programa de radio, pero usualmente no están involucrados en la presentación.
Productor televisivo	El productor de televisión es responsable de organizar los recursos humanos y técnicos necesarios para elaborar diferentes productos audiovisuales.
Publicidad	La publicidad es una forma de comunicación que intenta incrementar el consumo de un producto o servicio, insertar una nueva marca o producto dentro del mercado de consumo, Este producto puede ser impulsado a través de un medio de comunicación
Locutor	Un locutor es la persona certificada para hablar a través de la radio, como narrador de noticias, dar anuncios o poner música. Además de fungir como voz en off (persona que no sale en pantalla) para la televisión.
Oyente	Los oyentes son quienes escuchan un programa de radio o una cierta emisora.
Productora	Es la producción de contenidos para medios de comunicación como ser: televisión, publicidad y radio.

Tabla 3.39. Glosario de términos
Fuente: Elaboración propia

3.2.10 ESTABLECIMIENTO DE RECURSOS MOBILE-D

Este punto es exclusivo de la metodología Mobile-D donde se describe la herramienta a utilizar para la implementación de la aplicación móvil.

La elección de la plataforma es android que es influenciada en gran medida por el uso actual en el mercado, especialmente en la ciudad de La Paz y también en el país. Es que por esa razón se utilizaron las siguientes herramientas:

- Eclipse ADT (Android Development Tools)
- Android SDK Tools v. 22.0.1
- Android SDK Platform Tools v. 17

3.3 DISEÑO

3.3.1 TARJETAS CRC. XP

La finalidad de estas tarjetas es facilitar la implementación del diagrama de clases, las mismas también serán de ayuda para el modelado web. Su uso será exclusivo de la metodología XP. A continuación se describen las tarjetas CRC del sistema web.

En la tarjeta CRC servidor de streaming (ver tabla 3.40) se puede ver la clase y sus respectiva responsabilidad y colaborador.

Servidor de streaming	
Genera contenido de audio para las diferentes terminales (ordenadores de escritorio, laptops, tablets y teléfonos inteligentes).	Aplicativo web
	Aplicativo móvil

Tabla 3.40. Tarjeta C.R.C. Servidor de streaming
Fuente: Elaboración propia

En la tarjeta C.R.C aplicativo web (ver tabla 3.41) se puede ver la clase y su respectiva responsabilidad y colaboradores con los que se relaciona.

Aplicativo web	
Conforma el sistema web con sus diferentes módulos. Y el principal, recepciona el contenido del servidor de audio streaming.	Servidor de streaming
	Portal productora
	Portal de información
	Portal auspiciadores
	Portal contactos

Tabla 3.41. Tarjeta C.R.C. Aplicativo web
Fuente: Elaboración propia

En la tarjeta C.R.C portal productora (ver tabla 3.42) se puede ver la clase y sus respectivas responsabilidades y colaborador.

Portal productora	
Contiene anuncios importantes de la empresa.	Aplicativo web
Contiene redes sociales de la productora.	
Contiene el reproductor de audio streaming	

Tabla 3.42. Tarjeta C.R.C. Portal productora
Fuente: Elaboración propia

En la tarjeta C.R.C portal información (ver tabla 3.43) se puede ver la clase y su respectiva responsabilidad y colaborador.

Portal de información	
Contiene las actividades y servicios que realiza la productora en los medios de comunicación tanto para clientes como oyentes.	Aplicativo web

Tabla 3.43. Tarjeta C.R.C. Portal de información
Fuente: Elaboración propia

En la tarjeta C.R.C portal auspiciadores (ver tabla 3.44) se puede ver la clase y su respectiva responsabilidad y colaborador.

Portal auspiciadores	
Contiene a las firmas o empresas publicitarias que trabajan con la productora.	Aplicativo web

Tabla 3.44. Tarjeta C.R.C. Portal auspiciadores
Fuente: Elaboración propia

En la tarjeta C.R.C portal contactos (ver tabla 3.45) se puede ver la clase y su respectiva responsabilidad y colaborador.

Portal contactos	
Contiene información de ubicación de la productora para recibir contactos por parte de los oyentes y/o clientes para los servicios que brinda la empresa.	Aplicativo web

Tabla 3.45. Tarjeta C.R.C. Portal contactos
Fuente: Elaboración propia

En la tarjeta C.R.C aplicativo móvil (ver tabla 3.46) se puede ver la clase y su respectiva responsabilidad y colaborador.

Aplicativo móvil	
Recepciona el contenido de audio por parte del servidor.	Servidor de streaming

Tabla 3.46. Tarjeta C.R.C. Aplicativo móvil
Fuente: Elaboración propia

3.3.2 MODELO ESTRUCTURAL WEBML

El modelo estructural (ver figura 3.1) representa clases y relaciones a utilizar, las mismas son representadas en un diagrama de clases que brinda un mejor contexto del sistema.

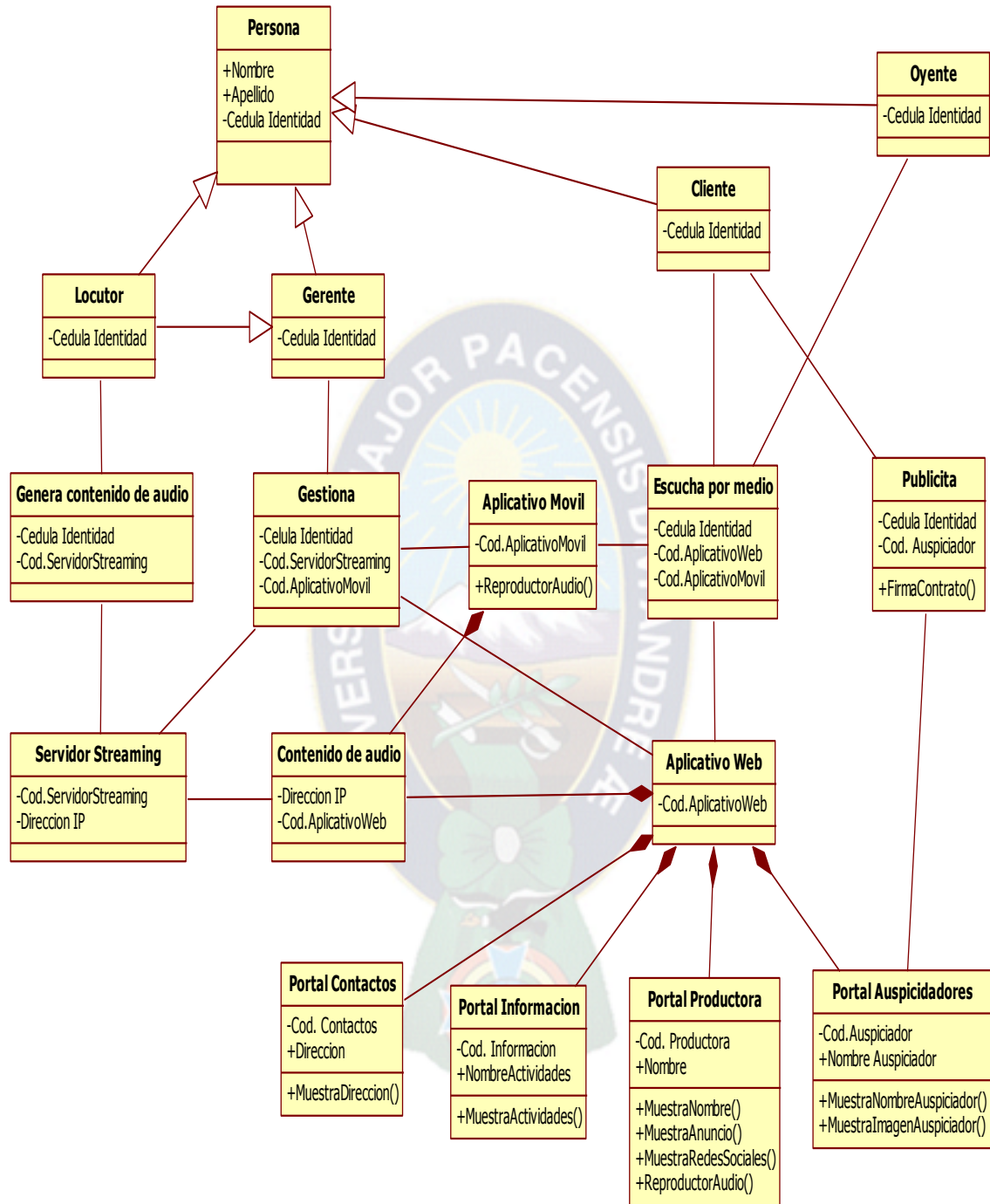


Figura 3.1. Diagrama de clases
Fuente: Elaboración propia

3.3.3 PRIMERA ITERACIÓN XP y WEB ML

Uno de los requisitos de la metodología WebML, incluye los modelos de presentación y de hipertexto, que a su vez está formado por la navegación y la composición del sitio visual. De esta manera se presentan dichos modelos para su posterior implementación y desarrollo.

3.3.3.1 MODELO DE HIPERTEXTO

El siguiente diagrama (ver Figura 3.2) muestra la navegación y la composición que se va a desarrollar el módulo de acceso de recursos al usuario final que pertenece la historia de usuario 3 y a sus respectivas tareas dentro a lo que corresponde al modelado web.

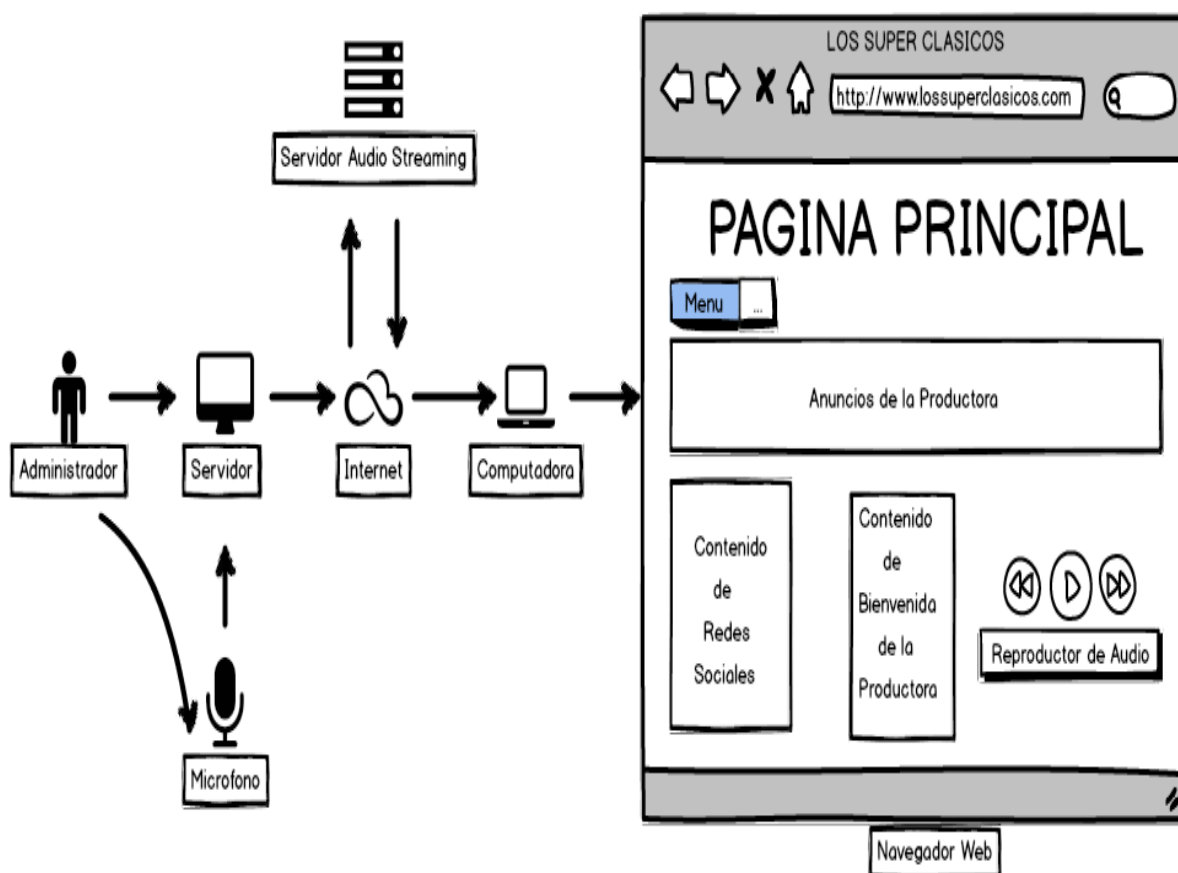


Figura 3.2. Diagrama de hipertexto Acceso a recursos al usuario final
Fuente: Elaboración propia

3.3.3.2 MODELO DE PRESENTACIÓN

La pantalla del portal de la productora (ver Figura 3.3) muestra el desarrollo del módulo de acceso de recursos al usuario final.

