

**SISTEMA INTRANET PARA REGISTRO, BUSQUEDA Y CONTROL DE MATERIAL
SONORO DE LA FAMILIA DE EMISORAS INSTITUCIONALES
UIS STEREO Y UIS AM DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER -
SEMCO 1.0**

**IVAN EDUARDO NIÑO ACEVEDO
ANDRES JULIAN VELANDIA ENCISO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2004

**SISTEMA INTRANET PARA REGISTRO, BUSQUEDA Y CONTROL DE MATERIAL
SONORO DE LA FAMILIA DE EMISORAS INSTITUCIONALES
UIS STEREO Y UIS AM DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER –
SEMCO 1.0**

**IVAN EDUARDO NIÑO ACEVEDO
ANDRÉS JULIAN VELANDIA ENCISO**

**Proyecto de Grado
para optar al título de Ingenieros de Sistemas**

**Director
MARTHA VITALIA CORREDOR MONTAGUT
Ingeniera de Sistemas**

**Tutor
IVAN DARIO MONTOYA OSORIO
Comunicador Social**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2004**

AGRADECIMIENTOS

Este proyecto ha sido el producto de un gran esfuerzo, es una prueba del conocimiento adquirido en la formación profesional en la Escuela de Ingeniería de Sistemas de la UIS y es una expresión de gratitud para nuestras familias.

A Dios, porque sin él nada es posible.

A nuestras madres, por ser ejemplos constantes de esfuerzo, sacrificio y trabajo, además de su infinita paciencia y constante apoyo.

A nuestra directora de proyecto, Ing. Marta Vitalia Corredor Montagut, por sus consejos y, sobre todo, por su don de gentes.

A los miembros de TELEUIS, especialmente a su director Iván Darío Montoya Osorio, por su colaboración y paciencia.

A Freddy Angarita, por poner a nuestra disposición su experiencia en el desarrollo de software y su característico sentido del humor que nos dio ánimo en muchas ocasiones.

CONTENIDO

	pág.
INTRODUCCIÓN	16
1. PRESENTACIÓN	19
1.1 PRESENTACIÓN DEL INFORME	19
1.2 DESCRIPCIÓN DEL PROYECTO	20
1.2.1 Objetivo General.	20
1.2.2 Objetivos Específicos	20
1.3 JUSTIFICACIÓN	20
1.3.1 Antecedentes del Proyecto.	20
1.3.2. Impacto	21
1.3.3 Viabilidad	22
1.4 DESARROLLO DEL SISTEMA	22
1.4.1 Hardware	22
1.4.2 Software	23
2. MARCO TEÓRICO	24
2. 1 SISTEMA OPERATIVO	26
2.1.1 Windows 2000	26
2. 2 SERVIDOR	26
2.2.1 Internet Information Server (IIS)	28
2.3 PÁGINAS WEB ESTÁTICAS	28
2.3.1 HTML	29

2.4 PÁGINAS WEB DINÁMICAS EN EL CLIENTE	29
2.4.1 Scripts	30
2.4.2 HTML Dinámico	30
2.4.3 ActiveX	31
2.4.4 Applets de Java	31
2.5 PÁGINAS WEB DINÁMICAS EN EL SERVIDOR	32
2.5.1 Hypertext Preprocessor (PHP)	32
2.5.2 Active Server Pages (ASP)	35
2.5.3 Programas CGI	37
2.6 ARQUITECTURA DE INFORMACIÓN	39
2.6.1 Modelo Cliente / Servidor.	39
2.6.2 Arquitectura de tres capas.	41
2.6.2.1 Capa de Presentación.	42
2.6.2.2 Capa de Datos.	42
2.6.2.3 Capa de Aplicación o de Negocios.	42
2.7 BASES DE DATOS	42
2.7.1 MySQL	43
2.8 PROGRAMACIÓN WEB	44
2.8.1 PHP	44
2.8.2 HTML	45
2.9 PLATAFORMA SOFTWARE.	45
2.9.1 Windows 2000 Server	45
2.10 SEGURIDAD INFORMÁTICA	45

2.10.1	Seguridad Lógica	45
2.10.1.1	Controles de acceso	46
2.10.2	Ataques más comunes en la Seguridad Informática	49
2.10.3	Formas de Protección	50
2.10.3.1	Administración de la seguridad	50
2.10.3.2	Gestión de claves “seguras”.	52
2.10.3.3	Seguridad en los servicios	52
2.10.4	Políticas de seguridad	53
2.10.4.1	Elementos de una política de seguridad informática	54
2.10.4.2	Estructura de las políticas de seguridad	55
2.10.4.3	Aspectos a tener en cuenta para establecer políticas de seguridad	56
3.	MARCO METODOLOGICO	57
3.1	FASE DE INICIO	57
3.1.1	Planeación	57
3.1.1.1	Hitos de la Fase de Inicio	57
3.1.1.2	Flujos, actividades y productos a entregar en la Fase de Inicio	58
3.1.2	Iteración Única: estudio de la viabilidad del proyecto	59
3.1.2.1	Captura de Requisitos	59
3.1.2.2	Análisis	73
3.1.2.3	Diseño	75
3.1.3	Evaluación	76
3.1.3.1	Desarrollo de los modelos de casos de uso	77
3.1.3.2	Vista de la Arquitectura	78

3.1.3.3 Cumplimiento de los criterios	78
3.2 FASE DE ELABORACION	79
3.2.1 Planeación	80
3.2.1.1 Hitos	80
3.2.1.2 Flujos, actividades y productos a entregar en la Fase de Elaboración	80
3.2.2 Iteración Única: Determinación de una arquitectura estable	81
3.2.2.1 Captura de Requisitos	81
3.2.2.2 Análisis	87
3.2.2.3 Diseño	89
3.2.2.4 Implementación	95
3.2.3 Evaluación	96
3.2.3.1 Desarrollo del modelo de casos de uso	96
3.2.3.2 Vista de la arquitectura	96
3.2.3.3 Cumplimiento de criterios	96
3.3 FASE DE CONSTRUCCION	98
3.3.1 Planeación	98
3.3.1.1 Hitos y asignaciones de tiempo	98
3.3.1.2 Flujos, actividades y productos a entregar en la Fase de Construcción	99
3.3.2 Subsistemas Generales	99
3.3.2.1 Subsistema de Gestión de Sesiones	99
3.3.2.2 Subsistema de Gestión de Interfaz	101
3.3.3 Subsistemas Específicos	102
3.3.3.1 Subsistema de Administración	102

3.3.4 Evaluación	103
4. CONCLUSIONES	105
5. RECOMENDACIONES	106
BIBLIOGRAFÍA	108
ANEXOS	109

LISTA DE TABLAS

	pág
Tabla 1. Requisitos Iniciales del Sistema	60
Tabla 2. Riesgos críticos identificados	63
Tabla 3. Proceso de negocio “Manejo de Base de Datos”	65
Tabla 4. Proceso de negocio “Envío y recepción de Mensajes”	65
Tabla 5. Proceso de negocio “Consultar Material”	65
Tabla 6. Actores del Sistema	67
Tabla 7. Requisitos adicionales identificados en la fase de inicio	73
Tabla 8. Casos de uso identificados en la Fase de Inicio	77
Tabla 9. Flujos desarrollados en la fase de Inicio	78
Tabla 10. Descripción del modelo de recepción de sugerencias	82
Tabla 11. Descripción del modelo de Formulación de sugerencias	82
Tabla 12. Descripción del modelo de Búsqueda de Material	84
Tabla 13. Descripción del modelo de administración de usuarios de la pizarra	84
Tabla 14. Descripción del modelo de envío y recepción de mensajes	85
Tabla 15. Descripción del modelo de monitoreo de mensajes	86
Tabla 16. Casos de uso identificados hasta el final de la fase de elaboración	96
Tabla 17. Tabla de asignación de tiempo a cada una de las iteraciones	99
Tabla 18. Tabla final de casos de uso del sistema	103

LISTA DE FIGURAS

	pág
Figura 1. Esquema de interpretación de PHP	34
Figura 2. Modelo Cliente / Servidor	39
Figura 3. Modelo general de casos de uso	64
Figura 4. Modelo de Objetos general	66
Figura 5. Modelo de casos de uso general del sistema	70
Figura 6. Casos de uso para la administración de usuarios, base de datos y pizarra.	72
Figura 7. Identificación de paquetes a partir de las clases de dominio.	74
Figura 8. Clases derivadas a partir del modelo de dominio.	75
Figura 9. Diagrama de despliegue del sistema	76
Figura 10. Subsistemas identificados a partir de los requerimientos del sistema	76
Figura 11. Modelo de recepción de sugerencias	82
Figura 12. Modelo de Formulación de sugerencias	83
Figura 13. Modelo de Búsqueda de Material	83
Figura 14. Modelo de administración de usuarios de la pizarra	84
Figura 15. Modelo de envío y recepción de mensajes	85
Figura 16. Modelo de monitoreo de mensajes	86
Figura 17. Paquete de análisis de gestión administrativa, conteniendo los paquetes de gestión de Material, Usuarios y Mensajes	87
Figura 18. Diagrama de colaboración para el caso de uso Agregar usuario	89
Figura 19. Subsistemas de diseño	91

Figura 20. Modelo de tres capas	91
Figura 21. Modelo de páginas dinámicas con código incrustado	92
Figura 22. Página Web con códigos de campo	93
Figura 23. Interfaz General de Usuario	94

LISTA DE ANEXOS

	pág.
ANEXO A. MODELO ENTIDAD RELACION SISTEMA SEMCO	109
ANEXO B. DESCRIPCIÓN DE LA CONFIGURACIÓN DE LOS DIRECTORIOS DEL SISTEMA	110
ANEXO C. INTERFASES DE USUARIO	115

RESUMEN

TITULO:

SISTEMA INTRANET PARA REGISTRO, BUSQUEDA Y CONTROL DE MATERIAL SONORO DE LA FAMILIA DE EMISORAS INSTITUCIONALES UIS STEREO Y UIS AM DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER – **SEMCO**. *

AUTORES:

NIÑO ACEVEDO, Iván Eduardo, VELANDIA ENCISO, Andrés Julián **

PALABRAS CLAVES:

Sistema intranet, proceso unificado, administración de material sonoro, PHP, MySQL.

DESCRIPCIÓN:

SEMCO es un sistema de información que permite la administración de algunos procesos de las emisoras institucionales UIS Stereo y UIS AM de la Universidad Industrial de Santander, como son la búsqueda bajo diferentes parámetros, tales como Intérprete, Compositor, Arreglista, Sello Discográfico, entre otros, del material sonoro con que cuenta la sala de Fonoteca; la administración de dicho material, entendida ésta como el préstamo de material y el proceso de inventario de la sala de Fonoteca, y el establecimiento de un canal de comunicación electrónico entre los miembros del equipo de trabajo.

El desarrollo del sistema está soportado en el proceso unificado, metodología que se apoya en el lenguaje de modelado unificado (UML) y está regida por tres aspectos: dirigida por casos de uso, centrado en la arquitectura y es iterativo e incremental. Por otro lado, el sistema está soportado en software libre como la base de datos MySQL, y el lenguaje de scripts PHP.

El sistema se desarrolló a lo largo de varias fases, desde la fase de inicio recopilando los requisitos, pasando por la fase de elaboración que permite definir la arquitectura y finalizando en la fase de construcción que permite desarrollar el producto a entregar a los usuarios finales.

* Trabajo de Grado

** Facultad de Ingenierías Físico mecánicas, Ingeniería de Sistemas e Informática, Marta Vitalia Corredor M.

ABSTRACT

TITLE:

INTRANET SYSTEM FOR THE REGISTRATION, SEARCH AND CONTROL OF AUDIBLE MATERIAL OF THE FAMILY OF INSTITUTIONAL RADIO STATIONS UIS STEREO AND UIS AM OF THE INDUSTRIAL UNIVERSITY OF SANTANDER – SEMCO. *

AUTHORS:

NIÑO ACEVEDO, Iván Eduardo, VELANDIA ENCISO, Andrés Julián **

KEY WORDS:

Intranet System, Unified Process, audible material management, PHP, MySQL.

DESCRIPTION:

SEMCO is an information system that allows for the administration of certain processes inside the institutional radio stations UIS Stereo and UIS A.M. of the Industrial University of Santander, such as searching for specific audible material under different search criteria (e.g. Composer, Label, Performer, amongst others); the management of such material, which implies its loan to the organization's staff members, and the process of inventory in the audible material's room, as well as the establishment of an electronic communication channel for the staff members.

The system hereby described has been developed under the Unified Process, which is a methodology supported by the Unified Modeling Language (UML), whose most important features must be mentioned: it is use-case driven; its approach is both iterative and incremental, and it is centered on architecture. Additionally, the system is backed up by open software such as PHP and MySQL.

The system has been developed throughout several phases, from the Inception Phase collecting requirements, continuing to the Elaboration Phase in order to define the architecture, all the way up to the Construction Phase, which allows for developing the application software to be delivered to the end users.

* Trabajo de Grado

** Facultad de Ingenierías Físico mecánicas, Ingeniería de Sistemas e Informática, Marta Vitalia Corredor M.

INTRODUCCIÓN

La radio es en la actualidad uno de los medios de comunicación mas útiles y populares del mundo. Es el medio de más fácil acceso y a través del cual se pueden transmitir las mejores ideas despertando la imaginación del oyente y por consiguiente ayudando al desarrollo mental.

Existen inmersas en este medio las emisoras de carácter educativo y cultural que tienen como finalidad mucho más que entregar música o información; quieren entregar a la gente educación, formación, cultura, en diferentes aspectos como el musical, artístico, noticioso, literario, etc. La Universidad Industrial de Santander como centro de formación universitaria no podía estar aislada de este proceso y posee en la actualidad un sistema radial compuesto por una emisora en Amplitud Modulada UIS AM y una en Frecuencia Modulada UIS Stéreo. Estas dos emisoras cumplen una labor educativa y cultural y poseen una alta tecnología en equipos de producción y transmisión.

La emisora UIS Stéreo cuenta en este momento más de siete años de funcionamiento y su compañera UIS AM está próxima a cumplir dos años. Durante todos estos años se han conseguido y almacenado materiales sonoros que garantizan la continuidad y variedad en la programación, materiales en diferentes formatos como Casetes, acetatos, discos compactos, MiniDisc, etc., que son almacenados en un cuarto llamado FONOTECA, el cual no posee actualmente un sistema de inventario y control que sea acorde al nivel tecnológico de las emisoras, teniendo en cuenta que la cantidad de material asciende a más de 10000 títulos.

Es por eso que surge la idea de crear un Sistema de Información que supla esta necesidad y que además proporcione facilidades de búsqueda que ayuden a propiciar un desempeño más dinámico por parte de realizadores, controladores y jefes de programación. Este sistema será un canal entre los diferentes personajes que participan en la realización de un espacio radial: los realizadores, quienes están a cargo de un espacio; los controladores, quienes están encargados de la parte técnica siguiendo las indicaciones del realizador; y el jefe de programación, quien está al tanto del programa

para corregir errores, además de facilitar los materiales que el realizador necesite, solicitados previamente. Únicamente en UIS Stéreo el número de espacios radiales supera los 30, y un alto porcentaje de éstos son conducidos, cada uno, por una persona diferente, en un horario de programación que se extiende desde las 7 de la mañana hasta las 11 de la noche. El tiempo necesario para que el realizador, el controlador y el jefe de programación coordinen sus labores es significativo, hecho que indudablemente afecta negativa y directamente otras labores importantes, tales como la producción, la pregrabación, la preparación o búsqueda de material informativo alusivo al espacio etc.

De nada serviría una sencilla base de datos almacenada en un computador que se encuentre dentro del cuarto de FONOTECA, el cual haya que encender, consultar y buscar cada vez que alguien necesite algún material; serviría como un sistema de inventario pero no estaría al nivel de exigencia que la dinámica de trabajo de las emisoras demanda.

La Intranet se convierte cada vez en la forma más usada por las empresas a nivel mundial, porque permite la comunicación interna entre los equipos de los empleados, brinda una comunicación por red que permite agilizar los procesos administrativos de la empresa, mejorar su rendimiento e incrementar su productividad. De esta forma un sistema instalado en un ambiente *cliente-servidor* es el más adecuado para satisfacer necesidades como las anteriormente planteadas.

Un sistema en ambiente cliente-servidor permite al usuario en una máquina, llamada "Cliente", requerir algún tipo de servicio ofrecido por una máquina a la que está unida, llamada "Servidor", mediante una red como una LAN (Red de Área Local) o una WAN (Red de Área Mundial). Para nuestro caso específico se trata de una LAN. El término *servidor* se aplica a cualquier programa que ofrece un servicio y que puede ser utilizado a través de la red. Los servidores aceptan peticiones recibidas por medio de la red, realizan el servicio y regresan el resultado a quien hizo la petición.

La metodología de Casos de Uso permite que la herramienta a desarrollar pueda ser utilizada según las necesidades del usuario, las cuales fueron manifestadas por los jefes de programación del sistema radial en charlas entabladas antes del inicio de la etapa de

diseño; además, permite la elaboración de módulos de desarrollo que giran alrededor de una necesidad o caso de uso específico.

1. PRESENTACIÓN

1.1 PRESENTACIÓN DEL INFORME

Aunque a nivel general no se le de gran importancia a esta sección del documento porque no se considera como una parte relevante en el desarrollo del proyecto y en el cumplimiento de los objetivos planteados, esta sección guía al lector al momento de estudiar el proyecto paso a paso para, a su vez, facilitarle su entendimiento y comprensión.

Este documento debe ser comprensible a cualquier tipo de lector, sin importar su grado de preparación en el tema. Por tal motivo debe, basándose en una implantación clara y concisa, mostrar el proceso que se llevó a cabo en el desarrollo del proyecto, dejando ver la fundamentación teórica utilizada como referencia para la implantación del sistema y el cumplimiento de los objetivos iniciales.

La información que se ofrece en el presente documento ha sido clasificada principalmente en dos partes que reflejan los pasos que se siguieron desde un planteamiento inicial, hasta la culminación del proyecto, esto con el fin de facilitar al lector su entendimiento.

Capítulo 1, el cual muestra una descripción detallada de todos y cada uno de los aspectos que fueron básicos en el surgimiento del proyecto, se listan los objetivos que se desean alcanzar y se muestra la importancia de su desarrollo y la viabilidad del mismo.

Capítulo 2, el cual presenta información general necesaria para el entendimiento del proyecto como lo son las generalidades acerca de una red Intranet, y del desarrollo de sistemas de información Web, así como la fundamentación teórica que es soporte para el desarrollo del proyecto. Esta información contempla aspectos teóricos y técnicos entre los cuales se encuentra la Seguridad Informática y todo el soporte software respectivamente.

Capítulo 3, el cual describe cada una de las fases del desarrollo del proyecto. En la fase de Inicio se realiza un análisis para justificar la puesta en marcha del proyecto; en la fase de elaboración se muestra la forma de construcción de la arquitectura estable para guiar el posterior desarrollo del sistema y, finalmente, en la fase de construcción se expone el desarrollo de una versión funcional del sistema.

1.2 DESCRIPCIÓN DEL PROYECTO

1.2.1 Objetivo General. Crear un completo Sistema de Información que automatice la búsqueda del material fonográfico bajo diferentes parámetros, el control de préstamos y entregas, las sugerencias sobre la adquisición de nuevos títulos y permita el acceso a información adicional alusiva a dicha búsqueda.

1.2.2 Objetivos Específicos. Implementar un Sistema de Información en red para el material fonográfico que:

- Facilite la administración de los recursos sonoros disponibles para la realización de los espacios radiales de las Emisoras.
- Facilite la consulta del material existente a los realizadores de programas, a los controladores o a algún miembro del equipo de trabajo de la entidad bajo diferentes parámetros sugeridos por los usuarios.
- Simplifique el proceso de inventario periódico por medio de la implementación de un módulo diseñado para tal fin.
- Sugiera la adquisición de nuevos títulos para su posterior adquisición
- Indique gráficamente la ubicación del material fonográfico requerido.

1.3 JUSTIFICACIÓN

1.3.1 Antecedentes del Proyecto. El material sonoro es de fundamental importancia en el desarrollo de un espacio radial. Si se habla de música, puede ayudar a hacer una pausa en medio de una extensa charla devolviendo de nuevo la atención del oyente,

puede ambientar una conversación o una larga intervención ayudando aún más a la creación imaginaria o puede sencillamente otorgar un largo espacio de relajación y disfrute. También existen programas completos almacenados en formatos sonoros como discos compactos, casetes o acetatos que son de vital importancia para una emisora y más aún si se hace referencia a La Voz de la Universidad, UIS Stereo.

En un comienzo, en la emisora el material era guardado en un pequeño armario de madera ubicado dentro de la misma cabina de transmisión. Con el paso del tiempo, gracias a donaciones, adquisiciones y convenios, se hizo necesaria la creación de un cuarto específico y una forma de controlar el material existente. Se optó entonces por clasificar el material por géneros musicales, asignando una letra a cada uno de ellos y enumerando el material existente para dicho género; adicionalmente se elaboraba un archivo de texto para controlar la entrada y salida de nuevos materiales.

Esta idea fue buena en un comienzo, pues permitía conocer los recursos existentes dentro de la Fonoteca, pero se fue volviendo impráctico a medida que el número de CD's, acetatos o casetes iba aumentando. Surgió entonces la necesidad de agilizar el proceso de consulta, búsqueda y control de este material que permita un desempeño óptimo del personal y de las emisiones tanto de UIS Stéreo como de UIS AM.

1.3.2. Impacto. El desarrollo de SEMCO será de gran ayuda para el nuevo centro de producción radial de TELEUIS, ya que ante una nueva emisora el ordenamiento de los recursos es de vital importancia; así mismo, el aumento de personal de trabajo implica indudablemente que se ejerza un mayor control sobre el material para evitar pérdidas. De igual forma se debe proporcionar a los usuarios una forma ágil de consulta para evitar demoras y confusiones.

Se invita al lector a imaginar por un instante al personal de UIS Stéreo y UIS AM buscando simultáneamente, de forma manual, el material fonográfico para sus emisiones en un cuarto estrecho con mas de 10000 títulos y sin saber por donde empezar. Para complicar un poco más el escenario se recuerda al lector que cada semana llega al

menos una docena de nuevos títulos para ordenar y almacenar. Sin duda se convertiría en un verdadero caos.

Una terminal de computador donde se pueda ejecutar esta búsqueda podría ayudar a solucionar el problema, la cual dice cuántos, cuáles y donde están exactamente los temas de interés relacionados con el artista o temática a tratar en un espacio radial determinado. En cuestión de segundos, al entrar al cuarto de materiales, es posible retirar el material necesario, conociendo su ubicación con toda certeza, y disponerse a realizar el programa.

1.3.3 Viabilidad. Para la realización de este proyecto se utilizarán herramientas de desarrollo de libre distribución, lo cual permite un uso libre de restricciones de tipo legal. Evidentemente, esta situación se revierte en bajos costos en el momento de la implementación e implantación.

Desde el mes de abril de 2002, TELEUIS cuenta con un moderno centro de producción y emisión. Ubicado en la sede UIS Bucarica, cuenta con tres cabinas: dos de emisión, una para cada una de las señales emitidas y otra para la producción de espacios de las dos emisoras. Estas tres cabinas tienen computadores en red mediante los cuales es posible enviar y recibir los materiales producidos sin necesidad de trasladarlos en medios físicos.

Esta institución necesita fortalecerse en el área de Sistemas de Información, razón por la cual este proyecto se vuelve absolutamente indispensable; así, su realización no sólo es viable, sino es además una imperiosa necesidad.

