



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA
Y ADMINISTRACIÓN

REIMPLEMENTACIÓN DEL SITIO DE VIDEOTECA DE LA FCA

DISEÑO DE UN SISTEMA O PROYECTO PARA UNA
ORGANIZACIÓN

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTAN:

RICARDO DEL ROSARIO CELAYA
AZUCENA PÉREZ ARIAS
HUGO VLADIMIR RESENDIZ BEIZA
FRANCISCO ALONSO VALDÉS ESCAMILLA
BRENDA SARAI ZARATE RIVERA

ASESOR:
DRA. BRIBIESCA CORREA GRACIELA



MÉXICO, D.F.

2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIAS Y
AGRADECIMIENTOS

Me place la idea de dedicar este trabajo:

A Dios por ser siempre mi compañero a lo largo de todo este tiempo y que de muchas maneras tuve la dicha de conocerlo.

A mis padres que me dieron el mejor regalo que se pueda recibir, la vida y que gracias a su amor, confianza y dedicación supieron encaminarme por el rumbo en el que voy.

A mis hermanas Lourdes y Alicia, por tanto cariño y apoyo que me han proporcionado, a mis hermanos que a lo largo del tiempo he encontrado y que de alguna manera han enriquecido mi vida, Bertín, Braulio, Guillermo, Alfredo y José Alfredo.

A mis tíos Javier y Victoria, por ser unas excelentes personas y por brindarme ese plus fuera de mi casa.

A mis amigos y amigas que en todo momento compartimos cosas inolvidables y para mi fortuna tengo la gracia de seguir contando con ellos: Izcoatl, Francisco, Hugo, Andrés, Juan Ramón, Araceli, Sandra, Julia, Azucena, Brenda, Tania Berenice, Saraí, Verónica Gabriela.

A todos los que mi memoria no pudo enfocar en este momento...

Gracias de corazón.

Ricardo Del Rosario Celaya

A Dios por haberme permitido llegar hasta este momento dándome salud, paciencia valentía, coraje, fuerza, felicidad y amor para lograr mis objetivos y porque sé que nunca me dejara sola.

A mis padres Roberto y Julia que son el mayor tesoro que Dios me dio, gracias por su amor y apoyo incondicional en todas las decisiones que he tomado, por levantarme cada vez que caía, por ayudarme a ser responsable y mejor persona, por su gran corazón y capacidad de entrega a nuestra familia.

A mi Abuelita Catalina por su amor y darme esas palabras de aliento para seguir luchando en todo momento.

A mis hermanos Sandra, Sergio y Andrea por su cariño, apoyo, paciencia y ayuda que me brindan a diario.

A mi Tía Reyna y primo Alberto por brindarme su tiempo en todas aquellas noches que se convertían en días, estudiando o platicando infinidad de cosas, gracias por tus consejos, orientaciones y enseñanzas. “No importa las veces que caigas y toques fondo, sino las veces que te vas a levantar para luchar por tus sueños”, porque todo es posible, sólo es cuestión de tiempo, paciencia y dedicación.

A mis sobrinos Martin, Jessica, Isaac y Ángel. Inteligentes y tremendos a la vez, dándole la chispa de alegría a la casa y teniendo siempre algo nuevo para sorprender.

A todos mis amigos que estuvieron conmigo a lo largo de la carrera, gracias por esos momentos de alegría, tristeza, paciencia, comprensión, confianza, apoyo y ayuda, brindándome su amistad, valor importante en mi vida.

A mis amigos de toda la vida, porque ni el tiempo ni la distancia han logrado romper los lazos de amistad que siempre nos han unido y porque sé que no se olvidan de mí.

A todas las personas y maestros que me he encontrado a lo largo del camino y que me brindaron su conocimiento y apoyo.

A mi equipo de trabajo para este proyecto Francisco, Hugo, Ricardo y Sarai

A ti amargoso por soportarme tanto, mil gracias por todo y disculpa mis locuras.

Gracias a todos ustedes he llegado a esta meta, los quiero mucho.

Azucena Pérez Arias

A mis padres Micaela y Víctor por su infinito apoyo y compartir su sabiduría de la vida.

A mis hermanos Violeta y Víctor por su cariño y comprensión.

A Uriel y Argenis por entender todo lo que soy.

A Izcoatl, Andrés, Juan Ramón y todos los que compartieron el gran viaje de varios años en las aulas de la facultad y fuera de ella.

A Sarai, Azucena, Francisco y Ricardo por compartir y no detenerse en este sueño que nos llevó a la puerta de nuestro destino.

A quien lo sabe.

Hugo Vladimir Resendiz Beiza

Principalmente quiero dedicar este trabajo de titulación a Dios y a mis padres por darme la vida, soporte y cariño además de instruirme y tener paciencia conmigo.

A mis hermanos: Luz María, Andrés y Dulce y a mi familia que me apoyo en todo el trayecto de mi vida escolar y personal.

A todos mis amigos y amigas, que comparto este trabajo y carrera como: Azucena, Hugo Vladimir, Ricardo, y Brenda Saraí. Sin olvidar a los demás que están todo el tiempo pendientes de mí y mis acciones, ya que cada uno de ellos los aprecio y estimo.

A todas las personas, maestros, compañeros y amigos que hemos coincidido en algún período de nuestras vidas y que por cualquier circunstancia o casualidad han compartido muchos momentos conmigo, ya que de todos ellos han surgido todo tipo de experiencias con las cuales se ha forjado la persona que soy.

Muchas Gracias.

Francisco Alonso Valdés Escamilla

Aún recuerdo el momento en el que tenía tu edad y me preguntaron ¿Qué quieres estudiar de grande?, a lo cual respondí “Quiero estudiar informática” y heme aquí a pocos pasos de convertirme en Lic. Informática.

Durante todo ese tiempo pasé por varias experiencias dentro de las cuales tuve aciertos y errores, pero fueron éstos últimos los que me ayudaron a crecer como persona. Tuve la dicha de conocer a diversas personas que con sus ideologías me hicieron ver la vida desde otra perspectiva y claro con todos esos aprendizajes forme mi propia ideología de vida.

En ocasiones me vi frustrada, llegué a pensar que la decisión que tomaban no era la correcta, tuve demasiados cuestionamientos para mi persona, pero aún con todo ello luché y no me dejé vencer; encontré las soluciones.

Con esto quiero que sepa esa persona que la vida le tiene planeada muchos retos, algunos fáciles algunos otros complicados, que en ocasiones creerá que no puede más, pero también, quiero que sepa que esos retos son los que le hacen ser una gran persona, le hacen crecer, que jamás deje de luchar ni de ver más allá de lo que todos pueden ver, que recuerde que al final del puente le espera mucho más.

Quiero que un día llegue a donde estoy y me diga “Recuerdo cuando tenías mi edad”. Todo el esfuerzo de no sólo cuatro años y medio, sino de veinte años, va dedicado a esa persona, a ti.

Daniel Rojas Rivera

Agradecimientos.

Conforme pasa el tiempo uno mismo se va encontrando con distintas personas, algunas de ellas simplemente pasan un instante pero ese instante se convierte en un aprendizaje de toda la vida.

Agradezco a todas esas personas especiales que han pasado por mi vida, empezando por mi madre Angélica María Rivera Solórzano que ha sido la primera persona que vi y la que siempre seguiré a mi lado.

A mi familia que siempre ha confiado en mí.

A Emiliano Castillo Jara, Ricardo Salinas Pérez, Hugo Israel Rodríguez Ortiz y a Hugo Vladimir Reséndiz Beiza, quienes me han dado una gran enseñanza de vida.

A Yair García Torres, y Jorge Alejandro Millán, que aunque apenas, han sido un gran apoyo para el término de este trabajo.

Por último a todos los que me consideran una amiga.

Brenda Saraí Zarate Rivera

Agradecimientos

Para el presente trabajo de titulación queremos hacer énfasis en la gratitud que tenemos hacia las personas que nos apoyaron en la elaboración de este trabajo, ya que sin su colaboración se habría dificultado el desarrollo del nuevo sistema de videoteca y todo el proceso que este conlleva. Así mismo agradecemos a todas las personas que de alguna manera estuvieron pendientes dando sus comentarios y de nuestros avances en todo el tiempo que requerimos para su asesoría y atención prestada en este proyecto y durante toda la carrera, es por eso que por este medio agradecemos a las siguientes personas:

Dra. Graciela Bribiesca Correa

Mtro. Ismael Israel Perea Camarillo

Ing. Daniel Manrique Martínez

También a:

Mtro. Jaime Ayala Pérez

Ing. Luis Arenas Hernández

Y sobre todo a la institución que nos dio la oportunidad de estudiar la profesión a la cual nos dedicaremos en nuestras vidas. ¡Gracias! Universidad Nacional Autónoma de México y a nuestra Facultad de Contaduría y Administración.

Índice

Dedicatorias y Agradecimientos	ii
Lista de tablas	x
Lista de figuras	x
Introducción	1
Capítulo 1. Justificación de la metodología utilizada en este proyecto de titulación.....	5
Metodologías contempladas para el desarrollo del sistema	6
Pesadas o tradicionales	6
Rational Method Composer (RMC)	6
Objectory.....	8
Agiles ó Ligeras	9
XP	9
SCRUM	9
Híbridas.....	10
ICONIX	10
Justificación de la metodología elegida.....	17
Capítulo 2. Análisis	18
Lista de requerimientos web	19
Lista de requerimientos del área de gestión.....	20
Diagrama de clases	22
Modelo de dominio	24
Prototipo	25
Casos de uso.....	29
Diagrama general de sistema videoteca.....	29
Ingresar videoteca.....	30
Administrar video.....	32
Administrar catálogos.....	37
Buscar video administración.....	49
Buscar video web	52
Capítulo 3. Diseño.....	54
Modelo E-R.....	55
Relación de atributos	56
Diagrama de robustez.....	59
Ingresar videoteca.....	59
Administrar video.....	60
Administrar catálogos.....	63
Buscar video administración.....	75
Buscar video web	76
Diagramas de secuencia.....	77
Ingresar videoteca.....	77
Administrar video.....	78
Administrar catálogos.....	81
Buscar video administración.....	93
Buscar video web	94

Diagramas de actividades	95
Ingresar videoteca.....	95
Administrar video.....	96
Administrar catálogos.....	99
Búsqueda administración.....	111
Búsqueda web	112
Capítulo 4. Implementación.....	113
Descripción y uso de <i>Ruby on Rails</i> en el proyecto.....	114
Patrón MVC.....	115
Modelo	116
Vista.....	116
Controlador	117
Migraciones.....	118
Pruebas.....	119
Prueba de complejidad ciclomática	120
Prueba de sistema	124
Conclusiones	127
Glosario	128
Siglarío.....	131
Bibliografía	133
Mesografía	134
Anexo	137

LISTA DE TABLAS

Tabla 1. Requerimientos web.....	20
Tabla 2. Requerimientos del área de gestión.....	21
Tabla 3. Requerimientos específicos para los casos de uso.....	28
Tabla 4. Atributos para la tabla videos de la base de datos.....	56
Tabla 5. Atributos para la tabla users de la base de datos.....	57
Tabla 6. Atributos para la tabla autors de la base de datos.....	57
Tabla 7. Atributos para la tabla perfils de la base de datos	57
Tabla 8. Atributos para la tabla categories de la base de datos.....	58
Tabla 9. Atributos para la tabla bitácoras de la base de datos	58
Tabla 10. Matriz de resultados del grafo.....	124

LISTA DE FIGURAS

Figura 1. (a).....	22
Figura 2. (b) Diagrama de clases, donde (a) y (b) son complementos y solo existe una clase bitácora.....	23
Figura 3. Modelo de dominio refinado donde se representan las clases con sus atributos, operaciones y cardinalidad.....	24
Figura 4. Prototipo del sistema videoteca sección video.....	25
Figura 5. Prototipo del sistema videoteca sección información.....	26
Figura 6. Diagrama general del sistema.....	29
Figura 7. Diagrama ingresar al sistema.....	30

Figura 8. Diagrama administrar video.....	32
Figura 9. Diagrama administrar catálogos.....	37
Figura 10. Diagrama administrar usuario.....	38
Figura 11. Diagrama catalogo autor.....	39
Figura 12. Diagrama crear perfil.....	39
Figura 13. Diagrama catalogo categoría.....	40
Figura 14. Búsqueda video administración.....	49
Figura 15. Diagrama buscar video web.....	52
Figura 16. Diagrama modelo Entidad-Relación.....	55
Figura 17. Diagrama de robustez ingresar videoteca.....	59
Figura 18. Diagrama de robustez crear video.....	60
Figura 19. Diagrama de robustez actualizar video.....	61
Figura 20. Diagrama de robustez borrar video.....	62
Figura 21. Diagrama de robustez crear usuario.....	63
Figura 22. Diagrama de robustez modificar usuario.....	64
Figura 23. Diagrama de robustez borrar usuario.....	65
Figura 24. Diagrama de robustez crear autor.....	66
Figura 25. Diagrama de robustez modificar autor.....	67
Figura 26. Diagrama de robustez borrar autor.....	68
Figura 27. Diagrama de robustez crear perfil.....	69
Figura 28. Diagrama de robustez modificar perfil.....	70
Figura 29. Diagrama de robustez borrar perfil.....	71
Figura 30. Diagrama de robustez crear categoría.....	72
Figura 31. Diagrama de robustez modificar categoría.....	73
Figura 32. Diagrama de robustez borrar categoría.....	74
Figura 33. Diagrama de robustez buscar video administración.....	75
Figura 34. Diagrama de robustez buscar video web.....	76
Figura 35. Diagrama de secuencia ingresar videoteca.....	77
Figura 36. Diagrama de secuencia crear video.....	78
Figura 37. Diagrama de secuencia actualizar video.....	79
Figura 38. Diagrama de secuencia borrar video.....	80
Figura 39. Diagrama de secuencia crear usuario.....	81
Figura 40. Diagrama de secuencia modificar usuario.....	82
Figura 41. Diagrama de secuencia borrar usuario.....	83
Figura 42. Diagrama de secuencia crear autor.....	84
Figura 43. Diagrama de secuencia modificar autor.....	85
Figura 44. Diagrama de secuencia borrar autor.....	86
Figura 45. Diagrama de secuencia crear perfil.....	87
Figura 46. Diagrama de secuencia modificar perfil.....	88
Figura 47. Diagrama de secuencia borrar perfil.....	89
Figura 48. Diagrama de secuencia crear categoría.....	90
Figura 49. Diagrama de secuencia modificar categoría.....	91
Figura 50. Diagrama de secuencia borrar categoría.....	92
Figura 51. Diagrama de secuencia buscar video administración.....	93
Figura 52. Diagrama de secuencia buscar video web.....	94
Figura 53. Diagrama de actividades ingresar videoteca.....	95

Figura 54. Diagrama de actividad crear video.	96
Figura 55. Diagrama de actividades actualizar video.....	97
Figura 56. Diagrama de actividades borrar video.	98
Figura 57. Diagrama de actividades crear usuario.	99
Figura 58. Diagrama de actividades modificar usuario.....	100
Figura 59. Diagrama de actividades baja de usuario.....	101
Figura 60. Diagrama de actividades crear autor.	102
Figura 61. Diagrama de actividades modificar autor.	103
Figura 62. Diagrama de actividades borrar autor.....	104
Figura 63. Diagrama de actividades crear perfil.	105
Figura 64. Diagrama de actividades modificar perfil.....	106
Figura 65. Diagrama de actividades borrar perfil.	107
Figura 66. Diagrama de actividades crear categorías.....	108
Figura 67. Diagrama de actividades modificar categorías.	109
Figura 68. Diagrama de actividades borrar categoría.....	110
Figura 69. Diagrama de actividades búsqueda administración.....	111
Figura 70. Diagrama de actividades búsqueda web.	112
Figura 71. Patrón MVC basado en, < http://2.bp.blogspot.com/_IJSFA7nl84U/SAV8oOwuubI/AAAAAAAAAPc/kklas9BZn8E/s320/MVC.png > [En línea] [05/08/10]	115
Figura 72. Técnicas de prueba de caja blanca y caja negra basado en< http://www.lsi.us.es/docencia/get.php?id=361 > [En línea][05/08/10]	120
Figura 73. Grafo de complejidad ciclomática.....	123
Figura 74. Video cargado.	125
Figura 75. Cambio de estado de video y actualización de la información.	125
Figura 76. Interfaz de inicio de sesión.....	126
Figura 77. Interfaz de gestión del sistema.	126



Universidad Nacional Autónoma De México
Facultad de Contaduría y Administración



INTRODUCCIÓN



Antecedentes

A partir de la necesidad del área de Fomento Editorial de la Facultad de Contaduría y Administración por distribuir el acervo de video de una manera más ágil, se implementó por primera vez un sistema basado en web, dicho sistema fue llamado “Videoteca”, las características principales eran administrar el contenido y mostrarlo a la comunidad mediante una página administrativa y una pública respectivamente.

Actualmente el área de Fomento Editorial de la Facultad de Contaduría y Administración requiere de una nueva imagen para dar una mayor difusión a su acervo, también se requiere de una documentación formal del sistema, que cuente con una estructura adecuada para la paginación, al mismo tiempo debe haber un control adecuado para la manipulación de la información en los catálogos, de igual manera es necesario realizar búsquedas eficientes de los videos, y finalmente cubrir algunos lineamientos principales para las páginas facultativas que se encuentren bajo dominio “unam.mx” determinados en el documento del CATIC (Consejo Asesor en Tecnologías de Información y Comunicación).

Justificación del trabajo de titulación

En la Facultad de Contaduría y Administración es importante como apoyo académico contar con un sistema ágil de videoteca para una mayor difusión de sus videos, es necesario desarrollar un nuevo sistema de videoteca que resulte de fácil acceso, con una nueva vista más amigable y que tenga una mejor gestión del mismo; al mismo tiempo que cuente con la documentación necesaria para actualizaciones futuras.

Objetivo

El presente trabajo de titulación tiene como finalidad desarrollar un nuevo sistema de videoteca para la Facultad de Contaduría y Administración, el cual consiste en desarrollar el sistema de videoteca nuevamente y su página web, elaborar la documentación necesaria y detallada del mismo; la cual podrá ser utilizada para actualizaciones futuras, tomando como base la metodología de desarrollo de sistemas orientada a objetos llamada ICONIX que gracias a su soporte que tiene en dos metodologías reconocidas en el mundo del desarrollo de sistemas orientados a objetos RMC (*Rational Method Composer*) y XP (*Xtreme Programming*) que nos proveen de los artefactos necesarios para llevar a cabo todo el ciclo de vida de desarrollo de un sistema orientado a objetos.



Ventajas del nuevo sistema de videoteca:

- Mostrar una nueva imagen que brindara tanto al usuario final como al usuario que gestiona una nueva interfaz novedosa y en la cual se distribuyan apropiadamente los espacios, también sacando provecho de esta para aumentar el interés en todo el acervo en video de la Facultad de Contaduría y Administración.
- Con el desarrollo de este nuevo sistema se facilitara la búsqueda de videos para el usuario y administrador por medio de un sistema inteligente de búsqueda, mostrando un ambiente más amigable y preciso.
- Asegurar que la misma se apegará a los lineamientos principales que debe cubrir cualquier página web académica de la UNAM que esté bajo el dominio .unam.mx, tal como los define el documento publicado por el CATIC.
- Podrá ser usada en cualquier sistema operativo que cuente con un navegador web que reúna los requisitos necesarios.
- El acceso a la información de los videos brindara un servicio de constante actualización de información por medio de RSS, permitiendo que el usuario pueda estar informado una vez que se haya suscrito a esta fuente de contenido.
- Proporcionara una nueva función de descarga de video a dispositivos personales, logrando una mayor difusión del acervo de videoteca.

Alcance del trabajo de titulación

El alcance de este trabajo de titulación es el desarrollo de un nuevo sistema de videoteca con su página web, elaboración de la documentación y la implementación.

Con este nuevo sistema de videoteca se pretende:

- Dejar la documentación pertinente para la implementación y administración del sistema como un referente para la evolución y crecimiento del sistema implementado.
- Reemplazar al sistema actual dando una mejora a la gestión del acervo en video de la Facultad de Contaduría y Administración.
- Orientar a la página web de la videoteca dentro de los lineamientos principales que pide el documento del CATIC.



- Así mismo darle una mejor estructura de navegación conservando un diseño en el cual se aproveche el espacio y tamaño de la página web y en donde se puedan ver los videos que el público necesita y quiere ver.
- Mejorar la administración de los catálogos y la navegación del sistema generando para el usuario una búsqueda que sea más eficaz e inteligente al momento de gestionar el contenido.
- Contar con nuevas funciones tales como una difusión para los videos en dispositivos personales, redes sociales, el uso de servicios de información referente a los videos como los es el uso de RSS.

Resumen Capítular

Capítulo 1. Justificación de la metodología. En este capítulo se hace una comparación entre las metodologías de desarrollo de sistemas, tanto ágiles, pesadas e híbridas y justificando la utilización de una metodología híbrida para el desarrollo del sistema de la videoteca.

Capítulo 2. Análisis. En este capítulo se detalla la lista de requerimientos específicos, diagramas de clases, modelo de dominio, diagramas y especificaciones de los caso de uso, así mismo se muestra el prototipo del sistema.

Capítulo 3. Diseño. En este capítulo se da seguimiento a diseñar el sistema con base al diagrama Modelo E-R, las tablas con la relación de atributos de la base de datos así mismo se detallan los diagramas de robustez, secuencia y actividades realizados con base a UML 2.0®.

Capítulo 4. Implementación. En este capítulo se explica parte de la generación de código para el nuevo sistema, utilizando el *framework* de *Rails* y explicando la arquitectura MVC que es la encargada de seccionar el sistema, se definen pruebas de software y sus características, asimismo se describen las pruebas con las cuales se evaluó el nuevo sistema.



Universidad Nacional Autónoma De México
Facultad de Contaduría y Administración



CAPÍTULO 1. JUSTIFICACIÓN DE LA METODOLOGÍA UTILIZADA EN ESTE PROYECTO DE TITULACIÓN



Metodologías contempladas para el desarrollo del sistema

Para dar inicio al presente trabajo de titulación se investigaron algunas metodologías de desarrollo de sistemas entre las cuales describimos:

Pesadas o tradicionales:

- *Rational Method Composer*(RMC)
- *Objectory*

Agiles ó Ligeras:

- XP
- SCRUM

Híbridas:

- ICONIX

Pesadas o tradicionales

Rational Method Composer (RMC)

Es un producto de *Rational* (IBM®). Se caracteriza por ser iterativo e incremental, está centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Está basado en 5 principios:

- Adaptar el proceso
- Equilibrar prioridades
- Demostrar valor iterativamente
- Colaboración entre equipos
- Elevar el nivel de abstracción



La estructura dinámica de RMC es la que permite que éste sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las 4 fases descritas anteriormente:

- Inicio(También llamado inepción)
- Elaboración
- Desarrollo(También llamado implementación, construcción)
- Cierre (También llamado transición)

RMC en cada una de sus fases (pertenecientes a la estructura estática) realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema (entre otros). Estos artefactos (entre otros) son los siguientes:

Inicio:

- Documento visión.
- Especificación de requerimientos.

Elaboración:

- Diagramas de caso de uso.

Construcción:

- Documento arquitectura que trabaja con las siguientes vistas:

Vista lógica:

- Diagrama de clases.
- Modelo E-R (Si el sistema así lo requiere).

Vista de Implementación:

- Diagrama de secuencia.
- Diagrama de estados.
- Diagrama de colaboración.

Vista Conceptual:

- Modelo de dominio.

Vista física:

- Mapa de comportamiento a nivel de *hardware*.



Cierre:

Tiene como objetivo garantizar que los requisitos se hayan cumplido, con consentimiento y satisfacción de las partes involucradas. Esta fase se da un seguimiento a los entregables y a menudo se inicia con una versión “beta” del sistema o aplicación.

Otras actividades de esta fase incluyen la preparación del ambiente, se completan, identifican y corrigen defectos. Y finaliza con un cierre dedicado al aprendizaje que queda como experiencia para futuras labores.

Objectory

Es un proceso organizado para la construcción industrial de *software*. Este proceso de diseño está guiado por casos de uso, una técnica que se basa en el entendimiento de un sistema en la forma en la cual es usado.

- Modelo de casos de uso. Se basa en la descripción de elementos o usuarios externos al sistema (actores) y funcionalidad del sistema (casos de uso).
- Modelo de objetos. Representa la estructura estática de objetos. Puede contener objetos entidad, de interfaz y de control, entre otros tipos y relaciones de herencia y comunicación.
- Diagrama de interacción. Muestran la secuencia de eventos entre paquetes u objetos necesarios para realizar un caso de uso.
- Diagrama de estado. Muestra los estados internos de un objeto complejo.

Algunos de los conceptos más importantes de esta metodología son:

- Objeto entidad. Representa información del sistema que debe sobrevivir cierto período de tiempo, por ejemplo, un caso de uso.
- Objeto de interfaz. Modela información y comportamiento que es dependiente de la interfaz actual del sistema.
- Objeto de control. modela funcionalidad que no corresponde a ningún objeto en particular y que se presenta en algunos casos de uso. Estos objetos generalmente operan sobre varios objetos entidad, realizan algún algoritmo y retornan algún resultado a un objeto de interfaz.
- Paquete. módulo que contiene código, traducible a un módulo en el lenguaje de implementación.
- Unidad. En pruebas, desde una clase hasta un subsistema.



Agiles ó Ligeras

XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito del desarrollo del *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los programadores y propiciando un buen clima de trabajo.

Se basa en una retroalimentación continua entre el cliente y el equipo de trabajo, comunicación fluida entre participantes, simplicidad en las soluciones implementadas y coraje para enfrentar cambios.

Adecuada para proyectos con requisitos imprecisos y muy cambiantes.

Características:

- Desarrollo incremental e iterativo.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas.
- Programación por parejas.
- Interacción con el usuario final.
- Ahorro del código.
- Simplicidad del código.
- Los requisitos cambian (clientes indecisos).
- Proyectos con alto grado de riesgo.
- Grupos pequeños de programadores entre 2 y 12.