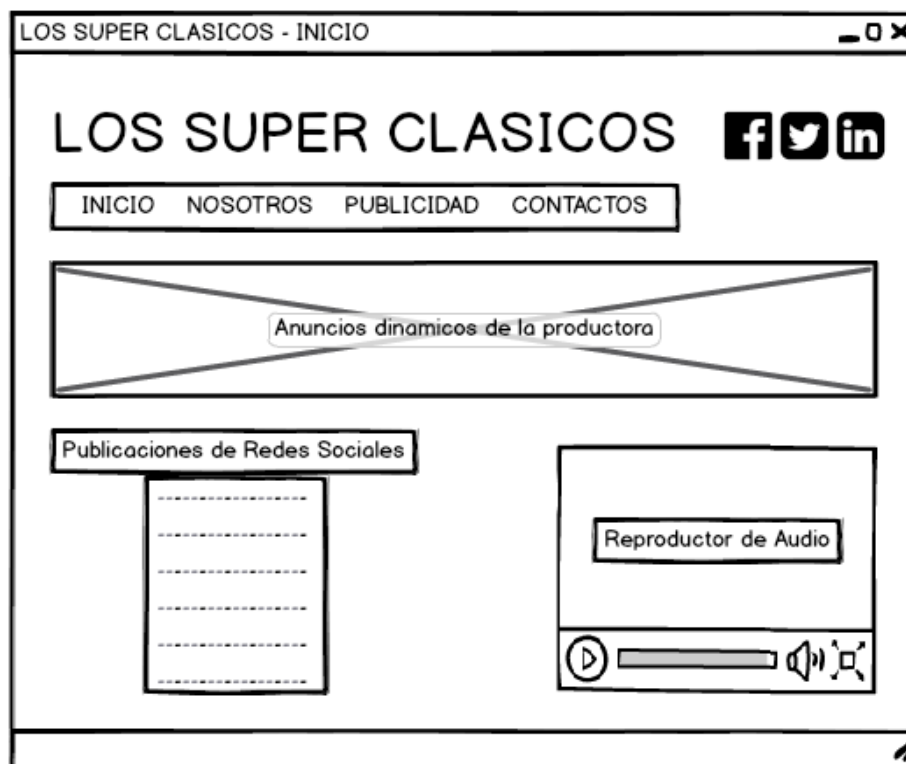


Figura 3.3. Diagrama de presentación Acceso a recursos al usuario final
Fuente: Elaboración propia

3.3.4 SEGUNDA ITERACIÓN XP y WEB ML

A continuación se describe el diagrama de hipertexto y diagrama de presentación para la historia de usuario correspondiente a la segunda iteración.

3.3.4.1 MODELO DE HIPERTEXTO

El siguiente diagrama (ver Figura 3.4) muestra la forma en que se va a desarrollar el módulo de portal de información de la productora que pertenece la historia de usuario 4 y a sus respectivas tareas dentro a lo que corresponde al modelado web.

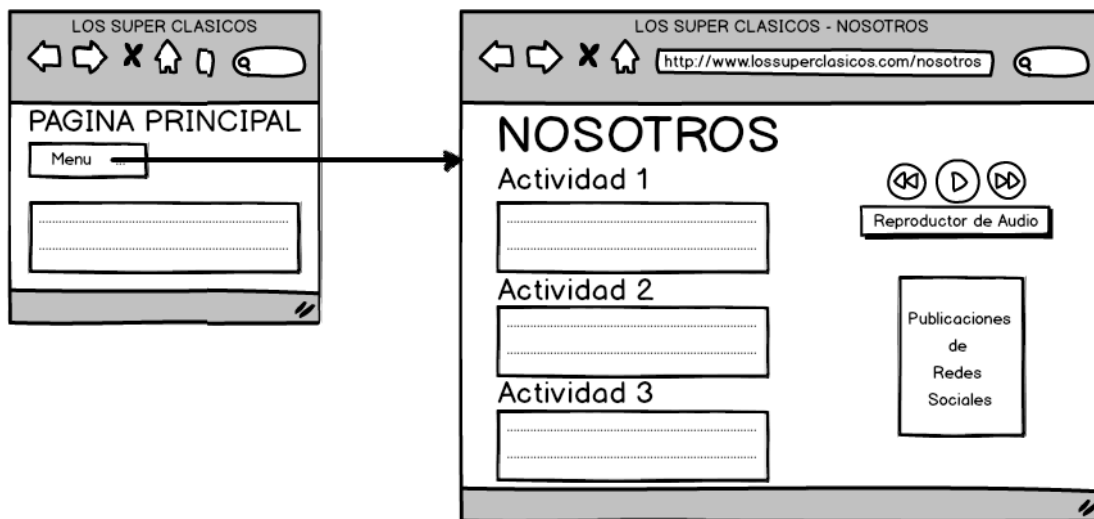


Figura 3.4. Diagrama de hipertexto Portal de información
Fuente: Elaboración propia

3.3.4.2 MODELO DE PRESENTACIÓN

La pantalla de información de la productora (ver Figura 3.5) muestra el desarrollo del módulo de portal de información de la productora.

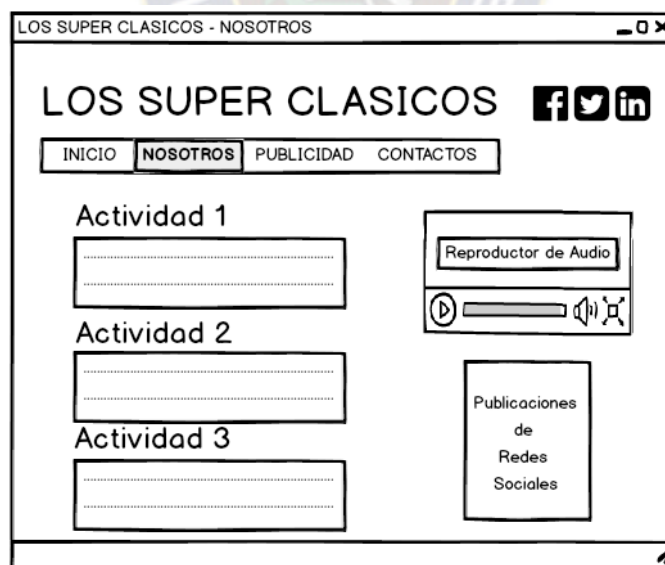


Figura 3.5. Diagrama de presentación Portal de información
Fuente: Elaboración propia

3.3.5 TERCERA ITERACIÓN XP y WEB ML

A continuación se describen los diagramas de hipertexto y diagramas de presentación para las historias de usuarios correspondientes a la tercera iteración.

3.3.5.1 MODELO DE HIPERTEXTO

El siguiente diagrama (ver Figura 3.6) muestra la forma en que se va a desarrollar el módulo del portal de auspiciadores que pertenece la historia de usuario 5 y a sus respectivas tareas dentro a lo que corresponde al modelado web.

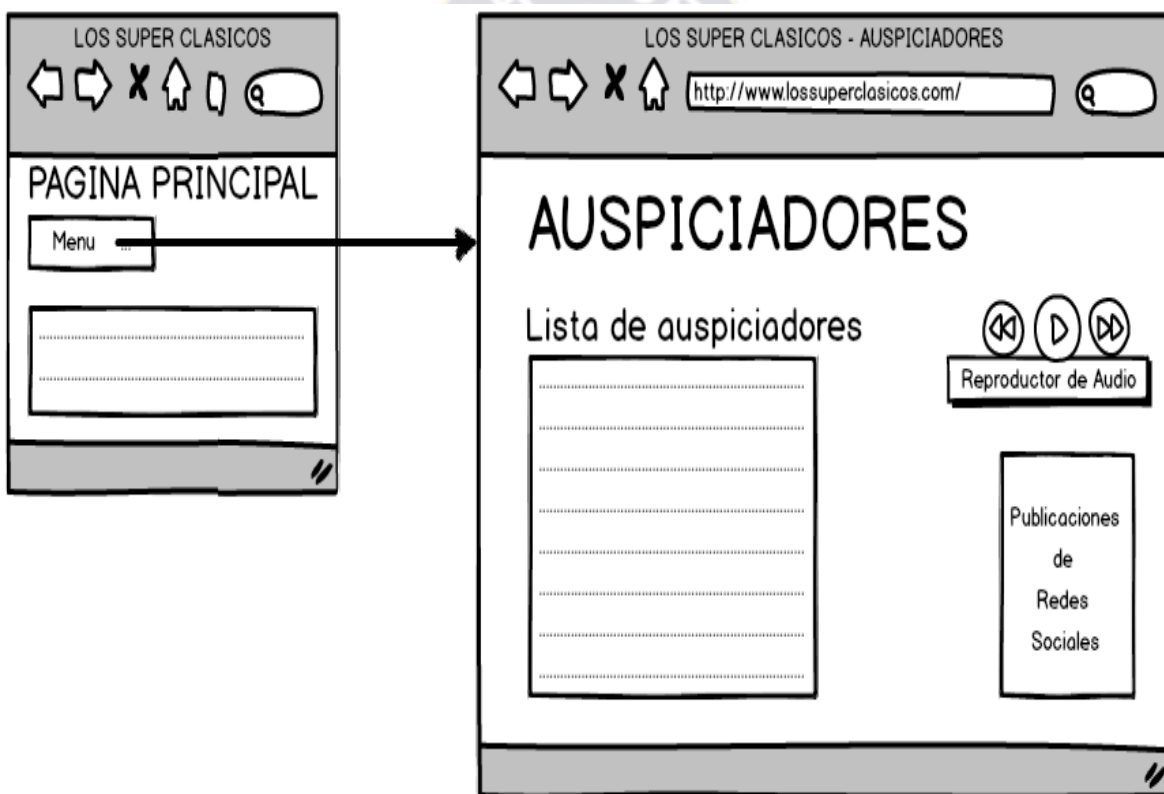


Figura 3.6. Diagrama de hipertexto Portal de auspiciadores
Fuente: Elaboración propia

El siguiente diagrama (ver Figura 3.7) muestra la forma en que se va a desarrollar el módulo del portal de contactos que pertenece la historia de usuario 6 y a sus respectivas tareas dentro a lo que corresponde al modelado web.

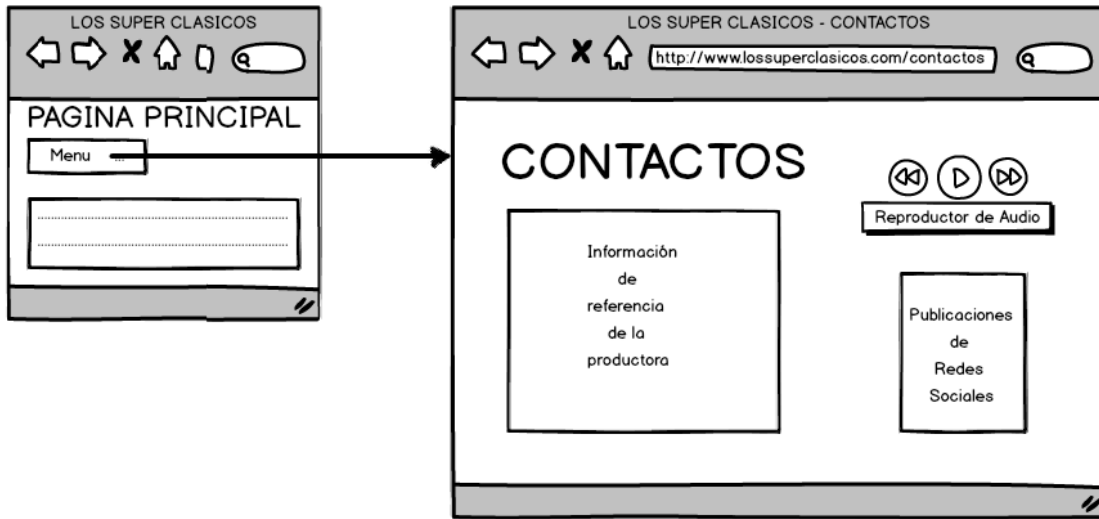


Figura 3.7. Diagrama de hipertexto Portal de contactos
Fuente: Elaboración propia

3.3.5.2 MODELO DE PRESENTACIÓN

La pantalla portal de auspiciadores de la productora (ver Figura 3.8) muestra el desarrollo del módulo de auspiciadores con los cuales trabaja de la productora.



Figura 3.8. Diagrama de presentación Portal de auspiciadores
Fuente: Elaboración propia

La pantalla portal de contactos (ver Figura 3.9) muestra el desarrollo del módulo de contactos donde puede encontrar dirección y ubicación de la productora.



Figura 3.9. Diagrama de presentación Portal de contactos
Fuente: Elaboración propia

3.4 DESARROLLO, PRODUCCIÓN Y ESTABILIZACIÓN

En esta etapa se realiza la programación del Sistema Web y el aplicativo móvil de acorde al plan de entregas realizado en la fase de planificación, Teniendo en cuenta que la dividiremos en 2 partes, la primera para XP y la segunda para Mobile-D respetando la secuencia de iteraciones planteadas anteriormente, pero destacando la particularidad de fases de cada metodología.

