

Universidad Autónoma Metropolitana unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Ingeniería en Computación

Reporte final

Linkuam, “Sistema de apoyo para la difusión de ideas innovadoras y emprendedoras”.

Jorge Alberto Bautista Hernández
201301215
Trimestre 11-P, 31 de julio de 2011

Asesora de proyecto terminal
Dra. Ma. Lizbeth Gallardo López
Profesor-investigador Asociado *D* del Departamento de Sistemas

Índice

1. Objetivo general	1
2. Objetivos particulares	1
2.1. Registrar las ideas creativas de los miembros de la UAM	1
2.2. Difundir ideas entre la comunidad universitaria	1
2.3. Difundir proyectos en proceso de desarrollo	1
2.4. Facilitar la comunicación con diferentes comunidades	1
3. Antecedentes	1
3.1. Referencias internas	1
3.2. Referencias externas	2
4. Justificación	3
4.1. Innovar y emprender	3
4.2. Innovar y emprender en la UAM	3
4.3. Servicio social	4
4.4. Proyecto terminal	5
4.5. Aprovechamiento de conocimiento generado	6
4.6. Compromiso con la sociedad	6
5. Desarrollo del proyecto	7
5.1. Diseño	7
5.1.1. Metodología de base empleada en el proyecto	7
5.1.2. Modelado del sistema	8
5.1.3. Módulos del sistema	32
5.1.4. Estructura de la base de datos	33
5.2. Implementación	37
5.2.1. Tecnología empleada	37
5.3. Conclusiones y perspectivas del proyecto	38
5.4. Anexos	38
Referencias	39

Índice de figuras

1. Diagrama de casos de uso.	8
2. Diagrama de dominio de la aplicación.	23
3. Diagrama de clases de la aplicación.	28
4. Diagrama de estados de la aplicación.	31
5. Módulos principales de la aplicación.	32
6. Estructura de base de datos de la aplicación.	36

Resumen

*Innovar*¹[1] y *emprender*²[1] son verbos que en la actualidad marcan la línea divisoria entre un seguidor y un líder. Para que México pueda progresar y mantener ese progreso durante un largo tiempo, debe convertirse en un líder, innovando y emprendiendo. Partiendo sobre todo de las que deberán de ser sus principales generadoras de talento, conocimiento y soluciones a problemas en nuestra sociedad: sus universidades.

Las universidades forman parte de las instituciones sociales, que juegan un papel fundamental en la transformación de las sociedades[2]. Cambiemos la forma en que las vemos, pensemos en ellas como centros para la convivencia y preparación de personas que pueden cambiar el rumbo de nuestro país y del mundo, y no solo como un lugar donde vienen los jóvenes a tomar una educación superior. Preparemos a los jóvenes para que sean capaces de tomar liderazgos en un ambiente como en el que México se encuentra actualmente inmerso.

La actividad de promover una cultura emprendedora e innovadora que ayude a este propósito, bien puede apoyarse en un sistema informático.

Este documento contiene un reporte sobre el desarrollo de LinkUAM.

¹Innovar significa “Mudar o alterar algo, introduciendo novedades”.

²Emprender significa “Acometer y comenzar una obra, un negocio, un empeño, especialmente si encierran dificultad o peligro”.

1. Objetivo general

El principal objetivo de este proyecto terminal es promover una cultura innovadora y emprendedora dentro de nuestra Universidad, a través del desarrollo de una aplicación web que facilite la difusión y ayude en la gestión de ideas de proyectos generadas por miembros de la UAM¹.

2. Objetivos particulares

2.1. Registrar las ideas creativas de los miembros de la UAM

Se desarrollará un módulo que permita a los miembros de nuestra Universidad registrar aquellas ideas que podrían ayudar a la UAM o a nuestra sociedad.

2.2. Difundir ideas entre la comunidad universitaria

Se desarrollará un módulo que permita iniciar, publicar, criticar y retroalimentar ideas (por ejemplo para proyectos de vinculación, para iniciar una empresa, proyectos de investigación, etc.) propuestas por miembros de la UAM e incluso por egresados de la misma. Esta información será difundida a los miembros de la comunidad universitaria, a través de una página Web.

2.3. Difundir proyectos en proceso de desarrollo

Un módulo que permita la publicación y difusión de propuestas de proyectos tanto de servicio social como de proyecto terminal; proporcionando así un repositorio de opciones, para los alumnos interesados en cubrir estos requerimientos.

2.4. Facilitar la comunicación con diferentes comunidades

Un módulo que facilite a miembros y no miembros de la Universidad, la publicación de problemas que enfrenta su comunidad o institución; y que por supuesto buscan apoyo dentro de nuestra comunidad estudiantil y académica.

3. Antecedentes

3.1. Referencias internas

- *Sistema basado en web para la gestión de capital intelectual referente a la organización de las metodologías de las TIC.*

El “Sistema basado en web para la gestión de capital intelectual referente a la organización de las metodologías de las TIC”², es una aplicación que permite la difusión y publicación de información relacionada con tecnologías informáticas, así como la posibilidad de publicar, descargar y vender cursos a través de él.

¹Entiendase como “miembro de la UAM” o “comunidad universitaria” a todo aquel que estudie y trabaje (o lo haya hecho) en cualquiera de los planteles de la UAM.

²Al momento de escribir este documento, la única alternativa para conseguir información sobre este sistema es con el coordinador de la carrera de Ingeniería en Computación, impartida en la UAM Azcapotzalco.

Nuestro proyecto terminal también tiene como objetivo difundir información, sin embargo, no será exclusiva de tecnologías TI. Tampoco planteamos ningún objetivo donde se requieran funcionalidades para poder comerciar información.

- ***Sistema de administración de proyectos terminales.***

El “Sistema de administración de proyectos terminales” [] permite gestionar la información de los proyectos terminales que se han desarrollado hasta el momento en la UAM-A. También permite llevar un control de los asesores y estudiantes relacionados con determinado proyecto.

Nuestro proyecto terminal propone difundir proyectos terminales a realizarse; no así un histórico sobre los ya realizados.

- ***Sistema de gestión de información de programas de responsabilidad social.***

El “Sistema de gestión de información de programas de responsabilidad social” [] tiene como objetivo gestionar las actividades de responsabilidad social, para poder medir sus alcances y resultados en el futuro.

Nuestro proyecto terminal también hace énfasis en el registro de actividades; sin embargo, no tiene como objetivo medir los alcances de las actividades que se registrarán y difundirán en él.

3.2. Referencias externas

- ***Red mexicana de emprendedores & Ángel Ventures México***

El objetivo principal de Red mexicana de emprendedores[3] y Ángel Ventures México[4] es servir de enlace entre proyectos innovadores y personas que pueden invertir monetariamente en éstos. Además clasifican las diferentes propuestas e inversiones haciendo aún más sencilla su vinculación.

Red mexicana de emprendedores plantea una idea: “Existen proyectos que necesitan capital y existen capitales que pueden ser invertidos; nosotros los vincularemos.”.

Al igual que “Red mexicana de emprendedores”, uno de los fines que tiene este proyecto terminal, es comunicar a miembros de la UAM que generen ideas con otros miembros que puedan ejecutarlas, aunque estas ideas no sean necesariamente empresas.¹

- ***Emprendimiento, Tecnológico de Monterrey***

Con el objetivo de continuar promoviendo y desarrollando el espíritu emprendedor en sus alumnos, el Tecnológico de Monterrey ha creado el Instituto para el Desarrollo de Emprendedores[5], a través del cual se les brindan las herramientas necesarias para crear, iniciar, consolidar y hacer crecer sus propias empresas.

El Tecnológico de Monterrey cuenta con muchas herramientas disponibles para alumnos emprendedores, desde asesoría en el inicio de sus proyectos, pasando por búsqueda de capital e incluso apoyo en cuestiones de patentes.

Nuestro proyecto terminal podría, en un futuro, servir de catapulta en cada una de las fases de los proyectos que alumnos emprendedores lleven a cabo, tal como lo hace el Tecnológico de Monterrey.

¹ “La ignorancia acerca del tema, ha llevado a convertir el emprendimiento en sinónimo de creación de empresas.” [6]

4. Justificación

4.1. Innovar y emprender

Innovar y emprender han llevado a muchos países a convertirse en los líderes del mundo en la actualidad y están llevando a otros a convertirse en los líderes del futuro. Sin enfocarnos solo a la cuestión del progreso monetario, los países más innovadores y emprendedores están generando soluciones para muchos de los problemas que afectan al mundo.

Lamentablemente, México no cuenta con una cultura innovadora ni emprendedora de calidad en sus universidades (públicas). Como resultado, la mayoría de nuestros alumnos no innovan o emprenden a lo largo de su estancia en la Universidad, se limitan a hacer lo que sus tareas les exigen. Esto no quiere decir que nuestros alumnos no sean capaces de innovar o emprender (¡todo lo contrario!), es simplemente una falta de atención por parte de académicos y autoridades de la Universidad hacia cuatro cosas: espacios donde la comunidad universitaria pueda publicar, criticar, iniciar y desarrollar ideas; espacios para promover una amplia comunicación entre nuestra comunidad universitaria (alumnos, académicos y autoridades universitarias de diferentes carreras y divisiones, incluso planteles); difusión de planes de apoyo que la UAM ofrece para el inicio y desarrollo de ideas emprendedoras; y tal vez la más importante de todas: hacer conscientes a todos los miembros de la UAM de que realmente pueden orientar sus esfuerzos a innovar, a emprender, y no necesariamente a seguir esquemas ya impuestos por la industria actual.

4.2. Innovar y emprender en la UAM

Durante el periodo de tiempo que he estado estudiando en la UAM Azcapotzalco, he tenido muy pocos profesores que aborden, en su clase, sobre las actividades que podemos (¿tal vez “debemos”?), desde el punto de vista de apoyar a nuestra sociedad) realizar cuando terminemos nuestros estudios, o incluso durante el transcurso de estos. Y de estos pocos académicos que lo hacen, la gran mayoría de ellos han impuesto a su clase una manera de pensar muy particular. Entre las ideas que imponen esa manera de pensar se citan las siguientes:

- «Preparense para trabajar en empresas que existen “allá afuera”.»
- «La competencia “allá afuera” está muy fuerte. »
- «Los planes de estudio de su carrera no sirven “allá afuera”.»
- «Esto es solo una práctica, realmente aprenderán “allá afuera”.»

Pasando por alto las “muchas quejas” y “pocas soluciones”, jamás nos mencionan que hay otra opción diferente al “allá afuera”, jamás nos mencionan que también existe la opción de comenzar una empresa y/o actividad remunerada propia, que no es forzoso que los estudiantes universitarios trabajemos para una empresa ya existente (teniendo que adaptarnos a su sistema), sino que ¡podemos ser los creadores de nuestra propia empresa! y que de hecho, nuestro país necesita más que nunca de estudiantes creadores de empresas con ideas innovadoras que impulsen nuestra economía y propongan soluciones a muchos problemas en los que México está inmerso. No quiero decir que trabajar en una empresa ya establecida sea malo, lo que intento transmitir es que muchas veces se limita nuestra visión del mundo laboral, omitiendo comentarios que nos hacen saber que también podemos innovar y emprender. No solamente académicos y autoridades de nuestra Universidad promueven y terminan fijando esta manera de pensar en nuestros estudiantes, hay otros factores por ejemplo: la pobre comunicación existente entre los alumnos de las diferentes divisiones de la UAM Azcapotzalco, e incluso de diferentes carreras:

Supongamos que uno de nuestros diseñadores industriales quisiera deleitar con sus creaciones al mundo¹, y emprender un negocio comenzando con una galería en algún lugar de Polanco, probablemente necesitaría de un par de diseñadores gráficos para crear una página web atractiva y la imagen de su local; un ingeniero en computación que desarrolle su sistema de pagos en línea; un contador para no tener problemas con la SHCP; un abogado para cuestiones legales e ingenieros en computación, mecánica y electrónica que le ayuden a impactar al público creando estantes móviles robóticos donde nuestro diseñador pueda mostrar sus obras. Los alumnos de universidades públicas no cuentan con recursos monetarios suficientes para poder producir estas ideas, nuestro diseñador tendría que preocuparse por algo más que la belleza y originalidad de sus obras. La inversión para la realización de esta idea, no es despreciable, considerando que debe pagar honorarios. Si la inversión pudiera dedicarse exclusivamente a la apertura del negocio, ahorrando en recursos humanos, esto facilitaría la ejecución de su idea. Consideramos que nuestro diseñador puede recurrir a su Universidad, en ella puede encontrar a otros estudiantes emprendedores para poder comenzar su negocio: diseñadores, ingenieros, abogados, etc. Incluso podría encontrar asesoría para poder atraer inversiones y así pagar los sueldos de profesionistas. ¡Es cuestión de comunicación!...He ahí el problema, en la UAM Azcapotzalco no existe la suficiente comunicación entre los alumnos incluso ni entre académicos, ni entre las autoridades de las diferentes Divisiones y/o carreras, como para que proliferen este tipo de actividades; actividades que beneficiarían tanto a los participantes en el negocio, como al prestigio de nuestra Universidad.