SCRUM

Sus principales características se pueden resumir en dos.

- El desarrollo de *software* se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente.
- La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.



Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costos se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

Híbridas

ICONIX

Es una metodología que abarca ambos conceptos de las metodologías ágiles y pesadas.

El proceso de ICONIX (*Rosenberg & Scott, 1999*) se define como un proceso de desarrollo de *software* práctico. ICONIX está entre la complejidad del RMC (*Rational Method Composer*) y la simplicidad y pragmatismo del XP (*Extreme Programming*), sin eliminar las tareas de análisis y diseño que XP no contempla.

Es un proceso simplificado en comparación con otros más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto. Fue elaborado por *Doug Rosenberg y Kendall Scott* a partir de la síntesis del proceso unificado (RMC). Presenta claramente las actividades de cada fase exhibe una secuencia de pasos que deben ser seguidos, ofrece soporte de UML®, dirigido por casos de uso y es un proceso iterativo e incremental.

Las tres características fundamentales de ICONIX son:

- Iterativo e incremental: varias iteraciones ocurren entre el desarrollo del modelo de dominio y la identificación de los casos de uso.
- Trazabilidad: cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación de los diferentes artefactos producidos.
- Dinámica de UML®: La metodología ofrece un uso “dinámico del UML®” como los diagramas de caso de uso, diagramas de secuencia y de colaboración.



Rosenberg y Scott destacan las siguientes tareas:

- Análisis de requisitos.
- Análisis y diseño preliminar.
- Diseño.
- Implementación como las principales tareas.

Análisis de Requisitos

Identificar en el “mundo real” los objetos y todas las relaciones de agregación y generalización entre ellos. Utilizar un diagrama de clases de alto nivel definido como modelo de dominio.

El trabajo es iniciado con un levantamiento informal de todos los requisitos que en principio deberían ser parte del sistema. Luego con los requisitos se construye el diagrama de clases, que representa las agrupaciones funcionales con que se estructura el sistema que se desarrolla. De generarse el sistema a este nivel de especificación, se obtendría el menú principal del sistema con la interfaces iniciales de los casos o actividades de cada división funcional. Los diagramas del segundo nivel o superior, accesibles a partir de cada escenario o estado del nivel anterior, representan los casos, actividades y secuencias de interacción de cada división funcional.

Presentar, si es posible, un prototipo rápido de las interfaces del sistema, los diagramas de navegación, etc., de forma que los clientes puedan comprender mejor el sistema propuesto. Con el prototipo se espera que las especificaciones iniciales estén incompletas.

En general se necesita entre 2 y 3 reuniones para establecer las especificaciones iniciales. La rapidez con la que se genera el sistema es esencial para que no se pierda el estado de ánimo sobre el proyecto y que los usuarios puedan comenzar a evaluar la aplicación en la mayor brevedad posible.

Durante la evaluación se debe capturar información sobre lo que les gusta y lo que les desagrada a los usuarios, al mismo tiempo poner atención al porque reaccionan los usuarios en la forma en que lo hacen.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. El proceso se repite varias veces y finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando es evidente que no se obtendrá mayor beneficio con una iteración adicional. El diseño de prototipos es una técnica popular de ingeniería para desarrollar modelos a escala(o simulados) de un producto o sus componentes. Cuando se aplica al desarrollo de sistemas de información el diseño de prototipos implica la creación de un modelo o modelos operativos de trabajo de un sistema o subsistema.



Existen cuatro tipos de prototipos:

1. Prototipo de viabilidad: para probar la viabilidad de una tecnología específica aplicable a un sistema de información.
2. Prototipo de Necesidades: utilizado para “descubrir” las necesidades de contenido de los usuarios con respecto a la empresa.
3. Prototipo de Diseño: es el que usa ICONIX. Se usa para simular el diseño del sistema de información final. Se centra en la forma y funcionamiento del sistema deseado. Cuando un analista crea un prototipo de diseño, espera que los usuarios evalúen este prototipo, como si formara parte del sistema final. Los usuarios deberían evaluar la facilidad de aprendizaje y manejo del sistema, así como el aspecto de las pantallas y los informes y los procedimientos requeridos para utilizar el sistema. Estos prototipos pueden servir como especificaciones parciales de diseño o evolucionar hacia prototipos de información.
4. Prototipo de Implantación: es una extensión de los prototipos de diseño donde el prototipo evoluciona directamente hacia el sistema de producción. Los prototipos de pantallas también proporcionan una manera de obtener las reacciones de los usuarios hacia la cantidad de información presentada sobre la pantalla de visualización. Tal vez el usuario decida que un diseño en particular es muy denso ya que existen demasiados detalles sobre la pantalla. En otros casos la información sobre la pantalla aunque no es excesiva en el sentido de causar que la pantalla se vuelva densa, tal vez sea mucho mayor que la que un individuo necesita durante todo el tiempo.

Ventajas:

- Los usuarios se hacen participantes más activos en los desarrollos del sistema. Suelen mostrarse más interesados en los prototipos de trabajo que en las especificaciones de diseño.
- La definición de necesidades se simplifica por el hecho de que muchos usuarios finales no comprenden o no son capaces de enumerar detalladamente sus necesidades hasta que ven un prototipo.
- La probabilidad de que los usuarios aprueben un diseño y luego rechacen su implantación se reducirá notablemente.
- Según se dice el diseño mediante prototipos reduce el tiempo de desarrollo, aunque algunos cuestionan este ahorro.



- Los prototipos suelen pasar a las fases de análisis y diseño con demasiada rapidez. Ello empuja al analista a pasar demasiado rápido a la codificación, sin haber comprendido las necesidades y los problemas. Condición deseable en un proceso ágil.

Identificar los casos de uso del sistema mostrando los actores involucrados. Utilizar para representarlo el modelo de casos de uso.

Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario; permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

Un caso de uso es una manera específica de utilizar un sistema. Es la imagen de una funcionalidad del sistema, desencadenada en respuesta a la estimulación de un actor externo.

El modelo de los casos de uso comprende los actores, el sistema y los propios casos de uso. El conjunto de funcionalidades de un sistema se determina examinando las necesidades funcionales de cada actor.

Los casos de usos reubican la expresión de las necesidades sobre los usuarios partiendo del punto de vista muy simple que dice que un sistema se construye ante todo para sus usuarios. La estructuración del método se efectúa respecto a las interacciones de una sola categoría de usuarios a la vez; esta partición del conjunto de necesidades reduce considerablemente la complejidad de la determinación de las necesidades.

Los casos de uso permiten a los usuarios estructurar y articular sus deseos; les obligan a definir la manera como querrían interactuar con el sistema, a precisar que informaciones quieren intercambiar y a describir lo que debe hacerse para obtener el resultado esperado.

Los casos de uso concretan el futuro sistema en una formalización próxima al usuario, incluso en ausencia de un sistema a criticar.

Organizar los casos de uso en grupos, o sea, utilizar los diagramas de paquetes.

Asociar los requisitos funcionales con los casos de uso y con los objetos del dominio (trazabilidad).

Un importante aspecto de ICONIX es que un requisito se distingue explícitamente de un caso de uso. En este sentido, un caso de uso describe un comportamiento; un requisito describe una regla para el comportamiento. Además, un caso de uso satisface uno o más requisitos funcionales; un requisito funcional puede ser satisfecho por uno o más casos de uso.



Análisis y Diseño Preliminar

Describir los casos de uso, como un flujo principal de acciones, pudiendo contener los flujos alternativos y los flujos de excepción. La principal sugerencia de ICONIX, en esta actividad, es que no se debe perder mucho tiempo con la descripción textual. Debería usarse un estilo consistente que sea adecuado al contexto del proyecto.

Realizar un diagrama de robustez. Se debe ilustrar gráficamente las interacciones entre los objetos participantes de un caso de uso. Este diagrama permite analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos participantes de cada caso de uso.

El análisis de robustez ayuda a identificar los objetos que participaran en cada caso de uso. Estos objetos que forman parte de los diagramas de robustez se clasifican dentro de los tres tipos siguientes:

- **Objetos de interfaz:** usados por los actores para comunicarse con el sistema. Son con los que los actores interactúan con el sistema, generalmente como ventanas, pantalla, diálogos y menús.
- **Objetos entidad:** son objetos del modelo del dominio. Son a menudo tablas y archivos que contiene archivos para la ejecución de dicho caso de uso.
- **Objetos de control:** es la unión entre la interfaz y los objetos entidad. Sirven como conexión entre los usuarios y los datos. Los controles son “objetos reales” en un diseño, pero usualmente sirven como una especie de oficinista para asegurar que no se olvide ninguna funcionalidad del sistema la cual puede ser requerida por algún caso de uso.

Esta técnica tan simple pero poderosa sirve como interfaz entre el “¿Qué?” y el “¿Cómo?” de un análisis. Además el análisis de robustez provee de una gran ayuda a saber si las especificaciones del sistema son razonables.

El análisis de robustez facilita el reconocimiento de objetos. Esto es un paso crucial ya que es casi seguro que se olvida algunos objetos durante el modelado del dominio y de esta manera se podrán identificar antes de que esto cause problemas serios, además sirve para identificar más y mejores clases, antes del desarrollo del diagrama de secuencias.

Las reglas básicas que se deben aplicar al realizar los diagramas de análisis de robustez:

- Actores solo pueden comunicarse con objetos interfaz.
- Las interfaces solo pueden comunicarse con controles y actores.



- Los objetos entidad solo pueden comunicarse con controles.
- Los controles se comunican con interfaces, objetos entidad y con otros controles pero nunca con actores.

Actualizar el diagrama de clases ya definido en el modelo de dominio con las nuevas clases y atributos descubiertas en los diagramas de robustez.

Diseño

Especificar el comportamiento a través del diagrama de secuencia. Para cada caso de uso identificar los mensajes entre los diferentes objetos. Es necesario utilizar los diagramas de colaboración para representar la interacción entre los objetos.

El diagrama de secuencia muestra interacciones entre objetos según un punto de vista temporal. El contexto de los objetos no se representa de manera explícita como en los diagramas de colaboración. La representación se concentra sobre la expresión de las interacciones.

A pesar de que a partir de los diagramas de casos de uso y de los diagramas de robustez ya se tiene entre un 75 y 80 por ciento de atributos de las clases identificadas, es hasta el diagrama de secuencia donde se empiezan a ver qué métodos llevarán las clases del sistema.

Eso se debe a que hasta que interactúan los objetos de las clases con los actores y con otros objetos de manera dinámica, hasta ese momento se tiene suficiente información como para poder empezar a especificar los métodos de las respectivas clases.

El diagrama de secuencia es el núcleo del modelo dinámico ya que describe los aspectos de un sistema que cambian con el tiempo. El modelo dinámico se utiliza para especificar e implementar los aspectos de control del sistema y muestra todos los cursos alternos que pueden tomar los casos de uso. Los diagramas de secuencia se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos (operaciones).

Terminar el modelo estático, adicionando los detalles del diseño en el diagrama de clases.

Verificar si el diseño satisface todos los requisitos identificados.



Implementación

Utilizar el diagrama de componentes, si fuera necesario para apoyar el desarrollo. Es decir, mostrar la distribución física de los elementos que componen la estructura interna del sistema.

El diagrama de componentes describe los elementos físicos y sus relaciones en el entorno de realización. El diagrama muestra las opciones de realización.

Escribir/ Generar el código.

La importancia de la interactividad, interactividad, accesibilidad y navegación en el software harán que el usuario se sienta seguro y cómodo al poder hacer uso de la aplicación sin inconvenientes tales como son los problemas de comunicación. Este y otros problemas como la realización de cambios, son factores que deben ser tenidos en cuenta.

Pero además hay que tener en cuenta factores como:

- La Reusabilidad: que es la posibilidad de hacer uso de los componente en diferentes aplicaciones.
- La Extensibilidad: que consiste en modificar con facilidad el software.
- La Confiabilidad: realización de sistemas descartando las posibilidades de error.

Realizar pruebas. Test de unidades, de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados.



Justificación de la metodología elegida

Finalmente después del análisis detallado de las metodologías que se contemplaron, se llegó a la conclusión de que la metodología ICONIX es la opción más óptima, puesto que es una fusión de tanto metodologías pesadas como ágiles, y es realmente adaptable si los recursos con los que se cuenta (humanos, financieros, infraestructura) pudieran ser limitados como en el caso de este proyecto, independientemente del lapso que tiene desde su creación y hasta el día de hoy ICONIX ha demostrado tener un alto grado de robustez al momento de generar entregables y documentación, no pierde su parte ágil al momento de desarrollar e implementar, además permitimos programar con el paradigma orientado a objetos obteniendo de esta manera un sistema adecuado para cubrir los requerimientos fundamentales del sistema.

Cabe mencionar que en un futuro este tipo de metodologías serán más comunes para el desarrollo e implementación de sistemas, ya que en la actualidad se requieren sistemas ágiles y versátiles en el tiempo de desarrollo y en su forma de controlar cambios. Así mismo al utilizar la metodología ICONIX en el desarrollo del nuevo sistema de Videoteca este será precursor en ser un sistema ágil desarrollado con una metodología híbrida.



Universidad Nacional Autónoma De México
Facultad de Contaduría y Administración



CAPÍTULO 2. ANÁLISIS



Análisis de requerimientos

Lista de requerimientos web

Nombre	Descripción	Importancia	Riesgo	Tipo	Nota
Distribuir contenido.	Distribuir y aprovechar el espacio de la página web.	3	7	U,S	
Estandarizar.	Cubrir con las normas que establece el CATIC para el desarrollo de sitios institucionales.	1	10	F,U,S,+ (Legal)	La estandarización en cada uno de sus puntos tomara su importancia dependiendo de la factibilidad en el proyecto.
Quitar animación.	Quitar la animación flash del encabezado y sustituirla, tomado en cuenta las normas del CATIC.	3	6	F	
Paginar videos.	Paginación de los videos, para la mejora de la distribución y búsqueda de los videos.	2	8	F,U	
Optimizar motor de búsqueda.	Optimizar el motor de búsqueda donde se tengan palabras clave más específicas y sencilla de usar.	2	9	F,U,P	
Crear criterio de visualización	Quitar criterio de visualización de videos más vistos, y remplazarlo con un criterio de videos más recientes. Ya que los videos con mayor demanda no han sido cambiados, es decir los más vistos siguen siendo los más vistos, y no permite la búsqueda o cambio de los videos mas nuevos e importantes además de ocupar un espacio importante en el sitio.	2	8	F,U,P,	
Generar video descargable para equipos portátiles.	Generar un video descargable para equipos portátiles.	5	5	+	



Crear un RSS.	Crear un RSS para los videos nuevos y más importantes.	5	5	F	
Crear el sitio más amigable.	Crear el sitio más amigable, a fin de que sea más intuitivo y fácil de utilizar por el usuario final.	4	7	U	

Tabla 1. Requerimientos web.

Lista de requerimientos del área de gestión

Nombre	Descripción	Importancia	Riesgo	Tipo	Nota
Paginar contenido de la lista de videos publicados.	Paginación del contenido de esta lista para una mejor distribución al momento de búsqueda de los videos agregados.	4	7	F, U	
Agregar Video.	Agregar un nuevo video al sistema de videoteca para ser publicado en la página web.	10	10	F, U	
Eliminar videos.	Borrar un video ya sea que este cargado, actualizado o publicado y que sea de baja demanda o que ya no es útil para la videoteca.	3	7	F, U	
Actualizar videos.	La actualización de los videos podrá optimizar la administración de la videoteca al poder reemplazar y actualizar un video que se encuentra dentro de esta, asimismo renombrar los archivos y/o videos publicados.	3	7	F, U	
Optimizar el motor de búsqueda y palabras clave.	Una optimización del motor de búsqueda y de las palabras clave para que sea más sencillo su uso.	2	7	U	
Jerarquizar categorías por áreas.	Agregar categorías dentro del árbol de categorías de los videos para optimizar la gestión.	2	5	U	



Universidad Nacional Autónoma De México

Facultad de Contaduría y Administración



Ingresar a la página de administración de la videoteca.	El ingreso al sistema con perfil administrador deberá permanecer activo sin un tiempo límite sin tener que firmarse nuevamente en el sistema.	2	5	F, +	Se deben tener en cuenta los límites y prioridades de seguridad que se cuentan o que se implantaran.
---	---	---	---	------	--

Tabla 2. Requerimientos del área de gestión.

Medición de *Riesgo e Importancia* en la matriz de requerimientos

El riesgo mide el impacto que tiene un requerimiento sobre el buen funcionamiento del sistema si no es desarrollado satisfactoriamente y se mide en una escala del 1 al 10.

La importancia mide la relevancia de un requerimiento en el proceso de desarrollo y se mide en una escala del 1 al 10



Diagrama de clases

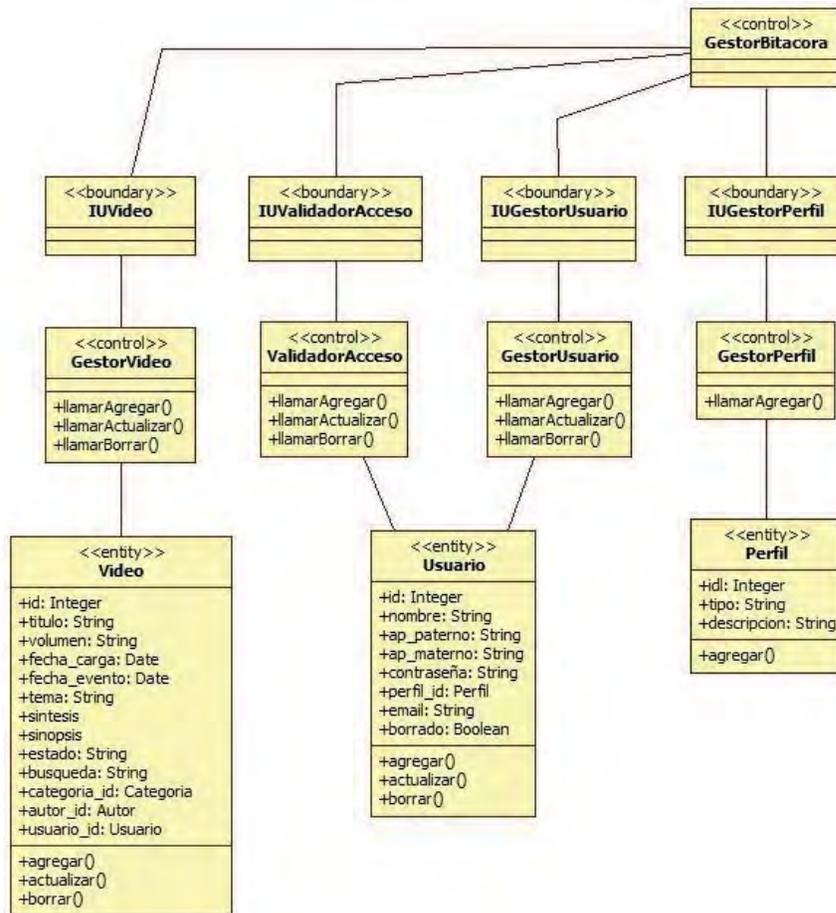


Figura 1. (a)



Continúa...

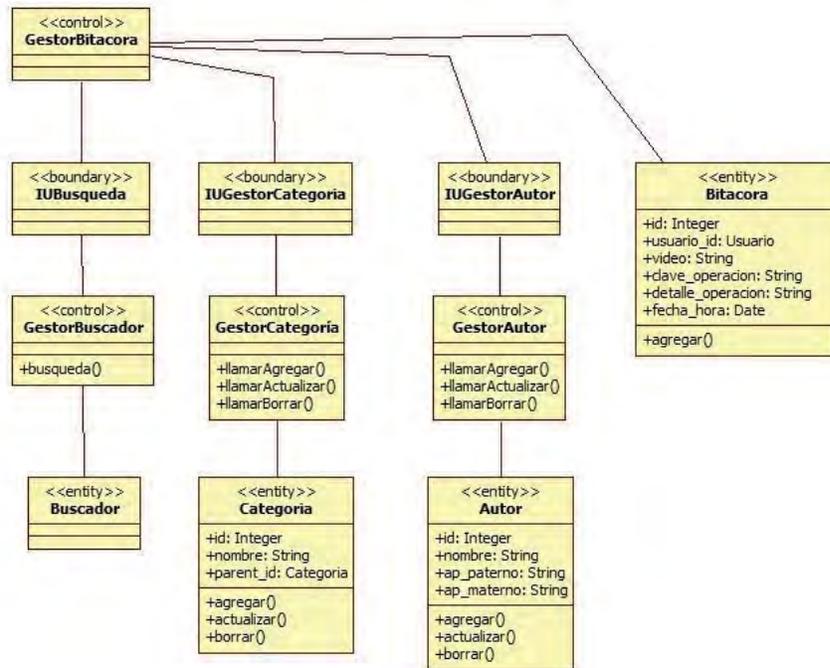


Figura 2. (b) Diagrama de clases, donde (a) y (b) son complementos y solo existe una clase bitácora.



Modelo de dominio

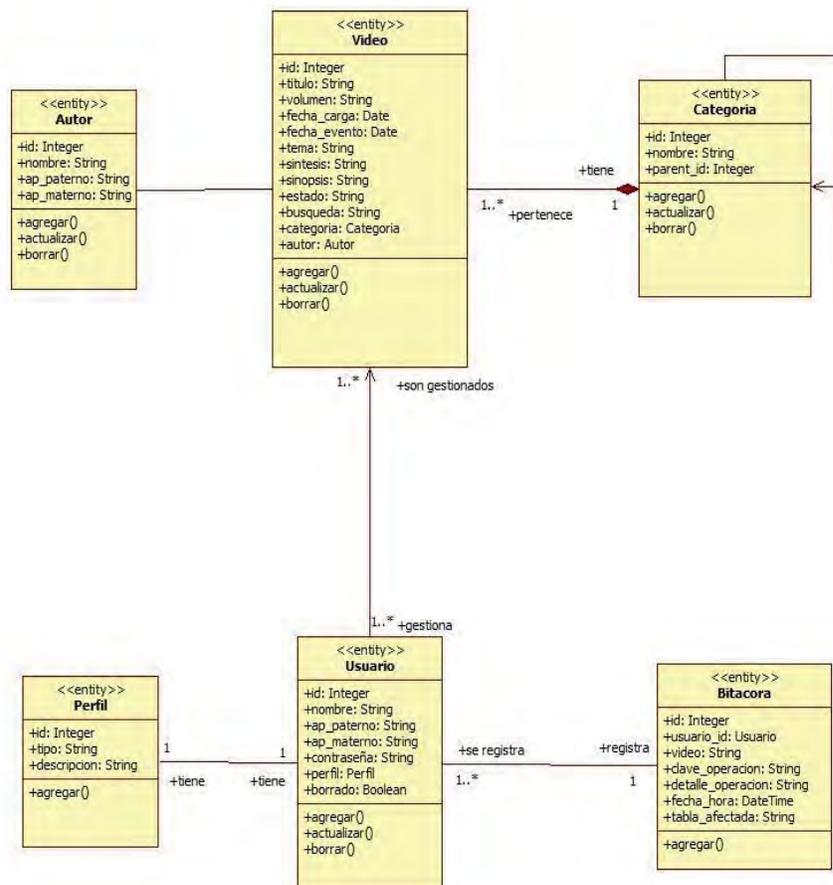


Figura 3. Modelo de dominio refinado donde se representan las clases con sus atributos, operaciones y cardinalidad.



Prototipo

En el prototipo preliminar del sistema se tienen varias secciones, en esta imagen están delimitadas por diferentes colores, cada una descrita de la siguiente manera:

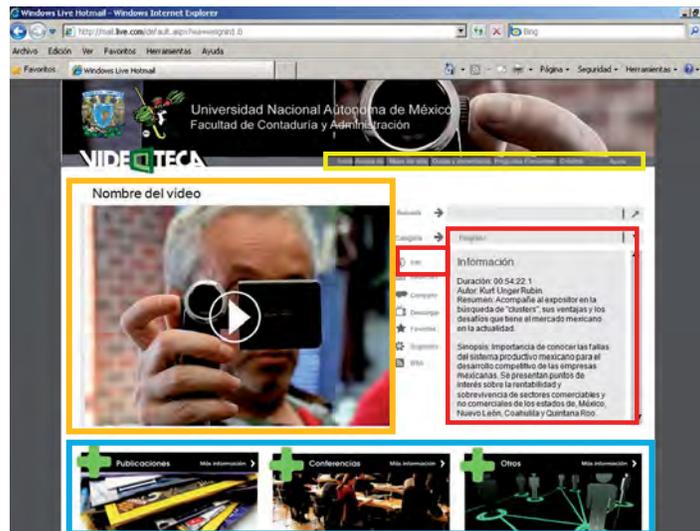


Figura 4. Prototipo del sistema videoteca sección video.

- La sección de video delimitada por color naranja es donde se reproducirá el video elegido por el usuario.
- La sección de descripción del video delimitada por color rojo es donde se detalla la información básica del video como autor, duración del video, categoría, sinopsis, etc.
- La sección de banners delimitada por color azul es donde se presenta la información y ligas a sitios los cuales se cambiarán conforme la importancia de estos y será actualizada por el área de Fomento Editorial.
- La sección del mapa de sitio delimitada por color amarillo se encuentran los links con los cuales se puede dirigirse a los apartados del portal que incluye inicio, mapa de sitio, dudas, comentarios, créditos y la ayuda de la página web.



Figura 5. Prototipo del sistema videoteca sección información.

- Existe también una sección de búsqueda delimitada por el color morado, donde se pueden poner datos referentes a algún video para poder localizarlo más fácilmente.

La parte delimitada por color azul se encuentran las siguientes secciones:

- En la sección de compartir, se puede recomendar el video a algún usuario con la finalidad de dar a conocer el acervo.
- En sección de descargar se cuentan con tres diferentes modalidades de resolución, baja calidad, media calidad, alta calidad, con su respectivo estimado tiempo de descarga para poder descargar el video elegido a tu dispositivo portátil.
- La sección de favoritos, proporciona los videos más vistos por los usuarios que visitan la videoteca.
- En la sección de sugeridos, despliega la información de los videos que son recomendados, ya sea por el tema, autor, etc.
- En la sección de recientes, muestra un listado de los videos que han sido agregados últimamente para que sean conocidos por la comunidad que visita la videoteca.
- En la sección de RSS, muestra un link en donde se puede inscribir a este servicio de distribución y recibir notificaciones actuales en línea.