3.4.1 DESARROLLO XP

3.4.1.1 PRIMERA ITERACIÓN

La primera iteración describe el desarrollo de la historia de usuario 1, 2 y 3 donde se hace referencia a implementar un servicio a un servidor de streaming, diseñar el sistema web de manera adaptativa y desarrollar el portal de la productora.

a) CAPTURA DE PANTALLAS

El desarrollo de la primera iteración se ve representado por las capturas de pantallas que se muestran a continuación.



Figura 3.10. Pantalla Portal de la Productora vista ordenador de escritorio

Fuente: Elaboración propia

La figura 3.10 muestra la pantalla principal del portal de la productora vista de desde un ordenador de escritorio, correspondiente a la primera iteración.



Figura 3.11. Pantalla Portal de la Productora vista teléfono inteligente
Fuente: Elaboración propia

La figura 3.11 muestra la pantalla principal del portal de la productora vista de desde un teléfono inteligente, enmarcándose que el sistema web tiene un diseño adaptativo a diferentes terminales.

3.4.1.2 SEGUNDA ITERACIÓN

La segunda iteración describe el desarrollo de la historia de usuario 4 donde se hace referencia a desarrollar el portal de información de la productora.

a) CAPTURA DE PANTALLAS

El desarrollo de la segunda iteración se lo vera representado en las capturas de pantallas que se muestran a continuación.



Figura 3.12. Pantalla Portal de información la Productora vista ordenador de escritorio
Fuente: Elaboración propia

La figura 3.12 muestra la pantalla del portal de información de la productora vista de desde un ordenador de escritorio, correspondiente a la segunda iteración.



Figura 3.13. Pantalla Portal de Información la Productora vista teléfono inteligente
Fuente: Elaboración propia

La figura 3.13 muestra la pantalla del portal de información de la productora vista desde un teléfono inteligente, enmarcándose que el sistema tiene un diseño adaptativo a diferentes terminales.

3.4.1.3 TERCERA ITERACIÓN

La tercera iteración describe el desarrollo de las historias de usuarios 5 y 6 donde se hace referencia a desarrollar el portal de auspiciadores de la productora y el portal de contactos de la productora.

a) CAPTURA DE PANTALLAS

El desarrollo de la tercera iteración se lo vera representado por capturas de pantallas que se muestran a continuación.



Figura 3.14. Pantalla Portal de auspiciadores la Productora vista ordenador de escritorio
Fuente: Elaboración propia

La figura 3.14 muestra la pantalla del portal de auspiciadores de la productora vista de desde un ordenador de escritorio, correspondiente a la tercera iteración.

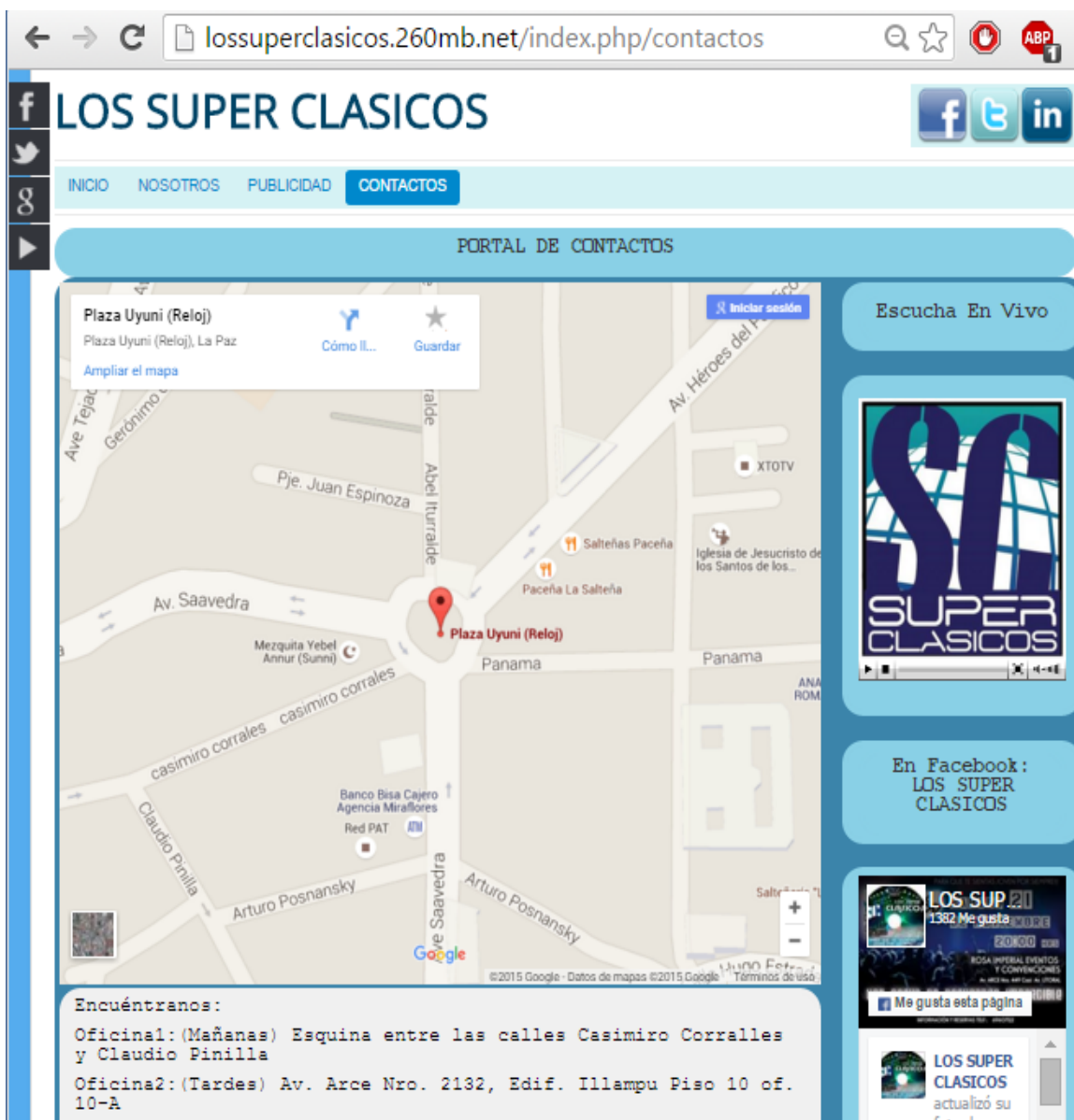


Figura 3.15. Pantalla Portal de contactos de la Productora vista ordenador de escritorio
Fuente: Elaboración propia

La figura 3.15 muestra la pantalla de portal de contactos de la productora vista de desde un ordenador de escritorio, correspondiente a la tercera iteración.

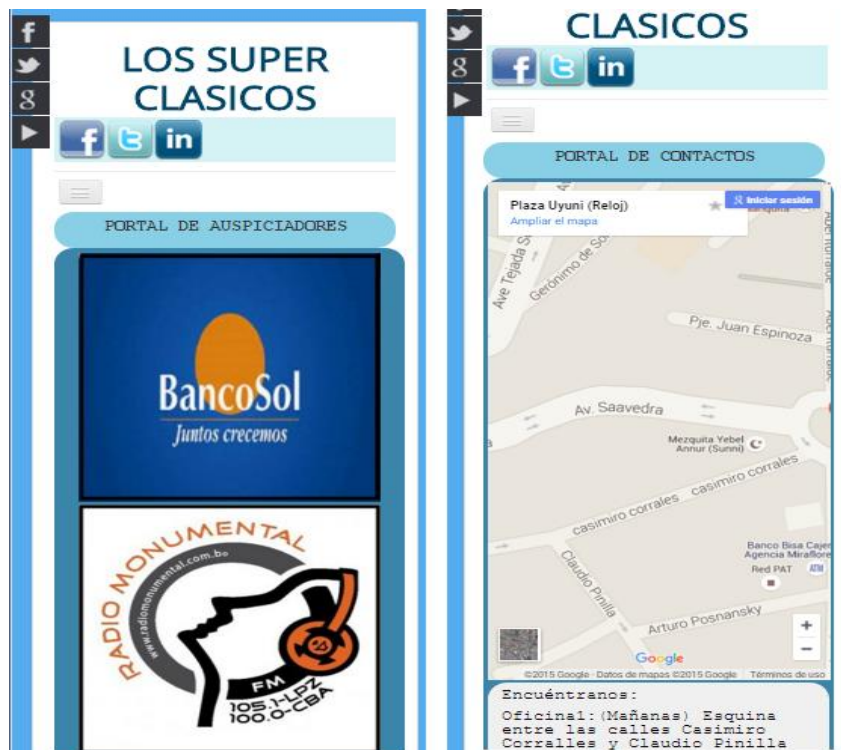


Figura 3.16. Pantalla Portal de Auspiciadores y Contactos vista teléfono inteligente
Fuente: Elaboración propia

La figura 3.16 muestra la pantalla del portal de auspiciadores y contactos de la productora vista desde un teléfono inteligente, enmarcándose que el sistema tiene un diseño adaptativo a diferentes terminales.

3.4.2 PRODUCCIÓN MOBILE-D

En esta fase se describe la última iteración del proyecto donde para la construcción de la aplicación móvil se sigue la regla de la metodología MOBILE-D: “programación en tres días”. A continuación se describe la misma.

3.4.2.1 CUARTA ITERACIÓN PARTE 1

La cuarta iteración primera parte describe el desarrollo de la historia de usuario 7 en su primera parte, más específicamente a la tarjeta de tarea 7.1 que hace referencia al diseño y elaboración de la interfaz de la aplicación móvil.

a) PLANEACIÓN

En tabla 3.47 se describe la planificación de la cuarta iteración primera parte destacándose las características y duración de la misma.

Características	Duración
En esta fase se diseñó y desarrolló la interfaz de la aplicación al usuario final. De acuerdo con los requerimientos, resaltando la insignia de la empresa.	3 días

Tabla 3.47. Planeación cuarta iteración primera parte
Fuente: Elaboración propia

b) CODIFICACIÓN

La interfaz en la aplicación móvil es una actividad en la programación Android, que contiene el diseño y estructura de la presentación la que tiene que ser atractiva visualmente para el usuario final.

c) LIBERACIÓN

La figura 3.17 muestra la captura de pantalla de la interfaz gráfica de la aplicación móvil de acuerdo a la historia de usuario 7 y la tarjeta de tarea 7.1



Figura 3.17. Captura de pantalla Liberación Historia de usuario 7 Tarjeta tarea 1.1
Fuente: Elaboración propia

3.4.2.2 CUARTA ITERACIÓN PARTE 2

La cuarta iteración segunda parte describe el desarrollo de la historia de usuario 7, haciendo referencia a la tarjeta de tarea 7.2 donde se integra de la dirección IP del servidor de audio streaming de la aplicación móvil.

a) PLANEACIÓN

En tabla 3.48 se describe la planificación de la cuarta iteración segunda parte destacándose las características y duración de la misma.

Características	Duración
Luego de haber diseñado la interfaz, se integra la dirección IP del servidor de streaming, que permite escuchar el contenido de audio por medio de la aplicación móvil desarrollada.	3 días

Tabla 3.48. Planeación cuarta iteración segunda parte
Fuente: Elaboración propia

b) CODIFICACIÓN

La integración de la dirección IP será el complemento para la aplicación móvil, la misma nos permitirá escuchar el contenido de audio enviado desde el servidor de streaming. Y con esto terminando el desarrollo del sistema con sus respectivos elementos.

c) LIBERACIÓN

La figura 3.18 muestra la captura de pantalla de la integración de la dirección IP del servidor de streaming reflejándose en el reproductor de audio. De acuerdo a la historia de usuario 7 y la tarjeta de tarea 7.2



Figura 3.18. Captura de pantalla Liberación Historia de usuario 7 tarjeta tarea 1.2
Fuente: Elaboración propia

3.4.3 ESTABILIZACIÓN MOBILE-D

En esta fase se realizaran las últimas acciones de integraciones de la aplicación, quedando el resultado reflejado final en la figura 3.19

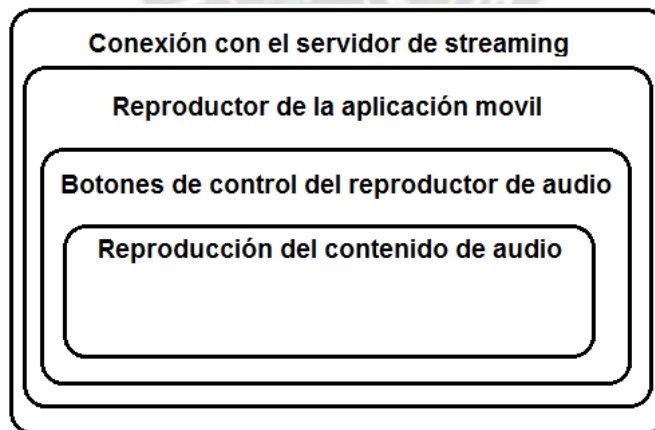


Figura 3.19. Fase de estabilización Integración de acciones del aplicativo
Fuente: Elaboración propia

El diagrama de estabilización de la figura 3.22 representa cada uno de los procesos realizados por la actividad en la aplicación.