Si nuestro alumno quisiera buscar dentro de la UAM estos recursos, ¿cómo los encontraría?, probablemente hay personas emprendedoras que quisieran participar en proyectos de este tipo, pero ¿cómo se enterarían de la existencia de este proyecto? Claro que se puede recurrir a pegar folletos en todas las paredes de la UAM, pero la contaminaríamos visualmente como lo hacemos (de forma lamentable) en tiempos de votaciones o de huelgas, sobretodo si cada alumno con una idea se decidiera a pegar sus propios papelititos.

En este proyecto planteamos una forma de iniciar y dar difusión a actividades emprendedoras e innovadoras dejando limpia nuestra UAM y manteniendo informados sobre ellas tanto a alumnos como a académicos y autoridades.

4.3. Servicio social

Para poder egresar de la UAM Azcapotzalco necesitamos cumplir con 480 horas de servicio social. Estas horas pueden realizarse tanto en instituciones de gobierno como en PyMES²[1].

Se supone que esta actividad “social” debe de beneficiar a la sociedad, sin embargo, muchos de los proyectos en los que nuestros estudiantes deciden realizar su servicio, se alejan mucho de este cometido social. Esto deriva en que nuestros estudiantes son explotados por las instituciones donde realizan su servicio, con pretextos como “experiencia” o “contrataciones futuras”. Este tipo de consecuencias puede deberse a dos razones: la apatía del estudiante por escoger un servicio social adecuado o la falta de variedad en actividades aceptadas que pueden realizarse como servicio social.

Actualmente, para elegir que actividad realizarán como servicio, nuestros estudiantes pueden elegir de una carpeta de proyectos o pueden ir a un institución y “escoger” entre un conjunto de actividades ya determinadas (si tienen suerte). ¿Qué pasaría si nuestros estudiantes pudieran proponer la actividad que quieren realizar como servicio social? Teóricamente deberían de proponer actividades mucho más útiles a la sociedad, en las que se sentirían muy cómodos trabajando. Suponiendo que esto último fuera verdad, no sería sencillo para un alumno encontrar a quien pudiera hacer válidas estas horas de servicio.

¹En este caso nos referimos a un emprendedor “entrepreneur”, ya que está creando su propia empresa o desarrollando su propio emprendimiento.[7]

²PyME significa “Empresa mercantil, industrial, etc., compuesta por un número reducido de trabajadores, y con un moderado volumen de facturación”.

Imaginemos que a un emprendedor estudiante de sociología se le ocurre la idea de realizar, junto con otros compañeros de diseño gráfico e industrial, un libro con las fotografías más representativas de la UAM¹, recopilando imágenes de sus lugares y momentos más bellos a través del tiempo, recorriendo sus diferentes planteles. Este libro se le regalaría a cada alumno de la Universidad, teniendo como objetivo aumentar el sentido de identidad en nuestra comunidad estudiantil (que buena falta nos hace). Esta idea interdisciplinaria es buena (a mi parecer), ¿pero a quién le es útil? ¿quién querría (y podría hacer válidas) 480 horas dedicadas a esa actividad interdisciplinaria?...nos encontramos con el mismo problema: mala (incluso inexistente) comunicación entre alumnos, académicos y autoridades de la Universidad.

En este proyecto planteamos una forma de proponer actividades para servicio social por parte de la comunidad universitaria, ampliando el abanico de proyectos aceptados, promoviendo así que las actividades realizadas realmente ayuden a la sociedad. Estas nuevas actividades propuestas, serán difundidas a estudiantes y académicos, logrando así que alguien interesado en ellas, pueda hacerlas válidas.

4.4. Proyecto terminal

Otro requisito para poder egresar de la carrera de Ingeniería en Computación, es la realización de un proyecto terminal, para lo cual requerimos al menos de un asesor que nos guíe a lo largo del desarrollo de este proyecto.

Lo que nosotros los alumnos hacemos normalmente, es buscar a nuestro asesor, vamos de cubículo en cubículo preguntando por la lista de proyectos terminales de cada profesor y buscamos uno que se nos resulte interesante. En el mejor de los casos, también proponemos un nuevo proyecto terminal al profesor, que con algo de suerte nos aceptará. Aquí suceden dos cosas. Primero, si el profesor nos ofrece su lista de proyectos y nosotros tomamos uno, estamos dejando de proponer nuestras ideas para la realización de nuestro proyecto. ¿Por qué no emprender ideas innovadoras con nuestro proyecto terminal? Segundo, si nosotros proponemos un nuevo proyecto terminal a un profesor y este nos lo acepta, ¿estamos seguros de que ese asesor es el adecuado para guiarnos?, tal vez exista un profesor en la UAM de otra carrera o división, con más experiencia en los temas relacionados con el proyecto, que permita maximizar los frutos y la utilización de éste.

Habiendo mencionado dos principales formas en que nuestros alumnos definen sus proyectos terminales y a sus asesores, nos parece pertinente proponer al menos 2 opciones más.

Como tercer alternativa a las formas de elección de proyecto terminal, los profesores podrían elegir a sus asesorados, es decir, los alumnos podrían proponer sus proyectos terminales (tomándose la libertad de emprender e innovar) y los profesores podrían proponerse como asesores de los proyectos terminales que se lleven mejor con su experiencia y/o temas de investigación. Esto, podría no solo maximizar los frutos de un proyecto terminal, haciendolo útil en uno o varios Departamentos de la UAM, en otras universidades o a la sociedad en general. Incluso podría aumentar el número de artículos de investigación que la UAM publica².

Como cuarta alternativa, los profesores podrían publicar y difundir entre la comunidad estudiantil los proyectos terminales que a ellos les parecen interesantes o que serían de ayuda para sus investigaciones.

En este proyecto planteamos una forma de ayudar a los alumnos a proponer sus propios proyectos terminales y que los profesores más adecuados para guiarlos, puedan proponerse como asesores. De la misma manera, planteamos una manera de que los profesores puedan difundir, entre la comunidad de estudiantes, las ideas que ellos tienen para realizar un proyecto terminal y que los estudiantes puedan escogerlo.

¹En este caso nos referimos a un emprendedor “intrapreneur”, ya que está utilizando su talento dentro de su organización, en este caso la UAM.[7]

²Este último fenómeno es posible ya que un profesor investigador puede encontrar, entre proyectos terminales propuestos por la comunidad universitaria, uno que funcione perfectamente a los fines de sus investigaciones.

4.5. Aprovechamiento de conocimiento generado

Existe un sentido de identidad muy débil en la mayoría de los alumnos de la UAM comparado con el de instituciones como el IPN y la UNAM. Esto se debe principalmente a la forma en que funciona nuestra Universidad: los alumnos pueden “escoger”¹ su horario de clases para poder adaptarlo a sus necesidades laborales o familiares, lo que provoca que en cada trimestre tengan compañeros nuevos, haciendo muy difícil el surgimiento de lazos fraternales fuertes entre estudiantes de nuestra Universidad.

Como un efecto secundario, esa escasez de sentido de identidad y de lazos fraternales, provoca que nuestros egresados no participen en actividades dentro de la Universidad posteriores a su titulación, ocasionando que la Universidad no tenga una retroalimentación de todo ese conocimiento y experiencia proporcionados a sus alumnos.

Si la UAM pudiera promover una constante comunicación con sus egresados una vez que estos ya han terminado sus estudios superiores, ellos podrían apoyar, entre otras cosas, a mejorar la calidad de nuestros proyectos terminales, a fortalecer los conocimientos con los que terminan nuestros nuevos egresados, a proporcionar ideas que pudiesen mejorar el estado de nuestra Universidad, ideas para servicio social, ideas para proyectos terminales, podrían proporcionarnos constantes fuentes de empleo para nuevos egresados, podrían incluso ser asesores de proyectos terminales gracias a la experiencia que han adquirido, entre otras muchas ventajas.

En este proyecto planteamos una forma de mantener una comunicación constante con los egresados de la Universidad Autónoma Metropolitana y sobre todo hacerlos participes en nuevas actividades posteriores a su titulación, aprovechando sus nuevos conocimientos y experiencias.

4.6. Compromiso con la sociedad

Los miembros de la Universidad Autónoma Metropolitana, así como todas las universidades públicas de nuestro país, tenemos un compromiso con nuestra sociedad: proponer soluciones a los problemas en los que México se encuentra inmerso y emprender nuevas ideas que hagan de nuestro país un líder en el mundo, mejorando el nivel de vida en todo nuestro territorio.

Muchos problemas presentes en nuestra sociedad pueden ser y están siendo resueltos por nuestra universidad, pero hay muchos otros problemas de los que ni siquiera estamos enterados. Si la sociedad tuviera una forma de comunicar sus problemas a la UAM, nuestras comunidades académicas y estudiantiles podrían resolverlos. La resolución de estos problemas dejarían beneficios tanto a la sociedad como a la misma UAM:

- Nos dejarían conocimiento y experiencia derivados de la investigación realizada para resolverlos.
- Nos darían presencia en más lugares del país, lo que aumentaría el prestigio de nuestra Universidad a nivel nacional, incluso internacional.
- Muchos de esos problemas podrían ser orientados para realizarse como proyectos terminales o servicio social, lo que le daría mucho más sentido al esfuerzo de nuestros académicos y estudiantes.

Este proyecto plantea una forma para crear un enlace de comunicación entre nuestra sociedad y nuestra Universidad, pudiendo así trabajar en conjunto para resolver problemas que al final, a todos nos afectan.

¹Usamos esta palabra entre comillas porque dada la falta de recursos y la mala planeación de los horarios de clase por parte de las autoridades de la Universidad, es casi imposible que la mayoría de nuestros alumnos queden satisfechos con la elección de sus horarios.

5. Desarrollo del proyecto

5.1. Diseño

5.1.1. Metodología de base empleada en el proyecto

La metodología usada para desarrollar este proyecto fue *Extreme Programming*[8]. Para poder tomar esta decisión, se tomaron en cuenta los siguientes factores:

- **Tiempo de desarrollo**

El sistema necesitaba realizarse en un máximo de un trimestre, ya que así fue estipulado en la propuesta de proyecto terminal¹.

- **Curva de aprendizaje**

La curva de aprendizaje de las tecnologías utilizadas para desarrollar LinkUAM fue larga. Se llevó una tercera parte del tiempo disponible para completar el desarrollo total. Esto se debió a que una de las decisiones tomadas al iniciar el desarrollo del sistema fue utilizar algunas de las tecnologías web con más prestigio en ese momento y muchas de ellas eran totalmente desconocidas para nosotros.

- **Número de desarrolladores**

Este proyecto terminal no se desarrolló en equipo. El desarrollo fue responsabilidad de una sola persona², por lo que, de haber elegido una metodología de desarrollo formal como puede ser *RUP*[10] los roles en el desarrollo del sistema y las fases que hubiesen compuesto a cada iteración, se hubieran multiplicado, haciendo imposible terminar en el tiempo máximo reglamentado para terminar un proyecto terminal³.

- **Arquitectura de las tecnologías utilizadas**

Seam Framework[9] fue la base del desarrollo de LinkUAM. Es una tecnología reconocida por habilitar el desarrollo rápido o ágil de aplicaciones web. Para sistemas que no requieren de una infraestructura muy compleja, como es el caso de LinkUAM, la arquitectura de *Seam Framework* permite dedicar menos tiempo al modelado y diseño de clases y más al desarrollo de la lógica del sistema⁴.

