



UNIVERSIDAD DE QUINTANA ROO  
DIVISIÓN DE CIENCIAS E INGENIERÍA

---

**DESARROLLO DE UNA APLICACIÓN PARA LA DIFUSIÓN  
CIENTÍFICA Y TECNOLÓGICA EN RELACIÓN A LAS  
FUENTES DE ENERGÍA SUSTENTABLES**

---

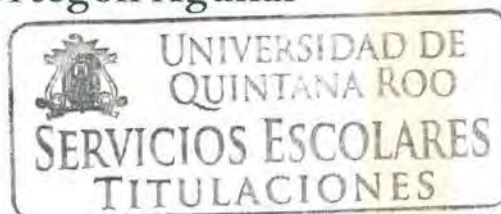
TRABAJO MONOGRÁFICO  
Para obtener el grado de  
**Ingeniería en Redes**

PRESENTA  
**Juan Manuel Alberti Rangel**

ASESORES  
**Dr. Homero Toral Cruz**

**Dra. Daniella Esperanza Pacheco Catalán**

**Dr. Jaime Silverio Ortegón Aguilar**



Chetumal Quintana Roo, México, Marzo de 2014



**UNIVERSIDAD DE QUINTANA ROO**  
**DIVISIÓN DE CIENCIAS E INGENIERÍA**

**TRABAJO MONOGRÁFICO ELABORADO BAJO SUPERVISIÓN DEL  
COMITÉ DE ASESORÍA Y APROBADA COMO REQUISITO PARCIAL PARA  
OBTENER EL GRADO DE:**


**INGENIERO EN REDES**

**Comité de Trabajo Monográfico**

**Asesor:**

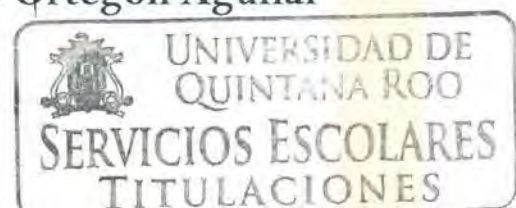
  
\_\_\_\_\_  
**Dr. Homero Toral Cruz**

**Asesor:**

  
\_\_\_\_\_  
**Dra. Daniella Esperanza Pacheco Catalán**

**Asesor:**

  
\_\_\_\_\_  
**Dr. Jaime Silverio Ortega Aguilar**



Chetumal Quintana Roo, México, Marzo de 2014

## Resumen

Las energías sustentables se han convertido en un tema de gran importancia en diversos sectores (industrial, investigación, etc.), debido a los grandes beneficios que prometen, principalmente económicos y su consideración con el medio ambiente. En el sector doméstico, la tendencia es similar, ahora se busca construir viviendas verdes. A pesar de estos beneficios, la difusión que existe para este tema es muy limitada. Hay mucha información disponible pero las plataformas existentes para su esparcimiento son un tanto imitadas, provocando con ello que la información llegue de forma esporádica a la mayoría de la población.

Por otro lado, el Internet juega un papel muy importante en el mundo de la comunicación y la difusión de contenido. Esta plataforma es capaz de compartir de forma casi instantánea información a cualquier rincón del mundo. Si lo comparamos con una revista especializada, el internet tiene mayor ventaja al momento de hacer llegar la información; sin embargo, tiene la desventaja de que no toda la información es fiable.

En la presente monografía se describe el desarrollo de una aplicación web, la cual pretende proporcionar a la población en general, contenido científico y fiable referente a las energías sustentables para poder lograr una mayor participación ciudadana en un tema que nos beneficia a todos. Esta página web también será una plataforma de interacción entre el usuario y los científicos que publiquen información ya que a través de ella podrán externar sus experiencias y comentarios respecto a algún tema relacionado con las energías sustentables.

---

## **Agradecimientos**

Al CONACYT por haber financiado el proyecto SOMEDICYT-191429 y haber asignado una beca para la realización del presente trabajo.

Agradezco especialmente a mis padres, han sido un ejemplo para mí y siempre han estado en todos y cada uno de los eventos que han marcado mi vida.

Agradezco también a mi esposa Ana Lilia por alentarme y darme esa fortaleza necesaria para continuar y alcanzar este objetivo, por estar junto a mí cuando más la he necesitado, ¡gracias Amor!

A mis hermanos, Andrei, Jessie y Guadalupe que han sido mi guía y ejemplo a seguir y que siempre serán mis mejores amigos.

A mis maestros de la Universidad, sin duda les aprendí mucho, más que maestros son compañeros y amigos.

A mis asesores: Dra. Daniella Esperanza Pacheco Catalán y Dr. Jaime Silverio Ortigón Aguilar.

Al Dr. Homero Toral Cruz, por darme la oportunidad de trabajar en este proyecto y por su tiempo dedicado al mismo.

---

## **Dedicatoria**

Este trabajo va dedicado a una personita que cambio mi vida, haciéndola mucho más feliz de lo que ya era, Joana, mi pequeña hija.

Ana Lilia, para ti también va dedicado, sin duda alguna has sido, eres y serás lo más importante en mi vida, gracias por tu comprensión, apoyo, comentarios y alientos que me brindaste durante el desarrollo de este proyecto.

Papá, este trabajo es la culminación del esfuerzo realizado durante mi carrera en la Universidad, sé que te hubiera encantado estar conmigo en estos momentos, pero me basta saber que siempre estuviste orgulloso de mi y de mis hermanos. Por eso dedico a tu memoria este, nuestro triunfo, del cual sin duda eres participe.

Mamá, este logro no es solo mío, es suyo también y tiene gran parte del mérito, me mostro el camino y me impulso para lograrlo.

## Tabla de contenido

<b>AGRADECIMIENTOS</b>	<b>II</b>
<b>DEDICATORIA</b>	<b>III</b>
<b>CAPÍTULO I INTRODUCCIÓN</b>	<b>1</b>
1.1 Justificación	1
1.2 Alcances y restricciones	2
<b>CAPÍTULO II CICLO DE VIDA DEL SOFTWARE</b>	<b>3</b>
<b>2.1. El ciclo de vida clásico</b>	<b>3</b>
2.1.1. Etapas	4
2.1.1.1 Análisis previo	4
2.1.1.2 Análisis de requerimientos	5
<b>2.1.1.3 Diseño</b>	<b>6</b>
2.1.1.4 Programación	7
2.1.1.5 Prueba	7
2.1.1.6 Mantenimiento	8
2.1.2 Lenguajes de cuarta generación	8
2.1.3 Inconvenientes del modelo de ciclo de vida en cascada	9
<b>2.2 Los ciclos de vida iterativos e incrementales.</b>	<b>11</b>
2.2.1. El ciclo de vida con prototipos	11
2.2.2 La programación exploratoria	12
2.2.3 El ciclo de vida del Rational Unified Process	13
<b>CAPÍTULO III ARQUITECTURA CLIENTE-SERVIDOR.</b>	<b>15</b>
<b>3.1 Separación de funciones</b>	<b>16</b>
<b>3.2 Modelos de distribución en aplicaciones Cliente - Servidor.</b>	<b>17</b>
3.2.1 Presentación distribuida	17
3.2.2 Aplicación distribuida	17
3.2.3 Datos distribuidos	18
<b>3.3 Arquitectura de dos y tres niveles.</b>	<b>18</b>
3.3.1 Arquitectura de dos niveles	19

2.3.2 Arquitectura de tres niveles	20
<b>3.4 Bloques básicos en arquitectura Cliente – Servidor.</b>	<b>21</b>
3.4.1 Cliente	21
3.4.2 Servidor	22
3.4.3 Middleware	23
<b>3.5 Descripción de un sistema Cliente – Servidor.</b>	<b>23</b>
<b>CAPÍTULO IV APLICACIONES WEB.</b>	<b>26</b>
<b>4.1 ¿Qué es una aplicación Web?</b>	<b>26</b>
<b>4.2 La parte cliente</b>	<b>27</b>
<b>4.3 La parte servidor</b>	<b>28</b>
<b>4.4 Transferencia de páginas Web</b>	<b>29</b>
<b>4.5 Entornos Web</b>	<b>32</b>
4.5.1 Internet	32
4.5.2 Intranet	33
4.5.3 Extranet	33
<b>4.6 Ventajas y desventajas de las aplicaciones web</b>	<b>34</b>
<b>CAPÍTULO V DESARROLLO</b>	<b>35</b>
<b>5.1 Análisis</b>	<b>35</b>
5.1.1 Introducción.	35
5.1.2 Diagrama de Casos de Uso.	37
5.1.3 Diagrama de clases.	62
<b>5.2 Diseño</b>	<b>66</b>
5.2.1 Lógica de Presentación.	66
5.2.2 Lógica de la aplicación.	67
5.2.3 Lógica de datos.	68
<b>5.3 Implementación</b>	<b>68</b>
5.3.1 Tecnología implementada	68
5.3.1.1 Ruby On Rails	68
5.3.1.2 HTML	71
5.3.1.3 CSS	72
5.3.1.4 JavaScript	72
5.3.2 Descripción	73
5.3.2.1 Inicio	74
5.3.2.2 Artículos	75

5.3.2.3 Investigadores	75
5.3.2.4 Videos	75
5.3.2.5 Foro / Blog	75
5.3.2.6 Empresas	77
5.3.2.7 Contacto	78
<b>5.4 Evaluación del proyecto</b>	<b>78</b>
<b>CAPÍTULO VI CONCLUSIONES</b>	<b>81</b>
<b>CAPÍTULO VII BIBLIOGRAFÍA</b>	<b>83</b>
<b>CAPÍTULO VIII GLOSARIO</b>	<b>84</b>
<b>CAPÍTULO IX ABREVIATURAS</b>	<b>85</b>
<b>CAPÍTULO X ANEXOS</b>	<b>87</b>
Anexo A Código Controladores	87
Anexo B Código Modelos	114
Anexo C Código Vistas	126
Anexo D Código Hojas de Estilo	179



**Índice de Tablas**

Tabla V-1 Registro De Usuario Nuevo .....	45
Tabla V-2 Ver Contenido De La Página Web .....	46
Tabla V-3 Iniciar Sesión De Usuario Registrado .....	46
Tabla V-4 Compartir Información En Redes Sociales .....	47
Tabla V-5 Crear Blog .....	47
Tabla V-6 Crear Una Publicación En El Blog .....	48
Tabla V-7 Editar Publicación Del Blog .....	49
Tabla V-8 Eliminar Blog .....	49
Tabla V-9 Borrar Comentarios De Una Publicación Del Blog .....	50
Tabla V-10 Comentar Publicación En Un Blog .....	51
Tabla V-11 Crear Un Tema Principal De Foros .....	51
Tabla V-12 Crear Un Foro.....	52
Tabla V-13 Borrar Tema Principal.....	52
Tabla V-14 Editar Tema Principal .....	53
Tabla V-15 Borrar Foro .....	53
Tabla V-16 Editar Foro.....	54
Tabla V-17 Editar Foro.....	54
Tabla V-18 Crear Un Hilo / Tema De Discusión .....	55
Tabla V-19 Eliminar Comentario Dentro De Un Foro .....	55
Tabla V-20 Comentar Un Hilo / Tema De Discusión .....	55
Tabla V-21 Agregar Artículos .....	56
Tabla V-22 Borrar Artículo .....	56
Tabla V-23 Agregar Nombre De Empresa .....	57
Tabla V-24 Borrar Nombre De Empresa .....	57
Tabla V-25 Agregar Investigador .....	58
Tabla V-26 Borrar Investigador .....	58
Tabla V-27 Agregar Video.....	59

Tabla V-28 Borrar Video .....	59
Tabla V-29 Editar Video .....	60
Tabla V-30 Asignar Privilegios A Los Usuarios .....	60
Tabla V-31 Eliminar Usuarios .....	61

**Índice de Ilustraciones**

Figura IV-1: Aplicación Web, Estructura Básica ..... 27

Figura IV-2.- Proceso De Transferencia De Páginas Web ..... 31

Figura V-1. Caso De Uso Visualizar Página..... 38

Figura V-2 Caso De Uso Artículos ..... 38

Figura V-3 Caso De Uso Investigadores ..... 39

Figura V-4 Caso De Uso Videos ..... 39

Figura V-5 Caso De Uso Foros ..... 40

Figura V-6 Caso De Uso Foros (Continuación) ..... 40

Figura V-7 Casos De Uso Foros (Continuación) ..... 41

Figura V-8 Casos De Uso Foros (Continuación) ..... 41

Figura V-9 Casos De Uso Blog ..... 42

Figura V-10 Casos De Uso Blog (Continuación) ..... 42

Figura V-11 Casos De Uso Usuarios ..... 43

Figura V-12 Casos De Uso Usuarios (Continuación) ..... 43

Figura V-13 Diagramas de Clases ..... 64

Figura V-14 Diagramas de Clases (Continuación) ..... 64

Figura V-15 Diagrama de Clases (Continuación) ..... 65

Figura V-16 Diagrama de Clases (Continuación) ..... 65

Figura V-17 Diagrama de Clases (Continuación) ..... 66

Figura V-18 Pagina Inicio..... 74

Figura V-19 Esquema Foros ..... 77

Figura V-20 Acceso a página web Safari 7.0 ..... 79

Figura V-21 Acceso a página web Mozilla Firefox 24.0..... 79

Figura V-22 Acceso a página web Google Chrome 31.0.1650 ..... 80

## Capítulo I Introducción

Actualmente, en las redes de comunicaciones existe una enorme cantidad de aplicaciones creadas bajo la arquitectura Cliente – Servidor. La arquitectura Cliente - Servidor es considerada como un modelo en el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos que cooperan entre sí para intercambiar información, servicios o recursos. Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (Clientes) y procesos que responden a esos servicios (Servidores). Estos procesos se pueden ejecutar en el mismo equipo o en diferentes.

En esta arquitectura, el servidor hace el trabajo que demanda más recursos, en cambio el cliente se encarga de manejar las funciones relacionadas con el despliegue, manipulación y recolección de datos. Para ello, se implementan plataformas de interfaces de usuario (GUI, Graphic User Interface). El cliente también es responsable de acceder a los servicios distribuidos en cualquier parte de una red de telecomunicaciones.

En la presente monografía, se plantea el desarrollo de una aplicación basada en la arquitectura Cliente – Servidor, la cual tendrá como objetivo la difusión científica y tecnológica en relación a las fuentes de energía sustentable.

### 1.1 Justificación

Hoy en día, las nuevas generaciones se encuentran más adaptadas al uso de nuevas tecnologías de comunicación e información, basadas en el uso del Internet, tales como: las redes sociales, páginas web, blogs, etc. Esto ha permitido a estas nuevas generaciones, demandar mayor cantidad de

información y sobre todo, que dicha información sea obtenida de la fuente principal y en tiempo real.

Por otro lado, unos de los tópicos de gran interés entre la comunidad científica a nivel mundial es el estudio y la difusión de las diferentes fuentes de energía sustentable.

Derivado de los puntos anteriores, en este trabajo se describe el desarrollo de una aplicación basada en la arquitectura Cliente-Servidor, para hacer la difusión científica y tecnológica de las diferentes fuentes de energía sustentables.

## **1.2 Alcances y restricciones**

Para tal efecto, se desarrollará una aplicación basada en la arquitectura cliente-servidor mediante el lenguaje de programación Ruby y su correspondiente framework Ruby On Rails (RoR), versión 1.9.3 de Ruby, y versión 2.3.2 de Ruby On Rails. Además, se utilizarán estilos en CSS y código de JavaScript para algunas funciones específicas del sistema, bajo el esquema HTML. La aplicación se desarrollará bajo una arquitectura de dos niveles y un modelo de Prestación Distribuida.

La aplicación estará formada por los siguientes módulos:

- Inicio
- Artículos
- Investigadores
- Videos
- Foros/Blog
- Empresas
- Contacto

## Capítulo II Ciclo de vida del software

El desarrollo de aplicaciones en software, implica algo más que la simple programación, existen un conjunto de etapas que la preceden y otras que la siguen.

El ciclo de vida del software está constituido por el conjunto de todas estas etapas. Los métodos y técnicas de la ingeniería del software se inscriben dentro del marco delimitado por el ciclo de vida del software, y, más concretamente, por las diferentes etapas que se distinguen.

La misma existencia de distintos modelos del ciclo de vida del software hace comprender que no hay ninguno que sea ideal o que no tenga grandes limitaciones. Sin embargo, es indispensable que todo proyecto se desarrolle dentro del marco de un ciclo de vida claramente definido, si se quiere tener una mínima garantía de cumplimiento de los plazos, y respetar los límites de los recursos asignados. Además, la garantía de calidad y las certificaciones de calidad también presuponen que el proceso de producción de software se desarrolle según el ciclo de vida establecido con etapas bien definidas.

### 2.1. El ciclo de vida clásico

En la Figura II-1, se presentan las etapas previstas en una versión particular del ciclo de vida clásico, así como los documentos que de ellas resultan.

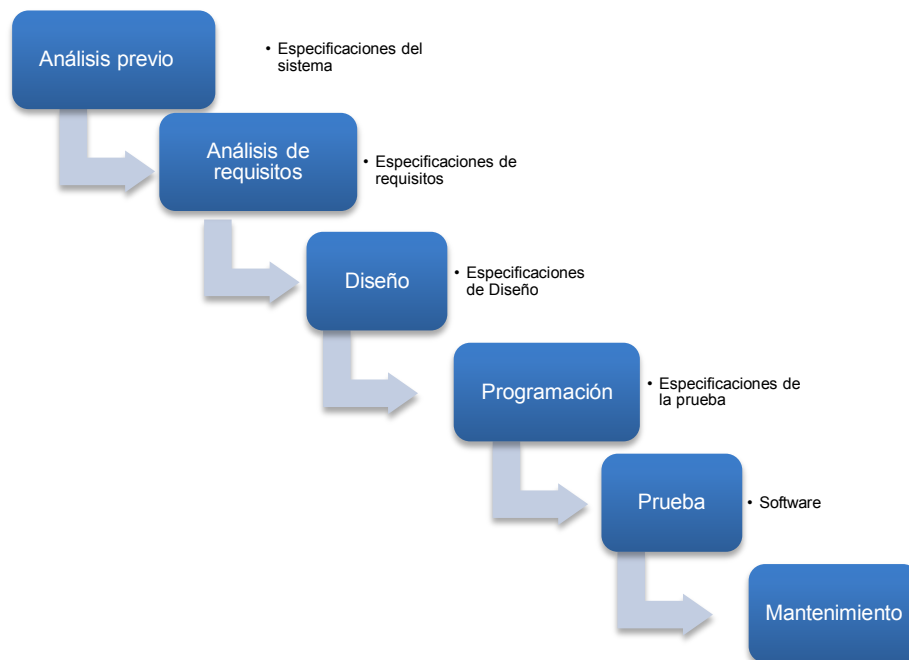


Figura II-1 Ciclo de Vida Clásico

A veces, el ciclo de vida clásico también se denomina ciclo de vida en cascada, lo cual quiere decir que en cada etapa se obtienen documentos, los cuales son la base de partida de la etapa siguiente, por tanto, éstos no pueden comenzar antes de que haya terminado la etapa anterior, así como también nunca regresan a etapas pasadas.

### **2.1.1. Etapas**

El ciclo de vida clásico, se divide en 6 etapas. A continuación se describen cada una de ellas:

#### ***2.1.1.1 Análisis previo***

La primera etapa se denomina análisis previo, y también análisis de sistemas o ingeniería de sistemas. En esta etapa, se definen los grandes rasgos del

sistema de software que tendrá que dar soporte informático a unas actividades determinadas de unos ciertos usuarios dentro del marco más general de la actividad de la empresa u organización (Campderrich Falgueras, 2001).

Además, este sistema tendrá que funcionar en un entorno de hardware y red determinado, que será necesario indicar, y quizá también tendrá que intercambiar información con otro software o compartir una base de datos. Estos hechos constituyen a otros aspectos del entorno del futuro software de los cuales se tendrá que dejar constancia.

Hay que tener en cuenta los recursos necesarios para el desarrollo del software y los condicionamientos temporales, especialmente los plazos impuestos desde fuera del proyecto, que a menudo están determinados por los hechos que han causado las necesidades de información que tiene que satisfacer dicho software y también restricciones eventuales y condiciones adicionales que sea necesario respetar; y en función de todo esto, se evalúa la viabilidad técnica, económica y legal del proyecto de desarrollo de dicho software.

El documento que resulta de esta etapa se denomina especificaciones del sistema, y sirve de base para tomar la decisión definitiva sobre la continuidad del proyecto.

### ***2.1.1.2 Análisis de requerimientos***

La segunda etapa es el análisis de requerimientos o simplemente análisis. Su objetivo es definir con detalle las necesidades de información que tendrá que resolver el software, sin tener en cuenta, por el momento, los medios técnicos con los que se tendrá que llevar a término el desarrollo del software (Campderrich Falgueras, 2001); como el lenguaje de programación, el gestor de base de datos, los componentes que se pueden reutilizar, etc.



En esta etapa detallamos los requerimientos de la etapa anterior; ahora solo pensamos en el software que es necesario desarrollar y sus interfaces con el entorno.

La figura responsable del análisis debe tener o adquirir conocimientos generales sobre el dominio de la aplicación y obtener información de los usuarios y de otras fuentes que le permitan hacerse una idea precisa de las funciones y de los requerimientos en general, del futuro software. Con esta información, se redacta el documento que llamaremos especificaciones de requerimientos, que tiene una doble función: 1) especificar qué debe hacer el software, con la suficiente precisión para que se pueda desarrollar, y 2) servir de base para un contrato, explícito o no, entre el equipo de desarrollo del software y sus futuros usuarios.

### **2.1.1.3 Diseño**

El diseño es la etapa siguiente. Si el análisis especifica el problema o “qué tiene que hacer el software”, el diseño especifica una solución a este problema o “como el software tiene que hacer su función (Campderrich Falgueras, 2001).

Del software, hay que diseñar varios aspectos diferenciados: su arquitectura general, las estructuras de datos (base de datos, etc.), la especificación de cada programa y las interfaces con el usuario; y éstos se tienen que llevar a cabo de manera que en base a todo esto, se pueda codificar el software, de manera similar a la que se realiza la construcción de un edificio o de una máquina a partir de los planos.

El documento resultante es la especificación del diseño. La etapa de diseño es el mejor momento para elaborar la especificación de prueba, que describe con qué datos se tiene que probar cada programa o grupo de programas y cuáles son los resultados esperados en cada caso.

#### **2.1.1.4 Programación**

La programación o codificación, que es la cuarta etapa, consiste en traducir el diseño a código procesable por una computadora. (Campderrich Falgueras, 2001).

Es en esta etapa donde se le da forma real al software, es en realidad cuando se elabora.

El entregable que se genera en esta etapa es el programa propiamente, con todas sus funcionalidades y componentes.

#### **2.1.1.5 Prueba**

La verificación es la última etapa del desarrollo del software y la penúltima del modelo de ciclo de vida del software que hemos considerado.

En esta etapa, se busca comprobar el software desde distintos puntos de vista de una manera planificada y, naturalmente, localizar y corregir dentro del software y su documentación los errores que se detecten.

La prueba se lleva a término en dos fases siguientes:

- En la primera se hacen distintas comprobaciones; primero para cada uno de los programas por separado y, después, por grupos de programas directamente relacionados,
- En la segunda fase se comprueba que el conjunto de programas de los resultados que se esperan, y que lo haga con el rendimiento deseado.

El primer equipo de desarrollo hace la última fase de la prueba, y si los resultados son satisfactorios, entonces se entrega el software al cliente, el cual

puede hacer una prueba parecida por su cuenta y con sus datos, con la finalidad de decidir si acepta el software. Con la aceptación por parte del cliente, se da entonces por terminado el desarrollo.

#### **2.1.1.6 Mantenimiento**

La última etapa del ciclo de vida es el mantenimiento o, si se prefiere, explotación del software, ya que siempre que se utilice el software habrá que darle mantenimiento, es decir, hacer cambios – pequeños o grandes—para corregir errores, mejorar las funciones o la eficiencia, o adaptarlo a un nuevo hardware o a cambios en las necesidades de información (Campderrich Falgueras, 2001).

Puesto que un software puede estar en explotación diez años o más, a menudo el coste total de mantenimiento durante la vida del software es de dos a cinco veces mayor que el costo de desarrollo.

#### **2.1.2 Lenguajes de cuarta generación**

Los entornos de programación de cuarta generación, son de muy alto nivel (en el sentido de que a menudo una sola instrucción, equivale a muchas instrucciones del lenguaje ensamblador), y en gran parte no son procedimientos que están integrados en un gestor de bases de datos relacionales. Incluyen herramientas de dibujo de pantallas, generación de listados y en ocasiones salidas gráficas y hojas de cálculo. Algunos entornos de programación pueden generar código en un lenguaje de tercera generación (Campderrich Falgueras, 2001).

Para aplicaciones sencillas, se puede pasar directamente de los requisitos a la codificación, pero en proyectos complejos es necesario llevar a cabo una etapa de diseño aunque sea simplificada.

### **2.1.3 Inconvenientes del modelo de ciclo de vida en cascada**

El inconveniente del modelo del ciclo de vida en cascada, radica, en que no es realista. Como se ha visto, el modelo de ciclo de vida en cascada tiene un comportamiento, en el cual, las etapas sucesivas del desarrollo se hacen de manera lineal, de forma que una fase no comienza mientras no se haya acabado la anterior, y no se vuelve nunca atrás.

También queda implícito en el modelo que, cuando se acaba una fase, se sabe al menos aproximadamente qué porcentaje del proyecto queda por hacer, ya que si el análisis se ha completado y su resultado es cien por ciento fijo, se puede saber con cierta precisión la duración del diseño e, incluso de la programación.

Ahora bien, en la realidad es posible que la especificación del sistema sea fiable en lo que respecta a las funciones, ya que no se espera que se describan punto por punto. Sin embargo, precisamente por esto último, el coste y al duración del proyecto se han calculado sobre una base muy poco sólida y tiene un gran margen de error.

No obstante, el problema más grave se presenta en el análisis de requerimientos, por el hecho de que éstos casi siempre son incompletos al principio o cambian antes de que se haya acabado de construir el software, y a menudo suceden ambas cosas a la vez. Y si la especificación de requerimientos es incompleta e insegura, es obvio que el diseño y la programación tendrán problemas y, sobre todo, retrasos y aumentos de costo importantes para el trabajo no previsto que se deberá hacer y también para el que será necesario rehacer.

Existen dos razones por las cuales es prácticamente imposible elaborar unos requerimientos completos y estables en el primer intento:

- En primer lugar, es difícil encontrar un conjunto de futuros usuarios que conozcan lo suficiente el entorno en el que se debe utilizar el software, que hayan reflexionado lo suficiente sobre lo que quieren conseguir y que, además, se pongan de acuerdo.
- El segundo lugar, porque el trabajo de consolidación de las peticiones de estos usuarios nunca será perfecto.

En cualquier caso, tenemos que contar con el hecho de que, una vez terminada oficialmente la etapa de análisis y comenzada la de diseño, todavía surgirán requisitos nuevos y cambios en los ya existentes.

Parece que la opción más razonable sea estudiar a fondo una pequeña parte de los requisitos que tenga una cierta autonomía, y diseñarla, programarla y probarla, y una vez que el cliente la haya dado por buena, hacer lo mismo con otra parte, y otra. Si partimos de un software ya construido en parte, se puede esperar que la idea que nos hacemos de los requerimientos restantes pueda ser cada vez más precisa y que también obtengamos una estimación cada vez más segura del costo y de la duración del proyecto completo; esto es lo que denominamos ciclo de vida iterativo e incremental (interactivo porque se repite dentro de un mismo proyecto e incremental porque procede por partes). Y se tendrá que considerar normal que, a veces, cuando se construya una parte, se vea que es necesario modificar una hecha con anterioridad (Campderrich Falgueras, 2001).

Los inconvenientes del modelo del ciclo de vida clásico no quieren decir que no pueda haber plazo o límite de costo para un proyecto de desarrollo de software, simplemente, hay que reconocer que no es realista creer que se pueda fijar de forma exacta la funcionalidad, el costo y la duración del proyecto, todo a la vez. Si el software tiene que funcionar en una fecha determinada y no se puede

aumentar el gasto en personal, será necesario estar dispuesto a aceptar que el software no realice todas las funciones deseadas; si la funcionalidad y la fecha de entrega del programa son innegociables, se tendrá que aumentar el número de programadores o analistas. Y esto no supone de ninguna renuncia en relación con los resultados que se alcanzaban hasta ahora, porque en la práctica muy pocos son los proyectos en los que no se producen desviaciones en cuanto a la funcionalidad, presupuesto o plazo, sino en varias de estas cosas al mismo tiempo.

Por tanto, el modelo de ciclo de vida en cascada puede ser válido si se aplica de manera que cada etapa, del análisis de requerimientos a la prueba, no prevea todo el conjunto del software, sino sólo una parte cada vez; entonces tendríamos un ciclo de vida iterativo e incremental basado en el ciclo de vida en cascada.

## **2.2 Los ciclos de vida iterativos e incrementales.**

El ciclo de vida en cascada ha sido muy criticado y se han propuesto algunos modelos alternativos, a continuación se describen algunos.

### **2.2.1. El ciclo de vida con prototipos**

Para ayudar a concretar los requisitos, se puede recurrir a construir un prototipo del software.

Un prototipo es un software provisional, construido con herramientas y técnicas que dan prioridad a la rapidez y a la facilidad de modificación antes que a la eficiencia en el funcionamiento, la cual, sólo servirá para que los usuarios puedan ver cómo sería el contenido de apariencia de los resultados de algunas de las funciones del futuro software.

Un prototipo sirve para que los usuarios puedan confirmar si lo que les muestran es efectivamente lo que necesitan, o bien lo puedan pedir por comparación, y entonces se prepara una nueva versión del prototipo teniendo en cuenta las indicaciones de los usuarios y se les enseña otra vez. En el momento en que el prototipo ha permitido concretar y confirmar los requerimientos, se puede comenzar un desarrollo según el ciclo de vida en cascada, en este caso, no obstante, partiendo de una base mucho más sólida.

El ciclo de vida con prototipos no se puede considerar plenamente un ciclo de vida repetitivo e incremental, ya que sólo el prototipo se elabora de manera iterativa, y no necesariamente incremental. Sin embargo, es un modelo de ciclo de vida que puede ser adecuado en algunos casos, en especial cuando basta con prototipar un número reducido de funciones para que las otras sean bastantes parecidas a éstas, de forma que las conclusiones a las que se llegue con el prototipo también les sean aplicables.