Para los siguientes casos de uso se deberá tomar en cuenta lo siguiente:

Circunstancias generales para los casos de uso

Nombre	Descripción	Importancia	Riesgo	Tipo	Nota
Uso en línea.	El sistema de videoteca será visualizado por medio del navegador web.	10	10	<i>Usability</i>	
Paginación.	Paginación del contenido para una mejor distribución al momento de mostrar los resultados de búsqueda de los videos agregados.	7	4	<i>Usability, +(Empaquetamiento)</i>	Forma de distribución de los videos.
Servidor videoteca.	Servidor HP® Proliant ML320S G5 Servidor tipo rack 1 procesador Xeon 3060 @2.4 GHz 2 GB RAM 2 fuentes de poder redundantes, 700W 2 conexiones de red 1 GBit 6 discos duros ATA hot-plug 3.5" de 750 GB, en arreglo	8	8	<i>Reliability, Performance, Supportability</i>	



RAID-5 (capacidad total: 3 TB) El sistema operativo: Linux Debian (<i>OpenSource</i>) (4.0/testing) con <i>kernel</i> 2.6.22, arquitectura 32-bit (i686) Los componentes básicos para el funcionamiento de la videoteca son: Apache® 2.2.11 MySQL® 5.0 Ruby® 1.8.7 Rails® 2.3.5 HTML 4, 5 <i>Prototype</i> 1.5.1 jQuery 1.3.2 swfObject 1.5				
---	--	--	--	--

Tabla 3. Requerimientos específicos para los casos de uso.



Casos de uso

Diagrama general de sistema videoteca

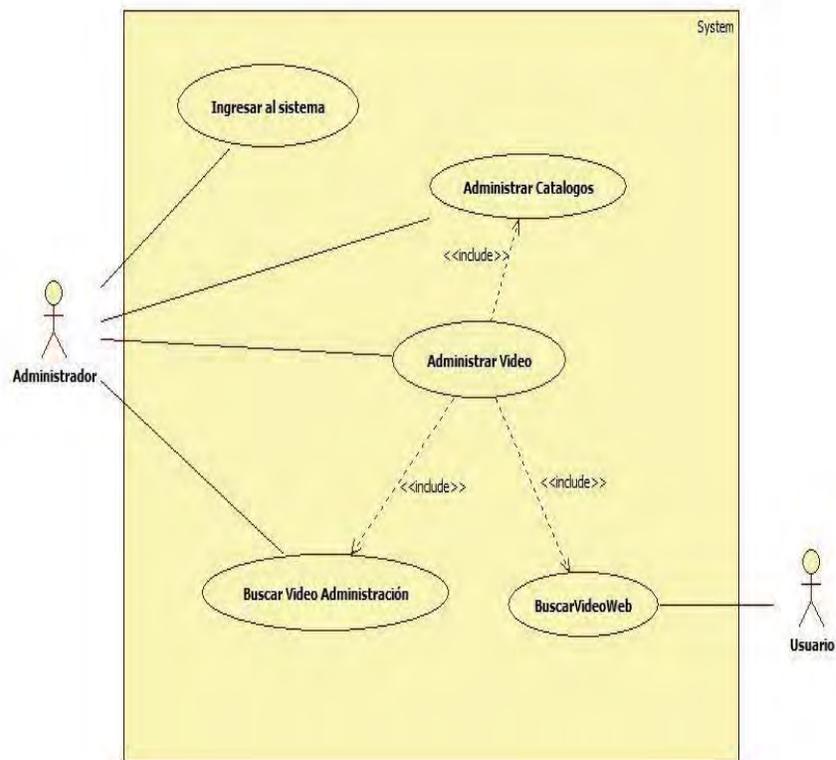


Figura 6. Diagrama general del sistema.



Ingresar videoteca

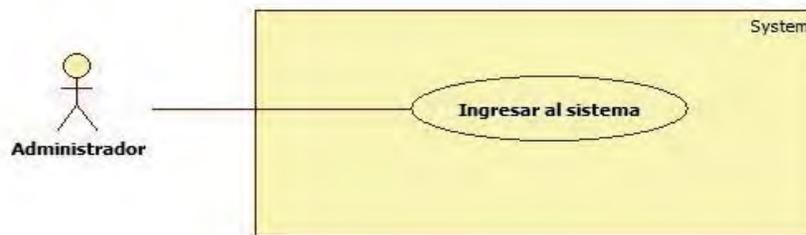


Figura 7. Diagrama ingresar al sistema.

Definición del Caso de Uso	
Propósito	Autenticación del administrador de la videoteca para acceder al sistema.
Unidad administrativa responsable	Fomento Editorial.
Modo de automatización	Este caso de uso es interactivo y será desarrollado a la medida.
Precondiciones	1. El administrador debe tener una cuenta para poder acceder al servicio, en caso contrario debe solicitar la creación de una cuenta.
Post-condiciones	1. Proporcionar acceso al sistema de acuerdo a un perfil de usuario.

Definición del flujo del proceso	
Actor primario	Administrador.
Actor secundario	N/A
Descripción:	Permitir al usuario ingresar al sistema de gestión de videoteca, verificando su nombre de usuario y contraseña para la gestión del contenido de la misma.



Flujo normal de los eventos

1	
Emisor	Acción: Ingresar al sistema de administración de videoteca.
Usuario	Ingresa al sistema de gestión de videoteca.
Sistema de videoteca	Muestra un formulario en el cual le pide su nombre de usuario y contraseña.
Usuario	Teclea su nombre de usuario y contraseña.
Sistema de videoteca	Valida y muestra el contenido principal del sistema de videoteca (http://videoteca.fca.unam.mx/admin) (flujo excepción 1, 2,3).

Flujo de excepción 1:

Emisor	Acción: El nombre de administrador esta Incorrecto.
Usuario	El usuario debe teclear su nombre de usuario correctamente.

Flujo de excepción 2:

Emisor	Acción: Contraseña incorrecta.
Usuario	El usuario debe teclear su contraseña correctamente.

Flujo de excepción 3:

Emisor	Acción: Nombre de administrador y/o contraseña olvidados.
Usuario	En este caso el usuario debe dirigirse con el administrador del sistema para que se le asigne un nuevo nombre de usuario y/o contraseña.



Administrar video

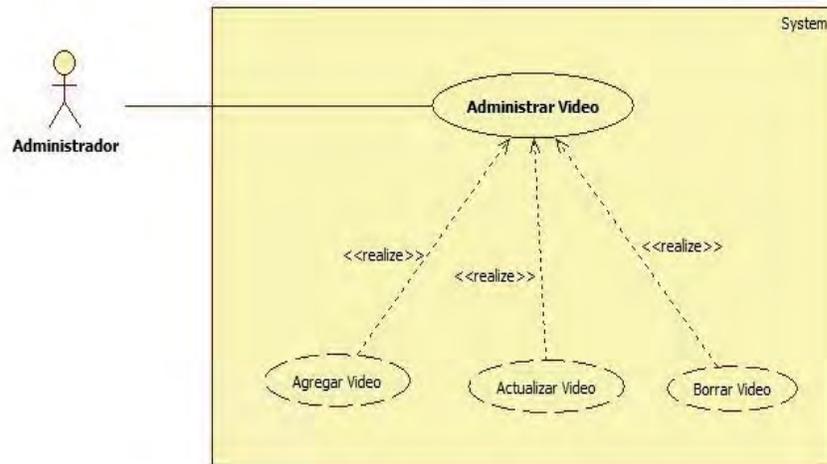


Figura 8. Diagrama administrar video.

Definición del Caso de Uso	
Propósito	Gestionar los videos dentro del sistema de videoteca para ser publicado en la página web.
Unidad administrativa responsable	Fomento Editorial.
Modo de automatización	Este caso de uso es interactivo y será desarrollado a la medida.
Precondiciones	El usuario debe tener una cuenta para poder acceder al sistema, en caso contrario debe solicitar la creación de una cuenta al usuario con perfil de administrador. El usuario deberá validarse e iniciar sesión exitosamente. Contar con el video para ser agregado al sistema. Contar con el programa para extraer los archivos de video.



Post-condiciones	Proporcionar acceso al sistema de acuerdo a un perfil de usuario. Guardar un registro en una bitácora de usuarios.
------------------	---

Circunstancias para el caso de uso administrar video

Nombre	Descripción	Importancia	Riesgo	Tipo	Nota
Mecatrón	Programa encargado de la extracción y conversión de los videos	10	10	Funcional	

Definición del Flujo del proceso

Actor primario	Administrador.
Actor secundario	Mecatrón
Descripción:	En este caso de uso de describe la manera de administrar los videos.

Flujo normal de los eventos

1:

Emisor	Acción: Crear video
Usuario	Ingresar el DVD al servidor.
Mecatrón	El programa extrae los archivos de video, decodifica y guarda los datos de volumen, fecha de carga, estado y duración. Genera 3 archivos de calidad de video. (.flv “baja calidad”, .mov “media calidad”, .mov “alta calidad”), los cuales se almacenan en el disco duro del servidor de videoteca, el archivo de video .flv es el que se mostrará en la página web en línea, la otros 2 archivos son solo para descarga.
Sistema de videoteca	El sistema publica el video dentro del sistema de gestión de videoteca con información inicial (volumen, fecha de carga, duración y estado este último se identificará con la letra “C” que corresponde a cargado) y se agregará a una lista de videos con el mismo estado.



1:	
Emisor	Acción: Crear video
Usuario	Ingresa al sistema de administración de videoteca.
Sistema de videoteca	Muestra un formulario en el cual le pide su nombre de usuario y contraseña al usuario. (Flujo de excepción 2,3 y 4).
Usuario	Ingresa a la sección de gestión de videos, e ingresa los datos (titulo, tema, síntesis, sinopsis, autor, categoría, estado, fecha de evento, búsqueda) a cualquier video con estado C dando click en link editar.
Usuario	El usuario da click al botón actualizar.
Sistema de videoteca	Valida que todos los campos hayan sido capturados correctamente (Flujo de excepción 7).
Sistema de videoteca	Cambia el estado de cargado a actualizado y agrega el video a la lista de videos con el estado de actualizado identificados por la letra "A".
Usuario	Verifica la lista de videos con estado A y la información actualizada (para cualquier video con estado A) y autoriza la publicación del mismo dando click en botón publicar.
Sistema de videoteca	Agrega video en la lista correspondiente a videos con el estado de publicado identificado con la letra "P" haciendo el cambio de estado "A" a "P" de manera automática, los videos con estado P serán los visualizados en el sistema, (flujo excepción 6).

2:	
Emisor	Acción: Actualización del video.
Usuario	Ingresa al sistema de gestión de videoteca.
Sistema de videoteca	Muestra un formulario en el cual le pide su nombre de usuario y contraseña al usuario.
Usuario	Captura su nombre de usuario y contraseña.
Usuario	El administrador teclea su nombre de usuario y contraseña (Flujos de excepción 2,3 y 4).
Sistema de videoteca	La aplicación valida y muestra el contenido principal de la aplicación.
Usuario	Ingresa a la sección de administración de videos, busca el video a modificar y da click en link modificar y se captura la información en los campos a elegir (fecha de evento,



2:	
Emisor	Acción: Actualización del video.
	título extendido, tema, autor, síntesis, sinopsis, categoría) y da click en botón modificar (Flujo de excepción 7).
Sistema de videoteca	El sistema muestra mensaje “video modificado satisfactoriamente”, guarda los cambios y actualiza la información.

3:	
Emisor	Acción: Borrar video.
Usuario	Ingresa al sistema de gestión de videoteca.
Sistema de videoteca	Muestra un formulario en el cual le pide su nombre de usuario y contraseña al usuario.
Usuario	El administrador teclea su nombre de usuario y contraseña.
Sistema de videoteca	El administrador teclea su nombre de usuario y contraseña (Flujos de excepción 2,3 y 4) presenta página de gestión del sistema.
Administrador	Ingresa a la sección de administración de videos, busca el video a borrar y da click en botón eliminar.
Sistema de videoteca	El sistema muestra una pantalla confirmando que el video será borrado con el mensaje “realmente quieres borrar video”.
Administrador	El administrador confirma que el video será borrado (flujo de excepción 5).
Sistema de videoteca	El sistema muestra un mensaje “video borrado exitosamente” (flujo de excepción 6).

Flujo excepción 1:	
Emisor	Acción: Información no actualizada.
Sistema de videoteca	Posible error de inserción en la base de datos.

Flujo excepción 2:	
Emisor	Acción: El nombre de usuario esta incorrecto.
Usuario	El usuario debe teclear su nombre de usuario correctamente.



Flujo excepción 3:	
Emisor	Acción: Contraseña incorrecta.
Usuario	El usuario debe teclear su contraseña correctamente.

Flujo excepción 4:	
Emisor	Acción: Nombre de administrador y/o contraseña olvidados.
Usuario	En este caso el usuario debe dirigirse con el administrador del servidor para que se le asigne un nuevo nombre de usuario y/o contraseña.

Flujo excepción 5:	
Emisor	Acción: Video equivocado.
Usuario	Elegir el video correcto a borrar.

Flujo excepción 6:	
Emisor	Acción: Video no borrado.
Sistema de videoteca	Posible error de inserción en la base de datos.

Flujo excepción 7:	
Emisor	Acción: Campos sin capturar.
Sistema de videoteca	Mandar mensaje con la leyenda “no puede estar vacío” anteponiendo el campo correspondiente.



Administrar catálogos

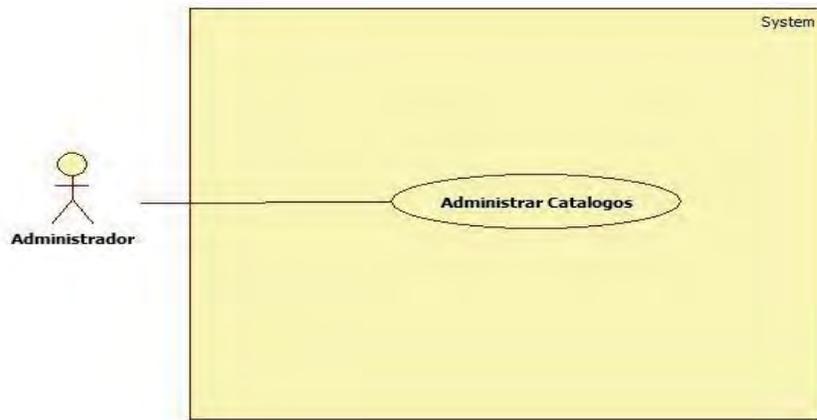


Figura 9. Diagrama administrar catálogos.

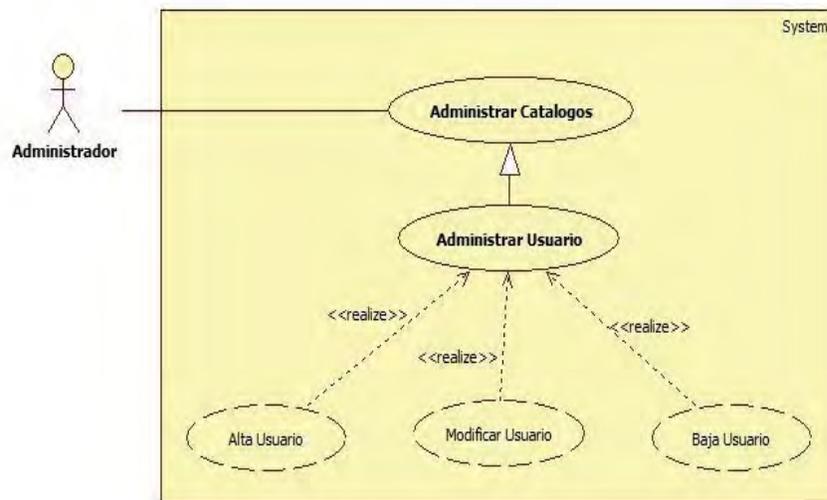


Figura 10. Diagrama administrar usuario.

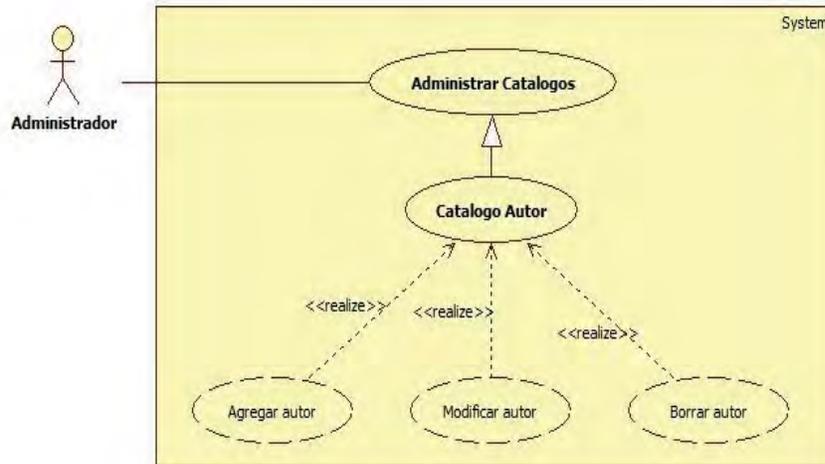


Figura 11. Diagrama catalogo autor.

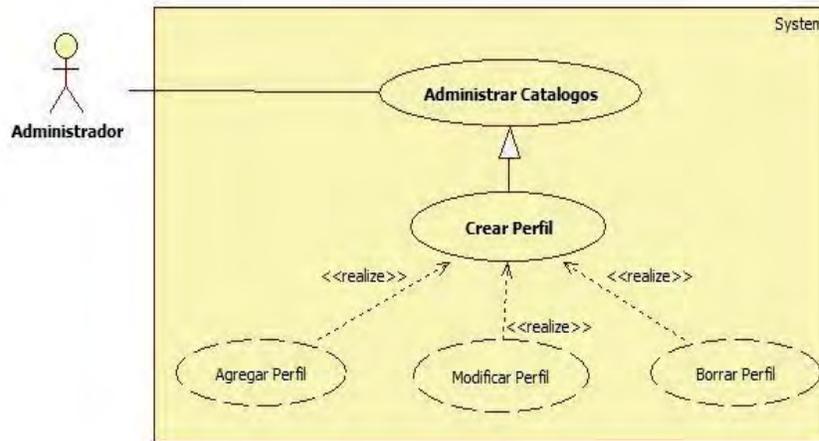


Figura 12. Diagrama crear perfil.

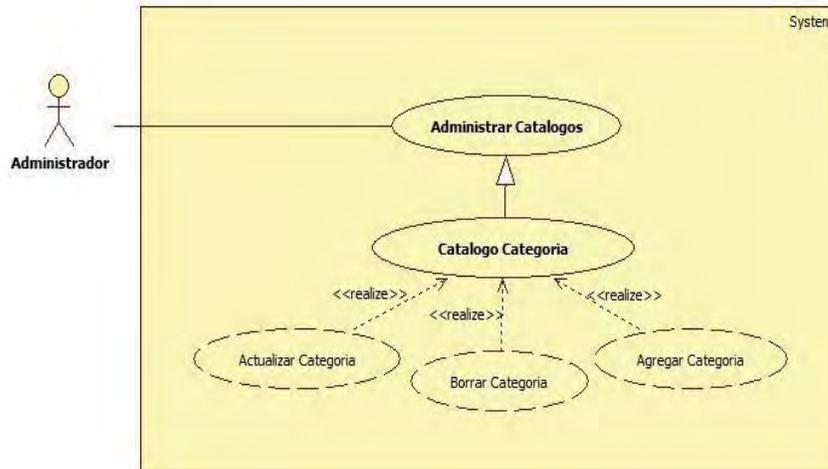


Figura 13. Diagrama catalogo categoría.

Definición del Caso de Uso	
Propósito	Gestionar la totalidad de los catálogos dentro del sistema de videoteca para ser publicado en la página web y en la interfaz administrativa.
Unidad administrativa responsable	Fomento Editorial.
Modo de automatización	Este caso de uso es interactivo y será desarrollado a la medida.
Precondiciones	a) El usuario debe tener una cuenta para poder acceder al sistema, en caso contrario debe solicitar la creación de una cuenta al usuario con perfil de administrador. b) El usuario deberá validarse e iniciar sesión exitosamente. c) Contar con los permisos necesarios para manipular la información contenida en los catálogos.



Post-condiciones	<ul style="list-style-type: none"> • Proporcionar acceso al sistema de acuerdo a un perfil de usuario. • Guardar un registro en una bitácora de usuarios. • Que el sistema actualice las bases de datos inmediatamente.
------------------	--

Circunstancias para el caso de uso administrar catálogos					
Nombre	Descripción	Importancia	Riesgo	Tipo	Nota
Jerarquizar	Agregar el árbol de categorías de los videos para optimizar la gestión.	6	8	<i>Usability</i>	
Enviar notificación	Enviar una confirmación al usuario por medio de un correo electrónico.	6	5	<i>Usability</i>	1. Los destinatarios deben tener una dirección de correo electrónico. 2. Los destinatarios han sido avisados de su alta en el sistema. 3. Esto se maneja en el



					catalogo de usuarios.
Uso de Plantilla	Debe existir una plantilla que permita establecer el título del correo y el cuerpo del mensaje. También debe ser posible modificar.	6	5	Usability	1. Esto se maneja en el catalogo de usuarios.

Definición del Flujo del proceso

Actor primario	Administrador.
Actor secundario	N/A
Descripción:	Permitir al administrador la gestión de los catálogos existentes.

Flujo normal de los eventos (catalogo usuarios)

1	
Emisor	Acción: Crear usuario.
Usuario	Ingresa al sistema de gestión de videoteca (Flujo de excepción 1,2 y 3).
Usuario	Elige el área de gestión de usuarios.
Sistema de videoteca	El sistema muestra el listado de usuarios en existencia.
Usuario	Da click en el link nuevo usuario.
Sistema de videoteca	Muestra los campos a llenar (nombre, apellido paterno, apellido materno, correo electrónico, password, confirmación de password, perfil).
Usuario	Ingresa los datos solicitados por el sistema y da click en botón crear.
Sistema de videoteca	Valida los campos capturados (flujo excepción 4).
Sistema de videoteca	Manda por correo electrónico los datos del usuario y contraseña que han sido dados de alta y los nuevos se agregan a la lista actualizada de usuario.
Sistema de videoteca	Manda a la sección de gestión de cuentas de usuario.



2	
Emisor	Acción. Modificar usuario (datos).
Usuario	Ingresa al sistema de gestión de videoteca (Flujo de excepción 1,2 y 3).
Usuario	Elige el área de gestión de usuarios.
Sistema de videoteca	El sistema muestra el listado de usuarios en existencia.
Usuario	Da clic en el link editar de algún usuario en específico para así modificar la información.
Sistema de videoteca	Muestra la información de la cuenta de usuario (nombre, apellido paterno, apellido materno, correo electrónico, password, confirmación de password perfil).
Usuario	Elige los campos a modificar.
Usuario	Cambia la información existente y da click en botón actualizar.
Sistema de videoteca	Valida que los campos modificados sean correctamente capturados (Flujo excepción 4).
Sistema de videoteca	Manda correo electrónico al usuario con la nueva información.
Sistema de videoteca	Manda a la sección de gestión de cuentas de usuario.

3	
Emisor	Acción. Borrar usuarios.
Usuario	Ingresa al sistema de gestión de videoteca (Flujo de excepción 1,2 y 3).
Usuario	Elige el área de gestión de usuarios.
Sistema de videoteca	Muestra información de usuario.
Sistema de videoteca	Elige la cuenta de usuario para dar de baja y da click en complemento y activar la casilla de borrado y se guardan los cambios.
Sistema de videoteca	Desactiva al usuario impidiéndole el acceso total. Pero guardando su registro en la base de datos.
Sistema de videoteca	Manda a la sección de gestión de cuentas de usuario.



Flujo normal de los eventos (catálogo autores)

1	
Emisor	Acción: Crear Autores.
Usuario	Ingresa a la página de gestión de videoteca (Flujo de excepción 1,2 y 3).
Usuario	Elige el área de gestión de autores.
Sistema de videoteca	El sistema muestra el listado de autores en existencia.
Usuario	Da click en link nuevo autor.
Sistema de videoteca	Muestra los campos a llenar (nombre, apellido paterno, apellido materno).
Usuario	Ingresa los datos solicitados por el sistema y da click en el botón crear.
Sistema de videoteca	Valida que todos los campos sean correctamente llenados. Si son correctos manda mensaje “autor creado satisfactoriamente” (Flujo excepción 4).
Sistema de videoteca	El sistema genera la nueva lista de autores.

2	
Emisor	Acción: Modificación de autor (datos).
Usuario	Ingresa a la página de gestión de videoteca (Flujo de excepción 1,2 y 3).
Usuario	Elige el área de gestión de autores.
Sistema de videoteca	Elige el autor a modificar y da click en link modificar.
Usuario	Muestra la información de la cuenta del autor (nombre, apellido paterno y apellido materno).
Usuario	Elige los campos a modificar.
Sistema de videoteca	Cambia la información existente y da click en actualización.
Sistema de videoteca	Valida que los campos estén completos si es así guarda los cambios (flujo excepción 4).
Sistema de videoteca	Manda a la sección de gestión de autores con la nueva lista.



3	
Emisor	Acción: Borrar autor.
Usuario	Elige el área de gestión de autores.
Usuario	Elige el autor para dar de baja.
Sistema de videoteca	Manda mensaje de advertencia para la baja de autor "Estas seguro que quieres eliminar el autor".
Usuario	Acepta la baja del autor.
Sistema de videoteca	Borra todo registro del autor elegido y todos los videos que correspondan al autor borrado cambian a "video sin autor".
Sistema de videoteca	Manda a la sección de gestión de cuentas de usuario.