Primero se establece conexión con el servidor de streaming, por medio de la dirección IP, posteriormente se genera el contenido. En el aplicativo móvil se tiene un reproductor de audio con sus respectivos botones de control como son: reproducir, pausa y volumen. Una vez oprimido el botón en la pantalla del aplicativo móvil se empieza a reproducir el contenido de audio.

3.5 PRUEBAS

Para esta fase se va a realizar una serie de pruebas hechas a los módulos y el aplicativo móvil ya implementado, empleados por la productora, se utilizara las pruebas de aceptación, que fueron calificadas dentro la organización.

3.5.1 PRUEBAS DE ACEPTACIÓN XP y MOBILE-D

La prueba de aceptación 1 (ver tabla 3.49) comprueba la ejecución de la Historia de Usuario 1 levantar el servidor de audio streaming, que luego de realizar las pruebas correspondientes se pudo obtener el siguiente resultado.

Prueba de Aceptación	
Numero: 1	Historia de Usuario Nro. 1
Nombre: Implementar un servicio de un servidor de audio streaming	
Descripción: El servidor de streaming permite generar el contenido de audio para ser recepcionado por medio de las terminales	
Condiciones de Ejecución: Encendido y ejecución del servidor de streaming, contando con conexión a internet,	
Entrada/Pasos de Ejecución: El locutor genera el contenido de audio, previa configuración del servidor de streaming, este envía el contenido a través de internet.	
Resultado Esperado: El contenido de audio es capturado por el sistema web y la aplicación móvil, para poder ser difundidos a los oyentes.	
Evaluación de Prueba: Aceptada	

Tabla 3.49. Verificar la implementación del servidor streaming
Fuente: Elaboración propia

La prueba de aceptación 2 (ver tabla 3.50) comprueba la ejecución de la Historia de Usuario 2 sistema web con diseño adaptativo, obteniendo el siguiente resultado.

Prueba de Aceptación	
Numero: 2	Historia de Usuario Nro. 2
Nombre: Sistema web con diseño adaptativo	
Descripción: El diseño adaptativo permite adecuar la maquetación del sistema web a cualquier terminal (ordenador de escritorio, laptop, tablet o teléfono inteligente.) sin perder la estructura de diseño en las diferentes terminales.	
Condiciones de Ejecución: El código fuente del diseño adaptativo tiene que estar inmerso en el sistema web previamente desarrollado por el programador. Contando el servidor web en ejecución y conexión a internet.	
Entrada/Pasos de Ejecución: El usuario final ingresando al sistema web desde cualquier dispositivo podrá observar el diseño del sistema que se adapta y va mostrándose de manera agradable al visitador.	
Resultado Esperado: El sistema web desarrollado con un diseño adaptativo se distribuye de manera elegante al vista a las diferentes terminales siendo estos ordenadores de escritorio, laptops, tablets o teléfonos inteligentes.	
Evaluación de Prueba: Aceptada	

Tabla 3.50. Verificación del sistema web con diseño adaptativo
Fuente: Elaboración propia

La prueba de aceptación 3 (ver tabla 3.51) comprueba la ejecución de la Historia de Usuario 3 portal de la productora, obteniendo el siguiente resultado.

Prueba de Aceptación	
Numero: 3	Historia de Usuario Nro. 3
Nombre: Portal de la productora	
Descripción: Es el principal por portal de la productora donde se muestran anuncios, redes sociales y lo más importante el reproductor de audio, por supuesto todo relacionado con la organización,	
Condiciones de Ejecución: Servidor de streaming y servidor web en ejecución, además contando con conexión a internet.	
Entrada/Pasos de Ejecución: El usuario ingresa al sistema web en la primera pantalla se observa el portal de la productora donde se destaca los anuncios que tiene la misma, las redes sociales y entre lo más destacado el reproductor que permite escuchar el contenido de audio planteado por el operador radial.	
Resultado Esperado: Los visitantes y administrador al sistema web tienen el contenido necesario relacionado con la productora.	
Evaluación de Prueba: Aceptada	

Tabla 3.51. Verificación del portal de la productora
Fuente: Elaboración propia

La prueba de aceptación 4 (ver tabla 3.52) comprueba la ejecución de la Historia de Usuario 4 portal de información de la productora, que luego de realizar las pruebas correspondientes se pudo obtener el siguiente resultado.

Prueba de Aceptación	
Numero: 4	Historia de Usuario Nro. 4
Nombre: Portal de información de la productora	
Descripción: En el portal de información se muestran las actividades que realiza la productora relacionada a los medios de comunicación.	
Condiciones de Ejecución: Servidor web en ejecución y conexión a internet.	
Entrada/Pasos de Ejecución: El usuario ingresa al sistema web y selecciona la pestaña “nosotros” ahí accede al portal información de la productora,	
Resultado Esperado: Los visitantes al sistema web conocen las diferentes actividades que realiza la productora.	
Evaluación de Prueba: Aceptada	

Tabla 3.52. Verificación del portal de información

Fuente: Elaboración propia

La prueba de aceptación 5 (ver tabla 3.53) comprueba la ejecución de la Historia de Usuario 5, luego de realizar las pruebas correspondientes se obtiene el siguiente resultado.

Prueba de Aceptación	
Numero: 5	Historia de Usuario Nro. 5
Nombre: Portal de auspiciadores	
Descripción: El portal de auspiciadores muestra las empresas con las que trabaja la productora de las cuales recibe réditos económicos.	
Condiciones de Ejecución: Servidor web en ejecución y conexión a internet	
Entrada/Pasos de Ejecución: El usuario ingresa al sistema web y selecciona la pestaña “auspiciadores” ahí accede al portal publicitario de la productora,	
Resultado Esperado: Los visitantes al sistema web conocen sobre los productos y servicios publicitarios que promociona la productora.	
Evaluación de Prueba: Aceptada	

Tabla 3.53. Verificación del portal de auspiciadores

Fuente: Elaboración propia

La prueba de aceptación 6 (ver tabla 3.54) comprueba la ejecución de la Historia de Usuario 6 portal de contactos, que luego de realizar las pruebas resulta lo siguiente.

Prueba de Aceptación	
Numero: 6	Historia de Usuario Nro. 6
Nombre: Portal de contactos	
Descripción: El portal de contactos muestra la dirección e información importante donde los clientes u oyentes pueden solicitar algún servicio que brinda la productora.	
Condiciones de Ejecución: Servidor web en ejecución y conexión a internet	
Entrada/Pasos de Ejecución: El usuario ingresa al sistema web y selecciona la pestaña “contactos” ahí accede al portal donde se muestra el contenido necesario para que el visitante conozca direcciones y referencias de la productora.	
Resultado Esperado: Los visitantes al sistema web tienen referencia de donde se encuentra ubicada la productora para contactar con algún servicio que presta la empresa.	
Evaluación de Prueba: Aceptada	

Tabla 3.54. Verificación del portal de contactos

Fuente: Elaboración propia

La prueba de aceptación 7 (ver tabla 3.55) comprueba la ejecución de la Historia de Usuario 7 por medio una aplicación móvil, escuchar el contenido de audio del servidor streaming, que luego de realizar las pruebas se pudo obtener el siguiente resultado.

Prueba de Aceptación	
Numero: 7	Historia de Usuario Nro. 7
Nombre: Aplicativo móvil	
Descripción: Aplicación móvil que permite escuchar la transmisión enviada por el servidor streaming.	
Condiciones de Ejecución: Ejecución y transmisión del servidor streaming mas conexión a internet,	
Entrada/Pasos de Ejecución: Descargar el aplicativo del sistema web o tienda de aplicaciones e instalar. Luego de la instalación oprimir el botón de reproducción dentro del reproductor de audio de la aplicación móvil.	
Resultado Esperado: El contenido de audio es capturado por el aplicativo móvil y es difundido por el usuario final.	
Evaluación de Prueba: Aceptada	

Tabla 3.55. Verificar la implementación del aplicativo móvil

Fuente: Elaboración propia

3.5.2 PRUEBAS DE STRESS XP

Las pruebas de stress nos ayudan a encontrar el volumen de datos o de tiempo en que la aplicación comienza a presentar deficiencias también es incapaz de responder a las peticiones. Son pruebas de carga conocidas de otra manera como de rendimiento, pero superando los límites esperados en el ambiente de producción y/o determinados en las pruebas.

Estas pruebas fueron realizadas con el software Webserver stress Tool 7, configurando previamente con 100 clicks por usuario, considerando que como máximo el programa nos permite a 10 clientes. Además accediendo en cada 20 segundos después de un click.

La figura 3.20 muestra el tiempo de conexión de usuarios al sistema web y se observa que el uso de la memoria del sistema puede llegar a 100% de su uso, pero después de un tiempo medido en segundos, el uso de la memoria va disminuyendo y se observa que llega a los 2150[s] aproximadamente, al finalizar una petición el uso de la memoria cae al 0%

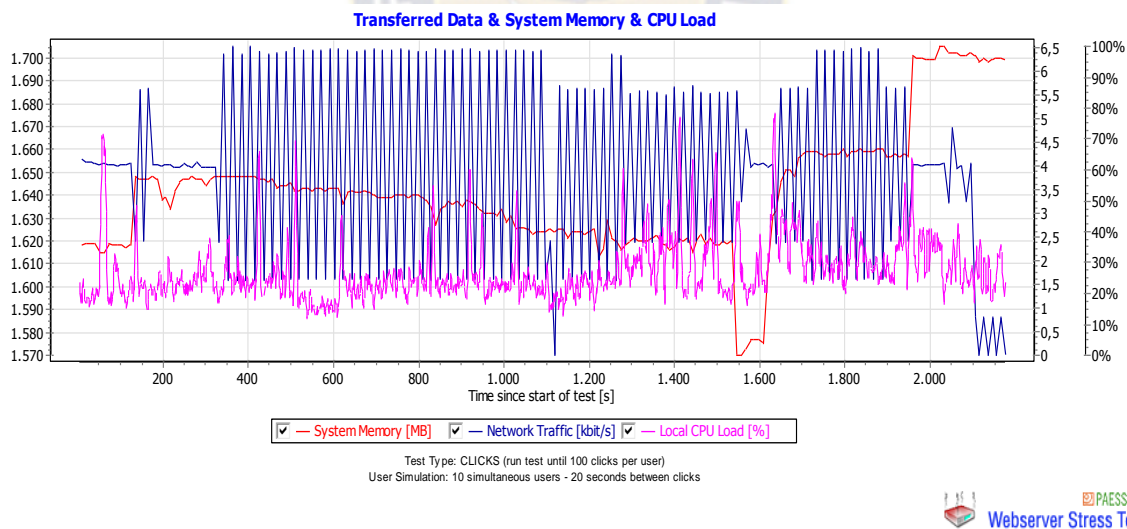


Figura 3.20. Datos transferidos, memoria del sistema y carga de la CPU

Fuente: Elaboración propia

La figura 3.21 nos muestra la transferencia de datos y el tiempo de respuesta del sistema a las peticiones del usuario.

La línea roja nos indica los tiempos de petición del usuario, la línea verde nos indica el tiempo de respuesta del servidor, la línea celeste representa el tiempo de recibo de respuesta del servidor, y la línea azul representa el tráfico de datos de kb/s del servidor. Donde se puede observar que el servidor está por encima respecto al tiempo de acuerdo con el tamaño de la petición del usuario y este se comporta de manera similar.

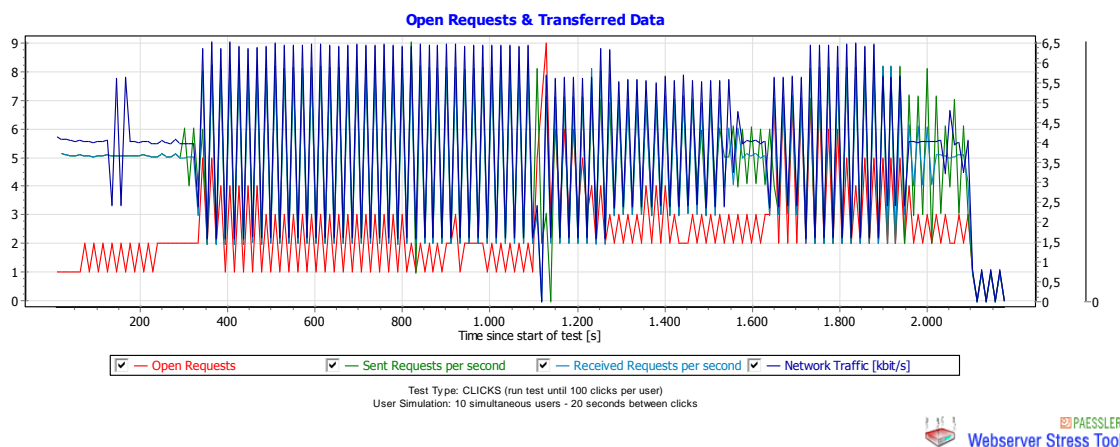


Figura 3.21. Solicitud de apertura y transferencia de datos
Fuente: Elaboración propia

La figura 3.22 muestra las peticiones de usuario hecho al sistema web en milisegundos, donde se observa, un resultado de 1 error en peticiones de página durante los 2000000 [ms].