### **2.2.2 La programación exploratoria**

La programación exploratoria consiste en elaborar una primera versión del software, o de una parte de éste, enseñarla a los usuarios para que la critiquen y, a continuación, hacerle los cambios que éstos sugieran, proceso que se repetirá tantas veces como sea necesario.

La diferencia principal con respecto a los prototipos es que aquí, el software es real desde el principio.

La programación exploratoria se puede considerar un ciclo de vida iterativo, pero no incremental, ya que el software está completo desde la primera versión. Como consecuencia de las numerosas modificaciones que sufre la calidad del software desarrollado de esta manera y su documentación tiende a ser deficiente, como la de un software que haya experimentado un mantenimiento largo e intenso.

### **2.2.3 El ciclo de vida del Rational Unified Process**

La empresa Rational Software ha propuesto este ciclo de vida como marco para el desarrollo de software que utiliza sus herramientas. Es claramente un ciclo de vida iterativo e incremental.

Se distinguen estas cuatro etapas, denominadas fases:

1. **Inicio:** Se establece la justificación económica del software y se delimita el alcance del proyecto.
2. **Elaboración:** Se estudia el dominio del problema, o simplemente dominio (parte de la actividad de la empresa dentro de la cual se utilizará el software), y se tienen en cuenta muchas de las necesidades de información y eventuales requerimientos no funcionales y restricciones; se establece la arquitectura general del software y se realiza una planificación del proyecto.
3. **Construcción:** Se desarrolla todo el producto de forma iterativa e incremental, teniendo en cuenta todas las necesidades de información que debe satisfacer y se desarrolla la arquitectura obtenida en la fase anterior.
4. **Transición:** Comprende la entrega del producto al cliente y el comienzo de utilización; aunque es posible que sea necesario hacer retoques en el software, y añadir nuevas funciones como consecuencia de errores detectados o de requisitos que se habían pasado por alto hasta el momento.

En cada una de estas fases se llevan a cabo, en diferentes proporciones, los siguientes componentes de proceso:

- Levantamiento de requerimientos,
- Analisis y diseño,
- Implementación,
- Prueba.



Cada unidad en la que se ejecutan pocos o muchos de los componentes de procesos es una interacción, y se aplica a un nuevo fragmento de software. Todas las fases tienen iteraciones.

### Capítulo III Arquitectura Cliente-Servidor.

La arquitectura Cliente - Servidor es considerada como un modelo en el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos que cooperan entre sí para intercambiar información, servicios o recursos.

Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (Clientes) y procesos que responden a esos servicios (Servidores). Estos procesos se pueden ejecutar en el mismo equipo o en diferentes.

La arquitectura Cliente – Servidor implica la realización de aplicaciones distribuidas. La principal ventaja que tenemos en esta arquitectura es que nos permite separar las funciones según los servicios disponibles, permitiendo situar cada función en la plataforma que sea más adecuada para su ejecución.

Otras ventajas de esta arquitectura son:

- La existencia de hardware económico. Esta es la ventaja más significativa, ya que a diferencia de una solución centralizada donde se utiliza hardware específico, el costo es considerablemente inferior. Además, con esta ventaja podemos usar hardware/software de diferentes fabricantes reduciendo aún más los costos, y aumentando notoriamente la flexibilidad en la implementación y actualización de una solución.
- Esta arquitectura facilita la integración entre sistemas diferentes y comparte información, permitiendo que máquinas ya existentes puedan ser utilizadas con una interfaz más amigable para el usuario. De esta forma integramos PC con sistemas grandes o medianos, sin la restricción de usar un sistema operativo en específico.

- También tenemos la ventaja de la rapidez con la que se puede implementar el mantenimiento y desarrollo de aplicaciones.
- En las redes de computadoras, esta arquitectura permite que múltiples procesadores puedan ejecutar alguna parte distribuida de una misma aplicación logrando concurrencia en los procesos.
- Se obtiene una mayor escalabilidad en la aplicación.
- Facilita el acceso a los datos sin importar donde se encuentre ni cual sistema operativo está utilizando el usuario.

### 3.1 Separación de funciones

La arquitectura Cliente – Servidor nos permite separar las funciones de procesamiento de una aplicación en tres niveles. Así como se observa en la figura III -1.

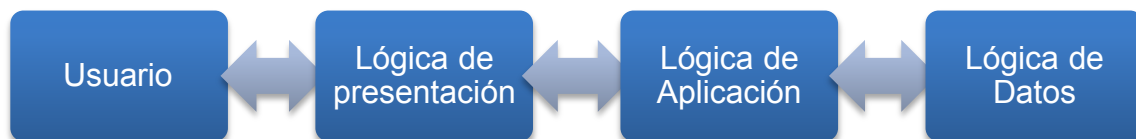


Figura III-1 Separación de funciones: Arquitectura Cliente - Servidor

- Lógica de presentación: esta es una función independiente del resto.
- Lógica de la aplicación: los flujos de trabajo pueden cambiarse de acuerdo a las necesidades o limitaciones de una computadora a otra.
- Lógica de datos: esta gestión debe ser independiente para poder ser distribuida según las necesidades de la empresa.

Si se logra un correcto diseño en un sistema distribuido, los tres niveles antes descritos pueden distribuirse y redistribuirse independientemente sin afectar a la aplicación.

### 3.2 Modelos de distribución en aplicaciones Cliente - Servidor.

De acuerdo a como se distribuyan las funciones básicas de una aplicación (Presentación, Aplicación y Datos) entre el cliente y el servidor, se pueden contemplar tres modelos: Presentación distribuida, aplicación distribuida y datos distribuidos.

#### 3.2.1 Presentación distribuida

En este modelo el cliente solo mantiene la lógica de la presentación, lo demás se ejecuta remotamente. En su forma más simple, es una interfaz gráfica de usuario (GUI) a la que pueden agregar controles para validar datos y con ello evitar la validación de los mismos en el servidor (ver Figura III-2).

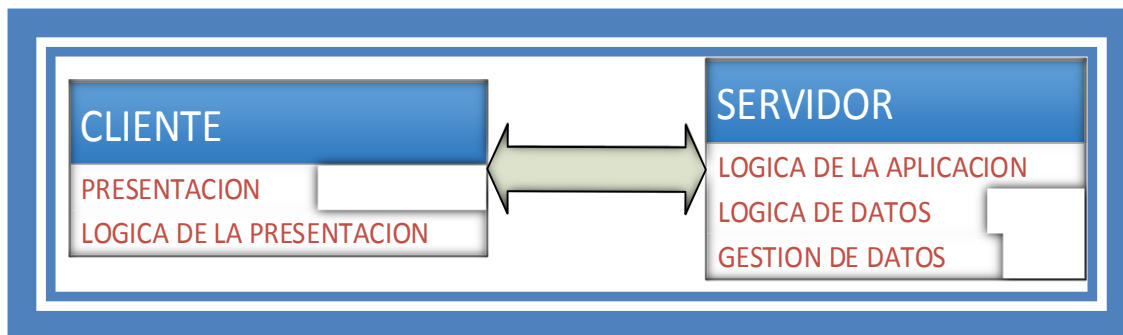


Figura III-2 Presentación Distribuida

#### 3.2.2 Aplicación distribuida

Este modelo es el más flexible, ya que permite tanto al cliente como al servidor mantener la lógica de la aplicación, en donde cada uno realiza funciones que le

sean más propias, ya sea por la estructura o por el rendimiento de la aplicación (ver Figura III-3).

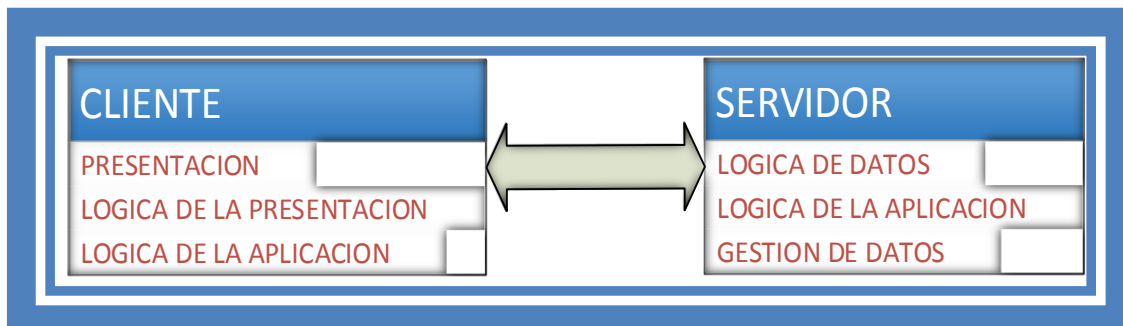


Figura III-3 Aplicación Distribuida

### **3.2.3 Datos distribuidos**

Los datos siempre son lo que se distribuyen, por lo tanto, la lógica de datos queda separada del resto de la aplicación (ver Figura III-4).

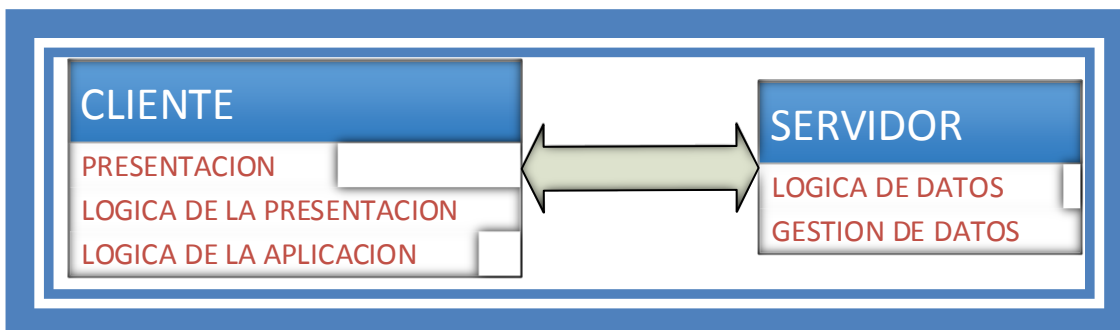


Figura III-4 Datos Distribuidos

### **3.3 Arquitectura de dos y tres niveles.**

Las aplicaciones de dos y tres niveles se distinguen en la manera en que se distribuye la aplicación entre el cliente y el servidor.

### **3.3.1 Arquitectura de dos niveles**

Las arquitectura de dos niveles se basa en un sistema que gestiona la base de datos, donde el cliente mantiene el control de la lógica de presentación, aplicación y de acceso a los datos; el servidor únicamente gestiona los datos. Las aplicaciones de dos niveles, suelen ser cerradas ya que dominan la lógica de los procesos cliente al gestor de la base de datos que se está usando.

En las arquitecturas de dos capas, existen los llamados clientes “gruesos” y los servidores “gruesos”. Estas arquitecturas de dos capas, generalmente se conforman por computadoras que no tienen la característica de un *Mainframe* (computadora central) y un cliente grueso dentro del cual se hace la mayor parte del procesamiento. Entonces, la interfaz gráfica de Usuario (GUI, Graphic Interface User) surge como el entorno que domina en las aplicaciones de escritorio y junto con él, emerge un nuevo planteamiento de la arquitectura de dos capas. Este enfoque plantea que en una red de propósito general, el servidor de archivos debe ser reemplazado por una servidor de bases de datos específico.

La mayor parte del procesamiento tiene lugar en los clientes “gruesos”, pero ahora la información llega al cliente utilizando un lenguaje estructurado de consulta (SQL, Structured Query Lenguaje) a través del cual se realizan peticiones al servidor de base de datos. Este último se limita a mostrar únicamente el resultado solicitado en la consulta previa.

En una relación directa, mientras más compleja sea la aplicación, más grueso debe ser el cliente y más potente debe ser el hardware que lo soporte.

Una configuración alternativa formada por un cliente “fino” y un servidor “grueso” es otra forma de interactuar en una arquitectura de dos capas. En esta configuración alternativa, el cliente invoca los procedimientos almacenados en el servidor de bases de datos. Este modelo de servidor “grueso” tiene un mejor

rendimiento, debido a que la carga de red es todavía pesada, pero más ligera que cuando se trabaja con un cliente “grosso”.

Un inconveniente de esta alternativa es que el uso de procedimientos almacenados hace depender el desarrollo excesivamente del software del vendedor. Otro inconveniente, parte del hecho de que los procedimientos están almacenados conjuntamente con las bases de datos, y cada base de datos contiene los procedimientos que deben modificarse cuando se cambia la lógica de la aplicación. Esto en grandes bases de datos distribuidas, conduce a una administración un tanto complicada.

### **2.3.2 Arquitectura de tres niveles**

En las arquitecturas de tres niveles, la lógica de presentación, aplicación y datos están separadas, de tal forma que mientras la lógica de presentación se ejecuta en el cliente, la lógica de la aplicación y la lógica de datos se distribuyen entre varias computadoras en caso de ser necesario.

El objetivo de aumentar de niveles en una aplicación distribuida, es lograr una mayor independencia entre un nivel y otro, lo cual facilita la portabilidad en entornos heterogéneos.

Una generación más benéfica de la arquitectura Cliente – Servidor añade una capa intermedia (Middle tier). Esta arquitectura se define por las siguientes tres capas de componentes:

- Front – End: Este se encarga de proporcionar la lógica de la presentación y de la interacción con el cliente.
- Back – End: Proporciona el acceso a los servicios dedicados tales como un servidor de bases de datos.
- Middle – Tier: Este permite a los usuarios compartir y controlar la lógica de aplicación a través del aislamiento de la aplicación real.

Una arquitectura multicapa aumenta la arquitectura Cliente – Servidor tradicional mediante la incorporación de uno o más componentes intermedios. El cliente interactúa con las capas intermedias a través de protocolos estándar como HTTP (*Hyper Text Transfer Protocol*).

La capa intermedia interactúa con el servidor de datos (back-end) mediante protocolos especializados. Esta capa alberga la mayor parte de la lógica de la aplicación, traduce las llamadas del cliente en consultas a la base de datos realizando también el proceso de traducción de la base de datos hacia el cliente.

La arquitectura de tres capas se puede extender a  $<n>$  capas, cuando la capa intermedia sea quien soporte las conexiones a diferentes tipos de servicios, integrándolos y acoplándolos al cliente y entre ellos.

### **3.4 Bloques básicos en arquitectura Cliente – Servidor.**

#### **3.4.1 Cliente**

En este bloque, se incluye el sistema operativo dentro de la cual estará una interfaz gráfica de usuario (GUI) o una interfaz orientada a objetos de usuario (OOUI, Object Oriented User Interface) (Coulouris, Dolilimore, Kindberg, & Blair, 2012).

Los clientes se dividen en tres tipos:

- Sin GUI: Estos pueden ser lectores de códigos de barras, demonios, o terminales tontos. Una terminal tonta, puede con tan solo un teclado y una pantalla de salida, aceptar y transmitir datos, o bien desplegarlos desde una computadora o servidor remoto.



- Con GUI: Estos sustituyen a las terminales sin gráficos y normalmente utilizan el modelo objeto/acción, que consiste en la selección de objetos y acciones para que estos se realicen.
- Con OOUI: En la interfaz de usuario orientada a objetos, los usuarios se encargan de manipular la forma en la que los objetos aparecen en pantalla, a este se le conoce como *drag and drop*. Algunos ejemplos de este tipo de clientes son los sistemas operativos Windows, Mac OSX, Genome y algunas páginas de internet que utilicen JavaBeans.

### **3.4.2 Servidor**

El servidor tendrá la responsabilidad de ejecutar únicamente de ejecutar el software especializado para brindar solución a determinadas situaciones. Entre las características de los servidores tenemos las siguientes:

- Debe esperar peticiones de clientes. En ocasiones estas peticiones se convierten en una sesión por cada cliente, en otras ocasiones puede ser en sesiones simultaneas.
- Atender solicitudes simultaneas, es decir debe brindar concurrencia sin que esto represente un riesgo para la integridad de los recursos compartidos.
- Su prioridad es la atención de las solicitudes recibidas.
- Debe tener capacidad de ejecutar tareas en segundo plano que no estén relacionadas con el servicio.
- Debe ser robusto, debe tener la suficiente capacidad en hardware como para estar disponible todo el tiempo atendiendo solicitudes con la misma rapidez todo el tiempo.
- Deben ser escalables y extensibles.

Los servidores se distinguen de dos formas: los servidores de serie o básicos, y los servidores de servicios extendidos.

### **3.4.3 Middleware**

El middleware es un software de conectividad que consiste en un conjunto de servicios que permiten interactuar a múltiples procesos que se ejecutan en distintas máquinas a través de una red de comunicaciones de datos. Los middleware ocultan la heterogeneidad y proveen de un modelo de programación conveniente para los desarrolladores de las aplicaciones.

La organización IETF (Internet Engineering Task Force), define un middleware como:

*“Un middleware puede ser visto como un conjunto de servicios y funciones reutilizables, expandibles, que son comúnmente utilizadas por muchas aplicaciones para funcionar bien dentro de un ambiente interconectado”*

Algunos ejemplos de middleware estándar para dominios específicos pueden ser ODBC/JDBC para bases de datos, Lotus para groupware, HTTP y SSL para Web, CORBA, DCOM y JAVA RMI para objetos distribuidos.

El middleware es un software distribuido especializado en interacciones entre cliente y servidor. Desde la API del cliente que se usa para invocar el servicio, el envío de la solicitud y la respuesta hasta el sistema que informa al servidor no incluye el software que proporciona el servicio ni la interfaz de usuario en el cliente.

El middleware debe incluir los protocolos de transporte, sistemas operativos de red.

### **3.5 Descripción de un sistema Cliente – Servidor.**

En el mundo de las redes con protocolo TCP/IP, las comunicaciones entre computadores se rigen por el sistema Cliente – Servidor. Este modelo intenta

proveer de usabilidad, interoperabilidad y escalabilidad las comunicaciones. También permite distribuir físicamente los procesos y los datos en forma más eficiente, lo que en una computación distribuida impacta directamente en el tráfico de la red logrando reducirlo considerablemente.

Un sistema Cliente – Servidor suele presentar las siguientes características:

- Una combinación de la parte cliente (Front – End) que interactúa con el usuario y la parte del servidor (Back – End) que interactúa con los recursos compartidos (Impresoras, bases de datos, etc....)
- Tanto la parte cliente como servidor tienen diferentes necesidades de recursos a la hora de ejecutarse.
- El entorno normalmente es heterogéneo y multifabricante. El hardware y el software del cliente y servidor suele diferir sin que esto afecte el desempeño. La comunicación se realiza a través de APIs (*Application Programming Interface*) y RPC (*Remote Procedure Call*) conocidos como OBCD (*Open DataBase Connectivity*) para las bases de datos.
- La parte cliente se implementa haciendo uso de una interfaz gráfica de usuario, la cual permite la introducción de datos a través de dispositivos de entrada tales como teclado, ratón, lápiz óptico, entre otros.
- En los sistemas Cliente – Servidor hay una relación de muchos a uno entre los protocolos asimétricos que utilizan para la comunicación los clientes y los servidores, en estos protocolos son los clientes los que siempre inician un diálogo mediante una solicitud, en cambio los servidores esperan sin oposición alguna por esas peticiones.
- En cuanto a encapsulamiento de servicios, los servidores son muy competentes cuando se trata de ejecutar un servicio, el determina como realizar el trabajo.
- Al ser el código y los datos centralizados en un servidor, se tiene como resultado que el mantenimiento sea más económico y la protección de la integridad de los datos compartidos.

- Los clientes y los servidores son procesos débilmente conectados que intercambian solicitudes de servicios y respuestas utilizando mensajes.
- Cuando se obtiene un diseño modular en una aplicación Cliente - Servidor se permite que la aplicación sea tolerante a fallos. Cuando se cuenta con una aplicación tolerante a fallos, estos pueden ocurrir sin causar una caída completa de la aplicación, también uno o más servidores pueden fallar, sin embargo, estas fallas pudieran no ser perceptibles al usuario si se cuenta con servidores que se hagan cargo de la contingencia. Estos servidores deberán tener disponibles los mismos servicios que los servidores caídos.
- Con la modularidad, una aplicación Cliente – Servidor puede lograr responder en automático a la necesidad de incrementar o disminuir la carga en el sistema.
- Los sistemas Cliente – Servidor pueden ser escalados horizontal o verticalmente:
  - Si es horizontal, se añaden o eliminan estaciones del lado del cliente, con un impacto en el rendimiento muy ligero.
  - Si es vertical, entonces se añaden servidores más robustos, o simplemente se agregan más servidores que trabajen en conjunto con los ya existentes.
  - Los servidores pueden compartir servicios a muchos clientes de forma simultánea, y con ello regular el acceso de estos a los recursos compartidos

## Capítulo IV Aplicaciones Web.

### 4.1 ¿Qué es una aplicación Web?

“Las aplicaciones web, son programas diseñados para ser usados completamente dentro del navegador o explorador de internet, con estas aplicaciones hoy en día es posible crear o editar documentos, editar fotografías o escuchar música sin tener la necesidad de instalar algún programa en nuestras computadoras” (Mora, 2001).

Entre las ventajas que ofrecen las aplicaciones web sobre el software de escritorio, es que las primeras son multiplataforma, funcionan de la misma forma sin importar el Sistema Operativo que se esté utilizando. Al ejecutarse a través del navegador, es usuario es libre de acceder a ellas desde el navegador que sea de su agrado.

Las aplicaciones web siempre se mantienen actualizadas y disponibles para utilizarla desde cualquier computadora.

Estas aplicaciones son un tipo especial de aplicación Cliente/Servidor donde el cliente (Navegador Web), el servidor (Servidor Web) y el protocolo (HTTP) deben estar estandarizados y no debería ser creado por el programador.

HTTP es parte de un conjunto de protocolos pertenecientes a la familia TCP/IP (*Transfer Control Protocol / Internet Protocol*). Estos protocolos nos dan la oportunidad de conectar sistemas heterogéneos con lo cual, el intercambio de información entre distintas computadoras se vuelve más sencillo.

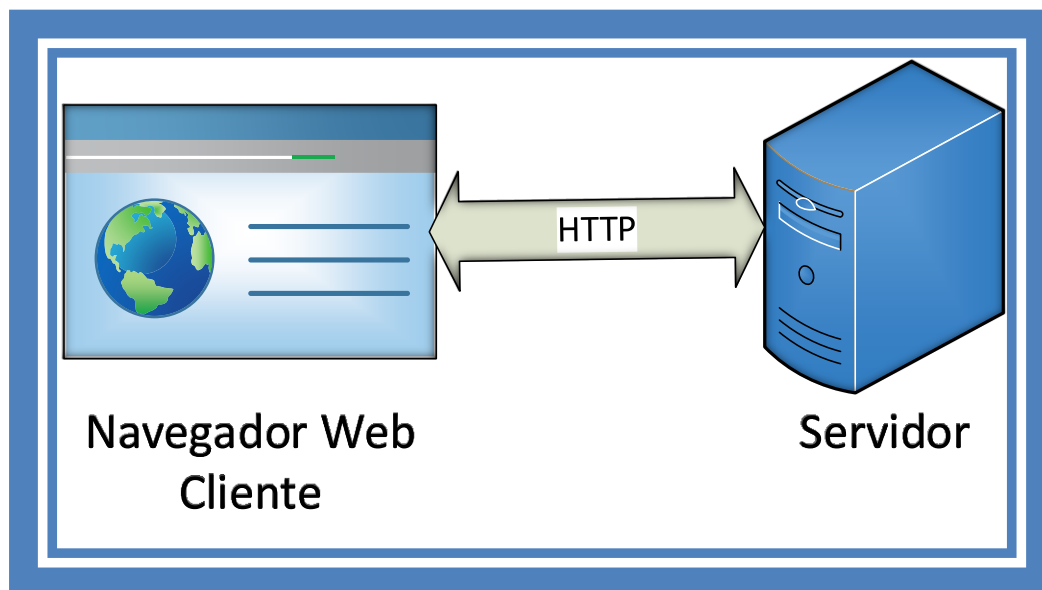


Figura IV-1: Aplicación Web, Estructura Básica

## 4.2 La parte cliente

En nuestra estructura básica de aplicación Web (ver Figura IV-1), el cliente será la parte de la aplicación que se encargará de interactuar con el usuario, gestionando sus peticiones hacia el servidor web mediante los protocolos que se usen, por ejemplo HTTP.

HTTP es un protocolo que permite dicha interacción y el cliente no está limitado a únicamente ese protocolo; sin embargo, puede realizar peticiones de transferencias de archivos (FTP), lecturas de correo electrónico (SMTP– POP–IMAP) o lecturas de noticias en la red (NNTP), por mencionar algunos.

El cliente en las aplicaciones web está formado principalmente por código HTML que da forma y estructura a la página web, además se agrega código ejecutable en un lenguaje que el navegador puede comprender (SCRIPTS) o se pueden incrustar pequeños programas hechos en JAVA llamados APPLETS. También se pueden incluir PLUG-INS, los cuales permiten visualizar contenido

multimedia como FLASH, aunque estos últimos en ocasiones presentan problemas de incompatibilidad en un ambiente multiplataforma.

En conclusión, el cliente web dentro de la aplicación tiene la misión de interpretar el código HTML y los diferentes recursos que contiene, tales como Imágenes, Sonidos, Formato de la página, Colores, tipo de letra, etc.)

### **4.3 La parte servidor**

El servidor dentro de la estructura básica de aplicaciones web es un programa que está a la espera permanente de peticiones provenientes del cliente a través del protocolo HTTP. En los sistemas basados en UNIX se llaman “Demonios” (Daemon) y en los sistemas Microsoft se les denomina “Servicios”.

El servidor se compone de páginas web estáticas, es decir siempre muestran el mismo contenido, también está compuesta por programas o Scripts los cuales se ejecutan dentro del servidor web cuando el cliente solicita alguna página. El resultado de ese script suele ser una página HTML estándar enviada al navegador del cliente. Los scripts ejecutados en el servidor están basados en la tecnología CGI (Common Gateway Interface).

Con CGI se volvía complejo y laboriosa la programación, esto debido a que HTTP no almacena el estado entre conexiones, por lo que era el programador el que tenía la responsabilidad de conservar dicho estado programando tareas ajenas al objetivo de la aplicación, lo que en muchas ocasiones era la causa de problemas diversos.

Microsoft entra al mundo del internet en 1995 y con ello saca al mercado su servidor IIS (Internet Information Server), con esto un nuevo campo para las aplicaciones web surge, el ISAPI (Internet Server Application Program Interface). Se trata de una colección de funciones que el servidor web pone para que los

programadores utilicen. Con ISAPI se podían crear DLL (Dynamic Link Library), las cuales se encargan de invocar los archivos que son requeridos únicamente,

El sistema ASP es una DLL del tipo ISAPI que se ejecuta automáticamente con los archivos cuya extensión es “.asp”, esta DLL interpreta el código que contiene como un script que debe ejecutarse en el servidor, este no implica que interpretara directamente el código, sino que en virtud del lenguaje en el que está escrito determina cual DLL deberá invocar para que realice la ejecución del código que contiene el script, posteriormente toma la información de salida que regresa el script y la envía al servidor web, el cual la reenvía al cliente.

Existen también otras tecnologías que están diseñadas para programar la parte del servidor de las aplicaciones web, tales como JSP (Java Server Pages), servlets, PHP, Ruby On Rails, entre muchas otras.

Ruby On Rails es la tecnología que se usara para el desarrollo de la aplicación web, que se describe en la presente monografía.

#### **4.4 Transferencia de páginas Web**

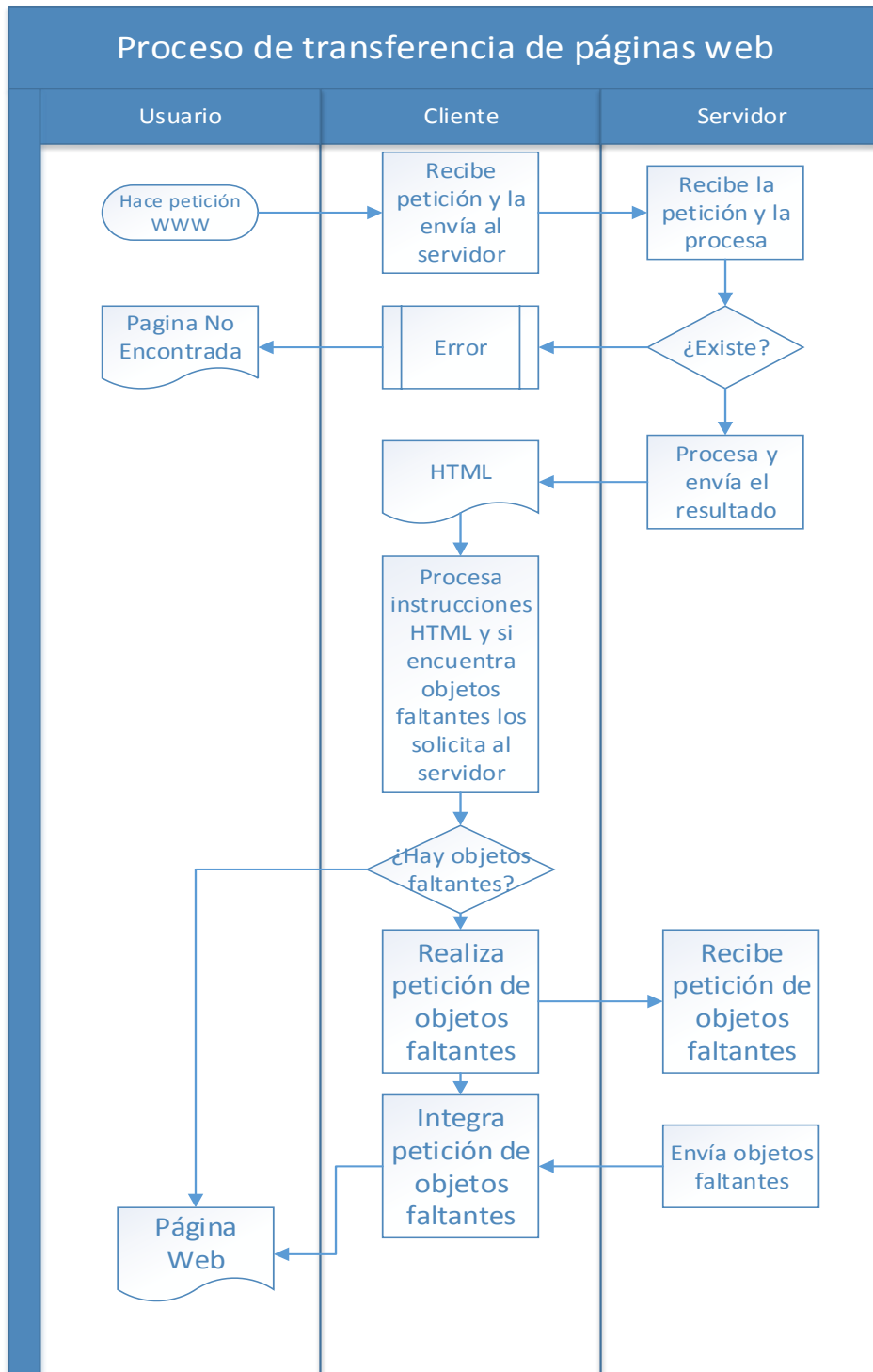
A continuación, se describe el proceso necesario desde que el usuario hace una petición hasta que el cliente muestra dicha página al usuario, como se ilustra en la Figura IV-2.