1.4 DESARROLLO DEL SISTEMA

Para el desarrollo de SEMCO 1.0 se requieren los siguientes recursos de hardware con estas especificaciones mínimas:

1.4.1 Hardware

Servidor de Prueba:

- Procesador Pentium III de 500MHz o superior
- Memoria RAM de 128MB

- Disco Duro de 10GB
- Tarjeta de Red

Ciente de Prueba:

- Procesador Pentium de 133MHz o superior
- Memoria RAM de 32MB
- Tarjeta de Red

Otros:

Servidor de la red Microsoft de la Emisora y, eventualmente, acceso a la red de área local (LAN) de la Universidad Industrial de Santander.

1.4.2 Software

- Manejador de base de datos : MySQL 3.23
- Lenguaje de Scripts : PHP 4.0
- Sistema Operativo : Windows 2000
- Servidor : IIS¹ 5.1
- Entorno gráfico de MySQL : MySQL Front versión 2.2
- Herramientas de diseño : UMLStudio 5.0.

¹ Internet Information Server, Microsoft Corporation.

2. MARCO TEÓRICO

Actualmente, la red Internet sirve de soporte a una gran cantidad y variedad de sistemas de información y comunicaciones que engloban áreas tan importantes como la investigación, el comercio electrónico, la publicación general de información, correo electrónico, sector multimedia y un sin fin de posibilidades de las que potencialmente se pueden beneficiar todas las personas. Una aplicación de las bondades de Internet es Intranet, que funciona de manera similar a la primera con la diferencia de ser de uso exclusivo de una organización particular.

Intranet es la herramienta que posibilita acumular, administrar, clasificar y presentar en forma amigable la comunicación e información interna de una organización.

Junto al creciente uso de redes internas en empresas, ha surgido la demanda de un mejor y más simple manejo de la información dentro de las mismas. Además, la globalización y el requerimiento de actuar según economía de escalas para competir, impulsa la fusión de empresas o expansión de éstas, dentro y fuera de una misma región. Esto conlleva la necesidad de trabajar sincronizadamente desde distintos lugares. Intranet permite acceder a toda la información vía Web en forma remota, desde cualquier punto en el mundo.

Ventajas principales:

- Tele-trabajo, acceso remoto y fácil navegación gracias a ser 100 % basado en la Web.
- Centraliza el trabajo y la información sin importar donde se encuentre el personal.
- Administra todo el flujo de conocimiento e información de una compañía.
- Define perfiles de usuario y niveles de seguridad que garantizan acceso sólo a las personas indicadas.
- Es el lugar en común para todos los miembros de la firma.
- Fácil de localizar desde cualquier ubicación.
- Asegura que ninguna información se pierda o no llegue a la persona apropiada.

- Comparte la información en forma general, por grupos o individualmente.
- Acorta tiempos facilitando la comunicación interna y el acceso a conocimiento.
- Ahorra costos en papel e impresión manejando todo digitalmente.
- Reduce costos telefónicos al permitir acceso remoto.
- Optimiza recursos por encontrarse disponible la información para todos los que pudieran necesitarla.
- Evita “vacíos” en el flujo de la comunicación interna y errores por falta de conocimiento.
- Seguimiento detallado de los datos manejados por la empresa.

Características básicas:

- Totalmente basado en la Web.
- Foros internos de discusión según temáticas.
- Carpetas para todos los tópicos relevantes.
- Determina códigos de acceso según niveles de seguridad.
- Crea lugares para publicar notas, artículos, opinión, etc.
- Quien origina la información define, en conjunto con el administrador de la Intranet, las personas que tienen acceso a dicha información.
- Facilita la realización de encuestas internas de manera ágil y simple.
- Permite o niega el acceso a los resultados de encuestas según el perfil de seguridad.
- Calendario personal.
- Agenda de contactos (base de datos sobre los contactos de la empresa).
- Publica eventos destacados, novedades, etc.
- Requiere un administrador de la Intranet.
- Posibilita acceso a publicaciones externas predeterminadas (por ejemplo: periódicos).
- Incluye buscador que simplifica el acceso a material acumulado.
- Opera como administrador de mensajes de e-mail para cada individuo.

Para la realización del “Sistema Intranet Para Registro, Búsqueda y Control de Material Sonoro de la Familia de Emisoras Institucionales UIS Stereo Y UIS AM de la Universidad

Industrial de Santander - SEMCO 1.0”, se concluyó utilizar, en su gran mayoría, herramientas de libre distribución, que son productos exclusivamente “hechos en Internet”, lo cual se traduce en costo casi nulo y con un grupo de desarrolladores de excelente calidad, logrando una solución económica sin que el sistema pierda su robustez y seguridad.

Dentro de estas herramientas se encuentra el sistema operativo Windows 2000, plataforma ampliamente utilizada para servidores Web; el lenguaje de programación PHP, que es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor; MySQL para el manejo de base de datos, Internet Information Server (IIS) servidor Web muy utilizado en la actualidad por su eficiencia, funcionalidad y velocidad.

Para entender las razones por las que se escogieron estas herramientas se muestra a continuación una división de los principales temas que es necesario profundizar.

2. 1 SISTEMA OPERATIVO

2.1.1 Windows 2000 Es el sistema operativo para redes que reemplazó al exitoso producto Windows NT. Fue construido enteramente sobre NT pero también le fueron añadidas numerosas e importantes mejoras en cuanto a confiabilidad, seguridad, mantenimiento, escalabilidad y soporte a hardware².

2. 2 SERVIDOR

Las condiciones para instalar un servidor son:

Hardware

El computador que se destine para el servicio Web debe reunir las siguientes características:

² Visite el enlace <http://labmice.techtarget.com/windows2000/default.htm> para mayor información al respecto.

Interfaz de red:	Puede ser una tarjeta de red Ethernet, o un módem para RDSI o para Frame Relay.
Sistema de almacenamiento de alto rendimiento:	Para mantener el espacio de información alojado en el servidor. Se recomienda una controladora SCSI y un mínimo de 2 MB de capacidad disponible.
Memoria central de elevada capacidad:	Para empezar recomendable 128 MB, para carga media 256 MB, y para una explotación fuerte, por lo menos 512 MB.
Procesador:	Es importante para la ejecución de los programas CGI's o para acceder a Sistemas de Gestión de Bases de Datos (SGBD).

Software

- Un sistema operativo multitarea y con soporte de red con características deseables como: robustez, fiabilidad y seguridad.
- Un servidor Web.
- Software de producción multimedia, editores de HTML, programas para el tratamiento de imágenes y gráficos.
- Sistema de Gestión de Bases de Datos. Imprescindibles para proveer contenidos dinámicos ya sea directamente, o a través de algún lenguaje de programación.

Antes de poner un servidor Web en funcionamiento hay que recopilar cierta información:

Memoria. El tamaño de memoria afecta al número de procesos que se puede tener en ejecución, es decir, en espera de atender peticiones.

Puerto de conexión. Generalmente es el 80.

Dirección IP. El servidor debe disponer de una dirección IP fija que le identifique de forma unívoca en la red. También tendrá un nombre asociado y registrado en un servicio de DNS.

Directorio de instalación. El servidor Web actúa en varios directorios simultáneamente.

Usuario y derechos de acceso. En un sistema multiusuario, cada usuario tiene unos derechos o privilegios de acceso a los recursos del sistema. En este entorno, el servidor Web también se ejecuta bajo un usuario (*usuario del servicio*) que se define explícitamente. Este usuario del servicio no debe comprometer jamás la seguridad del sistema.

A continuación se comentará acerca de las características del servidor que se utilizó durante el desarrollo, en la instalación y funcionamiento de SEMCO 1.0.

2.2.1 Internet Information Server (IIS). Es el software estándar que soporta comunicaciones en Internet sobre los Sistemas Operativos de la familia Windows: NT, XP y 2000.

Sus características principales son, entre otras, la facilidad de instalación y configuración, un API potente para desarrollar aplicaciones de Internet/Intranet, su amplia aceptación de su *Active Server Pages* basado en VBScript o JScript (para el desarrollo de Server Side Includes), y el hecho de ser gratuito (viene incluido en los sistemas operativos Windows NT y superiores).

2.3 PÁGINAS WEB ESTÁTICAS

Las páginas Web son textos ASCII escritos en el lenguaje HTML (Hypertext Markup Language) que se transfieren entre los servidores Web y los navegadores de los clientes mediante el protocolo HTTP (Hypertext Transfer Protocol). Las páginas Web eran estáticas en el sentido de que, a efectos de usuario, el único proceso realizado era el de visualización de sus contenidos (escritos en lenguaje HTML) por parte del explorador del cliente. Las páginas estáticas forman la base necesaria para la presentación de datos en muchos tipos de situaciones. También influye decisivamente la sencillez con que se pueden crear, instalar y mantener. Uno de los lenguajes que presenta páginas Web estáticas es HTML, cuyas características se presentan a continuación.

2.3.1 HTML. HTML es el lenguaje que se utiliza para crear páginas Web. Consta de un conjunto de elementos, denominados etiquetas o marcas, que se utilizan para incluir texto, dibujos, tablas, cabeceras, etc.

Las marcas que se introducen en un documento HTML no son visibles cuando se muestran en un navegador (explorador). Cuando un usuario solicita una página HTML a un servidor Web, este envía la página tal cual (incluidas las etiquetas y los atributos). En el momento en que el explorador recibe la página, interpreta las etiquetas que ésta contiene junto con los atributos de las mismas, mostrando al usuario el resultado final (sin etiquetas).

2.4 PÁGINAS WEB DINÁMICAS EN EL CLIENTE

Una página Web se considera dinámica en el cliente cuando está diseñada para que interprete o ejecute código relacionado con la página en el equipo del usuario.

Las ventajas que presenta este tipo de páginas respecto a las páginas dinámicas en el servidor se evidencian en que:

- Liberan al servidor de una gran carga de cómputo.
- En general, disminuyen la cantidad de ancho de banda empleado en comunicaciones.
- Normalmente ofrecen respuestas inmediatas al usuario.
- Permiten la validación interactiva de datos en formularios.
- Permiten utilizar recursos locales como el ratón, CPU, memoria, etc.

Por otra parte, los inconvenientes que presentan son:

- A menudo son dependientes del navegador (por ejemplo, Netscape no interpreta Visual Basic Script). Incluso existen diferencias en las versiones de cada lenguaje que soportan los diferentes navegadores.
- En ocasiones son poco seguras (por ejemplo, utilizando aplicaciones ActiveX de forma indiscriminada).
- A veces los navegadores no se pueden actualizar con facilidad para cubrir nuevas tecnologías (por ejemplo, al utilizar algunos modelos de Web TV).

- No pueden resolver situaciones en las que la propia naturaleza del problema es centralizada (por ejemplo, consultar en una base de datos centralizada sin acudir a ejecuciones en servidor).

Entre los lenguajes que presentan páginas Web dinámicas en el cliente son los que se enuncian a continuación.

2.4.1 Scripts. Los scripts de una página Web permiten la ejecución de código habitualmente asociado a eventos. Estos eventos se generan la mayoría de las veces a partir de las acciones que realiza el usuario con la página (pulsar en un botón, seleccionar una casilla de verificación, entrar en el ámbito de una caja de texto, etc.) Entre los lenguajes de guiones (scripts) se encuentra JavaScript.

Las aplicaciones más habituales de los lenguajes de guiones son:

- Validar datos en el cliente y comprobar la consistencia de los valores antes de mandar un formulario (como, por ejemplo, comprobar que una fecha tiene un rango adecuado y un formato correcto).
- Actualizar campos relacionados en formularios (por ejemplo, establecer las opciones de una lista desplegable en función del valor seleccionado en unos botones de radio).
- Realizar procesamientos que no requieren la utilización de información centralizada (por ejemplo, convertir pesetas en euros, visualizar el calendario del mes actual, etc.)
- Servir de base para la utilización de otras tecnologías (HTML Dinámico, XML, ActiveX).

2.4.2 HTML Dinámico. Este lenguaje permite hacer que los objetos aparezcan y desaparezcan, que se les asocie posiciones absolutas en pantalla (y, por tanto, la capacidad de movimiento variando las posiciones), que varíen su tamaño (cambiando la altura y el ancho), que unos aparezcan por delante de otros cuando esté solapados (empleando capas), etc.

HTML Dinámico también presenta ventajas estáticas de estructuración de la información a través de CSS (Cascade Style Sheets) que pueden ser usadas sin emplear lenguajes de guiones.

2.4.3 ActiveX. Esta tecnología permite ejecutar código máquina (creados por los programadores de aplicaciones Web) en los equipos de los usuarios. Es decir, se puede vincular como objeto en una página Web un programa que siga el estándar ActiveX y ejecutarlo en los equipos de los clientes que carguen la página.

La aplicación se puede desarrollar con los entornos de programación habituales de Microsoft (Visual Basic, C++, etc.) o con los de otras compañías que permitan generar objetos ActiveX. Un ejemplo que utilice esta tecnología es una aplicación Web que revise, compacte y repare las unidades de disco del cliente, compruebe las versiones de sus aplicaciones gratuitas y sugiera su actualización, proporcione utilidades de impresión y de comunicaciones, etc.

Los programas ActiveX pueden ser cargados con el Explorer de Microsoft (o Netscape modificado) y ejecutados sobre los sistemas operativos y arquitecturas hardware para los que han sido compilados. Fuera de estas condiciones, en general, los programas ActiveX no funcionarán debido a que el código habrá sido generado para un sistema operativo específico, un procesador concreto y una arquitectura fijada en cuanto a direcciones de E/S, etc.

2.4.4 Applets de Java. Un applet de Java es una aplicación Java que presenta unas ciertas restricciones y que puede ser vinculada a una página Web y enviada por Internet desde un servidor a un cliente.

La principal ventaja de los applets de Java reside en su sistema de seguridad, que permite ejecutar aplicaciones en los equipos de los clientes de forma segura. El inconveniente viene de la mano de la necesidad de mayores recursos en los equipos de los clientes, así como de un sistema de ejecución más lento basado en la interpretación de código intermedio.

2.5 PÁGINAS WEB DINÁMICAS EN EL SERVIDOR

Se caracterizan porque los cálculos asociados a las páginas se realizan en el equipo servidor. De esta manera, los visualizadores de los clientes reciben únicamente instrucciones HTML. Por tanto, la complejidad del sistema se traslada a los equipos servidores, liberando de carga a los equipos de los usuarios.

Las principales situaciones en las que se este mecanismo resulta adecuado son:

- Cuando se necesita acceder a información centralizada (habitualmente situada en bases de datos localizadas en el equipo servidor).
- Cuando los procesos que se ejecutan, por razones de seguridad, no se traspasan a los equipos de los usuarios (por ejemplo, la validación de fondos en una tarjeta bancaria).
- Cuando no se desea que los usuarios posean el código asociado a la página Web (por ejemplo, si un estudio de ingenieros de caminos realiza una página Web que resuelve, previo pago, problemas de cálculos de estructuras).
- Cuando se desea liberar de cálculos a los equipos de los clientes (por ejemplo, un sitio Web que genera animaciones tridimensionales y posee equipos especializados para este propósito).
- Cuando se desea estar seguro de que todos los clientes van a poder ver correctamente nuestras páginas Web, independientemente del tipo y versión de su visualizador y del equipo que posean.

2.5.1 Hypertext Preprocessor (PHP). PHP es un lenguaje de scripting Web de código fuente abierto. Es el acrónimo de Personal Home Page cuando fue creado en 1994 por Rasmus Lerdorf para hacer un seguimiento de los visitantes de su currículum online. Como su utilidad y sus capacidades empezaron a crecer (a la vez que empezaba a ser utilizado en más situaciones profesionales), su significado cambió por el de PHP: Hypertext Preprocessor (PHP: procesador de Hipertexto). Básicamente esta definición viene a significar que PHP manipula los datos antes de que estos se conviertan en HTML, el lenguaje de marcado de Hipertexto. Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

PHP es un lenguaje de Scripting del lado del servidor, de plataforma cruzada e incrustado en HTML.

En primer lugar del lado del servidor significa que todo el código PHP se produce en el servidor (en oposición a cuando se refiere al lado del cliente que es el ordenador con el que se visualiza el sitio Web). El Servidor es simplemente el ordenador que alberga las páginas que se ven cuando se acude a una dirección desde un navegador.

Cuando se declara que es de plataforma cruzada se está indicando que se puede utilizar en ordenadores que ejecutan Unix, Windows, Macintosh, OS/2 y otros sistemas operativos (se está hablando del sistema operativo del servidor y no del cliente). PHP no solo puede funcionar en casi cualquier sistema operativo, sino que, a diferencia de otros lenguajes de programación, puede trasladar su trabajo de una plataforma a otra con muy pocos o ningún cambio en absoluto. Como es lógico, también se podrá desarrollar PHP en cualquier sistema operativo, al igual que sucede con HTML.

Al decir que PHP está incrustado en HTML quiere decir que puede colocar el código PHP en la página HTML. Por último PHP es un lenguaje de scripting, en oposición a un lenguaje de programación. Esto quiere decir que PHP está diseñado para hacer algo después de que se produce un evento: por ejemplo cuando un usuario envía un formulario o acude a una URL (Uniform Resource Locator, Localizador de Recursos Uniforme; el término técnico que se utiliza para referirse a una dirección Web).

Los lenguajes de programación como Java, C y Perl se utilizan para crear aplicaciones únicas, que pueden o no estar vinculadas con la Web. El ejemplo más popular de scripting es JavaScript que normalmente manipula los eventos que se producen en el navegador, aspecto en el que es similar a PHP, aunque se trata de una tecnología del lado del cliente.

Otra forma de referirse a los distintos tipos de lenguajes es utilizar el término interpretado para los lenguajes como PHP y JavaScript que no pueden actuar por si mismos y de compilado para los lenguajes como C y Java que si pueden. La Figura 1 muestra como se interpreta PHP.

Esquema PHP

El código PHP es ejecutado en el servidor a diferencia de otros scripts como JavaScript que se ejecuta en la máquina cliente. PHP permite procesar la información de formularios, generar páginas con contenidos dinámicos o mandar y recibir cookies, entre otras actividades.

Figura 1. Esquema de interpretación de PHP



PHP es soporte para una gran cantidad de base de datos así como el soporte de otros servicios que usen protocolos con IMAP, SNMP, NNTP, POP3, HTTP y derivados. PHP es mejor, más rápido y más fácil de aprender que otras alternativas. Al diseñar sitios Web, las principales alternativas a PHP son: simplemente HTML, los scripts CGI (Common Gateway Interface, Interfaz de Pasarela Común), normalmente, pero no necesariamente escritos en Perl; ASP(Active Server Pages, Páginas de Servidor Activas); y JSP (Java Server Pages, Páginas de servidor de Java). JavaScript no es realmente una alternativa a PHP (o viceversa), al ser JavaScript una tecnología del lado del cliente y al no poder utilizarse para crear páginas HTML de la misma forma a como las crean PHP o CGI.

La ventaja que tiene PHP sobre HTML básico es que este último es un sistema limitado que no permite ninguna flexibilidad o sensibilidad. Los Webmasters saben que HTML por si solo no produce sitios Web atractivos y duraderos. Con este fin, las tecnologías del lado del servidor, como los scripts CGI, han ido ganando mayor popularidad. Estos sistemas

permiten a los diseñadores de páginas Web, crear aplicaciones Web generadas dinámicamente, teniendo en cuenta cualquier elemento que el programador desee.

¿Por qué se debería usar PHP en lugar de CGI, ASP o JSP para crear un sitio Web dinámico?: Porque PHP es más rápido de programar y más rápido de ejecutar que los scripts CGI. En comparación ASP requiere un cierto de conocimiento de VBScript y CGI requiere Perl o C y ambos son dos lenguajes más completos y por consiguiente más difíciles de aprender.

PHP es un lenguaje del lado del Servidor, lo que significa que el código que escribe en PHP reside en un ordenador o host que sirve páginas Web a los navegadores. En resumen PHP crea sobre la marcha una página HTML basándose en los parámetros de la elección y el servidor por tanto no contiene páginas HTML estáticas. Cada acción que haga PHP la hará en el servidor y después enviará la información apropiada al navegador.

Lo más importante para trabajar con PHP (puesto que se trata de un lenguaje de scripting del lado del servidor) es acceder al servidor con PHP. Lo más probable es que el ISP (Internet Service Provider, Proveedor de Servicios de Internet) o host tenga disponible esa opción en los servidores. La otra opción es instalar PHP en el servidor (Apache: www.apache.org + PHP: www.php.net). Por último se necesitará un editor de texto o un editor PHP. También se podrá usar editores con interfaz gráfica, también conocida como WYSIWYG (“What you see is what you get”, que traduce “lo que ve es lo que obtiene”), como Dreamweaver o GoLive. También se necesitará un programa FTP (File Transfer Protocol, protocolo de transferencia de archivos), para enviar el scripts al servidor. Otra opción es hacer telnet al servidor remoto y utilizar un editor para escribir scripts directamente en esta máquina.

2.5.2 Active Server Pages (ASP). Active Server Pages (ASP) es una nueva tecnología creada por Microsoft, destinada a la creación de sitios Web. No se trata de un lenguaje de programación en sí mismo (ya que los ASP se pueden programar en VBScript, JavaScript, PerlScript o en varios otros lenguajes), sino de un marco sobre el que construir aplicaciones basadas en Internet.

La tecnología ASP apareció por primera vez (versión 1.0) con el servidor Internet Information Server 3.0 de Microsoft en Diciembre de 1996. La versión 4.0 de IIS (el Option Pack para NT 4.0) incluye la versión 2.0 de ASP, y la versión 5.0 de IIS, distribuida con Windows 2000, incluye ASP 3.0.

Los predecesores de ASP incluyen CGI y Perl. Las tecnologías de Microsoft predecesoras de ASP incluyen IDC y WebDB. Otras tecnologías que compiten con ASP son ColdFusion (Allaire), JavaServer Pages (Sun Microsystems) y PHP (de libre distribución bajo Open System).

Características:

- ASP es totalmente gratuito para Microsoft Windows NT o Windows 95/98.
- El código ASP se puede mezclar con el código HTML en la misma página (no es necesario compilarlo por separado).
- El código ASP se puede escribir con un simple editor de textos como el Block de notas de Windows o UltraEdit.
- Cómo el código ASP se ejecuta en el servidor, y produce como salida código HTML puro, su resultado es entendible por todos los navegadores existentes.
- Mediante ASP se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tal de que proporcione un driver OLEDB u ODBC.
- ASP permite usar componentes escritos en otros lenguajes (C++, Visual Basic, Delphi), que se pueden llamar desde los guiones ASP.

Principales ventajas que ofrece ASP:

- Permite acceder a bases de datos de una forma sencilla y rápida.
- Las páginas se generan dinámicamente mediante el código de scripts, (guiones).
- El código de script se ejecuta en el servidor, y no se depende del navegador que se emplee.
- Desde una página ASP se pueden ejecutar servidores OLE en el servidor de Web, lo que abre un abanico de nuevas posibilidades sólo accesibles

previamente usando CGI y filtros ISAPI: acceso a base de datos, acceso a ficheros, logging en el sistema, envío de correo, etc.

La tecnología ASP se emplea principalmente para crear aplicaciones interactivas que funcionan en Internet.

Requerimientos para programar ASP:

- Un servidor de Web (Microsoft Personal Web Server para Windows 95/98 o Internet Information Server 3.0/4.0/5.0 para Windows NT 4.0)
- Editor de textos como el Bloc de notas de Windows para escribir los guiones.
- Si se quiere acceder a una base de datos, se puede usar una conexión ODBC, una conexión OLE-DB, o una ruta física. Como plataforma de bases de datos, una simple base de datos hecha en Microsoft Access es suficiente.

2.5.3 Programas CGI. Esta tecnología CGI (Common Gateway Interface) se utiliza para implementar páginas Web activas en el servidor. Los lenguajes Perl y C han sido muy empleados.