Flujo normal de eventos (catálogo perfil)

1	
Emisor	Acción: Crear perfil.
Usuario	Ingresa a la página de gestión de videoteca (Flujos de excepción 1,2 y 3).
Usuario	Elige la sección de gestión de perfiles.
Sistema de videoteca	Muestra la lista de perfiles dados de alta.
Usuario	Da click en link nuevo perfil.
Sistema de videoteca	Muestra los campos a llenar (tipo, descripción).
Usuario	Ingresa los datos solicitados por el sistema (Que tipo de perfil ocupara el usuario: administrador, invitado, etc. y su descripción).
Sistema de videoteca	Valida que todos los campos sean correctamente llenados. Si son correctos seguir con el proceso, (flujo excepción 4).
Sistema de videoteca	Crea el nuevo perfil y manda a la sección de gestión de perfiles con la nueva lista.



2	
Emisor	Acción: Modificación de perfil (datos).
Usuario	Ingresa a la página de gestión de videoteca (Flujos de excepción 1,2 y 3).
Usuario	Elige el área de gestión de perfiles.
Usuario	Elige el perfil a modificar y da click en link modificar.
Sistema de videoteca	Muestra la información del perfil (tipo, descripción).
Sistema de videoteca	Elige los campos a modificar.
Usuario	Cambia la información existente y da click en actualizar.
Sistema de videoteca	Valida que los campos modificados sean correctamente llenados (flujo excepción 4).
Sistema de videoteca	Manda a la sección de gestión de perfiles con la nueva lista.

3	
Emisor	Acción: Borrar perfil.
Usuario	Ingresa a la página de gestión de videoteca (Flujos de excepción 1,2 y 3).
Usuario	Elige el área de gestión de cuentas.
Usuario	Elige la sección de gestión de perfiles.
Usuario	Elige el perfil para dar de baja y da click en eliminar.
Sistema de videoteca	Manda mensaje de advertencia para la baja el perfil “Estas seguro de eliminar el perfil”.
Usuario	Acepta la baja del perfil.
Sistema de videoteca	Borra todo registro del perfil elegido.
Sistema de videoteca	Manda a la sección de gestión de cuentas de usuario.

Flujo normal de los eventos (catálogo categorías)

1	
Emisor	Acción: Crear categoría.
Usuario	Ingresa al sistema de gestión de videoteca (Flujos de excepción 1,2 y 3).
Usuario	Ingresa al área de gestión de categorías.
Usuario	Da click en link crear categoría.
Sistema de videoteca	El sistema muestra el campo donde se especifica el nombre de la categoría.
Usuario	Para crear una categoría hija el usuario elige la categoría padre y le da click en el botón crear, de no ser escogida la categoría “padre” va a crear la categoría “hija” por default en la última categoría creada.



Sistema de videoteca	Muestra en el árbol de jerarquización la nueva categoría creada.
----------------------	--

2	
Emisor	Acción. Modificar categoría.
Usuario	Ingresa al sistema de gestión de videoteca (Flujos de excepción 1,2 y 3).
Usuario	Elige la categoría a modificar y da doble click a la categoría elegida.
Sistema de videoteca	Muestra el campo a modificar.
Usuario	Actualiza el campo a modificar.
Usuario	Da click fuera del campo del árbol.
Sistema de videoteca	Guarda los cambios y muestra en el árbol de jerarquización los cambios efectuados.

3	
Emisor	Acción. Borrar categoría.
Usuario	Ingresa El sistema de gestión de videoteca (Flujos de excepción 1,2 y 3).
Usuario	Elige la categoría a eliminar.
Usuario	Elimina la categoría dando click en el botón de borrar (si la categoría contiene algún video estos se pasan a la categoría "videos sin categoría" establecida por default).
Sistema de videoteca	Muestra el mensaje de advertencia para eliminar "¿Estas seguro de eliminar el nodo? (Flujo de excepción 2).

Flujo de excepción 1:	
Emisor	Acción: El nombre del administrador es incorrecto.
Usuario	El Usuario debe teclear su Nombre de Usuario correctamente.

Flujo excepción 2:	
Emisor	Acción: Contraseña Incorrecta.
Usuario	El Usuario debe teclear su Contraseña correctamente.



Flujo excepción 3:

Emisor	Acción: Nombre de administrador y/o contraseña olvidados.
Usuario	En este caso el usuario debe dirigirse con el administrador del servidor para que se le asigne un nuevo nombre de usuario y/o contraseña.

Flujo excepción 4:

Emisor	Acción: Campos sin capturar.
Usuario	Campos sin capturar, mandar mensaje con la leyenda “no puede estar vacío” anteponiendo el campo correspondiente.



Buscar video administración



Figura 14. Búsqueda video administración.

Definición del Caso de Uso	
Propósito	Buscar un video para ser mostrado.
Unidad responsable	administrativa Fomento Editorial.
Modo de automatización	Este caso de uso es interactivo y será desarrollado a la medida.
Precondiciones	<ol style="list-style-type: none"> 1. El Administrador debe tener una cuenta para poder acceder al servicio, en caso contrario debe solicitar la creación de una cuenta. 2. El usuario deberá validarse e iniciar sesión exitosamente. 3. Contar con el video para ser buscado en el sistema.
Post-condiciones	N/A.

Definición del Flujo del proceso	
Actor primario:	Administrador.
Actor secundario:	N/A.
Descripción:	Permitir al administrador conectarse al sistema, para buscar un video en el sistema para modificarlo o borrarlo.



Flujo normal de los eventos

1	
Emisor	Acción: Buscar un video en el acervo.
Usuario	Ingresa al sistema de gestión de videoteca (Flujo de excepción 1, 2 y 3).
Administrador	Captura en el campo búsqueda las palabras relacionadas al video que quiera buscar y da click en botón buscar.
Sistema de videoteca	Lee palabras de búsqueda y hace la búsqueda mediante una consulta en la base de datos basada en las palabras capturadas por el administrador.
Sistema de videoteca	Muestra resultados obtenidos (flujo de excepción 4).
Administrador	Selecciona el video.
Sistema	Muestra el video seleccionado.

Flujo de excepción 1:

Emisor	Acción: El nombre del administrador es incorrecto.
Usuario	El solicitante debe teclear su nombre de usuario correctamente.

Flujo de excepción 2:

Emisor	Acción: Contraseña Incorrecta.
Usuario	El solicitante debe teclear su contraseña correctamente.

Flujo de excepción 3:

Emisor	Acción: nombre de administrador y/o contraseña olvidados.
Usuario	En este caso el solicitante debe dirigirse con el administrador del servidor para que se le asigne un nuevo nombre de usuario y/o contraseña.



Universidad Nacional Autónoma De México

Facultad de Contaduría y Administración



Flujo de excepción 4:	
Emisor	Acción: No existe video.
Sistema de Videoteca	No existe video relacionado con las palabras capturadas por el administrador. Regresa al flujo normal de eventos o salir del sistema.



Buscar video web

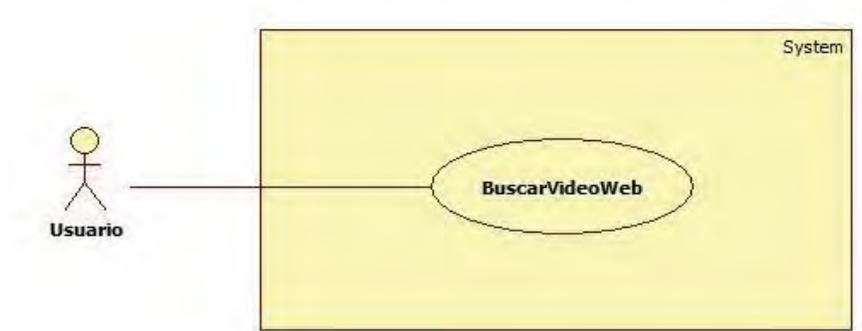


Figura 15. Diagrama buscar video web.

Definición del Caso de Uso	
Propósito	Buscar un video para ser mostrado.
Unidad administrativa responsable	Fomento Editorial.
Modo de automatización	Este caso de uso es interactivo y será desarrollado a la medida.
Precondiciones	1. Contar con el video para ser buscado en el sistema.
Post-condiciones	N/A.

Circunstancia de Uso					
Nombre	Descripción	Importancia	Riesgo	Tipo	Nota
Dominio de la videoteca.	Se debe conservar el dominio que tiene la videoteca http://videoteca.fca.unam.mx	10	10	<i>Supportability</i>	



Definición del Flujo del proceso

Actor primario:	Usuario.
Actor secundario:	N/A.
Descripción:	Permitir al usuario conectarse al sistema, para buscar un video.

Flujo normal de los eventos

1	
Emisor	Acción: Buscar video en la página web pública.
Usuario	Ingresa a la página web pública de videoteca http://videoteca.fca.unam.mx
Sistema de videoteca	Muestra el contenido principal de la aplicación.(flujo de excepción 1).
Usuario	Captura en el campo búsqueda las palabras relacionadas al video que quiera buscar y da click en botón buscar.
Sistema de videoteca	Lee palabras de búsqueda y hace la búsqueda mediante una consulta en la base de datos basada en las palabras capturadas por el administrador.
Sistema de videoteca	Muestra resultados obtenidos (flujo de excepción 2).
Usuario	Selecciona el video.
Sistema de videoteca	Muestra el video seleccionado.

Flujo de excepción 1:

Emisor	Acción. Falla del servidor.
Sistema de videoteca	Activar un servidor de respaldo con los archivos de video.

Flujo de excepción 2:

Emisor	Acción. No existe Video.
Sistema de videoteca	No existe video con los criterios seleccionados por el Usuario. Regresa al flujo normal de eventos o salir del sistema.



Universidad Nacional Autónoma De México
Facultad de Contaduría y Administración



CAPÍTULO 3. DISEÑO



Modelo E-R

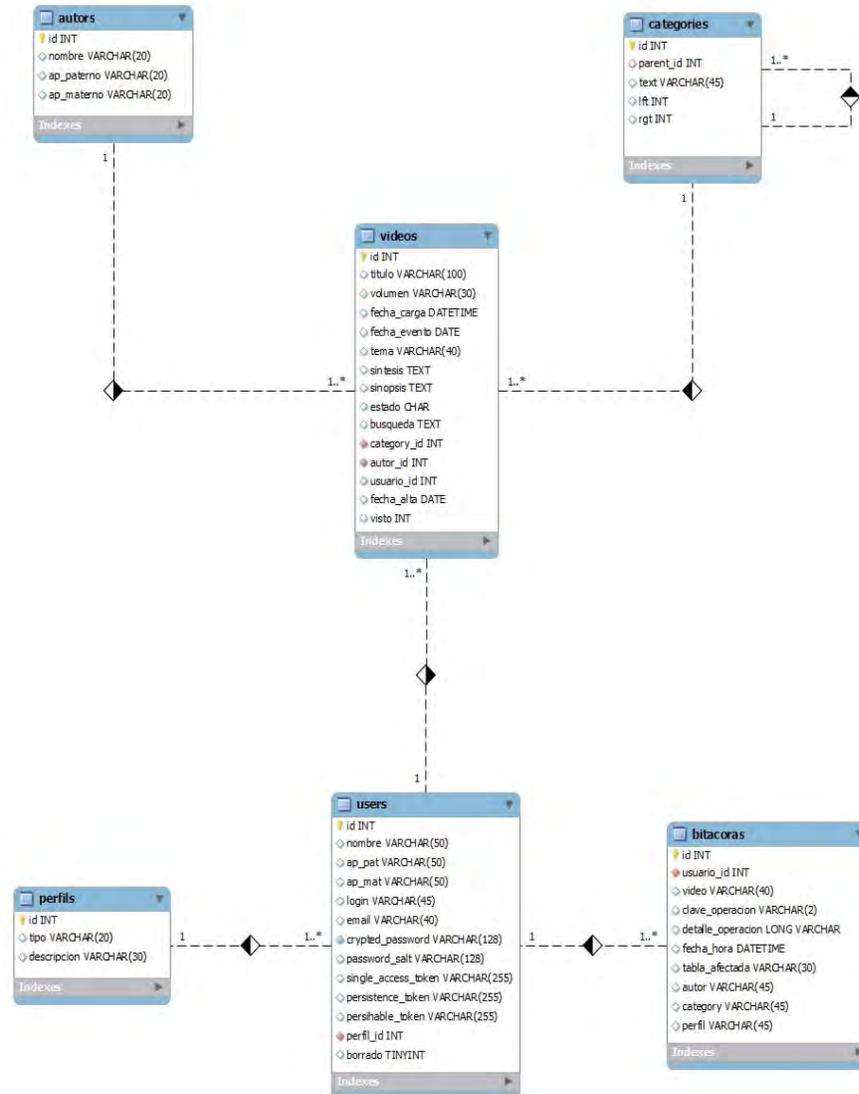


Figura 16. Diagrama modelo Entidad-Relación.



Relación de atributos

En *Rails* se genera automáticamente la base de datos sin embargo se usó la siguiente relación de atributos:

videos							
Columna	Tipo	Tamaño	Llave		Default	Restricción	NULL/NOTNULL
id	INTEGER		PK		nextval		NOTNULL
titulo	VARCHAR	100					NOTNULL
volumen	VARCHAR	30					NOTNULL
fecha_carga	DATE TIME						NOTNULL
fecha_evento	DATE						NOTNULL
tema	VARCHAR	40					NOTNULL
síntesis	TEXT						NOTNULL
sinopsis	TEXT						NOTNULL
estado	CHAR						NOTNULL
busqueda	TEXT						NOTNULL
category_id	INTEGER			FK		ON DELETE RESTRICT ON UPDATE CASCADE	NOTNULL
autor_id	INTEGER			FK		ON DELETE RESTRICT ON UPDATE CASCADE	NOTNULL
usuario_id	INTEGER			FK		ON DELETE RESTRICT ON UPDATE CASCADE	NOTNULL
fecha_alta	DATE						NOTNULL
visto	INTEGER						NOT NULL

Tabla 4. Atributos para la tabla videos de la base de datos.



users						
Columna	Tipo	Tamaño	Llave	Default	Restricción	NULL/NOTNULL
id	INTEGER		PK	nextval		NOTNULL
Nombre	VARCHAR	20				NOTNULL
ap_pat	VARCHAR	20				NOTNULL
ap_mat	VARCHAR	20				NOTNULL
Login	VARCHAR	45				NOT NULL
Email	VARCHAR	40				NOTNULL
crypted_password	VARCHAR	128				NOTNULL
password_salt	VARCHAR	128				NOT NULL
single_access_token	VARCHAR	255				NOTNULL
persistance_token	VARCHAR	255				NOTNULL
persihable_token	VARCHAR	255				NOTNULL
perfil_id	INTEGER			FK	ON DELETE RESTRICT ON UPDATE CASCADE	NOTNULL
Borrado	TINYINT					NOTNULL

Tabla 5. Atributos para la tabla users de la base de datos.

autores						
Columna	Tipo	Tamaño	Llave	Default	Restricción	NULL/NOTNULL
id	INTEGER		PK	nextval		NOTNULL
nombre	VARCHAR	20				NOTNULL
ap_paterno	VARCHAR	20				NOTNULL
ap_materno	VARCHAR	20				NOTNULL

Tabla 6. Atributos para la tabla autores de la base de datos.

perfiles						
Columna	Tipo	Tamaño	Llave	Default	Restricción	NULL/NOTNULL
id	INTEGER		PK	nextval		NOTNULL
Tipo	VARCHAR	20				NOTNULL
descripción	VARCHAR	30				NOTNULL

Tabla 7. Atributos para la tabla perfiles de la base de datos



categorias							
Columna	Tipo	Tamaño	Llave		Default	Restricción	NULL/NOTNULL
id	INTEGER		PK		nextval		NOTNULL
Text	VARCHAR	45					NOTNULL
parent_id	INTEGER			FK		ON DELETE / ON UPDATE CASCADE	NOTNULL
Lft	INTEGER						NOT NULL
Rgt	INTEGER						NOT NULL

Tabla 8. Atributos para la tabla categorias de la base de datos.

bitacoras							
Columna	Tipo	Tamaño	Llave		Default	restricción	NULL/NOTNULL
id	INTEGER		PK		nextval		NOTNULL
usuario_id	INTEGER			FK		ON DELETE / ON UPDATE RESTRICT	NOTNULL
Video	VARCHAR	40					NOTNULL
clave_operacion	VARCHAR	2					NOTNULL
detalle_operacion	LONG VARCHAR						NOTNULL
fecha_hora	DATETIME						NOTNULL
tabla_afectada	VARCHAR	30					NOTNULL
Category	VARCHAR	45					NOTNULL
autor	VARCHAR	45					NOTNULL
Perfil	VARCHAR	45					NOTNULL

Tabla 9. Atributos para la tabla bitácoras de la base de datos



Diagrama de robustez

Ingresar videoteca

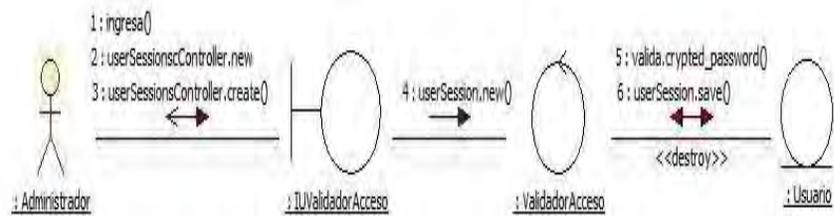


Figura 17. Diagrama de robustez ingresar videoteca.



Administrar video

- Crear video

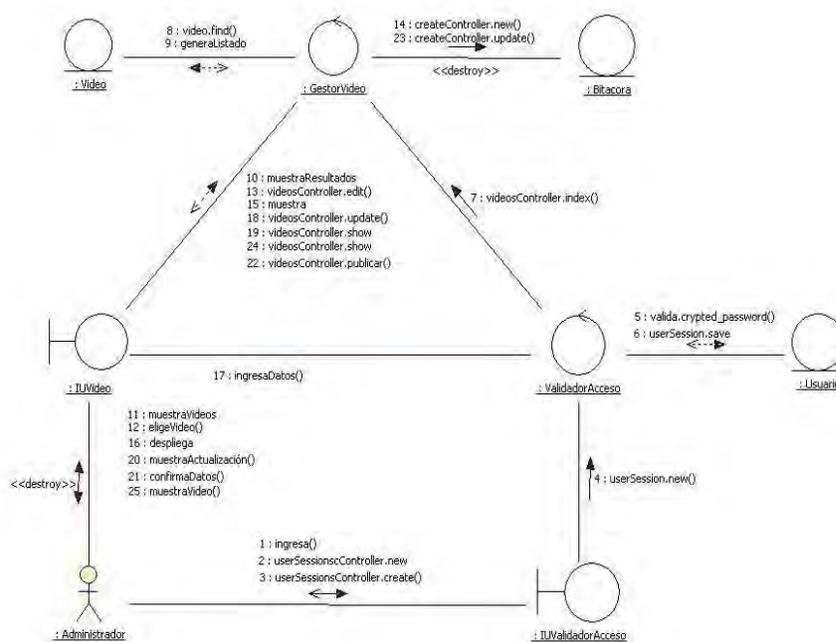


Figura 18. Diagrama de robustez crear video.



• Actualizar video

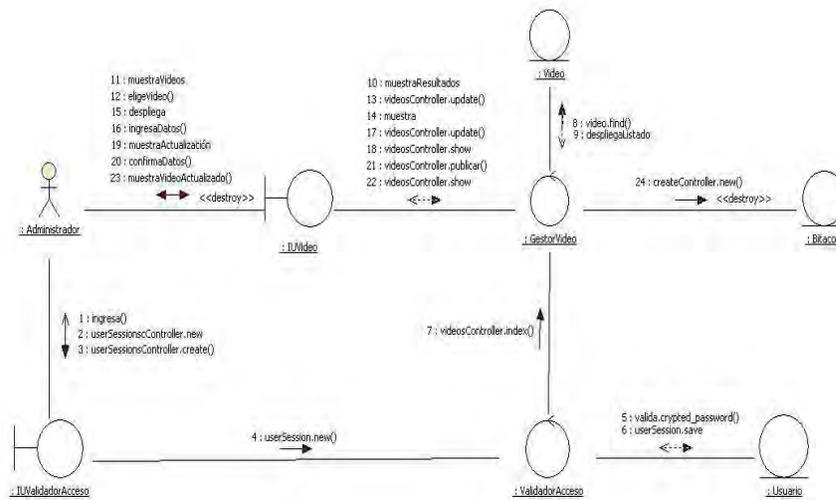


Figura 19. Diagrama de robustez actualizar video.



- Borrar video

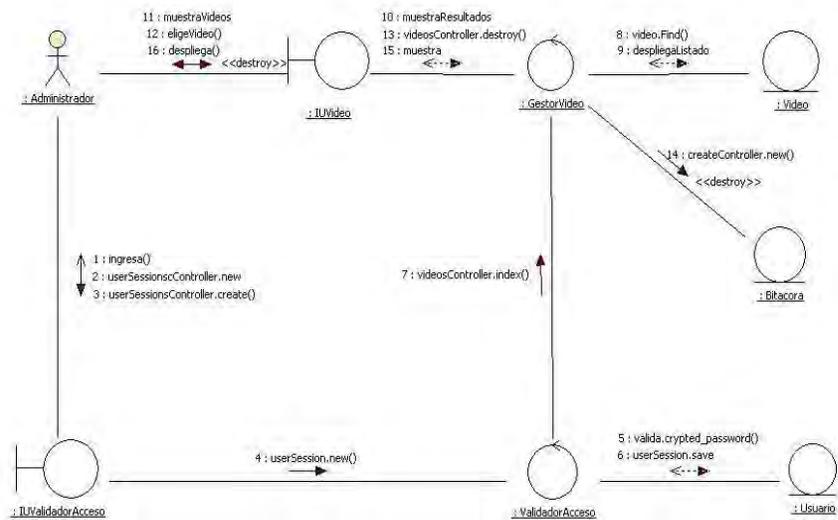


Figura 20. Diagrama de robustez borrar video.



Administrar catálogos

Usuario

- Crear usuario

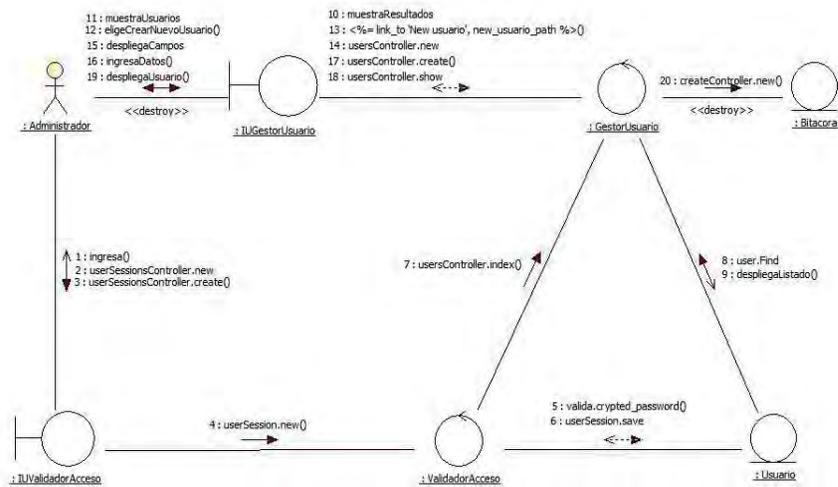


Figura 21. Diagrama de robustez crear usuario.



- **Modificar usuario**

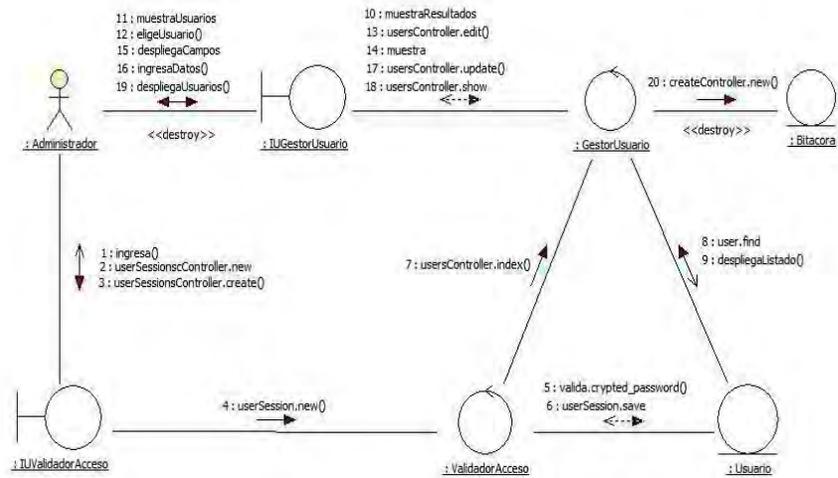


Figura 22. Diagrama de robustez modificar usuario.



- Borrar usuario

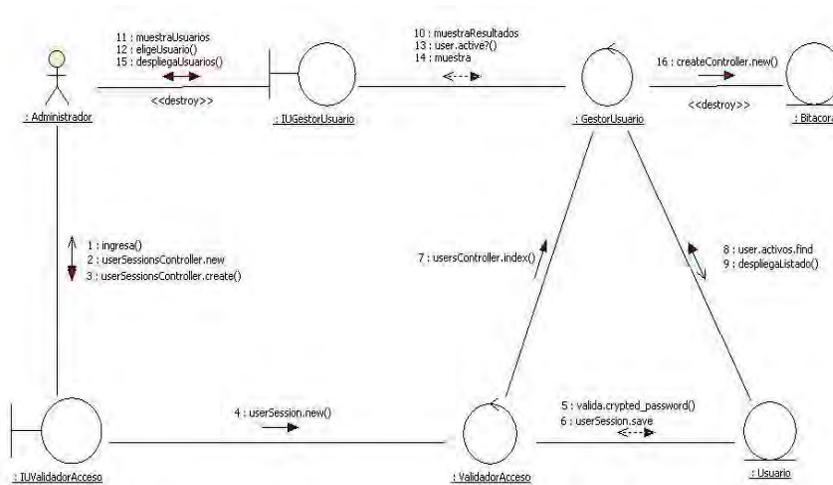


Figura 23. Diagrama de robustez borrar usuario.