- El usuario escribe la dirección del sitio al cual quiere acceder. URL (Universal Resource Locator).
- El cliente establece una conexión con el servidor y solicita la página u objeto.
- El servidor envía la pagina/objeto si este existe, en caso contrario devuelve un código de error.
- El cliente inicia el proceso de interpretación de los códigos HTML, si es el caso. Si el cliente encuentra instrucción que hagan referencia a objetos



multimedia (Audio, Video, Fotografías, etc...) establece automáticamente comunicación con el servidor que contiene dichos objetos para solicitarlos. El servidor no necesariamente es el mismo que almacena la aplicación, puede este ser un servidor de algún proveedor de servicios de almacenamiento en línea (Cloud).

- Se cierra la comunicación con el servidor
- El cliente muestra al usuario la página.



**Figura IV-2.- Proceso De Transferencia De Páginas Web**

## **4.5 Entornos Web**

Cuando hablamos de aplicaciones web, es casi instantáneo pensar en Internet como el único entorno en cual se ejecutan dichas aplicaciones, sin embargo existen otros medios en los que también se puede trabajar con las aplicaciones web, los cuales son usados de acorde a las necesidades de la empresa.

A continuación describiré los entornos web más comunes y de mayor demanda en el mundo. La descripción abarcará únicamente aspectos generales y no entrará en detalles, puesto que sería demasiado extenso.

### **4.5.1 Internet**

La historia de internet data desde el desarrollo de las redes de comunicación, lo cual ocurre en la década de los 60's. La principal idea era la de crear una red que permitiera la comunicación en general entre usuarios de varias computadoras ubicados en diferentes regiones. (Ballard, 1982)

Las versiones más antiguas de esta idea surgen a finales de los años 50's, pero fue hasta finales de los 80's y a lo largo de los 90's cuando se comenzaron a dar las implementaciones prácticas. A partir de los 90's con la introducción del WWW (World Wide Web), internet cambia y se convierte en lo que conocemos actualmente.

Internet, al contrario de otros servicios en línea tiene un diseño descentralizado, en el que cada computadora dentro de la internet es independiente. Es de sorprender que este sistema sin aparentemente tener orden funcione perfectamente.

### **4.5.2 Intranet**

Una intranet es una red de dispositivos que principalmente está basada en los protocolos del conjunto de TCP/IP. Esta red pertenece a una empresa u organización y sirve para difundir información de manera local a los usuarios, los cuales deberán tener autorización para poder acceder a la intranet.

La intranet puede o no tener acceso a internet, esa condición se determina de acuerdo a las necesidades de la empresa u organización y a los objetivos de las aplicaciones web.

Los sitios web que se encuentran dentro de las intranet funcionan de la misma forma que lo harían estando en la internet, pero en la intranet, un Firewall protege al sitio de ataques o accesos no autorizados.

Las intranets sirven al igual que el internet, para distribuir y compartir información. Hoy en día las empresas u organizaciones optan por las intranets, lo que las convierte en el segmento de mayor demanda y crecimiento dentro de internet, debido a su bajo costo de instalación y administración.

### **4.5.3 Extranet**

Una extranet no es más que una intranet en la que usuarios ajenos a la empresa u organización pueden acceder parcialmente.

A diferencia de una intranet, las extranet no se encuentran detrás de un firewall y solo es posible acceder a ellas si poseen un nombre de usuario y contraseña para identificarse. Esta identificación revelará al usuario su nivel de acceso a la intranet y las partes que serán visibles para él.

Las extranet se han convertido en el medio más usado por las empresas u organizaciones que comparten información entre ellas. También son usadas como medio de comunicación con clientes, proveedores o socios. Las extranet

son la base del comercio electrónico entre empresas B2B (Business to Business).

#### **4.6 Ventajas y desventajas de las aplicaciones web**

Cuando se trata de gestión de software, encontramos una ventaja enorme en las aplicaciones web debido a que las actualizaciones se realizan directamente en el servidor y no en el cliente. Con esto reducimos la gestión del número de usuarios que usan a aplicación a solo uno, el servidor. El beneficio no solo se refleja en el tiempo ahorrado, también en el costo, por ejemplo, si tenemos a cargo la red de una empresa, la cual tiene varios edificios ubicados en diferentes áreas geográficas, hacer una actualización usuario por usuario se convierte en una tarea difícil, sin embargo, con las aplicaciones web resulta mucho más fácil.

Se evita también la gestión de versiones, reduciendo a cero las posibilidades de que los clientes trabajen con distintas versiones de la aplicación.

También para las empresas, resulta eficiente el hecho de que se puedan integrar las aplicaciones a internet sin necesidad de invertir en herramientas o equipos adicionales al que ya tienen.

Otra ventaja, es contar con independencia de plataforma, en donde el cliente únicamente necesita un navegador web, sin que ello conlleve a generar una adaptación de código. La aplicación se ejecuta correctamente sin importar el Sistema Operativo que el cliente utilice.

Años atrás las aplicaciones web tenían la desventaja de ser menos versátiles que las aplicaciones de escritorio, incluso eran consideradas como aplicaciones de solo lectura. Esta situación ha ido desapareciendo con la aparición de tecnologías como java, JavaScript y ASP entre otras.

## Capítulo V Desarrollo

En el presente capítulo se describen las etapas del ciclo de vida del software de un caso de estudio, el cual consiste en el desarrollo de una Aplicación para la Difusión Científica y Tecnológica en Relación a las Fuentes de Energía Sustentables.

### 5.1 Análisis

#### 5.1.1 Introducción.

En esta etapa del ciclo de vida del software, el análisis, se mostrará el diagrama de clases y el diagrama de casos de uso, con ello se busca describir de forma clara la estructura del sistema, de cómo trabaja y como se desea que trabaje. Esto es muy importante, ya que aquí establecemos todos los criterios que deben de tomarse en cuenta a la hora de programar.

Los casos de uso describen un sistema en función de sus distintas formas de utilizarlo, cada una de las cuales es conocida como un caso de uso. Los casos de uso se componen por una secuencia de eventos que el usuario se encarga de iniciar.

Para comprender los casos de uso de un sistema, primero debemos saber cuáles son los usuarios. Se definen los Actores como el tipo de usuario que estará involucrado en cada caso de uso. Los actores son agentes externos al sistema.

El actor y el caso de uso, forman los elementos básicos de este diagrama.

Los actores son entidades diferentes a los usuarios, los usuarios representan a personas reales interactuando con el sistema, en cambio los actores muestran las funciones que una persona real puede realizar en el sistema.

A continuación, se muestran los actores que se utilizarán en el desarrollo de nuestra aplicación y se describe brevemente cada uno de ellos:

- **VISITANTE:** El visitante únicamente podrá visualizar cierto contenido de la página, tendrá la opción de registrarse para poder acceder a las demás funciones que la página ofrece.
- **PARTICIPANTE:** El participante podrá visualizar el contenido de la página y de los módulos, sin embargo, únicamente podrá hacer comentarios en foros y blogs, además de poder personalizar su perfil.
- **COLABORADOR:** El colaborador será un usuario con privilegios suficientes para poder crear, editar y borrar su blog personal, crear temas, foros e hilos de discusión dentro de los foros, además podrá editar cada una de las opciones del foro. También podrá eliminar comentarios de otros usuarios cuando considere que se ha violado las reglas del sitio, o cuando el comentario sea una ofensa o insulto. La función del colaborador, será la de participar activamente en la administración del contenido de la página.
- **ADMINISTRADOR:** El administrador será el usuario que tenga privilegios totales en el sitio. Además de poder realizar las tareas del colaborador, el administrador puede elevar los privilegios de los usuarios, crear y borrar empresas, crear y borrar investigadores y eliminar usuarios. La función del administrador será la de mantener las operatividad del sitio.

Por otro lado, los diagramas de clase nos servirán para poder determinar las relaciones que existen entre las clases que involucran el sistema, estas relaciones pueden ser asociativas, de herencia, de uso y de contenido.

### **5.1.2 Diagrama de Casos de Uso.**

A continuación, en las Figuras V-1 – Figuras V-12, observaremos los casos de uso que existen en nuestro sistema y cómo interactúan con los actores. La descripción de los casos de uso en las imágenes se presenta en función a los módulos con los que cuenta el sistema. Dichos módulos son:

- Inicio
- Artículos
- Investigadores
- Videos
- Foros/Blog
  - Foros
  - Blogs
- Empresas
- Usuarios (Este módulo no es completamente visible para todos los usuarios).



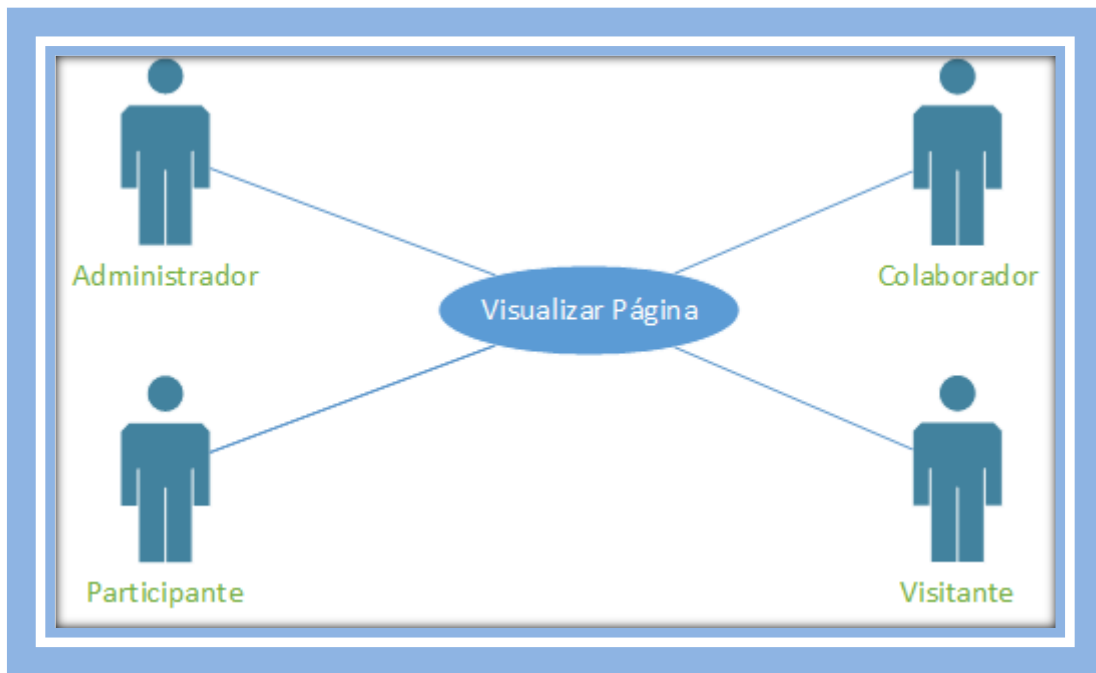


Figura V-1. Caso De Uso Visualizar Página

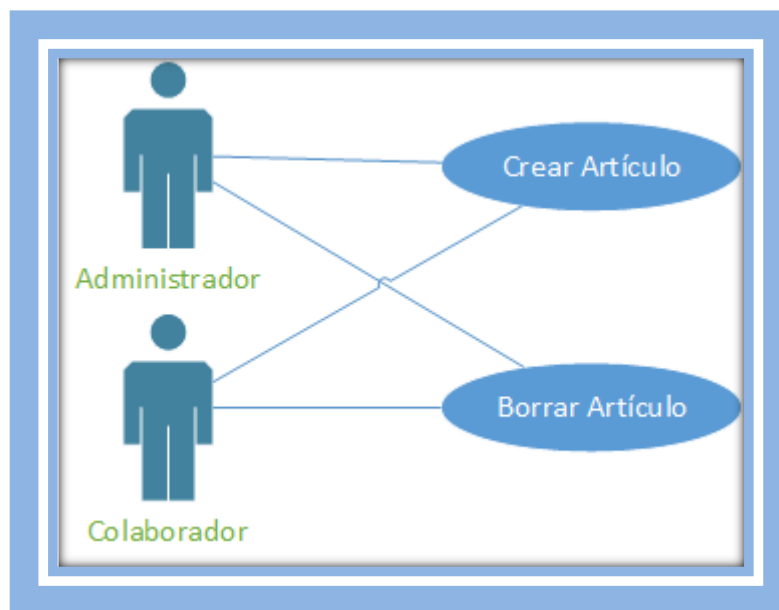


Figura V-2 Caso De Uso Artículos

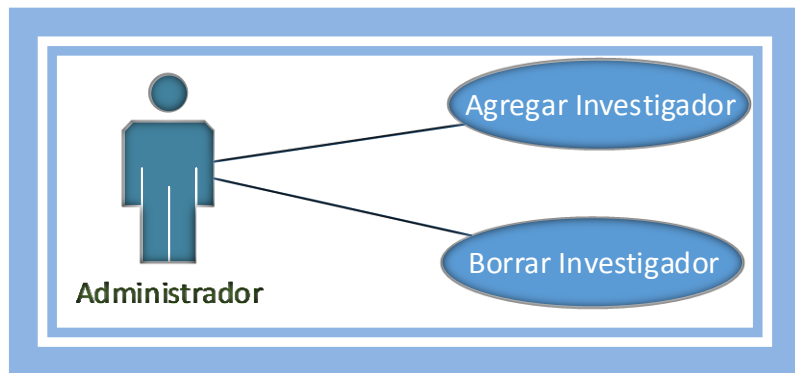


Figura V-3 Caso De Uso Investigadores

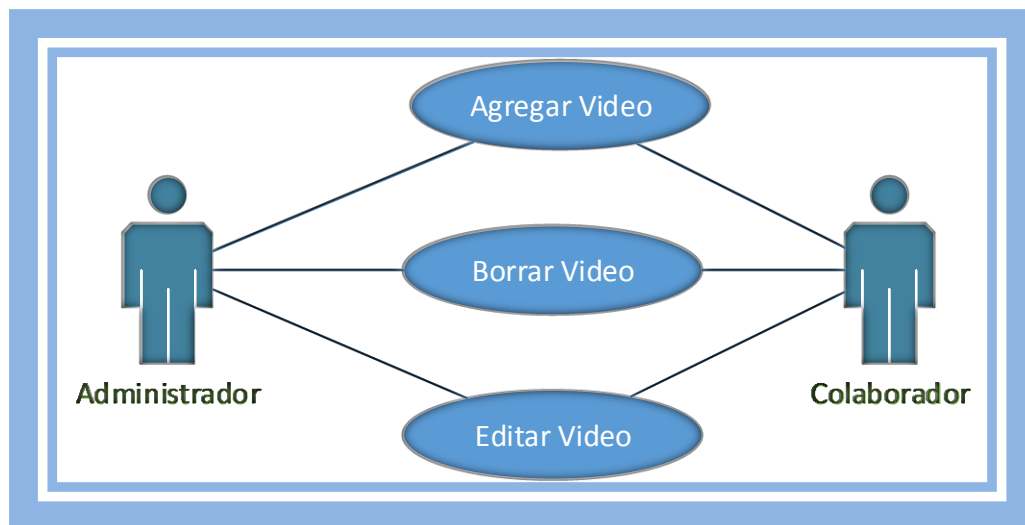


Figura V-4 Caso De Uso Videos

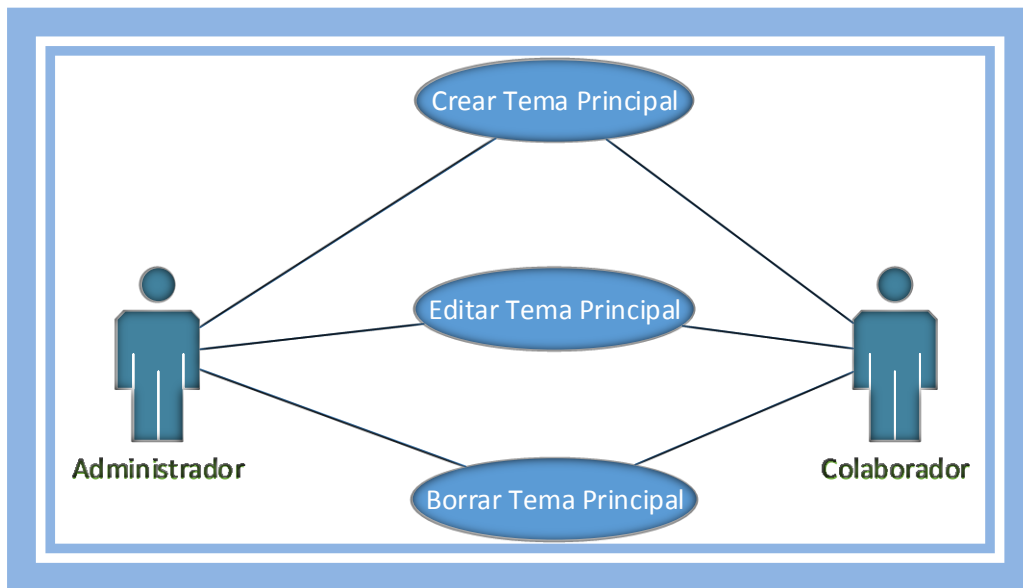


Figura V-5 Caso De Uso Foros

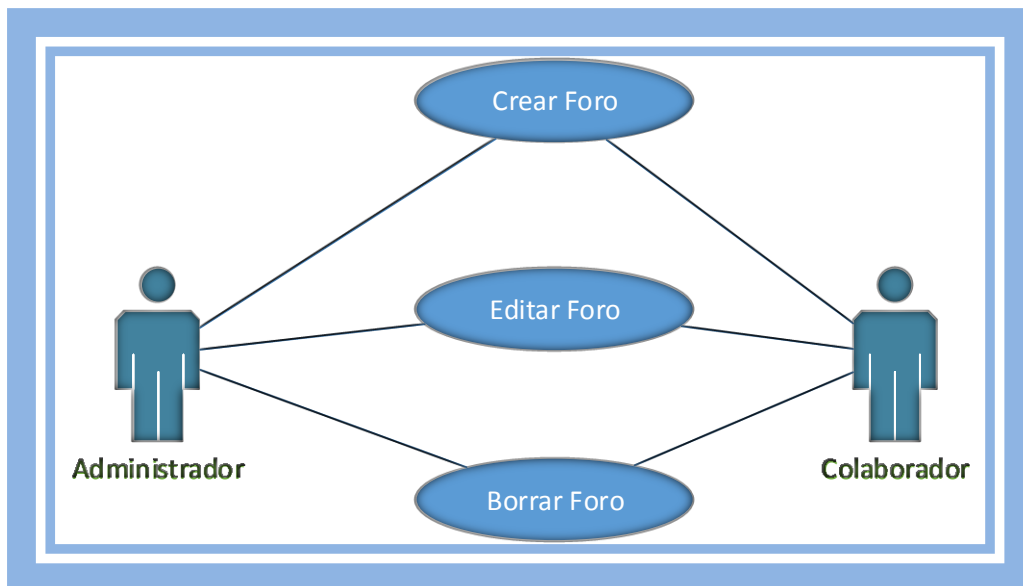


Figura V-6 Caso De Uso Foros (Continuación)

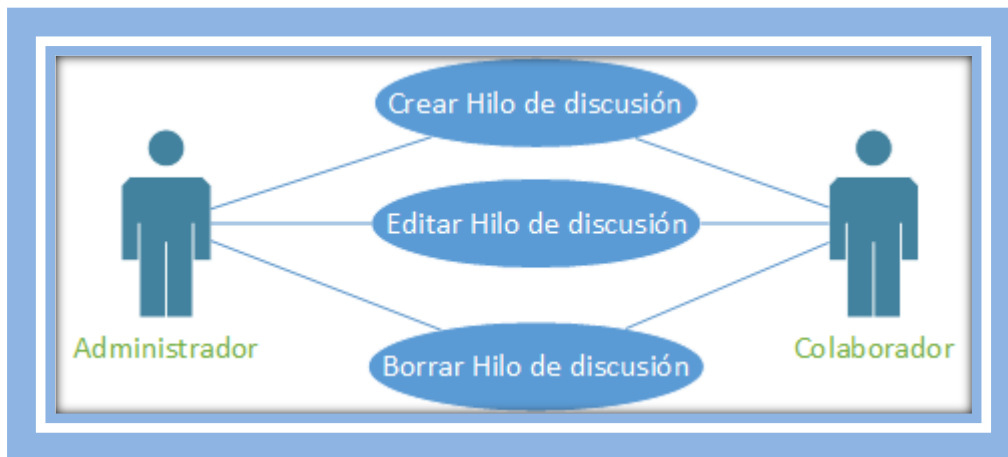


Figura V-7 Casos De Uso Foros (Continuación)

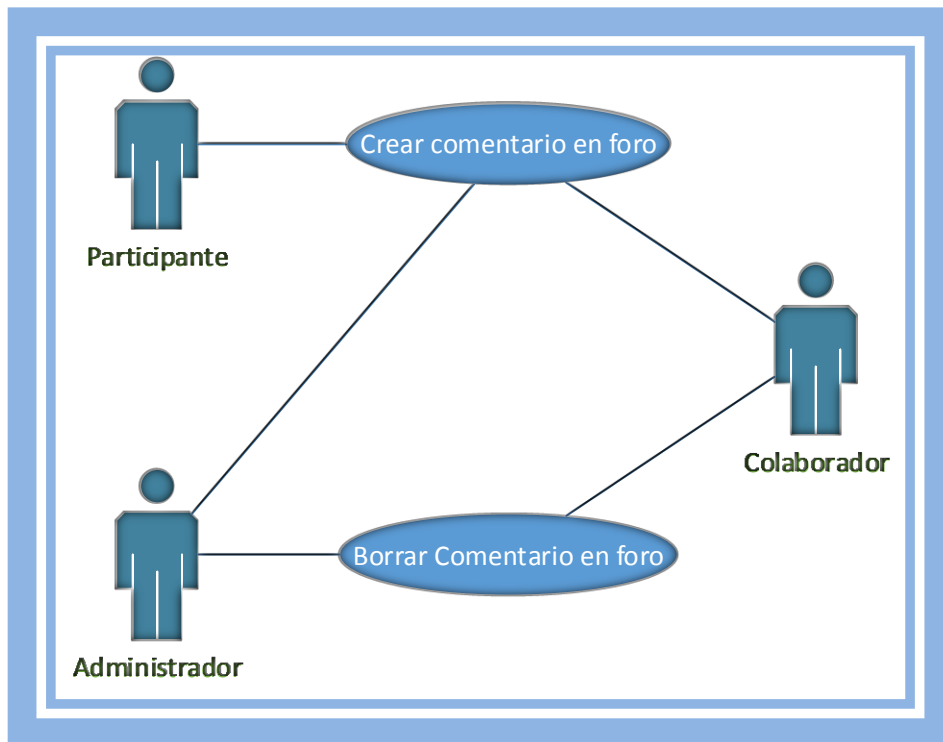


Figura V-8 Casos De Uso Foros (Continuación)

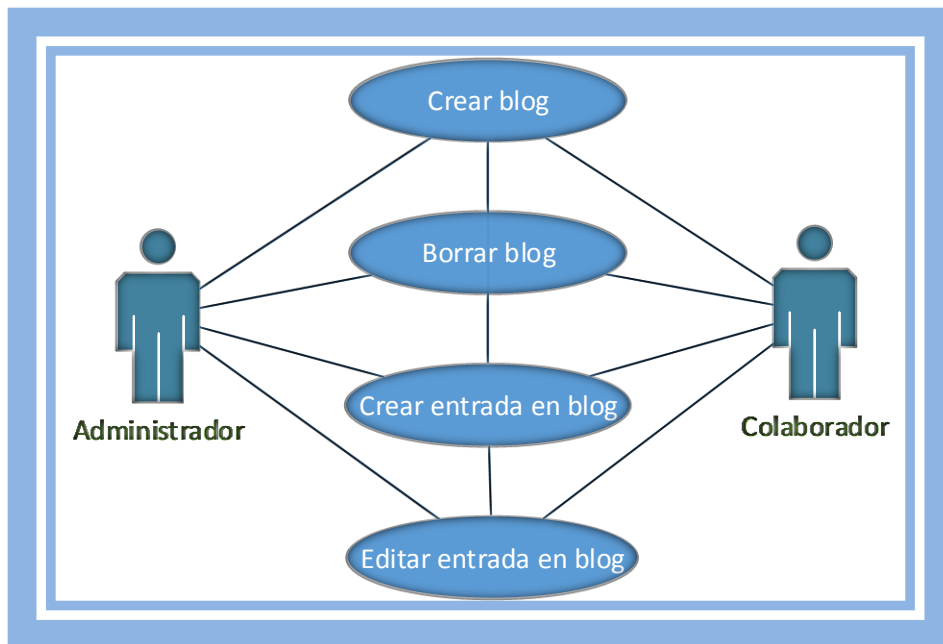


Figura V-9 Casos De Uso Blog

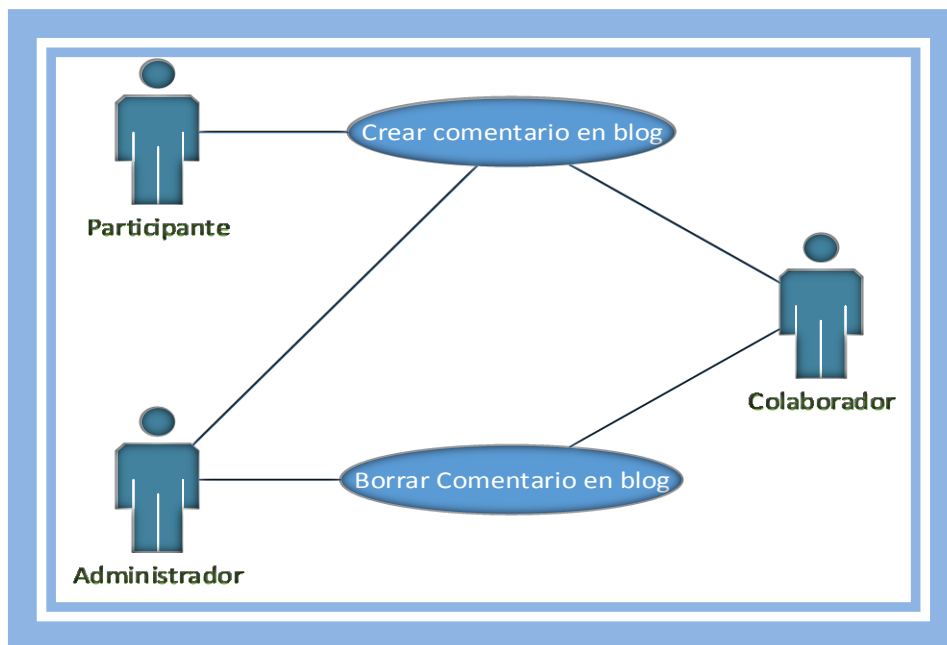


Figura V-10 Casos De Uso Blog (Continuación)

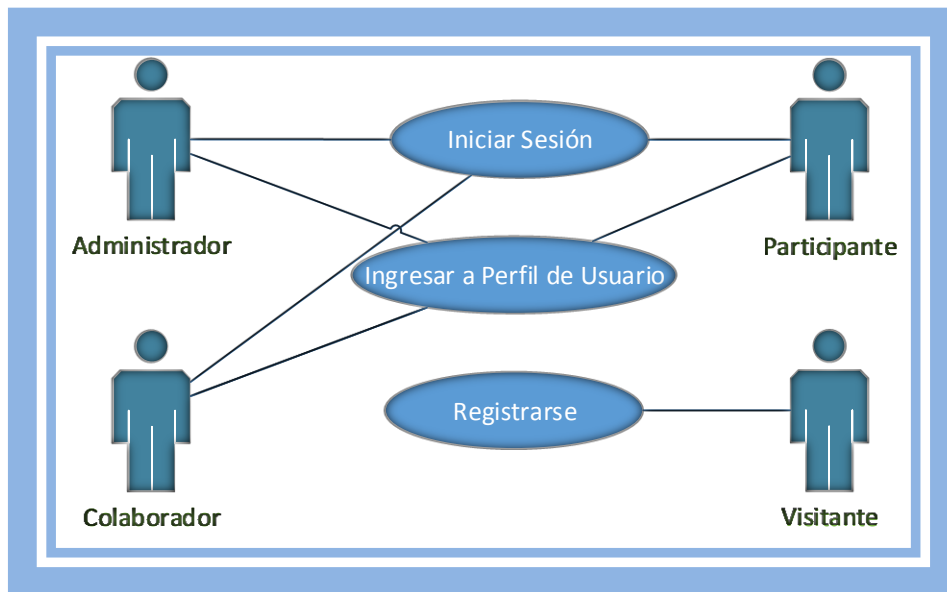


Figura V-11 Casos De Uso Usuarios

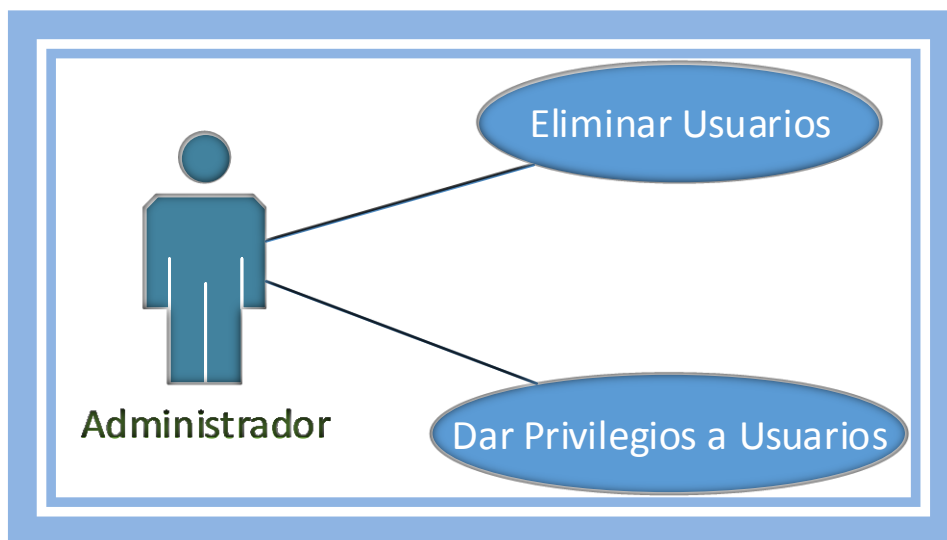


Figura V-12 Casos De Uso Usuarios (Continuación)

Después de observar el comportamiento de los actores con los casos de uso, ahora en las tablas V-1 a la V-32 se explica y detalla el flujo que debe seguir cada caso de uso.



