

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Documento de Diseño**

*Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco  
utilizando un canal RSS*

Elaboró

Francisco Javier Hernández Guzmán

Matricula: 208203175

Asesora

Dra. Ma. Lizbeth Gallardo López

Departamento de Sistemas

Trimestre 12O

Diciembre 2012

# Índice

<b>1.Introducción.....</b>	<b>3</b>
<b>2.Especificación del sistema.....</b>	<b>4</b>
2.1.Requisitos funcionales.....	4
2.2.Requisitos no funcionales.....	5
<b>3.Diagrama de casos de uso general .....</b>	<b>7</b>
<b>4.Escenarios de uso.....</b>	<b>7</b>
4.1.Caso de uso: Almacenar información.....	7
4.1.1.AI – 1: Subir archivo.....	8
4.1.2.AI – 2: Llenar formulario.....	10
4.2.Caso de uso: Indexar y clasificar información.....	11
4.2.1.ICI – 1: Clasificar información.....	11
4.3.Caso de uso: Publicar y difundir información.....	12
4.3.1.PDI – 1: Publicar información.....	13
4.3.2.PDI – 2: Difundir información.....	14
<b>5.Clases del sistema .....</b>	<b>15</b>
<b>6.Análisis de robustez.....</b>	<b>17</b>
<b>7.Módulos del sistema .....</b>	<b>18</b>
7.1.Módulo 1.....	18
7.1.1.Caso de uso: Almacenar información.....	18
7.2.Módulo 2.....	20
7.2.1.Caso de uso: Indexar y Clasificar Información.....	20
7.3.Módulo 3.....	21
7.3.1.Caso de uso: Publicar y Difundir Información.....	21
<b>8.Esquema de la base de datos .....</b>	<b>23</b>
8.1.Tabla diccionario.....	23
8.2.Tabla categoría.....	24
8.3.Tabla clasificación.....	24
8.4.Tabla administrador.....	24
8.5.Tabla anuncio.....	25

## **1. Introducción**

En el presente documento se describe la especificación del *Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS* a través de los requerimientos funcionales y no funcionales; además de los artefactos de diseño, tales como: diagramas de caso de uso generales y específicos, los escenarios de uso, el diagrama de clases, el esquema de la base de datos y el análisis de robustez. También se describe cada uno de los módulos que conforman el sistema, a saber: módulo 1, encargado de recolectar los anuncios que proporciona el administrador; módulo 2, encargado de clasificar y categorizar cada uno de los anuncios; y módulo 3, encargado de publicar y difundir los anuncios.

## **2. Especificación del sistema**

### *2.1.Requisitos funcionales*

El *Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS* cumple con los requisitos que se indican a continuación.

#### Registro de anuncios

El sistema proporciona dos modalidades para el registro de anuncios. El primero consiste en un formulario de tres campos, donde se solicitan los siguientes datos: Título, fecha y descripción.

La segunda modalidad permite al usuario subir un archivo que contenga varios anuncios con la condición de que debe ser un archivo con extensión *.txt*, esta segunda modalidad permite registrar varios anuncios en solo un paso, es decir evitará que el administrador agregue anuncio por anuncio.

#### Clasificación de anuncios

El sistema toma la fecha de ocurrencia del evento deportivo/cultural, el trámite académico/administrativo o el evento académico para determinar si se trata de un anuncio actual o un anuncio futuro.

Se define como *anuncio actual* a aquel anuncio que tiene lugar en la semana en curso; se define *anuncio futuro* a aquel anuncio que tiene lugar en semanas posteriores a la semana en curso.

Para el caso de los artículos del reglamento de estudios superiores, si el administrador no le asigna una fecha de difusión, el sistema le asigna una fecha de difusión aleatoria.

#### Publicación de anuncios

El sistema cuenta con una página web, en la cual se muestra el resultado de la gestión que hace el sistema con la información que se le proporciona.

En la página web, se muestran dos secciones, con los siguientes nombres: Anuncios actuales y Anuncios futuros.

#### Difusión de anuncios

Los anuncios son difundidos tres veces, la primera vez se realiza tres días antes del evento o trámite, la segunda vez un día antes del evento o trámite, y la tercera vez el día del evento o trámite.

### Actualización de anuncios

El administrador puede actualizar parcial o completamente la información que conforma un anuncio, cuando el sistema detecte que existe algún error en ese anuncio.

### Eliminación de anuncios

El sistema es el encargado de la gestión de la información, por lo que es el único que puede eliminar anuncios de la base de datos, esta acción la realiza cuando detecta que la fecha de un anuncio es anterior a la fecha actual del sistema (es decir el evento ya sucedió), entonces el sistema elimina permanentemente la información. Para el caso de los artículos y trámites académicos, el sistema los mantendrá almacenados y continuamente les asignará fechas aleatorias de difusión a los anuncios, los trámites académicos/administrativos tienen fechas preestablecidas por su naturaleza de ocurrencia periódica (cada trimestre).

### Seguridad

El sistema pide al administrador autenticarse, solicitándole un nombre de usuario y una contraseña, para que pueda tener acceso a las herramientas del sistema.

El sistema protege las contraseñas de los administradores bajo un esquema de cifrado Advanced Encryption Standard (AES).

## *2.2.Requisitos no funcionales*

### Funcionalidad

- Registro de administradores: Se pueden tener varios administradores, aunque no se cuenta con un módulo para el registro de nuevos administradores en línea, por seguridad.
- Seguridad: El sistema solo dará acceso a los administradores con credenciales de acceso.

### Facilidad de uso

- Una vez que el administrador obtenga acceso al sistema podrá observar todas las herramientas disponibles.
- El sistema contará con fuentes y colores adecuados para facilitar la lectura del texto contenido en el sistema.

## Fiabilidad

Cuando el sistema no pueda agregar anuncios a la base de datos, mostrará el siguiente mensaje: “Ocurrió un error al agregar el anuncio, vuelva más tarde”.

## Rendimiento

El sistema decidirá que información es obsoleta (eventos pasados) y desistirá de ella, de esta manera se ahorrará espacio de almacenamiento.

## Soporte

- El sistema estará disponible solo en idioma español.
- El administrador únicamente proporciona información al sistema, el sistema se encarga de la gestión de ésta, por lo que el administrador no interviene en esta actividad.

## Implementación

El sistema se desarrolla como un sistema web.

## Hardware:

- Notebook HP Pavilion dv4-1224la con procesador Intel(R) Pentium(R) Dual CPU T3400 2.17GHz, Memoria RAM de 3072 MB y Disco duro de 250GB.
- Servidor HP Proliant.

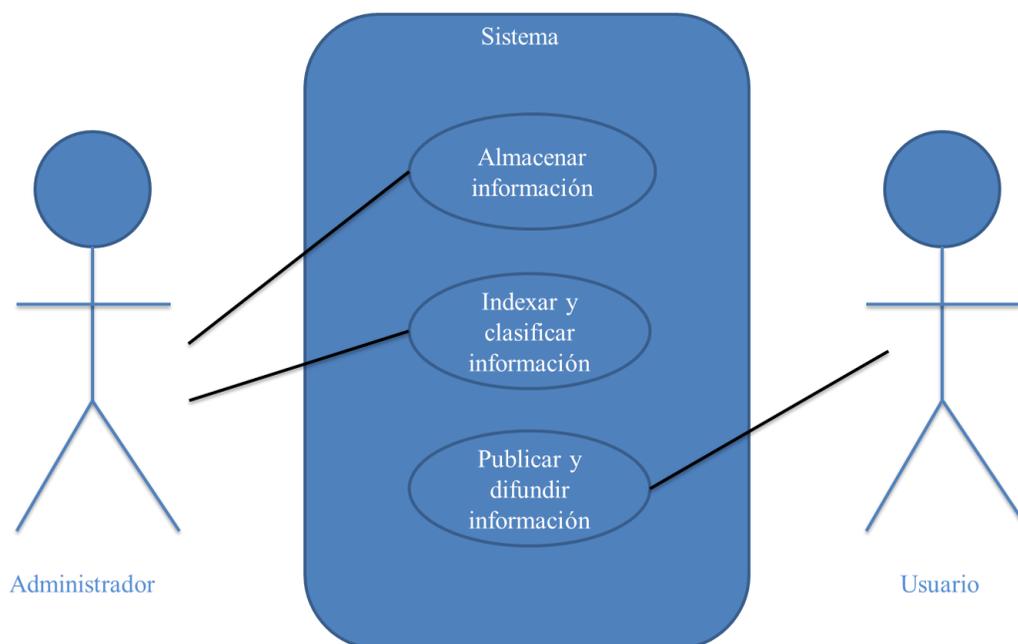
## Software:

- NetBeans: Entorno de desarrollo integrado de código abierto.
- Lenguaje de programación Java.
- MySQL: Manejador de bases de datos de código abierto.
- GlassFish: Servidor de aplicaciones de código abierto.
- Apache Struts: Framework web de código abierto.

## Interface

El sistema es accesible desde los navegadores Google Chrome, Internet Explorer 8 (o superior), o Firefox (Versión 10 en adelante).

### 3. Diagrama de casos de uso general



**Figura 1: Diagrama de casos de uso general**

En la figura 1, se presenta el diagrama general de casos de uso del proyecto, en él se muestran los tres módulos que conforman el sistema. El sistema está conformado por tres módulos generales: el primero está encargado de recoger los anuncios proporcionados por el usuario y almacenarlos en la base de datos; el segundo módulo clasifica y cataloga los anuncios; y el tercero está encargado de la difusión de los anuncios. Cada caso de uso general tiene, a su vez, tres casos de uso específicos que se detallan en la sección 7.

### 4. Escenarios de uso

#### 4.1. Caso de uso: Almacenar información

Subir archivo: El archivo debe contener anuncios con: título, fecha en caso de ser un trámite académico/administrativo o un evento deportivo/cultural, así como una descripción de no más de 280 caracteres.

Llenar formulario: Los datos que se registran son: Título, fecha en caso de ser un trámite académico/administrativo o un evento deportivo/cultural, así como una descripción de no más de 280 caracteres.

#### 4.1.1. AI – 1: Subir archivo

Descripción: Caso de uso que permite subir un archivo con extensión *.txt*, que contiene uno o más anuncios a difundir.

Actores:

- Primario: Administrador.
- Secundario: Ninguno.

Disparador: El actor primario solicita al sistema subir un nuevo archivo.

Precondiciones: No existe ningún anuncio registrado en el sistema igual a los nuevos anuncios contenidos en el archivo.

Poscondiciones: El o los anuncios quedan almacenados en la base de datos para su posterior clasificación, categorización y difusión.

Flujo principal:

1. El actor primario oprime el botón “Seleccionar archivo”.
2. El sistema muestra una ventana donde el usuario puede seleccionar el archivo que contiene los anuncios.
3. El actor primario selecciona el archivo que contienen los anuncios que desea difundir.
4. El sistema procesa el archivo.
5. El sistema busca (por nombre) si alguno de los anuncios del archivo ya se encuentran en la base de datos.
6. El sistema almacena solo los nuevos anuncios en la base de datos.

Flujos alternativos:

- a. El archivo no es de extensión .txt.
  1. El sistema detecta que el archivo contiene una extensión desconocida (No es .txt).
  2. El sistema muestra el siguiente mensaje “Solo se aceptan archivos con extensión .txt”.
  3. El actor primario selecciona otro archivo con una extensión soportada por el sistema (.txt).
  4. El flujo continúa en el punto 4 del Flujo principal.
  
- b. Alguno de los anuncios incluidos en el archivo ya se encuentra en la base de datos.
  1. El sistema detecta que un anuncio ya se encuentra en la base de datos.
  2. El sistema muestra una ventana con el listado de los anuncios que ya se encuentran almacenados en la base de datos, con la opción de editar el anuncio.
  3. El flujo continúa en el punto 6 del Flujo principal.
  
- c. La fecha ingresada para un anuncio es anterior a la fecha actual del sistema.
  1. El sistema detecta que la fecha del anuncio es anterior a la fecha actual del sistema.
  2. El sistema muestra una ventana con el listado de los anuncios con errores.
  3. El sistema muestra una opción para editar los anuncios.
  4. El flujo continúa en el en el punto 6 del Flujo principal.

Requerimientos no funcionales:

El tiempo de demora para el proceso “cargar archivo” depende de la velocidad de Internet del usuario.

#### 4.1.2. AI – 2: Llenar formulario

Descripción: Caso de uso que permite registrar en el sistema un nuevo anuncio.

Actores:

- Primario: Administrador.
- Secundario: Ninguno.

Disparador: El actor primario solicita al sistema registrar un nuevo anuncio.

Precondiciones: El anuncio no existe en la base de datos.

Poscondiciones: El anuncio queda registrado en la base de datos.

Flujo principal:

1. El actor primario llena el formulario con los datos de un nuevo anuncio.
2. El actor primario oprime el botón “Anunciar”.
3. El sistema busca (por el nombre) si el anuncio ya se encuentra almacenado en la base de datos.
4. El sistema almacena en la base de datos el nuevo anuncio.

Flujos alternativos:

- a. El anuncio ya se encuentra registrado en la base de datos.
  1. El sistema detecta que el anuncio ya se encuentra almacenado en la base de datos.
  2. El sistema muestra el siguiente mensaje “Ya existe un anuncio con el mismo nombre o con la misma descripción”.
- b. La fecha ingresada para el anuncio es anterior a la fecha actual del sistema.
  1. El sistema detecta que la fecha del anuncio es anterior a la fecha actual del sistema.
  2. El sistema muestra el siguiente mensaje “La fecha seleccionada es anterior a la actual”.
  3. El flujo continúa en el en el punto 1 del Flujo principal.

Requerimientos no funcionales:

El procedimiento de “Llenar formulario” no debe tomar más de dos minutos.

#### 4.2.Caso de uso: *Indexar y clasificar información*

Clasificar información: La información se clasifica en dos grandes conjuntos: Actual (eventos de la semana en curso) y por utilizar (eventos que sucederán en semanas futuras); a su vez dentro de cada conjunto la información se clasifica en tres categorías: eventos deportivos/culturales, tramites académicos/administrativos y artículos.

##### 4.2.1. ICI – 1: Clasificar información

Descripción: Caso de uso que permite clasificar los anuncios almacenados en la base de datos.

Actores:

- Primario: Ninguno.
- Secundario: Administrador.

Disparador: Es agregado un nuevo anuncio a la base de datos.

Precondiciones: Ninguna.

Poscondiciones: El nuevo anuncio queda clasificado en alguna de las tres categorías (Evento deportivo /cultural, Tramite académico/administrativo o Artículo).

Flujo principal:

1. El sistema agrega un nuevo anuncio a la base de datos.
2. El sistema hace una primera clasificación del anuncio con base en la fecha en la que sucederá el evento o tramite.
3. El anuncio recibe un identificador que define su clasificación en información Actual o Por utilizar.
4. El sistema determina a que categoría pertenece el anuncio con base en el título y de ser necesario en la descripción del anuncio.
5. El anuncio recibe un identificador que define su clasificación en alguna de las tres categorías.

Flujos alternativos:

- a. Se determina la categoría del anuncio con base en el título.
  1. El sistema coteja las palabras que conforman el título del anuncio con las palabras almacenadas en el diccionario del sistema.
  2. El sistema determina si se trata de un trámite, un evento o un artículo.
  3. El flujo continúa en el punto 5 del Flujo principal.
  
- b. Se determina la categoría del anuncio con base en el contenido de la descripción del anuncio.
  1. El sistema coteja las palabras que conforman la descripción del anuncio con las palabras almacenadas en el diccionario del sistema.
  2. El sistema determina si se trata de un trámite, un evento o un artículo.
  3. El flujo continúa en el punto 5 del Flujo principal.

Requerimientos no funcionales:

El resultado del proceso de clasificación es visible en la página del administrador.

#### *4.3.Caso de uso: Publicar y difundir información*

**Publicar información:** La información se publica en una página que solo es visible para el administrador del sistema; se muestra la información clasificada y la fecha programada para su difusión.

**Difundir información:** La información que se difunde consta de dos elementos: título, y descripción.

#### 4.3.1. PDI – 1: Publicar información

Descripción: Caso de uso que permite publicar en una página web los anuncios almacenados en la base de datos.

Actores:

- Primario: Ninguno.
- Secundario: Administrador.

Disparador: Es clasificado un nuevo anuncio en la base de datos.

Precondiciones: El anuncio debe estar clasificado.

Poscondiciones: El nuevo anuncio es publicado en la página del administrador.

Flujo principal:

1. El sistema clasifica un nuevo anuncio en la base de datos.
2. El sistema extrae de la base de datos el nuevo anuncio.
3. El sistema muestra el anuncio en la página web del actor secundario.

Flujos alternativos:

- a. El anuncio está clasificado en información Actual.
  1. El anuncio se muestra en el apartado “Anuncios actuales”.
  2. El anuncio se muestra con la etiqueta correspondiente a su clasificación: Evento deportivo/cultural, Trámite académico/administrativo o Artículo.
- b. El anuncio está clasificado en información Por utilizar.
  1. El anuncio se muestra en el apartado “Anuncios futuros”.
  2. El anuncio se muestra con la etiqueta correspondiente a su clasificación: Evento deportivo/cultural, Trámite académico/administrativo o Artículo.

Requerimientos no funcionales:

Para tener acceso a la página web donde se muestra esta información el administrador debe autenticarse.

#### 4.3.2. PDI – 2: Difundir información

Descripción: Caso de uso que permite difundir utilizando un canal RSS los anuncios publicados en la página web del administrador.

Actores:

- Primario: Ninguno.
- Secundario: Alumno.

Disparador: La fecha de difusión del anuncio es próxima (3 días, un día).

Precondiciones: El anuncio debe tener una fecha de difusión.

Poscondiciones: El anuncio es difundido utilizando un canal RSS, el alumno recibe en su lector de RSS el anuncio.

Flujo principal:

1. El sistema detecta que debe difundir un anuncio.
2. El sistema difunde el anuncio utilizando un canal RSS.
3. El actor secundario recibe en su lector de RSS el anuncio.

Flujos alternativos:

- a. La fecha de difusión del anuncio es próxima.
  1. El sistema detecta que la fecha de difusión del anuncio esta próxima (dentro de tres días, un día o el anuncio debe difundirse ese día).
  2. El sistema difunde el anuncio utilizando un canal RSS.
  3. El flujo continúa en el punto 3 del Flujo principal.

Requerimientos no funcionales:

El formato de visualización para los anuncios es proporcionado por el lector de RSS que el alumno utiliza.

## 5. Clases del sistema

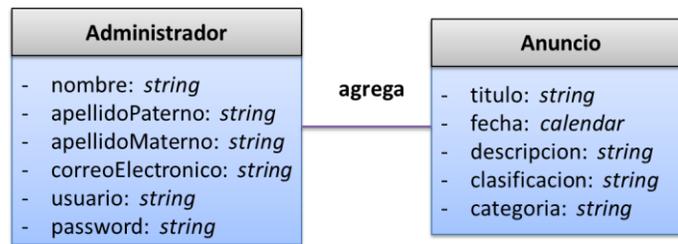


Figura 2: Clases contenedoras

Las clases de *Anuncio* y *Administrador* (clases contenedoras Figura 2), contienen los atributos y los *getters* y *setters* utilizados para obtener y modificar la información proporcionada por el usuario del sistema.

Las clases de tipo catálogo contienen los métodos para realizar operaciones básicas de adición, eliminación y actualización de anuncios y administradores (ver figura 3). Estas clases contienen las consultas SQL para explotar la base de datos.

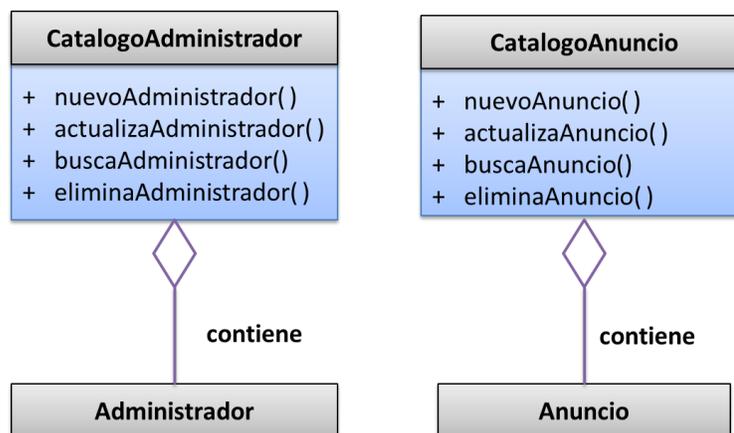
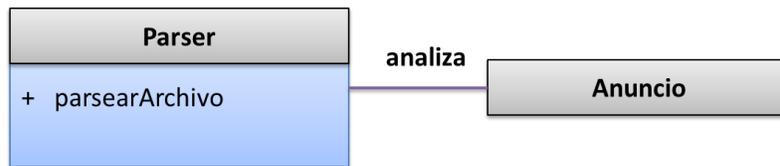


Figura 3: Clases catálogos

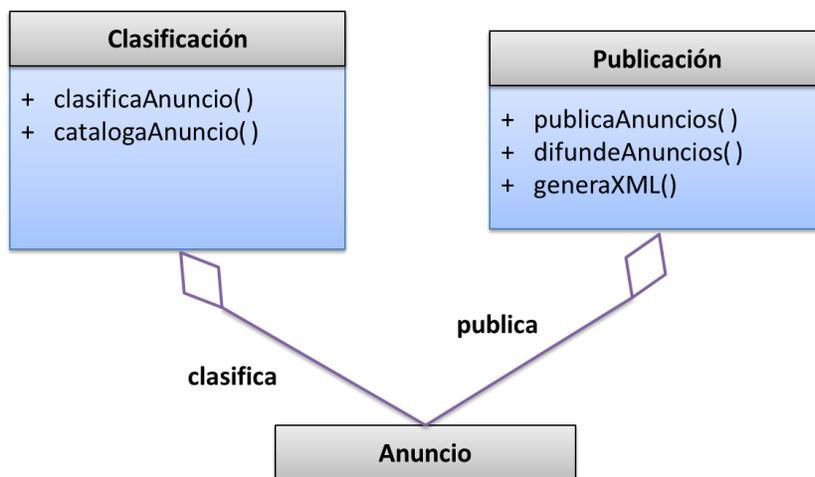
La clase *CatalogoAnuncio* está relacionada con la clase *Anuncio*, ya que en la clase *CatalogoAnuncio* se instancia la clase *Anuncio* para agregar nuevos objetos a la base de datos con los atributos de dicha clase.

