



UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN

**Diseño y Desarrollo de una nueva Aplicación Web  
para la Escuela de Computación de la  
Universidad Central de Venezuela  
con Tecnología Ruby On Rails**

**Trabajo Especial de Grado presentado ante la ilustre  
Universidad Central de Venezuela por las Bachilleres  
Mercy Ospina Torres y Sarabel Suárez Díaz.**

**Para optar al título de Licenciado en Computación  
Tutores: Profa. Yusneyi Y. Carballo Barrera y Prof. Sergio Rivas**

**Caracas, 17 de Octubre de 2008**

---

UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado “Diseño y Desarrollo de una nueva Aplicación Web para la Escuela de Computación de la Universidad Central de Venezuela con Tecnología Ruby On Rails” y presentado por las Brs. Mercy Ospina Torres de C.I. 10.870.485 y Sarabel Suárez Díaz de C.I. 16.902.631, a los fines de optar al título de **Licenciada en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 17 de Octubre de 2008, a las 2:00 pm horas, para que los autores lo defendieran en forma pública, lo que estos hicieron en la Sala PA III de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de \_\_\_\_ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_.

---

**Profa. Yusneyi Carballo Barrera**  
**(Tutora)**

---

**Prof. Sergio Rivas**  
**(Tutor)**

---

Profa. Jossie Zambrano  
(Jurado Principal)

---

Profa. Omaira Rodríguez  
(Jurado Principal)

---

Profa. Concettina Di Vasta  
(Jurado Suplente)

---

Prof. Johnny Sepúlveda  
(Jurado Suplente)

***Dedicatoria y Agradecimientos***

*A Dios quién siempre me ha acompañado en este largo recorrido, me ha dado una vida maravillosa y en los momentos de dificultad me ha llenado de fuerza necesaria para levantarme y continuar.*

*A ti mi hermosa abuelita que siempre estuviste a mi lado consintiéndome en todo lo que te pedía y aunque físicamente no estás siempre te tengo en mi corazón y pensamientos y sé que tú siempre estás a mi lado, te amo de más y extraño como nunca. Y por supuesto a mi abuelo yeyé que aunque te fuiste hace mucho siempre te he tenido en mi corazón y siempre te he amado y extrañado.*

*A mi mamá y a mi papá que me dieron la vida y siempre han estado conmigo en todo momento dando lo máximo para ofrecerme todo lo necesario para una vida feliz, gracias por haberme guiado por el mejor camino, y aunque hemos pasado momentos difíciles siempre he contado con su apoyo y amor. Los amo.*

*A mis hermanas Saray y Sinaí para quienes he tratado de ser el mejor ejemplo a seguir y ahora que comienzan la etapa que yo hoy culmino les brindo todo mi apoyo y amor.*

*A mis tías Ana y Maritza, mis tíos Niño y Nene, a mis primos Alex, Satmar, Abelmar, Marvic, Zarozky, Wladimir, Stefanie, Gamaliel y Rafael todos representan personas importantes, queridas y siempre han estado conmigo.*

*A todos mis niños consentidos que me llenan de alegría en todo momento Carimar, Hemyerli, Samantha, Javianny, Moises, Abel Alejandro y Mónica.*

*A Pedro Gil quien estuvo a mi lado durante toda la carrera y se convirtió en una persona muy especial que siempre me apoyó en todo lo que necesitaba y pesar de las dificultades sé que siempre estarás ahí al igual que yo estaré aquí.*

*A mis dos amigas de toda la vida Gaby y Jessica, gracias por siempre estar conmigo y escucharme en todo momento. Omar, Fernando, Luis Mora, Bianysa, Raúl y Vladimir mis compañeros y amigos de la carrera.*

*Por supuesto Mercy que no sólo eres mi compañera de tesis, te convertiste en una gran amiga que me ayudó, aconsejó y siempre estuviste en los momentos más difíciles Te Quiero Mucho, gracias por todo Amiga. También a tus hijos Kevin, Luis David y a tu esposo Luis que me recibieron muy bien en casa, los voy a extrañar a todos.*

*Y por supuesto a MÍ. La alegría que siento es inmensa estoy logrando una de mis más grandes metas.*

*Son muchas las personas que me han acompañado a muchas no he nombrado pero a todos quiero decirles que sin ustedes a mi lado no lo hubiera logrado, les doy las gracias de todo corazón por haber llegado a mi vida y por compartir momentos agradables y tristes, que me han dejado un gran aprendizaje, me han hecho crecer y valorar a todas las personas que me rodean, los quiero mucho y siempre los llevaré conmigo.*

*Sarabel Dayana Suárez Díaz*

### ***Dedicatoria***

*A Dios por darme esta maravillosa oportunidad de vivir y poner en mi camino a personas que de una u otra manera me han brindado su guía, apoyo y comprensión durante los momentos difíciles de la carrera y de mi vida.*

*A Luis, Kevin y Luis David, mi esposo e hijos que con su amor, apoyo y comprensión hicieron posible que lograra este sueño, los amos de corazón.*

*A mi madre por su bendición, por inculcarme su amor al estudio y su espíritu de superación, por creer en mi te amo.*

*A mi hermana Ahimara y mis sobrinos David, Dennis, Darwin, Thaiz, Ana, Angélica y Christian, para los que he tratado de ser un ejemplo de vida, y me han enseñado la importancia de la familia en cada paso de la vida, los quiero mucho y con esta etapa que hoy culmino les ofrezco todo mi apoyo y amor.*

### ***Agradecimientos.***

*En este momento siento un gran agradecimiento hacia muchas personas que dedicaron parte de su tiempo a apoyarme, guiarme y darme ánimo en los momentos difíciles.*

*A mi esposo Luis y mis hijos Kevin y Luis David, a Thaiz, Ahimara y Nelly mi pequeña gran familia que fueron un pilar fundamental en mi vida me tendieron la mano, me ayudaron y escucharon cuando más lo necesite.*

*A Loli y su familia, a Dolly, Elias, Daniel y mis suegros que supieron entenderme, me apoyaron y ayudaron cuando fue necesario.*

*A mis amigos Bianysa, Luis Mora, Maryoli, Carlos Luis y Manuel Dorado quienes me brindaron su amistad cuando entre en la Universidad y no conocía a nadie, y a Vladimir y Raúl, todos mis grandes amigos quienes me enseñaron el valor de la amistad, de la solidaridad y de la risa, siempre recordaré los buenos momentos que vivimos ya que son únicos e irrepetibles.*

*Un especial agradecimiento a Sara mi compañera de Tesis que más que compañera se convirtió en una gran amiga, casi mi hermana, y gran una aliada en la consecución de este logro, con quien compartí los mejores momentos y supo darme ánimo para continuar cuando nos sentíamos más agotadas y decaídas.*

*Y para finalizar a mí, por mi constancia y dedicación, ya que siento este logro como parte importante de mi crecimiento personal y profesional, que me ha dado aprendizajes de vida esenciales para ser una mejor persona.*

*Son muchas las personas que no he nombrado, pero que llegaron a mi vida y me brindaron su amistad, compartiendo conmigo momentos alegres y tristes, que me hicieron crecer como persona, a todos ellos gracias.*

*Mercy H. Ospina Torres*

### *Agradecimientos*

*Queremos dar gracias muy especiales a:*

*Nuestra tutora la profesora Yusneyi Carballo, quién nos brindó toda su ayuda y orientación en todo momento y que con sus consejos fue mucho más que una tutora, ya que nos motivó a dar lo mejor de nosotras.*

*Nuestro tutor el profesor Sergio Rivas, siempre muy atento a nuestras dudas, formándonos y dándonos las herramientas para solucionar muchos de los problemas que se nos presentaron.*

*La profesora Omaira Rodríguez de quien nació este proyecto, fue parte activa en cada una de sus etapas y confió en que su equipo de desarrollo haría el mejor trabajo posible.*

*La profesora Jossie Zambrano, por su aporte en distintas áreas del proyecto y por su guía y consejos, a la bebe por esperar, gracias a las dos.*

*Nuestra profe Tina Di Vasta, excelente persona y educadora, siempre pendiente de nosotras y dándonos su ayuda y conocimientos en todo lo que necesitáramos.*

*Todo el personal del Ceneac, el profesor Jhonny, Luis, Silvia y Mario.*

*Daniel Villavicencio quien nos brindó su ayuda para llevar la aplicación a un Servidor del Centro de Computación.*

*Sarabel Suárez y Mercy Ospina*

## **RESUMEN**

Este trabajo de la Facultad de Ciencias de la Universidad Central de Venezuela tiene como objetivo el desarrollo de una aplicación Web para la Escuela de Computación que sirva como medio de difusión de información institucional, académica y de investigación, accesible y de fácil navegación, usando estándares de diseño Web actuales y que cuente con funcionalidades que faciliten la actualización de su contenido. Debido a la naturaleza de la aplicación, donde unas funcionalidades deben ser del dominio público y otras sólo disponibles para usuarios autorizados, se sugirieron dos subsistemas denominados: subsistema público y subsistema administrativo.

Para llevar a cabo este objetivo se aplicaron herramientas y tecnologías de desarrollo Web tal como Ruby on Rails, Hojas de estilo en cascada, Lenguaje Extensible de Marcado de Hipertexto, en el caso de las funcionalidades se uso la metodología de desarrollo Proceso Unificado Ágil adaptada a las necesidades de la aplicación, y para el diseño se siguieron aquellos estándares actuales de diseño Web, lineamientos de usabilidad y patrones de interacción, considerados más importantes.

**Palabras Claves:** UCV, Escuela de Computación, Aplicación Web, tecnologías de desarrollo Web, Ruby on Rails, Modelado ágil. Metodología de desarrollo AUP, estándares de diseño Web, usabilidad, patrones de interacción.

### **Contactos:**

- Profa. Yusneyi Carballo B., correo-e: [yusneyi.carballo@ciens.ucv.ve](mailto:yusneyi.carballo@ciens.ucv.ve)
- Prof. Sergio Rivas, correo-e: [sergio.rivas@ciens.ucv.ve](mailto:sergio.rivas@ciens.ucv.ve)
- Mercy Ospina, correo-e: [mercy05@gmail.com](mailto:mercy05@gmail.com)
- Sarabel Suárez: correo-e: [sarabel.p@gmail.com](mailto:sarabel.p@gmail.com).

## Índice

<b>INTRODUCCIÓN</b>	<b>- 1 -</b>
<b>CAPÍTULO 1. PRESENTACIÓN DEL TRABAJO ESPECIAL DE GRADO</b>	<b>- 3 -</b>
<b>1.1 Planteamiento del Problema</b>	<b>- 3 -</b>
<b>1.2 Solución a Desarrollar</b>	<b>- 3 -</b>
<b>1.3 Objetivo General</b>	<b>- 4 -</b>
<b>1.4 Objetivos Específicos</b>	<b>- 4 -</b>
<b>1.5 Definición del Alcance y Restricciones</b>	<b>- 5 -</b>
<b>CAPÍTULO 2. MARCO TEÓRICO CONCEPTUAL</b>	<b>- 6 -</b>
<b>2.1 Aplicaciones Web</b>	<b>- 6 -</b>
2.1.1 Definición de Aplicaciones Web	- 6 -
2.1.2 Arquitectura Cliente-Servidor	- 6 -
<b>2.2 Arquitectura de Tres Capas (MVC y Model2)</b>	<b>- 7 -</b>
2.2.1 Patrón de arquitectura MVC y Model2	- 9 -
<b>2.3 Herramientas Tecnológicas para el desarrollo de Aplicaciones Web</b>	<b>- 10 -</b>
2.3.1 Tecnologías del lado del Cliente	- 11 -
2.3.1.1 HTML (Lenguaje de Marcado de Hipertexto)	- 11 -
2.3.1.2 XHTML (Lenguaje Extensible de Marcado de Hipertexto)	- 12 -
2.3.1.3 Hojas de Estilo en Cascada (CSS)	- 13 -
2.3.1.4 JavaScript	- 14 -
2.3.2 Tecnologías del lado del Servidor	- 15 -
2.3.2.1 Ruby on Rails	- 15 -
2.3.2.2 MySQL 5.0	- 19 -
2.3.2.3 Subversion	- 20 -
2.3.3 Servidores Web	- 23 -
2.3.3.1 Apache	- 23 -
2.3.3.2 Mongrel	- 24 -
<b>CAPÍTULO 3. MARCO METODOLÓGICO</b>	<b>- 26 -</b>
<b>3.1 Modelado Ágil</b>	<b>- 26 -</b>
<b>3.2 Metodología de desarrollo AUP</b>	<b>- 28 -</b>

---

<b>3.3</b>	<b>Lineamientos de Usabilidad</b>	<b>- 33 -</b>
3.3.1	Navegación o Navegabilidad	- 33 -
3.3.2	Elementos de Contenido	- 35 -
3.3.3	Diseño	- 36 -
<b>3.4</b>	<b>Estándares Web</b>	<b>- 38 -</b>
<b>3.5</b>	<b>Patrones de Diseño de Interacción</b>	<b>- 43 -</b>
<b>CAPÍTULO 4. DESARROLLO DE LA APLICACIÓN</b>		<b>- 46 -</b>
<b>4.1</b>	<b>Objetivo general de la aplicación</b>	<b>- 46 -</b>
<b>4.2</b>	<b>Objetivos específicos de la aplicación</b>	<b>- 46 -</b>
<b>4.3</b>	<b>Alcance de la aplicación</b>	<b>- 46 -</b>
<b>4.4</b>	<b>Características de las Aplicaciones Web Universitarias</b>	<b>- 47 -</b>
4.4.1	Navegación	- 47 -
4.4.2	Contenido	- 47 -
4.4.3	Diseño	- 48 -
<b>4.5</b>	<b>Adaptación de la metodología AUP</b>	<b>- 49 -</b>
4.5.1	INICIO	- 50 -
4.5.2	ELABORACIÓN:	- 63 -
4.5.3	CONSTRUCCIÓN:	- 81 -
4.5.3.1	Subsistema Público:	- 82 -
4.5.3.2	Subsistema Privado	- 94 -
4.5.4	TRANSICIÓN:	- 101 -
<b>CONCLUSIONES</b>		<b>- 112 -</b>
<b>RECOMENDACIONES</b>		<b>- 114 -</b>
<b>APORTES</b>		<b>- 114 -</b>
<b>REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES</b>		<b>- 116 -</b>



## Índice de Figuras

<i>Figura 2. 1. Arquitectura Cliente-Servidor</i>	- 7 -
<i>Figura 2. 2. Arquitectura de Tres Capas</i>	- 8 -
<i>Figura 2. 3. Arquitectura de tres capas para Aplicaciones Web</i>	- 10 -
<i>Figura 3. 1. Modelado Ágil refuerza Otros procesos de Software</i>	- 26 -
<i>Figura 3. 2. Ciclo de vida de la AUP</i>	- 28 -
<i>Figura 3. 3. Ciclo de vida del Desarrollo (AM)</i>	- 30 -
<i>Figura 3. 4. Ciclo de vida de las pruebas orientadas a objetos (FLOOT)</i>	- 31 -
<i>Figura 3. 5. Crecimiento de las versiones en el tiempo</i>	- 32 -
<i>Figura 3. 6. Ejemplo de Fly out menu o menú desplegable</i>	- 44 -
<i>Figura 3. 7. Ejemplo del patrón Search Area</i>	- 44 -
<i>Figura 3. 8. Ejemplo del Patrón Login</i>	- 45 -
<i>Figura 3. 9. Ejemplo del patrón Event-Calendar</i>	- 45 -
<i>Figura 4. 1. Principales Usuarios Subsistema Público</i>	- 51 -
<i>Figura 4. 2. Principales usuarios del subsistema administrativo.</i>	- 51 -
<i>Figura 4. 3. Perfil de Usuarios</i>	- 54 -
<i>Figura 4. 4. Diagrama de casos de uso Nivel 0</i>	- 54 -
<i>Figura 4. 5. Diagrama Casos de Uso Subsistema Público. Nivel 1</i>	- 55 -
<i>Figura 4. 6. Diagrama Casos de Uso Subsistema Administrativo. Nivel 1</i>	- 57 -
<i>Figura 4. 7. Modelo de Objeto de Dominio</i>	- 59 -
<i>Figura 4. 8. Arquitectura inicial</i>	- 60 -
<i>Figura 4. 12. Diagrama de Casos de Uso Ver Información General. Nivel 2</i>	- 64 -
<i>Figura 4. 13. Diagrama de Casos de Uso Ver Información de la Comunidad de la Escuela. Nivel 2</i>	- 66 -
<i>Figura 4. 14. Diagrama de Casos de Uso Ver Información Académica. Nivel 2</i>	- 68 -
<i>Figura 4. 15. Diagrama de Casos de Uso Ver Investigaciones. Nivel 2</i>	- 69 -
<i>Figura 4. 16. Diagrama de Casos de Uso Ver Noticias y Eventos. Nivel 2</i>	- 70 -
<i>Figura 4. 17. Diagrama de Casos de Uso Publicar Documentos. Nivel 2</i>	- 72 -
<i>Figura 4. 18. Diagrama de Casos de Uso Publicar Seminarios y Trabajos Especiales de Grado. Nivel 2</i>	- 73 -
<i>Figura 4. 19. Diagrama de Casos de Uso Publicar Ofertas de Trabajo y Pasantías. Nivel 2</i>	- 74 -
<i>Figura 4. 20. Diagrama de Casos de Uso Publicar Noticias y Eventos. Nivel 2</i>	- 76 -
<i>Figura 4. 21. Interfaz Página Principal</i>	- 79 -
<i>Figura 4. 22. Interfaz Página Secundaria. Página de Materias</i>	- 79 -
<i>Figura 4. 23. Diagrama de Clases Persistentes</i>	- 80 -

---

Figura 4. 24. Cronograma de desarrollo	- 82 -
Figura 4. 25. Diagrama de Clases Persistentes Iteración 1	- 83 -
Figura 4. 26. Interfaz Autoridades	- 83 -
Figura 4. 27. Interfaz Asesoría Académica	- 84 -
Figura 4. 28. Diagrama de Clases Persistentes Iteración 2	- 84 -
Figura 4. 29. Interfaz Vista Eventos del Día	- 85 -
Figura 4. 30. Interfaz Vista Noticia	- 85 -
Figura 4. 31. Diagrama de Clases Persistentes Iteración 3	- 86 -
Figura 4. 32. Controlador asociado a la funcionalidad Comunidad	- 87 -
Figura 4. 33. Código asociado a la Oferta de Trabajos y Pasantías	- 87 -
Figura 4. 34. Interfaz Ofertas de Pasantías y Trabajos	- 88 -
Figura 4. 35. Interfaz Vista Directorio	- 88 -
Figura 4. 36. Diagrama de Clases Persistentes Iteración 3	- 89 -
Figura 4. 37. Controladores asociados a las funcionalidades Ver Información Académica y Documentos	- 89 -
Figura 4. 40. Interfaz Vista Materias y Programación Docente	- 91 -
Figura 4. 41. Interfaz Vista Documentos	- 91 -
Figura 4. 42. Submenú de la opción Investigación	- 92 -
Figura 4. 43. Diagrama de Clases Persistentes Iteración 4	- 92 -
Figura 4. 44. Interfaz Vista Centros de Investigación y Laboratorios	- 93 -
Figura 4. 45. Interfaz Vista T.E.G. Culminados	- 93 -
Figura 4. 46. Diagrama de Clases Persistentes Iteración 6	- 94 -
Figura 4. 47. Interfaz Página de autenticación Subsistema Administrativo	- 95 -
Figura 4. 48. Interfaces Cambio de Contraseña y Actualizar Datos Personales	- 95 -
Figura 4. 49. Diagrama de Clases Persistentes Iteración 7	- 96 -
Figura 4. 50. Interfaz Listar Documentos	- 96 -
Figura 4. 51. Diagrama de Clases Persistentes Iteración 8	- 97 -
Figura 4. 52. Interfaz Editar seminario	- 97 -
Figura 4. 53. Diagrama de Clases Persistentes Iteración 9	- 98 -
Figura 4. 54. Interfaz Editar Oferta de Trabajo	- 98 -
Figura 4. 55. Diagrama de Clases Persistentes Iteración 10	- 99 -
Figura 4. 56. Interfaz Nueva Noticias	- 99 -
Figura 4. 57. Diagrama de Clases Persistentes Iteración 11	- 100 -
Figura 4. 58. Diagrama de Despliegue	- 102 -
Figura 4. 59. Modelo de la Encuesta vía Web.	- 103 -
Figura 4. 60. Ubicación del Menú	-104 -
Figura 4. 61. Gráfico Cambiar Opciones	- 104 -
Figura 4. 63. Gráfico Facilidad de Acceso a las Opciones	- 104 -
Figura 4. 64. Gráfico Cantidad de Opciones del Menú	- 105 -

*Figura 4. 65. Comentarios Navegación* \_\_\_\_\_ - 105 -

*Figura 4. 67. Gráfico Calidad de la Información* \_\_\_\_\_ - 106 -

*Figura 4. 68. Gráfico Tamaño de la Letra* \_\_\_\_\_ - 106 -

*Figura 4. 69. Comentarios Contenido* \_\_\_\_\_ - 107 -

*Figura 4. 72. Gráfico Colores Usados* \_\_\_\_\_ - 107 -

*Figura 4. 73. Comentarios Diseño* \_\_\_\_\_ - 109 -

## ***INTRODUCCIÓN***

La sociedad actual se ha visto influenciada por el uso de Internet, donde un creciente número de organizaciones cuentan con aplicaciones Web que les permiten difundir su información institucional y dar a conocer sus actividades a un grupo de usuarios en crecimiento como son los usuarios Web. Estas aplicaciones son conocidas como Sitio Web y su calidad es muy variada, considerándose más exitosas aquellas que presentan contenidos relevantes y útiles, que son constantemente actualizados, y que facilitan la visita de los usuarios aplicando estándares de usabilidad y diseño Web.

Al ser la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela una institución pionera en el uso de tecnologías de Internet se considera necesario que cuente con un Sitio Web de calidad que refleje sus actividades y procesos, y que muestre información actualizada y de interés para sus usuarios.

Para poder mantener dicha información actualizada esta aplicación debe contar con funcionalidades que permitan la actualización de manera fácil y segura, para que este proceso pueda llevarse a cabo por el personal que conforma la Escuela de Computación, es decir, personal que no necesariamente está capacitado en el desarrollo de aplicaciones Web, lo que garantizaría la actualización constante y fomentaría la participación de la comunidad de la Escuela en el mantenimiento de su Sitio Web adquiriendo de esta manera un sentido de pertenencia más amplio.

En tal sentido, el objetivo fundamental de este Trabajo Especial de Grado consiste en la implementación de una nueva aplicación Web para la Escuela de Computación que tenga como función difundir su información institucional, académica, de investigación y administrativa. Esta aplicación se denominará el Sitio Web de la Escuela de Computación y estará conformada por dos subsistemas: un primer subsistema denominado “Subsistema Público” que contará con funcionalidades para la consulta de la información antes mencionada, y un segundo subsistema, denominado “Subsistema Administrativo” que contará con funcionalidades de administración que permitan el manejo y la actualización, de manera fácil y segura, de la información mostrada por el Subsistema Público.

La nueva aplicación Web a implementar debe manejar diversas informaciones asociadas a la Escuela de Computación por lo que puede llegar a ser muy compleja. Esto hace necesario realizar un diseño estructurado que permita el desarrollo incremental de funcionalidades que se puedan ir incorporando a

ambos subsistemas, y utilizar un proceso de desarrollo ágil que tome en cuenta requerimientos cambiantes y un desarrollo rápido, iterativo y eficiente.

Para mejorar la comprensión del lector, este documento se ha estructurado en los siguientes capítulos:

**CAPÍTULO 1: PRESENTACION DEL TRABAJO ESPECIAL DE GRADO.** Describe el problema y la solución planteada, además se definen el objetivo general, los objetivos específicos, el alcance y las restricciones del presente trabajo de investigación.

**CAPÍTULO 2, MARCO TEÓRICO:** presenta los fundamentos conceptuales que sustentan el trabajo de investigación y la presentación de cada una de las tecnologías usadas para crear la aplicación Web.

**CAPÍTULO 3, MARCO PROCEDIMENTAL:** presenta la metodología de desarrollo usada, los lineamientos de diseño y los patrones de interacción utilizados para la solución.

**CAPÍTULO 4, MARCO APLICATIVO:** presenta la adaptación del Proceso de Desarrollo AUP al problema particular de investigación, detallando las fases de Inicio, Elaboración, Construcción y Transición, además de cada una de las iteraciones definidas para dicho proceso.

Finalmente se presentan las Conclusiones, Recomendaciones para futuros trabajos relacionados y las Referencias Bibliográficas y electrónicas consultadas durante la investigación.

## **CAPÍTULO 1.**

### **PRESENTACIÓN DEL TRABAJO ESPECIAL DE GRADO**

#### **1.1 Planteamiento del Problema**

Hoy en día se considera necesario para una organización contar con un Sitio Web que le permita darse a conocer, informar sobre sus actividades y en general, servir de soporte a sus procesos, y que además presente características de diseño que haga fácil y agradable la visita de los usuarios. El Sitio Web anterior de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela no cumple estos requisitos por las siguientes razones:

- Sus contenidos se encuentran desactualizados, algunos más de un año.
- Hace difícil la navegación a los usuarios debido a que:
  - No tiene un área de navegación definida.
  - Algunos enlaces no funcionan correctamente.
  - No se muestra claramente la estructura del sitio, debido a que los enlaces existentes no se muestran organizados o clasificados.
- No tiene información sobre las actividades desarrolladas en la escuela como eventos, charlas o conferencias, proyectos de investigación.
- Tiene escasa información sobre los procesos administrativos y/o docentes de la Escuela como solicitudes, normativas, grupos docentes de cada materia, así como información institucional como las autoridades, las comisiones, los centros y laboratorios de investigación, entre otros.
- El diseño no cumple con los estándares actuales de diseño y de usabilidad para el Web.

Los problemas descritos han hecho que este sea un Sitio que no representa una referencia apropiada para la Escuela de Computación.

#### **1.2 Solución a Desarrollar**

Considerando las deficiencias del sitio Web anterior, se observó la imperiosa necesidad de desarrollar e implementar un nuevo Sitio Web para la Escuela de Computación con las siguientes características:

- Este debe reflejar las actividades y procesos actuales que se llevan a cabo en la Escuela.

- Se debe tomar en cuenta que el público al que va dirigido el Sitio Web es muy variado y considerar las necesidades de los distintos usuarios con relación a la Escuela de Computación.
- Se debe observar como se han implementado aplicaciones similares.
- Se deben estudiar con detalle los elementos de diseño que permitan que la aplicación sea fácil de usar, agradable y que responda a los requerimientos de los usuarios.
- La información del Sitio debe poder ser actualizada por usuarios no especializados, de ser posible por la misma comunidad de la Escuela.
- Para el desarrollo se requiere que la metodología a aplicar permita una implementación rápida, sencilla, eficiente, con suficiente documentación y un desarrollo incremental que permita posteriores ampliaciones del sistema.
- Así mismo es importante que la tecnología a utilizar para el desarrollo permita el uso de metodologías ágiles.

### **1.3 Objetivo General**

Este Trabajo Especial de Grado tiene como objetivo la planificación, desarrollo e implementación de una nueva aplicación Web para la Escuela de Computación que sirva como medio de difusión de información institucional, académica, de investigación y administrativa relacionada con esta institución, accesible y de fácil navegación, desarrollada a partir de estándares de diseño Web actuales y que cuente con funcionalidades que permitan la actualización de su contenido de manera fácil y segura.

### **1.4 Objetivos Específicos**

1. Desarrollar una aplicación Web formada por dos subsistemas, un subsistema público que muestre información importante y de interés para el público general, y un subsistema administrativo que cuente con funcionalidades que faciliten la actualización de esta información.
2. Aplicar y adaptar el proceso de desarrollo Unificado Ágil (AUP por sus siglas en inglés) y seguir los principios de la modelación ágil para conseguir un desarrollo dinámico, rápido y evolutivo.
3. Aplicar estándares de diseño Web, lineamientos de usabilidad y patrones de interacción Web para lograr un diseño claro, limpio y que permita al usuario una fácil interacción con el sistema.
4. Llevar la aplicación Web a un servidor de prueba del Centro de Computación y realizar los ajustes necesarios en su configuración para su correcto funcionamiento.
5. Realizar encuestas a los usuarios para obtener su opinión acerca de diferentes aspectos relacionados con el diseño y el contenido de la aplicación.
6. Incorporar las mejoras resultantes del análisis de estas encuestas.

## **1.5 Definición del Alcance y Restricciones**

Se espera que la Aplicación Web a desarrollar tenga un aporte positivo para la Escuela de Computación, ya que permitirá, entre otras acciones:

1. Realizar consultas de los datos del personal de la Escuela de Computación.
2. Conocer la estructura organizativa de la Escuela.
3. Conocer los últimos eventos y noticias referentes a la Escuela o de interés para su comunidad.
4. Conocer ofertas de empleo, pasantías, trabajos de seminario y trabajos especiales de grado.
5. Consultar Normativas, Planillas y Recaudos para solicitudes.
6. Conocer los Centros de Investigación de la Escuela y sus proyectos.
7. Conocer las Comisiones que funcionan en la Escuela.
8. Conocer los recursos y servicios que ofrece la Escuela.

Esta aplicación Web también permitirá la actualización de información por parte del personal de la Escuela a través de un subsistema administrativo que contará con:

1. Autenticación de usuarios.
2. Manejo de funcionalidades de acuerdo al usuario autenticado.
3. Funcionalidades para la actualización de datos, las cuales se encargarán de validarlos y ofrecer al usuario el resultado del proceso.
4. Que toda la información del subsistema público tenga su contraparte en el subsistema administrativo.



## **CAPÍTULO 2.**

### **MARCO TEÓRICO CONCEPTUAL**

#### **2.1 Aplicaciones Web**

##### **2.1.1 Definición de Aplicaciones Web**

Una aplicación Web es un programa que se ejecuta en un servidor Web y que utiliza las páginas Web como la interfaz de usuario. Los usuarios utilizan estas aplicaciones accediendo al servidor Web que las contiene, a través de Internet o de una Intranet.

La característica común que comparten todas las aplicaciones Web es el hecho de centralizar el software para facilitar las tareas de mantenimiento y actualización de grandes sistemas. Cada vez que un usuario desea acceder a la aplicación Web, éste se conecta a un servidor donde se aloja la aplicación. De esta forma, la actualización de una aplicación es sencilla. Simplemente se reemplaza la versión antigua por la versión nueva en el servidor [1].