<i>Nombre</i>	Registro De Usuario Nuevo
<i>Descripción</i>	El usuario podrá registrarse en el sistema de la página web para poder visualizar y acceder al contenido completo de la página web.
<i>Actores</i>	1. Visitante
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá contar con una dirección de correo electrónico válida.</li> <li>2. El usuario deberá encontrarse dentro de la página web.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en el botón iniciar sesión.</li> <li>2. El usuario dará clic en la opción registrarse.</li> <li>3. El usuario ingresará los datos solicitados.</li> <li>4. El usuario validará su cuenta desde la bandeja de entrada de su correo electrónico.</li> <li>5. Se registrará el usuario.</li> </ol>

**Tabla V-1 Registro De Usuario Nuevo**

<i>Nombre</i>	Ver Contenido De La Página Web.
<i>Descripción</i>	El usuario podrá visualizar el contenido principal de la página web.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Visitante</li> <li>2. Participante</li> <li>3. Colaborador.</li> <li>4. Administrador.</li> </ol>
<i>Precondiciones</i>	1. El usuario deberá tener un explorador de internet.
<i>Flujo</i>	1. El usuario escribirá en la barra de direcciones del explorador, la URL de la página.



2. Seleccionara la sección que desea visualizar.
3. La información de la página se visualizará.

**Tabla V-2 Ver Contenido De La Página Web**

<i>Nombre</i>	Iniciar Sesión De Usuario Registrado
<i>Descripción</i>	El usuario podrá iniciar sesión en el sistema de la página web, para poder visualizar y acceder al contenido completo de la página web.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Administrador.</li> <li>2. Colaborador.</li> <li>3. Participante.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá estar dentro de la página web.</li> <li>2. El usuario deberá haber ingresado al apartado iniciar sesión.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario ingresará los datos necesarios en los campos requeridos.             <ol style="list-style-type: none"> <li>a. Si el inicio de sesión es por medio de una red social, el usuario deberá seleccionar el icono de la red social que desea.</li> <li>b. El usuario deberá seguir las instrucciones que le aparecen en pantalla.</li> </ol> </li> <li>2. El usuario iniciara sesión.</li> <li>3. La información de la página se visualizará.</li> </ol>

**Tabla V-3 Iniciar Sesión De Usuario Registrado**

<i>Nombre</i>	Compartir Información En Redes Sociales.
<i>Descripción</i>	Permitir al usuario compartir información de la página en las redes sociales, Facebook, Twitter y Google+.

<i>Actores</i>	1. Usuario registrado.
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario estar dentro de la página web.</li> <li>2. El usuario deberá tener una sesión iniciada.</li> <li>3. El usuario deberá estar donde se encuentra la información que desea compartir.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic sobre el icono de la red social deseada (Facebook, Twitter o Google+)</li> <li>2. El usuario seguirá las instrucciones en pantalla.</li> <li>3. La información que el usuario desea se compartirá.</li> </ol>

**Tabla V-4 Compartir Información En Redes Sociales**

<i>Nombre</i>	Crear Blog
<i>Descripción</i>	El usuario podrá crear un blog personal.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Administrador</li> <li>2. Colaborador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario podrá tener privilegios de colaboradores.</li> <li>2. El usuario deberá haber iniciado sesión</li> <li>3. El usuario no deberá contar con un blog previamente.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario ingresa a su perfil.</li> <li>2. El usuario dará clic en el botón hacer mi blog.</li> <li>3. El usuario seguirá las instrucciones e ingresará los datos requeridos.</li> <li>4. El usuario dará clic en el botón enviar.</li> <li>5. El blog que el usuario desea se creará.</li> </ol>

**Tabla V-5 Crear Blog**

<i>Nombre</i>	Crear Una Publicación En El Blog
---------------	----------------------------------

<i>Descripción</i>	Permitirá al usuario crear una nueva publicación en su blog personal.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá tener privilegios de colaborador o administrador.</li> <li>2. El usuario deberá haber iniciado sesión</li> <li>3. El usuario deberá contar con un blog creado.</li> <li>4. El usuario deberá estar dentro de su perfil.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá dar clic en el botón ir al blog.</li> <li>2. El usuario deberá dar clic en la opción nueva publicación.</li> <li>3. El usuario ingresará la información que será mostrada en la publicación.</li> <li>4. El usuario dará clic en el botón enviar.</li> <li>5. La publicación que el usuario desea crear se creará y se publicará en el blog.</li> </ol>

**Tabla V-6 Crear Una Publicación En El Blog**

<i>Nombre</i>	Editar Publicación Del Blog.
<i>Descripción</i>	Se podrá editar una publicación que ya ha sido publicada en el blog del usuario.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario podrá tener privilegios de colaborador y administrador.</li> <li>2. El usuario deberá haber iniciado sesión</li> <li>3. El usuario deberá estar dentro de su blog.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá visualizar la publicación que desea editar.</li> <li>2. El usuario modificará la información que desee</li> <li>3. El usuario deberá dar clic al botón enviar.</li> <li>4. La publicación que el usuario desea se habrá editado.</li> </ol>

**Tabla V-7 Editar Publicación Del Blog**

<i>Nombre</i>	Eliminar Blog
<i>Descripción</i>	El usuario podrá eliminar su blog personal.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá tener un blog creado.</li> <li>2. El usuario deberá estar dentro de su perfil.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá dar clic en el botón borrar mi blog.</li> <li>2. El blog que el usuario desea eliminar se borrará.</li> </ol>

**Tabla V-8 Eliminar Blog**

<i>Nombre</i>	Borrar Comentarios De Una Publicación Del blog.
<i>Descripción</i>	Se le permitirá eliminar un comentario dentro de una publicación, solo comentarios que puedan considerarse ofensivo o fuera de lugar.
<i>Actores</i>	1. Administrador.
<i>Precondiciones</i>	1. El usuario deberá estar dentro de una entrada de algún blog.
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará el comentario que desea eliminar.</li> <li>2. El usuario deberá dar clic en el botón borrar comentario.</li> <li>3. El comentario que el usuario desea borrar será borrado.</li> </ol>

**Tabla V-9 Borrar Comentarios De Una Publicación Del Blog**

<i>Nombre</i>	Comentar Publicación En Un Blog.
<i>Descripción</i>	Se le permitirá al usuario comentar en una publicación dentro de un blog.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> <li>3. Usuario registrado.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá encontrarse dentro de la sección de blogs.</li> <li>2. El usuario deberá haber ingresado a la publicación que desea comentar.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá dar clic en el la opción comentar.</li> <li>2. El usuario escribirá su comentario y deberá dar</li> </ol>

- clic en el botón comentar.
3. El comentario que el usuario desea será publicado.

**Tabla V-10 Comentar Publicación En Un Blog**

<i>Nombre</i>	Crear Un Tema Principal De Foros.
<i>Descripción</i>	El usuario podrá crear un tema principal.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Administrador</li> <li>2. Colaborador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá contar con privilegios de colaborador o administrador.</li> <li>2. El usuario deberá haber iniciado sesión.</li> <li>3. El usuario deberá haber ingresado al módulo de foros.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá dar clic en el icono de configuración.</li> <li>2. El usuario dará clic en la opción nuevo tema.</li> <li>3. El usuario escribirá el nombre del tema y deberá dar clic en el botón enviar.</li> <li>4. El tema principal que el usuario desea se creará.</li> </ol>

**Tabla V-11 Crear Un Tema Principal De Foros**

<i>Nombre</i>	Crear un foro
<i>Descripción</i>	El usuario podrá crear un nuevo foro.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá contar con privilegios de colaborador o administrador.</li> <li>2. El usuario deberá haber iniciado sesión.</li> </ol>

<i>Flujo</i>	<ol style="list-style-type: none"> <li>3. El usuario deberá haber ingresado al módulo de foros.</li> </ol> <ol style="list-style-type: none"> <li>1. El usuario dará clic en el icono de configuración que se encuentre dentro del tema donde se creará el foro.</li> <li>2. El usuario seleccionará la opción de nuevo foro.</li> <li>3. El usuario escribirá los datos requeridos y seleccionará la opción foro final.</li> <li>4. El usuario dará clic en el botón enviar.</li> <li>5. El foro que el usuario desea se creará.</li> </ol>
--------------	--

**Tabla V-12 Crear Un Foro**

<i>Nombre</i>	<i>Borrar Tema Principal</i>
<i>Descripción</i>	El usuario podrá crear un nuevo foro.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá contar con privilegios de colaborador o administrador.</li> <li>2. El usuario deberá haber iniciado sesión.</li> <li>3. El usuario deberá haber ingresado al módulo de foros.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en el icono de configuración que se encuentre dentro del tema donde se creará el foro.</li> <li>2. El usuario seleccionará la opción borrar tema.</li> <li>3. El tema que el usuario desea eliminar se borrará.</li> </ol>

**Tabla V-13 Borrar Tema Principal**

<i>Nombre</i>	Editar tema principal
<i>Descripción</i>	El usuario podrá editar un tema principal
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá contar con privilegios de colaborador o administrador.</li> <li>2. El usuario deberá haber iniciado sesión.</li> <li>3. El usuario deberá haber ingresado al módulo de foros.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en el icono de configuración</li> <li>2. El usuario seleccionará la opción de editar tema.</li> <li>3. El usuario modificará las opciones que desea y dará clic en el botón enviar.</li> <li>4. El tema que el usuario desea se editará.</li> </ol>

**Tabla V-14 Editar Tema Principal**

<i>Nombre</i>	Borrar Foro
<i>Descripción</i>	El usuario podrá eliminar un foro existente.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber ingresado al módulo foros.</li> <li>2. El usuario deberá haber ingresado al foro que desea eliminar.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en la opción Acciones del Foro.</li> <li>2. El usuario seleccionara la opción Borrar Foro.</li> <li>3. El foro será borrado.</li> </ol>

**Tabla V-15 Borrar Foro**



<i>Nombre</i>	Editar Foro
<i>Descripción</i>	El usuario podrá editar un foro existente.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber ingresado al módulo foros.</li> <li>2. El usuario deberá haber ingresado al foro que desea Editar.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario da clic en la opción Acciones del Foro.</li> <li>2. El usuario seleccionara la opción Editar Foro.</li> <li>3. El usuario ingresará la información que desea editar.</li> <li>4. El usuario dará clic en el botón Enviar.</li> <li>5. El usuario será editado.</li> </ol>

Tabla V-16 Editar Foro

Tabla V-17 Editar Foro

<i>Nombre</i>	Crear Un Hilo / Tema De Discusión.
<i>Descripción</i>	El usuario podrá crear un hilo de discusión dentro de un foro.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Administrador</li> <li>2. Colaborador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber ingresado al módulo foros.</li> <li>2. El usuario deberá haber ingresado al foro donde desea crear el hilo de discusión.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en el icono de configuración.</li> <li>2. El usuario seleccionara la opción Nuevo Hilo.</li> <li>3. El usuario ingresará la información necesaria.</li> </ol>

4. El usuario dará clic en el botón Enviar.
5. El nuevo Hilo de discusión será creado.

**Tabla V-18 Crear Un Hilo / Tema De Discusión**

<i>Nombre</i>	Eliminar Comentario Dentro De Un Foro.
<i>Descripción</i>	El usuario podrá eliminar un comentario
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá estar dentro del tema donde se encuentra el comentario que desea eliminar</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará el comentario que desea eliminar.</li> <li>2. El usuario dará clic en el botón borrar comentario.</li> <li>3. El comentario que el usuario desea eliminar se borrará.</li> </ol>

**Tabla V-19 Eliminar Comentario Dentro De Un Foro**

<i>Nombre</i>	Contener un Hilo / Tema de discusión.
<i>Descripción</i>	El usuario podrá contener un hilo de discusión dentro de un foro.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Usuario Registrado</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber ingresado al hilo de discusión en el que se desea comentar.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>2. El usuario se irá a la parte de abajo del tema/hilo.</li> <li>3. El Usuario escribirá su comentario.</li> <li>4. El usuario dará clic en el botón enviar.</li> <li>5. El comentario del usuario será publicado.</li> </ol>

**Tabla V-20 Comentar Un Hilo / Tema De Discusión**

<i>Nombre</i>	Agregar Artículos
<i>Descripción</i>	El usuario podrá agregar artículos
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber iniciado sesión.</li> <li>2. El usuario deberá estar dentro del módulo de artículos.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en la opción nuevo artículo.</li> <li>2. El usuario ingresará la información necesaria.</li> <li>3. El usuario dará clic en el botón enviar.</li> <li>4. El artículo del usuario se creará.</li> </ol>

**Tabla V-21 Agregar Artículos**

<i>Nombre</i>	Borrar Artículo
<i>Descripción</i>	El usuario podrá borrar un artículo.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>3. El usuario deberá haber iniciado sesión.</li> <li>4. El usuario deberá haber entrado en el módulo de artículos.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará el artículo que desea eliminar.</li> <li>2. El usuario seleccionará la opción borrar artículo.</li> <li>3. El artículo que el usuario desea eliminar se borrará.</li> </ol>

**Tabla V-22 Borrar Artículo**

<i>Nombre</i>	Agregar Nombre De Empresa
<i>Descripción</i>	El usuario podrá agregar el nombre de la empresa
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber iniciado sesión.</li> <li>2. El usuario deberá haber entrado en el módulo de empresa.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario seleccionará la opción nueva empresa.</li> <li>2. El usuario ingresará la información necesaria.</li> <li>3. El usuario dará clic al botón enviar.</li> <li>4. La empresa que el usuario desea se creará.</li> </ol>

**Tabla V-23 Agregar Nombre De Empresa**

<i>Nombre</i>	Borrar Nombre De Empresa
<i>Descripción</i>	El usuario podrá borrar el nombre de la empresa
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber iniciado sesión.</li> <li>2. El usuario deberá haber entrado en el módulo de empresa.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará la empresa que se desea eliminar.</li> <li>2. El usuario dará clic en la opción borrar empresa.</li> <li>3. La empresa que el usuario desea eliminar se borrará.</li> </ol>

**Tabla V-24 Borrar Nombre De Empresa**

<i>Nombre</i>	Agregar Investigador
<i>Descripción</i>	El usuario podrá agregar el nombre de un investigador.
<i>Actores</i>	1. Administrador.
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá ingresa al panel de administración.</li> <li>2. El usuario deberá ingresar en el módulo USER del panel de administración.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará la cuenta de usuario que desea agregar como investigador.</li> <li>2. El usuario seleccionará la opción Edit.</li> <li>3. El usuario seleccionará la casilla de investigador.</li> <li>4. El usuario dará clic en el botón Update User.</li> <li>5. El nombre de investigador que el usuario desea se creará.</li> </ol>

**Tabla V-25 Agregar Investigador**

<i>Nombre</i>	Borrar Investigador
<i>Descripción</i>	El usuario podrá borrar el nombre de un investigador.
<i>Actores</i>	1. Administrador.
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá ingresa al panel de administración.</li> <li>2. El usuario deberá ingresar en el módulo USER del panel de administración de flujo.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará la cuenta de usuario que desea agregar como investigador.</li> <li>2. El usuario seleccionará la opción Edit.</li> <li>3. El usuario desmarcará la casilla de investigador.</li> <li>4. El usuario dará clic en el botón Update User.</li> <li>5. El nombre de investigador que el usuario desea se eliminará.</li> </ol>

**Tabla V-26 Borrar Investigador**

<i>Nombre</i>	Agregar Video
<i>Descripción</i>	El usuario podrá agregar video
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber iniciado sesión.</li> <li>2. El usuario deberá haber entrado en el módulo de videos.</li> <li>3. El usuario deberá contar con el link del video que desea insertar.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario dará clic en la opción nuevo video.</li> <li>2. El usuario insertará el link del video que se desea agregar.</li> <li>3. El usuario ingresará la información requerida.</li> <li>4. El usuario dará clic en el botón enviar.</li> <li>5. El video que el usuario desea se agregará.</li> </ol>

**Tabla V-27 Agregar Video**

<i>Nombre</i>	Borrar Video.
<i>Descripción</i>	El usuario podrá borrar un video previamente agregado
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber iniciado sesión.</li> <li>2. El usuario deberá haber entrado en el módulo de videos</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará el video que desea borrar.</li> <li>2. El usuario dará clic en el icono de configuración y seleccionará la opción borrar video.</li> <li>3. El video se borrara.</li> </ol>

**Tabla V-28 Borrar Video**

<i>Nombre</i>	Editar Video
<i>Descripción</i>	El usuario podrá editar el video.
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Colaborador.</li> <li>2. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber iniciado sesión.</li> <li>2. El usuario deberá haber entrado en el módulo de videos</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará el video que desea editar.</li> <li>2. El usuario dará clic en el icono de configuración y seleccionará la opción editar video.</li> <li>3. El usuario ingresará la información que requiere editar.</li> <li>4. El usuario dará clic en el botón enviar.</li> <li>5. El video que el usuario desea se editará.</li> </ol>

**Tabla V-29 Editar Video**

<i>Nombre</i>	Asignar privilegios a los usuarios.
<i>Descripción</i>	El usuario podrá asignar privilegios a los usuarios
<i>Actores</i>	<ol style="list-style-type: none"> <li>1. Administrador.</li> </ol>
<i>Precondiciones</i>	<ol style="list-style-type: none"> <li>1. El usuario deberá haber ingresado al panel de administrador.</li> </ol>
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará al usuario que le desea asignar privilegios.</li> <li>2. El usuario seleccionará la opción edit.</li> <li>3. El usuario dará clic en la lista desplegable correspondiente a Role.</li> <li>4. El usuario seleccionará el privilegio que le desea asignar.</li> <li>5. El usuario dará clic en el botón Update User.</li> <li>6. Los nuevos privilegios que el usuario seleccionó se asignarán.</li> </ol>

**Tabla V-30 Asignar Privilegios A Los Usuarios**

<i>Nombre</i>	Eliminar usuarios.
<i>Descripción</i>	El usuario podrá eliminar usuarios
<i>Actores</i>	1. Administrador.
<i>Precondiciones</i>	1. El usuario deberá haber ingresado al panel de administrador.
<i>Flujo</i>	<ol style="list-style-type: none"> <li>1. El usuario visualizará al usuario que le desea eliminar.</li> <li>2. El usuario dará clic en la opción Delete.</li> <li>3. El usuario dará clic en aceptar en la ventana emergente que alerte la acción.</li> <li>4. El usuario que el usuario desea eliminar se borrará.</li> </ol>

**Tabla V-31 Eliminar Usuarios**



### 5.1.3 Diagrama de clases.

En el lenguaje de programación Ruby, todo lo que se puede manipular es un objeto, también lo son los resultados de estas operaciones, la única forma que existe de manipular los objetos es a través de los métodos. Dentro de una clase, podemos definir uno o varios métodos. A continuación se describe la lista de clases empleadas en la página web:

- **Administrator:** Esta clase representa al administrador de la página.
- **Avatar:** Esta clase representa la imagen del avatar por usuario.
- **Upload:** Esta clase representa las imágenes que los colaboradores suben a la página.
- **User:** Esta clase representa a todos los usuarios. Los usuarios pueden ser colaboradores o participantes, los colaboradores podrían tener acceso a todas las herramientas de administración.
- **Forummer::Forum:** Esta clase representa los foros. Los foros pueden tener sub foros y temas a través de la clase ForumTheme. Pertenecen a un tema y pueden ser foros finales (en los cuales se crean hilos de discusión).
- **Forummer::Theme:** Esta clase representa los temas, los temas pueden tener sub temas y sub foros a través de la clase ForumTheme.
- **Forummer::ForumTheme:** Esta clase representa el intermediario que determina si los foros pertenecen a otros foros o a temas.
- **Forummer::Thread:** Hilos de discusión donde se llevan a cabo las pláticas por medio de comentarios.
- **Forummer::Comment:** Esta clase representa los comentarios pertenecientes a usuarios e hilos de discusión.
- **Blogger::Blog:** Esta clase representa los blogs. Existe solamente un blog por usuario.

- **Blogger::Post:** Posts Esta clase representa las pertenecientes a blogs. Dentro de los blogs reside la información que el usuario desea publicar en su blog.
- **Blogger::Comment:** Esta clase representa los comentarios que pertenecen a los usuarios y a los posts.
- **Workshop::Video:** Esta clase representa a los videos. Los videos pertenecen a los usuarios.
- **Article:** Esta clase representa los artículos publicados, con enlaces hacia páginas externas
- **Company:** Esta clase representa las empresas. Las empresas se almacenan y se muestran en forma de lista en la página.
- **Contact:** Esta clase representa al modelo utilizado para el funcionamiento del Mailer, el cual se encarga de realizar él envío de correos de contacto a los usuarios. Es necesario mantener un representante de los campos para llenar el formulario de contacto a pesar de que no se guarde ninguna información.
- **Research\_field:** Esta clase representa las listas de las áreas de investigación que se asignan a los usuarios con nivel de colaborador.

En las Figuras V-13 – Figura V-17, observaremos los diagramas de clases que existen en la página, así como sus dependencias con otras clases.

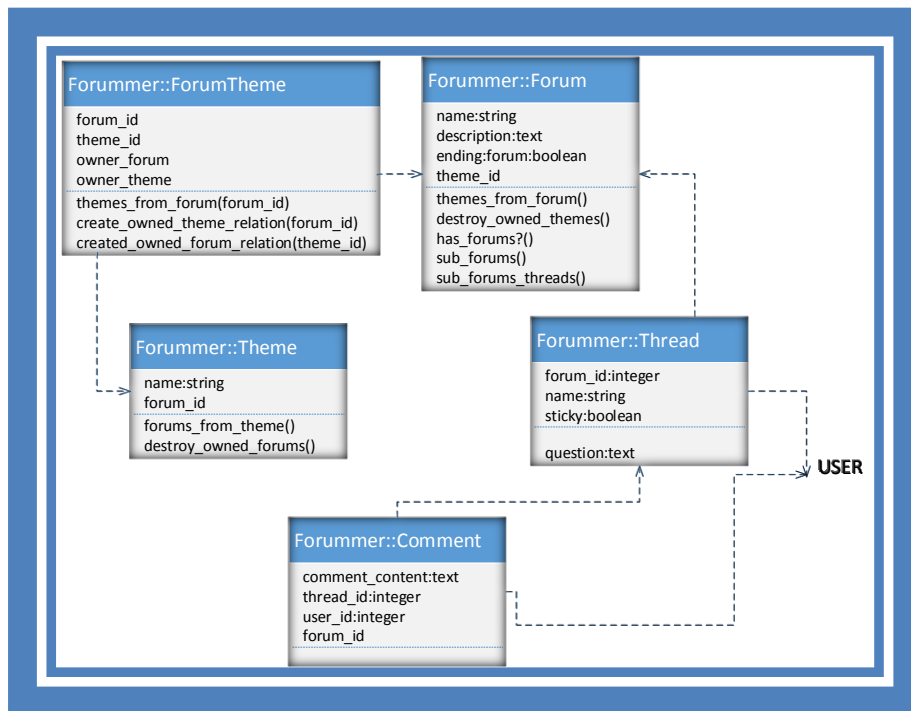


Figura V-13 Diagramas de Clases

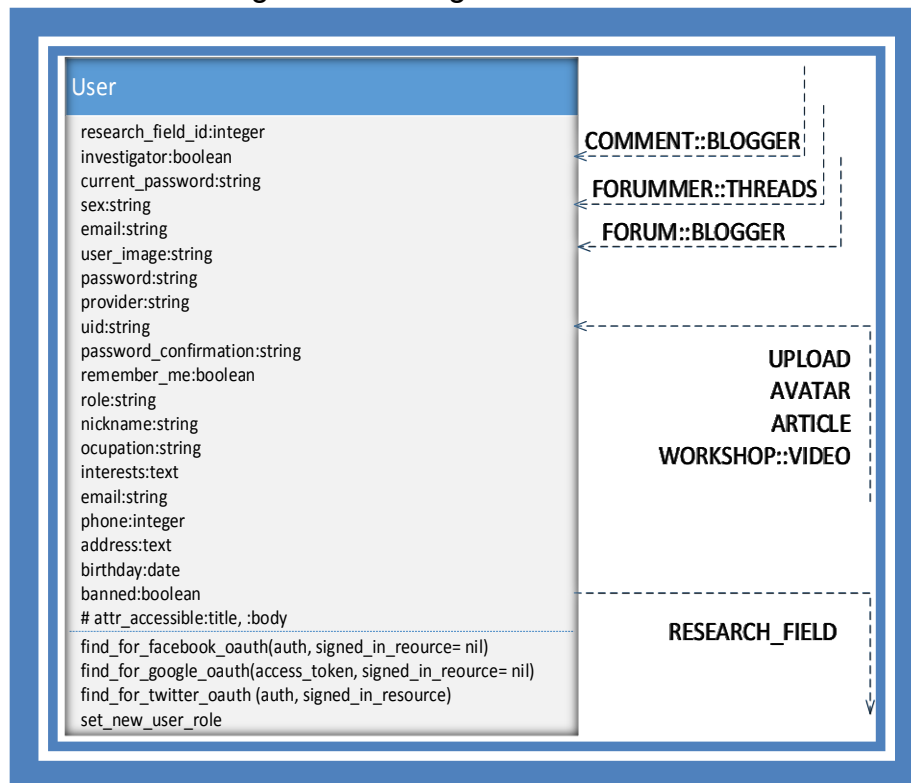


Figura V-14 Diagramas de Clases (Continuación)

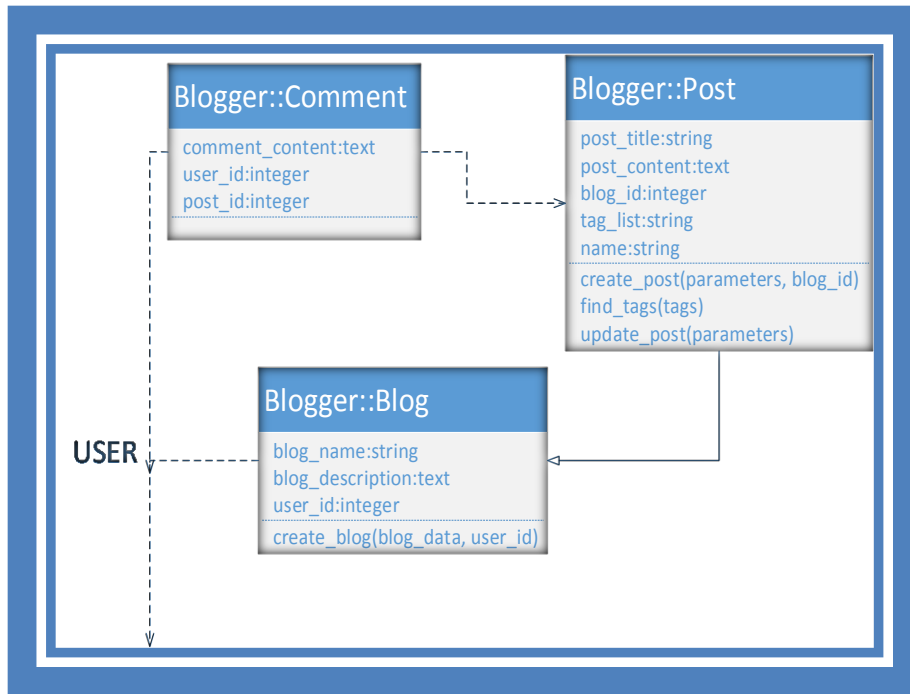


Figura V-15 Diagrama de Clases (Continuación)

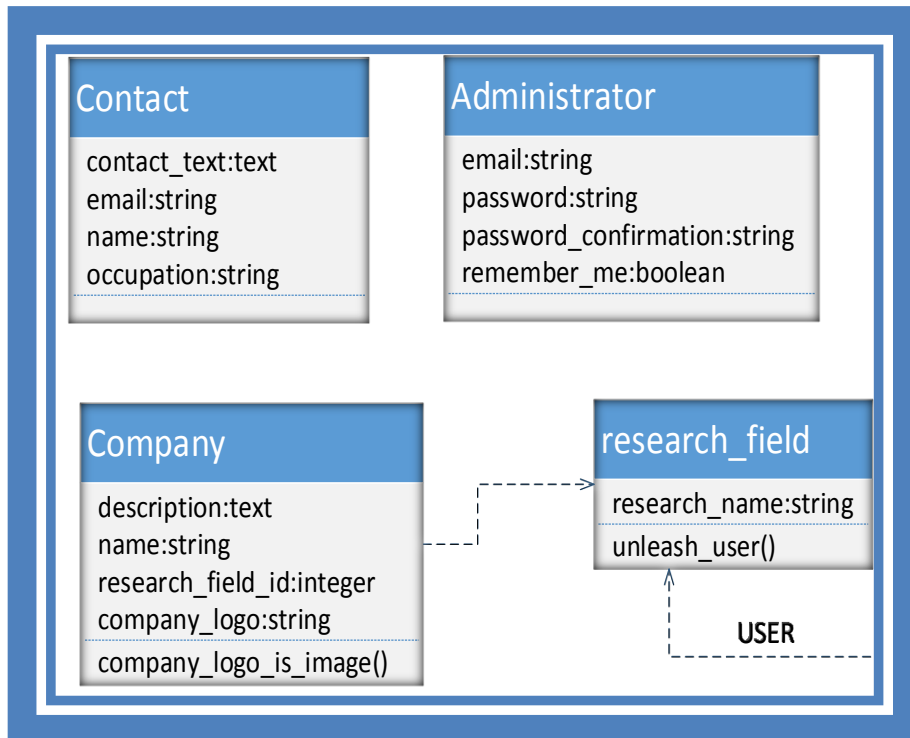


Figura V-16 Diagrama de Clases (Continuación)

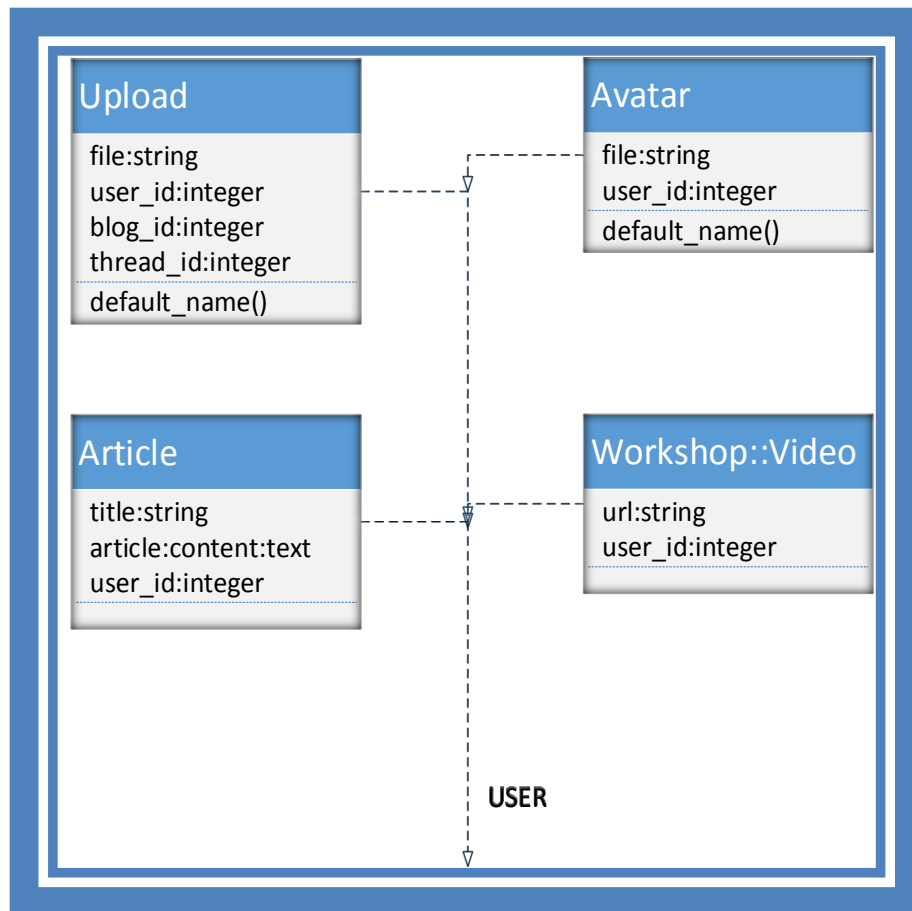


Figura V-17 Diagrama de Clases (Continuación)

## 5.2 Diseño

### 5.2.1 Lógica de Presentación.

Se implementa una interfaz de usuario basada en HTML, CSS y JavaScript, con ello se logra una presentación sencilla y ligera para el cliente, lo que se traduce en un menor tiempo de espera del usuario.

