



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE INGENIERÍA

**DESARROLLO DE UNA APLICACIÓN WEB  
PARA LA PUBLICACIÓN DE PODCASTS EN  
LÍNEA**

**INFORME DE TRABAJO PROFESIONAL**

QUE PARA OBTENER EL TÍTULO DE  
**INGENIERO EN COMPUTACIÓN**

P R E S E N T A :

**JAVIER ALEJANDRO GARCÍA ROMERO**

ASESORA:

M.C. MA. JAQUELINA LÓPEZ BARRIENTOS



CIUDAD UNIVERSITARIA, MÉXICO D.F.  
2008.

1. Panorama.....	5
2. Objetivo.....	5
3. Introducción .....	6
4. Definición de la problemática.....	7
5. Justificación .....	8
5.1 Mercado por grupos.....	8
5.2 Ventaja competitiva.....	9
5.3 Primeros en México y Latinoamérica.....	11
5.4 Beneficios que traerá el desarrollo al usuario final (podcaster).....	11
5.5 Beneficios que traerá el desarrollo al administrador.....	12
6. Metodología del proyecto.....	13
6.1 Metodología y diseño Orientado a objetos.....	13
6.2 Consideraciones para adoptar un estándar o construir una metodología.....	14
6.3 Importancia de las herramientas CASE.....	16
6.4 Ingeniería de Software.....	16
7. Análisis.....	17
7.1 Requerimientos.....	18
7.1.1 Requerimientos en la administración.....	18
7.2 Requerimientos sobre el servicio para el usuario final.....	18
7.3 Requerimientos para la interfaz.....	19
8. Planeación.....	20
8.1 Creación del grupo de trabajo.....	22
8.2 Estimación de Tiempos y Costos.....	22
Definir procedimientos de cambio.....	24
8.3 Acordar criterios de aceptación.....	24
9. Diseño.....	25
9.1 Diagrama Entidad Relación.....	27
9.2 Modelo de Casos de Uso.....	27
9.2.1 Especificaciones de Alto Nivel.....	27
10. Desarrollo.....	30
10.1 Metodología empleada para el desarrollo.....	30
10.2 Ciclo de desarrollo del sistema.....	30
10.2.1 Framework.....	30
10.2.2 Requerimientos para la instalación.....	30
10.2.3 Creación del proyecto.....	31
10.2.4 Creación del modelo.....	31
10.2.5 Creación de un módulo.....	32
10.2.6 Arquitectura MVC.....	32
11. Pruebas.....	34
11.1 Plan de Pruebas.....	35
12. Documentación.....	37
12.1 Documentación del desarrollo.....	37
12.2 Documentación de código.....	37
13. Manutención del sistema.....	38
14. Seguridad.....	40
14.1 Seguridad de la aplicación.....	40
14.1.1 Evaluación de vulnerabilidades.....	40
14.1.2 Factores que facilitan intrusiones .....	40
14.1.3 Control de acceso al servidor.....	40
14.1.4 Respaldos.....	40
14.2 Seguridad de los usuarios.....	40
15. Resolución de problemas.....	41
16. Participación Profesional.....	42

<b>17. Resultados y Aportaciones.....</b>	<b>43</b>
<b>18. Conclusiones.....</b>	<b>44</b>
<b>ANEXO.....</b>	<b>46</b>
<b>19. Bibliografía.....</b>	<b>47</b>

## DESARROLLO DE UNA APLICACIÓN WEB PARA LA PUBLICACIÓN DE PODCASTS EN LÍNEA

### 1. Panorama

Internet es la última gran tecnología global que no deja de crecer y redescubrirse día a día. Por tal razón, debemos de aprovechar las tendencias que le va marcando al mundo. Una de las más recientes, y en pleno crecimiento, son los podcasts.

Los podcasts, con el uso de archivos de audio y video, facilitan la distribución de contenido a la medida y en tiempo real a diversos grupos de usuarios de Internet.

La flexibilidad que brindan los podcasts en el manejo de la información, le permite una amplia gama de aplicaciones. Por citar un ejemplo, los podcasts han sido utilizados en los últimos años para enseñar en los salones de clases. Tal es el caso de una escuela en Wells, Maine en los Estados Unidos, en donde algunos profesores distribuyen sus clases en podcasts como material de apoyo para sus alumnos<sup>1</sup>. La importancia que han generado iguales iniciativas en otros puntos del país, trajo como consecuencia la creación del EPN (The Education Podcast Network) que busca almacenar todos los recursos creados en podcasts para la Educación.

En México, también se han encontrado iniciativas similares. Por ejemplo, en la Escuela Nacional Preparatoria No 1 'Gabino Barreda' de la UNAM, una clase de quinto año de Inglés ha trabajado con podcasts para mejorar la comprensión auditiva en los alumnos, así como para desarrollar la habilidad de hablar en otro idioma<sup>2</sup>.

En nuestro país, el uso de los podcasts se limita a ofrecer principalmente algunos contenidos de radio, televisión, medios impresos, música, ocio, tecnología y deportes. Hasta la fecha se han encontrado varios experimentos en la creación de podcasts, de los cuáles solo algunos de ellos se han mantenido vigentes. Se espera que el crecimiento del número de podcasters en México siga creciendo y estos ofrezcan contenidos de calidad.

El desarrollo de la aplicación Web, la primera en México y Latinoamérica, permitirá a los usuarios de Internet de México y el mundo publicar y difundir su información por medio de los podcasts.

### 2. Objetivo

Desarrollar una aplicación Web que permita a los usuarios de Internet publicar un podcast.

Los podcasts requieren de una plataforma de difusión para que los usuarios los conozcan, aprendan a utilizarlos y los aprovechen.

El diseño de la aplicación Web abarca el manejo de información en bases de datos, el diseño de un modelo lógico que por medio de métodos, acciones y plantillas permitan el buen funcionamiento de dicha aplicación.

---

<sup>1</sup> <http://www.apple.com/education/profiles/wells/>

<sup>2</sup> <http://edelgado.podomatic.com/>

Cabe resaltar que el desarrollo se basó en tres metas principales. En primera instancia el costo que generó la creación de la aplicación. En segunda instancia el plazo en el que se concluyó el proyecto. Finalmente los resultados de la aplicación completa.

Con el presente documento, en el que reporto mis actividades durante el desarrollo de la aplicación, demuestro mi capacidad de análisis, diseño, planeación, producción, resolución de problemas y manutención de un sistema electrónico que demuestre la formación de un profesional de la Ingeniería en el campo de la computación.

### 3. Introducción

Podcasting, de dónde se desprende la palabra podcast, es un medio de comunicación que consiste en la distribución de audio y video por Internet para ser escuchado en una computadora o, mediante un formato de suscripción gratuita, en un dispositivo portátil como el popular reproductor de mp3 llamado iPod o un celular con grandes capacidades de memoria.

Un podcast es la colección de archivos de audio y/o video que tienen un tema en particular como un programa de radio. Los usuarios de Internet pueden suscribirse a un podcast, de tal forma que cuando exista un nuevo archivo, o episodio del podcast, éste se descargue a la computadora o dispositivo portátil del usuario automáticamente.

El término podcast proviene de la palabra *iPod* y la palabra sajona *Broadcast* que en español significa emisión, aunque a ésta se le asocia regularmente con radio y televisión.

A la persona que graba un podcast se les conoce como *podcaster* y a su audiencia como *podlisteners* o, en español, *podescuchas*.

La distribución de archivos de audio y video se realiza mediante *RSS feeds* o alimentación mediante RSS con la que se puede realizar una suscripción a un podcast. El término RSS, por sus siglas en inglés, significa "RDF Site Summary", aunque también es conocido como "Really Simple Syndication" o "Sindicación Realmente Simple". El RSS es un sistema que difunde los contenidos, en éste caso los archivos de un podcast que se actualizan constantemente y los entrega a los usuarios de Internet al momento.

La comunicación es el proceso mediante el cuál dos entidades intercambian información con diferentes métodos, dónde el intercambio requiere retroalimentación. Podría sugerirse que un *podcaster*, al solo distribuir sus ideas en audio y video y el *podlistener* los descarga y escucha, no esta recibiendo retroalimentación. Esta idea es incorrecta, ya que como sucede con la televisión, la audiencia responde realizando acciones motivadas por lo que vio o escuchó, además una aplicación Web para un *podcaster* puede darle el espacio a su audiencia de comentar las ideas y puntos de vista sobre lo que ha escuchado.

Como medio de comunicación, los podcasts permiten la apertura de nuevas y mejores formas de compartir información.

#### **4. Definición de la problemática**

Tanto en México como en Latinoamérica no existía ninguna aplicación Web que permitiera a los crecientes usuarios interesados en ingresar al mundo de los podcast (tanto como *podescuchas* como *podcasters* potenciales) tener un lugar donde alojar sus podcasts y difundirlos. Por otro lado, en español apenas se encontraba un sitio que cumplía con este requisito, pero desafortunadamente cerró. Esto nos dejaba un panorama en Internet en el que no existía sitio Web alguno, en español, que diera un servicio de podcasting.

La creación de una aplicación Web para podcasters y podescuchas en español debía crearse con la idea principal de difundir las bondades que tienen los podcasts como medio de comunicación alternativo a los medios tradicionales controlados por minorías y no por la gente en general, que por el momento serán los usuarios de Internet.

## 5. Justificación

La facilidad para crear un podcast, en la que principalmente solo se necesita un micrófono y una computadora, lo cuál significa una pequeña inversión, ha incrementado la popularidad de distribución de los programas de podcasts.

Los podcasts son una tecnología que va ganando terreno día a día dentro de los usuarios a nivel mundial. La razón se debe a que las necesidades de comunicación entre las personas siguen en constante movimiento buscando mejores formas de hacerlo. Los podcasts permiten difundir a todo el mundo lo que para uno es importante y que puede ser importante para muchos más de manera fácil, eficiente y a bajo costo.