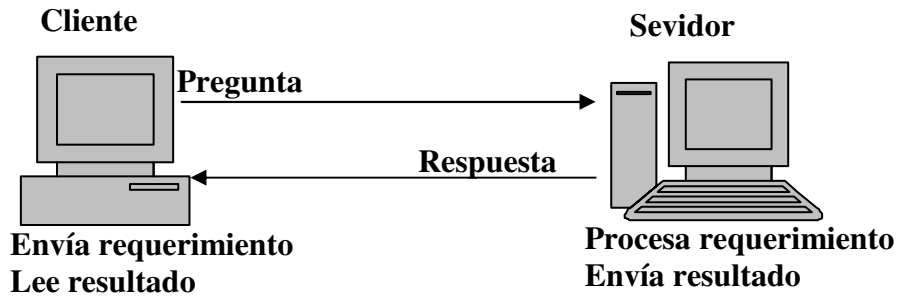
##### **2.1.2 Arquitectura Cliente-Servidor**

Las aplicaciones Web son basadas en la arquitectura Cliente-Servidor [2]. Esta arquitectura trabaja básicamente de forma que el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras [3].

Desde el punto de vista funcional, se puede definir la arquitectura Cliente-Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma [3].

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el

servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario [4].



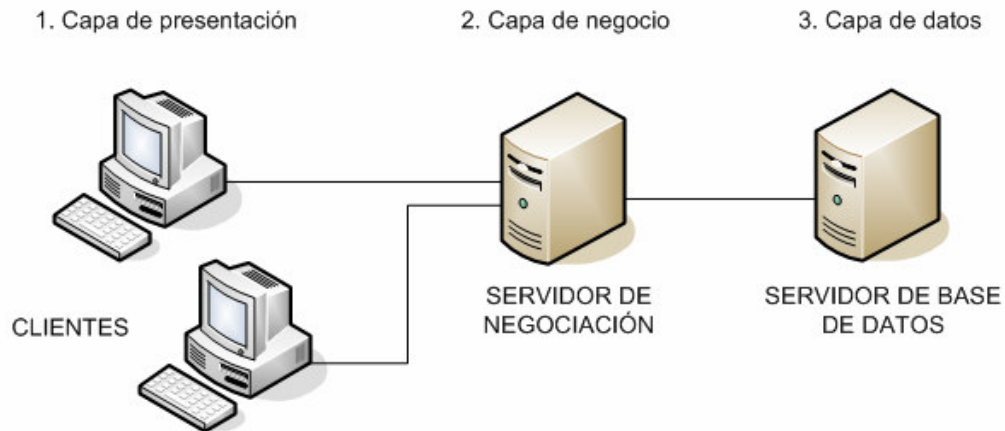
**Figura 2. 1. Arquitectura Cliente-Servidor**

Como se puede ver en la Figura 2-1 en esta arquitectura la computadora de cada uno de los usuarios, llamada Cliente, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente, se conoce como Servidor. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes, y de procesarla según le convenga. Los distintos servidores también pueden intercambiar información dentro de ésta arquitectura.

Los clientes o servidores pueden estar conectados en una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es Internet. Cliente-Servidor es el modelo de interacción más común entre aplicaciones en una red, todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

## **2.2 Arquitectura de Tres Capas (MVC y Model2)**

En la actualidad muchos desarrolladores organizan sus aplicaciones en tres capas como se muestra en la Figura 2-2: Capa de Presentación, Capa de Negocio y Capa de datos. En el contexto de la arquitectura cliente-servidor La Capa de Presentación es la que representa la interfaz con el Cliente. La Capa de Negocios y la Capa de datos están presentes en el Servidor donde se han separado y encapsulado en el procesamiento de los datos por un lado y la lógica de negocios por otro.



**Figura 2. 2. Arquitectura de Tres Capas<sup>1</sup>**

En resumen las responsabilidades de cada capa se distribuyen de la siguiente forma. [ 5]:

- **Capa de Presentación:** Es la responsable de generar la interfaz de usuario con la información provista por la Capa de Negocios. es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema a este, le comunica la información del servidor y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz grafica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.
- **Capa de Negocios:** Es donde residen los programas que se ejecutan, recibe las peticiones del usuario y envía las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.
- **Capa de Datos:** Es responsable de proveer acceso a los datos. Esta formada por uno o más Sistemas Manejadores de Base de Datos que realizan todo el almacenamiento de los datos de la aplicación, y reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocios.

<sup>1</sup> Tomado de [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_tres\\_niveles](http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles)

### 2.2.1 Patrón de arquitectura MVC y Model2

La arquitectura de tres capas se refleja en el patrón de arquitectura MVC (Modelo-Vista-Controlador) desarrollado a mediados de los setenta en la comunidad *Smalltalk*<sup>2</sup> para reducir el acoplamiento, o grado de dependencia entre módulos, entre la lógica de negocios y la de presentación.

Con MVC, la lógica de negocios se coloca en objetos de dominio separados, y la lógica de presentación se aísla en una vista que presenta los datos de los objetos de dominio. El controlador administra la navegación entre vistas, procesa las entradas del usuario y organiza los objetos del dominio adecuados entre el modelo y la vista.

En el entorno de desarrollo Web se utiliza una variante ligeramente distinta de MVC llamada Model2, la cual utiliza los mismos principios pero los ajusta a aplicaciones Web sin estado o “*stateless*”.<sup>3</sup>

La tecnología Web plantea algunos desafíos como lo es el hecho de que no mantiene un estado permanente de conexión entre el Cliente y el Servidor (*stateless*), lo que dificulta que el Modelo pueda notificar a la Vista de la ocurrencia de cambios. El Navegador debe constantemente preguntar al servidor por el estado de la aplicación. [6]

Con Model2 un navegador llama al controlador a través de una solicitud HTTP, el controlador interactúa con el modelo para obtener los datos y validar la entrada del usuario, y pone a disposición los objetos de dominio para su visualización. Luego el controlador llama al generador de vistas adecuado basándose en los resultados de la validación y el URL de la solicitud, luego la capa de vista genera una página Web para el usuario, este proceso se puede observar en la figura 2.3.

---

<sup>2</sup> Smalltalk es un sistema informático que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Ha tenido gran influencia sobre otros lenguajes como *Java* o *Ruby*, y de su entorno han surgido muchas de las prácticas y herramientas de desarrollo promulgadas actualmente por las metodologías ágiles (refactorización, desarrollo incremental, desarrollo dirigido por tests, etc.).

<sup>3</sup> El concepto de “*stateless*” se define como el no poseer información de lo acontecido anteriormente. El “World Wide Web” (www) es intrínsecamente “*stateless*” debido a su forma de operar pues al solicitar una página (solicitud HTTP) el servidor la procesa sin tener conocimiento de la página que proceso anteriormente, es decir no se almacena el estado anterior de la aplicación.

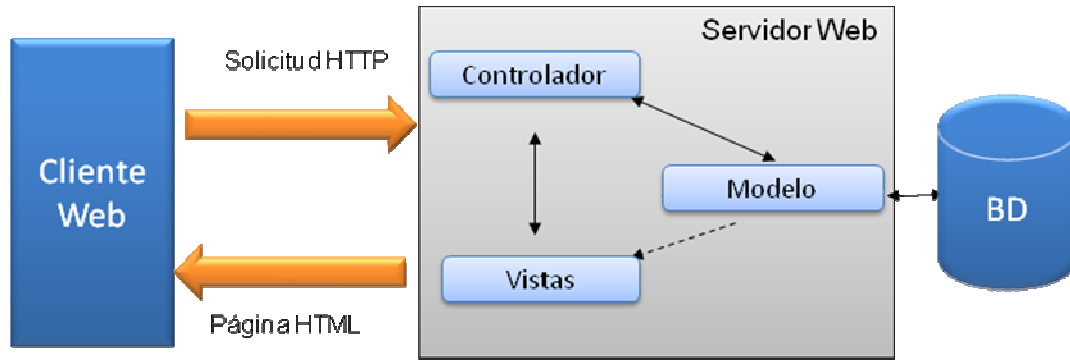


Figura 2. 3. Arquitectura de tres capas para Aplicaciones Web

### 2.3 Herramientas Tecnológicas para el desarrollo de Aplicaciones Web

El desarrollo de una aplicación Web requiere el uso de un conjunto de tecnologías que permitan diseñar y estructurar el contenido de las páginas, implementar las funcionalidades y el dinamismo de las mismas, alojarlas y ponerlas en funcionamiento para su utilización por parte de los usuarios.

Entre estas tecnologías tenemos:

- **Tecnologías del lado del Cliente:** Son las que están insertadas en la página HTML del cliente y son interpretadas y ejecutadas por el Navegador. Estas son utilizadas fundamentalmente para mostrar la información y dar estética al Sitio Web. Las tecnologías utilizadas en el desarrollo de la aplicación Web son HTML, XHTML, CSS y JavaScript.
- **Tecnologías del lado del Servidor:** Permite construir código que se ejecuta en el Servidor Web justo antes de que se envíe la página al cliente a través de la red, existen diversidad de tecnologías del lado del servidor tales como PHP, JSP y Ruby on Rails, en el presente trabajo se destaca Ruby on Rails por ser una tecnología innovadora que facilita la implementación de aplicaciones Web, minimizando el tiempo de desarrollo.
- **Servidores Web:** Es un servidor que proporciona infraestructura y servicios a una aplicación cliente soportando la ejecución y disponibilidad de las aplicaciones desplegadas, gestionando la mayor parte de las funciones de lógica de negocio y de acceso a los datos de dichas aplicaciones, en el presente trabajo se describirán los servidores Apache y Mongrel.

- **Sistemas manejadores de Base de Datos:** Consisten en un conjunto de datos relacionados entre si y un conjunto de herramientas de software o hardware para tener acceso a dichos datos, y a su vez procesarlos y administrarlos. Se estudiará MySQL como sistema manejador de base de datos de software libre, fácilmente integrable con aplicaciones descritas en Ruby on Rails.

### 2.3.1 Tecnologías del lado del Cliente

#### 2.3.1.1 HTML (Lenguaje de Marcado de Hipertexto)

HTML [7] es el acrónimo en inglés de *Hyper Text Markup Language*, es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de 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, este es el formato estándar de las páginas Web. HTML es el estándar usado en la World Wide Web, y se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML describe el aspecto visual que debe tener una página mediante la utilización de etiquetas. Las etiquetas consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en el navegador el texto, las imágenes y los demás elementos, en la pantalla del computador.

Los enlaces, vínculos o hipervínculos (también conocidos como links, su denominación inglesa) son simplemente caminos hacia otras páginas de la World Wide Web. Al pulsar sobre él, instruimos al navegador para abrir la página a la que hace referencia.

Un documento en HTML se compone de dos partes: la cabecera y el cuerpo. La primera se engloba entre las etiquetas `<head>cabecera</head>` y el cuerpo se identifica por estar contenido entre las etiquetas `<body>cuerpo</body>`. De tal forma que la cabecera y cuerpo se enmarcan entre las etiquetas `<html>` y `</html>`.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato como Microsoft Word, Microsoft Wordpad, Notepad. También se pueden utilizar editores específicos que insertan

automáticamente las etiquetas correspondientes al formato del texto que se escriba o a las imágenes que se inserten; casi todos los procesadores de texto actuales incluyen esta posibilidad.

### 2.3.1.2 XHTML (Lenguaje Extensible de Marcado de Hipertexto)

XHTML [8][9] es el acrónimo inglés de eXtensible Hypertext Markup Language, es el lenguaje de marcado creado con el objetivo de sustituir a HTML como estándar para las páginas Web<sup>4</sup>.

Su objetivo es avanzar en el proyecto del *World Wide Web Consortium*<sup>5</sup> de lograr una Web semántica, donde la información, y la forma de presentarla estén claramente separadas. De esta forma XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo y *JavaScript*<sup>6</sup> su aspecto y diseño en distintos medios como lo son, computadoras, PDAs, teléfonos móviles, impresoras, entre otros.

La necesidad de una versión más estricta de HTML se sintió principalmente porque el contenido de la World Wide Web ahora puede visualizarse desde numerosos dispositivos aparte de las computadoras tradicionales, donde no pueden dedicarse recursos suplementarios para afrontar la complejidad añadida de la sintaxis del HTML, por lo tanto XHTML surge como el lenguaje cuyo etiquetado, más estricto que HTML, va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella.

Los aspectos más importantes a tener en cuenta a la hora de utilizar XHTML son los siguientes:

- Los documentos deben estar bien formados: Un formato correcto en un documento XHTML es muy importante. Esto quiere decir que todos los elementos deben tener etiquetas de cierre, deben estar escritos de una forma determinada y además todos los elementos deben estar anidados correctamente.

---

<sup>4</sup> XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML

<sup>5</sup> Para más información leer la sección 3.4.4 del Capítulo III referente a estándares Web

<sup>6</sup> Javascript es un lenguaje de programación Web, tratado con más detalle en la sección 2.3.1.4 de este Capítulo.

- Los nombres de atributos y elementos deben ir en minúsculas: tanto los elementos como los atributos deben ir en minúsculas para todos los elementos HTML y los nombres de atributos. Esto es importante ya que XML interpreta las mayúsculas y las minúsculas de forma diferente.
- Los valores de las etiquetas deben ir siempre entre comillas: todos los valores de los atributos deben ir entre comillas, incluso aquellos que sean numéricos.
- Existen varias versiones de XHTML. Para utilizar una versión concreta, se debe incluir antes del elemento html del código de la página Web, la Definición del Tipo de Documento (DTD) que se pretende utilizar.

### **2.3.1.3 Hojas de Estilo en Cascada (CSS)**

Las hojas de estilo en cascada [10] son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los navegadores.

La idea que se encuentra detrás del desarrollo de las hojas de estilo en cascada es separar la estructura de un documento de su presentación lo que presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

El lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, entre otros.

Las ventajas de utilizar las hojas de estilo en cascada son:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.



- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un Sitio Web, con lo que aumenta considerablemente la accesibilidad.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser leída por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

Las desventajas de utilizar las hojas de estilo en cascada son:

- No todos los navegadores soportan las Hojas de Estilos.
- No existe un estándar general para todos los navegadores, por lo que el resultado del CSS difieren de un navegador a otro.

#### **2.3.1.4 JavaScript**

JavaScript [11] es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página Web. Es un lenguaje de programación interpretado que se ejecuta en el navegador del cliente, sin necesidad de que intervenga el servidor.

Con *JavaScript* se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones *JavaScript* y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Entre las acciones típicas que se pueden realizar en *JavaScript* están los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, *JavaScript* permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

*JavaScript* tiene dos características principales, por un lado que es un lenguaje basado en objetos, es decir, su paradigma de programación es orientado a objetos, y por otro es un lenguaje orientado a eventos, lo que implica que gran parte de la programación en *JavaScript* se centra en describir objetos y escribir funciones que respondan a eventos del usuario, por ejemplo, movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página.

### 2.3.2 Tecnologías del lado del Servidor

#### 2.3.2.1 Ruby on Rails

Ruby on Rails [12] [13] [14] es un entorno de desarrollo Web de código abierto, escrito en lenguaje Ruby y sigue el paradigma de la arquitectura Modelo Vista Controlador (MVC). Ruby on Rails facilita el desarrollo, despliegue y mantenimiento de las aplicaciones Web.

##### *Ruby*

Es un lenguaje de scripts, multiplataforma, totalmente orientado a objetos. Ruby es un lenguaje de programación interpretado y su implementación oficial es distribuida bajo una licencia de software libre.

Entre las características del lenguaje se encuentran:

- Posibilidad de hacer llamadas directamente al sistema operativo.
- Muy potente para el manejo de cadenas y expresiones regulares.
- No se necesita declarar las variables.
- La sintaxis es simple y consistente.
- Gestión de memoria automática.
- Todo es un objeto.
- Métodos Singleton.

##### *Rails*

Es un marco de trabajo de código abierto para Ruby que sirve para desarrollar aplicaciones Web, está basado en el patrón de diseño Modelo Vista Controlador, específicamente en su variante Model2.

Rails está basado en dos principios de desarrollo:

- **Don't Repeat Yourself (no te repitas):** según este principio, ninguna pieza de código debe estar duplicada, esto se debe a que la duplicación incrementa la dificultad en los cambios y la evolución posterior.

Con este principio lo que se busca es que el desarrollo se más rápido y con menos errores, lo que hará que el código sea fácil de entender, mantener y mejorar.

Las definiciones deben hacerse una sola vez, para luego ser reutilizadas de ser necesario. Un ejemplo es al definir un formulario, este puede ser llamado desde cualquier vista y cuantas veces sea necesario con tan solo una línea de código.

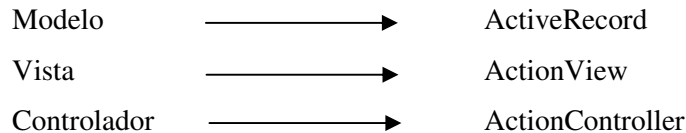
- **Convention Over Configuration (Convención sobre Configuración):** según este principio el seguir las convenciones de Rails se traduce en escribir menos código, ya que solo se definen aquellas configuraciones que no son convencionales.

En lugar de archivos de configuración, Rails utiliza una serie de convenciones simples que le permiten averiguarlo todo mediante reflexión y descubrimiento. De esta forma el código de la aplicación y la base de datos ya contienen todo lo que este necesita saber.

Un ejemplo de este principio es que, si hay una clase Estudiante en el modelo, la tabla correspondiente de la base de datos es estudiantes, pero si la tabla no sigue la convención (por ejemplo alumnos) debe ser especificada manualmente (`set_table_name "alumnos"`).

La arquitectura de Rails trabaja con el patrón MVC, por lo tanto el marco de trabajo (framework) obliga al programador a realizar y mantener de una manera separada los códigos para los modelos, controladores y vistas. Esto se evidencia al momento de crear una aplicación, ya que automáticamente se genera una estructura de directorios muy bien definida que separa y encapsula cada porción de código correspondiente a los componentes del modelo MVC, y otros componentes adicionales como librerías externas, librerías definidas por el desarrollador, archivos de configuración, archivos .js, entre otros.

Al utilizar el entorno de desarrollo Ruby on Rails implementando la arquitectura MVC, la correspondencia que se genera es:



### **Modelo - ActiveRecord**

El Modelo consiste en las clases persistentes (que representan a las tablas de la base de datos). En Ruby on Rails, las clases del Modelo extienden la clase *ActiveRecord*.

Esta clase implementa el patrón de diseño *ActiveRecord* que permite corresponder objetos a relaciones, es decir, provee una interfaz entre las tablas de una base de datos relacional, y el código de la aplicación escrita en ruby que manipula los registros de la base de datos.

Rails añade automáticamente atributos y métodos a la clase basándose en las columnas de la tabla, por lo que automáticamente se corresponden las clases con las tablas y los atributos con las columnas.

### Relaciones *ActiveRecord* [16]

Trabajar con relaciones es una de las tareas más importantes de los marcos de trabajo de persistencia. La idea es manejar las relaciones con un alto rendimiento de cara al usuario final y gran sencillez de cara al desarrollador. *ActiveRecord* aprovecha las convenciones de Ruby para simplificar el acceso a datos relacionados, definiendo las siguientes relaciones:

- *Belongs\_to*: Significa “pertenece a” y representa la relación “de muchos a uno” en el extremo de “muchos”
- *Has\_many*: Representa el lado “uno” de la misma relación anterior, es decir es la otra cara de la relación *belongs\_to*.
- *Has\_one*: Representa la relación “uno a uno”, se puede usar en conjunto con “*belongs\_to*” o con “*has\_one*”
- *Has\_and\_belongs\_to\_many*: Representa la relación de “muchos a muchos” donde hay una tabla adicional con las claves primarias de las tablas relacionadas, pero no hay más datos.

### **Vista – ActionView**

La Vista es la lógica de visualización, o cómo se muestran los datos de las clases del Controlador.

Existe una vista para cada acción del controlador y son éstas las que definen la presentación que tendrán los datos en los archivos HTML que se enviarán al navegador.

El método que se emplea en Rails para gestionar las vistas es usar Ruby Embebido, que son archivos de extensión `.rhtml`, que contienen una combinación de fragmentos de código HTML con código en Ruby.

### ***Controlador – ActionController***

El Controlador corresponde a la interacción del usuario e invocan a la lógica de la aplicación, que a su vez manipula los datos de las clases del Modelo y muestra los resultados usando las Vistas.

El *ActionController* se encuentra entre un *ActiveRecord* (interface para la base de datos) y un *ActionView* (motor de presentación de los datos). El *ActionController* provee facilidades para la manipulación y organización de la data que se extrae de la base de datos, y la que viene como entrada en los formularios Web; para luego redireccionar a la plantilla de vista correspondiente con los datos resultantes del procesamiento previo.

Cada *Controller* es una clase de ruby, y cada método dentro de dicha clase representa un *Action*.

Los *ActionControllers* utilizan las rutas para saber qué controlador se usará, qué método del mismo se ejecutará y qué parámetros se le pasarán. Por defecto, las rutas presentan la siguiente estructura:

`http://host/<controlador>/<método>[/<id>]`

Al crear una nueva aplicación de Ruby on Rails se crearán una serie de directorios y archivos, los cuales representan una jerarquía organizada y predeterminada por Rails para estructurar los proyectos, llamada también Andamiaje (*scaffold*).

Cada directorio cumple una función específica, a continuación se hará la descripción de algunos de estos.

- **App:** Almacena los componentes de la aplicación: vistas y *helpers*, controladores y modelos.
- **Config:** Contiene la configuración de la aplicación: configuración de la base de datos (`database.yml`), estructura del entorno Rails (`environment.rb`) y el enrutador de peticiones Web (`routes.rb`).
- **Public:** es similar al directorio *public* de un Servidor Web. Contiene los archivos *JavaScript*, imágenes, hojas de estilos y código html común para las interfaces.
- **Script:** Este directorio contiene *scripts* que inician y administran las diversas herramientas que se pueden utilizar en Rails; por ejemplo, se tienen los *scripts* que generan código (*generate*) e indican un Servidor Web (Server).

Rails sabe donde encontrar todos los componentes que necesita dentro de la estructura de directorios presentada anteriormente de tal forma que no es necesario especificar detalles del despliegue de la aplicación. El trabajo de los desarrolladores se centra en crear y editar archivos dentro del directorio app, específicamente en los subdirectorios, *controller*, *views* y *model*.

### 2.3.2.2 MySQL 5.0

MySQL [17] es uno de los Sistemas Manejadores de Bases de Datos de software libre más popular, desarrollado, distribuido y soportado por MySQL AB. Esta es una empresa comercial, fundada por los desarrolladores de MySQL.

MySQL es el Sistema Manejador de Bases de Datos (SMBD), probablemente, más usado en el ámbito del software libre, debido a su rapidez y facilidad de uso. Esta aceptación es debida, en parte, a que cuenta con un gran número de librerías y otras herramientas que permiten su uso a través de una variedad de lenguajes de programación, además de su fácil instalación y configuración.

#### *Características [18]*

Las características más importantes de MySQL se describen a continuación:

- Multihilo, núcleo de aplicación completamente multihilo. Puede usar múltiples CPUs, si están disponibles.
- Disponibilidad en diversas plataformas y sistemas operativos (Linux, Windows, Solaris, MacOS X, OS/2, AIX, entre otros).
- Provee motores de almacenamiento que tratan con distintos tipos de tablas, tanto transaccionales como no transaccionales. Entre ellos tenemos MyISAM, Memory, Merge, InnoDB, BDB.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Ruby and TCL.
- El código de MySQL esta comprobado por Purify (herramienta para verificación de problemas de memoria).
- Soporta un ambiente Cliente/Servidor.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible.

### ***Tipos de Datos***

- Tipos Numéricos: Estos tipos incluyen los tipos numéricos exactos (*INTEGER*, *SMALLINT*, *DECIMAL*, y *NUMERIC*), así como los tipos de datos aproximados (*FLOAT*, *REAL*, y *DOUBLE PRECISION*). Tipo *BIT* para almacenar valores de un bit (soportado por las tablas *MyISAM*, *MEMORY*, *InnoDB* y *BBD*). Como extensión del estándar *SQL*, *MySQL* soporta los tipos enteros *TINYINT*, *MEDIUMINT*, y *BIGINT*.
- Tipos de Fecha y hora: Los tipos de fecha y hora para representar valores temporales son *DATETIME*, *DATE*, *TIMESTAMP*, *TIME*, y *YEAR*.
- Tipos de cadenas y caracteres: *CHAR*, *VARCHAR*, *BINARY*, *VARBINARY*, *BLOB*, *TEXT*, *ENUM*, y *SET*.
- Data de Longitud fija y variable.

### ***Sentencias y Funciones***

- Soporte de operadores en la sentencia *SELECT* y la clausura *WHERE*.
- Soporte de clausuras *GROUP BY* y *ORDER BY*, al igual que el grupo de funciones: *count()*, *avg()*, *sum()*, *max*, *min()*, entre otras.
- En esta versión de *MySQL* se incorpora nuevas funcionalidades como los disparadores (*triggers*), procedimientos almacenados (*Stored procedures*) y vistas.

### ***Ventajas de MySQL [19]***

- Mayor velocidad al realizar las operaciones, lo que hace uno de los manejadores que ofrece mayor rendimiento.
- Bajo consumo de memoria, lo que permite ser ejecutado en un computador con escasos recursos.
- Facilidad de configuración e instalación con respecto a la mayoría de los sistemas manejadores de base de datos comerciales existentes.
- No hay límites en el tamaño de registros.

#### **2.3.2.3 Subversion**

Subversion [20] es un sistema de control de versiones libre y de código fuente abierto, el cual maneja archivos y directorios a través del tiempo. Para ello cuenta con un árbol de archivos en un repositorio central. Este repositorio es como un servidor de archivos ordinario, excepto porque recuerda todos los cambios hechos a sus archivos y directorios, esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos computadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración, y la rapidez en el desarrollo, ya que se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y debido a que el trabajo se encuentra bajo el control de versiones, la calidad del mismo no es afectada por la pérdida de ese conducto único, ya que si, por ejemplo, se ha hecho un cambio incorrecto a los datos, simplemente se deshace ese cambio.

### **Control de Versiones**

Una versión, revisión o edición de un producto, es el estado en el se encuentra en un momento dado en su desarrollo o modificación. Se llama **control de versiones** a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico).

### **Características de Subversión**

Subversion proporciona:

1. **Versionado de directorios:** Subversion implementa un sistema de archivos versionado “virtual” que sigue los cambios sobre árboles de directorios completos a través del tiempo. Ambos, archivos y directorios, se encuentran bajo el control de versiones.

2. **Verdadero historial de versiones:** Otros sistemas de control de versiones como CVS está limitado al versionado de archivos, por lo que operaciones como copiar y renombrar (las cuales pueden ocurrir sobre archivos, pero que realmente son cambios al contenido del directorio en el que se encuentran) no son soportadas. Adicionalmente, no se puede reemplazar un archivo versionado con algo nuevo que lleve el mismo nombre sin que el nuevo elemento herede el historial del archivo antiguo, que quizás sea completamente distinto al anterior. Con Subversion, se pueden añadir, borrar, copiar, y renombrar archivos y directorios, y cada archivo añadido comienza con un historial nuevo, limpio y completamente suyo.

3. **Envíos atómicos:** Una colección cualquiera de modificaciones o bien entra por completo al repositorio, o bien no lo hace en absoluto. Esto permite a los desarrolladores construir y enviar los cambios como



fragmentos lógicos e impide que ocurran problemas cuando sólo una parte de los cambios enviados lo hace con éxito.

**4. Versionado de metadatos:** Cada archivo y directorio tiene un conjunto de propiedades (claves y sus valores) asociado a él. Se puede crear y almacenar cualquier par arbitrario de clave/valor. Las propiedades son versionadas a través del tiempo, al igual que el contenido de los archivos.

**5. Elección de las capas de red:** Subversion tiene una noción abstracta del acceso al repositorio, facilitando a las personas implementar nuevos mecanismos de red. Subversion puede conectarse al servidor HTTP Apache como un módulo de extensión. Esto le proporciona una gran ventaja en estabilidad e interoperabilidad, y acceso instantáneo a las características existentes que ofrece este servidor (autenticación, autorización, compresión de la conexión, etc.). También tiene disponible un servidor de Subversion independiente, y más ligero. Este servidor habla un protocolo propio, el cual puede ser encaminado fácilmente a través de un túnel SSH.

**6. Manipulación consistente de datos:** Subversion expresa las diferencias del archivo usando un algoritmo de diferenciación binario, que funciona idénticamente con archivos de texto y archivos binarios. Ambos tipos de archivos son almacenados e igualmente comprimidos en el repositorio, y las diferencias son transmitidas en ambas direcciones a través de la red.

**7. Ramificación y etiquetado eficientes:** El costo de ramificación y etiquetado no necesariamente es proporcional al tamaño del proyecto. Subversion crea ramas y etiquetas simplemente copiando el proyecto, usando un mecanismo similar al enlace duro<sup>7</sup>. De este modo estas operaciones toman solamente una cantidad de tiempo pequeña y constante.

**8. Hackability:** Subversion está implementado como una colección de bibliotecas compartidas en C con APIs bien definidas. Esto hace a Subversion extremadamente fácil de mantener y reutilizable por otras aplicaciones y lenguajes.

### **Ventajas de usar un sistema de control de versiones**

Usar un sistema de control de versiones nos proporciona muchas ventajas:

---

<sup>7</sup> Un 'enlace duro' es una referencia, o indicador a los datos físicos sobre un sistema de archivos. En la mayoría de los sistemas de archivos, todos los archivos son enlaces duros. El nombre de un archivo es simplemente una etiqueta que apunta a los datos reales. Así, se puede asociar más de un nombre (etiqueta) a los mismos datos.

- Poder en todo momento ver cuál es la versión más actual del proyecto, quien y cuándo la ha subido.
- Comparar cualquier par de versiones, dónde vemos todas las partes añadidas, modificadas o eliminadas de los archivos entre las dos versiones.
- Volver atrás en el tiempo a una versión que funcionaba, si los cambios que hemos hecho no nos han llevado por el buen camino.
- Crear distintas ramas del proyecto, para que sigan caminos separados a partir de un punto.
- Etiquetar distintas revisiones que representan una versión oficial del proyecto (para indicar que la revisión 4283 coincide con la versión del proyecto 2.4.0, la etiquetaremos por ejemplo: 2\_4\_0).

### 2.3.3 Servidores Web

#### 2.3.3.1 Apache

Apache es un Servidor de páginas Web de tecnología de código abierto (*open source*) y de libre distribución, para uso comercial y desarrollado por la *Apache Software Foundation* [21].

Este servidor está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos existentes hacen que a menudo sean necesarias diferentes características o funcionalidades o que una misma característica sea implementada de diferentes maneras para obtener mayor eficiencia. Apache se adapta a gran cantidad de entornos gracias a su diseño modular, el cual permite a los diseñadores de sitios Web elegir las funcionalidades que van a ser incluidas en el servidor seleccionando que módulos se van a usar.

La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro de cada módulo. Los módulos de Apache se pueden clasificar en tres categorías:

- **Módulo base:** Módulo que contiene las funciones básicas de Apache.
- **Módulos multiprocesos:** Responsables de la unión con los puertos del computador, aceptando peticiones y enviando hijos a atender las peticiones.
- **Módulos adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor, ejemplos:

- **Mod\_php:** Módulo para la creación de páginas dinámicas en php.
- **Mod\_ruby:** Módulo para la creación de páginas dinámicas en ruby.
- **Mod\_perl:** Módulo para la creación de páginas dinámicas en perl.