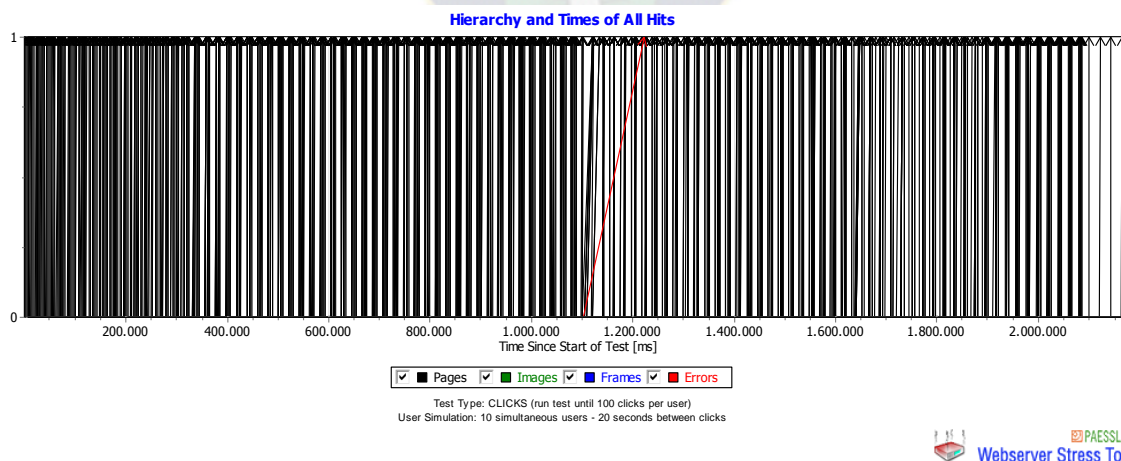


Figura 3.22. Peticiones de usuario
Fuente: Elaboración propia

La figura 3.23 muestra la capacidad máxima de uso simultáneo de usuarios. La línea azul llega al punto de embotellamiento “gridlock”, indicando la capacidad máxima de carga de usuario simultáneamente.

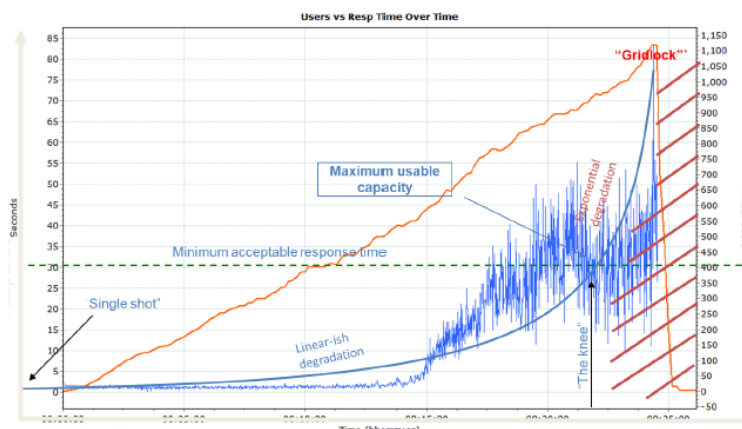


Figura 3.23. Capacidad de carga del sistema
Fuente: Elaboración propia

Por lo tanto se puede concluir que, el servidor es capaz de responder un máximo de 1119 usuarios activos simultáneamente. El tiempo y usuarios mínimos aceptables es de que en 30 segundos con la capacidad de 400 usuarios activos simultáneamente.

3.5.4 PRUEBAS DE FUNCIONAMIENTO MOBILE-D

En la tabla 3.56 se describe los dispositivos móviles en los cuales se llevaron a cabo las pruebas de funcionamiento del aplicativo de la productora, correspondiente a cuarta iteración de desarrollo.

Dispositivo Móvil	Versión de Android
Samsung Galaxy Ace	2.3.6
Samsung Galaxy S3 Mini	4.2.2
Sony Xperia E4g	4.4.4
Tablet Titan ME 1008	4.2.2

Tabla 3.56. Dispositivos móviles utilizados para la prueba de funcionamiento
Fuente: Elaboración propia

CAPÍTULO 4

CALIDAD Y SEGURIDAD

4.1 INTRODUCCIÓN

En este capítulo se hará el análisis posterior al desarrollo e implementación del sistema web y el aplicativo móvil, en este análisis se comprobará la calidad del software mediante un análisis y haciendo uso de uno de los estándares más utilizados y conocidos.

4.2 CALIDAD DE SOFTWARE

La ingeniería del software se diferencia de otras áreas, al no estar basada en leyes cuantitativas básicas, en su lugar se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad de algún producto de software.

Para valorar la calidad de los productos de software o sistemas que se desarrollan se proporcionan información adecuada sobre los datos referentes de la misma a la calidad del producto, permitiendo una visión más profunda sobre el cumplimiento de los objetivos del proyecto.

Medir la calidad de un software determina una de las tareas más complicadas que se presenta en el desarrollo de un sistema. Pero gracias a esta necesidad se fueron creando diferentes formas de medición de las mismas.

A estas las llamamos métricas y entre algunas podemos mencionar las siguientes:

- Modelo de McCall
- Modelo de Boehm
- Modelo ISO 9126

Para nuestro proyecto implementado utilizaremos el modelo de calidad de la ISO 9126, donde se medirá aspectos como la funcionalidad, fiabilidad, usabilidad, portabilidad, eficiencia y mantenibilidad [Pressman, 2002].

4.3 ISO 9126

La norma ISO 9126 (International Standard Organization – Organización Internacional de Normalización) nos ayudara a medir la calidad del sistema siguiendo criterios planteados.

4.3.1 FUNCIONALIDAD

La funcionalidad del sistema está dada por el dominio de la información al cual está asociado un valor de complejidad. Los dominios de información son:

- **Número de entradas de usuario.-** Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación. Las entradas deben ser restringidas de las peticiones que se contabilizan por separado.
- **Número de salidas de usuario.-** La salida se refiere a informes, pantallas, mensajes de error.
- **Número de peticiones de usuarios.-** Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva
- **Número de archivos.-** Se cuenta cada archivo maestro lógico. En otras palabras las tablas existentes en la base de datos.
- **Número de interfaces externas.-** Se cuenta todas las interfaces legibles por el ordenador que son utilizados para transmitir la información.

La funcionalidad es medida a través del punto función (PF), que proporciona una medida objetiva, cuantitativa y auditable del tamaño de la aplicación, basada en la visión del usuario de la aplicación. [Pressman, 2002].

Para calcular el punto función se utiliza la siguiente relación:

$$PF = Cuenta\ Total * (X + Min(Y) * \sum Fi)$$

Dónde:

PF: Medida de funcionalidad

Cuenta Total: Es la suma de los siguientes datos (Nº de entradas, Nº de salidas, Nº de peticiones, Nº de archivos, Nº de interfaces externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%

Min (Y): Error mínimo aceptable al de la complejidad.

$\sum F_i$: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

La tabla 4.1 muestra la cuenta total de los dominios de información establecidos en el sistema web y aplicativo móvil de acuerdo a los parámetros de medición.

PARÁMETRO DE MEDICIÓN	Factor de ponderación				
	CUENTA	SIMPLE	MEDIO	COMPLEJO	TOTAL
Nº de entradas de usuario	4	3	4	6	16
Nº de salidas de usuario	10	4	5	7	50
Nº de peticiones de usuario	14	3	4	6	56
Nº de archivos	4	7	10	15	40
Nº de interfaces externas	6	5	7	10	42
Cuenta Total					204

Tabla 4.1. Factor de ponderación para la funcionalidad
Fuente: Elaboración propia

La ponderación será media, ya que se trata de un sistema de no muy alta complejidad.

FACTOR	VALOR
Sin importancia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Tabla 4.2. Valores de los puntos función
Fuente: Pressman, 2002

Para ajustar la complejidad de los puntos función según el factor de ajuste se asignan valores de acuerdo a los siguientes casos como se ve en la tabla 4.2

La tabla 4.3 muestra los factores de ajustes de complejidad con sus respectivos valores previamente establecidos en la tabla anterior.

FACTOR DE AJUSTE DE VALOR	PESO
1. ¿El sistema requiere respaldo y recuperación confiables?	5
2. ¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?	5
3. ¿Existen funciones de procesamiento distribuidas?	2
4. ¿El desempeño es crucial?	5
5. ¿El sistema correrá en un entorno operativo existente enormemente utilizado?	5
6. ¿El sistema requiere entrada de datos en línea?	5
7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?	4
8. ¿Los ALI se actualizan en línea?	5
9. ¿Las entradas, salidas, archivos o consultas son complejos?	3
10. ¿El procesamiento interno es complejo?	3
11. ¿El código se diseña para ser reutilizable?	5
12. ¿La conversión y la instalación se incluyen en el diseño?	5
13. ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?	3
14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?	5
$\sum F_i$	60

Tabla 4.3. factor de ajuste de valor
Fuente: Pressman, 2010

Para calcular puntos función (PF), usa la siguiente relación: [Pressman, 2010]

$$PF = \text{conteo total} * [0.65 + (0.01 * \sum F_i)]$$

Ahora con los valores obtenidos en las tablas 4.1 y 4.3 se obtiene el siguiente resultado:

$$PF = 204 * [0.65 + (0.01 * 60)]$$

$$PF = 255$$

A continuación calculamos el ajuste, que se lo obtiene de la ecuación anterior pero utilizando los factores de la tabla 4.3 con sus pesos máximos siendo este:

$$\sum Fi = 5 * 14$$

$$\sum Fi = 70$$

De aquí tenemos:

$$PF \text{ ajuste} = 204 * [0.65 + (0.01 * 70)]$$

$$PF \text{ ajuste} = 275.4$$

Con estos resultados podemos ahora si calcular la funcionalidad del sistema, expresándola en porcentaje:

$$\text{Funcionalidad} = (PF / PF \text{ ajuste}) * 100$$

$$\text{Funcionalidad} = (255 / 275.4) * 100$$

$$\text{Funcionalidad} = 92.6 \%$$

Con el resultado obtenido se puede interpretar de la siguiente manera: que de cada 10 personas, 9 consideran que el sistema responde de manera óptima a las funcionalidades requeridas por la productora.

4.3.2 FIABILIDAD

La cantidad de tiempo en el que el software está disponible para usarlo según los siguientes subatributos:

- Nivel de madurez: no fallar a causa de fallas del software.
- Tolerancia de fallos: indica el grado en que el sistema mantiene un nivel de respuesta ante fallos del sistema o interfaces.
- Recuperabilidad: indica la capacidad que tiene el sistema para restablecer su nivel de respuesta después de un fallo crítico o error del hardware.

La fiabilidad en términos estadísticos como “la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico”.

[Pressman, 2002].

La medida de fiabilidad es el tiempo entre fallos y está dado por:

$$\text{TMEF} = \text{TMDF} + \text{TMDR}$$

Dónde: TMEF, es el tiempo medido entre fallos.

TMDF, es el tiempo medio de fallos.

TMDR, es el tiempo de reparación.

Además de una medida de fiabilidad, debemos calcular la medida de disponibilidad del software de la siguiente forma:

$$\text{Disponibilidad} = [\text{TMDF} / (\text{TMDF} + \text{TMDR})] \times 100\%$$

Para hacer la prueba de fiabilidad calculamos lo siguiente:

$$\text{TMEF} = 24 \text{ hrs.} + 1 \text{ hrs.} = 25 \text{ hrs.}$$

Entonces la disponibilidad es:

$$\text{Disponibilidad} = [24 \text{ hrs.} / (24 \text{ hrs.} + 1 \text{ hrs.})]$$

$$\text{Disponibilidad} = 0.96 * 100\% = 96\%$$

Con el resultado obtenido se puede interpretar: que el sistema tiene la capacidad de ser utilizado libre de errores en promedio 96 de cada 100 veces.

4.3.3 USABILIDAD

La usabilidad representa facilidad de uso que el usuario final percibirá del sistema. Esta métrica nos muestra el esfuerzo necesario para aprender a manipular el sistema.

Se obtiene a través de los datos obtenidos en una pequeña encuesta que se realizó directamente a un grupo determinado de usuarios finales.