Algo similar pasa con la clase *CatalogoAdministrador*, que instancia a la clase *Administrador*, para agregar objetos a la base de datos con los atributos de esta última.



**Figura 4: Clase Parser**

La clase *Parser* (ver figura 4) contiene el método encargado de separar en *tokens* el contenido del archivo proporcionado por el usuario, para conformar la estructura de cada anuncio y proceder a catalogar, clasificar y almacenar la información en la base de datos.



**Figura 5: Clases controladoras**

Las clases de *Publicación* y *Clasificación* (ver figura 5) son utilizadas para realizar las operaciones de clasificación y categorización de los anuncios proporcionados por el administrador, para posteriormente publicarlos en la página web y así difundirlos.

## 6. Análisis de robustez

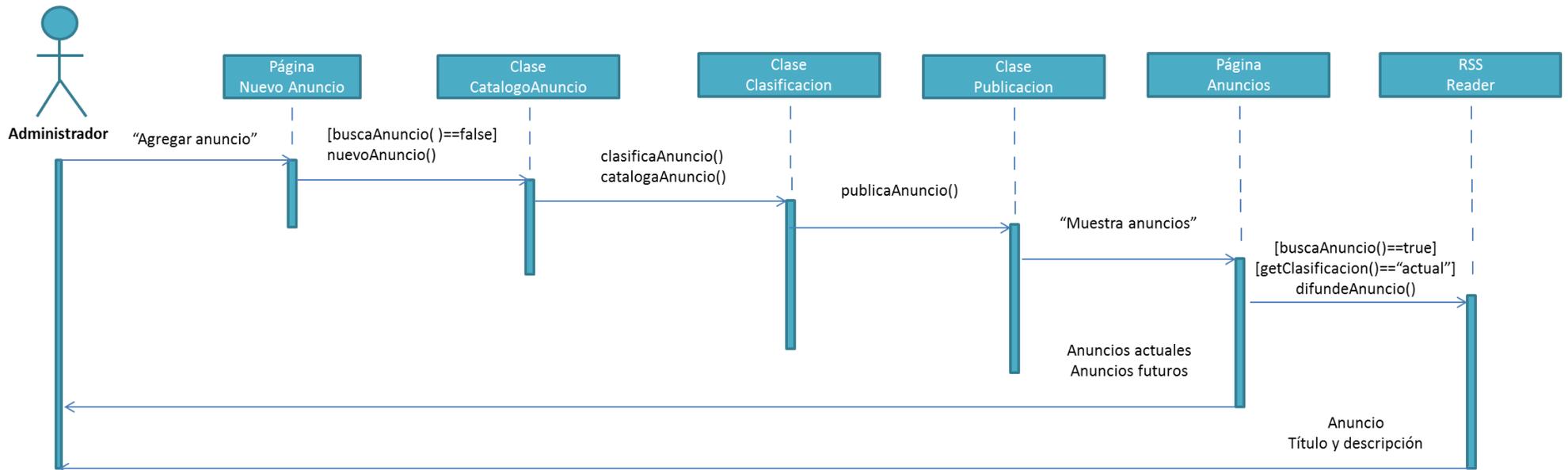


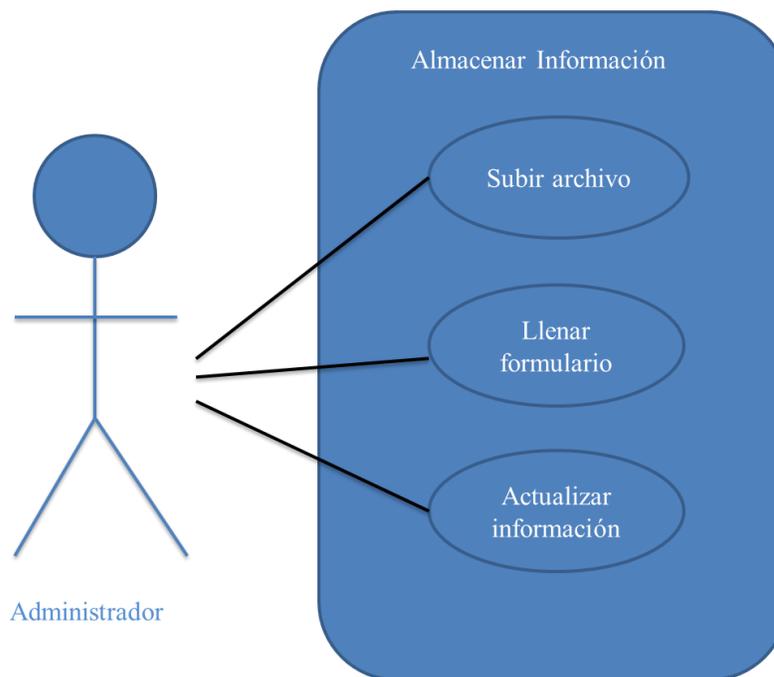
Figura 6: Diagrama general de secuencia

En la figura 6 se muestra el diagrama general de secuencia; se optó por mostrar de forma general el proceso desde que el administrador ingresa la información hasta que el anuncio es difundido, con el objetivo de que el proceso sea más claro y no se pierda la línea de los procesos.

## 7. Módulos del sistema

### 7.1. Módulo 1

#### 7.1.1. Caso de uso: Almacenar información



**Figura 7: Caso de uso Almacenar Información**

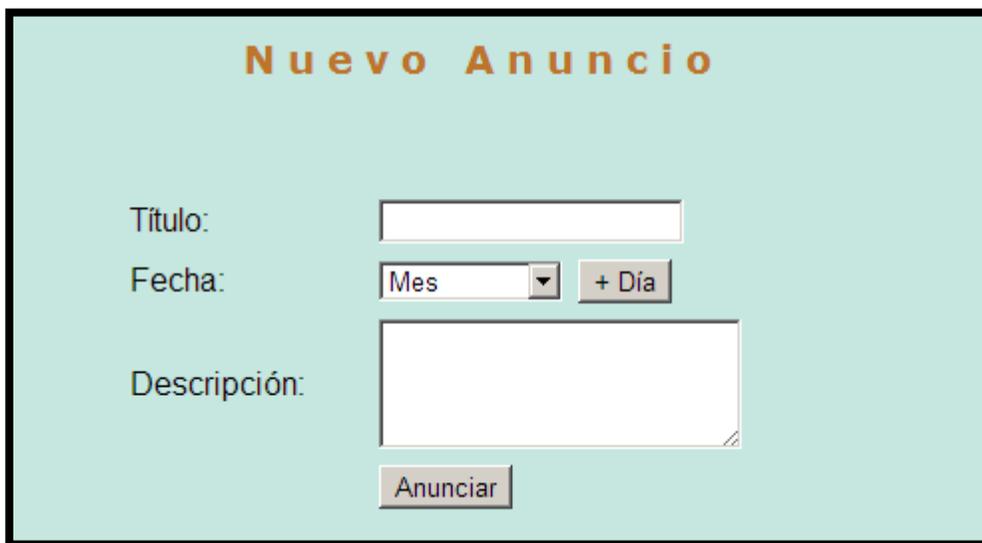
En la figura 7 se muestra el caso de uso: *Almacenar Información*, así como los casos de uso específicos que lo conforman:

- *Subir archivo*, cuya interfaz gráfica se muestra en la figura 9, es un módulo que permite cargar archivos con extensión *.txt*.
- *Llenar formulario*, permite proporcionar anuncios al sistema de manera individual, tal y como se muestra en la figura 8.
- *Actualizar información*, permite actualizar los anuncios que no lograron ser catalogados en el sistema y permite definir una categoría para estos.

El caso de uso general *Almacenar información*, describe el *Módulo 1* del sistema. El módulo 1 es el único que interactúa con el administrador para el ingreso de los anuncios, posteriormente la información es almacenada en una base de datos.

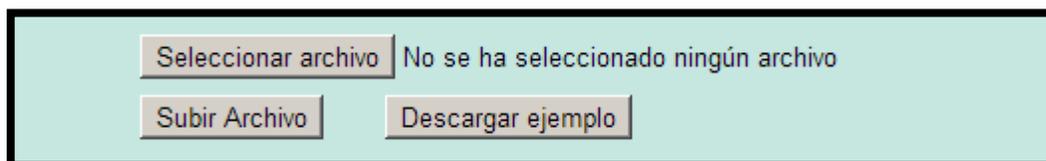
El módulo 1 cuenta con una interfaz gráfica donde se solicita al usuario un archivo con los anuncios a almacenar; también cuenta con un formulario donde se puede agregar uno a uno cada elemento, a saber: título, cuerpo y fecha, que conforma un anuncio de tipo trámite, evento ó artículo.

En las Figuras 8 y 9, se muestran las capturas de pantalla del formulario *Nuevo Anuncio* y la interfaz utilizada para cargar un archivo.



**Figura 8: Formulario para agregar un nuevo anuncio.**

La figura 8 muestra la interfaz gráfica del sistema encargada de recoger la información asociada a un anuncio, en esta ventana el administrador debe proporcionar el título del anuncio que desea difundir, así como la fecha en la que desea que sea difundido y una breve descripción del anuncio, de no más de 280 caracteres.

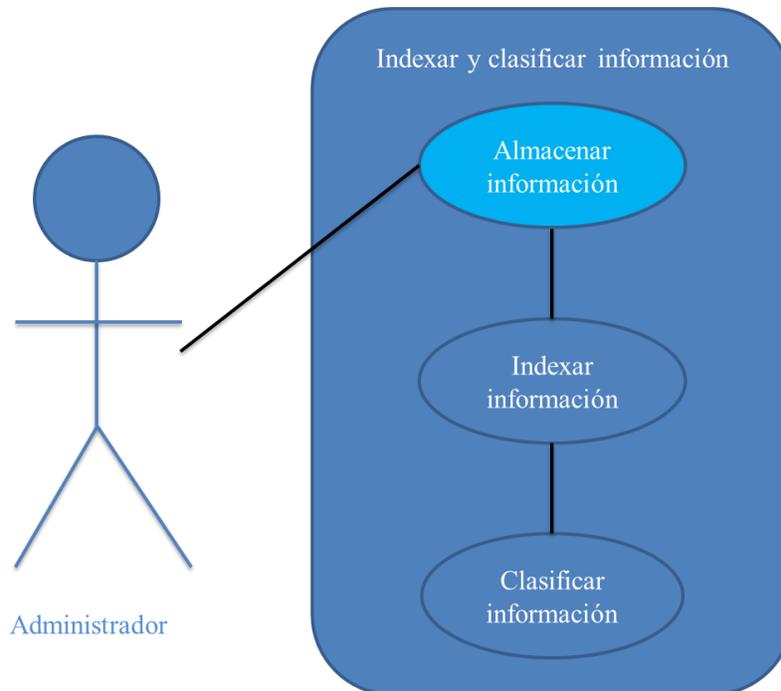


**Figura 9: Sección para seleccionar un archivo y subirlo.**

En la figura 9 se muestra la parte complementaria del módulo 1, esta parte es la encargada de solicitar un archivo con extensión *.txt*, este archivo debe contener una relación de los anuncios que se desean difundir y debe contener: título, cuerpo y fecha del anuncio.

## 7.2.Módulo 2

### 7.2.1. Caso de uso: Indexar y Clasificar Información



**Figura 10: Caso de uso Indexar y Clasificar Información**

En la figura 10 se muestra el caso de uso general: *Indexar y Clasificar Información*, así como los casos de uso específicos que lo conforman:

- *Indexar información*, se encarga de ordenar cronológicamente los anuncios, en función de la fecha de difusión programada para cada uno de ellos.
- *Clasificar información*, se encarga de clasificar los anuncios a partir de su fecha de realización y cataloga los anuncios en base a las palabras clave que contenga, a saber: artículos, trámites académicos, eventos deportivos y eventos culturales.

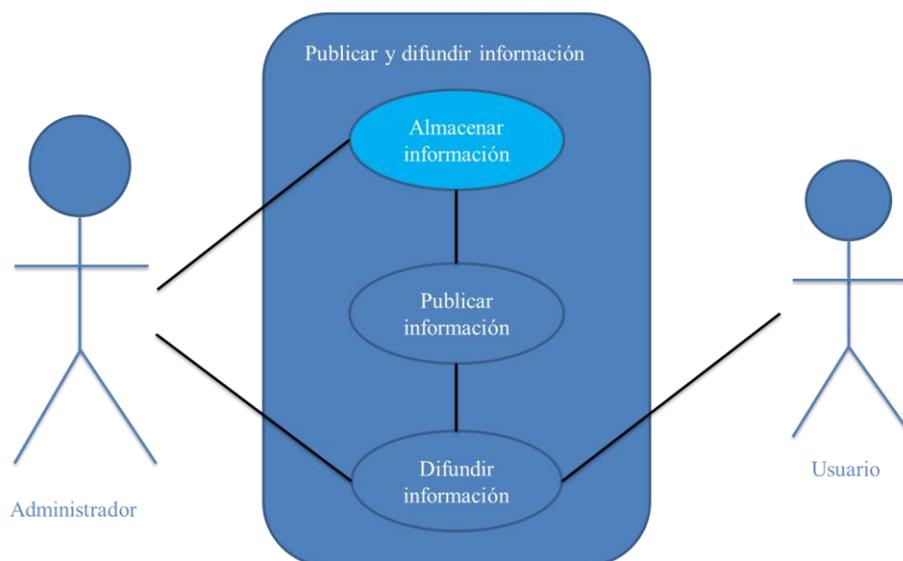
El caso de uso general *Indexar y Clasificar Información* describe el funcionamiento del módulo 2 del sistema; este módulo se encarga clasificar y catalogar los anuncios almacenados en la base de datos, los anuncios son clasificados en anuncios actuales (eventos que ocurrirán en la semana en curso) y anuncios futuros (eventos que ocurrirán posteriormente a la semana en curso). Posteriormente el anuncio es catalogado en una de las siguientes categorías: Artículo, trámite académico/administrativo, actividad deportiva, actividad cultural y evento académico.

El sistema clasifica los anuncios tomando en cuenta la fecha que el administrador proporciona para su difusión; para el caso de los artículos, estos pueden tener o no tener fecha de difusión, en caso de que el administrador no proporcione la fecha de difusión, el sistema le asigna una fecha aleatoria.

Una vez que los anuncios son clasificados y catalogados, son publicados por el sistema, posteriormente son difundidos utilizando un canal RSS, según su fecha de ocurrencia. Este módulo general *Indexar y Clasificar Información* no cuenta con interfaz gráfica.

### 7.3.Módulo 3

#### 7.3.1. Caso de uso: Publicar y Difundir Información



**Figura 11: Caso de uso Publicar y Difundir Información**

En la figura 11 se muestra el caso de uso general: *Publicar y Difundir Información*, así como los casos de uso específicos que lo conforman:

- *Publicar información*, se encarga de mostrar en la página web del administrador, los anuncios que ya han sido clasificados y catalogados.
- *Difundir información*, se encarga de difundir los anuncios utilizando un canal RSS para que éstos lleguen a la comunidad universitaria.

El caso de uso *Publicar y Difundir Información* describe el funcionamiento del *Módulo 3* del sistema. El módulo 3 se encarga de publicar la información clasificada y almacenada en la base de datos en la página web del administrador. Posteriormente la información es difundida mediante un canal RSS, al que los alumnos pueden suscribirse (vía Internet) y recibir la información en su *smartphone*, *tablet* o computadora.

En el módulo 3 también se gestiona la información para determinar, qué información permanece en la base de datos y qué información se elimina. Los anuncios que permanecen en la base de datos son: Artículos y Trámites académicos/administrativos porque deben ser difundidos cada trimestre a la comunidad universitaria, con el propósito de que los estudiantes conozcan o recuerden sus derechos y obligaciones. Por ejemplo el proceso de altas, bajas y cambios en la primera semana, o el proceso de renunciaciones en la quinta semana. Los anuncios son difundidos de la siguiente manera:

- Se difunden 3 días antes del evento.
- Un día antes del evento.
- Y el día del evento.

Posteriormente la información que no está catalogada como artículo o trámite es eliminada de la base de datos. Por ejemplo: los eventos deportivos o culturales. Estos anuncios son eliminados de la base de datos, porque son actividades o eventos que solo tienen lugar una sola vez; por ejemplo, una conferencia (evento académico) o un partido de fútbol (evento deportivo), son eventos o actividades que solo ocurren una vez y no se repiten al siguiente trimestre, al menos no la misma actividad como tal, por esta razón no se considera importante mantenerlo en la base de datos.

Aunado a esto, con una buena gestión de información, se puede ahorrar espacio de almacenamiento, manteniendo en la base de datos solo aquellos anuncios que son constantes en su ocurrencia, cada trimestre, como son: artículos del reglamento y trámites académico-administrativos.

## 8. Esquema de la base de datos

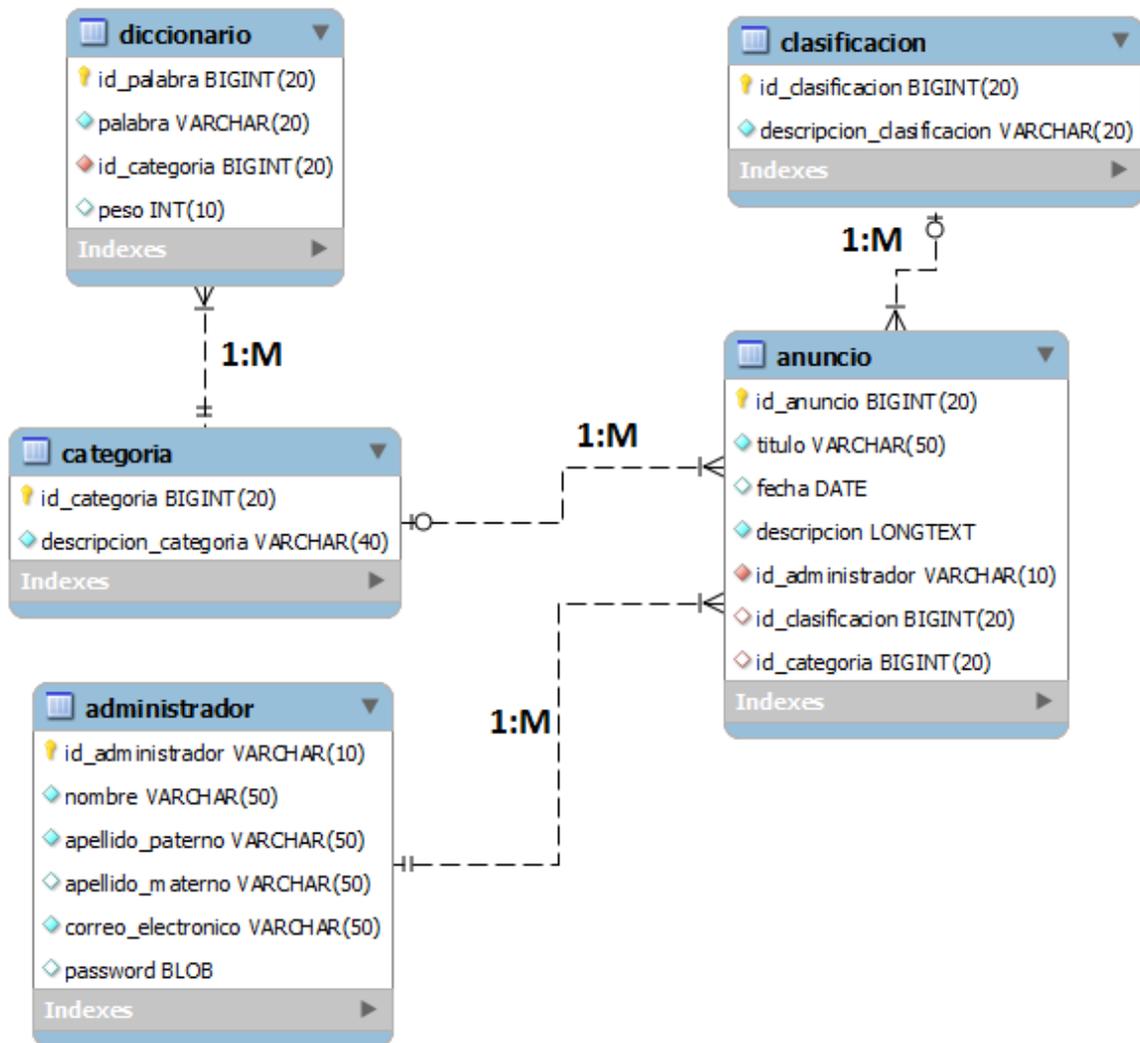


Figura 12: Estructura de la base de datos

En la figura 12 se muestra la estructura de la base de datos del sistema, en la base de datos se almacenan los anuncios y toda la información asociada a estos.