#### Características

- Es de fácil configuración a través de un archivo de configuración llamado `httpd.config`.
- Independencia de plataforma. Apache funciona en casi todas las plataformas actuales, lo que permite escoger la plataforma que mejor se adapte a las necesidades y a cambiar de plataforma cuando se requiera.
- Respuestas personalizadas ante errores del servidor. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado *script* cuando ocurra un error en concreto.
- Apache puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos.
- Es un servidor altamente configurable de diseño modular, lo que hace sencillo ampliar sus capacidades.

#### 2.3.3.2 Mongrel

La arquitectura modular de Apache le permite interactuar y funcionar en conjunto con otros servidores, entre estos se encuentra el servidor **Mongrel** [22], el cual fue diseñado fundamentalmente para facilitar el desarrollo, puesta en producción y extensión de aplicaciones Ruby.

Mongrel es un servidor Web pequeño, rápido y muy seguro, que se encarga de servir peticiones http, especialmente para aplicaciones Ruby, integrándose muy bien con el marco de trabajo Rails. Está desarrollado en su mayoría en lenguaje Ruby, pero utiliza algunas extensiones RubyC para el análisis sintáctico del URI (identificador uniforme de recurso) lo que lo hace más veloz. Sin embargo Mongrel es un servidor que por sí solo no representa la mejor opción debido a que este no puede manejar múltiples peticiones. Por esta razón se utiliza en combinación con el servidor Apache, para aprovechar al máximo las ventajas que ofrece cada uno.

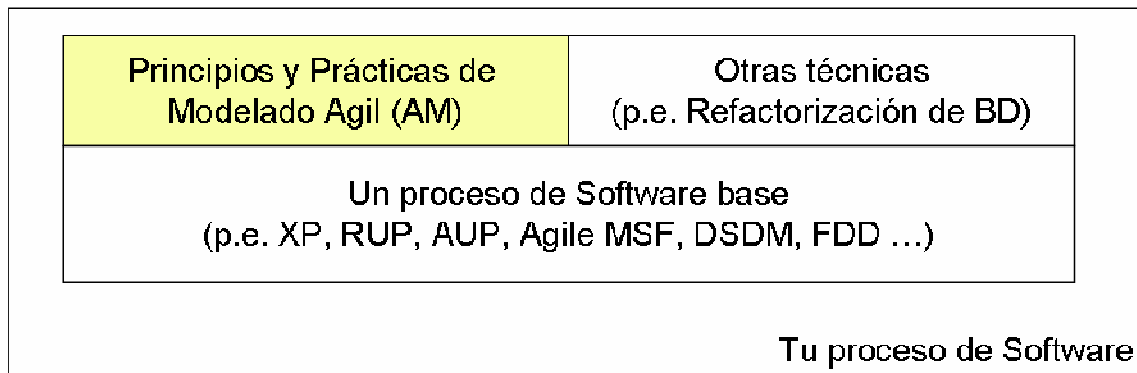
La labor del servidor Apache es recibir las peticiones para luego redireccionarlas a algunos de los servidores Mongrel. Normalmente como un solo servidor de Mongrel es insuficiente para atender para atender un elevado número de peticiones, se pueden correr al mismo tiempo varios servidores Mongrel, y

Apache se encarga de balancear la carga entre las peticiones recibidas con el fin de que cada uno reciba un número de peticiones promedio entre todos los demás, logrando así un mejor desempeño.

**CAPÍTULO 3.**  
**MARCO METODOLÓGICO**

**3.1 Modelado Ágil**

El Modelado Ágil (*Agile Modeling*, AM) [23] fue propuesto por Scott Ambler como un “conjunto de valores, principios y prácticas para modelar software que pueda ser aplicados a un proyecto de desarrollo de una forma efectiva y ligera, que complementan a las diferentes metodologías o procesos de desarrollo, ágiles o no, de manera de hacerlas adaptables a requerimientos cambiantes del cliente y lograr un proceso acorde a las necesidades de cada proyecto.



Copyright 2001-2006 Scott W. Ambler

**Figura 3. 1. Modelado Ágil refuerza Otros procesos de Software**

También puede definirse como una metodología basada en prácticas (con una fuerte inclinación a la producción de código) que se utiliza para el modelado y documentación de sistemas, la cual puede agregarse a otras metodologías establecidas para adaptarlas a las necesidades del proyecto, como se muestra en la figura 3.1<sup>8</sup>.

Los principales objetivos del Modelado Ágil son:

1. Definir y mostrar como poner en práctica una colección de valores, principios y prácticas que conlleven a un modelado ligero efectivo.

<sup>8</sup> Tomada de <http://www.agilemodeling.com/> y traducida al español de la versión original en Inglés.

2. Explorar la aplicación de técnicas de modelado en proyectos de software a través de un enfoque ágil, tal como XP, DSDM<sup>9</sup>, SCRUM<sup>10</sup> o AUP.
3. Explorar el como mejorar el modelado bajo procesos prescriptivos, tales como el Proceso Rational Unificado (RUP), o el Proceso Unificado Empresarial (EUP).

Los principios de AM especificados por Scott Ambler [24] incluyen:

- Presuponer simplicidad. La solución más simple es la mejor.
- El contenido es más importante que la representación. Pueden ser documentos formales o informales. Lo que importa no es el soporte físico o la técnica de representación, sino el contenido.
- Adaptarse al cambio. Aceptar que los requerimientos cambian.
- Habilitar el esfuerzo siguiente. Garantizar que el sistema es suficientemente robusto para admitir mejoras posteriores.
- Cambio incremental.
- Adquirir conocimiento de los modelos del problema, e identificar cuáles son sus fuerzas y sus debilidades.
- Adaptación local. Producir sólo el modelo que resulte suficiente para el propósito.
- Maximizar la inversión del cliente.
- Modelar con un propósito. Sólo se modela aquello que se necesita.
- Modelos múltiples. Múltiples paradigmas en convivencia, según se requiera.
- Comunicación abierta y honesta.
- Trabajo de calidad.
- Realimentación rápida.
- El software es el objetivo primario. Debe ser de alta calidad y coincidir con lo que el usuario espera.

---

<sup>9</sup> DSDM: Desarrollo de Software Dirigido por Modelos. es una propuesta para el desarrollo de software en la que se le atribuye a los modelos el papel principal de todo el proceso.

<sup>10</sup> SCRUM: Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software, propuesto por los Japoneses Hirotaka Takeuchi e Ikujiro Nonaka como modelo para el desarrollo de productos tecnológicos.

### 3.2 Metodología de desarrollo AUP

AUP [25] es una versión simplificada del *Rational Unified Process* (RUP). En él se describe un procedimiento sencillo y fácil de entender para el desarrollo de software de aplicación comercial ágil, utilizando las técnicas y los conceptos que aún permanecen todavía fieles a RUP.

El enfoque aplica técnicas ágiles e incluyen desarrollo basado en pruebas (TDD), Modelado basado en desarrollo ágil (AMDD), gestión de cambios ágil, y refactorización de base de datos para mejorar su productividad.

La Figura 3-2 muestra el ciclo de vida de la AUP. Haciendo una comparación de esta metodología con RUP se observa que las disciplinas han cambiado. En primer lugar, la disciplina Modelado de AUP abarca las disciplinas Modelado de Negocios, Requisitos y Análisis y Diseño de RUP. La disciplina Modelado es una parte importante de la AUP, como se puede ver, pero no dominan el proceso – se desea que el proceso permanezca ágil mediante la creación de solo los modelos y documentos estrictamente necesarios. En segundo lugar, la disciplina Configuración y Manejo del cambio es ahora la disciplina configuración de Gestión. En el desarrollo ágil las actividades de gestión de cambios suelen ser parte de los esfuerzos de manejo de requerimientos, que forma parte de la disciplina Modelado.

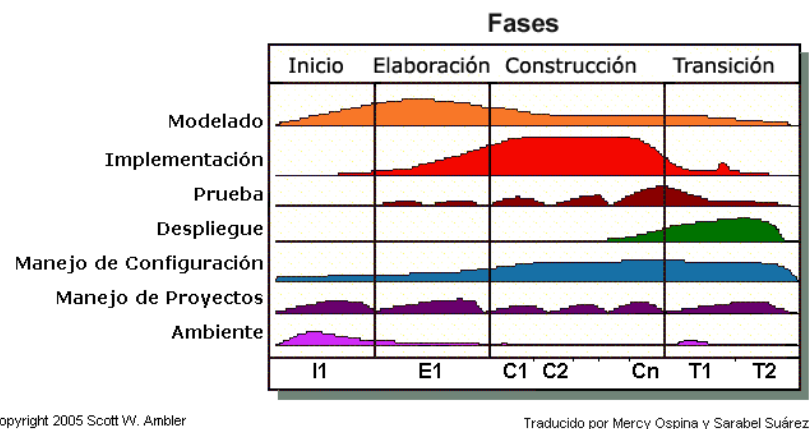


Figura 3. 2. Ciclo de vida de la AUP<sup>11</sup>

UP Ágil es formada por cuatro fases que se ejecutan en serie:

<sup>11</sup> Tomado de <http://www.ambyssoft.com/unifiedprocess/agileUP.html>

1. **Inicio.** El objetivo es identificar el alcance inicial del proyecto, una potencial arquitectura de la aplicación, y para obtener la financiación inicial del proyecto y la aceptación de los interesados.
2. **Elaboración.** El objetivo es probar la arquitectura de la aplicación. En esta fase el equipo de desarrollo se prepara para la fase de construcción creando el ambiente de desarrollo mediante la configuración del hardware, el software y las herramientas necesarias para la creación de la aplicación.
3. **Construcción.** El objetivo es construir software de trabajo, en una base regular e incremental que cumpla las necesidades altamente prioritarias de los interesados del proyecto.
4. **Transición.** El objetivo es validar y desplegar la aplicación en su entorno de producción. Se pueden llevar a cabo pruebas exhaustivas durante esta fase, incluidas las pruebas beta. El ajuste del producto se lleva a cabo aquí, así como la revisión para hacer frente a defectos importantes. El tiempo y esfuerzo que se puede invertir en esta fase varía de un proyecto a otro.

### Disciplinas e Iteraciones

Cada fase se divide en disciplinas y estas a su vez se pueden dividir en varias iteraciones dependiendo de la complejidad del problema. Las disciplinas se realizan de manera sistemática, definiendo las actividades de desarrollo que realizan los miembros del equipo para construir, validar y entregar software de trabajo que satisfaga las necesidades de sus interlocutores. Las disciplinas son las siguientes:

- **Modelado:** El objetivo de esta disciplina es entender el negocio de la organización, el dominio del problema hacia el que está dirigido el proyecto, e identificar una solución viable para resolver el dominio del problema.

Para esta disciplina no es necesario crear todos los modelos existentes, solo los más apropiados a la situación, ni es necesario modelar con muchos detalles en las fases de Inicio y Elaboración (ver Figura 3-3).

En la fase de Construcción se deben realizar tormentas de ideas, siguiendo la filosofía Justo a Tiempo (JIT por sus siglas en inglés) para obtener los detalles necesarios para cada iteración (Ver Figura 3-3).



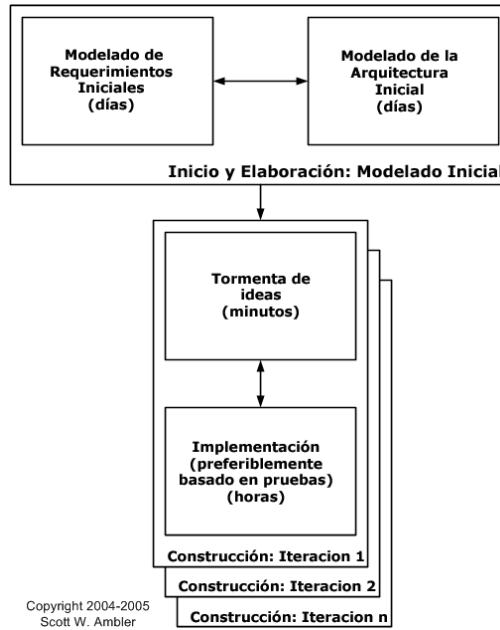


Figura 3.3. Ciclo de vida del Desarrollo (AM)<sup>12</sup>

- **Aplicación:** El objetivo de esta disciplina es transformar el (los) modelo(s) en código ejecutable y llevar a cabo un nivel básico pruebas, en particular, las pruebas unitarias. En esta disciplina se recomienda trabajar en parejas y bajo el esquema de desarrollo basado en pruebas.
- **Pruebas:** El objetivo de esta disciplina es realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, la validación de que el sistema funciona tal como está establecido, y verificar que se cumplan los requisitos.

La figura 3.4, muestra un diagrama de todo el ciclo de vida de las pruebas orientadas a objetos (FLOOT Full Lifecycle Object Oriented Testing)

<sup>12</sup> Tomada de [www.agilemodeling.com/](http://www.agilemodeling.com/)

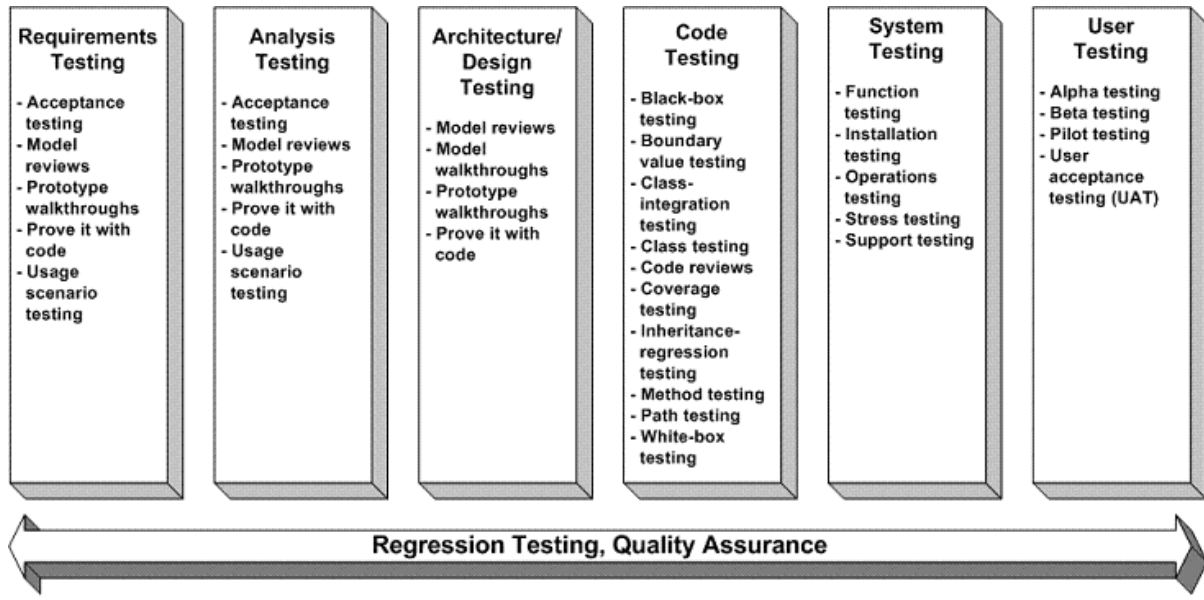


Figura 3. 4. Ciclo de vida de las pruebas orientadas a objetos (FLOOT)<sup>13</sup>

- **Despliegue:** El objetivo de esta disciplina planificar para la entrega del sistema y ejecutar el plan para hacer que el sistema este a disposición de los usuarios finales.
- **Gestión de configuración.** El objetivo de esta disciplina es gestionar el acceso a los artefactos del proyecto. Esto incluye no sólo el seguimiento de versiones de los artefactos a lo largo del tiempo, sino también el control y la gestión de cambios en ellos.
- **Gestión de Proyectos.** El objetivo de esta disciplina es dirigir las actividades que lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, dirección de personas (la asignación de tareas, seguimiento de los progresos, etc.), y coordinar con las personas y los sistemas fuera del alcance del proyecto para garantizar que se entrega a tiempo y dentro del presupuesto.
- **Medio ambiente.** El objetivo de esta disciplina es apoyar el resto del esfuerzo por garantizar que el proceso adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc.) están disponibles para el equipo según sea necesario.

<sup>13</sup> Tomada de <http://www.ambysoft.com/essays/floot.html>

### Entrega de versiones incrementales en el tiempo

En lugar del enfoque de "big bang" donde se cumple con la entrega de todos los programas a la vez, los equipos de desarrollo AUP suelen emitir revisiones al final de cada iteración en escenarios de pre-producción. Una versión de desarrollo de una aplicación es algo que podría ser liberado en producción si pasa a través de un módulo de aseguramiento de calidad (QA), las pruebas, y procesos de despliegue, en pre-producción.

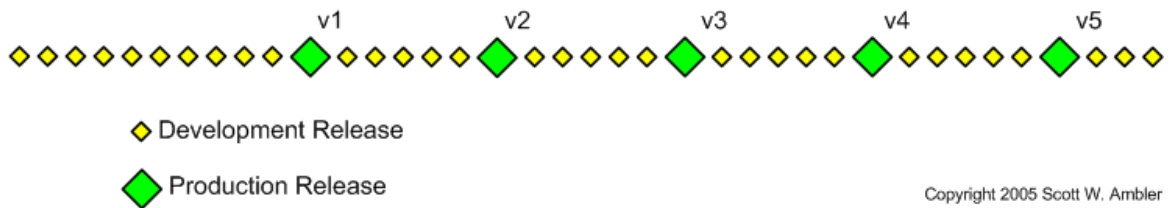


Figura 3. 5. Crecimiento de las versiones en el tiempo

En la Figura 3-5, se puede ver que la primera versión de producción a menudo toma más tiempo que las versiones posteriores; en la primera versión de un sistema se debe tomar en cuenta el tiempo dedicado a crear el ambiente de trabajo y la adaptación del equipo de desarrollo, lo que permitirá tener una colaboración eficaz. La primera versión de producción podría tomar doce meses para entregar, la segunda versión de nueve meses, y luego otras versiones pueden ser entregadas cada seis meses.

### Filosofías de la AUP

El UP Ágil se basa en los siguientes principios:

1. *Confianza en las capacidades del personal.* No es necesaria una documentación detallada del proceso ya que casi nadie la lee, esto se sustituiría por guías de alto nivel y / o capacitación. El producto AUP ofrece flexibilidad en este aspecto dejando que los desarrolladores tomen la decisión de cuales documentos son necesarios.
2. *Simplicidad.* Se busca que los documentos sean concisos y precisos, además de cortos.
3. *Agilidad.* El Agile UP se ajusta a los valores y principios de la Agile Alliance.
4. *Centrado en las actividades de alto valor.* La atención se centra en las actividades que realmente cuentan, no cada cosa que podría suceder en un proyecto.
5. *Independencia de herramientas.* Se puede usar cualquier conjunto de herramientas con el Agile UP.

6. *Adaptación* para satisfacer las necesidades del proyecto. La metodología AUP es fácilmente a las necesidades de cada proyecto.

### 3.3 Lineamientos de Usabilidad

Para el desarrollo de este trabajo especial de grado se tomaron en cuenta tres factores fundamentales de Usabilidad al momento de implementar una aplicación Web:

- Navegación
- Contenido
- Diseño

#### 3.3.1 Navegación o Navegabilidad

Debido a que la Web es un sistema de navegación, gracias a que los vínculos de hipertexto permiten al usuario moverse por un amplio espacio de información conformada por páginas Web, al realizar el estudio del diseño de un sitio Web se debe tomar en cuenta los aspectos de navegación de manera que el usuario se pueda contestar las siguientes preguntas: ¿Dónde estoy?, ¿Dónde he estado?, ¿A dónde puedo ir?

¿DÓNDE ESTOY?: Es importante para los usuarios saber donde están para llegar a entender la estructura del sitio y poder interpretar el significado del vínculo que acaba de seguir.

La ubicación actual del usuario debe mostrarse a dos niveles diferentes:

1. En relación con la Web como un todo
2. En relación con la estructura del sitio

Es necesario identificar el Sitio en todas las páginas, ya que forman un subconjunto de la Web, por lo tanto, la primera regla de navegación consiste en incluir el logotipo (u otro identificador del sitio) en cada página. Este deberá tener una colocación coherente (preferentemente, la esquina superior izquierda si la página está en un lenguaje que se lea de izquierda a derecha) y puede convertirse en un vínculo de hipertexto con la página de inicio, de forma que los usuarios puedan llegar a su página de inicio desde cualquier otra página.

La ubicación relativa a la estructura del sitio suele darse mostrando estructuras del sitio y resaltando el área donde se encuentra la página. También es importante tener un título principal claro en la página que indique su nombre o contenido principal. Por último, el título de la página de la definición de encabezado HTML debe utilizarse para generar un nombre significativo en cada una de las páginas, de forma que los usuarios puedan localizarla fácilmente en su lista de marcadores.

### *Estructura del Sitio*

La estructuración del sitio Web debe hacerse mediante clasificaciones temáticas útiles e inteligibles por el usuario. Uno de los errores más habituales es reproducir la estructura jerárquica de la organización en el sitio Web, de forma que ante un cambio en la estructura de la organización habrá que rehacer el sitio Web.

¿DÓNDE HE ESTADO?: Dado que la tecnología Web estándar no tiene estado, no es tarea fácil para los diseñadores de páginas contestar directamente a la pregunta "¿Dónde he estado?", ya que el sitio no lo sabe sin acudir a *cookies* u otras medidas de control del usuario. Algunos de los mecanismos de desplazamiento de los navegadores actuales ofrecen herramientas que ayudan al usuario sobre esta cuestión. El botón Atrás lleva al usuario directamente a la página anterior<sup>14</sup>, la lista de historial incluye una lista de páginas recientemente visitadas, y los vínculos de hipertexto aparecen en un color diferente si señalan a las páginas anteriormente visitadas.

Los enlaces deben tener colores fácilmente identificables, y han de ser colores distintos para enlaces visitados y no visitados<sup>15</sup>. Saber los vínculos que conducen a las páginas anteriormente visitadas es útil por dos motivos: ayuda a los usuarios a aprender la estructura del sitio y les impide perder el tiempo yendo a la misma página muchas veces.

¿A DÓNDE PUEDO IR?: A esta pregunta responden las numerosas opciones de navegación y todos los demás vínculos que pueda haber en la página. Además si presuponemos que el usuario ha entendido medianamente la estructura del sitio, se habrá hecho una idea general de otras páginas a donde puede ir. Dado que es imposible mostrar todos los destinos posibles en todas las páginas, resulta obvio que una buena estructura de sitio constituye una ventaja muy importante a la hora de ayudar a los usuarios a responder a la pregunta "¿A Dónde puedo ir?"

---

<sup>14</sup> Nielsen Jakob. Usabilidad: Diseño de sitios Web. Pag. 214

<sup>15</sup> Nielsen, Jakob. Alertbox: Change the Color of Visited Links

### *Área de Navegación:*

El área de navegación debe ocupar un lugar relevante dentro de la página, preferentemente junto al cuerpo de contenido de esta, y debe contener accesos directos a tareas importantes del sitio Web.

Los elementos del área de navegación suelen ir dentro de una lista. En el caso de ser muy abundantes es muy útil para el usuario su distribución en áreas temáticas: crear categorías para dichos elementos.

### **3.3.2 Elementos de Contenido**

El contenido y su calidad diferencian un Sitio Web de los demás, la elección de los contenidos del sitio debe ajustarse a su objetivo y al público al que va dirigido, en el desarrollo de aplicaciones Web no se dispone siempre de todo el contenido a publicar y de tenerlo este tiene que ser repensado para adaptarse a las limitaciones de la Web.

La redacción de los contenidos para la Web deben ajustarse en la medida de lo posible a las siguientes características [26]:

- Uso de lenguaje sencillo y directo, para transmisión clara de las ideas. Han de evitarse las estructuras complejas (empleo de oraciones cortas y que contengan una sola idea) y los conceptos abstractos.
- En caso de necesidad de mención de conceptos abstractos es recomendable usar ejemplos concretos o comparaciones que faciliten su comprensión.
- No se debe dar por hecho que el usuario tiene conocimientos previos sobre el tema que se trata.
- Empleo de palabras cortas, evitando, en la medida de lo posible, palabras largas y/o difíciles de pronunciar.
- Dirigirse al usuario de forma directa y personal, utilizando de segunda persona (“Si tienes alguna pregunta no dudes en contactar con nosotros”) en lugar de tercera persona (“En caso de que algún usuario tenga alguna pregunta que no dude en contactar con nosotros”).
- Adaptar el lenguaje utilizado a la audiencia potencial del sitio, y que dicho lenguaje sea positivo. Se han de evitar el lenguaje negativo y las negaciones.
- Cuidado al utilizar lenguaje figurativo o metafórico, ya que puede resultar confuso para una parte de los usuarios.

- Uso de voz activa en lugar de pasiva, haciendo que el documento sea más vivo y con menor complicación.
- Sistematización en el uso de las palabras, utilizando siempre la misma palabra para denominar una misma cosa, aunque la repetición de palabras afecte al estilo de redacción.
- Elección de signos de puntuación sencillos, evitando el punto y coma, los guiones y las comas.

### *Contenido de la Página de Inicio*

La página principal es el lugar más importante de un sitio Web, por el que acceden al sitio la mayor parte de sus usuarios. Es importante seleccionar adecuadamente los contenidos que aparecerán en esta, proporcionándoles suficiente espacio para facilitar su legibilidad y visibilidad.

La saturación en la página principal por un exceso de contenidos produce un colapso informacional, los elementos dejan de ser visibles, al no tener espacios vacíos entre ellos. La consecuencia es que el usuario no encuentra la información que busca o le cuesta mucho trabajo encontrarla.

### **3.3.3 Diseño**

Los elementos de diseño definen la interfaz que tendrá la aplicación con el usuario, y su principal objetivo es destacar los contenidos del sitio a diferentes niveles (contenidos principales y secundarios), y si está bien planificado puede facilitar el aprendizaje del uso de la aplicación y su posterior reconocimiento.

Se deben tomar en cuenta ciertos aspectos al momento de diseñar la interfaz de usuario, como el uso de elementos multimedia e imágenes, la resolución de pantalla, el uso de colores, la distribución del espacio y de los elementos dentro de dicho espacio, el formato del texto de los contenidos, entre otros.

Para la elección de cada uno de los factores anteriores se debe tomar en cuenta el objetivo del sitio Web, la información, su estructura y su relevancia, los usuarios de la aplicación cuya interfaz se requiere diseñar, entre otras.

### *Elementos multimedia e imágenes*

Algunos elementos multimedia pueden ser muy atractivos, pero dentro de un sitio público traen consigo problemas de carga lenta del sitio Web, poca compatibilidad en diferentes navegadores y versiones de los mismos, además pueden distraer al usuario y no aportar utilidad o valor adicional a la aplicación.

El uso de las imágenes es un elemento importante ya que, por un lado pueden hacer un Sitio Web más atractivo, pero por otro lado ocupan espacio y pueden generar que una página se cargue con menos rapidez, por lo tanto se deben elegir cuidadosamente para que estén acorde al contenido, y al objetivo de la aplicación Web.

#### *Resolución de pantalla*

La resolución de pantalla es un elemento importante ya que determinará la aparición de desplazamientos verticales u horizontales, el cual puede presentar ciertos inconvenientes a los usuarios, la existencia de desplazamiento vertical no es negativa ya que el usuario entiende que el contenido puede ser algo mayor que la pantalla, y en muchos casos es difícil o imposible de eliminar. Por otro lado el desplazamiento horizontal si es muy negativo, ya que resulta mucho más incómodo: puede interrumpir las líneas de texto, haciendo que el usuario tenga que desplazarse una y otra vez en la página.

Para evitar los desplazamientos hay que adecuar<sup>16</sup> la anchura y la altura de las páginas a las resoluciones de pantalla más habituales, tomando como referencia la más pequeña de todas ellas.

#### *Formato del Texto*

El contenido debe ser posible de leer y entender por cualquiera de los usuarios del Sitio, por lo que los elementos de diseño aplicados al texto deben tomar en cuenta las limitaciones que puedan presentar algunos usuarios, para ello se recomienda:

- Texto de poca anchura.
- Margen e interlineados suficientes.
- Tipografía del tamaño que permita una lectura cómoda a la mayoría de los usuarios.
- Elección de tipografía tipo *Sans-Serif* que es más adecuado para la lectura en pantalla.

Adicionalmente la distribución del texto debe ser ordenada

---

<sup>16</sup> Nielsen Jakob. Alertbox: Screen Resolution and Page Layout



### *Separación de presentación y contenido*

Las páginas del sitio deben dividirse en dos partes<sup>17</sup>:

- Estructura.
- Presentación.

La estructura es el contenido del documento, normalmente en código HTML o XHTML.

La presentación es la parte donde se define el aspecto visual de la estructura (tamaños de fuente, colores, etc.), la cual se debe realizar mediante hojas de estilo CSS y no con el uso de etiquetas HTML, ya que permite la separación tanto física (archivos diferentes) como lógica de esta y el contenido

Al separar la presentación y el contenido la accesibilidad es mayor, y el peso del documento es menor, facilitando además un diseño homogéneo del Sitio, donde todas las páginas coinciden en su aspecto y permite mayor flexibilidad en caso de requerir cambios de diseño.

### **3.4 Estándares Web**

Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí. [27]

El cumplimiento de estándares trata de evitar situaciones que puedan generar confusión, como el hecho de que distintas palabras, situaciones o acciones signifiquen lo mismo.

Los estándares Web son un conjunto de recomendaciones dadas por el World Wide Web Consortium [28] (W3C) y otras organizaciones internacionales acerca de cómo crear e interpretar documentos basados en tecnología Web. Son un conjunto de tecnologías orientadas a brindar beneficios a la mayor cantidad de usuarios, asegurando la vigencia de todo documento publicado en la Web.

El W3C (World Wide Web Consortium) [24], es una organización independiente y neutral, que desarrolla estándares relacionados con la Web también conocidos como Recomendaciones, los cuales sirven como

---

<sup>17</sup> Zeldman, Jeffrey. Diseño con estándares Web. pag. 69-70

referencia para construir una Web accesible, interoperable y eficiente, en la que se puedan desarrollar aplicaciones cada vez más robustas.

Para ayudar a los desarrolladores que deseen utilizar sus Recomendaciones, el W3C ofrece una serie de herramientas que permiten verificar si se hace una correcta aplicación de las especificaciones. Manuales de directivas o buenas prácticas de tecnologías concretas, y los validadores sintácticos de los lenguajes, son ejemplos de estas ayudas

Algunos estándares o recomendaciones de la W3C son:

**Accesibilidad:** [29] Este estándar llamado también WAI por las siglas en inglés de Iniciativa de Accesibilidad Web, hace referencia a un diseño Web que va a permitir que personas con algún tipo de discapacidad puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto mermadas sus habilidades a consecuencia de la edad.

La accesibilidad Web engloba muchos tipos de discapacidades, incluyendo problemas visuales, auditivos, físicos, cognitivos, neurológicos y del habla.

Hacer un sitio Web accesible puede ser algo sencillo o complejo, depende de muchos factores como por ejemplo, el tipo de contenido, el tamaño y la complejidad del sitio, así como de las herramientas de desarrollo y el entorno.