Una vez que fue completada la curva de aprendizaje necesaria para poder comenzar el desarrollo de LinkUAM en forma, la lista de factores anteriores provocaron que la decisión de usar *Extreme Programming* como metodología de desarrollo fuera tomada, ya que nos permitiría aprovechar más el tiempo restante en el desarrollo de la lógica del sistema en lugar de diseñar toda su arquitectura. Así mismo la arquitectura propuesta por *Seam Framework* complementó muy bien la manera en que se decidió desarrollar el sistema. La metodología de desarrollo también nos ayudó a adaptarnos y solucionar las dificultades que el sistema nos iba imponiendo, y no tanto a intentar predecirlas, lo cual hubiese llevado más tiempo.

¹Cabe mencionar que finalmente, debido a una mala planeación de mi parte, el desarrollo del sistema tuvo que extenderse a 2 trimestres.

²Este aspecto es el único que no cumple con la metodología *Extreme Programming*, ya que esta exige que se desarrolle siempre en parejas.

³En el momento de redactar este documento, el tiempo reglamentado es de 2 trimestres.

⁴Este último punto no implica que el sistema no tenga una arquitectura y un diseño formal, simplemente *Seam Framework* propone una arquitectura ideal para desarrollo ágil.

5.1.2. Modelado del sistema

■ Casos de uso

En este reporte solo se mostrará la documentación de los casos de uso más importantes y/o complejos del sistema, los cuales se presentan en la *Figura 1*:

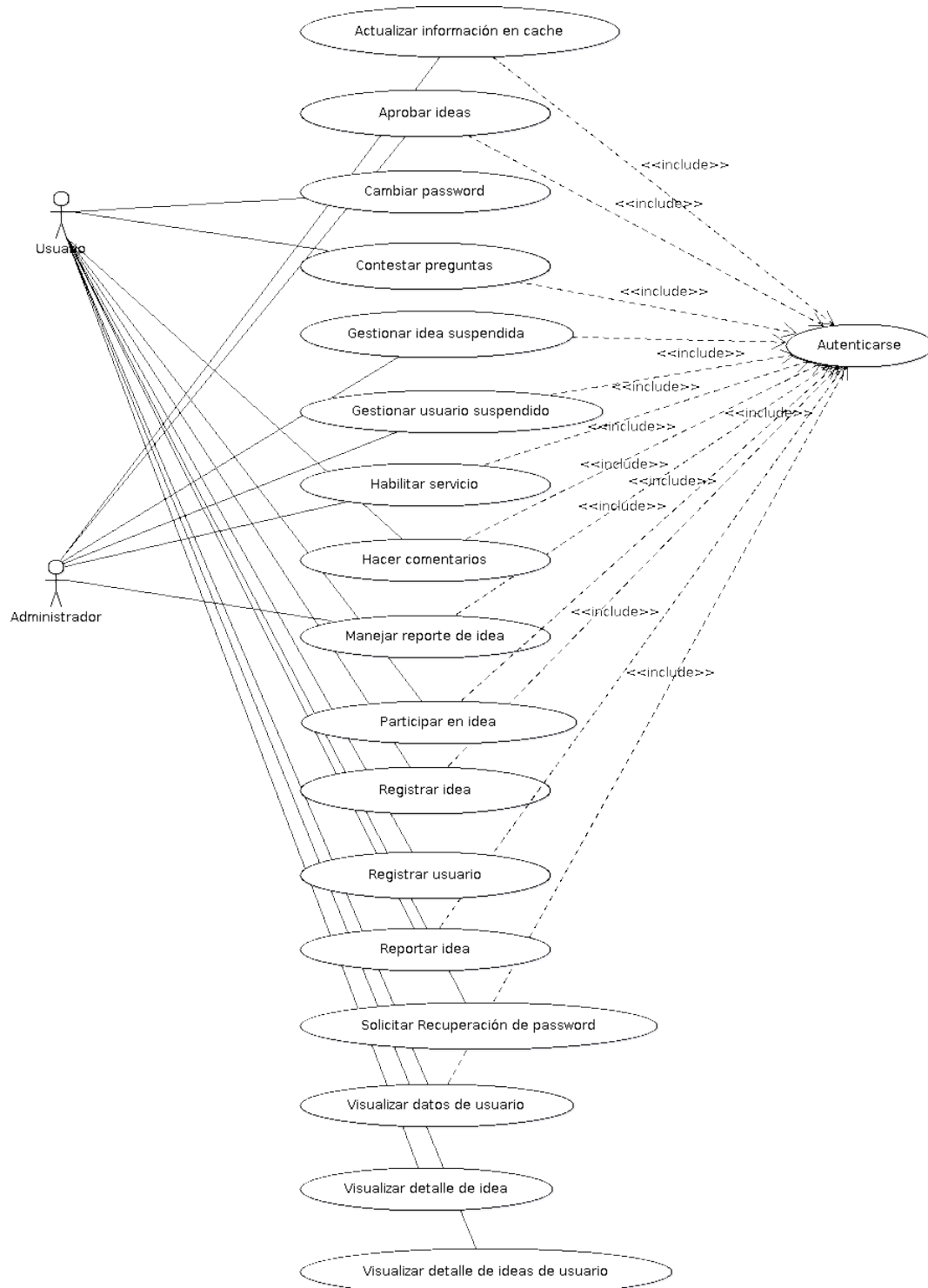


Figura 1: El usuario Administrador puede utilizar todos los casos de uso que contiene este diagrama. Por claridad, en el diagrama se le asocian únicamente los que le son exclusivos.

Caso de uso: Actualizar información en *cache*

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Administrador.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta actualizar el contenido de los catálogos almacenados en la caché del sistema: Frases, instituciones, tipos de idea, vistas.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *ADMINISTRADOR*.
 4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *ADMINISTRADOR*.
 5. El usuario selecciona la pestaña *Recarga catálogos*.
 6. El sistema muestra al usuario una lista de los catálogos que utiliza el sistema.
 7. El usuario presiona el botón *Recargar* perteneciente al catálogo que desea actualizar.
 8. El sistema notifica al usuario de la conclusión de la operación.
 9. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.
- **Postcondición**

La información de uno o más catálogos ha sido actualizada.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
 2. En el paso 8, el usuario decide recargar más catálogos.
 - I. El usuario repite el proceso desde el paso 7 mientras sea necesario.
- **Consideraciones especiales**
 1. El proceso descrito en este caso de uso, es idéntico para recargar los catálogos: *Recargar frases*, *Recargar instituciones*, *Recargar tipos de idea*, *Recargar vistas*.

Caso de uso: Aprobar ideas

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Administrador.
- **Prerequisitos:** Usuario administrador autenticado en el sistema que quiere aprobar ideas.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *ADMINISTRADOR*.

4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *ADMINISTRADOR*.
5. El usuario selecciona la pestaña *Aprobar ideas*.
6. El sistema muestra al usuario una lista de ideas que no han sido aprobadas.
7. El usuario presiona el botón *Aprobar idea* perteneciente a la idea o conjunto de ideas que desee aprobar.
8. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.

- **Postcondición**

Una o más ideas han sido aprobadas por un usuario con rol de administrador y por lo tanto desde ese momento serán visibles para cualquier usuario que ingrese a LinkUAM.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.

- **Consideraciones especiales**

1. El proceso descrito en este caso de uso, es idéntico para aprobar preguntas, respuestas y bitácoras.

Caso de uso: Autenticarse

- **Casos de uso relacionados:** Todos los que requieran autenticación. Aprobar ideas, Contestar preguntas, Habilitar servicio, Hacer comentarios, Recargar catálogos, Registrar idea, Gestionar ideas suspendidas, Gestionar usuarios suspendidos, Visualizar datos de usuario, Visualizar ideas de usuario.

- **Roles:** Cualquiera.

- **Prerequisitos:** Usuario ha ingresado a LinkUAM a través de un navegador web y quiere utilizar funcionalidades que requieren autenticación.

- **Secuencia normal**

1. El usuario hace click en el botón *Iniciar sesión* o intenta utilizar funcionalidades que requieran que se identifique en el sistema.
2. El sistema almacena la vista en la que se encuentra actualmente el usuario para regresar a ella después de terminar el proceso de autenticación.
3. El sistema redirige al usuario a la vista destinada para autenticarse.
4. El sistema muestra un formulario con un campo de texto para el nombre de usuario y otro para la contraseña o password, así como un botón para enviar la información.
5. El usuario introduce usuario y contraseña.
6. El usuario presiona el botón *Iniciar sesión*.
7. El sistema redirige al usuario a la vista en la que se encontraba antes de comenzar el proceso de autenticación.

- **Postcondición**

El usuario a quedado autenticado en el sistema y puede utilizar las funcionalidades que requieran autenticación.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 6, si el nombre de usuario o contraseña no son correctos.
 - I. El sistema avisará al usuario de su error.
 - II. El usuario repite el proceso desde el paso 5 mientras sea necesario.
3. En el paso 4, si el usuario no está registrado en el sistema.
 - I. Caso de uso *Registrar usuario*

Caso de uso: Cambiar password

- **Casos de uso relacionados:** Solicitar recuperación de password.
- **Roles:** Cualquiera.
- **Prerequisites:** Usuario recibió un correo electrónico para continuar con el proceso de cambio de password.
- **Secuencia normal**
 1. El usuario hace click en el hipervínculo que recibió en el correo enviado por el sistema.
 2. El sistema verifica que el hipervínculo sea válido.
 3. El sistema redirige al usuario a la vista de cambio de password y muestra dos campos de texto. Un campo es para introducir el nuevo password y el otro para confirmarlo. También muestra un botón para enviar la información.
 4. El usuario introduce su correo electrónico y presiona el botón *Solicitar renovación de password*.
 5. El sistema verifica que ambas contraseñas introducidas sean iguales.
 6. El sistema redirige al usuario a la página principal, le notifica que la operación fue exitosa y que ya puede autenticarse.
- **Postcondición**
Un usuario ha cambiado su contraseña.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
 2. En el paso 5, las contraseñas no son iguales.
 - I. El sistema avisa al usuario que las contraseñas no son iguales y le pide repetir el proceso.
- **Consideraciones especiales**
 1. Este caso de uso es la continuación del caso: *Solicitar recuperación de password*.

Caso de uso: Contestar preguntas

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Cualquiera.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta contestar las preguntas que hayan hecho acerca de sus ideas.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *Ideas*.
 4. El sistema muestra al usuario todas las pestañas relacionadas con ideas. También mostrará un botón con la leyenda *Contestar preguntas*.
 5. El usuario hace click sobre el botón con la leyenda *Contestar preguntas*.
 6. El sistema muestra al usuario una ventana (ventana A) que contiene una o varias preguntas sin contestar.
 7. El usuario hace click en el botón *Contestar* presente en la *ventana A* correspondiente a la pregunta que desea contestar.
 8. El sistema muestra al usuario una nueva ventana (ventana B) que contiene el formulario para poder contestar la pregunta.
 9. El usuario llena el formulario con la respuesta y presiona el botón *Contestar* en la ventana *B*.
 10. La ventana *B* es cerrada, la lista de preguntas presente en la ventana *A* es actualizada quedando solo las preguntas que no han sido respondidas y un correo electrónico ha sido enviado al usuario que realizó la pregunta, notificándole que ha sido respondida.
 11. El usuario cierra la ventana *A*.
 12. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.
- **Postcondición**

Una o más preguntas han sido contestadas.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige al usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
 2. En el paso 10, el usuario puede decidir seguir contestando preguntas.
 - I. El usuario repite el proceso desde el paso 7 el número de veces que sea necesario.

Caso de uso: Gestionar idea suspendida

- **Casos de uso relacionados:** Autenticarse, Manejar reporte de idea, Reportar idea, Gestionar usuario suspendido.
- **Roles:** Administrador.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta rescatar o desechar definitivamente una idea que fue suspendida al gestionar un reporte.

