



# **ESCUELA POLITÉCNICA DEL EJÉRCITO EXTENSIÓN LATACUNGA**

## **CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO  
EN SISTEMAS E INFORMÁTICA**

***Desarrollo de un prototipo del sistema de automatización  
para el audio de emisión en vivo de la radio Majestad del  
Distrito Metropolitano de Quito aplicando la Ingeniería de  
Requisitos en todas sus fases***

**DIEGO OSWALDO PULE LÓPEZ**

**LATACUNGA – ECUADOR**

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**2012**

CARRERA DE INGENIERÍA EN SISTEMAS E  
INFORMÁTICA

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, PULE LÓPEZ DIEGO OSWALDO,

**DECLARO QUE:**

El proyecto de grado denominado “**Desarrollo de un prototipo del Sistema de Automatización para el audio de emisión en vivo de la Radio Majestad del Distrito Metropolitano de Quito aplicando la Ingeniería de Requisitos en todas sus fases**” ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 17 de Febrero del 2012.

PULE LÓPEZ DIEGO OSWALDO

**ESCUELA POLITÉCNICA DEL EJÉRCITO**  
**CARRERA DE INGENIERÍA SISTEMAS E INFORMÁTICA**

**CERTIFICADO**

ING. JOSE LUIS CARRILLO (DIRECTOR)  
ING. JAVIER MONTALUISA (CODIRECTOR)

**CERTIFICAN:**

Que el trabajo titulado **“Desarrollo de un prototipo del Sistema de Automatización para el audio de emisión en vivo de la Radio Majestad del Distrito Metropolitano de Quito aplicando la Ingeniería de requisitos en todas sus fases”** realizado por el señor: PULE LÓPEZ DIEGO OSWALDO ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, **SI** recomiendan su publicación.

El mencionado trabajo consta de UN empastado y UN disco compacto el cual contiene los archivos en formato portátil de Acrobat. Autorizan al señor: PULE LÓPEZ DIEGO OSWALDO que lo entregue al ING. SANTIAGO JACOME, en su calidad de Director de Carrera.

Latacunga, 17 de Febrero de 2012.

-----  
Ing. José Luis Carrillo  
**DIRECTOR**

-----  
Ing. Javier Montaluisa  
**CODIRECTOR**

# ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA DE SISTEMAS E  
INFORMÁTICA

## AUTORIZACIÓN

Yo, PULE LÓPEZ DIEGO OSWALDO

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo **“Desarrollo de un prototipo del Sistema de Automatización para el audio de emisión en vivo de la Radio Majestad del Distrito Metropolitano de Quito aplicando la Ingeniería de requisitos en todas sus fases”** cuyo contenido, ideas y criterios son de MI exclusiva responsabilidad y autoría.

Latacunga, 17 de Febrero del 2012.

PULE LÓPEZ DIEGO OSWALDO

## **AGRADECIMIENTO**

A mi director de tesis, Ing. José Luis Carrillo, por iniciarme en el mundo de la investigación y apoyarme a continuar con mi trabajo a pesar de las dificultades obtenidas.

A mi madre por todo el apoyo incondicional para continuar con mis estudios durante este tiempo.

*Diego Oswaldo*

# ÍNDICE DE CONTENIDOS

PORTADA .....	I
DECLARACIÓN .....	II
CERTIFICADO .....	III
AUTORIZACIÓN .....	IV
AGRADECIMIENTO .....	V
ÍNDICE DE CONTENIDOS .....	VI
ÍNDICE DE TABLAS .....	IX
ÍNDICE DE FIGURAS .....	X
RESUMEN .....	XI
SUMMARY .....	XIII
CAPÍTULO 1 .....	1
MARCO TEÓRICO .....	1
1.1. INTRODUCCIÓN .....	1
1.2. INGENIERÍA DE SOFTWARE .....	3
1.3. INGENIERÍA DE REQUISITOS .....	4
1.3.1. <i>La necesidad de una ingeniería de requisitos</i> .....	5
1.3.2. <i>Requisito</i> .....	8
1.3.3. <i>Fases de la ingeniería de requisitos</i> .....	9
1.3.4. <i>Obtener requisitos</i> .....	11
1.3.5. <i>Obtención</i> .....	17
1.3.6. <i>Análisis</i> .....	17
1.3.7. <i>Documentación</i> .....	18
1.3.8. <i>Verificación</i> .....	20
1.3.9. <i>Validación</i> .....	21

1.4. HERRAMIENTAS REPRESENTATIVAS.....	22
1.5. METODOLOGÍA DE DESARROLLO DE SOFTWARE .....	23
1.5.1. <i>Programación Extrema (XP)</i> .....	24
1.6. DISEÑO WEB Y ESTÁNDARES.....	33
1.7. GUIA WEB - MODELO CLÁSICO.....	34
<b>CAPÍTULO II .....</b>	<b>36</b>
<b>FASES DE LA INGENIERÍA DE REQUISITOS.....</b>	<b>36</b>
2.1. GENERALIDADES .....	36
2.2. FASES DE LA INGENIERÍA DE REQUISITOS .....	36
2.2.1. <i>Obtención de requisitos</i> .....	36
2.2.2. <i>Analizar Requisitos</i> .....	49
2.2.3. <i>Documentar los requisitos</i> .....	50
2.2.4. <i>Verificar los requisitos</i> .....	55
2.2.5. <i>Validar los requisitos</i> .....	56
2.3. RESUMEN .....	57
<b>CAPÍTULO III .....</b>	<b>59</b>
<b>ELABORACIÓN DEL PROTOTIPO.....</b>	<b>59</b>
3.1. INTRODUCCIÓN.....	59
3.2. DESCRIPCIÓN GENERAL DEL SISTEMA .....	60
3.3. METODOLOGÍA DE DESARROLLO.....	61
3.4. HERRAMIENTAS DE DESARROLLO.....	62
3.4.1. <i>Apache</i> .....	62
3.4.2. <i>MYSQL V 5.1.13.</i> .....	63
3.4.3. <i>ECLIPSE</i> .....	64
3.5. SISTEMA DE AUTOMATIZACION .....	66
3.5.1. <i>Macroprocesos</i> .....	66
3.5.2. <i>Especificación de requisitos</i> .....	67
3.6. ANÁLISIS DE INFORMACIÓN .....	72
3.6.1. <i>Priorización y estimación</i> .....	72
3.6.2. <i>Distribución funcional</i> .....	73
3.6.3. <i>Estimación de la duración del proyecto</i> .....	74
3.6.4. <i>Plan de entregas</i> .....	75

<b>3.9.1. Diseño de la Experiencia de Usuario</b> .....	79
<b>3.10. PROTIPO DE INTERFACES</b> .....	84
<b>3.11. Pruebas</b> .....	90
<b>3.11.1. Pruebas de la primera iteración</b> .....	90
<b>3.11.2. Pruebas de la segunda iteración</b> .....	91
<b>3.11.3. Pruebas de la tercera iteración</b> .....	92
<b>3.12. IMPLEMENTACIÓN</b> .....	93
<b>3.12.1. Hospedaje del sitio</b> .....	93
<b>3.12.2. Medición de la capacidad de automatización</b> .....	93
<b>3.12.3. Alojamiento del sitio</b> .....	94
<b>CAPÍTULO IV</b> .....	95
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	95
<b>4.1. CONCLUSIONES</b> .....	95
<b>4.2. RECOMENDACIONES</b> .....	96
<b>BIBLIOGRAFÍA</b> .....	97
<b>ANEXOS</b> .....	100
ANEXO A .....	101
ANEXO B .....	107
ANEXO C .....	117
ANEXO D .....	118
ANEXO E .....	119



## ÍNDICE DE TABLAS

TABLA 2.1. CRONOGRAMA DE REUNIONES.....	44
TABLA 2.2: SOLICITUDES DEL SISTEMA.....	45
TABLA 2.3. MATRIZ DE ATRIBUTOS DE LOS STAKEHOLDERS .....	52
TABLA 2.4. MATRIZ DE ATRIBUTOS PARA LOS ACTORES .....	52
TABLA 2.5. MATRIZ DE CARACTERÍSTICAS DEL SOFTWARE .....	53
TABLA 2.6. MATRIZ DE CASOS DE USO .....	54
TABLA 2.7. MATRIZ DE ATRIBUTOS DE LAS CLASES.....	54
TABLA 2.8. HISTORIAS DE USUARIOS FINALES.....	55
TABLA 2.9. INTERROGANTES VALIDACIÓN DE REQUISITOS.....	56
TABLA 3.10. PROCESOS AUTOMATIZADOS DE RADIO MAJESTAD.....	66
TABLA 3.11. GESTIÓN DE USUARIOS.....	67
TABLA 3.12. GESTIÓN DE LISTAS DE REPRODUCCIÓN.....	68
TABLA 3.13. GESTIÓN DE CONTRATOS PARA PUBLICIDAD.....	68
TABLA 3.14. GESTIÓN DE CLIENTES .....	69
TABLA 3.15. GESTIÓN DE FACTURACIÓN.....	69
TABLA 3.16. GESTIÓN DE PROGRAMAS .....	70
TABLA 3.17. GESTIÓN DE PAQUETES PUBLICITARIOS.....	70
TABLA 3.18. GESTIÓN DE ACCESOS VÁLIDOS AL SISTEMA .....	71
TABLA 3.19. GESTIÓN DE INTENTOS DE ACCESO AL SISTEMA.....	71
TABLA 3.20. GESTIÓN DE ACCESOS VÁLIDOS AL SISTEMA .....	72
TABLA 3.21. PRIORIZACIÓN Y ESTIMACIÓN DE TIEMPOS DE IMPLEMENTACIÓN	73
TABLA 3.22. DISTRIBUCIÓN FUNCIONAL DE LAS HISTORIAS DE USUARIOS .....	73
TABLA 3.23. ESTIMACIÓN DE LA DURACIÓN DEL PROYECTO.....	74
TABLA 3.24. PLAN DE ENTREGAS DE ACUERDO A ITERACIONES.....	75
TABLA 3.25. PLAN DE ENTREGAS DE ACUERDO A LOS MÓDULOS.....	76
TABLA 3.26. PLAN PRIMERA ITERACIÓN .....	77
TABLA 3.27. PLAN SEGUNDA ITERACIÓN .....	78
TABLA 3.28. PLAN TERCERA ITERACIÓN.....	78

## ÍNDICE DE FIGURAS

FIGURA 1.1: INGENIERÍA DE REQUISITOS.....	5
FIGURA 1.2: RESULTADOS DEL INFORME DEL GAO.....	6
FIGURA 1.3: RESULTADOS DEL INFORME CHAOS.....	7
FIGURA 1.4: DESARROLLO DE UN SISTEMA USANDO XP.....	25
FIGURA 1.5: BASES DE EXTREME PROGRAMMING.....	27
FIGURA 1.6: PROCESO DE EXTREME PROGRAMMING.....	29
FIGURA 3.7: TARJETA DE SONIDO.....	60
FIGURA 3.8: INGRESO AL SISTEMA.....	80
FIGURA 3.9: PROGRAMACIÓN.....	81
FIGURA 3.10: PAQUETES.....	81
FIGURA 3.11: USUARIOS.....	82
FIGURA 3.12: CLIENTES.....	82
FIGURA 3.13: FACTURAS.....	83
FIGURA 3.14: CONTRATOS.....	83
FIGURA 3.15: PÁGINA DE INICIO AL SISTEMA.....	84
FIGURA 3.16: CONSOLA DEL SISTEMA.....	84
FIGURA 3.17: PROGRAMAS.....	85
FIGURA 3.18: PAQUETES.....	85
FIGURA 3.19: USUARIOS.....	86
FIGURA 3.20: MAJESTAD STUDIO.....	86
FIGURA 3.21: CLIENTES.....	87
FIGURA 3.22: PROGRAMACIÓN.....	87
FIGURA 3.23: FACTURACIÓN.....	88
FIGURA 3.24: CONTRATOS.....	88
FIGURA 3.25: ACCESO AL SISTEMA.....	89
FIGURA 3.26: ERROR DE USO DEL SISTEMA.....	89

## RESUMEN

El mundo tecnológico en el cual actualmente nos encontramos, da lugar a que día a día las empresas destinen gran cantidad de su presupuesto a automatizar ciertas actividades que se llevan a cabo dentro de su organización, con la finalidad de satisfacer las más exigentes necesidades del cliente; esta investigación es una recopilación de requerimientos basados en mejorar la automatización de la señal en vivo que emite Radio Majestad de la ciudad de Quito, y en base a los cuales se plantea desarrollar un prototipo para tal efecto.

La metodología que se empleó para desarrollar el prototipo es la denominada Extreme Programming (Programación Extrema), por su rapidez y agilidad en el desarrollo de productos de software, orientados a la mayor satisfacción del cliente. El desarrollo de software usando la metodología XP (Extreme Programming), se sustenta en aspectos muy importantes como son la comunicación, simplicidad, valor y retroalimentación.

A continuación se realiza una breve descripción acerca de los contenidos de cada uno de los tres capítulos en los cuales se ha distribuido esta tesis:

El capítulo I, hace referencia a la fundamentación teórica, la cual sirve de base para el desarrollo de este proyecto; en este capítulo se analizan todo lo concerniente con la ingeniería de software, la ingeniería de requisitos, la metodología XP empleada en el desarrollo del prototipo y aspectos básicos sobre el desarrollo de guías web.

El capítulo II, hace referencia a la obtención de requisitos, mediante la identificación de los interesados, analizando sus respectivos puntos de vista para al final en conjunto recopilar los requisitos necesarios, los

mismos que serán analizados, documentados, verificados y validados para ser empleados en el prototipo a desarrollar.

El capítulo III, está relacionado con la propuesta de la investigación, aquí se presenta de manera detallada la descripción del sistema, el mismo que ha sido diseñado empleado herramientas informáticas de última generación y acordes a los procesos a automatizar, se presenta el prototipo de interfaces web y pruebas, para al final proceder con la implementación del mismo.

El capítulo IV, enuncian las conclusiones y recomendaciones tendientes a la presente investigación.

Finalmente, se presentan las referencias bibliográficas y se adjuntan los anexos que se han recopilado durante el desarrollo de todo el trabajo investigativo.

## **SUMMARY**

The technological world in which we are currently results in that every day companies allocate large amount of its budget to automate certain activities carried out within your organization, in order to satisfy the most demanding customer needs, this research is a compilation of requirements based on improving the automation of the live signal emitted Majesty Radio Quito, and on the basis of which is proposed to develop a prototype for this purpose.

The methodology used to develop the prototype is called Extreme Programming (Extreme Programming), for his speed and agility in the development of software products, aimed at increased customer satisfaction. Software development methodology using XP (Extreme Programming), is based on important aspects such as communication, simplicity, value and feedback.

Here is a brief description about the contents of each of the three chapters which was distributed this thesis:

Chapter I, refers to the theoretical foundation, which provides the basis for the development of this project in this chapter discusses everything related to software engineering, requirements engineering, the XP methodology used in the development of prototype and basic aspects of the development of web guides.

Chapter II refers to the gathering of requirements, by identifying stakeholders, analyzing their respective points of view together to finally collect the requirements, they will be analyzed, documented, verified and validated for use in the prototype to be developed.

Chapter III, is related to the proposed research, here is a detailed description of the system, it has been designed employee-art tools and chords to automate processes, presents the prototype interfaces web and testing, only to proceed with implementation.

Chapter IV sets out the conclusions and recommendations to the current investigation.

Finally, some references and appendices are attached have been compiled during the development of the entire research work.

# **CAPÍTULO 1**

## **MARCO TEÓRICO**

### **1.1. INTRODUCCIÓN**

Imaginarse por un momento una sala de un centro de cálculo a principios de los años 50 en algún organismo del gobierno de los Estados Unidos. El local está ocupado por una enorme computadora cuya elaboración ha costado varios millones de dólares y cuyo mantenimiento tiene ocupados a varios técnicos a la vez las 24 horas del día los 365 días al año.

La siguiente media hora es el turno semanal de uno de los muchos científicos que trabajan en el Departamento de Defensa de los Estados Unidos. Su trabajo consiste en calcular tablas numéricas para trayectorias de misiles usando ecuaciones sumamente complejas y un uso relativo de muchas variables. Lleva más de una semana repasando su programa línea por línea en su oficina para asegurarse de que no contiene errores sintácticos ni semánticos, ya que un fallo en la compilación podría hacerle perder su valiosa media hora semanal.

Cuando comienza su turno, la computadora comienza a leer las tarjetas perforadas que previamente había entregado al operador, compila el programa sin errores y comienza su ejecución. El científico se dispone a esperar a que la impresora comience a imprimir las tablas con los resultados de las ecuaciones. Cuando faltan algunas líneas para completarlas, el hardware falla y los encargados de mantenimiento deben parar la máquina para repararla. Quizá y posiblemente con un poco de suerte tenga los resultados la próxima semana.

Si se compara esta situación con la actual, pocos son los aspectos comunes que podríamos encontrar. Sin embargo, podemos hacer un

ejercicio de análisis e identificar algunas de las características del desarrollo de software en los comienzos de la informática allá en los años 50.

El hardware es mucho más caro que el software-. La computadora y el mantenimiento cuestan millones de dólares. Comparado con ese costo, el sueldo del científico que escribe el programa es ridículo, así que ¿Por qué preocuparse por el costo del desarrollo del software?.

El desarrollo del hardware es más complejo que el del software. La tecnología hace que el hardware sea complejo de construir y mantener. El software habitual en aquella época eran simples programas no muy grandes ni complejos debido a la escasa cantidad de memoria y solían estar escritos por una sola persona, normalmente empleada en la organización que utiliza el hardware. Los requisitos que tiene que cumplir el software son simples. Por tanto ¿Por qué preocuparse por la complejidad del software?

El hardware es poco fiable-. Debido a la tecnología que se utiliza para su implementación, en cualquier momento la computadora puede sufrir algún desperfecto, así que ¿Para qué preocuparse por la calidad del software?.

Esta desinteresada situación respecto al desarrollo de software cambió cuando, gracias a los avances en la tecnología, aumenta la capacidad de memoria, procesamiento, fiabilidad y obviamente se reducen los costos de desarrollo y mantenimiento del hardware. Se empiezan a comercializar las primeras computadoras y la demanda de software más complejo empieza a crecer rápidamente, destapando algo llamado “La crisis del software”, término utilizado por primera vez en la conferencia organizada por la Comisión de Ciencias de la OTAN en Garmisch, Alemania, en octubre de 1968, para designar la gran cantidad de problemas que



presentaban y aun presentan el desarrollo de software y el alto índice de fracasos y deserción en los proyectos de desarrollo.

¿Qué podía hacerse antes una situación en la que los proyectos de software tenían un alto riesgo de fracasar? La respuesta parecía obvia: construir software de forma similar a como se construye hardware, aviones, barcos o edificios, es decir aplicar los métodos de la ingeniería al desarrollo de software.

## **1.2. INGENIERÍA DE SOFTWARE**

En la ingeniería de sistemas y la ingeniería de software, la Ingeniería de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los inversores, que pueden entrar en conflicto entre ellos.

Muchas veces se habla de requerimientos en vez de requisitos; esto se debe a una mala traducción del inglés. La palabra requirement debe ser traducida como requisito, mientras que requerimiento se traduce al inglés como request.

El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones, etc.

El aspecto más importante en la ingeniería de requisitos es la comunicación, esta característica es la que hace de la ingeniería de requisitos una disciplina especialmente compleja ya que hay un factor

que, aunque lleva siendo estudiado mucho tiempo, apenas se conoce aun, es el factor humano.

Este factor es el responsable de que la ingeniería de requisitos tenga aspectos sociales, culturales y no solo técnicos.

### **1.3. INGENIERÍA DE REQUISITOS**

La comprensión de los requisitos de un problema está entre las tareas más difíciles que enfrenta un ingeniero de software. Cuando se piensa por primera vez acerca de ello, la ingeniería de requisitos no parece nada complicado; al fin y al cabo, ¿El cliente no sabe lo que requiere?, ¿Los usuarios finales no deberían entender bien las características y funciones que les proporcionarán un beneficio?, es realmente increíble, pero en muchas ocasiones la respuesta es: “los clientes no saben que es lo que quieren hasta que se les enseña”. Y aun si los clientes y usuarios finales son explícitos en sus necesidades, estos requisitos pueden cambiar durante el proyecto.

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, que es lo que el cliente quiere y como interactuarán los usuarios finales con el software.

Dejando de lado, el diseño y la construcción de un elegante programa de computadora, que resuelve un programa incorrecto que no satisface las necesidades de nadie, es muy importante entender lo que el cliente quiere antes de comenzar a diseñar y construir un sistema computacional.

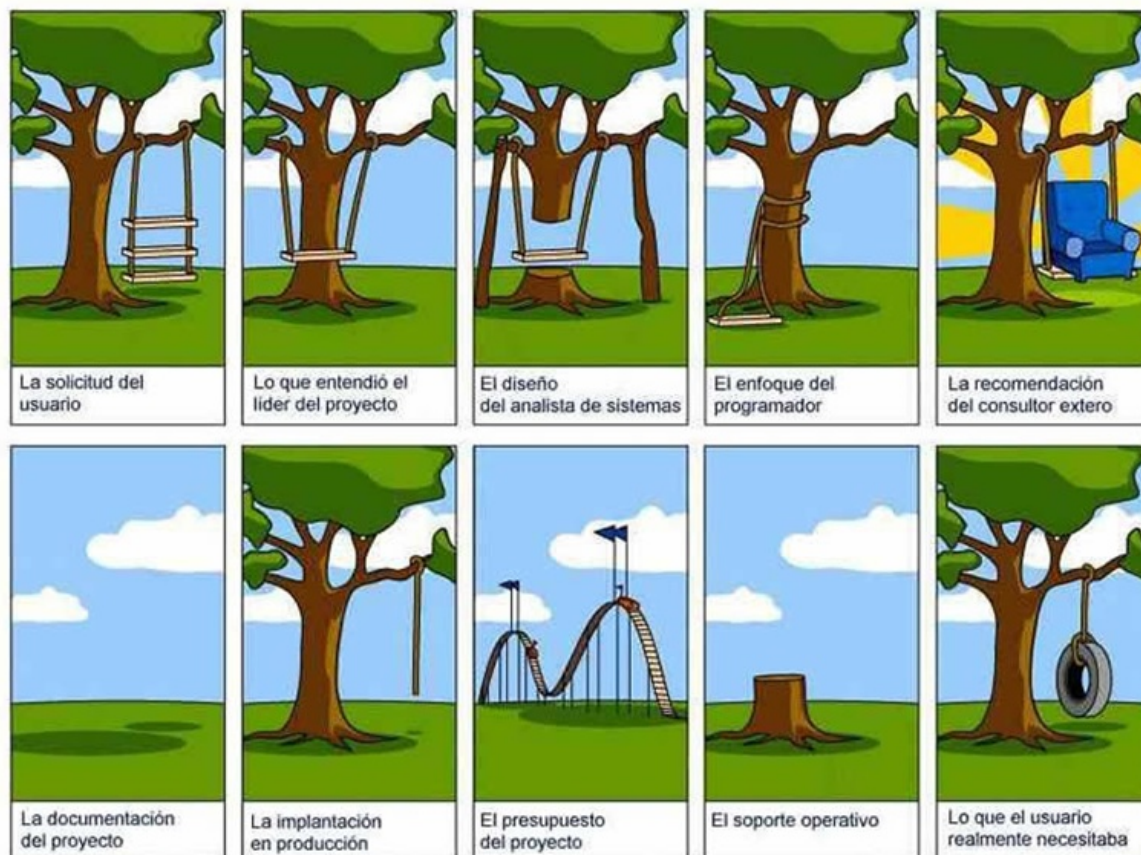


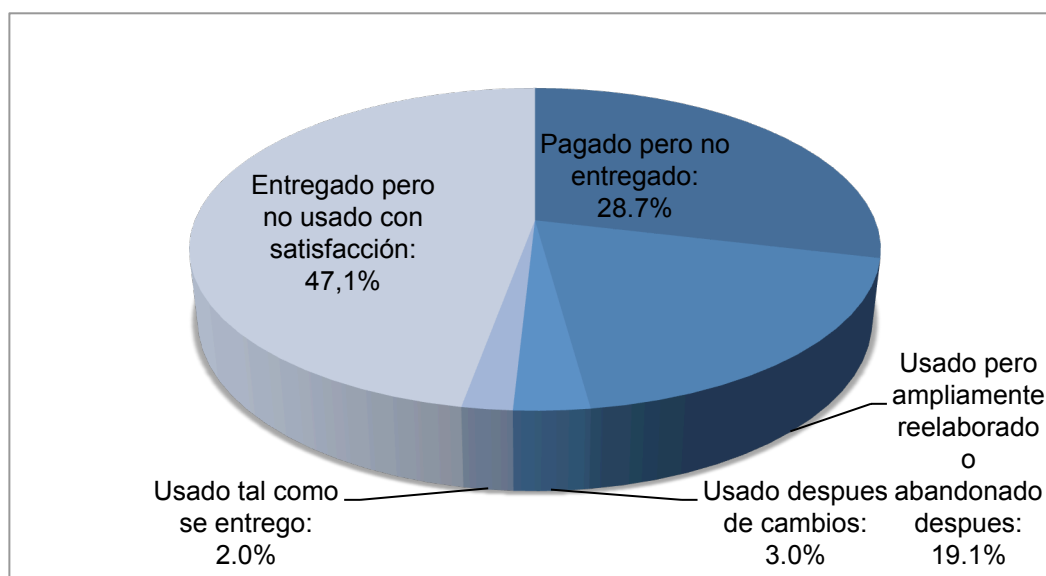
Figura 1.1: Ingeniería de Requisitos

### 1.3.1. La necesidad de una ingeniería de requisitos

En 1979, la oficina de presupuestos del gobierno norteamericano (GovernmentAccount Office, GAO) realizó un estudio (GAO 1979) seleccionando 9 proyectos de desarrollo de software para el gobierno norteamericano cuyos contratos sumaban una cantidad total de \$ 6'800.000 USD.