### 8.1. Tabla diccionario.

- *id\_palabra*: identificador único para cada palabra del diccionario.
- *palabra*: palabra del diccionario.
- *id\_categoria*: identificador de la categoría a la cual pertenece la palabra del diccionario.

- *peso*: valor de la palabra. Utilizado para categorizar los anuncios con mayor rapidez. Este campo puede tomar dos valores: 1 o 2; cuando el campo tiene peso uno, prácticamente cualquier anuncio que contenga esta palabra puede ser categorizado de inmediato en la categoría a la cual pertenece la palabra.

Por ejemplo, en el anuncio con la siguiente descripción:

“Se llevará a cabo un partido de **fútbol** en la cancha empastada de la universidad.”

Podemos decir que la palabra fútbol es una palabra con peso 1 ya que es una palabra que puede determinar con facilidad a un anuncio en la categoría de evento deportivo.

Cabe destacar que ésta tabla está relacionada con la tabla categoría, ya que una palabra puede existir en la base de datos solo si ésta tiene una categoría.

### 8.2.Tabla categoría.

- *id\_categoria*: identificador único para cada categoría de la base de datos.
- *descripcion\_categoria*: contiene el nombre de cada categoría.

En esta tabla se almacenan las categorías utilizadas para catalogar los anuncios. Para el proyecto terminal se consideraron las siguientes categorías: Artículo, Trámite académico/administrativo, Evento cultural, Evento deportivo, Evento académico.

### 8.3.Tabla clasificación.

- *id\_clasificacion*: identificador único para cada clasificación.
- *descripcion\_clasificacion*: contiene el nombre de cada clasificación.

En esta tabla se almacenan los tipos de clasificación que puede tomar un anuncio, éstas son: información actual o información por utilizar. En información actual son clasificados los anuncios que contengan información acerca de los eventos que sucederán en la semana en curso. Por el contrario en información por utilizar están clasificados los anuncios que tienen información de eventos que sucederán una o varias semanas después de la semana en curso.

### 8.4.Tabla administrador.

- *id\_administrador*: identificador único para cada administrador.
- *nombre*: nombre de pila del administrador.
- *apellido\_paterno*: apellido paterno del administrador.
- *apellido\_materno*: apellido materno del administrador.
- *correo\_electronico*: correo electrónico asociado al administrador.
- *password*: contraseña utilizada para ingresar al sistema.

En esta tabla se almacena toda la información asociada al administrador del sistema. Pueden existir varios administradores del sistema, los usuarios serán creados por el administrador responsable y este será el único que podrá autorizar y dar de alta nuevos usuarios. Por seguridad no se tiene un módulo en línea para agregar nuevos administradores.

La tabla está relacionada directamente con la tabla de anuncios ya que no puede existir un anuncio a menos que haya sido proporcionado por un administrador, y éste este registrado en la base de datos.

#### 8.5. Tabla anuncio.

- *id\_anuncio*: identificador único para cada anuncio.
- *titulo*: título del anuncio.
- *fecha*: fecha de difusión del anuncio.
- *descripcion*: descripción del anuncio.
- *id\_administrador*: identificador del administrador responsable del anuncio.
- *id\_clasificacion*: identificador de la clasificación a la cual pertenece el anuncio.
- *id\_categoria*: identificador de la categoría a la cual pertenece el anuncio.

En esta tabla se almacena toda la información asociada a los anuncios, está relacionada con la tabla categoría, la tabla administrador y con la tabla clasificación, esto porque el anuncio debe tener los identificadores de: quién agrego el anuncio, a qué categoría pertenece y cuál es su clasificación.

Para poder implementar la base de datos, se incluye en el CD el script de la base de datos, así como el código fuente del sistema; este script está diseñado de tal manera que al momento de ejecutarse en el servidor MySQL, la base de datos creada contendrá lo siguiente:

- Diccionario con las palabras de cada categoría: artículos, trámites académicos, eventos deportivos, eventos culturales y eventos académicos.
- Categorías: artículos, trámites académicos, eventos deportivos, eventos culturales y eventos académicos.
- Credenciales de un administrador: Se tiene el nombre de usuario y la contraseña por defecto para el sistema, usuario: *admin*; contraseña: *administrador*.

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Manual de Usuario**

*Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco  
utilizando un canal RSS*

Elaboró

Francisco Javier Hernández Guzmán

Matricula: 208203175

Asesora

Dra. Ma. Lizbeth Gallardo López

Departamento de Sistemas

Trimestre 12O

Diciembre 2012

## Índice

1. Introducción .....	3
2. Login .....	4
3. Página inicial .....	5
4. Formulario nuevo anuncio .....	6
5. Módulo para anuncios sin clasificar .....	8
6. Módulo para agregar nuevas palabras .....	9
7. Muestra de anuncios .....	11

## **1. Introducción**

En el presente documento, se hace un recorrido por los módulos que integran el *Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS*; se presentan las impresiones de pantalla de cada uno de los módulos sobre los cuales un administrador puede interactuar. Cabe destacar que el administrador solo proporciona la información de los anuncios al sistema, y que en ningún caso interviene en cómo se clasifica no cómo se almacenan los anuncios en la base de datos. El administrador, sin embargo, puede agregar nuevas palabras al sistema y asignarlas a una categoría, para que estas palabras faciliten la tarea de categorización de los anuncios. En caso de que el sistema no logre categorizar uno o varios anuncios, el sistema le dará la opción para que él mismo los catalogue.

## 2. Login

El administrador para que tenga acceso al sistema deberá autenticarse, mediante un usuario y una contraseña (ver figura 1). Las credenciales para ingresar al sistema en este momento son:

- Usuario: admin
- Contraseña: administrador



The image shows a login interface with a light blue background. At the top, the text "¿Deseas difundir anuncios?" is displayed in a bold, orange font. Below it, the text "Inicia Sesión" is also in a bold, orange font. The form contains two input fields: "Usuario:" followed by a white text box, and "Contraseña:" followed by a white text box. Below the password field is a grey button with the text "Entrar" in white.

**Figura 1: Inicio de sesión del sistema**

### 3. Página inicial

Una vez que el usuario ingresa sus credenciales, el sistema le dará acceso a las herramientas (ver figura 2). El administrador puede:

- Agregar nuevas palabras.
- Clasificar anuncios que no pudieron ser clasificados por el sistema.
- Agregar un nuevo anuncio.
- Agregar un archivo con anuncios.
- Ver el resultado de la clasificación de los anuncios.
- Cerrar sesión.

Anuncios sin clasificar   Agregar palabras   Ver anuncios   Cerrar sesión

## Nuevo Anuncio

Título:

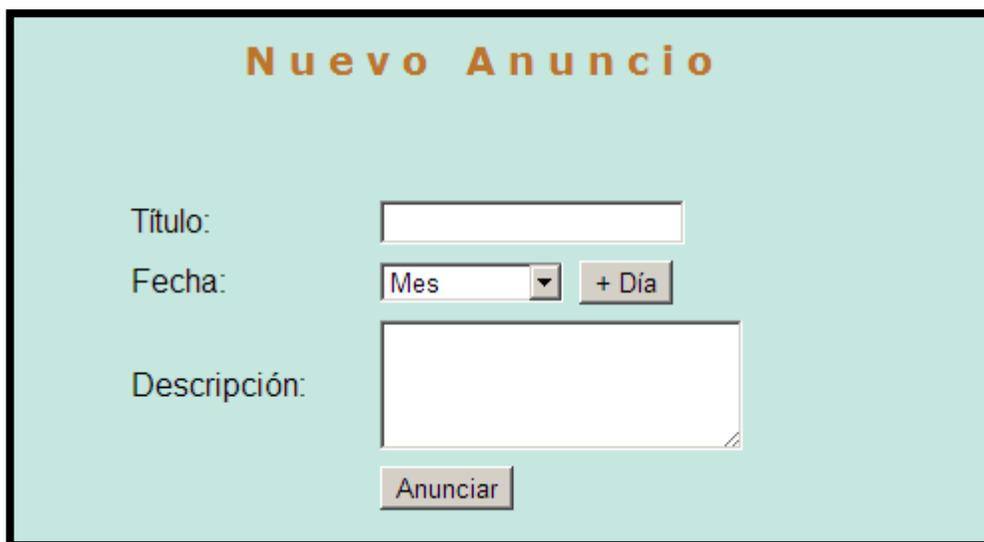
Fecha:

Descripción:

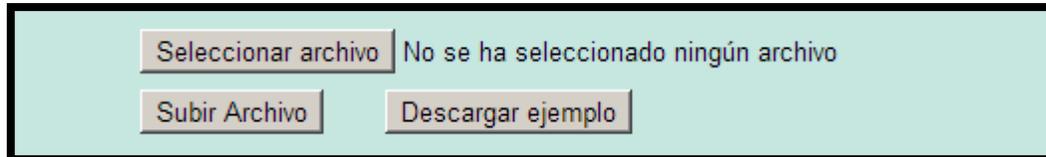
No se eligió ningún archivo

**Figura 2: Página inicial**

#### 4. Formulario nuevo anuncio



**Figura 3: Formulario para agregar un nuevo anuncio.**



**Figura 4: Sección para seleccionar un archivo y subirlo.**

En las figuras 3 y 4 se muestran las dos formas en las que el administrador puede proporcionar información al sistema. La figura 3 corresponde a un formulario que el administrador debe llenar para poder agregar anuncios (uno a uno) a la base de datos. Los datos que debe proporcionar son: título del anuncio, fecha de ocurrencia del evento que se desea anunciar y una breve descripción de no más de 280 caracteres. La figura 4 corresponde al apartado en donde el usuario puede proporcionar un archivo con la relación de los anuncios que desea difundir; el sistema cuenta con un ejemplo descargable que el administrador puede utilizar como plantilla, esto con el objetivo de que no haya errores al momento de procesar la información en el sistema. En la figura 5 se muestra la plantilla que el administrador puede utilizar para guiarse al momento de hacer la relación de anuncios que desea difundir.

//Nota: Este es un ejemplo de la estructura que deben llevar los anuncios.

//Nota: La fecha debe ir en el siguiente formato: yyyy-mm-dd

//Nota: La descripción del anuncio debe ir de corrido, es decir sin saltos de línea.

//Ejemplo de Artículo

Artículo x

Descripción del artículo.

//Ejemplo de Evento

Evento x

2012-11-10

Descripción del evento x.

**Figura 5: Plantilla descargable que el administrador debe utilizar para proporcionar la relación de anuncios en un archivo con extensión *.txt*.**

## 5. Módulo para anuncios sin clasificar

Anuncios sin clasificar		
Anuncio	Categoría	Acción
Semana de la salud	Categoría ▼	Listo
Terror en la UAM	Categoría ▼	Listo

Volver

**Figura 6: Módulo anuncios sin clasificar**

En la figura 6 se muestra el módulo en el cual el administrador puede observar los anuncios que no pudieron ser clasificados; con esta herramienta, el administrador podrá clasificar los anuncios en alguna de las categorías que se tengan disponibles; por lo pronto se tienen las siguientes categorías disponibles:

1. Artículo del reglamento de estudios superiores de la UAM
2. Trámite académico/administrativo
3. Evento cultural
4. Evento deportivo
5. Evento académico

En la categoría de *artículo*, se pueden agregar artículos del reglamento de estudios superiores de la UAM, que el administrador desee difundir.

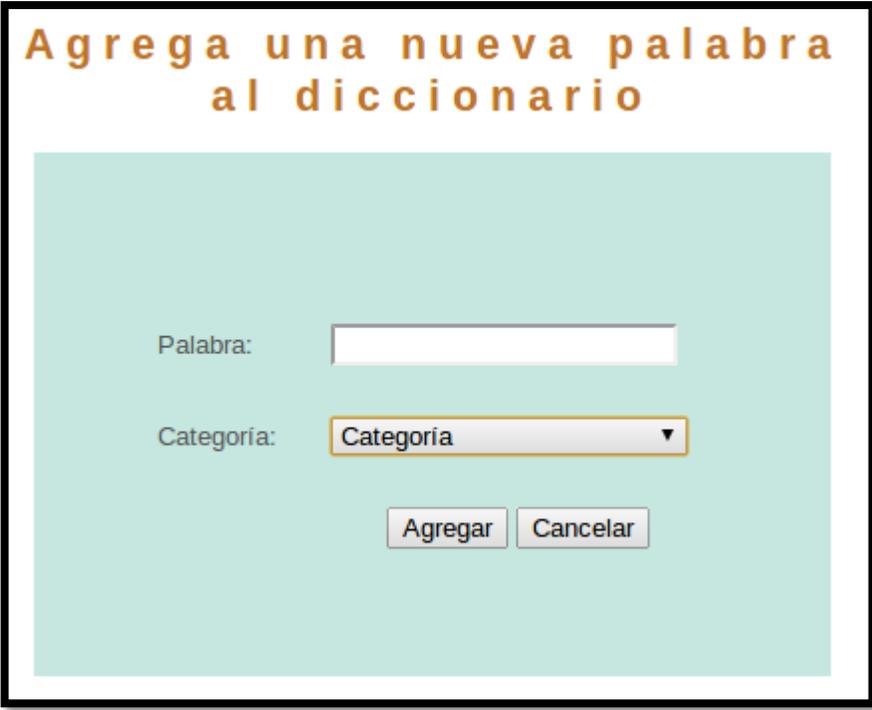
En la categoría de *trámite académico/administrativo*, el administrador puede agregar anuncios que corresponden a trámites que los alumnos pueden realizar en la universidad, tales como altas, bajas, renuncias a UEA, etc.

En la categoría de *evento cultural*, el administrador puede agregar anuncios que corresponden a eventos culturales tales como: conciertos, obras de teatro, exposiciones de arte, etc.

En la categoría de *evento deportivo*, el administrador puede agregar anuncios que corresponden a eventos deportivos tales como: partidos de fútbol, básquetbol, carreras, etc.

En la categoría de *evento académico*, el administrador puede agregar anuncios que corresponden a eventos académicos dentro de la Universidad, tales como: coloquios, presentación de temas de investigación, simposios, etc.

## 6. Módulo para agregar nuevas palabras



The image shows a web form titled "Agrega una nueva palabra al diccionario" in orange text. The form has a light teal background and contains the following elements:

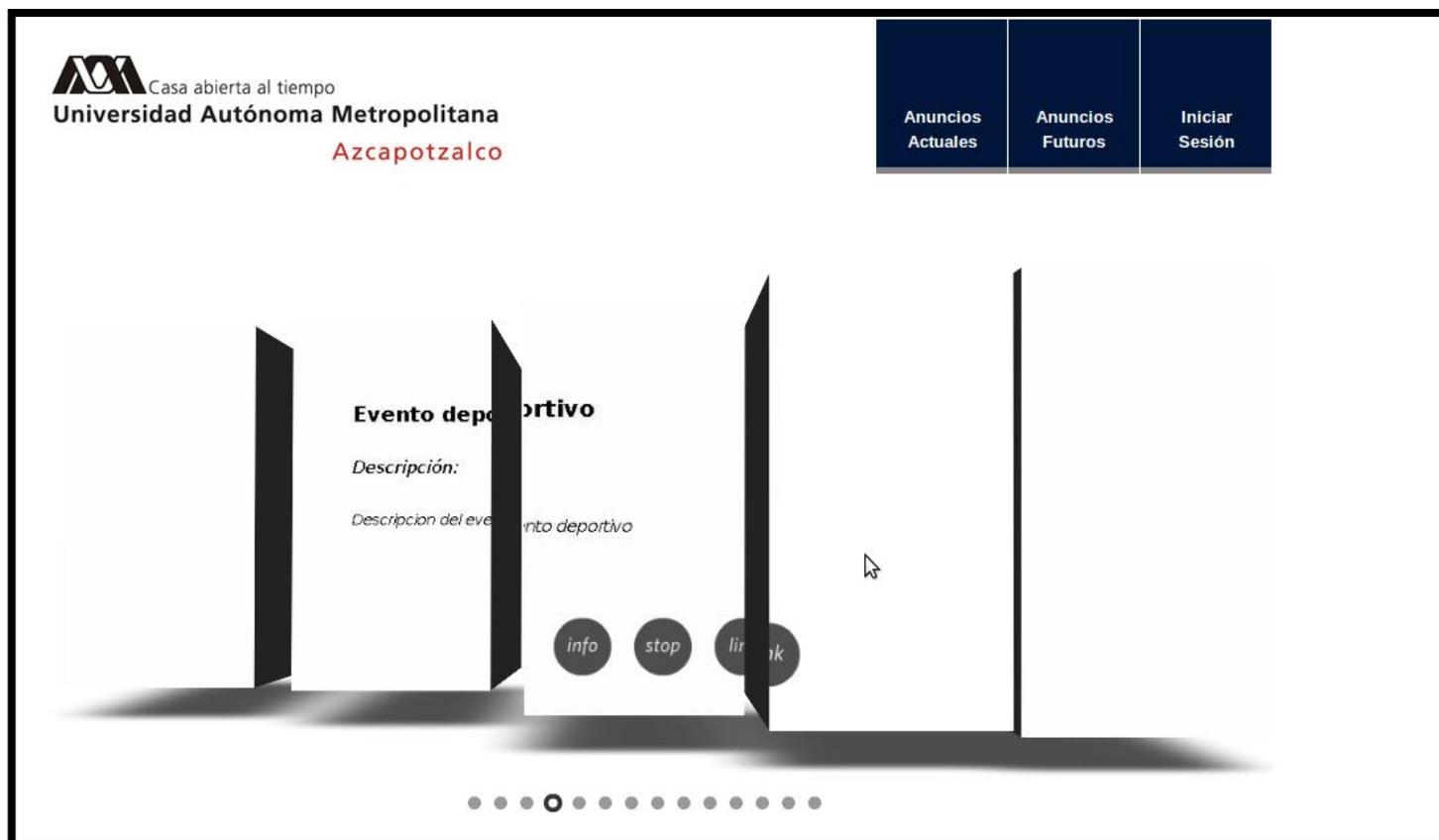
- A label "Palabra:" followed by a white text input field.
- A label "Categoría:" followed by a dropdown menu with "Categoría" selected and a downward arrow.
- Two buttons at the bottom: "Agregar" and "Cancelar".

**Figura 7: Módulo para nuevas palabras**

En la figura 7 se muestra el módulo utilizado para agregar nuevas palabras al diccionario; para lograrlo, el administrador debe elegir una categoría para la palabra que quiere agregar; esta palabra sirve para que el sistema discrimine los anuncios adecuadamente, en el proceso de categorizar los anuncios. Por ejemplo, el administrador desea agregar al sistema la palabra *altas*.

1. El administrador agrega a la base de datos la palabra *altas*.
  - 1.1. A esta palabra el administrador le asigna la categoría de *trámite académico/administrativo*.
  - 1.2. El sistema ya tiene palabras almacenadas con la misma categoría, palabras como: **renuncias, bajas, cambios**, inscripción, etc.
2. El administrador agrega al sistema un anuncio con la siguiente descripción: “En la primer semana del trimestre los alumnos pueden realizar: **altas, bajas y cambios** a UEA-Grupo, alumnos inscritos al trimestre anterior y que presentaron examen de recuperación.”
3. El sistema llama al método *catalogaAnuncio* de la clase *Clasificacion.java* y pasa como argumento la descripción del anuncio.
4. El sistema extrae del diccionario almacenado en la base de datos las palabras con categoría trámite académico/administrativo.
5. El sistema cruza la descripción del anuncio contra las palabras del diccionario, con objeto de encontrar en la descripción coincidencias entre las palabras del diccionario y las palabras de la descripción del anuncio.
6. El sistema encuentra las siguientes coincidencias: altas, bajas y cambios.
7. El sistema cataloga al anuncio como un anuncio de tipo trámite académico/administrativo.

## 7. Muestra de anuncios



**Figura 8:** Página que muestra el resultado de la clasificación y categorización de anuncios

En la figura 8 se muestra la sección en la que el administrador puede ver los anuncios almacenados en la base de datos una vez que se realizaron los procesos de categorización y clasificación de los anuncios. Es decir, en este apartado el administrador puede observar todos los anuncios almacenados en la base de datos del sistema. La página está dividida en dos secciones, la primera: *anuncios actuales*, muestra solo los anuncios de eventos que ocurrirán en la semana en curso; la segunda: *anuncios futuros*, muestra solo los anuncios de eventos que ocurrirán posteriormente a la semana en curso. Esta sección está accesible desde el módulo principal del sistema, oprimiendo el botón “ver anuncios”, ubicado en la parte superior derecha de la página inicial (ver figura 2).