La tabla 4.4 muestra el modelo de las preguntas para realizar la encuesta a los usuarios finales.

Nº	PREGUNTA	VALOR
1	¿El sistema satisface el contenido requerido por el usuario final?	
2	¿Las salidas del sistema están de acuerdo al gusto del usuario final?	
3	¿Cómo considera el ingreso de información al sistema?	
4	¿Cómo considera las vistas adaptativas del sistema web?	
5	¿Cómo considera la manipulación del aplicativo móvil?	
TOTAL		

Tabla 4.4. Preguntas facilidad de uso

Fuente: Elaboración propia

La tabla 4.5 muestra la escala de ponderación para responder a la encuesta.

DESCRIPCIÓN	VALOR
Pésimo	1
Malo	2
Regular	3
Optimo	4
Muy Bueno	5

Tabla 4.5. Escala de ponderación para usabilidad

Fuente: Elaboración propia

La tabla 4.6 muestra los resultados de la encuesta que se realizó a 5 usuarios finales encabezada por el gerente de la productora.

PREGUNTA	USUARIO 1	USUARIO 2	USUARIO 3	USUARIO 4	USUARIO 5	PROMEDIO
1	4	5	5	5	5	4.8
2	5	4	5	5	4	4.6
3	4	5	5	5	5	4.8
4	5	5	4	4	5	4.6
5	4	5	5	5	5	4.8
Z Xi						23.6

Tabla 4.6 . Resultados de la encuesta de usabilidad

Fuente: Elaboración propia

Con el resultado obtenido, considerando a “n=5” y utilizando la siguiente ecuación tendremos:

$$\text{Usabilidad} = [(Zx / n) * 100\%] / n$$

$$\text{Usabilidad} = [(23.6 / 5) * 100\%] / 5$$

$$\text{Usabilidad} = [4.72 * 100\%] / 5$$

$$\text{Usabilidad} = [472\%] / 5$$

$$\text{Usabilidad} = 94.4 \%$$

De acuerdo al porcentaje obtenido se interpreta de la siguiente manera: en promedio 9 de cada 10 usuarios consideran que tienen facilidad manipular el sistema.

4.3.4 MANTENIBILIDAD

Para la evaluación de la mantenibilidad, se desarrolló preguntas estas preguntas son valoradas en porcentaje por el desarrollador del sistema al momento de la culminación del proyecto.

Este valor tiene consideración por la experiencia y la forma de trabajo de cada programador, el mismo puede ser relativo respecto a otros desarrolladores.

La tabla 4.7 muestra las preguntas y resultados obtenidos de la evaluación mantenibilidad.

FACTOR DE AJUSTE	VALOR
¿Puede ser modificado el sistema?	96 %
¿Deja identificar las partes que deben ser modificadas?	97 %
¿Permite implementar una modificación específica?	96 %
¿Presenta efectos inesperados como posibles errores?	96 %
TOTAL	96.25 %

Tabla 4.7. Evaluación de mantenibilidad
Fuente: Elaboración propia

Tras el resultado obtenido se concluye:

$$\text{Mantenibilidad} = 96.25 \%$$

Con el resultado encontrado se interpreta de la siguiente manera: el esfuerzo necesario para realizar mantenimiento al sistema es mínimo.

4.3.5 PORTABILIDAD

La portabilidad es una característica que indica la capacidad del sistema de funcionar correctamente independientemente de la plataforma tecnológica sobre la cual opera.

Para la estimación, se aplica la siguiente ecuación:

$$\text{Portabilidad} = 1 - (\# \text{ días para portar el sistema} / \# \text{ días para implementar el sistema})$$

Reemplazando valores se tiene:

$$\text{Portabilidad} = 1 - (1 / 30)$$

$$\text{Portabilidad} = 0.966 * 100\%$$

$$\text{Portabilidad} = 96.6\%$$

Se traduce como la capacidad del mismo para funcionar en un 96.6% en la totalidad de las plataformas tecnológicas.

4.3.6 EFICIENCIA

El grado en que el software emplea los recursos del sistema, bajo los siguientes atributos:

- Comportamiento temporal: características que influyen en el tiempo de respuesta, proceso y productividad cuando se ejecuta su función.
- Utilización de recursos: usar cantidades y tipos de recursos apropiados, en condiciones especificadas.

En tabla 4.8 se puede observar el cuestionario empleado para obtener la eficiencia del sistema:

Nro.	Factor de ajuste	Valor obtenido i (0-100)
1	¿Es de respuesta rápida al utilizar sus funciones?	97
2	¿Tiene rendimiento de acuerdo a los factores que utiliza?	95
3	¿Responde adecuadamente cuándo utiliza sus funciones?	98
4	¿El tiempo de respuesta a sus consultas es adecuado?	97
Eficiencia		96.8

Tabla 4.8. Cuestionario de factor de eficiencia

Fuente: Yali, 2011

La eficiencia es del 96.8%, lo que significa que el sistema emplea los recursos de manera óptima. Con lo que de cada 10 peticiones al sistema 9 son de manera adecuada y rápida.

4.4 CALIDAD GLOBAL

Para poder obtener la calidad global del sistema, se saca la media de todas las medidas expresadas en porcentaje hasta el momento. La tabla 4.9 muestra la calidad global del sistema expresada en porcentajes.

CRITERIOS	RESULTADO
Funcionalidad	92.6 %
Fiabilidad	96 %
Usabilidad	94.4 %
Mantenibilidad	96.3 %
Portabilidad	96.6%
Eficiencia	96.8%
Calidad Global	95.5%

Tabla 4.9. Calidad global

Fuente: Elaboración propia

Con este resultado concluimos que: de 95 usuarios de cada 100 consideran que sistema web y aplicativo móvil de calidad.

4.5 SEGURIDAD

La seguridad es uno de los aspectos más importantes, siendo el software desarrollado para la web y tomando en cuenta que los datos almacenados son de mucha importancia para la organización.

Por lo tanto las políticas y estrategias de control que se consideran con las siguientes:

- Políticas de control de acceso al Sistema Web.
- Políticas de respaldo de la base de datos.
- Políticas de registro de los eventos realizados por los usuarios.

4.5.1 POLÍTICAS DE CONTROL DE ACCESO AL SISTEMA WEB

- El acceso al sistema es controlado mediante el factor doble de seguridad que posee Joomla mediante unos de sus componentes. Ingresando así el usuario, la contraseña y la palabra clave definida por el administrador.
- La encriptación se realiza mediante la función MD5 utilizado en la librería de “encriptación”, desarrollado para la encriptación de contraseñas de los usuarios.
- Se hace el control de sesiones para que usuarios no autenticados de esta forma no puedan ingresar al sistema.
- El control de tiempo inactivo de la sesión en el sistema web, permite cerrar de manera automática la cuenta del usuario, que por cualquier motivo no lo haya hecho.

4.5.2 POLÍTICAS DE RESPALDO A LA BASE DE DATOS

- El acceso a la base datos se encuentra configurada con una sesión previamente registrada para manipulación.
- Los archivos de configuración y acceso a la base de datos del gestor de contenidos Joomla de acuerdo a su factor doble seguridad hace que las conexiones sean seguras.

- El respaldo de la base de datos es muy importante por lo cual se realiza el backups o resguardo para la seguridad de la información.
- Implementación en el sistema web del componente RSFirewall de Joomla que tiene un escáner activo estando alerta de un intruso a la base datos como una inyección SQL y bloqueando al mismo.
- Si se intenta deliberadamente ingresar a la administración del sistema, se activan captchas evitando así ataques por robots.
- Escáner RSFirewall para cuantificar número de intentos determinado para falsear la seguridad del sitio, automáticamente la IP de donde provienen los ataques quede bloqueada dejando así al intruso inutilizado.

4.5.3 POLÍTICAS DE REGISTROS DE EVENTOS

Los logs o eventos realizados por el administrador del sistema web permite el seguimiento y monitoreo de los accesos de los usuarios a las diferentes funciones que ofrece el sistema como: Ingresos al sistema, actualizaciones o eliminaciones que el usuario haya realizado, las peticiones al sistema.

Para el buen uso del sistema web se recomienda lo siguiente:

- Se recomienda al administrador principal del Sistema Web, asignar o aumentar ciertos privilegios, solo a usuarios capacitados para el manejo el mismo.
- Se recomienda a los usuarios el cambio periódico de sus contraseñas para la seguridad de sus cuentas y principalmente del sistema.
- Se recomienda realizar el mantenimiento preventivo y correctivo del sistema para su buen funcionamiento y prevención de fallas.

CAPÍTULO 5

COSTO/BENEFICIO

5.1 INTRODUCCIÓN

El objetivo del capítulo es demostrar a los miembros de la productora que con la implementación y utilización del sistema se obtendrá muchos beneficios que los esperados.

Para tal efecto se usó de ciertas herramientas y heurísticas que nos ayudaran a calcular en VAN (Valor Actual Neto), C/B (Costo Beneficio) y el TIR (Tasa Interna de Retorno).

Para poder alimentar el cálculo de VAN se hará uso de COCOMO II que es una herramienta que nos ayudara a estimar el costo del sistema basado en el tamaño del mismo y otras características.

Después de realizar los cálculos necesarios para la obtención de los resultados esperados estaremos en la capacidad de afirmar si el proyecto es viable, redituable y comprobar que es buena opción invertir en el proyecto.

5.2 COCOMO II

El Modelo Constructivo de Costes (COCOMO) es un modelo matemático de base empirica, utilizando para la estimación de costes de software.

Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro “Software Engineering Economics”.

COCOMO II consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones:

$$E = a (\text{KLDC})^b, \text{ persona-mes}$$

$$D = c (E)^d, \text{ meses}$$

$$P = E / D, \text{ personas}$$

Dónde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código, en miles.

A la vez cada modelo se subdivide en modos, los mismos son:

- **Modo orgánico:** Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- **Modo semi – libre o semi acoplado:** Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

La tabla 5.1 muestra los coeficientes del proyecto de software de acuerdo a los tres modos expuestos anteriormente.

Proyecto de Software	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semi-Acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 5.1. Coeficientes: a, b, c y d COCOMO II
Fuente: Pressman, 2002

5.3 COSTO DEL SISTEMA

El costo del sistema se lo planteara en tres partes: desarrollo de software, implementación y elaboración del proyecto.

5.3.1 COSTO DE DESARROLLO DEL SOFTWARE

Para el cálculo del desarrollo del software se tendrá como partida el punto función no ajustado valor ya encontrado en el capítulo anterior. Recordando es el siguiente:

$$PF = 204$$

Este resultado se debe convertir a KLDC (Kilos de Líneas de Código), para ello se utiliza la siguiente la tabla 5.2

Lenguaje	Nivel	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Tabla 5.2. Conversión de puntos función a KLDC
Fuente: Pressman, 2002

Como el desarrollo del sistema está basado en PHP y JAVA realizaremos una media entre estos 2 lenguajes y de acuerdo a la tabla 5.2 se tiene:

$$\text{Factor LDC/PF} = (29 + 53) / 2$$

$$\text{Factor LDC/PF} = 41$$

Con el factor LDC/PF = 41 y reemplazando este dato más el punto función no ajustado se calcula en la siguiente ecuación:

$$\text{LDC} = \text{PF} * \text{Factor LDC/PF}$$

$$\text{LDC} = 204 * 41$$

$$\text{LDC} = 8364$$

Para convertirlo a KLDC dividimos LDC entre 1000

$$\text{KLDC} = \text{LDC} / 1000$$

$$\text{KLDC} = 8364 / 1000$$

$$\text{KLDC} = 8.36$$

A continuación haremos el cálculo del esfuerzo necesario para la programación del sistema, para ello utilizamos la siguiente ecuación:

$$E = a (\text{KLDC})^b$$

Para hallar el esfuerzo “E” definimos antes el tipo del proyecto que en nuestro caso es orgánico y utilizamos de los datos de la tabla 5.1. Con esto se reemplaza en la fórmula:

$$E = a (\text{KLDC})^b$$

$$E = 2.4 (8.36)^{1.05}$$

$$E = 22.31 \text{ persona-mes}$$

Ahora para hallar el tiempo del proyecto usamos los datos de la tabla 5.1, recordando que el proyecto es de tipo orgánico y reemplazando en la siguiente formula:

$$D = c (E)^d \text{ meses}$$

$$D = 2.5 (22.31)^{0.38}$$

$$D = 4.61 \approx 5 \text{ meses}$$

De aquí concluimos que el proyecto deberá tener un desarrollo de 5 meses.