Autor

- Crear autor

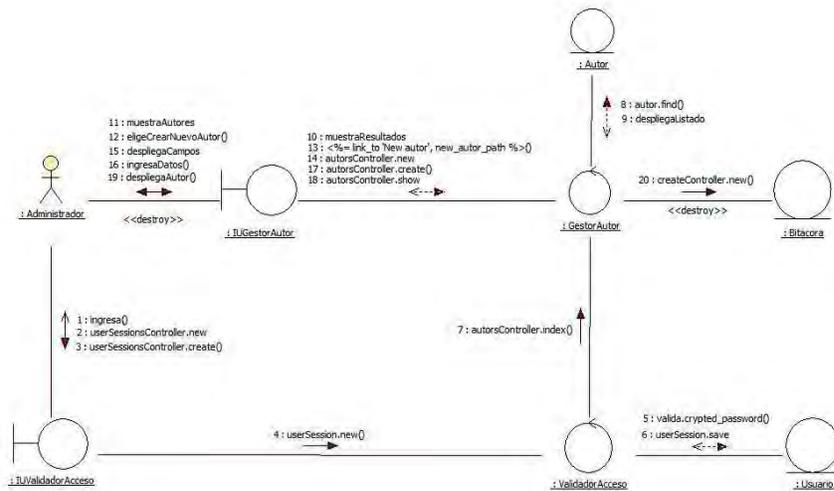


Figura 24. Diagrama de robustez crear autor.



- Modificar autor

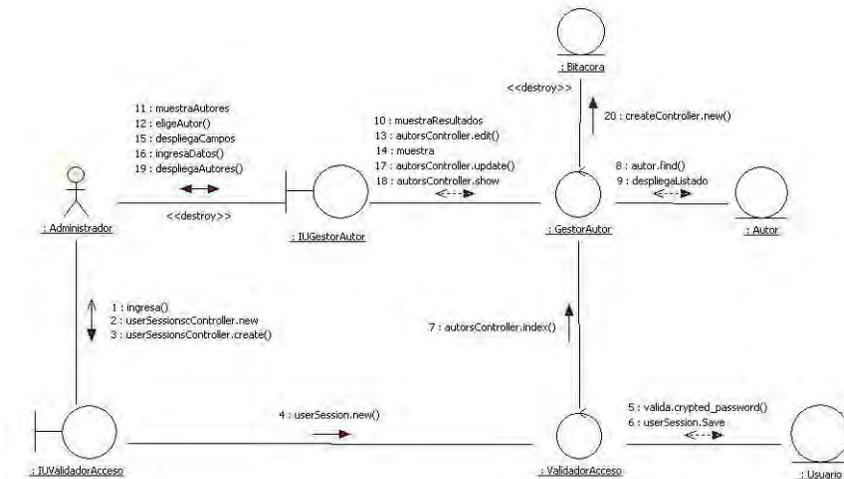


Figura 25. Diagrama de robustez modificar autor.



- Borrar autor

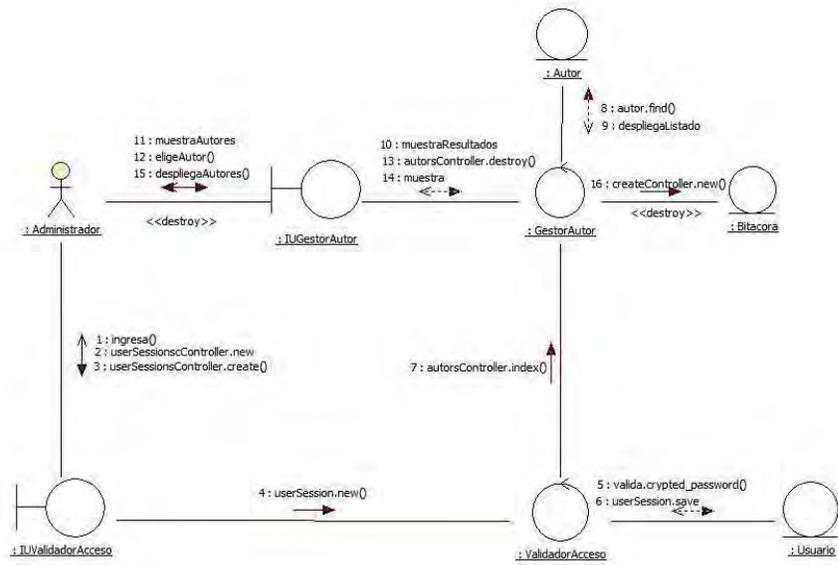


Figura 26. Diagrama de robustez borrar autor.



Perfil

- Crear perfil

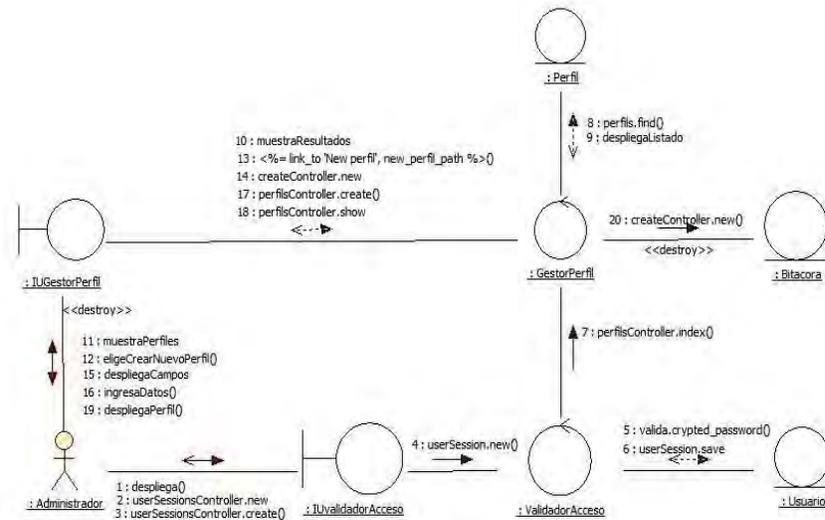


Figura 27. Diagrama de robustez crear perfil.



- Modificar perfil

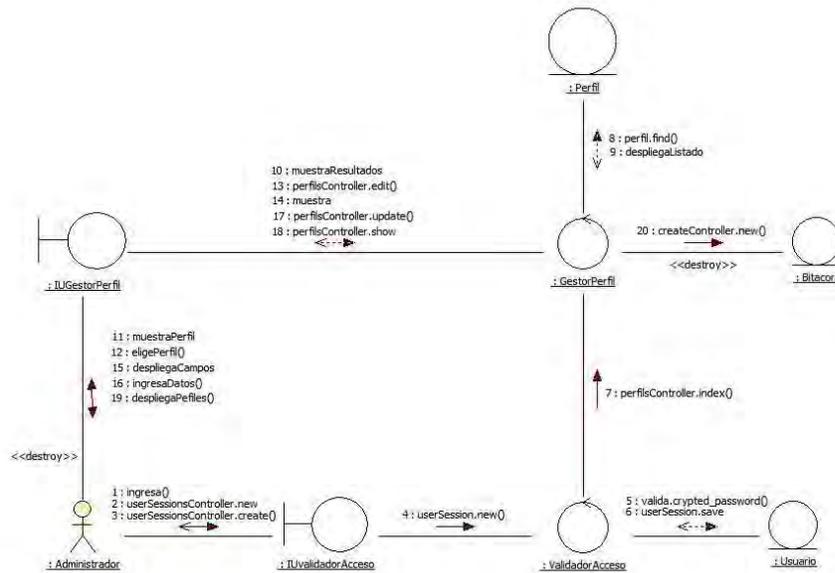


Figura 28. Diagrama de robustez modificar perfil.



- Borrar perfil

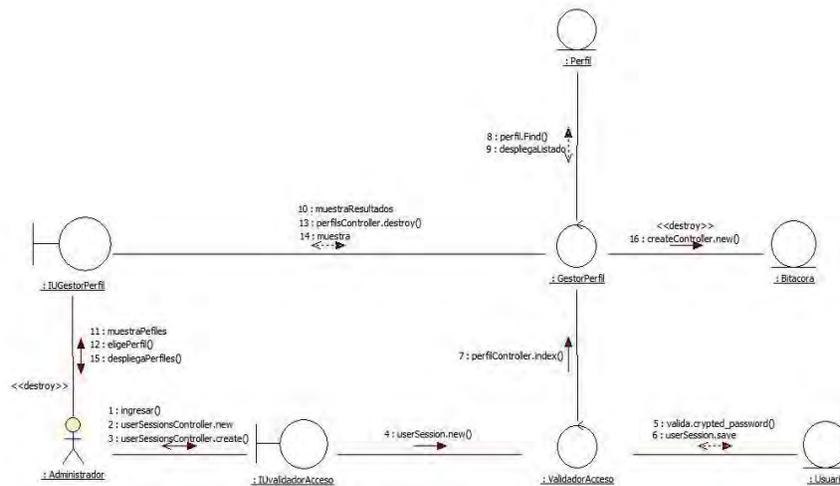


Figura 29. Diagrama de robustez borrar perfil.



- Modificar categoría

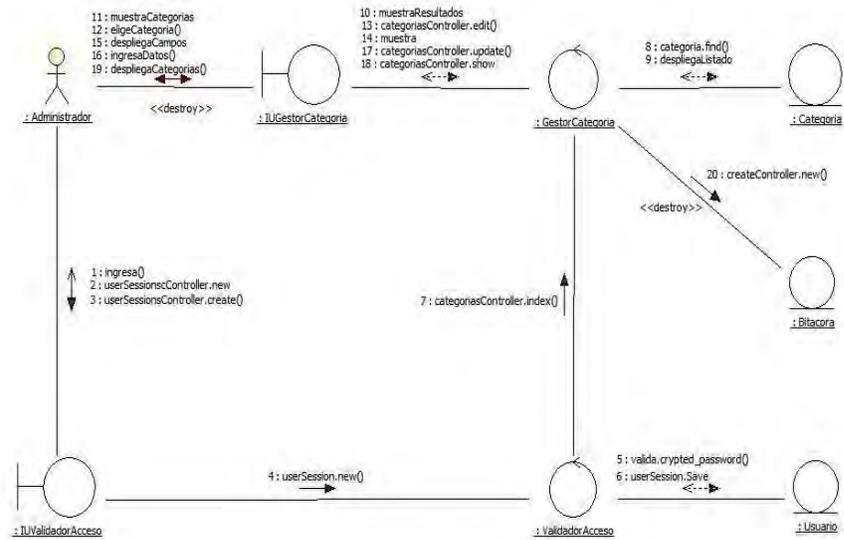


Figura 31. Diagrama de robustez modificar categoría.



- Borrar categoría

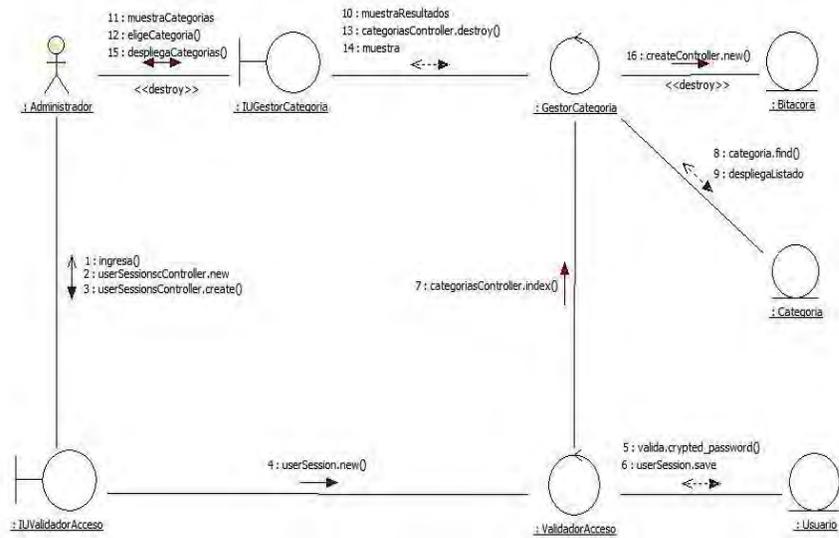


Figura 32. Diagrama de robustez borrar categoría.



Buscar video administración

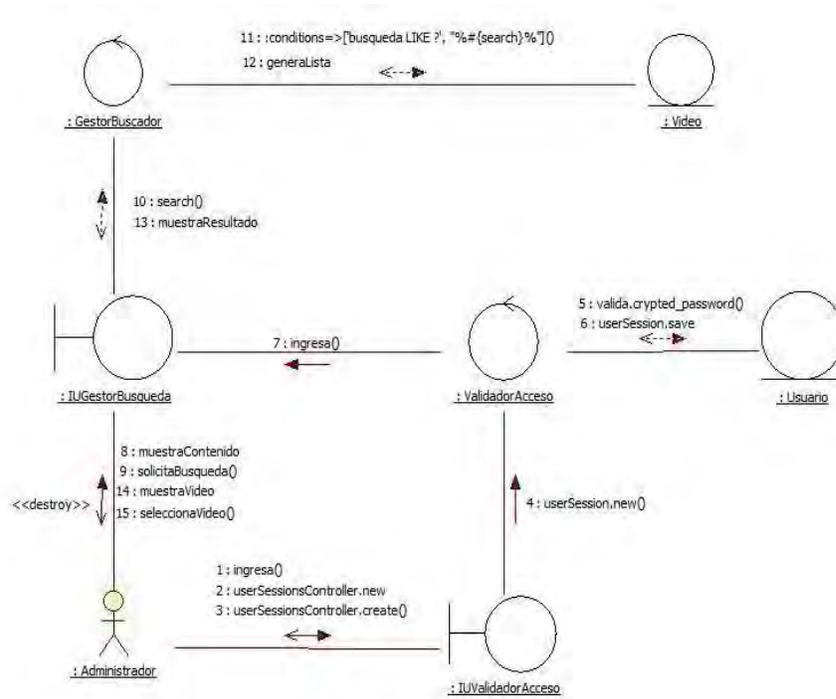


Figura 33. Diagrama de robustez buscar video administración.



Buscar video web



Figura 34. Diagrama de robustez buscar video web.



Diagramas de secuencia

Ingresar videoteca

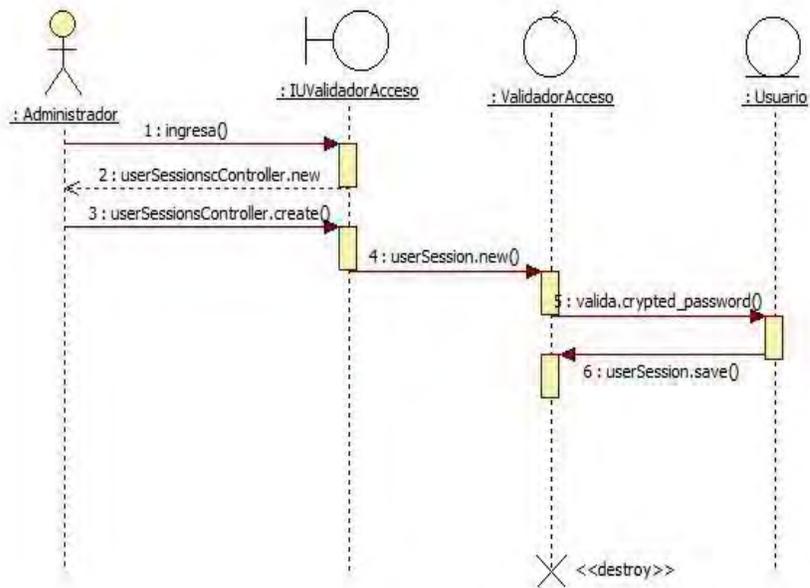


Figura 35. Diagrama de secuencia ingresar videoteca.



Administrar video

- Crear video

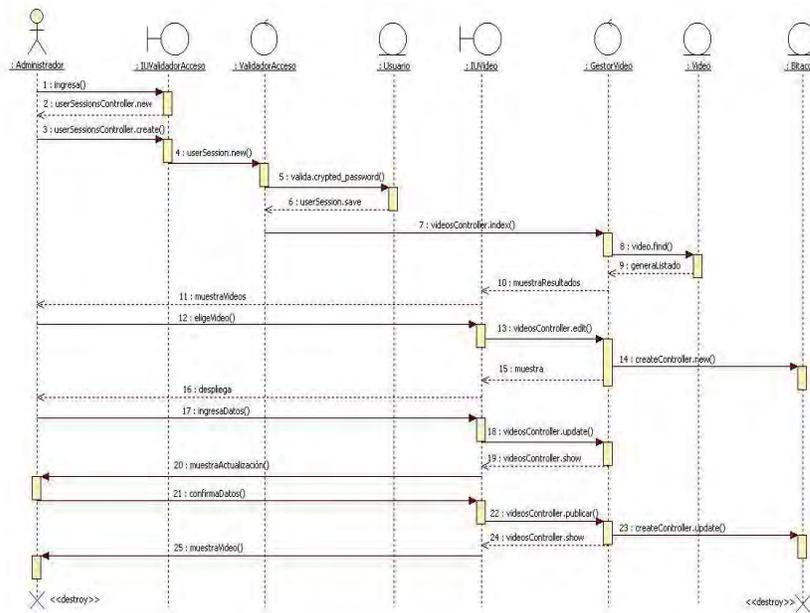


Figura 36. Diagrama de secuencia crear video.



- Actualizar video

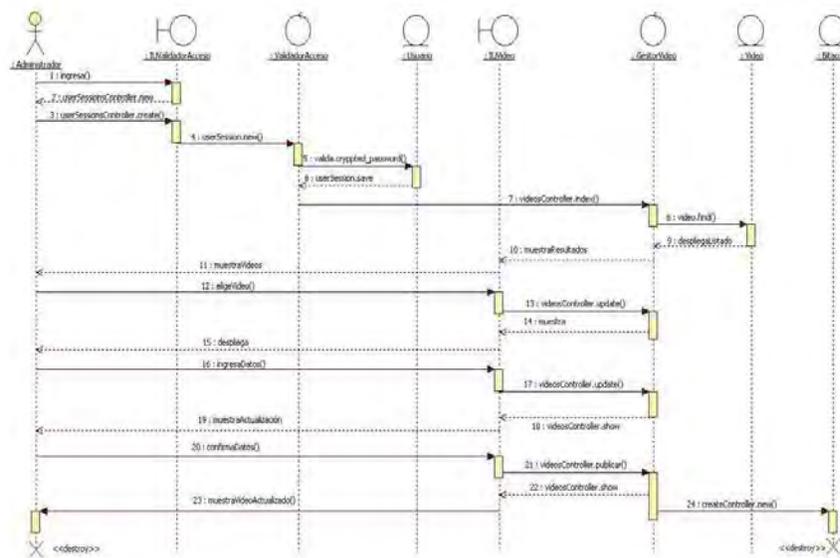


Figura 37. Diagrama de secuencia actualizar video.



- Borrar video

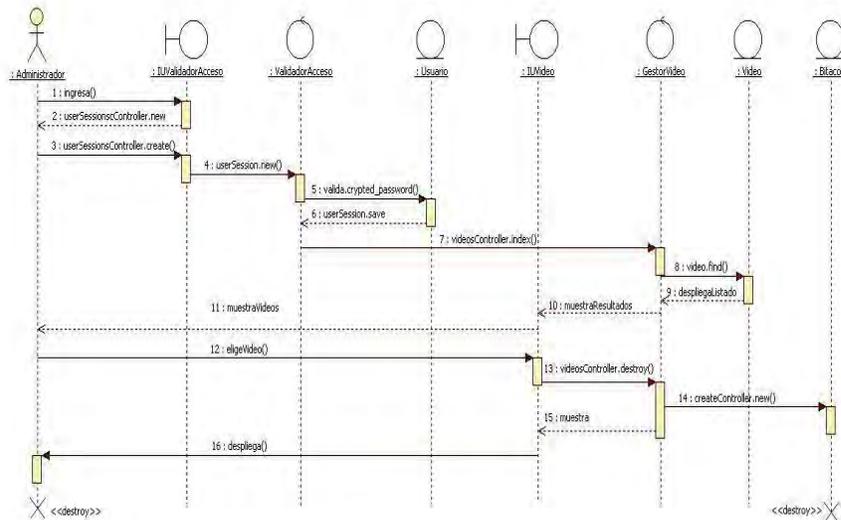


Figura 38. Diagrama de secuencia borrar video.



Administrar catálogos

Usuario

- Crear usuario

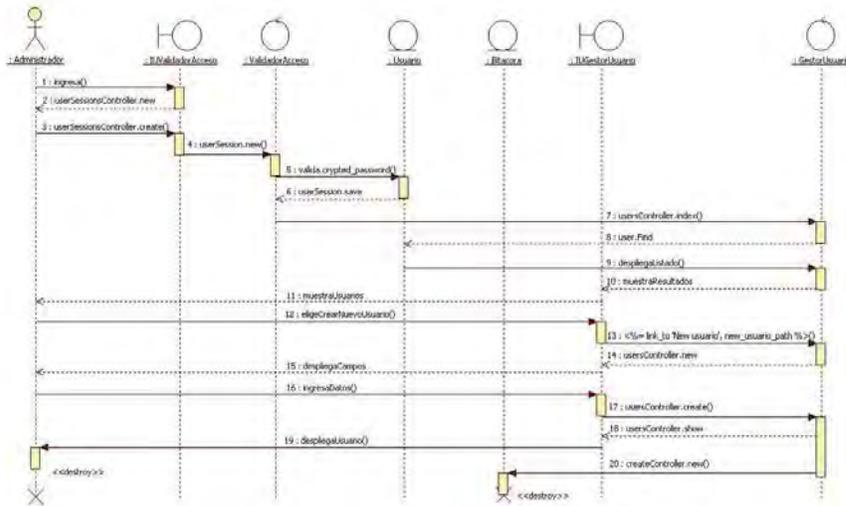


Figura 39. Diagrama de secuencia crear usuario.



- **Modificar usuario**

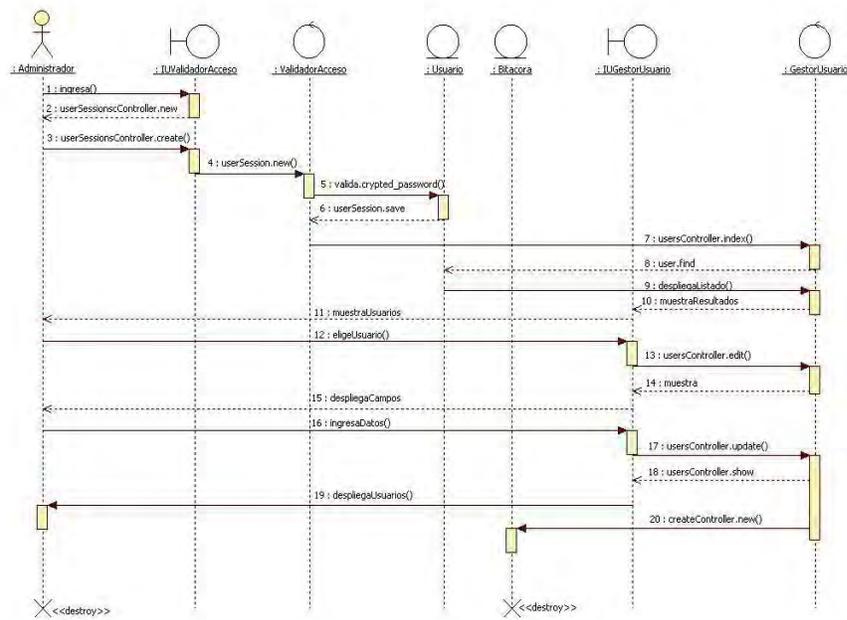


Figura 40. Diagrama de secuencia modificar usuario.



- **Borrar usuario**

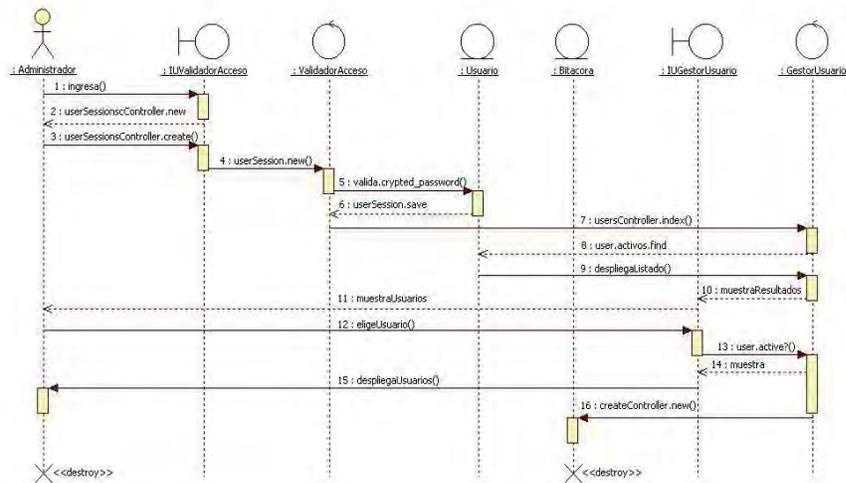


Figura 41. Diagrama de secuencia borrar usuario.



Autor

- Crear autor

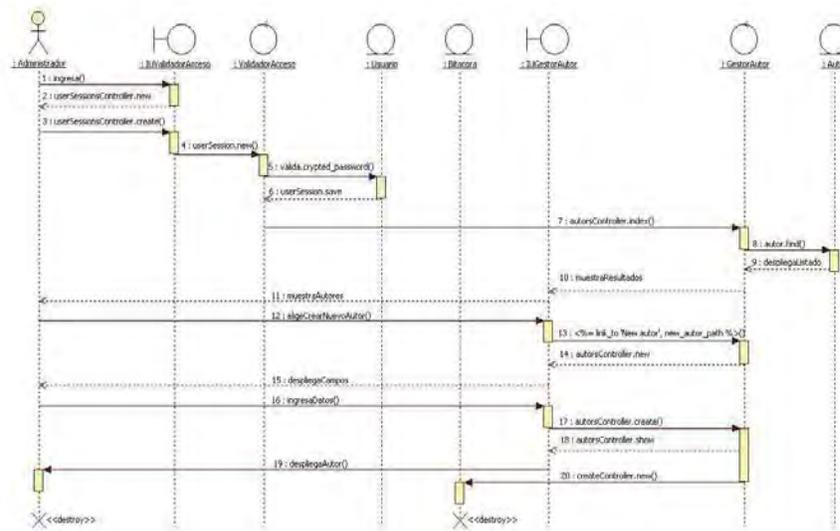


Figura 42. Diagrama de secuencia crear autor.