El funcionamiento básico de un programa CGI se basa en:

- Lectura de datos provenientes de un formulario situado en una página Web, para ello se emplea el canal estándar de entrada.
- Procesamiento de la información, lo que puede llevar incluido el acceso a bases de datos mediante pasarelas del lenguaje.
- Escritura sobre el canal estándar de salida de las páginas HTML de respuesta. Esto conlleva la introducción de numerosas sentencias de escritura de códigos HTML.

Ventajas:

- La respuesta tecnológica que se planteó a la ineficacia y desperdicio de recursos de las soluciones CGI, fue la creación de aplicaciones ISAPI

(Internet Server API) y la versión de Netscape para sus servidores Web: NSAPI.

- Con ambos métodos (CGI,ISAPI) de realizar aplicaciones que se ejecutan en el servidor, se pueden emplear las facilidades que proporcionan IDC (Internet Database Conector) para el acceso a bases de datos con IIS y ODBC.

Desventajas:

- Resulta difícil mantener las páginas Web que se le devuelven al cliente, cuyas instrucciones HTML se encuentran insertadas en el propio código del programa CGI, mezclándose sus funcionalidades. De esta manera, no se puede hacer uso de editores HTML para depurar las páginas.
- La ejecución del programa CGI es muy ineficiente, debido al proceso de carga del código en memoria que se realiza cada vez que un usuario requiere su ejecución por medio de las páginas Web que lo invoca. La existencia de múltiples clientes simultáneos supone múltiples copias del programa en memoria del servidor (con el coste añadido de todas estas cargas). Este funcionamiento resulta obsoleto, y además, compromete el funcionamiento de las máquinas que soportan servidores Web muy utilizados.
- Una aplicación ISAPI es una DLL de Windows que se ejecuta en el mismo espacio de direcciones que el servidor de Web. Estas aplicaciones cuyo código es reentrante, pueden soportar las peticiones simultáneas de diversos clientes con una sola imagen en memoria. En definitiva, la realización de aplicaciones ISAPI le confiere al servidor un alto grado de eficiencia, lo que aumenta la seguridad en la capacidad del sistema para dar respuesta a peticiones simultáneas de varios usuarios.
- La creación de aplicaciones ISAPI es costosa debido a su complejidad técnica, al tiempo necesario para realizar las compilaciones necesarias con el servidor de Web, a la dificultad para hacer las pruebas en una máquina que esté dando servicios en red, etc.

2.6 ARQUITECTURA DE INFORMACIÓN

2.6.1 Modelo Cliente / Servidor. La arquitectura cliente-servidor permite al usuario en una máquina, llamado el cliente, requerir algún tipo de servicio de una máquina a la que está unido, llamado el servidor, mediante una red como una LAN (Red de Área Local) o una WAN (Red de Área Mundial).

Es el modelo de base utilizado en la mayoría de las redes de computadoras. El término servidor se aplica a cualquier programa que ofrece un servicio y que puede ser accedido a través de la red. Los servidores aceptan peticiones recibidas por medio de la red, realizan el servicio y regresan el resultado al que generó la petición.

Figura 2. Modelo Cliente / Servidor



Si el principal objetivo de una máquina es el de soportar un programa servidor en particular, entonces el término "servidor" puede ser aplicado también a dicha máquina. Un servidor entonces, provee un servicio, es pasivo, es decir, espera a que los clientes le soliciten un servicio. Un servidor comienza su ejecución antes del inicio de cualquier interacción y (usualmente) continúa aceptando peticiones y enviando respuestas indefinidamente.

Por otro lado, un cliente es cualquier programa que efectúa una petición y espera una respuesta; (generalmente) termina después de usar un servidor un número finito de veces. Solicita servicios del servidor y es activo, es decir, es el que inicia las conexiones. Las aplicaciones cliente / servidor en la actualidad requieren un esquema híbrido de tal

manera que puedan procesarse transacciones, diseño de bases de datos, experiencia de comunicaciones, uso de una interfase gráfica de usuario GUI y uso de Internet. Las aplicaciones más avanzadas requieren conocimiento de objetos distribuidos e infraestructura de componentes.

Entre las características de los sistemas cliente / servidor se encuentran:

- Servicio: Un esquema cliente / servidor puede verse como una relación entre procesos corriendo o ejecutándose en máquinas separadas. El servidor es un proveedor de servicios. El cliente es un consumidor de servicios. En esencia, el esquema cliente / servidor provee una clara separación basándonos en la idea del servicio.
- Recursos compartidos: Un servidor puede atender muchos clientes al mismo tiempo y regular el acceso de los mismos a los recursos compartidos.
- Protocolos asimétricos: Existen relaciones muchos-a-uno entre los clientes y un servidor. Los clientes siempre inician el diálogo solicitando el requerimiento a un servicio. Los servidores están siempre pasivos esperando los requerimientos de los clientes.
- Localización transparente: El servidor es un proceso que puede estar en la misma máquina que el cliente o en diferentes máquinas sobre la red. El software cliente / servidor frecuentemente enmascara la localización del servidor para los clientes, redireccionando las llamadas al servicio cuando se hace necesario. Un programa puede ser un cliente, un servidor o ambos.
- Escalabilidad: Los sistemas cliente / servidor deben escalarse tanto horizontal como verticalmente. Horizontalmente significa poder adicionar o retirar estaciones de trabajo con muy bajo impacto sobre el rendimiento; verticalmente significa migrar a estaciones más grandes y rápidas o a sistemas distribuidos sobre la red.

- **Integridad:** El código en el servidor y los datos en él están administrados de forma centralizada, lo cual ofrece integridad y seguridad a los datos. Al mismo tiempo, los clientes son personales e independientes.

Existen diferentes tipos de servidores entre los cuales se encuentran:

1. **Servidor de archivos:** Con un servidor de archivos, un cliente lo que hace es requerimientos de los mismos sobre una red.
2. **Servidor de Bases de Datos:** Con un servidor de bases de datos se pasan requerimientos SQL como mensajes a la base de datos. El resultado de cada comando SQL es retornado sobre la red, el código que procesa el requerimiento SQL y los datos, se encuentran en la misma máquina.
3. **Servidor de transacciones:** Con un servidor de transacciones, se pueden hacer llamados a procedimientos o servicios que se encuentran en el servidor donde se encuentra la base de datos. Se puede afirmar que una transacción existe cuando se agrupan varios requerimientos SQL.
4. **Servidor de aplicaciones Web:** Este modelo consiste en máquinas cliente simples que hablan con servidores Web. Un servidor Web retorna documentos cuando el cliente pregunta por el nombre de los mismos.

Como se mencionó antes, los procesos del servidor pueden residir en una máquina que también actúa como cliente de otro servidor. Además de proporcionar este tipo de recursos, un servidor puede dar acceso a otras redes, actuando como un servidor de comunicaciones que conecta a otros servidores o mainframes o minicomputadoras que actúan como hosts de la red.

2.6.2 Arquitectura de tres capas. La arquitectura de una aplicación es la vista conceptual de la estructura de esta. La arquitectura de tres capas es una arquitectura donde hay una separación física entre el cliente que solicita la información, los programas que la procesan y los datos sobre los que operan. Es decir, se cuenta con una lógica de

presentación, del negocio y unos datos de forma separada. Las “tres capas” se distinguen por ser más manejables, por trabajar en diferentes tipos de plataformas y permitir el balance de las solicitudes del cliente entre varios servidores. La seguridad es fácil de implementar ya que el software de la aplicación esta fuera del cliente, entonces, el coste se reduce notablemente ya que una modificación en una capa no debe afectar a las demás.

2.6.2.1 Capa de Presentación. En esta capa se brinda la interfaz necesaria para presentar información y reunir datos. De la misma forma, se aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios.

La capa de presentación se puede identificar por medio de la interfaz de usuario, o normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. El usuario proporciona el contexto de presentación, por lo general en un navegador como Microsoft Internet Explorer o Netscape, el cual permite ver datos, usando como medio HTML o cualquier otro tipo de aplicación.

2.6.2.2 Capa de Datos. La capa de datos es responsable de almacenar, recuperar y mantener los datos, así como de verificar la integridad de los mismos.

2.6.2.3 Capa de Aplicación o de Negocios. La capa de aplicación es el “puente” entre un usuario, representado en la capa de presentación, y la capa de datos. Los servicios de esta capa responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con ello aplicando procedimientos formales y reglas de negocio a los datos relevantes. Esto aísla al usuario de la interacción directa con la base de datos.

2.7 BASES DE DATOS

Base de Datos es un conjunto exhaustivo no redundante de datos fiables, homogéneos estructurados organizados independientemente de su utilización y su implementación en

máquinas accesibles en tiempo real, y compatibles con usuarios concurrentes con necesidad de información diferente y no predecible en tiempo.

Las bases de datos proporcionan la infraestructura requerida para la Toma de Decisiones y para los Sistemas de Información Estratégicos, ya que estos sistemas requieren la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas.

Una base de datos relacional debe cumplir con las siguientes características:

- **Atomicidad:** La atomicidad de una transacción garantiza que todas sus acciones sean realizadas o ninguna sea ejecutada.
- **Consistencia:** La consistencia garantiza que las reglas que hayan sido declaradas para una transacción sean cumplidas.
- **Aislamiento:** Esto garantiza que las transacciones que se estén realizando en el sistema sean invisibles a todos los usuarios hasta que estas hayan sido declaradas finales. El aislamiento "Isolation" garantiza que los usuarios del sistema no observen estos cambios intermedios hasta que sea finalizada la última acción de actualización.
- **Durabilidad:** La durabilidad de una transacción garantiza que al instante en el que se finaliza la transacción esta perdure a pesar de otras consecuencias, esto es, si el disco duro falla, el sistema aún será capaz de recordar todas la transacciones que han sido realizadas en el sistema.

2.7.1 MySQL. Es un Motor de Base de Datos con todas las características y ventajas de un motor comercial. Su principal objetivo de diseño fue la velocidad. Una característica importante es que consume muy pocos recursos, tanto de CPU como de memoria y soporta conectividad ODBC para ser utilizado con aplicaciones Windows y JDBC 2.0 para Java. Es bastante estable y si lo juntamos con IIS y PHP hace un buen equipo para servir páginas WEB con contenido dinámico.

Para el caso particular de SEMCO, este es el motor de base de datos utilizado.

2.8 PROGRAMACIÓN WEB

2.8.1 PHP. El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc...No es un lenguaje de marcas como podría ser HTML, XML o WML. Es un lenguaje cercano a JavaScript o a C.

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML.

Dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones para Internet.

Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra.

La sintaxis que utiliza, la toma de otros lenguajes muy extendidos como C y Perl.

Para quien conozca las ASP (Active Server Pages) de Microsoft, PHP es muy parecido, sólo que más rápido, gratuito y multiplataforma.

El funcionamiento es el siguiente:

- Se escriben las páginas HTML pero con el código PHP inserto.
- Se guarda la página en el servidor Web.
- Un navegador solicita una página al servidor.
- El servidor interpreta el código PHP.
- El servidor envía el resultado del conjunto de código HTML y el resultado del código PHP que también es HTML

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes para el usuario, al que le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

2.8.2 HTML. (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces (hyperlinks)* que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones multimedia* (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).

2.9 PLATAFORMA SOFTWARE.

2.9.1. Windows 2000 Profesional. Este sistema operativo ofrece diferentes versiones que se pueden adaptar a las necesidades de determinadas empresas, estas y sus clientes sólo almacenarán sus datos confidenciales en un entorno que les garantice la seguridad del mismo. En la actualidad es el sistema operativo existente en los computadores del sistema radial de TELEUIS.

2.10 SEGURIDAD INFORMÁTICA

2.10.1 Seguridad Lógica. Ya que el activo más importante que posee una organización es la información, deben existir técnicas, más allá de la seguridad física, que la aseguren. Estas técnicas las brinda la seguridad lógica.

La seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo.

Los objetivos que se plantean en la seguridad lógica son, entre otros:

1. Restringir el acceso a los programas y archivos.
2. Asegurar que los operadores puedan trabajar sin una supervisión minuciosa y no puedan modificar los programas ni los archivos que no les corresponden.
3. Asegurar que se estén utilizando los datos, archivos y programas correctos en y por el procedimiento correcto.
4. Que la información transmitida sea recibida sólo por el destinatario al cual ha sido enviada y no a otro.
5. Que la información recibida sea la misma que ha sido transmitida.

2.10.1.1 Controles de acceso. Estos controles pueden implementarse en el sistema operativo, sobre los sistemas de aplicación, en bases de datos, o en un paquete de seguridad; constituyen una importante ayuda para proteger al sistema operativo de la red, al sistema de aplicación y demás software utilizado; así mismo, ayudan a mantener la integridad de la información y protegen información confidencial de accesos no autorizados.

A continuación se encontrarán algunos de los controles básicos que deberían establecerse en un sistema:

- **Identificación y autenticación.** Es la primera línea de defensa para la mayoría de los sistemas computarizados, permite prevenir el ingreso de personas no autorizadas. Es la base para la mayor parte de los controles de acceso y para el seguimiento de las actividades de los usuarios.

Se denomina **identificación** al momento en que el usuario se da a conocer en el sistema, y **autenticación** a la verificación que realiza el sistema sobre esta identificación. La seguridad informática se basa, en gran medida, en la efectiva administración de los permisos de acceso a los recursos informáticos, basados en la identificación, autenticación y autorización de accesos. Esta administración abarca:

1. Proceso de solicitud, establecimiento, manejo, seguimiento y cierre de las cuentas de usuarios. Es necesario considerar que la solicitud de habilitación de un

permiso de acceso para un usuario determinado debe provenir de un superior y, de acuerdo con sus requerimientos específicos de acceso, debe generarse el perfil en el sistema de seguridad, en el sistema operativo o en la aplicación según corresponde.

2. Revisiones periódicas sobre la administración de las cuentas y los permisos de acceso establecidos. Las mismas deben realizarse desde el punto de vista del sistema operativo, y aplicación por aplicación, pudiendo ser llevadas a cabo por personal de administración del sistema; siempre sobre la base de que cada usuario dispone del mínimo permiso que requiera de acuerdo con sus funciones.
3. Las revisiones deben orientarse a verificar la adecuación de los permisos de acceso de cada individuo de acuerdo con sus necesidades operativas, la actividad de las cuentas de usuarios o la autorización de cada habilitación de acceso. Para esto, deben analizarse las cuentas en busca de periodos de inactividad o cualquier otro aspecto anormal que permita una redefinición de la necesidad de acceso.
4. Detección de actividades no autorizadas. Además de realizar auditorias o efectuar el seguimiento de los registros de transacciones, existen otras medidas que ayudan a detectar la ocurrencia de actividades no autorizadas. Algunas de ellas se basan en evitar la dependencia hacia personas determinadas, estableciendo la obligatoriedad de tomar vacaciones o efectuando rotaciones periódicas a las funciones asignadas a cada una.
5. Nuevas consideraciones relacionadas con cambios en la asignación de funciones de los empleados. Para implementar la rotación de funciones o en caso de reasignar funciones por ausencias temporales de algunos empleados, es necesario considerar la importancia de mantener actualizados los permisos de acceso.
6. Procedimientos a tener en cuenta en caso de desvinculaciones de personal con la organización, llevadas a cabo en forma amistosa o no. Los despidos del personal de sistemas presentan altos riesgos ya que en general se trata de empleados con capacidad para modificar aplicaciones o la configuración del sistema. No obstante, el personal de otras áreas usuarias de los sistemas también pueden causar daños, por ejemplo, introduciendo información errónea a las aplicaciones intencionalmente.

- **Modalidades de acceso.** Se refiere al modo de acceso que se permite al usuario sobre los recursos y a la información. Esta modalidad puede ser:

1. Lectura: el usuario puede únicamente leer o visualizar la información pero no puede alterarla. Debe considerarse que la información puede ser copiada o impresa.
2. Escritura: este tipo de acceso permite agregar datos, modificar o borrar información.
3. Ejecución: este acceso otorga al usuario el privilegio de ejecutar programas.
4. Borrado: permite al usuario eliminar recursos del sistema.
5. Todas las anteriores.

- **Control de acceso interno.** Palabras claves (passwords). Generalmente se utilizan para realizar la autenticación del usuario y sirven para proteger los datos y aplicaciones. Los controles implementados a través de la utilización de palabras clave resultan de muy bajo costo. Sin embargo, se debe estimular una cultura en cuanto a la utilización de passwords, para evitar que los usuarios las escriban o elijan palabras fácilmente deducibles, con lo que se vea disminuida la utilidad de esta técnica.

- **Listas de control de accesos.** Se refiere a un registro donde se encuentran los nombres de los usuarios que obtuvieron el permiso de acceso a un determinado recurso del sistema, así como la modalidad de acceso permitido.

- **Control de acceso externo.** Dispositivos de control de puertos. Estos dispositivos autorizan el acceso a un puerto determinado y pueden estar físicamente separados o incluidos en otro dispositivo de comunicaciones, como por ejemplo un módem. Se debe partir del principio de cerrar todos los puertos e ir abriéndolos de acuerdo a las necesidades que expresen los usuarios.

- **Firewalls o Puertas de seguridad.** Permiten bloquear o filtrar el acceso entre dos redes, usualmente una privada y otra externa (por ejemplo Internet). Los firewalls

permiten que los usuarios internos se conecten a la red externa al mismo tiempo que previenen la intromisión de atacantes o virus a los sistemas de la organización.

- **Accesos Públicos.** Para los sistemas de información consultados por el público en general, o los utilizados para distribuir o recibir información, deben tenerse en cuenta medidas especiales de seguridad, ya que se incrementa el riesgo y se dificulta la administración. Debe considerarse para estos casos de sistemas públicos que un ataque externo o interno puede acarrear un impacto negativo en la imagen de la organización.
- **Administración.** Una vez establecidos los controles de acceso sobre los sistemas y la aplicación, es necesario realizar una eficiente administración de estas medidas de seguridad lógica, lo que involucra la implementación, pruebas y modificaciones sobre los accesos de los usuarios de sistemas.

La política de seguridad que se desarrolle respecto a la seguridad lógica debe guiar a las decisiones referidas a la determinación de los controles de accesos.

La definición de los permisos de acceso requiere determinar cual será el nivel de seguridad necesario sobre los datos, por lo que es imprescindible clasificar la información, determinando el riesgo que produciría una eventual exposición de la misma a usuarios no autorizados. Una vez clasificados los datos, deberán establecerse las medidas de seguridad para cada uno de los niveles.

2.10.2 Ataques más comunes en la Seguridad Informática. En los primeros tiempos, los ataques involucraban poca sofisticación técnica. Los atacantes internos como, operadores o programadores, utilizaban sus permisos para alterar archivos o registros. Los atacantes externos ingresaban a la red averiguando simplemente una contraseña válida. Al pasar el tiempo, se han desarrollado formas cada vez más sofisticadas de ataque para explotar “agujeros” en el diseño, la configuración y operación de los sistemas. Vale la pena nombrar aquellas falencias que normalmente se presentan en los elementos de un sistema informático:

1. Los protocolos de comunicación utilizados carecen, en su mayoría, de seguridad o esta ha sido implementada, tiempo después de su creación en forma de “parche”.
2. Existen agujeros de seguridad en los sistemas operativos.
3. Existen agujeros de seguridad en las aplicaciones.
4. Existen errores en las configuraciones de los sistemas.
5. Los usuarios carecen de información respecto al tema de la seguridad.

Las consecuencias de los diferentes tipos de ataques que se podrían presentar en un sistema informático pueden clasificarse en:

1. Corrupción de Datos: la información que no contenía defectos pasa a tenerlos.
2. Negación del Servicio: servicios que deberían estar disponibles no lo están.
3. Pérdida: los datos llegan a destinos a los que no deberían llegar.

Para mantener un sistema seguro los administradores deberían tener en cuenta los siguientes aspectos:

1. Mantener las máquinas actualizadas y seguras físicamente.
2. Mantener personal especializado en cuestiones de seguridad.
3. Aunque una máquina no contenga información valiosa, hay que tener en cuenta que puede resultar útil para un atacante
4. Por último, pero quizás lo más importante, la continua capacitación del usuario.

2.10.3 Formas de Protección. Existe un sinnúmero de herramientas que pueden utilizarse con el objeto de aumentar la seguridad en un sistema determinado, aunque la implementación de ninguna representará el 100% de la seguridad deseada; será la suma de algunas de ellas las que permitan afirmar que un sistema es confiable.

2.10.3.1 Administración de la seguridad. Es posible dividir las tareas de administración de seguridad en tres grandes grupos:

- Autenticación: se refiere a establecer las entidades que pueden tener acceso a los recursos de cómputo que ofrecen.
- Autorización: es el hecho de que las entidades autorizadas a tener acceso a los recursos de cómputo, tengan acceso únicamente a las áreas de trabajo sobre las cuales ellas deben tener dominio.
- Auditoria: se refiere a la continua vigilancia de los servicios en producción.

Por lo general, las políticas son el primer paso que da una organización para entrar en un ambiente de seguridad, puesto que reflejan su “voluntad de hacer algo” que permita detener un posible ataque antes de que éste suceda. A continuación se citan algunos de los métodos de protección que pueden emplearse:

1. Sistemas orientados a conexión de red: monitorizan las conexiones que se intentan establecer en una red o equipo en particular, siendo capaces de efectuar una acción sobre la base de métricas como origen y destino de la conexión, servicio solicitado, permisos, etc. Las acciones que pueden emprender suelen ir desde el rechazo de la conexión hasta alerta al administrador.
2. Sistemas de análisis de vulnerabilidades: analizan sistemas en busca de vulnerabilidades conocidas anticipadamente. La “desventaja” de estos sistemas es que pueden ser utilizados tanto por personas autorizadas como por personas que buscan acceso no autorizado al sistema.
3. Sistemas de protección a la integridad de información: sistemas que mediante criptografía tratan de asegurar que no ha habido alteraciones indeseadas en la información que se intenta proteger.
4. Sistemas de protección a la privacidad de la información: herramientas que utilizan criptografía para asegurar que la información sólo sea visible para quien tiene autorización. Su aplicación se realiza principalmente en las comunicaciones entre dos entidades.

En resumen, un modelo de seguridad debe estar formado por múltiples componentes que pueden ser incorporados de manera progresiva, con el fin de obtener un método más efectivo para disuadir el uso no autorizado de sistemas y servicios de red.

2.10.3.2 Gestión de claves “seguras”. Ya que es común la mala elección de contraseñas, es útil conocer algunos aspectos que podrían tenerse en cuenta al momento de escoger una clave:

1. No utilizar contraseñas que sean palabras o nombres (el usuario, miembros de la familia, mascotas, lugares etc.).
2. No usar contraseñas completamente numéricas con algún significado (teléfono, fecha de nacimiento, etc.).
3. Elegir una contraseña que mezcle caracteres alfabéticos (minúsculas y mayúsculas) y numéricos.
4. Deben ser largas, de 8 caracteres o más.
5. Deben ser fáciles de recordar para no verse obligado a escribirlas.

• **Normas para proteger una clave.** La protección de la contraseña recae tanto sobre el administrador como sobre el usuario. Al comprometer una cuenta se puede estar comprometiendo todo el sistema.

Algunos consejos a seguir son:

1. No permitir ninguna cuenta sin contraseña.
2. No mantener las contraseñas por defecto del sistema.
3. Nunca compartir con nadie la contraseña. Si se hace cambiarla inmediatamente.
4. No escribir la contraseña en ningún sitio. Si se escribe no debe identificarse como tal y no debe identificarse al propietario en el mismo lugar.
5. No teclear la contraseña si alguien esta mirando.
6. No enviar la contraseña por correo electrónico ni mencionarla en una conversación.
7. No mantener una contraseña indefinidamente. Cambiarla regularmente.