El hombre desde sus orígenes ha buscado y encontrado diferentes formas de comunicarse. Con el paso de los siglos, éstas se volvieron más eficientes y menos costosas. Lo que desde mi punto de vista ayudó a que todos los métodos de comunicación, que han sido concebidos y los que vendrán, tuvieran éxito fueron la confianza y la difusión que tuvieron por parte de organizaciones para acercarlas a las personas. Un nuevo sistema de comunicación merece exactamente el mismo respeto y confianza. Al final de cuentas, solo el tiempo le dará la razón.

La razón principal de crear la aplicación Web es para impulsar la difusión de los podcast por facilitar la distribución de audio y video. Es por eso que surge la necesidad de crear un sitio Web que permita a los usuarios publicar sus podcasts y, por el hecho de estar en Internet, hacerlos universalmente accesibles. El proyecto se convertiría finalmente en *poderato.com*.

La aplicación Web brindará a los usuarios una interfaz simple y fácil de usar para publicar sus podcasts. Éstos deberán ser entregados a los escuchas al momento en que fueron creados y permitir la retroalimentación. Además, al *podcaster* se le dará información importante como el número de descargas, de donde provienen sus oyentes y el impacto que está generando en la sociedad. El proyecto va más allá de ser un servicio dentro de un sitio Web. Pretende continuar con el crecimiento tecnológico en México proponiendo un servicio de calidad que pueda ser utilizado en cualquier parte del mundo. Hoy en día, nuestras aplicaciones de uso diario, cuando navegamos en Internet, son de un número diverso de nacionalidades. Nuestro desarrollo responde a la necesidad de tener una aplicación que pueda ser utilizada en la “canasta básica de servicios Web” y que sea completamente hecha en México.

### 5.1 Mercado por grupos

El mercado es global. Algunos de los nichos a los cuáles podemos enfocar la aplicación Web son:

- Educación y Gobierno
- Entretenimiento
- Computación e Internet
- Economía y Negocios
- Radio, Televisión y Revistas
- Artes y Humanidades
- Deportes y Recreación
- Ciencias
- Noticias, Sociedad y Cultura
- Cine, Música y Videojuegos

## 5.2 Ventaja competitiva

La primera ventaja que tiene el proyecto frente a otros sitios que ofrecen el mismo servicio es la Ideología. Mientras *los otros sitios* nacieron a consecuencia del *boom* de las nuevas tecnologías en Internet y el de los podcasts, la aplicación Web nace con el fin de brindar y mostrar el servicio a México, Latinoamérica y el mundo.

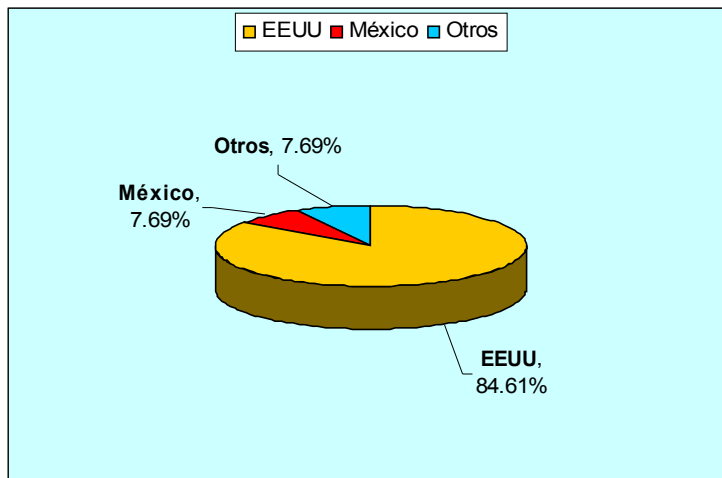
La segunda ventaja es la suscripción gratuita. De esta ponemos al alcance de todos, un servicio eficiente, mismo que no obtienen en otros sitios que no ofrecen este tipo de suscripciones.

Como tercera ventaja destaco las herramientas que la aplicación da a sus usuarios, tales como son las estadísticas, el sitio personal a la medida y el reproductor de podcasts.

En la tabla 1 se muestra la popularidad de los doce sitios en el mundo que dan el mismo servicio que el de la aplicación Web.

Sitio Web	Lugar de Origen	Ranking
<b>podomatic.com</b>	<b>San Francisco, CA</b>	<b>9,475</b>
<b>libsyn.com</b>	<b>Herndon, VA</b>	<b>11,575</b>
<b>odeo.com</b>	<b>Medford, New Jersey</b>	<b>14,659</b>
<b>castpost.com</b>	<b>Scottsdale, Arizona</b>	<b>43,169</b>
<b>podbean.com</b>	<b>Scottsdale, Arizona</b>	<b>48,494</b>
<b>mypodcast.com</b>	<b>Cooper City, FL</b>	<b>65,408</b>
<b>switchpod.com</b>	<b>Pittsburgh, PA</b>	<b>145,387</b>
<b>clickcaster.com</b>	<b>Boulder, CO</b>	<b>163,364</b>
<b>gcast.com</b>	<b>Scottsdale, Arizona</b>	<b>143,682</b>
<b>poderato.com</b>	<b>México City, MX</b>	<b>419,800</b>
<b>podblaze.com</b>	<b>Solana Beach, CA</b>	<b>589,469</b>
<b>podshowcreator.com</b>	<b>Farmingdale, NJ</b>	<b>945,074</b>
<b>www.podcastfm.co.uk</b>	<b>Leeds West Y, GB</b>	<b>1,385,435</b>

Tabla 1. Los diez sitios más populares que dan el servicio de podcasting<sup>1</sup>



<sup>1</sup> Datos proporcionados por Alexa.com. Fecha: 31 de Marzo 2008.



Figura 1. Los diez sitios más populares que dan el servicio de podcasting

Sitio Web	Dueño	Fecha de Creación
<b>Podomatic.com</b>	<b>Podomatic, Inc.</b>	<b>12-Jul-05</b>
<b>libsyn.com</b>	<b>Wizzard Media</b>	<b>07-Dic-04</b>
<b>odeo.com</b>	<b>SonicMountain</b>	<b>10-Jul-05</b>
<b>podbean.com</b>	<b>PodBean</b>	<b>28-Ago-06</b>
<b>castpost.com</b>	<b>Broad Holdings</b>	<b>01-Jun-05</b>
<b>mypodcast.com</b>	<b>First Beat Media</b>	<b>12-Ene-07</b>
<b>switchpod.com</b>	<b>Wizzard Media</b>	<b>15-Jun-05</b>
<b>clickcaster.com</b>	<b>ClickCaster, Inc</b>	<b>01-Jun-05</b>
<b>gcast.com</b>	<b>GarageBand Records</b>	<b>01-Abr-05</b>
<b>poderato.com</b>	<b>nGWiSE Comunicaciones</b>	<b>23-May-07</b>
<b>podblaze.com</b>	<b>The Info Gurus, LLC</b>	<b>21-Abr-05</b>
<b>podshowcreator.com</b>	<b>Pi Internet Bussiness Solutions</b>	<b>04-Feb-06</b>
<b>podcastfm.co.uk</b>	<b>EPOIA Interactive Studios, LLC</b>	<b>02-Abr-06</b>

Tabla 2. Los diez sitios más populares que dan el servicio de podcasting<sup>2</sup>

La tabla 1 nos muestra que, a diferencia de nuestro proyecto, no existe otro sitio desarrollado en México o Latinoamérica, además de que las interfaces de los sitios se presentan sólo en inglés.

A parte del análisis superficial que podemos obtener con el uso de los datos, debemos también advertir que toda la información se había estado guardando exclusivamente por servicios norteamericanos, lo cual limitaba al *podcaster* al momento de decidir en donde guardar su contenido.

<sup>2</sup> Datos proporcionados por Alexa.com. Fecha: 31 de Marzo 2008.

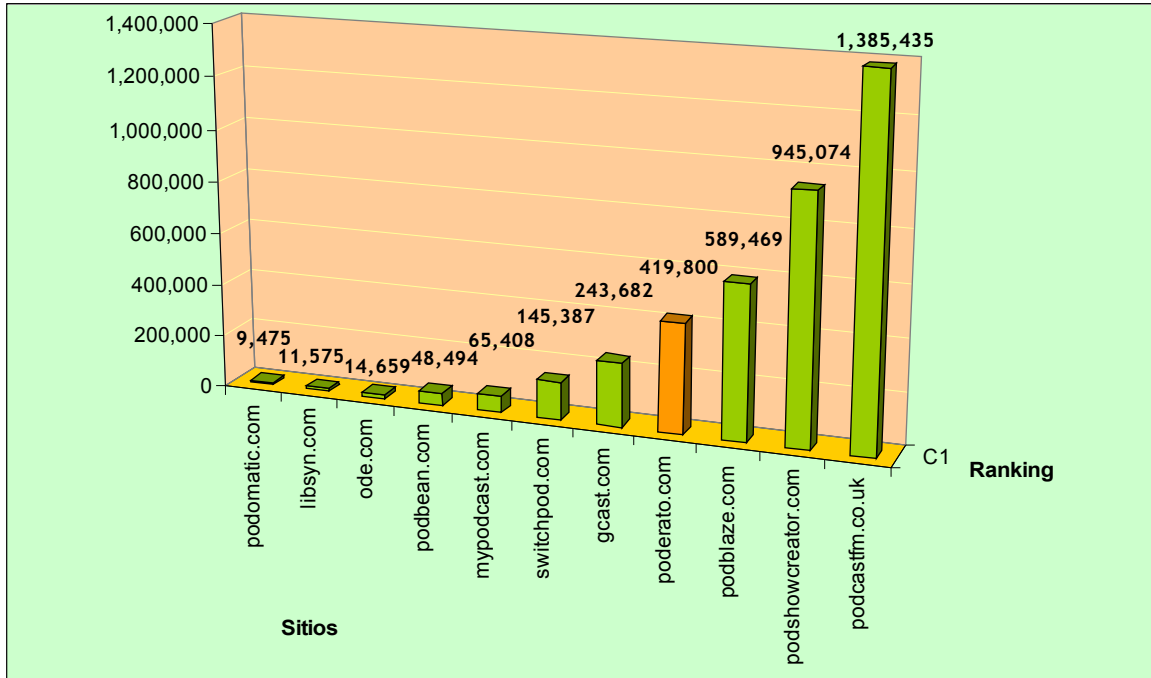


Figura 2. Comparación de nuestro proyecto con los diez sitios más populares que dan el servicio de podcasting<sup>3</sup>

### 5.3 Primeros en México y Latinoamérica

La aplicación Web desarrollada es el primer sitio Web en dar el servicio de *hosting* para podcasting en México y Latinoamérica, además de ser en este momento el único sitio en dar el servicio en español en el mundo.