Para calcular la cantidad en número de programadores se utiliza la siguiente formula, reemplazando los datos ya encontrados:

$$P = E / D$$

$$P = 22.31 / 5$$

$$P = 4.46 \approx 4 \text{ programadores}$$

El salario promedio de un programador oscila entre los 212 a 312 \$us, en nuestro caso tomaremos un promedio de 250\$us. A partir de este monto podemos calcular el costo total del desarrollo del software.

$$\text{Costo mensual de desarrollo} = \text{Nro. de programadores} * \text{Salario promedio}$$

$$\text{Costo mensual de desarrollo} = 4 * 262.5$$

$$\text{Costo mensual de desarrollo} = 1050 \text{ $us}$$

Como el desarrollo de software se lo estima en 5 meses resultado que se lo vio anteriormente, se tiene:

$$\text{Costo total del desarrollo} = \text{Costo mensual de desarrollo} * \text{Nro. de meses}$$

$$\text{Costo total del desarrollo} = 1050 * 5$$

$$\text{Costo total del desarrollo} = 5250 \text{ $us}$$

5.3.2 COSTO DE IMPLEMENTACIÓN

La productora cuenta con un área de sistemas por lo cual cuentan con una red interna funcional y con servicio de internet. Por lo tanto el único costo de implementación que se tendrá será la configuración de la parte del servidor. El mismo tendrá un costo de 100Sus.

5.3.3 COSTO DE ELABORACIÓN DEL PROYECTO

Los costos de elaboración del proyecto se refieren principalmente a los gastos que se realizan a lo largo de las diferentes fases de la metodología XP y Mobile-D Estas las podemos ver expresadas en la tabla 5.3

Detalle	Importe (\$us)
Análisis y diseño del proyecto	200
Material de escritorio	30
Internet	50
Otros	20
Total	300

Tabla 5.3. Costo de elaboración del proyecto
Fuente: Elaboración propia

5.3.4 COSTO TOTAL DEL SOFTWARE

El costo total del software se lo obtiene de la sumatoria del costo de: desarrollo, implementación y elaboración del proyecto. La tabla 5.4 expresa estos resultados.

Detalle	Importe (\$us)
Costo de desarrollo	5250
Costo de implementación	100
Costo de elaboración del proyecto	300
Total	5650

Tabla 5.4. Costo total del software
Fuente: Elaboración propia

5.4 VALOR ACTUAL NETO

El VAN o valor actual neto es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto.

La fórmula que utilizaremos para hallar el valor actual neto será:

$$VAN = \sum \frac{Ganancias}{(1+k)^n} - \sum \frac{Costos}{(1+k)^n}$$

Dónde:

VAN: Valor Actual Neto

Ganancias: Ingreso de flujo anual

Costos: Salidas de flujo anual

n: Numero de periodo

k: Tasa de descuento o tasa de interés al préstamo

Los gastos y ganancias que se estiman en un lapso de 4 años los mostramos en la tabla 5.5, para este caso en particular utilizamos una de descuento del 12% ya que es la tasa actual de interés del préstamo en las entidades financieras.

Año	Costos	Ganancias	Costos/(1+i) ⁿ	Ganancias/(1+i) ⁿ	Resultado
1	5650	0	5044.64	0	
2	1100	2000	876.9	1594.4	717.5
3	620	3500	441.3	2491.2	2049.9
4	300	5000	190.7	3177.6	2986.9
Σ	7670	10500	6553.5	7263.2	
$VAN = \sum \frac{Ganancias}{(1+0.12)^n} - \sum \frac{Costos}{(1+0.12)^n}$					709.7

Tabla 5.5. Calculo del VAN
Fuente: Elaboración propia

La tabla 5.6 muestra si un proyecto es rentable y de acuerdo a ciertos criterios más el valor del VAN concluiremos si es rentable o no.

Valor del VAN	Interpretación
$VAN > 0$	El proyecto es rentable
$VAN = 0$	El proyecto también es rentable, ya que se incorpora la ganancia de la tasa de interés.
$VAN < 0$	El proyecto no es rentable.

Tabla 5.6. Criterio de interpretación del VAN
Fuente: Elaboración propia

De aquí concluimos: considerando que el $VAN = 709.7 \approx 709$ y siguiendo los criterios de la tabla 5.6 se afirma que nuestro proyecto es rentable ya que 709 es mayor a 0.

5.4.1 COSTO / BENEFICIO

Para hallar el costo/beneficio de un proyecto se aplica la siguiente ecuación:

$$\text{Costo/Beneficio} = \frac{\sum \text{Ganancias}}{\sum \text{Costos}}$$

De aquí, reemplazando en la ecuación anterior los valores conocidos de la tabla 5.5

$$\text{Costo/Beneficio} = 10500 / 7670$$

$$\text{Costo/Beneficio} = 1.4 \text{ \$us}$$

Con este resultado interpretamos de la siguiente manera: por cada dólar invertido en el proyecto de software, la institución genera una ganancia de 0.4 \$us.

5.5 TASA INTERNA DE RETORNO

Cuando en la fórmula del VAN el valor de “k” es igual a “0” pasa a llamarse TIR (Tasa Interna de Retorno). La TIR es la rentabilidad que nos proporciona al proyecto.

La ecuación que utilizaremos es la siguiente:

$$TIR = \sum \frac{Ganancias - Costos}{(1 - i)^n}$$

Dónde:

TIR: Tasa Interna de Retorno

Ganancias: Flujo de entrada de un periodo

Costos: Flujo de salida de un periodo

i: Tasa de interés al ahorro

n: Numero de periodo

La tabla 5.7 muestra una mejor comprensión de ecuación TIR y expresando los resultado encontrados en las misma.

Año	Costos	Ganancias	$\frac{Ganancias - Costos}{(1 - i)^n}$
1	5650	0	- 6420.5
2	1100	2000	1162.2
3	620	3500	4226.1
4	300	5000	7837.3
TIR			6805.1

Tabla 5.7. Calculo de la tasa interna de retorno

Fuente: Elaboración propia

Con esta información se concluye que el proyecto nos dará una rentabilidad de 6805.1 \$us durante los 4 años de inversión.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 INTRODUCCIÓN

En el último capítulo se realizara un análisis completo acerca de los resultados obtenidos con la implementación del Sistema Web de Gestión de Audio Streaming Orientado a Diseños Adaptativos. Acerca de la evaluación que se realiza al sistema, así como también las visiones que se puedan a llegar a tener para un futuro dentro de la institución.

El análisis a realizar será también al documento de investigación presente, considerando elementos como metodologías, diseño y también se dará recomendaciones a futuras investigaciones similares.

6.2 CONCLUSIONES

- Se desarrolló el Sistema Web de Gestión de Audio Streaming Orientado a Diseños Adaptativos, en su plenitud con todos los requerimientos planteados por la Institución, satisfaciendo así las necesidades definidas en un principio.
- Se implementó un servicio de audio streaming donde se contempló los requerimientos físicos y lógicos de la empresa.
- Se desarrolló el sistema web dinámico e interactivo que permite gestionar los contenidos de la productora.
- Se contempló en el desarrollo del sistema web un diseño adaptativo, visualizando de forma elegante en los diferentes terminales.
- Se desarrolló un aplicativo móvil que complementa al sistema web de gestión de audio streaming.

- Se verificó el desempeño: con la estabilidad de información en el sistema web y la fidelidad del contenido de audio streaming en el aplicativo móvil.

6.3 RECOMENDACIONES

- Desarrollar nuevos sistemas dentro de la empresa para las diferentes necesidades como ser: control de procesos administrativos, control de procesos administrativos, entre otros.
- Realizar un mantenimiento constante de los contenidos: mediante backups periódicos debido a la importancia de la información que mantiene el sistema web.
- Extender nuevas funcionalidades al sistema si se necesitan nuevos requerimientos por parte de la productora.
- Desarrollar sistemas información con nuevas metodologías que van manifestándose debido a su versatilidad e innovación ayudando al desarrollo de las nuevas exigencias del mercado.
- Al momento de desarrollar sistemas optar por los sistemas web y móviles debido a que hoy en día el uso de internet es gran ayuda para los usuarios independientes como las organizaciones.

BIBLIOGRAFÍA

ABRAHAMSSON. “Agile software development of mobile information systems”. Proceedings of the 19th international conference on Advanced information systems engineering. 2007. p1-4.

ABRAHAMSSON & HANHINEVA & HULKKO & IHME & JÄÄLINOJA & KORKALA & KOSKELA & KYLLÖNEN & SALO “Mobile-D: an agile approach for mobile application development”. Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications. 2004.

BARESI L. Garzotto Paolini “Extendido UML para el modelado Web Modeling Web Applications” Pittsburgh USA 2001.

BAUER F. “Ingeniería del software” Garmisch Alemania 1968. p205.

BECK K. “Extreme Programming Explained”. Pearson Education. Addison Wesley 2000.

BECK K. “Una explicación de la Programación extrema: aceptar el cambio” Madrid, Addison-Wesley. 2002. p189.

BRAMBILLA & MANOLESCU “Diseño de Aplicaciones Web” Publicado en San Francisco USA Ediciones Morgan Kaufmann 2002.

CERI Stefano, Piero Fraternali “Lenguaje de Modelado Web (Web ML): Un lenguaje para diseñar sitios Web”. San Francisco, Morgan Kaufmann Publishers. 2000. p216

CESAR F. & ACEBAL “Programacion Extrema (XP): un nuevo método de desarrollo de software. Dpto. de Informática. Area de Lenguajes y Sistemas Informáticos”. 2002.

ECHEVERRY T. Delgado C. Luz E. Caso práctico de la metodología XP. Al desarrollo de software. (Tesis de ingeniería). Universidad tecnológica de Pereira. 2007.

FILIP Chereches Tosa, Bogdan Brinzarea, Cristian Darie. “Construyendo Aplicaciones Web Interactivas”. España. Packt Publishing, 2008, p352.

FLING “Mobile Design and Development” O’Reilly, ISBN: 978-0-596-15544-5. 2009.

FLORES Rios Carla Julieta, 2013: Sistema Web de Seguimiento y Control de Proceso Judiciales Caso Ministerio de Educación

HANS VAN VLIET “Software Engineering. Principles and Practice”. 3ra edición, 2002.

IEEE Computer Society “Standard Glossary of Software Engineering Terminology” 2003.

INTECO “Ingeniería de software: Metodologías y ciclos de vida” Madrid: Instituto Nacional de Tecnologías de la Comunicación 2009.

JEFFRIES R. & ANDERSON “Addison – Wesley Programación Extrema Instalada” 2001.

KORTH Herry, Silverschatz Abraham & Sudarshan “Fundamentos de Base de Datos” 5ta edición. España McGraw-Hill. 2002. 600p

MATT Stephens, Doug Rosenberg, “Extreme Programming Refactored: The Case Against XP”, USA, APress, 1970. p432.

MURUGESAN & DESHP & HANSEN & GINIGE “Web Engineering: A New Discipline for Development of Web-Based Systems”. 2001.

NEWKIRK J. & MARTIN R. C “Practicas de Programacion Extrema” USA 2001.

PASTOR Oscar “Metodologias de Diseño de Aplicaciones Web. España. 2008.

PEÑA & GUTIERREZ “Ingeniería para los sistemas web: Una propuesta metodológica” Instituto Politécnico Nacional – Centro De Investigación En Computación 2004.

PRESSMAN “Ingeniería del Software un enfoque práctico”. McGraw-Hill 3ra Ed. 2002

PRESSMAN “Software Engineering: A practitioner`s perspective”. McGraw-Hill. 2004.

PRESSMAN “Ingeniería del Software un enfoque práctico”. McGraw-Hill. 7ma Ed. 2010.

RAMÍREZ Vique Robert “Métodos para el desarrollo de aplicaciones móviles” Universidad Abierta de Cataluña. Editorial UOC. 2013.

SCOTT N. Ambler “Agile Modeling: Effective Practices for Extreme Programming the unified Process”, United States of America, John Wiley & Sons, 2002. p404.

STEFANO Ceri, Piero Fraternali. “Lenguaje de Modelado Web (WebML): Un lenguaje para diseñar sitios Web”. San Francisco, Morgan Kaufman. 2000. p216.

VARGAS Agudelo Fabio Alberto “La Ingeniería de Software en el desarrollo de aplicaciones para dispositivos móviles”. 2011.

WILLIAM “Exploración de Programación Extrema”. Addison-Wesley. USA. 2002.

YALI, F. “Sistema de control y evaluación para el proyecto C.D.U”. La Paz, UMSA. 2011

ZAPATA & VARGAS “Una revisión de la literatura en consistencia entre problemas y objetivos en ingeniería de software y gerencia organizacional”. Revista EIA. 2009.