HTML se compone de dos partes:

- El contenido, es el texto que se visualizará en la pantalla del usuario.

- Las etiquetas y atributos, estas estructuran el texto en la página web con encabezados, párrafos, listas, enlaces, etc..., las etiquetas y los atributos no se muestran en pantalla, pero si el resultado de haberlos utilizado.

Para los estilos más definidos, se emplea CSS (Cascading Style Sheets), con el logramos tener una presentación más estética de los documentos HTML. Con CSS obtenemos mayor libertad en la elección de colores, fuentes, formato, etc...

Esta implementación gráfica, no interfiere ni condiciona el software del usuario, es una interfaz totalmente transparente para el cliente, el cual no necesita ningún requerimiento especial, ni software adicional para poder acceder a la aplicación.

Todos los datos son procesados del lado del servidor y entregados de manera pertinente a través de esta GUI o interfaz de usuario.

### **5.2.2 Lógica de la aplicación.**

Debido a la estructura de nuestra página web, la lógica de la aplicación reside en el servidor. El servidor es quien se encarga de las tareas más importante. El cliente únicamente tiene la responsabilidad de procesar la interfaz gráfica de usuario. Esto motiva a que los clientes no requieran de requisitos especiales para poder interactuar y visualizar el contenido.

La lógica de la aplicación mantiene una aproximación orientada a objetos. Con el ello el código queda organizado en objetos que contienen datos y el comportamiento de los mismos. Estos objetos trabajan generalmente mediante herencias o dependencias, lo cual les muestra el camino para resolver la lógica de la aplicación.

### **5.2.3 Lógica de datos.**

Debido al tamaño de la aplicación, la base de datos se encuentra dentro del mismo servidor que aloja la aplicación. Esto no es motivo para observar una disminución en el rendimiento de la aplicación, así como el aumento en el tráfico de la red no es lo suficientemente significativo para considerarlo una problemática.

La base de datos se gestiona a través de MySQL (My Structured Query Language), se eligió esta alternativa debido a que es una opción Open Source. También nos ofrece una velocidad considerable al realizar operaciones de consulta, convirtiéndola en una de las herramientas más utilizadas en el mundo del desarrollo de aplicaciones.

Otro de los puntos importantes para la elección de MySQL es el bajo costo en requerimientos para elaborar la base de datos, es tan bajo su consumo de recursos, que podemos usar una máquina con escasos recursos para realizar la gestión y no tendríamos mayor inconveniente. Es fácil de configurar e instalar, además de que trabaja con una amplia variedad de sistemas operativos.

## **5.3 Implementación**

### **5.3.1 Tecnología implementada**

#### ***5.3.1.1 Ruby On Rails***

Ruby On Rails es un framework que ha ganado mucho terreno en el campo del desarrollo de aplicaciones web debido a su sencillez y a la gran potencia que brinda.

Ruby On Rails es un entorno de desarrollo web que es muy fácil de aprender y que no cuesta mucho trabajo dominar, únicamente se necesita tener

conocimientos sobre Ruby. Con Ruby On Rails se pueden crear aplicaciones sumamente sencillas, pero también aplicaciones con una complejidad mayor.

Es por ello que se decidió utilizar esta tecnología para el desarrollo del proyecto, aprovechando que Ruby On Rails está enfocado a la producción, nos genera un considerable ahorro de tiempo en tareas que con otros lenguajes hubiéramos tenido que realizar manualmente. Por ejemplo, para el diseño de la base de datos, en el lenguaje PHP se tiene que hacer una declaración manual e individual de cada tabla y columna, con Ruby On Rails solo tenemos que usar la sintaxis que nos brinda para esa tarea y él se encarga del resto.

Otro punto importante en la elección de Ruby On Rails es que es software libre, lo cual significa que el código fuente lo podemos modificar para mejorarlo y afinarlo, o incluso extenderlo.

Ruby On Rails está compuesto por gemas. Una gema es un código añadido que permite a nuestro proyecto integrar nuevas funciones predefinidas, como por ejemplo el login de los usuarios. Con el uso de estas gemas, también generamos un ahorro en el tiempo de programación.

También, la filosofía DRY (Don't Repeat Yourself) -no te repitas- que Ruby On Rails implementa, provoca que las definiciones que haya que hacer, se hagan una sola vez, ya que todos los componentes están integrados y no hay necesidad de generar puentes para relacionarlos, logrando con ello ahorro en tiempo de programación. Un ejemplo de esto se da con ActiveRecord, en el que las definiciones de clases no tienen la necesidad de especificar los nombres de las columnas, Ruby los detecta automáticamente desde la base de datos.

El proyecto que se presenta en este trabajo monográfico se desarrolló con la versión 1.9.3 de Ruby y con la versión 2.3.2 de Ruby On Rails. Estas versiones son las más estables que existen hasta el momento.

A continuación se presenta la lista de gemas utilizadas en el proyecto.



```
gem 'taps'
gem 'rails', '3.2.9'
gem 'heroku'
gem 'therubyracer'
gem 'haml-rails'
gem 'holder_rails'
gem 'activeadmin', github: 'gregbell/active_admin'
gem 'meta_search'
gem 'twitter-bootstrap-rails'
gem 'omniauth'
gem 'omniauth-facebook'
gem 'omniauth-twitter'
gem 'omniauth-google-oauth2'
gem 'simple_form'
gem 'pry', :group => :development
gem 'jbuilder'
gem 'acts-as-taggable-on'
gem 'tinymce-rails'
gem 'carrierwave'
gem "fog"
gem 'social-share-button'
gem "rmagick"
gem "kaminari"
gem 'jquery-ui-rails'
gem "newrelic_rpm"
gem "ransack"
gem 'social-share-button'
gem 'youtube_it'
gem 'fastimage'
gem 'tldmail'
gem "recaptcha", :require => "recaptcha/rails"
group :development do
  gem 'mysql2'
end

group :test do
  gem 'sqlite3'
end

group :production do
  gem 'mysql2'
end

group :assets do
  gem 'sass-rails', '~> 3.2.3'
  gem 'coffee-rails'#, '~> 3.2.1'
  gem 'coffee-script-source', '~> 1.4.0'
```

```
gem 'less-rails'  
gem 'jquery-rails'  
gem 'uglifyer', '>= 1.0.3'  
gem 'jquery-fileupload-rails'  
end
```

### 5.3.1.2 HTML

HTML es un lenguaje que se emplea en el desarrollo de páginas web. Está compuesto por una serie de etiquetas las cuales el navegador se encarga de interpretarlas y dar forma en la pantalla del usuario.

HTML dispone de etiquetas para imágenes, hipervínculos, saltos de página, saltos de línea, tablas, etc.... Podemos decir que HTML nos sirve para crear páginas web, darle estructura y contenido.

En la página web que se desarrolló, se utiliza la versión HTML5, la cual nos brinda mejores características en el manejo de las imágenes. Por ejemplo, en versiones anteriores de HTML, los diseñadores insertaban la imagen y era complicado poder interactuar con ellas directamente desde el HTML, con la nueva versión, se pueden crear imágenes dinámicas.

Otro punto a favor de HTML5 con respecto a versiones anteriores es a la hora de insertar un video en la página. Con las versiones anteriores era necesario instalar un programa en la computadora que permita la ejecución del video desde la web. Con HTML5 solo se necesitara un navegador que soporte HTML5, con ello logramos una menor responsabilidad del usuario a la hora de visualizar el video.

También hay una mejora significativa al momento de descargar la página. HTML5 permite descargar elementos en segundo plano, es decir, le da prioridad a los más rápidos en cargar, mientras que los menos importantes se van descargando mientras el usuario ya interactúa con los más importantes.

Este punto es muy importante, hoy en día el desempeño de las páginas web, comienza desde su velocidad de carga, con esta tecnología logramos reducir el tiempo de carga, sin duda un aspecto positivo para los usuarios que gustan de ver la información lo más rápido posible.

### **5.3.1.3 CSS**

CSS es un lenguaje utilizado en la presentación de documentos HTML. En términos coloquiales, un documento HTML viene siendo una página web. Entonces, podemos decir que CSS sirve para organizar la presentación y el aspecto de una página web. Al igual que HTML, CSS es interpretado por el navegador web del usuario.

Para poder lograr el objetivo visual de la página, era necesario usar CSS, aunque no era la única solución, si era la que más nos ayudaría a reducir sintaxis y tiempo en la elaboración de las vistas.

Se aprovechó la filosofía de CSS, la cual separa la estructura del documento HTML de su presentación. CSS es un lenguaje sumamente intuitivo, lo cual nos dio la facilidad de realizar las vistas de forma sencilla.

### **5.3.1.4 JavaScript**

JavaScript es un lenguaje de programación que se utiliza para crear pequeños programas, los cuales se encargaran de realizar una función en específico dentro de la página. Este es un lenguaje que se ejecuta en el navegador. Gracias a que es compatible con la mayoría de los navegadores modernos, lo convierte en uno de los lenguajes de programación más usados.

Se elige el uso de JavaScript debido a que se buscaban lograr ciertos comportamientos en la página que sean dinámicos, es decir que el usuario interactúe con ellos. Para ello CoffeScript, la cual es una gema en una versión

más avanzada de JavaScript, se logra crear un sistema que se encargue de las imágenes que los usuarios ingresarían a la hora de crear una publicación en su blog o una publicación en algún foro y para las fotos que los usuarios establecerían como avatar.

### **5.3.2 Descripción**

La página web, esta seccionada en 3 bloques, estos bloques son:

- **Encabezado:** Aquí encontraremos el logo de nuestra página, la barra del buscador, el botón para iniciar sesión y el menú que da acceso a los módulos de la página.
- **Contenido:** Este bloque será donde la información de cada sección se muestre. Este bloque es dinámico, ya que dependiendo la sección a la que ingreses será la información que te muestre.
- **Footer (Pie de página):** En este bloque encontráramos la información de los derechos de autor de la página y los iconos para acceder a las redes sociales de la misma.

El encabezado y el footer se mostrarán siempre en todas las secciones de la página. La página web consta de 7 secciones, las cuales se establecen en el menú. A continuación, se explicarán cada una de las secciones.



Figura V-18 Pagina Inicio

### 5.3.2.1 Inicio

Esta es la sección principal de la página, aquí se mostrará la información más reciente que se ha ingresado al sitio en las secciones de blog y artículos. Las publicaciones en blog y artículos, se mostrarán en la parte de abajo del

contenido en tres sub-secciones llamadas Destacados. También existe un carrusel de información, el cual muestra un artículo diferente cada 4 segundos.

### **5.3.2.2 Artículos**

La sección de artículos está pensada para que los usuarios, principalmente los colaboradores, muestren información científica en forma de artículos.

### **5.3.2.3 Investigadores**

En esta sección encontramos el nombre de todos los investigadores que están registrados en la página web, esta información tiene la finalidad de brindarles a los usuarios la oportunidad de contactar a los investigadores y externarles sus dudas o comentarios a través de un correo electrónico.

### **5.3.2.4 Videos**

En esta sección se publicarán videos que traten temas relacionados con las energías renovables. Estos videos tienen la finalidad de mostrar de forma práctica las ventajas en la utilización de las energías renovables.

### **5.3.2.5 Foro / Blog**

#### **5.3.2.5.1 Foros**

En esta sección se encuentran los foros de discusión y permitirá a los usuarios acceder a ellos para externar dudas o comentarios con respecto a un tema en específico. La estructura de los foros se detalla a continuación:

Los foros, son un espacio dedicado a la publicación de discusiones u opiniones con el objetivo principal de que la comunidad enriquezca el tema, opinando y comentando. En el foro existen reglas, las cuales son necesarias para mantener un ambiente de respeto entre los usuarios.

Está un foro principal, este esta visible solo con acceder a la sección de foros en el sitio. Dentro de ese foro principal, se establecerán las secciones que organizarán los foros, a estas secciones se les llamará temas. Dentro de esos temas se crearán los foros, un foro creado dentro de un tema específico, deberá seguir la línea que el tema indica, es decir, no debemos crear dentro de un foro de energías sustentable un foro de tecnología. El encargado de administrar y organizar el contenido de los foros serán los usuarios con perfil de Colaborador.

Los foros creados dentro de los temas, se pueden convertir en foros finales, los foros finales son los que albergaran los hilos de discusión. Un hilo de discusión puede ser una pregunta formulada por un usuario, alguna información relevante que un colaborador quiera compartir y de la cual pueda surgir una interacción positiva, siempre dentro de un ambiente cordial.

Para poder crear temas, foros e hilos de discusión, deberás ser un usuario con privilegios de colaborador.

En la Figura V-19, se muestra un ejemplo jerárquico de la estructura de los foros, en las que se puede manejar dos opciones, en la primera crear una especie de sub-foros, los cuales tendrán temas y dentro cada tema un foro final que permita crear hilos de discusión. En la segunda opción, simplemente dentro de un tema, creamos el foro final y con eso ya podremos crear hilos de discusión.

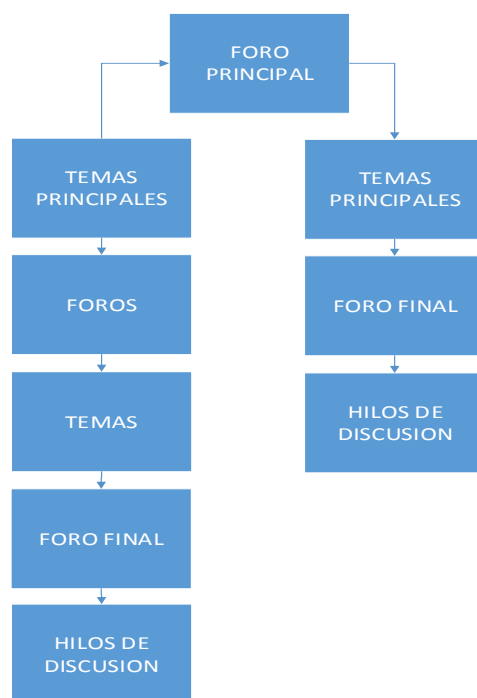


Figura V-19 Esquema Foros

### 5.3.2.5.2 Blogs

En esta sección, los usuarios con privilegios de colaborador y administrador, tendrán la oportunidad de crear un blog personal. Un blog es un sitio web que recopila cronológicamente textos o artículos de uno o varios autores, poniendo primero el más reciente. En cada artículo, es común que los lectores puedan escribir sus comentarios y el autor responderlos, logrando un dialogo, el cual debe llevarse a cabo de forma responsable y respetuosa.

Los usos o temáticas de los blogs, son muy diversos, se pueden usar desde algo personal hasta algo científico o educativo, incluso hay quienes definen un blog como un diario personal.

### 5.3.2.6 Empresas

En esta sección, los usuarios podrán conocer la información de las empresas relacionadas con el uso de energías renovables. La idea principal de esta



sección, es brindarles a los usuarios que estén interesados en algún producto o alguna solución en específico, la oportunidad de contactar directamente con las empresas y así conocer con más detalle los productos.

#### **5.3.2.7 Contacto**

En esta sección, los usuarios podrán contactar con el administrador de la página y externarle dudas y comentarios. Los mensajes que se envíen a través del formulario de contacto, los recibirá el administrador en la bandeja de entrada de su correo electrónico, desde ahí podrá responder todas tus dudas.

### **5.4 Evaluación del proyecto**

La evaluación del proyecto consistió en analizar en distintos navegadores web el comportamiento gráfico de la página. Para ello, se utilizaron los siguientes navegadores web:

- Safari versión 7.0
- Mozilla Firefox versión 24.0
- Google Chrome versión 31.0.1650.57

Los resultados nos muestran que existe una pequeña variación en cuanto a la tonalidad de colores de un navegador a otro, esto a causa de la interpretación del lenguaje CSS en cada navegador. En cuanto a la presentación de la información, en los tres navegadores se despliegan de forma correcta y no presentan ningún inconveniente. Los programas hechos en JavaScript funcionan correctamente.

En general se puede decir que la página web funciona correctamente en los 3 navegadores web usados para la evaluación.

En las Figura V-20, Figura V-21 y Figura V.22 se muestran las capturas de pantalla mostrando el acceso a la página principal en cada navegador.



Figura V-20 Acceso a página web Safari 7.0



Figura V-21 Acceso a página web Mozilla Firefox 24.0



Figura V-22 Acceso a página web Google Chrome 31.0.1650

## Capítulo VI Conclusiones

Durante los últimos 15 años, el crecimiento del uso de Internet se incrementó de forma exponencial, pasando a ser una herramienta indispensable en la mayoría de las empresas, a tal grado que una falla en el acceso a Internet, puede resultar en pérdidas millonarias.

Derivado de ese crecimiento, el mercado de las páginas web también ha tenido un crecimiento simultáneo. Ahora las empresas –desde grandes corporativos hasta empresas pequeñas– tienen un sitio web, mediante el cual ofrecen sus servicios, dan a conocer información relevante o simplemente interactúan con sus clientes. Es gracias al Internet y a las páginas web en conjunto, que el comercio se globalizó, es decir, desde cualquier parte del mundo puedes comprar a través de Internet a una empresa que se encuentre en cualquier país del mundo.

El Internet y las páginas web, también se pueden usar en otros ámbitos como el educativo, por ejemplo, existen muchas universidades ofreciendo programas académicos (licenciatura y posgrados) en línea, cosa que jamás se hubiera imaginado dos décadas atrás. Así como el ámbito educativo, el sector de investigación ha hecho uso del Internet, para la difusión científica y tecnológica.

Con el objetivo realizar la difusión científica y tecnológica referente a las energías renovables, nace este proyecto que busca aprovechar el uso de la plataforma del internet para dar a conocer las ventajas de implementar el uso de las energías renovables en nuestra vida cotidiana.

Para lograr el objetivo, se desarrolló un sistema basado en el lenguaje de programación Ruby y su framework para diseño web Ruby On Rails. Además se utilizaron estilos en CSS y código de JavaScript para algunas funciones específicas del sistema, todo esto bajo el esquema que HTML maneja.

La página desarrollada mantiene una arquitectura de dos niveles y un modelo de distribución de presentación distribuida, puesto que en el equipo del usuario únicamente reside la presentación y su lógica, mientras que la lógica de la aplicación, la lógica de datos y la gestión de datos se encuentran en un servidor.

El siguiente paso en este proyecto, es desarrollar una aplicación que publique automáticamente la información del sitio en los diferentes canales que existen en las redes sociales, para con esto tener una constante de información y presencia, lo cual dará como resultado una mayor afluencia al sitio y con ello una mayor difusión del contenido. De igual forma, como segunda etapa del presente proyecto se propone su migración a dispositivos móviles.

## Capítulo VII Bibliografía

- Ballard, D. a. (1982). *Computer vision*. Englewood Cliffs.: Prentice-Hall Inc.
- Campderrich Falgueras, B. (2001). *Ingenieria del Software*. Barcelona, España: Universitat Oberta de Catalunya.
- Coulouris, G., Dolilimore, J., Kindberg, T., & Blair, G. (2012). *Distributed Systems - Concepts and Desing* (5ta edición ed.). Pearson.
- Dolado, J., & Fernandez, L. (2000). *Medicion para la gestion en la ingenieria del software*. Ra-Ma.
- Elmasri, R., & Navath, S. B. (2007). *Fundamentos de Sistemas de Bases de Datos*. Madrid, España: Pearson Education S.A.
- Gundavaram, S. (1996). *CGI Programming on the World Widw Web* (Primera ed.). O'Reilly & Associates, Inc.
- Hoque, R., & Sharma, T. (1998). *Programming Web Components* (Primera ed.). McGraw-Hill.
- Mora, S. L. (2001). *Programacion en Internet: Clientes Web*. Alicante, España: Editorial Club Universitario.
- Tanenbaum, A. S., & Van Steen, M. (2006). *Distributed Systems - Principles and Paradigms* (2ª edición ed.). Amsterdam, Holanda: Pearson.
- Wall, D. (2004). *Multi - Tier Application Programming with PHP*. Morgan Kaufmann.

## Capítulo VIII Glosario

- Applets** Un Applet es un programa escrito en Java que forma parte de los componentes de una página web. Han sido usados para satisfacer las necesidades que no se pueden lograr solo con HTML
- Genome** Es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Linux y derivados de UNIX
- Hardware** Se refiere a cualquier componente físico que trabaja o interactúa de alguna forma con una computadora.
- Mac OS X** Es un Sistema operativo basado un UNIX, el cual es desarrollado, comercializado y vendido por Apple Inc.
- Main Frame** Es una computadora central, generalmente se le denomina mainframe al equipo más costoso, potente y grande con el que cuenta una empresa.
- Open Source** Es una expresión con la que se conoce al software distribuido y desarrollado libremente.
- Plug-ins** Es el complemento de una aplicación, el cual se relaciona con otra para aportar una función específica.
- Scripts** Es un conjunto de instrucciones que normalmente se almacenan en un archivo de texto y deben ser ejecutadas línea por línea en tiempo real.
- Software** Es la parte lógica de un sistema informático. Comprende un conjunto de componentes lógicos necesarios que hacen posible la realización de tareas prácticas.
- Windows** Es un sistema operativo con una interfaz gráfica, diseñado para computadoras personales.

## Capítulo IX Abreviaturas

<b>APIs</b>	Aplication Programming Interface
<b>ASP</b>	Application Service Provider
<b>B2B</b>	Business to Business
<b>CGI</b>	Common Gateway Interface
<b>CORBA</b>	Common Object Request Broker Architecture
<b>CSS</b>	Cascading Style Sheets
<b>DCOM</b>	Distributed Component Object Model
<b>DLL</b>	Dynamic Link Library
<b>FTP</b>	File Transfer Protocol
<b>GUI</b>	Graphic User Interface
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hyper Text Transfer Protocol
<b>IETF</b>	Internet Engineering Task Force
<b>IIS</b>	Internet Information Server
<b>IMAP</b>	Internet Message Access Protocol.
<b>ISAPI</b>	Internet Server Aplication Program Interface
<b>ISP</b>	Internet Service Provider
<b>JDBC</b>	Java DataBase Connectivity
<b>JSP</b>	Java Server Pages
<b>MySQL</b>	My Structured Query Language
<b>NNTP</b>	Network News Transfer Protocol
<b>OBCD</b>	Open DataBase Connectivity
<b>OOUI</b>	Object Oriented User Interface
<b>PHP</b>	Personal Home Page
<b>POP</b>	Post Office Protocol
<b>RMI</b>	Remote Method Invocation.
<b>RoR</b>	Ruby On Rails



<b>RPC</b>	Remote Procedure Call
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>SQL</b>	Structured Query Language
<b>SSL</b>	Secure Sockets Layer
<b>TCP/IP</b>	Transfer Control Protocol / Internet Protocol
<b>URL</b>	Universal Resource Locator
<b>WWW</b>	World Wide Web

## Capítulo X Anexos

### Anexo A Código Controladores

#### Blog\_controller

```
# encoding: utf-8
class Blogger::BlogsController < ApplicationController
  def index
    @posts = Blogger::Post.page(params[:page]).per(3).order("updated_at DESC")
  end
  def edit
    @blog = Blogger::Blog.find(params[:id])
  end
  def update
    @blog = Blogger::Blog.find(params[:id])
    if @blog.user == current_user
      @blog.update_attributes(params[:blogger_blog])
      flash[:notice] = "¡Actualizado con éxito!"
      redirect_to @blog
    else
      flash[:alert] = "No tienes permisos para editar este blog"
      render :edit
    end
  end
  def destroy
    @blog = Blogger::Blog.find(params[:id])
    if @blog.user == current_user
      @blog.destroy
      flash[:notice] = "¡Borrado con éxito!"
      redirect_to users_profile_path(@blog.user)
    else
      flash[:alert] = "No tiene derechos para borrar el blog"
      redirect_to users_profile_path(@blog.user)
    end
  end
  def new
    @blog = current_user.blog
    if @blog
      redirect_to blogger_blog_path(@blog)
    elsif current_user.role != "Collaborator"

```

```
        redirect_to blogger_blogs_path
      else
        @blog = Blogger::Blog.new
      end
    end
  end
  def create
    @blog = Blogger::Blog.create_blog(params[:blogger_blog], current_user.id)
    if @blog.errors.count > 0
      render :new
    else
      flash[:notice] = "¡Blog creado!"
      redirect_to @blog
    end
  end
  def show
    @blog = Blogger::Blog.find(params[:id])
    @posts = @blog.posts.page(params[:page]).per(3).order("updated_at DESC")
  end
end
```

## Comments\_controller

```
class Blogger::CommentsController < ApplicationController
  def create
    Blogger::Comment.create(params[:blogger_comment])
    @post = Blogger::Post.find(params[:blogger_comment][:post_id])
    @comment = Blogger::Comment.new
    redirect_to blogger_blog_post_path(@post.blog, @post)
  end
  def destroy
    comment = Blogger::Comment.find(params[:id])
    @post = comment.post
    comment.destroy
    @comment = Blogger::Comment.new
    flash[:notice] = "Comentario borrado"
    redirect_to blogger_blog_post_path(@post.blog, @post)
  end
end
```

## Post\_controller

```

# encoding: utf-8
class Blogger::PostsController < ApplicationController
  before_filter :current_blog, :only => [:show, :new]

  def index
    if params[:blog]
      @blog = Blogger::Blog.find(params[:blog])
      @posts = @blog.posts.tagged_with(params[:tag]).order("updated_at
DESC").page(params[:page]).per(3)
      @tag = params[:tag]
    else
      @posts = Blogger::Post.tagged_with(params[:tag]).order("updated_at
DESC").page(params[:page]).per(3)
      @tag = params[:tag]
    end
  end
  def new
    @post = Blogger::Post.new
    @upload = Upload.new
  end
  def create
    @post = Blogger::Post.create_post(params[:blogger_post], params[:blog_id])
    if @post.errors.count > 0
      @upload = Upload.new
      render :new
    else
      flash[:notice] = "Publicación hecha con éxito"
      redirect_to blogger_blog_path(@post.blog)
    end
  end
  def show
    @post = Blogger::Post.find(params[:id])
    @comment = Blogger::Comment.new
  end
  def edit
    @post = Blogger::Post.find(params[:id])
    @upload = Upload.new
  end
  def update
    Blogger::Post.find(params[:id]).update_post(params[:blogger_post])
    redirect_to blogger_blog_post_path(params[:blog_id], params[:id])
  end

  def tags

```

```
    respond_to do |format|
      format.html
      format.json {render json: empty_query?}
    end
  end

  def destroy
    Blogger::Post.destroy(params[:id])
    flash[:notice] = "Publicacion borrada"
    redirect_to Blogger::Blog.find(params[:blog_id])
  end

  private

  def current_blog
    @blog = Blogger::Blog.find(params[:blog_id])
  end

  def empty_query?
    tags = ActsAsTaggableOn::Tag.where("name LIKE ?", "%#{params[:q]}%")
    if tags.empty? || tags.blank?
      [{id: "<<<#{params[:q]}>>>", name: "Nueva: #{params[:q]}"}]
    else
      tags
    end
  end
end
```

## Comments\_controller

```
class Forummer::CommentsController < ApplicationController
  def create
    @new_comment = Forummer::Comment.new(thread_id:
params[:forummer_comment][:thread_id].to_i,
                                         user_id: current_user.id,
                                         comment_content:
params[:forummer_comment][:comment_content])
    if @new_comment.save
      redirect_to forummer_forum_thread_path(@new_comment.thread.forum,
@new_comment.thread)
    else
      @thread = Forummer::Thread.find(params[:forummer_comment][:thread_id])
      @upload = Upload.new
      redirect_to forummer_forum_thread_path(@thread.forum, @thread)
    end
  end
  def destroy
    @comment = Forummer::Comment.find(params[:id])
    if @comment.destroy
      redirect_to
forummer_forum_thread_path(@comment.thread.forum,@comment.thread)
    else
      redirect_to forummer_forum_thread_path(@comment.thread.forum,@comment.thread)
    end
  end
  def edit
    @comment = Forummer::Comment.find(params[:id])
    @upload = Upload.new
  end
  def update
    @comment = Forummer::Comment.find(params[:id])
    if @comment.update_attributes(params[:forummer_comment])
      redirect_to forummer_forum_thread_path(@comment.thread.forum, @comment.thread)
    else
      @upload = Upload.new
      render :edit
    end
  end
end
```