A finales de Mayo de 2007, se puso en marcha la versión *beta* del proyecto con el nombre podErato y con dirección electrónica podErato.com.

### 5.4 Beneficios que traerá el desarrollo al usuario final (podcaster)

Las posibilidades que da la aplicación final a los usuarios finales son trascendentales. No olvidemos que los podcasts son un medio de comunicación en expansión y la aplicación Web es una plataforma para el uso de los mismos. A continuación enlisto los beneficios más representativos:

- Facilita un nuevo medio de comunicación que permitirá al usuario informar, entretener y enseñar a otros.
- Permite que la información llegue a cualquier punto del planeta en tiempo real.
- Beneficiará al usuario con la publicación de su podcast, de manera gratuita, y le mantendrá informado sobre la popularidad del mismo.

<sup>3</sup> Datos proporcionados por Alexa.com

- d) Fijará la generación de ingresos para usuarios con programas en demanda.

### **5.5 Beneficios que traerá el desarrollo al administrador**

El administrador de la aplicación Web obtendrá como beneficios:

- a) Manejo de los datos almacenados por el sistema.
- b) Acceso a las estadísticas del sistema.
- c) Administración de las cuentas de los usuarios.
- d) Actualizaciones de módulos.

## **6. Metodología del proyecto**

Para que el desarrollo de la aplicación resultara exitoso, adopté un conjunto de procedimientos y técnicas, además de poner énfasis en la documentación.

Las técnicas nos indican cómo debe ser realizada una actividad determinada identificada en la metodología. Se combina el empleo de modelos o representaciones gráficas junto con el empleo de procedimientos detallados.

Como dato fundamental, se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología del desarrollo y que no todas las técnicas aplican en todos los casos. Además se debe tener mucho cuidado cuando se quiere cambiar una técnica por otra.

### **6.1 Metodología y diseño Orientado a objetos**

La creciente tendencia de crear programas cada vez más grandes y complejos llevó a los desarrolladores a crear una nueva forma de programar que les permita crear sistemas de niveles empresariales y con reglas de negocios muy complejas. Para estas necesidades ya no basta la programación estructurada ni mucho menos la programación lineal. Es así como aparece la programación orientada a objetos (POO).

La POO viene de la evolución de la programación estructurada; básicamente la POO simplifica la programación con la nueva filosofía y nuevos conceptos que tiene. La POO se basa en dividir el programa en pequeñas unidades lógicas de código. A estas pequeñas unidades se les llama objetos. Los objetos son unidades independientes que se comunican entre ellos mediante mensajes.

Para el desarrollo de este proyecto se eligió el paradigma orientado a objetos. Para el desarrollo de software orientado a objetos no basta usar un lenguaje orientado a objetos. También se necesitará realizar un análisis y diseño orientado a objetos. El modelo visual es la clave para realizar el análisis Orientado a Objetos.

Desde los inicios del desarrollo de software orientado a objetos han existido diferentes metodologías para hacer dicho modelado, pero sin lugar a duda, el Lenguaje de Modelado Unificado (UML) puso fin a la guerra de metodologías. UML consta de todos los elementos y diagramas que permiten modelar los sistemas basándose en el paradigma orientado a objetos. Es por las anteriores razones que el desarrollo de este sistema utiliza las mejores prácticas de UML.

En el proceso de modelado orientado a objetos podemos distinguir las tres fases tradicionales de elaboración de un sistema informático:

- a) Análisis: Se ocupa de qué, de entender el dominio del problema.
- b) Diseño: responde al cómo, y se centra en el espacio de la solución.
- c) Implementación: fase en la que se adapta la solución a un entorno de programación específico.

## 6.2 Consideraciones para adoptar un estándar o construir una metodología

En el momento de adoptar un estándar o construir una metodología, se han de considerar unos requisitos deseables, por lo que se proponen una serie de criterios de evaluación de dichos requisitos.

En la tabla 3 se presenta el listado de criterios.

### Consideraciones para la elección de una metodología de desarrollo

- La metodología debe ajustarse a los objetivos
- La metodología debe cubrir el ciclo entero de desarrollo de software
- La metodología debe integrar las distintas fases del ciclo de desarrollo
  - Es importante poder referirse a otras fases de un proyecto y fusionarlo con las frases previas. Es importante poder moverse no solo hacia delante en el ciclo de la vida, sino hacia atrás de forma que se pueda comprobar el trabajo realizado y se puedan efectuar correcciones.
- Fácil interacción entre etapas del ciclo de desarrollo
  - Es necesaria una validación formal de cada fase antes de pasar a la siguiente. La información que se pierde en una fase determinada queda perdida para siempre, con un impacto en el sistema resultante.
- La metodología debe detectar y corregir los errores cuanto antes.
  - Uno de los problemas mas frecuentes y costosos es el aplazamiento de la detección y corrección de los problemas en las etapas finales del proyecto. Cuanto más tarde sea detectado el error, más caro será corregirlo. Por lo tanto cada fase del proceso de desarrollo de software deberá de incluir una actividad de validación explícita.
- La metodología debe soportar la determinación de la exactitud del sistema a través del ciclo de desarrollo.
  - La exactitud del sistema implica muchos asuntos, incluyendo la correspondencia entre el sistema y sus especificaciones, así como que el sistema cumpla con las necesidades del usuario. Por ejemplo, los métodos usados para análisis y especificación del sistema deberían colaborar a terminar con el problema del entendimiento entre los informáticos, los usuarios. Y otras posibles partes implicadas.
- La metodología debe ser la base de una comunicación efectiva.
  - Ha de haber una comunicación efectiva entre analistas, programadores, usuarios y gestores, con pasos bien definidos para realizar progresos visibles durante la actividad del desarrollo.

## Desarrollo de una aplicación Web para la publicación de podcasts en línea

- La metodología debe funcionar en un entorno dinámico orientado al usuario
  - A lo largo de todo el ciclo de vida del desarrollo se debe producir una transferencia de conocimientos hacia el usuario. La clave del éxito es que todas las partes implicadas han de intercambiar información libremente. La participación del usuario es de importancia vital debido a que sus necesidades evolucionan constantemente. Por otra parte la adquisición de conocimientos del usuario permitirá la correcta toma de decisiones. Es esencial contar con una buena técnica de diagramación.
- La metodología debe especificar claramente los responsables de resultados
  - Debe especificar claramente quienes son los participantes de cada tarea a desarrollar, detallando de manera clara los resultados de los que serán responsables
- Una empresa debe adoptar una metodología que sea útil para un gran número de sistemas que vaya a construir. Por esta razón no es práctico adoptar varias metodologías en una misma empresa
- Las metodologías deberán ser capaces de abordar sistemas de distintos tamaños y rangos de vida
- La metodología debe servir para sistemas de distinta complejidad
- La metodología se debe poder enseñar.
  - Incluso en una organización sencilla, serán muchas las personas que la van a utilizar, incluso los que se incorporen posteriormente a la empresa. Cada persona debe entender las técnicas específicas de la metodología, los procedimientos organizacionales y de gestión que la hacen efectiva, las herramientas automatizadas que soportan la metodología y las motivaciones que subyacen en ella.
- La metodología debe estar soportada por herramientas CASE
  - La metodología debe estar soportada por herramientas automatizadas que mejoren la productividad, tanto del ingeniero de software en particular, como la del desarrollo en general.
- La metodología debe soportar la eventual evolución del sistema
  - Normalmente durante su tiempo de vida los sistemas tienen muchas versiones. Existen herramientas CASE para la gestión de la configuración y otras denominadas "Ingeniería inversa" para ayudar en el mantenimiento de los sistemas no estructurados, permitiendo estructurar los componentes de estos facilitando así su mantenimiento.
- La metodología debe contener actividades contundentes a mejorar el proceso de desarrollo de software.
  - Para mejorar el proceso es básico disponer de datos numéricos que evidencian la efectividad de la aplicación del proceso con respecto a cualquier producto software resultante del proceso. Para disponer de estos datos, la metodología debe contener un conjunto de mediciones de proceso para identificar la calidad y costo asociado a cada etapa del proceso. Sería ideal el uso de herramientas CASE.

Tabla 3 Criterios para adoptar un estándar

### 6.3 Importancia de las herramientas CASE

Con la aparición de las herramientas CASE junto con los generadores de código, el ciclo de desarrollo de software en cascada ha cambiado a un ciclo de vida basado en transformaciones.

En el desarrollo de la aplicación Web se hizo uso de Symfony que es un *framework* de aplicación Web para proyectos programados en el lenguaje PHP en su versión 5.

Su objetivo es acelerar la creación y mantenimiento de aplicaciones Web, y reemplazar las tareas de codificación repetitiva por un mejor organización que permita un mejor manejo y control de la aplicación a desarrollar.

#### Ventajas

Symfony se utiliza para la creación de aplicaciones robustas en un contexto empresarial. Lo cuál significa que se tiene un control completo sobre la configuración: desde la estructura de directorios hasta las librerías externas, en dónde prácticamente todo puede ser adecuado a las necesidades del programador. Para alcanzar las directrices del desarrollo de una empresa, symfony está equipado con herramientas adicionales que ayudan en los procesos de **prueba, depuración y documentación** del proyecto.

### 6.4 Ingeniería de Software

A mediados de la carrera, en la Facultad de Ingeniería de la UNAM, estudié la Ingeniería de Software (IS), o Ingeniería de Programación. Sin duda, una de las asignaturas más interesantes, pues nos enseñó que a un proyecto de software se le debe:

- a) Analizar
- b) Diseñar
- c) Hacer una planeación
- d) Desarrollar
- e) Documentar el código fuente
- f) Realizar manuales para el usuario y para el administrador
- g) Proyectar tiempo de vida del software y futuras actualizaciones

Aplicar las técnicas que la IS ofrece, ayudó a que el proyecto no adoleciera de:

- a) Retrasos considerables en la planificación
- b) Poca productividad
- c) Elevadas cargas de mantenimiento
- d) Demandas no acordes con las ofertas
- e) Baja calidad y fiabilidad del producto
- f) Dependencia de los realizadores
- g) Pérdida de tiempo
- h) Incapacidad de ofrecer el servicio u objetivo final satisfactoriamente

En resumen, la IS comprende un conjunto de técnicas orientadas al desarrollo de un programa o software. Las cuáles apliqué para el desarrollo de la aplicación Web.