De esta cantidad, sólo \$119.00 USD correspondían a un proyecto que había utilizado tal como se había entregado. Dicho proyecto se trataba de un preprocesador de lenguaje para COBOL, por lo que era un problema relativamente simple cuyos requisitos eran comprendidos por los clientes y desarrolladores y estos no fueron cambiados durante el desarrollo.

El resto de los 6.8 millones de dólares se distribuyeron como puede verse en la figura 1.1, en la que se puede destacar el enorme porcentaje de dinero invertido en proyectos cancelados o no satisfactorios.

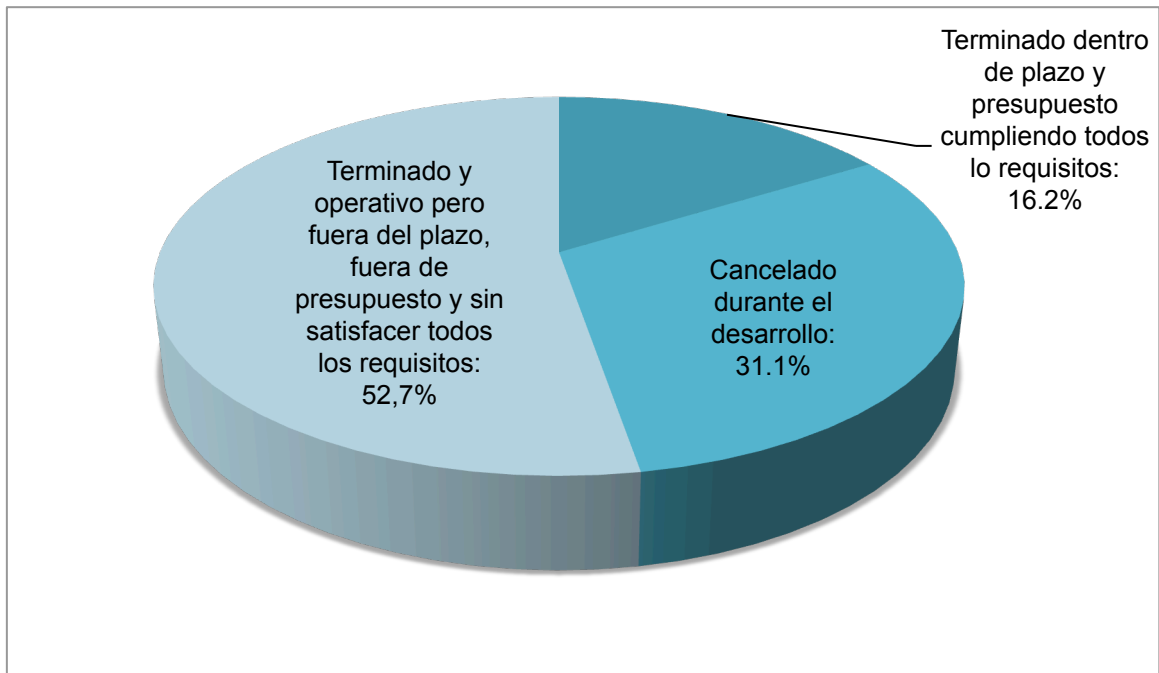


**Figura 1.2: Resultados del informe del GAO**

En 1995, el grupo de Standish realizó un estudio (el informe CHAOS) mucho más amplio y significativo que el GAO cuyos resultados, que a pesar de haber pasado más de 25 años, no reflejaban una mejoría sustancial (TSG 1995).

Los resultados generales, que pueden verse en la figura 1.2, al compararse con los datos (GAO 1979) representan una mejora en proyectos que se entregan cumpliendo todos los requisitos, 2% frente al 16.2% (sólo el 9% en grandes compañías), pero empeoran ligeramente respecto a los que se abandonan durante el desarrollo, 28.7% frente a 31.1%.

Sin incluir al 16.2% de los proyectos terminados correctamente, la media del gasto final fue del 18,9% del presupuesto original, el tiempo necesario para su realización del 22,2% del plazo original y se cumplieron una media del 61% de los requisitos iniciales, cifras que también empeoraron en el caso de grandes compañías.



**Figura 1.3: Resultados del informe CHAOS**

Las encuestas realizadas a los directores de los proyectos que participaron en el estudio indicaron que, en su opinión, los tres principales factores de éxito eran:

- Participación activa de los usuarios
- Apoyo a los directivos
- Enunciación clara de los requisitos

Mientras que los tres principales factores de fracaso fueron:

- Falta de información por parte de los usuarios.
- Especificaciones y requisitos incompletos
- Especificaciones y requisitos cambiantes

Estos informes ponen de manifiesto el hecho de que, a pesar que las herramientas para desarrollar software han evolucionado enormemente, se sigue produciendo software que no es satisfactorio para los clientes y usuarios. Esto indica que los principales problemas que han dado origen a la crisis del software residen en las primeras etapas del desarrollo, cuando hay que decidir las características del producto software a desarrollar.

“La parte más difícil de desarrollar un sistema de software es decir que desarrollar. Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después”.<sup>1</sup>

“Otro hecho comprobado es el costo de un cambio de requisitos, una vez entregado el producto, es entre 60 y 100 veces superior al costo que hubiera representado el mismo durante las fases iniciales del desarrollo”<sup>2</sup>;

por lo que no es de extrañar que aquellos proyectos en los que no se determinan correctamente los requisitos y cambian frecuentemente durante el desarrollo, superen con creces su presupuesto inicial.

### **1.3.2. Requisito**

Una de las características de la ingeniería de requisitos es la falta de uniformidad en la terminología empleada, tanto para los conceptos básicos como para los procesos y los productos [Davis 1993, Brackett 1990, Pohl 1997]. Uno de los conceptos afectados por dicha falta de uniformidad es el de requisito.

---

<sup>1</sup>F.P. Brooks [Brooks 1995, pág. 199]

<sup>2</sup>Pressman 1997, Davis 1993, pág. 37

La definición que se puede encontrar en la [IEEE 1990] dice lo siguiente:

Requisito (1). (a) Una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo. (b) Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, especificación u otro documento. (c) Una representación en forma de documento de una condición o capacidad como las expresadas en (a) o (b).

Sin embargo, a pesar de esta aparente simplicidad del concepto, es frecuente encontrar el término requisito calificado con adjetivos que pueden resultar confusos en un primer momento: de sistemas, hardware, software, de usuario, de cliente, funcional, no funcional, etc.

### **1.3.3. Fases de la ingeniería de requisitos<sup>3</sup>**

Desde un punto de vista general, las principales fases son las siguientes:

- Obtener requisitos
- Analizar requisitos
- Documentar requisitos
- Verificar requisitos
- Validar los requisitos

La ingeniería de requisitos empieza con una fase preliminar de inicio, la cual es una tarea que define el ámbito y la naturaleza del problema que debe resolverse. Después continúa con la obtención de requisitos, que es una tarea que ayuda al cliente a definir sus necesidades; posteriormente sigue con el análisis, que es la fase donde se redefine y modifican los

---

<sup>3</sup>McConnell, Steve (1996). Rapid Development: Taming Wild Software Schedules, 1st ed., Redmond, WA: Microsoft Press.

requisitos básicos. Cuando el cliente ha definido el problema se lleva a cabo la verificación de requisitos, donde se define cuales son las prioridades, cuales aspectos son esenciales y en qué momento se requieren. Por último, el problema se especifica de alguna manera, y después es validado y revisado para asegurar que la concepción del problema que tiene el ingeniero de software coincide con la percepción del cliente.

El objetivo final del proceso de la ingeniería de requisitos es darle a todas las fases una explicación escrita y lo más detallada posible sobre el problema. Esta puede lograrse por medio de varios productos de trabajo: casos de uso, lista de funciones y características, modelos de análisis o alguna especificación.

El ingeniero de software revisa los productos de trabajo de la ingeniería de requisitos junto con el cliente y los usuarios finales para asegurarse que se hayan entendido lo que en realidad pretendían decirle. Es necesario hacer una advertencia: aun después de que todas las partes están de acuerdo, las cosas cambian, y continuarán haciéndolo a través del ciclo la vida del proyecto.

La mayoría de los proyectos comienzan cuando se identifica una necesidad de negocio o se descubre un nuevo mercado o servicio potencial. Los participantes de la comunidad de negocios (es decir, los gerentes, gente de mercadotecnia, gerentes de producto) defienden un caso de negocios para la idea, tratan de identificar la amplitud y profundidad del mercado, hacen análisis preliminares de factibilidad, e identifican una descripción funcional del ámbito del proyecto. Toda esta información está sujeta a cambios, pero es suficiente para presentar conversaciones con la organización de ingeniería de software.



Al inicio del proyecto los ingenieros de software hacen una serie de preguntas libres de contexto, como por ejemplo:

- ¿Quién está detrás de la solicitud de este trabajo?
- ¿Quién usará la solución?
- ¿Cuál será el beneficio económico de una solución exitosa?
- ¿Existe otra fuente para la solución requerida?

El objetivo es establecer una comprensión básica del problema, las personas que quieren una solución, la naturaleza de la solución que se desea, y la efectividad de la comunicación preliminar entre el cliente y el desarrollador.

#### **1.3.4. Obtener requisitos**

La obtención de requisitos es la actividad de la ingeniería de requisitos en la que se estudia el dominio del problema y se interactúa con los clientes y usuarios para obtener y registra información sobre sus necesidades, estas interacciones pueden realizarse para recopilar información o para resolver conflictos que se hayan detectado en la información recopilada, avanzando de esta forma en las dimensiones del sistema.

A nivel de investigación, el obtener requisitos es sin duda la actividad a la que menos atención se le ha prestado en la ingeniería de software es por esta razón, que en la presente tesis se realizara dos veces la actividad de obtención requisitos; en la primera iteración consistirá básicamente en obtener la mayor cantidad posible de información, asumiendo que lo más probable es que dicha información sea incompleta, ambigua y contenga contradicciones.

En las siguientes actividades, de la obtención de requisitos consistirá principalmente en la resolución de conflictos encontrados en la información obtenida, la resolución de estos conflictos se llevará a cabo, normalmente, mediante algún tipo de negociación entre los participantes.

### **a. Problemas con la obtención de requisitos**

La mayor parte de los problemas del desarrollo de software están relacionados con la ingeniería de requisitos, y dentro de esta con la obtención de requisitos.

Aunque se disponga de excelentes lenguajes de especificación de requisitos e incluso aunque se consiga que los clientes y usuarios validen una determinada especificación, si no se han obtenido los requisitos correctos, todo el trabajo de desarrollo terminará con un producto técnicamente correcto pero inútil, ya que no satisfará las necesidades que dieron origen a su desarrollo.

Los problemas a los que se enfrenten en la obtención de requisitos son variados los más importantes son los siguientes: problemas de articulación, de comunicación, de limitaciones cognitivas, de conducta humana y técnicos.

#### **a.1. Problemas de articulación**

Los problemas de articulación están relacionados con la expresión de sus necesidades por parte de los clientes, usuarios y la comprensión de dichas necesidades, por parte de los desarrolladores. Algunos de estos problemas son los siguientes:

Los clientes y usuarios pueden ser conscientes de sus necesidades pero no son capaces de expresarlas de manera apropiada. Es lo que en sociología se denomina el problema de decir-hacer y en filosofía se denomina conocimiento tácito, “las personas saben cómo hacer muchas cosas que no saben describir”<sup>4</sup>, aquí se describe un claro ejemplo: Si una persona con hambre llega a un restaurante pero no puede decidir que quiere comer o no puede entender el menú, decirle al camarero que tiene hambre no es una articulación apropiada de unos requisitos a los que el camarero pueda responder.

“Algunos de los usuarios pueden no expresar sus necesidades por miedo a parecer incompetentes ante lo demás o porque los desarrolladores juegan un papel excesivamente dominante en el proceso, provocando que la falta de conocimiento tecnológico de los usuarios les haga sentir en inferioridad de condiciones”<sup>5</sup>.

“Los clientes pueden no llegar a tomar decisiones porque no pueden prever las consecuencias de su decisión o porque no entienden las alternativas que se les plantea”<sup>6</sup>.

En otras ocasiones no se toman decisiones porque no hay una sola persona que tenga una visión global, por lo que puede haber varios puntos de vista que tenga que integrarse.

Algunos desarrolladores no escuchan apropiadamente a los clientes y usuarios, porque creen haber entendido sus necesidades rápidamente, o bien se dedican a pensar inmediatamente sobre aspectos de implementación y no se colocan en el lugar de clientes y usuarios.

---

<sup>4</sup>Coguen 1994

<sup>5</sup>Macaulay 1999

<sup>6</sup>Kontoya y Sommerville 1996, Nuseibeh et al, 1994

## **a.2. Problemas de comunicación**

Algunas de las dificultades en la comunicación entre clientes, usuarios y desarrolladores son las siguientes.

“Los clientes, usuarios y desarrolladores tiene culturas y vocabularios diferentes, con la posibilidad de que los mismos términos tengan significados distintos en los diferentes vocabularios, o que su significado se vea enormemente afectado por el contexto, ya que en estos momentos del desarrollo, la información está fuertemente contextualizada”<sup>7</sup>.

No solo la cultura y el vocabulario son distintos, las preocupaciones sobre el sistema a desarrollar también suelen serlo. Mientras los clientes y usuarios suelen preocuparse por aspectos de bajo nivel como utilización de recursos, algoritmos, etc. Es importante no olvidar que el principal interés de los clientes no es un sistema software en sí mismo, sino los efectos positivos resultantes de la introducción del sistema en su organización.

El medio de comunicación que se utilice debe ser entendible por todos los participantes. Se suele utilizar lenguaje natural porque es el único medio de comunicación común a todos los participantes, a pesar de su inherente ambigüedad. “La utilización de otro tipo de técnicas como diagramas o lenguajes artificiales pueden presentar problemas de comprensión”<sup>8</sup>.

La comunicación puede verse afectada también por sus aspectos puramente sociales. “El ingeniero de software debe ser capaz de comunicarse y tratar con todo tipo de personas y ser capaz de manejar conflictos personales y políticos”<sup>9</sup>.

---

<sup>7</sup>Goguen 1994

<sup>8</sup> Davis 1995

<sup>9</sup>Goguen 1994, Macaulay 1998, Macaulay 1999

### **a.3. Problemas de conducta humana**

La naturaleza social de la licitación de requisitos provoca que surtan problemas de conducta humana, algunos de los cuales son los siguientes:

“La suposición o el temor a que el sistema a desarrollar cambie su forma de trabajar o incluso ponga en peligro su puesto de trabajo, puede provocar que algunos usuarios retengan información o incluso saboteen el desarrollo, por ejemplo proporcionando información falsa”<sup>10</sup>.

Puede haber conflictos y ambigüedades en los roles que cada persona debe jugar en el proceso de licitación. Dentro del grupo de clientes y usuarios, algunos pueden pensar que, aunque conozcan ciertas necesidades o ciertos aspectos importantes, es responsabilidad de otros participantes más afectados el hacerlas explícitas, con lo que el resultado final es que nadie dice nada. Otra situación similar pueden producirse si algunos clientes y usuarios piensan que los desarrolladores les harán todas las preguntas necesarias sobre el dominio del problema y los desarrolladores piensan que los clientes y usuarios les proporcionarán toda la información necesaria sin necesidad de preguntar por su parte, con lo que pueden quedar aspectos sin tratar.

### **a.4. Problemas técnicos**

Otros problemas en la obtención de requisitos pueden considerarse como técnicos, entre los cuales se tiene:

El software tiene que resolver problemas cada vez más complejos, por lo que sus requisitos son también a su vez más complejos, y contemplan detalles más específicos del dominio del problema.

---

<sup>10</sup> [Sawyer y Kontoya 1999]

Los requisitos cambian con el tiempo, ya que a medida que los clientes y usuarios van conociendo sus propias necesidades y las posibilidades que les ofrece la tecnología puede surgir la necesidad de reconsiderar decisiones anteriores o descubrir nuevas necesidades<sup>11</sup>. Esta es una de las principales causas de la naturaleza iterativa de la Ingeniería de Requisitos.

Otra fuente de cambios es el hecho de que el hardware y el software cambian rápidamente, haciendo asequible requisitos que antes era inabordables por su complejidad o por su coste.

En algunos sistemas puede haber muchas fuentes de requisitos, por lo que a veces no basta con consultar con los clientes y usuarios, sino que también es necesario hacerlo con técnicos, personal de mantenimiento, consultar normativas, estándares, etc.

Es necesario también tomar en cuenta que los sistemas novedosos requieren un esfuerzo mucho mayor de licitación, y que las fuentes de información pueden ser muy distintas dependiendo de la naturaleza del sistema a desarrollar.

## **b. Técnicas para la obtención de requisitos**

Las técnicas más habituales en la obtención de requisitos son las entrevistas, el Join Application development (JAD<sup>12</sup>) o desarrollo conjunto de aplicaciones, el brainstorming, o tormenta de ideas y la utilización de escenarios más conocidos como casos de uso.

---

<sup>11</sup>Cuando más se ve, más se necesita Davis 1995, El Principio 15.

<sup>12</sup>[www.carolla.com/wp-jab.htm](http://www.carolla.com/wp-jab.htm)

### **1.3.5. Obtención**

La información conseguida con el cliente durante el inicio y la obtención se expande y se refina durante la elaboración. Esta actividad de la ingeniería de requisitos se enfoca en el desarrollo de un modelo técnico refinado de las funciones, características y restricciones del software.

La elaboración es una acción del modelo del análisis que se compone de una serie de tareas de modelado y refinamiento. La elaboración se conduce mediante la creación y el refinamiento de escenarios del usuario, que describen la forma en que el usuario final (y otros actores) interactuarán con el sistema. Cada caso de uso se analiza para obtener clases de análisis: entidades del dominio de negocios visibles para el usuario final. Se definen los atributos de cada clase de análisis y se identifican los servicios (operaciones y métodos) que requieren cada clase. Se identifican las relaciones y la colaboración entre las clases y se produce una variedad de diagramas de UML complementarias.

El resultado final de la elaboración es un modelo de análisis que define el dominio de la información, las funciones y el comportamiento del problema.

### **1.3.6. Análisis**

Si se dan los recursos limitados del negocio, no resulta inusual que los clientes y usuarios pidan más de lo que se puede lograr. También es relativamente común que diferentes clientes o usuarios propongan requisitos que entran en conflicto en el momento de argumentar que su versión es “esencial para nuestras necesidades especiales”.

El ingeniero de requisitos debe conciliar estos conflictos por medio de un procesos de análisis y negociación. Se pide a los clientes, usuarios y otros interesados que ordenen sus requisitos y después discutan los conflictos relacionados con la prioridad. Se identifican y analizan los riesgos asociados con cada requisitos. Se hacen “estimaciones” preliminares del esfuerzo requerido para su desarrollo y después se utilizan para evaluar el impacto de cada requisito en el costo del proyecto y sobre el tiempo de entrega. Mediante un enfoque iterativo, los requisitos se eliminan, combinando o modificando de forma que cada parte alcance cierto grado de satisfacción.

### **1.3.7. Documentación**

En el contexto de los sistemas basados en computadora y en software, el término “documentación” tiene significados diferentes para personas distintas. Un documento puede ser una especificación escrita, un conjunto de modelos gráficos, un modelo matemático formal, una colección de escenarios de uso, un prototipo o cualquier combinación de éstos.

Algunos sugieren que, para especificar un documento, se debe desarrollar y utilizar una “plantilla estándar” Argumentan que esto conduce a que los requisitos sean presentados de una manera más consistente y por ende más entendible. Sin embargo, algunas veces es necesario ser flexible mientras se desarrolla una especificación documentada.

Respecto de sistemas grandes el mejor enfoque podría ser un documento escrito que combina descripciones en el lenguaje natural y modelos gráficos. Por otro lado, en cuanto a productos o sistemas más pequeños, podría ser que no se necesite más que escenarios de uso, cuando dichos sistemas residan en ambientes técnicos que se comprendan bien.



La documentación de especificaciones es el producto del trabajo final que genera la ingeniería de requisitos. Sirve como base para las actividades de ingeniería de software subsecuentes.

Describe la función y el desempeño de un sistema basado en computadora y las restricciones que regirán su desarrollo.

#### **a. Gestión de requisitos**

Los requisitos para los sistemas basados en computadoras cambian y que el deseo y necesidad de cambiarlos persiste durante el ciclo de vida del sistema.

La gestión de requisitos es un conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y rastrear los requisitos y los cambios a éstos en cualquier momento mientras se desarrolla el proyecto, la gestión formal de requisitos se inicia sólo para proyectos grandes, los cuales tienen cientos de requisitos inidentificables; en los proyectos pequeños esta función de la ingeniería de requisitos es bastante menos formal. Muchas de estas actividades son idénticas a las actividades de la gestión de la configuración del software.