- **Secuencia normal**

1. El usuario hace click en el botón *Panel de control*.
2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
3. El usuario hace click en la pestaña de *ADMINISTRADOR*.
4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *ADMINISTRADOR*.
5. El usuario selecciona la pestaña *Papelera de reciclaje*.
6. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *Rescatar ideas*.
7. El sistema muestra al usuario una lista de ideas que han sido suspendidas debido a la gestión de un reporte que tuviese una violación grave a las políticas de LinkUAM.
8. El usuario puede aplicar a cada elemento de la lista de ideas, una de 2 acciones disponibles: rescatar idea o desechar idea.
9. El usuario selecciona el botón que corresponda con la acción que desee aplicar sobre la idea suspendida.
10. Si el usuario seleccionó *Rescatar idea*, el estatus de la idea cambiará a *IDEA_NO_APROBADA*, dando la posibilidad de que sea reactivada. Si el usuario seleccionó *Desechar idea*, el estatus de la idea cambiará a *IDEA_DESECHADA*, eliminando cualquier posibilidad de rescatarla.
11. El sistema notifica al usuario que ha terminado la operación y la lista de ideas suspendidas se actualiza, quitando la idea que acaba de gestionarse.
12. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.

- **Postcondición**

Una o más ideas suspendidas fueron gestionadas.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 11, el usuario puede decidir seguir gestionando ideas suspendidas.
 - I. El usuario repite el proceso desde el paso 9 el número de veces que sea necesario.

Caso de uso: Gestionar usuario suspendido

- **Casos de uso relacionados:** Autenticarse, Manejar reporte de idea, Gestionar idea suspendida, Reportar idea
- **Roles:** Administrador.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta rescatar o desechar definitivamente a un usuario que fue suspendido al gestionar un reporte.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.

3. El usuario hace click en la pestaña de *ADMINISTRADOR*.
4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *ADMINISTRADOR*.
5. El usuario selecciona la pestaña *Papelera de reciclaje*.
6. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *Rescatar usuarios*.
7. El sistema muestra al usuario una lista de usuarios que han sido suspendidos debido a la gestión de un reporte que tuviese una violación grave a las políticas de LinkUAM.
8. El usuario puede aplicar a cada elemento de la lista de usuarios, una de 2 acciones disponibles: Rescatar usuario o desechar usuario.
9. El usuario selecciona el botón que corresponda con la acción que desee aplicar sobre el usuario suspendido.
10. Si el usuario seleccionó *Rescatar usuario*, el estatus del usuario cambiará a *USUARIO_ACTIVADO*. Si el usuario seleccionó *Desechar usuario*, el estatus del usuario cambiará a *USUARIO_DESECHADO*, eliminando cualquier posibilidad de rescatarlo.
11. El sistema notifica al usuario que ha terminado la operación y la lista de usuarios suspendidos se actualiza, quitando el usuario que acaba de gestionarse.
12. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.

- **Postcondición**

Una o más usuarios suspendidos fueron gestionados.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 11, el usuario puede decidir seguir gestionando usuarios suspendidos.
 - I. El usuario repite el proceso desde el paso 9 el número de veces que sea necesario.

Caso de uso: Habilitar/Deshabilitar servicio

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Administrador.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta habilitar o deshabilitar un servicio.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *ADMINISTRADOR*.
 4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *ADMINISTRADOR*.
 5. El usuario hace click en la pestaña *Servicios del sistema*.

6. El sistema muestra una lista de los servicios disponibles. Cada servicio está acompañado de un botón para activarlo o desactivarlo.
 7. El usuario hace click en el botón Activar/Desactivar correspondiente al servicio que desea modificar.
 8. El sistema avisa al usuario que la operación fue exitosa.
 9. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.
- **Postcondición**
Se ha cambiado el estado activo/inactivo de uno o más servicios.
 - **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
 2. En el paso 8, el usuario puede decidir seguir habilitando o deshabilitando otros servicios.
 - I. El usuario repite el proceso desde el paso 7 el número de veces que sea necesario.
 - **Consideraciones especiales**
 1. El proceso descrito en este caso de uso, es idéntico para modificar el estado de los servicios: *Servicio de envío de correo, Activador de preguntas, Activador de respuestas, Activador de bitácoras, Activador de ideas, Cargador de tweets, Cargador de fotos, Otras ideas, Visitas, Estadísticas*.

Caso de uso: Hacer comentarios/preguntas

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Cualquiera.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta preguntar o hacer un comentario en una idea. Usuario se encuentra visualizando el detalle de una idea en la vista correspondiente.
- **Secuencia normal**
 1. El usuario (usuario A) hace click en el botón *Comentar/Preguntar en esta idea* presente en la parte de abajo de la vista.
 2. El sistema muestra al usuario un formulario para poder escribir el comentario o pregunta y un botón para poder enviar la información.
 3. El usuario llena el formulario con su comentario o pregunta, y presiona el botón *Enviar pregunta*.
 4. El sistema muestra al usuario un cuadro de diálogo avisándole que la pregunta fue enviada y que se le avisará por correo cuando esta sea respondida.
 5. El usuario cierra el cuadro de diálogo.
 6. El sistema notifica del suceso al usuario autor de la idea (usuario B) en la que se realizó la pregunta, por medio de un correo electrónico.
- **Postcondición**
Una o más preguntas han sido realizadas al autor de la idea (usuario B).
- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 5, el usuario puede decidir seguir realizando preguntas.
 - I. El usuario repite el proceso desde el paso 1 el número de veces que sea necesario.

Caso de uso: Manejar reporte de idea

- **Casos de uso relacionados:** Autenticarse, Reportar idea, Gestionar idea suspendida, Gestionar usuario suspendido.
- **Roles:** Administrador.
- **Prerequisitos:** Un usuario con rol de Administrador necesita gestionar reportes hechos por los usuarios acerca de los contenidos de las ideas.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *ADMINISTRADOR*.
 4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *ADMINISTRADOR*.
 5. El usuario selecciona la pestaña *Reportes de usuario*.
 6. El sistema muestra al usuario una lista de reportes realizados por los usuarios y que no han sido gestionados aún. Se podrá aplicar una de 3 diferentes acciones para gestionar un reporte de abuso: *suspender a usuario* en caso de que los reportes sean muy graves, *suspender la idea* porque efectivamente viola las políticas del sitio, o *desechar reporte* en caso de que este no tenga validez.
 7. El usuario selecciona el botón que corresponda con la acción que el usuario desea aplicar sobre cierto reporte.
 8. Si el usuario selecciona *Suspender usuario*, el usuario será dado de baja del sistema cambiando su estatus a *USUARIO_INACTIVO*. Si selecciona *Suspender idea*, la idea dejará de ser visible para toda la comunidad de LinkUAM cambiando su estatus a *IDEA_INACTIVA*. Por último, si el usuario selecciona *Desechar reporte*, será el reporte el que cambiará de estatus a *REPORTE_LEIDO* y ni el usuario ni la idea sufrirán ninguna consecuencia.
 9. El sistema notifica al usuario que ha terminado la operación y la lista de reportes se actualiza, quitando el reporte que se acaba de gestionar.
 10. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.
- **Postcondición**

Uno o varios reportes han sido gestionados.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.

2. En paso 9, el usuario decide seguir gestionando reportes.
 - I. El usuario repite el proceso a partir del paso 7 mientras sea necesario.

Caso de uso: Participar en idea

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Cualquiera.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta participar en una idea. Usuario se encuentra visualizando el detalle de una idea en la vista correspondiente.
- **Secuencia normal**
 1. El usuario hace click en el hipervínculo *Yo quiero participar* situado junto a la descripción de cada necesidad de la idea.
 2. El sistema notifica al usuario que ya está participando en la idea y que recibirá los datos del autor por correo. Así mismo, el usuario autor de la idea recibirá los datos del usuario que está participando.
 3. El sistema envía un correo al usuario que está participando con los datos del autor de la idea para que puedan contactarse.
 4. El sistema envía un correo al usuario autor de la idea con los datos del usuario que está participando para que puedan contactarse.
 5. El usuario cierra sesión en el sistema haciendo click en el botón *Salir*.
- **Postcondición**

Se ha generado una nueva participación para una idea.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
 2. En el paso 2, el usuario desea participar en más necesidades.
 - I. El usuario repite el proceso desde el paso 1 el número de veces necesarias.

Caso de uso: Registrar idea

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Cualquiera.
- **Prerequisitos:** El usuario esta autenticado en el sistema.
- **Secuencia normal**
 1. El usuario hace click en el hipervínculo *Emprende una idea* o en el botón *Empréndela*.
 2. El sistema redirigirá al usuario a la página de registro de ideas.
 3. El sistema muestra un formulario con los siguientes campos para capturar: título, descripción, necesidades, categorías, cantidad de ayuda, comunidades a las que se les avisará de su creación y una imagen que la represente. También muestra un botón *Publicar esta idea* para enviar la información.
 4. El usuario llena la información y presiona el botón *Publicar esta idea*.
 5. El sistema verifica que al menos el título y la descripción sean válidos y no vayan en blanco.

6. El sistema verifica que si el usuario subió una imagen para representar a su idea, esta no tenga un tamaño mayor al permitido por el sistema.
 7. Si el usuario seleccionó comunidades, el sistema notificará de la creación de la idea a todos los miembros de estas a través de correo electrónico.
 8. El sistema registra la nueva idea y todos los datos capturados.
 9. El sistema redirige al usuario a la página principal, le notifica que la operación fue exitosa y que la idea que acaba de crear se encuentra inactiva hasta que un administrador la apruebe.
- **Postcondición**
 1. Una nueva idea ha sido creada en el sistema.
 2. Si el usuario seleccionó comunidades, se le enviará un correo a cada usuario miembro de estas comunidades.
 - **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
 2. En el paso 5, el título y/o la descripción de la idea están vacías o no son válidos.
 - I. El sistema avisa al usuario que los datos introducidos no son válidos y le pide volver a llenar los campos.
 3. En el paso 6, la imagen tiene un tamaño mayor al permitido por el sistema.
 - I. El sistema avisa al usuario que la imagen es muy grande y muestra un error notificándole.
 4. En el paso 8, si el usuario no seleccionó comunidades.
 - I. Ningún usuario registrado en el sistema será notificado acerca de la creación de esta idea.

Caso de uso: Registrar usuario

- **Casos de uso relacionados:** Ninguno.
- **Roles:** Cualquiera.
- **Prerequisitos:** Un usuario nuevo quiere registrarse en LinkUAM para poder acceder a las funcionalidades que requieren autenticación.
- **Secuencia normal**
 1. El usuario hace click en el hipervínculo *Inscríbete* o el usuario se encuentra en la vista de autenticación al sistema y hace click en *Regístrate*.
 2. El sistema redirige al usuario a la vista de registro de usuarios.
 3. El sistema muestra un formulario con los siguientes campos para capturar: nombre, apellido paterno y materno, institución a la que pertenece, nombre de usuario, password, correo electrónico, imagen que lo representa y temas de interés.
 4. El usuario llena la información y presiona el botón *Completar mi registro*.
 5. El sistema verifica que todos los campos, a excepción de la imagen y los temas de interés sean llenados y tengan información válida.
 6. El sistema verifica que si el usuario subió una imagen, esta no tenga un tamaño mayor al permitido por el sistema.
 7. El sistema registra al nuevo usuario y todos los datos capturados.

8. Si el usuario seleccionó algún tema de interés, el sistema lo hará miembro de la comunidad con el mismo nombre para que pueda recibir notificaciones de ideas creadas y que estén relacionadas con los temas que escogió.
9. El sistema envía un correo al nuevo usuario registrado confirmando su inscripción.
10. El sistema redirige al usuario a la página principal, le notifica que la operación fue exitosa y que ya puede iniciar sesión.

- **Postcondición**

1. Un nuevo usuario ha sido creado en el sistema.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 5, los campos no fueron llenados o la información que contienen no es válida.
 - I. El sistema avisa al usuario que los datos introducidos no son válidos y le pide volver a llenar los campos.
3. En el paso 6, la imagen tiene un tamaño mayor al permitido por el sistema.
 - I. El sistema avisa al usuario que la imagen es muy grande y muestra un error notificándole.
4. En el paso 8, si el usuario no seleccionó comunidades.
 - I. El usuario no será registrado en ninguna comunidad de usuarios y por lo tanto no recibirá notificaciones de creación de ideas.