## 7. Análisis

Analizar es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema. En la figura 3 se ejemplifica el ciclo que le da vida a la creación de un sistema informático.

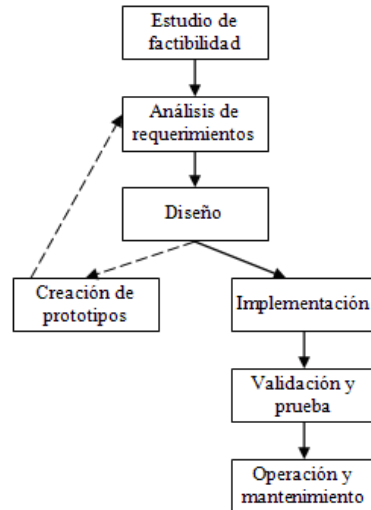


Figura 3. Ciclo de vida típico de los sistemas informáticos

Por lo general se piensa que el análisis es el primer paso en la elaboración de una aplicación. Sin embargo, antes de éste se debe de realizar un estudio de factibilidades, en dónde el más importante resultó ser el financiero. Este estudio es analizado a través de otros estudios que a continuación se enlistan:

- a) Estudio de mercado
- b) Estudio técnico
- c) Estudio de las finanzas

En el estudio de mercado se encontró una gran oportunidad de crecimiento y posicionamiento al no existir servicio alguno en México y Latinoamérica.

Sobre el estudio técnico, se concluyó que se cubrían todas las necesidades referentes a equipo, sistema operativo, servidor y software relacionado.

Los siguientes elementos no requirieron inversión al utilizar versiones libres:

- a) **Manejador de bases de datos**  
Nombre: PhpPgAdmin  
Versión: 4.1.1
- b) **Framework PHP**  
Nombre: Symfony  
Versión: 1.0.11
- c) **Base de datos**  
Nombre: PostgreSQL  
Versión: 7.4.17



**d) Lenguaje de Programación**

Nombre: PHP

Versión: 5.2.2

**e) Servidor**

Nombre: Apache

Versión: 2.0

**f) Base de datos de países y sus estados**

**g) Base de datos de rango de IP (Protocolos de Internet) para obtener la ubicación del usuario**

En cuanto a las finanzas, el análisis arrojó que en el desarrollo inicial sólo se invertiría en el sueldo de dos integrantes y la parte proporcional de hosting de un servidor dedicado.

Teniendo resultados positivos en todas las áreas, pasé al análisis de requerimientos.

### **7.1 Requerimientos**

Los requerimientos de la aplicación es el conjunto de características, acciones y servicios que el sistema deberá ofrecer. Los requerimientos siempre deben apuntarse en papel. Resulta más fácil organizarlos cuando los das tu mismo.

#### **7.1.1 Requerimientos en la administración**

A continuación los requerimientos para la administración

- a) Debe ser intuitiva.
- b) Debe contener un reporte de los podcasts y episodios problemáticos.
- c) Debe mostrar un listado de usuarios problemáticos (aquellos usuarios que han sido reportados por la comunidad ya sea por tener un podcast inadecuado o por escribir comentarios ofensivos).
- d) Podrá revisar peticiones de nuevos moderadores.
- e) Podrá agregar nuevos administradores.
- f) Debe mostrar un listado de *tags* no populares.

#### **7.2 Requerimientos sobre el servicio para el usuario final**

A continuación los requerimientos para el usuario final

- a) Deberá contener una administración intuitiva.
- b) Presentar sus episodios en una página personal eficiente.
- c) Deberá presentar una sección de asistencia técnica.
- d) Deberá avisar cada vez que una sección ha sido actualizada de manera exitosa, o de lo contrario mostrar al usuario el error en turno.
- e) Deberá enviar correos electrónicos al usuario al registrarse, cuando alguien le realice un comentario y para darle avisos por parte del administrador.
- f) Deberá presentar estadísticas del podcast en tiempo real.
- g) Permitirá al usuario decorar su sitio a su gusto.

### **7.3 Requerimientos para la interfaz**

A continuación los requerimientos para la interfaz

- a) Deberá ser intuitiva.
- b) Presentará listados con los podcasts mas descargados, más votados y más comentados.
- c) Facilitará las acciones de valoración a cualquier episodio.
- d) Permitirá la descarga del podcast, tanto a una computadora como a un dispositivo portátil.
- e) Concederá la publicación de comentarios.

## **8. Planeación**

La planeación del proyecto abarca el alcance que se le dará, la elección de los integrantes del grupo de trabajo, la estimación de tiempos y costos, los procedimientos de cambios, y los criterios de aceptación.

El alcance de la aplicación Web desarrollada comprende desde el análisis y el diseño del sitio hasta su construcción, limitada por lo establecido en los requerimientos.

La aplicación Web fijará las bases de una plataforma ideal para la difusión y aprovechamiento de los podcasts.

Ya se cuenta con las proyecciones para que en el transcurso del año 2008, se implementen nuevos módulos que den al usuario una mejor experiencia con el sitio.

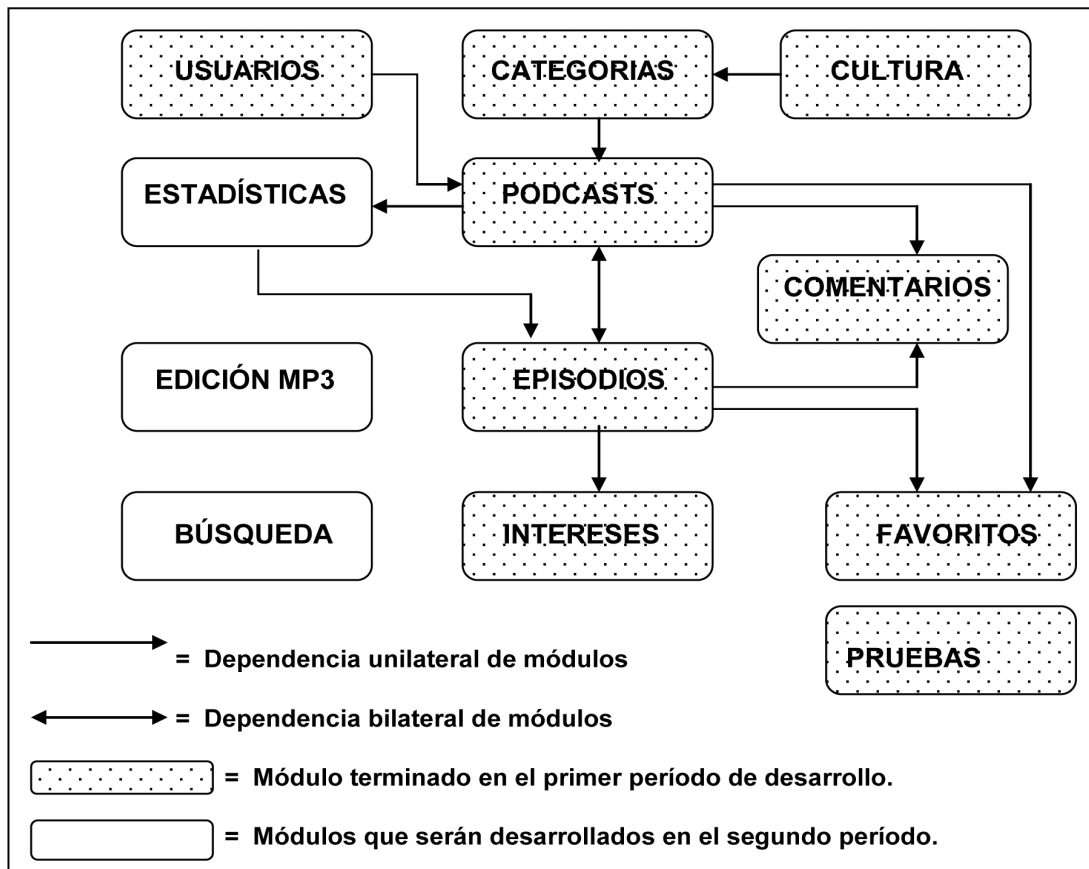


Figura 4. Relación de módulos desarrollados en un primer y segundo período

En la figura 4 se muestran tanto los módulos que se decidió, por su importancia, desarrollar en el primer período de desarrollo del proyecto (Enero 2007 – Mayo 2007), así como los módulos previstos a ser desarrollados en el segundo período de desarrollo (Junio 2007 – Noviembre 2007).

Los módulos diseñados para la aplicación Web son:

- Usuarios.** Comprende el registro y comportamiento de los usuarios (podcasters, listener, administrador).
- Categorías.** Define las categorías a los que pertenecerán los podcasts.
- Cultura.** Permite la presentación de la interfaz de usuario y administración en diferentes idiomas.
- Podcast.** Módulo que registra y administra los podcasts de los usuarios.
- Episodios.** Módulo que registra y administra los episodios de los usuarios.
- Comentarios.** Facilita el ingreso de comentarios a podcasts y episodios.
- Favoritos.** Permite integrar una red social entre los podcasters y listeners.
- Intereses.** Define la popularidad de los episodios de un podcast.
- Estadísticas.** Registra, organiza y presenta las estadísticas de los podcast.
- Edición de MP3.** Facilita la manipulación de *tags* en los archivos mp3.
- Búsqueda.** Habilita la función de búsqueda de podcasts y episodios.

### **8.1 Creación del grupo de trabajo**

En todo trabajo, debemos rodearnos de la gente idónea. Para el desarrollo de la aplicación, iniciamos tres integrantes. El primero revisó el avance y, por su experiencia, apoyó en los módulos de mayor complejidad. El segundo integrante y yo primero estudiamos a detalle el *framework* que a la postre sería la base del proyecto para después completar el ciclo de desarrollo.