La gestión de requisitos comienza con la identificación. Cada requerimiento se asigna a un solo identificador. Una vez identificados los requisitos se desarrollan las tablas de rastreabilidad. A continuación se muestra de una manera esquemática, una de las tablas de rastreabilidad, cada una de ellas relaciona los requisitos con uno o más aspectos del sistema o de su ambiente. Entre las muchas tablas de rastreabilidad posibles están las siguientes:

**Tabla de rastreabilidad de las características.** Muestra la manera en que los requisitos se relacionan las características del sistema/producto observables para el cliente.

**Tabla de rastreabilidad de la fuente.** Identifica la fuente de cada requisito.

**Tabla de rastreabilidad de dependencia.** Indica la forma en que los requisitos están relacionados entre sí.

**Tabla de rastreabilidad del subsistema.** Establece categorías entre los requisitos de acuerdo con el (los) subsistemas (s) que gobierna (n).

**Tabla de rastreabilidad de la interfaz.** Muestra la forma en que los requisitos se relacionan con las interfaces internas y externas del sistema.

En muchos casos, estas tablas de rastreabilidad se mantienen como parte de la base de datos de los requisitos de forma que se puedan buscar con rapidez para entender como el cambio en que un requisito afectará diferentes aspectos del sistema que se construirá.

### **1.3.8. Verificación**

Con frecuencia resulta útil examinar cada requisito frente a una serie de preguntas en forma de lista de verificación. Esta es una pequeña lista de preguntas que se pueden tomar como referencia:

- ¿Los requisitos están establecidos de manera clara?, ¿Estos pueden malinterpretarse?
- ¿La fuente del requisito, por ejemplo una persona, una regulación o un reglamento está completamente bien identificado?, ¿El

enunciado final del requisito ha sido examinado por la fuente original o comparándolo con ella?

- ¿Cuáles otros requisitos están relacionados con éste?
- ¿El requisito se puede probar? Si es así, ¿Se pueden especificar las pruebas para ejercitar el requisito?
- ¿El requisito es rastreable para cualquier modelo del sistema que haya sido creado?
- ¿La especificación está estructurada de una forma que conduzca a su comprensión, referencia y traducción fácil en productos de trabajo más técnicos?
- ¿Cuáles requisitos parecen ser implícitos?

### **1.3.9. Validación**

La calidad de los productos de trabajo, procedentes de la ingeniería de requisitos se evalúa durante este paso de validación. La validación de requisitos examina la especificación para asegurar que todos los requisitos de software se han establecido de manera precisa; que se han detectado las inconsistencias, omisiones y/o errores y que éstos han sido corregidos, y que los productos de trabajo cumplen con los estándares establecidos para el proceso, proyectos y productos.

El mecanismo primario para la validación de requisitos es la revisión técnica formal. El equipo de revisión que valida los requisitos incluye ingenieros de software, clientes, usuarios y otros interesados que examinan la especificación y buscan errores en el contenido o la interpretación, áreas que tal vez requieran una clarificación, información faltante, inconsistencias (que es un problema importante cuando se desarrollan productos o sistemas grandes), conflictos entre los requisitos, o requisitos irreales (inalcanzables).

## 1.4. HERRAMIENTAS REPRESENTATIVAS

The Atlantic Systems Guide, Inc., ha preparado una lista razonablemente completa de herramientas para la ingeniería de requisitos, ésta se puede encontrar el <http://www.systemsguild.com> las herramientas que se presentan a continuación se enfocan en la gestión de requisitos:

EasyRM, desarrollado por CyberneticIntelligence GMBH ([www.easy-rm.com](http://www.easy-rm.com)), construye un diccionario/glosario específico del proyecto que contiene descripciones y atributos detallados de los requisitos.

OnYourMark Pro, desarrollado por Omni-Vista ([www.omni-vista.com](http://www.omni-vista.com)), construye una base de datos de los requisitos, establece relaciones entre éstos, y permite a los usuarios analizar la relación entre los requisitos y los calendarios/costos.

RationalRequisitePro, desarrollado por Rational Software ([www.rational.com](http://www.rational.com)), permite a los usuarios desarrollar una base de datos de los requisitos, representa las relaciones entre éstos y los organiza, prioriza y rastrea.

RTM (Requirements Management Tool), desarrollado por Integrad Chipware ([www.chipware.com](http://www.chipware.com)), es una herramienta para la descripción y rastreabilidad de requisitos que también soporta ciertos aspectos del control del cambio y gestión de pruebas.

Umbrello es una herramienta libre para crear y editar diagramas UML, que ayuda en el proceso del desarrollo de software. Fue desarrollada por Paul Hensgen, y está diseñado principalmente para KDE (K Desktop Environment), aunque funciona en otros entornos de escritorio<sup>13</sup>.

---

<sup>13</sup><http://es.wikipedia.org/wiki/Umbrello>

Umbrello maneja gran parte de los diagramas estándar UML pudiendo crearlos de manera manual o también importándolos a partir de código en C++, Java, Python, IDL, Pascal/Delphi, Ada, o también Perl (haciendo uso de una aplicación externa). Así mismo, permite crear un diagrama y generar el código automáticamente en los lenguajes antes citados, entre otros. El formato de fichero que utiliza está basado en XML (eXtensible Markup Language).

Se debe hacer notar que muchas tareas de la gestión de requisitos se pueden realizar con una simple hoja de cálculo o un sistema pequeño para el manejo de bases de datos.

## **1.5. METODOLOGÍA DE DESARROLLO DE SOFTWARE**

En ingeniería de software, una metodología es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información.

Una metodología de desarrollo de software se refiere a un conjunto estandarizado de conceptos, prácticas y criterios que sirven para enfocar un tipo de problemática en particular que luego es tomado como referencia para enfrentar y resolver nuevos problemas en que se usan dichos conjuntos estandarizados para planear y controlar el proceso de desarrollo de nuevos sistemas de información.

A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. La estructura digital para una metodología de desarrollo de software consiste en:

Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software.

Herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

Estas estructuras digitales son a menudo vinculadas a algún tipo de organización, que además de desarrollar, apoya el uso y promueve dicha metodología. Estas metodologías son almacenadas en algún tipo de documentación formal y/o estandarizada.

Las Metodologías de Desarrollo de Software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelamiento de sistemas que permitan desarrollar software de calidad.

#### **1.5.1. Programación Extrema (XP)**

La programación extrema o *eXtremeProgramming* (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

##### **a. Descripción de Extreme Programming**

Extreme Programming (Programación Extrema), es una metodología ágil de desarrollo de productos de software, que se focaliza en la satisfacción

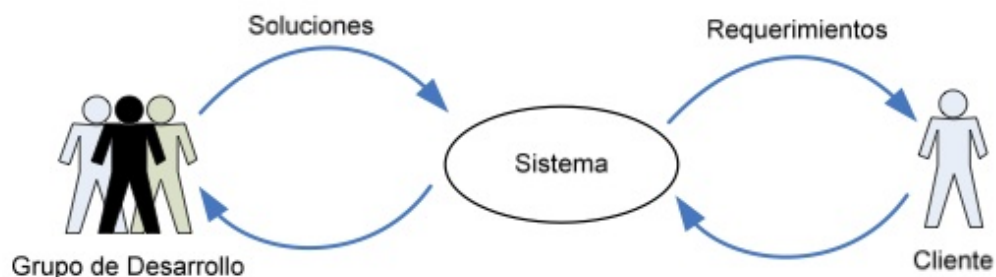
del cliente, manteniendo como doctrina “el entregar el software que el cliente necesita cuando lo necesita”.

El trabajo en conjunto es uno de los pilares de Extreme Programming, donde el cliente forma parte del equipo de desarrollo, junto con los programadores, comprometiéndose a obtener un producto de calidad, disminuyendo de forma notable el tiempo invertido en las pruebas de aceptación del sistema.

El desarrollo de software usando XP (Extreme Programming), es fortalecido con cuatro aspectos importantes que son:

- Comunicación
- Simplicidad
- Valor
- Retroalimentación

XP es una metodología adecuada para equipos pequeños de desarrollo, y no requiere de personas expertas en un área en específico, dirigida a mitigar los riesgos del desarrollo, debido a que el sistema a desarrollar está de acuerdo a las necesidades cambiantes de un cliente.



**Figura 1.4: Desarrollo de un Sistema usando XP**

Una de las ventajas de XP es realizar entregas funcionales del sistema, por lo cual el grupo de desarrollo emplea prototipos evolutivos, y mediante un test de funcionalidad el cliente acepta la entrega o plantea cambios.

“Es una metodología de acción antes que una metodología de control; a diferencia de las metodologías tradicionales, XP no obliga a los miembros del equipo a tener demasiados documentos, restricciones y actividades que a lo largo del proceso de desarrollo, muchas veces incrementan el tiempo del mismo”.<sup>14</sup>

### **b. Historias de usuarios**

Las Historias de Usuarios son tarjetas escritas por el cliente en un lenguaje natural para este, indicando las necesidades que el sistema debe satisfacer. Además de describir las necesidades, las historias de usuario también tienen la finalidad de manejar la creación de las pruebas de aceptación, de tal forma que al concluir una historia se verifique que esta ha sido llevada a cabo correctamente por el grupo de desarrollo.

Uno de los errores comunes cometidos con las historias de usuario es que son confundidas con requerimientos tradicionales, diferenciándose estos en el nivel de detalle, una historia de usuario debe proporcionar el detalle necesario como para estimar el riesgo y el tiempo que toma llevar a cabo dicha historia, la ventaja es que el programador recibe una descripción detallada de los requerimientos de una forma personal.

### **c. Características de XP**

La principal característica es que incentiva la programación, de tal forma que crea un ambiente ideal para trabajar en equipos pequeños de van desde 2 o más integrantes por grupo de desarrollo, siendo las bases de la

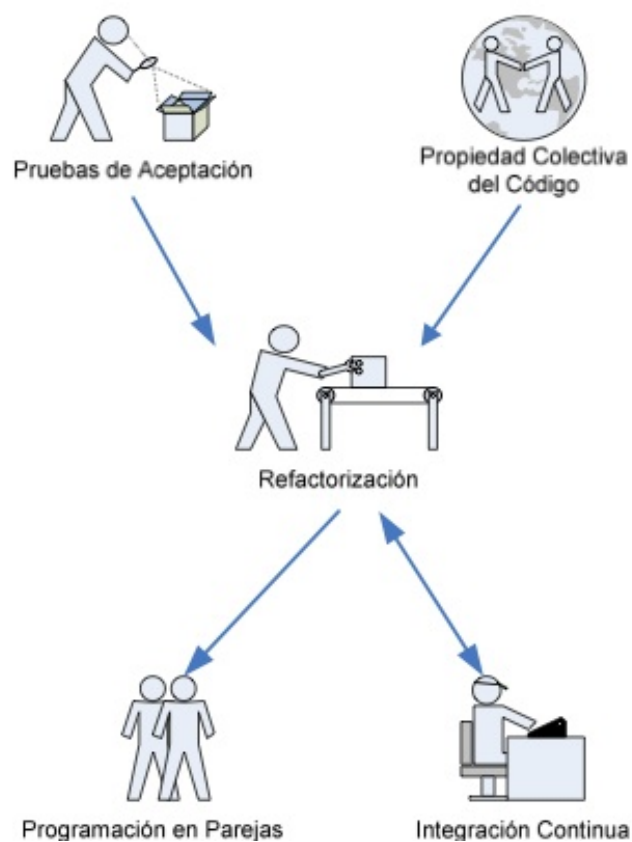
---

<sup>14</sup>DAPENA, José. Desarrollo en Comunidad con eXtremeProgramming. Italia



programación extrema: la Programación en Parejas, Propiedad Colectiva del Código, Pruebas de Aceptación, Refactorización e Integración Continua.

Estas bases permiten a los desarrolladores participar de una forma activa en el desarrollo, independientemente de quien programó y de cómo lo hizo.



**Figura 1.5: Bases de eXtreme Programming**

### **a.1. Programación en parejas (PairProgramming)**

Extreme Programming propone la opción de dividir al grupo de desarrollo en pares, estos tendrán un solo computador, el primer individuo será el encargado de buscar la mejor forma de satisfacer la necesidad del cliente,

mientras que el segundo tiene una tarea más estratégica, debido a que tiene que idear las pruebas para el módulo que están desarrollando, ver que pruebas son factibles y cuáles no, buscar la forma de simplificar el sistema sin dejar a un lado la funcionalidad.

### **a.2. Pruebas de Aceptación (AcceptanceTests)**

“Cualquier parte funcional del sistema que no tenga una prueba simplemente no existe”.<sup>15</sup> XP propone diseñar las pruebas, previo a la construcción de un módulo, de tal forma que sirva de criterio de aceptación de la misma, además de ser una base para detectar mejoras en la funcionalidad de sistema con la finalidad de satisfacer las necesidades del cliente. Es preciso diferenciar que las pruebas de unidad que propone XP son muy diferentes de las pruebas de sistema que emplean otras metodologías.

### **a.3. Integración Continua**

Una vez concluidas las pruebas unitarias se procede a integrar los módulos, es recomendable hacer uso de una sola máquina que servirá de repositorio para las partes del sistema, la pareja hace uso de esta máquina para anexar su parte y debe analizar que no entre en conflicto con las demás.

### **d. Fases de la metodología XP**

Las fases del ciclo de vida de un proceso XP son: Exploración, Planificación, Iteraciones, Producción, Mantenimiento y Muerte del proyecto.

---

<sup>15</sup>BECK, Kent. Extreme Programming Explained: Embrace Change. Primera Edición. Addison – Wesley. Estados Unidos de América. 2000.



## Extreme Programming Project



Figura 1.6: Proceso de eXtreme Programming

### d.1. Exploración

En la fase de exploración los desarrolladores deben familiarizarse con las herramientas que van a trabajar, además de aprender a trabajar entre sí. Durante la exploración, los programadores usan herramientas de desarrollo que luego serán manejadas en la producción del sistema, además exploran activamente las posibilidades para la arquitectura del sistema.

Los programadores también deben experimentar con los límites de la tecnología que están usando. Deben realizar simulaciones con cargas realísticas con el hardware y la red. Mientras el equipo de desarrollo está encargado de la tecnología, el cliente debe practicar la escritura de las historias. La clave es conseguir una retroalimentación rápida por parte de los clientes con las primeras historias para que aprendan a especificar lo que los programadores en verdad necesitan.

## **d.2. Planificación**

El propósito de esta fase es para los clientes y programadores estar de acuerdo en una fecha en la cual el conjunto de historias más pequeño y representativo va a ser realizado. La planificación, debe abarcar todas las historias a ser implementadas y repartidas en las iteraciones que se decidan efectuar.

La esencia de la planificación es para el equipo de desarrollo, estimar el tiempo en semanas que tarda en realizar cada historia de usuario, incluidas las pruebas. El cliente y los diseñadores deciden en conjunto qué historia es la más importante o tiene la prioridad más alta a ser completada.

## **d.3. Iteraciones**

Esta fase requiere tener el Plan de Entrega definido, el cual contiene un conjunto de iteraciones y cada iteración tiene un conjunto de historias asignadas. Cada iteración origina un conjunto de casos de prueba funcionales para cada una de las historias fijadas para esa iteración.

En la primera iteración, son escogidas las historias que obligan a que "el sistema completo" sea construido, aun cuando sea solamente la estructura. Las historias para las iteraciones subsecuentes están completamente en la discreción del cliente. Al final de la última iteración, es el momento para entrar en la producción.

## **d.4. Puesta en Producción**

En la fase de producción las impresiones del producto elaborado son recogidas, con el fin de poder mejorarlo en las próximas entregas,

permitiendo así añadir características adicionales a la versión actual. XP propone realizar entregas funcionales del sistema, estas están listas para ser integradas al sistema una vez que hayan aprobado las pruebas unitarias, pero esto no significa que al momento de integrarlo con los distintos módulos, este funcione de forma correcta, por esta razón es necesario realizar pruebas al momento de integrar el sistema.

En estos casos es indispensable utilizar pruebas en paralelo de todas las partes que componen el sistema. Una vez cumplidas estas pruebas es posible certificar que el sistema puede entrar en producción. El diseño de estas pruebas ayuda al equipo de trabajo a poner a punto al sistema, y gracias a estas el grupo será capaz de estimar de una mejor forma el tiempo y el riesgo que involucraría realizar los cambios en el sistema.

Durante esta etapa el equipo de trabajo baja el ritmo del desarrollo, sin que esto implique que el sistema deje de evolucionar, tomando en cuenta que mientras más sea la experiencia que tenga el grupo de trabajo en el sistema, mayor será su visión al momento de afrontar los cambios requeridos, añadiendo estos cambios a la entrega presente o a futuras entregas.

#### **d.5. Mantenimiento**

El mantenimiento es el estado normal de un proyecto de XP. Es necesario simultáneamente, producir la nueva funcionalidad, mantener el sistema existente corriendo e incorporar nuevas personas en el equipo.

#### **d.6. Muerte del Proyecto**

Un proyecto llega a su final cuando el cliente no aporta con más historias de usuario para que el grupo de trabajo las implemente. Con esto puede

darse por concluido el desarrollo del sistema, y es posible decir que el sistema ha alcanzado toda la funcionalidad que el cliente deseaba. Otra razón para que un proyecto concluya es la falta de recursos por parte del cliente, ocasionando que el proyecto sea abandonado.

## **e. Roles de XP**

Para el correcto funcionamiento de un equipo de trabajo es necesario tener definidas las responsabilidades que deben cumplir cada uno de sus miembros. Un equipo XP consta de los siguientes roles:

### **e.1. Programador**

Es el encargado de realizar los programas y las pruebas respectivas. Pero para que esto funcione de verdad debe existir una comunicación con las demás personas del equipo de trabajo.

Un programador debe tener habilidades como:

La comunicación para poder coordinar y tener éxito

La simplicidad tanto en el manejo del programa como en el código que genera.

### **e.2. Cliente**

Él sabe lo que el programador debe realizar y esto lo refleja en los requisitos donde detalla sus necesidades. Además debe trabajar estrechamente con el equipo de trabajo para poder ayudar de la mejor manera con las pruebas del sistema.

### **e.3. Tester**

Debido a que las pruebas son parte de la responsabilidad de los programadores, el tester debe centrarse en el cliente, apoyarlo a escribir las pruebas de funcionamiento del sistema, ejecutar las pruebas y publicar estos resultados. Con lo que es posible verificar que el producto final funcione correctamente.

### **e.4. Entrenador**

Es el responsable del proceso en sí. Está encargado de mantener la calma en el equipo en momentos de presión y también de tomar decisiones cuando ocurre algún error en el diseño.

### **e.5. Consultor**

Hay ocasiones en las que el equipo necesita profundos conocimientos técnicos y cuando esto ocurre es necesario de un consultor. El objetivo de un consultor es enseñar al equipo a resolver sus problemas.

### **e.6. Big Boss**

El bigboss es la persona encargada de mantener la comunicación entre los miembros del equipo de trabajo.

## **1.6. DISEÑO WEB Y ESTÁNDARES**

Una de las características que hacen popular el diseño de sistemas web es la facilidad en mostrar los contenidos de forma gráfica, además de permitir vincular los documentos de interés al sistema y ofrecer servicios que agilizan el trabajo del personal de la empresa.

El equipo de desarrollo cumplirá con ciertas características de publicación, lo que permitirá conseguir lo siguiente:

- Que las páginas se desplieguen rápidamente.
- Que las páginas sean visualizadas de la misma forma en la que se las diseña.

Para cumplir con lo anterior, los programadores harán uso de las buenas prácticas obtenidas de la experiencia de los profesionales del desarrollo web.

### **1.7. GUIA WEB - MODELO CLÁSICO**

XP es una metodología enfocada más en la acción que en el control, su éxito radica en la comunicación entre el grupo de desarrollo, incluyendo al cliente como miembro del mismo.

El desarrollo inicia con la recolección de las historias de usuario, estas son organizadas por prioridad en el plan de entrega. Una vez establecido el plan de entrega el grupo de desarrollo procede a implementar el requerimiento, el cual va a ser evaluado por el tester y el cliente, siendo este último el que constatará los resultados de la prueba, dando sus recomendaciones o nuevos requerimientos de acuerdo a la lógica del negocio, de tal forma que se pueda continuar con la siguiente iteración.

Al momento de diseñar el sistema, el grupo de trabajo hará uso de la Guía de Desarrollo para sitios Web, esta guía ofrece un conjunto de buenas prácticas y estándares que se cumplirán de acuerdo a la necesidad y a los requerimientos que los usuarios deseen añadir al sistema con una interfaz web.



Para poder llevar a cabo tanto la metodología como la guía, se hará uso de las etapas Análisis, Diseño, Pruebas, Implantación del modelo clásico, que permitirá realizar una correcta combinación entre la Metodología de Desarrollo Extreme Programming con la Guía de Desarrollo para sitios Web, distribuyendo en cada etapa del modelo clásico las fases del ciclo de vida de XP, a la vez aplicando las buenas prácticas de desarrollo Web al momento de realizar el diseño del sistema.

Los entregables para cada etapa de desarrollo serán los siguientes:

### **Análisis**

Plan de entregas

Plan de iteraciones

### **Diseño**

Diagramas de interacción de los usuarios con el sistema

### **Implementación**

Seguimiento y ejecución de las iteraciones

Base de datos

Prototipo de interfaces

### **Pruebas**

Pruebas con los requisitos obtenidos.

Pruebas de integración

Evaluación de resultados

### **Implantación**

Hospedaje definitivo del sitio

Pruebas de aceptación del sistema

Pruebas de integración

Pruebas de compatibilidad de navegadores

## **CAPÍTULO 2**

### **FASES DE LA INGENIERÍA DE REQUISITOS**

#### **2.1. GENERALIDADES**

En un escenario ideal, los clientes y los ingenieros de software trabajan juntos en el mismo equipo. En tales casos, la ingeniería de requisitos se trata sólo de guiar conversaciones significativas con colegas que son miembros bien conocidos del equipo. Sin embargo, en la vida real las cosas son muy diferentes, es por esa razón que se aplica una serie de métodos para obtener los requerimientos o necesidades del problema, aquí se exponen de manera ordenada las cinco fases de la ingeniería de requisitos que usualmente se usan en la ingeniería de software.

#### **2.2. FASES DE LA INGENIERÍA DE REQUISITOS**

##### **2.2.1. Obtención de requisitos**

En el caso de Radio Majestad de la ciudad de Quito D.M. para el sistema de automatización del audio en vivo se realizó las siguientes formulaciones de las primeras preguntas:

- ¿Quién está detrás de la solicitud de este proyecto?
- ¿Quién usara la solución?
- ¿Cuál será el beneficio económico de una solución exitosa?

Estas preguntas ayudaron a identificar a todos los participantes que tienen interés en el desarrollo del software, es decir todos los principales involucrados de la estación de radio. Además, estas preguntas

permitieron identificar un beneficio medible de una implementación a futuro exitosa y las mejores alternativas para el desarrollo del prototipo.

Las respuestas a las preguntas fueron las siguientes:

El gerente general de la estación de radio, al igual que el director de programación de la radio, son los más interesados en el desarrollo del software.

El departamento de ventas usara la solución para poder tener una guía en la elaboración de facturas al igual que los locutores de la estación. En esta pregunta lo más interesante es que ellos no miden la parte económica sino la auditoria de los audios que se pasan al aire.

La siguiente serie de preguntas permitió que el equipo de desarrollo de software (Diego Pule) comprenda mejor el problema y dejo que el cliente (Radio Majestad) exprese sus percepciones acerca de la mejor solución:

- ¿Cómo se podría medir un buen resultado con el desarrollo de un prototipo exitoso?
- ¿Cuáles son los problemas que debería de atacar la presentación de la solución, entrega del prototipo?
- ¿Cómo se podría mostrar el ambiente de negocios en el que se probara el prototipo?