## Forum\_themes\_controller

```
class Forummer::ForumThemesController < ApplicationController
  def index
    if params[:main_forum]
      @forum = Forummer::Forum.find(params[:main_forum])
      @themes = Forummer::ForumTheme.themes_from_forum(params[:main_forum])
      @sticky_threads = @forum.threads.where(sticky: true)
      @threads = @forum.threads.where(sticky: false)
    else
      @forum = Forummer::Forum.find(params[:forum])
      @themes = Forummer::ForumTheme.themes_from_forum(params[:forum])
      @sticky_threads = @forum.threads.where(sticky: true)
      @threads = @forum.threads.where(sticky: false)
    end
  end
end
```



## forums\_controller

```

class Forummer::ForumsController < ApplicationController
  def main_forum
    @main_forum = Forummer::Forum.find_by_name("Foro Principal")
    redirect_to forummer_forum_themes_path(:main_forum => @main_forum)
  end
  def new
    @theme = params[:theme]
    @new_forum = Forummer::Forum.new
  end
  def create
    @new_forum = Forummer::Forum.new(name: params[:forummer_forum][:name],
ending_forum:params[:forummer_forum][:ending_forum], description:
params[:forummer_forum][:description] )
    if @new_forum.save
      Forummer::ForumTheme.create_owned_forum_relation(params[:forummer_forum][:theme_
id])
      redirect_to forummer_forum_themes_path(forum:
Forummer::Forum.order("created_at").last.id)
    else
      @theme = params[:forummer_forum][:theme_id]
      render :new
    end
  end
  def destroy
    Forummer::Forum.find(params[:id]).destroy
    Forummer::ForumTheme.where(owner_theme: true, forum_id:params[:id]).first.destroy
    redirect_to forummer_main_forum_path
  end
  def edit
    @forum = Forummer::Forum.find(params[:id])
  end
  def update
    @forum = Forummer::Forum.find(params[:id])
    if @forum.update_attributes(params[:forummer_forum])
      redirect_to forummer_forum_themes_path(forum: @forum.id )
    else
      render :edit
    end
  end
  def faq
  end
end

```

## themes\_controller

```
class Forummer::ThemesController < ApplicationController
  def new
    @forum_id = params[:forum]
    @new_theme = Forummer::Theme.new
  end
  def create
    @new_theme = Forummer::Theme.new(name: params[:forummer_theme][:name])
    if @new_theme.save
      Forummer::ForumTheme.create_owned_theme_relation(params[:forummer_theme][:forum_
id])
      redirect_to forummer_forum_themes_path(forum:
params[:forummer_theme][:forum_id])
    else
      @forum_id = params[:forummer_theme][:forum_id]
      render :new
    end
  end
  def destroy
    Forummer::Theme.find(params[:id]).destroy
    Forummer::ForumTheme.where(owner_forum: true, theme_id:params[:id]).first.destroy
    redirect_to forummer_forum_themes_path(forum: params[:forum_id])
  end
  def edit
    @theme = Forummer::Theme.find(params[:id])
    @forum_id = params[:forum_id]
  end
  def update
    @theme = Forummer::Theme.find(params[:id])
    if @theme.update_attributes(name: params[:forummer_theme][:name])
      redirect_to forummer_forum_themes_path(forum:
params[:forummer_theme][:forum_id])
    else
      @forum_id = params[:forummer_theme][:forum_id]
      render :edit
    end
  end
end
```

## thread\_controller

```
class Forummer::ThreadsController < ApplicationController
  def new
    @forum = Forummer::Forum.find(params[:forum])
    @new_thread = Forummer::Thread.new
    @upload = Upload.new
  end

  def create
    @new_thread = Forummer::Thread.new(params[:forummer_thread])
    @new_thread.user_id = current_user.id
    if @new_thread.save
      redirect_to forummer_forum_thread_path(params[:forummer_thread][:forum_id],
      Forummer::Thread.order("created_at").last.id)
    else
      @forum = Forummer::Forum.find(params[:forummer_thread][:forum_id])
      @upload = Upload.new
      render :new
    end
  end

  def show
    @new_comment = Forummer::Comment.new
    @thread = Forummer::Thread.find(params[:id])
    @forummer_comments = @thread.forummer_comments.page(params[:page]).per(10)
    @upload = Upload.new
  end

  def destroy
    @thread = Forummer::Thread.find(params[:id])
    if @thread.destroy
      redirect_to forummer_forum_themes_path(forum: @thread.forum.id)
    else
      redirect_to forummer_forum_thread_path(@thread.forum, @thread)
    end
  end

  def edit
    @thread = Forummer::Thread.find(params[:id])
    @upload = Upload.new
  end

  def update
    @thread = Forummer::Thread.find(params[:id])
    if @thread.update_attributes(params[:forummer_thread])
      redirect_to forummer_forum_thread_path(@thread.forum, @thread)
    else

```

```
@upload = Upload.new
  render :edit
end
end
end
```

## Omniauth\_callback\_controller

```
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksController

  def facebook
    @user = User.find_for_facebook_oauth(request.env["omniauth.auth"], current_user)
    if @user.persisted?
      sign_in_and_redirect @user, :event => :authentication
    else
      session["devise.facebook_data"] = request.env["omniauth.auth"]
      redirect_to new_user_registration_url
    end
  end

  def twitter
    request.env["omniauth.auth"].delete(:extra)
    @user = User.find_for_twitter_oauth(request.env["omniauth.auth"],
current_user)
    if @user.persisted?
      sign_in_and_redirect @user, :event => :authentication
    else
      session["devise.twitter_data"] = request.env["omniauth.auth"]
      redirect_to new_user_registration_url
    end
  end

  def google_oauth2
    # You need to implement the method below in your model (e.g. app/models/user.rb)
    @user = User.find_for_google_oauth2(request.env["omniauth.auth"], current_user)

    if @user.persisted?
      flash[:notice] = I18n.t "devise.omniauth_callbacks.success", :kind => "Google"
      sign_in_and_redirect @user, :event => :authentication
    else
      session["devise.google_data"] = request.env["omniauth.auth"]
      redirect_to new_user_registration_url
    end
  end
end
```

## Registrations\_controller

```
class Users::RegistrationsController < Devise::RegistrationsController
  before_filter :check_captcha, only: :create
  def create
    super
  end

  private

  def check_captcha
    if verify_recaptcha == false
      flash[:alert] = "Captcha incorrecto"
      redirect_to new_user_registration_path
    end
  end
end
```

## Videos\_controller

```
# encoding: utf-8
class Workshop::VideosController < ApplicationController
  def show
    @video = Workshop::Video.find(params[:id])
    @client = YouTubeIt::Client.new
  end

  def destroy
    @video = Workshop::Video.find(params[:id])
    if @video.destroy && current_user == @video.user
      flash[:notice] = "Borrado con éxito"
      redirect_to workshop_videos_path
    else
      flash[:alert] = "No tienes permisos para borrar este video"
      redirect_to workshop_videos_path
    end
  end

  def index
    @videos = Workshop::Video.page(params[:page]).per(6).order("updated_at DESC")
    @client = YouTubeIt::Client.new
  end

  def new
    if current_user.role == User::COLLABORATOR
      @video = Workshop::Video.new
    else
      flash[:alert] = "No tienes permisos para acceder a esta sección de la página"
      redirect_to workshop_videos_path
    end
  end

  def edit
    @video = Workshop::Video.find(params[:id])
    if current_user.role != User::COLLABORATOR && @video.user != current_user
      flash[:alert] = "No tienes permisos para acceder a esta sección de la página"
      redirect_to workshop_videos_path
    end
  end

  def update
    @video = Workshop::Video.find(params[:id])
    @video.url = params[:workshop_video][:url]
    client = YouTubeIt::Client.new
  end
end
```

```
save_youtube_thumbnails(client.video_by(@video.url).thumbnails)
save_unique_id(client.video_by(@video.url).unique_id)
if @video.save && @video.user == current_user
  flash[:notice] = "Actualizado con éxito"
  redirect_to workshop_video_path(@video)
else
  flash[:alert] = "No tiene permisos para actualizar este video"
  render :edit
end
end

def create
  unless current_user.role != User::COLLABORATOR
    begin
      @video = Workshop::Video.new(params[:workshop_video])
      client = YouTubeIt::Client.new
      thumbnails = client.video_by(@video.url).thumbnails
      save_youtube_thumbnails(client.video_by(@video.url).thumbnails)
      save_unique_id(client.video_by(@video.url).unique_id)
      @video.user_id = current_user.id
      if @video.save
        redirect_to workshop_videos_path
      else
        render :new
      end
    rescue OpenURI::HTTPError
      flash[:alert] = "No es un video de youtube, favor de revisar el enlace"
      render :new
    end
  else
    flash[:alert] = "No tienes permisos para acceder a esta sección de la página"
    redirect_to workshop_videos_path
  end
end

private
def save_youtube_thumbnails(thumbnails)
  @video.thumbnail_1 = thumbnails[0].url
  @video.thumbnail_2 = thumbnails[1].url
  @video.thumbnail_3 = thumbnails[2].url
end
def save_unique_id(unique_id)
  @video.unique_id = unique_id
end
end
```



## aplication\_controller

```
class ApplicationController < ActionController::Base
  protect_from_forgery
  before_filter :get_avatar
  before_filter :banned?

  def get_avatar
    if user_signed_in?
      @avatar = Avatar.find_by_user_id(current_user.id)
      if @avatar.nil?
        @avatar = Avatar.new
      end
    end
  end

  def banned?
    if current_user.present? && current_user.banned?
      sign_out current_user
      flash[:error] = "Esta cuenta fue suspendida"
      new_user_session_path
    end
  end
end
```

## articles\_controller

```
# encoding: utf-8
class ArticlesController < ApplicationController
  def index
    @articles = Article.page(params[:page]).per(6).order("updated_at DESC")
  end
  def new
    if user_signed_in? && current_user.role == User::COLLABORATOR
      @article = Article.new
      @upload = Upload.new
    else
      flash[:alert] = "No tienes permisos para acceder a esta sección"
      redirect_to articles_path
    end
  end
  def create
    @article = Article.new(params[:article])
    if @article.save
      flash[:notice] = "Artículo Creado"
      redirect_to article_path(@article)
    else
      render :new
    end
  end
  def show
    @article = Article.find(params[:id])
  end
  def destroy
    @article = Article.find(params[:id])
    if user_signed_in? && current_user == @article.user
      flash[:notice] = "Artículo borrado"
      @article.destroy
      redirect_to articles_path
    else
      flash[:alert] = "No tienes permisos para borrar este artículo"
      redirect_to articles_path
    end
  end
  def edit
    @article = Article.find(params[:id])
    @upload = Upload.new
    unless user_signed_in? && current_user == @article.user
      flash[:alert] = "No tienes permisos para editar este artículo"
      redirect_to articles_path
    end
  end
end
```

```
end
def update
  @article = Article.find(params[:id])
  if user_signed_in? && current_user == @article.user
    if @article.update_attributes(params[:article])
      flash[:notice] = "Artículo actualizado"
      redirect_to articles_path
    else
      @upload = Upload.new
      render :edit
    end
  else
    flash[:alert] = "No tienes permisos para editar este artículo"
    redirect_to articles_path
  end
end
end
end
```

## avatars\_controller

```
class AvatarsController < ApplicationController
  def create
    @avatar = Avatar.find_by_user_id(params[:avatar][:user_id])
    if @avatar
      @avatar.update_attributes(params[:avatar])
    else
      @avatar = Avatar.create!(params[:avatar])
    end
    respond_to do |format|
      format.js
    end
  end
end
```

## companies\_controller

```
# encoding: utf-8
class CompaniesController < ApplicationController

  def index
    @companies = Company.page(params[:page]).per(6).order("updated_at DESC")
  end

  def new
    if user_signed_in? && current_user.role == User::COLLABORATOR &&
!current_user.research_field.nil?
      @company = Company.new
      @upload = Upload.new
    else
      if user_signed_in? && current_user.role == User::COLLABORATOR &&
!current_user.research_field.nil?
        flash[:alert] = "No tienes permisos, verifica tu campo de investigación con el
administrador"
        redirect_to companies_path
      else
        flash[:alert] = "No tienes permisos para acceder a esta sección"
        redirect_to companies_path
      end
    end
  end

  def create
    @company = Company.new(params[:company])
    if @company.save
      flash[:notice] = "Compañía Creada"
      redirect_to company_path(@company)
    else
      @upload = Upload.new
      render :new
    end
  end

  def show
    @company = Company.find(params[:id])
  end

  def edit
    @company = Company.find(params[:id])
    @upload = Upload.new
  end
end
```

```
unless user_signed_in? && current_user.role == User::COLLABORATOR &&
@company.research_field == current_user.research_field
  flash[:alert] = "No tienes permisos para editar esta empresa"
  redirect_to companies_path
end
end

def update
  @company = Company.find(params[:id])
  @upload = Upload.new
  if user_signed_in? && current_user.role == User::COLLABORATOR &&
@company.research_field == current_user.research_field
    if @company.update_attributes(params[:company])
      flash[:notice] = "Empresa actualizada"
      redirect_to company_path(@company)
    else
      render :edit
    end
  else
    flash[:alert] = "No tienes permisos para cambiar la empresa"
    redirect_to companies_path
  end
end

def destroy
  @company = Company.find(params[:id])
  if user_signed_in? && current_user.role == User::COLLABORATOR &&
@company.research_field == current_user.research_field
    if @company.destroy
      flash[:notice] = "¡Empresa borrada!"
      redirect_to companies_path
    else
      flash[:alert] = "No se pudo borrar la empresa"
      redirect_to companies_path
    end
  else
    flash[:alert] = "No tienes permisos para borrar la empresa"
    redirect_to companies_path
  end
end
end
```

## contacts\_controller

```
class ContactsController < ApplicationController
  def index
    @contact = Contact.new
  end
  def create
    ContactMailer.contact_mailing(Contact.new(params[:contact])).deliver
    redirect_to contacts_path
  end
end
```

## investigators\_controller

```
class InvestigatorsController < ApplicationController
  def index
    @investigators =
    User.investigators.page(params[:page]).per(6).order("updated_at DESC")
  end
end
```

## searches\_controller

```
# encoding: utf-8
class SearchesController < ApplicationController
  def search_home
    unless params[:q].blank?
      @forums = Forummer::Forum.search(name_or_description_cont: params[:q]).result
      @threads =
Forummer::Thread.search(name_or_question_or_forummer_comments_comment_content_cont:
params[:q]).result
      @blogs = Blogger::Blog.search(blog_name_or_blog_description_cont:
params[:q]).result
      @posts = Blogger::Post.search(post_content_or_post_title_cont:
params[:q]).result

      if user_signed_in? && current_user.role == User::COLLABORATOR
        @users = User.search(email_or_name_or_nickname_cont: params[:q]).result
      else
        @users = remove_non_collaborators(User.search(email_or_name_or_nickname_cont:
params[:q]).result)
      end
    else
      flash[:alert] = "La búsqueda no puede estar vacia"
      redirect_to blogger_blogs_path
    end
  end

  def remove_non_collaborators(users)
    new_array = []
    users.each do |user|
      if user.role == User::COLLABORATOR
        new_array << user
      end
    end
    return new_array
  end
end
```



## statics\_pages\_controller

```
class StaticPagesController < ApplicationController
  def home
    articles = Article.last(3).reverse
    videos = Workshop::Video.last(3).reverse
    threads = Forummer::Thread.last(3).reverse
    posts = Blogger::Post.last(3).reverse
    @objects = mixer(articles, videos, threads, posts)
  end

private

  def mixer( articles, videos, threads, posts)
    array = []
    if !articles.empty?
      articles.each do |article|
        array << article
      end
    end

    if !videos.empty?
      videos.each do |video|
        array << video
      end
    end

    if !threads.empty?
      threads.each do |thread|
        array << thread
      end
    end

    if !posts.empty?
      posts.each do |post|
        array << post
      end
    end

    return array.sort_by { |obj| obj.created_at }
  end
end
```

## uploads\_controller

```
class UploadsController < ApplicationController
  def create
    @upload = Upload.create(params[:upload])
    respond_to do |format|
      format.js
    end
  end
  def destroy
    @upload = Upload.find(params[:id])
    render :destroy
  end
end
```

## user\_profiles\_controller

```
# encoding: utf-8
class UsersProfilesController < ApplicationController
  def update
    @user = User.find(params[:id])

    if params[:user][:password].blank?
      params[:user].delete("password")
      params[:user].delete("password_confirmation")
      params[:user].delete("current_password")

      if @user.update_attributes(params[:user])

        flash[:notice] = "Actualizado con éxito"
        redirect_to users_profile_path(@user)
      else
        flash[:error] = @user.errors.full_messages
        redirect_to edit_user_registration_path
      end
    end

    elsif @user.update_with_password(params[:user])
      sign_in @user, :bypass => true

      flash[:notice] = "Actualizado con éxito"
      redirect_to users_profile_path(@user)
    else
      flash[:error] = @user.errors.full_messages
      redirect_to edit_user_registration_path
    end
  end

  def show
    @user = User.find(params[:id])
  end

  def ban_user
    @user = User.find(params[:id])
    if current_user.role == User::COLLABORATOR
      @user.banned = true
      @user.save
      flash[:alert] = "Usuario suspendido"
      redirect_to users_profile_path(@user)
    else
      flash[:alert] = "No tienes Permisos para suspender usuarios"
      redirect_to users_profile_path(@user)
    end
  end
end
```

```
    end
  end

  def unban_user
    @user = User.find(params[:id])
    if current_user.role == User::COLLABORATOR
      @user.banned = false
      @user.save
      flash[:alert] = "Usuario libre"
      redirect_to users_profile_path(@user)
    else
      flash[:alert] = "No tienes permisos para quitar suspensiones"
      redirect_to users_profile_path(@user)
    end
  end
end
```

## Anexo B Código Modelos

### Blog

```
class Blogger::Blog < ActiveRecord::Base
  # attr_accessible :title, :body
  attr_protected nil
  belongs_to :user
  has_many :posts, :class_name => 'Blogger::Post', dependent: :destroy
  validates_uniqueness_of :user_id
  validates_presence_of :blog_name

  def self.create_blog(blog_data, user_id)
    blog = Blogger::Blog.where(user_id: user_id).first
    unless blog
      blog = Blogger::Blog.create(:blog_name => blog_data[:blog_name],
                                  :blog_description => blog_data[:blog_description],
                                  :user_id => user_id
                                )
    end
    blog
  end
end
```

### Comment

```
class Blogger::Comment < ActiveRecord::Base
  # attr_accessible :title, :body
  attr_protected nil
  belongs_to :post, :class_name => 'Blogger::Post'
  belongs_to :user
  validates_presence_of :comment_content
  validates_uniqueness_of :comment_content
  validates_length_of :comment_content, :minimum => 5, :maximum => 500, :allow_blank
=> false
end
```

## post

```

class Blogger::Post < ActiveRecord::Base
  scope :excluding_last, lambda {
    last = Blogger::Post.last(order: "updated_at desc", limit: 1)
    return [] unless last
    where("id <> #{last.id}")
  }

  attr_accessible :post_title, :post_content, :blog_id, :tag_list, :name
  belongs_to :blog, :class_name => 'Blogger::Blog'
  has_many :comments, :class_name => 'Blogger::Comment', dependent: :destroy
  validates_presence_of :post_title, :post_content
  validates_uniqueness_of :post_title
  acts_as_taggable

  def self.create_post(parameters, blog_id)
    post = Blogger::Post.new(
      post_title:parameters[:post_title],
      post_content:parameters[:post_content],
      blog_id:blog_id
    )
    post.tag_list = post.find_tags(parameters[:tag_list])
    post.save
    post
  end

  def find_tags(tags)
    split_tags = tags.split(",")
    split_tags.each_with_index do |tag, index|
      unless tag.numeric?
        split_tags[index] = ActsAsTaggableOn::Tag.create(name: tag.gsub(/<<<(.*)>>>/,
'\1')).id
      end
    end
    ActsAsTaggableOn::Tag.where(id: split_tags).map(&:name).join(",")
  end

  def update_post(parameters)
    parameters[:tag_list] = find_tags(parameters[:tag_list])
    self.update_attributes(parameters)
  end
end

```

## Comment

```
class Forummer::Comment < ActiveRecord::Base
  attr_accessible :comment_content, :thread_id, :user_id
  attr_accessor :forum_id
  validates :comment_content, :presence => true

  belongs_to :user
  belongs_to :thread, :class_name => "Forummer::Thread"
end
```

## Forum

```
class Forummer::Forum < ActiveRecord::Base
  has_many :themes, :through => :forum_themes, :class_name => "Forummer::Theme"
  has_many :forum_themes, :class_name => "Forummer::ForumTheme"
  has_many :threads, dependent: :destroy
  attr_accessible :name, :description, :ending_forum
  attr_accessor :theme_id
  before_destroy :destroy_owned_themes
  validates :name, :description, :presence => true, :uniqueness => true

  def themes_from_forum
    themes = []
    self.forum_themes.where(owner_forum: true).each do |relation|
      themes << relation.theme
    end
    themes
  end

  def destroy_owned_themes
    self.forum_themes.where(owner_forum: true).each do |relation|
      relation.theme.destroy
      relation.destroy
    end
  end

  def has_forums?
    if self.themes_from_forum.empty?
      return false
    else
      forums = []
      self.themes_from_forum.each do |theme|
        unless theme.forums_from_theme.empty?
          theme.forums_from_theme.each do |sub_forum|

```

```
        forums << sub_forum
      end
    end
  end

  if forums.length > 0
    return true
  else
    return false
  end
end
end

def sub_forums
  sub_forums = []
  if self.themes_from_forum.empty?
    return sub_forums
  else
    self.themes_from_forum.each do |theme|
      unless theme.forums_from_theme.empty?
        theme.forums_from_theme.each do |sub_forum|
          sub_forums << sub_forum
        end
      end
    end
    return sub_forums
  end
end

def sub_forums_threads
  threads = []
  if self.has_forums?
    self.sub_forums.each do |sub_forum|
      if sub_forum.ending_forum?
        sub_forum.threads.each do |thread|
          threads << thread
        end
      elsif sub_forum.has_forums?
        sub_forum.sub_forum_threads.each do |thread|
          threads << thread
        end
      end
    end
    return threads
  elsif self.ending_forum?
    self.threads.each do |thread|
```



```

        threads << thread
      end
      return threads
    else
      return threads
    end
  end
end
end

```

## Forum\_theme

```

class Forummer::ForumTheme < ActiveRecord::Base
  belongs_to :forum, :class_name => "Forummer::Forum"
  belongs_to :theme, :class_name => "Forummer::Theme"
  attr_accessible :forum_id, :theme_id, :owner_forum, :owner_theme

  def self.themes_from_forum(forum_id)
    themes = []
    self.where(forum_id: forum_id, owner_forum: true).each do |relation|
      themes << relation.theme
    end
    themes
  end

  def self.create_owned_theme_relation(forum_id)
    self.create(forum_id: forum_id,
                theme_id: Forummer::Theme.order("created_at").last.id,
                owner_forum: true
               )
  end

  def self.create_owned_forum_relation(theme_id)
    self.create(forum_id: Forummer::Forum.order("created_at").last.id,
                theme_id: theme_id,
                owner_theme: true
               )
  end
end
end

```

## Theme

```

class Forummer::Theme < ActiveRecord::Base
  has_many :forums, :through => :forum_themes, :class_name => "Forummer::Forum"
  has_many :forum_themes, :class_name => "Forummer::ForumTheme"

```

```
attr_accessible :name
attr_accessor :forum_id
before_destroy :destroy_owned_forums
validates :name, :presence => true, :uniqueness => true

def forums_from_theme
  forums = []
  self.forum_themes.where(owner_theme: true).each do |relation|
    forums << relation.forum
  end
  forums
end

def destroy_owned_forums
  self.forum_themes.where(owner_theme: true).each do |relation|
    relation.forum.destroy
    relation.destroy
  end
end
end
```

## Thread

```
class Forummer::Thread < ActiveRecord::Base
  belongs_to :forum, :class_name => "Forummer::Forum"
  belongs_to :user
  has_many :forummer_comments, dependent: :destroy, :class_name => "Forummer::Comment"
  attr_accessible :forum_id, :name, :sticky, :question
  validates :name, :presence => true
  validates :question, :presence => true
end
```

## Video

```
class Workshop::Video < ActiveRecord::Base
  attr_accessible :url, :thumbnail_1, :thumbnail_2, :thumbnail_3, :user_id
  belongs_to :user
  validates :url, :presence => true
end
```

## Administrator

```
class Administrator < ActiveRecord::Base
  # Include default devise modules. Others available are:
  # :token_authenticatable, :confirmable,
  # :lockable, :timeoutable and :omniauthable
  devise :database_authenticatable,
         :recoverable, :rememberable, :trackable, :validatable

  # Setup accessible (or protected) attributes for your model
  attr_accessible :email, :password, :password_confirmation, :remember_me
  # attr_accessible :title, :body
end
```

## Article

```
class Article < ActiveRecord::Base
  attr_accessible :title, :article_content, :user_id
  belongs_to :user
  validates :title, :presence => true
  validates :article_content, :presence => true
end
```

## Avatar

```
class Avatar < ActiveRecord::Base
  attr_accessible :file, :user_id
  belongs_to :user
  mount_uploader :file, AvatarUploader

  validates_presence_of :file

  before_create :default_name

  def default_name
    self.name ||= File.basename(file.filename, '*.*').titleize if file
  end
end
```

## Blogger

```
module Blogger
  def self.table_name_prefix
    'blogger_'
  end
end
```

## Company

```
# encoding: utf-8
class Company < ActiveRecord::Base
  attr_accessible :description, :name, :research_field_id, :company_logo

  belongs_to :research_field
  validates :description, :name, :research_field_id, :company_logo, presence: true
  validate :company_logo_is_image, :on => :create
  validate :company_logo_is_image, :on => :update

  def company_logo_is_image
    allowed_types = [:jpg, :png, :jpeg]
    type = FastImage.type(self.company_logo)
    unless allowed_types.include? type || !type.nil?
      errors.add(:company_logo, "No es una imagen válida")
    end
  end
end
```

## Contact

```
class Contact < ActiveRecord::Base
  attr_accessible :contact_text, :email, :name, :occupation
end
```

## Forummer

```
module Forummer
  def self.table_name_prefix
    'forummer_'
  end
end
```

## Research\_field

```
class ResearchField < ActiveRecord::Base
  attr_accessible :research_name
  has_many :companies
  has_many :users

  before_destroy :unleash_users

  def unleash_users
    users = User.where(research_field_id: self.id)
    users.each do |user|
      user.update_attribute(:research_field_id, nil)
    end
  end
end
```

## Upload

```
class Upload < ActiveRecord::Base
  attr_accessible :file, :user_id
  belongs_to :user
  attr_accessor :blog_id
  attr_accessor :thread_id
  mount_uploader :file, FileUploader

  validates_presence_of :file

  before_create :default_name
```

```

def default_name
  self.name ||= File.basename(file.filename, '*').titleize if file
end
end

```

## User

```

# encoding: utf-8
class User < ActiveRecord::Base
  # Include default devise modules. Others available are:
  # :token_authenticatable, :confirmable,
  # :lockable, :timeoutable and :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :trackable, :validatable
  devise :omniauthable, :omniauth_providers => [:facebook, :twitter, :google_oauth2]
  # Setup accessible (or protected) attributes for your model
  attr_accessible :research_field_id, :investigator, :current_password, :sex, :email,
  :user_image, :password, :provider, :uid, :password_confirmation, :remember_me, :role,
  :name, :nickname, :ocupation, :interests, :email, :phone, :address, :birthday, :banned
  # attr_accessible :title, :body
  PARTICIPANT = 'Participant'
  COLLABORATOR = 'Collaborator'
  MALE = 'Hombre'
  FEMALE = 'Mujer'

  scope :investigators, -> {where(investigator: true)}

  OCUPATIONS = [ "Estudiante", "Maestro", "Investigador", "Interesado", "Otro" ]

  validates_format_of :phone,
                    :with => /\A[0-9]{10}\Z/,
                    :allow_blank => true,
                    :allow_nil => true

  ROLES = [PARTICIPANT, COLLABORATOR]
  SEX = [MALE, FEMALE]

  has_one :blog, :class_name => 'Blogger::Blog', dependent: :destroy
  has_many :comments, :class_name => 'Blogger::Comment', dependent: :destroy
  has_many :forummer_comments, :class_name => 'Forummer::Comment', dependent: :destroy
  has_many :uploads, dependent: :destroy
  has_one :avatar, dependent: :destroy
  has_many :threads, :class_name => 'Forummer::Thread', dependent: :destroy
  has_many :videos, :class_name => 'Workshop::Video', dependent: :destroy

```

```
has_many :articles, dependent: :destroy
belongs_to :research_field
after_create :set_new_user_role

def self.find_for_facebook_oauth(auth, signed_in_resource = nil)
  user = User.where(:provider => auth.provider, :uid => auth.uid).first
  unless user
    user = User.create(name:auth.extra.raw_info.name,
                      provider:auth.provider,
                      uid:auth.uid,
                      email:auth.info.email,
                      password:Devise.friendly_token[0,20],
                      role: PARTICIPANT,
                      user_image:auth.info.image
                      )
  end
  user
end

def self.find_for_google_oauth2(access_token, signed_in_resource=nil)
  data = access_token.info
  user = User.where(email: data["email"], provider: access_token["provider"] ).first

  unless user
    user = User.create(name: data["name"],
                      email: data["email"],
                      password: Devise.friendly_token[0,20],
                      provider: access_token["provider"],
                      role: PARTICIPANT,
                      user_image: data["image"]
                      )
  end
  user
end

def self.find_for_twitter_oauth(auth, signed_in_resource = nil)
  user = User.where(:provider => auth.provider, :uid => auth.uid).first
  unless user
    user = User.create( name:auth.info.name,
                      provider:auth.provider,
                      uid:auth.uid,
                      email:"#{auth.info.nickname}@default.com",
                      password:Devise.friendly_token[0,20],
                      role: PARTICIPANT,
                      user_image:auth.info.image,
                      nickname: auth.info.nickname
                      )
  end
end
```

```
    end
  user
end

def set_new_user_role
  update_attribute(:role, PARTICIPANT)
end
end
```