Nuestra forma de administrar los recursos y repartir responsabilidades se manejó de la siguiente manera:

- a) Priorizar los módulos principales.
- b) Desarrollar módulos que no tuvieran relación directa alguna.
- c) Aprovechar la arquitectura MVC para separar el trabajo en el modelo de datos y la interfaz.

Esta forma de trabajar contribuyó con el desempeño en el desarrollo de la aplicación.

### **8.2 Estimación de Tiempos y Costos**

La estimación de tiempo para la aplicación Web supone el establecimiento de un calendario. La estimación de costos se enfoca en un presupuesto recomendado para completar exitosamente el ciclo de vida del proyecto. Ambas estimaciones resultaron a partir del análisis de requerimientos.

Elaboré el calendario de tiempos que fue analizado por todos los integrantes del proyecto y se realizaron adecuaciones. Cuando se tuvo la versión final del documento, se comenzó a trabajar.

A continuación, en la figura 5, muestro el calendario de actividades para la primera versión del sistema:

## Desarrollo de una aplicación Web para la publicación de podcasts en línea



























	<b>- Definir diseño Web</b>	<b>38 días ?</b>	<b>lun 08/01/07</b>	<b>mié 28/02/07</b>
	Interfaz Gráfica	38 días?	lun 08/01/07	mié 28/02/07
	Hojas de Estilo (CSS)	38 días?	lun 08/01/07	mié 28/02/07
	Usabilidad	38 días?	lun 08/01/07	mié 28/02/07
	<b>Definir Slogans</b>	5 días	lun 02/07/07	vie 06/07/07
	<b>- Crear Sección Informativa</b>	<b>10 días ?</b>	<b>lun 05/03/07</b>	<b>vie 16/03/07</b>
	¿Qué es Podcast?	10 días?	lun 05/03/07	vie 16/03/07
	¿Cómo lo Descargo?	10 días?	lun 05/03/07	vie 16/03/07
	¿Cómo creo el mío?	10 días?	lun 05/03/07	vie 16/03/07
	<b>Crear Sección Preguntas y Respuestas</b>	10 días	lun 19/03/07	vie 30/03/07
	<b>- Estrategia Publicitaria</b>	<b>10 días ?</b>	<b>lun 02/04/07</b>	<b>vie 13/04/07</b>
	Estadísticas de visitas/descargas	10 días?	lun 02/04/07	vie 13/04/07
	Buscar clientes posibles	10 días?	lun 02/04/07	vie 13/04/07
	Publicidad en nuestro portal	10 días?	lun 02/04/07	vie 13/04/07
	<b>- Privacidad</b>	<b>10 días ?</b>	<b>lun 16/04/07</b>	<b>vie 27/04/07</b>
	Crear política de privacidad	10 días?	lun 16/04/07	vie 27/04/07
	Definir marco jurídico y su alcance	10 días?	lun 16/04/07	vie 27/04/07
	<b>- Infraestructura</b>	<b>15 días ?</b>	<b>lun 16/04/07</b>	<b>vie 04/05/07</b>
	Analizar espacio en servidor a contratar	15 días?	lun 16/04/07	vie 04/05/07
	Analizar ancho de banda a contratar	15 días?	lun 16/04/07	vie 04/05/07
	Analizar espacio a proporcionar a clientes	15 días?	lun 16/04/07	vie 04/05/07
	Analizar ancho de banda a proporcionar a	15 días?	lun 16/04/07	vie 04/05/07
	<b>- Cobro por espacio extra</b>	<b>15 días ?</b>	<b>lun 07/05/07</b>	<b>vie 25/05/07</b>
	Definir diferentes tipos de cuentas	15 días?	lun 07/05/07	vie 25/05/07
	Definir métodos de pago	15 días?	lun 07/05/07	vie 25/05/07
	<b>Definir historias del administrador</b>	5 días	lun 28/05/07	vie 01/06/07
	<b>- Actividades sobre el Servicio</b>	<b>95 días ?</b>	<b>lun 02/04/07</b>	<b>vie 10/08/07</b>
	Definir Categorías	10 días?	lun 02/04/07	vie 13/04/07
	Definir creación de Cuenta	5 días?	lun 16/04/07	vie 20/04/07
	<b>- Modulo Nuevo Podcast</b>	<b>20 días ?</b>	<b>lun 23/04/07</b>	<b>vie 18/05/07</b>
	Editar Podcast	15 días?	lun 23/04/07	vie 11/05/07
	Cancelar Cuenta	6 días?	vie 11/05/07	vie 18/05/07
	Desplegar Podcast	16 días?	vie 18/05/07	vie 08/06/07
	<b>- Opina Sobre el Podcast</b>	<b>6 días ?</b>	<b>vie 08/06/07</b>	<b>vie 15/06/07</b>
	Alquien opino sobre tu Podcast	6 días?	vie 08/06/07	vie 15/06/07
	Agrega Tag	5 días?	lun 18/06/07	vie 22/06/07
	Recomienda mi Podcast	5 días?	lun 25/06/07	vie 29/06/07
	Recomienda Poderato	5 días?	lun 02/07/07	vie 06/07/07
	Buscador	10 días?	lun 09/07/07	vie 20/07/07
	Modulo de Seguridad al crear cuentas	7 días?	lun 23/07/07	mar 31/07/07
	Modulo de Seguridad al subir podcast	3 días?	mié 01/08/07	vie 03/08/07
	Radio	5 días?	lun 06/08/07	vie 10/08/07

Figura 5. Diagrama de actividades de desarrollo de la aplicación Web

### **Definir procedimientos de cambio**

En todo proyecto se deben considerar modificar o cambiar completamente uno o más módulos del desarrollo. Los cambios suelen ser por que algo no está funcionando en el sistema como debería, o por que existen nuevas tecnologías que logran hacer acciones más eficientes.

Cuando se desea actualizar algún *plug-in*, biblioteca o software instalado en el servidor, se debe investigar si las nuevas versiones son cien por ciento compatibles tanto con el servidor, como con el tipo de tecnología que estamos utilizando para programar, así como con nuestra base de datos y otros programas actualmente instalados.

Al momento de escribir este reporte, no se han realizado cambios.

### **8.3 Acordar criterios de aceptación**

Como parte de la estructuración y el buen desarrollo del sistema se fijaron criterios que rigen el desarrollo de la aplicación Web.

Entre los criterios encontramos:

#### **a) Consultas a la BD**

El número de consultas a la base de datos deben ser reducidas a su máxima expresión. Es un hecho que cada parte del sitio requiere un número diferente de consultas a la base de datos, sin embargo debe tomarse en cuenta que el número de consultas en promedio, por página, no debe sobrepasar las 5 consultas.

#### **b) Tiempo en cargar**

Por lo general, un sitio puede requerir un tiempo prolongado si abusa de *scripts* o código innecesario. Para nuestra aplicación Web, el tiempo de carga debe ser mínimo y comparable con cualquier página de la aplicación.

## 9 Diseño

El diseño en una aplicación Web es como el bosquejo en un dibujo. En el que, sin detallar, se determina la funcionalidad general de nuestro sistema.

Me apoyé en los **Casos de Uso** para el desarrollo de módulos y en el **Diagrama Entidad Relación** para diseñar la base de datos.



## 9.1 Diagrama Entidad Relación

El diagrama entidad relación (DER) que diseñe me ayudó a modelar los datos de mi aplicación Web. A partir del modelo, obtuve las entidades y sus atributos, así como su interrelación. A continuación en la figura 6 muestro el DER de la aplicación Web.