Las respuestas a las interrogantes planteadas fueron:

El gerente comentó que un buen indicador sería la falta de quejas por parte del director de programación que muchas veces no tiene control sobre la lista de reproducción de música; la gerente de ventas comentó que sería mas fácil para una pequeña auditoria sobre los comerciales que se van a pasar al aire.

Uno de los problemas mas complejos es la fiabilidad (la probabilidad de que un sistema funcione o desarrolle una cierta función, bajo condiciones fijadas y durante un período determinado) del sistema, el software anterior o el que se usa actualmente llamado Jazler<sup>16</sup> radio, es muy complejo y difícil de manejar ya que contiene varias funciones innecesarias.

Para la evaluación del prototipo del sistema de automatización de radio Majestad, en el ambiente de negocios se lo puede probar con una computadora conectada al sistema el cual registre el audio que se envía en vivo.

Más que medir un buen resultado se espera que el prototipo o el sistema completo sea seguro, fácil de usar, fiel, y que satisfaga los requisitos obtenidos.

Esta serie final de preguntas se enfocó en medir el rendimiento en la actividad de comunicación de la ingeniería de software; se las conocen como “Meta preguntas”.

- ¿Las preguntas que estoy realizando son importantes para solucionar el problema?
- ¿Estoy haciendo demasiadas preguntas?
- ¿Alguien más puede proporcionar información adicional?
- ¿Alguna pregunta que usted crea que se me paso por alto?

Estas preguntas ayudaron a “romper el hielo” y a realizar una conversación esencial para una obtención exitosa de una sola reunión.

---

<sup>16</sup><http://www.jazler.com/>

### **a. Identificación de los interesados**

En esta subetapa del inicio se logro identificar a todos los sospechosos usuales y la mayor parte de los involucrados que se benefician directa o indirectamente, al inicio solo encontramos al gerente de la radio, pero a medida que se realizaba la pregunta de rigor: ¿A quien mas debo preguntar?, la lista de involucrados creció, obviamente cada interesado tiene una visión diferente del sistema, obteniendo beneficios diferentes cuando éste se empiezo a desarrollar de manera exitosa.

Aquí se elaboró una lista de las personas que contribuyeron en la obtención de requisitos:

- Gerente de la radio – Fabián Custode (El cargo es definir la programación de la estación general, maneja el sueldo de los empleados).
- Jefe de Ventas – María Augusta Viteri (Vendedora de la estación, se encarga de cobrar a los clientes, coordinar las listas de reproducción de comerciales, pago de comisiones, promociones, y coordinación de ventas con otros vendedores).
- Director de Programación – Jimmy Pavón (Encargado de realizar las listas de reproducción de música, se especializa en que las canciones no se repitan, coloca promocionales, coloca las mejores canciones de la semana, da prioridad a artistas nacionales).
- Locutor – Iván Calderón.
- Operador – Miguel Morales.

### **b. Reconocimiento de múltiples puntos de vista**

Debido a que existen muchos “clientes” diferentes, los requisitos se exploraron desde diferentes puntos de vista.

El gerente estaba interesado en que el sistema funcione, desarrolle la automatización (el funcionamiento automático de las listas de comerciales y de música de manera autónoma), bajo las condiciones fijadas y durante un período determinado del sistema; es decir que el sistema realice todo y no se “cuelgue””, y además que él pueda “visualizar” todo lo que hacen en el sistema.

Jefe de ventas solo estaba interesado en que el sistema elabora facturas con el mayor detalle posible, y que cumpla todos los requisitos que el dueño de la radio lo exige (el gerente de la radio no es el dueño).

Director de programación estaba interesado solo en las listas de música, que se reproduzca en orden y sin problemas, que se pueda colocar cada día los archivos precisos en línea.

Locutor, considera que lo único importante para él, es la rapidez del sistema y que ofrezca la mayor cantidad de detalles sobre el tema a colocar, así como también y que disponga de toda la base musical para siempre.

El portero de la estación, lo que más le interesa es que cuando se presente algún altercado con el suministro de energía eléctrica ya sea en los estudios o en el transmisor, éste sea lo más fácil de recuperar,

Pero en lo que la mayoría coincidía fue en el hecho que el sistema sea lo más sencillo de usar y que todos se encuentren familiarizados con el mismo, y algo importante que se encuentre en internet, y finalmente como es de esperar, al gerente le interesaba que sea lo más económico posible en cuanto a su mantenimiento.

Conforme se recopiló la información desde múltiples puntos de vista, algunos requisitos llegaron a ser inconsistentes otros muy ambiciosos y hasta entraron en conflicto unos con otros. El trabajo de la presente tesis fue categorizar toda la información recopilada de los interesados (incluido los requisitos inconsistentes, ambiciosos y conflictivos), por ejemplo:

El locutor quería un buscador mediante el cual encuentre lo más rápido posible una canción, por lo que este archivo de audio, debería tener metadatos con la información de la letra de la canción, año de publicación del álbum, número de pista, autor, y hasta la portada del disco, a más de ello necesitaba que el buscador sea 100% igual al buscador del programa de iTunes<sup>17</sup>, pero el director de programación omitió todo eso ya que estaba interesado en que el sistema lea una lista de reproducción de dicho programa.

### **c. Trabajo con respecto a la colaboración**

Lo más complicado en el inicio de la obtención de requisitos es la colaboración entre los involucrados ya que se presentan pequeñas peleas sin sentido, para obtener esta tan valiosa colaboración identificamos áreas en común (es decir, los requisitos en los que todos los interesados están de acuerdo) y áreas de conflicto o inconsistencia (esto es, los requisitos solicitados por un interesado que entran en conflicto con la necesidades de otro), Ésta es, por supuesto, la última subcategoría que presento para la presente tesis una gran desafío.

---

<sup>17</sup> Es un reproductor de medios multimedia desarrollado por Apple con el fin de reproducir, organizar y sincronizar iPods, iPhones, iPad. Es compatible con ordenadores basados en sistemas operativos Mac OS X, Windows 2000, Windows XP, Windows Vista y Windows 7.  
<http://es.wikipedia.org/wiki/iTunes>

La mejor forma de resolver los conflictos entre los requerimientos, y al mismo tiempo que se entienda la mayor importancia con relación a todos fue un pequeños esquema de votación basado en puntos de prioridad, estos “puntos”, se entregaron a cada uno de los involucrados los cuales los usaron según su criterio; una vez que los puntos de prioridad del interesado se han agotaron, dicho cliente no puedo realizar ninguna otra acción sobre sus requerimientos. Los puntos totales que asignen a cada requisito todos los interesados indican la importancia general de cada requisito.

A cada involucrado se le dio puntos de prioridad para cada uno de sus requisitos en formato de estrellas entendiéndose que a tres estrellas el requisito tiene mayor prioridad, y con menos estrellas o ninguna que requiere menos atención.

El formato de preguntas y respuestas que se describió en la sección anterior es útil para romper el hielo en una etapa inicial, pero no es un enfoque que haya sido exitosa de manera decisiva para una obtención de requisitos más detallada, de hecho estas preguntas y respuesta solo se usaron en un primer encuentro y luego se las reemplazaron por un formato de obtención de requisitos que combinaron elementos de la resolución elaborados para la negociación y especificación del problema.

#### **d. Recopilación Conjunta De Requisitos**

Para estimular un esfuerzo conjunto y orientado a la presente tesis en la recopilación de requisitos con el equipo de participantes y el desarrollador, se propuso varios elementos de solución, negociando diferentes enfoques y especificando todo un conjunto preliminar de requerimientos para la solución.



Se propusieron muchos y diferentes enfoques para la recopilación conjunta de los requisitos, para cada uno se uso un escenario muy diferente, pero se aplicaron variaciones sencillas en directrices generales, las normas generales que se tomaron en cuenta para el trayecto de la presente tesis fueron las siguientes:

Las reuniones fueron dirigidas, por el egresado Diego Pule, y con cada uno de los participantes, respetando los horarios de trabajo.

Una de las reglas fue que cada participante entregue una o más hojas describiendo cuáles eran sus necesidades y justificaciones sobre el sistema a desarrollar.

Se estableció una agenda de dos semana en las cuales los participantes se comprometieron a entregar las ideas de una maneja seria y formal, pero también muy informal en las reuniones para estimular el flujo de ideas.

El mecanismo conjunto para aclarar las ideas fueron, hojas de trabajo, algunos gráficos, comparación de algunos sistemas existentes en el mercado, hojas adheribles, y el correo electrónico [diegopule@gmail.com](mailto:diegopule@gmail.com)

La meta final fue identificar el problema de forma global y abstracta para luego desarrollar el prototipo aplicando la metodología Programación Extrema.

Para comprender mejor el flujo de eventos y sucesos conforme van ocurriendo y se van presentando en secuencia, se presentaron dos eventos que condujeron a dos reuniones finales que sirvieron para la recopilación de requisitos que ocurrieron de la siguiente manera:

Primera reunión. Se idealizó un evento en el cual el prototipo estuviera funcionando al 100%, durante una semana sin problemas externos.

Segunda reunión. Se idealizo un evento en el cual el prototipo estuviera funcionando al 100%, durante 24 horas con dos problemas externos mas comunes.

Durante la fase de inicio las preguntas y respuestas básicas establecen el ámbito del problema y la percepción global de la solución, luego de estas reuniones se les pidió a los involucrados como tarea que entreguen una “solicitud del sistema” de una o de dos páginas y siguiendo el siguiente calendario para la reunión con la respectiva tarea.

**Tabla 2.1. Cronograma de reuniones**

Semana de reunión en Febrero 21 al 26 2011 con los involucrados					
Cliente	Lunes	Martes	Miércoles	Jueves	Viernes
Gerente	x	x	x	x	x
Ventas		x		x	
Programador			x	x	
Locutor A	x				x
Portero		x			x

El formato de la solicitud del sistema que se lo repartió a todos los involucrados días antes de las reuniones tenía el siguiente formato, se les pidió realizar una lista de objetos que son parte del ambiente que rodea al sistema, otros objetos que este producirá, y objetos que realiza para realizar sus funciones. Además, se le pidió a cada asistente que elabore una lista de los servicios (procesos o funciones) que manipulen o interactúen con los objetos. Por último también se preparo una pequeña lista de posibles restricciones referentes a costo, tiempo seguridad, reglas

del negocio y criterios de rendimiento con referencia velocidad, exactitud, fiabilidad y seguridad. Se les pidió a todos los involucrados, 5 en total: Gerente, jefe de ventas, programador, locutor, portero que las listas no sean exhaustivas, sino que reflejen la percepción que cada uno tiene del sistema a desarrollar.

**Tabla 2.2: Solicitudes del sistema**

Solicitud del sistema	
Cargo del involucrado:	
Actividad detallada del cliente:	Objetos:

Si un sistema o un producto servirán a muchos usuarios se debe tener la completa seguridad de que los requisitos se obtienen de una muestra representativa de los usuarios. Si solo uno de los usuario define todos los requisitos el riesgo de rechazo del sistema final es muy alto, es por esa razón que en el formato de solicitud del sistema se pide un percepción final del sistema funcionando.

En realidad, muchos de los colaboradores ayudaron con mas información durante las reuniones y mucha mas de esta información estuvo disponible, pero con la información adicional disponible la ambigüedad y las criticas estuvieron presentes es por esa razón que para la elaboración del prototipo el equipo de desarrollo se rigió directamente a las hojas de solicitud del sistema no sin tomar en cuenta que existieran errores e inconsistencias pero hasta ese momento las hojas de solicitud son suficientes.

El equipo completo de recopilación de requisitos lo comprendieron los cinco involucrados y el equipo de desarrollo de software Diego Pule, cada persona desarrollo las listas de solicitud antes mencionadas. Los objetos descritos para MajeSoft<sup>18</sup> incluyeron: panel de control, automatización, dos pc, tarjeta de audio, licencia de software, eventos finales por ejemplo: corte del suministro de corriente eléctrica, listas de mp3. La listas de servicios también incluyeron la configuración del sistema, la colocación de puesta en marcha para el audio en vivo, auditoria, entrega de un reporte de errores, de archivos colocados al aire, monitoreo del sistema, también se presentaron las listas con sus restricciones respectivas (por ejemplo: El sistema debe de reconocer quienes ingresaron al sistema, cuáles fueron los errores, debe de ser amigable al usuario con una interfaz orientada a la web) y también criterios de rendimiento, por ejemplo la carga de la lista de música debe de estar lista para su reproducción en el menor tiempo posible, en las dos últimas reuniones se priorizaron los eventos.

En la primera reunión realizada el día lunes 28 de febrero del 2011, para la recopilación de documentos, el primer tópico que se tomo en cuenta fue la necesidad y la justificación para el desarrollo del prototipo, todos los involucrados estuvieron de acuerdo en la necesidad de un nuevo sistema de automatización, el anterior software utilizado era el sistema de radio automatización llamado Jazler Radio<sup>19</sup>, el cual presentaba algunas limitaciones y restricciones para cada uno de los involucrados, una vez que nos pusimos de acuerdo de la necesidad de un nuevo sistema y su justificación se colocaron en un pizarrón las principales ideas. En esta la crítica y el debata estuvieron completamente prohibidos.

---

<sup>18</sup>Nombre asignado al prototipo del sistema de aquí en adelante en el presente trabajo se llamara así al "Prototipo del sistema de automatización del audio en vivo de Radio Majestad"

<sup>19</sup><http://www.jazler.com/Downloads/JazlerIIManual-Spanish.pdf>

Después de que las listas individuales sobre el área de un tópico se presentaron, el grupo creó una lista combinada, dicha lista elimina los asuntos redundantes, se incorporó ideas nuevas que surgieron durante el debate, pero no se eliminó nada de nada. Después de haberse elaborado la lista combinada se organizó y pulió de mejor manera la lista de requisitos para que se pueda reflejar de una manera apropiada el sistema que se desarrollara. El objeto final en este punto es crear una lista condensada en un área específica: objetos, servicios, restricciones, rendimiento, seguridad. Luego estas listas se las integro para siguientes acciones.

Cuando se completaron las listas ya condensadas el equipo se dividió en sub-equipos menores (desarrollador - involucrado), cada uno trabajo para desarrollar “casos de uso”<sup>20</sup> para uno o más asuntos de cada una de estas listas.

Estas son las listas que se condensaron por los interesados:

#### Lista 1 Gerente de la estación

- Controlar todo desde el internet
- Tener acceso a las listas de comerciales y canciones que se van a pasar a continuación.
- Poder ver la lista de comerciales que se van a incluir
- Tener a disposición la lista de canciones que el director de programación elaboro para que el sistema de automatización las realice.

---

<sup>20</sup>Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes que participarán en un caso de uso se denominan actores

## Lista 2 Jefe de Ventas

- Tener completo control sobre la colocación de comerciales en la lista de reproducción
- Poder administrar de manera fácil la lista de comerciales
- Tener un reporte de todos los comerciales que se pasaron
- Acceder de manera rápida a la lista de comerciales
- Colocar identidades a cada uno de los comerciales para

## Lista 3 Director de programación

- Buscador fácil de canciones.
- Guardar la lista de reproducción para una futura auditoria de canciones.
- Que dicho buscador lo realice dentro de si es posible las letras de las canciones

## Lista 4 Locutor de la estación

- Desea la mayor información posible de la canción que va a sonar a continuación
- Tener a disposición la lista anterior que el anterior locutor paso las canciones y de ser posible las del día de mañana
- Tener acceso a la lista de canciones y poder reemplazarlas de manera rápida

## Lista 5 Operador

- Confiable es decir que no se cuelgue
- Al tener algún problema con el suministro energético que sea fácil la reincorporación de las funciones

### **2.2.2. Analizar Requisitos**

Cuando se complementaron las listas ya condensadas, cada persona involucrada se reunió con el desarrollador del proyecto para trabajar en conjunto en la elaboración de mini especificaciones para cada uno de los asuntos de las listas. Cada una de estas mini especificaciones es una explicación muy concisa y muy bien analizada de cada requerimiento de las listas ya elaboradas.

#### **Análisis de los requerimientos de las listas elaboradas por los involucrados y el desarrollador.**

##### **Gerente de la estación - desarrollador**

El sistema de automatización tiene que tener un dominio por ejemplo: [www.radiomajestad.com](http://www.radiomajestad.com) el cual solo las persona con la clave de acceso pueda ingresar, cualquier persona puede “ver” la pagina de acceso pero solo los usuarios registrados podrán ingresar a dicha pagina, para el desarrollo del prototipo se tomo en cuenta la dirección local electrónica [www.localhost/Radio](http://www.localhost/Radio)

##### **Jefe De Ventas – Desarrollador**

Manejo sencillo de la exploración de archivos, creación de paquetes publicitarios todo en un ambiente tipo Web, navegador, manejo de ordenes publicitarias, ambiente parecido a las ordenes enviadas desde las agencias de publicidad.

## **Director de programación – desarrollador**

Fácil búsqueda de los archivos dentro del disco duro del sistema, creación de la lista de manera fácil, información de la canción, para la vista del locutor

## **Operador – desarrollador**

Sea de fácil acceso tipo “Internet” es decir de forma Web, reposición fácil en caso de algún evento catastrófico como falta de fluido eléctrico.

### **2.2.3. Documentar los requisitos**

Cuando se terminó la materialización de los requisitos con las mini especificaciones y ya con una visión general del sistema, fue conveniente en esta etapa del desarrollo, documentar los requisitos, para esto se creó escenarios de usuario o casos de uso para cada una de estas mini especificaciones adjuntando la respectiva documentación, pero también fue necesario crear los archivos respectivos para la ingeniería de requisitos esto se logró gracias al software informático de la empresa Rational, llamado RationalRequistePro<sup>21</sup>, el desglose para la documentación de requisitos usando software de la Rational.

A continuación se detallan los documentos Word utilizados en la herramienta RationalRequisite Pro para declarar los requisitos software, es decir, el documento visión, el documento glosario y tanto las especificaciones de los casos de uso incorporados hasta la segunda iteración de la fase de construcción, además de las especificaciones de casos de prueba de los casos de uso incorporados hasta la última iteración de la fase de construcción.

---

<sup>21</sup><http://www-01.ibm.com/software/awdtools/reqpro/>



Para mayor detalle de los documentos elaborados se encuentran en la parte de anexos de la presente tesis:

Documento Glosario. (VER ANEXO A)

Documento Visión. (VER ANEXO B)

Las especificaciones de casos de uso están divididas según el subsistema al que pertenezcan, atendiendo a los subsistemas definidos en el documento Visión. Los documentos se encuentran en la sección de anexos de la presente tesis:

A continuación se presenta la documentación gestionada mediante el programa de Rational, el Requisite Pro para la gestión de requerimientos. En Requisite Pro están definidos: requisitos, matrices de atributos y matrices de trazabilidad para hacer un seguimiento de un proyecto.

#### **a. Stakeholders**

Los representantes de los usuarios y portavoces de las necesidades de la empresa son los stakeholders. En el presente proyecto solamente se ha tratado con un stakeholder como representante de los usuarios y necesidades de la estación, sin embargo se han dividido representativamente según los distintos departamentos.

**Tabla 2.3. Matriz de atributos de los stakeholders**

Requirements:	Prioridad	Estado	Dificultad	Representante	Location
<b>STK1: GERENCIA</b>	Alta	Aprobado	Media	Diego Pule	D. Visión
<b>STK2: VENTAS</b>	Alta	Aprobado	Alta	Diego Pule	D. Visión
<b>STK3: PROGRAMACIÓN</b>	Alta	Aprobado	Alta	Diego Pule	D. Visión
<b>STK4: PROGRAMAS</b>	Media	Aprobado	Media	Diego Pule	D. Visión
<b>STK5: MANTENIMIENTO</b>	Media	Propuesto	Media	Diego Pule	D. Visión
<b>STK6: RESPALDO</b>	Alta	Propuesto	Media	Diego Pule	D. Visión
+ <Click here to create a requirement>					Database

### b. Actores

Se define este requerimiento para listar los usuarios potenciales del sistema, en este proyecto se han definido los siguientes actores dentro del programa de Rational: Gerente, Jefe de ventas, Director de programación, técnico operador.

**Tabla 2.4. Matriz de atributos para los actores**

Requirements:	Prioridad	Estado	Dificultad	Location	Subsistema
<b>ACT1: GERENTE</b>	Alta	Aprobado	Alta	D. Visión	Gerencia
<b>ACT2: JEFE DE VENTAS</b>	Alta	Aprobado	Alta	D. Visión	Ventas
<b>ACT3: DIRECTOR DE PROGRAMACIÓN</b>	Alta	Aprobado	Alta	D. Visión	Programa
<b>ACT4: LOCUTOR</b>	Media	Aprobado	Media	D. Visión	Programa
<b>ACT5: PORTERO DE LA ESTACIÓN</b>	Media	Aprobado	Media	D. Visión	Gerencia
<b>ACT6: SISTEMAS</b>	Media	Aprobado	Media	D. Visión	Programa
+ <Click here to create a requirement>					

### c. Características de software

Las características software son las necesidades de los usuarios propuestas por los stakeholders de la empresa, son los requisitos que

debe cumplir el sistema para satisfacer las necesidades de los trabajadores y de la empresa.

Las características definidas son las que aparecen en la matriz de atributos, siendo las indicadas como sub-características las derivadas según una clasificación jerárquicas.

**Tabla 2.5. Matriz de características del software**

Requirements:	Prioridad	Estado	Dificultad	Asignado a:
<b>CSW1: DEPARTAMENTO DE GERENCIA</b> Gestión de usuarios Gestión de accesos Gestión de errores de acceso	Media	Aprobado	Media	
<b>CSW2: DEPARTAMENTO DE VENTAS</b> Gestión de comerciales Gestión de contratos Gestión de paquetes	Alta	Aprobado	Alta	
<b>CSW3: DEPARTAMENTO DE PRODUCCIÓN</b> Ingreso de archivos Gestión de listas Automatización de listas	Alta	Aprobado	Media	
<b>CSW4: ANIMACIÓN</b> Información	Baja	Aprobado	Baja	
<b>CSW5: DEPARTAMENTO DE MANTENIMIENTO</b> Gestión de listas	Media	Aprobado	Media	
+ <Click here to create a requirement>				

#### **d. Casos de uso**

Derivados de las características software, son el resultado del análisis de las necesidades de los usuarios, cuyas especificaciones están recogidas

en el paquete Especificaciones de Casos de Uso definido en Requisite Pro.

**Tabla 2.6. Matriz de casos de uso**

Requirements:	Prioridad	Estado	Dificultad	Location
<b>CUS1: ingreso al sistema</b>	Alta	Aprobado	Media	
<b>CUS2: ingreso de usuario</b>	Alta	Aprobado	Media	
<b>CUS3: Programación de comerciales</b>	Alta	Por aprobar	Alta	Jefe de Ventas
<b>CUS4: Audio de fecha a fecha</b>	Alta	Aprobado	Alta	
<b>CUS5: Explorar archivo</b>	Alta	Por aprobar	Baja	
<b>CUS6: Búsqueda orden</b>	Media	Aprobado	Baja	
<b>CUS7: Contratos</b>	Media	Aprobado	Alta	Jefe de Ventas
<b>CUS8: Añadir contrato</b>	Alta	Aprobado	Alta	
+ <Click here to create a requirement>				

### e. Clases

Las clases son requerimientos derivados de los casos de uso como necesidad de representación del modelo de datos.