2.10.3.3 Seguridad en los servicios. Cada servicio ofrecido para comunicación con un servidor, es un posible problema para la seguridad, es claro que lo ideal sería no ofrecer

ninguno, y poseer una máquina completamente aislada del resto; evidentemente esto no es posible en la mayoría de sistemas.

Por lo tanto, ya que es necesaria la conectividad entre equipos, se ha de ofrecer los mínimos servicios necesarios para que todo funcione correctamente; esto choca con las políticas de la mayoría de fabricantes y empresas, que por defecto mantiene la mayoría de servicios abiertos al instalar un equipo nuevo, es responsabilidad del administrador preocuparse de cerrar los que no sean estrictamente necesarios.

2.10.4 Políticas de seguridad. Actualmente la seguridad informática ha tomado gran importancia, dadas las cambiantes condiciones y nuevas plataformas de computación disponibles.

Además la posibilidad de interconectarse a través de redes ha llevado la aparición de serias amenazas a la seguridad y confidencialidad tanto de la información como de los sistemas computarizados. Esto ha llevado a que muchas organizaciones hayan desarrollado documentos guía que tienen como objeto orientar en el uso adecuado de las nuevas tecnológicas.

Cada organización debería definir un plan de seguridad de acuerdo con sus propias características, teniendo en cuenta la detección de necesidades, el análisis de riesgos, la definición de políticas y procedimientos, programas de concientización, planes de contingencia, así como la instalación, configuración y mantenimiento de infraestructura, todo ello para mantener protegida la información sin obstaculizar la operación cotidiana.

Las políticas de seguridad informática surgen como una herramienta organizacional para concientizar a cada uno de los miembros de una organización sobre la importancia, la sensibilidad de la información y los servicios críticos que permiten a la compañía desarrollarse y mantenerse en su sector de negocios.

Una política de seguridad se define como la especificación de los requerimientos para el control de acceso a la información, aplicaciones y servicios de una organización. Estos requerimientos a su vez especifican los recursos que pueden ser accedidos (información,

servicios, aplicaciones), con qué permisos (lectura, modificación, eliminación, transferencia), cuándo y por quién.

No es una descripción técnica de mecanismos de seguridad, ni una expresión legal que involucre sanciones a conductas de los empleados, es más bien una descripción de los que se desea proteger y el por qué de ello.

Crear políticas consiste en especificar las necesidades de control de acceso a la información. En particular, las políticas deberán reflejar los objetivos de la institución con respecto a la importancia relativa de cada recurso a proteger y la manera en que la información contribuye a la misión de cada institución. Por su parte, las normas, procedimientos, prácticas y estándares, deberán acoplar esas necesidades con los recursos técnicos existentes, lo cual incidirá en la modernización de los esquemas técnicos de seguridad.

2.10.4.1 Elementos de una política de seguridad informática. Las políticas de seguridad se deben orientar las decisiones que se toman en relación con la seguridad. Por tanto, requieren una disposición de cada uno de los miembros de la empresa para lograr una visión de lo que se considera importante.

Las políticas de seguridad deben considerar entre otros, los siguientes elementos:

- **Alcance**, incluyendo sistemas, recursos y personal sobre la cual aplica.
- **Objetivos** y descripción clara de los elementos involucrados en su definición.
- **Responsabilidades** de cada uno de los servicios y recursos informáticos a todos los niveles de la organización.
- **Requerimientos** mínimos para configuración de la seguridad de los sistemas que cobija el alcance de la política.
- **Consecuencias** de violaciones y no cumplimiento de las políticas.
- **Obligaciones** de los usuarios con respecto a la información a la que él o ella tiene acceso.

Las políticas de seguridad deben ofrecer explicaciones comprensibles sobre por qué deben tomarse ciertas decisiones, transmitir por qué son importantes éstos u otros

recursos o servicios. De igual forma deben mantener un lenguaje común libre de tecnicismos y términos legales que impidan una comprensión clara de las mismas, sin sacrificar su precisión y formalidad dentro de la empresa.

2.10.4.2 Estructura de las políticas de seguridad. La estructura de este tipo de políticas se define con base en la amplitud de las situaciones que pretenda abarcar una política, la cual puede ser clasificada en los siguientes niveles jerárquicos:

- Política general: Visión de seguridad respecto a todo el servicio de Internet.
- Política específica de un tema: Se orienta a tópicos que tienen un interés específico.
- Política particular a una aplicación: Se enfoca a decisiones tomadas por la administración para proteger aplicaciones o sistemas particulares.

Partiendo de las políticas generales, el grado de detalle y complejidad requeridos aumenta conforme se avanza hacia las políticas particulares. Asimismo, entre más detallada y compleja es una política, se requiere actualizarla con mayor frecuencia, y es más complicado el proceso de su implantación.

Por otro lado, de acuerdo al recurso al que está dirigida una política, ésta puede ser clasificada de la siguiente manera.

- Orientada a los recursos lógicos: Cuando el recurso tiene que ver con las técnicas empleadas para generar, explotar o intercomunicar tanto aplicaciones como los datos asociados a las mismas.
- Orientada a la gestión: Cuando el recurso tiene que ver con aspectos administrativos, de personal o con la misma estructura organizacional de la dependencia.
- Orientada a los recursos físicos: Cuando se trate de bienes materiales como instalaciones y equipos.
- Orientada a la respuesta ante incidentes: Cuando se trate de los recursos empleados durante y después de una contingencia.

2.10.4.3 Aspectos a tener en cuenta para establecer políticas de seguridad. Si bien las características de las políticas de información que se han mencionado hasta el momento, muestran una visión de la importancia su formulación, a continuación se presentan algunos aspectos generales recomendados para la formulación de las mismas.

- Efectuar un análisis de riesgos informáticos, a través del cual se logre valorar los activos de la organización; este análisis permitirá afinar el establecimiento de las políticas de seguridad.
- Involucrar a los responsables de los recursos o servicios, pues ellos poseen la experiencia y son fuente principal para establecer el alcance y las definiciones de las políticas de seguridad.
- Comunicar a todo el personal involucrado en el desarrollo de las políticas, los beneficios y riesgos relacionados con los recursos y bienes, y sus elementos de seguridad.
- Monitorear periódicamente el cumplimiento y la eficacia de las políticas en el hacer de la organización, para realizar actualizaciones oportunas de las mismas.

3. MARCO METODOLOGICO

3.1 FASE DE INICIO

En esta fase es necesario hacer un análisis detallado que permita justificar el desarrollo del proyecto. Aún no es indispensable definir todos los casos de uso, únicamente aquellos que son absolutamente necesarios para soportar el análisis previamente realizado. La primera actividad es definir el **Alcance general del proyecto**. Diferentes a los objetivos específicos de todo el proyecto, tenemos algunos objetivos específicos particulares de esta fase, los cuales son:

- Definición de alcance y límites del proyecto.
- Definición de una posible arquitectura (adecuada a las necesidades de usuario).
- Identificación y solución de las posibles dificultades críticas para la realización del proyecto.

3.1.1 Planeación. Quizá el aspecto primordial de la fase de inicio es lograr el conocimiento de la organización para la cual se realiza el proyecto, en nuestro caso se trata de la familia de emisoras institucionales UIS Stereo y UIS A.M. No obstante, es necesario que también se identifiquen las dificultades a superar para iniciar exitosamente el proyecto.

3.1.1.1 Hitos de la Fase de Inicio. Un hito se define como un punto de referencia que se utiliza para medir el progreso alcanzado durante el proyecto. En este orden de ideas, la fase de inicio tiene su razón de ser en la determinación de la justificación del proyecto. Como se mencionó anteriormente, es necesario analizar la organización o entidad interesada en el proyecto para definir los requisitos iniciales.

Así mismo, se deben cumplir los requisitos planteados con anterioridad:

- Definir el alcance y los límites del proyecto. Las respuestas a las siguientes preguntas nos darán una perspectiva más clara:

- ¿Se posee una clara visión del proyecto?
- ¿Han sido identificados los elementos que eventualmente harán parte del proyecto?
- ¿Existe una sugerencia inicial con respecto a la interfaz de usuario, previa consideración de los requisitos iniciales?
- Definir una arquitectura, de acuerdo con las necesidades de los usuarios. De nuevo, las respuestas a las siguientes preguntas despejarán la perspectiva:
 - ¿Existe una base pedagógica que soporte satisfactoriamente cada uno de los elementos del proyecto?
 - ¿Existe una plataforma tecnológica adecuada para desarrollar el proyecto?
 - Si existe, ¿cuenta esta plataforma con características de tolerancia a fallos?
 - ¿Existen suficientes recursos para desarrollar el proyecto?
 - ¿Es necesaria la adquisición de algún recurso?
 - ¿Qué tan portátil, actualizable y escalable es la plataforma tecnológica?
- Identificar y resolver las eventuales dificultades que son relevantes para la puesta en marcha del proyecto. Una vez más, responder satisfactoriamente las siguientes preguntas servirá para tener una mejor visión de estos aspectos:
 - ¿La puesta en marcha del proyecto se ve significativamente amenazada por ciertas dificultades ya identificadas?
 - ¿Puede el inicio del proyecto verse afectado en su duración y / o puesta en marcha debido a limitaciones tecnológicas conocidas?
 - ¿Se han planteado diversas posibilidades de solución a estas dificultades?

3.1.1.2 Flujos, actividades y productos a entregar en la Fase de Inicio. Para garantizar la viabilidad del proyecto se deben definir los requisitos generales del proyecto y estructurar un modelo del mismo dentro durante la Fase de Inicio. No se trata obviamente de un modelo muy detallado o refinado ya que su utilidad será la definición del ámbito de la arquitectura del tema.

El producto final de esta fase es:

- La definición de un modelo general del sistema.
- La definición de los casos de uso generales.

- Sugerir la posible arquitectura del sistema.
-

3.1.2. Iteración Única: estudio de la viabilidad del proyecto. Este apartado describe las actividades a realizar en esta fase y que luego se ejecutarán en el contexto del proyecto actual.

3.1.2.1 Captura de Requisitos. Esta es una de las tareas más importantes en esta fase. Aquí se definen los requisitos iniciales del sistema. Además, permite que los diferentes participantes del proyecto definan las características generales del sistema, el contexto del mismo, los actores y casos de uso, y los posibles requisitos adicionales.

- **Lista de Características**

En esta fase se presentan las características generales del sistema que han sido propuestas por todos los participantes en el proyecto. Las características se traducen en requisitos y éstos, a su vez, necesitan de su nombre, su descripción, su estado (propuesto – aprobado), sus recursos, su prioridad y su nivel de riesgo. Los requisitos están agrupados por módulos para facilitar su comprensión.

Los módulos generales del sistema son los siguientes:

- | | |
|----------------------------------|------------------------------|
| A. Manejo de Usuarios | B. Consulta de Material |
| C. Modificación de Base de datos | D. Enlaces de profundización |
| E. Envío de mensajes | F. Sugerencia de títulos |
| G. Planilla de Programación | |

La lista de requisitos iniciales del sistema se presenta a continuación: (ver Tabla 1)

Tabla 1. Requisitos Iniciales del Sistema

	Nombre	Descripción	Estado	Recursos	Prioridad	Nivel de Riesgo
A	Manejo de Usuarios	Creación y edición usuarios ³ y de información específica sobre estos.	Aprobado	Computador con acceso a Internet.	Importante	
A1	Creación, actualización y borrado de usuarios	Publicación de una pantalla que permita la creación, actualización y borrado de usuarios.	Aprobado	Computador con acceso a Internet.	Importante	
A2	Creación, actualización y borrado de perfil de usuarios	Publicación de una pantalla que permita la creación, actualización y borrado de perfil de usuarios.	Aprobado	Computador con acceso a Internet.	Importante	
A3	Creación, actualización y borrado de login usuarios	Publicación de una pantalla que permita la creación, actualización y borrado de login usuarios.	Aprobado	Computador con acceso a Internet.	Importante	
B	Consulta de Material ⁴	Publicación de una pantalla que permita conocer la existencia y ubicación física del material sonoro desde diferentes parámetros de búsqueda.	Aprobado	Computador con acceso a Internet.	Importante	

³ Administrador, Operador de Audio, Programador de Espacios Radiales e Invitado.

⁴ Los criterios de búsqueda son: Tipo de Material, Artista, Compositor, Arreglista, Región, Instrumentista, Género, Subgénero.

	Nombre	Descripción	Estado	Recursos	Prioridad	Nivel de Riesgo
C	Modificación de Base de datos	Publicación de una pantalla que permita la creación, actualización y borrado de materiales sonoros existentes. Publicación de una pantalla que permita el préstamo de material sonoro.	Aprobado	Computador con acceso a Internet.	Importante	
D	Enlaces de profundización	Publicación de una pantalla que permita visualizar enlaces a Websites relacionados con la búsqueda realizada.	Sugerido	Computador con acceso a Internet.	Secundario	
E	Pizarra de mensajes	Publicación de un formulario que permita la interacción de los usuarios por medio de mensajes de relevancia para los miembros del equipo de trabajo de las emisoras.	Aprobado	Computador con acceso a Internet.	Importante	
E1	Envío de mensajes	Publicación de un formulario que permita el envío de mensajes entre usuarios del sistema.	Aprobado	Computador con acceso a Internet.	Importante	
E2	Recepción de mensajes	Publicación de una pantalla que permita la recepción de mensajes enviados por otros usuarios del sistema.	Aprobado	Computador con acceso a Internet.	Importante	
F	Sugerencia de títulos	Publicación de un formulario que permita sugerir títulos de material al administrador.	Aprobado	Computador con acceso a Internet.	Importante	

	Nombre	Descripción	Estado	Recursos	Prioridad	Nivel de Riesgo
G	Planilla de Programación	Publicación de un listado que permite verificar, publicar y consultar la planilla de programación de las emisoras.	Sugerido	Computador con acceso a Internet.	Rutinario	

- **Riesgos críticos**

En esta fase se deben identificar los diferentes factores que podrían retrasar o impedir la realización del proyecto. Estos riesgos deben ser, por supuesto, superados para poder garantizar la realización del proyecto.

A continuación se presenta una lista de los riesgos críticos asociados a este proyecto. Esta lista contiene, para cada riesgo, el posible impacto en el desarrollo del proyecto y una posible contingencia.

Tabla 2. Riesgos críticos identificados

Descripción	Impacto	Contingencia
Dificultades para lograr el diseño de una interfaz de usuario.	La interfaz de usuario es un elemento esencial para todas las personas que vean el sistema; por lo tanto, debe estar bien diseñado y soportado conceptualmente.	Conseguir asesoría de diseñadores gráficos. Debido a que la interfaz gráfica no es necesaria para el desarrollo de los módulos, se puede comenzar con una provisional.
No se cuenta con una dirección IP que haga que el servidor sea visible desde ubicaciones externas a las instalaciones de UIS STEREO - UIS AM.	Los usuarios no tendrán acceso al servidor desde fuera de las instalaciones de las emisoras.	Realizar gestiones ante la División de Servicios de Información de la Universidad para lograr que al servidor le sea asignada una dirección IP válida.

- **Contexto del Sistema**

Al definir un concepto del sistema se obtiene una concepción más detallada de cómo se va a desarrollar el proyecto y se puede limitar mejor el sistema. Cuando se intenta comprender el contexto de un sistema se pueden utilizar dos herramientas: **modelado de dominio** y **modelado de negocio**. El modelado de dominio permite crear una estructura de objetos a partir de los conceptos que forma el contexto del sistema. Por otro lado, el modelado de negocio describe los procesos que se llevan a cabo en el sistema, y nos permite establecer actividades y saber quién las lleva a cabo.

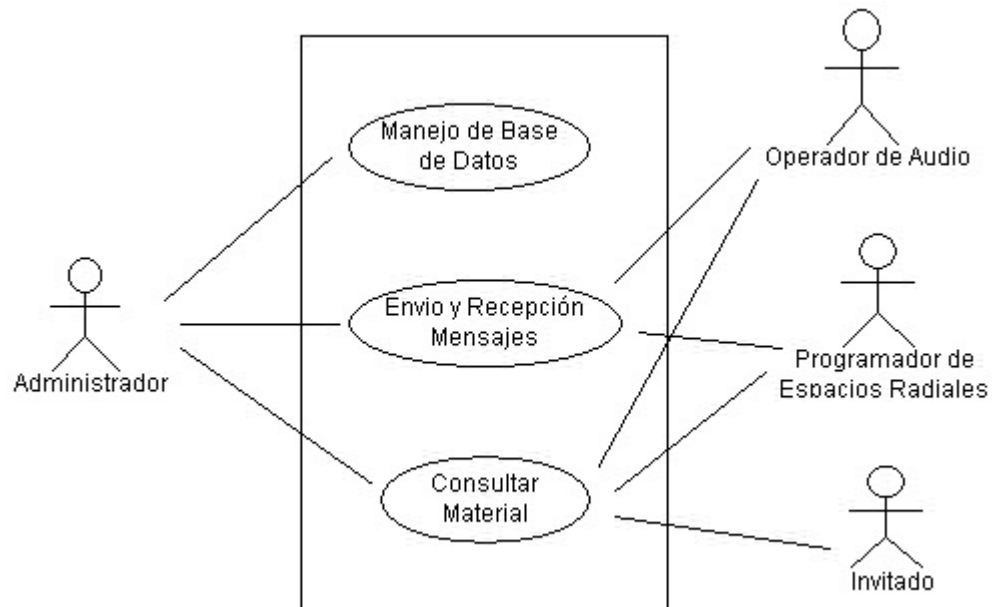
- **Modelo de Negocio**

A continuación se presenta el objetivo general del proyecto “Sistema Intranet para Registro, Búsqueda y Control de Material Sonoro de la Familia de Emisoras Institucionales UIS Stereo Y UIS AM De La Universidad Industrial De Santander – **Semco**”. Posteriormente se desarrollará el modelo de negocio a través de los modelos de casos de uso y de objetos.

El objetivo general del presente proyecto es “Crear un completo Sistema de Información que automatice la búsqueda del material fonográfico bajo diferentes parámetros, el control de préstamos y entregas, las sugerencias sobre la adquisición de nuevos títulos y permita el acceso a información adicional alusiva a dicha búsqueda.” A partir de los objetivos del proyecto, se han generado los siguientes casos de uso que van a definir el contexto del sistema.

- ◆ **Modelo de casos de uso** (Ver Figura 3)

Figura 3. Modelo general de casos de uso



Las siguientes tablas contienen una descripción de los casos de uso del negocio:

Tabla 3. Proceso de negocio “Manejo de Base de Datos”

Proceso de Negocio	Manejo de Base de Datos
Objetivo	Manipular la base de datos que contiene la información que utiliza el sistema, de manera que los datos allí contenidos siempre sean reales y confiables.
Descripción	El administrador decide qué información debe ser actualizada y procede a llenar los formularios disponibles para dicha operación.
Posibilidades	Facilidad de manejo de la base de datos sin tener formación técnica, ahorrando así la contratación de personal externo a la organización.
Tiempo de Ejecución	Depende de la transacción a realizar en la base de datos.
Costo de Ejecución	Costo de acceso (Tiempo de Internet)

Tabla 4. Proceso de negocio “Envío y recepción de Mensajes”

Proceso de Negocio	Envío y recepción de Mensajes
Objetivo	Establecer un canal de comunicación dedicado exclusivamente a temas que son relevantes para los miembros del equipo de trabajo de la emisora.
Descripción	El administrador regula el contenido de los mensajes y, al igual que los demás usuarios (excepto el INVITADO), puede publicarlos y leerlos. Los Operadores de Audio y Programadores de Espacios Radiales deben: (1) registrarse, (2) ver los últimos mensajes publicados y decidir si responden a éste, o (3) empiezan un tema nuevo.
Posibilidades	Conocer lo que otras personas de la organización comentan sobre aspectos propios de la misma, de manera que se puede participar haciendo aportes constructivos.
Tiempo de Ejecución	Depende del contenido del mensaje, o de lo que desee realizar una vez en el sistema de mensajes.
Costo de Ejecución	Costo de acceso (Tiempo de Internet)

Tabla 5. Proceso de negocio “Consultar Material”

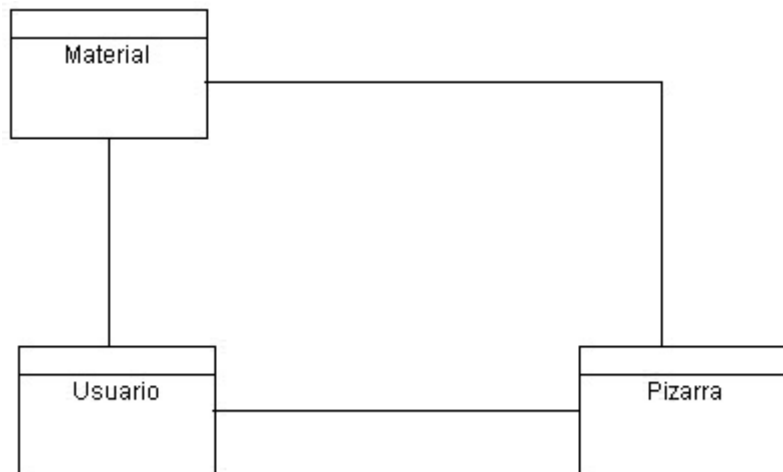
Proceso de Negocio	Consultar Material
Objetivo	Conocer con qué material sonoro se cuenta en las emisoras para la producción de los diferentes espacios radiales. Así mismo, se desea conocer detalles importantes acerca de dicho material, en caso de que existan.
Descripción	Cualquiera de los usuarios del sistema elige el criterio de búsqueda dentro de los ofrecidos por el sistema y escribe algunas palabras claves según el criterio seleccionado; el sistema le indica como primera información el criterio de búsqueda y después permite que el usuario conozca información más detallada, si la hubiere, acerca del material en cuestión.
Posibilidades	Apoyar los procesos básicos de producción de espacios radiales en las emisoras, agilizando la búsqueda de la herramienta fundamental

	de trabajo: el material sonoro.
Tiempo de Ejecución	Depende de la búsqueda a realizar.
Costo de Ejecución	Costo de acceso (Tiempo de Internet)

- **Modelo de Dominio o de Objetos**

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen, o los eventos que suceden en el entorno en el que trabaja el sistema. El modelo de objetos permite observar la interacción entre ellos, a través de las actividades reflejadas en el diagrama de casos de uso del negocio. (Ver Figura 4)

Figura 4. Modelo de Objetos general



- **Actores y Casos de Uso**

A partir del modelo de negocio es posible obtener de manera casi directa la colección inicial de casos de uso del sistema. De esta manera, el modelo del negocio permite obtener los casos de uso más importantes dentro de cada proceso del negocio, y además facilita la determinación del conjunto adecuado de pasos incrementales en el proceso iterativo de desarrollo.

▪ **Actores**

Definir los actores es un paso esencial para obtener de manera correcta los requisitos. Para obtener los actores, se debe partir del modelo del negocio y observar los siguientes pasos:

- Identificar al menos a un usuario que pueda representar a un actor candidato. De esta manera se obtendrán los actores relevantes.
- Encontrar una coincidencia mínima entre los roles que desempeñan las instancias de los diferentes casos de uso en relación con el sistema, con el fin de evitar que dos o más actores sean, en esencia, los mismos. De ser así, se debe crear un actor generalizado que tenga asignados los roles comunes a los actores que se superponen.
- Describir brevemente las funciones de cada actor e indicar para qué utiliza el sistema.

En este proyecto se identificaron los actores que describe la Tabla 6.