Muchas de las características accesibles de un sitio se implementan de forma sencilla si se planean desde el principio del desarrollo del sitio Web o al comienzo de su rediseño. La modificación de sitios Web inaccesibles puede requerir un gran esfuerzo, sobre todo aquellos que no se "etiquetaron" correctamente con etiquetas estándares de XHTML, y sitios con cierto tipo de contenido, como multimedia.

Para saber si un sitio Web es accesible existen varias herramientas [30] tal como TAW (El Test de Accesibilidad Web), la cual es una herramienta de la Fundación CTIC y SIDAR. Este test es una herramienta de validación en castellano basada en las reglas desarrolladas por WAI, y está disponible a través de su página Web<sup>18</sup> y como una versión de descarga para ejecutarla localmente.

---

<sup>18</sup> <http://www.tawdis.net/taw3/cms/es/>

**Hojas de estilo (CSS):** Es un mecanismo para dar estilo a documentos HTML y XML, que consiste en reglas simples a través de las cuales se establece cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores control total sobre el estilo y formato de sus documentos separando contenido y presentación.

Los documentos que separan presentación de contenido usan menos código, además, CSS permite conseguir efectos que antes requerían el uso de *JavaScript* e imágenes, por lo que los sitios basados en este estándar utilizan menos ancho de banda y se muestran más rápido a los usuarios, mejorando dramáticamente la experiencia de estos.

La W3C ofrece un validador de hojas de estilos en línea que está disponible en el siguiente enlace <http://jigsaw.w3.org/css-validator/>.

**XHTML:** Es una versión más estricta y limpia de HTML que nace ante la limitación de uso de éste con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.

Los documentos basados en XHTML válido son más relevantes para los motores de búsqueda, contienen mayor información y menos código, por lo que un sitio basado en este estándar Web tendrá una mejor posición.

La W3C ofrece un validador de XHTML en línea que está disponible en el siguiente enlace <http://validator.w3.org/>

**Internacionalización:** [31] La Internacionalización es un concepto que implica un diseño Web sin barreras para los diferentes idiomas, sistemas de escritura, códigos de caracteres y otras convenciones locales que existen. El carácter global de la Web requiere un sistema en el que exista facilidad a la hora de crear y procesar información para una audiencia variada, permitiendo así publicar material e intercambiar datos en cualquier idioma. La Internacionalización también es conocida como I18N, que es la abreviatura de Internacionalización ya que entre la "i" y la "n" existen 18 letras.

El W3C inicia la Actividad de Internacionalización en un intento por asegurar que estos formatos y protocolos puedan utilizarse de forma universal en todos los idiomas y en todos los sistemas de escritura.

Por lo tanto, la creación de un sitio Web internacional permite garantizar su utilización universal incluyendo todos los idiomas y culturas.

### **Identificadores de idioma y codificación**

Usuarios de diferentes países y con diferentes culturas necesitan servicios adaptados correctamente para procesar información usando su idioma de origen, su sistema de escritura, su sistema de medida, sus calendarios y otras reglas y convenciones culturales, la especificación de un conjunto particular de convenciones culturales es importante para que un sitio Web procese la información que intercambia con el usuario correctamente. Hay muchas preferencias que un sitio Web debe ofrecer para que sea considerado usable y aceptable por los usuarios a nivel mundial.

Al existir un variado número de preferencias o de circunstancias culturales o de idioma, es importante utilizar identificadores basados en el idioma y lugar como referente para recoger información sobre las preferencias de los usuarios. Por ejemplo HTML usa el atributo lang para indicar el idioma de segmentos de contenido. XML utiliza el atributo xml:lang con el mismo objetivo.

La utilización de estos identificadores de idioma son muy importantes. Por ejemplo, es un requisito imprescindible identificar el idioma de nuestra página al inicio del documento, antes de head, de la siguiente forma:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
```

También es importante marcar cualquier cambio de idioma que se pueda producir a lo largo del texto en una página Web. Un ejemplo sería:

```
<p xml:lang="en">The quick brown fox jumps over the dog.</p>
```

```
<p xml:lang="en-GB">What colour is it?</p>
```

```
<p xml:lang="en-US">What color is it?</p>
```

Los idiomas se especifican a través de un código estándar, como se aprecia en los ejemplos anteriores. Primero se indica el idioma (en, inglés), y si se desea, el país (en-US, inglés de EEUU)

No hay que olvidar tampoco indicar, si incluimos un enlace a otra página, el idioma en el que esa página se va a mostrar, esta vez se indicará con hreflang

`<a href="http://www.w3.org/International/questions/" hreflang="en">`

Preguntas frecuentes sobre Internacionalización</a>

Otro aspecto importante es hacer frente a los problemas de codificación en documentos (X)HTML. Se trata de un método para transformar una secuencia de bits en una secuencia de caracteres. Los servidores envían documentos HTML a agentes de usuario como cadenas de bits; a su vez, los agentes de usuarios los interpretan como una secuencia de caracteres. El método de conversión va desde una simple transformación hasta algoritmos y esquemas complejos. Una forma de solucionar problemas de codificación es servir todas las páginas en un conjunto de caracteres, por ejemplo UTF-8, un conjunto de longitud variable que utiliza grupos de bits para representar el estándar Unicode para el alfabeto de varios idiomas. UTF-8 puede representar los caracteres de una amplia variedad de idiomas. Los navegadores envían de vuelta los datos en el mismo codificado que la página que contiene la información.

Es muy importante que el conjunto de caracteres de cualquier documento XML o (X)HTML esté claramente etiquetado. Esto puede hacerse utilizando el parámetro `charset`, en la cabecera `Content-Type` de HTTP, de la siguiente manera:

`Content-Type: text/html; charset=UTF-8`

Para XHTML, hay que usar la etiqueta `<meta>` dentro de `<head>`, de la siguiente forma:

`<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />`

**Web Móvil:** [32] Cuando se habla de Web Móvil se está haciendo referencia a una Web en la que el usuario puede acceder a la información desde cualquier lugar, independientemente del tipo de dispositivo que utilice para ello.

Actualmente, existe una demanda cada vez mayor por parte de los usuarios en lo referente a una disponibilidad incondicional de la Web, pero la realidad en el mercado es otra ya que aunque la oferta de dispositivos móviles está creciendo de forma asombrosa en los últimos años, ofreciéndonos infinidad de dispositivos desde los que llevar a cabo operaciones que normalmente realizábamos desde el equipo de sobremesa, existen limitaciones a la hora de acceder a los servicios desde esos dispositivos móviles. En la mayoría de las ocasiones, el resultado es una experiencia de usuario poco satisfactoria al encontrarnos con numerosos problemas para acceder a la Web desde los dispositivos móviles.

El W3C, con el objetivo de convertir el acceso a la Web desde un dispositivo móvil en algo tan sencillo y cómodo como lo es desde los equipos de sobremesa, ha puesto en marcha la Iniciativa de Web Móvil que busca resolver los problemas de interoperabilidad y usabilidad que actualmente dificultan el acceso a la Web desde dispositivos móviles y hacer posible uno de los objetivos principales del W3C que consiste en alcanzar una Web única.

La Web móvil llega a lugares donde el cable no puede llegar, lugares que anteriormente eran impensables, como está ocurriendo en los países en vías de desarrollo. Esto trae consigo nuevas oportunidades para realizar negocios, para trabajar, para gestionar nuestro tiempo de ocio, y para muchas otras cosas, lo que nos permite incrementar nuestra capacidad de movimiento.

La Web móvil se presenta como un auténtico reto tanto para usuarios como para desarrolladores ya que, por un lado, el usuario encuentra problemas al intentar acceder a los sitios Web desde los dispositivos móviles, y por otro, los proveedores de contenido encuentran dificultades para crear sitios Web que funcionen adecuadamente en todos los tipos de dispositivos y configuraciones.

En la actualidad, el trabajo del W3C en temas de Web móvil se centra principalmente en dos áreas: generación de buenas prácticas y descripción de dispositivos móviles. El Grupo de Trabajo de Buenas Prácticas en Web Móvil ha sido creado para desarrollar pautas, puntos de verificación y buenas prácticas con el objetivo de ayudar a los proveedores de contenido a desarrollar contenido Web que funcione correctamente en dispositivos móviles. Por otro lado, el Grupo de Trabajo de Descripción de Dispositivo se ha creado para guiar el desarrollo de mecanismos de descripción de dispositivos, es decir, bases de datos de descripciones que los desarrolladores de contenido podrán utilizar para adaptar los contenidos a los diferentes dispositivos.

### **3.5 Patrones de Diseño de Interacción**

Una herramienta para facilitar la elección de soluciones a problemas concretos de diseño son los denominados patrones de diseño, originalmente propuestos por el arquitecto Christopher Alexander en el contexto del diseño y construcción urbanística. Partiendo de la definición original de Alexander, un patrón de diseño es una solución a un problema que se usa repetidamente en contextos similares con algunas variantes en la implementación.

Para que una solución pueda ser considerada un patrón de diseño debe ser eficaz, es decir, que se haya demostrado que resuelve satisfactoriamente el problema, y además debe ser reutilizable, que pueda ser aplicada en diferentes casos. Los Patrones de Diseño de Interacción tienen el objetivo de resolver problemas de usabilidad.

En años más recientes investigadores como el Martin Van Wellie, Jennifer Tidwell, Jaime Muñoz han desarrollado colecciones de Patrones de Interacción para la Web, dichas colecciones captan la experiencia de programadores y diseñadores expertos en el desarrollo de interfaces usables y condensan esta experiencia en una serie guías o recomendaciones, que puedan ser usadas por otros desarrolladores.

De la revisión de los patrones de Wellie se tomaron los siguientes como referencia para el diseño del Sitio Web a implementar.

**Homepage:** En este patrón los usuarios necesitan entender si están en el lugar correcto, y de ser así, cómo pueden llevar a cabo una determinada actividad en el sitio. La página principal (homepage) es la puerta de entrada al Sitio Web, por lo que es importante que el usuario se ubique correctamente y tenga claro el objetivo y su contenido.

**Fly out menú:** (menú desplegable) En este patrón se requiere que los usuarios tengan acceso directo las opciones de sub-navegación, pero el espacio en pantalla para la navegación es limitado.



Figura 3. 6. Ejemplo de *Fly out menu* o menú desplegable

**Search Area:** (área de búsqueda) Se utiliza cuando se requiere que los usuarios puedan encontrar alguna información particular en el Sitio



Figura 3. 7. Ejemplo del patrón *Search Area*

Por lo general se coloca en un área rectangular ubicada en un sitio prominente de la página

**Login:** Se requiere que los usuarios se identifiquen de forma que los datos almacenados acerca de estos puedan ser usados en el proceso para dar acceso a funcionalidades del Sitio

Please log in to access your personal data. [Help?](#)

username

password

[Forgotten your password?](#)

Store username and password on this computer.

If you are not already registered, you can [register as a new visitor](#)

Figura 3. 8. Ejemplo del Patrón *Login*

**Date selector:** Los usuarios requieren seleccionar una fecha o un periodo de tiempo. Se recomienda usar un calendario gráfico donde se pueda hacer clic en la fecha deseada

**Event Calendar:** Los usuarios requieren ser informados de eventos pasados o futuros, se usa junto con el patrón Date Selector

**Events Calendar**

Welcome to Baylor's online events calendar. Please select from the boxes at the right to view events for a day, week or month, and to filter events based on a category.

**Featured Events: Week of Jan 9 - Jan 15**

DATE/TIME	EVENT	PUBLISHER
1/10 (All Day)	Class Sessions Begin	Office of the Provost
1/11 (3:30 - 5 pm)	Spring Faculty Meeting	Office of the Provost

Jan 2005

Event Categories

Figura 3. 9. Ejemplo del patrón *Event-Calendar*



## **CAPÍTULO 4.**

### **DESARROLLO DE LA APLICACIÓN**

#### **4.1 Objetivo general de la aplicación**

Esta aplicación tiene como objetivo construir un Sitio Web que represente a la Escuela de Computación en la Internet, mostrando información de relevancia tales como noticias, eventos, personal y autoridades, actividades y centros de investigación, actividades administrativas, servicios y recursos, entre otros; y que sea fácil de actualizar.

#### **4.2 Objetivos específicos de la aplicación**

1. Mostrar los últimos eventos y noticias referentes a la Escuela.
2. Dar a conocer ofertas de empleo, pasantías, seminarios.
3. Mostrar información sobre los trabajos especiales de grado presentados y por presentar.
4. Mostrar información sobre los Centros de Investigación.
5. Dar a conocer los recursos y servicios que ofrece la Escuela.
6. Permitir la consulta de planillas, normativas y recaudos para solicitudes.
7. Permitir la consulta del personal de la Escuela y sus autoridades.
8. Permitir la actualización de la información de forma fácil y segura
9. Presentar un diseño limpio, fácil de usar y que cumpla con los estándares Web actuales.

#### **4.3 Alcance de la aplicación**

La Escuela de Computación lleva a cabo una gran cantidad de procesos tanto administrativos como docentes y de investigación que de querer implementarse en su totalidad en este desarrollo sería necesario invertir una cantidad excesiva de tiempo, por lo que se hace necesario plantear un desarrollo incremental considerando esta una primera fase, donde se implementarán aquellas funcionalidades que se consideren más importantes y/o necesarias para la Escuela, tomando en cuenta que también existen otros trabajos de investigación o aplicaciones que abarcan procesos particulares por lo que la aplicación actual podría servir de enlace a estos, como es el caso de la página de postgrado.

#### 4.4 Características de las Aplicaciones Web Universitarias

Para comprender cuales pueden ser las características más resaltantes que debería tener el Sitio Web de la Escuela de Computación se procedió a estudiar los Sitios Web de diferentes Universidades cuyo objetivo fuera similar a los planteados para nuestro Sitio.

Se estudiaron las páginas de varias universidades y departamentos o escuelas de computación y se observó lo siguiente:

##### 4.4.1 Navegación

En la mayoría de los casos tienen una o más áreas de navegación en forma de menú, de uno o más niveles, que se ubican en la parte superior, luego del logo y/o en la parte izquierda, el menú se encuentra separado en secciones claramente definidas.

Presentan un buscador, un mapa del sitio, y en algunos de los casos un enlace a la intranet.

También se puede observar el uso de marcadores de ubicación o *bread crumbs*.

##### 4.4.2 Contenido

Presentan una página principal donde su contenido puede ser:

- Novedades (Noticias y eventos).
- Categorías desde las que se puede acceder a los contenidos del sitio.
- Contenidos más consultados.
- Contenidos referidos a una época concreta como Inscripciones o exámenes en el caso de Universidades.

## Información de Contacto

Debe existir alguna forma de ponerse en contacto con la organización mediante un buzón de sugerencias o una herramienta similar. En la gran mayoría de los sitios visitados se observó información como dirección, correo de contacto y teléfonos de contacto.

Contenido de las demás páginas: Para las páginas secundarias se observa una serie de contenidos según la sección, entre las cuales pudimos encontrar:

- Futuros estudiantes.
- Académica (materias, plan de estudios, calendario, etc.).
- Información general (objetivo, misión, visión).
- Investigación.
- Directorio.
- Servicios.
- Publicaciones.
- Fecha de actualización.
- Promociones.
- Extensión.
- Enlaces de interés.
- Soporte a varios idiomas (menos del 50%).

### **4.4.3 Diseño**

En cuanto a los elementos de diseño se pudo observar:

- Uso del logo y colocación en la parte superior izquierda, como parte del *banner*, el logo además tiene un enlace a la página principal.
- Uso del logo de la universidad y de la facultad, con sus respectivos enlaces.
- Uso de los colores de la universidad o facultad.
- Uso de imágenes propias de la Escuela, Facultad o Universidad.
- Sitio homogéneo (todas las páginas presentan una estructura y diseño similar).
- Uso de capas y hojas de estilo (separación de diseño y contenido).
- Poco uso de animaciones flash. Los que usaron flash cargaron más lentamente.

#### 4.5 Adaptación de la metodología AUP

Para el desarrollo de la aplicación descrita en el presente trabajo de investigación se usó como guía la metodología de desarrollo AUP la cual como ya se indicó anteriormente sigue los principios de la Modelación ágil, con la finalidad de que este se realizara de manera rápida, sencilla, eficiente y con la suficiente documentación. También se tomo esta decisión por estar familiarizados con los artefactos o entregables UML.

A continuación se describe como se aplican las fases de AUP a nuestro trabajo de investigación:

➤ **Inicio:**

En esta etapa se tuvieron diversas reuniones con el cliente para determinar el alcance inicial del sistema, definir los requerimientos iniciales y una posible arquitectura del sistema, para construir un modelo inicial de requerimientos y un modelo inicial de arquitectura siguiendo los lineamientos de la Modelación ágil. Para los modelos se usaron diagramas de casos de uso, modelos de objeto y prototipos de interfaz

Asimismo se hizo un análisis global para determinar la posible interfaz de la aplicación, y los principios de usabilidad y estándares de diseño Web que debería cumplir esta.

Se determinaron además algunas reglas de negocio que aunque implícitas debían ser cumplidas por la aplicación.

Como resultado se obtuvieron los casos de uso iniciales, el modelo de arquitectura inicial, el modelo de objetos del dominio y el prototipo inicial de las principales interfaces.

➤ **Elaboración:**

En esta etapa se determino la arquitectura del sistema, se realizaron el diagrama de clases inicial, el prototipo de interfaz de usuario y el modelo de arquitectura.

➤ **Construcción:**

Para esta etapa se definieron en primer lugar las iteraciones del subsistema público, las del subsistema administrativo se comenzaron a definir luego de implementar las dos primeras iteraciones del público, en cada iteración se implementaron más de un casos de uso que estuvieran relacionados, y en general aquellos con los que se contaba con información suficiente. Se dejaron para el final aquellos casos de uso donde la

información era escasa o difícil de conseguir, o donde no estaban suficientemente claros los requerimientos.

➤ **Transición:**

Es esta fase se colocó una versión beta del proyecto en producción, donde se realizó una encuesta de aceptación a un grupo de usuarios

### 4.5.1 INICIO

**Requerimientos iniciales:**

Se identificaron los siguientes requerimientos:

- Se debe mostrar información de la Escuela como su historia, objetivos, visión, misión y autoridades.
- Se debe mostrar la información del personal docente y administrativo de la Escuela.
- Debe tener una sección para las noticias y eventos que sean de interés para los usuarios del sitio.
- Debe centralizar la información de las materias que se dictan en la Escuela.
- Debe centralizar la información referida a los centros de investigación de la Escuela.
- Se debe mostrar información sobre ofertas de trabajos y pasantías, así como también ofertas de temas de seminario.
- Se deben mostrar los documentos de docencia, reportes técnicos y documentos administrativos.

El sistema debe contar con funcionalidades que permitan la actualización de la información de los requerimientos mencionados anteriormente de manera sencilla, por lo que se propone dividir el sistema en dos subsistemas:

- **El Subsistema Público**, el cual muestra la información y esta podrá ser vista por cualquier tipo de usuario.
- **El Subsistema Administrativo**, permite el acceso sólo a usuarios autorizados, los cuales tendrán diferentes perfiles, y podrán actualizar la información que se muestra en el Subsistema Público.

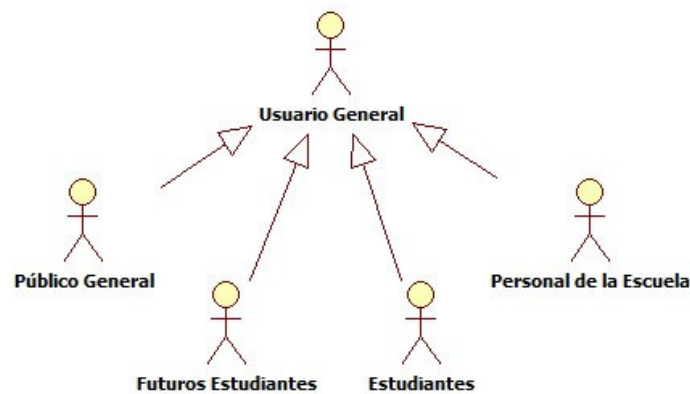
**Casos de uso iniciales:**

➤ **Principales usuarios:**

En la figura 4.1 se muestra una clasificación de los usuarios del subsistema público, la cual se realizó con la finalidad de identificar los distintos tipos de usuarios del subsistema y poder determinar las necesidades de cada uno, sin embargo el usuario que interactuará con el sistema es el usuario general.

En la figura 4.2 se muestran los usuarios del subsistema administrativo, en este caso si cumplen distintos roles en el subsistema.

➤



**Figura 4. 1. Principales Usuarios Subsistema Público**



**Figura 4. 2. Principales usuarios del subsistema administrativo.**

➤ **Perfiles de usuario**

Usuario	Descripción	Necesidades	Perfil Administrativo
Aspirante a Estudiante de Computación	Personas que desean estudiar de computación en pregrado o postgrado.	<ul style="list-style-type: none"> <li>➤ Conocer información de la Escuela: Ubicación, teléfonos, correo electrónico, carreras que ofrece.</li> <li>➤ Consultar las modalidades de ingreso.</li> <li>➤ Conocer las aptitudes requeridas para ser estudiante de computación</li> <li>➤ Conocer los estudios ofertados en postgrado</li> </ul>	➤ No tiene.
Estudiante	Estudiantes de la Licenciatura.	<ul style="list-style-type: none"> <li>➤ Consultar información sobre profesores o docentes</li> <li>➤ Consultar información de materias y horarios</li> <li>➤ Conocer las facilidades y apoyos con los que cuenta</li> <li>➤ Conocer el calendario académico.</li> <li>➤ Conocer noticias y eventos de la Escuela</li> <li>➤ Saber sobre pasantías, Seminarios y TRABAJO ESPECIAL DE GRADO ofertados</li> <li>➤ Consultar el plan de estudios (pensum)</li> <li>➤ Consultar solicitudes que puedan realizar y los recaudos necesarios</li> <li>➤ Conocer sobre Asesoramiento Académico</li> <li>➤ Conocer sobre los proyectos de servicio Comunitario ofertados.</li> </ul>	➤ No tiene
Egresado	Graduado universitario en la Licenciatura de Computación u otra.	<ul style="list-style-type: none"> <li>➤ Conocer sobre las promociones de graduandos</li> <li>➤ Conocer las ofertas de Postgrado</li> <li>➤ Conocer las oportunidades de empleo dentro de la</li> </ul>	➤ No tiene

Usuario	Descripción	Necesidades	Perfil Administrativo
		Escuela ➤ Conocer noticias y eventos de la Escuela	
Docentes	Graduado universitario, perteneciente al Personal Docente de la Escuela.	<ul style="list-style-type: none"> <li>➤ Los mismos de los estudiantes</li> <li>➤ Conocer la estructura organizativa y física de la Escuela</li> <li>➤ Consultar información sobre Investigación: Centros, Laboratorios</li> <li>➤ Consultar publicaciones asociadas a Docencia</li> <li>➤ Conocer los recursos de la Escuela y como reservarlos</li> <li>➤ Conocer acerca de los cursos de extensión de la Escuela</li> <li>➤</li> </ul>	<ul style="list-style-type: none"> <li>➤ Poder publicar documentos</li> <li>➤ Poder actualizar o eliminar documentos publicados.</li> <li>➤ Poder actualizar sus datos</li> <li>➤ Poder ofertar pasantías, trabajos, seminarios y/o trabajos especiales de grado.</li> <li>➤ Poder actualizar las ofertas publicadas.</li> <li>➤ Poder publicar noticias y eventos y modificar o eliminar las noticias y eventos publicados.</li> <li>➤ Poder cambiar su contraseña.</li> <li>➤ Actualizar Información de Centros y Laboratorio de Investigación al que pertenece.</li> <li>➤ Actualizar la información sobre la página Web de la materia que dicte.</li> </ul>
Auxiliar Docente y Preparadores	Estudiantes de la Escuela.	➤ Las mismas de los docentes	➤ No tiene actualmente.
Personal Administrativo de la Escuela	Personas en su mayoría familiarizados con el uso del computador, como herramienta de trabajo	➤ Las mismas de los docentes	<ul style="list-style-type: none"> <li>➤ Poder publicar documentos administrativos</li> <li>➤ Poder actualizar o eliminar documentos publicados.</li> <li>➤ Poder actualizar sus datos</li> <li>➤ Poder ofertar pasantías y trabajos.</li> <li>➤ Poder actualizar las ofertas publicadas.</li> <li>➤ Poder publicar noticias y eventos.</li> <li>➤ Poder modificar o eliminar las noticias y eventos publicados.</li> <li>➤ Poder cambiar su contraseña</li> </ul>
Público General	Cualquier persona interesada en consultar información de la Escuela	<ul style="list-style-type: none"> <li>➤ Conocer información de la Escuela: Ubicación, teléfonos, correo electrónico</li> <li>➤ Conocer acerca de los servicios que ofrece la Escuela</li> </ul>	➤ No tiene



Usuario	Descripción	Necesidades	Perfil Administrativo
		<ul style="list-style-type: none"> <li>➤ Saber a quién dirigirse para ofertar empleos y pasantías</li> </ul>	
Administrador	Este es un usuario particular que pertenece a la comunidad de la Escuela de Computación y que requiere identificación para acceder a sus funcionalidades	<ul style="list-style-type: none"> <li>➤</li> </ul>	<ul style="list-style-type: none"> <li>➤ Manejo de otros usuarios como Docentes, personal administrativo y preparadores</li> <li>➤ Manejo de noticias y eventos relacionados con la Escuela</li> <li>➤ Actualizar Información de Cargos, Centros y Laboratorio de Investigación, publicaciones, y demás contenido del Sitio</li> </ul>

Figura 4. 3. Perfil de Usuarios

**Descripción de los casos de uso:** A continuación se muestra el diagrama de casos de uso de nivel 0 y los diagramas de casos de uso principales por cada subsistema.

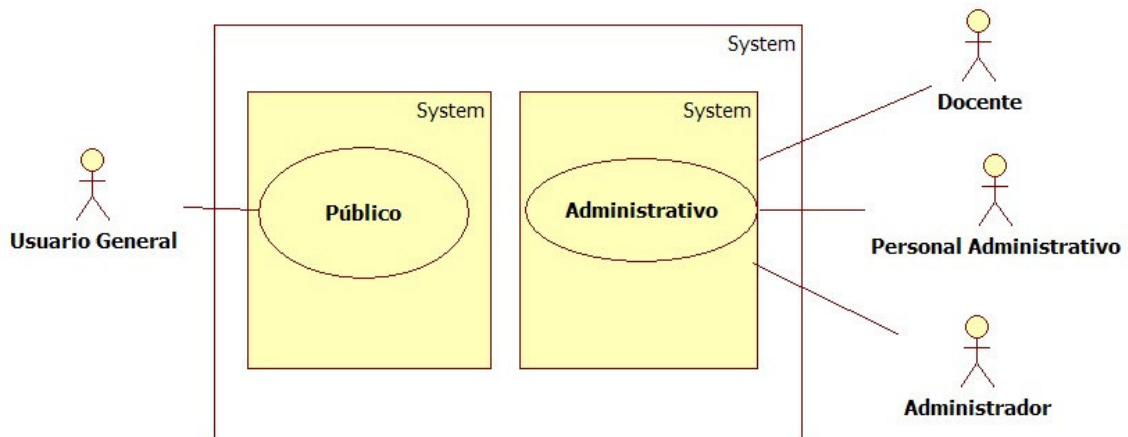


Figura 4. 4. Diagrama de casos de uso Nivel 0

➤ Subsistema público

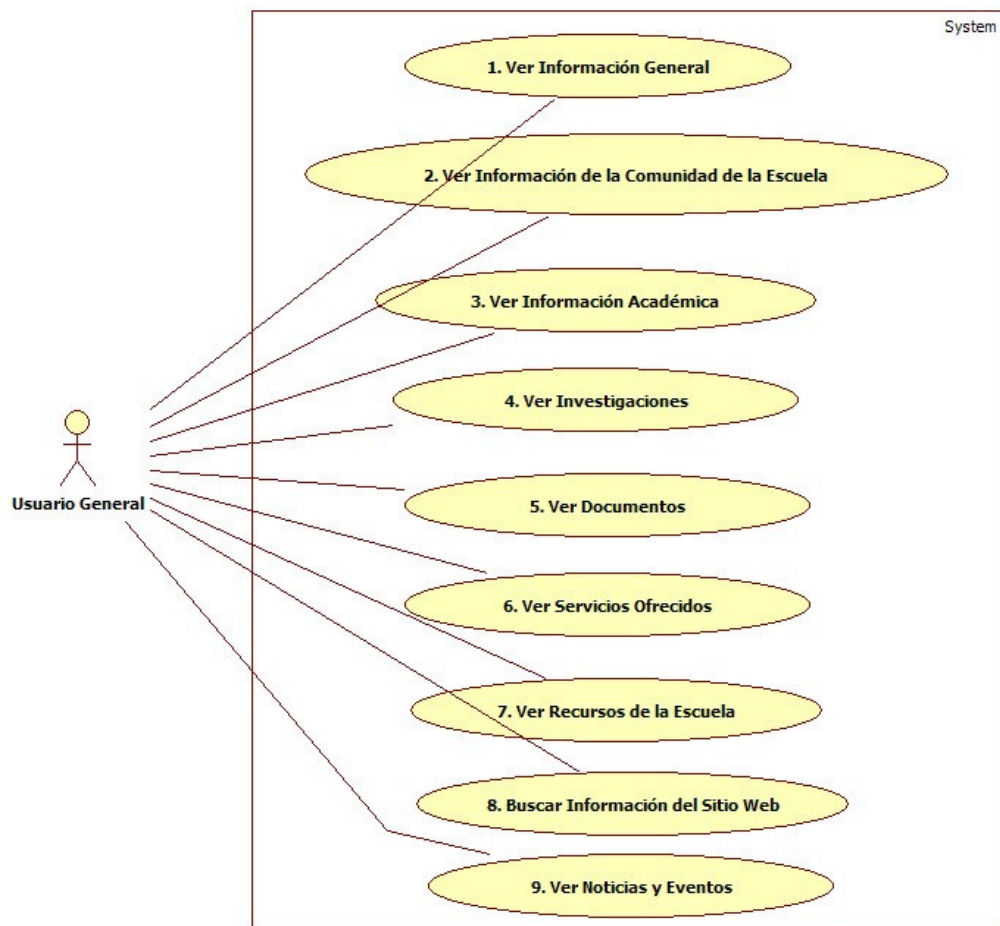


Figura 4. 5. Diagrama Casos de Uso Subsistema Público. Nivel 1

1.

**Nombre:** Ver Información General

**Precondición:** Ninguna

**Descripción:** Muestra toda la información de la Escuela referente a su historia, misión, visión, objetivos, autoridades, estructura, estudios ofrecidos e información de contacto.

2.

**Nombre:** Ver Información de la Comunidad de la Escuela

**Precondición:** Ninguna

**Descripción:** Muestra toda la información de la comunidad de la Escuela, tal como el personal docente y administrativo, información de interés para los estudiantes de pregrado y postgrado.

3.

**Nombre:** Ver Información Académica

**Precondición:** Ninguna

**Descripción:** Muestra las materias que se dictan en la Escuela, la información asociada a estas y la programación docente.

4.

**Nombre:** Ver Investigaciones

**Precondición:** Ninguna

**Descripción:** Muestra información de los Centros de Investigación, sus proyectos y todo lo referido a los seminarios y trabajos especiales de grado.

5.

**Nombre:** Ver Documentos

**Precondición:** Ninguna

**Descripción:** Muestra documentos generados por la Escuela tales como documentos de docencia, reportes técnicos, normativas y demás documentos administrativos.

6.

**Nombre:** Ver Servicios Ofrecidos

**Precondición:** Ninguna

**Descripción:** Muestra información sobre los servicios ofrecidos por la Escuela.

7.

**Nombre:** Ver Recursos de la Escuela

**Precondición:** Ninguna

**Descripción:** Muestra información sobre los recursos con los que cuenta la Escuela.

8.

**Nombre:** Buscar Información del Sitio Web

**Precondición:** Realizar una búsqueda

**Descripción:** Permite a los usuarios realizar búsquedas específicas sobre el Sitio Web.

9.

**Nombre:** Ver Noticias y Eventos de la Escuela

**Precondición:** Ninguna

**Descripción:** Muestra las noticias y eventos de interés.

➤ **Subsistema Administrativo**

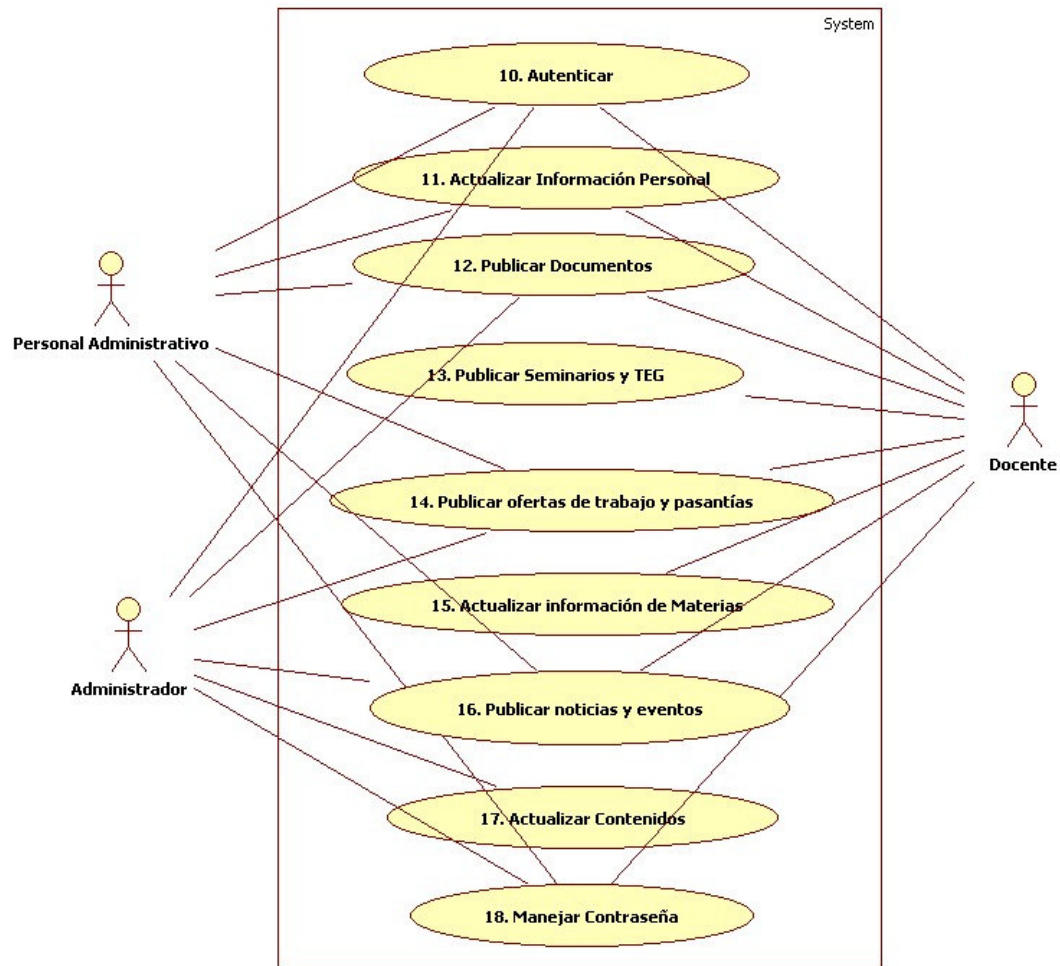


Figura 4. 6. Diagrama Casos de Uso Subsistema Administrativo. Nivel 1

10.

**Nombre:** Autenticar

**Precondición:** Ninguna

**Descripción:** Pide usuario y contraseña y verifica que el usuario está registrado en el sistema y autorizado para actualizar información.

11.

**Nombre:** Actualizar Información Personal

**Precondición:** Estar autenticado

**Descripción:** Permite al usuario ver sus datos personales y modificarlos.

12.

**Nombre:** Publicar Documentos

**Precondición:** Estar autenticado

**Descripción:** Permite al usuario ver los documentos que ha publicado, modificarlos, eliminarlos y publicar nuevos documentos.

13.

**Nombre:** Publicar Seminarios y Tesis

**Precondición:** Estar autenticado

**Descripción:** Permite al usuario ver los seminarios y tesis que ha publicado, modificarlos, eliminarlos y publicar nuevos seminarios o tesis.

14.

**Nombre:** Publicar Ofertas de Trabajo y Pasantías

**Precondición:** Estar autenticado

**Descripción:** Permite al usuario ver las ofertas de trabajo y pasantías que ha publicado, modificarlas, eliminarlas y publicar nuevas.

15.

**Nombre:** Actualizar Información de Materias

**Precondición:** Estar autenticado y pertenecer al grupo docente de la materia o ser administrador.

**Descripción:** Permite al usuario cambiar la página Web de la materia.

16.

**Nombre:** Publicar Noticias y Eventos

**Precondición:** Estar autenticado

**Descripción:** Permite al usuario ver las noticias y eventos que ha publicado, modificarlas, eliminarlas y publicar nuevas.

17.

**Nombre:** Actualizar Contenidos

**Precondición:** Estar autenticado y ser administrador

**Descripción:** Permite al usuario manejar usuarios, centros y laboratorios y autoridades.

18.