**Tabla 2.7. Matriz de atributos de las clases**

Requirements:	Prioridad	Estado	Dificultad	Location
<b>CLS1: Gerencia</b>	Media	Incorporado	Alta	Database
<b>CLS2: Jefe de Ventas</b>	Alta	Incorporado	Alta	Database
<b>CLS3: Ventas</b>	Alta	Incorporado	Alta	Database
<b>CLS4: Listas de canciones</b>	Media	Incorporado	Media	Database
<b>CLS5: Listas de comerciales</b>	Alta	Incorporado	Media	Database
<b>CLS6: Resultados de ventas</b>	Media	Incorporado	Media	Database
+ <Click here to create a requirement>				

## f. Especificación de requerimientos para XP

Las Historias de Usuarios son tarjetas escritas por el cliente en un lenguaje natural para éste, indicando las necesidades que el sistema debe satisfacer.

A continuación se describen las historias de usuario finales, las mismas que han sido agrupadas en los respectivos componentes del sistema para aplicarlos en el desarrollo del sistema usando la metodología XP.

**Tabla 2.8. Historias de usuarios finales**

<b>Historia de Usuario</b>	
<b>Número: 001</b>	<b>Usuario:</b>
<b>Nombre historia:</b>	
<b>Prioridad en negocio:</b>	Alta
<b>Riesgo en desarrollo:</b>	Alta
<b>Horas estimadas: 24</b>	<b>Iteración asignada: 1</b>
<b>Programador responsable:</b> Equipo XP (Diego Pule)	
<b>Descripción:</b>	
<b>Observaciones:</b>	

### 2.2.4. Verificar los requisitos

Consiste en comprobar el correcto funcionamiento de un requisito en la aplicación.

Para verificar este funcionamiento simplemente se creó un diseño del sistema con capturas de pantalla con la ayuda de hipervínculos que luego

fueron usadas en la parte de desarrollo del prototipo usando la metodología XP en la sección de prototipo de interfaces.

### 2.2.5. Validar los requisitos

Al crear cada elemento del modelo de análisis, este fue examinado para conocer su consistencia, sus omisiones y ambigüedades. A los requisitos que representa el modelo el cliente les da jerarquía y se agrupan en paquetes de requisitos que se implementan como incremento de software y se le entregan. La revisión final del modelo de análisis se enfocó en las siguientes preguntas:

**Tabla 2.9. Interrogantes validación de requisitos**

PREGUNTAS	SI	NO
¿Cada requisito fue consistente con el objetivo general del sistema?	X	
¿Todos los requisitos fueron presentados con detalle técnico para cada etapa del desarrollo?	X	
¿Cada requisito fue necesario en realidad o fue una característica agregada irrelevante para el objetivo final del sistema?	X	
¿Algunos requisitos entran en conflicto con otros?		X
¿Cada requisito se puede probar una vez que este haya sido implementado?	X	
¿El modelo de requisitos se ha sometido a toda una “participación” para que exponga en forma progresiva información más detallada acerca del sistema de automatización?	X	
¿Se han usado patrones de requisitos para simplificar el modelo de requisitos?	X	
¿Todos los patrones se han validado de manera apropiada?	X	
¿Todos los patrones son consistentes con los requisitos del cliente?	X	

Estas preguntas fueron realizadas para asegurar que la aplicación de la ingeniería de requisitos en el desarrollo de software de automatización es

un fiel reflejo de las necesidades del cliente y que proporciona una base sólida para el diseño y desarrollo del sistema.

### **2.3. Resumen**

Antes de que el diseño y la construcción del sistema de automatización del audio en vivo de Radio Majestad de la ciudad de Quito basado en computadoras puedan comenzar a desarrollarse, es necesario entender a cabalidad los requisitos. Esto se logra realizando una serie de tareas de ingeniería; las cuales se llevan a cabo durante las actividades de comunicación con el cliente y modelado que han sido definidas para el proceso genérico de desarrollo de software. El miembro del equipo de desarrollo de software realiza cinco funciones distintas dentro de dicha ingeniería:

- Obtener requisitos
- Analizar requisitos
- Documentar requisitos
- Verificar los requisitos
- Validar los requisitos.

Al inicio del proyecto el desarrollador y el cliente (así como los otros interesados) establecen los requisitos básicos del problema, se definieron las restricciones predominantes del proyecto y se especificaron las características y funciones más importantes que estarán presentes en el desarrollo del prototipo para que alcance sus objetivos. Esta información fue expandida y refinada durante la obtención de requisitos que fueron empleadas durante las reuniones que encabezó el desarrollador Diego Pule y también en el desarrollo de casos de uso.

La elaboración posterior fue expandir los requisitos para crear un modelo de análisis; es decir se elaboro una colección de elementos del modelo basados en casos de uso, en actividades y en clases, de comportamiento y orientados al flujo. En la creación de estos elementos se puede utilizar una variedad de notaciones de modelado. El modelo puede referirse a patrones de análisis, características del dominio del problema que son concurrentes a través de diferentes aplicaciones.

Cuando se identifican lo requisitos y se crea el modelo de análisis, el equipo de software, el cliente y otros interesados en el proyecto negocian la prioridad, disponibilidad y costo relativo de cada requisito. El objetivo de esta negociación y el modelo de análisis como un todo se validan contrastándolos con las necesidades del cliente para asegurar la construcción y desarrollo del sistema de automatización con las funciones correctas y dejar a un lado todo los errores.



## **CAPÍTULO 3**

### **ELABORACIÓN DEL PROTOTIPO**

#### **3.1. INTRODUCCIÓN**

Las estaciones de radio en los últimos años han evolucionado de manera gradual tanto en infraestructura como en tecnología, recordemos que hace algunos años para tener el audio en vivo en optimas condiciones se necesitaban de dos personas llamadas comúnmente como operadores o controladores los cuales estaban encargados de colocar toda la programación musical de la estación de radio, tandas comerciales, jingles e identificativos; en la actualidad ya no es necesario que trabajen tantas personas y en especial que estén manejando los equipos de manera constante las 24 horas, todo esto es debido a la incorporación de una sola maquina llamada computadora u ordenador este dispositivo contempla todas los “requerimientos” que supuestamente necesita la estación de radio a reemplazado a dichos operadores en el audio en vivo, pero la pregunta es ¿Realmente la computadora puede realizar todo eso?, otra pregunta que siempre resalta es ¿Qué tan confiable es este equipo si funciona las 24 horas del día los 365 días al año?, a finales de los años 90 aparecieron en el mercado algunos paquetes de software que estaban destinados a crear una automatización de radio, pero eran sistemas demasiado caros o muy sencillos, con requerimientos muy elevados, ya que en ese entonces las capacidades de las computadoras eran muy limitadas, en la actualidad es posible encontrar a precios muy asequibles equipos con gran capacidad además de rebasar con creces los requerimientos de un software de automatización de los años 90, es por eso que la presente tesis se encamina solo a la parte del desarrollo de un prototipo que cumpla todos los requisitos que dejaron de lado los más importantes paquetes informáticos de automatización radial que se licencian en el mercado.

### 3.2. DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema de automatización de radio Majestad de audio en vivo estará alojado en un servidor de forma local usando el software libre Apache Tomcat versión 7, lamentablemente no se puede usar la aplicación bajo un sistema operativo de software libre como es el caso de Ubuntu debido no a que el sistema de automatización no funcione sino debido al inconveniente de compatibilidad de driver de hardware, la estación de radio Majestad utiliza justamente para el audio en vivo un tipo de tarjeta de audio profesional como es la Delta 44 de la empresa M – Audio, lamentablemente el fabricante de la tarjeta no desarrolla drivers para sistemas operativos de código abierto ver figura estará desarrollado en software libre y se podrá acceder desde cualquier navegador, el sistema a usar será Windows 7 versión ultimate, una computadora Intel core 2 duo de tendrá varios módulos principalmente satisfaciendo las necesidades de la estación de radio estos son:

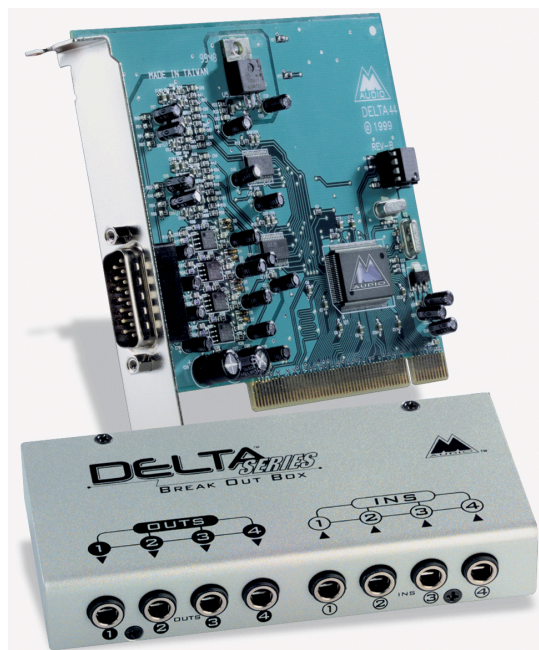


Figura 3.7: Tarjeta de sonido

- Ingreso del administrador
- Programas
- Paquetes
- Usuarios
- Majestad estudio
- Clientes
- Programación
- Facturación
- Contratos
- Acceso al sistema
- Error de uso de sistema
- Salir

### **3.3. METODOLOGÍA DE DESARROLLO**

En el desarrollo del prototipo del sistema de automatización de Radio Majestad de la ciudad de Quito, es necesario determinar un conjunto de actividades que permitirán transformar los requisitos de un usuario en un sistema de software, satisfaciendo todas sus necesidades cuando este lo requiera.

De acuerdo a las características del sistema de automatización y a las necesidades del usuario resultó seleccionada la metodología Extreme Programming, debido a que está presenta resultados a los clientes de acuerdo a como va avanzando el proyecto. Además para la fase del diseño de las interfaces y usabilidad del sistema el equipo de desarrollo se basará en el sistema actual de automatización que posee la estación de radio.

### **3.4. HERRAMIENTAS DE DESARROLLO**

Para el desarrollo del prototipo es necesario conocer las características de las distintas herramientas que participarán en el desarrollo del proyecto, su funcionalidad y las ventajas que estas nos ofrecen. A continuación se describen las herramientas a utilizarse en el desarrollo del prototipo.

#### **3.4.1. Apache**

Apache es un servidor Web desarrollado por la Apache Software Foundation, es un software libre de código abierto que funciona sobre cualquier plataforma. Tiene capacidad para servir páginas de contenido estático y dinámico. Debido a su eficiencia, funcionalidad, velocidad y estabilidad se ha convertido en uno de los mejores servidores y según estudios realizados por Netcraft<sup>22</sup> es el servidor Web más popular del mundo.

Para el desarrollo del portal se va a utilizar la versión Apache Tomcat versión 7.0.2

#### **Características de Apache 10**

- Funciona sobre diversas plataformas como Unix, Windows y Mac OS.
- Soporta CGI, Perl, PHP.
- Tiene soporte directo SQL para módulos que lo necesiten.
- Es flexible, rápido, continuamente actualizado y adaptado a los nuevos protocolos.

---

<sup>22</sup>Netcraft es un sitio dedicado a monitorear servicios webs del planeta. Entre sus aportes a la minería de datos mide la evolución de hosts, servidores, marcas de web servers, etc.

- Apache es un servidor altamente configurable de diseño modular. Este diseño permite a los administradores del sitio web elegir las características que van a ser incluidas en el servidor seleccionando los módulos que se van a cargar, ya sea al compilar o al ejecutar el servidor.
- Se adapta a diferentes entornos y necesidades con la API de programación y los módulos de apoyo.
- Permite personalizar respuestas ante posibles errores que pueda presentar en el servidor.
- Permite la creación de ficheros de log, de este modo tiene un mayor control sobre lo que ocurre en el servidor.
- Posee soporte IPV6

### **3.4.2 MYSQL V 5.1.13.**

Es un sistema de gestión de base de datos relacional, multi hilo y multiusuario. Está desarrollado en ANSI C en su mayor parte. Es un software libre que posee un licenciamiento dual, es decir ofrece una licencia GNU GPL y otra que permite a una empresa incorporarlo en productos privados. MySQL está patrocinado por una empresa privada la misma que ofrece soporte y servicios. Para el desarrollo del portal se va a utilizar 5.1.13.

#### **Características de mysql 5.1.13.**

- Funciona sobre múltiples plataformas como: Linux, Mac OS X, Solaris, Novell Netware, Windows. Permite a aplicaciones escritas en C, C++ C#, Pascal, Delphi, Java, Lisp, Perl, PHP, Python acceder a la base de datos MySQL para lo cual cada uno utiliza una API determinada, además existe una interfaz ODBC que

permite que cualquier lenguaje de programación que lo soporte se comunique con las base de datos.

- Brinda sistemas de almacenamiento transaccionales y no transaccionales.
- Procedimientos almacenados.
- Triggers a nivel de filas.
- Vistas actualizables
- Cursores.
- Tiene alta seguridad, debido a que las cuentas de registro están almacenadas en la base de datos de la cuenta.
- Contraseñas seguras ya que el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
- Soporta grandes bases de datos, es decir bases de datos con gran número de tablas y registros.
- Se puede conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma.
- Los servidores Windows permiten conexiones con memoria compartida.
- El servidor puede desplegar mensajes de error en varios idiomas.
- Tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.

### **3.4.3 ECLIPSE<sup>23</sup>**

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java DevelopmentToolkit (JDT) y el compilador (ECJ)

---

<sup>23</sup>[http://es.wikipedia.org/wiki/Eclipse\\_%28software%29](http://es.wikipedia.org/wiki/Eclipse_%28software%29)

que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de ModelDrivenEngineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse fue liberado originalmente bajo la CommonPublicLicense, pero después fue re-licenciado bajo la Eclipse PublicLicense. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL)

### **Características de Eclipse**

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización.

Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion.4 e integración con Hibernate.5.

### 3.5. SISTEMA DE AUTOMATIZACION

#### 3.5.1 Macroprocesos

La actividad de automatización que se desarrolla en la Estación de Radio Majestad en la Ciudad de Quito para administrar el audio en vivo, se pueden mostrar y presentar mediante procesos y macroprocesos, los mismos que se muestran en la siguiente tabla:

**Tabla 3.10. Procesos automatizados de Radio Majestad**

Usuarios	Macro procesos	Procesos que comprende
Administrador	Ingreso de usuarios	Ingreso de datos del usuario. Ingreso al sistema. Creación de Usuarios.
	Majestad Studio	Ingreso al programa de automatización de las listas de archivos de audio
	Programación	Examinar archivo a enlistar Agregar archivo de audio Agregar el archivo con la respectiva fecha a reproducir.
	Contratos	Añadir contrato Editar contrato. Controlar fecha y hora del contrato.
	Clientes	Añadir clientes Editar clientes
	Facturación	Añadir clientes para la facturación. Ver contratos de los clientes.
	Programas	Añadir programa Editar programa Eliminar programa



	Paquetes	Añadir paquete Editar paquete.
	Acceso al sistema	Muestra los accesos validos al sistema
	Errores de accesos	Muestra los intentos de acceder al sistema

### 3.5.2 Especificación de requisitos

Las Historias de Usuarios son tarjetas escritas por el cliente en un lenguaje natural para este, indicando las necesidades que el sistema debe satisfacer. A continuación se describen las historias de usuario finales, las mismas que han sido agrupadas en los respectivos componentes del sistema. Algunas de estas historias fueron eliminadas o cambiadas a lo largo del proyecto, a medida que se aplicó la ingeniería de requisitos en todas sus fases.

**Tabla 3.11. Gestión de usuarios**

<b>Historia de Usuario</b>	
<b>Número: 001</b>	<b>Usuario: Administrador</b>
<b>Nombre: Administración de Usuarios (Usuarios)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Alto</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: El administrador es el encargado de establecer la creación de usuarios mediante una contraseña asignada</b>	
<b>Observaciones: El administrador es el único que puede crear y eliminar los distintos usuarios.</b>	
<b>Documentos Resultantes: Lista de Usuarios, Lista de accesos validos, Lista de accesos no validos.</b>	

**Tabla 3.12. Gestión de listas de reproducción**

<b>Historia de Usuario</b>	
<b>Número: 002</b>	<b>Usuario: Administrador, Director de programación</b>
<b>Nombre: Gestión de listas de reproducción (Programación)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Alto</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: Mediante el botón de examinar el administrador y/o el director de programación buscan el tema a enlistar, mediante la tecla de agregar se agrega una lista para la programación.</b>	
<b>Observaciones: El administrador y/o el director de programación son los que realizan esta acción.</b>	
<b>Documentos Resultantes: Lista de canciones a reproducir con fecha y tiempo estimado.</b>	

**Tabla 3.13. Gestión de contratos para publicidad**

<b>Historia de Usuario</b>	
<b>Número: 003</b>	<b>Usuario: Administrador, jefe de ventas</b>
<b>Nombre: Gestión de contratos para la publicidad (Contratos)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Alto</b>
<b>Horas Estimadas: 16</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón "Añadir Contrato" el jefe de ventas crea una (número) # de orden, con un numero de cédula o RUC, y con su respectivo paquete, es importante colocar la fecha de inicio con la fecha de finalización del contrato.</b>	
<b>Observaciones: El administrador y/o el jefe de ventas son los que realizan esta acción.</b>	
<b>Documentos Resultantes: lista del comercial a pasar en determinada fecha y hora.</b>	

**Tabla 3.14. Gestión de clientes**

<b>Historia de Usuario</b>	
<b>Número: 004</b>	<b>Usuario: Administrador, jefe de ventas</b>
<b>Nombre: Gestión de clientes (Clientes)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Media</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón “Clientes” el jefe de ventas o el administrador pueden crear el tipo de clientes que deseen ya sea para uso frecuente u ocasional también se describe el uso de agencias para su respectivo descuento, también sirve para actualizar los datos.</b>	
<b>Observaciones: El administrador y/o el jefe de ventas son los que realizan esta acción.</b>	
<b>Documentos Resultantes: lista de clientes con su número de cedula y/o RUC, nombre, dirección, descuento, tipo.</b>	

**Tabla 3.15. Gestión de facturación**

<b>Historia de Usuario</b>	
<b>Número: 005</b>	<b>Usuario: Administrador, jefe de ventas</b>
<b>Nombre: Gestión de facturación (Facturación)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Alto</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón “Facturación”, podemos ver el tipo de contrato, además del respectivo descuento, se pueden añadir clientes de manera directa.</b>	
<b>Observaciones: El administrador y/o el jefe de ventas son los que realizan esta acción.</b>	
<b>Documentos Resultantes: lista de clientes con su respectivo contrato.</b>	

**Tabla 3.16. Gestión de programas**

<b>Historia de Usuario</b>	
<b>Número: 006</b>	<b>Usuario: Administrador, locutor (s)</b>
<b>Nombre: Gestión de programas (Programas)</b>	
<b>Prioridad en Negocio: Bajo</b>	<b>Riesgo en desarrollo: Bajo</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón “Programas” se puede añadir el programa o nombres de programas con el nombre del locutor con su respectivo horario</b>	
<b>Observaciones: El administrador y/o locutor son los que realizan esta acción.</b>	
<b>Documentos Resultantes: lista de programas con la hora de inicio y hora final.</b>	

**Tabla 3.17. Gestión de paquetes publicitarios**

<b>Historia de Usuario</b>	
<b>Número: 007</b>	<b>Usuario: Administrador, jefe de ventas</b>
<b>Nombre: Gestión de paquetes publicitarios (Paquetes)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Alto</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón “Añadir Pack” el jefe de ventas y/o administrador crea el nombre del paquete, da un descripción, crea un numero de pasadas, crea el valor unitario de cada comercial y los inserta.</b>	
<b>Observaciones: El administrador y/o el jefe de ventas son los que realizan esta acción.</b>	
<b>Documentos Resultantes: Muestra la lista de paquetes con todas las descripciones posibles.</b>	

**Tabla 3.18. Gestión de accesos válidos al sistema**

<b>Historia de Usuario</b>	
<b>Número: 008</b>	<b>Usuario: Administrador.</b>
<b>Nombre: Gestión de Accesos Validos al Sistema (Accesos al Sistema)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Medio</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón “Acceso al sistema” enlista el número de ingresos al sistema de automatización.</b>	
<b>Observaciones: El administrador y/o el jefe de ventas son los que realizan esta acción.</b>	
<b>Documentos Resultantes: lista de accesos validos al sistema.</b>	

**Tabla 3.19. Gestión de intentos de acceso al sistema**

<b>Historia de Usuario</b>	
<b>Número: 009</b>	<b>Usuario: Administrador.</b>
<b>Nombre: Gestión de intentos de acceso al sistema (Errores de Acceso)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Medio</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón “Errores de Acceso” enlista el numero de intentos de ingreso erróneos al sistema de automatización.</b>	
<b>Observaciones: El administrador y/o el jefe de ventas son los que realizan esta acción.</b>	
<b>Documentos Resultantes: lista de intentos de acceso erróneos al sistema.</b>	

**Tabla 3.20. Gestión de accesos válidos al sistema**

<b>Historia de Usuario</b>	
<b>Número: 010</b>	<b>Usuario: Administrador.</b>
<b>Nombre: Gestión de Accesos Validos al Sistema (Accesos al Sistema)</b>	
<b>Prioridad en Negocio: Alta</b>	<b>Riesgo en desarrollo: Alta</b>
<b>Horas Estimadas: 30</b>	<b>Iteración Asignada: 1</b>
<b>Programador Responsable: Equipo XP (Diego Pule)</b>	
<b>Descripción: En el botón "Majestad Studio" muestra la consola de automatización de las dos principales listas, la lista de música programada por el director de programación y la lista de comerciales a pasar a determinada hora</b>	
<b>Observaciones: Solo es una pantalla que muestra la automatización, tiene los botones de automatización pero son manejados por el locutor.</b>	
<b>Documentos Resultantes: lista de archivos de audio ya programados.</b>	

### 3.6. ANÁLISIS DE INFORMACIÓN

#### 3.6.1. Priorización y estimación

En la siguiente tabla se presenta la estimación del tiempo que se requiere para la implementación y la prioridad de las distintas historias de usuario recolectadas y corregidas, tomando en cuenta que un día laborable para el equipo de desarrollo XP es de 4 horas.

**Tabla 3.21. Priorización y estimación de tiempos de implementación**

Historias de Usuario	Tiempo Estimado		
	Prioridad	Días Estimados	Horas Estimadas
<b>001. Administración de Usuarios</b>	Alta	3,5	14
<b>002. Datos</b>	Alta	3	12
<b>003. Contratos</b>	Alta	3,5	14
<b>004. Listas de canciones</b>	Alta	4	16
<b>005. Acceso al sistema</b>	Alta	2	8
<b>006. Errores al sistema</b>	Alta	7	28
<b>007. Facturas</b>	Alta	2,5	10
<b>008. Inicio</b>	Alta	3	12
<b>009. Consola</b>	Alta	3	12
<b>010. Datos del Programa</b>	Alta - Media	3	12

### 3.6.2. Distribución funcional

Se identificaron los distintos módulos de la aplicación, con el propósito de tomar en cuenta la distribución funcional de las historias de usuario recolectadas en la elaboración del plan de entrega, a más de la prioridad y el tiempo de implementación estimado. En la siguiente tabla se muestra la funcionalidad detallada como gestión.