Caso de uso: Reportar idea

- **Casos de uso relacionados:** Autenticarse, Manejar reporte de idea, Gestionar idea suspendida, Gestionar usuario suspendido.
- **Roles:** Cualquiera.
- **Prerequisitos:** Usuario autenticado en el sistema que intenta reportar el contenido de una idea que no cumple con las políticas de LinkUAM para que sea gestionado por los administradores. Usuario se encuentra visualizando el detalle de una idea en la vista correspondiente.
- **Secuencia normal**
 1. El usuario hace click en el botón *Reportar idea*.
 2. El sistema muestra al usuario un formulario para que escoja la política de LinkUAM que el contenido de una idea no está respetando. El formulario también muestra un campo de texto para describir mejor el reporte y un botón *Enviar reporte* para enviar la información.
 3. El usuario llena el formulario, y presiona el botón *Enviar reporte*.
 4. El sistema muestra al usuario un cuadro de diálogo notificándole que el reporte fue hecho.
 5. El usuario cierra el cuadro de diálogo.
 6. El sistema registra este nuevo reporte, haciendolo gestionable para los usuarios administradores.

- **Postcondición**

Uno o más reportes quedaron registrados en el sistema.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 5, el usuario puede decidir seguir realizando reportes.
 - I. El usuario repite el proceso desde el paso 1 el número de veces que sea necesario.

Caso de uso: Solicitar recuperación de password

- **Casos de uso relacionados:** Cambiar password.

- **Roles:** Cualquiera.

- **Prerequisitos:** Usuario intentando autenticarse en el sistema sin lograrlo. El usuario se encuentra en la vista de autenticación al sistema.

- **Secuencia normal**

1. El usuario hace click en el hipervínculo *Olvidaste tu password*.
2. El sistema redirige al usuario a la vista de recuperación de password y muestra un campo de texto para poder introducir el correo electrónico con el que se registró, así como un botón para enviar la información al sistema con la leyenda *Solicitar renovación de password*.
3. El usuario introduce su correo electrónico y presiona el botón *Solicitar renovación de password*.
4. El sistema verifica que el correo electrónico pertenezca a un usuario en el sistema.
5. El sistema envía un correo electrónico al usuario que necesita renovar su password para poder continuar con el proceso.
6. El sistema redirige al usuario a la página principal y le notifica que la operación fue exitosa.

- **Postcondición**

Un correo electrónico fue enviado al usuario que necesita cambiar su password.

- **Excepciones y flujos alternos**

1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.
2. En el paso 4, el correo electrónico introducido no pertenece a ningún usuario.
 - I. El sistema avisa al usuario de que el correo que introdujo no pertenece a ningún usuario y le pide que repita la operación.

- **Consideraciones especiales**

1. El proceso se continua en el caso de uso *Cambiar password*

Caso de uso: Visualizar datos de usuario

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Cualquiera.
- **Prerequisitos:** Usuario autenticado intentando ver los datos con los que se encuentra registrado en el sistema.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *Datos de usuario*.
 4. El sistema muestra al usuario todos los datos con los que este se registró al sistema.
- **Postcondición**

Datos de usuario autenticado visualizados.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige al usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.

Caso de uso: Visualizar detalle de idea

- **Roles:** Cualquiera.
- **Prerequisitos:** Usuario autenticado intentando ver el detalle de una idea.
- **Secuencia normal**
 1. El usuario hace click en una idea para poder pasar a la página de detalle.
 2. El sistema muestra al usuario el título, descripción, número de seguidores, número de participaciones, calificación de la idea, cantidad que la ayuda necesita, institución a la que pertenece. Además dentro del detalle de la idea se puede acceder a las funcionalidades de *Reportar idea*, *Participar en idea*, *Preguntar o comentar*, *Votar por la idea*, *Seguir idea*.
- **Postcondición**

Idea detallada visualizada.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.

Caso de uso: Visualizar detalle de ideas de usuario

- **Casos de uso relacionados:** Autenticarse.
- **Roles:** Cualquiera.

- **Prerequisitos:** Usuario autenticado en el sistema que intenta visualizar un resumen de las idas que ha creado, de las que actualmente sigue, de las personas que están participando en sus ideas y de las ideas que fueron rechazadas por el administrador.
- **Secuencia normal**
 1. El usuario hace click en el botón *Panel de control*.
 2. El sistema muestra al usuario las pestañas de *ADMINISTRADOR*, *Ideas*, *Datos de usuario*, *Avisos pendientes*.
 3. El usuario hace click en la pestaña de *Ideas*.
 4. El sistema muestra al usuario otro conjunto de pestañas pertenecientes a la sección de *Ideas*.
 5. El usuario puede seleccionar entre las secciones: *Ideas creadas por tí*, *Ideas que sigues actualmente*, *Ideas en las que estás participando* o *Ideas rechazadas* y navegar a través de ellas.
- **Excepciones y flujos alternos**
 1. En cualquier momento el sistema falla.
 - I. El sistema cancela y revierte cualquier operación que no haya sido terminada por completo en la base de datos.
 - II. El sistema redirige a usuario a una página de error genérica informándole lo sucedido y recomendándole regresar a la página inicial.

- Diagrama de dominio

En la *Figura 2* presentamos el diagrama de dominio de la aplicación con el objetivo de comprender mejor como funciona el negocio de esta. Por razones de espacio solo es posible describir una vista previa de éste. El diagrama completo y a gran tamaño se adjunta con este documento.

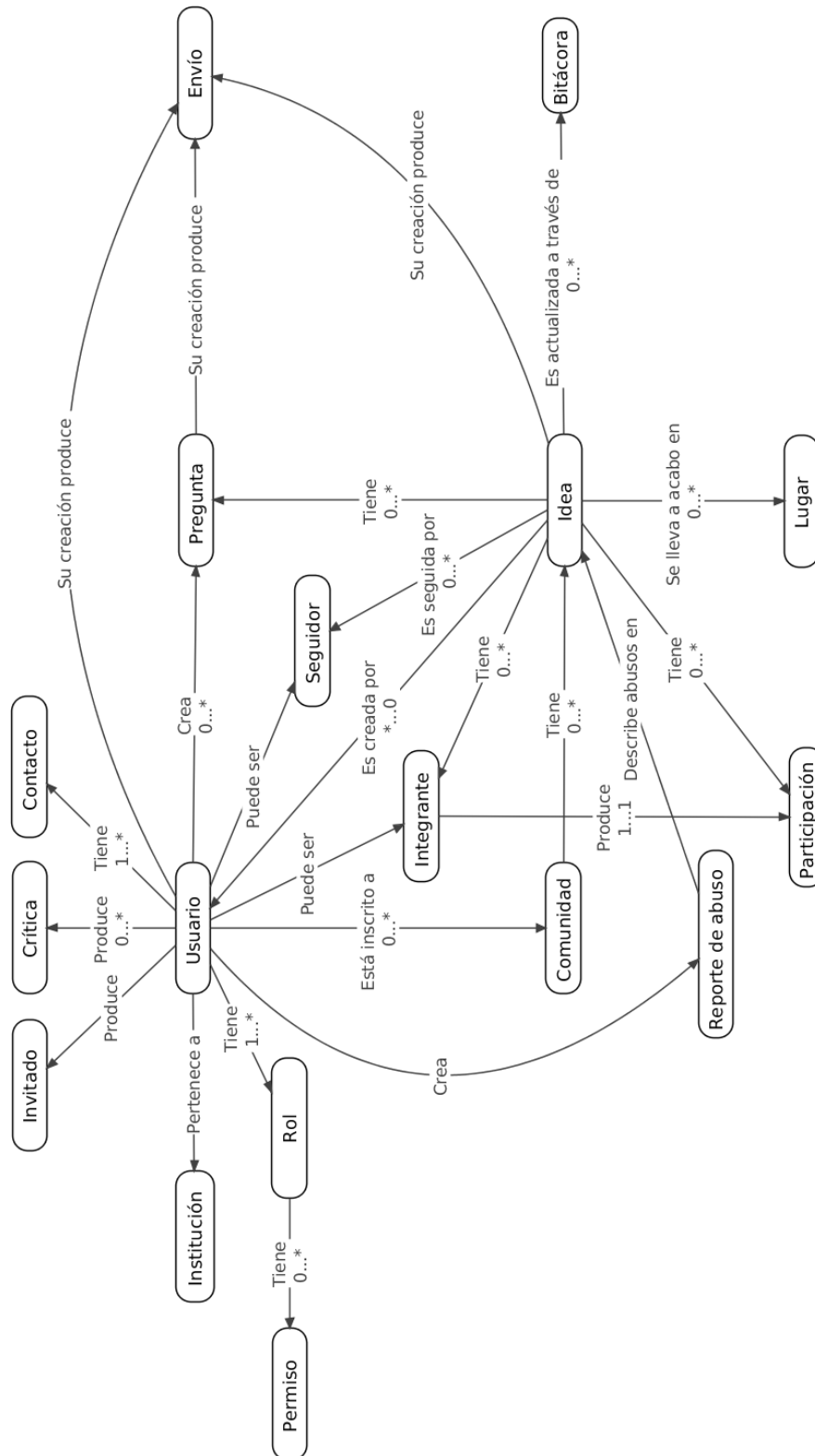


Figura 2: *Diagrama de dominio de la aplicación.*

- **Diagrama de clases** En esta sección serán descritas brevemente las clases más importantes del sistema. En el manual del usuario se describen todas las clases con más detalle.

El siguiente conjunto de clases contienen las operaciones CRUD[11] y todos aquellos métodos relacionados a cada una de las tablas que representan¹.

- | | |
|--|----------------------------------|
| • AccionReporteAbusoHome | • InstitucionHome |
| • AvisoHome | • IntegranteHome |
| • BitacoraHome | • InvitacionHome |
| • CanicaHome | • JustificacionHome |
| • ClasificacionHome | • LugarHome |
| • ComoLograrloHome | • MiembroComunidadHome |
| • ComunidadHome | • NecesidadHome |
| • ConfiguracionUsuarioHome | • ObjetivoEspecificoHome |
| • ContactoUsuarioHome | • ObjetivoGeneralHome |
| • CriticaHome | • PermisoHome |
| • EnvioHome | • PreguntaHome |
| • EstatusAvisoHome | • ReporteAbusoHome |
| • EstatusAyudaMonetariaHome | • RolHome |
| • EstatusAyudaRecursosHumanosHome | • SeguidorHome |
| • EstatusBitacoraHome | • TicketRegistroHome |
| • EstatusCriticaHome | • TipoAvisoHome |
| • EstatusEnvioHome | • TipoAvisoHome |
| • EstatusFraseHome | • TipoBitacoraHome |
| • EstatusIdeaHome | • TipoContactoUsuarioHome |
| • EstatusIntegranteHome | • TipoCriticaHome |
| • EstatusInvitacionHome | • TipoEnvioHomeHome |
| • EstatusMiembroComunidadHome | • TipoIdeaHome |
| • EstatusNecesidadHome | • TipoLugarHome |
| • EstatusPreguntaHome | • TipoReporteAbusoHome |
| • EstatusReporteAbusoHome | • TipoTicketRegistroHome |
| • EstatusSeguidorHome | • TipoVisitaHome |
| • EstatusTicketRegistroHome | • TipoVistaHome |
| • EstatusUsuarioHome | • UsuarioHome |
| • FraseHome | • VisitaHome |
| • IdeaHome | • VistaHome |
| • ImagenHome | |

¹Por ejemplo: la clase *AvisoHome* representa la tabla *aviso*, la clase *CriticaHome* representa la tabla *critica*.

El siguiente conjunto de clases contienen las operaciones que no están relacionadas específicamente a cierta tabla del sistema. Colaboran, entre otras cosas, con la activación de los servicios, manejo de catálogos, autenticación de usuarios, algunas utilerías, valores globales del sistema, valores constantes del sistema, y otros.

- **ActivadorEventos**

Esta clase contiene la lógica necesaria para levantar eventos en el sistema que activan funcionalidades.¹

- **Authenticator**

Esta clase contiene la lógica que implanta la funcionalidad del lado del servidor para que un usuario pueda autenticarse.

- **BuscandoEnAnotacionesException**

Esta clase es una excepción que se levanta en el sistema cuando se buscan datos en las anotaciones de las entidades[12] y algo sale mal.

- **Buscar**

Esta clase contiene la lógica para ejecutar la funcionalidad de buscar ideas del lado del servidor.