Tabla 6. Actores del Sistema

Actor	Descripción	Responsabilidades (papel que juega)	Necesidades (para qué utiliza el sistema)
Administrador	Representa a la(s) persona(s) encargada(s) de la administración del sistema SEMCO 1.0, es decir, del mantenimiento de la información del sistema.	<ul style="list-style-type: none"> • Mantenimiento de usuarios. • Mantenimiento de la base de datos. • Recepción de sugerencias acerca de material a adquirir. • Realizar búsquedas de material sonoro en SEMCO 1.0. • Gestionar la materialización de tales sugerencias. • Verificar y publicar la programación de las emisoras. • Moderar el uso de la pizarra. 	Utiliza el sistema para la creación, modificación y eliminación de usuarios y materiales sonoros. Revisa las sugerencias hechas por los usuarios del sistema a raíz de las búsquedas no exitosas en el mismo. Verifica y publica la planilla de programación de las emisoras. Aprueba o desaprueba el préstamo

Actor	Descripción	Responsabilidades (papel que juega)	Necesidades (para qué utiliza el sistema)
			<p>de material sonoro a alguno de los usuarios que pueden pedir material en préstamo. Además, utiliza la pizarra para publicar y leer mensajes, así como para moderar su uso. La moderación del uso de la pizarra la realiza mediante la publicación de las reglas de uso, y mediante una verificación de cada mensaje allí publicado, asegurando que el medio de comunicación no sea utilizado para asuntos irrelevantes para el equipo de trabajo de las emisoras.</p>
Operador de Audio	Representa a la persona que se dedica a ejercer la labor de Controlador de Audio en las emisoras.	<ul style="list-style-type: none"> • Revisar y enviar mensajes publicados en la pizarra. • Solicitar en calidad de préstamo y entregar el material sonoro. • Realizar búsquedas de material sonoro en SEMCO 1.0. • Consultar la programación de las emisoras. 	<p>Utiliza el sistema para recibir y enviar mensajes de interés para los miembros del equipo de trabajo.</p> <p>Formula sugerencias acerca de material que debería ser adquirido por las emisoras.</p> <p>Consulta la planilla de programación de las emisoras.</p>

Actor	Descripción	Responsabilidades (papel que juega)	Necesidades (para qué utiliza el sistema)
			<p>Solicita el material sonoro en calidad de préstamo.</p> <p>Realiza búsquedas de material sonoro que necesita para la elaboración de espacios radiales.</p>
Programador de Espacios Radiales	Representa a la persona que conduce, dirige y/o realiza un espacio radial especializado.	<ul style="list-style-type: none"> • Revisar y enviar mensajes publicados en la pizarra. • Solicitar en calidad de préstamo y entregar el material sonoro. • Realizar búsquedas de material sonoro en SEMCO 1.0. • Consulta la programación de las emisoras. 	<p>Utiliza el sistema para recibir y enviar mensajes de interés para los miembros del equipo de trabajo.</p> <p>Consulta la planilla de programación de las emisoras.</p> <p>Solicita el material sonoro en calidad de préstamo.</p> <p>Realiza búsquedas de material sonoro que necesita para la elaboración de espacios radiales.</p>
Invitado	Representa a la persona que simplemente desea consultar información de la emisora, sin que necesariamente sea miembro del equipo de trabajo.	<ul style="list-style-type: none"> • Consulta la programación de las emisoras. • Realizar búsquedas de material sonoro en SEMCO 1.0. 	<p>Consulta la planilla de programación de las emisoras.</p> <p>Realiza búsquedas de material sonoro que necesita para la elaboración de espacios radiales.</p>

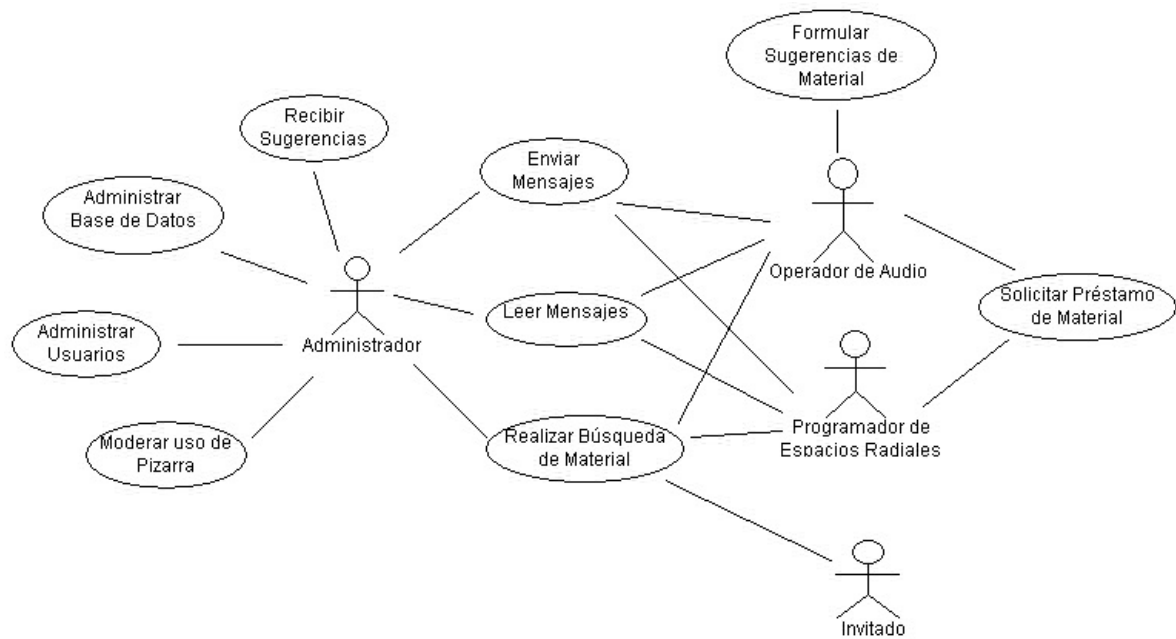
▪ **Casos de Uso**

Los casos de uso son fundamentales para definir una visión global del sistema e identificar los subsistemas que forman parte del mismo.

La descripción de los casos de uso depende de su complejidad; de no poseerla, una descripción breve será suficiente. Los casos de uso se agrupan según el paquete de análisis al que pertenezcan. Cada paquete presenta el modelo de casos de uso y una descripción general de cada caso de uso; sólo se detallan aquellos que se crean convenientes para la comprensión del funcionamiento general del sistema.

◆ **Modelo de casos de uso general**

Figura 5. Modelo de casos de uso general del sistema



Descripción del modelo general de casos de uso
El sistema ofrece la posibilidad a los usuarios de hacer consultas de material sonoro. Se ofrecen varios criterios de búsqueda.
El administrador está encargado del mantenimiento de la base de datos, de los usuarios, y de la moderación del uso de la pizarra. También se encarga de mantener actualizada la planilla de

programación de las emisoras, así como de recibir las sugerencias hechas por el sistema en cuanto a material sonoro no existente y luego es responsable de gestionar su eventual adquisición. Al igual que los demás tipos de usuario, tiene la posibilidad de interactuar con ellos mediante el uso de la pizarra.

El actor “Invitado” fue creado teniendo en cuenta la posibilidad real que existe de que algún otro miembro de TELEUIS que por sus características no pueda ser incluido en alguno de los demás tipos de usuario desee realizar la búsqueda de material existente, cualquiera sea su motivo.

La descripción de un caso de uso depende de su complejidad. Si el caso de uso es sencillo una descripción breve será suficiente. En caso contrario, es necesario proveer una explicación más exhaustiva.

◆ Administración de Usuarios

La siguiente figura presenta en detalle los casos de uso “Administrar usuarios”, “Administrar Material” y “Administrar Pizarra”. (Ver Figura 6)

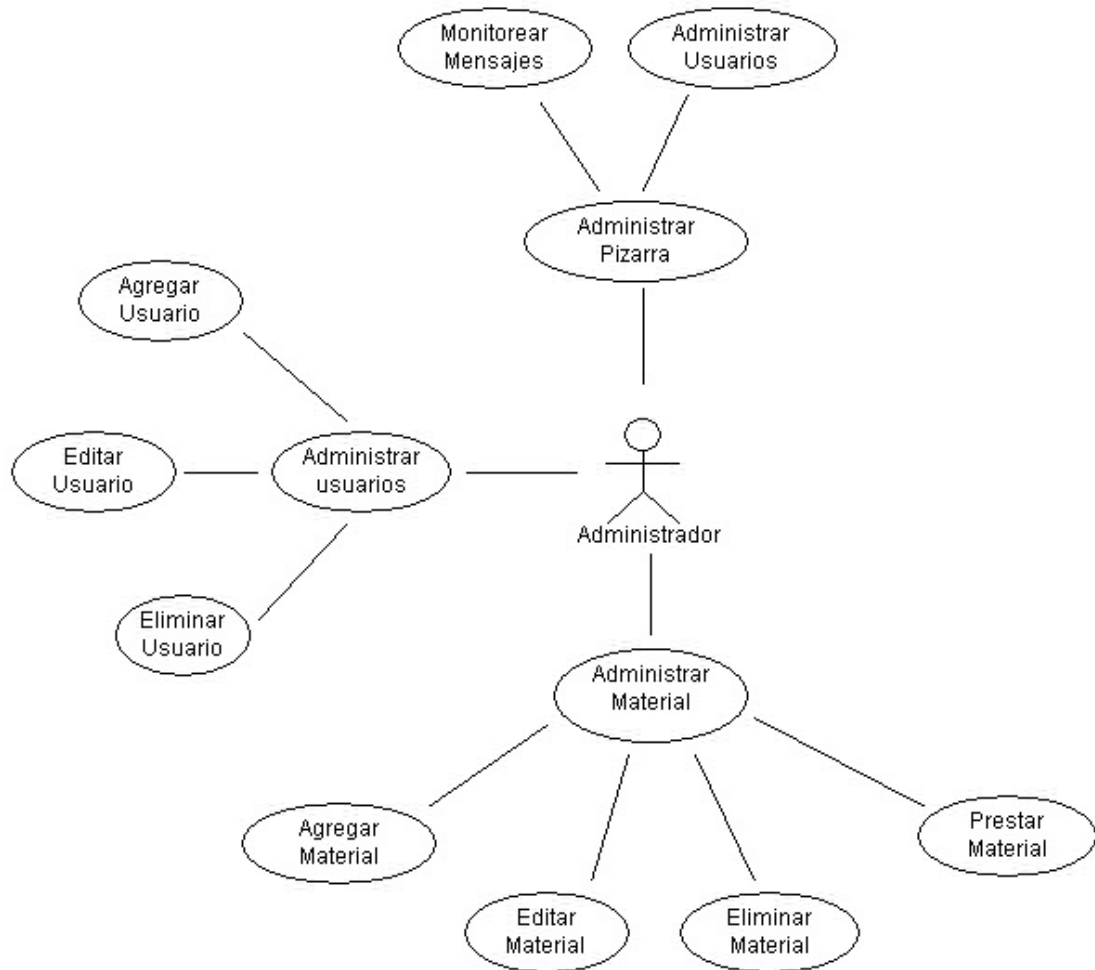
Descripción del módulo de administración de la pizarra
<p>El sistema ofrece al administrador la posibilidad de otorgar permisos a ciertos usuarios para que utilicen la pizarra, así como de administrar el contenido de los mensajes que allí se publican.</p> <p>Para la administración del contenido de los mensajes el administrador cuenta con una interfaz de usuario intuitiva y fácil de utilizar. El contenido se controla convencionalmente, es decir, leyendo los mensajes publicados y estimando la relevancia de los mismos para el equipo de trabajo de la organización. El administrador está en capacidad de leer cualquier mensaje y cualquier respuesta a éste por parte de otro usuario, así como también de eliminar estos mensajes (junto a sus respuestas, si las hubiere) del sistema.</p>

Casos de Uso Identificados	Breve Descripción
Monitorear mensajes	Monitorear el contenido de los mensajes publicados en la pizarra.
Administrar usuarios	Registrar usuarios de la pizarra.

Descripción del módulo de administración de usuarios
<p>El sistema ofrece al administrador la posibilidad de crear, modificar y eliminar usuarios.</p> <p>Para llevar a cabo esta actividad, el administrador cuenta con una interfaz intuitiva y fácil de utilizar. El administrador decide qué derechos le otorga a un nuevo usuario, según su posición dentro de la organización, así como también está en capacidad de asignar más o menos derechos de uso a un usuario ya registrado.</p>

Casos de Uso Identificados	Breve Descripción
Agregar usuario	Crear usuarios en el sistema.
Editar usuario	Editar usuarios en el sistema.
Eliminar usuario	Eliminar usuarios del sistema.

Figura 6. Casos de uso para la administración de usuarios, base de datos y pizarra.



Descripción del módulo de administración de material

El sistema ofrece al administrador la posibilidad de crear, modificar y eliminar material, así como administrar el préstamo del mismo.

Para el préstamo de material, el administrador está en capacidad de registrar quién toma prestado el material, cuándo lo toma prestado y cuándo debe ser devuelto. Esta información queda almacenada en la base de datos.

Casos de Uso Identificados	Breve Descripción
Agregar Material	Crear material en el sistema.
Editar Material	Editar material en el sistema.
Eliminar Material	Eliminar material del sistema.
Prestar Material	Registrar el préstamo de material en el sistema.

- **Requisitos adicionales**

Los requisitos adicionales son aquellos que no pueden ser asociados con ninguno de los casos de uso, pero que, de una u otra manera, tienen impacto sobre ellos. Estos requisitos adicionales se presentan en la siguiente tabla:

Tabla 7. Requisitos adicionales identificados en la fase de inicio

Requisito	Descripción
Requisitos de interfaz de usuario	Se deberá contar con una interfaz que permita acceder rápidamente al sistema. El sistema deberá funcionar de manera similar en la mayoría de los navegadores Web disponibles en el mercado.
Requisitos de Hardware	Clientes: PC Intel mínimo 486.
Requisitos de Red	Permitir visibilidad desde la Red Interna de las Emisoras o, adicionalmente, desde la Intranet de la Universidad o desde Internet.
Restricciones de Diseño	No se utilizarán componentes comerciales en su desarrollo. Se deberá separar la capa de presentación de los datos de la lógica del sistema.
Restricciones de Implementación	Software del servidor: Sistema Operativo Windows 2000. Servidor Web: Internet Information Server. Software cliente: Netscape 4.0 o superior; Internet Explorer 4.0 o superior.

3.1.2.2 Análisis. El análisis nos permite examinar los objetos y los casos de uso planteados, de manera que formen un modelo inicial de análisis. Es modelo se refinará en las siguientes fases. El objetivo del modelo inicial es definir de manera más concisa los casos de uso y plantear la arquitectura candidata.

- **Análisis de la arquitectura**

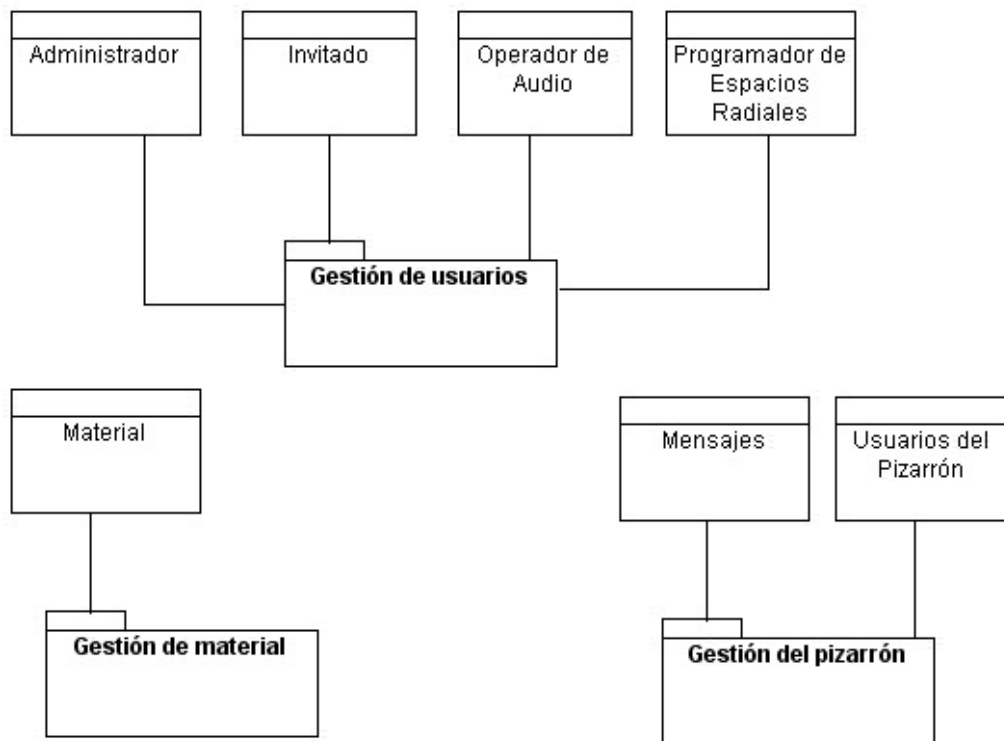
Se va a desarrollar un modelo inicial del sistema. Para ello se definen los paquetes de análisis y las entidades obtenidas a partir del modelo de negocio.

- **Paquetes de análisis general**

Los paquetes de análisis permiten organizar los elementos del modelo de análisis de manera que sean piezas manejables. Estos paquetes se crean a partir del análisis de los requisitos y del modelo de negocio. Para este fin, se asignan casos de uso a cada paquete teniendo en cuenta los siguientes criterios:

- Agrupar los casos de uso que dan soporte a un determinado proceso de negocio.
- Agrupar los casos de uso que dan soporte a un determinado actor del sistema.
- Agrupar los casos de uso que tengan relaciones de generalización o extensión.

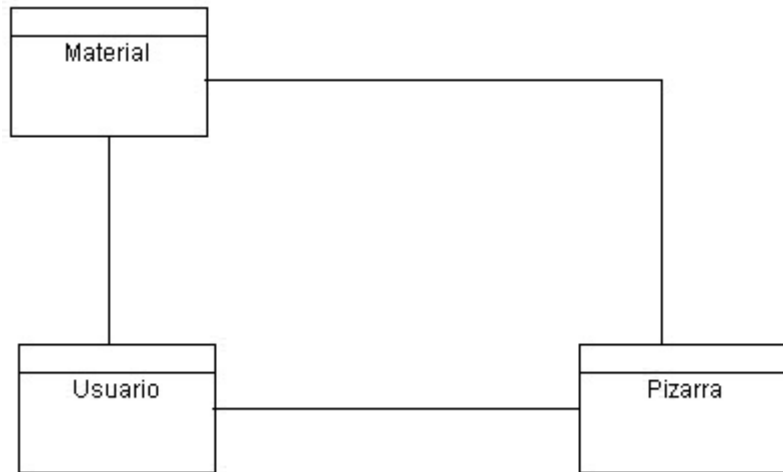
Figura 7. Identificación de paquetes a partir de las clases de dominio.



- **Clases de entidad derivadas del modelo de dominio**

Las clases descritas en la Figura 8 se derivaron del modelo de dominio. Estas se convertirán en entidades en el modelo *entidad – relación* del modelo de diseño y luego se convertirán en tablas de la base de datos en la fase de implementación.

Figura 8. Clases derivadas a partir del modelo de dominio.



3.1.2.3 Diseño. El objeto del diseño es definir la arquitectura candidata. El diseño de clases e interfaces se delega a fases posteriores.

- **Diseño de la arquitectura**

A continuación se realizará un esbozo de la arquitectura de red a utilizar, así como la identificación de los subsistemas.

- **Configuración de red**

El sistema funcionará utilizando una arquitectura de tres capas, la cual permite separar la lógica de negocio de la base de datos y de la lógica de presentación.

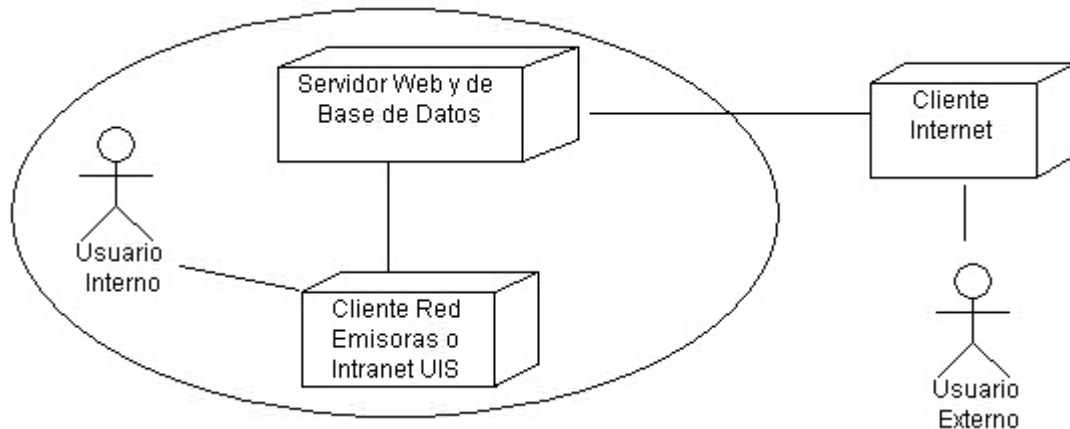
El sistema se ejecutará sobre un servidor y uno o varios clientes. La lógica de negocios y la funcionalidad de la base de datos se ejecutarán en el servidor, mientras que la capa de presentación de la interfaz se ejecutará en el cliente.

La conexión entre el servidor y el cliente se dará utilizando el protocolo HTTP⁵, ya sea desde la red interna de las Emisoras, de la Universidad, o desde Internet.

⁵ “Hypertext Transfer Protocol” o “Protocolo de Transferencia de Hipertexto”.

La siguiente figura muestra el diagrama de despliegue del sistema.

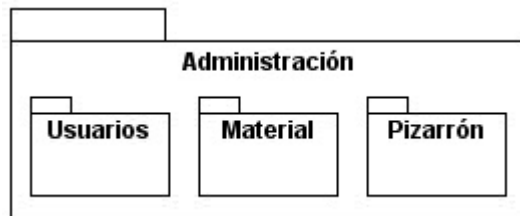
Figura 9. Diagrama de despliegue del sistema



- **Identificación de los subsistemas**

Los subsistemas constituyen una forma de organizar el modelo de diseño en porciones manejables, identificables a partir de los requerimientos iniciales, para dividir el trabajo de diseño. La Figura 10 muestra los subsistemas identificados.

Figura 10. Subsistemas identificados a partir de los requerimientos del sistema



3.1.3 Evaluación. Al terminar cada fase es necesario realizar una evaluación para verificar si se han alcanzado los hitos propuestos y si es posible proseguir. En la fase de Inicio la evaluación está orientada a decidir si se continua con el proyecto o no.

3.1.3.1 Desarrollo de los modelos de casos de uso. Debido a que los casos de uso son cruciales para el desarrollo del proyecto, a continuación se presenta una tabla de los casos de uso identificados en esta fase.

La tabla de casos de uso presenta el nombre de cada caso de uso y el grado de conocimiento que de cada uno se tiene.

El grado de conocimiento se define teniendo en cuenta los siguientes parámetros:

- **Identificado:** Si se ha determinado la conveniencia de crear este caso de uso según los criterios presentados para encontrar casos de uso relevantes.
- **Descrito:** Si se ha realizado una descripción breve del caso de uso.
- **Analizado:** Si se ha realizado una descripción paso a paso del funcionamiento del caso de uso. Este análisis también puede involucrar creación de diagramas de análisis, según la complejidad del caso de uso.
- **Diseñado e Implementado:** estos dos estados han sido agrupados dado que, en algunas situaciones, antes de llevarse a cabo la implementación de un caso de uso se realiza su diseño a través de diagramas de secuencia. En los demás casos, con el análisis de los casos de uso es suficiente para realizar su implementación.