**Nombre:** Manejar de Contraseña

**Precondición:** Ser usuario con privilegios

**Descripción:** Permite al usuario cambiar su contraseña o solicitar una nueva en caso de ser olvidada.

**Modelo de Objeto de Dominio:** En la figura 4.7 se muestra el modelo de la aplicación con los principales objetos, a utilizar por ambos subsistemas, destacando que el subsistema público no requiere el objeto usuario.

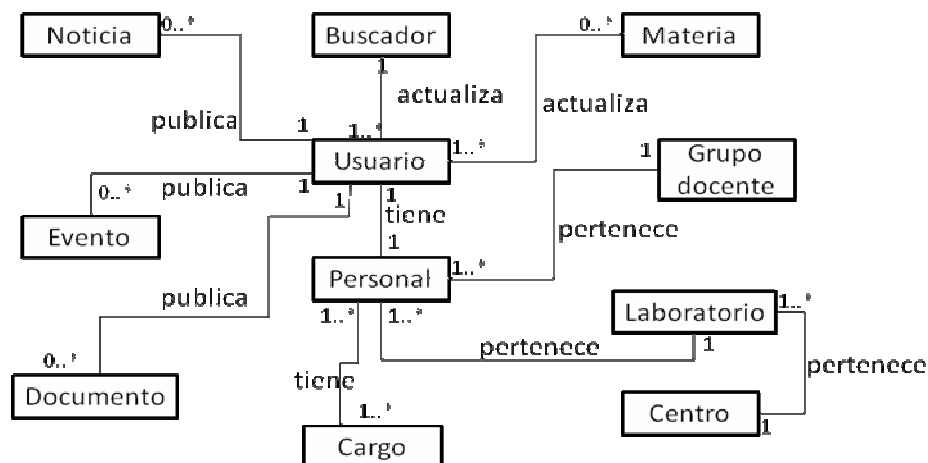


Figura 4. 7. Modelo de Objeto de Dominio

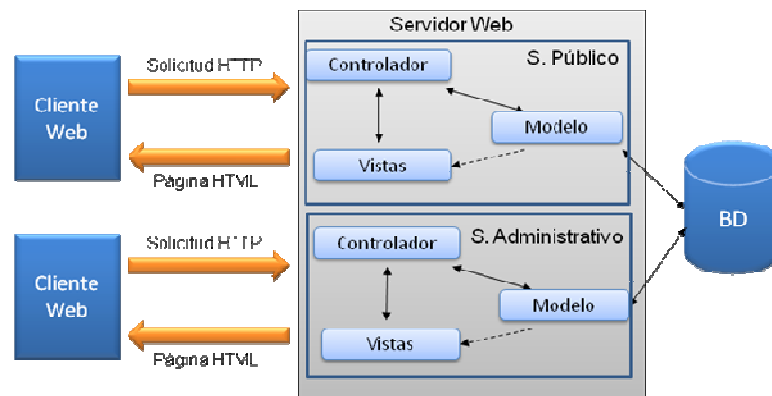
**Arquitectura inicial:** La cual será utilizada por ambos subsistemas

➤ **Arquitectura Física**

Como se indico en el punto 2.1.2 las aplicaciones Web están basadas en arquitectura cliente-servidor, esta arquitectura se muestra en la figura 2.1.1.

➤ **Arquitectura Lógica**

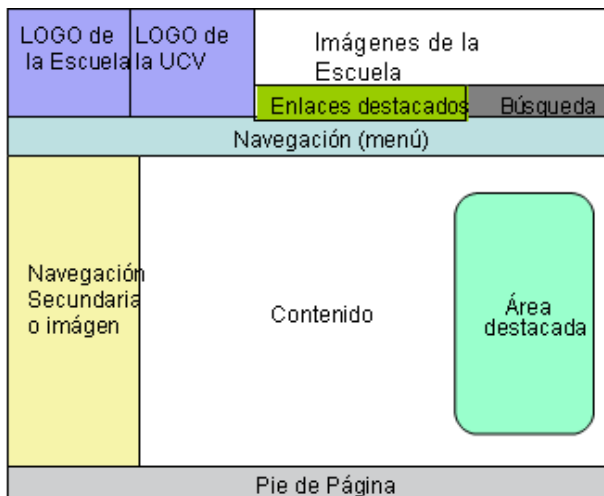
El sistema desarrollado en el presente trabajo de investigación está basado en una arquitectura de tres capas utilizando el patrón de diseño Model2 descrito en el punto 2.2.



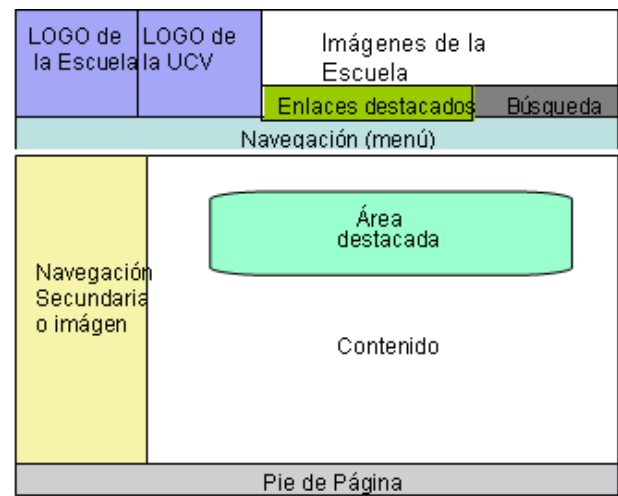
**Figura 4. 8. Arquitectura inicial**

**Prototipo de interfaz inicial:**

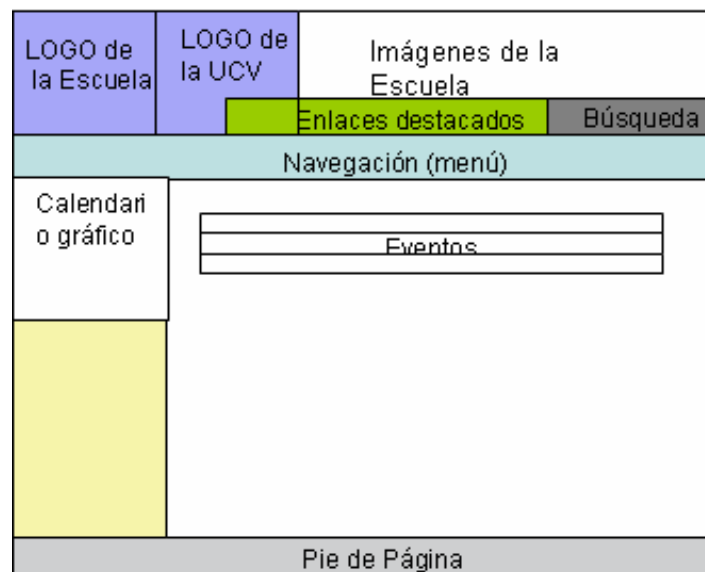
Se crearon las interfaces que se observan en las figuras 4.9, 4.10 y 4.11 como guía para el diseño de las páginas del sitio.



**Figura 4. 10. Prototipo de Interfaz Página Principal**



**Figura 4. 9. Prototipo de Interfaz Páginas Secundarias**



**Figura 4. 11. Prototipo de Interfaz Página de Eventos**

En estos prototipos se tomaron en cuenta los siguientes elementos de diseño:

Patrones de interacción:

- Homepage: Referido al diseño de la página principal como la puerta de entrada de un sitio Web, se adaptó a las necesidades de la aplicación.
- Search area: Ya que se requiere disponer de un área de búsqueda.



- Date selector: Debido a la necesidad de seleccionar fechas por parte del usuario se decidió usar un calendario gráfico ya que es menos propenso a errores de entrada.
- Even Calendar: Se requiere crear una página que muestre eventos pasados o futuros.

Lineamientos de usabilidad:

Navegación

- Identificación del Sitio en todas sus páginas usando para esto el logo de la Escuela, en la parte superior izquierda.
- Disposición de un área para navegación o menú claramente definida, donde se dispongan los enlaces a las demás páginas del Sitio.

Contenido:

- Uso del pie de página para colocar información de contacto de primera mano.

Diseño:

- Uso de áreas de diferente color para destacar ideas principales o secciones de contenido importante.
- Uso de áreas para imágenes que ayuden al reconocimiento del Sitio por los usuarios y que puedan ser usadas para crear un sentido de pertenencia.

**Reglas de negocio:** Se observó la necesidad de identificar algunos lineamientos propios de la Escuela de Computación con la finalidad de tomar en cuenta aspectos propios de los procesos y actividades de la institución, que pudieran definir algunos comportamientos de la aplicación, los cuales se denominarán reglas de negocio.

1. Los documentos de docencia pertenecen a un docente.
2. Un documento de docencia sólo puede ser publicado, modificado o eliminado por el docente al que pertenece.
3. Los documentos administrativos deben ser publicados por personal administrativo.
4. Los seminarios y TRABAJO ESPECIAL DE GRADO pueden ser ofertados por un docente.
5. Las fechas de presentación de un TRABAJO ESPECIAL DE GRADO deben ser publicadas por el tutor académico.

6. Los seminarios y TRABAJO ESPECIAL DE GRADO son creados de manera separada, el TRABAJO ESPECIAL DE GRADO puede ser creado a partir de un seminario existente o no, para tomar en cuenta los cambios de tema que pueda haber.
7. Las noticias y eventos pueden ser publicados y por cualquier usuario autenticado.
8. El usuario que publique una noticia o evento será responsable de este, para su modificación o eliminación.
9. Las pasantías y trabajos pueden ser publicados por cualquier usuario autenticado, y este puede o no hacerse responsable o ser contacto de la pasantía o trabajo.
10. La información de las materias, por tratarse de datos que provienen de control de estudios no podrán ser modificados, a excepción de la dirección de la página Web la cual podrá ser modificada por el grupo docente.
11. Puede haber más de un Administrador.
12. La información de los centros y laboratorios puede ser modificada por los Administradores o por cualquier miembro del centro a modificar.
13. Los usuarios autenticados podrán cambiar su contraseña en el momento que deseen.
14. Cada usuario autenticado puede editar su información personal.
15. Los administradores podrán crear o eliminar usuarios, centros, laboratorios y cambiar autoridades.

#### **4.5.2 ELABORACIÓN:**

##### **Diagramas de Casos de Uso**

**1. Información General:** Este caso de uso está relacionado con la Información orientada a dar a conocer la Escuela y su estructura organizativa.

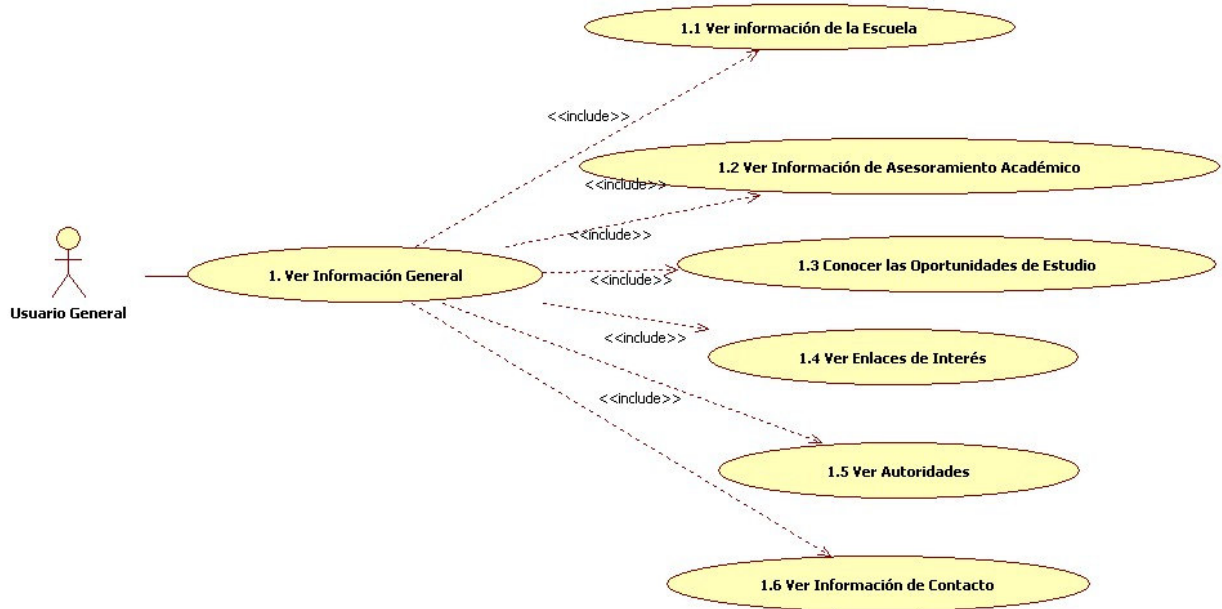


Figura 4. 12. Diagrama de Casos de Uso Ver Información General. Nivel 2

### 1.1

**Nombre:** Ver Información de la Escuela

**Precondición:** Seleccionar Información General

**Descripción:** Se muestra al usuario la misión, visión, objetivos e historia de la Escuela.

### 1.2

**Nombre:** Ver Información de Asesoramiento Académico

**Precondición:** Seleccionar Información General.

**Descripción:** Se muestra información acerca de la asesoría académica que se presta en la Escuela.

### 1.3

**Nombre:** Conocer las Oportunidades de Estudio

**Precondición:** Seleccionar Información General.

**Descripción:** Se da a conocer las modalidades de ingreso a la Escuela y el plan de Estudio de la carrera.

### 1.4

**Nombre:** Ver Enlaces de Interés

**Precondición:** Seleccionar Información General.

**Descripción:** Se muestran enlaces a otras páginas de interés para la Escuela.

### **1.5**

**Nombre:** Ver Autoridades

**Precondición:** Seleccionar Información General.

**Descripción:** Se muestra la lista de autoridades de la Escuela (cargo y profesor que lo ocupa)

### **1.6**

**Nombre:** Ver Información de Contacto

**Precondición:** Seleccionar Información General.

**Descripción:** Se muestra información que permite al usuario ubicar físicamente la Escuela o contactar telefónicamente o por correo electrónico.

**2. Ver Comunidad de la Escuela:** Su objetivo es mostrar información relacionada con la comunidad o sobre los beneficios que le aporta.

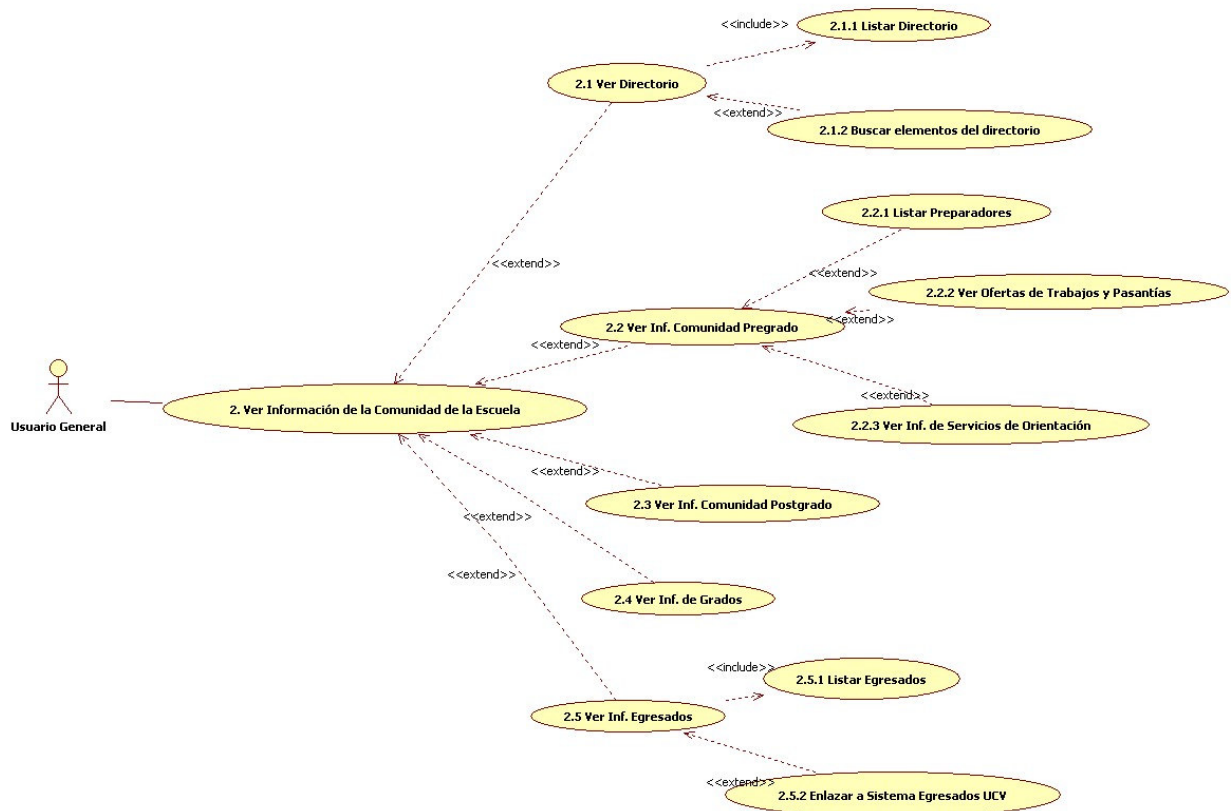


Figura 4. 13. Diagrama de Casos de Uso Ver Información de la Comunidad de la Escuela. Nivel 2

## 2.1

**Nombre:** Ver Directorio

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se lista el personal que labora en la Escuela y se permite realizar búsquedas por nombre, por tipo y área profesional, esta última sólo en el caso de los docentes.

### 2.2.1

**Nombre:** Listar Preparadores

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se muestra la lista de preparadores de la Escuela de Computación.

### 2.2.2

**Nombre:** Ver Ofertas de Trabajos y Pasantías

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se muestran las ofertas de trabajo o pasantías.

### 2.2.3

**Nombre:** Ver Información Servicio de Orientación

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se enlaza a la página del Servicio de Orientación.

### 2.3

**Nombre:** Ver Información de la Comunidad Postgrado

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se enlaza con el sistema de postgrado.

### 2.4

**Nombre:** Ver Información de Grados

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se muestra información referente a las graduaciones de pregrado y postgrado del año..

### 2.5

**Nombre:** Ver Información de Egresados

**Precondición:** Seleccionar Comunidad.

**Descripción:** Se listan los egresados de la Escuela de Computación y además se ofrece un enlace con el sistema de Egresados de la UCV.

**3. Ver Información Académica:** su objetivo es mostrar la información académica de la Escuela

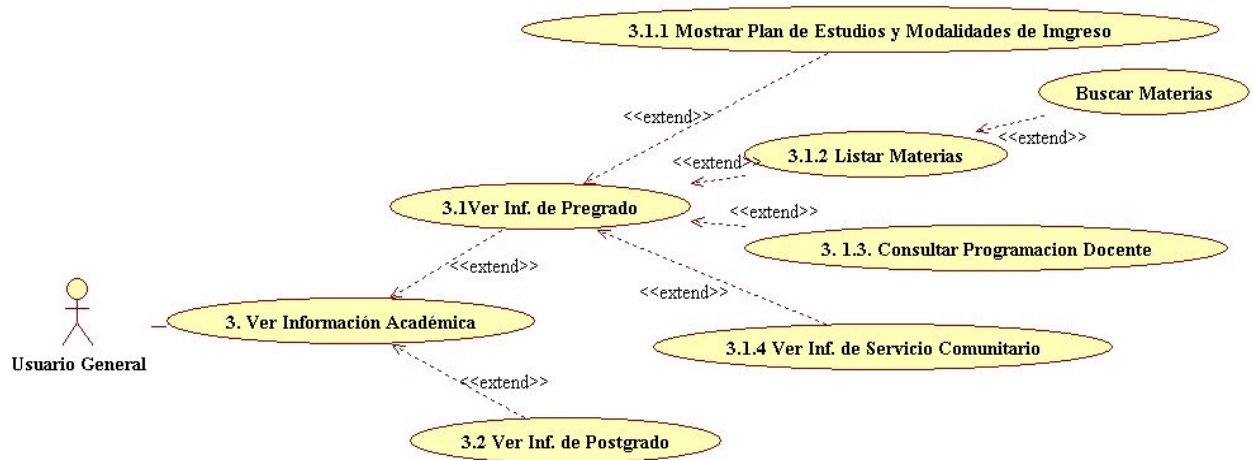


Figura 4. 14. Diagrama de Casos de Uso Ver Información Académica. Nivel 2

### 3.1.1

**Nombre:** Mostrar Plan de Estudios y Modalidades de Ingreso

**Precondición:** Seleccionar Académico.

**Descripción:** Muestra la información referente al plan de estudios de la carrera y las diferentes modalidades de ingreso a la Escuela.

### 3.1.2

**Nombre:** Listar Materias

**Precondición:** Seleccionar Académico.

**Descripción:** Se muestran las materias que se dictan en la carrera, y se permite realizar la búsqueda de materias.

### 3.1.3

**Nombre:** Consultar Programación Docente

**Precondición:** Seleccionar Académico.

**Descripción:** Se muestra la programación docentes.

### 3.1.4

**Nombre:** Ver Información Servicio Comunitario

**Precondición:** Seleccionar Académico.

**Descripción:** Se muestran información referente al Servicio Comunitario.

**4. Ver Investigaciones:** muestra toda la información referente al área de investigación de la Escuela.

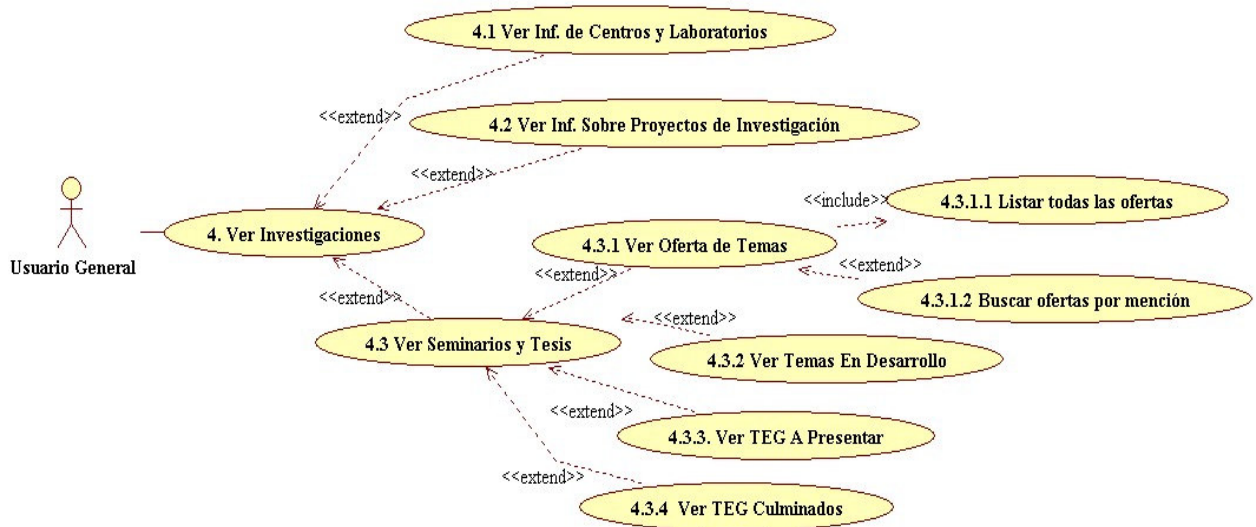


Figura 4. 15. Diagrama de Casos de Uso Ver Investigaciones. Nivel 2

#### 4.1

**Nombre:** Ver Información de Centros y Laboratorios

**Precondición:** Seleccionar Investigación

**Descripción:** Muestra información sobre los Centros y Laboratorios que hacen vida en la Escuela de Computación y además se ofrece el enlace la página de cada uno.

#### 4.2

**Nombre:** Ver Información Sobre Proyectos de Investigación

**Precondición:** Seleccionar Investigación

**Descripción:** Se listan los Proyectos de Investigación que se llevan a cabo por cada Centro de Investigación.

#### 4.3.1

**Nombre:** Ver Ofertas de Temas

**Precondición:** Seleccionar Investigación

**Descripción:** Se listan las ofertas de temas de investigación y se permite buscar por menciones.



4.3.2

**Nombre:** Ver Temas en Desarrollo

**Precondición:** Seleccionar Investigación

**Descripción:** Se listan los trabajos de investigación que están en desarrollo.

4.3.3

**Nombre:** Ver Trabajos Especiales de Grado a Presentar

**Precondición:** Seleccionar Investigación

**Descripción:** Se listan los Trabajos Especiales de Grado que se van a presentar.

4.3.4

**Nombre:** Ver Temas en Culminados

**Precondición:** Seleccionar Investigación

**Descripción:** Se listan los trabajos de investigación que han sido culminados.

**5. Ver Noticias y Eventos:** muestra las noticias y eventos publicados de la Escuela.

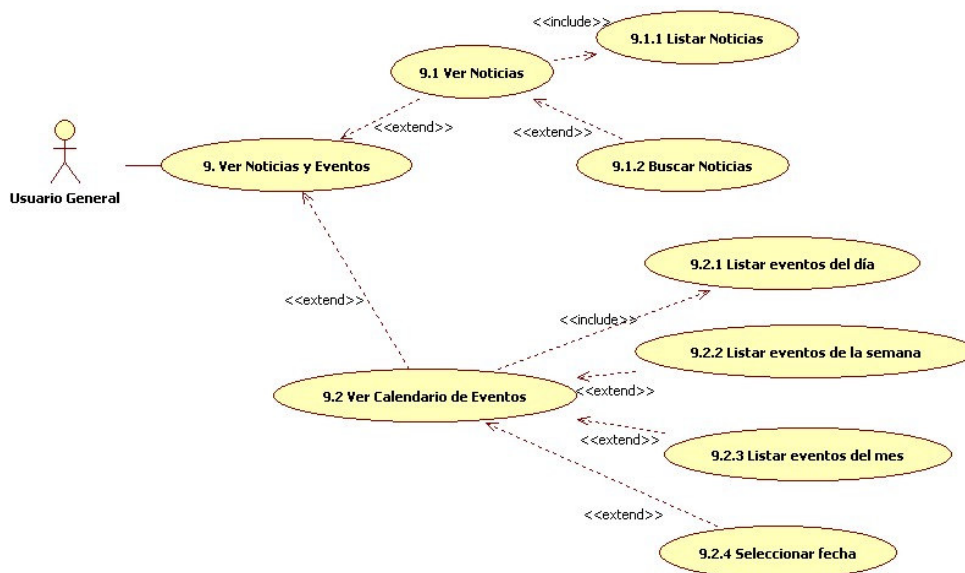


Figura 4. 16. Diagrama de Casos de Uso Ver Noticias y Eventos. Nivel 2

9.1

**Nombre:** Ver Noticias

**Precondición:** Seleccionar Información General o seleccionar una noticia de la sección de noticias.

**Descripción:** Se listan las noticias publicadas.

### 9.2.1

**Nombre:** Ver Eventos del día

**Precondición:** Seleccionar Información General o seleccionar una fecha en el calendario de la sección de eventos.

**Descripción:** Se muestran los eventos publicados del día actual o seleccionado.

### 9.2.2

**Nombre:** Ver Eventos de la semana

**Precondición:** Seleccionar Información General o seleccionar una fecha en el calendario de la sección de eventos.

**Descripción:** Permite ver todos los eventos de la semana.

### 9.2.3

**Nombre:** Ver Eventos del mes

**Precondición:** Seleccionar Información General o seleccionar una fecha en el calendario de la sección de eventos.

**Descripción:** Permite ver todos los eventos del mes.

### 9.2.4

**Nombre:** Seleccionar Fecha

**Precondición:** Seleccionar Información General o seleccionar una fecha en el calendario de la sección de eventos.

**Descripción:** Permite seleccionar una fecha específica a mostrar los eventos.

## Subsistema Privado

**6. Publicar Documentos:** permite publicar noticias y eventos así como manipular los publicados.

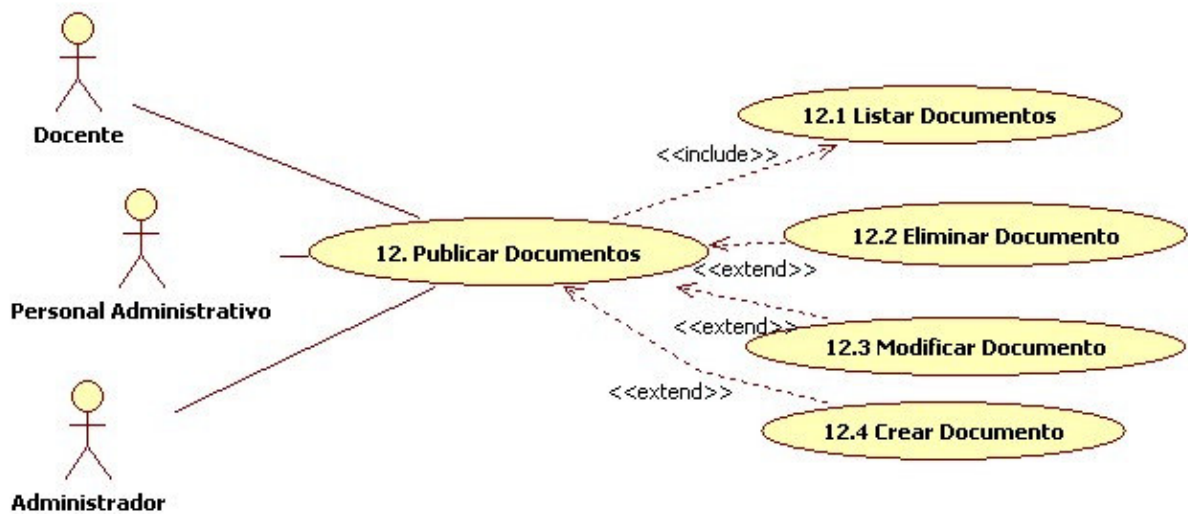


Figura 4. 17. Diagrama de Casos de Uso Publicar Documentos. Nivel 2

### 12.1

**Nombre:** Listar Documento

**Precondición:** Estar Autenticado

**Descripción:** Muestra al usuario autenticado los documentos que ha publicado.

### 12.2

**Nombre:** Eliminar Documento

**Precondición:** Estar Autenticado y haber seleccionado un documento.

**Descripción:** Permite al usuario autenticado eliminar documentos que ha publicado.

### 12.3

**Nombre:** Modificar Documento

**Precondición:** Estar Autenticado y haber seleccionado un documento.

**Descripción:** Permite al usuario autenticado modificar documentos que ha publicado.

### 12.4

**Nombre:** Crear Documento

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado publicar documentos.

**7. Publicar Seminarios y TRABAJO ESPECIAL DE GRADO:** permite publicar ofertas de seminarios y trabajos especiales de grado y manipular los existentes.

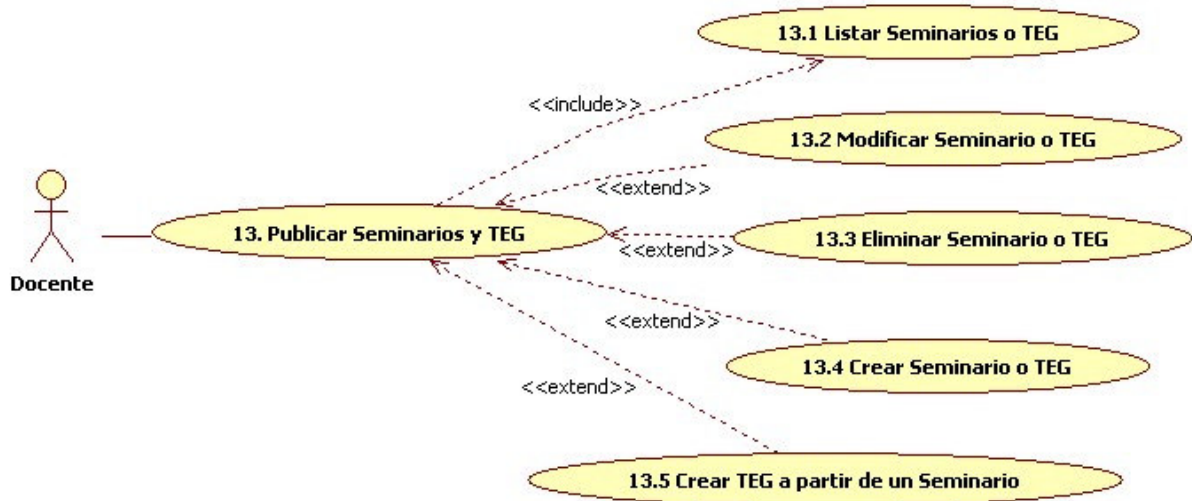


Figura 4. 18. Diagrama de Casos de Uso Publicar Seminarios y Trabajos Especiales de Grado. Nivel 2

### 13.1

**Nombre:** Listar seminarios y trabajos especiales de grado

**Precondición:** Estar Autenticado

**Descripción:** Muestra al usuario autenticado seminarios y trabajos especiales de grado publicados por él.

### 13.2

**Nombre:** Modificar seminarios o trabajos especiales de grado

**Precondición:** Estar Autenticado y haber seleccionado un seminario o trabajo especial de grado.

**Descripción:** Permite al usuario autenticado modificar seminario o trabajo especial de grado que ha publicado.

### 13.3

**Nombre:** Eliminar seminario o trabajo especial de grado

**Precondición:** Estar Autenticado y haber seleccionado seminario o trabajo especial de grado.

**Descripción:** Permite al usuario autenticado eliminar un seminario o trabajo especial de grado que haya publicado.

13.4

**Nombre:** Crear seminario o trabajo especial de grado

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado publicar seminarios o trabajos especiales de grado.

13.5

**Nombre:** Crear un trabajo especial de grado a partir de un seminario

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado crear un trabajo especial de grado a partir de los datos de un seminario existente.

**8. Publicar ofertas de trabajo y pasantía:** permite publicar ofertas de trabajo y pasantía y manipular los existentes.

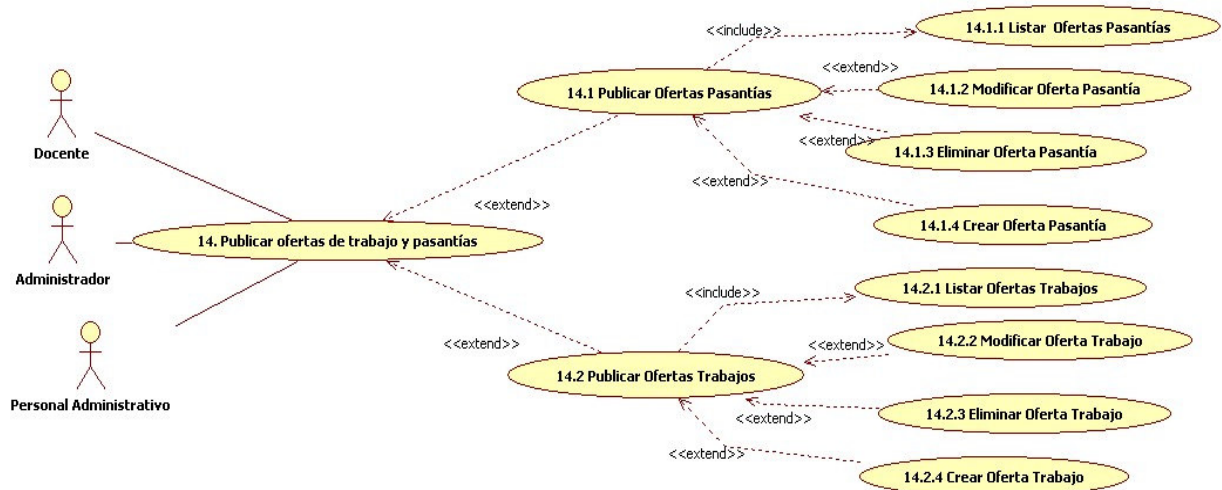


Figura 4. 19. Diagrama de Casos de Uso Publicar Ofertas de Trabajo y Pasantías. Nivel 2

14.1.1

**Nombre:** Listar Ofertas Pasantías

**Precondición:** Estar Autenticado

**Descripción:** Muestra al usuario autenticado las ofertas de pasantías que ha publicado.

14.1.2

**Nombre:** Modificar Oferta de Pasantía

**Precondición:** Estar Autenticado y haber seleccionado una oferta de pasantía.

**Descripción:** Permite al usuario autenticado modificar las ofertas de pasantías que ha publicado.

#### 14.1.3

**Nombre:** Eliminar Oferta de Pasantía

**Precondición:** Estar Autenticado y haber seleccionado una oferta de pasantía.

**Descripción:** Permite al usuario autenticado eliminar ofertas de pasantías que ha publicado.

#### 14.1.4

**Nombre:** Crear Oferta de Pasantía

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado publicar ofertas pasantías.

#### 14.2.1

**Nombre:** Listar Ofertas Trabajos

**Precondición:** Estar Autenticado

**Descripción:** Muestra al usuario autenticado las ofertas de trabajos que ha publicado.

#### 14.2.2

**Nombre:** Modificar Oferta de Trabajo

**Precondición:** Estar Autenticado y haber seleccionado una oferta de trabajo.

**Descripción:** Permite al usuario autenticado modificar las ofertas de trabajos que ha publicado.

#### 14.2.3

**Nombre:** Eliminar Oferta de Trabajo

**Precondición:** Estar Autenticado y haber seleccionado una oferta de trabajo.

**Descripción:** Permite al usuario autenticado eliminar ofertas de trabajos que ha publicado.

#### 14.2.4

**Nombre:** Crear Oferta de Trabajo

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado publicar ofertas trabajos.

**9. Publicar Noticias y Eventos:** permite publicar ofertas de noticias y eventos y manipular los existentes.

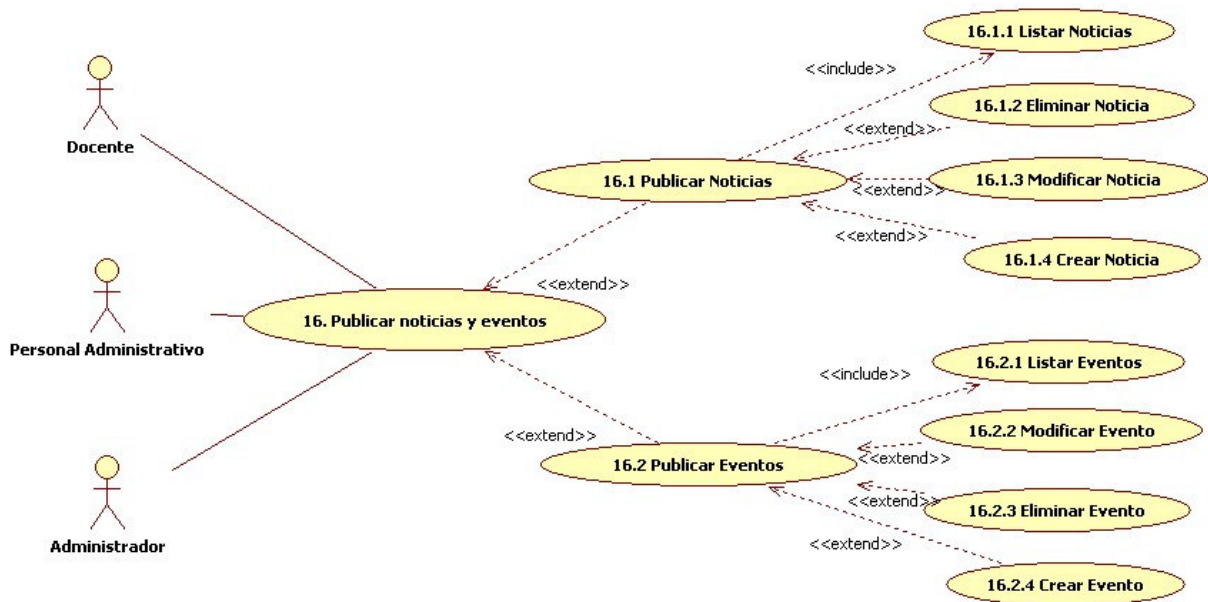


Figura 4. 20. Diagrama de Casos de Uso Publicar Noticias y Eventos. Nivel 2

### 16.1.1

**Nombre:** Listar Noticias

**Precondición:** Estar Autenticado

**Descripción:** Muestra al usuario autenticado las noticias que ha publicado.

### 16.1.2

**Nombre:** Modificar Noticia

**Precondición:** Estar Autenticado y haber seleccionado una noticia.

**Descripción:** Permite al usuario autenticado modificar las noticias que ha publicado.

### 16.1.3

**Nombre:** Eliminar Noticia

**Precondición:** Estar Autenticado y haber seleccionado una noticia.

**Descripción:** Permite al usuario autenticado eliminar noticias que ha publicado.

### 16.1.4

**Nombre:** Crear Noticia

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado publicar noticias.

### 16.2.1

**Nombre:** Listar Eventos

**Precondición:** Estar Autenticado

**Descripción:** Muestra al usuario autenticado los eventos que ha publicado.

### 16.2.2

**Nombre:** Modificar Evento

**Precondición:** Estar Autenticado y haber seleccionado un evento.

**Descripción:** Permite al usuario autenticado modificar los eventos que ha publicado.

### 16.2.3

**Nombre:** Eliminar Evento

**Precondición:** Estar Autenticado y haber seleccionado un evento.

**Descripción:** Permite al usuario autenticado eliminar eventos que ha publicado.

### 16.2.4

**Nombre:** Crear Evento

**Precondición:** Estar Autenticado

**Descripción:** Permite al usuario autenticado publicar eventos.

### **Prototipo de Interfaz dinámico.**

En esta fase se tomaron los prototipos de interfaz iniciales y se llevaron a prototipos funcionales, en la figura 4.21 y la figura 4.22 se muestran interfaces de páginas del subsistema público. Para la creación de estas interfaces se siguieron aplicando elementos de diseño, los cuales son:

Estándares Web:

- Se uso como lenguaje de marcado XHTML, y como lenguaje de formato CSS.
- Se separó en la mayor medida posible, el contenido del formato, y se limitó el uso de JavaScript de manera que el contenido fuera parte del código de la página.
- Para accesibilidad que el color de fondo permitiera leer el contenido, en su mayoría se uso fondo blanco con letras de color gris oscuro. Donde el fondo fuera gris o azul se usó letra negra.



- Evitar el uso de los colores verde y el rojo para resaltar contenidos, en caso de usarlo agregar otros elementos.

Patrones de interacción:

- En este punto se incorporó el patrón de interacción Fly out menú debido a la necesidad de crear un menú desplegable, para que el usuario pueda tener acceso a todas las opciones de navegación.

Lineamientos de usabilidad:

- En el menú desplegable no pueden haber más de 7 opciones ya que el ser humano puede retener hasta un máximo de 7 elementos a la vez.
- Clasificación y agrupamiento de los enlaces de manera lógica.
- Uso del color de la Escuela (azul), colores contrastantes como el naranja y colores complementarios como gris y otros azules de la gama.
- Formato de texto sans serif, de un tamaño no menor de 12px.
- Se definió un diseño parcialmente líquido para que la aplicación se vea correctamente en resoluciones pequeñas (800 x 600) y que no se pierda mucho espacio de pantalla en resoluciones grandes.



Figura 4. 21. Interfaz Página Principal



Figura 4. 22. Interfaz Página Secundaria. Página de Materias

**Diagrama de clases:** En la figura 4.23 se muestra el diagrama de clases persistentes del subsistema administrativo, el cual se diferencia del subsistema público por la clase usuario, la cual no está presente en el subsistema público.

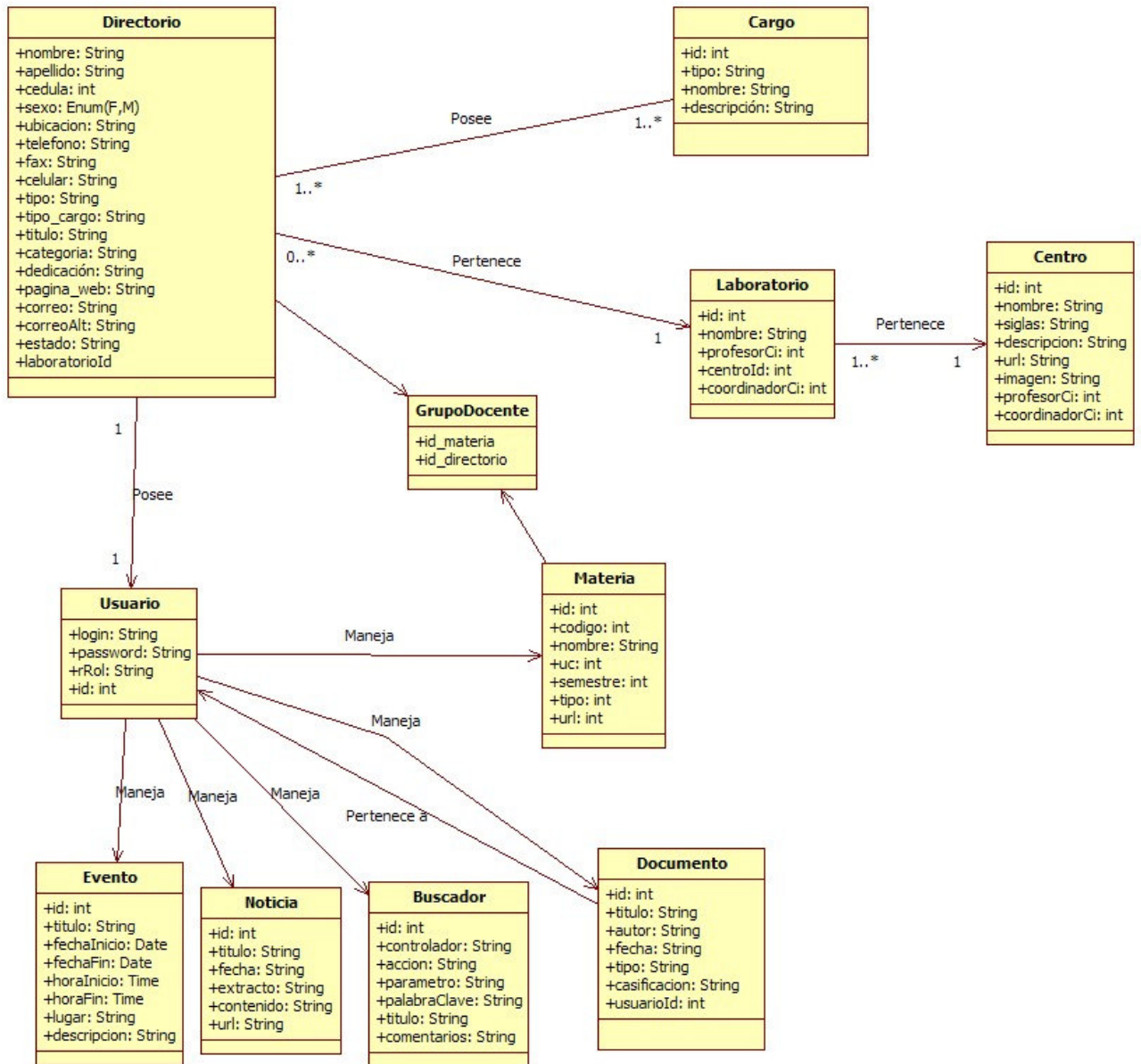


Figura 4. 23. Diagrama de Clases Persistentes

## Plataforma de Desarrollo

Hardware (Características mínimas):

- Pentium 4 o equivalente.
- Memoria RAM 512Mb.
- Disco Duro 40Gb.
- Monitor SVGA.

Software:

- Ruby 1.8.6.
- Rails 1.2.6.
- MySql 5.0.
- XHTML
- JavaScript 1.3.
- CSS 2.
- Subversion.
- Webrick como servidor Web de desarrollo.
- Aptana Studio 1.1.6 con RadRails como entorno de programación.
- Varios tipos de navegadores: Explorer, Mozilla, Opera y Safari.

### 4.5.3 CONSTRUCCIÓN:

En esta fase el desarrollo se dividió en dos etapas, la primera referida a la parte pública de la aplicación denominada “Subsistema público”, el cual muestra la información a los usuarios generales, y una segunda etapa que se refiere a la parte administrativa y se denomina “Subsistema Administrativo, y es aquella donde se crean, eliminan o modifican elementos y que va dirigido a personal de la Escuela ya que requiere elementos de seguridad como autenticación de usuario para protección de los datos.

Para la primera etapa se definieron 5 iteraciones y para la segunda etapa se definieron 6 iteraciones, no todas se definieron al inicio de la fase. Las iteraciones del sistema administrativo se llevaron a cabo una vez completada su contraparte del sistema público, a excepción de la primera iteración, ya que debido a la complejidad de la información manejada allí su contraparte administrativa se llevó a cabo al final.

En la Figura 4.24. se puede observar el Cronograma de desarrollo de las iteraciones correspondientes a esta fase:

<b>Iteración</b>	<b>Duración</b>
1	12 semanas
2	4 semanas
3	6 semanas
4	4 semanas
5	4 semanas
6	4 semanas
7	2 semanas
8	2 semanas
9	2 semanas
10	2 semanas
11	4 semanas

**Figura 4. 24. Cronograma de desarrollo**

#### **4.5.3.1 Subsistema Público:**

##### **Iteración 1**

Esta iteración fue la más larga ya que se comenzó a desarrollar en PHP debido a requerimientos iniciales del cliente. Luego cuando se migró a Ruby on Rails el desarrollo se aceleró considerablemente.

En esta iteración se desarrollaron los casos de uso referido a la publicación de información general de la Escuela, definiendo detalladamente los requerimientos de este y las funcionalidades asociadas. En esta iteración se crearon las páginas correspondientes a Misión, Visión, Objetivos, Historia, Autoridades, Departamento, Comisiones, Estudios Ofrecidos e Información de Contacto

##### **Diagrama de Clases**

En esta etapa de la Construcción se elaboró el Diagrama de Clases persistente correspondiente a los elementos del modelo asociados a las funcionalidades correspondientes a la primera iteración, específicamente a Ver Autoridades. Cada una de estas clases hereda de la clase ActiveRecord.

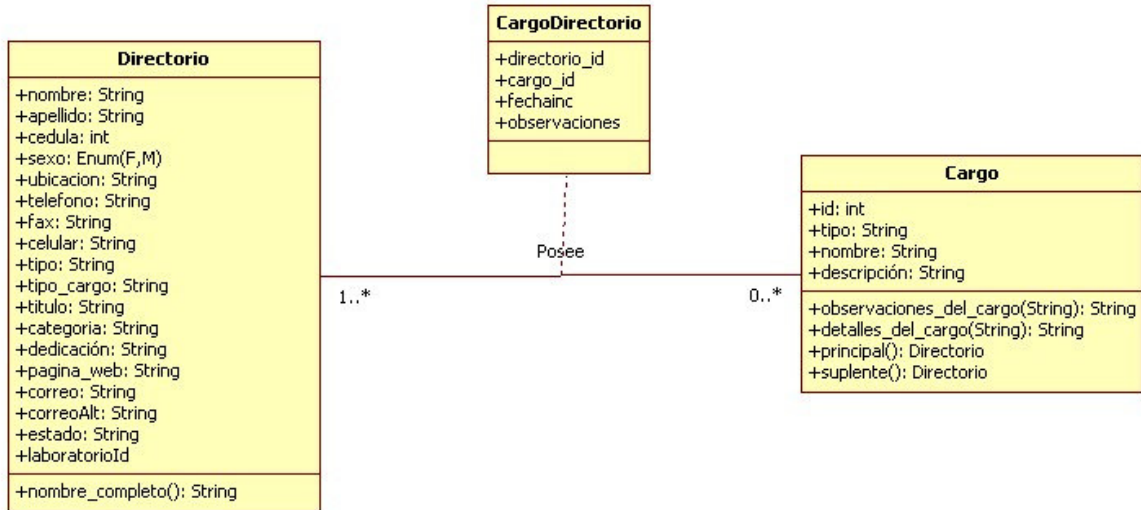


Figura 4. 25. Diagrama de Clases Persistentes Iteración 1

La implementación de esta funcionalidad requiere que se muestren los cargos y cada uno de los elementos del directorio asociados al cargo.

Las demás funcionalidades no llevan asociados elementos del modelo

### Principales Interfaces



Figura 4. 26. Interfaz Autoridades



Figura 4. 27. Interfaz Asesoría Académica

## Iteración 2:

En esta iteración se desarrollaron los casos de uso referido a las noticias y eventos. Para la funcionalidad de eventos se utilizó un plugin de rails llamado `calendar_date_select` lo que facilitó la creación de calendarios dinámicos y el manejo de fechas.

**Diagrama de Clases:** A continuación se muestra el diagrama de clases persistente de la iteración 2, donde se observan las clases `Evento` y `Noticia`, y la clase `Directorio`, ya que este es responsable de un evento o noticia,

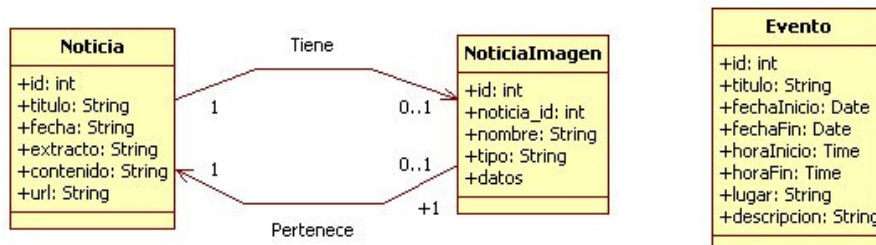


Figura 4. 28. Diagrama de Clases Persistentes Iteración 2

## Principales Interfaces

Para mostrar los eventos se diseñaron las vistas `calendario_dia`, `calendario_semana`, `calendario_mes`, las cuales reciben una fecha y los eventos asociados al día, a la semana o al mes de esta fecha respectivamente.



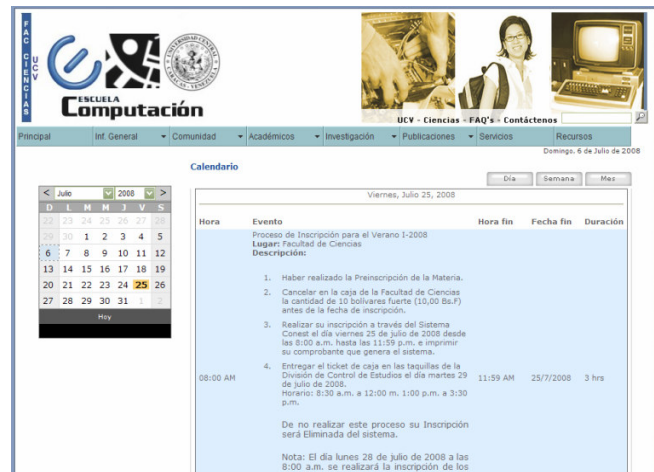


Figura 4. 29. Interfaz Vista Eventos del Día

Desde el menú se accede a *calendario\_dia*, las demás vistas se pueden visitar a través de las vistas mencionadas en el párrafo de arriba.

Las Noticias se pueden acceder desde el menú Información General o desde la página principal.

La vista *noticias* muestra las noticias del año ordenadas por mes desde la más reciente, o muestra la noticia seleccionada en la página principal.

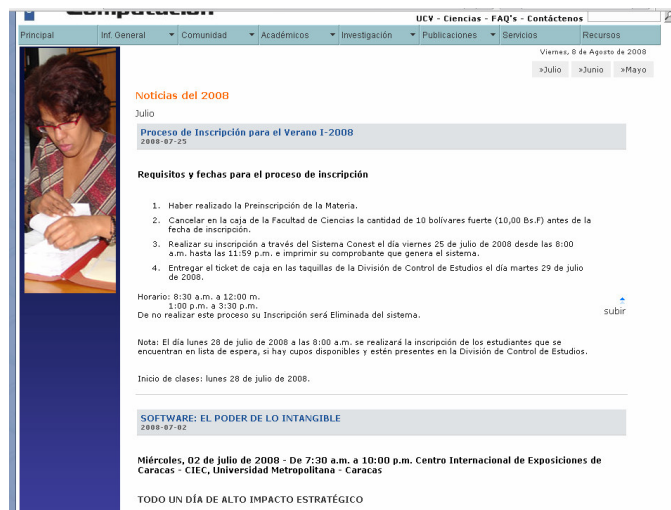


Figura 4. 30. Interfaz Vista Noticia

### Iteración 3:

En esta iteración se desarrollaron los casos de uso referido a la información de la Comunidad de la Escuela, definiendo detalladamente los requerimientos de este y las funcionalidades asociadas. Es esta



iteración se desarrollaron las páginas correspondientes a: Directorio de la Escuela, Ofertas de pasantías y trabajos, y promociones de grados.

### Diagrama de Clases

Se incorporaron las clases PasantiaTrabajo y Egresado, para las funcionalidades de mostrar las pasantías y trabajos ofertados, y para mostrar las promociones de graduados de diferentes años, la Clase Directorio se usa para la funcionalidad Directorio

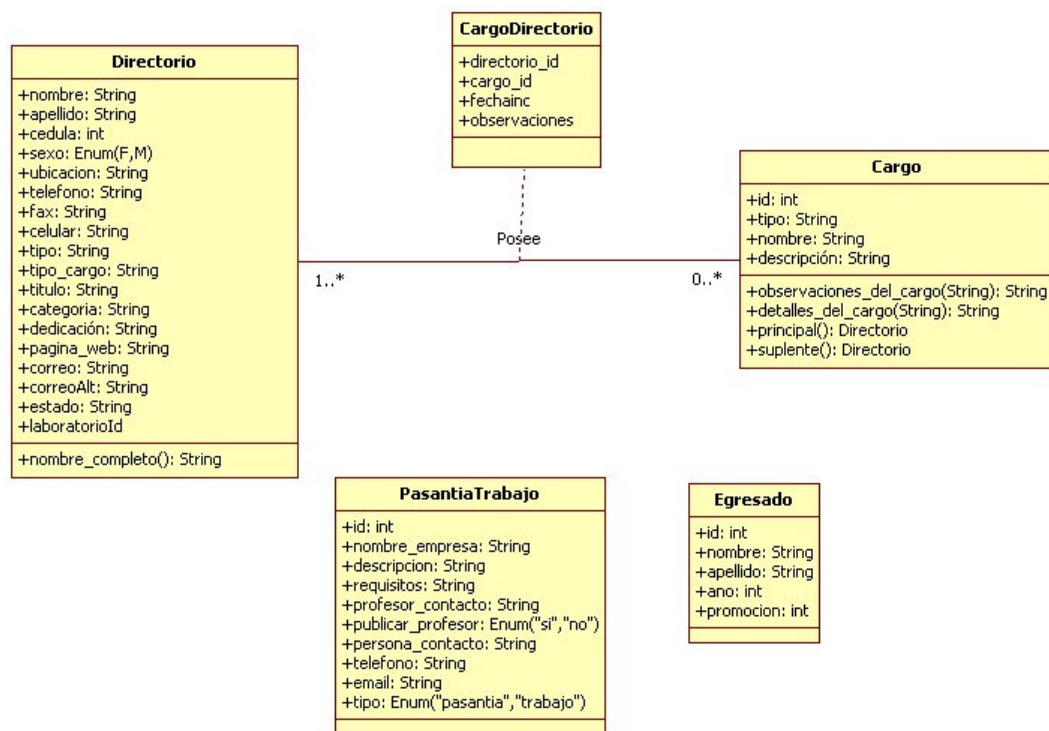


Figura 4. 31. Diagrama de Clases Persistentes Iteración 3

Para las funcionalidades de Comunidad se creó el controlador ComunidadController el cual hereda del ApplicationController, en la figura 4.32 se muestra el diagrama de clases correspondiente a este controlador.



**Figura 4. 32. Controlador asociado a la funcionalidad Comunidad**

A continuación se muestra el código asociado a la oferta de pasantías y trabajos, el método pasantía\_trabajo, llama a la vista pasantía\_trabajo pasando los tipos para que la vista construya el select.

El método list\_pasantia\_trabajos es llamado, via ajax, desde la vista pasantía\_trabajo y se encarga de construir el listado según el tipo seleccionada, actualizando la vista pasantía\_trabajo