- **ControladorServicios**

Esta clase contiene la lógica necesaria para ejecutar la activación o inactivación de los servicios dentro del sistema.

- **EnviadorALogin**

Esta clase auxilia a las funcionalidades que requieren enviar al usuario a la página de login.

- **Estadisticas**

Esta clase funciona como contenedor de las estadísticas del sistema: Número de usuarios, participaciones, preguntas, ideas generadas, usuarios seguidores.

- **FacesMensajero**

Esta clase auxilia a algunas funcionalidades para mandar mensajes al usuario.

- **ImageFormat**

Esta clase contiene las constantes que definen los tipos de datos que se manejan dentro del sistema.

- **ImageManager**

Esta clase contiene la lógica necesaria para ejecutar todas las funcionalidades que se refieren a almacenamiento de imágenes dentro del sistema.

- **ImageSize**

Esta clase contiene las constantes que definen los tamaños de imagen que se manejan dentro del sistema.

- **InicializadorServiciosQuartz**

Esta clase contiene la lógica necesaria para inicializar todos los servicios que requieren estar activos desde que la aplicación se levanta en el servidor.

- **LinkEventosManager**

Esta clase contiene constantes que identifican los eventos manejados dentro del sistema.

- **LinkServicioAprobadorBitacoras**

Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para aprobar bitácoras.

¹Complementando esto, existen clases con métodos que funcionan como escuchadores.

- **LinkServicioAprobadorIdeas**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para aprobar ideas.
- **LinkServicioAprobadorPreguntas**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para aprobar preguntas.
- **LinkServicioAprobadorRespuestas**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para aprobar respuestas.
- **LinkServicioCargadorEstadisticas**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para cargar las estadísticas del sistema.
- **LinkServicioCargadorFotosUsuario**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para cargar las fotos de usuario que se presentan en la sección ¿Quién es LinkUAM?.
- **LinkServicioCargadorOtrasideas**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para cargar las ideas que se presentan del lado derecho en algunas páginas del sistema.
- **LinkServicioCargadorTweets**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para cargar los últimos tweets ¹ <http://www.twitter.com>.
- **LinkServicioMail**
Esta clase funciona como un servicio dentro del sistema y contiene toda la lógica del lado del servidor que se requiere para realizar los envíos de correo generados por ciertos módulos del sistema y que complementan algunas funcionalidades como la recuperación de password.
- **ManagerEstadisticas**
Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo de estadísticas .
- **ManagerFrases**
Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo de frases.
- **ManagerFotosUsuario**
Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo fotos de usuario.
- **ManagerInstituciones**
Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo de instituciones.
- **ManagerOtrasIdeas**
Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo de *Otras ideas*.

¹Un *tweet* es un post en .

- **ManagerTiposIdea**

Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo de tipos de idea presentados en la página principal del sistema.

- **ManagerTweets**

Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga los tweets que se muestran en la página principal del sistema.

- **ManagerVistas**

Esta clase contiene un método observador de eventos generados por la clase *ActivadorEventos*. Al detectar el evento se recarga el catálogo de vistas, que no es otra cosa más que una lista con las vistas que se manejan dentro del sistema.

- **MedidorLongitudColumna**

Esta clase contiene la lógica necesaria para obtener la longitud de una columna de la base de datos. Esto es realizado con objetivos de validación. Utiliza a la clase *MetadatosEnAnotaciones*.

- **MenusDinamicos**

Esta clase contiene la lógica necesaria para hacer dinámicos algunos componentes de la vista.

- **MetadatosEnAnotaciones**

Esta clase contiene la lógica necesaria para realizar la búsqueda en anotaciones.

- **PasswordManager**

Esta clase contiene la lógica necesaria para hashear passwords y comprobar que sean iguales a los almacenados en la base de datos.

- **Path**

Esta clase contiene los paths usados dentro del sistema para almacenar imágenes. También almacena los paths del folder de desarrollo y contenedor de aplicaciones cuando el sistema se encuentra en modo debug.

- **RecuperadorPassword**

Esta clase contiene las operaciones necesarias poder cambiar el password de un usuario.

- **SessionActions**

Esta clase contiene acciones relacionadas con la sesión del usuario.

- **SessionStatus**

Esta clase contiene acciones para obtener información acerca del estado de la sesión de usuario.

- **TipoMiembroAnotable**

Esta clase contiene constantes que auxilian a la clase *MetadatosEnAnotaciones* diciéndole de que tipo de fuente anotable sacará la información: método, atributo o clase.

- **ValoresGlobales**

Esta clase funciona como contenedor de los valores de inicio de la aplicación. Algunos de estos valores pueden ser modificados solo antes de inicial la aplicación, a través de XML y otros cuando la aplicación ya está funcionando.

El diagrama de clases completo se presenta en la *Figura 3*. Por razones de espacio solo es posible describir una vista previa de él. El diagrama completo y a gran tamaño se adjunta con este documento.

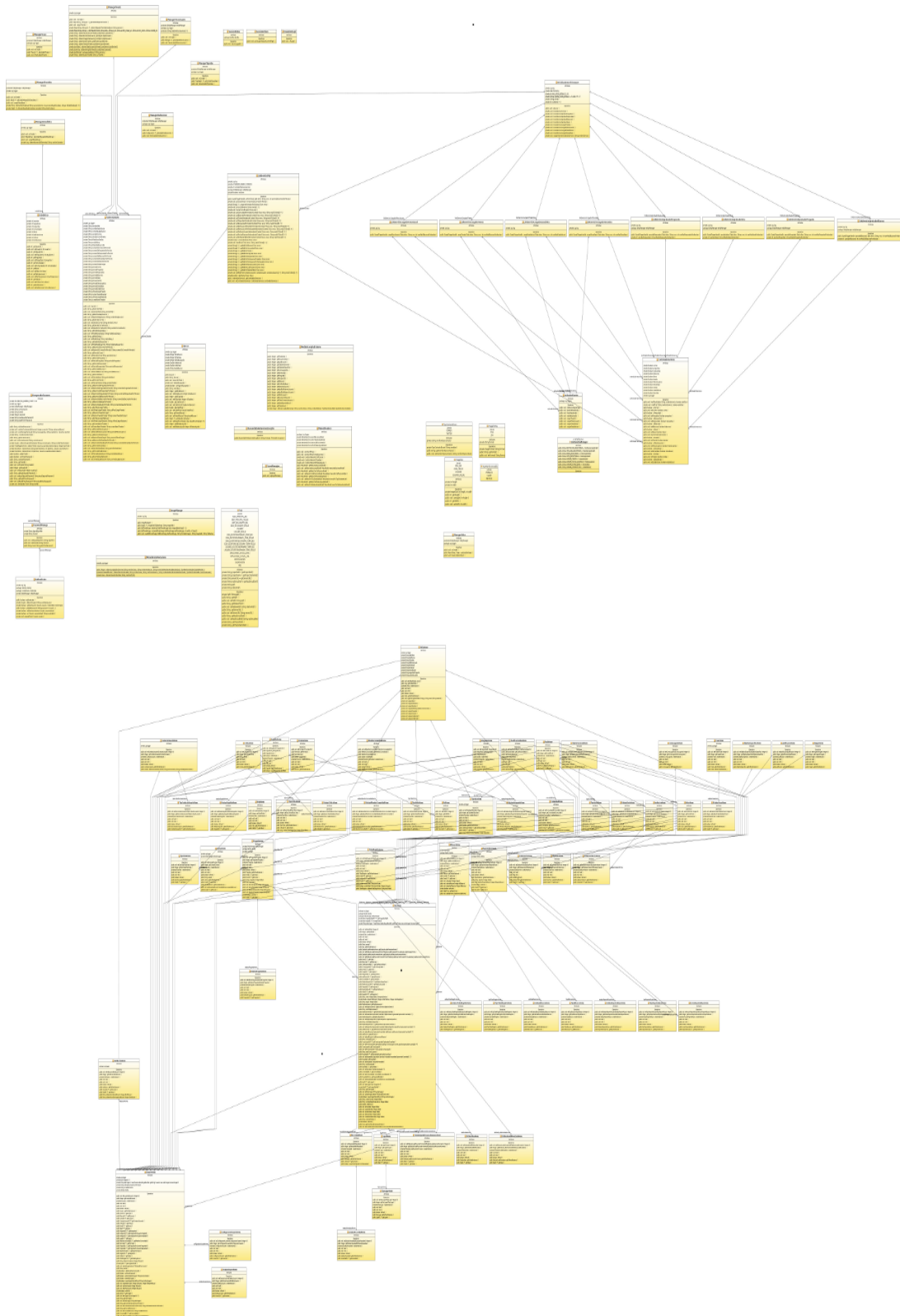


Figura 3: *Diagrama de clases de la aplicación.*

■ Diagrama de estados

En esta sección presentaremos la información correspondiente a las vistas del sistema y al diagrama de estados. La siguiente lista describe la función de cada vista en LinkUAM.

- **ActualizarBitacora (En PanelControl)**
Muestra un formulario en el cuál el usuario pondrá crear nuevas entradas de la bitácora de una idea. Actualizando de esta manera su información.
- **ActualizarBitacoraConfirmacion (En PanelControl)**
Muestra una ventana confirmando la actualización de una bitácora.
- **ADMINISTRADOR (En PanelControl)**
Muestra información, opciones y funcionalidades que solo corresponden a un administrador.
- **Aprobar bitácoras (En PanelControl)**
Muestra al administrador controles para poder activar las bitácoras creadas por los usuarios.
- **Aprobar ideas (En PanelControl)**
Muestra al administrador controles para poder activar las ideas creadas por los usuarios.
- **Aprobar preguntas (En PanelControl)**
Muestra al administrador controles para poder activar las preguntas hechas por los usuarios.
- **Aprobar respuestas (En PanelControl)**
Muestra al administrador controles para poder activar las respuestas hechas por los usuarios.
- **Avisos pendientes (En PanelControl)**
Vista que no tiene funcionalidades actualmente.
- **BuscarIdea**
Muestra al usuario los resultados de las búsquedas, también tiene controles para realizarlas.
- **ComoAyudas**
Muestra a los usuarios información sobre como pueden apoyar al desarrollo del sistema.
- **ContestarPregunta (En PanelControl)**
Muestra al usuario un formulario para poder contestar una pregunta.
- **ContestarPreguntaConfirmacion**
Muestra información confirmando que el usuario ha respondido una pregunta.
- **Datos de usuario (En PanelControl)**
Muestra al usuario la información personal con la que se encuentra inscrito al sistema.
- **Error**
Página genérica de error a donde el sistema redirige cuando ocurre un error grave.
- **Estadísticas**
Muestra estadísticas sobre el sistema como el número de usuarios, seguidores, preguntas, comunidades, ideas, participaciones, instituciones y críticas presentes actualmente en el sistema.
- **Home**
Página inicial y principal del sistema.

- **IdeaCrear**
Muestra controles para poder crear una nueva idea.
- **IdeaDetalle**
Muestra la información detallada de una idea.
- **Ideas (En PanelControl)**
Muestra al usuario las secciones: *Ideas creadas por tí*, *Ideas en las que estás participando*, *Ideas que sigues actualmente*, *Ideas rechazadas*.
- **Ideas creadas por tí (En PanelControl)**
Muestra al usuario las ideas que han sido creadas por el, así como controles para poder ver los seguidores, participaciones y actualizar la biácora de cada una.
- **Ideas en las que estás participando (En PanelControl)**
Muestra al usuario las ideas en las que se encuentra actualmente participando.
- **Ideas que sigues actualmente (En PanelControl)**
Muestra al usuario las ideas que se encuentra siguiendo actualmente.
- **Ideas rechazadas (En PanelControl)**
Muestra al usuario las ideas que fueron rechazadas por un administrador.
- **Licencia**
Muestra al usuario la información sobre la licencia de LinkUAM.
- **Login**
Muestra al usuario controles para poder autenticarse en el sistema.
- **MostrarDatosUsuario (En PanelControl)**
Muestra al usuario los datos con los que se inscribió al sistema.
- **MostrarParticipaciones (En PanelControl)**
Muestra al usuario las participaciones que existen en una de sus ideas.
- **MostrarPreguntas (En PanelControl)**
Muestra al usuario las preguntas que tiene pendientes acerca de sus ideas.
- **MostrarSeguidores (En PanelControl)**
Muestra al usuario los seguidores que tiene en una de sus ideas.
- **NuevoPassword**
Muestra al usuario los controles para introducir un nuevo password y su confirmación.
- **Opiniones**
Muestra al usuario las críticas y opiniones que fueron hechas por la comunidad de LinkUAM.
- **PanelControl**
Muestra al usuario opciones y funcionalidades para poder administrar su cuenta en LinkUAM. En el caso de un administrador, su panel de control también muestra controles para administrar el sistema.
- **Papelera de reciclaje (En PanelControl)**
Muestra controles para poder rescatar ideas y/o usuarios. Permite que vuelvan a estar activos en el sistema.
- **QueEs**
Muestra una explicación de lo que es LinkUAM.
- **QuienEsLinkUAM**
Muestra fotos de 28 usuarios escogidos de manera aleatoria simbolizando a la comunidad entera.