Tabla 8. Casos de uso identificados en la Fase de Inicio

Casos de uso identificados	Identificado	Descrito	Analizado	Diseñado e Implementado	¿Se comprende completamente?
Monitorear mensajes	F. Inicio	F. Inicio			No
Administrar usuarios	F. Inicio	F. Inicio			No
Agregar usuario	F. Inicio	F. Inicio			No
Editar usuario	F. Inicio	F. Inicio			No
Eliminar usuario	F. Inicio	F. Inicio			No
Agregar Material	F. Inicio	F. Inicio			No
Editar Material	F. Inicio	F. Inicio			No
Eliminar Material	F. Inicio	F. Inicio			No
Prestar Material	F. Inicio	F. Inicio			No

3.1.3.2 Vista de la Arquitectura. Al describir la arquitectura se obtiene una mayor comprensión del sistema, se organiza el desarrollo y se fomenta la reutilización. Por lo tanto, se debe construir una arquitectura que permita implementar los casos de uso del sistema de una forma económica; además, se deben tener en cuenta otros factores como el Sistema Operativo y de Bases de Datos a escoger, los productos para el desarrollo del sistema y los requisitos no funcionales del sistema.

Sin embargo, en la fase de inicio solo se busca una arquitectura candidata, la cual está soportada en los casos de uso más relevantes para la viabilidad del proyecto. En la Tabla 9 se presenta un sumario de los flujos desarrollados en esta fase.

Tabla 9. Flujos desarrollados en la fase de Inicio

Flujo	Sección	Vista	Comentario
Captura de requisitos	Casos de uso	- Modelo de casos de uso general - Descripción del modelo general de casos de uso	Presenta los actores y casos de uso más importantes del sistema.
Análisis	Paquetes de Análisis Generales	- Identificación de paquetes de análisis a partir de las clases de dominio. - Identificación de los paquetes de análisis a partir de los casos de uso.	Principales paquetes de análisis encontrados.
Diseño	Diseño de la arquitectura	Subsistemas a partir de los paquetes de análisis.	Subsistemas más importantes para la arquitectura pertenecientes al modelo de diseño.

3.1.3.3 Cumplimiento de los criterios. El desarrollo de la fase de inicio estuvo orientado a determinar el alcance y la viabilidad del proyecto. Se realizó un estudio preliminar del negocio a través de los casos de uso y de objetos.

Teniendo en cuenta este estudio se determinó que el alcance del sistema es brindar un espacio a los miembros del equipo de trabajo de las emisoras institucionales UIS AM y UIS STEREO para facilitar sus labores diarias utilizando una herramienta software que realice automáticamente varias tareas que, hasta el momento, han tenido que ser realizadas manualmente. Así mismo, esta herramienta software creará un canal de

comunicación electrónico para tratar temas de relevancia para la realización de sus labores dentro de la organización.

El sistema tendrá un módulo de *Consulta de Administración de Material Fonográfico y Usuarios del sistema*, un módulo de *Control de Préstamo de Material*, un módulo de *Gestión Automática de Material*, un módulo para manejo de *Información Adicional* y, por último, un módulo de *Pizarra*, o medio de comunicación entre los miembros de la organización que son usuarios del sistema.

Aunque en esta fase no se definió totalmente la arquitectura del sistema, se cuenta con un bosquejo sobre el cual se puede seguir edificando y analizando para agregar o reorganizar otras funciones o módulos.

Se determinó también que el sistema debe estar en un ambiente Web para facilitar el acceso desde cualquier cliente, en cualquier plataforma y a un costo muy bajo o inexistente.

Así mismo, se definieron dificultades que puedan surgir en la realización del proyecto, así como estrategias para poder superarlas.

3.2 FASE DE ELABORACION

En esta fase se construye una arquitectura estable para guiar el sistema a lo largo de su vida futura. También se estudia el sistema propuesto al punto de planificar la fase de construcción con gran precisión. Los objetivos de esta fase son:

- Establecer una sólida base de la arquitectura para guiar el trabajo durante la fase de construcción.
- Continuar la observación y el control de los riesgos críticos que aún queden.
- Identificar los nuevos riesgos que puedan aparecer.
- Recopilar los casos de uso para aproximadamente el 80% de los requisitos funcionales del sistema.

3.2.1 Planeación. Al iniciar la fase de elaboración se cuenta con un modelo de casos de uso parcialmente completo y una descripción de la arquitectura candidata. También se cuenta con unos bocetos de un modelo de análisis y de un modelo de diseño. Estos bocetos servirán de punto de partida para iniciar la búsqueda de una arquitectura estable que sea consistente durante la construcción del sistema.

3.2.1.1 Hitos. El punto de referencia para medir el progreso en esta fase está dado con el cumplimiento de los objetivos planteados anteriormente:

- Extender los requisitos mediante la respuesta a preguntas como:
 - o ¿Han sido identificados los requisitos, actores y casos de uso necesarios para diseñar la línea base de la arquitectura?
 - o ¿Han sido estos detallados suficientemente como para lograr los objetivos de esta fase?
- Definir la línea base de la arquitectura mediante el análisis y la búsqueda de respuestas a:
 - o ¿Satisface la línea base de la arquitectura no sólo a los requisitos recopilados formalmente hasta ahora sino también las necesidades de todos los usuarios?
 - o ¿Es la arquitectura base lo suficientemente robusta como para resistir la fase de construcción y la adición de características que puedan ser necesarias en posteriores versiones del sistema?
- Mitigar los riesgos significativos, lo que se logrará encontrando respuestas a las preguntas:
 - o ¿Se ha mitigado de forma adecuada los riesgos críticos ya sea eliminándolos o preparando un plan de emergencia?
 - o ¿Han sido identificados todos los riesgos significativos?

3.2.1.2 Flujos, actividades y productos a entregar en la Fase de Elaboración. Dado que el fin de la presente fase es determinar una arquitectura estable que permita el desarrollo posterior en la fase de construcción, se deben obtener nuevos casos de uso y detallar los necesarios para comprender la arquitectura del sistema.

Además, se debe realizar un análisis de los paquetes que lo componen para, posteriormente, realizar el diseño de los subsistemas.

Al final de ésta fase se debe definir el diseño de la base de datos y la interfaz.

3.2.2 Iteración Única: Determinación de una arquitectura estable. Debido a que en la fase de inicio se describió cómo es el procedimiento para realizar cada una de las actividades de este apartado, sólo se describirán brevemente aquellas que no hayan sido descritas anteriormente. Para el desarrollo de cada actividad y los productos generados sólo se presentará lo necesario para la comprensión de la metodología.

3.2.2.1 Captura de Requisitos. En este flujo de trabajo se establecerán la prioridad y estructura de los casos de uso, en especial de aquellos que sean de vital importancia para la definición de la arquitectura del sistema. Para ello, se determinarán la mayoría de los casos de uso (alrededor del 80%) para garantizar que la arquitectura definida cumpla con todos los requisitos exigidos.

- **Actores y Casos de Uso**

Para esta actividad se deben identificar los casos de uso adicionales a los presentados en la fase de inicio.

- **Casos de uso**

En la fase de inicio se definió un modelo de casos de uso general. Este modelo fue de utilidad para trazar la arquitectura inicial del sistema. En esta fase, el modelo general será ampliado y analizado de manera detallada para ajustarlo a los requerimientos presentados en la fase de inicio. Para cada modelo de casos de uso se identificarán los casos de uso que no hayan sido identificados anteriormente.

- **Modelo de recepción de sugerencias**

El caso de uso que se muestra en la siguiente figura permite realizar las actividades concernientes a la *recepción y gestión de sugerencias de material*, de manera tal que existan interfaces de usuario de fácil manejo sin tener que acceder directamente la base de datos.

Figura 11. Modelo de recepción de sugerencias

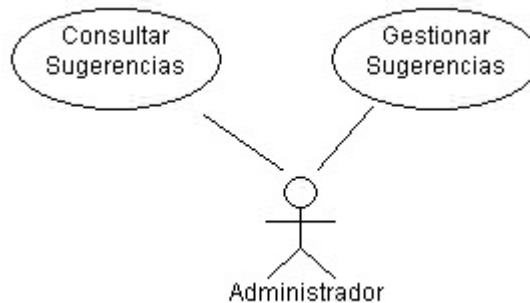


Tabla 10. Descripción del modelo de recepción de sugerencias.

Descripción del modelo de recepción de sugerencias
El sistema le permite al administrador consultar las sugerencias hechas por los usuarios habilitados para tal fin, para determinar cuáles de esos títulos buscados deberían ser adicionados a la sala de Fonoteca.

Casos de uso identificados	Breve descripción
Consultar Sugerencias	Permite que el administrador lea en una pantalla del sistema cuáles búsquedas infructuosas han sido registradas por el mismo y, por lo tanto, cuáles sugerencias han sido hechas por los usuarios a quienes se les otorga permiso para hacerlo.
Gestionar sugerencias	Por medio del uso de la pizarra, el administrador está en capacidad de conocer la necesidad de adquirir cierto material sonoro para la sala de Fonoteca.

▪ **Modelo de Formulación de sugerencias**

El caso de uso que se muestra en la siguiente figura permite realizar las actividades concernientes a la *formulación de sugerencias de material*, de manera tal, que existan interfaces de usuario de fácil manejo sin tener que acceder directamente la base de datos. (Ver Figura 12)

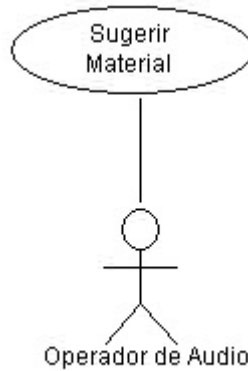
Tabla 11. Descripción del modelo de Formulación de sugerencias

Descripción del modelo de Formulación de sugerencias
El sistema le permite al Operador de Audio sugerir la adquisición de nuevos títulos al Administrador, después de consultar material sin éxito en el sistema.

Casos de uso identificados	Breve descripción
-----------------------------------	--------------------------

Sugerir Material	Permite que el Operador de Audio sugiera algún material a adquirir por la Emisora por medio de una pantalla en donde este usuario proporciona algunas características necesarias para distinguir el material a conseguir.
------------------	---

Figura 12. Modelo de Formulación de sugerencias



- **Modelo de Búsqueda de Material**

El caso de uso que se muestra en la siguiente figura permite realizar las actividades concernientes a la *Búsqueda de material sonoro*, de manera tal, que existan interfaces de usuario de fácil manejo sin tener que acceder directamente la base de datos.

Figura 13. Modelo de Búsqueda de Material

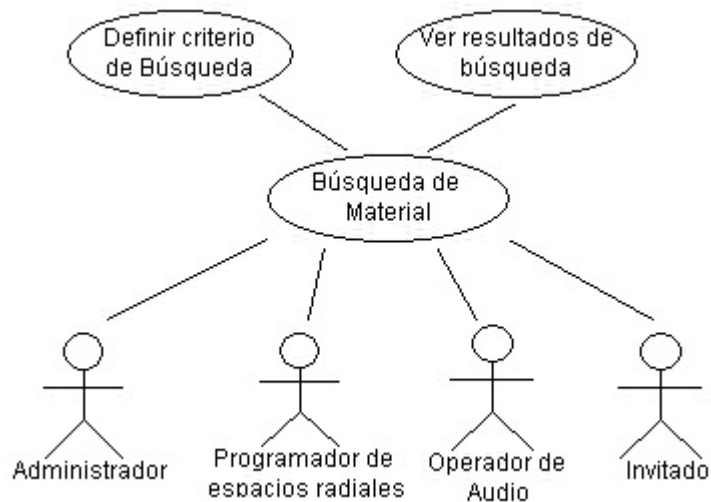


Tabla 12. Descripción del modelo de Búsqueda de Material

Descripción del modelo de Búsqueda de Material	
El sistema les permite a todos los usuarios consultar, mediante la selección de un criterio de búsqueda definido, el material existente en la sala de Fonoteca.	

Casos de uso identificados	Breve descripción
Definir criterio de búsqueda	Permite que el usuario escoja alguno de los criterios de búsqueda de material predefinidos.
Ver resultados de búsqueda	Permite visualizar en un listado los materiales sonoros que coinciden con el criterio de búsqueda elegido.

▪ **Modelo de administración de usuarios de la Pizarra**

El caso de uso que se muestra en la siguiente figura permite realizar las actividades concernientes a la *Administración de usuarios de la pizarra* (como crear, editar y eliminar usuarios), de manera tal, que existan interfaces de usuario de fácil manejo sin tener que acceder directamente la base de datos.

Figura 14. Modelo de administración de usuarios de la pizarra



Tabla 13. Descripción del modelo de administración de usuarios de la pizarra

Descripción del modelo de administración de usuarios del pizarra
El sistema le permite al administrador administrar los usuarios que usan la pizarra, en el mismo instante en que el usuario es creado para el sistema SEMCO, mediante los casos de

uso Crear, Editar y Eliminar Usuarios.	
Casos de uso identificados	Breve descripción
Crear usuario	Permite que el administrador agregue un nuevo usuario.
Editar usuario	Permite editar un usuario.
Eliminar usuario	Permite eliminar un usuario.

- **Modelo de Enviar y Recibir Mensajes**

El caso de uso que se muestra en la siguiente figura permite realizar las actividades concernientes al *Envío y Recepción de mensajes* (como crear, editar y leer mensajes). (Ver Figura 15).

Figura 15. Modelo de envío y recepción de mensajes

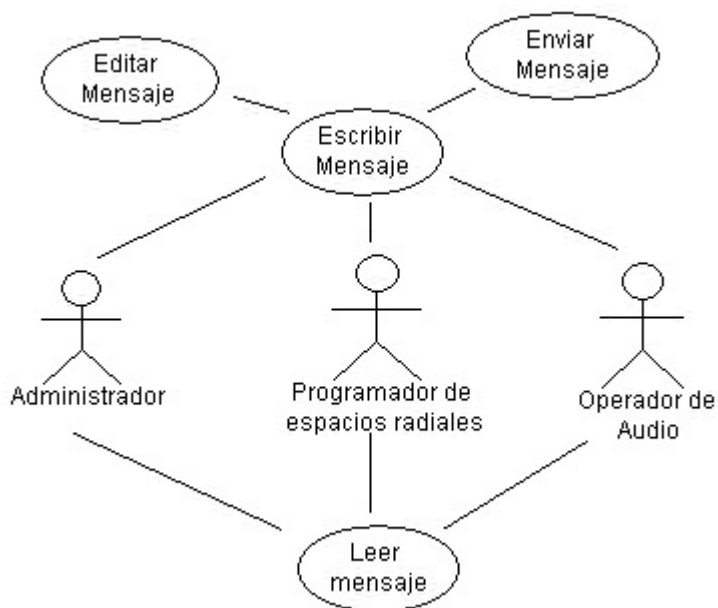


Tabla 14. Descripción del modelo de envío y recepción de mensajes

Descripción del modelo de envío y recepción de mensajes	
El sistema les permite a los usuarios, a excepción del <i>Invitado</i> escribir, editar, leer y enviar mensajes.	

Casos de uso identificados	Breve descripción
----------------------------	-------------------

Escribir Mensaje	Permite que el usuario escriba un nuevo mensaje.
Editar Mensaje	Permite editar un mensaje.
Enviar Mensaje	Permite enviar un mensaje.
Leer Mensaje	Permite leer mensajes publicados en la pizarra.

- **Modelo de Monitorear Mensajes**

El caso de uso que se muestra en la siguiente figura permite realizar las actividades concernientes al *Monitoreo de mensajes*.

Figura 16. Modelo de monitoreo de mensajes

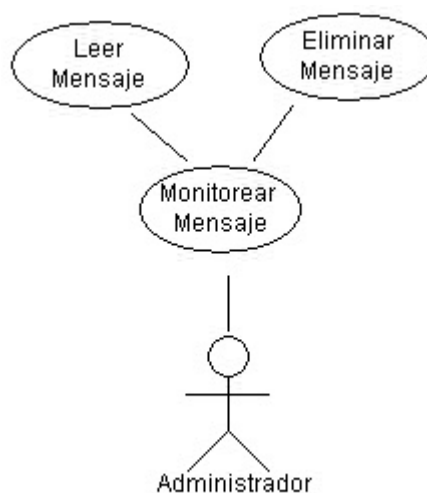


Tabla 15. Descripción del modelo de monitoreo de mensajes

Descripción del modelo de monitoreo de mensajes	
El sistema le permite al administrador leer y eliminar mensajes enviados por otros usuarios.	

Casos de uso identificados	Breve descripción
Leer Mensaje	Permite leer mensajes publicados en la pizarra por otros usuarios.
Eliminar Mensaje	Según el contenido del mensaje o el tiempo de duración, el administrador está en capacidad de eliminar un mensaje de la pizarra.

3.2.2.2 Análisis. En la fase de elaboración se refinará el modelo de análisis realizado en la fase de inicio, utilizando los casos de uso que son significativos desde el punto de vista de la arquitectura y aquellos que requieran más refinamiento, para lograr una mejor comprensión del sistema.

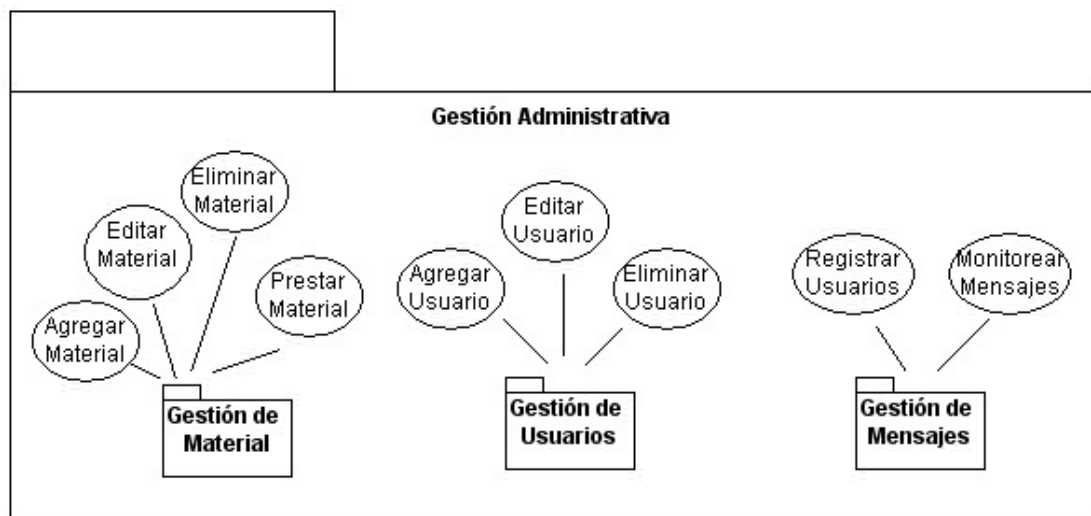
- **Análisis de la arquitectura**

En esta fase se agregarán los nuevos casos de uso a cada uno de los paquetes existentes y se rediseñarán o crearán nuevos.

- **Paquetes de análisis de gestión**

En la siguiente figura se muestra el paquete de análisis de los usuarios, materiales y mensajes.

Figura 17. Paquete de análisis de gestión administrativa, conteniendo los paquetes de gestión de Material, Usuarios y Mensajes



- **Análisis de los casos de uso**

En esta fase todavía existen casos de uso que no son totalmente comprensibles tal como están descritos en el modelo de casos de uso; por lo tanto, deben ser refinados en función de las clases del análisis que existen en el ámbito de los requisitos pero que no se implementan necesariamente en forma directa en el sistema.

Para realizar este análisis se deben identificar primero las clases del análisis (entidad, control e interfaz) y la manera como se relacionan entre sí para llevar a cabo el flujo de sucesos del caso de uso. Para este fin se han utilizado diagramas de colaboración, cuyo objetivo es identificar requisitos y responsabilidades sobre las clases.

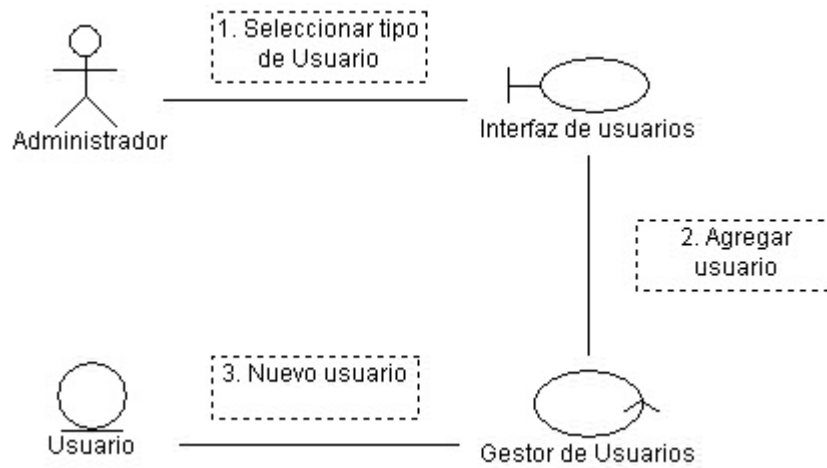
- **Gestión administrativa**

En el paquete de gestión administrativa se ha escogido el caso de uso de **agregar usuario**. Al intentar seleccionar un criterio para elegir el caso de uso a mostrar con este diagrama de colaboración se pensó en el más común de todos: los casos de uso en los que más clases interactúan. Sin embargo, para este proyecto en particular, el usuario Administrador es quien realmente interactúa con el sistema y con los usuarios en todo momento en las mismas condiciones; es decir, no existe un caso de uso en que interactúen varias clases de la entidad **usuarios**.

En la siguiente figura se observa el diagrama de colaboración para agregar usuarios y se utiliza una clase de la entidad **usuarios**. Esta almacena la información de los usuarios del sistema, sean Administradores, Operadores de Audio, Programadores de Espacios Radiales e Invitados.

La clase de control **gestor de usuarios** es la encargada de incluir el nuevo usuario en el sistema. La clase de interfaz denominada **interfaz de usuario** se encarga de permitir al usuario ingresar la información adecuada para la adición de usuarios.

Figura 18. Diagrama de colaboración para el caso de uso Agregar usuario



3.2.2.3 Diseño. El objetivo del diseño es crear una arquitectura estable y sólida, además de un plano del modelo de implementación. Para ello, se deben comprender en profundidad aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, los componentes y los sistemas operativos, entre otros.

- **Diseño de la arquitectura**

Con el diseño de la arquitectura se pretende definir los modelos de diseño y despliegue y su arquitectura, mediante la identificación de los nodos, configuraciones de red, subsistemas, interfaces, software del sistema y capas intermedias.

- **Nodos y configuración de red**

En el diagrama de despliegue definido en la fase de inicio (ver Figura 9) se definió la arquitectura base del sistema. Esta cuenta con nodos cliente y servidor. Vale la pena recordar que en el nodo servidor se encuentra la **lógica de aplicaciones** y la **capa de datos**, mientras que en el cliente se encuentra la **capa de presentación**.

La configuración de cada uno de los nodos se presenta a continuación:

- Nodo Servidor: Se cuenta con un servidor con las siguientes características de software y hardware:
 - Procesador Intel Pentium III – 1.2 GHz
 - Disco Duro de 30 GB
 - Memoria RAM de 256 MB
 - Tarjeta de Red de 10/100 Mbps

Para el acceso al servidor se cuenta con la infraestructura interna de la red de las Emisoras, la cual cuenta con cableado estructurado, switches ubicados en cada uno de los cuartos de producción y un Backbone que intercomunica a éstos.

En esta fase no es indispensable contar con una dirección IP válida, ya que las pruebas preliminares se realizarán con otros dos equipos de similares características.