## workshop

```
module Workshop
  def self.table_name_prefix
    'workshop_'
  end
end
```

## worshoper

```
module Workshoper
  def self.table_name_prefix
    'workshoper_'
  end
end
```



## Anexo C Código Vistas

\_article.html.haml

```
.well.media.bordered-well
  .row-fluid
    .forum-track{style: "max-width:674px; padding:0;"}
      %h3.pull-left.forum-header.search-content{style: "max-width:614px; padding-
left:0; padding-right:0; "}
        = link_to article.title, article_path(article), class: "normal-link"
        - if user_signed_in? && current_user == article.user
          .btn-group.pull-right
            = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
              %i.icon-cog
              %span.caret
              %ul.dropdown-menu.pull-right
                %li= link_to "Editar Artículo", edit_article_path(article)
                %li= link_to "Borrar Artículo", article_path(article), method: :delete,
data: { confirm: "¿Esta seguro?" }

    .row-fluid
      %strong
        Publicado:
        = l( article.created_at, format: "%d %b. %Y")
      .post-content.word-wrapping-break
        = article.article_content.html_safe

= render "global_article_footer", article: article
```

\_global\_article\_footer.html.haml

```
.row-fluid.post-footer

  .pull-right.share-container
    %span.pull-left.share-span
      Compartir:
      = social_share_button_tag(article.title, url: article_url(article), img:
"/assets/logo-verde.png" )
```

edit.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      EDITAR ARTÍCULO
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
```

```

.row-fluid{style: "margin-top:5px;"}
  = link_to 'Regresar', :back, :class => 'normal-link pull-right'
  = simple_form_for @article do |f|
    = f.input :title, label: "Título"
    = f.input :article_content, label: "Contenido", input_html: { class: 'tinymce',
rows: 40, cols: 300 }
    = f.input :user_id, as: :hidden, input_html: { value: @article.user.id }
    .row-fluid
      = f.button :submit, 'Enviar', class: "submit-btn pull-right"

- if user_signed_in? && current_user.role == User::COLLABORATOR
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  = render "uploads/upload_form"

= tinymce

```

index.html.haml

```

.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      ARTÍCULOS
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
  .row-fluid{style: "margin-top: 5px; height:35px;"}
    - if user_signed_in? && current_user.role == User::COLLABORATOR
      .btn-group.pull-right
        = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
          %i.icon-cog
          %span.caret
          %ul.dropdown-menu.pull-right
            %li= link_to "Nuevo Articulo", new_article_path

  .row-fluid.blog-container{style: "margin-top: 5px;"}
    .span9
      = render @articles

      = paginate @articles
    .span3
      .well.bordered-well
        %h4.blue-text
          Archivo
        %ul.archive-ul

```

```

- Article.select("*").order("created_at DESC").group_by { |m|
m.created_at.beginning_of_month}.each do |date , articles|
  .accordion-group
    %li.closedli.accordion-heading
      = link_to l(date, format:"%B %Y") , "#Collapse#{date.strftime("%B")}",
class: "accordion-toggle black-bold-link"
    .accordion{id: "Collapse#{date.strftime("%B")}" }
      %ul.post-list-archive
        - articles.each do |inner_article|
          %li.closedli.post-li.word-wrapping-break
            = link_to inner_article.title, article_path(inner_article),
class: "black-bold-link"

```

new.html.haml

```

.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      NUEVO ARTÍCULO
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid{style: "margin-top:5px;"}
    = link_to 'Regresar', :back, :class => 'normal-link pull-right'
    = simple_form_for @article do |f|
      = f.input :title, label: "Título"
      = f.input :article_content, label: "Contenido", input_html: { class: 'tinymce',
rows: 40, cols: 300 }
      = f.input :user_id, as: :hidden, input_html: { value: current_user.id }
    .row-fluid
      = f.button :submit, 'Enviar', class: "submit-btn pull-right"

  - if user_signed_in? && current_user.role == User::COLLABORATOR
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    = render "uploads/upload_form"

= tinymce

```

show.html.haml

```

.container.new-object-form-layout.post-container
  .row-fluid

```

```

= image_tag "Linea_Separadora.png", class: "linea-separadora"
%h2.forum-header
  = @article.title
= image_tag "Linea_Separadora.png", class: "linea-separadora"

.row-fluid.blog-container
- if user_signed_in? && @article.user == current_user
  .btn-group.pull-right
    = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
      %i.icon-cog
      %span.caret
    %ul.dropdown-menu
      %li
        = link_to "Editar Publicación", edit_article_path(@article)
      %li
        = link_to "Borrar Publicación", article_path(@article), method: "delete"

.row-fluid{style: "margin-top:5px;"}
  = link_to 'Regresar', articles_path, :class => 'normal-link pull-right'
.row-fluid
  %strong
  Publicado:
  = @article.created_at.strftime("%d %b. %Y")

.row-fluid
  .post-content.word-wrapping-break
  = @article.article_content.html_safe

_avatar.html.haml

= image_tag avatar.file_url(:avatarize).to_s, data: {src:
avatar.file_url(:avatarize).to_s }

```

avatar\_upload.html.haml

```

- if user_signed_in? && current_user == @user
  = simple_form_for Avatar.new, :url => avatars_path, :html => { multipart: true,
:method => :post } do |avatar|
    = avatar.input :user_id, as: :hidden, :input_html => {value: current_user.id}
    = avatar.file_field :file, name: "avatar[file]", hidden: true
    = link_to "#", class: "faker-upload btn avatar-upload-faker" do
      cambiar foto
    .upload-percentage

```

## create.js.erb

```
<% if @avatar.new_record? %>
  alert("Error al subir: <%= j @avatar.errors.full_messages.join(', ').html_safe %>");
<% else %>
  $(".avatar").html("<%= j render(@avatar) %>");
  $(".upload-percentage").html("");
<% end %>
```

## edit.html.haml

```
.container.registration-content.new-object-form-layout.new-blog-container
  .row-fluid.container
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      =@blog.blog_name
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

    .row-fluid{style: "margin-bottom:5px;"}
      = link_to 'Regresar', users_profile_path(@blog.user), :class => 'normal-link pull-right'

  = simple_form_for @blog, :defaults => {:input_html => {:class => 'field-inputs'}} do
|b|
  = b.input :blog_name, :label => 'Nombre del Blog'
  = b.input :blog_description, :label => 'Descripción'
  = b.button :submit, 'Enviar', :class => 'pull-right submit-btn'
```

## index.html.haml

```
.container.new-object-form-layout.profile-container
  .blogs-container.blog-container
    = render @posts

    .row-fluid.posts-index
      = paginate @posts
```

## index.js.erb

```
$('.blogs-container').append('<%= j render(@posts) %>');
<% if link_to_next_page(@posts, "siguiente") %>
  $('.pagination').replaceWith('<%= j paginate(@posts) %>');
<% else %>
  $('.pagination').remove();
<% end %>
```

## new.html.haml

```
.container.registration-content.new-object-form-layout.new-blog-container
  .row-fluid.container
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
```

```

    Mi Blog
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

    .row-fluid{style: "margin-bottom:5px;"}
    = link_to 'Regresar', users_profile_path(current_user), :class => 'normal-link
pull-right'

    = simple_form_for @blog, :defaults => {:input_html => {:class => 'field-inputs'}} do
|b|
    = b.input :blog_name, :label => 'Nombre del Blog'
    = b.input :blog_description, :label => 'Descripción'
    = b.button :submit, 'Enviar', :class => 'pull-right submit-btn'

```

show.html.haml

```

.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
    = @blog.blog_name
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid.blog-container
    .span9
      - if user_signed_in? && current_user == @blog.user
        .row-fluid.blog-controls
          = link_to "Nueva Publicación", new_blogger_blog_post_path(@blog), :class =>
'pull-right submit-btn', style: "margin-right:0;"

        = render @posts

        .row-fluid
          = paginate @posts
        .span3
          .well.bordered-well.blog-profile
            - if @blog.user.avatar
              = link_to users_profile_path(@blog.user), class: "thumbnail avatar-thumb
container" do
                = image_tag @blog.user.avatar.file_url(:thumb).to_s
            - else
              = link_to users_profile_path(@blog.user), class: "thumbnail avatar-thumb
container" do
                = image_tag "Avatar.png"
            %strong.blue-text.blog-user-name
            = @blog.user.name
            %p.word-wrapping-break
            = @blog.user.address if !@blog.user.address.blank?

          .well.bordered-well
            %h4.blue-text
            Archivo
            %ul.archive-ul
              - @blog.posts.order("created_at DESC").group_by { |m|
m.created_at.beginning_of_month}.each do |date , post|
                .accordion-group
                %li.closedli.accordion-heading

```

```

      = link_to l(date, format: "%B %Y"), "#Collapse#{date.strftime("%B")}",
class: "accordion-toggle black-bold-link"
    .accordion{id: "Collapse#{date.strftime("%B")}" }
      %ul.post-list-archive
        - post.each do |innerpost|
          %li.closedli.post-li.word-wrapping-break
            = link_to innerpost.post_title,
blogger_blog_post_path(innerpost.blog, innerpost), class: "black-bold-link"

    .well.bordered-well
      %h4.tag-filter-list.blue-text
      Etiquetas
      - eliminate_duplicates(blog_tag_list(@blog)).each do |tag|
        %p.tag-filter
          = link_to tag, tag_path(tag, blog: @blog), class: "black-bold-link"

```

create.json.jbuilder

```

json.user_image      @new_comment.user.user_image
json.comment_content @new_comment.comment_content
json.id              @new_comment.id

```

comment\_form.html.haml

```

- if user_signed_in?
  = simple_form_for @comment do |new_comment|
    = new_comment.input :comment_content, :label => 'Comentar (máximo de 500
caracteres)'
    = new_comment.input :user_id, :as => :hidden, :input_html => {:value =>
current_user.id}
    = new_comment.input :post_id, :as => :hidden, :input_html => {:value => @post.id}
    .row-fluid
      = new_comment.button :submit, 'Comentar', :class => 'submit-btn pull-right',
style: "margin-right:0;"
- else
  %h5
  %p.pull-left
    Para comentar necesitas
    = link_to 'iniciar sesión', new_user_session_path, :class => 'normal-link'

```

comment\_partial.html.haml

```

= image_tag "Linea_Separadora.png", class: "linea-separadora"
%h4.blue-text
  Comentarios
  - @post.comments.each do |comment|
    .comment.media
      - if comment.user.avatar
        = link_to users_profile_path(comment.user), class: "thumbnail avatar-thumb pull-
left" do
          = image_tag comment.user.avatar.file_url(:thumb).to_s
      - else

```

```

      = link_to users_profile_path(comment.user), class: "thumbnail avatar-thumb pull-
left" do
      = image_tag "Avatar.png"
      %p.blue-text
      .media-body
      .row-fluid
      %small.muted
      = comment.created_at.strftime("%d %b. %Y")
      %p.media-paragraph.word-wrapping-break.blogger-comment-content
      = comment.comment_content
      .row-fluid
      - if user_signed_in? && (current_user == comment.user || current_user.role ==
User::COLLABORATOR)
      = button_to 'Borrar comentario', blogger_comment_path(comment), :method =>
:delete, :class => 'submit-btn pull-right', style: "margin-right:0;"
      = image_tag "Linea_Separadora.png", class: "linea-separadora"

```

global\_post\_footer.html.haml

```

.row-fluid.post-footer
- unless post.tag_list.empty?
  .pull-left
  Etiquetas:
  = raw post.tag_list.map {|t| link_to t, tag_path(t), class: "black-bold-dotted-
link"}.join(', ')

  .pull-right.share-container
  %span.pull-left.share-span
  Compartir:
  = social_share_button_tag(post.post_title, url: blogger_blog_post_url(post.blog,
post), img: "/assets/logo-verde.png" )
  .pull-right.comment-count-container
  .comment-count
  %span= post.comments.count

  = link_to 'Comentar', blogger_blog_post_path(post.blog,post), :class => 'normal-link
pull-right'

```

\_post.html.haml

```

.well.media.bordered-well
- if params[:action] == "index"
  .profile.pull-left
  - if post.blog.user.avatar
    = link_to users_profile_path(post.blog.user), class: "thumbnail avatar-thumb"
do
    = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
    = image_tag post.blog.user.avatar.file_url(:thumb).to_s
  - else
    = link_to users_profile_path(post.blog.user), class: "thumbnail avatar-thumb"
do
    = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"

```



```

    = image_tag "Avatar.png"
  - if !post.blog.user.nickname.blank?
    %p.blue-text{style: "margin-top:5px;"}
    = link_to post.blog.user.nickname, users_profile_path(post.blog.user),
class: "normal-link"
    %p.blue-text{style: "margin-top:5px;"}
    = link_to post.blog.blog_name, blogger_blog_path(post.blog), class: "normal-
link"

    .media-body
    .row-fluid
    %h3.media-heading.blue-text.post-title
    = link_to post.post_title, blogger_blog_post_path(post.blog,post), class:
'normal-link '

    - if user_signed_in? && post.blog.user == current_user
    .btn-group.pull-right
    = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
    %i.icon-cog
    %span.caret
    %ul.dropdown-menu
    %li
    = link_to "Editar Publicación", edit_blogger_blog_post_path(post.blog,
post)
    %li
    = link_to "Borrar Publicación", blogger_blog_post_path(post.blog, post),
method: "delete"

    .row-fluid
    %strong
    Publicado:
    = l( post.created_at, format: "%d %b. %Y")
    .post-content.word-wrapping-break
    = post.post_content.html_safe

= render "blogger/posts/global_post_footer", post: post

```

\_post\_footer.html.haml

```
.row-fluid.post-footer
- unless post.tag_list.empty?
  .pull-left
    Etiquetas:
    = raw post.tag_list.map {|t| link_to t, tag_path(t), class: "black-bold-dotted-link"}.join(', ')

  .pull-right.share-container
    %span.pull-left
      Compartir:
      = social_share_button_tag(post.post_title, url:
blogger_blog_post_url(post.blog,post), img: "/assets/logo-verde.png" )

= link_to 'Regresar', :back, :class => 'pull-right normal-link'
```

create.html.haml

```
.container
  .well
    = @post.post_title
    = @post.post_content.html_safe
    = link_to 'Regresar', blogger_blog_path(@post.blog)
    = render "blogger/posts/global_post_footer", post: @post
```

edit.html.haml

```
.container.new-object-form-layout{style: "position:relative;"}
= image_tag "Linea_Separadora.png", class: "linea-separadora"


### forum-header


  Editar Publicación
  = image_tag "Linea_Separadora.png", class: "linea-separadora"

= link_to 'Ver Publicación', blogger_blog_post_path(@post.blog, @post), :class =>
'pull-right submit-btn', style: "margin-right:0;"

= simple_form_for @post, :url => blogger_blog_post_path do |post|
  = post.input :post_title, label: 'Título'
  = post.input :tag_list, label: 'Etiquetas (separadas por comas)', input_html:
{data: {load: @post.tags.to_json}}
  = post.input :post_content, label: 'Contenido', input_html: { class: 'tinymce',
rows: 40, cols: 300 }
  .row-fluid
    = post.button :submit, "Enviar"

- if user_signed_in? && current_user.role == User::COLLABORATOR
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  = render "uploads/upload_form"

= tinymce
```

index.html.haml

```
.container.new-object-form-layout.profile-container
  .blogs-container.blog-container

    - if @blog
      .row-fluid
        = image_tag "Linea_Separadora.png", class: "linea-separadora"
        %h2.forum-header
          = @blog.blog_name
          %small.blue-text
            Etiqueta:
            %span= @tag
        = image_tag "Linea_Separadora.png", class: "linea-separadora"

      .row-fluid{style: "margin-bottom:5px;"}
        = link_to 'Regresar al blog', blogger_blog_path(@blog), :class => 'normal-link
pull-right'
    - else
      .row-fluid
        = image_tag "Linea_Separadora.png", class: "linea-separadora"
        %h2.forum-header
          = @tag
        = image_tag "Linea_Separadora.png", class: "linea-separadora"

      .row-fluid{style: "margin-bottom:5px;"}
        = link_to 'Regresar a la lista de publicaciones', blogger_blogs_path, :class
=> 'normal-link pull-right'

=render @posts

  .row-fluid.posts-index
    = paginate @posts
```

show.html.haml

```
.container.new-object-form-layout.post-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      = @post.blog.blog_name
      %small.muted= link_to "Ir al blog", blogger_blog_path(@post.blog), class:
"normal-link"
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid.blog-container
    %h2.media-heading.blue-text.post-title
      = @post.post_title
    - if user_signed_in? && @post.blog.user == current_user
      .btn-group.pull-right
        = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
          %i.icon-cog
```

```
      %span.caret
    %ul.dropdown-menu
      %li
        = link_to "Editar Publicación", edit_blogger_blog_post_path(@post.blog,
@post)
      %li
        = link_to "Borrar Publicación", blogger_blog_post_path(@post.blog,
@post), method: "delete"

    .row-fluid
      %strong
        Publicado:
        = @post.created_at.strftime("%d %b. %Y")

    .row-fluid
      .post-content.word-wrapping-break
        = @post.post_content.html_safe

= render "blogger/posts/post_footer", post: @post

    .row-fluid.comments
      = render 'blogger/posts/comments_partial'
    .row-fluid.comment-form
      = render 'blogger/posts/comment_form'
```

\_company.html.haml

```
.row-fluid.blog-container{style: "margin-bottom: 10px;"}
- if user_signed_in? && current_user.role == User::COLLABORATOR &&
company.research_field == current_user.research_field
  .btn-group.pull-right{style: "margin-left: 10px;"}
    = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
      %i.icon-cog
      %span.caret
      %ul.dropdown-menu.pull-right
        %li= link_to "Borrar Empresa", company_path(company), method: :delete, data: {
confirm: "¿Esta seguro?" }
        %li= link_to "Editar Empresa", edit_company_path(company)
  .media
    = image_tag company.company_logo, class: "media-object pull-left company-logo-
style"

    .media-body{style: "padding: 0 10px 0 10px;"}
      .media-heading
        %h4
          = company.name

      .row-fluid
        .word-wrapping-break
          = company.description.html_safe

= image_tag "Linea_Separadora.png", class: "linea-separadora"
```

edit.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      NUEVA EMPRESA
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid{style: "margin-top:5px;"}
    = link_to 'Regresar', :back, :class => 'normal-link pull-right'
    = simple_form_for @company do |f|
      = f.input :name, label: "Nombre"
      = f.input :description, label: "Descripción", input_html: { class: 'tinymce',
rows: 40, cols: 300 }
      = f.association :research_field, label: "Campo de desarrollo", label_method:
:research_name, value_method: :id
      = f.input :company_logo, label: "URL del logo (necesita ser imagen, de preferencia
utilizar el sistema)"
      .row-fluid
        = f.button :submit, 'Enviar', class: "submit-btn"

    - if user_signed_in? && current_user.role == User::COLLABORATOR
      = image_tag "Linea_Separadora.png", class: "linea-separadora"
      = render "uploads/upload_form"

= tinymce
```

index.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      EMPRESAS
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    - if user_signed_in? && current_user.role == User::COLLABORATOR
      .row-fluid{style: "margin-top: 5px; height:35px;}
        .btn-group.pull-right
          = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
            %i.icon-cog
            %span.caret
          %ul.dropdown-menu.pull-right
            %li= link_to "Nueva Empresa", new_company_path
        = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid.blog-container{style: "margin-top: 5px;}
    = render @companies

    = paginate @companies
```

new.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      NUEVA EMPRESA
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid{style: "margin-top:5px;}
    = link_to 'Regresar', :back, :class => 'normal-link pull-right'
    = simple_form_for @company do |f|
      = f.input :name, label: "Nombre"
      = f.input :description, label: "Descripción", input_html: { class: 'tinymce',
rows: 40, cols: 300 }
      = f.association :research_field, label: "Campo de desarrollo", label_method:
:research_name, value_method: :id, collection: ResearchField.where(id:
current_user.research_field.id)
      = f.input :company_logo, label: "URL del logo (necesita ser imagen, de preferencia
utilizar el sistema)"
      .row-fluid
        = f.button :submit, 'Enviar', class: "submit-btn"

    - if user_signed_in? && current_user.role == User::COLLABORATOR
      = image_tag "Linea_Separadora.png", class: "linea-separadora"
      = render "uploads/upload_form"

= tinymce
```

show.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      EMPRESA
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid.blog-container{style: "margin-top: 5px;"}

    - if user_signed_in? && current_user.role == User::COLLABORATOR &&
@company.research_field == current_user.research_field
      .btn-group.pull-right{style: "margin-left: 10px;"}
        = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
          %i.icon-cog
          %span.caret
          %ul.dropdown-menu.pull-right
            %li= link_to "Borrar Empresa", company_path(@company), method: :delete,
data: { confirm: "¿Esta seguro?" }
            %li= link_to "Editar Empresa", edit_company_path(@company)
      .row-fluid{style: "margin-top:5px; width:200px;"}
        = link_to 'Regresar', companies_path, :class => 'normal-link pull-left'

  .media
    = image_tag @company.company_logo, class: "media-object pull-left company-logo-
style"

    .media-body{style: "padding: 0 10px 0 10px;"}
      .media-heading
        %h4
          = @company.name

      .row-fluid
        .word-wrapping-break
          = @company.description.html_safe
```

Correo de contacto

```
Usuario: <%= @contact.name %>
Correo: <%= @contact.email %>
Ocupación: <%= @contact.occupation %>
Mensaje: <%= @contact.contact_text %>
```

Index.html.haml

```

.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
  %h2.forum-header
    CONTACTO
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
  .row-fluid.map
    .span8
      <iframe width="630" height="325" frameborder="0" scrolling="no" marginheight="0"
marginwidth="0"
src="https://maps.google.com.mx/maps?ie=UTF8&cid=3474708155229719444&q=Centro+
de+Investigaci%C3%B3n+Cient%C3%ADfica+de+Yucat%C3%A1n+CICY&gl=MX&hl=es-
419&t=m&iwloc=A&ll=21.029792,-
89.637494&spn=0.006295,0.006295&output=embed"></iframe><br /><small><a
href="https://maps.google.com.mx/maps?ie=UTF8&cid=3474708155229719444&q=Centro+
de+Investigaci%C3%B3n+Cient%C3%ADfica+de+Yucat%C3%A1n+CICY&gl=MX&hl=es-
419&t=m&iwloc=A&ll=21.029792,-
89.637494&spn=0.006295,0.006295&source=embed" style="color:#0000FF;text-
align:left">Ver mapa más grande</a></small>
    .span4
      .investigator-profile-box{style: "background-height:350px; height:350px;}
      .row-fluid{style: "height:80px;}
      %h4.blue-text.cotact-cicy-title
        Centro de Investigación Científica de Yucatán CICY
      .row-fluid
      %p.black-text.contact-cicy-address
        Chuburna de Hidalgo, 97200 Mérida, Yucatán
      %br
      01 999 942 8330

      .row-fluid
      %p.black-text.contact-cicy-address{style: "font-weight:bold;}
        Energía Renovable
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
  .row-fluid.mailer
  %h4.blue-text
    Dudas y Comentarios
  %p.span8.black-text{style: "margin-left:0;}
    Llena el formulario y un representante de nuestro equipo te respondera tan
pronto como sea posible. Gracias

  .row-fluid
  = simple_form_for @contact do |f|
    = f.input :name, label: "Nombre Completo"
    = f.input :email, label: "Correo"
    = f.input :occupation, label: "Ocupación"
    = f.input :contact_text, label: "Mensaje"
  .row-fluid
  = f.button :submit, 'Enviar', class: "submit-btn"

```

Edit.html.haml

```

.container.new-object-form-layout{style: "width:100%;"}
  = image_tag "Linea_Separadora.png", class: "linea-separadora"

```



```


### forum-header


  Editar tu comentario
%p.header-welcome
  Edita tu comentario de manera dinámica
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  = simple_form_for @comment do |comment|
    = comment.error_notification
    = comment.input :comment_content, label: "Editar comentario: ", input_html: {
class: 'tinymce', rows: 40, cols: 300 }
    = comment.button :submit

  = render "uploads/upload_form"

= tinymce

```

advancedsearch\_field.html.haml

```

.row-fluid.well
  = search_form_for @search, url: forummer_search_forums_path, html: {method:
:post} do |f|
    .row-fluid
      .span4
        .forum-track
          = image_tag "Linea_Separadora.png", class: "linea-separadora"
          %h3.forum-header
            Buscar en discusiones
          = image_tag "Linea_Separadora.png", class: "linea-separadora"

          = f.label :titulo_contiene_texto
          = f.text_field :name_present

          = f.label :discusion_contiene_texto
          = f.text_field :question_cont

        .span4
          .forum-track
            = image_tag "Linea_Separadora.png", class: "linea-separadora"
            %h3.forum-header
              Buscar en comentarios
            = image_tag "Linea_Separadora.png", class: "linea-separadora"

            = f.label :comentario_contiene_texto
            = f.text_field :forummer_comments_comment_content_cont

        .span4
          .forum-track
            = image_tag "Linea_Separadora.png", class: "linea-separadora"
            %h3.forum-header
              Buscar en foros
            = image_tag "Linea_Separadora.png", class: "linea-separadora"

            = f.label :foro_contiene_texto
            = f.text_field :forum_name_cont
          = image_tag "Linea_Separadora.png", class: "linea-separadora"
        .row-fluid

```

```
= f.submit "Buscar", class: "submit-btn pull-right"
```

```
endig_forum.html.haml
```

```
%thead
  %tr
    %th.information-cell
      Información
    %th.answer-cell
      Última respuesta
      - if user_signed_in?
        .btn-group.pull-right
          = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
            %i.icon-cog
            %span.caret
            %ul.dropdown-menu.pull-right
              %li= link_to "Nuevo Hilo", new_forummer_thread_path(forum: @forum.id)
%tbody.forum-threads
  - if !@sticky_threads.empty?
    %tr.forum-theme
      %td
        Discusiones
      %td
        - @sticky_threads.each do |sticky_thread|

          %tr.theme-forums
            %td
              .media
                .pull-left
                  = image_tag "logo-verde.png", class: "media-object"
                .media-body
                  %p.media-heading= link_to "#{sticky_thread.name}",
forummer_forum_thread_path(@forum, sticky_thread), class: "normal-link"
                  %small.muted.media= sticky_thread.created_at
            %td.answers
              - unless sticky_thread.forummer_comments.empty?
                .media
                  .pull-left
                    %p=
sticky_thread.forummer_comments.order("created_at").last.user.email
                    %small=
sticky_thread.forummer_comments.order("created_at").last.created_at
              - else
                No se encontraron respuestas

          - elsif user_signed_in? && (current_user.role == "Collaborator" &&
@sticky_threads.empty?)
            %tr
              %td
                %p.text-warning Te recomendamos crear como mínimo un hilo de discusión con las
reglas de etiqueta para este foro
              %td
```

```
|  |
| --- |
| Preguntas generales |
| - @threads.each do |thread|         |  |  | | --- | --- | |  | link_to "#{thread.name}", forummer_forum_thread_path(@forum, thread), class: "normal-link" | | thread.created_at | thread.forummer_comments.empty?           ? thread.forummer_comments.order("created_at").last.user.email           : No hay respuestas | |
| footer-cell | footer-cell |

```

\_themes.html.haml

```
|  |  | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| theme.name | - if user_signed_in? && current_user.role == "Collaborator"          caret  - link_to "Nuevo Foro", new_forummer_forum_path(theme: theme.id) - link_to "Borrar tema", forummer_theme_path(theme, forum_id: @forum.id), method: "delete" - link_to "Editar tema", edit_forummer_theme_path(theme, forum_id: @forum.id) |
| theme.forums_from_theme.each do |forum|         - if !forum.ending_forum? && forum.themes_from_forum.empty? && user_signed_in? && current_user.role == "Collaborator"           |  |  | | --- | --- | |  | forum | |

```

```

        .media-body
        %p.media-heading= link_to "#{forum.name}",
forummer_forum_themes_path(forum: forum.id), class: "normal-link"
        %small.muted.media= forum.description.nil? ? "Foro sin descripción" :
forum.description

        %td.count-cell
        = forum.themes_from_forum.count
        %td.count-cell
        = forum.sub_forums_threads.count
        %td
        .media
        .pull-left
        %p.text-error En construcción

- elsif !forum.ending_forum? && !forum.themes_from_forum.empty? &&
!forum.has_forums? && user_signed_in? && current_user.role == "Collaborator"
    %tr.theme-forums
        %td
        .media
        .pull-left
        = image_tag "logo-gris.png", class: "media-object"
        .media-body
        %p.media-heading= link_to "#{forum.name}",
forummer_forum_themes_path(forum: forum.id), class: "normal-link"
        %small.muted.media= forum.description.nil? ? "Foro sin descripción" :
forum.description
        %td.count-cell
        = forum.themes_from_forum.count
        %td.count-cell
        = forum.sub_forums_threads.count
        %td
        .media
        .pull-left
        %p.text-error En construcción

- elsif !forum.ending_forum? && forum.has_forums?
    %tr.theme-forums
        %td
        .media
        .pull-left
        = image_tag "logo-verde.png", class: "media-object"
        .media-body
        %p.media-heading= link_to "#{forum.name}",
forummer_forum_themes_path(forum: forum.id), class: "normal-link"
        %small.muted.media= forum.description.nil? ? "Foro sin descripción" :
forum.description

        %td.count-cell
        = forum.themes_from_forum.count
        %td.count-cell
        = forum.sub_forums_threads.count
        %td
        - if threads_from_forums(forum.sub_forums).empty?
        .media

```

```

        .pull-left
        %p Te invitamos a entrar

    - elsif comments_from_threads(threads_from_forums(forum.sub_forums)).empty?
      .media
      .pull-left
      %p= link_to threads_from_forums(forum.sub_forums).sort! {|a,b|
a.created_at <=> b.created_at}.last.name,
forummer_thread_path(threads_from_forums(forum.sub_forums).sort! {|a,b| a.created_at
<=> b.created_at}.last), class: "normal-link"
      %small.muted= threads_from_forums(forum.sub_forums).sort! {|a,b|
a.created_at <=> b.created_at}.last.created_at

    - else
      .media
      .pull-left
      %p= link_to
comments_from_threads(threads_from_forums(forum.sub_forums)).sort! {|a,b| a.created_at
<=> b.created_at}.last.thread.name,
forummer_thread_path(comments_from_threads(threads_from_forums(forum.sub_forums)).sort
! {|a,b| a.created_at <=> b.created_at}.last.thread), class: "normal-link"
      %small.muted=
comments_from_threads(threads_from_forums(forum.sub_forums)).sort! {|a,b| a.created_at
<=> b.created_at}.last.thread.created_at

    - elsif forum.ending_forum?
      %tr.theme-forums
      %td
      .media
      .pull-left
      = image_tag "logo-verde.png", class: "media-object"
      .media-body
      %p.media-heading= link_to "#{forum.name}",
forummer_forum_themes_path(forum: forum.id), class: "normal-link"
      %small.muted.media= forum.description.nil? ? "Foro sin descripción" :
forum.description
      %td.count-cell
      = forum.themes_from_forum.count
      %td.count-cell
      = forum.sub_forums_threads.count
      %td
      - if forum.threads.empty?
        .media
        .pull-left
        %p Te invitamos a hacer una pregunta

      - elsif comments_from_threads(forum.threads).empty?
        .media
        .pull-left
        %p= link_to forum.threads.order("created_at").last.name,
forummer_thread_path(forum.threads.order("created_at").last)
        %small.muted= forum.threads.order("created_at").last.created_at

    - else
      .media
      .pull-left
      %p= link_to comments_from_threads(forum.threads).sort! {|a,b|

```

```
a.created_at <=> b.created_at}.last.thread.name,
forummer_thread_path(comments_from_threads(forum.threads).sort! {|a,b| a.created_at
<=> b.created_at}.last.thread), class: "normal-link"
      %small.muted= forum.threads.order("created_at").last.created_at
```

```
%tr.table-footer
  %td.footer-cell
  %td.footer-cell
  %td.footer-cell
  %td.footer-cell
```

index.html.haml

```
.forum-track.container
= image_tag "Linea_Separadora.png", class: "linea-separadora"
%h3.forum-header
  BIENVENIDO
%p.header-welcome
  Si es su primera visita a los foros no te olvides de revisar el
  %span= link_to "FAQ", forummer_faq_path, class: "normal-link"
  \.
  - if user_signed_in?
    Esperamos tu visita sea satisfactoria. Para comenzar a ver mensajes selecciona
    un foro a continuación
  - else
    Tal vez tengas que registrarte antes de poder participar, da clic
    %span= link_to "AQUI", new_user_registration_path , class: "normal-link"
    para registrarte. Para comenzar a ver mensajes selecciona un foro a
    continuación.