**Tabla 3.22. Distribución funcional de las historias de usuarios**

Módulo	Historias de Usuario
<b>Gestión de Usuarios</b>	001. Administración de Usuarios
	002. Datos del Usuario
	003. Datos de los demás usuarios
<b>Gestión de Listas de Canciones</b>	004. Datos de canciones
	005. Calidad de reposición de la lista de canciones

	006. Enlistar las canciones
	007. Exploración de canciones
<b>Gestión de Listas</b>	008. Ingreso de Listas
	009. Tiempo de automatización
	010. Ingreso de listas de comerciales
	011. Errores de canciones
<b>Gestión de Información</b>	012. Base de datos
	013. Búsqueda de canciones
	014. Inicio
	015. Respaldo de Información
<b>Gestión de servicios</b>	016. Automatización
	017. Facturas
	018. Consulta de clientes

### 3.6.3. Estimación de la duración del proyecto

Para establecer la estimación total de la duración del proyecto se tomo en cuenta el tiempo total de horas que tomará la implementación de las historias de usuario, además que el equipo de desarrollo (Diego Pule) trabajará 5 días a la semana y 4 horas diarias.

**Tabla 3.23. Estimación de la duración del proyecto**

<b>Actividades</b>	<b>Tiempo (Horas)</b>
<b>Requerimientos definidos en la etapa de exploración.</b>	400
<b>Requerimientos futuros (a)</b>	12
<b>Imprevistos (b)</b>	20
<b>Pruebas de aceptación y reuniones (c)</b>	80
<b>TOTAL</b>	<b>512</b>



- 3% del total de horas estimado para la implementación de los requerimientos recolectados en la etapa de exploración, para requerimientos no descubiertos inicialmente.
- 5% del tiempo de horas estimado para requerimientos futuros.
- 20% del total de horas estimado para la implementación.

Después de realizar los cálculos respectivos se estima que la duración del proyecto será de 6 a 7 meses.

Una vez determinada la estimación de la duración del proyecto, se establece el plan de entrega haciendo uso de las estimaciones de las historia de usuario.

### 3.6.4. Plan de entregas

Las siguientes tablas detallan el plan de entregas.

**Tabla 3.24. Plan de entregas de acuerdo a iteraciones**

Historias de usuario	Tiempo estimado		Iteración asignada			Entrega Asignada		
	Días estimados	Horas estimadas	1	2	3	1	2	3
<b>001. Administración de Usuarios</b>	3,5	14	X			X		
<b>002. Datos del canciones</b>	3	12	X			X		
<b>003. Inserción de canciones</b>	3,5	14	X			X		
<b>004. Inserción de listas comerciales</b>	3	12	X			X		
<b>005. Automatización</b>	8	32	X			X		
<b>006. Facturas</b>	6	24		X			X	
<b>007. Lista de errores</b>	4	16		X			X	
<b>008. Lista de accesos al sistema</b>	2,5	10			X			X

<b>009. Programas</b>	6	24			X			X
<b>010. Clientes</b>	2,5	10			X			X

**Tabla 3.25. Plan de entregas de acuerdo a los módulos**

Módulo	Historias de Usuario	Tiempo estimado		Iteración asignada			Entrega Asignada		
		Días estimados	Horas estimadas	1	2	3	1	2	3
<b>Gestión de Usuarios</b>	001. Administración de Usuarios	7,5	30	X			X		
	002. Datos del Usuario	6	24		X			X	
	003. Datos de los demás usuarios	5	20	X			X		
<b>Gestión de listas de audios</b>	004. Datos de canciones	7,5	30	X			X		
	005. Calidad de reposición de la lista de canciones	6	24	X			X		
	006. Enlistar las canciones	5	20	X			X		
	007. Exploración de canciones	2,5	10			X			X
<b>Gestión de Información</b>	008. Base de datos	3	12			X			X
	009. Búsqueda de canciones	2,5	10			X			X
	010. Respaldo de Información	5	20	X			X		
	011. Inicio de listas	5	20	X			X		
<b>Gestión de servicios</b>	012. Facturas	4	16	X			X		
	013. Automatización	4	16	X			X		
	014. Consulta de clientes	2,5	10			X			X

**Tabla 3.26. Plan primera iteración**

<b>Id</b>	<b>Nombre de tarea</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Duración</b>
1	Diseño de páginas con hipervínculos	20/05/11	25/05/11	4 días
2	Colores del sistema	23/05/11	23/05/11	1 día
3	Nombre del sistema	25/05/11	26/05/11	2 días
4	Diseño de la menú	30/05/11	30/05/11	1 día
5	Programación	30/05/11	02/06/11	4 días
6	Audio para la fecha listar	31/05/11	31/05/11	1 día
7	Exploración de archivos	31/05/11	07/06/11	6 días
8	Agregar archivo	03/06/11	03/06/11	1 día
9	Búsqueda de archivo asignado	06/06/11	08/06/11	2 días
10	Regreso al sistema	08/06/11	17/06/11	8 días
11	Añadir usuario	13/06/11	15/06/11	3 días
12	Acceso al sistema lista	16/06/11	16/06/11	1 día
13	Errores de Acceso lista	17/06/11	17/06/11	1 día
14	Programas	20/06/11	21/06/11	2 días
15	Añadir programa	21/06/11	21/06/11	1 día

<b>Id</b>	<b>Nombre de tarea</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Duración</b>
1	Añadir programas	21/06/11	23/06/11	3 días
2	Nombre	23/06/11	23/06/11	1 día
3	Conductor	23/06/11	29/06/11	5 días
4	Notas	23/06/11	24/06/11	2 días
5	Hora inicio	27/06/11	30/06/11	4 días
6	Hora fin	30/06/11	30/06/11	1 día
7	Insertar datos de programas	28/06/11	06/07/11	7 días

**Tabla 3.27. Plan segunda iteración**

<b>Id</b>	<b>Nombre de tarea</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Duración</b>
1	Diseño de menú	22/07/11	22/07/11	1 día
2	Colores con diseño del menú	25/07/11	27/07/11	3 días
3	Tablas dinámicas	26/07/11	29/07/11	4 días
4	Diseño de consola lista de canciones	27/07/11	29/07/11	3 días
5	Diseño de lista de comerciales	29/07/11	04/08/11	5 días
6	Clientes	01/08/11	10/08/11	8 días
7	Manejo de listas programadas	03/08/11	05/08/11	3 días
8	Pruebas de listas dinámicas	04/08/11	05/08/11	2 días
9	Listas dinámicas errores de acceso	08/08/11	11/08/11	4 días
10	Listas dinámicas accesos validos	09/08/11	10/08/11	2 días
11	Programación	11/08/11	11/08/11	1 día
12	Diseño de Iconos menú principal	11/08/11	15/08/11	2 días
13	Paquetes	15/08/11	16/08/11	2 días
14	Enlistar paquetes	16/08/11	23/08/11	6 días
15	Usuarios	17/08/11	22/08/11	4 días

**Tabla 3.28. Plan tercera iteración**

<b>Id</b>	<b>Nombre de tarea</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Duración</b>
1	Automatización	26/09/11	03/10/11	6 días
2	Gestión completa de listas	28/09/11	30/09/11	3 días
3	Gestión de listas de comerciales	03/10/11	06/10/11	4 días
4	Facturas	05/10/11	06/10/11	2 días
5	Paquetes	06/10/11	10/10/11	3 días
6	Creación de paquetes	10/10/11	12/10/11	3 días
7	Listas dinámicas	12/10/11	18/10/11	5 días
8	Añadir paquetes	14/10/11	19/10/11	4 días
9	Buscar paquetes	17/10/11	19/10/11	3 días
10	Editar paquetes	20/10/11	20/10/11	1 día
11	Añadir clientes	24/10/11	28/10/11	5 días
12	Buscar clientes	25/10/11	26/10/11	2 días

13	Lisas dinámicas	03/11/11	08/10/11	4 días
14	Editar clientes	07/11/11	10/11/11	4 días
15	Tipo de cliente	10/11/11	11/11/11	2 días
16	Código de clientes	14/11/11	16/11/11	3 días
17	Mostrar precio de paquete	15/11/11	15/11/11	1 día
18	Contratos	15/11/11	18/11/11	4 días
19	Añadir contrato	17/11/11	18/11/11	2 días
20	Pautaje	18/11/11	18/11/11	1 día

## **DISEÑO**

### **Mapa del Sitio**

Ingreso al sistema

Menú principal

MAJESTAD STUDIO

Programación

Contratos

Clientes

Facturación

Programas

Paquetes

Usuarios

#### **3.9.1. Diseño de la Experiencia de Usuario**

El objetivo de esta sección es que el cliente - usuario que use el sistema se encuentre familiarizado con la aplicación que desea realizar de una manera más simple y sencilla, de tal forma que se sienta cómodo y seguro de usarlo.

Para llevar a cabo esta misión el equipo de desarrollo hasta uso de los diagramas de iteración, los cuales representan gráficamente las posibles acciones que tiene el usuario.

### Ingreso al sistema

La Figura indica la posibilidad de reacción que tiene el sistema ante el ingreso al sistema de usuario registrado.

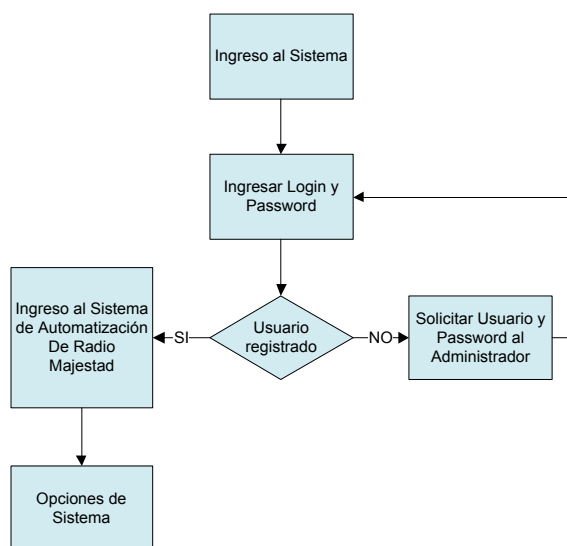
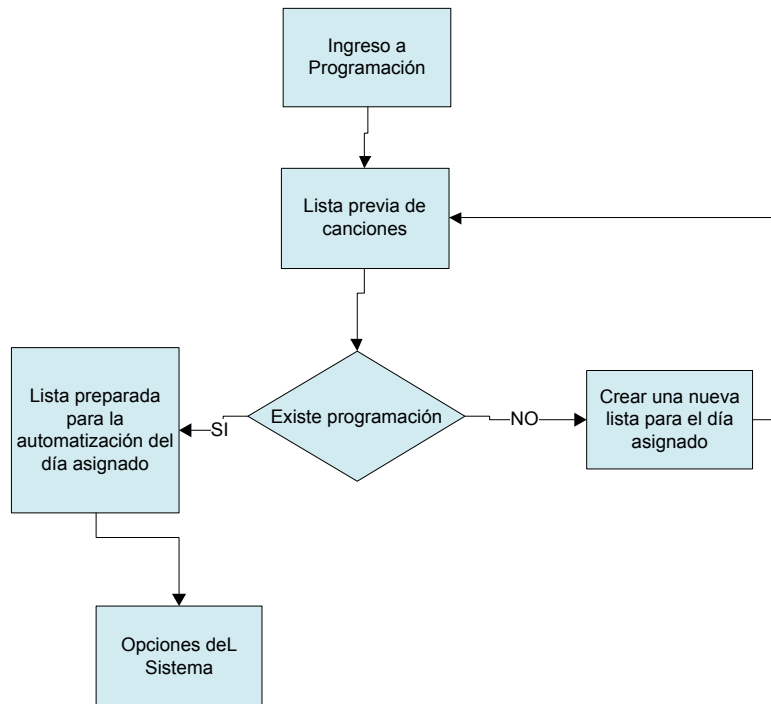


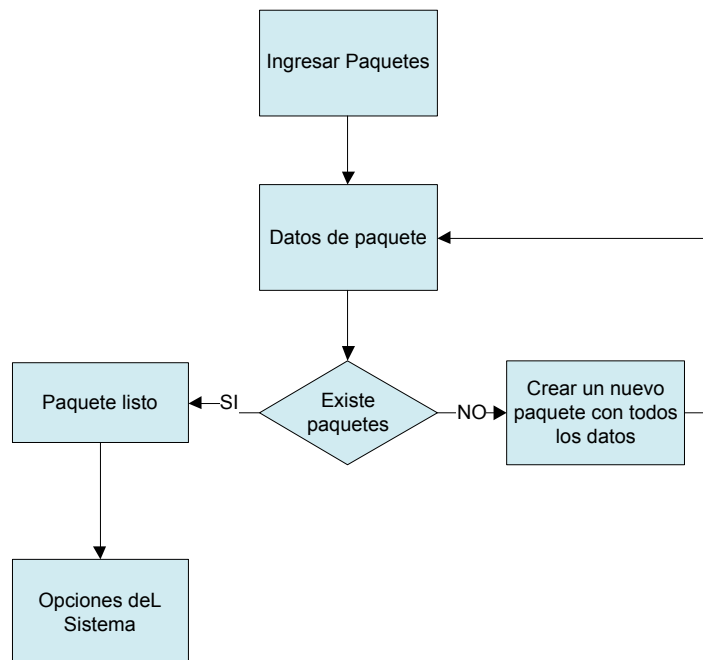
Figura 3.8: Ingreso al sistema

### Consola del sistema

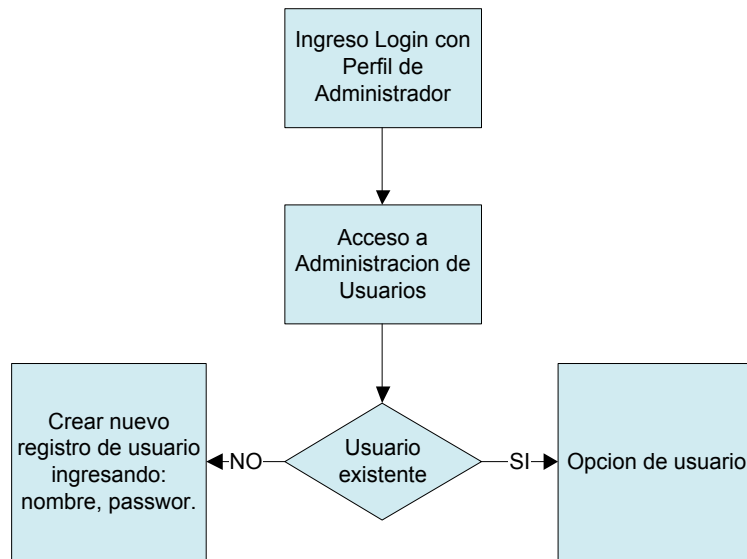
En la figura se muestran las opciones que presenta el sistema MajestoSoft, según cada usuario.



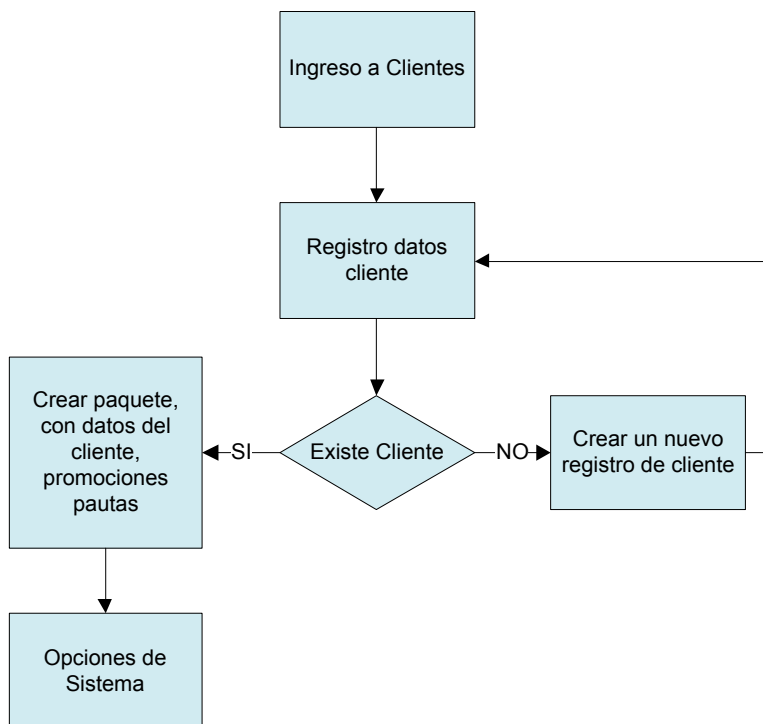
**Figura 3.9: Programación**



**Figura 3.10: Paquetes**

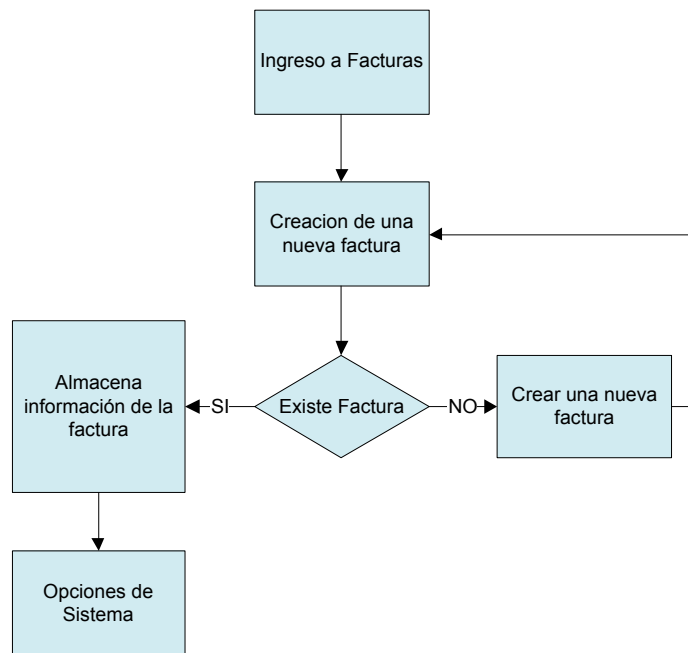


**Figura 3.11: Usuarios**

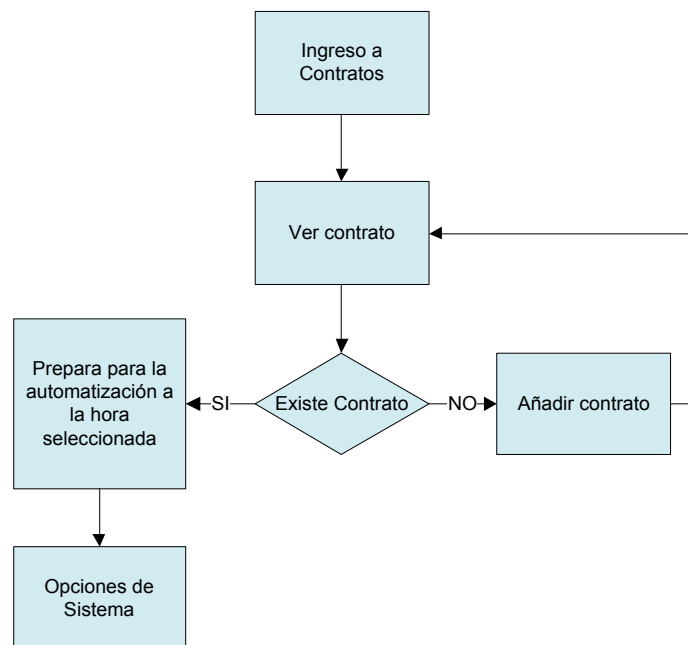


**Figura 3.12: Clientes**





**Figura 3.13: Facturas**



**Figura 3.14: Contratos**

### 3.10. PROTIPO DE INTERFACES

En la presente sección se presentan las interfaces del sitio web denominado MajestoSoft 1.0.



Figura 3.15: Página de Inicio al sistema



Figura 3.16: Consola del Sistema

## Programas

[Añadir Programas](#)

Ver  Registros Buscar:

Código	Nombre	Conductor	Notas	Hora Inicio	Hora Fin		
1	Programa 1	Conductor 1		09:00:00	10:00:00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	Programa 2	Conductor 2		10:00:00	11:00:00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
4	Programa 3	Conductor 3		12:00:00	13:00:00	<a href="#">Editar</a>	<a href="#">Eliminar</a>

Mostrando del 1 al 3 de 3 registros ◀ ▶

[Inicio](#)

Figura 3.17: Programas

## Paquetes

[Añadir Pack](#)

Ver  Registros Buscar:

Código	Nombre	Descripción	Pasadas	Valor Unitario	Responsable		
1	Por Pasada	Costo por pasada	0	20	1	<a href="#">Editar</a>	
2	Pack4	Paquete de 4 pasadas al día	4	18	1	<a href="#">Editar</a>	
3	Pack8	Paquete de 8 pasadas al día	8	16	1	<a href="#">Editar</a>	
6	Pack16	Paquete de 16 pasadas al día	16	8	1	<a href="#">Editar</a>	

Mostrando del 1 al 4 de 4 registros ◀ ▶

[Inicio](#)

Figura 3.18: Paquetes



Figura 3.19: Usuarios

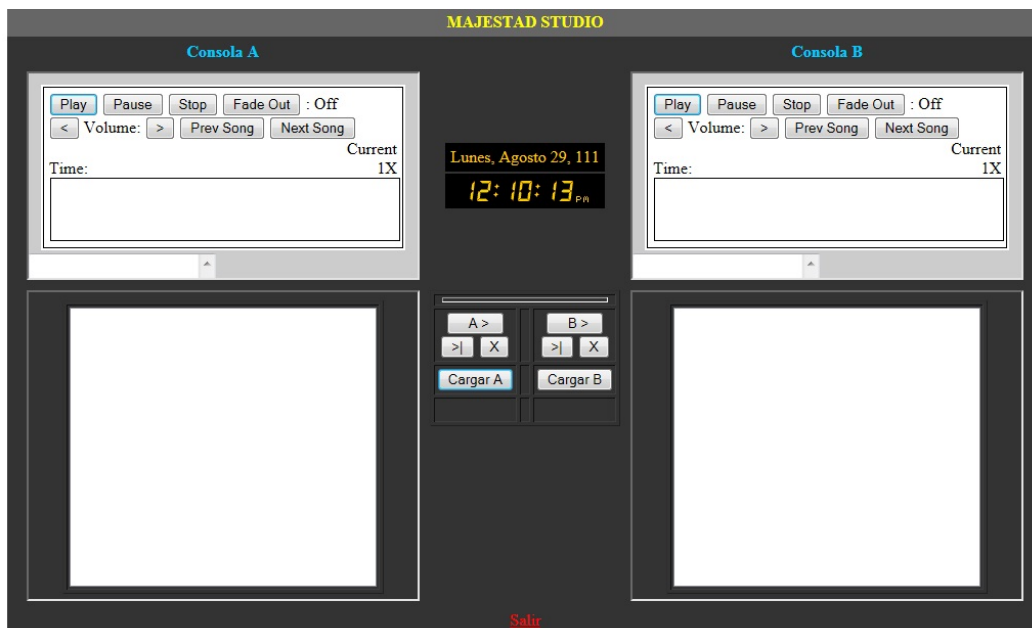


Figura 3.20: Majestad Studio




## Clientes

**Lista de Clientes** [Añadir Clientes](#)

Ver  Registros Buscar:

Código	Cedula/Ruc	Nombre	Dirección	Descuento	Tipo	
1	1234567890	UDLA	Quito	Udla	Frecuente	<a href="#">Editar</a>
2	1234567891	UEmisferios	Quito	UEmisferios	Frecuente	<a href="#">Editar</a>
3	1234567892	CNT	Quito	5	Frecuente	<a href="#">Editar</a>
4	1234567893	LANpass	Quito	3	Frecuente	<a href="#">Editar</a>
8	0102030405	cliente1	dir3	1	Agencia	<a href="#">Editar</a>
9	0102050405001	cliente 2	dir2	10	Agencia	<a href="#">Editar</a>
10	0123456789011	cliente2	das	50	Agencia	<a href="#">Editar</a>

Mostrando del 1 al 7 de 7 registros ◀ ▶

Usuario Activo: Administrador [Inicio](#)

Figura 3.21: Clientes




## Programación

Audios para Fecha:

Archivo:

Programacion para: null

Ver  Registros Buscar:

Orden	Fecha	File	
No hay datos que mostrar			

Mostrando del 0 al 0 de 0 registros ◀ ▶

Responsable: Administrador [Inicio](#)

Figura 3.22: Programación

## Facturación

### Lista de Clientes

[Añadir Clientes](#)

Código	Cedula/Ruc	Nombre	Dirección	Descuento	Tipo	
1	1234567890	UDLA	Quito	Udla	Frecuente	<a href="#">Ver Contratos</a>
2	1234567891	UEmisferios	Quito	UEmisferios	Frecuente	<a href="#">Ver Contratos</a>
3	1234567892	CNT	Quito	5	Frecuente	<a href="#">Ver Contratos</a>
4	1234567893	LANpass	Quito	3	Frecuente	<a href="#">Ver Contratos</a>
8	0102030405	cliente1	dir3	1	Agencia	<a href="#">Ver Contratos</a>
9	0102050405001	cliente 2	dir2	10	Agencia	<a href="#">Ver Contratos</a>
10	0123456789011	cliente2	das	50	Agencia	<a href="#">Ver Contratos</a>

Usuario Activo: Administrador

[Inicio](#)

Figura 3.23: Facturación

## Contratos

### Contrato

[Añadir Contrato](#)

Código	# Orden	RUC	Pack	Desde	Hasta	Valor	
3	12	1	Pack4	2011-08-12	2011-09-12	123	<a href="#">Editar</a>
C:\audios\publicidad\claro.mp3 <a href="#">Pautaje:</a> 1) 14:00 <sup>±</sup> ;							
1	1	1	Pack16	2011-08-01	2011-09-01	10	<a href="#">Editar</a>
C:\audios\publicidad\udla.mp3 <a href="#">Pautaje:</a> 1) 11:00 <sup>±</sup> ; 2) 11:30 <sup>±</sup> ; 3) 11:35 <sup>±</sup> ;							
2	25	4	Pack4	2011-08-01	2011-09-01	12	<a href="#">Editar</a>
C:\audios\publicidad\lanpas.mp3 <a href="#">Pautaje:</a> 1) 10:00 <sup>±</sup> ; 2) 10:30 <sup>±</sup> ; 3) 10:35 <sup>±</sup> ; 4) 10:40 <sup>±</sup> ; 5) 10:45 <sup>±</sup> ; 6) 10:50 <sup>±</sup> ; 7) 10:55 <sup>±</sup> ; 8) 12:00 <sup>±</sup> ;							

[Inicio](#)

Figura 3.24: Contratos

## Accesos Validos Al Sistema

### Lista de Accesos

Ver  Registros Buscar:

Número	Usuario	Fecha
1	Administrador	2012-01-15 15:42:08.0
2	Administrador	2012-01-14 10:09:31.0
3	Administrador	2012-01-14 10:05:47.0
4	Administrador	2012-01-14 07:42:44.0
5	Administrador	2012-01-12 09:35:22.0
6	Administrador	2012-01-12 09:34:32.0
7	Administrador	2012-01-11 22:49:31.0
8	Administrador	2011-10-27 16:00:15.0
9	Administrador	2011-09-28 16:28:30.0
10	Administrador	2011-09-18 09:42:12.0

Mostrando del 1 al 10 de 94 registros Inicio  
 Usuario Activo: Administrador

Figura 3.25: Acceso al Sistema

## Errores de Ingreso

### Lista de Accesos

Ver  Registros Buscar:

Número	Acceso	Fecha
1	Error de Ingreso al Sistema: Usuario:admin Contraseña: admini	2011-08-22 16:04:11.0
2	Error de Ingreso al Sistema: Usuario:admin Contraseña: admini	2011-08-22 15:46:44.0
3	Error de Ingreso al Sistema: Usuario:admin Contraseña: admini	2011-08-16 19:48:07.0
4	Error de Ingreso al Sistema: Usuario:admin Contraseña: ad	2011-08-07 09:49:14.0
5	Error de Ingreso al Sistema: Usuario:admin ContraseÃ±a: aadmin	2011-08-02 22:37:07.0
6	Error de Ingreso al Sistema: Usuario:asd ContraseÃ±a: asd	2011-07-24 19:49:48.0

Mostrando del 1 al 6 de 6 registros

Usuario Activo: Administrador

Inicio

Figura 3.26: Error de uso del Sistema

### **3.11. Pruebas**

Se tomaron como base para las pruebas de aceptación las historias de usuario ya que estas definen las condiciones a satisfacer de cada parte con el desarrollo de cada parte del sistema de automatización.

#### **3.11.1. Pruebas de la primera iteración**

##### **Historia de Usuario No. 001 Administración de Usuarios**

###### **Verificación de usuario registrado**

###### **Descripción**

El usuario al ingresar a la página que contiene el sistema se encontrará con el logotipo de la estación de radio y con el nombre de Majesodft. Tras realizar un clic hay una ventana con campos que solicitan el ingreso de usuario y contraseña. Si el usuario no está registrado no puede ingresar al Sistema. Tras la respectiva identificación, el usuario tendrá acceso a la consola del sistema.

###### **Condiciones de ejecución**

Ninguna

###### **Entrada**

El usuario ingresa a la página de inicio del sitio que contiene el sistema, [www.radiomajestad.com](http://www.radiomajestad.com).

En la siguiente ventana se presenta un formulario en el que se solicita el nombre de usuario (nombre), la contraseña.

El usuario al llenar los dos campos, selecciona los dos siguientes y presiona el botón “entrar”.



El sistema verifica estos campos en la base de datos, comprueba si existe el usuario y el tipo de perfil que tiene.

El sistema de automatización muestra las opciones de este.

### **Registro de Nuevo Usuario**

#### **Descripción**

Si el usuario no se encuentra registrado en el sistema, el administrador puede registrarlo.

#### **Condiciones de ejecución**

Ninguna

#### **Entrada**

El administrador ingresa a la página de usuarios del sistema.

Ingresa a la opción de añadir usuarios.

Se llenan los campos nombre, usuario, contraseña.

Se verifica que todos los campos obligatorios estén llenos.

Se presiona el botón "insertar".

### **3.11.2. Pruebas de la segunda iteración**

#### **Historia de Usuario No. 008 Ingreso Comerciales**

#### **Creación de la lista de canciones para la automatización de la estación de radio**

#### **Descripción**

Al inicio del día y dentro del horario de 8 a 9 am, el director de programación ingresa al sistema para la elaboración de la lista de canciones a automatizar.

### **Condiciones de ejecución**

El director de programación debe estar registrado en el sistema.

El director de programación tiene que tener ya la lista de canciones para automatizar.

### **Entrada**

El director de programación ingresa al sistema con su usuario y contraseña.

Se realiza clic en la consola del sistema.

Se despliega la opción de búsqueda de las canciones para elaborar la lista de canciones para la automatización del día.

Se escoge el curso dando clic en “listar”.

### **3.11.3. Pruebas de la tercera iteración**

#### **Historia de Usuario No. 010 Ingreso a la Consola Majestad Studio.**

#### **Automatización del audio en vivo de la estación de radio Majestad**

#### **Descripción**

Al inicio del día y dentro del horario de 8 a 9 am, el director de programación ingresa al sistema para la elaboración de la lista de canciones a automatizar, el director de ventas a creado la lista de comerciales a automatizar

#### **Condiciones de ejecución**

El director de programación debe estar registrado en el sistema.

El director de programación ya ingreso la lista de canciones para el día.

El jefe de ventas ya ingreso la lista de comerciales para el día y estarán listos para ser escuchados.

## **Entrada**

El director de programación ingresa al sistema con su usuario y contraseña.

El jefe de programación deja el sistema para que este reproduzca las dos listas de archivos de audio, ya sea el de las canciones y también la de los comerciales para las próximas 24 horas.

## **3.12. IMPLEMENTACIÓN**

### **3.12.1. Hospedaje del sitio**

En esta parte de la investigación se dio a conocer y se evaluaron las distintas alternativas para la elección del hospedaje del sistema ya que el proveedor de la estación de radio en internet es de la empresa Punto Net.

### **3.12.2. Medición de la capacidad de automatización**

Un sistema de automatización implantado y colocado de manera local en una estación de radio puede llegar a automatizarla hasta de 4 años o de manera indefinida pero en la práctica eso es irrealizable porque el simple hecho que la programación de radio Majestad necesita colocar canciones nuevas o lanzar nuevas canciones; en la actualidad la música llega muy deprisa gracias al internet es por eso que en las pruebas de automatización se las realiza de manera diaria siempre y cuando la programación anterior se la realice en "papel" el día anterior solo las pruebas más "difíciles" de automatización se las realizo el fin de semana con una automatización de 48 horas, pero el sistema de automatización está preparado para automatizar la estación de radio de hasta un periodo de 4 días en caso de presentarse

### **3.12.3. Alojamiento del sitio**

Para que una página Web pueda ser levantada, debe estar “alojada” o almacenada en un computador y que el mismo esté conectado permanentemente a la Internet, de ahí que debe ser necesariamente un servidor. Un espacio en servidor con estas características recibe el nombre de host, y el servicio recibe el nombre de web hosting o alojamiento web.

Para poder hospedar el sistema en internet, primero es necesario conocer las posibilidades que existen y elegir la más conveniente.

Para este sistema se ha elegido un hosting con soporte para apache Tomcat y que puede tener un tráfico mensual de subida de 3GB, además alojamiento de Base de Datos en MySQL.

En un futuro no lejano la estación de radio estará pensada para alojar el sistema de automatización en un servidor propio acompañado de los archivos de audio.

## **CAPÍTULO IV**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **4.1. CONCLUSIONES**

- La sistematización y obtención de requisitos al principio parece algo tedioso, poco aplicable, sin embargo a medida que se va desarrollando el proyecto se concluye que es de suma importancia en un proyecto, ya que permite identificar pequeños detalles que muchas veces se pasan por alto, ocasionando pérdida de tiempo en el desarrollo del mismo.
  
- La ingeniería de requisitos es de suma importancia ya que contesta una de las interrogantes más importantes en el desarrollo del software ¿Qué es lo que realmente necesita el cliente?, no lo que supuestamente quiere y deja a un lado expectativas muy ambiciosas y poco realistas.
  
- Al hacer referencia al desarrollo de un prototipo funcional de un sistema informático, éste es de suma importancia ya que permite identificar errores, corregir requisitos preliminares para obtener un producto final acorde a las exigencias y necesidades del cliente.
  
- Con el desarrollo de la tecnología y la implementación de Internet, es mucho más fácil orientar un proyecto a la web ya que casi todos los involucrados se encontraron familiarizados con la navegación y sobretodo porque la Internet al ser una red de redes con cobertura mundial permite el acceso a este tipo de sistemas web sin importar la hora o el lugar en el que estemos, pues basta una conexión a internet para llevarlo a cabo.

## 4.2. RECOMENDACIONES

- Es recomendable tener una idea clara del proyecto final para poder presentarlo ya sea como prototipo o como una implementación durante el desarrollo y con todos los requisitos obtenidos.
- Se recomienda usar la metodología de Programación Extrema con la ingeniería de requisitos ya que aplicándolas juntas se optimiza mucho el tiempo, por la simple razón de que en los tiempos asignados en la programación el programador muestra los avances al cliente satisfaciendo los requisitos ya obtenidos.
- Se recomienda a los docentes de la carrera de sistemas reforzar los conocimientos en la programación en lenguaje Java, además de que es muy potente esta herramienta de desarrollo, permite crear aplicaciones muy versátiles con un bajo coste y en un menor tiempo.
- Cumplir los tiempos establecidos para las reuniones y presentación de avances mientras se desarrolla el proyecto, son aspectos importantes que no hay que dejar de lado, ya que con el paso del tiempo los requisitos pueden cambiar o la persona encargada de un departamento puede ser removida y por ende se estaría en la necesidad de reformular nuevos requisitos de acuerdo a nuevas exigencias y necesidades.
- Mantener una secuencia de todas las fases de la ingeniería de requisitos es imprescindible en el desarrollo de un proyecto, ya que si se omite un paso se pueden crear malas interpretaciones o ambigüedad en los requisitos.

## **BIBLIOGRAFÍA**

### **CONSULTADA**

HOFMANN, Hubert, (1993), "Requirements Engineering", Universidad de Zurich

MALAN, Ruth, (1999), "Functional Requirements and Use Cases". Hewlette-Packard Company

MCCONNELL, Steve, (1996). "Rapid Development: Taming Wild Software Schedules", Primera Edición, Editorial Redmond

OBERG, Roger, (1998), "Applying Requirements Management with Use Cases". Rational Software Corporation

PRESSMAN, Roger, (S/N), "Ingeniería de software: un enfoque práctico", Cuarta Edición, McGrawHill

SAIEDIAN, H., (1999), "Requirements Engineering: Making the connection between the software developer and customer", Universidad de Nebraska

SENN, James A., (1992), "Análisis y Diseño de Sistemas de Información". Segunda Edición. McGraw Hill

STELLMAN, Andrew, (2005), "Applied Software Project Management", Cambridge

WIEGERS, Karl E., (2003). "Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle", Segunda Edición, Editorial Redmond

## **VIRTUAL**

**Departamento de Lenguajes y Sistemas Informáticos**  
**Escuela Técnica Superior de Ingeniería Informática**  
**Universidad de Sevilla**  
**Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio**  
**Comparativo**

Fecha de acceso: Noviembre 23 del 2011

Disponible en: [www.lsi.us.es/docs/informes/LSI-2002-4.pdf](http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf)

### **Apache Http Server Project**

Fecha de acceso: Noviembre 23 del 2011

Disponible en: <http://httpd.apache.org/download.cgi>

### **Seminario Web de MySQL**

Fecha de acceso: Noviembre 23 del 2011

Disponible en: <http://www.mysql.com/>

### **Apache OpenOffice**

Fecha de acceso: Noviembre 30 del 2011

Disponible en: <http://es.openoffice.org/>

### **Elevate Recording**

Fecha de acceso: Noviembre 30 del 2011

Disponible en: <http://www.m-audio.com/>

### **Eclipse**

Fecha de acceso: Enero 20 del 2012

Disponible en: <http://www.eclipse.org/>



## **javaHispano**

Fecha de acceso: Enero 20 del 2012

Disponible en: <http://www.javahispano.org/>

## **Ubuntu**

Fecha de acceso: Enero 20 del 2012

Disponible en: <http://www.ubuntu.com/>

# **ANEXOS**

---

**Majesoft 1.0**

---

**Sistema de Automatización para el audio en vivo para Radio Majestad 01**

**Tesis: Diego Pule**

**Versión 1.0**

## Historial de Revisiones

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
10/10/2011	0.8	Versión preliminar del glosario, a falta de completar con próximas reuniones	Diego Pule López
21/10/2011	0.9	Versión para ser aprobada el día 24/11	Fabián Custode
24/10/2011	1.0	Versión aprobada por el Stakeholder	Diego Pule López
25/11/2011	2.0	Versión modificada con la incorporación de nueva terminología derivada de modificaciones en los requisitos	Diego Pule López
12/12/2011	2.1	Versión revisada en la primera iteración de construcción. Términos actualizados	Diego Pule López
25/12/2011	2.9	Versión para ser aprobada por el Stakeholder en la segunda iteración de la fase de construcción	Diego Pule López
02/01/2011	3.0	Versión revisada para la segunda iteración, pendiente de ser validada por el Stakeholder.	Diego Pule López

---

# Majesoft 1.0

---

## Tabla de Contenidos

CAPÍTULO 1.....	1
SISTEMA DE AUTOMATIZACIÓN PARA EL AUDIO EN VIVO PARA RADIO MAJESTAD 01 .....	101
TESIS: DIEGO PULE .....	101
VERSIÓN 1.0.....	101
HISTORIAL DE REVISIONES.....	1
TABLA DE CONTENIDOS .....	2
GLOSARIO .....	4
INTRODUCCIÓN.....	4
PROPÓSITO .....	4
ALCANCE .....	4
REFERENCIAS .....	4
ORGANIZACIÓN DEL GLOSARIO .....	4
DEFINICIONES .....	4
AUTOMATIZACIÓN .....	4
AVANCE .....	4
CASTING .....	5
COMERCIAL .....	5
CUE .....	5
CUÑA .....	5
DEMO .....	5
INTERNO .....	5
MENCIÓN .....	6
OPERADOR .....	6
PAUTA .....	6
PICCOLO .....	6

---

# Majesoft 1.0

---

PRODUCCIÓN	6
PRODUCTOR	6
PRODUCTOR / OPERADOR	6
PROGRAMA	7
PROMO	7
REAIL	7
TANDA	7
ESTEREOTIPOS.....	7

## Glosario

### Introducción

Este documento recoge todos y cada uno de los términos manejados a lo largo de todo el proyecto de desarrollo del sistema de automatización de audio en vivo de radio Majestad de la ciudad de Quito. Se trata de un diccionario informal de datos y definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

### Propósito

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo del sistema de automatización. También sirve como guía de consulta para la clarificación de los puntos conflictivos del proyecto.

### Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos para la empresa Radio Majestad. De tal modo que la terminología empleada en el departamento de gerencia, el departamento de ventas, el departamento de producción, y el operador, sea reflejada con claridad en este documento.

### Referencias

El presente glosario hace referencia al siguiente documento:

- Documento Visión del Proyecto Automatización de Audio en vivo.

### Organización del Glosario

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional del Español.

### Definiciones

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto de desarrollo de un sistema para el sistema de automatización del audio en vivo de Radio Majestad.

### Automatización

Es la reproducción de dos listas de archivos de audio los cuales se reproducirán a determinada fecha y hora.

### Avance

Es un archivo de audio que puede durar de entre 1 a máximo 30 segundos en los cuales se de una pequeño adelanto de el

programa de radio que va a continuar también puede ser de un anuncio comercial.

### **Casting**

Pequeña prueba de un programa puede durar hasta media hora.

### **Comercial**

Anuncio de una determinada marca comercial, el comercial puede durar hasta un minuto, todo depende del precio que asigne el propietario de la estación.

### **Cue**

Es una palabra inglesa que significa escuchar solo adentro de la estación es un sonido que no se emite al aire.

### **Cuña**

Tiene una significación igual a comercial pero para elaborar la presente tesis se usan las dos palabras.

### **Demo**

Es un lista a probar durante el audio que se emite en vivo en la estación.

### **Interno**

Tiene un significado parecido al cue pero la diferencia principal es que esta dura mucho más tiempo, es más existen programas completos que se realizan en interno y que muchas veces no se emiten al aire.



**Mención**

Es una locución que dura hasta 30 segundos pero ésta se la realiza lo más sencilla posible sin efectos de sonido.

**Operador**

Es la persona encargada de monitorear y/o revisar que el sistema esté funcionando a la perfección.

**Pauta**

Lista de comerciales que se van a transmitir en determinado momento, esta lista por lo general no se la cambia.

**Piccolo**

Forma en la cual el comercial es enviado a la estación de radio.

**Producción**

Se dice al tiempo requerido y usado para la creación de una cuña publicitaria

**Productor**

Es la persona encargada de realizar la producción

**Productor / Operador**

Es la persona que se encarga de ayudar al productor a realizar la producción.

## **Programa**

Es el espacio de tiempo usado para colocar la música junto con la publicidad que puede ser de hasta 4 horas.

## **Promo**

Es una avance muy elaborado que se realiza para el lanzamiento de una producto comercial o de un programa de radio

## **Reail**

Se usaba antes para expresar el uso de la grabadora magnetofónica pero ahora se lo usa para determinar el archivo que se encuentra en el disco duro, tiene referencia con los comerciales.

## **Tanda**

Lista de comerciales o lista de canciones a reproducir, se las usa de manera indistinta

## **Estereotipos**

[This section contains or references specifications of Unified Modeling Language (UML) stereotypes and their semantic implications—a textual description of the meaning and significance

---

## Majesoft 1.0

---

of the stereotype and any limitations on its use—for stereotypes already known or discovered to be important for the system being modeled. The use of these stereotypes may be simply recommended or perhaps even made mandatory; for example, when their use is required by an imposed standard or when it is felt that their use makes models significantly easier to understand. This section may be empty if no additional stereotypes, other than those predefined by the UML and the Rational Unified Process, are considered necessary.]

---

**Majesoft 1.0**

---

**Sistema de Automatización para el audio en vivo para Radio Majestad 01**

**Tesis: Diego Pule**

**Versión 1.0**

## Historial de Revisiones

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
22/10/2011	0.9	Propuesta inicial del documento. Visión con las primeras capturas de requisitos funcionales del sistema.	Diego Pule
28/10/2011	1.0	Versión 1.0 en estado de complementación para su aprobación.	Diego Pule
29/10/2011	1.0	Versión 1.0 para la aprobación al final de la fase de inicio	Diego Pule
10/12/2011	2.9	Versión modificada en la segunda iteración de construcción. Pendiente de revisión por el Stakeholder.	Diego Pule
02/012011	3.0	Versión revisada para la segunda iteración de construcción. Pendiente de validación por el Stakeholder	Diego Pule

---

# Majesoft 1.0

---

## Tabla de Contenidos

CAPÍTULO 1.....	1
SISTEMA DE AUTOMATIZACIÓN PARA EL AUDIO EN VIVO PARA RADIO MAJESTAD 01 .....	101
TESIS: DIEGO PULE .....	101
VERSIÓN 1.0.....	101
HISTORIAL DE REVISIONES.....	1
TABLA DE CONTENIDOS .....	2
GLOSARIO .....	4
INTRODUCCIÓN.....	4
PROPÓSITO .....	4
ALCANCE .....	4
REFERENCIAS .....	4
ORGANIZACIÓN DEL GLOSARIO .....	4
DEFINICIONES .....	4
AUTOMATIZACIÓN .....	4
AVANCE .....	4
CASTING .....	5
COMERCIAL .....	5
CUE .....	5
CUÑA .....	5
DEMO .....	5
INTERNO .....	5
MENCIÓN .....	6
OPERADOR .....	6
PAUTA .....	6
PICCOLO .....	6

---

# Majesoft 1.0

---

PRODUCCIÓN	6
PRODUCTOR	6
PRODUCTOR / OPERADOR	6
PROGRAMA	7
PROMO	7
REAIL	7
TANDA	7
ESTEREOTIPOS.....	7
SISTEMA DE AUTOMATIZACIÓN PARA EL AUDIO EN VIVO PARA RADIO MAJESTAD 01 .....	9
TESIS: DIEGO PULE .....	9
VERSIÓN 1.0.....	9
HISTORIAL DE REVISIONES.....	1
TABLA DE CONTENIDOS .....	2
VISIÓN.....	5
INTRODUCCIÓN.....	5
PROPÓSITO	5
ALCANCE	5
DEFINICIONES, ACRÓNIMOS, Y ABREVIACIONES	5
REFERENCIAS	6
POSICIONAMIENTO.....	6
OPORTUNIDAD DE NEGOCIO	6
SENTENCIA QUE DEFINE EL PROBLEMA	7
SENTENCIA QUE DEFINE LA POSICIÓN DEL PRODUCTO	8
DESCRIPCIÓN DE STAKEHOLDERS (PARTICIPANTES EN EL PROYECTO) Y USUARIOS .....	8
RESUMEN DE STAKEHOLDERS	9

---

# Majesoft 1.0

---

RESUMEN DE USUARIOS	9
ENTORNO DE USUARIO	10
PERFIL DEL STAKEHOLDERS	10
<i>Representante de Gerencia</i>	10
PERFILES DE USUARIO	11
<i>Jefe de Ventas</i>	11
<i>Jefe de Programación</i>	11
<i>Jefe de Programas</i>	12
<i>Locutor</i>	12
<i>Operador</i>	12
<i>Jefe de Ventas apoyo</i>	13
DESCRIPCIÓN GLOBAL DEL PRODUCTO .....	14
PERSPECTIVA DEL PRODUCTO	14
RESUMEN DE CARACTERÍSTICAS	14
COSTO Y PRECIO	14
DESCRIPCIÓN GLOBAL DEL PRODUCTO .....	15
5.1 <a href="#">DEPARTAMENTO DE GERENCIA</a>	15
5.2 <a href="#">DEPARTAMENTO DE VENTAS</a>	15
5.3 <a href="#">DEPARTAMENTO DE PRODUCCIÓN</a>	15
5.4 <a href="#">GESTIÓN DE AUTOMATIZACIÓN</a>	15
5.5 <a href="#">GESTIÓN DE VENTAS</a>	16
OTROS REQUISITOS DEL PRODUCTO .....	16
REQUISITOS DE SISTEMA	16
REQUISITOS DE DESEMPEÑO	17
REQUISITOS DE ENTORNO	17
REQUISITOS DE DOCUMENTACIÓN .....	17
A.    ATRIBUTOS DE CARACTERÍSTICAS .....	17



**Tesis: Diego Pule**

## **Introducción**

### **Propósito**

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de automatización del audio en vivo de Radio Majestad del Distrito Metropolitano de Quito. El documento se centra en la funcionalidad requerida por los participantes en el proyecto

Esta funcionalidad se basa principalmente en la gestión de automatización de la estación de forma tal que la estación trabaje o funcione de manera automática en un determinado tiempo

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso.