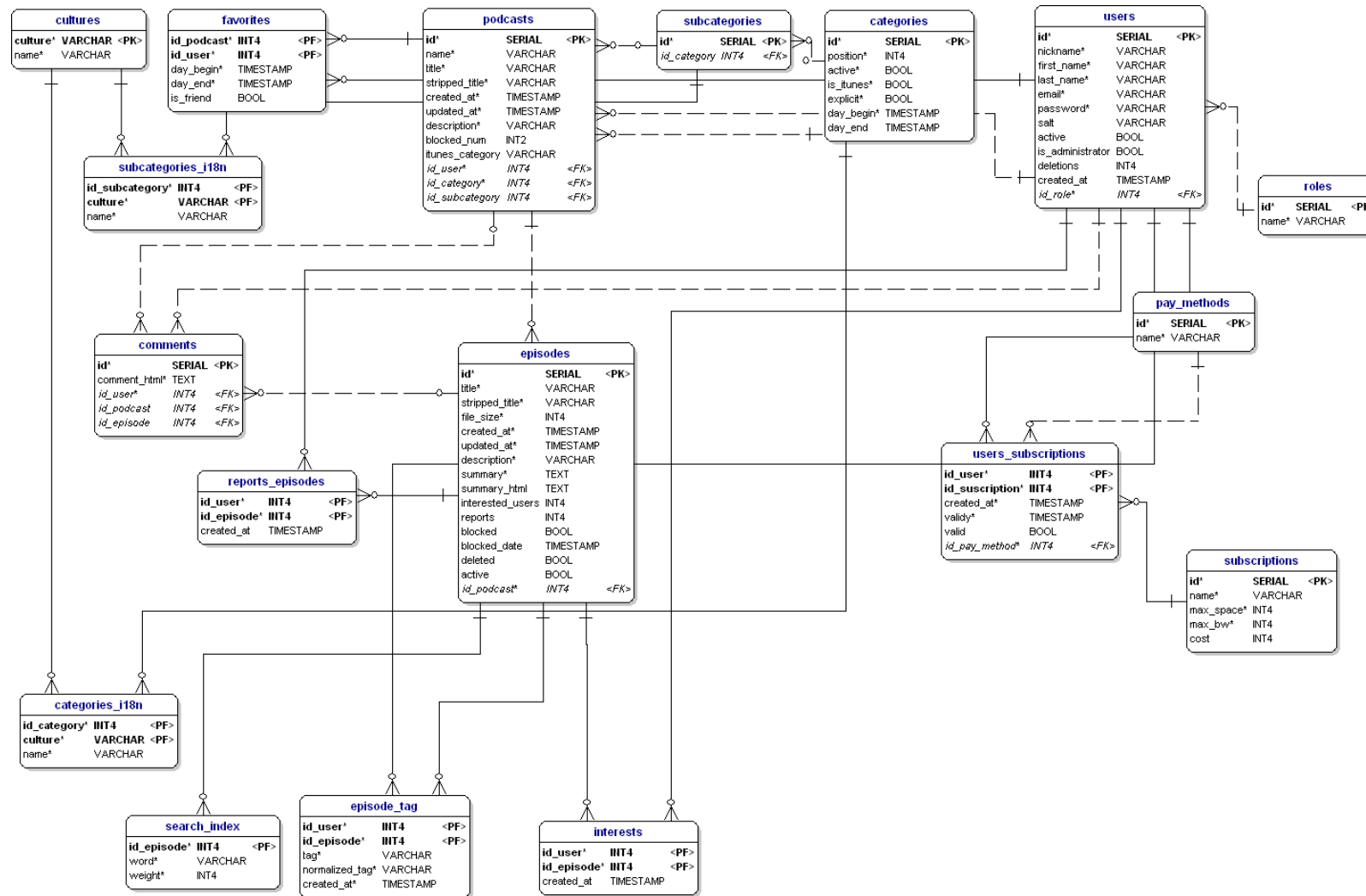


Figura 6 Diagrama Entidad Relación de la aplicación Web

## 9.2 Modelo de Casos de Uso

Al analizar se hace distinción y separación de cada parte de los requerimientos, con la meta de conocer a la perfección lo que se debe desarrollar, así como la posibilidad de recomendar mejoras al sistema. En consecuencia realicé un conjunto de casos de uso de lo que se necesitaba hacer.

Con el modelado de casos de uso rescaté las acciones principales, ya sea por su relevancia o por su complejidad, que la aplicación Web servirá al usuario.

A continuación mostrare algunos de los casos de uso del Modelo de Uso que redacté para la aplicación Web. Para los casos de usos identifiqué tres actores distintos que interactúan en la aplicación Web: el usuario, el sistema (aplicación Web) y el administrador.

Para ejemplificar, en las siguientes tablas muestro 3 casos de uso que considero indispensables en la aplicación Web.

### 9.2.1 Especificaciones de Alto Nivel

#### Caso 1:

**Descripción:** *El módulo de subida de archivos es quizás el más importante, pues permitirá a los podcasters publicar sus archivos de audio a Internet.*

<b>Actores</b>	Usuario
<b>Descripción</b>	El usuario sube un archivo de audio para su publicación.
<b>Condiciones</b>	El usuario debe estar autenticado por el sistema.

Tabla 4 Caso de Uso: Subir un archivo

#### Caso 2:

**Descripción:** *El módulo de estadísticas provee al podcaster de información relevante sobre el accionar de su podcast. Registra, organiza y presenta los datos que obtiene.*

<b>Actores</b>	Sistema (aplicación Web)
<b>Descripción</b>	El sistema recibe la petición, por parte del usuario, para mostrar las estadísticas.
<b>Condiciones</b>	El usuario debe estar autenticado por el sistema.

Tabla 5 Caso de Uso: Presentación de Estadísticas

**Caso 3:**

**Descripción:** El reporte de archivos permite a la comunidad del sitio reportar archivos ilegales o de mal gusto. El administrador podrá evaluar el archivo y decidir eliminarlo o indultarlo.

<b>Actores</b>	Administrador
<b>Descripción</b>	El Administrador es notificado sobre un reporte asignado a un episodio de podcast.
<b>Condiciones</b>	El administrador debe estar autenticado por el sistema.

Tabla 6 Caso de Uso: Reporte de archivos

**Especificaciones extendidas****Caso 1:**

**Descripción:** El módulo de subida de archivos es quizás el más importante, pues permitirá a los podcasters publicar sus archivos de audio a Internet.

<b>Actores</b>	Usuario (iniciador)
<b>Descripción</b>	El usuario sube un archivo de audio para su publicación.
<b>Curso típico de los eventos</b>	
<b>Acción del actor</b> 1. El usuario selecciona el archivo y lo sube.	<b>Respuesta del Sistema</b> 2. El sistema recibe la petición y sube el archivo mientras muestra el avance. 3. Avisa que el archivo ha sido publicado.
<b>Acciones alternativas</b>	
Paso 2. El archivo no es de un formato aceptado por el servidor y lo rechaza.	
Paso 3. Ocurre un error en la subida y el servidor manda un mensaje de alerta.	

Tabla 7 Caso de Uso: Subir un archivo

**Caso 2:**

**Descripción:** El módulo de estadísticas provee al podcaster de información relevante sobre el accionar de su podcast. Registra, organiza y presenta los datos que obtiene.

<b>Actores</b>	Sistema (aplicación Web)
<b>Descripción</b>	El sistema recibe la petición, por parte del usuario, para mostrar las estadísticas.
<b>Curso típico de los eventos</b>	
<b>Acción del actor</b> 1. El usuario selecciona la liga de estadísticas por podcast.	<b>Respuesta del Sistema</b> 2. El sistema obtiene la petición y ejecuta el método getPodcastStatistics(). 3. Presenta la información en orden decreciente.
<b>Acciones alternativas</b>	
Paso 2. El sistema no encuentra estadísticas y le avisa al usuario.	

Tabla 8 Caso de Uso: Presentación de Estadísticas

**Caso 3:**

**Descripción:** El reporte de archivos permite a la comunidad del sitio reportar archivos ilegales o de mal gusto. El administrador podrá evaluar el archivo y tomar la decisión.

<b>Actores</b>	Administrador
<b>Descripción</b>	El Administrador es notificado sobre un reporte asignado a un episodio de podcast.
<b>Curso típico de los eventos</b>	
<b>Acción del actor</b> 1. El administrador recibe en su sesión la notificación.  2. Lee la notificación y elimina el archivo correspondiente al episodio.	<b>Respuesta del Sistema</b>  3. El sistema borra todo el historial generado por el archivo.
<b>Acciones alternativas</b>	
2. El administrador no encuentra ningún problema con el episodio y quita el reporte.	

Tabla 9 Caso de Uso: Reporte de archivos

## **10. Desarrollo**

El desarrollo inicia cuando el análisis y el diseño han sido concluidos y entendidos por todos los miembros del equipo.

### **10.1 Metodología empleada para el desarrollo**

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con directrices específicas, tales como: estructura modular, fuerte cohesión, alta rentabilidad, etc. Un paradigma de programación es una colección de modelos conceptuales que juntos modelan el proceso de diseño y determina, al final, la estructura de un programa.

Esa estructura conceptual de modelos está pensada de forma que estos determinen la forma correcta de los programas y controlan el modo en que pensamos y formulamos soluciones, y al llegar a la solución, ésta se debe de expresar mediante un lenguaje de programación. Para que este proceso sea efectivo las características del lenguaje deben reflejar adecuadamente los modelos conceptuales de ese paradigma.

### **10.2 Ciclo de desarrollo del sistema**

El ciclo de desarrollo del sistema es el ciclo de vida del sistema, por lo que inicia cuando se toma la decisión de desarrollar el proyecto hasta que se entrega la versión final. El ciclo comprende lo que se ha venido escribiendo en este reporte: análisis de los requerimientos, diseño y desarrollo. Sin embargo también existen elementos importantes como el grupo de trabajo, estimación de tiempo y costos, definición de los criterios de aceptación y documentación. Todos estos elementos los elaboré y a continuación presento.

#### **10.2.1 Framework**

Como se describió en la importancia de las herramientas CASE, se decidió utilizar Symfony como framework de desarrollo. La ventaja de utilizar un framework es que a partir del diseño de la base de datos, se generan automáticamente módulos, clases y métodos, lo que representa un avance importante al inicio del desarrollo.

#### **10.2.2 Requerimientos para la instalación**

La pequeña cantidad de prerrequisitos hacen que symfony sea fácil de instalar sobre cualquier configuración.

Se necesita:

- ❖ Sistema Operativo Unix o Windows
- ❖ Servidor Web
- ❖ PHP5
- ❖ Sistema de Base de Datos

Además de que tiene una muy pequeña carga de recursos, por lo que los beneficios del framework no afectan el incremento por costos de alojamiento de servidor.

### 10.2.3 Creación del proyecto

Descargué e instalé el framework en el servidor.

```
$ pear upgrade PEAR
```

```
$ pear channel-discover pear.symfony-project.com
```

```
$ pear install symfony/symfony
```

Instalé el proyecto en nuestro directorio de trabajo.

```
$ mkdir /home/sfprojects/askeet  
$ cd /home/sfprojects/askeet  
$ symfony init-project askeet
```

Inicialicé la aplicación.

```
$ symfony init-app frontend
```

Después, modifique el archivo `httpd.conf` para que las llamadas al servidor fueran recibidas por el framework como se ve a continuación.

```
<VirtualHost *:80>  
  ServerName poderato  
  DocumentRoot "/home/sfprojects/poderato/web"  
  DirectoryIndex index.php  
  Alias /sf /usr/local/lib/php/data/poderato/web/sf  
  
  <Directory "/home/sfprojects/poderato/web">  
    AllowOverride All  
  </Directory>  
</VirtualHost>
```

### 10.2.4 Creación del modelo

Con el diagrama entidad relación de nuestra aplicación, cree la base de datos. Además, de manera automática generé el modelo de datos.