- Nodo Cliente: Los requerimientos mínimos para el cliente son:
 - Procesador de 200 Mhz.
 - Memoria RAM de 32 MB
 - Tarjeta de Red

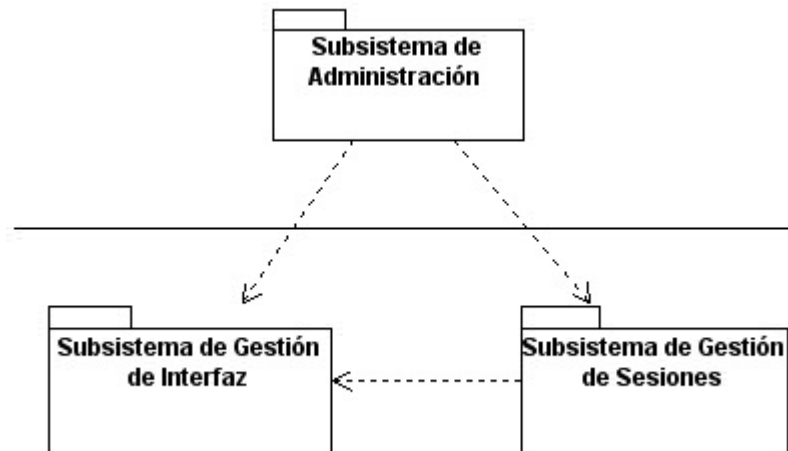
El nodo cliente deberá estar configurado para funcionar dentro de la red interna de las Emisoras con el fin de acceder al nodo servidor.

▪ **Subsistemas**

Los subsistemas permiten dividir el trabajo de diseño y, posteriormente, el de implementación. Para obtener los subsistemas de diseño se deben tomar como base los paquetes de análisis definidos anteriormente.

La siguiente figura presenta los subsistemas de diseño.

Figura 19. Subsistemas de diseño



- **Diseño basado en el modelo de tres capas**

El desarrollo del sistema se realizará a partir del concepto de la arquitectura de tres capas, el cual consiste en dividir el sistema en capas lógicas capaces de interactuar entre sí.

La primera es la **capa de datos**, que encapsula el manejo de la base de datos y procesos relacionados.

La segunda capa es la **capa lógica** o **capa de negocio**, la cual llevará a cabo la lógica del negocio y la tercera es la **capa de presentación**, la cual encapsula las funciones de interfaz de usuario.

Figura 20. Modelo de tres capas



▪ Capa de datos

La capa de datos está determinada por la base de datos y las funciones para su manejo.

Las funciones más importantes de esta capa son:

- Manejar la integridad referencial de las tablas
- Manejar las interfaces de inserción, modificación y eliminación de registros.

Las funciones de esta capa estarán desarrolladas en el lenguaje PHP.

▪ Capa Lógica

La capa lógica o de negocio contendrá los scripts y las funciones que llevan a cabo la lógica del sistema basándose en los casos de uso.

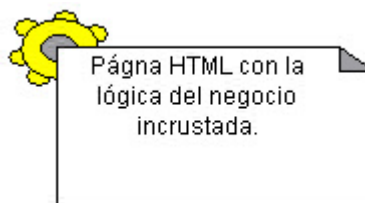
▪ Capa de presentación

Esta capa es la encargada de la interacción con el usuario final. Esta capa se implementará de manera que su mantenimiento no interfiera con la capa lógica; para ello, en el diseño se utilizarán plantillas.

◆ Diseño guiado por plantillas

En la mayoría de los casos, la lógica del negocio se encuentra incrustada en el código HTML de una página. Cuando se desarrollan aplicaciones de esta forma la interfaz debe diseñarse primero para luego insertar el código PHP que posee la lógica del negocio. Esto hace que el mantenimiento de las páginas sea más complicado. Además, no se divide el trabajo entre un equipo de diseño y uno de programación, ya que el equipo de diseño tendría que manejar el código PHP incrustado en las páginas y el equipo de programación tendría que preocuparse por el código HTML de la página.

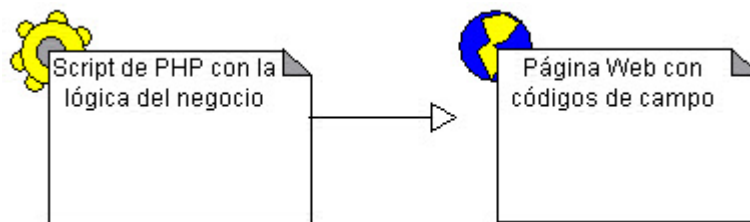
Figura 21. Modelo de páginas dinámicas con código incrustado



Una aproximación diferente es separa el diseño de la funcionalidad, permitiendo que la lógica del negocio esté en un archivo independiente al archivo HTML que contiene la interfaz. Esto se puede lograr mediante el uso de librerías de PHP. Al utilizar esta alternativa, el funcionamiento de la aplicación es diferente:

Se tiene una página HTML con ciertos códigos de campo a los cuales es posible asignarles contenido mediante el uso de las librerías. De esta manera, la lógica del negocio realiza sus operaciones de manera independiente a la interfaz y, únicamente cuando lo requiera, utilizará las librerías para asignarle valores a los códigos de campo y enviar la interfaz resultante al cliente.

Figura 22. Página Web con códigos de campo



De esta manera, el mantenimiento de la interfaz es independiente del mantenimiento del código PHP y viceversa.

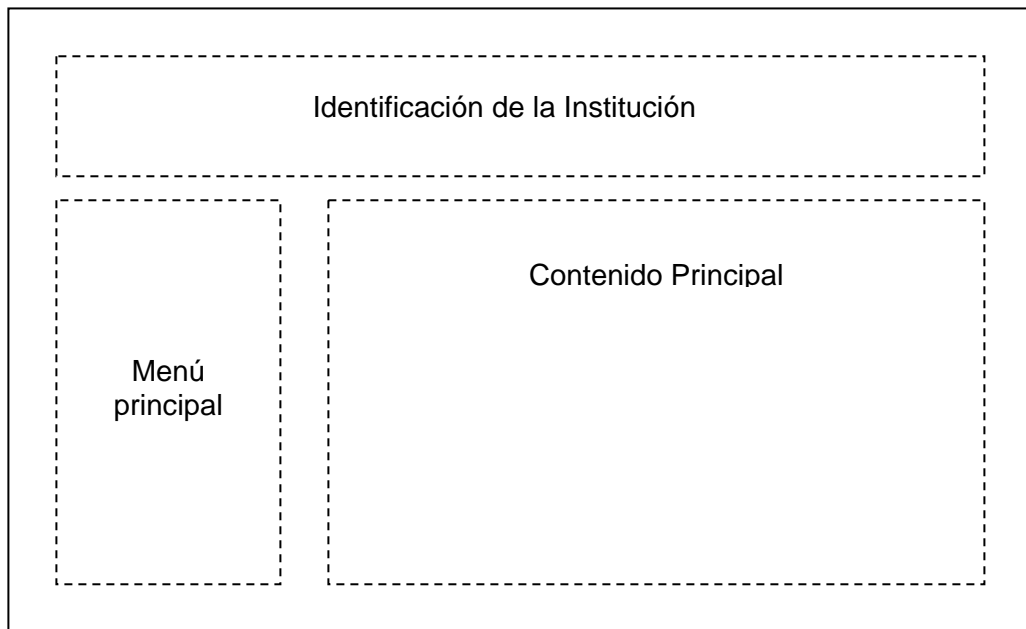
- **Diseño de la interfaz**

Para el diseño de la interfaz se debe tener en cuenta los tipos de usuario que van a utilizar el sistema. Estos tipos ya está plenamente identificados y son: *Administrador*, *Operador de Audio*, *Programador de Espacios Radiales* e *Invitado*. Si se observan los casos de uso del sistema se puede notar que el Administrador comparte varios casos de uso con los demás usuarios, de modo que la interfaz de usuario será muy similar para los diferentes usuarios. La Figura 23 muestra el esquema general de Interfaz de usuario.

La sección *Identificación de la Institución* contiene el logotipo del Sistema de Información y el logotipo de TELEUIS. La sección de *Menú principal* cambia de acuerdo al usuario que

se haya registrado en el sistema; a su vez, la sección de *Contenido Principal* cambia de acuerdo a la opción seleccionada en el Menú Principal.

Figura 23. Interfaz General de Usuario



El subsistema de gestión de interfaz se encargará de presentar, activar o desactivar las opciones, dependiendo del tipo de usuario que ingrese al sistema. Por ejemplo, al Administrador se le presentará la opción de *Material*, pero a los demás usuarios no se les dará esta alternativa.

- **Diseño de la base de datos**

De manera frecuente, las clases de análisis guardan estrecha relación con las entidades en el modelo entidad – relación. El Anexo A ilustra el diagrama entidad – relación para el sistema SEMCO.

- **Software a utilizar en el sistema**

El sistema será desarrollado bajo el Sistema Operativo **Windows XP**. A su vez, el sistema será desarrollado utilizando el lenguaje de programación **PHP** (Hypertext Processor);

lenguaje interpretado de alto nivel, ejecutado del lado del servidor. El sistema de gestión de base de datos será **MySQL**, sistema de libre distribución que posee todas las características de cualquier gestor de base de datos comercial.

Los tres productos WINXP, PHP y MySQL, no reportan ningún costo a la realización del proyecto, ya que el primero de ellos cuenta con su respectiva licencia de uso personal, y los dos últimos son productos desarrollados bajo licencia GNU. Además, se puede obtener soporte de miles de usuarios y desarrolladores alrededor del mundo.

3.2.2.4 Implementación. A partir de los elementos de diseño obtenidos, se implementan los componentes críticos y se realizan pruebas correspondientes. El resultado de este proceso es una arquitectura base, implementada a partir de menos del 10% de los casos de uso identificados.

En esta fase la implementación sólo se enfoca a instalar y configurar el software del sistema, así como de asignar los componentes necesarios a cada uno de los nodos detallados en el diagrama de despliegue del sistema.

- **Software del sistema**

Se instaló en el sistema Operativo Windows 2000. Se realizó la configuración de los parámetros de red y la infraestructura para conectar el servidor a la Red de las Emisoras. Los parámetros de red son los siguientes:

Nombre del Host:	produccion
Dirección IP:	10.10.0.1
Submáscara de red	255.255.255.0

Se instaló y configuró el servidor de bases de datos MySQL, así como la interfaz gráfica de manejo *MySQL Front 2.2*.

Se instaló y configuró el servidor Web *Internet Information Service* y se configuró el módulo de PHP con acceso a bases de datos MySQL.

Se implementó el modelo entidad – relación definido anteriormente en la base de datos MySQL.

3.2.3 Evaluación. En la fase de elaboración, la evaluación está orientada a verificar si la arquitectura del sistema es estable y puede adaptarse a cambios que puedan ocurrir durante el proceso de construcción y, luego, en la vida útil del sistema.

Para definir de adecuadamente la estructura se deben analizar los casos de uso que puedan afectarla de manera directa; además, se deben identificar y mitigar los posibles riesgos críticos que puedan surgir o que ya estén identificados.

3.2.3.1 Desarrollo del modelo de casos de uso. En esta fase, un mínimo del 80% de los casos de uso debe ser analizado exhaustivamente. Cabe aclarar que algunos casos de uso son tan sencillos que no requieren profundización. Para tales casos de uso una breve descripción es suficiente.

Tabla 16. Casos de uso identificados hasta el final de la fase de elaboración.

Casos de uso identificados	Identificado	Descrito	Analizado	Diseñado e Implementado	¿Se comprende completamente?
Monitorear mensajes	Fase Inicio	Fase Inicio			Si
Leer Mensaje	Fase Elaboración	Fase Elaboración			Si
Eliminar Mensaje	Fase Elaboración	Fase Elaboración			Si
Administrar usuarios	Fase Inicio	Fase Inicio			Si
Agregar usuario	Fase Inicio	Fase Inicio			Si
Editar usuario	Fase Inicio	Fase Inicio			Si
Eliminar usuario	Fase Inicio	Fase Inicio			Si

Agregar Material	Fase Inicio	Fase Inicio			Si
Editar Material	Fase Inicio	Fase Inicio			Si
Eliminar Material	Fase Inicio	Fase Inicio			Si
Prestar Material	Fase Inicio	Fase Inicio			Si
Consultar Sugerencias	Fase Elaboración	Fase Elaboración			Si
Gestionar Sugerencias	Fase Elaboración.	Fase Elaboración			Si
Sugerir Material	Fase Elaboración	Fase Elaboración			Si
Definir Criterio de búsqueda	Fase Elaboración	Fase Elaboración			Si
Ver resultados de búsqueda	Fase Elaboración	Fase Elaboración			Si
Escribir mensaje	Fase Elaboración	Fase Elaboración			Si
Editar mensaje	Fase Elaboración	Fase Elaboración			Si
Enviar mensaje	Fase Elaboración	Fase Elaboración			Si
Leer mensaje	Fase Elaboración	Fase Elaboración			Si

3.2.3.2 Vista de la arquitectura. En esta fase se identificaron la mayoría de los casos de uso del sistema. De éstos, se analizaron los más adecuados para obtener las clases de entidad, control de interfaz y gestión.

Las clases de entidad se utilizaron en el diseño del modelo entidad-relación. Las clases de control de interfaz y control de sesión se convertirán en funciones o scripts, en la siguiente fase.

3.2.3.3 Cumplimiento de criterios. Al identificar la mayoría de los casos de uso del sistema, fue posible crear una arquitectura base para la fase de construcción.

Al utilizar un diseño de tres capas se obtiene una mayor flexibilidad del sistema debido a que se puede modificar una de ellas sin mayor repercusión sobre las otras.

3.3 FASE DE CONSTRUCCION

En esta fase se desarrolla una versión funcional del sistema para ser objeto de pruebas y ajustes que lo lleven a ser un producto terminado.

Los objetivos específicos de esta fase son identificar, describir y analizar todos los casos de uso del sistema, realizar ajustes a la arquitectura cuando sea necesario y crear los manuales de usuario y de administración del sistema.

3.3.1 Planeación. La planeación en esta fase consiste en determinar el tiempo, las iteraciones a realizar y los productos a entregar en cada iteración de desarrollo del sistema.

3.3.1.1 Hitos y asignaciones de tiempo. Para lograr que el desarrollo del sistema se lleve a cabo en los plazos propuestos es necesario dividirlo en módulos. Cada módulo corresponde a los subsistemas identificados en la fase de elaboración. (Ver Figura 19)

Generalmente los subsistemas son divididos en dos categorías:

- Subsistemas Generales: Son la base operativa para los demás subsistemas, aunque no son visibles al usuario del sistema. Los subsistemas generales son el Subsistema de gestión de Sesiones y el Subsistema de Interfaz.
- Subsistemas Específicos: Son aquellos que realizan una tarea específica y son visibles al usuario del sistema. Para el presente proyecto, el Subsistema de Administración es el único que pertenece a esta categoría.

El proceso de desarrollo se dividirá en tres iteraciones, donde cada una de ellas corresponde a un subsistema.

La Tabla 17 ilustra las diferentes iteraciones y el tiempo requerido para su desarrollo.

Tabla 17. Tabla de asignación de tiempo a cada una de las iteraciones.

Subsistemas	Tiempo Requerido (Semanas)
Sesiones	2
Interfaz	2
Administración	4
Tiempo total de desarrollo	8

El criterio para definir si la fase de construcción está completa se obtiene mediante la respuesta a preguntas como: ¿se logró identificar y describir todos los casos de uso de cada subsistema? ¿Se construyó una versión funcional de cada subsistema? ¿El sistema posee toda su funcionalidad? ¿Se han elaborado los manuales de usuario y administración?

3.3.1.2 Flujos, actividades y productos a entregar en la Fase de Construcción.

Además de entregar una versión funcional del sistema, se deben entregar las versiones finales de los casos de uso y de la arquitectura del sistema. El objetivo de esta fase será presentar la implementación de cada uno de los subsistemas desarrollados.

Las pruebas a cada uno de los subsistemas fueron realizadas por medio de reuniones periódicas con los miembros de las emisoras institucionales; por tal motivo, sólo se presentará una breve descripción de cómo se definieron.

3.3.2 Subsistemas Generales. Los subsistemas generales son requeridos por los demás subsistemas para su funcionamiento, tal como se aprecia en el diagrama de diseño. (Ver Figura 19).

3.3.2.1 Subsistema de Gestión de Sesiones. En general, un sistema de gestión de sesiones permite preservar datos mientras el usuario navega por el sitio Web. Esto permite construir aplicaciones personalizadas, debido a que el sistema puede *conocer* al usuario que lo está accediendo y, dependiendo de sus acciones, presentarle las opciones adecuadas.

Anteriormente, la gestión de sesiones se realizaba con *cookies*. La principal desventaja de esta alternativa era que los datos de sesión quedaban almacenados en el cliente y, por consiguiente, podrían ser modificados de manera inescrupulosa para acceder ilegalmente a recursos del servidor. Con esto en mente, los desarrolladores de PHP implementaron un conjunto de funciones de gestión de sesiones, que utiliza *cookies*, pero adicionalmente almacena información de la sesión en el servidor.

El soporte de sesiones de PHP permite transportar variables importantes para la lógica del negocio a través de las solicitudes del cliente. Cuando se finaliza un *script* de PHP, todas las variables que han sido registradas se almacenan automáticamente. Luego, es posible recuperar el valor de dichas variables. Las variables están disponibles hasta que el cliente termina la sesión de manera explícita, o de manera automática, una vez cumplido cierto tiempo de inactividad por parte del usuario.

Cabe anotar que el sistema no almacena las contraseñas de usuario en un archivo o en la base de datos, pues la validación de la contraseña se realiza por medio del sistema operativo. Esto aumenta la seguridad del sistema, pues garantiza que ningún usuario puede cambiar su contraseña o la de otro usuario.

- **Implementación**

Cuando un usuario ingresa al sistema, éste valida la información registrada. Si los datos son correctos, se inicia la sesión y se crean las variables necesarias para la lógica del negocio para rastrear al usuario durante su permanencia en el sistema.

Esta sesión será destruida si el usuario selecciona la opción **Salir del Sistema** o pasado cierto tiempo⁶ de inactividad por parte del usuario. Si los datos son incorrectos, el sistema redireccionará automáticamente al usuario a la página de inicio y le indicará un mensaje de error.

⁶ Debe ser susceptible de ser modificado por el administrador del sistema.

El objetivo principal del subsistema de gestión de sesiones es suministrar el primer nivel de seguridad en el sistema. Además, es el encargado de mantener las variables de sesión necesarias para la lógica del negocio.

Debido a que este módulo se creó como un conjunto de funciones en un archivo separado, debe ser incluido al inicio de todos los *scripts* de PHP, para luego invocar la función de chequeo de sesión, *session_check()*.

3.3.2.2 Subsistema de Gestión de Interfaz. El objetivo principal de este subsistema es suministrar las opciones de interfaz de usuario, dependiendo del tipo de éste. Por medio de la utilización del subsistema de gestiones se construye la página a partir de una plantilla HTML y la información que va a ser insertada en los códigos de campo de ésta.

Este subsistema funciona de manera coordinada con el subsistema de gestión de sesiones, pues utiliza las variables de sesión creadas por éste para determinar aspectos del ambiente relevantes para la lógica del negocio, tales como el tipo de usuario para construir la página presentada a éste.

- **Implementación**

El subsistema de gestión de interfaz no se puede agrupar en un conjunto de funciones independientes a la lógica del negocio, y se maneja como fracciones de código en cada uno de los subsistemas específicos para ocultar o mostrar opciones, dependiendo del tipo de usuario que haya iniciado sesión.

Sin embargo, esto no contradice el planteamiento original de la separación de la interna de la funcionalidad, ya que la función del subsistema de interfaz es asignar valores a los códigos de campo de las plantillas HTML, que son totalmente independientes de éste.

Los pasos realizados por el subsistema son los siguientes:

- Crear el objeto de plantillas
- Definir la página que es plantilla

- Dependiendo de las variables de sesión, designar valor a los códigos de campo.

3.3.3 Subsistemas Específicos. Estos subsistemas son los que conforman los módulos del sistema desde el punto de vista de usuario y tienen su sustento en los casos de uso identificados. Debido a que los casos de uso ya han sido descritos, es este apartado sólo se presentarán los aspectos más importantes de cada uno de ellos.

Los subsistemas específicos interactúan con los subsistemas generales para presentar las opciones permitidas dependiendo del tipo de usuario.

3.3.3.1 Subsistema de Administración. Este subsistema es el encargado de manejar usuarios, material y mensajes.

- **Implementación**

El subsistema de Administración permite al usuario Administrador realizar cambios al sistema SEMCO en tres aspectos diferentes: Usuarios, Material, Mensajes.

El subsistema permite al Administrador registrar un nuevo usuario, otorgarle permisos para hacer uso del sistema, y modificar algún aspecto relacionado con el usuario hasta el punto de poder eliminarlo del sistema si fuese necesario.

Así mismo, este subsistema hace posible la administración del material sonoro existente en la sala de Fonoteca de las emisoras, al permitir agregar nuevos materiales, editar materiales existentes, eliminar materiales si fuese necesario, administrar el préstamo de los mismos a usuarios autorizados y, también, recibir sugerencias acerca de la adquisición de nuevo material.

Igualmente, el subsistema de administración permite controlar los mensajes que son enviados entre los usuarios para determinar su validez y relevancia dentro del contexto de la institución.

- **Pruebas**

Las pruebas estuvieron enfocadas hacia la verificación del proceso de manejo de usuarios, de material y de mensajes.

3.3.4 Evaluación. En la fase de construcción la evaluación está orientada a verificar si el sistema desarrollado ha alcanzado su capacidad inicial de operación. De esta manera, el producto final de esta fase es el sistema mismo.

En la fase de elaboración se definieron los casos de uso generales, de manera que los casos de uso faltantes son derivaciones o adaptaciones de éstos, motivo por el cual se omite su análisis.

Igualmente, se omite el análisis de la arquitectura, la cual no sufrió variaciones.

A continuación se presenta la tabla final de casos de uso, ilustrando en cuál fase fueron identificados, descritos, analizados, diseñados e implementados.

En este proyecto, los casos de uso fueron identificados plenamente desde la fase de elaboración, de manera que la tabla final no ha sufrido modificación alguna.

Tabla 18. Tabla final de casos de uso del sistema

Casos de uso identificados	Identificado	Descrito	Analizado	Diseñado e Implementado
Monitorear mensajes	Fase Inicio	Fase Inicio		F. Construcción
Leer Mensaje	Fase Elaboración	Fase Elaboración		F. Construcción
Eliminar Mensaje	Fase Elaboración	Fase Elaboración		F. Construcción
Administrar usuarios	Fase Inicio	Fase Inicio		F. Construcción
Agregar usuario	Fase Inicio	Fase Inicio		F. Construcción
Editar usuario	Fase Inicio	Fase Inicio		F. Construcción
Eliminar usuario	Fase Inicio	Fase Inicio		F. Construcción
Agregar Material	Fase Inicio	Fase Inicio		F. Construcción
Editar Material	Fase Inicio	Fase Inicio		F. Construcción
Eliminar Material	Fase Inicio	Fase Inicio		F. Construcción
Prestar Material	Fase Inicio	Fase Inicio		F. Construcción

Consultar Sugerencias	Fase Elaboración	Fase Elaboración		F. Construcción
Gestionar Sugerencias	Fase Elaboración.	Fase Elaboración		F. Construcción
Sugerir Material	Fase Elaboración	Fase Elaboración		F. Construcción
Definir Criterio de búsqueda	Fase Elaboración	Fase Elaboración		F. Construcción
Ver resultados de búsqueda	Fase Elaboración	Fase Elaboración		F. Construcción
Escribir mensaje	Fase Elaboración	Fase Elaboración		F. Construcción
Editar mensaje	Fase Elaboración	Fase Elaboración		F. Construcción
Enviar mensaje	Fase Elaboración	Fase Elaboración		F. Construcción
Leer mensaje	Fase Elaboración	Fase Elaboración		F. Construcción

Para finalizar, los manuales de usuario y administración estarán disponibles posteriormente en la ayuda en línea del sistema.