- Modificar autor

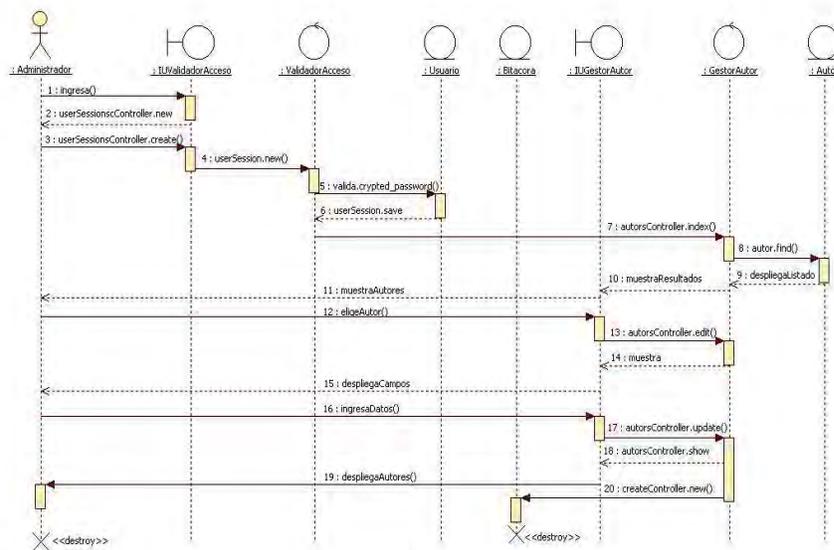


Figura 43. Diagrama de secuencia modificar autor.



- **Borrar autor**

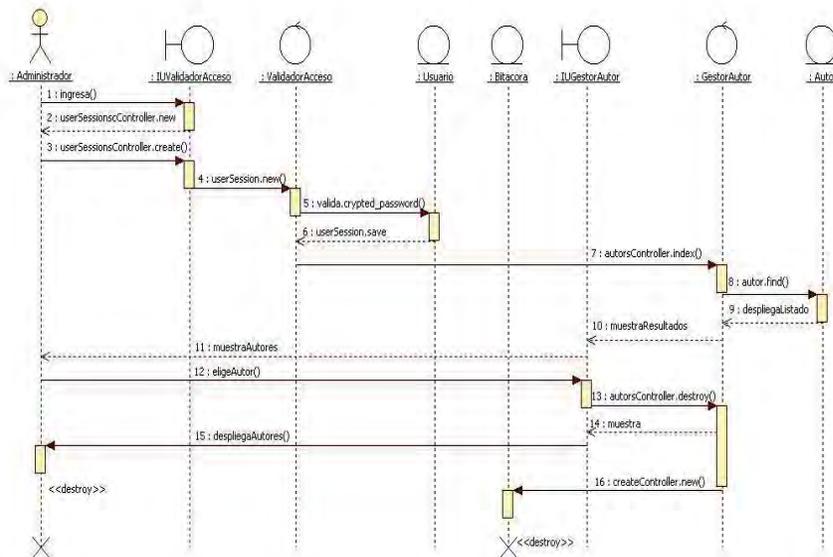


Figura 44. Diagrama de secuencia borrar autor.



Perfil

- Crear perfil

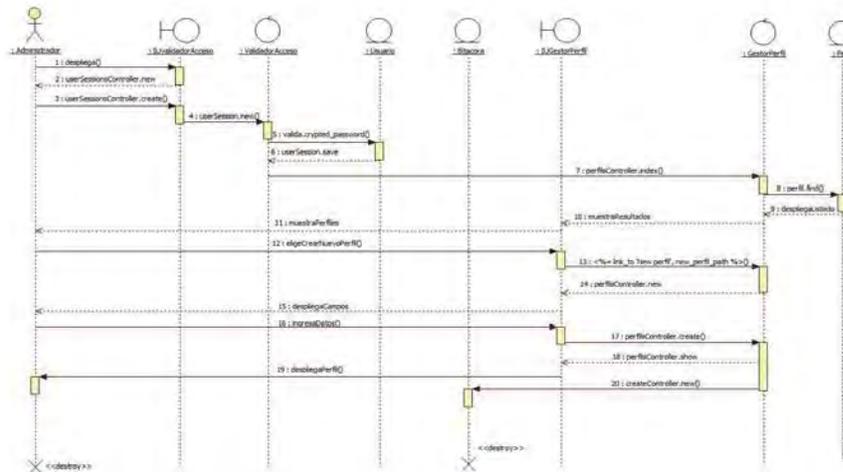


Figura 45. Diagrama de secuencia crear perfil.



- Modificar perfil

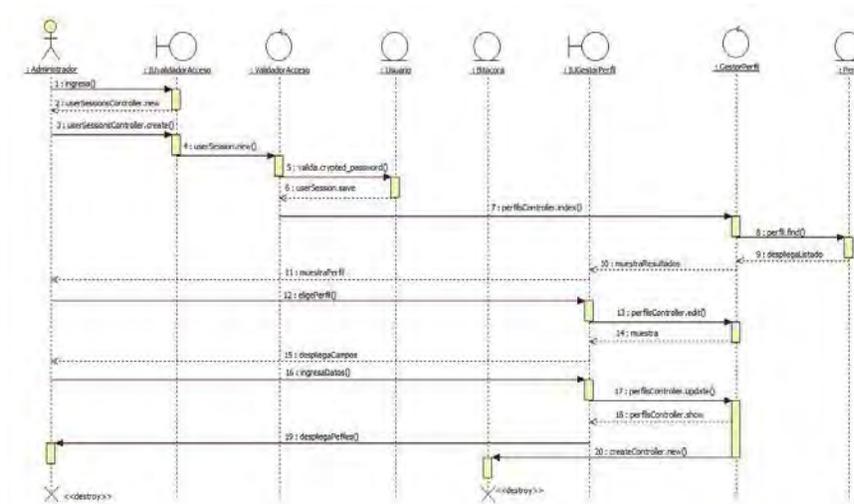


Figura 46. Diagrama de secuencia modificar perfil.



- **Borrar perfil**

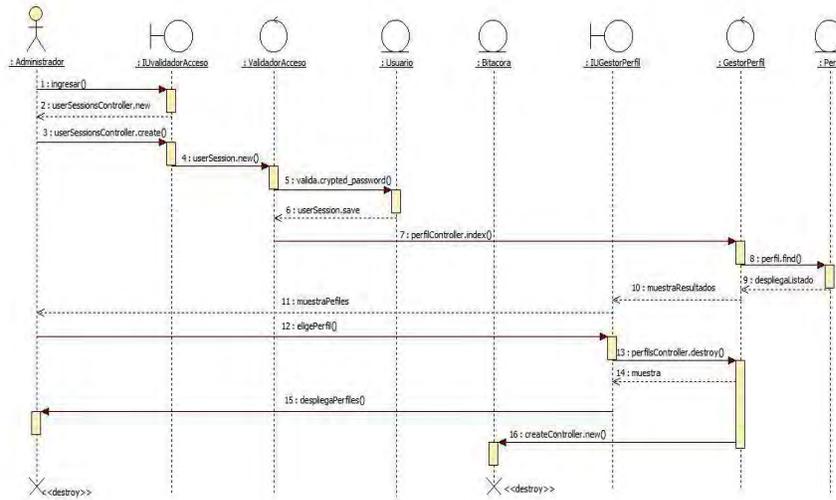


Figura 47. Diagrama de secuencia borrar perfil.



Categoría

- Crear categoría

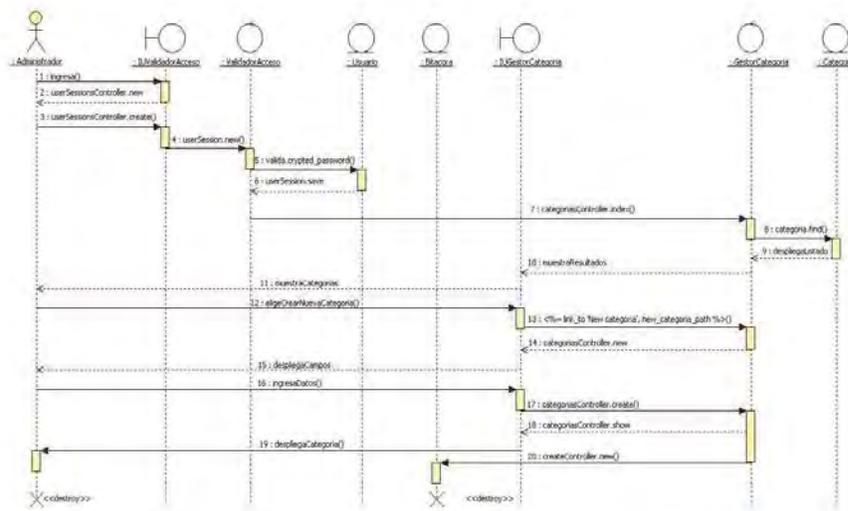


Figura 48. Diagrama de secuencia crear categoría.



- **Modificar categoría**

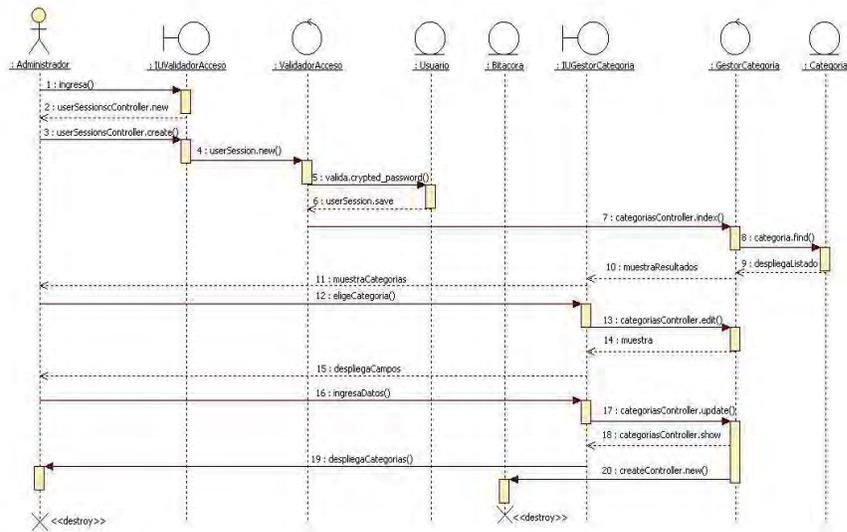


Figura 49. Diagrama de secuencia modificar categoría.



- Borrar categoría

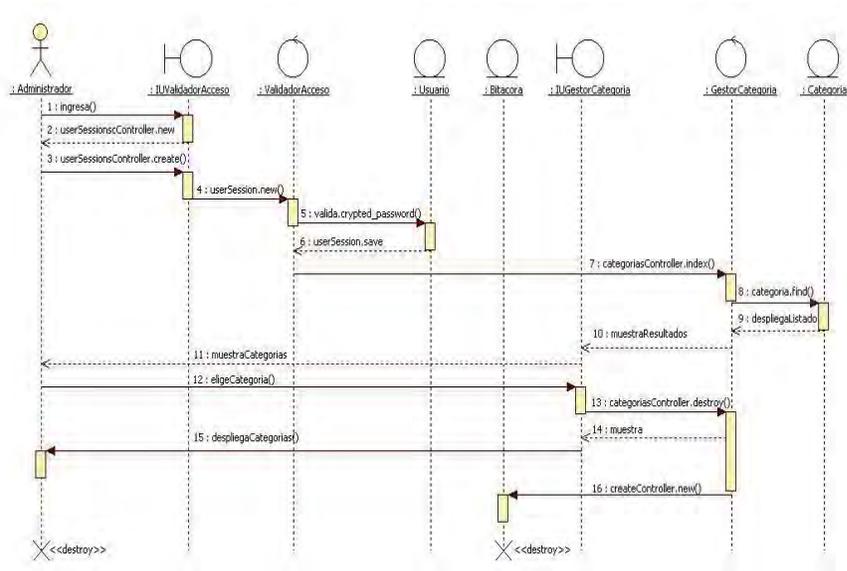


Figura 50. Diagrama de secuencia borrar categoría.



Buscar video administración

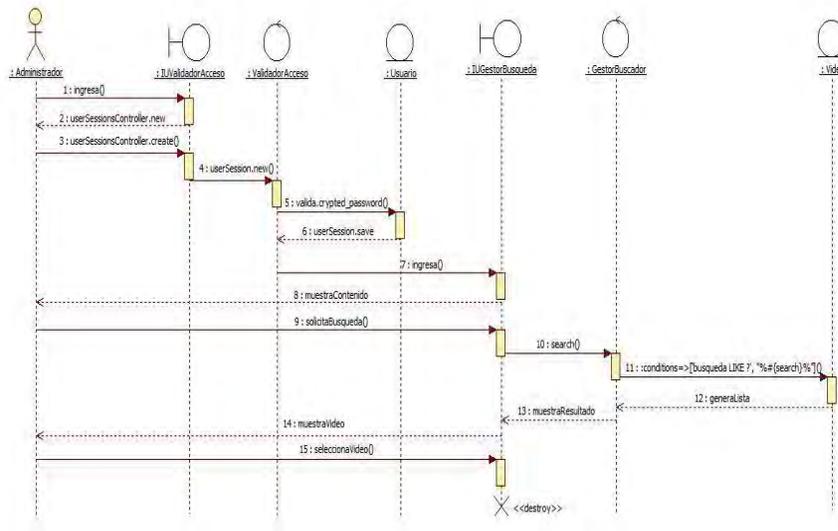


Figura 51. Diagrama de secuencia buscar video administración.



Buscar video web

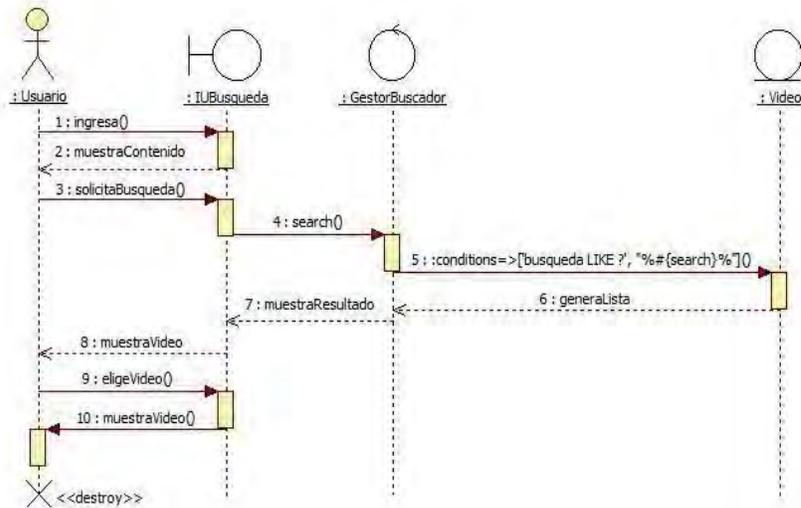


Figura 52. Diagrama de secuencia buscar video web.



Diagramas de actividades

Ingresar videoteca



Figura 53. Diagrama de actividades ingresar videoteca.



Administrar video

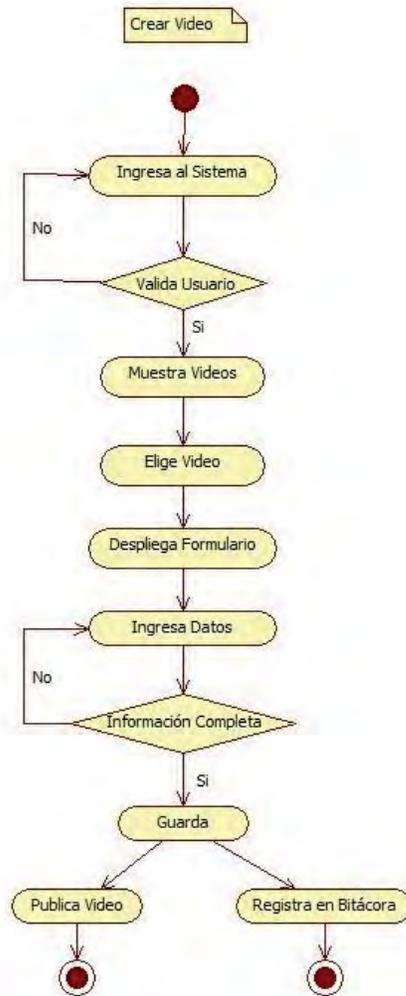


Figura 54. Diagrama de actividad crear video.

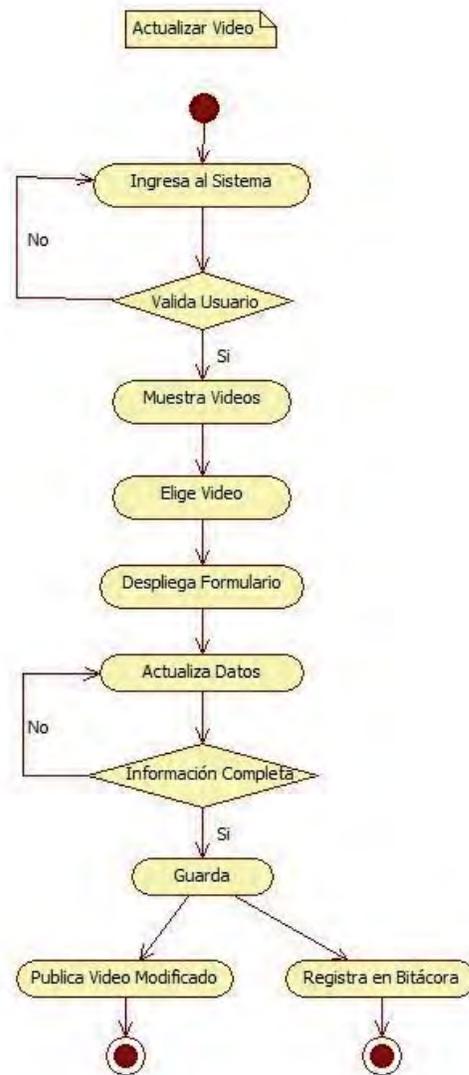


Figura 55. Diagrama de actividades actualizar video.



Figura 56. Diagrama de actividades borrar video.



Administrar catálogos

Usuario



Figura 57. Diagrama de actividades crear usuario.



Figura 58. Diagrama de actividades modificar usuario.



Figura 59. Diagrama de actividades baja de usuario.



Autor

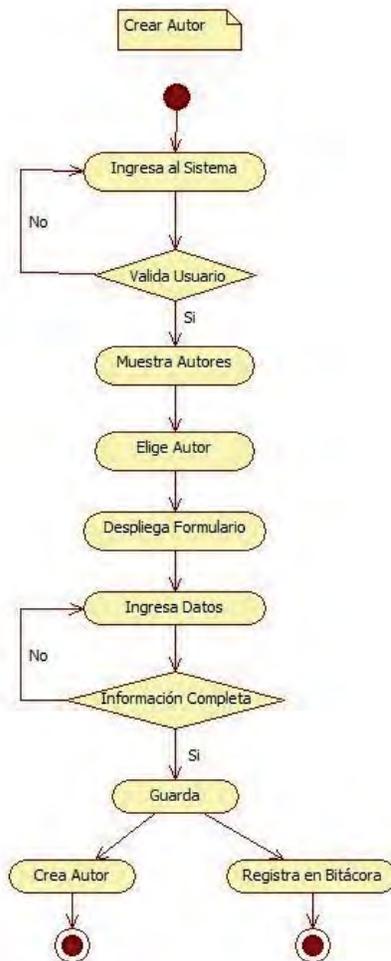


Figura 60. Diagrama de actividades crear autor.

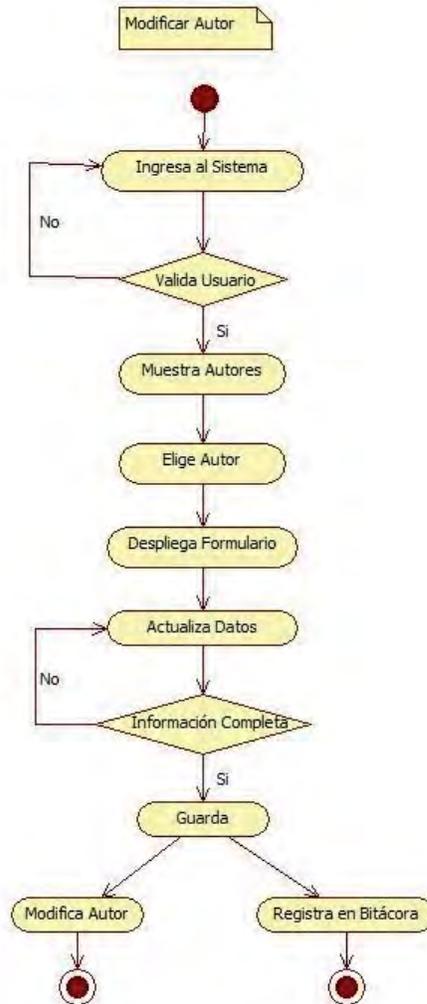


Figura 61. Diagrama de actividades modificar autor.



Figura 62. Diagrama de actividades borrar autor.



Perfil

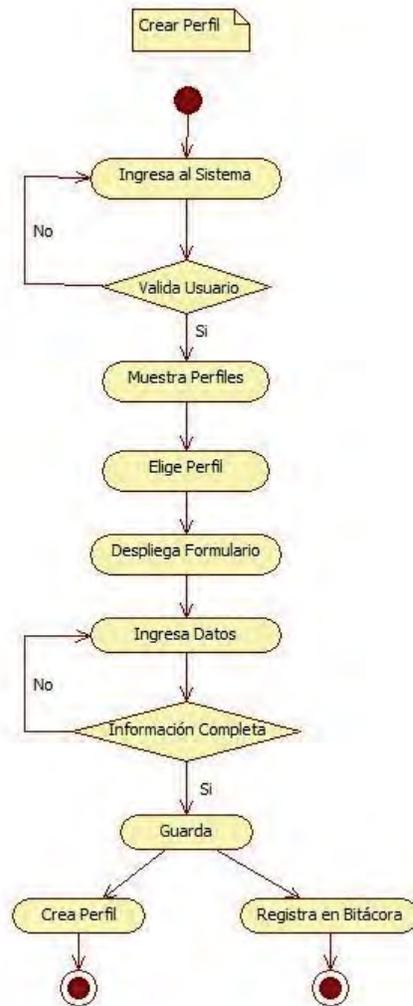


Figura 63. Diagrama de actividades crear perfil.

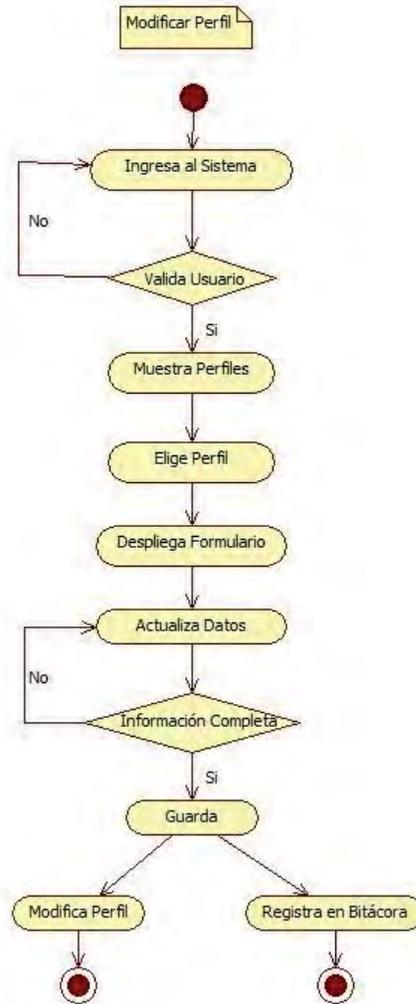


Figura 64. Diagrama de actividades modificar perfil.



Figura 65. Diagrama de actividades borrar perfil.



Categoría

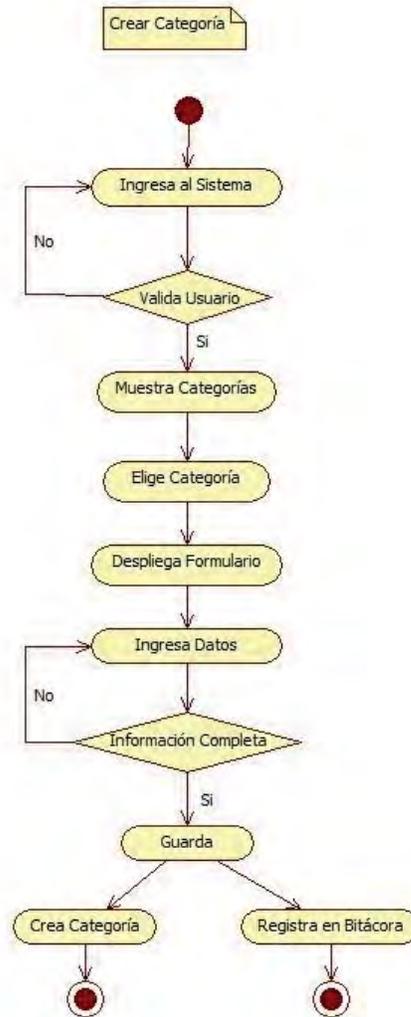


Figura 66. Diagrama de actividades crear categorías.