Primero generé el *schema* de la aplicación y después generé el modelo.

```
$ symfony propel-build-schema xml
```

```
$ symfony propel-build-model
```

Después de generado el modelo, nuestro árbol de archivos generó:

```
modules/  
  administrator  
  comment  
  feed  
  episodes  
  podcasts  
  sidebar  
  tag  
  user
```

### 10.2.5 Creación de un módulo

La creación de un módulo permite desarrollar las clases, métodos y acciones para lograr los objetivos definidos en los alcances. La creación de un módulo lo generé de la siguiente manera:

```
$ symfony propel-generate-crud frontend podcasts Podcasts
```

### 10.2.6 Arquitectura MVC

El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

#### a) Modelo

Mi trabajo consistió en generar el modelo que se encuentra en la librería. La librería de Modulo en el proyecto se encuentra de la siguiente forma:

```
lib/  
  model/  
    Episodes  
    EpisodesPeer  
    Podcasts  
    PodcastsPeer  
    Categories  
    CategoriesPeer  
    Interests  
    InterestsPeer  
    Favorites  
    FavoritesPeer
```

## **b) Vista**

Para la vista, definí templates que presentarían la información. Los archivos que se crearon para la vista fueron:

```
templates/  
  layout.php  
  podhome.php  
  main.php
```

## **c) Controlador**

En el controlador definí todas las acciones referentes a los módulos diseñados. Las acciones en común fueron:

```
actions/  
  EditSuccess.action.class  
  FormSuccess.action.class  
  CreateSuccess.action.class  
  DeleteSuccess.action.class
```



## 11. Pruebas

En la medida que fue creciendo el desarrollo de la aplicación, realicé pruebas que nos proporcionaron información relevante sobre nuestro avance. Encontramos los métodos que no estaban dando los resultados esperados o las clases que debían ser depuradas en busca de hacer que la respuesta de la aplicación fuera rápida y eficiente.

Las pruebas unitarias son uno de los grandes avances en la programación desde la orientación a objetos. Permiten un proceso de desarrollo seguro y complementan la documentación del desarrollo ya que ilustran lo que una aplicación debe hacer. En el *framework*, utilizamos una herramienta para aplicar las pruebas unitarias.

Para realizar las pruebas utilicé *Simple Test*<sup>1</sup>, ya que es estable, está bien documentado y tiene una amplia gama de opciones.

Primero lo instalé desde la línea de comando del servidor de la aplicación.

```
$ pear install simpletest 1.0.0.tgz
```

Después, para escribir programas que utilicen la librería Simple Test, modifiqué la cabecera aplicada a todos los archivos.

```
<?php
require_once('simpletest/unit_tester.php');
require_once('simpletest/reporter.php');

?>
```

Creé el módulo **test** en donde subí los scripts necesarios para la inicialización de pruebas.

Para realizar la prueba se necesita identificar los archivos y sus métodos que se desean valorar y crear un archivo de pruebas por cada uno de ellos. Cada archivo contendrá una clase y los métodos particulares de cada archivo y que realizarán la prueba.

Para inicializar las pruebas se ejecutó:

```
$ symfony test frontend
```

En donde *frontend* es el nombre de identificación de nuestra aplicación.

Si las pruebas pasaban, se mostraba en pantalla:

```
$ symfony test frontend
Test suite in (test/frontend)
OK
```

<sup>1</sup> [http://www.lastcraft.com/simple\\_test.php](http://www.lastcraft.com/simple_test.php)

```
Test cases run: 1/1, Passes: 5, Failures: 0, Exceptions: 0
```

En caso contrario, se mostrará las veces y el número de pruebas que fallaron. Para solucionarlo, debí eliminar prueba por prueba y correr el comando repetidamente hasta encontrar elementos conflictivos.

### 11.1 Plan de Pruebas

Las pruebas no es demostrar que el sistema trabaja correctamente. Para implementar un plan de pruebas se necesita organizar a un equipo de personas de la siguiente forma.

- El desarrollador de cada módulo es responsable de probar su propio código.
- El equipo de desarrollo es responsable de ejecutar las pruebas y, de ser conveniente, cada desarrollador prueba el código escrito por otros desarrolladores.

El equipo de pruebas:

1. Determinará que módulo esta listo para ser puesto en producción.
2. Proveerá servicios de pruebas al administrador del proyecto.
3. Ejecuta pruebas.
4. Reporta errores.
5. Se asegura de que el equipo de desarrollo conozca los defectos.
6. Describen los problemas del producto.
7. El nivel de pruebas realizado.
8. Estimado de la calidad del producto.

El equipo de desarrollo:

1. Realiza la depuración (el debugging).
2. Realiza pruebas de usabilidad.
3. Realiza manuales técnicos y documentación.

Se realizaron varias pruebas a la aplicación de la siguiente forma:

1. Identificación (alcance de las pruebas)
2. Inventario (identificar módulos o acciones a ser probadas)
3. Priorizar (iniciar con los módulos o acciones mas importantes)
4. Implementación (ejecutar las pruebas)
5. Análisis Retrospectivo (evaluar el proceso realizado)

Para ejemplificar las pruebas realizadas a los *scripts*, presentaré las realizadas a los archivos que son parte de la biblioteca auxiliar del proyecto.

Archivo	Descripción
Tag.class.php	Identifica palabras claves y las guarda para su uso.
MyFilterTag.class.php	Detecta el <i>user-agent</i> del usuario y lo guarda para su uso. También define el idioma del sitio.
myFileValidator.class.php	Valida los archivos mp3 que se suben al sitio.
myImageValidator.class.php	Valida las imágenes que se suben al sitio.
myPasswordValidator.class.php	Valida que la contraseña de los usuarios sea una cadena de caracteres y coincida con la de la base de datos.
myUniquePodcast.class.php	Verifica que el usuario tenga una sola cuenta.
toptenRSS.class.php	Genera dinámicamente el archivo RSS de los mejores episodios de podcasts.

Archivo	Prueba 1	Prueba 2
Tag.class.php	<b>Pasó</b>	<b>Pasó</b>
MyFilterTag.class.php	Se encontró que el <i>script</i> se mandaba a llamar dos veces innecesariamente.	<b>Pasó</b>
	<b>Solución:</b> Reubicar el código que hacía que fuera llamado por segunda vez.	
myFileValidator.class.php	Se encontró que el <i>script</i> no hacía diferencia entre formatos de archivos.	<b>Pasó</b>
	<b>Solución:</b> Incluir el uso de <i>mime-type</i> para analizar el formato de los archivos.	
myImageValidator.class.php	Se encontró que el <i>script</i> no hacía diferencia entre formatos de archivos.	<b>Pasó</b>
	<b>Solución:</b> Incluir el uso de <i>mime-type</i> para analizar el formato de los archivos.	
myPasswordValidator.class.php	<b>Pasó</b>	<b>Pasó</b>
myUniquePodcast.class.php	Se encontró que ésta clase consultaba numerosas veces a la base de datos.	----
	<b>Solución:</b> Se decidió eliminar la clase.	
toptenRSS.class.php	Se encontró que no se completaba la generación del archivo RSS.	<b>Pasó</b>
	<b>Solución:</b> Las pruebas arrojaron la línea de código errónea y fue corregida.	

Tabla 10 Listado de archivos probados mediante pruebas unitarias

También se realizaron pruebas en otros archivos del proyecto los cuales no requirieron el uso *Simple Test*. Los errores encontrados se solventaron y documentaron.

## 12. Documentación

Uno de los temas al que se le hizo mas hincapié en la asignatura de 'Ingeniería de Software' fue la documentación debido a que se cuentan registros de las acciones, plantillas o clases. Esto nos permitirá ahorrar tiempo si deseamos modificar algún archivo de nuestro sistema. Se manejó la documentación de dos formas, la primera consiste en la elaboración de un documento en el que se explica un desarrollo en particular y la segunda se trata de documentar dentro del código fuente las funciones, clases o plantillas explicando de manera clara el funcionamiento de cada uno de estos elementos.

### 12.1 Documentación del desarrollo

Durante el desarrollo de la aplicación Web, es de enorme importancia la documentación de los avances en el desarrollo. Para tener un control, se propuso la creación de un documento plantilla en el que se describieran las actividades realizadas en cada desarrollo.

El formato del reporte se puede apreciar en la figura 7.

1. Título del desarrollo: _____
2. Fecha de inicio del desarrollo: _____
3. Fecha de finalización del desarrollo: _____
4. Archivos actualizados: _____
5. Indicar si el desarrollo depende de alguna clase, método o biblioteca de archivos en particular ajena al desarrollo: _____
6. Observaciones: _____
7. Estatus del desarrollo: _____
8. Datos del desarrollador: _____

Figura 7 Reporte sobre desarrollo terminado

### 12.2 Documentación de código

También realicé la documentación de los *scripts*, plantillas, acciones, y archivos de módulo que programé. La documentación incluía una cabecera con el nombre del autor, la fecha de última actualización y una breve descripción de lo que hace el archivo. Además de documentar por líneas los métodos o fragmentos mas representativos de cada archivo.

```
    /*
       Nombre del creador:
       Fecha de última modificación:
       Status del archivo:
       Explicación general de las clases o funciones:
    */

// Explicación de una clase o función en una sola línea

/* Explicación de una clase o función en
mas de una línea
/*
```

Figura 8 Documentación de clases o funciones en el código fuente

### **13. Manutención del sistema**

La manutención que se realiza al proyecto es a través de las pruebas continuas que se realizan para encontrar errores y solventar los detectados. Además, el proyecto tiene un contacto directo con sus usuarios para conocer su opinión sobre el desempeño del sitio y, si hay sugerencias, se analizan y se desarrollan para poder ofrecer el mejor servicio posible. El mantenimiento es constante en todas las áreas que comprenden al proyecto.

El mantenimiento se divide en:

#### **a) Correctivo**

Se corrigen las fallas cuando éstas se presentan. Aunque tiene desventajas al compararlo con el mantenimiento preventivo, debe considerarse tener a la mano los elementos de programación como de documentación necesarios para solventar cualquier problema.

#### **b) Preventivo**

En la manutención preventiva se implementa un plan de trabajo que se aplica de manera periódica en el cuál se busca detectar y corregir fallas de la aplicación. Las ventajas de aplicar el mantenimiento preventivo son:

- Confiabilidad del funcionamiento del sistema
- Disminución de las posibilidades para desactivar algún módulo en producción
- Registros de las fallas y sus soluciones para próximos eventos

#### **c) Perfectivo**

Significa la modificación del código y de los componentes de la aplicación, después de su puesta en funcionamiento para mejorar el rendimiento o eficacia de su accionar.

#### **d) Adaptativo**

En este tipo de mantenimiento se modifica el código o componentes de la aplicación, cuando estos ya se encuentran en producción, para adaptarlo a las nuevas tecnologías.