- **Recarga catálogos (En PanelControl)**

Muestra controles para poder recargar en cualquier momento los catálogos del sistema.

- **RecuperarPassword**

Auxilia a un usuario cuando ha olvidado su password. Muestra controles para poder enviar su correo electrónico al sistema y que este le devuelva por correo electrónico un link para poder resetearlo.

- **Reportes de abuso**

Muestra controles para poder reportar una idea cuando no cumple con las políticas de LinkUAM.

- **Rescatar ideas (En PanelControl)**

Muestra controles para poder rescatar ideas de la papelerera de reciclaje.

- **Rescatar usuarios (En PanelControl)**

Muestra controles para poder rescatar usuarios de la papelerera de reciclaje.

- **Servicios del sistema (En PanelControl)**

Muestra controles para poder activar y desactivar los servicios del sistema.

- **UsuarioCrear**

Muestra un formulario con controles para poder inscribirse al sistema.

En la *Figura 4* se presenta el diagrama de estados del sistema. Este representa la navegación que se puede realizar en LinkUAM a través de un navegador web. Por razones de espacio solo es posible describir una vista previa de él. El diagrama completo y a gran tamaño se adjunta con este documento.

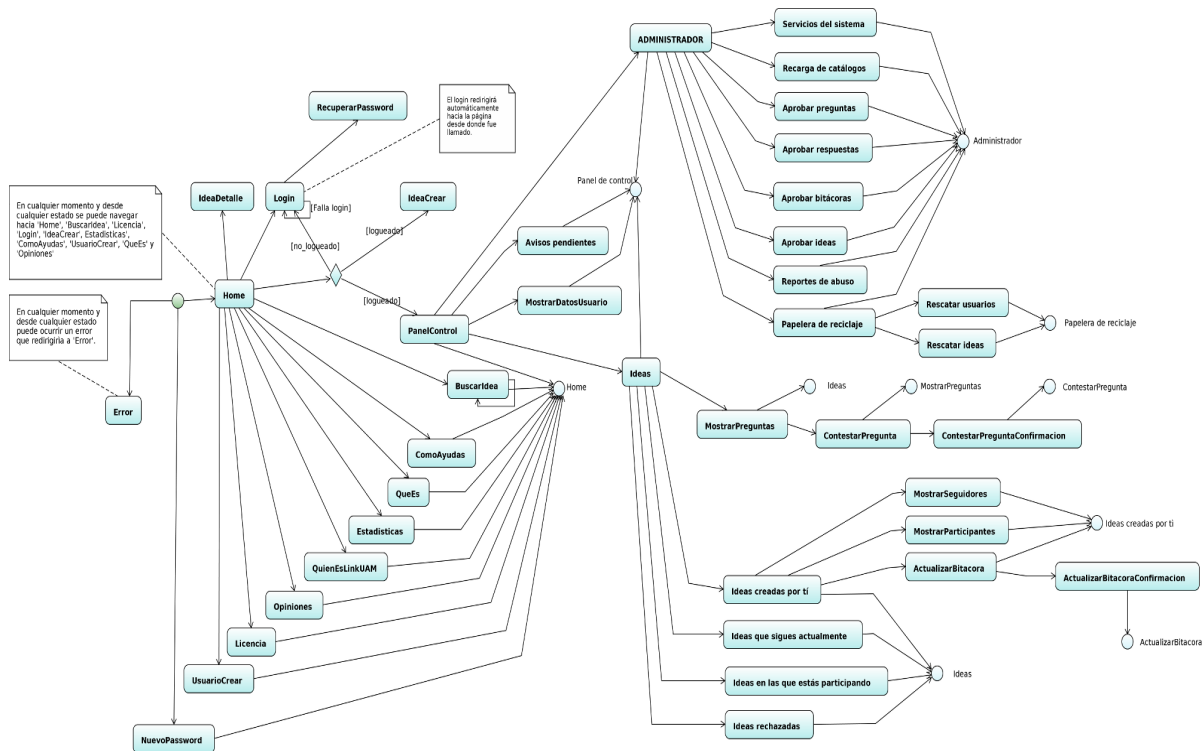


Figura 4: Diagrama de estados de la aplicación.

5.1.3. Módulos del sistema

Hasta este punto del documento hemos descrito los siguientes diagramas:

- **Diagrama de casos de uso**, en el que describimos las principales operaciones que se llevan a cabo en LinkUAM.
- **Diagrama de dominio**, en el que observamos una representación más apegada al mundo real que al informático, de los componentes del sistema y sus relaciones.
- **Diagrama de clases**, en el cual se describe la forma en que las clases del sistema están relacionadas.
- **Diagrama de estados**, en el que se muestran las formas en que se puede navegar a través del sistema con un navegador web.

En la *Figura 5* se describe un diagrama con los principales módulos del sistema tal como se definen en la propuesta inicial de este proyecto terminal. Cabe mencionar que algunos de estos módulos no fueron implantados¹ en la versión final de LinkUAM, optando por posponer su desarrollo para futuras versiones.

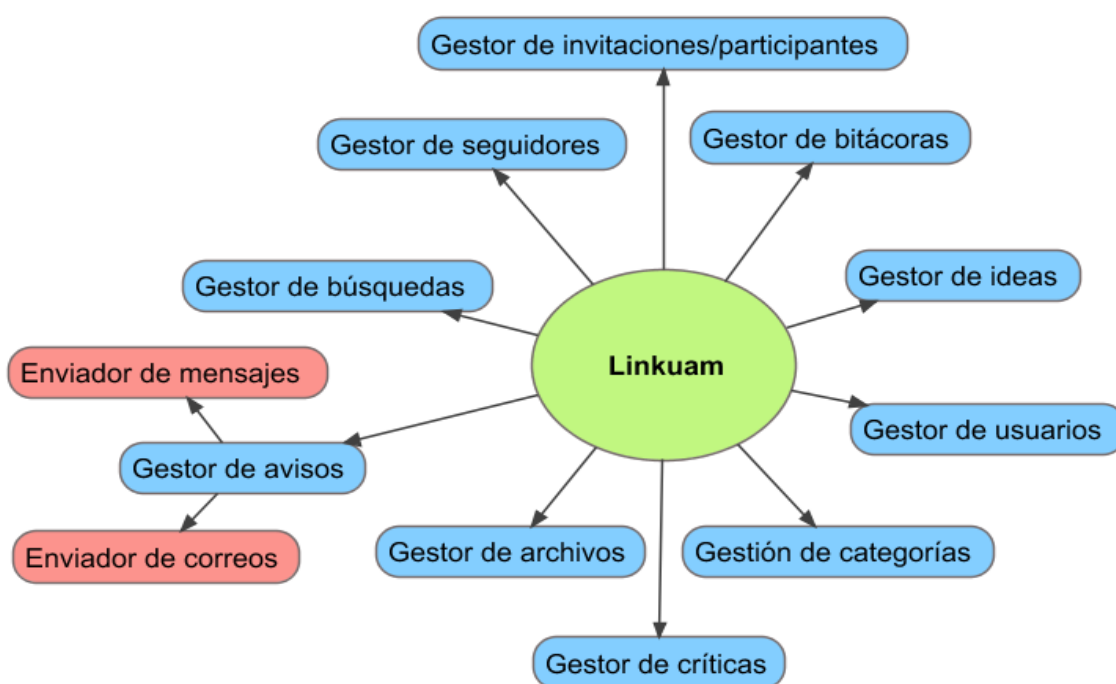


Figura 5: *Módulos principales de la aplicación.*

¹Específicamente, el módulo que no se implantó fue: *Enviador de mensajes*.

5.1.4. Estructura de la base de datos

Ahora describiremos cada una de las tablas de la base de datos de LinkUAM y una vista previa de su estructura.

Tabla	Descripción
accion_reporte_abuso	Tabla que define la acción que se toma cuando un reporte_abuso es tratado.
aviso	Tabla que contendrá los avisos para los usuarios.
bitacora	Tabla que contendrá avisos sobre los principales acontecimientos ocurridos en la idea.
canica	Tabla que define la información del integrante que llenó una necesidad, así como la necesidad llenada.
clasificacion_idea	Tabla que describe los diferentes tipos a los que pertenece una idea.
como_lograrlo	Tabla que contendrá uan explicación acerca de como lograr una idea.
comunidad	Esta tabla contendrá una lista con las comunidades de usuarios del sistema link.
configuracion_usuario	Tabla que contendrá las preferencias de un usuario.
contacto_usuario	Tabla que contendrá los diferentes medios por los cuales comunicarse con un usuario.
critica	Tabla que contendrá las críticas/comentarios hechas al sistema por los usuarios de Link.
envio	Tabla que contendrá los registros de los envíos hechos (y no hechos) por el sistema.
estatus_aviso	Tabla que contendrá los registros que describen los estados que puede tomar un aviso.
estatus_ayuda_monetaria	Contendrá los posibles estatus monetarios que pudiese tener una idea(incluyendo INHABILITADO, PIDO y OFREZCO).
estatus_ayuda_recursos_humanos	Esta tabla contendrá los posibles estados de ayuda relacionada con recursos humanos que pudiese tener una idea (incluyendo el inhabilitado).
estatus_bitacora	Tabla que contendrá los diferentes estados que puede tomar una bitacora.
estatus_comunidad	Tabla que contendrá los diferentes estados que puede tener una comunidad.
estatus_critica	Tabla que definirá el estatus de una crítica hecha por un usuario acerca del sistema.
estatus_envio	Tabla que contiene los registros relacionados con los diferentes estatus que puede tomar un envío.
estatus_frase	Tabla que contiene los posibles estados que puede tomar una clase.

Tabla 1: Descripción de tablas de la base de datos del sistema.

estatus_idea	Tabla que contiene los diferentes estados que puede tomar una idea en determinado momento.
estatus_integrante	Representa el estatus que puede tener el integrante de una idea en determinado momento.
estatus_invitacion	Tabla que contendrá los diferentes estados que puede tener una invitación.
estatus_miembro_comunidad	Tabla que describe los diferentes estados que puede tener un miembro de una comunidad.
estatus_necesidad	Tabla que contendrá los diferentes estatus que puede tener una necesidad.
estatus_pregunta	Contendrá los registros que representen todos los posibles estados que pueda tener una pregunta.
estatus_reporte_abuso	Tabla que define el estatus de un reporte de abuso.
estatus_seguidor	Representa los diferentes estados que puede tener un follower/seguidor de alguna idea.
estatus_ticket_registro	Tabla que describe los diferentes estados que puede tomar un ticket de registro de usuarios.
estatus_usuario	Tabla que contendrá registros que describirán el estado de un usuario.
frase	Tabla que contendrá las frases motivadoras para los emprendedores.
idea	Contiene las ideas presentes en la aplicación.
imagen	Tabla que contendrá registros de nombres de imágenes.
institucion	Tabla que contendrá la institución a la que pertenecerá el usuario.
integrante	Contiene los registros de un usuario preteneciente a las filas de alguna idea.
invitacion	Tabla que contendrá las invitaciones de ingreso a linkuam hechas por usuarios.
justificacion	Tabla que contendrá la justificación de una idea.
lugar	Tabla que contiene los lugares disponibles para ser ocupados por las ideas.
miembro_comunidad	Esta tabla contendrá una lista de todos los usuarios que se hayan inscrito a alguna comunidad, así como la comunidad a la que pertenecen.
necesidad	Tabla que contiene los registros de las necesidades que tiene una idea para realizarse.