### **Alcance**

El documento Visión se ocupa, como ya se ha apuntado, del sistema de la automatización del audio en vivo de la estación. Dicho sistema será desarrollado por el egresado Diego Pule.

El sistema permitirá a los encargados de la estación controlar todo lo relativo a la edición y colocación de listas de canciones y de comerciales en la estación mismos que serán programados con anterioridad y con los requerimientos mencionados.

### **Definiciones, Acrónimos, y Abreviaciones**

---

# Majesoft 1.0

---

XP: Son las siglas de Programación extrema. Se trata de una metodología para describir el proceso de desarrollo de software.

## Referencias

- Glosario.
- Plan de desarrollo de software.
- XP (Extreme Programming).
- Diagrama de casos de uso.

## Posicionamiento

### Oportunidad de Negocio

Este sistema permitirá a la estación automatizar el audio en vivo (principalmente dos listas una de canciones previamente seleccionadas y la otra lista de comerciales previamente elaborada) lo cual supondrá un acceso rápido y sencillo a las listas de reproducción, gracias a interfaces gráficas sencillas y amigables.

Además, las listas accedidas estarán siempre actualizadas, lo cual es un factor muy importante para poder llevar un control de la estación.

---

# Majesoft 1.0

---

El sistema también permite al gerente acceder a los servicios del sistema de manera fácil y confiable.

## Sentencia que define el problema

El problema de	Controlar las canciones y comerciales, de forma que se administran y enlistan de manera fácil y confiable. Gestionar las órdenes de publicidad de clientes y/o agencias de publicidad. Gestionar los pedidos de los locutores. Gestionar un seguimiento de la facturación de la estación.
Afecta a	Departamento de programación, Jefe de ventas, Técnicos de la estación, Director de programación, Usuarios de ventas, Gerente, Locutores, etc.
El impacto asociado es	Listar todas la canciones a colocar en determinada fecha exacta con la respectiva lista de comerciales además de mostrar un reporte de los accesos válidos de las personas que ingresaron o intentaron ingresar al sistema durante su funcionamiento, el sistema tendrá además en el caso de algún acontecimiento un fácil reinicio no complicado frente a la persona que se encuentre en ese momento
Una solución adecuada sería	Automatizar el proceso, usando una red local con una base de datos accesible desde los distintos puntos de la red (tres exactamente) usando interfaces amigables y sencillas con las que acceder a dichos procesos.

---

# Majesoft 1.0

---

## Sentencia que define la posición del Producto

Para	Departamento de programación, Operadores, Técnico de programación, Encargados de transporte, Usuarios de ventas de cada región, Departamento de Contabilidad / Facturación.
Quienes	A quién controla: las órdenes de pedidos musicales y la facturación.
El nombre del producto	Es un sistema software (Majesoft).
Que	Automatiza las listas necesarias para la respectiva reproducción
No como	El sistema actual. (Jazler 6.0)
El producto	Permite gestionar las distintas listas de reproducción de la estación mediante una interfaz gráfica sencilla y amigable. Además proporciona un acceso rápido y actualizado a la información de dichas listas.

## Descripción de Stakeholders (Participantes en el Proyecto) y Usuarios

Para proveer de una forma efectiva y confiable los servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No

---

# Majesoft 1.0

---

describe sus requisitos específicos ya que éstos se capturan en otro ítem.

## Resumen de Stakeholders

Nombre	Descripción	Responsabilidades
Fabián Custode	Representante de Radio Majestad	El stakeholder realiza: Representa a todos los usuarios posibles del sistema. Seguimiento del desarrollo del proyecto. Aprueba requisitos y funcionalidades

## Resumen de Usuarios

Nombre	Descripción	Stakeholder
<u>Gerencia</u>	Responsable del Departamento de Gerencia	<u>GERENCIA</u>
<u>Jefe de Ventas</u>	Encargado de las ventas de la estación, administración de facturas contabilidad y envió de estos documentos a la matriz	<u>VENTAS</u>
<u>Programación</u>	Encargado directo de la lista de canciones a colocar en lista en otra máquina y presenta un reporte al final del año	<u>PROGRAMACIÓN</u>
<u>Programas</u>	Responsable de ventas del producto a los clientes, mediante visitas al domicilio del cliente. Informa de las ofertas y confecciona las órdenes de pedido.	<u>PROGRAMAS</u>

---

# Majesoft 1.0

---

<u>Operador</u>	Supervisor del Departamento de Ventas, encargado de otorgar incentivos y del control de estadísticas.	<u>Ventas</u>
<u>Respaldo</u>	Encargado de la facturación y cobranzas, política de cobro de los clientes.	<u>Contabilidad</u> / <u>Facturación</u>

## Entorno de usuario

Los usuarios entrarán al sistema identificándose sobre un ordenador con un sistema operativo Windows XP y tras este paso entrarán a la parte de aplicación diseñada para cada uno según su papel en la empresa. Este sistema es similar a cualquier pagina web de internet y por tanto los usuarios estarán familiarizados con su entorno.

## Perfil del Stakeholders

### Representante de Gerencia

<b>Representante</b>	Fabián Custode
<b>Descripción</b>	Representante Global de la estación de radio Majestad FM
<b>Tipo</b>	Gerente
<b>Responsabilidades</b>	Encargado de mostrar las necesidades de cada usuario del sistema. Además, lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema
<b>Criterio de Éxito</b>	<b>A definir por el cliente</b>
<b>Grado de participación</b>	Revisión de requerimientos, estructura del sistema
<b>Comentarios</b>	Ninguno

---

# Majesoft 1.0

---

## Perfiles de Usuario

### Jefe de Ventas

<b>Representante</b>	<u>Ventas</u>
<b>Descripción</b>	Jefe del Departamento de ventas de la estación.
<b>Tipo</b>	Jefe de ventas.
<b>Responsabilidades</b>	Responsable del Departamento de Ventas encargada de enlistar lo comerciales antes se los hacía a mano, luego con el programa Jazler, ahora con los requisitos se cambia al sistema, sólo se encarga de la lista de comerciales.
<b>Criterio de Éxito</b>	A definir por el cliente
<b>Grado de participación</b>	A definir por el cliente
<b>Comentarios</b>	Ninguno

### Jefe de Programación

<b>Representante</b>	<u>Programación</u>
<b>Descripción</b>	Responsable de la lista de canciones a programar a diario, por semana y por meses.
<b>Tipo</b>	Usuario anexo al sistema es decir la lista lo realiza en otro equipo.
<b>Responsabilidades</b>	No repetir canciones, complacer los gustos de los oyentes, tener una buena conexión con los locutores, verificar que no se repitan listas completas sólo se encarga de la lista de canciones
<b>Criterio de Éxito</b>	A definir por el cliente
<b>Grado de participación</b>	A definir por el cliente
<b>Comentarios</b>	Ninguno.

---

# Majesoft 1.0

---

## Jefe de Programas

<b>Representante</b>	<u>Gerencia</u>
<b>Descripción</b>	Responsable de los programas y tipos de música.
<b>Tipo</b>	Usuario experto.
<b>Responsabilidades</b>	Encargado directo de los programas de la estación, control de horarios, preparación de los mismos, gestión de calamidades
<b>Criterio de Éxito</b>	A definir por el cliente
<b>Grado de participación</b>	A definir por el cliente
<b>Comentarios</b>	Ninguno.

## Locutor

<b>Representante</b>	<u>Programación</u>
<b>Descripción</b>	Anuncio de canciones y menciones
<b>Tipo</b>	Usuario básico.
<b>Responsabilidades</b>	Responsable de ventas del producto a los clientes, mediante visitas a domicilio del cliente. Informa de las ofertas y confecciona las órdenes de pedido. También participa en las incidencias de pedidos poniéndose en contacto con el cliente para la resolución de los mismos. Puede cancelar pedidos en estado de elaboración.
<b>Criterio de Éxito</b>	A definir por el cliente
<b>Grado de participación</b>	A definir por el cliente
<b>Comentarios</b>	Ninguno.

## Operador



---

# Majesoft 1.0

---

<b>Representante</b>	<u>Ventas</u>
<b>Descripción</b>	Operadora de ventas de los productos
<b>Tipo</b>	Usuario experto.
<b>Responsabilidades</b>	Responsable de ventas de los paquetes a los clientes a través de venta directas como por ejemplo visitas personales. Informa de las ofertas , elabora las órdenes de pedido. También participa en las incidencias de la tandas poniéndose en contacto con el cliente para la resolución de lo mismos.
<b>Criterio de Éxito</b>	A definir por el cliente
<b>Grado de participación</b>	A definir por el cliente
<b>Comentarios</b>	Ninguno.

## Jefe de Ventas apoyo

<b>Representante</b>	<u>Ventas</u>
<b>Descripción</b>	Jefe del Departamento de Ventas de un paquete determinado.
<b>Tipo</b>	Usuario experto.
<b>Responsabilidades</b>	Jefe del departamento de ventas, verificar los datos de los cliente promociones y paquetes.
<b>Criterio de Éxito</b>	A definir por el cliente
<b>Grado de participación</b>	A definir por el cliente
<b>Comentarios</b>	Ninguno.

---

# Majesoft 1.0

---

## Descripción Global del Producto

### Perspectiva del producto

El producto a desarrollar es un sistema de automatización para la empresa *Radio Majestad*, con la intención de automatizar su funcionamiento. Las áreas a tratar por el sistema son: gerencia, gestión de dos listas de reproducción: canciones y comerciales, contabilidad, facturación y operación (locución).

### Resumen de características

A continuación se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

<b>Beneficio del cliente</b>	<b>Características que lo apoyan</b>
Mayor facilidad de automatización	Aplicación web de fácil intuición.
Gestión de listas a programar.	Sistema de optimización con programación de listas de canciones y comerciales
Mayor facilidad para la gestión de comerciales.	Base de datos centralizada con la información de los procesos.
Posibilidad de fácil reanudación en caso de pérdida de fluido eléctrico.	Aplicación web para cambiar las listas de canciones.
Automatización completa de todo el audio en vivo.	Sistema automatizado de varias horas 18 horas más o menos.

### Costo y precio

*El costo del desarrollo del sistema asume en su totalidad el desarrollador:*

*Egresado Diego Pule*

## Descripción Global del Producto

### 5.1 [Departamento de Gerencia](#)

Departamento encargado de la gestión administrativa, asignación de trabajo, aprobación del trabajo.

### 5.2 [Departamento de Ventas](#)

Departamento responsable de las ventas de la estación contratos de comerciales a transmitir además de la orden de menciones

### 5.3 [Departamento de Producción](#)

Departamento que dirige y gestiona la creación de identificativos para la estación además de crear los avances de la estación estos se encuentran dentro de la lista de canciones y dependiendo del promocional también pueden estar incluidos en las tandas de comerciales, muchas veces las listas de canciones pasan por este departamento ya que se los respaldan desde este departamento.

### 5.4 [Gestión de Automatización](#)

Es la principal función del sistema de la estación ya que se encarga de gestionar la reproducción de las canciones a determinada hora ya sea partiendo desde las 9 horas por ejemplo, pero dependiendo del jefe de producción, del gerente o del productor. Es una lista que ha simple vista parece sencilla pero la persona responsable de esta lista tiene que responder sobre la relevancia de las canciones. La lista de comerciales debe reproducirse a determinada hora y fecha previamente aprobada por el gerente y/o jefe de ventas, esta lista es muy importante para las utilidades de la estación.

#### 5.4.1 [Automatización de lista de canciones](#) .

La lista de canciones será previamente elaborada por el productor el cuál luego la pasará al gerente que será aprobada aunque muchas veces se pasa por alto este proceso, ya la lista elabora el productor se encarga de copiar los archivos al computador que luego serán enlistados para su reproducción a determinada hora al momento de presentar la presente tesis la lista de canciones se iniciaba a las 9 horas, en caso de algún incidente como el corte del fluido eléctrico la lista siempre se reproduce

desde la primera canción fácilmente se puede adelantar la canción hasta la siguiente del corte eléctrico.

#### 5.4.2 Automatización de comerciales

La lista de canciones será previamente elaborada por el departamento de ventas el cuál luego la pasará al gerente que será aprobada nunca se pasa por alto este proceso ya la lista elabora el productor se encarga de copiar los archivos al computador que luego serán enlistados para su reproducción a determinada hora y fecha exactas al momento de presentar la presente tesis la lista de comerciales era de 24 horas, en caso de algún incidente como el corte del fluido eléctrico la lista seguirá reproduciendo los comerciales a las misma hora exacta los comerciales que no se transmitieron será reportados para el siguiente día.

#### 5.5 Gestión de Ventas

El departamento de ventas dispone de dos servicios distintos de ventas:

##### 5.5.1 Información de los comerciales transmitidos

Un representante de ventas o la operadora pueden elaborar la lista o bien para su propios clientes: clientes directos uno solo o varios, clientes por agencias de publicidad o hasta por paquetes que son elaborados por el jefe de ventas o por el gerente

##### 5.5.2 Gestión de los datos de los clientes

Un representante de ventas o el gerente pueden modificar los datos de los clientes, ya sea dirección RUC o paquete, esta información es sólo para uso interno y no tiene ningún valor para la facturación ya que ésta se la realiza con otro sistemas propio de la estación

## Otros Requisitos del Producto

### Requisitos de Sistema

[A definir por el cliente]

---

# Majesoft 1.0

---

## Requisitos de Desempeño

[A definir por el cliente]

## Requisitos de Entorno

[A definir por el cliente]

## Requisitos de Documentación

[A definir por el cliente]

### A. Atributos de Características

Número y nombre de la característica	Estado	Beneficio	Esfuerzo	Riesgo	Estabilidad	Asignación
5.1 Depart. De Gerencia	Propuesta: Sí Aprobada: Sí Incorporada: No	Útil	Bajo	[A definir por el cliente]	[A definir por el cliente]	Ninguna

## Majesoft 1.0

5.2 Depart. de Ventas	Propuesta: Sí Aprobada: Sí Incorporada: No	Útil	Bajo	[A definir por el cliente]	[A definir por el cliente]	Ninguna
5.3 Depart. de Producción	Propuesta: Sí Aprobada: Sí Incorporada: No	Importante	Medio	[A definir por el cliente]	[A definir por el cliente]	Ninguna
5.4 Gestión de Automatización	Propuesta: Sí Aprobada: Sí Incorporada: Sí	Crítica	Alto	[A definir por el cliente]	[A definir por el cliente]	María Augusta Viteri Fabián Custode
5.4.1 Automatización lista de canciones	Propuesta: Sí Aprobada: Sí Incorporada: Sí	Crítica	Alto	[A definir por el cliente]	[A definir por el cliente]	Fabián Custode Jefe de Producción
5.4.2 Automatización de listas de comerciales	Propuesta: Sí Aprobada: Sí Incorporada: No	Crítica	Alto	[A definir por el cliente]	[A definir por el cliente]	María Augusta Viteri Fabián Custode
5.5 Gestión de Ventas	Propuesta: Sí Aprobada: Sí Incorporada:	Crítica	Alto	[A definir por el cliente]	[A definir por el cliente]	María Augusta Viteri Fabián Custode

---

## Majesoft 1.0

---

	Sí			cliente]	cliente]	
5.5.1 Información de comerciales transmitidos	Propuesta: Sí Aprobada: Sí Incorporada: Sí	Útil	Medio	[A definir por el cliente]	[A definir por el cliente]	María Augusta Viteri Fabián Custode
5.5.2 Gestión de datos de clientes	Propuesta: Sí Aprobada: Sí Incorporada: No	Importante	Medio	[A definir por el cliente]	[A definir por el cliente]	María Augusta Viteri Fabián Custode

## ANEXO C

### Listas de colaboración:

#### Gerente

- Acceso restringido, avisos de quienes ingresaron al sistema con hora y posibles errores.
- La lista de canciones tiene que ser lo mas detallada posible.
- Solo él tiene acceso como administrador.
- Colores fáciles de identificar.
- Que el programa sea único para la estación.

#### Gerente de ventas

- Quiere tener un control sencillo al sistema, único y que pueda verificar las ventas.
- El programa no requiere que imprima reportes.
- El locutor no tiene que tener acceso a las ventas que ha realizado, solo el gerente y la gerente de ventas.

#### Locutor de la estación

- Manejo sencillo de la estación.
- Total accesos a las canciones.
- Poder modificar la lista de reproducción a su antojo.
- Acceso a toda la información de las canciones.

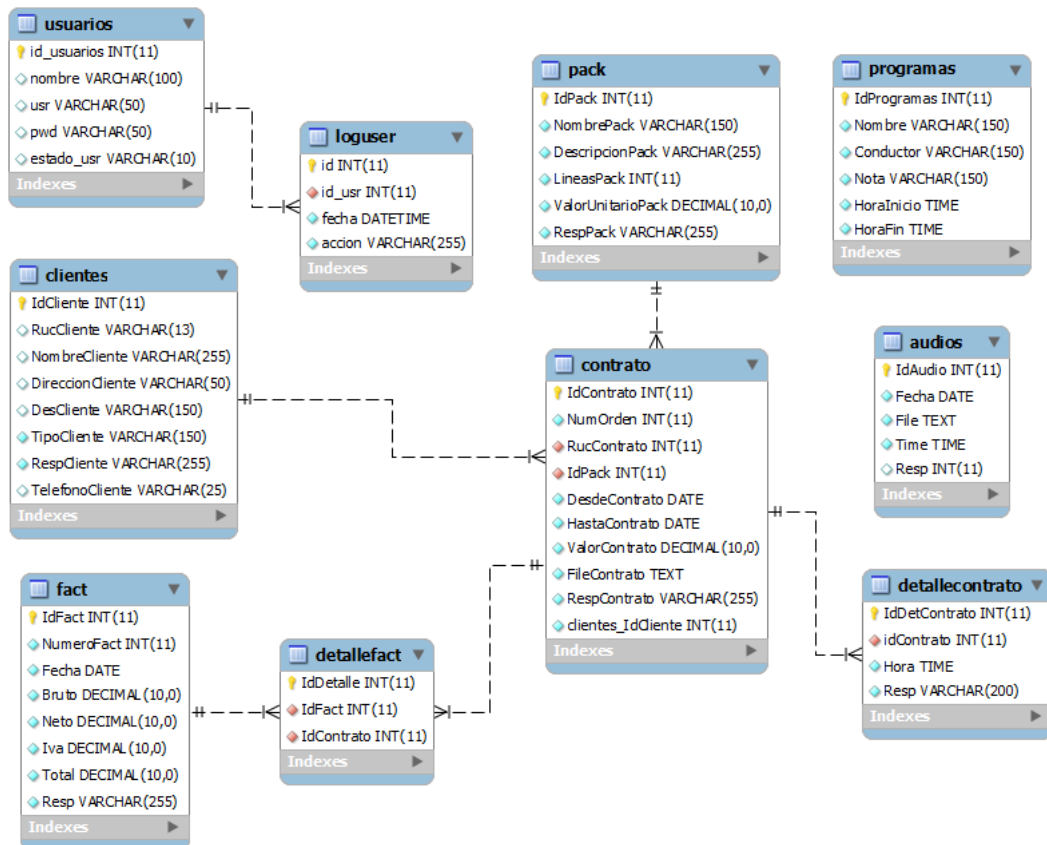
#### Director de programación

- Solo él tiene acceso a la lista de canciones y solo él puede modificarlas.



## ANEXO D

### BASE DE DATOS



## ANEXO E



89.7 QUITO	96.1 STO. DOMINGO	104.1 TUNGURAHUA	107.5 LOS RIOS
88.5 MANABI	97.5 ESMERALDAS	104.1 COTOPAXI	

### CERTIFICACIÓN

Yo, Sr. Fabián Custode, Gerente de radio Majestad, luego de revisar el informe técnico del jefe de Sistemas de la estación y a petición verbal del interesado, tiendo a bien certificar que:

El proyecto de Grado "Desarrollo de un prototipo del Sistema de Automatización para el audio de emisión en vivo de la Radio Majestad del Distrito Metropolitano de Quito aplicando la Ingeniería de requisitos en todas sus fases.", elaborado por el Sr. Pule López Diego Oswaldo, alumno egresado de la carrera de Sistemas e Informática de la Escuela Politécnica del Ejercito; es de beneficio para esta estación de radio y permitió conocer un prototipo que satisface todas los requerimientos que necesita la estación para su correcta automatización.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo la parte interesada hacer uso del presente certificado de la forma que estime conveniente.

Sr. Fabian Custode

GERENTE DE RADIO MAJESTAD



FC/ams

www.radiomajestad.com

#### QUITO:

Av. 6 de Diciembre 3981 y Checoslovaquia  
Telefax: (02) 2269918 / (02) 2440598  
Estudio: (02) 2241041

#### SANTO DOMINGO:

Av. Quito y Cocaniguas Edif. Radio Majestad  
Telfs.: (02) 2766583 / (02) 2762802 Fax: (02) 2750118  
Estudio: (02) 2759800

**Latacunga, mayo 2012**

**AUTOR:**

---

**DIEGO OSWALDO PULE LÓPEZ  
171654291-3**

**DIRECTOR DE CARRERA:**

---

**Ing. Santiago Jácome**

**UNIDAD DE ADMISIÓN Y REGISTRO:**

---

**Dr. Rodrigo Vaca Corrales**