REFERENCIAS DE INTERNET

Lacey-Bordeaux Emma. (2015). “Noruega será el primer país del mundo en eliminar la radio FM”. New York - EEUU. CNN. Recuperado de:
<http://cnnespanol.cnn.com/2015/04/21/noruega-sera-el-primer-pais-del-mundo-en-eliminar-la-radio-fm/#0>

OPHELIA PASTRANA (2014).”5 beneficios de aplicar metodologías ágiles en el desarrollo de software”. Recuperado de: <http://www.i2btech.com/blog-i2b/tech-deployment/5-beneficios-de-aplicar-metodologias-agiles-en-el-desarrollo-de-software/>

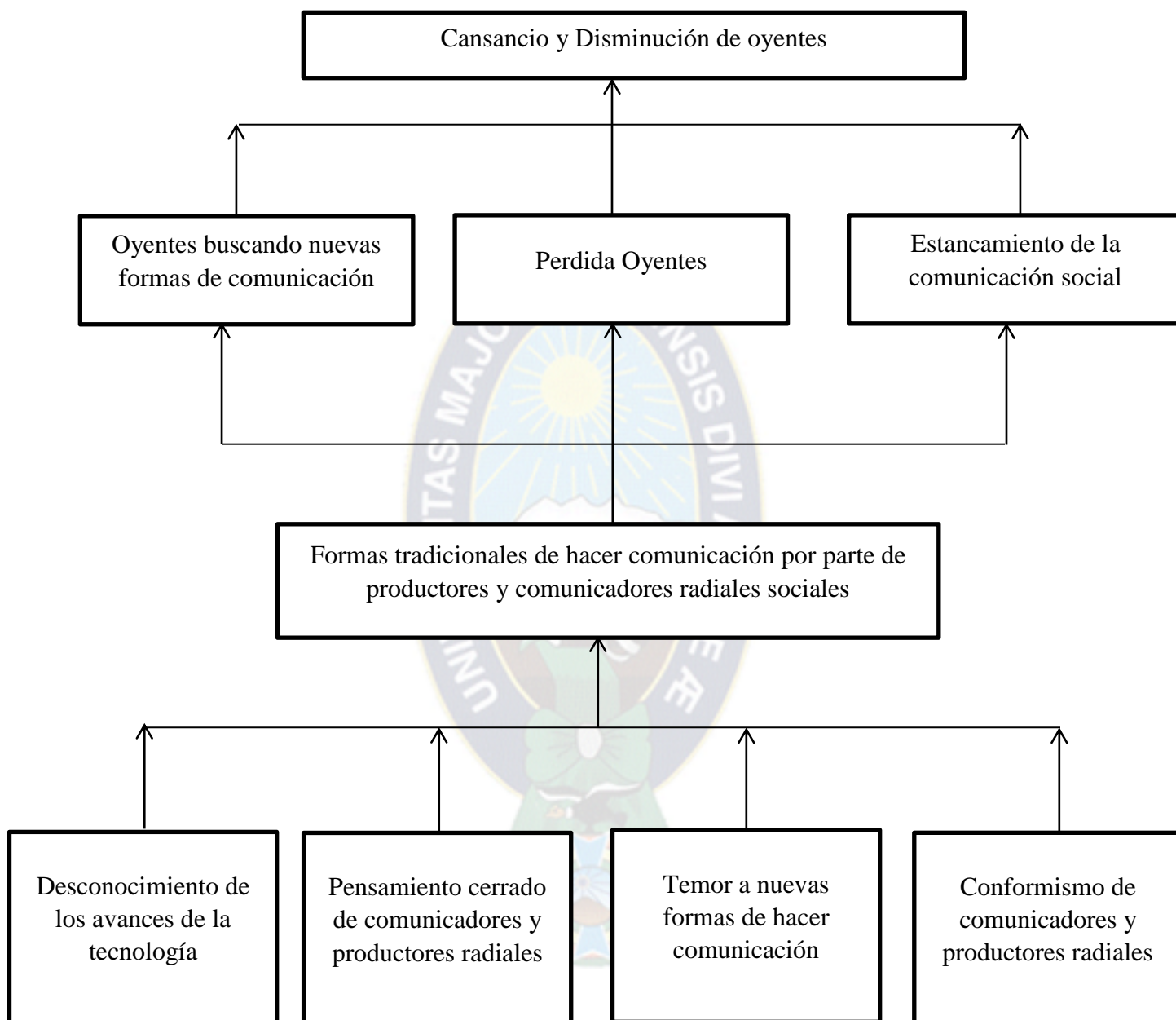
ONESS. (2015). “ONESS PROJECT”. Recuperado de:
<http://oness.sourceforge.net/proyecto/html/ch05s02.html>

VTT and K2 Quality Partners. (2015). Agile Software Technologies. Recuperado de:
<http://agile.vtt.fi/mobile.html>

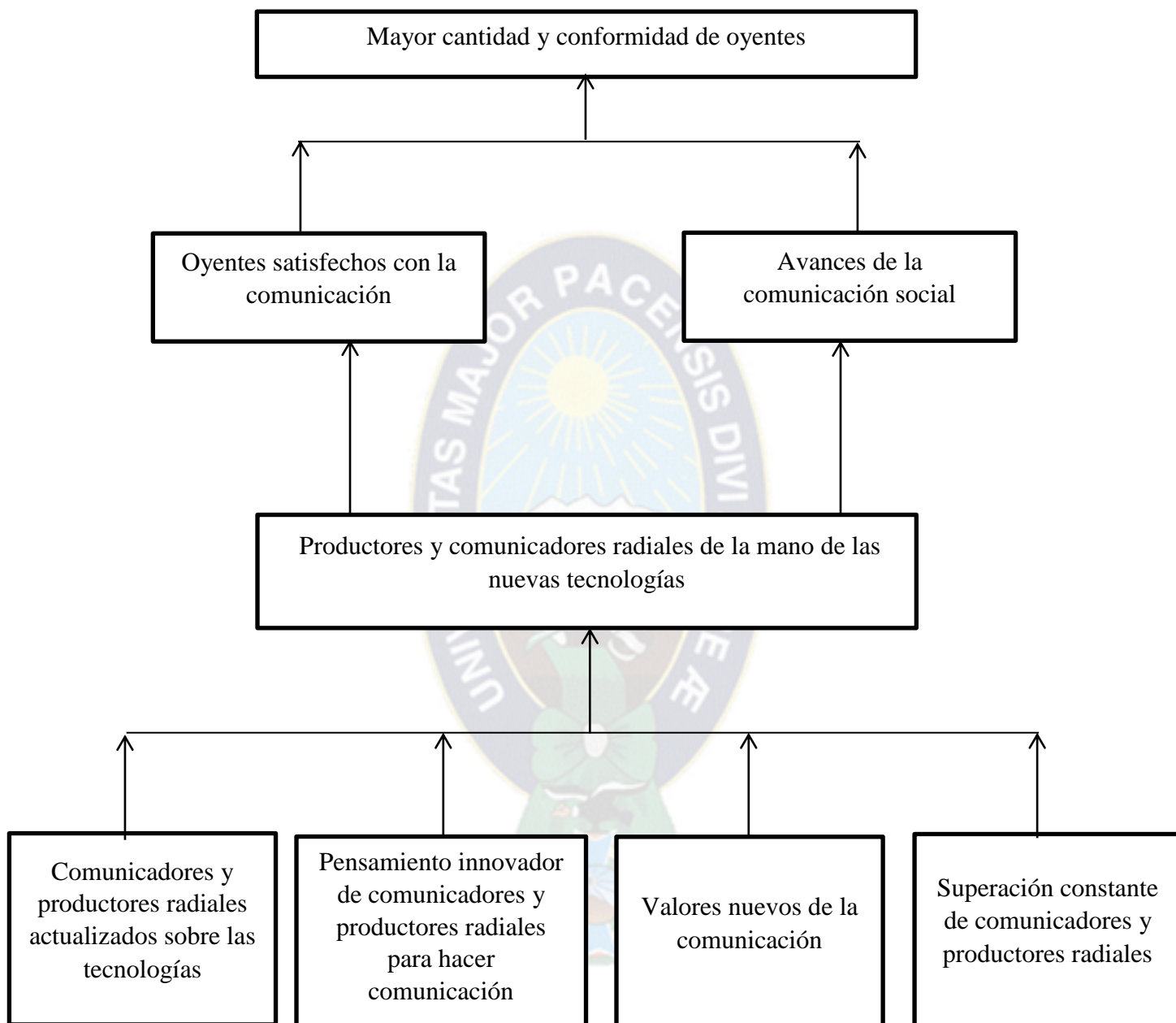
WEBML. (2015). “The Web Modeling Language”. Recuperado de: <http://www.webml.org/>

WIKIPEDIA. (2015). “Streaming”. Recuperado de: <http://es.wikipedia.org/wiki/Streaming>

ANEXO A – ARBOL DE PROBLEMAS



ANEXO B – ARBOL DE OBJETIVOS



ANEXO C – MARCO LÓGICO

RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACION	SUPUESTOS
<p>FIN</p> <p>Implementar y desarrollar un servicio de audio streaming orientado a diseños adaptativos.</p>	<ul style="list-style-type: none"> - Calidad de proyecto. - Cumplimiento de la norma ISO 9126 	<ul style="list-style-type: none"> - Aumento de números de oyentes. - Incrementa del número de visitantes al sistema web. - Usuarios que descargan aplicativo móvil en gran numero 	<ul style="list-style-type: none"> - Contar con recursos económicos para adquirir el hosting del sistema web. - Contar con un equipo de buenas características para sostener el servicio a un servidor de audio streaming.
<p>PROPOSITO</p> <p>Estar de mano de la tecnología, para llegar a nuevos oyentes, en todo lugar y no solamente de manera local sino globalmente. Y dejar de lado las formas tradicionales de hacer comunicación</p>	<ul style="list-style-type: none"> - Implementación del sistema en 3 meses. - Reducción de tiempo en la realización de actividades. - Mejor llegada del contenido de audio en los oyentes 	<ul style="list-style-type: none"> - Oyentes que escuchan el programa desde lugares lejanos y envían la su opinion desde el que encuentran. - Sistema implementado y desarrollado en la empresa 	<ul style="list-style-type: none"> - No cerrarse en la forma de hacerse comunicación de manera común. - Informarse de mejor sobre las nuevas tecnologías.
<p>COMPONENTES</p> <ul style="list-style-type: none"> - Módulo de gestión de un servicio a un 	<ul style="list-style-type: none"> - Funcionamiento fluido de audio del servicio a un servidor 	<ul style="list-style-type: none"> - Emisión de audio en el servidor. - Emisión de 	<ul style="list-style-type: none"> - Contar con herramientas para implementar el

<p>servidor de streaming de audio.</p> <ul style="list-style-type: none"> - Control de administración del material de audio en el servicio streaming. - Gestión del contenido en el sistema web. - Seguimiento al sistema para tener un buen desempeño. 	<p>streaming.</p> <ul style="list-style-type: none"> - Registro y autenticación del administrador - Cambio constante del contenido en el sistema web - Actualización del material de audio y contenido en sistema web 	<p>audio en el sistema Web.</p> <ul style="list-style-type: none"> - Emisión de audio en el aplicativo móvil. - Actualización de contenido del sistema web en el Internet. - Actualización en internet del contenido en el aplicativo móvil 	<p>servidor de streaming.</p> <ul style="list-style-type: none"> - Contar con un buen material de audio y conocimiento sobre el contenido. - Conocimiento y experiencia del manejo del sistema.
<p>ACTIVIDADES</p> <ul style="list-style-type: none"> - Recopilación necesaria para los procesos. - Implementación del servicio a un servidor de streaming de audio - Diseño adaptativo, desarrollo del sistema web y el aplicativo móvil. - Verificación y pruebas para el funcionamiento correcto del sistema. 	<ul style="list-style-type: none"> - Entrevista con gerente de la productora. - Entrevista para el diseño del sistema web con el gerente de la empresa. - Entrevista para el diseño del aplicativo móvil con el gerente de la empresa. - Pruebas de funcionamiento y stress. 	<ul style="list-style-type: none"> - Desarrollo de historias de usuario. - WebML para el diseño adaptativo para el sistema web. - Diseño del aplicativo móvil. - Historias de usuarios para recabar los requerimientos del sistema. 	<ul style="list-style-type: none"> - Predisposición del gerente de la empresa para entrevista sobre el sistema, - Predisposición del gerente de la empresa para entrevista sobre el diseño adaptativo del sistema web. - Predisposición del gerente de la empresa para entrevista sobre el diseño del aplicativo móvil.

CRONOGRAMA DE AVANCE

ACTIVIDAD	DURACIÓN EN DÍAS	DEL 1 DE JULIO AL 30 DE NOVIEMBRE																					
		JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DIC	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Redacción del Capítulo II - Marco Teórico	14	■	■																				
Desarrollo del Capítulo III - Marco Aplicativo	7			■																			
1ra Fase - XP Planificación del proyecto	21			■	■	■																	
2da Fase – XP Diseño	28					■	■	■	■														
3ra Fase - XP Codificación	28									■	■	■	■	■									
4ta Fase – XP Pruebas	28														■	■	■	■					
Redacción del Capítulo III - Marco Aplicativo	28					■	■	■	■														
Redacción del Capítulo IV - Calidad y Seguridad	28									■	■	■	■										
Redacción del Capítulo V - Análisis Costo Beneficio	28														■	■	■	■					
Redacción del Capítulo VI – Conclusión y Recomendación	28																		■	■	■		