## **14. Seguridad**

La seguridad se diagnostica de la siguiente manera:

### **14.1 Seguridad de la aplicación**

Se presentan los elementos de seguridad referentes a la aplicación Web.

#### **14.1.1 Evaluación de vulnerabilidades**

Se buscó determinar las fallas de seguridad de la aplicación, establecer el status de seguridad por períodos y así permitir tomar medidas preventivas para evitar abusos al sistema por terceros.

#### **14.1.2 Factores que facilitan intrusiones**

Para evitar intrusiones, realizo periódicamente las siguientes acciones:

- b) cambio de contraseñas a los accesos por ftp y a las bases de datos mediante línea de comando.
- c) evito el uso indebido de comandos o de utilidades poco seguras
- d) perfeccionar la programación de los módulos mas representativos
- e) resolver fallas de configuración al instante de ser detectada alguna anomalía
- f) Manutención del sistema operativo actualizándolo periódicamente
- g) Autenticar a los usuarios cada vez que se registran o inician sesión al sistema

#### **14.1.3 Control de acceso al servidor**

Se busca tener protección en el control de acceso al servidor con el uso de firewalls, en dónde se abren y cierran solo los puertos necesarios. Además se mantiene vigilado el accionar de los usuarios del sistema mediante sistemas de monitoreo.

#### **14.1.4 Respaldos**

Para evitar pérdidas de datos, ya sea por algún usuario malintencionado o por cualquier otra razón, se realizan respaldos continuos. Sin embargo, es imposible respaldar todos los archivos del servidor.

Se respalda:

- ❖ Todos la información contenida en la base de datos
- ❖ Todos los módulos que a su vez contienen los archivos del MVC
- ❖ La carpeta de Imágenes, animaciones flash, y librerías.

No se respalda:

- a) Los archivos de audio

## **14.2 Seguridad de los usuarios**

La seguridad del usuario contempla las siguientes acciones automatizadas por parte del sistema:

- a) Activación de cuenta por Email para verificar que el usuario existe
- b) Verificación periódica del status del usuario
- c) No se presenta ningún tipo de información confidencial del usuario en el sitio Web
- d) La autenticación del usuario es necesaria para realizar modificaciones
- e) Si no se ha detectado modificaciones del usuario autenticado por un lapso de 20 minutos, el sistema vuelve a pedir el inicio de sesión por parte del usuario.

### **15. Resolución de problemas**

Para la resolución de problemas he implementado un sistema automatizado que se maneja de la siguiente forma:

1. Identificación del problema
2. Búsqueda de las posibles causas
3. Aplicación de la mejor solución
4. Período de pruebas
5. Documentación del problema y su solución

Todos los problemas resueltos se verifican periódicamente para conocer su status.



## 16. Participación Profesional

Mi participación dentro del proyecto es el siguiente:

### a) Analista y diseñador

A partir de los requerimientos, agrupé toda la información y datos obtenidos para identificar lo que se deseaba realizar, para después repartir responsabilidades e identificar el tiempo para cada tarea. Además, diseñé la forma y arquitectura de trabajo.

### b) Diseñador de base de datos

Con base a mi experiencia en las bases de datos, analicé los requerimientos y diseñé la base de datos del proyecto.

### c) Programador

Forme parte del grupo de programadores del proyecto con la tecnología PHP5 y el uso del framework Symfony, utilizando un enfoque de orientación a objetos.

### d) Realización de pruebas y documentación

Finalmente y como complemento a las buenas prácticas de la ingeniería en programación, apliqué pruebas unitarias y documenté lo realizado en el proyecto (por módulo y por código de *script*).

## **17. Resultados y Aportaciones**

Los resultados obtenidos fueron una aplicación Web 2.0 programada en PHP5 con el uso de un *framework*, completa en toda la extensión de la palabra.

### *Estructura de datos*

La aplicación tiene una organización de archivos adecuada, ya que su estructura divide claramente el modelo, la vista y el controlador del proyecto.

### *Modelo de datos*

Siguiendo la metodología propuesta, se diseñó una base de datos de la cuál obtuvimos nuestro modelo de datos. Las entidades, sus atributos y las relaciones se revisaron hasta obtener un diseño eficiente que permitiera un desarrollo sin complicaciones.

### *Ingeniería de Software*

Durante todo el desarrollo se aplicaron todas las recomendaciones de la Ingeniería de programación, lo cuál resultó en un proyecto completo, seguro y listo para ser presentado a todo el mundo por medio de Internet.

En mi opinión, se desarrolló una aplicación que tiene todas las características técnicas y de programación para tener éxito. Debemos recordar que el sitio Web representará un servicio para todos los usuarios de Internet que deseen subir o escuchar un podcast. La aportación en el mundo de los podcast la conoceremos con el paso del tiempo y las buenas decisiones que se tomen a futuro.

## **18. Conclusiones**

Se desarrolló una aplicación Web que permite a los usuarios de Internet publicar un podcast. El sitio en Internet que brinda una plataforma de difusión de los podcast para los usuarios es: [www.poderato.com](http://www.poderato.com).

La aplicación Web final maneja la información en una base de datos cuyo manejador es Postgresql en su versión 7.0. A partir del diseño de la base de datos, se creó el modelo de datos el cual complementa el modelo vista controlador con acciones y planillas.

### *Inversión en la creación del proyecto*

El principal objeto de estudio, antes de comenzar a diseñar, fue verificar qué tan viable era construir el proyecto. Entonces pensamos en los costos. Nuestro análisis, en la parte financiera, arrojó que no se invertiría en otra cosa que no fueran las comisiones para los involucrados por el desarrollo, por lo que se dio luz verde, sin embargo en el transcurso del desarrollo se requirió de la inversión en elementos que no se habían contemplado.

### *Tiempo de desarrollo*

El tiempo que nos tomaría desarrollar la aplicación fue objeto de un análisis detallado. Nuestro análisis indicó que en un período X de meses tendríamos la aplicación completa, incluyendo la documentación, pruebas y el plan de manutención de la misma. El tiempo final en el que se completó el proyecto excedía, aunque por poco, nuestra planeación original debido a que en el lapso del desarrollo se incluyeron eventos que no estaban en el plan de trabajo original.

### *Perspectivas a futuro*

Además de lo desarrollado hasta este punto, se tiene un plan de trabajo en el que se pretende trabajar a detalle el uso de los archivos mp3, las estadísticas y la usabilidad en el sitio para que el usuario tenga una mejor experiencia.

### *Reflexión*

Por otra parte me gustaría reflexionar en cuanto a la parte financiera de un proyecto Web. Por lo general, los ingenieros estamos listos para analizar, diseñar, desarrollar, hacer pruebas y documentar todo lo referente a una aplicación. Sin embargo, cuando llega la hora de hacer cuentas, vender el proyecto, o simplemente en la recuperación de la inversión, generalmente se complica debido a la falta de experiencia en ese rubro. A lo largo de la carrera tenemos a penas un par de asignaturas que nos llevan por el camino (Costos y Evaluación de Proyectos y Economía), pero al final de cuentas no queda otra más que investigar por nuestra cuenta hasta encontrar el modelo a seguir que satisfaga una aplicación en particular.

*Beneficios adicionales*

En el transcurso del desarrollo de la aplicación Web obtuve experiencia en elementos de programación y de sistemas que no conocía. Aprendí a instalar, usar y modificar nuevas tecnologías, bibliotecas o módulos de aplicación (como el utilizado en las pruebas).

Después de que culmina la primera etapa de desarrollo de la aplicación Web, cada vez estoy mas convencido de las grandes posibilidades de desarrollo que la Web proporciona.

## **ANEXO**

### **Glosario de términos**

#### **PHP5**

Symfony está desarrollado en PHP5 y dedicado a construir aplicaciones Web con el mismo lenguaje. PHP5 es un lenguaje de programación que se ejecuta del lado del Servidor Web en donde se encuentra instalado y devuelve el resultado del lado del Cliente. La diferencia entre PHP5 y sus versiones anteriores es nuevo modelo orientado a objetos.

#### **Programación Orientada a Objetos (POO)**

Symfony hace un extensivo uso de los mecanismos orientados a objetos, en este caso, encaminados en PHP5. La idea detrás de la programación orientada a objetos es que un programa de computadora pueda ser vista como la composición de una colección de unidades individuales, u objetos, que actúan unas sobre las otras, oponiéndose a la visión tradicional en la cuál un programa puede ser visto como una colección de funciones, o simplemente como un listado de instrucciones para la computadora.<sup>2</sup> PHP5 implementa los paradigmas de clase, objeto, método, herencia y más de la orientación a objetos.

#### **Mapeo Objeto-relacional (ORM)**

Interface que traduce la lógica de objetos a la lógica relacional de una base de datos.

#### **Ingeniería de Software**

La Ingeniería de software designa el conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la sola actividad de programación. Forman parte de esta disciplina las ciencias computacionales y el manejo de proyectos.

#### **Podcaster**

Un podcaster es la persona u organización que graba un podcast y lo publica.

#### **Script**

Guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Son ejecutados por un intérprete de línea de órdenes.

#### **Framework**

Base de programación definida con el que se puede desarrollar un proyecto de software.

#### **Tag**

Palabra clave que define una característica del contenido.

#### **MVC**

Modelo Vista Controlador. Es un patrón de diseño que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

#### **Plug-in**

Programa que puede ser instalada en un sistema que realiza una tarea determinada.

---

<sup>2</sup> Enciclopedia Wikipedia 2008. Definición de “Programación Orientada a Objetos (POO)”

## 19. Bibliografía

- ❖ Fisher, Matt. Developer's Guide to Web Application Security [ILLUSTRATED]. Editorial Syngress. 2005
- ❖ Freire, Eunice. Web Design & Development. Editorial Charles Media. 2006
- ❖ Garfinkel, Simson. Web Security, Privacy and Commerce. 2da Edición. Editorial O'Reilly. 2007
- ❖ Potencier, Fabien. The definitive Guide to Symfony. 1ra Edición. Apress. 2007
- ❖ [http://eisc.univalle.edu.co/materias/Material\\_Desarrollo\\_Software/softwaretestingx.pdf](http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/softwaretestingx.pdf)