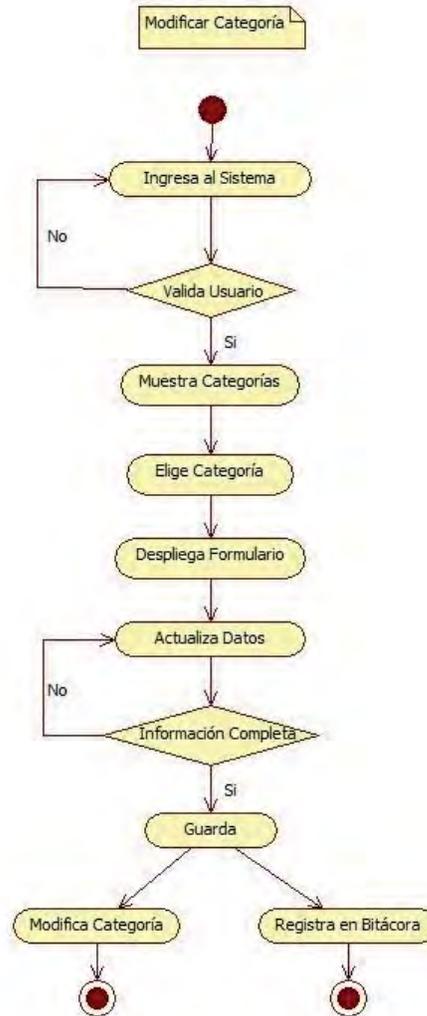


Figura 67. Diagrama de actividades modificar categorías.



Figura 68. Diagrama de actividades borrar categoría.



Búsqueda administración

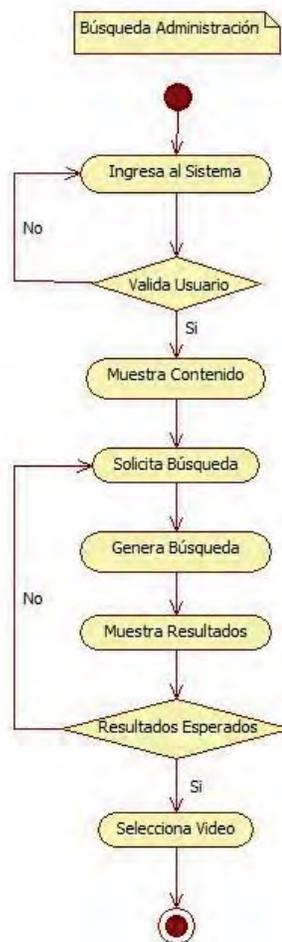


Figura 69. Diagrama de actividades búsqueda administración.



Búsqueda web



Figura 70. Diagrama de actividades búsqueda web.



Universidad Nacional Autónoma De México
Facultad de Contaduría y Administración



CAPITULO 4.

IMPLEMENTACIÓN



Descripción y uso de *Ruby on Rails* en el proyecto.

Ruby on Rails no es sólo un lenguaje, es un conjunto de un lenguaje (*Ruby*) sobre el que trabaja un *framework* (*Rails*.)

Ruby on Rails nos da muchas ventajas si lo comparamos con otros *frameworks*. Una de las principales es que es *open source* y otra que posee toda una comunidad de apoyo.

Rails está basado en dos principios principales de desarrollo:

- DRY (*Don't Repeat Yourself*): “no te repitas a ti mismo”, con esto podemos tener un formulario, y llamarlo las veces que queramos y desde donde queramos, simplemente con una línea código, o tal vez tener una tabla en nuestra base de datos, y manipular a los registros como un objeto y a sus campos como un atributo, sin necesidad de que declaremos nada.
- Convención sobre configuración:

```
class Video < ActiveRecord::Base  
end
```

Con esa declaración de una clase, mapeamos a una tabla en nuestra base de datos, dicho de otra manera *Rails* buscara una tabla llamada 'videos' en nuestra base de datos.

Para la implementación (generación de código) se ha utilizado *Ruby on Rails*, también conocido como **RoR** o *Rails* es un *framework* de aplicaciones web de código abierto escrito en el lenguaje de programación *Ruby*, Los *frameworks* son parte fundamental en la ingeniería del software, ya que promueven la reutilización del código siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros *frameworks* y con un mínimo de configuración.

El lenguaje de programación *Ruby* permite la metaprogramación que esta es el uso de programas que manipulan a otros para el desarrollo de nuestro sistema de la cual *Rails* hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible.

Rails se distribuye a través de *RubyGems*, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones *Ruby*. *Rails* usa convenciones, paquetes de programación integrados y código predefinido, diseñado para completar y usar inmediatamente sin necesidad de configuración.



A diferencia de otros ambientes de programación, como aquellos basados en *Java* que requieren usar varios *frameworks*, los cuales deben ser configurados para que funcionen entre sí, para obtener todas las capacidades deseadas.

Patrón MVC

La aplicación se divide en tres partes:

- Modelo: responsable de mantener los datos de la aplicación.
- Vista: se encarga de la interfaz y presentar la información al usuario.
- Controlador: hace cálculos y une todas las piezas. Recibe eventos del exterior, interactúa con el modelo y actualiza la información de las vistas.

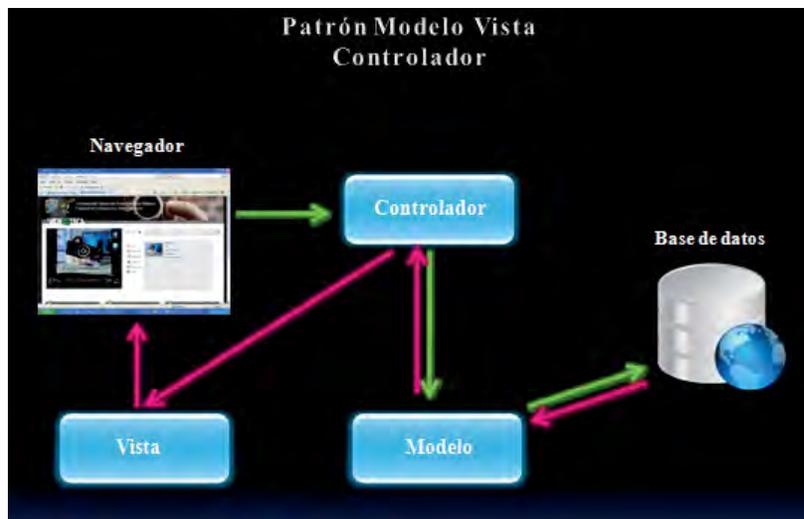


Figura 71. Patrón MVC basado en, http://2.bp.blogspot.com/_IJSFA7nI84U/SAV8oOwuubI/AAAAAAAAAPc/kklas9BZn8E/s320/MVC.png [En línea] [05/08/10]

Para generar el directorio base de nuestra aplicación *Rails* es tan sencillo como teclear el siguiente comando:

```
rails -d mysql videoteca
```



Con el parámetro `-d mysql` indicamos que base de datos vamos a utilizar, el segundo parámetro será el nombre de nuestra aplicación.

Ahora nuestra aplicación llamada videoteca es manejada bajo la arquitectura MVC, como ejemplo de MVC en una aplicación web, podríamos decir que la **vista** es la página HTML y el código que provee de datos dinámicos a la página, el **controlador** es el Sistema de Gestión de Base de Datos y el **modelo** es el modelo de datos.

Modelo

El modelo (*Módulo: ActiveRecord*): una clase representa una tabla, se descubren automáticamente los campos, se pueden declarar relaciones con otros modelos/tablas, se puede personalizar y añadir métodos

Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos

En *Rails*, los modelos están basados en *ActiveRecord*. Lo más importante sobre los modelos de una aplicación *Rails* es que te dan la información correcta cuando el controlador se la pide. Ejemplo de una clase definida dentro del modelo:

```
class Video < ActiveRecord::Base
  belongs_to :autor

  validates_presence_of :titulo, :message => "No puede
  estar vacio"

  def nombreMetodo
    #definición de instrucciones
  end
end
```

Vista

La vista (*Módulo: ActionView*): HTML con *Ruby* embebido, una por cada acción de cada controlador, dependen de los controladores, plantillas en formato *Embedded Ruby*, y HTML repetitivo, a funciones externas (*helpers*).

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. En *Rails* la vista se genera usando RHTML (HTML con código *Ruby*) o RXML (XML con código *Ruby*). El controlador elige que vista usar, y hace disponible los datos que necesita.



Ejemplo de una vista simple:

```
<h1>Lista de Videos</h1>
<br>
<% form_tag videos_path, :method => 'get' do %>
  <p>
    <%= text_field_tag :criteria, params[:search] %>
    <%= submit_tag "Buscar", :titulo => nil %>
  </p>
<% end %>
<table>
  <tr>
    <th>Titulo</th>
  </tr>
  <% @videos.each do |video| %>
    <tr>
      <td><%=h video.titulo %></td>
      <td><%= link_to 'Mostrar', video %></td>
      <td><%= link_to 'Editar', edit_video_path(video) %></td>
      <td><%= link_to 'Eliminar', video, :confirm => "¿Estas
seguro de querer eliminar el video #{video.titulo}?", :method
=> :delete %></td>
    </tr>
  <% end %>
</table>
<br />
<div class='digg_pagination'>
<%= will_paginate @videos%>
</div>
```

Controlador

El controlador (*Módulo: ActionPack*): cada controlador es una clase de *Ruby*, cada método es una acción, y tiene andamios (*scaffolds*) para avanzar más rápido.

Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. En una aplicación web el controlador recibe la petición del usuario, interactúa con el modelo para procesar los datos y hace disponible esos datos a la vista. Los controladores son la única parte del MVC que deben ser definidos. El controlador puede procesar los datos y mostrarlos, es autosuficiente.



Ejemplo de una clase definida en el controlador.

```
class VideosController < ApplicationController
  def index
    if request.format.html?
      @videos = Video.search(params[:criteria], params[:page])
    else
      @videos = Video.find(:all)
    end
    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @videos }
      format.rss { render :layout => false }
    end
  end
end
```

Migraciones

La migración dentro de *Rails* es una parte importante ya que con ella podemos generar cualquier comportamiento para nuestro modelo de datos.

El siguiente comando, es básico para nuestra aplicación en *Rails* ya que con el generamos la migración para la base de datos:

```
rake db:migrate
```

Ejemplo de una clase para la migración de una base de datos:

```
class CreateVideos < ActiveRecord::Migration
  def self.up
    create_table :videos do |t|
      t.string :titulo, :limit => 100
      t.references :category
    end
  end
  def self.down
    drop_table :videos
  end
end
```



Pruebas

Una prueba del *software* se puede definir como una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, registrándose los resultados obtenidos. Se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar fallos en el *software* y corregirlos hasta llegar a obtener fiabilidad, o un cierto nivel de confianza en el *software* probado.

Características de una buena prueba de *software*:

- Una buena prueba ha de tener una alta probabilidad de encontrar un fallo. Para alcanzar este objetivo el responsable de la prueba debe entender el *software* e intentar desarrollar una imagen mental de cómo podría fallar.
- Una buena prueba no debe ser redundante. El tiempo y los recursos son limitados, así que todas las pruebas deberían tener un propósito diferente.
- Una buena prueba no debería ser ni demasiado sencilla ni demasiado compleja, pero si se quieren combinar varias pruebas a la vez se pueden enmascarar errores, por lo que en general, cada prueba debería realizarse separadamente.

El tipo de pruebas generalmente usadas para probar *software* son:

- Pruebas Unitarias. Comienzan con la prueba de cada módulo.
- Pruebas de Integración. A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces.
- Prueba del Sistema. El *software* ensamblado totalmente con cualquier componente hardware que requiere se prueba para comprobar que se cumplen los requisitos funcionales.
- Pruebas de Aceptación. El cliente comprueba que el *software* funciona según sus expectativas.

Las pruebas proporcionan distintos criterios para generar fallos en los programas. Estas técnicas se agrupan en:

- Técnicas de caja blanca ó estructural, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
- Técnicas de caja negra ó funcional, que realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa.



No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

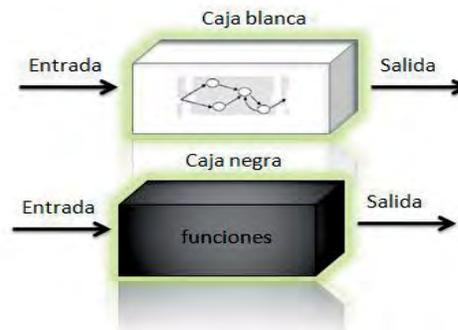


Figura 72. Técnicas de prueba de caja blanca y caja negra basado en <http://www.lsi.us.es/docencia/get.php?id=361> [En línea][05/08/10]

Para este nuevo sistema se aplicaron dos tipos de pruebas

- Prueba de complejidad ciclomática (caja blanca)
- Prueba de sistema (caja negra)

Prueba de complejidad ciclomática

Es una métrica del software propuesta por *Thomas McCabe* que proporciona una medida cuantitativa de la complejidad lógica de un programa. El valor de la complejidad ciclomática define el número de caminos independientes de dicho programa, y por lo tanto, el número de casos de prueba a realizar para asegurar que se ejecuta cada camino del programa.

Los pasos a realizar para aplicar esta técnica son:

1. Representar el programa en un grafo de flujo
2. Calcular la complejidad ciclomática
3. Determinar el conjunto básico de caminos independientes
4. Derivar los casos de prueba que fuerzan la ejecución de cada camino

A continuación, se detallan cada uno de estos pasos.

1. Representar el programa en un grafo de flujo



El grafo de flujo se utiliza para representar flujo de control lógico de un programa. Para ello se utilizan los tres elementos siguientes:

Nodos: representan cero, una o varias sentencias en secuencia. Cada nodo comprende como máximo una sentencia de decisión.

Aristas: líneas que unen dos nodos.

Regiones: áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más.

Nodos predicado: cuando en una condición aparecen uno o más operadores lógicos (*AND*, *OR*, *XOR*, etc) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado.

Cada construcción lógica de un programa tiene una representación.

2. Calcular la complejidad ciclomática

- El número de regiones del grafo coincide con la complejidad ciclomática, $V(G)$.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como $V(G) = \text{Aristas} - \text{Nodos} + 2$
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como $V(G) = \text{Nodos Predicado} + 1$

3. Determinar el conjunto básico de caminos independientes

Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una condición, respecto a los caminos existentes. En términos del diagrama de flujo, un camino independiente está constituido por lo menos por una arista que no haya sido recorrida anteriormente a la definición del camino. En la identificación de los distintos caminos de un programa para probar se debe tener en cuenta que cada nuevo camino debe tener el mínimo número de sentencias nuevas o condiciones nuevas respecto a los que ya existen. De esta manera se intenta que el proceso de depuración sea más sencillo.

El conjunto de caminos independientes de un grafo no es único. No obstante, a continuación, se muestran algunas indicaciones para identificar dichos caminos:

- Elegir un camino principal que represente una función válida que no sea un tratamiento de error. Debe intentar elegirse el camino que atravesase el máximo número de decisiones en el grafo.
- Identificar el segundo camino mediante la localización de la primera decisión en el camino de la línea básica alternando su resultado mientras se mantiene el máximo número de decisiones originales del camino inicial.



- Identificar un tercer camino, colocando la primera decisión en su valor original a la vez que se altera la segunda decisión del camino básico, mientras se intenta mantener el resto de decisiones originales.
- Continuar el proceso hasta haber conseguido tratar todas las decisiones, intentando mantener como en su origen el resto de ellas.

Este método permite obtener $V(G)$ caminos independientes cubriendo el criterio de cobertura de decisión y sentencia.

4. El último paso es construir los casos de prueba que fuerzan la ejecución de cada camino por medio de la matriz.

Código seleccionado a probar:

```
def self.search(search, page)
  if search.nil?
    conditions = nil
  else
    conditions = ['busqueda LIKE ?', "%#{search}%"]
  end
  paginate :per_page => 1, :page => page,
    :conditions => conditions,
    :order => 'fecha_carga'
end
def cambia_estado
  if self.estado == "C"
    else self.estado = "A"
  end
end
end
```



Grafo de flujo

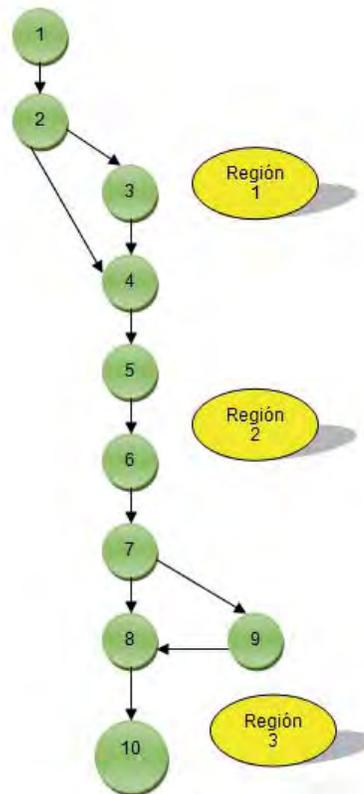


Figura 73. Grafo de complejidad ciclomática.

Cálculo de la complejidad del flujo anterior

$$V(G) = \text{Regiones} = 3$$

$$V(G) = \text{Aristas} - \text{Nodos} + 2 = 11 - 10 + 2 = 3$$

$$V(G) = \text{Nodos Predicado} + 1 = 2 + 1 = 3$$



Caminos

- 1.- 1,2,4,5,6,7,8,10
- 2.- 1,2,3,4,5,6,7,9,8,10
- 3.- 1,2,3,4,5,6,7,8,10
- 4.- 1,2,4,5,6,7,9,8,10

Estos cuatro caminos constituyen el camino básico para el grafo de flujo correspondiente.

Matriz de complejidad ciclomática	1	2	3	4	5	6	7	8	9	10	Resultados
1		1									1 - 1 = 0
2			1	1							2 - 1 = 1
3				1							1 - 1 = 0
4					1						1 - 1 = 0
5						1					1 - 1 = 0
6							1				1 - 1 = 0
7								1	1		2 - 1 = 1
8										1	1 - 1 = 0
9								1			1 - 1 = 0
10											0
											2 + 1 = 3

Tabla 10. Matriz de resultados del grafo.

Cumple con la complejidad ciclomática

Prueba de sistema

Este tipo de pruebas tiene como propósito de verificar que se han integrado adecuadamente todos los elementos del sistema (*hardware*, otro *software*, etc.) y que realizan las funciones adecuadas.

Concretamente se debe comprobar que:

- Se cumplen los requisitos funcionales establecidos.
- El funcionamiento y rendimiento de las interfaces de usuario.
- La adecuación de la documentación de usuario.



El nuevo sistema realiza un requerimiento funcional que es el cambio de estados del video y la actualización de la información inicial.



Figura 74. Video cargado.

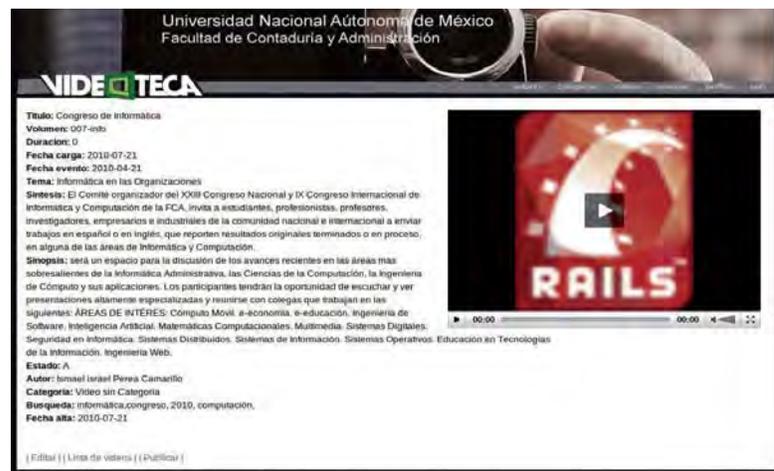


Figura 75. Cambio de estado de video y actualización de la información.



El nuevo sistema cumple con el funcionamiento y rendimiento de las interfaces de usuario por que al momento de ingresar existe una interfaz que permite el correcto inicio de sesión seguido del direccionamiento a la interfaz de administración del sistema.



Figura 76. Interfaz de inicio de sesión.



Figura 77. Interfaz de gestión del sistema.

El nuevo sistema cumple con la adecuada documentación para el usuario por medio del manual (Anexo).



CONCLUSIONES

Al desarrollar un nuevo sistema de Videoteca para la Facultad de Contaduría y Administración, con los parámetros de una metodología de desarrollo de sistemas orientada a objetos se logró documentar todo el ciclo de vida y así tener una referencia para poder actualizarlo cuando sea necesario. Se siguieron las especificaciones de la metodología de nombre ICONIX, ya que es una nueva convergencia dentro del paradigma de desarrollo de sistemas, que por ser una metodología híbrida nos permitió realizar el sistema de manera suelta como las metodologías ágiles y con la robustez (los entregables y documentación) de las metodologías tradicionales.

Con este nuevo desarrollo, la página facultativa de la videoteca cuenta ahora con los lineamientos principales del CATIC, dando una imagen más homogénea e institucional.

Asimismo la utilización de técnicas actualizadas de programación web, nos ayudó a poner al sistema dentro de estándares de uso internacional como los especificados por el consorcio del W3C, además de facilitarnos la tarea de mantener una difusión constante de todas las actualizaciones en contenido que se van generando en la videoteca mediante la integración de las RSS y la funcionalidad de descarga de los videos a dispositivos personales aumentando la difusión e interés de los videos.

Finalmente, con todo esto conseguiremos que la comunidad de la Facultad de Contaduría y Administración tenga un material más para complementar su aprendizaje en las distintas áreas de conocimiento de las tres licenciaturas que se imparten. Además se podrá dar a conocer todo el acervo en video con el que se cuenta dentro de la facultad. Al mismo tiempo se difundirá a otras instituciones.



GLOSARIO

Actor:

Un actor es una entidad externa (fuera del sistema) que interacciona con el sistema participando en un caso de uso. Los actores pueden ser gente real, otros sistemas o eventos externos.

ActiveRecord:

Es un objeto que representa una fila en una tabla de una base de datos, encapsula su acceso a la base de datos y añade lógica de negocio. Es la aproximación más obvia, poniendo el acceso a la base de datos en el propio objeto de negocio.

Borrar usuario:

Referido al caso de uso administrar catálogos, en la parte administración de usuario el borrado de alguna cuenta de usuario se referirá a la desactivación de la cuenta para contemplaciones pertinentes a los registros de la bitácora.

Boundary:

Parte del sistema (*System boundary*) es representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman, representa las interacciones con los usuarios y/o actores que intervienen en el sistema.

Caso de uso:

Especifica el comportamiento de un sistema o una parte del mismo, y es una descripción de un conjunto de secuencias de acciones, donde cada secuencia representa la interacción de los elementos externos del sistema (sus actores) con el propio sistema. Representa un requerimiento funcional del sistema.

Clase:

Una clase está representada por un rectángulo que dispone de tres apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos; cada clase debe tener un nombre único, que las diferencie de las otras.

Control:

Contienen comportamiento que no pertenece naturalmente ni a objetos entidad ni de interfaz. Son normalmente objetos transitorios, como ser un controlador de reportes.

Diagrama de actividades:

Describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente contienen actividades.



Diagrama de colaboración:

Muestra las interacciones que ocurren entre los objetos que participan en una situación determinada.

Diagrama de caso de uso:

Describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Diagrama de clases:

Muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras.

Diagrama entidad – relación:

Muestran el diseño conceptual de las aplicaciones de bases de datos. Representan varias entidades en el sistema de información, las relaciones y restricciones existentes entre ellas.

Diagrama de robustez:

Es una variante del diagrama de colaboración. Este diagrama nos va a permitir identificar los tres tipos de objetos que dan lugar al diagrama de diseño: *boundary*, *control*, *entity*.

Diagrama de secuencia:

Muestran el intercambio de mensajes (la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

Entity:

Entidades. Pueden ser clases, objetos, usuarios o sistemas de comportamientos.

Framework:

Término usado en programación orientada a objetos para definir un conjunto de clases que definen un diseño abstracto para solucionar un conjunto de problemas relacionados. Los *Frameworks* ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez.

Helpers:

Son juegos de funciones, ordenados por temática en diferentes paquetes, que nos ayudan en la realización de tareas habituales en las aplicaciones web. Existen *helpers* para muchas cosas distintas, como para trabajo con *arrays*, fechas, *cookies*, *emails*, URLs, formularios, etc.



Kernel:

El *kernel* ó núcleo de *linux* se puede definir como el corazón de este sistema operativo. Es el encargado de que el software y el hardware de la computadora o servidor puedan trabajar juntos.

Link:

Es una palabra, una frase o un gráfico de un hipertexto que conecta con otra información en la *web* o internamente, los enlaces (o vínculos) de una página *web* conectan con otras páginas principales de otra *web*.

Mecatrón:

Es un robot virtual que extrae los títulos de los DVD, cada título se subdivide en tres diferentes archivos .flv que se visualizara en línea .mov en dos calidades, alta y media para poder descargarlos.

Metaprogramación

La metaprogramación es codificar programas que a su vez codifican otros programas.

Password:

Palabra por lo general secreta que se emplea como herramienta de seguridad para identificar a usuarios de una aplicación, archivo, o red. Su propósito es prevenir accesos no autorizados a información personal o confidencial.

Realize:

Es estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML®. Una restricción identifica un comportamiento forzado de una clase o relación, es decir mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

Scaffold:

Scaffolding es un método para contruir aplicaciones basadas en bases de datos, esta técnica está soportada por algunos *frameworks* del tipo MVC en el cuál el programador escribe una especificación que describe cómo debe ser usada la base de datos. el código que la aplicación usará para crear, leer, actualizar y eliminar registros de la base de datos, esto es conocido como CRUD (*create, read, update, delete*).El *Scaffolding* fue popularizado por el *framework Ruby on Rails* es utilizado por otros *frameworks*.

Sprint (Scrum):

Un Sprint en Scrum es el término que denomina a una iteración que está acotada en el tiempo, usualmente entre 2 y 4 semanas. Cada sprint es un incremento ejecutable que se muestra al cliente.



SIGLARIO

CATIC:

Pertenece a las siglas del Consejo Asesor en Tecnologías de Información y Comunicación, que se encarga de establecer los lineamientos que deben de seguirse para la creación de páginas *web* pertenecientes a la UNAM. (Antes llamado CAC).

.Flv:

Es la terminación de un video en un formato contenedor propietario usado para transmitir video por Internet usando *Adobe Flash Player*.