```

def pasantia_trabajo
  @tipos = PasantiaTrabajo.find(:all, :select => 'DISTINCT tipo')
end

#metodo que muestra la lista de trabajos o pasantías, segun la búsqueda, en ajax

def list_pasantia_trabajos
  tipo = params[:tipo] || 'todos'
  if tipo!='todos'
    @pasantia_trabajo_pages, @pasantia_trabajos = paginate :pasantia_trabajos, :per_page => 2,
:conditions => ['tipo=?', tipo]
  end
  render :layout => false
end
    
```

**Figura 4. 33. Código asociado a la Oferta de Trabajos y Pasantías**

### Principales Vistas o Interfaces

La vista para ofertas de Pasantías y trabajos consulta via ajax las ofertas de trabajo o pasantías mostrandolas en la parte inferior



Figura 4. 34. Interfaz Ofertas de Pasantías y Trabajos

La vista *Directorio* muestra el listado de Personal de la Escuela, almacenado en Directorio y un buscador que permite escoger el tipo (Docente o personal administrativo), o el nombre o apellido, y en caso de ser Docente, escoger el área profesional. El resultado de ejecutar la búsqueda se muestra en la misma vista para facilitarle la labor al usuario.

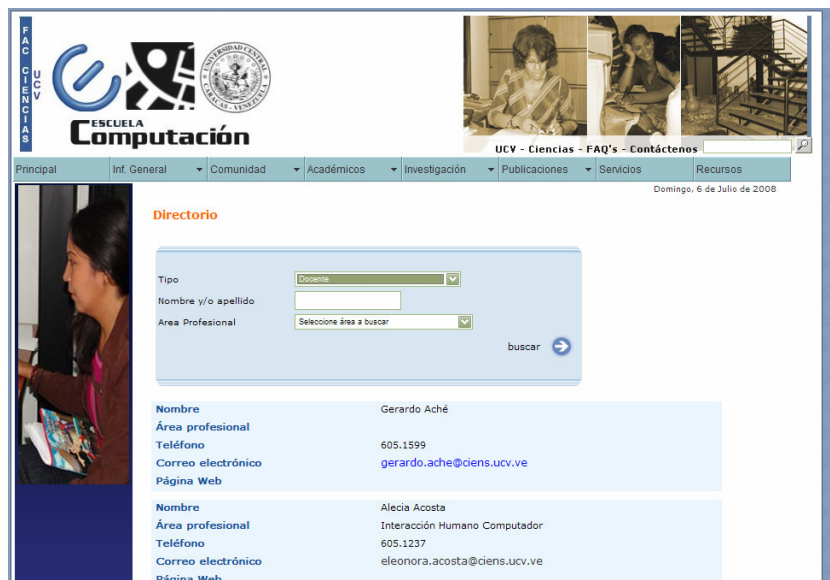


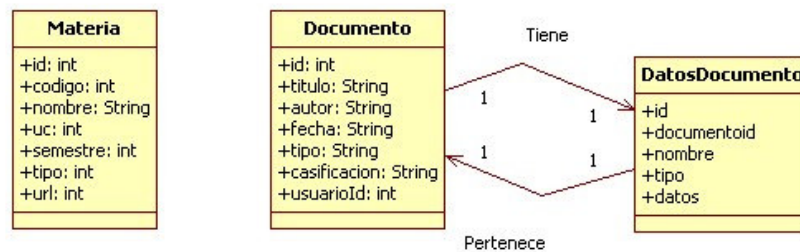
Figura 4. 35. Interfaz Vista Directorio

**Iteración 4:**

En esta iteración se desarrollaron los casos de uso referido a la información Académica de la Escuela y sus publicaciones, los requerimientos principales fueron:

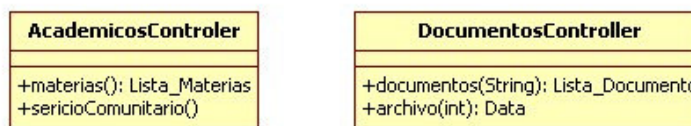
- En cuanto a las publicaciones se tiene como requisito clasificarlas por tipo para facilitar la ubicación por parte del usuario, luego se agregó una clasificación adicional para subcategorías dentro de un tipo.
- En cuanto a la opción de materias se deben mostrar por semestre y se colocar la información más relevante de la materia. El nombre de esta debe llevar a la página principal de la materia o en su defecto a la nota informativa.

**Diagrama de Clases:** En la figura 4.36 se muestran las clases persistentes asociadas a los elementos de los modelos que interactúan con las funcionalidades Ver Materias y Ver Documentos



**Figura 4. 36. Diagrama de Clases Persistentes Iteración 3**

Para esta iteración se crearon los controladores DocumentoController y AcademicosController, en la figura 4.37 se muestra el diagrama de clases de estos controladores.



**Figura 4. 37. Controladores asociados a las funcionalidades Ver Información Académica y Documentos**

A continuación se muestran los métodos del Controlador DocumentosController y la porción del menú de la opción Publicaciones

```

def documentos
  #Se lee el parámetro de entrada tipo
  tipo = params[:tipo]
  #Se hace la búsqueda de Documentos del tipo especificado, se clasifican y se ordenan de
  la más reciente a la más antigua.
  #Se coloca en la variable de instancia @documentos para que este disponible para la
  vista
  @documentos = Documento.find(:all,
:conditions=>["tipo=?",tipo],:order=>'clasificacion_documento_id, fecha DESC')
  #Se pasa el tipo a la vista, para indicar al usuario el tipo de documento que esta
  viendo
  @tipo = tipo
end

def archivo
  #Este metodo carga el documento para ser mostrado por pantalla o ser guardado por el
  usuario
  @archivo=DatosDocumento.find(:first,:conditions=>["documento_id=?",params[:id]])
  @nombre_archivo="#{@archivo.nombre.split('/')}"
  send_data(@archivo.datos,
:filename=> @nombre_archivo,
:type=>@archivo.tipo,
:disposition=>"attachment")
end

```

**Figura 4. 38. Código asociado al Controlador DocumentosController**

```

<li><a class="MenuBarItemSubmenu" href="#">Publicaciones</a>
  <ul>

    <li><%=link_to 'Normas',:controller=>'documentos',:action=>'documentos',
:tipo=>'Normativa'%>
    </li>

    <li><%=link_to 'Planillas y Actas', :controller=>'documentos',
:action=>'documentos', :tipo=>'Planilla'%>
    </li>

    <li><%=link_to 'Docencia',:controller=>'documentos', :action=>'documentos',
:tipo=>'Docencia'%>
    </li>

    <li><%=link_to 'Reportes técnicos', :controller=>'documentos', :action=>'documentos',
:tipo=>'Técnicos'%>
    </li>