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Manual Técnico**

*Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco  
utilizando un canal RSS*

Elaboró

Francisco Javier Hernández Guzmán

Matricula: 208203175

Asesora

Dra. Ma. Lizbeth Gallardo López

Departamento de Sistemas

Trimestre 12O

Diciembre 2012

## Índice

<b>1.Introducción.....</b>	<b>3</b>
<b>2.Hardware .....</b>	<b>4</b>
<b>3.Características de Red .....</b>	<b>4</b>
<b>4.Software .....</b>	<b>4</b>
<b>5.Estructura del directorio del sistema .....</b>	<b>5</b>
5.1. <i>Web Pages</i> .....	5
5.2. <i>SourcePackages</i> .....	6
5.2.1. <i>Conexión</i> .....	6
5.2.2. <i>Controladores</i> .....	7
5.2.3. <i>Modelo</i> .....	7
5.3. <i>Test Packages y Test Libraries</i> .....	7
5.4. <i>Libraries</i> .....	8
5.5. <i>Configuration Files</i> .....	8
<b>6.Proceso de instalación del sistema .....</b>	<b>9</b>
6.1. <i>Desplegar la aplicación en un servidor</i> .....	9
6.2. <i>Desplegar la aplicación en una PC</i> .....	10
6.3. <i>Acceder a la aplicación a través del dispositivo móvil</i> .....	10
6.4. <i>Instalación de la base de datos</i> .....	10
<b>7.Pruebas.....</b>	<b>11</b>
<b>8.Problemas durante el desarrollo.....</b>	<b>16</b>

## 1. Introducción

En el proyecto terminal que lleva por nombre: *Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS*, se desarrolló un sistema web utilizando el servidor de aplicaciones GlassFish, el manejador de base de datos MySQL, y el lenguaje de programación Java.

En el presente documento se muestra la estructura del directorio de la aplicación desarrollada en el proyecto terminal. También se presentan el software y hardware utilizado en el desarrollo del proyecto; así como el proceso de instalación que se debe seguir para que otro usuario pueda montar el sistema y realizar las pruebas aquí mostradas.

Se realizaron pruebas para los dos tipos de entrada de información de los anuncios con los que cuenta el sistema. El primero consta de llenar un formulario web; y para el segundo se proporciona un archivo con extensión *.txt* con la información a difundir.

## 2. Hardware

El sistema se desarrolló en una computadora portátil con las siguientes características:

Notebook HP Pavilion dv4-1224la con procesador Intel(R) Pentium(R) Dual CPU T3400 2.17GHz, Memoria RAM de 3072 MB y Disco duro de 250GB.

En este mismo equipo se montó la aplicación y se verificó su correcto funcionamiento.

Para verificar que la información se difunde correctamente por el canal RSS, se empleó un *smartphone* con las siguientes características:

Samsung Galaxy S GT-I9000T con sistema operativo Android y procesador de 1 GHz.

## 3. Características de Red

Si no se cuenta con un dominio público, es decir no se cuenta con un servicio de Web Hosting para el despliegue del sistema; entonces el equipo en el cual está montado el sistema, así como el dispositivo móvil empleado para recibir la información deben estar conectados a la misma red inalámbrica. De esta manera, el dispositivo móvil puede recibir las actualizaciones que el sistema realiza porque se encuentran en la misma red.

## 4. Software

- Sistema operativo GNU/Linux.
- NetBeans IDE: Entorno de desarrollo integrado de código abierto.
- MySQL: Manejador de bases de datos de código abierto.
- GlassFish: Servidor de aplicaciones de código abierto.
- Apache Struts: Framework web de código abierto.
- RssDemon: Lector de fuentes RSS

El software utilizado para el desarrollo del proyecto es de código abierto, evitando así problemas de licenciamiento.

Ubuntu (el sistema operativo en el que se desarrolló el proyecto) es un sistema estable, y cuenta con diversas herramientas que facilitan el desarrollo de software.

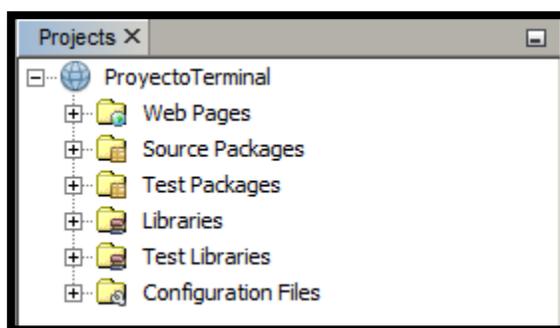
NetBeans es un entorno de desarrollo integrado muy completo; cuenta con un depurador que se encarga, incluso, de revisar el correcto funcionamiento de páginas JSP. Aunado a esto, el IDE facilita el desarrollo al tener disponibles frameworks como Struts, así como los controladores para manejadores de bases de datos.

MySQL es un sistema gestor de bases de datos muy estable y podemos encontrar documentación en su página web oficial, esa fue una de las principales razones por las que se decidió utilizar este manejador de bases de datos.

Se utilizó Apache Struts en su versión 1.3.10, ya que es la versión más estable; además, la documentación se encuentra en la página Web de Apache. Struts es una framework para el desarrollo de aplicaciones Web basados en el patrón de diseño MVC y empleando la plataforma Java EE.

RssDemon se utiliza en el dispositivo móvil como el software predeterminado para la lectura de la fuente RSS del canal del sistema. Se utilizó este software por su característica de permitir leer fuentes RSS aún cuando el dispositivo está en modo offline. En efecto, las fuentes se descargan al móvil cuando el teléfono está conectado a internet; en un momento posterior, el usuario puede leer las fuentes aunque ya no esté conectado a internet.

## 5. Estructura del directorio del sistema

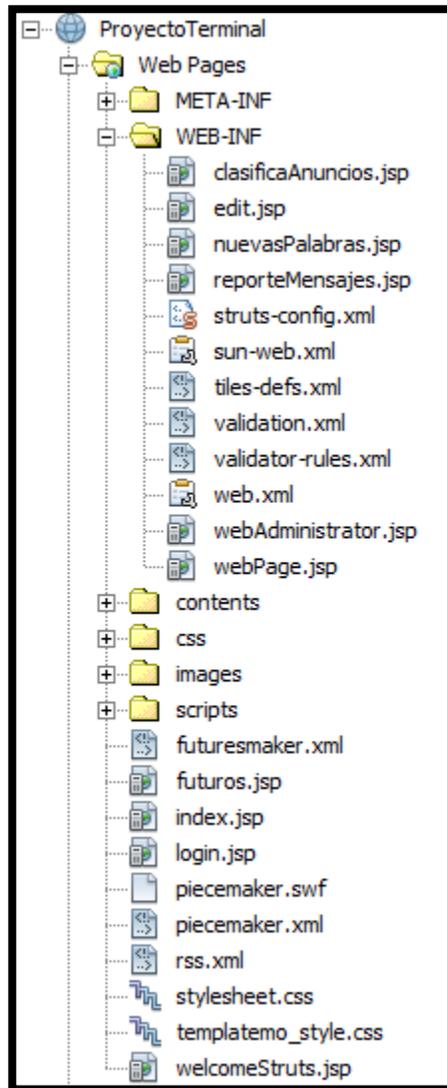


**Figura 1: Esquema general del directorio**

En la figura 1 se muestra el esquema general del directorio del *sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco* utilizando un canal RSS, en el que se pueden observar 6 carpetas. A continuación se presenta la descripción de cada una de ellas y su contenido.

### 5.1. Web Pages

En la carpeta *Web Pages* podemos encontrar todas las páginas Web a las que se pueden acceder sin necesidad de autenticarse en el sistema; ésta carpeta contiene, a su vez, subcarpetas en donde se encuentran archivos con las hojas de estilos empleadas en el diseño de las páginas. En la carpeta *Web Pages* además podemos encontrar otra carpeta llamada *WEB-INF* en esta carpeta están almacenadas todas las páginas a las que solo son accesibles mediante autenticación en el sistema (ver figura 2).



**Figura 2: Esquema de la carpeta Web Pages**

## 5.2. SourcePackages

En la carpeta *SourcePackages*, podemos encontrar cinco paquetes, siguiendo el patrón de diseño: Modelo – Vista – Controlador (MVC). Los paquetes son:

### 5.2.1. Conexión

En este paquete se encuentra la clase que permite la conexión y desconexión a la base de datos llamada SDN.sql (Sistema para Difusión de Noticias).

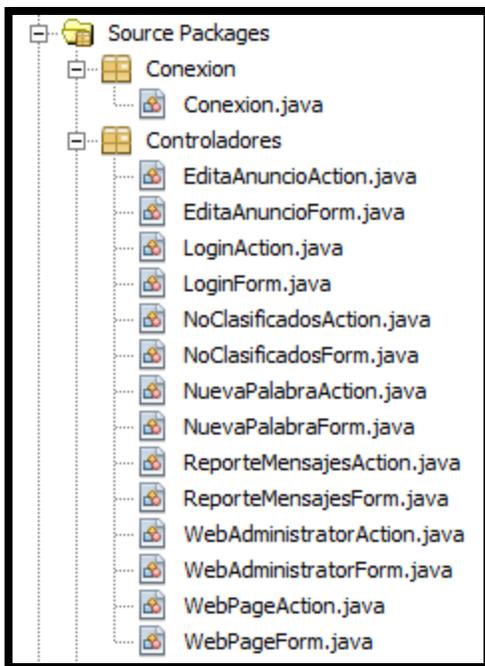
### 5.2.2. Controladores

En este paquete se encuentran todas las clases tipo *Action* y todas las clases tipo *Form*. Las clases tipo *Form* son utilizadas para recoger la información de los formularios Web del sistema. Las clases tipo *Action*, contienen el código que controla el flujo del sistema, según las acciones que tome el usuario.

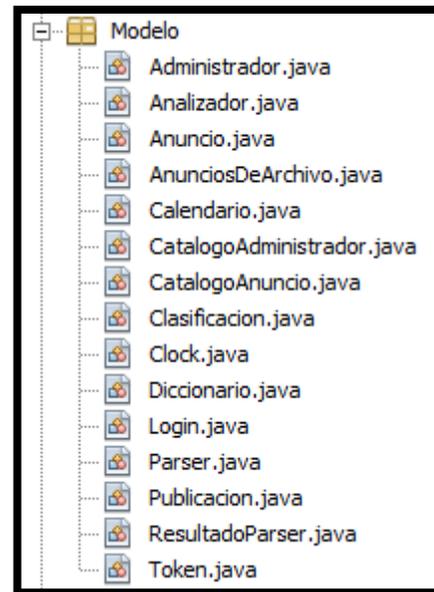
### 5.2.3. Modelo

En este paquete se encuentran las clases que definen todo la lógica de negocio.

En las figuras 3 y 4 se muestra la estructura del directorio de la carpeta *SourcePackages*.



**Figura 3: Esquemas paquetes  
Conexión y Controladores**



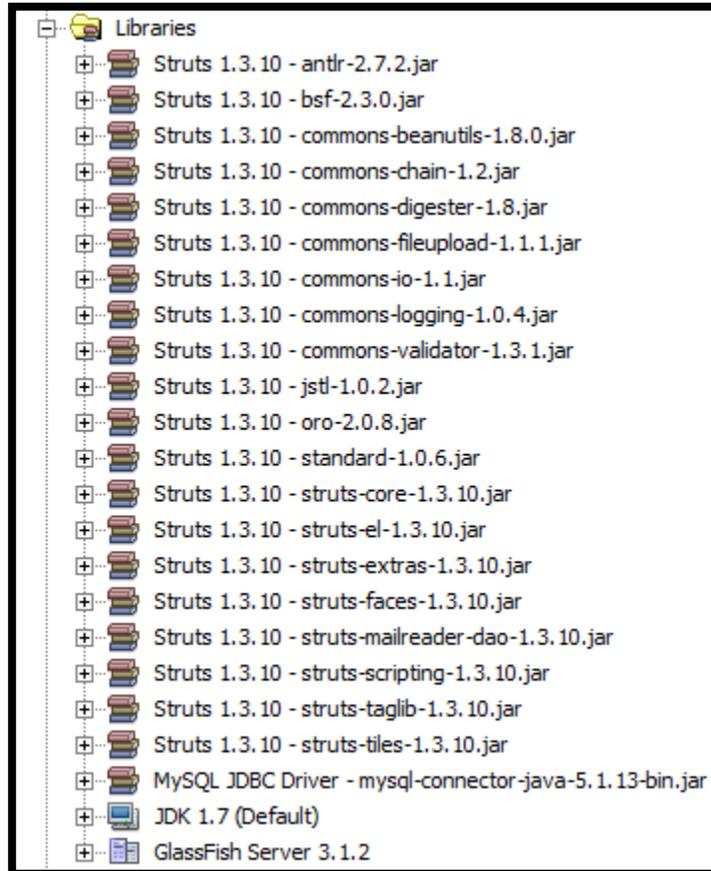
**Figura 4: Esquema paquete  
Modelo**

### 5.3. *Test Packages* y *Test Libraries*

En estas carpetas se almacenan las clases de prueba y las bibliotecas de prueba, para el proyecto terminal no fueron utilizadas.

#### 5.4. Libraries

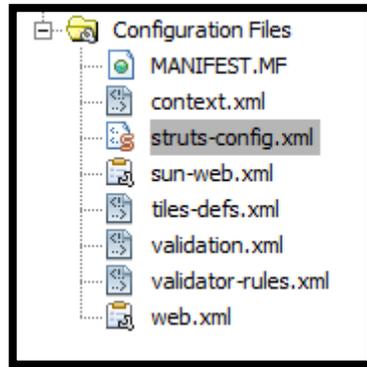
En esta carpeta se encuentran todas las bibliotecas necesarias para el correcto funcionamiento del sistema (ver figura 5).



**Figura 5: Bibliotecas necesarias para el sistema.**

#### 5.5. Configuration Files

En esta carpeta se encuentran todos los archivos de configuración del sistema; es decir, se especifica qué página debe lanzarse primero, cuando el usuario ingresa al sistema; además se especifica qué página *jsp* debe responder, cuando el usuario presiona un botón, así como el re-direccionamiento de las páginas al momento de navegar; también se especifica la codificación que deben tener todas las páginas para que puedan ser visualizados caracteres especiales como la ñ y tildes (ver figura 6).



**Figura 6: Archivos de configuración.**

## 6. Proceso de instalación del sistema

Para poder instalar la aplicación y utilizarla, se requiere la instalación previa del software siguiente:

En el servidor:

- GlassFish en su versión 3.1.2. Disponible en:  
<http://glassfish.java.net/es/downloads/3.0.1-final.html>
- MySQL Server en su versión 5.1.2. Disponible en:  
<http://dev.mysql.com/downloads/mysql/>
- El JDK de Java. Disponible en:  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

En el cliente:

- Navegador: Google Chrome, Internet Explorer 8 (o superior), o Firefox (Versión 10 en adelante).
- RssDemon para el dispositivo móvil, software recomendado, aunque se pueden utilizar otros como Google Reader, Any RSS reader, etc.

### 6.1. Desplegar la aplicación en un servidor

Para desplegar la aplicación web en el servidor, se debe colocar el archivo *ProyectoTerminal.war* (ubicado en la carpeta `\ProyectoTerminal\dist\`) en la carpeta de *autodeploy* del servidor de aplicaciones GlassFish.

Es decir mover: `\ProyectoTerminal\dist\ProyectoTerminal.war`

Hacia: `\Glassfish-3.1.2\glassfish\domains\domain1\autodeploy\`

Posteriormente, debe arrancarse el servidor utilizando el archivo por lotes incluido en la carpeta bin del servidor web GlassFish: `\Glassfish-3.1.2\glassfish\bin\startserv`

### *6.2.Desplegar la aplicación en una PC*

Si el sistema se despliega de manera local en una PC, ésta y el dispositivo móvil empleado para recibir las actualizaciones por RSS, deben estar conectados a la misma red inalámbrica.

### *6.3.Acceder a la aplicación a través del dispositivo móvil*

Para poder recibir los anuncios difundidos mediante RSS en el dispositivo móvil, el usuario debe suscribir el dispositivo al canal de información del sistema.

Añadir fuentes → Agregar URL de la fuente.

La URL de la fuente se especifica a continuación:

`http://IPdelHost:8080/ProyectoTerminal/rss.xml`

### *6.4.Instalación de la base de datos*

Para poder instalar la base de datos en el servidor MySQL se deben seguir los siguientes pasos:

1. Abrir una terminal.
2. Acceder al servidor MySQL: `mysql -h ip_servidor -u usuario -p`
3. Una vez dentro de la consola MySQL, si la base de datos no existe, podemos crearla con: `create database sdn`
4. Indicamos la base de datos a usar: `use sdn;`
5. El proceso de importación, sería ejecutar: `source /path/script_base_de_datos.sql`

El script ya trae precargado el usuario: *admin* y la contraseña: *administrador*.

## 7. Pruebas

El sistema cuenta con dos tipos de entrada de información, una “manual” llenando un formulario que solicita título del anuncio a difundir, fecha en la que ocurrirá el evento a anunciar y descripción del evento (ver figura 7).



The image shows a web form titled "Nuevo Anuncio" (New Advertisement) on a light green background. The form contains the following elements:

- Título:** A single-line text input field.
- Fecha:** A date selection interface consisting of a dropdown menu currently showing "Mes" (Month) and a "+ Día" (Day) button.
- Descripción:** A multi-line text area.
- Anunciar:** A button to submit the advertisement.
- File Upload Section:** A "Seleccionar archivo" (Select file) button, followed by the text "No se eligió ningún archivo" (No file selected).
- Subir Archivo:** A button to upload a file.
- Descargar ejemplo:** A button to download an example file.

**Figura 7: Formulario para nuevo anuncio.**

Los campos están validados de manera que no puede existir ningún campo vacío, y la fecha ingresada debe ser igual a la actual o posterior, en caso de que se dejen campos vacíos, la fecha ingresada sea anterior a la actual, o la descripción sea mayor a 280 caracteres se mostrarán mensajes de error como se muestran en las figuras 8, 9 y 10.

**Nuevo Anuncio**

¿Y el Título del anuncio?

Título:

Fecha:

Descripción:

Figura 8: Campos vacíos.

**Nuevo Anuncio**

La fecha seleccionada es anterior a la actual

Título:

Fecha:

Descripción:

Figura 9: Fecha anterior a la actual.

**Nuevo Anuncio**

El tamaño de la descripción debe ser menor a 280 caracteres

Título:

Fecha:

Descripción:

Figura 10: Tamaño de la descripción muy grande.

En caso de que se quiera administrar de información al sistema utilizando un archivo, éste debe tener extensión *.txt*, de lo contrario enviará un mensaje de error (ver figura 11).

**Nuevo Anuncio**

Solo se aceptan archivos con extensión *.txt*

Título:

Fecha:

Descripción:

No se eligió ningún archivo

**Figura 11: Extensión de archivo incorrecta.**

Una vez que se ingresa un archivo con la extensión requerida y se hace el proceso de exploración del archivo para extraer los anuncios contenidos en él, el sistema guía al administrador a una página donde se muestran los anuncios procesados correctamente, así como los anuncios que tuvieron problemas para ser almacenados acompañados del motivo del error (ver figuras 12 y 13).

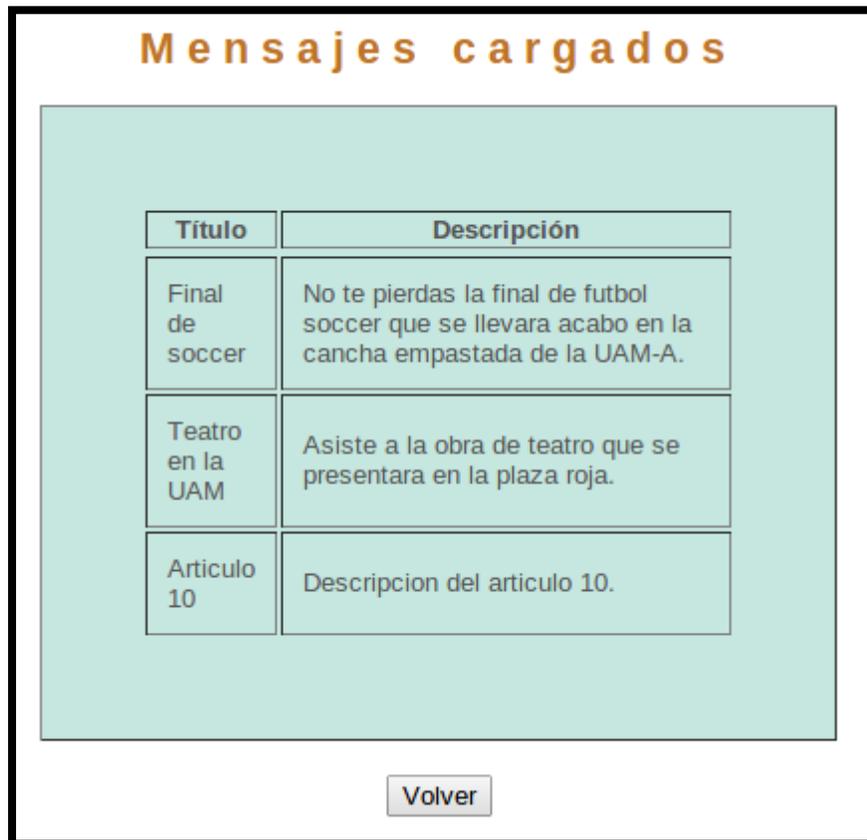


Figura 12: Mensajes cargados con éxito



Figura 13: Mensajes con errores



## 8. Problemas durante el desarrollo

Durante el desarrollo de este proyecto, surgieron varios contratiempos como: Problemas de bibliotecas que no se descargaron antes de iniciar la programación. También se presentaron problemas en la base de datos, había veces en las que el navegador arrojaba “*Too many connections*”, este problema era generado por las conexiones que se abrían durante la prueba del sistema, las cuales no se estaban cerrando; es decir, al momento de escribir el código fuente y abrir conexiones a base de datos, las conexiones no se cerraban, y ocasionaban problemas de muchas conexiones a la vez del mismo usuario.