FURPS+:

Es un acrónimo que representa un modelo para clasificar los atributos de calidad del *software* (funcionales y no funcionales requerimientos):

- **Functionality.**- Conjunto de características, capacidades, la Generalidad, de Seguridad
- **Usability.**- Los factores humanos, Estética, Coherencia, Documentación.
- **Reliability.**- La frecuencia y severidad de la insuficiencia, de valorización, la previsibilidad, exactitud, el tiempo medio hasta el fallo.
- **Performance.**- Tiempos de respuesta, productividad, precisión, disponibilidad de recuperación de fallo y grado de previsión.
- **Supportability.**- Adaptabilidad, facilidad de mantenimiento, internacionalización, facilidad de configuración.

El "+" en **FURPS+** también nos ayuda en:

- Implementación.- Restricciones impuestas para la interacción con sistemas
- Interfaz.- Restricciones impuestas para la interacción con sistemas externos (no es GUI)
- Operaciones.- Gestión del sistema, pautas administrativas, puestas en marcha
- Empaquetamiento.- Forma de distribución
- Legales.- Licencias, derechos de autor etc.

HTML:

Pertenece a las siglas de *HyperText Markup Language* (Lenguaje de Marcado de Hipertexto), es un sistema de código o marcas que se utiliza para crear la estructura de un sitio web, el cual será leído y mostrado por los navegadores según la disposición desarrollada por el autor.



.Mov:

Es la terminación de un archivo de video estándar de la compañía APPLE®.

OMG:

Por sus siglas en ingles *Object Management Group* que es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML®. Es una organización sin ánimo de lucro que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones para las mismas

RAID:

Originalmente proviene de *Redundant Array of Inexpensive Disks* - conjunto redundante de discos baratos. Actualmente de *Redundant Array of Independent Disks* - conjunto redundante de discos independientes). Sistema de almacenamiento de información que utiliza múltiples discos duros en donde se distribuyen y/o duplican datos. En su concepto más simple, RAID utiliza múltiples discos como si se tratara de una unidad lógica sola. El sistema operativo y el usuario ven un solo disco, pero en realidad la información es almacenada en todos los discos.

RMC:

Por sus siglas en inglés *Rational Method Composer*, es una plataforma de procesos de gestión flexible con herramientas más completa de la industria y la más rica colección de procesos para ayudar a las empresas implementar procesos efectivos para el *software* y proyectos exitosos de TI.

RSS:

Son las siglas de *RDF Site Summary or Rich Site*, es un formato para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos.

RUP:

Por sus siglas en inglés *Rational Unified Process* es un proceso iterativo de desarrollo de programas, marco creado por el *Rational Software Corporation*, una división de IBM® desde 2003. RUP® no es un solo proceso normativo concreto, sino más bien un proceso de adaptación, destinado a ser adaptado por los organismos de desarrollo y los equipos de proyectos de software que seleccionará a los elementos del proceso que son apropiados para sus necesidades.

RHTML:

Un archivo RHTML es una combinación de HTML y *ruby*. El código *ruby* vive dentro de los delimitadores `<% %>`, o `<%= %>` si desea el resultado de una función desplegado en el HTML.



SCRUM:

Es un método ágil para el desarrollo y gestión de *software* basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de *software*.

TI:

Son las siglas pertenecientes a Tecnología de la Información es un término general que describe cualquier tecnología que ayuda a producir, manipular, almacenar, comunicar, y/o esparcir información.

UML:

Por sus siglas en inglés *Unified Modeling Language*, es un lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML® ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

XP:

La Programación Extrema (PX), mejor conocida por su nombre en inglés *Extreme Programming* (XP), es una de las llamadas Metodologías Ágiles de desarrollo de *software* más exitosas de los tiempos recientes.

BIBLIOGRAFÍA

LARMAN Craig UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado, PRENTICE HALL, Madrid 2003, 583p, ISBN 842-0534-38

LAWRENCE Pfleedge Shari, Ingeniería de Software Teoría y Práctica, Pearson Education, Buenos Aires 2002, 792 p, ISBN 987-9460-71-5

THOMAS Dave, Heinemeier David, Agile Web Development with Rails Second Edition, Hansson, Pragmatic Bookshelf, EEUU 2009, 685 p, ISBN 10:0-0-9776166-3-0, ISBN 13:0-0-9776166-3-3



MESOGRAFÍA

Actor. <<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

ActiveRecord. <<http://es.debugmodeon.com/articulo/active-record>> [En línea] [27/06/10]

Boundary. <<http://usuarios.multimania.es/ooopere/uml.htm>> [En línea] [27/06/10]

Caso de uso. <<http://www.dcc.uchile.cl/~clgutier/casosUso.doc>> [En línea] [27/06/10]

CATIC. <<http://www.cac.unam.mx/>> [En línea] [27/06/10]

Clase. <<http://usuarios.multimania.es/ooopere/uml.htm>> [En línea] [27/06/10]

Control. <<http://www.chaco.gov.ar/utn/disenodesistemas/apuntes/oo/ApunteUML.pdf>> [En línea] [27/06/10]

Diagrama de actividades.
<<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

Diagrama de colaboración.
<<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

Diagrama de caso de uso.
<<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

Diagrama de clases. <<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

Diagrama entidad – relación. <<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

Diagrama de robustez.
<<http://www.humbertocervantes.net/cursos/ayd/practicas/practica5/practica.html>> [En línea] [27/06/10]

Diagrama de secuencia. <<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>> [En línea] [27/06/10]

Entity. <<http://cnx.org/content/m11658/latest/>> [En línea] [27/06/10]



.Flv. <http://es.wikipedia.org/wiki/Flash_Video> [En línea] [27/06/10]

Framework. <<http://www.dccia.ua.es/dccia/inf/ asignaturas/RG/pdf/colecciones-java.pdf>> [En línea] [27/06/10]

FURPS+. <<http://sophia.javeriana.edu.co/~lcdiaz/ADOO2008-/requerimientosFURPS.pdf>> [En línea] [27/06/10]

Helpers <<http://www.desarrolloweb.com/articulos/helpers-codeigniter.html>> [En línea] [27/06/10]

HTML. <www.redaccioncontenidos.com.ar/category/glosario-seo/> [En línea] [27/06/10]

Kernel. <<http://www.linux-es.org/kernel>> [En línea] [27/06/10]

Link.

<http://www.unipamplona.edu.co/unipamplona/hermesoft/portallG/home_28/recursos/01_general/09052008/glosario.jsp> [En línea] [27/06/10]

Metaprogramación.

<http://revistas.concytec.gob.pe/scielo.php?script=sci_arttext&pid=S1810-99932007000100012&lng=es&nrm=iso> En línea] [27/06/10]

.Mov. <<http://es.wikipedia.org/wiki/QuickTime>> [En línea] [27/06/10]

OMG. <http://es.wikipedia.org/wiki/Object_Management_Group> [En línea] [27/06/10]

Password. <www.anetcom.es/servicios/glosario.asp> [En línea] [27/06/10]

RAID. <<http://www.alegsa.com.ar/Dic/raid.php>> [En línea] [27/06/10]

Realize. <<http://usuarios.multimania.es/ooopere/uml.htm>> [En línea] [27/06/10]

RMC. <<http://www.ibm.com/software/awdtools/rmc>> [En línea] [27/06/10]

RSS. <<http://es.wikipedia.org/wiki/RSS>> [En línea] [27/06/10]

RUP. <<http://www-01.ibm.com/software/awdtools/rup/>> [En línea] [27/06/10]

Scaffold <<http://techtastico.com/post/que-es-el-scaffolding-o-scaffold/>> [En línea] [27/06/10]



Sprint <<http://www.dosideas.com/wiki/Sprint>> [En línea] [27/06/10]

UML. <<http://www.uml.org/>> [En línea] [27/06/10]

SCRUM. <<http://www.mountaingoatsoftware.com/topics/scrum>> [En línea] [27/06/10]

TI. <http://es.wikipedia.org/wiki/Tecnolog%C3%ADa_de_la_informaci%C3%B3n> [En línea] [27/06/10]

XP. <<http://www.programacionextrema.org/>> [En línea] [27/06/10]



Universidad Nacional Autónoma De México
Facultad de Contaduría y Administración



ANEXO



Manual de usuario

Ingresar al Sistema

Para ingresar al sistema se debe de autenticarse con el nombre de usuario y contraseña que ya ha sido proporcionada anteriormente.

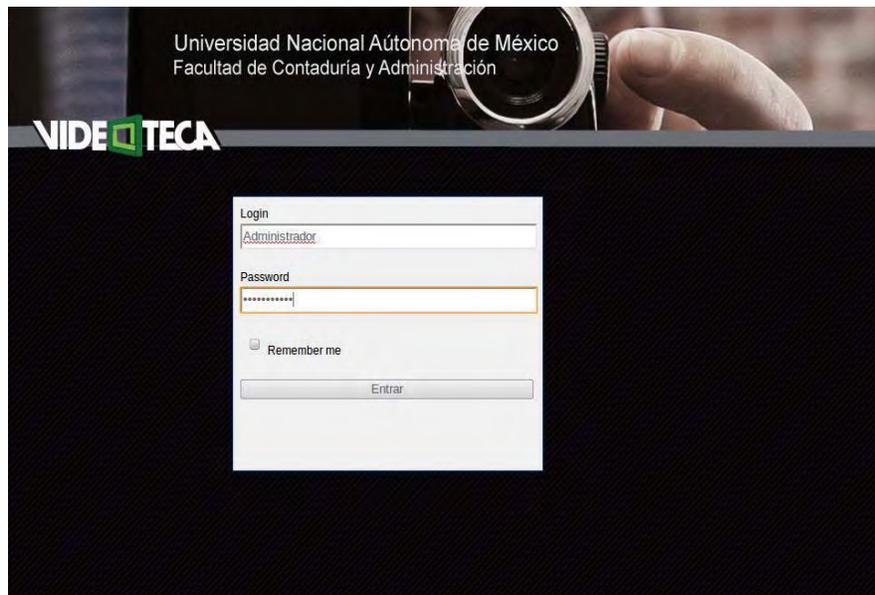


Figura 1. Interfaz de ingreso al sistema.



Área de Gestión

Ya autenticado el sistema desplegará el área de gestión que se compone de los videos cargados y de menú catálogos.

Figura 2. Interfaz de la lista de videos cargados.

El menú de catálogos se compone de:

- Autores
- Categorías
- Videos
- Usuario
- Perfiles
- Salir

Figura 3. Interfaz del menú catálogos.



Menú Autores

Para la gestión de los autores se cuenta con las funcionalidades de creación, edición y eliminado.

Para dar de alta un nuevo Autor se debe de dar click al botón de Nuevo Autor, el cual desplegara un formulario donde pide los siguientes datos a capturar: Nombre, Apellido Paterno, Apellido Materno una vez llenados los campos dar click en crear, si el Autor fue creado satisfactoriamente mostrara un listado con los datos de dicho autor.



Figura 4. Interfaz de captura de nuevo autor.

Para regresar al menú de ese catálogo dar click en Lista de autores



Figura 5. Interfaz de la lista de autores existentes.



Para la edición del autor dar click en Editar el cual desplegara el formulario con los datos que fueron capturados anteriormente para editar la información que se desee se elige el campo a modificar y se realizan los cambios pertinentes, una vez satisfecho se da click en Actualizar y se guardan los cambios realizados.



Figura 6. Interfaz de edición de autor anteriormente capturado.

Para eliminar un usuario basta con solo dar click en Eliminar, el cual mostrara una pantalla de advertencia diciendo “¿Estás seguro de eliminar al autor *****?”.



Figura 7. Interfaz de baja de autor.



Menú Categorías

Para la gestión de categorías se cuenta con las funcionalidades de creación, edición y eliminado.

Para crear una nueva categoría se debe de elegir el botón de Crear especificando el nombre que llevara, así mismo dentro de esta categoría se puede crear una nueva categoría y para esto se elige la Categoría “padre” y se le da click en el botón Crear.



Figura 8. Interfaz de categorías de video.



Figura 9. Interfaz de creación de un nuevo nodo.



Universidad Nacional Autónoma De México

Facultad de Contaduría y Administración



Para eliminar una categoría basta con elegir a la categoría deseada y dar click en borrar, el cual mostrara una pantalla de advertencia diciendo “¿Está seguro de borrar el nodo nuevo?”.

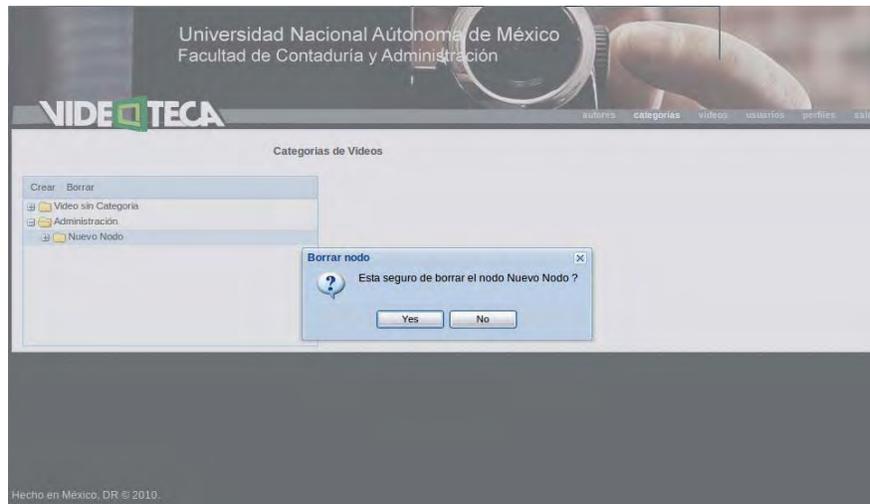


Figura 10. Interfaz de baja del nodo creado.

Menú Video

Para la gestión de los videos se cuenta con las funcionalidades de edición y eliminado. Nota: El Video ya ha sido cargado dentro del sistema por medio del robot automatizado llamado “Mecatron”.



Figura 11. Interfaz mostrando la carga al servidor por medio de mecatrón.



Para publicar un nuevo Video se debe de dar click al botón de Editar, el cual desplegara un formulario dividido en 5 secciones al darle click se desplegaran sus respectivos formularios.

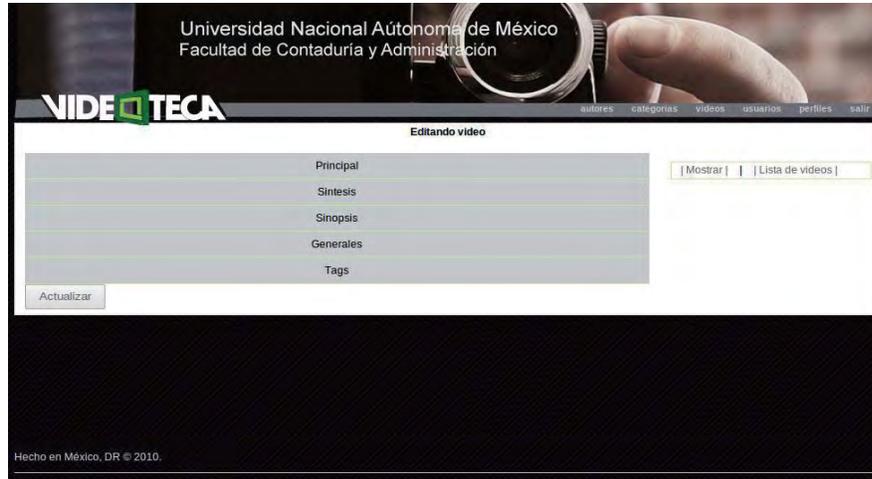


Figura 12. Interfaz de edición de video.

Cada sección despliega un formulario a llenar

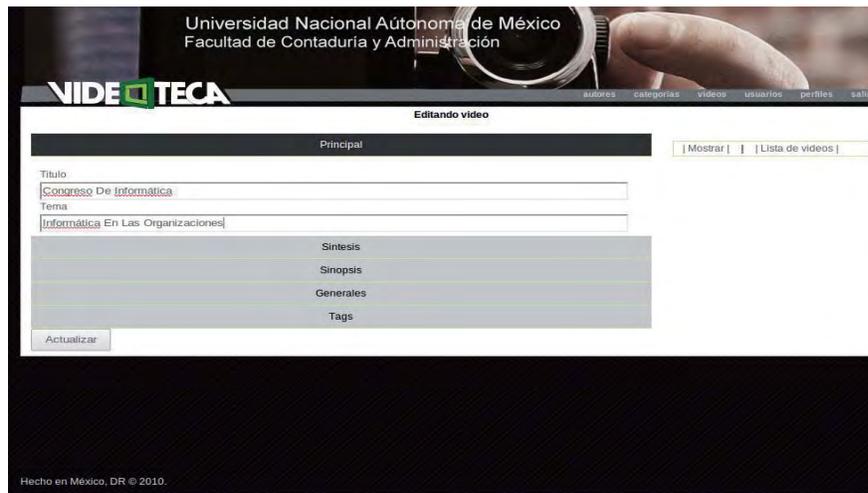


Figura 13. Interfaz de captura del menú principal de video.

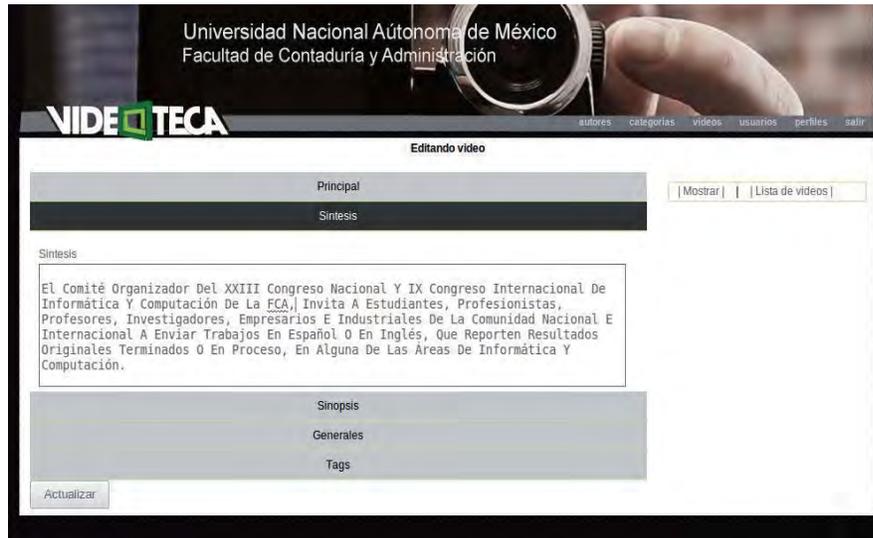


Figura 14. Interfaz de captura del menú síntesis de video.

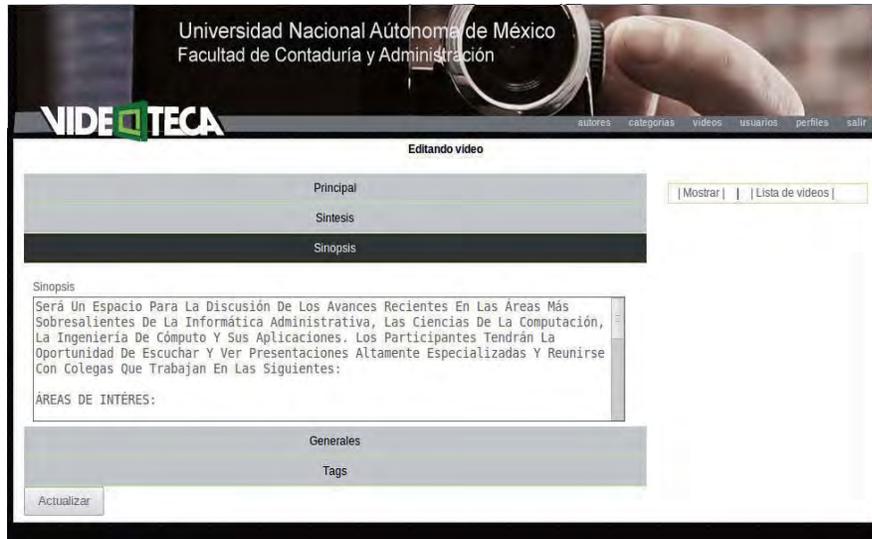


Figura 15. Interfaz de captura del menú sinopsis de video.



Universidad Nacional Autónoma de México
Facultad de Contaduría y Administración

VIDEOTECA autores categorías videos usuarios perfiles salir

Editando video

Principal | Mostrar | Lista de videos |

Síntesis

Sinopsis

Generales

Autor
Ismael Israel Perea Camarillo

Category
Video sin Categoría

Estado
A

Fecha Evento
2010 April 21

Tags

Actualizar

Figura 16. Interfaz de captura del menú general de video.

Universidad Nacional Autónoma de México
Facultad de Contaduría y Administración

VIDEOTECA autores categorías videos usuarios perfiles salir

Editando video

Principal | Mostrar | Lista de videos |

Síntesis

Sinopsis

Generales

Tags

Busqueda
Informática, Congreso, 2010, Computación, |

Actualizar

Figura 17. Interfaz de captura del menú tags de video.



Una vez llenados todos los campos dar click en Actualizar y mostrara el video actualizado con la información anteriormente capturada.



Figura 18. Interfaz actualizada con la información del video capturada.

Para Publicar el video que ya ah sido actualizado y se está conforme con la información solo basta con dar click en el botón Publicar.



Figura 19. Interfaz con la información precisa del video para ser publicada.



Para eliminar un video se elige el video y solo se debe dar click en el botón de Eliminar.



Figura 20. Interfaz de baja de video.

Menú Usuarios

Para la gestión de los usuarios se cuenta con las funcionalidades de creación, edición y eliminado.

Para dar de alta un nuevo usuario se debe de dar click al botón de Nuevo Usuario, el cual desplegara un formulario donde pide los siguientes datos a capturar: Nombre, Apellido Paterno, Apellido Materno.



Figura 21. Interfaz de captura del menú general para un nuevo usuario.



Universidad Nacional Autónoma De México

Facultad de Contaduría y Administración



Al dar click en “Complemento” desplegara los campos faltantes como: Login, E-mail, Password, Confirmación de password y el tipo de perfil una vez llenados los campos dar click en crear, si el Usuario fue creado satisfactoriamente mostrara un listado con los datos de dicho usuario.

Universidad Nacional Autónoma de México
Facultad de Contaduría y Administración

VIDEOTECA

autores categorías videos usuarios perfiles salir

Nuevo usuario

General | Lista de usuarios |

Complemento

Login
E-mail: Editorial
E-mail: Editorial@Correo-Eca-Unam.mx
Password
Confirmación de password:
Perfil
Administrador

Crear

Figura 22. Interfaz de captura del menú complemento para un nuevo usuario.

Para regresar al menú de ese catálogo dar click en Lista de usuarios

Universidad Nacional Autónoma de México
Facultad de Contaduría y Administración

VIDEOTECA

autores categorías videos usuarios perfiles salir

Lista de usuarios

Nombre	Apellido paterno	Apellido materno	Login	E-mail	Borrado	Mostrar	Editar	Eliminar
Administrador	Administrador	Administrador	admin	monitoreo@fca.unam.mx				

| Nuevo Usuario |

Hecho en México, DR © 2010.

Figura 23. Interfaz de la lista de usuarios existentes.



Para la edición del usuario dar click en Editar el cual desplegara el formulario con los datos que fueron capturados anteriormente para editar la información que se desee se elige el campo a modificar y se realizan los cambios pertinentes, una vez satisfecho se da click en Actualizar y se guardan los cambios realizados.



Figura 24. Interfaz de edición del usuario anteriormente capturado.

Para eliminar un usuario basta con solo dar click en Eliminar, el cual mostrara una pantalla de advertencia diciendo “¿Estás seguro de eliminar al usuario *****?”.



Figura 25. Interfaz de desactivación del usuario.



Menú Perfiles

Para la gestión de los perfiles se cuenta con las funcionalidades de creación, edición y eliminado.

Para dar de alta un nuevo perfil se debe de dar click al botón de Nuevo Perfil, el cual desplegara un formulario donde pide los siguientes datos a capturar: Tipo, Descripción una vez llenados los campos dar click en crear, si el perfil fue creado satisfactoriamente mostrara un listado con los datos de dicho perfil.

La imagen muestra la interfaz de usuario de un sistema llamado VIDEOTECA. En la parte superior, hay un encabezado con el logo de la Universidad Nacional Autónoma de México y la Facultad de Contaduría y Administración. Debajo del encabezado, se encuentra el logo de VIDEOTECA y un menú de navegación con los siguientes ítems: autores, categorías, videos, usuarios, perfiles y salir. El título principal de la página es "Nuevo perfil".

El formulario principal está dividido en una pestaña "General". A la derecha de esta pestaña hay un botón que dice "[Lista de perfiles]". El formulario contiene los siguientes campos:

- Tipo: un campo de texto con el valor "Invitado".
- Descripción: un campo de texto vacío.
- Servicio Social: un campo de texto vacío.

Debajo de los campos hay un botón "Crear". En la parte inferior izquierda de la interfaz, se puede leer "Hecho en México, DR © 2010."

Figura 26. Interfaz de captura para nuevo perfil.



Para regresar al menú de ese catálogo dar click en [Lista de perfiles](#).



Figura 27. Interfaz de lista de perfiles existentes.

Para la edición del perfil dar click en [Editar](#) el cual desplegara el formulario con los datos que fueron capturados anteriormente para editar la información que se desee se elige el campo a modificar y se realizan los cambios pertinentes, una vez satisfecho se da click en [Actualizar](#) y se guardan los cambios realizados.



Figura 28. Interfaz de captura para la edición de perfil anteriormente capturada.



Para eliminar un perfil basta con solo dar click en Eliminar, el cual mostrara una pantalla de advertencia diciendo “¿Estás seguro de eliminar al perfil *****?”.



Figura 29. Interfaz de baja de perfil



Salir del Sistema

Para salir del sistema se debe dar click en el botón de Salir y automáticamente saldrá del sistema enviándote a la página de ingreso.



Figura 30. Interfaz al salir del sistema.