4. CONCLUSIONES

Al finalizar el desarrollo del sistema SEMCO 1.0 se puede concluir lo siguiente:

- El producto, cuya descripción se presenta en este documento, hizo que la administración de los recursos sonoros disponibles para la realización de los espacios radiales de las Emisoras fuera una tarea mucho más fácil que en el pasado.
- La consulta del material sonoro en la sala de Fonoteca dejó de ser tediosa y difícil, y se tornó fácil, intuitiva y, sobre todo, óptima, ya que los resultados de las búsquedas proporcionan datos claves para la ubicación física del material requerido.
- El proceso de inventario periódico del material de la Fonoteca es, gracias a la implantación de SEMCO, confiable y rápido, gracias a la posibilidad de obtener listados de las existencias directamente desde la Base de datos.
- El uso de una metodología como el *proceso unificado* para el desarrollo de software permite la planeación y desarrollo del sistema de una manera clara y ordenada. Por otro lado, se aprovechó una de las ventajas de esta metodología como es una de las premisas que guía el proceso, es *iterativo e incremental*; esto permite que se logren controlar los riesgos desde un principio, cuando se está menos presionado por cumplir los plazos, así como, la reducción de costos en los riesgos, debido a que en caso de tener que repetir la iteración se pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero.
- El aprovechamiento de una de las principales ventajas de la arquitectura de 3 capas como es la independencia entre capas. Es así como se realizó la implementación de la capa de presentación, por medio de plantillas, lo cual permitió una mayor independencia de ésta con respecto a la de aplicaciones,

facilitando modificaciones sencillas y rápidas en cualquiera de estas capas sin afectar el funcionamiento de las otras.

- El desarrollo de proyectos de tecnología Intranet, que explica una vez más la necesidad del desarrollo de este tipo de sistemas de información que permitan a una entidad almacenar y hacer uso de información que es de interés exclusivo y, en un momento dado, pueda ser de acceso general.
- El desarrollo del proyecto centrado en el empleo de casos de uso permitió avanzar a través de una serie de flujos de trabajo y enlazar las actividades desarrolladas en cada uno de ellos.
- El uso de herramientas de software libre⁷ como PHP y MySQL permitieron a los autores confirmar las ventajas que traen consigo el uso de estas herramientas, como es la inversión de pequeñas sumas de dinero en software y beneficios de seguridad, estabilidad y desempeño de las aplicaciones, que para el caso de pequeñas medianas empresas es una ventaja de fácil aplicación.

⁷ "Software Libre" se refiere a la libertad de los usuarios de correr, copiar, distribuir, estudiar, cambiar y mejorar el software. El término no se refiere a que sea gratis o que no se pueda comercializar.

5. RECOMENDACIONES

Al finalizar este proyecto, se han detectado *recomendaciones* a tener en cuenta para posteriores versiones del sistema, las cuales son:

- Es necesaria la inclusión de las Emisoras Institucionales UIS Stereo y UIS AM en la LAN de la Universidad para así tener la posibilidad de que el sistema SEMCO sea consultado por un mayor número de usuarios y desde cualquier lugar del campus universitario.
- Eventualmente, debería ser posible la publicación del sistema en Internet como la página Web institucional de las emisoras, con las ventajas que esto podría traer consigo, tal como la posibilidad de transmitir en línea la programación de UIS Stereo.
- El administrador del sistema debe realizar copias de seguridad, por lo menos una vez al mes, debido al crecimiento de la información, esto con miras al proceso de inventario que se realiza periódicamente en la sala de Fonoteca.

BIBLIOGRAFÍA

BOBADILLA, Jesús y ALCOCER Alejandro. **Superutilidades para Webmasters**. McGraw Hill. Primera edición. España, 1999.

GOMEZ, Luis Carlos. **Guía para el Desarrollo de Proyectos de Grado**. Bucaramanga. UIS, 1993.

JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. **El Proceso Unificado de Desarrollo de Software**. Primera edición. Addison Wesley. España, 2000.

----- **Manual de Referencia**. Primera edición. Addison Wesley. España, 2000.

PRESSMAN, Roger. **Ingeniería del software. Un enfoque práctico**. Cuarta edición. McGraw Hill. España, 1998.

RATSCHILLER, Tobias y GERKEN, Till. **Creación de Aplicaciones Web con PHP 4**. Primera edición. Prentice Hall. España, 2001.

SAETHER, Stig, AULBACH, Alexander y SCHMID, Egon. **Manual de PHP**. Quinta edición. Grupo de documentación de PHP, 2001

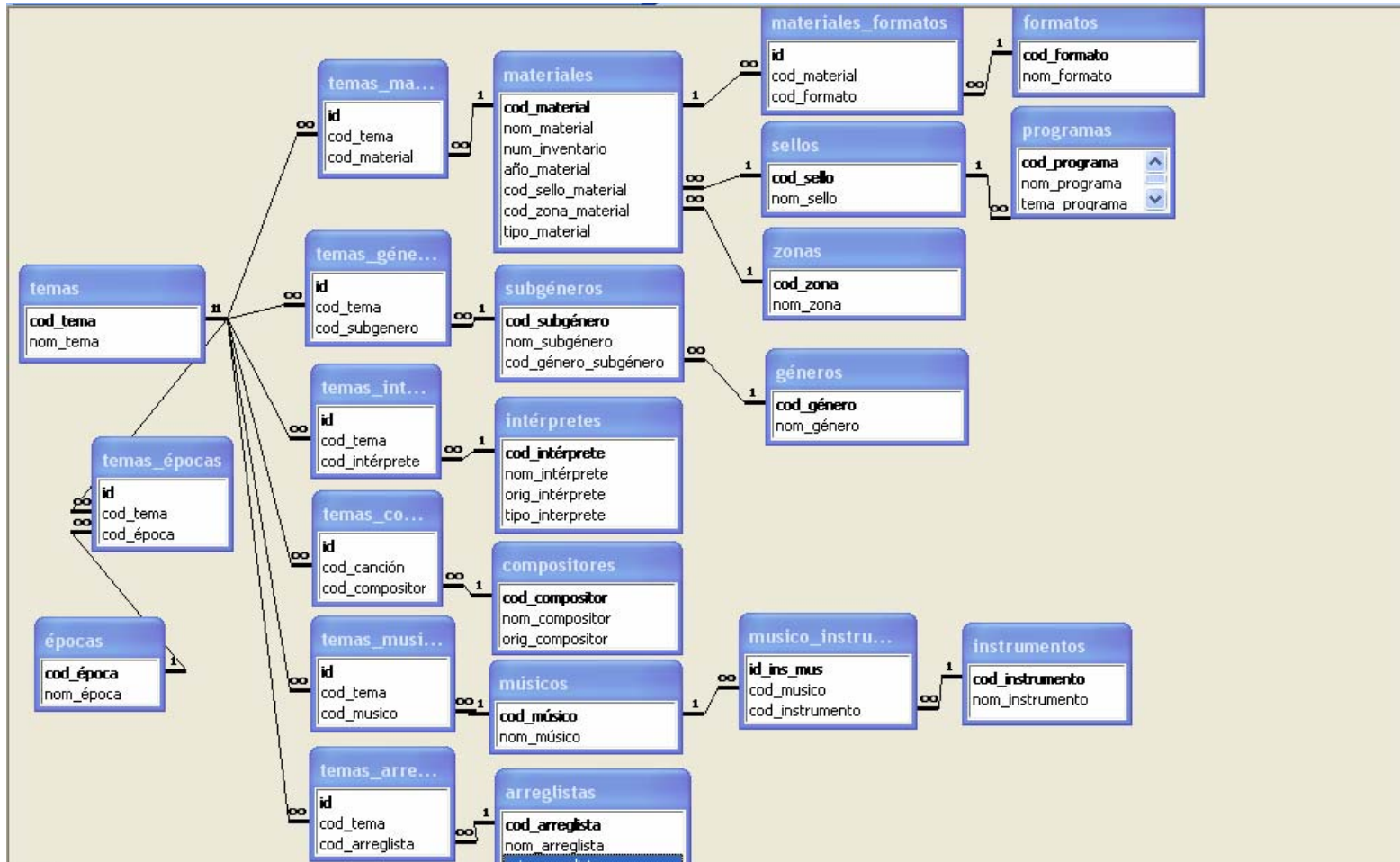
SUÁREZ, Edwin. **Sistema de Información Intranet – Extranet para la seccional de la UIS Barrancabermeja**. Bucaramanga. Junio, 2000.

www.php.net Sitio oficial de PHP

www.phpbuilder.com Sitio oficial de desarrolladores Web

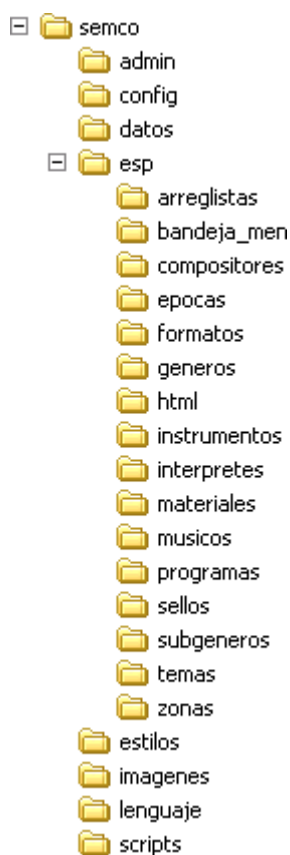
www.devarticles.com Sitio Internet para desarrolladores de software.

ANEXO 1. MODELO ENTIDAD RELACION SISTEMA SEMCO



ANEXO B. DESCRIPCIÓN DE LA CONFIGURACIÓN DE LOS DIRECTORIOS DEL SISTEMA

En el siguiente gráfico se muestra la estructura de archivos desarrollada con el producto SEMCO:



En su mayoría, los archivos usados para almacenar los Scripts PHP tienen un esquema de nombres el cual permite su fácil identificación.

Normalmente, tienen el prefijo *sem*, el cual indica el nombre de la aplicación; posteriormente, un carácter de subrayado seguido de la utilidad que realiza, codificada por tres letras definidas arbitrariamente por los autores del proyecto. Luego, la tarea específica para el cual fue diseñado; en algunos casos, la letra “e”, la cual denota edición o administración y la letra “a”, que significa actualización.

Dentro de la carpeta *esp*, se encuentran algunos archivos poli funcionales, los cuales fueron nombrados de diferente manera.

Dentro de la carpeta *admin* se encuentran los siguientes archivos:

Archivo	Uso
sem_usu_e.php	Presenta el formulario para la edición de usuarios del sistema.
sem_usu_a.php	Realiza la llamada a la capa de datos para realizar la operación de actualización de los datos de usuario.

En la carpeta *config* se encuentran archivos de configuración del sistema. Estos archivos son compartidos por las diferentes páginas del sistema, lo cual facilita su instalación y personalización.

Archivo	Uso
Config.php	Contiene los valores de las constantes de configuración del sistema, tales como el número de registros para realizar el proceso de paginación de listados, la configuración de la conexión de la base de datos, entre otros elementos (el archivo se encuentra documentado, con comentarios php, lo cual facilita su comprensión y modificación según las necesidades).
Funciones.php	Contiene funciones de uso general en el sistema.
Seguridad.php	Realiza el proceso de validación de usuarios. Este archivos es de vital importancia para el sistema, ya que proporciona el mecanismo de seguridad y prepara el sistema permitir ediciones o para mostrar contenido en modalidad sólo texto.

Dentro del directorio *datos*, se encuentran las capas de conexión y la capa de datos y de negocio del sistema; los cuales proporcionan servicios a toda la aplicación al almacenar toda la lógica del negocio.

Archivo	Uso
cconexion.php	Este archivo contiene una serie de funciones que permiten crear capas de datos con otros motores de datos diferentes a MySQL, lo cual agrega escalabilidad al sistema.

cdatos.php	Contiene la capa de datos y la lógica del negocio de todo el sistema de información; este archivo contiene las consultas y los algoritmos creados para manejar los datos de la aplicación en general.
------------	---

La carpeta *estilos* mantiene los archivos *css* necesarios para mejorar la presentación del sistema ante el usuario final. Al crear archivos de estilos en cascada se permite cambiar la apariencia del sistema, en cuanto a esquemas de colores, tipos de letra, etc. No se pueden modificar elementos como la diagramación externa sin modificar la lógica del negocio.

Archivo	Uso
Semco.php	Archivo <i>css</i> del sistema Semco.

La carpeta *imágenes*, contiene los gráficos creados para mejorar la experiencia del usuario final en cuanto a presentación y navegabilidad el sitio.

El sistema SEMCO tiene capas de lenguaje, lo cual lo proyectan como producto internacional y permiten la fácil modificación y personalización de los mensajes que se le presentan al usuario final.

Archivo	Uso
espanol.php	Capa de lenguaje para el usuario final.
espanol-admin.php	Capa de datos para la zona administrativa.

La carpeta *Scripts* contiene archivos de secuencias de comandos que permiten realizar validaciones del lado del cliente y realizar operaciones de lógica del negocio del lado cliente, lo cual descarga el procesamiento del lado del servidor y proporciona flexibilidad en la administración y visualización de los datos.

La carpeta *esp*, es el núcleo funcional de SEMCO. En esta carpeta se encuentran varios subdirectorios que tienen un orden lógico específico; es decir, las carpetas encontradas dentro de *esp*, tienen el mismo nombre que tiene la tabla de la base de datos que administra. Esto con el fin de proporcionar un esquema para realizar operaciones en la lógica del negocio de manera automática.

Para el caso anterior se tienen las carpetas: *arreglistas*, *bandeja_men*, *compositores*, *epocas*, *formatos*, *generos*, *html*, *instrumentos*, *interpretes*, *materiales*, *musicos*, *programas*, *sellos*, *subgéneros*, *temas* y *zonas*. Estas carpetas contienen los archivos siguiendo el esquema *sem_<utilidad>.php*, el cual es un esquema predecible para realizar operaciones en la lógica del sistema.

Dentro de la carpeta *esp*, se encuentra una serie de archivos de uso común dentro del sistema.

Archivo	Uso
au.php	Configura y realiza las llamadas a la capa de datos para realizar actualizaciones sobre las tablas del sistema, en especial sobre aquellas que contienen la información base de SEMCO.
b.php	Configura y llama la capa de datos para realizar operaciones de borrado de registros en las diferentes secciones del sistema.
bandeja.php	Muestra un listado de los mensajes enviados al usuario que actualmente sea el usuario activo.
entrada_bandeja.php	Presenta un formulario para que los usuarios autorizados envíen mensajes a los demás usuarios del sistema.
entrada_bandeja_a.php	Realiza la llamada a la capa de datos para enviar el mensaje a los usuarios seleccionados en la anterior página.
index.php	Es una página de marcos que llama al menú del sistema y establece la navegabilidad del sistema.
inter.php	Esta página genera dinámicamente (basándose en la información contenida en la tabla <i>conftablas</i>) la presentación para tablas intermedias del sistema; es decir, presenta un mecanismo para asociar información para las tablas intermedia del sistema, las cuales caracterizan a los temas almacenados y administrados por el sistema.
inter_au.php	Hace el llamado a la capa de datos para ejecutar las

	acciones seleccionadas en la página Inter.php.
intro.php	Contiene una serie de gráficos que identifican el sistema de información.
l.php	Genera dinámicamente los listados para las tablas que se tengan configuradas en conftablas. Esta página genera los encabezados, el listado de registros y la paginación de resultados, y tiene enlaces que conducen al usuario a los detalles de cada registro.
login.php	Presenta un formulario para que los usuarios entren su nombre de usuario único y su clave para ingresar al sistema.
log_auth.php	Está diseñado para realizar las operaciones de autenticación de usuarios del sistema. Una vez identificado el usuario esta página envía el usuario a la página <i>index.php</i> .
menu.php	Contiene una secuencia de comandos que permite al usuario navegar por las diferentes opciones del sistema de información.
sem_búsqueda.php	Presenta un formulario para la entrada de criterios de búsqueda y una vez digitados, un listado con los resultados de la misma.
sem_auto.php	Esta página implementa la funcionalidad que evita la entrada de registros repetidos; busca interactivamente usando los textos que entra el usuario para determinar si el registro ya existe.
semco.php	Es una página de marcos que alberga la página de entrada de datos (por ejemplo, temas), y la página sem_auto.php lo cual complementa la funcionalidad de búsqueda automática.
top.php	Es la página ubicada en la parte superior de la diagramación general del sistema, allí se encuentran los gráficos que identifican el software ante el usuario final.

ANEXO C. INTERFASES DE USUARIO

- Formato de creación y/o actualización de usuarios

Seleccione Usuario <input type="text" value="admin - Claudia Mantilla"/>			
INFORMACIÓN PERSONAL			
Nombre	<input type="text" value="Claudia"/>	Apellido	<input type="text" value="Mantilla"/>
Documento	<input type="text" value="63488212"/>	Tipo de documento	<input type="text" value="C.C."/>
Sexo	<input type="text" value="Femenino"/>	Teléfono	<input type="text" value="6520769"/>
Celular	<input type="text" value="0"/>	E-mail	<input type="text"/>
Dirección	<input type="text" value="carrera 38# 41-47"/>		
INFORMACIÓN DEL USUARIO			
Nombre de Usuario	<input type="text" value="admin"/>	Tipo de Usuario	<input type="text" value="Administrador"/>
Clave	<input type="password" value="*****"/>	Confirmar Clave	<input type="password" value="*****"/>
<input type="button" value="Actualizar"/>		<input type="button" value="Borrar"/>	

Permite modificar la información de un usuario ya registrado o incluir uno Nuevo, este formato es sólo desplegado al usuario administrador.

- Formato de ingreso de material nuevo

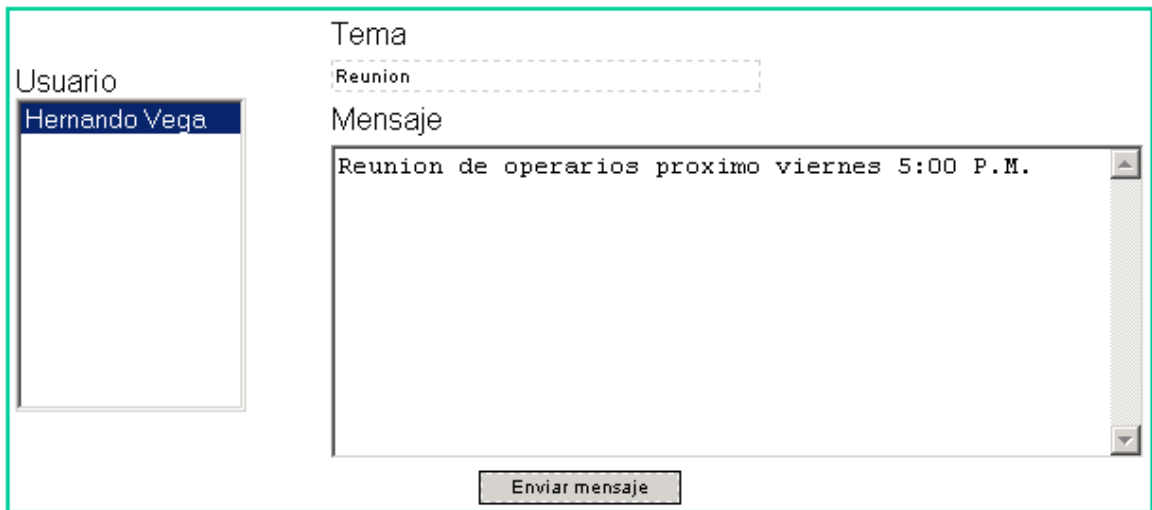
NOMBRE	
Nombre	<input type="text" value="Jazz World 05/95"/>
Número de Inventario	<input type="text" value="ACD-01"/>
Año	<input type="text" value="1995"/>
Autor	<input type="text" value="RFI"/>
Sello	<input type="text" value="RFI"/>
Zona	<input type="text" value="Zona A5"/>

[Asociaciones](#)

SEMCO

Permite registrar la información relevante del material fonográfico, actualizar o borrar la de alguno ya existente.

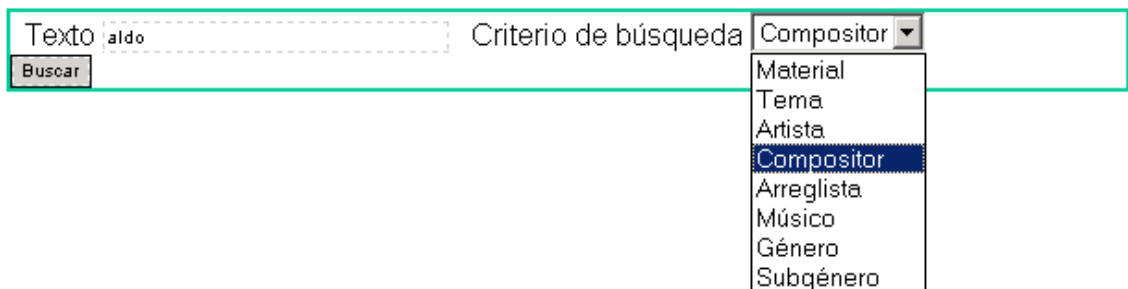
- Formato de envío de mensajes



The screenshot shows a web interface for sending messages. On the left, there is a list box labeled 'Usuario' with 'Hernando Vega' selected. To the right, there is a 'Tema' field containing the text 'Reunion'. Below that is a larger text area labeled 'Mensaje' containing the text 'Reunion de operarios proximo viernes 5:00 P.M.'. At the bottom center, there is a button labeled 'Enviar mensaje'.

Permite enviar mensajes a otros usuarios autorizados del sistema eligiendo en la lista de usuarios aquel a quien se quiera enviar dicho mensaje, el nombre del usuario actual no aparece en el listado.

- Búsqueda de temas según parámetros



The screenshot shows a search interface. It features a text input field containing the word 'aldo' and a 'Buscar' button. To the right of the input field is the label 'Criterio de búsqueda' and a dropdown menu. The dropdown menu is open, showing a list of search criteria: Material, Tema, Artista, Compositor (which is highlighted), Arreglista, Músico, Género, and Subgénero.

En el cuadro de texto se escribe lo que se quiere buscar y se elige el criterio bajo el cual se va a efectuar dicha búsqueda.

- Formato de resultados

Texto	aldo	Criterio de búsqueda	Compositor
Buscar			

TEMA	MATERIAL	# INVENTARIO	ZONA	VER
Silenzio	Jazz World 05/95	ACD-01	7	Ver

Resultado de la búsqueda indicando el tema relacionado con el compositor solicitado, el material donde se encuentra, su número de inventario, su zona de localización física y un link para ampliar información.

Información detallada de la búsqueda (modo modificación)

TEMAS
ACTUAL: SILENZIO

Modo Texto Modo Modificación

Músicos ▾

Gilbert Pereira
 Jean-Marc Phillips
 Louis Sclavis
 Daniel Goyone
 Olivier Ker Ourio
 Paolo Fresu
 Stefano Di Batista
 Franco D'Andrea
 Jean-Michel Pilo
 Michel Benita
 Furio Di Castri
 Aldo Romano

[Agregar Músicos](#)
 Actualizar

Es permitido sólo al administrador modificar la información alusiva a la búsqueda realizada, agregar o eliminar información.

- Información detallada de la búsqueda (modo texto)

TEMAS
ACTUAL: SILENZIO

Modo Texto Modo Modificación

Músicos ▾

MÚSICOS

Olivier Ker Ourio
Paolo Fresu
Stefano Di Batista
Franco D'Andrea
Jean-Michel Pilo
Michel Benita
Furio Di Castri
Aldo Romano

Posterior a la muestra de resultados se puede obtener información adicional a la búsqueda, en este caso los músicos participantes en el tema compuesto por el compositor en cuestión.