Existen algunos problemas con el manejo de caracteres del idioma español, como tildes. Este problema se resolvió configurando las páginas JSP con *encoding* tipo UTF8 y se utilizó esa misma codificación en la base de datos.

Otro contratiempo que se presentó, fue al momento de leer la información de los archivos y realizar los procesos correspondientes de clasificación y categorización; al utilizar vectores dinámicos para almacenar y manejar la información, el sistema se volvía lento al recorrer los vectores por completo. Básicamente y lo primero que se hizo para disminuir el tamaño de los vectores fue, eliminar los artículos del lenguaje (el, la, los, etc.), esta acción redujo en tamaño los vectores de almacenamiento y en efecto los campos de búsqueda, dejando solo palabras significativas de los anuncios.

Lo siguiente que se hizo fue ajustar la memoria asignada a los vectores de almacenamiento, es decir, cuando se crea un vector dinámico, la máquina virtual de Java asigna una determinada cantidad de memoria para los vectores, pero hay veces que no se ocupa toda esa memoria y por lo tanto se desperdician recursos; empleando el método *vector.trimToSize()*, se ajusta la cantidad de memoria, dejando solo la memoria que en verdad se está utilizando, y de esta forma se ahorran recursos de almacenamiento.

Una vez que se tienen los vectores sólo con la memoria justa y necesaria, se dejaron de utilizar ciclos *for* anidados para el recorrido de los vectores, y se optó por utilizar algunas clases precompiladas de Java para buscar elementos en los vectores, así como para realizar operaciones de adición y eliminación de elementos de los vectores.

Con estas pequeñas acciones se logró una reducción visible en el tiempo de ejecución del sistema cuando procesa la información de los archivos.

Cabe destacar que en la propuesta de proyecto terminal, se planteó que el sistema aceptaría documentos con extensión *.txt* y *.pdf*. Sin embargo, el sistema solo acepta archivos con extensión *.txt*, porque por un lado no se tiene, por el momento, una forma de acceder directamente a los campos de un archivo pdf; y por otro lado, para quien edite los anuncios tendrá que hacerlo sobre un archivo tipo *.doc*, *.odt*, ó *.txt*; por lo tanto, se decidió que los archivos que contienen los anuncios tengan extensión *.txt*.

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Reporte Final de Proyecto Terminal II**

*Sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco  
utilizando un canal RSS*

Elaboró

Francisco Javier Hernández Guzmán

Matrícula: 208203175

Asesora

Dra. Ma. Lizbeth Gallardo López

Departamento de Sistemas

Trimestre 12O

Diciembre 2012

# Índice

<b>Resumen .....</b>	<b>3</b>
<b>1.Introducción.....</b>	<b>4</b>
<b>1.1.Objetivos .....</b>	<b>4</b>
1.1.1.Objetivo general.....	4
1.1.2.Objetivos particulares.....	4
<b>2.Antecedentes .....</b>	<b>5</b>
<b>3.Justificación .....</b>	<b>6</b>
<b>4.Metodología de base empleada en el proyecto .....</b>	<b>7</b>
<b>5.Desarrollo del proyecto .....</b>	<b>9</b>
5.1.Diseño .....	9
5.1.1.Diagrama de casos de uso general.....	9
5.1.2.Caso de uso: Almacenar información.....	10
5.1.3.Caso de uso: Indexar y Clasificar Información.....	12
5.1.4.Caso de uso: Publicar y Difundir Información .....	15
5.1.5.Arquitectura general del sistema .....	17
5.1.6.Estructura de la base de datos.....	18
5.1.7.Clases del sistema.....	21
5.2.Implementación.....	23
5.2.1.Hardware .....	23
5.2.2.Software .....	23
5.2.3.Funcionalidad.....	24
<b>6.Conclusiones y perspectivas del proyecto.....</b>	<b>32</b>
6.1.Logros .....	32
6.2.Cuestiones sin resolver.....	33
6.3.Trabajo a futuro.....	33
<b>Anexos .....</b>	<b>34</b>
Anexo A: Método clasificaAnuncio de la clase Clasificacion .....	34
Anexo B: Método catalogaAnuncio de la clase Clasificacion .....	36
Anexo C: Estructura básica de un archivo XML para difusión RSS.....	38
<i>Método generaXML de la clase Publicacion .....</i>	<i>39</i>
Anexo D: Método nuevoAnuncio de la clase CatalogoAnuncio.....	40
<b>Glosario .....</b>	<b>41</b>
<b>Bibliografía .....</b>	<b>42</b>

## **Resumen**

En este proyecto terminal se desarrolló un *sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS*. Se trata de un sistema web diseñado bajo el patrón de diseño MVC (Modelo Vista Controlador). La metodología seguida en el desarrollo del proyecto fue el Proceso Unificado, un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso. El sistema está dividido en tres módulos. El primer módulo se encarga de recolectar los anuncios proporcionados por el usuario. El segundo módulo está encargado de catalogar los anuncios en: artículos del reglamento de estudios superiores de la UAM, trámites académicos/administrativos, eventos deportivos y eventos culturales; además de clasificarlos por la fecha de difusión. El tercer módulo tiene las tareas de: Publicar en la página web del administrador los anuncios ya clasificados y catalogados, eliminar los eventos que ya ocurrieron y difundir los anuncios a la comunidad universitaria mediante un canal RSS.

## **1. Introducción**

En la UAM Azcapotzalco suceden cosas todos los días, desde eventos culturales y deportivos, hasta trámites académicos y administrativos de suma importancia para los estudiantes. Muchas de estas noticias son difundidas vía correo electrónico (institucional) a los estudiantes, por lo que muchos de estos nunca se enteran, porque no lo consultan y si lo hacen, lo realizan con poca frecuencia; también se utilizan medios de difusión impresos o se publica la información en la página web de la institución.

La página web de la UAM Azcapotzalco cuenta con varios menús y submenús, por lo que a veces resulta difícil encontrar la información que se está buscando y aunque cuenta con un buscador, éste arroja resultados no solo de la institución, sino de sitios web externos, por lo que resulta aún más difícil encontrar la información.

Los administradores del sitio web tienen que gestionar una gran cantidad de información a diario para mantenerlo actualizado, esto es una gran labor ya que deben estar pendientes de qué información deben retirar ya del sitio, cuál deben subir y cuál deben actualizar. A pesar del trabajo que implica la gestión muchos estudiantes no se enteran de la información.

En este proyecto terminal se desarrolló un sistema para gestionar información relacionada con trámites de tipo académicos y administrativos, eventos deportivos y culturales, así como artículos del Reglamento de Estudios Superiores de la UAM; en este sistema el administrador únicamente se encarga de alimentar la base de datos con la información correspondiente a un conjunto de anuncios; posteriormente, el sistema se encargará de clasificar, indexar, procesar y difundir los anuncios, utilizando un canal RSS, al que los estudiantes deberán suscribirse para poder recibirlos.

### **1.1.Objetivos**

#### **1.1.1. Objetivo general**

Diseñar e implementar un sistema gestor de información para difundir mediante un canal RSS noticias sobre los trámites académicos y administrativos; así como los eventos deportivos y culturales que se llevan a cabo en la UAM Azcapotzalco.

#### **1.1.2. Objetivos particulares**

- Construir una agenda que contenga los trámites académico/administrativos más importantes que pueden realizar los estudiantes de la UAM Azcapotzalco durante el trimestre.
- Diseñar e implementar la base de datos relacional que alimentará al sistema gestor de información.

- Diseñar e implementar un módulo que permita cargar a la base de datos la información que será difundida.
- Diseñar e implementar un módulo que realice procesos de indexación y clasificación a la información almacenada en la base de datos.
- Diseñar el formato de archivo basado en XML que recoja la información publicada por el sistema gestor.
- Diseñar e implementar un módulo que determine qué información debe ser eliminada, mantenida o publicada para que posteriormente sea difundida mediante un canal RSS.

## 2. Antecedentes

A continuación se hace referencia a trabajos realizados dentro y fuera de la UAM.

Referencias internas:

- “Gestor de contenidos web” propuesto en el trimestre 10-I, es un proyecto terminal en el cual se implementa un sistema web en el que se pueden editar y publicar páginas web, subir archivos y gestionar cuentas de usuarios, todo esto lo puede realizar uno o varios usuarios/administradores.

La diferencia entre este proyecto y el que se propone, reside en la forma en que se gestiona la información. El sistema web a desarrollar solo necesitará la intervención de un “Administrador” al momento de almacenar información en la base de datos, ya que posteriormente el sistema se encargará de hacer la gestión y en base a algunos parámetros decidirá cuándo difundirla.

- “Lista de RSS de la Unidad Azcapotzalco”, es una lista de 29 fuentes RSS empleadas por la UAM Azcapotzalco para dar a conocer a la comunidad universitaria los eventos: semanales, deportivos, institucionales, así como becas, conferencias, congresos, convocatorias, cursos, etc.

La diferencia entre esta herramienta y el proyecto que se propone, yace en que la lista de fuentes RSS es administrada por un gestor de canales RSS, y éste a su vez es gestionado por un administrador que también se encarga de gestionar las noticias a difundir. El proyecto que se propone solo necesitará un administrador para alimentar a la base de datos y el sistema se encargará de gestionar la información y decidir cuándo difundirla.

Referencias externas: Durante la investigación, el hecho de utilizar el nombre de “Sistema gestor de información” dio varios resultados, a continuación un par de ejemplos.

- Drupal™, es una plataforma gestora de contenido de código abierto, con la cual se pueden administrar usuarios, documentos y construir aplicaciones empresariales añadiendo módulos para ampliar y hacer más robusto el sistema.
- Joomla!, es un sistema gestor de contenidos utilizado para crear, construir y editar una variedad de sitios web y aplicaciones.

La diferencia entre los proyectos antes mencionados y el proyecto a desarrollar, reside en que los primeros proveen una serie de herramientas para la edición del contenido web, y el administrador es el encargado de gestionar la información almacenada en el sistema; en cambio en el proyecto que se propone, el administrador solo proveerá de información a la base de datos mediante archivos con un formato específico o mediante una interfaz gráfica y el sistema gestor se encargará de procesarla, indexarla, clasificarla y hacerla pública cuando sea necesario para que posteriormente sea difundida utilizando la tecnología RSS.

### **3. Justificación**

La UAM Azcapotzalco utiliza distintos medios de difusión (impresos, web, etc.), para dar a conocer las actividades que se llevan a cabo en la unidad; a pesar de esto, algunos estudiantes no se enteran de las noticias que la unidad difunde debido a que no consultan el correo institucional, no leen los medios impresos, y consultan con poca frecuencia la página web de la unidad.

Un problema importante surge cuando los alumnos de nuevo ingreso no se enteran de las actividades relevantes que semana a semana son llevadas a cabo en la Universidad como son: altas, bajas y cambios en la primer semana; renunciadas a alguna Unidad de Enseñanza–Aprendizaje (UEA) en la 5ª semana, etc. Es muy importante que los estudiantes conozcan el Reglamento de Estudios Superiores (RES) de la UAM, ya que en él se estipulan una serie de artículos que describen los derechos y obligaciones que tienen los estudiantes de esta universidad. Por ejemplo, en el artículo 25 se estipula que los alumnos tienen derecho a renunciar a una UEA; éste artículo es de suma importancia para los estudiantes, por lo que todos deben conocerlo; debido a la falta de un medio de difusión lo suficientemente atractivo que permita dar a conocer los artículos del RES, algunos estudiantes desconocen los derechos y obligaciones que deben cumplir como alumnos de la UAM Azcapotzalco.

La importancia de este proyecto reside en que se desarrollará un sistema que automatizará los procesos de clasificación, indexación, publicación y eliminación de información (a partir de ahora llamaremos *gestión* a los procesos antes mencionados); los administradores de sitios web de la UAM Azcapotzalco sólo se encargarán de proporcionar la información al sistema y éste se ocupará de los procesos de gestión correspondientes, por lo que no deberán preocuparse por la frecuencia con la que deben publicar información o cómo será

visualizada, ya que el *lector RSS* que empleen los estudiantes para suscribirse al canal de noticias le dará el formato para la visualización. Aunado a esto se aumenta la probabilidad de que muchos más estudiantes se enteren de lo que está sucediendo o lo que sucederá en la universidad gracias a la difusión RSS. Por ejemplo, en la 5ª semana un alumno conecta su celular, laptop o cualquier otro dispositivo (deberá contar con un lector RSS) a Internet, éste recibirá la noticia donde se le informa que puede renunciar a una UEA y se le enunciará el artículo donde se describe este derecho.

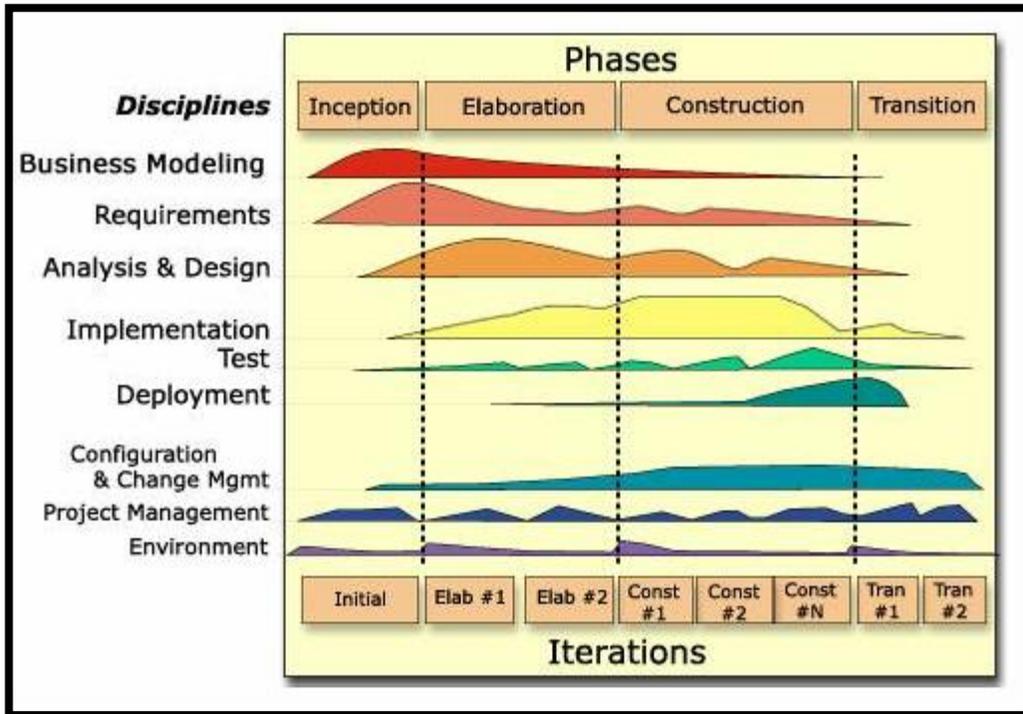
#### **4. Metodología de base empleada en el proyecto**

Para el desarrollo de este proyecto terminal se utilizó el proceso de desarrollo de software denominado *Proceso Unificado (UP)*, que se caracteriza por estar dirigido por casos de uso, por estar centrado en la arquitectura, y por ser un modelo iterativo e incremental.

Un proyecto UP organiza el trabajo y las iteraciones en cuatro fases fundamentales [1]:

1. **Inicio:** Visión aproximada, análisis del negocio, alcance, estimaciones imprecisas.
2. **Elaboración:** Visión refinada, implementación iterativa del núcleo central de la arquitectura, detección de los riesgos del proyecto, identificación de otros requisitos y alcance, estimaciones más realistas.
3. **Construcción:** Implementación iterativa del resto de los requisitos y de los módulos que se comprenden mejor; realización de pruebas de unidad sobre los módulos implementados; así como la preparación para el despliegue.
4. **Transición:** Pruebas de integración y de despliegue.

En la Figura 1 se ilustran las etapas del Proceso Unificado (UP).



**Figura 1: Disciplinas del UP [2]**

Es recomendable que cada iteración dure entre dos y seis semanas, esto con el propósito de obtener una rápida retroalimentación y así disminuir los riesgos del proyecto. Un riesgo del proyecto puede ser que un requerimiento del cliente se modifique; si la iteración sobre el requerimiento original dura diez semanas y al final se modifica, esto puede implicar un esfuerzo adicional para realizar la modificación.

Algunos de los beneficios del desarrollo iterativo son:

- Mitigación tan pronto como sea posible de riesgos a nivel técnico, de requisitos, de objetivos, y demás.
- Progreso visible en las primeras etapas.
- Una temprana retroalimentación, compromiso y adaptación de los usuarios al nuevo sistema; esto nos lleva a un sistema refinado que se ajusta más a las necesidades reales del personal involucrado.
- Gestión de la complejidad, es decir, al desarrollar el proyecto basado en un modelo iterativo, se pueden definir fases de desarrollo, y en cada fase implementar un módulo del sistema.

- El conocimiento adquirido en una iteración se puede utilizar para mejorar el propio proceso de desarrollo, iteración a iteración.

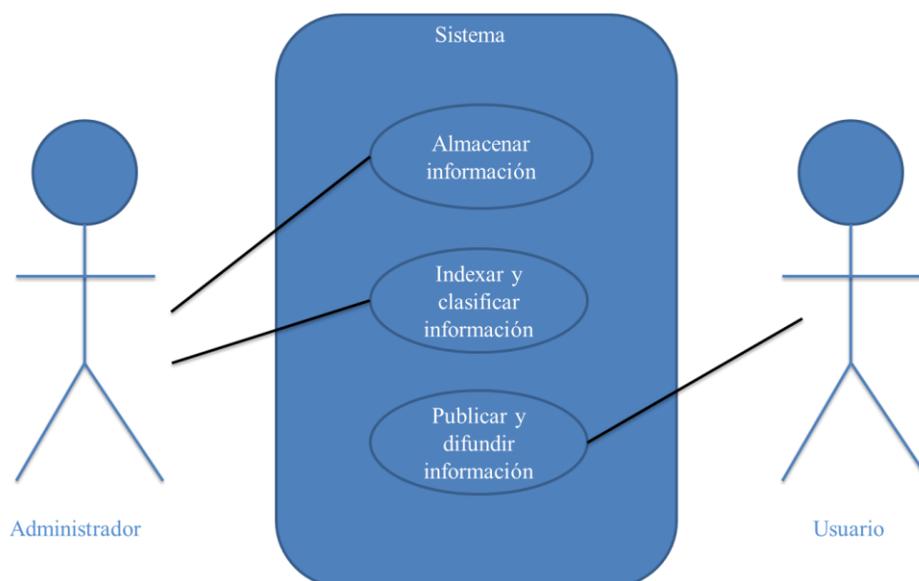
## 5. Desarrollo del proyecto

La sección 5.1 presenta parte del diseño del *sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS*. Se detalla la especificación de algunos módulos, a saber: clasificación y categorización de los anuncios, los cuales son esenciales en el sistema. La sección 5.2 presenta algunos detalles de implementación, poniendo énfasis en los mismos módulos; así mismo se expone la tecnología empleada y su justificación.

### 5.1. Diseño

A continuación se presentan los artefactos siguientes: casos de uso, la arquitectura del sistema, el diseño de la base de datos y el diagrama de clases; explicando con detalle la especificación de cada módulo del sistema.