Tabla 1: *Descripción de tablas de la base de datos del sistema.*

objetivo_especifico	Tabla que describirá los objetivos específicos de una idea.
objetivo_general	Tabla que describe el objetivo(s) general de una idea.
permiso	Tabla que contendrá los permisos de un usuario. Cada permiso está integrado por una combinación Usuario-Rol.
pregunta	Tabla que contiene las preguntas hechas por los usuarios del sistema en las ideas en las que se encuentren interesados.
reporte_abuso	Tabla que contendrá reportes de información no apropiada dentro del sistema.
rol	Tabla que contendrá los diferentes roles que puede tener un usuario.
seguidor	Contendrá los registros de las ideas que siguen los usuarios.
ticket_registro	Tabla que contiene los tickets para activar un usuario.
tipo_aviso	Tabla que contendrá los registros que describirán los tipos de aviso que pueden existir.
tipo_bitacora	Representa los tipos de mensajes que pueden existir como elementos de una bitácora.
tipo_contacto_usuario	Tabla que contendrá los diferentes tipos de contacto que puede tener un usuario.
tipo_critica	Tabla que contendrá los tipos de crítica que el usuario puede hacer al sistema.
tipo_envio	Tabla que contendrá los diferentes tipos de envío que el sistema puede manejar.
tipo_idea	Esta tabla representa los tipos de ideas a los que una idea puede pertenecer.
tipo_lugar	Tabla que contiene los tipos de lugares disponibles para ser ocupados por las ideas.
tipo_reporte_abuso	Tabla que contendrá los tipos de reporte que podrá hacer el usuario en caso de detectar información no adecuada dentro del sistema.
tipo_ticket_registro	Tabla que contendrá los registros que describen los diferentes estatus que puede tomar un ticket de registro.
tipo_visita	Tabla que contendrá los tipos de visitas manejados por linkUam.
tipo_vista	Tabla que contiene los tipos de vista existentes en el sistema.
usuario	Tabla que contendrá a los usuarios registrados en el sistema.
visita	Tabla que contendrá los registros de las visitas al sitio.
vista	Tabla que contendrá la descripción de las vistas del sistema (.xhtml).

Tabla 1: Descripción de tablas de la base de datos del sistema.

En la *Figura 5*, presentamos la estructura completa de la base de datos. Por razones de espacio solo es posible describir una vista previa de ésta. La estructura completa y a gran tamaño se adjunta con este documento.

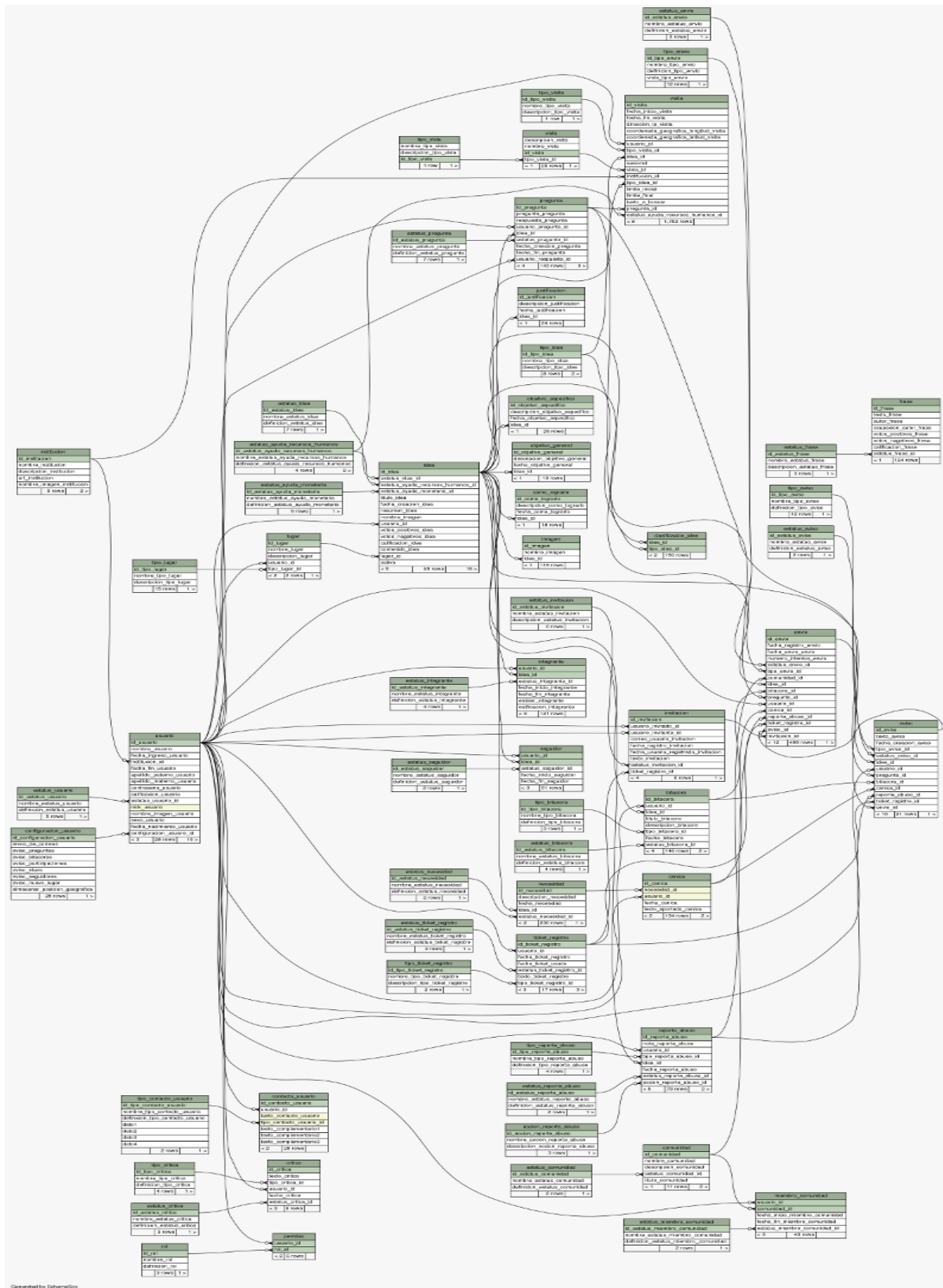


Figura 6: Estructura de base de datos de la aplicación.

5.2. Implementación

5.2.1. Tecnología empleada

- **Software** Para el desarrollo de LinkUAM se siguió la política de utilizar solo software libre y tratando de usar solo las versiones más recientes, dentro de lo posible.

Dada la naturaleza que tienen actualmente las aplicaciones web, algunas tecnologías de software fueron utilizadas para generar la vista del sistema y otras para implantar la funcionalidad del lado del servidor.

- **Tecnologías usadas para generar la vista**

- *PrimeFaces*[13]

PrimeFaces es una implementación más de *JSF*[14], lo que permite utilizarlo en cualquier proyecto donde *JSF* esté presente. Esta tecnología fue utilizada para generar la vista ya que investigando sobre esta tecnología, nos dimos cuenta de que es una implementación que ha retomado los esfuerzos de otras muy importantes como *RichFaces*[15] y *IceFaces*[16], aprendiendo de sus errores y complementando sus funcionalidades.

- *jQuery*[17]

Esta tecnología permitió atacar algunos problemas en la implantación de la vista sin tener que agregar nuevas bibliotecas al sistema¹.

- *JavaScript*[18]

A pesar de que *PrimeFaces* y *jQuery* proporcionan muchas facilidades para el desarrollo de la vista del sistema, *JavaScript* fue utilizado para corregir algunos detalles.

- *SeamFramework*[9]

Es una tecnología que fue utilizada mayormente implantando la funcionalidad del sistema en el servidor, sin embargo, también tiene componentes que facilitan la generación de la vista.

- **Tecnologías usadas para implantar la funcionalidad del lado del servidor**

- *Java*[19]

Java es una plataforma con mucho soporte para el desarrollo de aplicaciones web. Y con soporte no me refiero solo a la cantidad de información que existe sobre ella en la red, sino que muchas empresas y organizaciones aportan bibliotecas de código abierto desarrolladas bajo esta plataforma, que facilitan el desarrollo de proyectos.

- *SeamFramework*[9]

SeamFramework permite utilizar tecnologías difíciles de implantar, de manera relativamente sencilla. Proporciona interfaces de programación sencillas para tecnologías como *Quartz*[20], *JavaMail*[21], *JAAS*[22] entre otras.

- *JBoss AS*[23]

Es un servidor de aplicaciones utilizado con mucha frecuencia en entornos empresariales. Además, tiene mucho soporte y por el historial de versiones que vi en su página, está en constante desarrollo. Por otro lado, es un poco más difícil de utilizar que otros servidores de aplicaciones, pero he leído que es mucho más estable, aunque no lo he comprobado.

- **Hardware**

¹*jQuery* ya estaba presente en el sistema porque *PrimeFaces* apoya mucha de su funcionalidad en esta biblioteca.

Para el desarrollo de este proyecto se dispuso de una computadora con procesador AMD 64 X2 a 1.8 GHZ, 1GB de memoria RAM, sistema operativo Debian y disco duro de 120 GB.

5.3. Conclusiones y perspectivas del proyecto

Se describirán las conclusiones de acuerdo a los objetivos planteados.

Los objetivos específicos de LinkUAM planteados al inicio del desarrollo de este proyecto terminal, fueron 3:

- Desarrollar un módulo que ayude a difundir ideas entre la comunidad universitaria.
- Desarrollar un módulo que ayude a difundir proyectos en proceso de desarrollo, como servicios sociales y proyectos terminales.
- Desarrollar un módulo que ayude a facilitar la comunicación con diferentes comunidades.

Los tres objetivos fueron cubiertos sin embargo, a la fecha no se han puesto a prueba en situaciones reales.

En cuando al objetivo general de este sistema, solo logrando posicionarlo al alcance de la comunidad universitaria, lograremos saber si puede cumplirse.

5.4. Anexos

Se adjuntan 4 diagramas que por razones de espacio no pueden visualizarse cómodamente en este reporte:

- *diagramaDominio.png*
- *diagramaDeEstados.png*
- *diagramaDeClases.png*
- *estructuraDeBaseDeDatos.png*

Referencias

- [1] Real Academia Española, <http://www.rae.es>
- [2] Instituto Tecnológico Autónomo de México (ITAM), “*El papel de las universidades*” <http://biblioteca.itam.mx>
- [3] Red mexicana de emprendedores, <http://www.angelesinversionistas.com.mx>
- [4] Ángel Ventures México, <http://www.angelesinversionistas.com.mx>
- [5] Emprendimiento, Tecnológico de Monterrey, <http://www.itesm.edu>
- [6] Jairo Edgardo Arias Durán, *Resumen del artículo “Desarrollo de la Cultura Emprendedora”* <http://www.emplenet.org.co>
- [7] Annelissie Arrázola M., “*EMPRENDEDURISMO*” <http://produccionintelectual.nur.edu>
- [8] Extreme Programming: A gentle introduction, <http://www.extremeprogramming.org/>
- [9] Seam Framework, <http://seamframework.org/>
- [10] Rational Unified Process, http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational
- [11] Create, read, update and delete, http://en.wikipedia.org/wiki/Create,_read,_update_and_delete
- [12] Hibernate entities, <http://docs.jboss.org/hibernate/>
- [13] PrimeFaces, <http://www.primefaces.org/showcase/>
- [14] JSF, <http://www.oracle.com/technetwork/java/javase/javaserverfaces-139869.html>
- [15] RichFaces, <http://richfaces-showcase.appspot.com/>
- [16] IceFaces, <http://component-showcase.icefaces.org/component-showcase/showcase.iface>
- [17] jQuery, <http://jquery.com/>
- [18] JavaScript, <http://es.wikipedia.org/wiki/JavaScript>
- [19] Java, <http://www.oracle.com/us/technologies/java/index.html>
- [20] Quartz, <http://www.quartz-scheduler.org/>
- [21] JavaMail, <http://www.oracle.com/technetwork/java/javamail/index.html>
- [22] JAAS, <http://es.wikipedia.org/wiki/JAAS>
- [23] JBoss, <http://www.jboss.org/overview.html>