  </ul>
</li><!-- Fin de Publicaciones -->

```

**Figura 4. 39. Código asociado a la opción del menú Publicaciones**

En el código anterior se puede observar que los elementos de la lista de la opción del menú correspondiente a Publicaciones, son enlaces dirigidos al mismo controlador y a la misma acción, diferenciándose entre ellos por el parámetro tipo.

## Principales Vistas o Interfaces

La **Vista de Materias** muestra un documento que contiene la programación vigente actual y el listado de materias con un formulario en la parte superior que permite buscar materias a partir del tipo, u ordenarlas por semestre u orden alfabético.

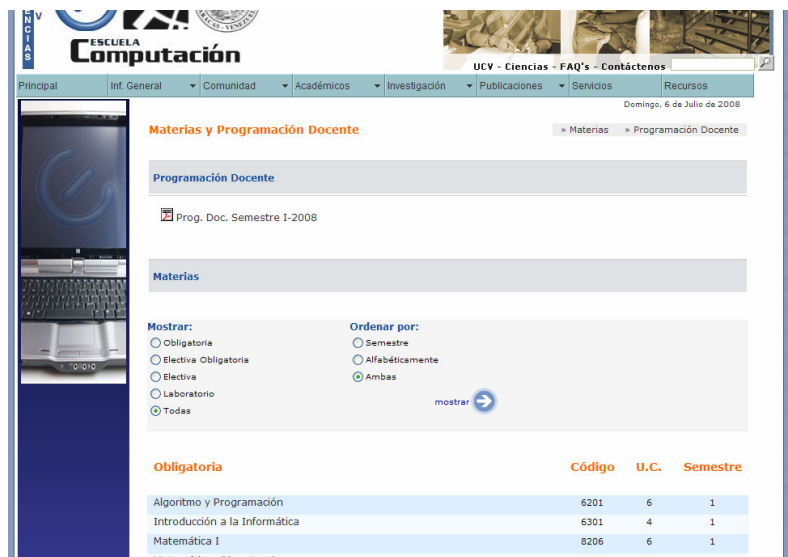


Figura 4. 40. Interfaz Vista Materias y Programación Docente

La vista *Documentos* muestra los documentos según la opción que se haya elegido en el menú

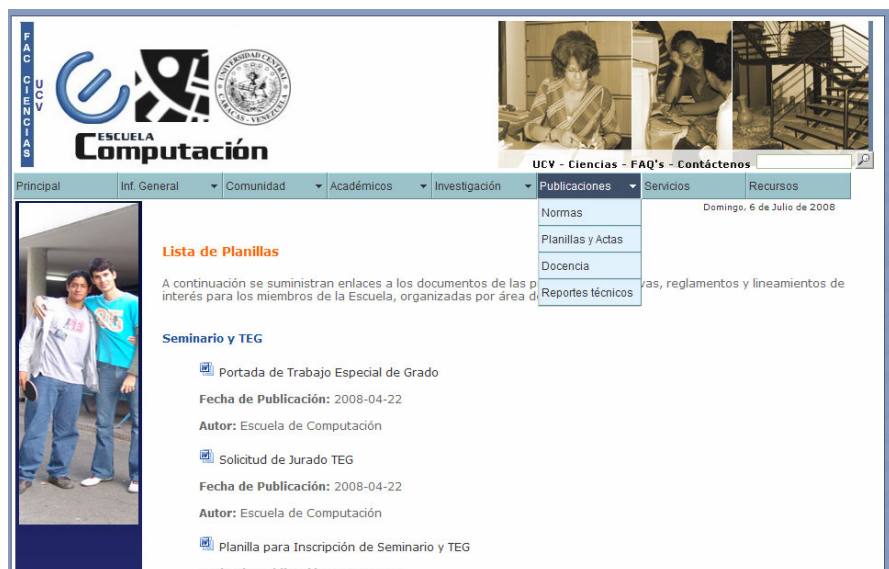


Figura 4. 41. Interfaz Vista Documentos

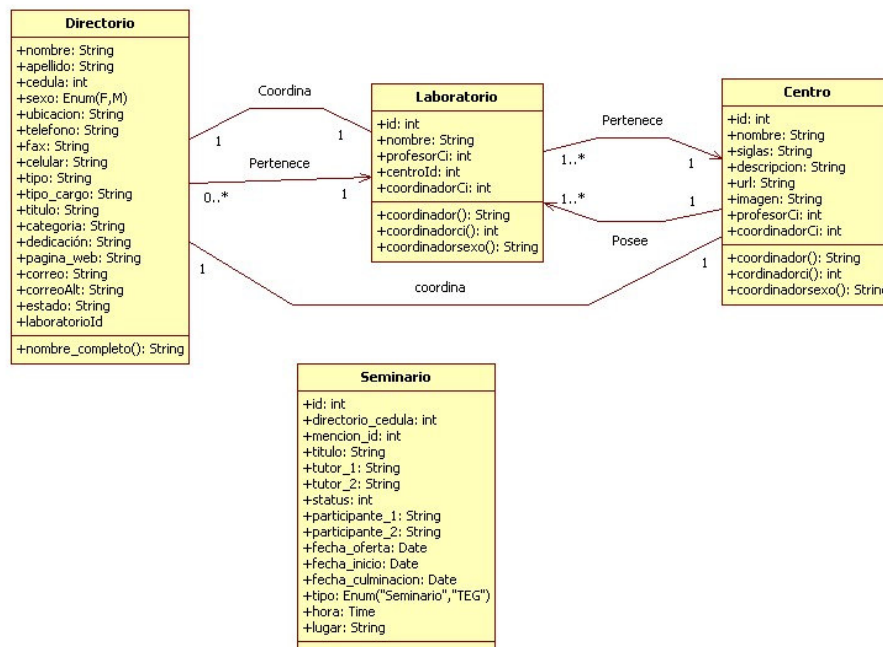
**Iteración 5:**

En esta iteración se desarrollaron los casos de uso referidos a las áreas de Investigación, Servicios y Recursos, para ello se colocaron 3 opciones en el menú, en la opción Investigación, se despliegan las funcionalidades asociadas a esta.



**Figura 4. 42. Submenú de la opción Investigación**

**Diagrama de Clases:** A continuación se muestran las clases persistentes correspondientes a los elementos del modelo en esta iteración, es decir las que manejan la información de los Centros, Laboratorios y Seminarios y Trabajos especiales de grado.



**Figura 4. 43. Diagrama de Clases Persistentes Iteración 4**



## Principales Interfaces

La vista *Centros* muestra un área destacada con el listado de los Centros de Investigación de la Escuela de Computación, para luego dar una breve descripción de cada uno, sus laboratorios asociados, coordinadores respectivos y un enlace a la página del centro.



Figura 4. 44. Interfaz Vista Centros de Investigación y Laboratorios

La vistas referidas a seminario muestra las funcionalidades asociadas a oferta de temas, seminarios en desarrollo, trabajos especiales de grado a presentar y culminados, por ejemplo la vista *seminario\_culminado* muestra los trabajos de investigación culminados desde el más reciente al más antiguo.



Figura 4. 45. Interfaz Vista T.E.G. Culminados



La vista *servicios* muestra una descripción estática de los servicios de la Escuela y la vista de *recursos* muestra los recursos con los que cuenta la escuela.

### 4.5.3.2 Subsistema Privado

#### Iteración 6:

En esta iteración se desarrollaron los casos de uso referido a autenticar, actualizar información personal y manejar contraseña. En el modulo de autenticación se identifica al tipo de usuario y se crearon filtros para determinar las funcionalidades asociadas a cada uno.

**Diagrama de Clases:** En la siguiente figura se puede observar el Diagrama de Clases Persistentes de esta iteración.

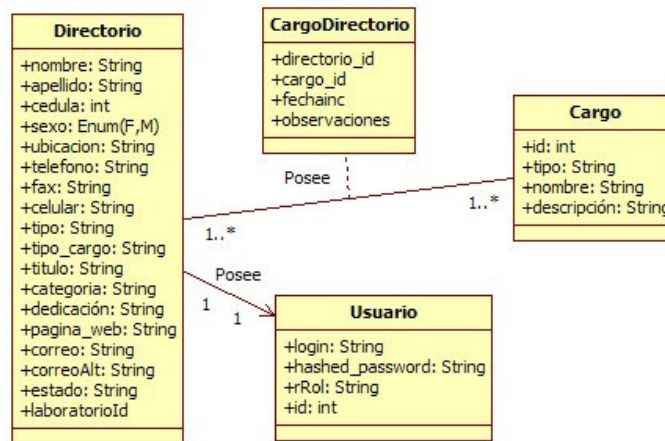


Figura 4. 46. Diagrama de Clases Persistentes Iteración 6

En esta iteración los datos del usuario y su personal asociado se almacenan en la sesión, además se desarrollo un filtro de entrada, el cual se encarga de verificar si un usuario está autenticado y si este es Administrador, esto con la finalidad de evitar que personas sin autorización ingresen a las páginas del subsistema.

#### Principales Interfaces

La vista *index* pide al usuario su Usuario y su Contraseña para poder acceder al sistema.



Figura 4. 47. Interfaz Página de autenticación Subsistema Administrativo

La vista *cambio\_password* permite al usuario cambiar su contraseña actual y la vista *directorio/edit* permite al usuario cambiar sus datos personales.



Figura 4. 48. Interfaces Cambio de Contraseña y Actualizar Datos Personales

### Iteración 7:

En esta iteración se desarrollaron los casos de uso referido a publicar documentos, definiendo detalladamente los requerimientos de este y las funcionalidades asociadas.

**Diagrama de Clases:** En la figura 4.49 se muestran las clases persistentes asociadas a esta iteración

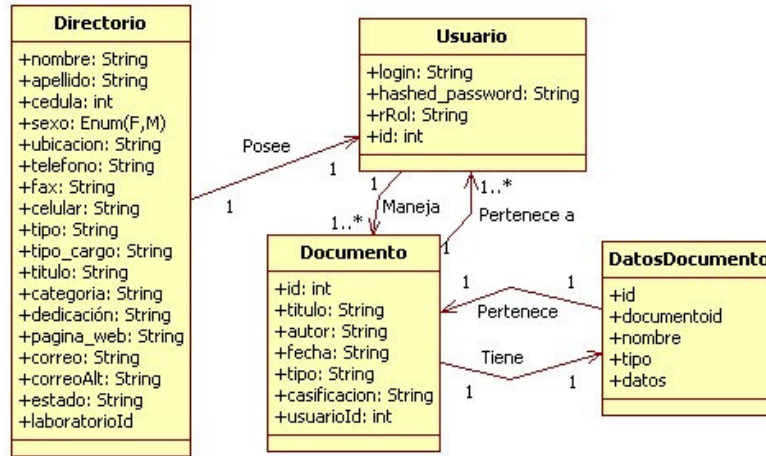


Figura 4. 49. Diagrama de Clases Persistentes Iteración 7

### Principales Interfaces

A continuación se muestra la vista list de documentos la cual muestra el listado de los documentos publicados por el usuario (el administrador los puede ver todos). A partir de esta, el usuario puede eliminar, modificar o ver los detalles de algún documento de la lista, o publicar un documento nuevo.



Figura 4. 50. Interfaz Listar Documentos

### Iteración 8:

En esta iteración se desarrollaron los casos de uso referido a publicar seminarios y trabajos especiales de grado, para lo cual se determinaron con más detalle los requerimientos asociados.

**Diagrama de Clases:** En la siguiente figura se muestran las clases persistentes asociadas a la iteración 8.

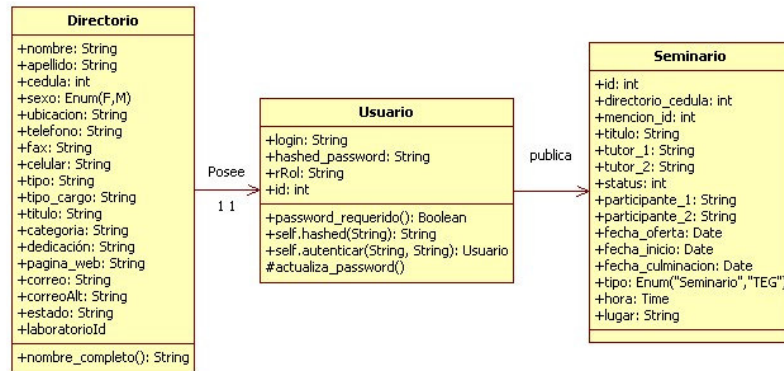


Figura 4. 51. Diagrama de Clases Persistentes Iteración 8

### Principales Interfaces

A continuación se muestra la vista edit de seminarios, la cual permite modificar los datos de un seminario ya publicado y seleccionado de la lista

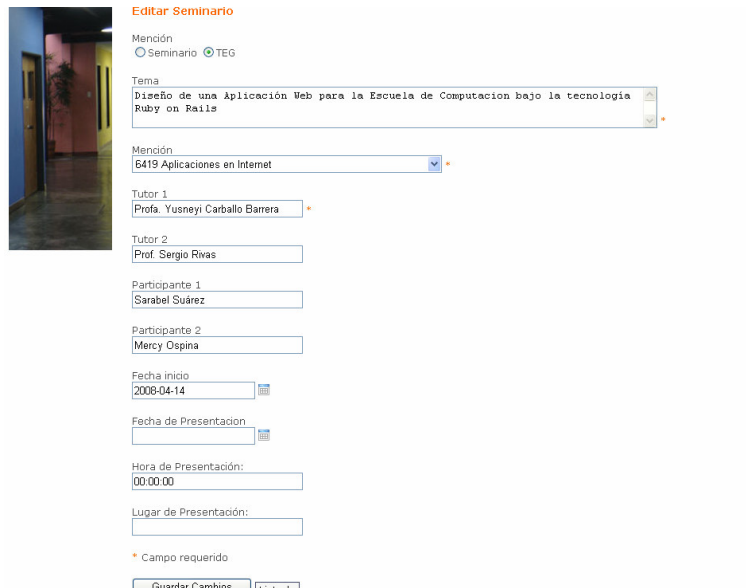


Figura 4. 52. Interfaz Editar seminario

### Iteración 9:

En esta iteración se desarrollaron los casos de uso referido a publicar ofertas de trabajos y pasantías, definiendo detalladamente los requerimientos de este y las funcionalidades asociadas.

**Diagrama de Clases:** En la siguiente figura se observan las clases persistentes asociadas a esta iteración donde PasantiaTrabajo es la clase que mantiene la información de las pasantías y los trabajos a publicar en la Escuela

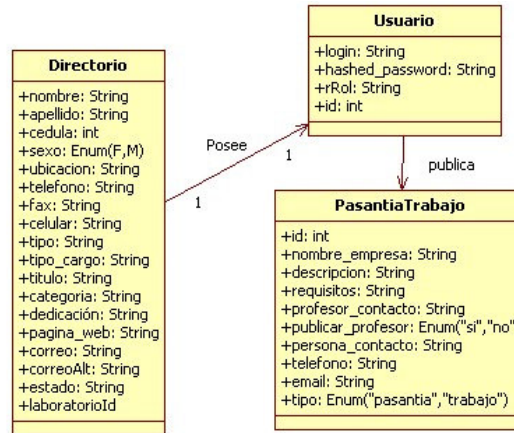


Figura 4. 53. Diagrama de Clases Persistentes Iteración 9

### Principales Interfaces

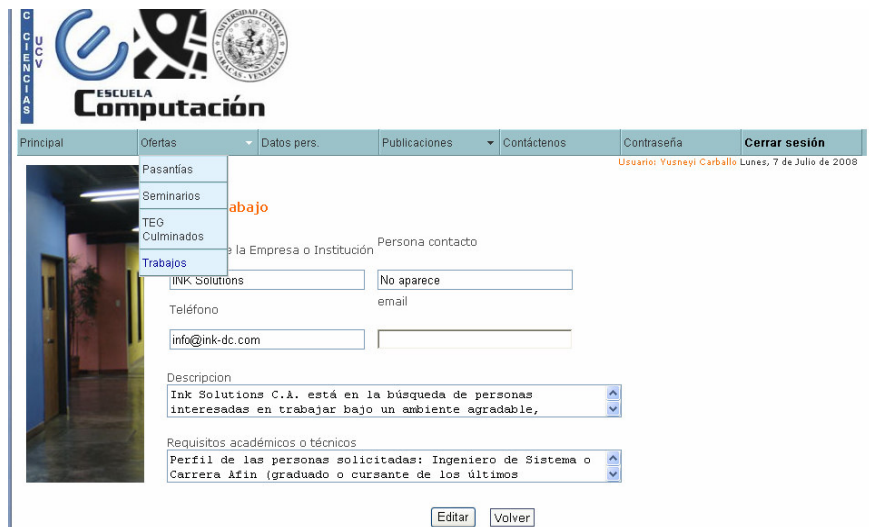
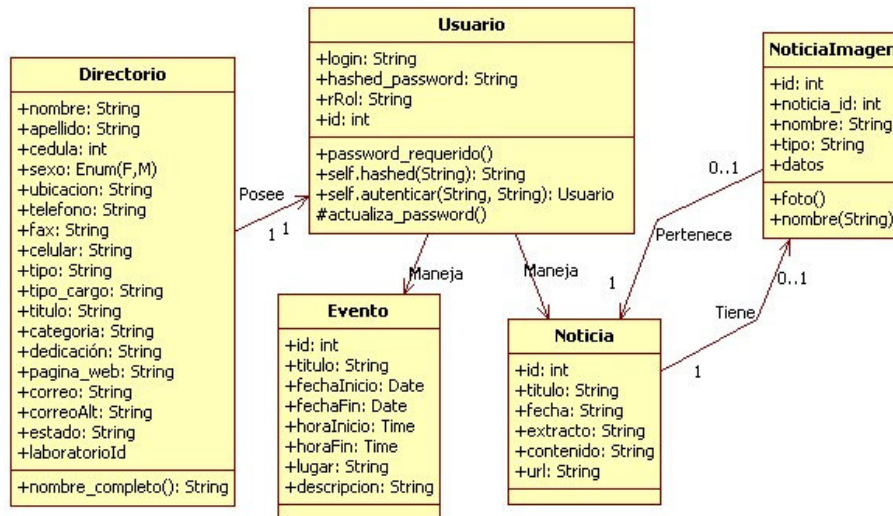


Figura 4. 54. Interfaz Editar Oferta de Trabajo

### Iteración 10:

En esta iteración se desarrollaron los casos de uso referido a publicar noticias y eventos.

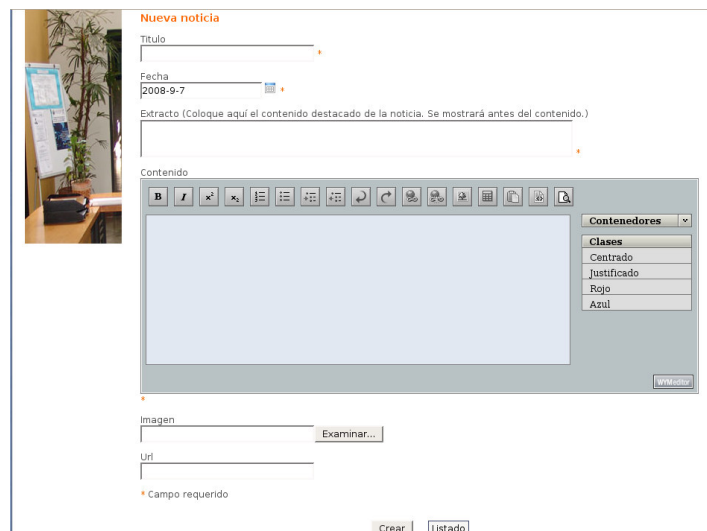
**Diagrama de Clases:** La figura 4.55 muestra las clases persistentes necesarias para la implementación de la iteración 10



**Figura 4. 55. Diagrama de Clases Persistentes Iteración 10**

### Principales Interfaces

La siguiente vista muestra la creación de una nueva noticia usando un editor de texto.



**Figura 4. 56. Interfaz Nueva Noticias**





#### 4.5.4 TRANSICIÓN:

##### **Ambiente de Producción**

En esta fase se colocó una versión beta de la aplicación en un servidor del centro de computación, el cual tiene las siguientes características.

Actualmente está ejecutándose en un computador que no tiene arquitectura de servidor debido a que está siendo usado para migrar el servidor Web actual, además a la máquina física se le han definido varias máquinas virtuales, y la aplicación Web de la escuela corre en una de estas máquinas virtuales. Las características de este computador son:

##### Máquina Física:

###### Hardware:

- 1 GB de RAM
- Disco Duro 160 GB
- Procesador Pentium D 3.2 GHZ

###### Software:

- Linux Debian Etch 4.0
- Monitor de máquinas virtuales XEN
- SSH server

##### Máquina Virtual donde se ejecuta la aplicación de la Escuela de Computación:

###### Hardware:

- 512 MB RAM
- Disco Duro 2 GB
- Procesador Pentium D 3.2 Ghz

###### Software (Máquina Virtual):

- Linux Debian Etch 4.0
- Java 5
- Ruby 1.8 y 1.9 (con las gemas Rails 1.2.6 y Mongrel)
- SSH server
- MySQL Server 5



- Apache 2
- PHP5
- Tomcat 5.

Dentro de la máquina virtual se creó una carpeta donde está alojado el sistema, y dentro de esta cada subsistema tiene un subdirectorío propio, por lo que cada uno debe implementar las clases descritas en la fase de construcción y ejecuta un hilo del servidor mongrel, lo cual permite que las peticiones a cada uno sean atendidas de manera independiente.

### Diagrama de despliegue

A continuaciones se observa el diagrama que muestra el despliegue de la aplicación

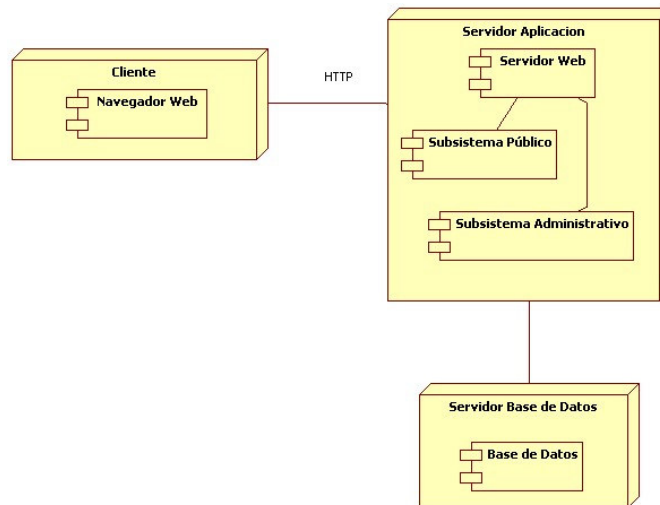


Figura 4. 58. Diagrama de Despliegue

### Encuesta de aceptación a usuarios finales.

En esta versión se colocó una encuesta en cada subsistema para evaluar aspectos de diseño, navegabilidad y contenido, que tiene por objetivo que el usuario evaluara aspectos puntuales y diera su opinión sobre estos, para medir el grado de aceptación de la aplicación. Una vez que se obtuvo un número considerable de encuestas llenas se procedió al análisis.

## Modelo de la Encuesta

Navegación
Diseño
Contenido

Nombre:  Tipo de Usuario: Estudiante ▼

---

**¿Qué opinas de la ubicación del menú de Navegación?**

Bien Ubicado  Mal ubicado

Podría estar mejor | Dónde:

**¿Piensas que se le deberían agregar o quitar opciones al menú?**

Dejar como está

Agregar  Quitar | ¿Cuáles opciones?

**Facilidad de uso del menú**

Abre con facilidad y se mantienen las opciones abiertas con suficiente tiempo

El tamaño de letra es apropiado para leer las opciones

Los submenús tienen solo las opciones requeridas

**Otros comentario sobre la organización del menú**

**¿Encontraste la información que buscabas?**

Sí, fácilmente  Sí, con dificultad  No conseguí lo que estaba buscando

**¿Qué información desearías que estuviera en el Sitio Web?:**

Navegación
Diseño
Contenido

**¿Qué opinas de las imágenes utilizadas en el Sitio?**

Son apropiadas

Son pocas se deben agregar más.

**Danos tus sugerencias sobre las imágenes**

**¿Qué opinas de los color usados?**

Adecuados  No adecuados

**Danos tus comentarios sobre los colores**

**¿Qué opinas del tamaño de letra? (verifica que tu navegador tenga la letra en tamaño normal)**

Pequeña  Adecuada  Grande

**Danos tus comentarios sobre la letra**

Navegación
Diseño
Contenido

**Calidad de la información encontrada**

Excelente  Buena  Regular  Mala

**Danos tus sugerencias sobre la información que encontraste en el Sitio**

**Alguna otra sugerencia sobre algún aspecto que consideres importante y no fue evaluado**

Figura 4. 59. Modelo de la Encuesta vía Web.

## Análisis de los datos

Se realizaron 54 encuestas del Sitio Público, con la siguiente distribución:

Publico General: 15

Estudiantes: 36

Administrativo: 1

Docente: 2

Estas fueron procesadas graficando los datos para una mejor comprensión.

A continuación se muestra el resultado de las respuestas a cada pregunta clasificadas por categoría y los comentarios de los usuarios por categoría.

➤ **Opciones de Navegación:**

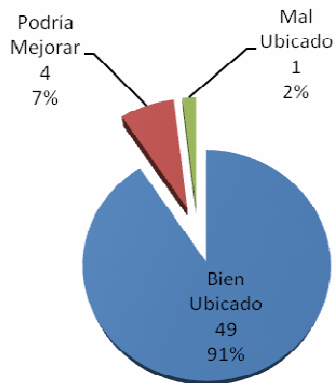


Figura 4. 60. Ubicación del Menú

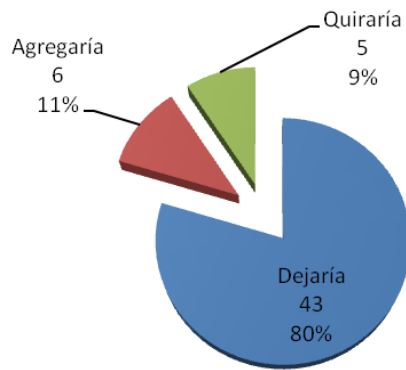


Figura 4. 61. Gráfico Cambiar Opciones

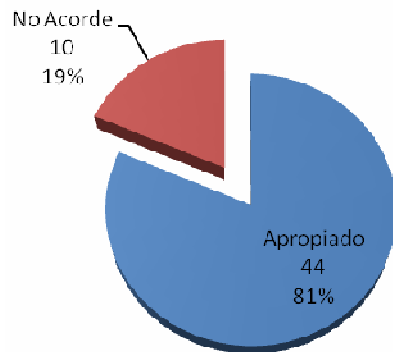


Figura 4. 63. Gráfico Facilidad de Acceso a las Opciones

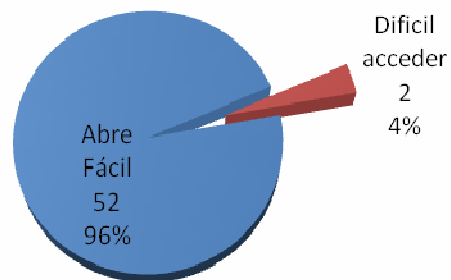


Figura 4. 62. Gráfico Tamaño de la Letra del Menú

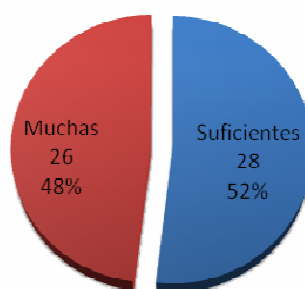


Figura 4. 64. Gráfico Cantidad de Opciones del Menú

**Comentarios de los usuarios sobre aspectos de navegación**

Sobre la organización del menú	Poner todo con solo 2 niveles.
	Reorganizar por tópico, es decir, siempre es bueno tener acceso a información como noticias, calendario y contáctenos de la mano (de fácil acceso), recomiendo jugar con el menú y reestablecer la información para mejor organización y búsqueda de información clara e intuitiva.
	Las secciones de servicios y recursos podrían ir dentro del menú de información general, para minimizar las opciones dentro del menú principal.
	En cuanto a la organización me parece que esta bien, tal vez tratar de que cada campo de los submenús mantengan la misma altura. Adicionalmente, podría quedar mejor si el menú queda en el tope de la página al momento de recorrer la misma.
	El color de letra (blanco) de los submenús que tienen más submenús no se puede leer.
	Un poco simples pero eficientes.
	Es una página muy atractiva, con la información necesaria, de verdad encontré todo lo que buscaba y hasta más.
Sobre las opciones del menú	Debería agregarse en el submenú comunidad, las actividades de extensión, asociadas a la escuela.
	Link de interés, con referencias a otras escuelas de computación.
	Servicios, creo que no tiene el contenido que uno esperaría ver en ese menú.
	Hay opciones que están repetidas en varias categorías.

Figura 4. 65. Comentarios Navegación

➤ **Opciones de Contenido:**

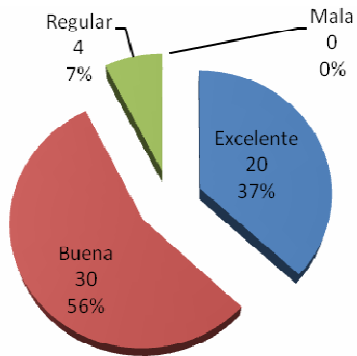


Figura 4. 67. Gráfico Calidad de la Información

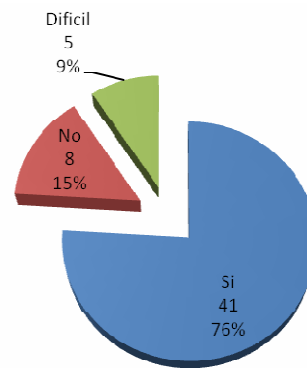


Figura 4. 66. Gráfico Ubicó la Información

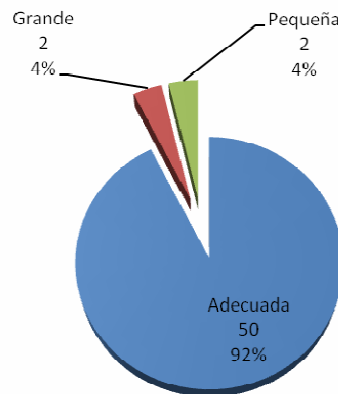


Figura 4. 68. Gráfico Tamaño de la Letra

**Comentarios de los usuarios sobre aspectos de contenido**

Los cursos que se ofrecen o links a esos departamentos.
Información acerca de las tesis presentadas en años anteriores.
Ayudas en línea para tesis, seminarios, etc., reglamento de la escuela de computación, enlaces de interés y relacionados, libros electrónicos, manuales en línea.
En general me parece bien. Quizás agregar un espacio para blogs de los estudiantes con opiniones, aunque pueden colocarlo como algo que envíen al correo y se publique en algún espacio.
Pudiese colocarse información de convenios con otras instituciones.
Tener los trabajos de grado aprobados y listado con sus autores me pareció interesante, sin embargo las

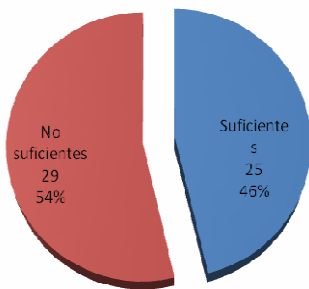
capacidades de búsqueda y navegación son limitadas, a pesar de que tiene paginación. Le faltan algunas cosas.

Me parece bastante completa. Quizás, en el calendario también se pudieran agregar los eventos más relevantes de postgrado, por ejemplo las fechas de inscripción. En la sección Centros de Investigación y Laboratorios, por cada centro aparece un cuadro con información que se ve desplazado hacia abajo tanto en Firefox 2.0 como IExplorer 7.

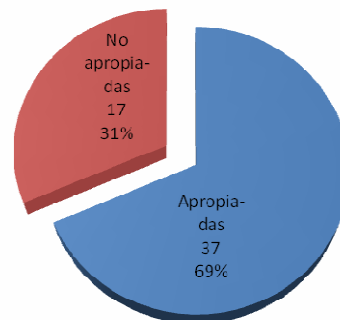
Está bien organizada y además es fácil de encontrar.

**Figura 4. 69. Comentarios Contenido**

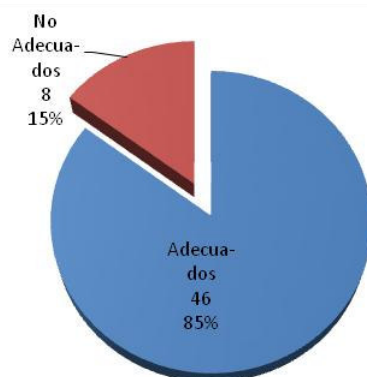
➤ **Opciones de Diseño:**



**Figura 4. 71. Gráfico Cantidad de Imágenes**



**Figura 4. 70. Gráfico Uso de Imágenes**



**Figura 4. 72. Gráfico Colores Usados**

**Comentarios de los usuarios sobre aspectos de diseño**

Sobre las imágenes	Repartir los logos de la escuela, facultad y universidad en lo ancho de la plantilla, creo que todos juntos recalcan mucho. Es más si es la Escuela de Computación podemos colocar la facultad y la universidad con otra iconografía y recalcar solo la de la escuela.
	Están bien pero deberían tener en la imágenes colores mas llamativos que motiven a los navegadores o al publico a ver el contenido que tiene en la pagina.
	Algunas fotos que luzcan con un poco mas de actualidad.
	Que salgan aleatoriamente te confunde, piensas q estas en diferentes secciones aun cuando te encuentre sección donde has estado antes. Debería ser consistente cada sección una respectiva imagen. Si son diferentes y aleatorias que tengan la misma línea (todas verdes con personas, o todas azules con equipos de computación) dependiendo la sección.
	Hay algunas imágenes que no les veo relación con el contenido de la página donde se encuentra.
	La imagen que tiene el nombre de la escuela de computación en vertical no se ve bien. Deberían colocar imágenes de la escuela mas que de su personal, también podrían utilizar más imágenes de tecnología así como la que tienen del <i>pendrive</i> .
	Creo que no se ve muy bien la distribución de los logos, todos juntos en la izquierda me parece que no luce muy bien... Quizá Poner la imagen de la derecha de fondo, clarita y distribuir los logos sobre esta. O quitar la imagen y distribuir los logos mejor.
Sobre los colores	Colores sobrios adecuados al sitio.
	La combinación del azul con naranja está bien, pues son colores complementarios, pero el menú verde no combina. Además que tiene otras tonalidades de azules. Por otra arte lo colores usados en las fotos del banner principal deberían llamar mas la atención y no ser tan opacos.
	Son agradables a la vista, no cansan los ojos.
	El verde de fondo no se ve muy bien. El gris en la página de materias no queda bien.
	Las fotos las colocaría a color.
	Me encantaron los colores, me gusta el azul y como lo combinan.
	De la observación de los gráficos y los comentarios anteriores se tomaron las siguientes resoluciones:
	Revisar el menú para eliminar las opciones repetidas, revisar las clasificaciones y reducir lo más posibles los subniveles a dos.

	Revisar los submenús para verificar y solucionar el problema de la letra en blanco.
	Eliminar las opciones que ofrecen poca información o son enlaces externos y agregarlas a páginas relacionadas o se dejarlas para una página de enlaces de interés.
	Crear una página para enlaces de interés.
	Crear un buscador para la página de trabajos especiales de grado culminados.
	Se cambiarán los colores de algunas áreas destacadas que están en gris a azul.
	En cuanto a las imágenes por limitaciones de tiempo no se cambiarán por ahora, pero se planificará su estudio para arreglarlas a corto o mediano plazo.

**Figura 4. 73. Comentarios Diseño**

➤ **Acciones tomadas**

De la observación de los gráficos y los comentarios anteriores se tomaron las siguientes resoluciones:

1. Revisar el menú para eliminar las opciones repetidas, revisar las clasificaciones y reducir lo más posibles los subniveles a dos.
2. Revisar los submenús para verificar y solucionar el problema de la letra en blanco.
3. Eliminar las opciones que ofrecen poca información o son enlaces externos y agregarlas a páginas relacionadas o se dejarlas para una página de enlaces de interés.
4. Crear una página para enlaces de interés y cargar los enlaces a otras Escuelas.
5. Crear un buscador para la página de trabajos especiales de grado culminados.
6. Se cambiarán los colores de algunas áreas destacadas que están en gris a azul.
7. En cuanto a las imágenes por limitaciones de tiempo no se cambiarán por ahora, pero se planificará su estudio para arreglarlas a corto o mediano plazo.

**Iteración 12:**

Esta iteración se realizó para introducir las mejoras producto del análisis de los resultados de las encuestas y tuvo una duración de tres semanas y fue realizada entre Julio y Agosto de 2008. Los cambios realizados son referidos a:

- Mejoras de organización y colores de letra en el menú
- Un buscador para los trabajos especiales de grado culminados,
- Una página para los enlaces de interés,



- Arreglos de tipos de letras, errores ortográficos o gramaticales y arreglo de algunos de los elementos existentes.
- Cambios de colores de las áreas destacadas.

### **Validación de Estándares.**

Se evaluó la validez de las páginas que conforman el subsistema público usando los validadores XHTML y CSS de la W3C obteniendo los siguientes resultados:

### **Validación XHTML**

Usando el validador XHTML de la W3C se evaluó cada una de las páginas que conforman este subsistema, y se obtuvo como resultado que 34 de las 36 páginas que lo conforman cumplen con el estándar, es decir un 94,4%.

Las páginas que no se lograron validar fueron las siguientes:

- La página principal por el uso del elemento html <marquee> utilizado para mostrar las noticias más recientes, debido a que no es un elemento válido en el estándar usado.
- La página de tesis culminadas, ya que la función autocompletar que provee Ruby on Rails y usada en el formulario de esta página, inserta estilos dentro del cuerpo, lo cual no es permitido por el estándar.

### **Validación CSS**

Usando el validador CCS de la W3C se evaluaron todos los estilos CSS usados en el subsistema público, los cuales cumplen con el estándar con excepción de los estilos div.contenedor y div.imagen vertical. Estos estilos tienen como finalidad definir el ancho y alto de la página mostrada ya que, para aprovechar los distintos tamaños de resolución se utilizó un diseño parcialmente líquido, es decir que se adapta a la resolución entre cierto rango de valores, esto se logra usando los atributos: min-width y max-width.

Sin embargo aun existen versiones del navegador Internet Explorer (anteriores a la 7) que no soportan estos atributos, por lo que se tuvieron que utilizar expresiones que funcionan en esos casos, pero que no son válidas.

### **Validación WAI**

Para este estándar se verificó que se cumplieran con las especificaciones técnicas que exige el estándar tales como:

- Que se use el atributo alt de las imágenes y el title de los enlaces para dar una descripción alternativa.
- Que se usen etiquetas label con los campos de un formulario.
- Asegurarse de que entre el color de fondo y el de primer plano haya suficiente contraste para que puedan ser diferenciado por personas con problemas visuales.

Se usó el validador TAW y se determinó que se cumple con el nivel de aceptación A, lo cual se considera una base para lograr una aplicación accesible.

Gracias al uso de los estándares Web la aplicación es operativa en los navegadores Internet Explorer v7.x, Mozilla Firefox v2.x y v3, Apple Safari v3.x, Opera v9.51 y Netscape v9.0, los cuales cumplen dichos estándares. En versiones anteriores del navegador Internet Explorer se tuvieron que utilizar recursos adicionales, específicamente en el uso de CSS, versiones anteriores del navegador Opera presentan algunos problemas al cargar los estilos del menú.

## **CONCLUSIONES**

Al concluir la realización de este Trabajo Especial de Grado, se presenta la construcción de una aplicación Web, denominada Sitio Web de la Escuela de Computación, conformada por dos subsistemas que se complementan, uno de ellos muestra la información al usuario y se presenta como la imagen de la Escuela de Computación en la Web, el otro permite mantener esta información actualizada.

La implementación del subsistema Público presentó cierto nivel de dificultad al momento de realizar el diseño de las interfaces debido a la exigencia por parte del equipo de desarrollo de lograr una interfaz de calidad, altamente usable y que cumpliera con los principales estándares Web. Para poder cumplir con este objetivo se utilizaron distintos recursos, como entrevistas a docentes del área de usabilidad durante la etapa de Elaboración, encuestas vía Web en la etapa de Transición para ser llenadas por los visitantes del sitio Web, y que nos permitirían tener algunos parámetros de medición y análisis; y evaluación de las páginas por medio del uso de los validadores de XHTML y CSS de la W3C. De los resultados obtenidos del análisis de las encuestas se introdujeron algunas mejoras a la aplicación. Como consecuencia del cumplimiento de este objetivo se logró que la aplicación se pueda ejecutar correctamente en los distintos navegadores existentes en el mercado.

El proceso de desarrollo UP Ágil al seguir la filosofía del Modelado Ágil y ser fácilmente adaptable a las necesidades del proyecto fue clave para el éxito del mismo, gracias a la aplicación de prácticas de proceso en diseño, codificación y pruebas, que soportaron el desarrollo rápido de una Aplicación expuesta a cambios durante sus diferentes etapas, y promoviendo la comunicación y trabajo en equipo de los programadores comprometidos en el cumplimiento de las labores en cada iteración.

El uso de Ruby on Rails como tecnología del lado del servidor agilizó considerablemente el proceso de desarrollo, ya que se aprovecharon las diversas estructuras y métodos predefinidos que tiene, de manera que si se desarrollaran con otros lenguajes requerirían de un mayor esfuerzo y tiempo de codificación.

El manejo de versiones a través de Subversion permitió desempeñar un mejor trabajo en equipo, considerando que el código desarrollado era compartido y modificado por los integrantes del proyecto, incluyendo el cliente, quienes realizaban actualizaciones sobre una misma versión del sistema, detectando así si se afectaba el funcionamiento de piezas de código ya consolidadas.

En general la implementación de esta aplicación Web genera una serie de ventajas importantes para la Escuela de Computación, entre las que se encuentran:

- Permite publicar de manera rápida y fácil noticias y eventos de la Escuela, que luego podrán ser vistos por toda la comunidad.
- Permite publicar y consultar ofertas de pasantías, trabajos y seminarios.
- Cuenta con un directorio completo del personal que labora en la Escuela, el cual puede ser actualizado por el mismo personal.
- Muestra las investigaciones y la información de los centros y laboratorios que las llevan a cabo.
- Muestra información institucional que permite a la Escuela de Computación darse a conocer, mostrando su historia, autoridades y sus principales actividades y servicios.
- Utiliza tecnologías de código abierto, lo cual sigue las políticas de uso de software libre de la Universidad Central de Venezuela.

Debido a lo afirmado en los párrafos anteriores podemos concluir que se cumplieron los objetivos planteados al comienzo de este documento, y atribuimos en gran parte el logro de estos a la planificación y organización de requerimientos y actividades necesarias para la creación del sistema.

Sin embargo debido a limitaciones de tiempo y de información algunas funcionalidades no pudieron ser desarrolladas en su totalidad como aquellas referidas a Recursos y Servicios ofrecidos por la Escuela.

### ***RECOMENDACIONES***

Se recomienda para futuros trabajos ampliar la información referente a los Servicios prestados por la Escuela y los Recursos con los que cuenta, ya que se considera de gran utilidad para la comunidad, de tal forma de mejorar y ampliar el sistema.

Actualmente la información de los graduandos en la Licenciatura de Computación, por año debe obtenerse del sistema de control de estudios y cargarse directamente a la base de datos, de manera manual, por lo que se hace necesario una interfaz que integre ambos sistemas para que de esta manera los datos sobre los graduandos de Computación pueda tenerse actualizados y disponibles desde el Sitio de la Escuela.

El sitio requiere de un Administrador o grupo de Administradores que se encargue de manejar aspectos de información delicada en el sistema como el cambio de autoridades, creación de usuarios, centros, laboratorios y materias, que no pueden ser llevados a cabo por usuarios de otro nivel, por lo que se recomienda la creación de una comisión para la Administración del Sitio Web y de esta forma garantizar su oportuna y constante actualización.

También se recomienda adaptar la aplicación a dispositivos móviles, usando para ello las herramientas provistas por las hojas de estilo para otros dispositivos, aprovechando el hecho de que se usó el estándar CSS lo cual hace esta tarea más sencilla de implementar.

### ***APORTES***

Esperamos que la presente investigación sea un aporte significativo a la Escuela de Computación ya que desarrolla e implementa un sitio Web para la Escuela, entre cuyos objetivos están tener una interfaz usable y contenidos de calidad, que maneje sus principales procesos, sirva de punto de contacto e información a sus usuarios, y que puede ser extendido fácilmente agregando nuevas funcionalidades. Es importante destacar el hecho de que esta aplicación cuenta con funcionalidades que permiten la actualización de la información mostrada, por parte del mismo personal de la Escuela, lo cual podrá garantizar la actualización constante del sitio Web.

Por otro lado esta contribución puede servir de guía a aquellos investigadores que deseen o requieran trabajar con la metodología de desarrollo ágil UP, y con los estándares de desarrollo Web, entre los que se destacan los estándares XHTML, CSS y WAI.

## REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES

- [1] Informaticamilenium.com.mx (2008) Páginas Web, Internet, Sitio Web, Portal, Hospedaje, Definición. Tomado de la página Web <http://www.informaticamilenium.com.mx/Paginas/espanol/sitioWeb.htm>.
- [2] Departamento de Control de Calidad y Auditoría Informática (Junio 2001). Sistemas en Arquitectura Cliente/Servidor. Tomado de la página Web <http://sistemas.dgsca.unam.mx/publica/pdf/clienteservidor.PDF>
- [3] Cliente/Servidor. Consultado de la página Web: <http://www.csi.map.es/csi/silice/Global71.html>
- [4] Masadelante.com (2006) Tipos de Servidores. Tomado de la página Web <http://www.masadelante.com/faq-tipos-de-servidores.htm>.
- [5] Wikipedia (Junio de 2008), Programación por capas, Tomado de la página Web [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_tres\\_niveles](http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles)
- [6] Sabja Russ Cristian, Soto Lagunas Alejandro y Antillanca Espina Héctor (Sin Fecha). Método para transformar el diseño de una aplicación basada en la arquitectura MVC++ a una aplicación Web, Departamento de Ingeniería Informática, Universidad de Santiago de Chile, 917-0124 [Publicación en línea]. Tomado de la siguiente dirección Web: <http://www.diInformaciónusach.cl/ArchivosSubidos%5C212007184919Articulo%20MVC++%20a%20WEB%20v7.pdf>
- [7] Adelat.org (2006). Lenguajes del lado del cliente o del servidor. Consultado de la página Web: [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html)
- [8] W3c.es (2008). Guía breve de XHTML. Consultado de la página Web: <http://www.w3c.es/divulgacion/guiasbreves/XHTML>
- [9] Sidar.org. XHTML Básico. Consultado de la página Web: <http://www.sidar.org/recur/desdi/traduc/es/xhtml/xhtml-basic.html>
- [10] Pozo R. Juan (2001). Traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification". Tomado de la WWW: <http://html.conclase.net/w3c/html401-es/present/styles.html#h-14.1>
- [11] Rivas Santos, Víctor. Curso de JavaScript. Tomado de la WWW: [http://geneura.ugr.es/~victor/cursillos/JavaScript/js\\_intro.html](http://geneura.ugr.es/~victor/cursillos/JavaScript/js_intro.html)
- [12] Sobrerailes.com. SobreRaíles. En marcha con Ruby on Rails. Tomado de la página Web [http://sobrerailes.com/pages/en\\_marcha\\_con\\_rails](http://sobrerailes.com/pages/en_marcha_con_rails)

- [13] Carlson, Lucas y Richardson Leonard (2007) Curso de Ruby, Procedimientos para la creación de secuencias de comandos orientadas a objetos. Madrid - España. Ediciones Anaya Multimedia.
- [14] Es.wikipedia.org (2008). Ruby on Rails. Tomado de la página Web [http://es.wikipedia.org/wiki/Action\\_Mailer](http://es.wikipedia.org/wiki/Action_Mailer)
- [15] Wiki.rubyonrails.com (2007). WEBrick in Ruby on Rails. Tomado de la página Web <http://wiki.rubyonrails.com/rails/pages/WEBrick>
- [16] Tate, Bruce A. y Hibbs, Curt (2007). Ruby on Rails. Madrid - España. Ediciones Anaya Multimedia
- [17] MySQL (2008). Manual de Referencia MySQL 5.0 (Español), tomado de la página Web <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- [18] MySQL (2008), Principales características de MySQL, tomado de la página Web <http://dev.mysql.com/doc/refman/5.0/es/features.html>
- [19] PosgreSQL Vs. MySQL, Daniel Pecos, Ventajas de MySQL, tomado de la página Web [http://www.netpecos.org/docs/mysql\\_postgres/x108.html](http://www.netpecos.org/docs/mysql_postgres/x108.html)
- [20] Control de Versiones con Subversión, Ben Collins-Sussman, (2004), tomado de la página Web <http://svnbook.red-bean.com/nightly/es/index.html>
- [21] Apache2.com (2008). The Apache Software Foundation - Apache httpd server - Web Server. Tomado de la página Web <http://www.apache2.com>
- [22] Apache Best Practice Deployment (2008), Tomado de la página Web <http://mongrel.rubyforge.org/wiki/Apache>
- [23] Ambler, Scott W. (2006). Agile Modeling (AM) Home Page, Effective Practices for Modeling and Documentation, Tomado de la página Web <http://www.agilemodeling.com/>
- [24] Scott W. Ambler. Agile Modeling: Effective Practices for EXtreme Programming and the Unified Process. J. Wiley, 2002.
- [25] Scott W. Ambler (Junio 2006). The Agile Unified Process (AUP), Tomado de la página Web <http://www.ambysoft.com/unifiedprocess/agileUP.html>
- [26] García Gómez, Juan Carlos. (2005). Úsalo, usabilidad para todos: Escribir para la red. Tomado de la página Web: <http://usalo.es/77/escribir-para-la-red/> .
- [27] W3C (2008). Guía breve sobre estándares Web. Tomado de la página Web: <http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>
- [28] W3C (2008). ¿Qué es el Consorcio World Wide Web (W3C)?. Tomado de la página Web: <http://www.w3c.es/Consorcio/>
- [29] WAI (Septiembre del 2005) Introducción a la Accesibilidad Web. Tomado de la página Web <http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>



- [30] Biblioteca de la Universidad Europea de Madrid. ¿Cómo saber si mi página Web es accesible?, Tomado de la página Web [http://biblioteca.upc.es/Rebiun/nova/InformesGrupoTrabajo/revisiones\\_automaticas.htm](http://biblioteca.upc.es/Rebiun/nova/InformesGrupoTrabajo/revisiones_automaticas.htm)
- [31] W3C (2008). Guía breve de Internacionalización. Tomado de la página Web <http://www.w3c.es/Divulgacion/GuiasBreves/internacionalizacion>
- [32] W3C (2008). Guía breve de Web Móvil. Tomado de la página Web <http://www.w3c.es/Divulgacion/GuiasBreves/webmovil>