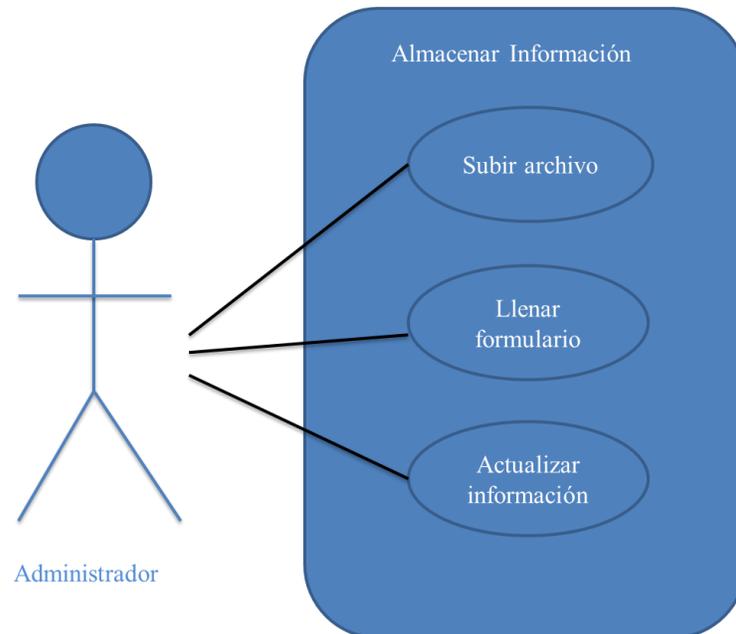
#### 5.1.1. Diagrama de casos de uso general



**Figura 2: Diagrama de casos de uso general**

En la figura 2, se presenta el diagrama general de casos de uso del proyecto, en él se muestran los tres módulos que conforman el sistema. El sistema está conformado por tres módulos generales: el primero está encargado de recoger los anuncios proporcionados por el usuario y almacenarlos en la base de datos; el segundo módulo clasifica y cataloga los anuncios; y el tercero está encargado de la difusión de los anuncios. Cada caso de uso general tiene, a su vez, tres casos de uso específicos que a continuación se presentan:

### 5.1.2. Caso de uso: Almacenar información



**Figura 3: Caso de uso Almacenar Información**

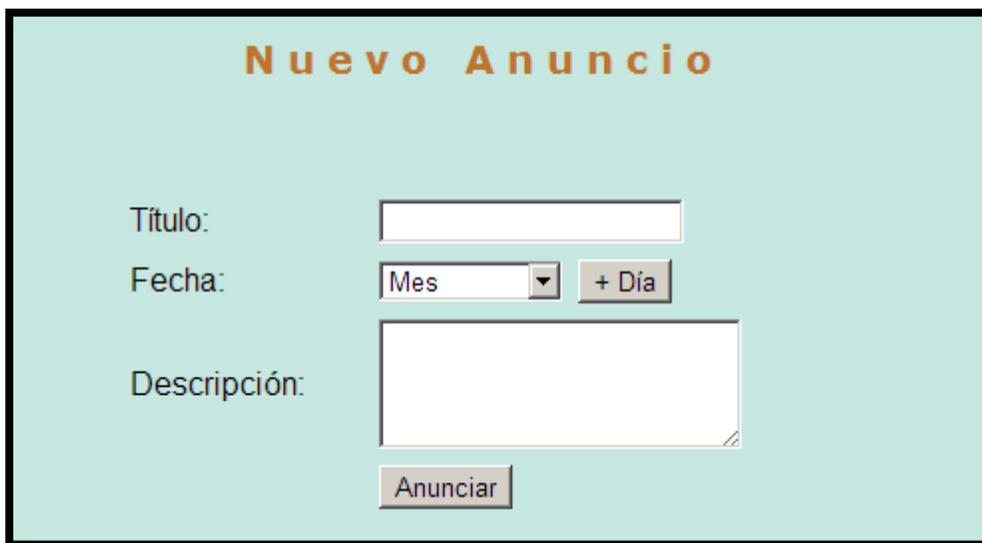
En la figura 3 se muestra el caso de uso: *Almacenar Información*, así como los casos de uso específicos que lo conforman:

- *Subir archivo*, cuya interfaz gráfica se muestra en la figura 5, es un módulo que permite cargar archivos con extensión *.txt*.
- *Llenar formulario*, permite proporcionar anuncios al sistema de manera individual, tal y como se muestra en la figura 4.
- *Actualizar información*, permite actualizar los anuncios que no lograron ser catalogados en el sistema y permite definir una categoría para estos.

El caso de uso general *Almacenar información*, describe el *Módulo 1* del sistema. El módulo 1 es el único que interactúa con el administrador para el ingreso de los anuncios, posteriormente la información es almacenada en una base de datos.

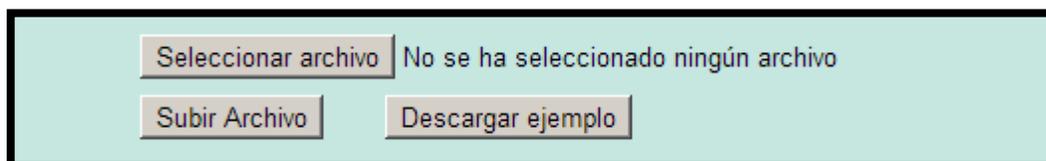
El módulo 1 cuenta con una interfaz gráfica donde se solicita al usuario un archivo con los anuncios a almacenar; también cuenta con un formulario donde se puede agregar uno a uno cada elemento, a saber: título, cuerpo y fecha, que conforma un anuncio de tipo trámite, evento ó artículo.

En las Figura 4 y 5, se muestran las capturas de pantalla del formulario *Nuevo Anuncio* y la interfaz utilizada para cargar un archivo.



**Figura 4: Formulario para agregar un nuevo anuncio.**

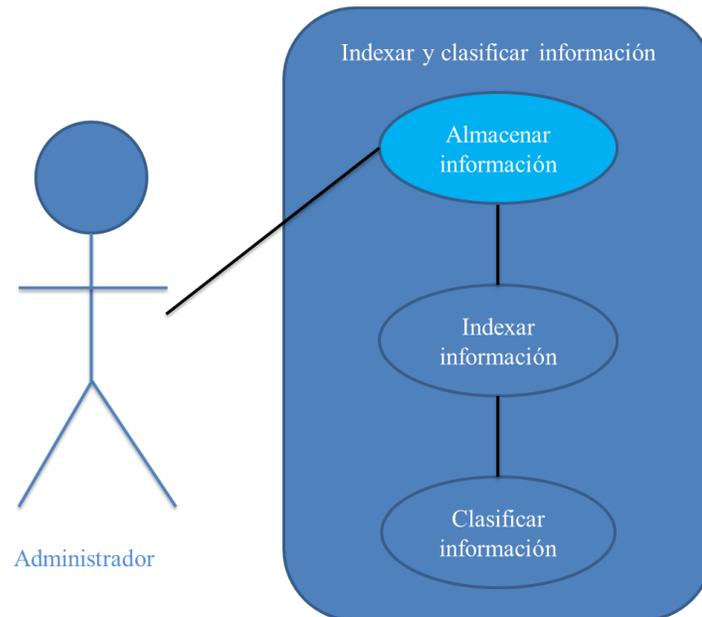
La figura 4 muestra la interfaz gráfica del sistema encargada de recoger la información asociada a un anuncio, en esta ventana el administrador debe proporcionar el título del anuncio que desea difundir, así como la fecha en la que desea que sea difundido y una breve descripción del anuncio, de no más de 280 caracteres.



**Figura 5: Sección para seleccionar un archivo y subirlo.**

En la figura 5 se muestra la parte complementaria del módulo 1, esta parte es la encargada de solicitar un archivo con extensión *.txt*, este archivo debe contener una relación de los anuncios que se desean difundir y debe contener: título, cuerpo y fecha del anuncio.

### 5.1.3. Caso de uso: Indexar y Clasificar Información



**Figura 6: Caso de uso Indexar y Clasificar Información**

En la figura 6 se muestra el caso de uso general: *Indexar y Clasificar Información*, así como los casos de uso específicos que lo conforman:

- *Indexar información*, se encarga de ordenar cronológicamente los anuncios, en función de la fecha de difusión programada para cada uno de ellos.
- *Clasificar información*, se encarga de clasificar los anuncios a partir de su fecha de realización y cataloga los anuncios en base a las palabras clave que contenga, a saber: artículos, trámites académicos, eventos deportivos y eventos culturales.

El caso de uso general *Indexar y Clasificar Información* describe el funcionamiento del *Módulo 2* del sistema; este módulo se encarga clasificar y catalogar los anuncios almacenados en la base de datos, los anuncios son clasificados en anuncios actuales (eventos que ocurrirán en la semana en curso) y anuncios futuros (eventos que ocurrirán posteriormente a la semana en curso). Posteriormente el anuncio es catalogado en una de las siguientes categorías: Artículo, trámite académico/administrativo, actividad deportiva, actividad cultural y evento académico.

El sistema clasifica los anuncios tomando en cuenta la fecha que el administrador proporciona para su difusión; para el caso de los artículos, estos pueden tener o no tener fecha de difusión, en caso de que el administrador no proporcione la fecha de difusión, el sistema le asigna una fecha aleatoria.

Una vez que los anuncios son clasificados y catalogados, son publicados por el sistema, posteriormente son difundidos utilizando un canal RSS, según su fecha de ocurrencia. Este módulo general *Indexar y Clasificar Información* no cuenta con interfaz gráfica.

**Función** Clasifica\_Anuncio (*fecha*)

//*fecha* es un tipo de dato String que contiene la fecha en la que será difundido el anuncio.

*fechaSistema* ← tomaFechaDeSistema();

**Si** *fechaSistema* y *fecha* se ubican en semanas diferentes

**Entonces**

**Regresa** *anuncioFuturo*;

**De lo contrario**

**Regresa** *anuncioActual*;

**Fin**

**Pseudocódigo 1: Algoritmo para clasificar anuncios que no son artículos.**

El pseudocódigo 1 muestra el algoritmo para clasificar anuncios que no son artículos. Los anuncios que son artículos son detectados, se les asigna fecha aleatoria y se catalogan.

En el anexo A se muestra el código fuente del método *clasificaAnuncio*.

**Función** *Cataloga\_Anuncio (texto)*

*//texto* es un tipo de dato String que contiene el título o la descripción del anuncio a difundir.

**Si** *texto* empieza con “artículo”

**Entonces**

**Regresa** *esArticulo*;

**De lo contrario**

*vector* ← **devideCadena**(*texto*);

*vector* ← **eliminaArticulos**(*vector*); *//Artículo*: Precisa la referencia de un sustantivo.

*//El argumento determina la categoría de las palabras a extraer de la base de datos.*

*vector2* ← **extraerPalabrasDiccionario**(*tramite*);

*coincidenciasTramite* ← **cuentaCoincidencias**(*vector*, *vector2*);

*vector2* ← **extraerPalabrasDiccionario**(*cultural*);

*coincidenciasCultural* ← **cuentaCoincidencias**(*vector*, *vector2*);

*vector2* ← **extraerPalabrasDiccionario**(*deporte*);

*coincidenciasDeporte* ← **cuentaCoincidencias**(*vector*, *vector2*);

*vector2* ← **extraerPalabrasDiccionario**(*academico*);

*coincidenciasAcademico* ← **cuentaCoincidencias**(*vector*, *vector2*);

*//maximo* es la variable que aloja la categoría que tuvo más coincidencias al ser comparado el texto con las palabras del diccionario.

*maximo* ← **calculaMaximo**(*coincidenciasTramite*, *coincidenciasCultural*, *coincidenciasDeporte*, *coincidenciasAcademico*);

**Si** *maximo* == *coincidenciasTramite*

**Entonces**

**Regresa** *esTramite*

**Si** *maximo* == *conicidenciasCultural*

**Entonces**

**Regresa** *esCultural*

**Si** *maximo* == *conicidenciasDeporte*

**Entonces**

**Regresa** *esDeportivo*

**Si** *maximo* == *conicidenciasAcademico*

**Entonces**

**Regresa** *esAcademico*

**Fin**

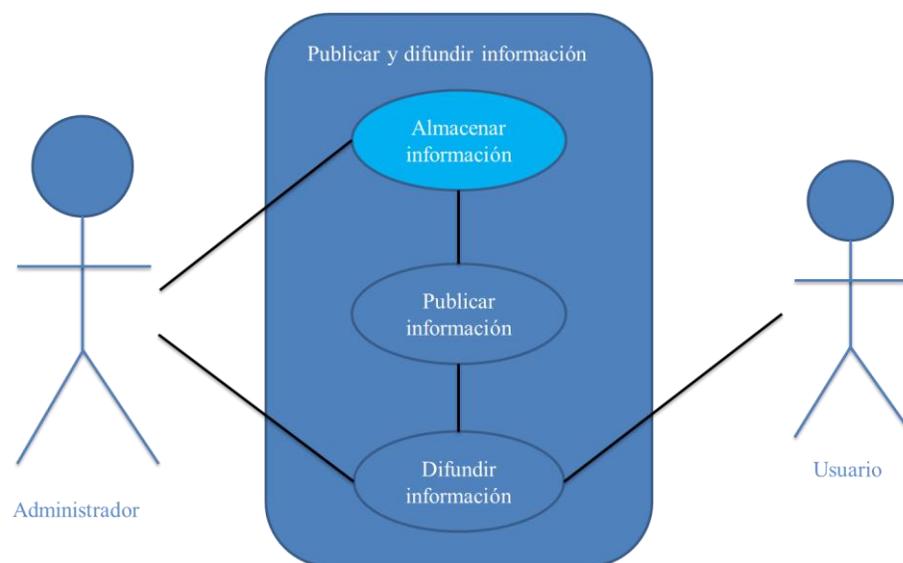
**Pseudocódigo 2: Algoritmo para catalogar anuncios.**

El pseudocódigo 2 muestra el algoritmo para catalogar los anuncios almacenados en la base de datos; el algoritmo recibe una cadena de texto que en primer lugar es el título del anuncio; el algoritmo lee la primera palabra de la cadena, si la primera palabra es igual a "artículo", automáticamente el algoritmo determina que el anuncio es un artículo.

El algoritmo divide la cadena en *tokens* (un vector de palabras), a este vector de palabras lo somete a un proceso en el cual se eliminan los artículos del lenguaje (el, la, los, etc.), esto con el fin de disminuir el tamaño del vector. Una vez hecho lo anterior, se procede a extraer de la base de datos las palabras del diccionario correspondientes a cada categoría (una categoría a la vez), éstas se almacenan en un vector. Posterior a esto, se compara el vector de las palabras del texto contra el vector con las palabras del diccionario, y se buscan coincidencias entre las palabras de ambos vectores. Para cada coincidencia de palabras, un contador va aumentando, al final de las comparaciones, se almacena el contador con el número de coincidencias.

El proceso de comparación se realiza para cada categoría, una vez que se tienen los números de coincidencias de cada categoría almacenados en variables, se calcula el máximo entre éstas, y así la categoría con mayor número de coincidencias, es la categoría a la cual pertenece el anuncio. Como se mencionó anteriormente, primero se aplica el proceso al título del anuncio, si con el título no es suficiente para categorizar el anuncio, se procede a procesar la descripción del anuncio. En el anexo B se muestra el código fuente del método *catalogaAnuncio*.

#### 5.1.4. Caso de uso: Publicar y Difundir Información



**Figura 7: Caso de uso Publicar y Difundir Información**

En la figura 7 se muestra el caso de uso general: *Publicar y Difundir Información*, así como los casos de uso específicos que lo conforman:

- *Publicar información*, se encarga de mostrar en la página web del administrador, los anuncios que ya han sido clasificados y catalogados.
- *Difundir información*, se encarga de difundir los anuncios utilizando un canal RSS para que éstos lleguen a la comunidad universitaria.

El caso de uso *Publicar y Difundir Información* describe el funcionamiento del *Módulo 3* del sistema. El módulo 3 se encarga de publicar la información clasificada y almacenada en la base de datos en la página web del administrador. Posteriormente la información es difundida mediante un canal RSS, al que los alumnos pueden suscribirse (vía Internet) y recibir la información en su *smartphone*, *tablet* o computadora [3].

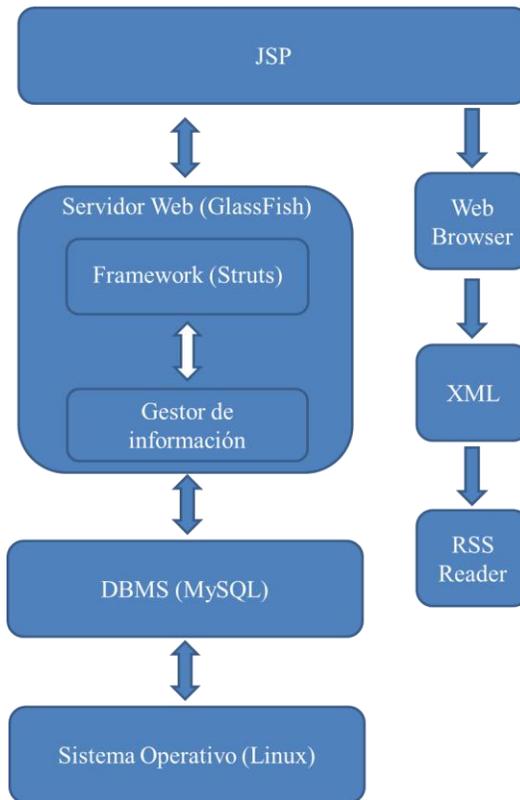
En el módulo 3 también se gestiona la información para determinar, qué información permanece en la base de datos y qué información se elimina. Los anuncios que permanecen en la base de datos son: Artículos y Trámites académicos/administrativos porque deben ser difundidos cada trimestre a la comunidad universitaria, con el propósito de que los estudiantes conozcan o recuerden sus derechos y obligaciones. Por ejemplo el proceso de altas, bajas y cambios en la primera semana, o el proceso de renunciaciones en la quinta semana. Los anuncios son difundidos de la siguiente manera:

- Se difunden 3 días antes del evento.
- Un día antes del evento.
- Y el día del evento.

Posteriormente la información que no está catalogada como artículo o trámite es eliminada de la base de datos. Por ejemplo: los eventos deportivos o culturales. Estos anuncios son eliminados de la base de datos, porque son actividades o eventos que solo tienen lugar una sola vez; por ejemplo, una conferencia (evento académico) o un partido de fútbol (evento deportivo), son eventos o actividades que solo ocurren una vez y no se repiten al siguiente trimestre, al menos no la misma actividad como tal, por esta razón no se considera importante mantenerlo en la base de datos.

Aunado a esto, con una buena gestión de información, se puede ahorrar espacio de almacenamiento, manteniendo en la base de datos solo aquellos anuncios que son constantes en su ocurrencia, cada trimestre, como son: artículos del reglamento y trámites académico-administrativos.

### 5.1.5. Arquitectura general del sistema



**Figura 8: Arquitectura general del sistema**

La figura 8 muestra la arquitectura general del sistema, el sistema está conformado por una base de datos (MySQL), un servidor de aplicaciones web (GlassFish) que corre sobre un servidor con sistema operativo Linux. Se utilizan páginas JSP para recibir y mostrar información.

Posteriormente es generado un archivo XML con los anuncios para que puedan ser difundidos por el canal RSS. El sistema está programado para difundir los anuncios cuando se acerca su fecha de ocurrencia, una vez que se cumple esta condición, el sistema selecciona de la base de datos los anuncios que debe difundir y posteriormente los imprime en un archivo con extensión .xml, en el anexo C, se muestra la estructura básica de un archivo XML utilizado para la difusión RSS, así como la clase que hace posible generar este archivo.

### 5.1.6. Estructura de la base de datos

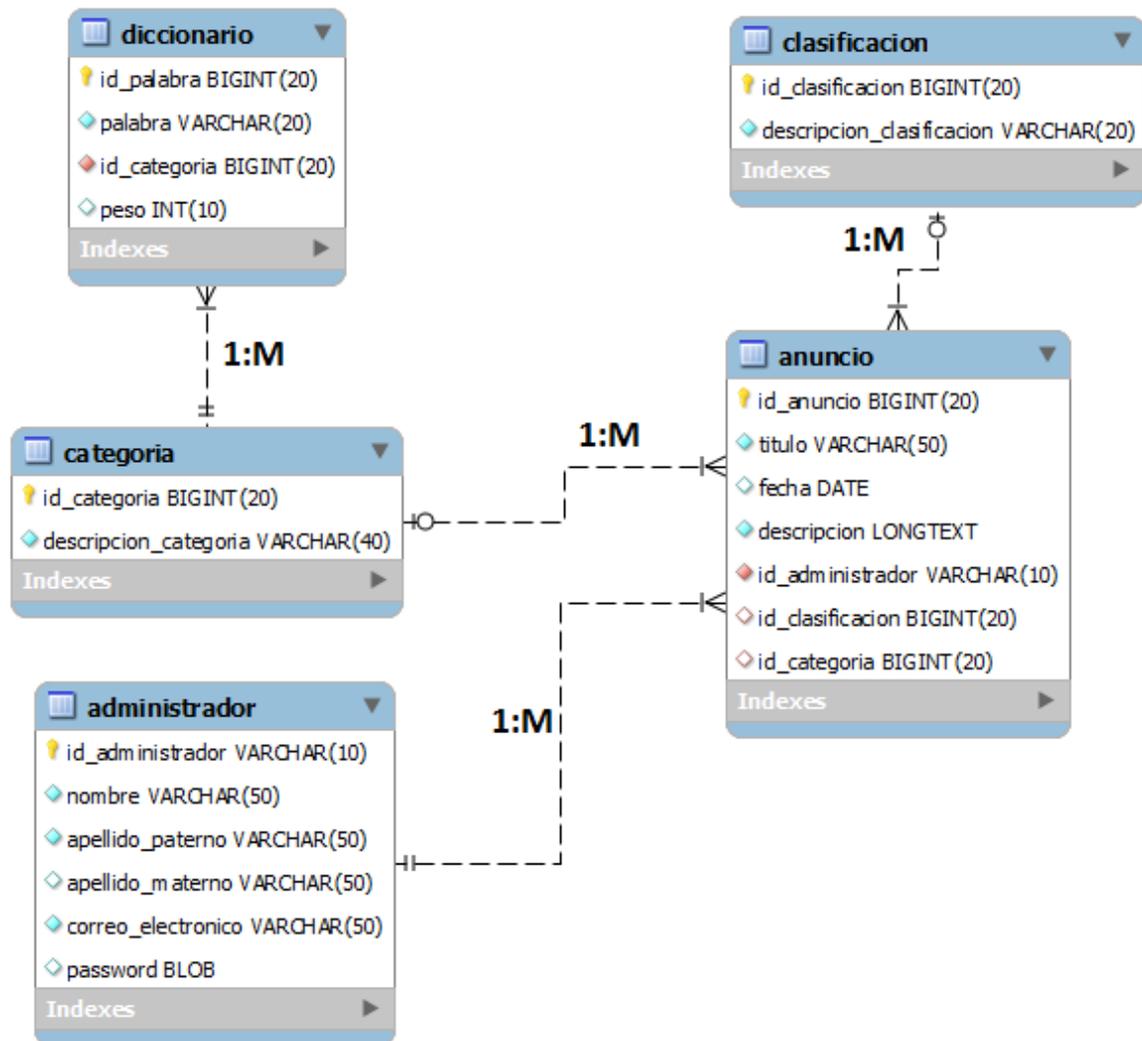


Figura 9: Estructura de la base de datos

En la figura 9 se muestra la estructura de la base de datos del sistema, en la base de datos se almacenan los anuncios y toda la información asociada a estos.

#### Tabla diccionario

- *id\_palabra*: identificador único para cada palabra del diccionario.
- *palabra*: palabra del diccionario.

- *id\_categoria*: identificador de la categoría a la cual pertenece la palabra del diccionario.
- *peso*: valor de la palabra. Utilizado para categorizar los anuncios con mayor rapidez. Este campo puede tomar dos valores: uno ó dos; cuando el campo tiene peso uno, prácticamente cualquier anuncio que contenga una palabra determinada puede ser categorizado de inmediato en la categoría a la cual pertenece dicha palabra. Por ejemplo, en el anuncio con la siguiente descripción:

“Se llevará a cabo un partido de **fútbol** en la cancha empastada de la Universidad.”

Podemos decir que la palabra fútbol es una palabra con peso 1 ya que es una palabra que puede determinar con facilidad a un anuncio en la categoría de evento deportivo.

Cabe destacar que esta tabla está relacionada con la tabla categoría, ya que una palabra puede existir en la base de datos solo si ésta tiene una categoría.

#### Tabla categoría

- *id\_categoria*: identificador único para cada categoría de la base de datos.
- *descripcion\_categoria*: contiene el nombre de cada categoría.

En esta tabla se almacenan las categorías de los anuncios. Para el proyecto terminal se consideraron las siguientes categorías: artículo, trámite académico/administrativo, evento cultural, evento deportivo, evento académico.

#### Tabla clasificación

- *id\_clasificacion*: identificador único para cada clasificación.
- *descripcion\_clasificacion*: contiene el nombre de cada clasificación.

En esta tabla se almacenan los tipos de clasificación que puede tomar un anuncio, éstas son: información actual o información por utilizar. En información actual son clasificados los anuncios que contengan información acerca de los eventos que sucederán en la semana en curso. Por el contrario en información por utilizar están clasificados los anuncios que tienen información de eventos que sucederán una o varias semanas después de la semana en curso.

### Tabla administrador

- *id\_administrador*: identificador único para cada administrador.
- *nombre*: nombre de pila del administrador.
- *apellido\_paterno*: apellido paterno del administrador.
- *apellido\_materno*: apellido materno del administrador.
- *correo\_electronico*: correo electrónico asociado al administrador.
- *password*: contraseña utilizada para ingresar al sistema.

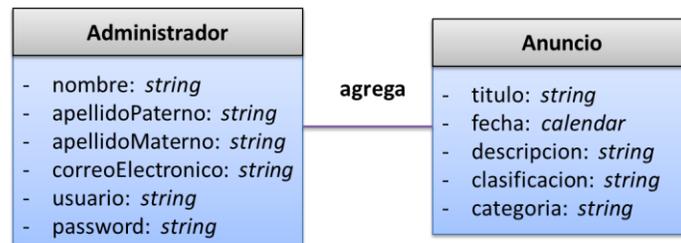
En esta tabla se almacena toda la información asociada al administrador del sistema. Pueden existir varios administradores del sistema, los usuarios serán creados por el administrador responsable y éste será el único que podrá autorizar y dar de alta nuevos usuarios. Por seguridad no se tiene un módulo en línea para agregar nuevos administradores. La tabla está relacionada directamente con la tabla de anuncios ya que no puede existir un anuncio a menos que haya sido proporcionado por un administrador, y éste este registrado en la base de datos.

### Tabla anuncio

- *id\_anuncio*: identificador único para cada anuncio.
- *titulo*: título del anuncio.
- *fecha*: fecha de difusión del anuncio.
- *descripcion*: descripción del anuncio.
- *id\_administrador*: identificador del administrador responsable del anuncio.
- *id\_clasificacion*: identificador de la clasificación a la cual pertenece el anuncio.
- *id\_categoria*: identificador de la categoría a la cual pertenece el anuncio.

En esta tabla se almacena toda la información asociada a los anuncios, está relacionada con la tabla categoría, la tabla administrador y con la tabla clasificación, esto porque el anuncio debe tener los identificadores de: quién agregó el anuncio, a qué categoría pertenece y cuál es su clasificación.

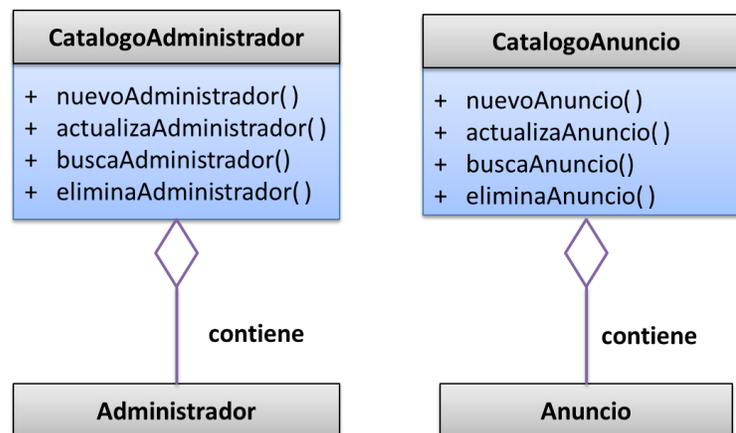
### 5.1.7. Clases del sistema



**Figura 10: Clases contenedoras**

Las clases de Anuncio y Administrador (clases contenedoras Figura 10), contienen los atributos y los *getters* y *setters* utilizados para obtener y modificar la información proporcionada por el usuario del sistema.

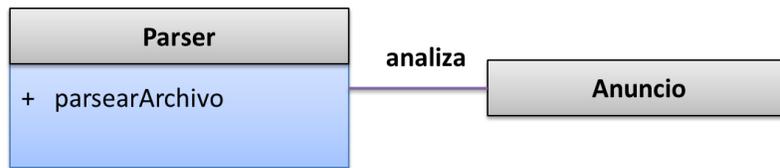
Las clases de tipo catálogo contienen los métodos para realizar operaciones básicas de adición, eliminación y actualización de anuncios y administradores (ver figura 11). Estas clases contienen las consultas SQL para explotar la base de datos (ver anexo D).



**Figura 11: Clases catálogos**

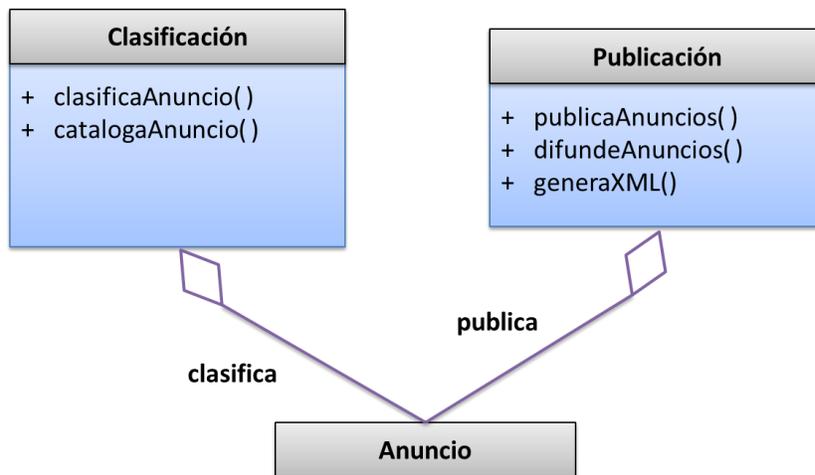
La clase *CatalogoAnuncio* está relacionada con la clase Anuncio, ya que en la clase *CatalogoAnuncio* se instancia la clase Anuncio para agregar nuevos objetos a la base de datos con los atributos de dicha clase.

Algo similar pasa con la clase *CatalogoAdministrador*, que instancia a la clase Administrador, para agregar objetos a la base de datos con los atributos de esta última.



**Figura 12: Clase Parser**

La clase *Parser* (ver figura 12) contiene el método encargado de separar en *tokens* el contenido del archivo proporcionado por el usuario, para conformar la estructura de cada anuncio y proceder a catalogar, clasificar y almacenar la información en la base de datos.



**Figura 13: Clases controladoras**

Las clases de *Publicación* y *Clasificación* (ver figura 13) son utilizadas para realizar las operaciones de clasificación (*código fuente: ver anexo A*) y categorización (*código fuente: ver anexo B*) de los anuncios proporcionados por el usuario, para posteriormente publicarlos en la página web del administrador y así difundirlos.

## **5.2.Implementación**

### *5.2.1. Hardware*

- Notebook HP Pavilion dv4-1224la con procesador Intel(R) Pentium(R) Dual CPU T3400 2.17GHz, Memoria RAM de 3072 MB y Disco duro de 250GB.
- Servidor HP Proliant.

### *5.2.2. Software*

- Sistema operativo GNU/Linux.
- NetBeans IDE: Entorno de desarrollo integrado de código abierto.
- MySQL: Manejador de bases de datos de código abierto [4].
- GlassFish: Servidor de aplicaciones de código abierto [5].
- Apache Struts: Framework web de código abierto.

El software utilizado para el desarrollo del proyecto es de código abierto, evitando así problemas de licenciamiento.

Ubuntu (el sistema operativo en el que se desarrolló el proyecto) es un sistema estable, y cuenta con diversas herramientas que facilitan el desarrollo de software.

NetBeans es un entorno de desarrollo integrado muy completo; cuenta con depurador que se encarga, incluso, de revisar el correcto funcionamiento de páginas JSP. Aunado a esto, el IDE facilita el desarrollo al tener disponibles frameworks como Struts, así como los controladores para manejadores de bases de datos.

MySQL es un sistema gestor de bases de datos muy estable y podemos encontrar documentación en su página web oficial, esa fue una de las principales razones por las que se decidió utilizar este manejador de bases de datos.

Se utilizó Apache Struts en su versión 1.3.10, ya que es la versión más estable; además, la documentación se encuentra en la página Web de Apache. Struts es una framework para el desarrollo de aplicaciones Web basados en el patrón de diseño MVC y empleando la plataforma Java EE.

RssDemon se utiliza en el dispositivo móvil como el software predeterminado para la lectura de la fuente RSS del canal del sistema. Se utilizó este software por su característica de permitir leer fuentes RSS aún cuando el dispositivo está en modo offline. En efecto, las fuentes se descargan al móvil cuando el teléfono está conectado a internet; en un momento posterior, el usuario puede leer las fuentes aunque ya no esté conectado a internet.

### 5.2.3. Funcionalidad



The image shows a login interface with a light green background. At the top, the text '¿Deseas difundir anuncios?' is displayed in a bold, orange font. Below it, 'Inicia Sesión' is written in a similar font. The main area contains two input fields: 'Usuario:' and 'Contraseña:'. Below the password field is a button labeled 'Entrar'.

**Figura 14: Inicio de sesión del sistema**

Para poder ingresar al sistema, el administrador requiere de un usuario y una contraseña (ver figura 14).

Cabe destacar que las contraseñas se almacenan y protegen en la base de datos con un esquema de cifrado AES Advanced Encryption Standard, un estándar de cifrado simétrico dispuesto por el NIST (National Institute of Standards and Technology).

Las principales características del esquema de cifrado AES son su fácil diseño, su versatilidad en ser implementado en diferentes dispositivos, así como ser inmune a los ataques conocidos hasta la fecha; por ejemplo: ataques a la estructura matemática del cifrador, ataques para recuperar claves por fuerza bruta, etc. Soporta bloques de datos de 128 bits y claves de 128, 192, y 256 bits [6].

Una vez que se ingresan el nombre de usuario y la contraseña correcta, se puede visualizar la siguiente pantalla.

Anuncios sin clasificar   Agregar palabras   Ver anuncios   Cerrar sesión

## Nuevo Anuncio

Título:

Fecha:

Descripción:

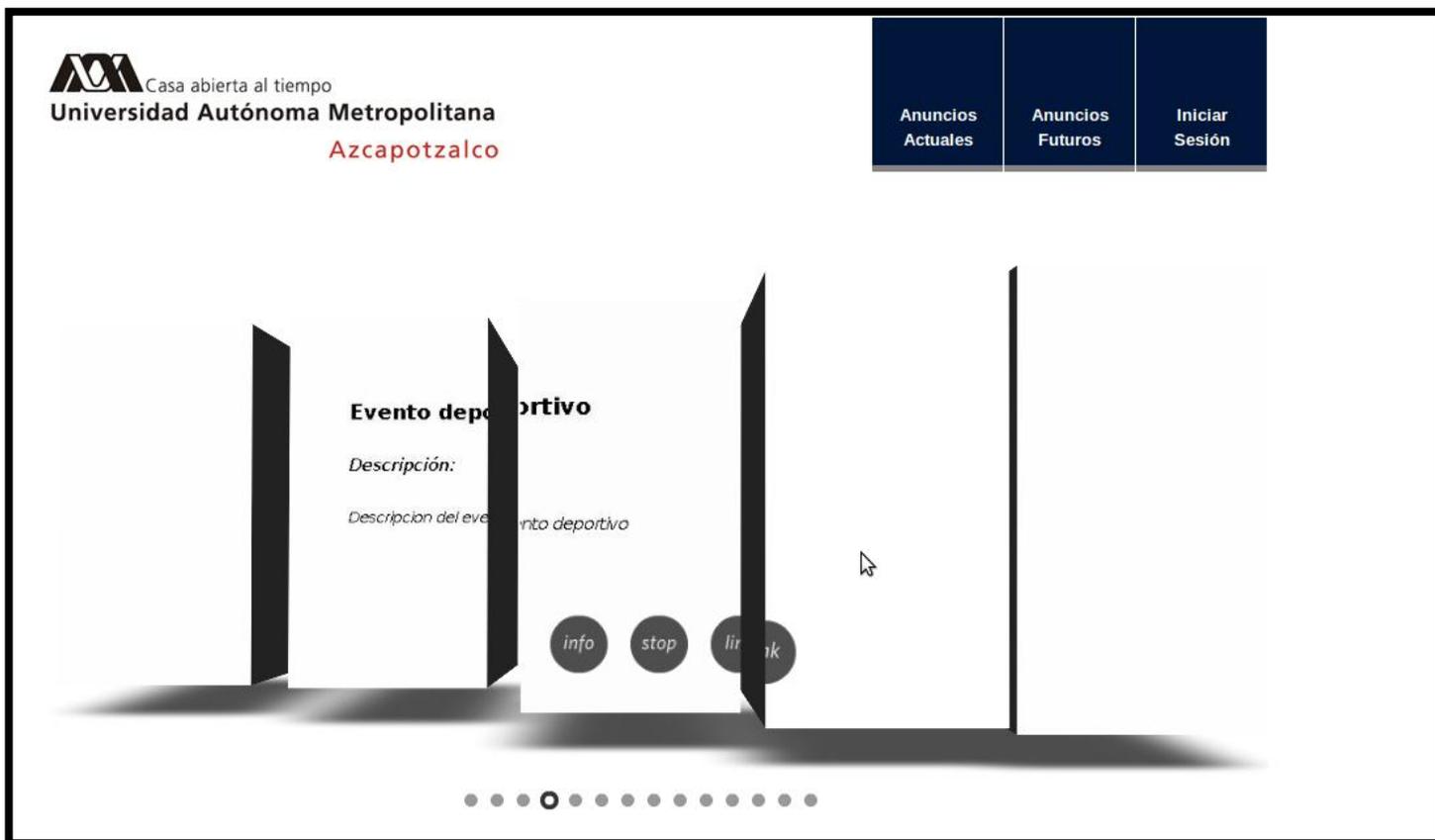
No se eligió ningún archivo

**Figura 15: Página inicial**

En la figura 15 se muestra la pantalla inicial del sistema, sólo se puede acceder a esta pantalla una vez que el administrador inició sesión. La pantalla muestra un área donde el administrador puede agregar anuncios de manera individual; cuando el número de anuncios es grande, el administrador puede optar por agregar los anuncios de manera conjunta en un archivo con extensión *.txt*.

Los anuncios contenidos en el archivo deben llevar un formato específico para que puedan ser almacenados por el sistema, a saber: título, fecha y descripción de no más de 280 caracteres. Para que el administrador no tenga ningún problema al momento de añadir anuncios al archivo, el sistema cuenta con un ejemplo descargable, de esta manera el administrador puede utilizarlo como plantilla.

En esta pantalla el administrador tiene la opción de "Ver anuncios", este botón lo guía a la sección donde se muestra el resultado de la clasificación y categorización de los anuncios (ver figura 16).



**Figura 16: Página que muestra el resultado de la clasificación y categorización de anuncios**

En esta página el administrador puede observar el resultado de la clasificación y categorización de los anuncios, así como el nombre y la descripción del anuncio.

En la página inicial del administrador (ver figura 15), también podemos encontrar el botón “Agregar palabras”, este botón nos guía a un módulo en el cual se pueden agregar nuevas palabras acompañadas de su categoría, al diccionario de la base de datos.

Otro botón que podemos encontrar en la página inicial es “Anuncios sin clasificar”, este botón nos guía a un módulo donde se muestran todos los anuncios que no pudieron ser clasificados por el sistema. En este módulo el administrador puede asignarles la categoría a la que pertenecen los anuncios, y así ser actualizados en la base de datos.

En las figuras 17 y 18 se muestran los dos módulos descritos anteriormente.

## Anuncios sin clasificar

Anuncio	Categoría	Acción
Semana de la salud	Categoría ▼	Listo
Terror en la UAM	Categoría ▼	Listo

**Figura 17: Módulo anuncios sin clasificar**

## Agrega una nueva palabra al diccionario

Palabra:

Categoría: Categoría ▼

**Figura 18: Módulo Nuevas palabras**

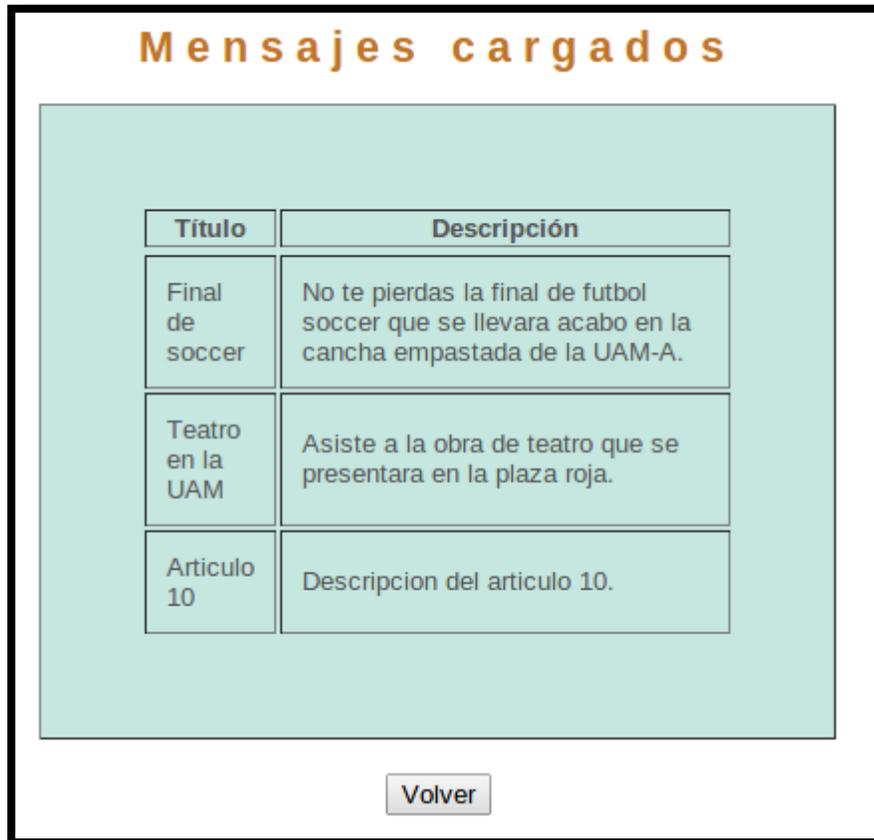
Como ya se mencionó anteriormente, la página inicial, cuenta con dos opciones para agregar anuncios, para el caso particular de agregar anuncios mediante un archivo, en caso de que el archivo seleccionado no sea de extensión *.txt* arrojará el mensaje “Solo se aceptan archivos con extensión *.txt*”, como se muestra en la figura 19.



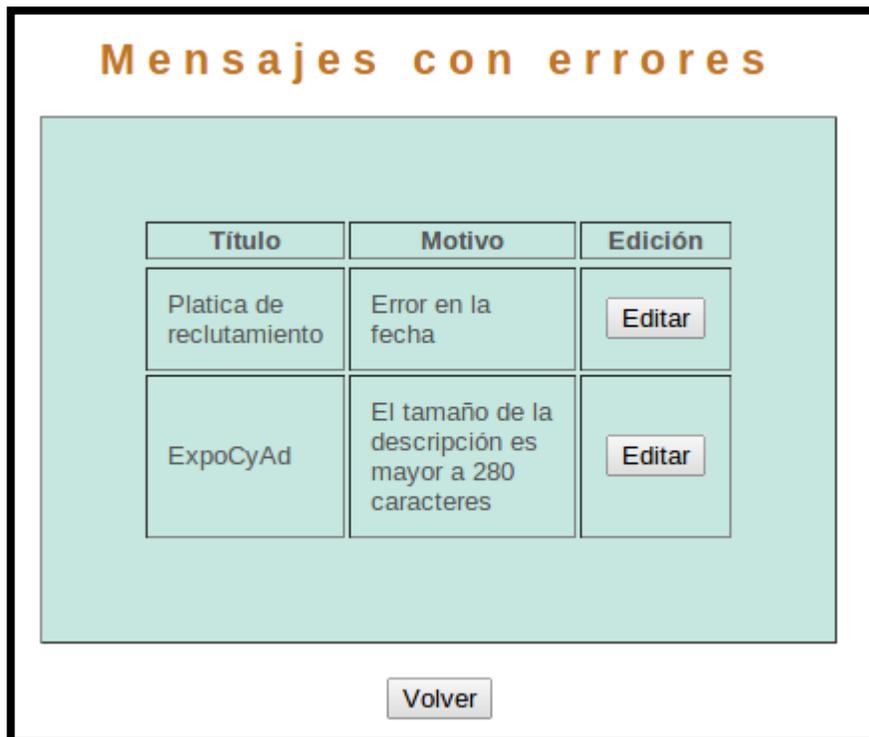
The image shows a web form titled "Nuevo Anuncio" on a light blue background. At the top, the title "Nuevo Anuncio" is displayed in a brown, serif font. Below the title, a red error message reads "Solo se aceptan archivos con extensión .txt". The form contains several input fields: a text box for "Título:", a date selector for "Fecha:" with a dropdown menu showing "Mes" and a "+ Día" button, and a larger text area for "Descripción:". Below these fields is an "Anunciar" button. At the bottom of the form, there are three buttons: "Seleccionar archivo" (which is currently disabled and shows the text "No se eligió ningún archivo"), "Subir Archivo", and "Descargar ejemplo".

**Figura 19: Error en la extensión del archivo**

Una vez que se ingresa un archivo con la extensión requerida y se hace el proceso de exploración del archivo para extraer los anuncios contenidos en él, el sistema guía al administrador a una página donde se muestran los anuncios procesados correctamente así como los anuncios que tuvieron problemas para ser almacenados acompañados del motivo del error. En la figura 20 y en la figura 21 se muestra la sección descrita anteriormente.



**Figura 20: Mensajes cargados con éxito**



**Figura 21: Mensajes con errores**

El contenido del archivo con el que se realizó la prueba y cuyo resultado del proceso se muestra en las figuras 20 y 21, es el siguiente:

```
//Nota: La fecha debe ir en el siguiente formato: yyyy-mm-dd
//Nota: La descripción del anuncio debe ser continua, es decir sin saltos de línea.

Final de soccer
2012-11-18
No te pierdas la final de futbol soccer que se llevara acabo en la cancha empastada de
la UAM-A.

Teatro en la UAM
2012-11-27
Asiste a la obra de teatro que se presentara en la plaza roja.

Articulo 10
Descripción del artículo 10.

Platica de reclutamiento
noviembre-10
Asiste a la charla por parte de reclutadores que quieren conocer el talento
universitario.

ExpoCyAd
2012-12-11
123456789101112131415161718192021222324252627282923031323334353637383
949414243444546474849505152309urx,,,,,,,,,,,,,,,,,,,,,,,,,,,,,2iewqooooooooooooo
jejdasdljiroejwskjvheriuhgjkdfbxcuigsebrkjewshduzsrxtcfrygtvubhjukldsfjgikbewd
sghrdsyhewdsuuhjkewasojkledasijklqwadscvbnaspoifudjksncbvsqwertyu832p45
'60398i1qytghwehdfsjfcxvnbxcvxxmxcv,.bnphogjpi69584376|1234567890'¿+poiuytr
ewqasdfghjklñ{}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}
_{}}{ñlkjhgfdsaaaaaaaaaaaaaaaaqwertyuiop++++¿'0987654321
```

Figura 22: Caso para prueba de con archivo.

El proceso es el siguiente. Cuando el administrador carga el archivo en el sistema, el sistema lo abre, y utiliza la clase *Parser* para explorar el contenido del archivo, y así determinar la estructura de cada anuncio y los errores, si es que contiene, los anuncios son almacenados en un vector dinámico. El resultado del proceso son dos vectores, un vector contiene la lista de los anuncios procesados satisfactoriamente, el otro vector contiene una lista de los anuncios que contienen errores.

El sistema llama al método *clasificaAnuncio* de la clase *Clasificacion*, para determinar en qué momento serán difundidos los anuncios. El sistema pasa como argumento la lista de anuncios sin errores. Después, el sistema llama al método *catalogaAnuncio*, de la misma clase, para determinar la categoría de cada anuncio. El sistema pasa como argumento al método, primero el título del anuncio, si el método no puede determinar la categoría del anuncio solo con el título del anuncio, el sistema pasa como argumento la descripción del anuncio.

Una vez realizados los procesos de clasificación y categorización, el sistema almacena en la base de datos los anuncios procesados satisfactoriamente. El sistema muestra al administrador una página en la que se listan los anuncios procesados satisfactoriamente, y los anuncios con errores, se muestra el motivo del error y se le da opción de editar el anuncio (ver figuras 20 y 21).

Una vez que se editan todos los anuncios con errores (opcional), el sistema llama al método *publicaAnuncios* de la clase *Publicacion*, el sistema pasa como parámetro la clasificación de los anuncios. El método recibe la clasificación de los anuncios para extraer de forma ordenada los anuncios de la base de datos y posteriormente publicarlos en la página web del administrador. Al momento de mostrar la información en la página web del administrador, ésta se divide en dos secciones, *Anuncios actuales* y *Anuncios futuros* (ver figura 16).

El sistema revisa la base de datos cada 24 horas para extraer los anuncios que tendrán ocurrencia uno, dos y hasta tres días posteriores a la fecha actual del servidor, esto con el objetivo de extraer los anuncios de la base de datos y difundirlos utilizando un canal RSS.

## **6. Conclusiones y perspectivas del proyecto**

Este reporte de proyecto terminal describe el desarrollo del *sistema gestor de información para la difusión de noticias de la UAM Azcapotzalco utilizando un canal RSS*; este sistema almacena, clasifica, y difunde información que es proporcionada por un administrador. El sistema está dividido en tres módulos; el primer módulo es el encargado de recolectar los anuncios, ya sea mediante un archivo, o mediante un formulario. El segundo módulo, se encarga de clasificar los anuncios proporcionados por el administrador; los anuncios se clasifican en *anuncios actuales (con ocurrencia en la semana en curso)* y *anuncios futuros (con ocurrencia posterior a la semana en curso)*; además, el sistema cataloga los anuncios en cuatro categorías diferentes: artículos, trámites académicos/administrativos, eventos deportivos y eventos culturales. El tercer módulo se encarga de publicar en la página web del administrador los anuncios con su respectiva clasificación; aunado a esto, el sistema difunde los anuncios de los eventos que están próximos a ocurrir, utilizando un canal RSS. Estos anuncios pueden llegar a un *smarthphone*, una tableta o computadora, siempre y cuando tengan acceso a internet y previamente se hayan inscrito al canal de difusión del sistema.

### 6.1. Logros

Los objetivos particulares establecidos para el proyecto fueron los siguientes:

- Construir una agenda que contenga los trámites académico/administrativos más importantes que pueden realizar los estudiantes de la UAM Azcapotzalco durante el trimestre.
- Diseñar e implementar la base de datos relacional que alimente al sistema gestor de información.
- Diseñar e implementar un módulo que permita cargar a la base de datos la información que será difundida.
- Diseñar e implementar un módulo que realice procesos de indexación y clasificación a la información almacenada en la base de datos.
- Diseñar el formato de archivo basado en XML que recoja la información publicada por el sistema gestor.
- Diseñar e implementar un módulo que determine qué anuncios deben ser eliminados ó mantenidos para que sean difundidos mediante un canal RSS posteriormente.

El sistema construido puede realizar lo siguiente:

- Recupera la información de los anuncios desde archivos con extensión *.txt*.
- Almacena en una base de datos la información de los anuncios proporcionada por el administrador a través de un formulario.

- Clasifica los anuncios por la fecha en que ocurrirán
- Categoriza los anuncios según su tipo, a saber: artículos, trámites académicos/administrativos, eventos deportivos y eventos culturales.
- Publica en una página web los anuncios una vez clasificados y catalogados.
- Difunde los anuncios utilizando un canal RSS en las fechas señaladas en el mismo anuncio.
- Gestiona los anuncios almacenados en la base de datos; es decir, conserva en la base de datos los anuncios que se catalogan como artículos y trámites académicos/administrativos; y elimina de la base de datos los eventos deportivos y eventos culturales.

Para poder difundir la información mediante RSS, el sistema genera automáticamente un archivo XML que contiene los anuncios a difundir. Este archivo es generado y actualizado periódicamente, es decir cada vez que está cerca la ocurrencia de un evento o trámite.

## 6.2.Cuestiones sin resolver

Cabe destacar que en la propuesta de proyecto terminal, se planteó que el sistema aceptaría documentos con extensión *.txt* y *.pdf*. Sin embargo, el sistema solo acepta archivos con extensión *.txt*, porque por un lado no se tiene, por el momento, una forma de acceder directamente a los campos de un archivo pdf; y por otro lado, para quien edite los anuncios tendrá que hacerlo sobre un archivo tipo *.doc*, *.odt*, ó *.txt*; por lo tanto, se decidió que los archivos que contienen los anuncios tengan extensión *.txt*.

## 6.3.Trabajo a futuro

El sistema es escalable, es decir, se le pueden agregar módulos conforme sean necesarios. Algunas mejoras que se pueden realizar sobre el sistema son, por ejemplo, enriquecer la difusión de información por RSS con difusión de mensajes de texto utilizando las aplicaciones de mensajería instantánea gratuitas disponibles en la web.

Mediante esta nueva difusión, los mensajes pueden ser más personalizados, es decir a los estudiantes se les puede hacer llegar información relacionada solo con ellos, como la fecha de sus exámenes globales, fechas de re-inscripción, estatus de sus trámites, etc.

Algunos módulos que se pueden añadir al sistema son:

- Módulo para agregar nuevas categorías a la base de datos.
- Módulo para agregar nuevos administradores a la base de datos (por el momento solo pueden agregarse directamente a la base de datos).

## Anexos

### Anexo A: Método clasificaAnuncio de la clase Clasificacion

```
/**
 * Método que recibe una fecha de tipo String, regresa 1 si la fecha
 * pertenece a un evento actual, o regresa 2 si la fecha pertenece a un
 * evento futuro.
 *
 * @param fecha Fecha en la que será difundido un anuncio.
 * @return
 */
public int clasificaAnuncio(String fecha) {
    StringTokenizer token = new StringTokenizer(fecha, "-");
    int[] dateParts = new int[token.countTokens()];
    for (int i = 0; i < dateParts.length; i++) {
        dateParts[i] = Integer.parseInt(token.nextToken());
    }
    Calendar date = Calendar.getInstance();
    int month = date.get(Calendar.MONTH) + 1;
    int day = date.get(Calendar.DATE);
    int dayOfWeek = date.get(Calendar.DAY_OF_WEEK);
    if (month == 12 && dateParts[1] == 1) {
        if (dateParts[2] - day > 7) {
            return 2;
        } else {
            return 1;
        }
    }
    if ((dateParts[1] - month) < 2) {
        if (((dateParts[1] - month) == 1) && dateParts[2] > 6
            || ((dateParts[2] - day) >= 7)) {
            return 2;
        } else {
            switch (dayOfWeek) {
                case 1:
                    if (Math.abs(dateParts[2] - day) > 0) {
                        return 2;
                    } else {
                        return 1;
                    }
                case 2:
                    if (Math.abs(dateParts[2] - day) > 6) {
                        return 2;
                    } else {
                        return 1;
                    }
                } else {
                    return 1;
                }
            }
        }
    }
}
```

```
case 3:
    if (Math.abs(dateParts[2] - day) > 5) {
        return 2;
    } else {
        return 1;
    }
case 4:
    if (Math.abs(dateParts[2] - day) > 4) {
        return 2;
    } else {
        return 1;
    }
case 5:
    if (Math.abs(dateParts[2] - day) > 3) {
        return 2;
    } else {
        return 1;
    }
case 6:
    if (Math.abs(dateParts[2] - day) > 2) {
        return 2;
    } else {
        return 1;
    }
case 7:
    if (Math.abs(dateParts[2] - day) > 1) {
        return 2;
    } else {
        return 1;
    }
default:
    return 0;
}
}
} else {
    return 2;
}
}
```

## Anexo B: Método catalogaAnuncio de la clase Clasificacion

```
/**
 * Método que recibe un String con el título o descripción del anuncio,
 * y devuelve un entero según la categoría que le asigne.
 * Devuelve 10 si se trata de un Artículo.
 * Devuelve 20 si se trata de un Trámite Académico/Administrativo.
 * Devuelve 30 si se trata de un Evento Cultural.
 * Devuelve 40 si se trata de un Evento Deportivo.
 * Devuelve 50 si se trata de un Evento Académico.
 * @param texto
 * @return
 */
public int catalogaAnuncio(String texto) {
    if (texto.toLowerCase().startsWith("articulo") ||
        texto.toLowerCase().startsWith("artículo")) {
        return 10;
    }
    StringTokenizer token = new StringTokenizer(texto);
    ArrayList<String> vector = new ArrayList();
    int size = token.countTokens();
    for (int i = 0; i < size; i++) {
        vector.add(token.nextToken().replace(',', ' ').trim().toLowerCase());
        vector.set(i, vector.get(i).replace(',', ' ').trim());
    }
    eliminaPalabras(vector);
    vector.trimToSize();
    int palabrasTramite = cuentaPalabras(vector, "20");
    if (palabrasTramite == anuncioCategorizado) {
        return 20;
    }
    int palabrasCultura = cuentaPalabras(vector, "30");
    if (palabrasCultura == anuncioCategorizado) {
        return 30;
    }
    int palabrasDeporte = cuentaPalabras(vector, "40");
    if (palabrasDeporte == anuncioCategorizado) {
        return 40;
    }
    int palabrasAcademico = cuentaPalabras(vector, "50");
    if (palabrasAcademico == anuncioCategorizado) {
        return 50;
    }
}
```

```

int max = Math.max(Math.max(palabrasTramite, palabrasCultura),
    Math.max(palabrasDeporte, palabrasAcademico));
if ((palabrasTramite + palabrasCultura + palabrasDeporte + palabrasAcademico) != 0) {
    if ((palabrasTramite == max) && (palabrasTramite != palabrasCultura)
        && (palabrasTramite != palabrasDeporte)
        && (palabrasTramite != palabrasAcademico)) {
        return 20;
    } else if (palabrasCultura == max && (palabrasCultura != palabrasTramite)
        && (palabrasCultura != palabrasDeporte)
        && (palabrasCultura != palabrasAcademico)) {
        return 30;
    } else if (palabrasDeporte == max && (palabrasDeporte != palabrasTramite)
        && (palabrasDeporte != palabrasCultura)
        && (palabrasDeporte != palabrasAcademico)) {
        return 40;
    } else {
        return 50;
    }
} else {
    return anuncioSinCategoria;
}
}

```

## Anexo C: Estructura básica de un archivo XML para difusión RSS

Declaración del tipo de documento [7]

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Especificación RSS

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<rss version="2.0">  
  
</rss>
```

Especificación del canal

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<rss version="2.0">  
  
<channel>  
    <title></title>  
    <link></link>  
    <description></description>  
  
</channel>  
  
</rss>
```

Elementos no variables

¿Qué se va a difundir?

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<rss version="2.0">  
  
<channel>  
    <title>El nombre del canal RSS</title>  
    <link></link>  
    <description>Descripción de lo que se va a difundir</description>  
  
<item>  
    <title>Título del anuncio</title>  
    <link></link>  
    <description>Contenido de esta información</description>  
  
</item>  
  
</channel>  
  
</rss>
```

Elementos variables

## *Método generaXML de la clase Publicacion*

```
/**
 * Método que recibe como parámetro un vector de anuncios a difundir,
 * posteriormente genera un archivo con extensión .xml, al cual se suscriben
 * los usuarios finales, para recibir en sus lectores de RSS los anuncios
 * actuales.
 * @param anuncios
 */
public void generaXML(ArrayList<Anuncio> anuncios) {

String start = "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>"
+ "\n <rss version=\"2.0\"> \n <channel> "
+ "\n <title>Difusión de anuncios</title> "
+ "\n <link>http://localhost:8080/ProyectoTerminal</link>"
    + "\n <description>El presente canal es utilizado para hacer "
    + "llegar a la comunidad universitaria,"
    + "información de interés.</description> \n <language>es</language>";
String end = "</channel> \n </rss>";
    FileWriter fichero = null;
    PrintWriter pw;
    try {
        fichero = new FileWriter("/home/user/NetBeansProjects/ProyectoTerminal/web/rss.xml");
        pw = new PrintWriter(fichero);
        pw.println(start);
        for (int i = 0; i < anuncios.size(); i++) {
            pw.println("<item>");
            pw.println("<title>" + anuncios.get(i).getTitulo() + "</title>");
            pw.println("<link>http://localhost:8080/ProyectoTerminal</link>");
            pw.println("<description>" + anuncios.get(i).getFecha() + ": " +
                anuncios.get(i).getDescripcion() + "</description>");
            pw.println("</item>");
        }
        pw.println(end);
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if (fichero != null) {
                fichero.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

## Anexo D: Método nuevoAnuncio de la clase CatalogoAnuncio

```
/**
 * Retorna 1 si se insertaron los datos correctamente, de lo contrario
 * devuelve 0.
 * @param anuncio Contiene los datos proporcionados por el usuario para un
 * nuevo anuncio.
 * @param sesion Contiene los datos del usuario almacenados en la sesión
 * actual.
 * @return
 */

Public int nuevoAnuncio(Anuncio anuncio, HttpSession sesion) {
int flag = 0;
try {
Connection cn = conect.getConnection();
String query = "Insertinto anuncio (titulo,fecha,descripcion,id_administrador,"
        + "id_clasificacion,id_categoria) values('" + anuncio.getTitulo() +
        "', '" + anuncio.getFecha() + "', '"
        + anuncio.getDescripcion() + "', '" + sesion.getAttribute("usuario") +
        "', " + Integer.parseInt(anuncio.getClasificacion()) + ", " +
Integer.parseInt(anuncio.getCategoria()) + "')";
Statement st = cn.createStatement();
flag = st.executeUpdate(query);
conect.cierraConexion(cn);
    } catch (SQLException e) {
e.printStackTrace();
    } finally {
Return flag;
    }
}
```

## Glosario

RSS	Really Simple Syndication, un formato XML para indicar o compartir contenido en la web.
Feed	Fuente web o canal web, es un medio de redifusión de contenido web. Se utiliza para suministrar información actualizada frecuentemente a sus suscriptores.
Elementos no variables	Parte estática del documento XML, donde se describe nombre del canal, liga del sistema y descripción del sistema.
Elementos variables	Parte del documento XML que cambian dependiendo del anuncio que se quiera difundir. Es decir puede ser un elemento (ítem) o más.
Sistema de información	Conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo.
Clasificación de anuncios	Cuando hablamos de clasificación, nos referimos al tempo en el que ocurren los eventos que se anuncian, es decir: Anuncios actuales (con ocurrencia en la semana en curso) y Eventos futuros (con ocurrencia posterior a la semana en curso)
Categorización de anuncios.	A cada anuncio se le asigna una categoría, dependiendo de las palabras que estos contengan. Categorías como: artículos, trámites académicos/administrativos, eventos culturales y eventos deportivos.
Framework	Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.
Struts	Herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition).
NetBeans IDE	Entorno de desarrollo integrado de código abierto.
MySQL	Manejador de bases de datos de código abierto.
GlassFish	Servidor de aplicaciones de código abierto.
Java	Es un lenguaje de programación de alto nivel orientado a objetos.
JSP	Java Server Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
XML	Extensible Markup Language permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

## Bibliografía

- [1] Craig Larman, *UML y Patrones*, Pearson 2003.
- [2] Fuente de la imagen (2012, Noviembre 15). *El Proceso Unificado de Desarrollo de Software* [En línea]. Disponible en: <http://yaqui.mx1.uabc.mx/~molguin/as/RUP.htm>
- [3] Lector RSS (2012, Septiembre 10). RSS Explicado [En línea]. Disponible en: <http://www.rss.nom.es/lector-rss/>
- [4] MySQL (2012, Noviembre 18). MySQL 5.0 Reference Manual [En línea]. Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/>
- [5] GlassFish (2012, Septiembre 10). GlassFish Community [En línea]. Disponible en: <http://glassfish.java.net/es/>
- [6] AES (2012, Noviembre 15). *Advanced Encryption Standard* [En línea]. Disponible en: [http://computacion.cs.cinvestav.mx/~jjangel/aes/AES\\_v2005\\_jjaa.pdf](http://computacion.cs.cinvestav.mx/~jjangel/aes/AES_v2005_jjaa.pdf)
- [7] RSS Tutorial (2012, Noviembre 26). RSS Document Example [En línea]. Disponible en: <http://www.w3schools.com/rss/default.asp>