= image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @forum.name == "Foro Principal"
  = link_to "Regresar al inicio", forummer_main_forum_path, class: "normal-link"

- unless @forum.forum_themes.where(owner_theme: true).empty? ||
last_forum(@forum).name == "Foro Principal"
  %span foro anterior:
  = link_to "#{last_forum(@forum).name}", forummer_forum_themes_path(forum:
last_forum(@forum).id), class: "normal-link"

- if user_signed_in? && current_user.role == "Collaborator" && @forum.name != "Foro
Principal"
  .btn-group
    = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
      Acciones del foro
      %span.caret
    %ul.dropdown-menu.pull-right
      %li= link_to "Borrar foro", forummer_forum_path(@forum), method: "delete"
      %li= link_to "Editar foro", edit_forummer_forum_path(@forum)

.forum-table-container.container
  %table.forum-table.table.table-bordered
    - unless @forum.ending_forum?
```

```
%thead
%tr
  %th.forum-cell
    %span= @forum.name

  %th.theme-cell
    Temas

  %th.discussions-cell
    Discusiones

  %th.last-message-cell
    Último Mensaje
    - if user_signed_in? && current_user.role == "Collaborator" &&
!@forum.ending_forum
      .btn-group.pull-right
        = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"}
do
    %i.icon-cog
    %span.caret
    %ul.dropdown-menu.pull-right
      %li= link_to "Nuevo Tema", new_forummer_theme_path(forum: @forum.id)

%tbody
  = render "themes"

- else
= render "ending_forum"
```

edit.html.haml

```
.container.new-object-form-layout
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  %h3.forum-header
    Editar foro
  = image_tag "Linea_Separadora.png", class: "linea-separadora"

  %h4 En caso de cambiar la bandera de "Foro final", ningun contenido sera borrado
  (hilos de discusión, temas, foros secundarios, etc...)
  = simple_form_for @forum do |f|
    = f.error_notification
    = f.input :name, label: "Nombre del Foro"
    = f.input :ending_forum, label: "Foro Final? (convierte el foro en un foro final
    permitiendo al usuario crear hilos de discusión)"
    = f.input :description, label: "Descripción"
    .row-fluid
      = f.button :submit, 'Enviar'
```

faq.html.haml

```
.container.pattern-background{style:"padding:10px; width:940px;"}
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  %h1.forum-header
    F A Q
  = image_tag "Linea_Separadora.png", class: "linea-separadora"

  %ul.faq-list
    %li
      %i.icon-circle-arrow-right
      = link_to '¿Que son los foros?', "#preg1", class: "normal-link"
    %li
      %i.icon-circle-arrow-right
      = link_to '¿Cualquiera puede participar?', "#preg2", class: "normal-link"
    %li
      %i.icon-circle-arrow-right
      = link_to '¿Si no deseo registrarme?', "#preg3", class: "normal-link"
    %li
      %i.icon-circle-arrow-right
      = link_to '¿Hay alguna regla?', "#preg4", class: "normal-link"

  .faq-answer#preg1
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    = link_to '¿Que son los foros?', "#preg1", class: "normal-link answer-question"
    %p
      Los foros son una zona dedicada a la resolucio de dudas, un lugar donde puedes
      entrar a compartir información sobre los temas selectos y conseguir una respuesta a
      tus preguntas o dudas.

    = link_to "#", class: "back-top-link" do
      Volver arriba
      %i.icon-circle-arrow-up
```



```
.faq-answer#preg2
= image_tag "Linea_Separadora.png", class: "linea-separadora"
= link_to '¿Cualquiera puede participar?', "#preg1", class: "normal-link answer-
question"
%p
  ¡Por supuesto!, solamente necesitas registrarte via este formulario:
  %span= link_to "Registro.", new_user_registration_path, class: "normal-link
answer-question", style: "width:100px;"

= link_to "#", class: "back-top-link" do
  Volver arriba
  %i.icon-circle-arrow-up
```

```
.faq-answer#preg3
= image_tag "Linea_Separadora.png", class: "linea-separadora"
= link_to '¿Si no deseo registrarme todavia puedo participar?', "#preg1", class:
"normal-link answer-question"
%p
  Los usuarios que acceden a los foros pueden leer toda la informacion que
nuestros participantes publiquen de manera totalmente libre pero el registro te
permitira compartir, hacer preguntas y formar parte de esta comunidad.

= link_to "#", class: "back-top-link" do
  Volver arriba
  %i.icon-circle-arrow-up
```

```
.faq-answer#preg4
= image_tag "Linea_Separadora.png", class: "linea-separadora"
= link_to '¿Hay alguna regla?', "#preg1", class: "normal-link answer-question"
%p
  Cada foro tiene sus propias reglas pero las principales son:
  %ul.faq-list{style: "margin-left:15px;"}
    %li
      %i.icon-circle-arrow-right
      %span Realizar preguntas o comentar conforme a la temática.
    %li
      %i.icon-circle-arrow-right
      %span No subir contenido fuera de lugar, ilegal u ofensivo.
    %li
      %i.icon-circle-arrow-right
      %span Respeto entre los usuarios de esta tu comunidad.

= link_to "#", class: "back-top-link" do
  Volver arriba
  %i.icon-circle-arrow-up

= image_tag "Linea_Separadora.png", class: "linea-separadora"
```

new.html.haml

```
.container.new-object-form-layout
= image_tag "Linea_Separadora.png", class: "linea-separadora"
%h3.forum-header
```

```

Nuevo foro
= image_tag "Linea_Separadora.png", class: "linea-separadora"
= simple_form_for @new_forum do |f|
  = f.error_notification
  = f.input :name, label: "Nombre del Foro"
  = f.input :theme_id, as: :hidden, input_html: { value: @theme }
  = f.input :ending_forum, label: "Foro Final? (convierte el foro en un foro final
  permitiendo al usuario crear hilos de discusión)"
  = f.input :description, label: "Descripción"
  .row-fluid
    = f.button :submit, 'Enviar'

```

edit.html.haml

```

.container.new-object-form-layout
= image_tag "Linea_Separadora.png", class: "linea-separadora"
%h3.forum-header
  Editar Tema
= image_tag "Linea_Separadora.png", class: "linea-separadora"

= simple_form_for @theme do |f|
  = f.error_notification
  = f.input :name, label: "Nombre del Tema"
  = f.input :forum_id, as: :hidden, input_html: { value: @forum_id }
  = f.button :submit, 'Enviar'

```

new.html.haml

```

.container.new-object-form-layout
= image_tag "Linea_Separadora.png", class: "linea-separadora"
%h3.forum-header
  Nuevo Tema
= image_tag "Linea_Separadora.png", class: "linea-separadora"

= simple_form_for @new_theme do |f|
  = f.input :name, label: "Nombre del Tema"
  = f.input :forum_id, as: :hidden, input_html: { value: @forum_id }
  = f.button :submit, 'Enviar'

```

forummer\_comment.html.haml

```

.well.comment-container-with-uploads
= simple_form_for @new_comment do |comment|
  = comment.error_notification
  = comment.input :comment_content, label: "Comentar: ", input_html: { class:
'tinymce', rows: 40, cols: 300 }
  = comment.input :thread_id, as: :hidden, input_html: { value: params[:id] }
  .row-fluid
    = comment.button :submit, "Enviar"

- if current_user.role == "Collaborator"

```

```

    = render "uploads/upload_form"

= tinymce

```

## edit.html.haml

```

.container.new-object-form-layout
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  %h3.forum-header
    Editar hilo de discusión

  = image_tag "Linea_Separadora.png", class: "linea-separadora"

  = simple_form_for @thread do |f|
    = f.error_notification
    = f.input :name, label: "Titulo de la pregunta"
    - if user_signed_in? && current_user.role == "Collaborator"
      = f.input :sticky, label: "Hilo reglamentario? (convierte al hilo en
reglamentario haciendo que aparezca al principio de la lista)"
    = f.input :question, label: "Escribe tu pregunta:", input_html: { class: 'tinymce',
rows: 40, cols: 300 }
    .row-fluid
      = f.button :submit, 'Enviar'

    - if user_signed_in? && current_user.role == User::COLLABORATOR
      = image_tag "Linea_Separadora.png", class: "linea-separadora"
      = render "uploads/upload_form"

= tinymce

```

## new.html.haml

```

.container.new-object-form-layout{style: "position:relative;"}
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  %h3.forum-header
    Nuevo Hilo de discusión
  %p.header-welcome
    Llena el siguiente formulario para crear un nuevo hilo de discusión
  = image_tag "Linea_Separadora.png", class: "linea-separadora"

  = link_to 'Regresar', :back, :class => 'pull-right normal-link'

  = simple_form_for @new_thread do |f|
    = f.error_notification
    = f.input :name, label: "Titulo de la pregunta"
    = f.input :forum_id, as: :hidden, input_html: { value: @forum.id }
    - if user_signed_in? && current_user.role == "Collaborator"
      = f.input :sticky, label: "Hilo reglamentario? (convierte al hilo en
reglamentario haciendo que aparezca al principio de la lista)"
    = f.input :question, label: "Escribe tu pregunta:", input_html: { class: 'tinymce',
rows: 40, cols: 300 }
    .row-fluid
      = f.button :submit, 'Enviar'

```

```

- if user_signed_in? && current_user.role == User::COLLABORATOR
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  = render "uploads/upload_form"

= tinymce

```

## show.html.haml

```

.forum-track.container
  = link_to "Regresar", forummer_forum_themes_path(forum: @thread.forum.id), class:
"normal-link"
.forum-table-container.container
  .thread-question
    .thread-header-question
      = @thread.name

      - if (user_signed_in? && current_user.role == User::COLLABORATOR) ||
current_user == @thread.user
        .btn-group.pull-right
          = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
            %i.icon-cog
            %span.caret
          %ul.dropdown-menu.pull-right
            %li= link_to "Borrar Hilo", forummer_thread_path(@thread), method:
"delete"
            %li= link_to "Editar", edit_forummer_thread_path(@thread)
        .thread-content.media
          .thread-coment-profile.pull-left.well
            - if @thread.user.avatar
              = link_to users_profile_path(@thread.user), class: "thumbnail" do
                = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
                = image_tag @thread.user.avatar.file_url(:thumb).to_s
            - else
              = link_to users_profile_path(@thread.user), class: "thumbnail" do
                = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
                = image_tag "Avatar.png"

            %small= @thread.user.name if !@thread.user.name.blank?
            %p= @thread.user.ocupation if !@thread.user.ocupation.blank?

          .thread-comment-content.media-body
            - if @thread.user.banned?
              %h3.banned-strip Usuario suspendido
            - else
              %p= @thread.question.html_safe

- unless @forummer_comments.empty?
  .thread-question
    .thread-header
      Respuestas

    - @forummer_comments.each do |comment|

```

```

.thread-content.media
.thread-coment-profile.pull-left.well
- if comment.user.avatar
  = link_to users_profile_path(comment.user), class: "thumbnail" do
    = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
    = image_tag comment.user.avatar.file_url(:thumb).to_s
- else
  = link_to users_profile_path(comment.user), class: "thumbnail" do
    = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
    = image_tag "Avatar.png"

= comment.user.name if !comment.user.name.blank?
%p= comment.user.ocupation if !comment.user.ocupation.blank?

.thread-comment-content.media-body
.row-fluid.comment-header
  %span.muted
  %span Fecha:
  = comment.created_at

  -if (user_signed_in? && current_user.role == User::COLLABORATOR) ||
comment.user == current_user
  .btn-group.pull-right
    = link_to "#", class: "btn dropdown-toggle",
data:{toggle:"dropdown"} do
      %i.icon-cog
      %span.caret
      %ul.dropdown-menu.pull-right
        %li= link_to "Borrar Comentario",
forummer_comment_path(comment), method: "delete"
        %li= link_to "Editar Comentario",
edit_forummer_comment_path(comment)

  .row-fluid.forummer-comment-content
  - if comment.user.banned?
    %h3.banned-strip Usuario suspendido
  - else
    %p= comment.comment_content.html_safe

= paginate @forummer_comments

- if user_signed_in?
  = render "forummer_comment"
- else
  .well
  Para comentar necesitas
  = link_to 'iniciar sesión', new_user_session_path, :class => 'normal-link'

```

\_investigator.html.haml

```
.investigator-profile-box
  .row-fluid{style: "height:145px;"}
    .span6
      .investigator-image-box
        - if investigator.avatar
          = image_tag investigator.avatar.file_url(:thumb).to_s, class: "investigator-
profile-picture"
          = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
        - else
          = image_tag "Marco Fotos Investigadores.png", class: "user-marquee"
          = image_tag "Avatar.png"
      .span6
        %h4.media-heading.blue-text.investigator-name{style: "margin-top:10px;
color:#1F93FF;"}
          = link_to "#{investigator.name.blank? ? investigator.email :
investigator.name}", users_profile_path(investigator), class: "normal-link"

        .row-fluid{style: "margin-left:15px;"}
          %h4.media-heading.investigation-black-title
            Línea de Investigación
          %p.investigator-text
            = investigator.research_field.nil? ? "Sin línea de investigación" :
investigator.research_field.research_name

        .row-fluid{style: "margin-left:15px;"}
          %h4.media-heading.investigation-black-title
            Correo electrónico
          %p.investigator-text
            = investigator.email
```

index.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      INVESTIGADORES
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid.blog-container{style: "margin-top: 5px;"}
    = render partial: "investigator", as: :investigator, collection: @investigators
    = paginate @investigators
```

\_first\_page.html.haml

```
-# Link to the "First" page
-# available local variables
```

```
-# url:          url to the first page
-# current_page: a page object for the currently displayed page
-# total_pages:  total number of pages
-# per_page:     number of items to fetch per page
-# remote:       data-remote
%span.first
  = link_to_unless current_page.first?, raw(t 'views.pagination.first'), url, :remote
=> remote, class: "normal-link"
```

#### \_gap.html.haml

```
-# Non-link tag that stands for skipped pages...
-# available local variables
-# current_page: a page object for the currently displayed page
-# total_pages:  total number of pages
-# per_page:     number of items to fetch per page
-# remote:       data-remote
%span.page.gap
  = raw(t 'views.pagination.truncate')
```

#### \_last\_page.html.haml

```
-# Link to the "Last" page
-# available local variables
-# url:          url to the last page
-# current_page: a page object for the currently displayed page
-# total_pages:  total number of pages
-# per_page:     number of items to fetch per page
-# remote:       data-remote
%span.last
  = link_to_unless current_page.last?, raw(t 'views.pagination.last'), url, {:remote
=> remote, class: "normal-link"}
```

#### \_next\_page.html.haml

```
-# Link to the "Next" page
-# available local variables
-# url:          url to the next page
-# current_page: a page object for the currently displayed page
-# total_pages:  total number of pages
-# per_page:     number of items to fetch per page
```

```

-# remote:          data-remote
%span.next
= link_to_unless current_page.last?, raw(t 'views.pagination.next'), url, :rel =>
'next', :remote => remote, class: "normal-link"

```

#### `_paginator.html.haml`

```

-# The container tag
-# available local variables
-# current_page: a page object for the currently displayed page
-# total_pages: total number of pages
-# per_page:     number of items to fetch per page
-# remote:      data-remote
-# paginator:   the paginator that renders the pagination tags inside
= paginator.render do
  .row-fluid
    .pagination.pull-right
      %ul
        %li= first_page_tag unless current_page.first?
        %li= prev_page_tag unless current_page.first?
        - each_page do |page|
          - if page.left_outer? || page.right_outer? || page.inside_window?
            %li= page_tag page
          - elsif !page.was_truncated?
            %li= gap_tag
        %li= next_page_tag unless current_page.last?
        %li= last_page_tag unless current_page.last?

```

#### `_prev_page.html.haml`

```

-# Link to the "Previous" page
-# available local variables
-# url:          url to the previous page
-# current_page: a page object for the currently displayed page
-# total_pages: total number of pages
-# per_page:     number of items to fetch per page
-# remote:      data-remote
%span.prev
= link_to_unless current_page.first?, raw(t 'views.pagination.previous'), url, :rel
=> 'prev', :remote => remote, class: "normal-link"

```



## Application.html.haml

```

!!!
%html
  %head
    %title Proyecto Energía Sustentable
    = stylesheet_link_tag "application", :media => "all"
    = javascript_include_tag "application"
    = csrf_meta_tags
  %body
    .container.full-width-background
      .row-fluid.top-container.main-container
        .logo-container
          = image_tag "Logo 2.png", class: "pull-left"
        .header-actions-sign-in-search.pull-right
          = render "static_pages/login_button"
          = render "static_pages/home_search_field"
        .menu-container.row-fluid
          %ul
            %li.menu-li
              = link_to 'INICIO', "http://www.proyectoenergiasustentable.org/"
            %li.menu-li
              = link_to 'ARTÍCULOS',
                "http://www.proyectoenergiasustentable.org/?page_id=8"
            %li.menu-li
              = link_to 'INVESTIGADORES',
                "http://www.proyectoenergiasustentable.org/?page_id=131"
            %li.menu-li
              = link_to 'VIDEOS',
                "http://www.proyectoenergiasustentable.org/?page_id=137"
            %li.menu-li
              = link_to "#" do
                = "FORO/BLOG"
              %ul.sub-menu-ul
                %li.menu-li.sub-menu-li
                  = link_to 'FOROS', forummer_main_forum_path
                %li.menu-li.sub-menu-li
                  = link_to 'BLOGS', blogger_blogs_path
            %li.menu-li
              = link_to "#" do
                = "EMPRESAS/ORG"
              %ul.sub-menu-ul
                %li.menu-li.sub-menu-li
                  = link_to 'EMPRESAS',
                    "http://www.proyectoenergiasustentable.org/?page_id=302"
                %li.menu-li.sub-menu-li

```

```

        = link_to 'ORGANIZACIONES',
"http://www.proyectoenergiasustentable.org/?page_id=304"
        %li.menu-li.sub-menu-li
        = link_to 'ASOCIACIONES',
"http://www.proyectoenergiasustentable.org/?page_id=306"
        %li.menu-li
        = link_to "CONTACTO",
"http://www.proyectoenergiasustentable.org/?page_id=55"

```

```

.full-width.yield-content
= render "static_pages/flash"
= yield

```

```

.container.full-width-background{style: "padding-top:20px; bottom:0px;}
.transparent-background
.footer.main-container
%ul.container
%li
    CICY 2013. DERECHOS RESERVADOS
%li
    = image_tag "icono-mensajes.png"
%li
    = image_tag "icono-facebook.png"
%li
    = image_tag "icono-twitter.png"
%li
    = image_tag "icono-youtube.png"

```

\_blog.html.haml

```

.thread-question
.media.thread-content
.forum-track{style: "padding:0;"}
%h3.forum-header
    = link_to blog.blog_name, blogger_blog_path(blog), class: "normal-link"
.media-body.search-content.word-wrapping-break
    = blog.blog_description.nil? ? "Foro sin descripción" : blog.blog_description

```

\_forum.html.haml

```

.thread-question
.media.thread-content

```

```

.forum-track{style: "padding:0;"}
  %h3.forum-header
    = link_to forum.name, forummer_forum_path(forum), class: "normal-link"
.media-body.search-content.word-wrapping-break
  = forum.description.nil? ? "Foro sin descripción" : forum.description

```

#### \_post.html.haml

```

.thread-question
  .media.thread-content
    .forum-track{style: "padding:0;"}
      %h3.forum-header
        = link_to post.post_title, blogger_blog_post_path(post.blog, post), class:
"normal-link"
      .media-body.search-content.word-wrapping-break
        = post.post_content.html_safe

```

#### \_thread.html.haml

```

.thread-question
  .media.thread-content
    .forum-track{style: "padding:0;"}
      %h3.forum-header
        = link_to thread.name, forummer_thread_path(thread), class: "normal-link"
      .media-body.search-content.word-wrapping-break
        = thread.question.html_safe

```

#### \_user.html.haml

```

.thread-question
  .media.thread-content
    .pull-left.thread-coment-profile
      - if user.avatar
        = link_to users_profile_path(user), class: "thumbnail" do
          = image_tag user.avatar.file_url(:thumb).to_s
      - else
        = link_to users_profile_path(user), class: "thumbnail" do
          = image_tag "Avatar.png"
    .media-body
      .row-fluid
        .span6
          - unless user.name.blank?
            %p
              Nombre:

```

```

      %small= user.name
    - unless user.occupation.blank?
      %p
      Ocupación:
      %small= user.occupation
    - unless user.nickname.blank?
      %p
      Nombre de usuario:
      %small= user.nickname
    - unless user.sex.blank?
      %p
      Sexo:
      %small= user.sex
    - unless user.phone.blank?
      %p
      Teléfono:
      %small= user.phone
    - unless user.email.blank?
      %p
      Correo:
      %small= mail_to user.email, nil ,class: "normal-link"
    .span6
    - unless user.address.blank?
      %p
      Dirección:
      %small= user.address
    - if user.blog
      %p= link_to user.blog.blog_name, blogger_blog_path(user.blog), class:
"normal-link"

    - if !user.threads.nil? && user.threads.count > 0
      %p
      Participación en los foros:
      %small
      = user.threads.count
      %span publicaciones

```

#### Search\_home.html.haml

```

.container.new-object-form-layout.profile-container
- unless @forums.empty?
  .thread-question
  .thread-header Foros
  - @forums.each do |forum|
    = render partial: "forum", locals: {forum: forum} || ""

```

```

- unless forum == @forums.last
  = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @threads.empty?
  .thread-question
  .thread-header
  Discusiones
  - @threads.each do |thread|
    = render partial: "thread", locals: {thread: thread} || ""
    - unless thread == @threads.last
      = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @blogs.empty?
  .thread-question
  .thread-header
  Blogs
  - @blogs.each do |blog|
    = render partial: "blog", locals: {blog: blog} || ""
    - unless blog == @blogs.last
      = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @posts.empty?
  .thread-question
  .thread-header
  Publicaciones
  - @posts.each do |post|
    = render partial: "post", locals: {post: post} || ""
    - unless post == @posts.last
      = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @users.empty?
  .thread-question
  .thread-header
  Usuarios
  - @users.each do |user|
    = render partial: "user", locals: {user: user} || ""
    - unless user == @users.last
      = image_tag "Linea_Separadora.png", class: "linea-separadora"

```

`_article.html.haml`

```

.container.new-object-form-layout.profile-container
- unless @forums.empty?
  .thread-question

```

```

.thread-header Foros
- @forums.each do |forum|
  = render partial: "forum", locals: {forum: forum} || ""
  - unless forum == @forums.last
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @threads.empty?
.thread-question
.thread-header
  Discusiones
- @threads.each do |thread|
  = render partial: "thread", locals: {thread: thread} || ""
  - unless thread == @threads.last
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @blogs.empty?
.thread-question
.thread-header
  Blogs
- @blogs.each do |blog|
  = render partial: "blog", locals: {blog: blog} || ""
  - unless blog == @blogs.last
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @posts.empty?
.thread-question
.thread-header
  Publicaciones
- @posts.each do |post|
  = render partial: "post", locals: {post: post} || ""
  - unless post == @posts.last
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

- unless @users.empty?
.thread-question
.thread-header
  Usuarios
- @users.each do |user|
  = render partial: "user", locals: {user: user} || ""
  - unless user == @users.last
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

```

\_flash.html.haml

```
= bootstrap_flash
```

```
_home_search._field.html.haml
```

```
= form_tag search_home_path, method: :post, class: "home-search-form pull-right" do
  = text_field_tag :q, nil , class: "pull-left home-search"
  %button{class: "search-transparent-button"}
    %i.icon-search
```

```
_login_button.html.haml
```

```
- if user_signed_in?
  .login-container.pull-right
    = link_to current_user.email, users_profile_path(current_user), :class => "pull-
right gray-btn"
    = link_to "Cerrar sesión", destroy_user_session_path, :class => "pull-right gray-
btn", :method => :delete
- else
  .login-container.pull-right
    = link_to "Iniciar sesión", new_user_session_path, :class => "pull-right gray-
btn"
```

```
_thread.html.haml
```

```
.well.media.bordered-well
  .media-body
    .row-fluid
      %h3.media-heading.blue-text.post-title
        = link_to thread.name, forummer_thread_path(thread), class: 'normal-link '

    .row-fluid
      %strong
        Publicado:
        = l( thread.created_at, format: "%d %b. %Y")

    .post-content.word-wrapping-break
      - if thread.user.banned?
        %h3.banned-strip Usuario suspendido
      - else
        %p= thread.question.html_safe
```

```
_video.html.haml
```

```
.well.media.bordered-well
```

```

.forum-track{style: "max-width:674px; padding:0;"}
- client = YouTubeIt::Client.new
  %h3.pull-left.forum-header.search-content{style: "max-width:614px; padding-left:0;
padding-right:0; "}
  = client.video_by(video.url).title
- if user_signed_in? && current_user == video.user
  .btn-group.pull-right
    = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
      %i.icon-cog
      %span.caret
    %ul.dropdown-menu.pull-right
      %li= link_to "Borrar Video", workshop_video_path(video), method: :delete,
data: { confirm: "Esta seguro" }
      %li= link_to "Editar Video", edit_workshop_video_path(video)

```

#### Home.html.haml

```

.container.new-object-form-layout.profile-container
  .blogs-container.blog-container
    - @objects.each do |object|
      - if object.class == Forummer::Thread
        = render partial: "thread", locals: {thread: object}
      - elsif object.class == Article
        = render partial: "article", locals: {article: object}
      - elsif object.class == Workshop::Video
        = render partial: "video", locals: {video: object}
      - elsif object.class == Blogger::Post
        = render partial: "blogger/posts/post", locals: {post: object}

```

#### \_upload.html.haml

```

%li.upload-thumbnail.well.only-pattern{style: "padding:5px", data: { uploadid:
upload.id }}
  .thumb
    = button_to "x", upload_path(upload), remote: true , method: :delete, data: {
confirm: "¿Desea borrar esta imagen?" }
    .image-thumbnail
      = image_tag upload.file_url(:thumb).to_s, data: {src:
upload.file_url(:thumb).to_s }
    .thumb-url
      %input.image-url-input{value: upload.file_url.to_s, type:"text"}

```



`_upload_form.html.haml`

```

.files-upload-form
  #flash

  = simple_form_for @upload, :html => {:multipart => true} do |upload|
    = upload.input :user_id, as: :hidden, :input_html => {value: current_user.id}
    %p{style: "font-weight: bold; font-size: 18px;} Subir archivos (.jpg, .png, .gif)
    = upload.file_field :file, multiple: true, name: "upload[file]", hidden: true
    = link_to "#", class: "faker-upload btn" do
      Elegir archivos
      %i.icon-upload{style: "font-size:18px; margin:5px 0 0 5px;}

  %script#template-upload{type:"text/x-tmpl"}
    .upload
      {%=o.name%}
    .progress
      .bar{style:"width: 0%"}

%ul.files-list.row-fluid
  = render partial: "uploads/upload", collection: current_user.uploads, as: :upload

```

`Créate.js.erb`

```

<% if @upload.new_record? %>
  alert("Error al subir: <%= j @upload.errors.full_messages.join(', ').html_safe %>");
<% else %>
  $(".files-list").prepend("<%= j render(@upload) %>");
<% end %>

```

`Destroy.js.erb`

```

<% if @upload.destroy %>
  $(".li[data-uploadid='<%= @upload.id %>']").fadeOut(300, function() {
$(this).remove() } )
<% else %>
  alert("No se pudo borrar la imagen por las siguientes razones: <%= j
@upload.errors.full_messages.join(', ').html_safe %>");
<% end %>

```

`New.html.erb`

```

<h2>Resend confirmation instructions</h2>

<%= form_for(resource, :as => resource_name, :url => confirmation_path(resource_name),
:html => { :method => :post }) do |f| %>
  <%= devise_error_messages! %>

  <div><%= f.label :email %><br />
  <%= f.email_field :email, :autofocus => true %></div>

  <div><%= f.submit "Reenviar las instrucciones para confirmación" %></div>
<% end %>

<%= render "devise/shared/links" %>

```

### Confirmation\_instructions.html.erb

```

<p>Welcome <%= @email %>!</p>

<p>You can confirm your account email through the link below:</p>

<p><%= link_to 'Confirm my account', confirmation_url(@resource, :confirmation_token
=> @resource.confirmation_token) %></p>

```

### Reset\_password\_instructions.html.erb

```

.container.registration-content.new-object-form-layout
%p
  Hola
  = @resource.email
  \!

  %p Esta es una solicitud de recuperación de contraseña. Por favor has click abajo si
solicitaste este correo, de lo contrario ignoralo.

  %p
    = link_to 'Cambiar mi contraseña', edit_password_url(@resource,
:reset_password_token => @resource.reset_password_token)

  %p Tu contraseña no cambiara hasta que des click al enlace de arriba.

```

### Unlock\_instructions.html.erb

```
<p>Hello <%= @resource.email %>!</p>
```

```
<p>Your account has been locked due to an excessive number of unsuccessful sign in attempts.</p>
```

```
<p>Click the link below to unlock your account:</p>
```

```
<p><%= link_to 'Unlock my account', unlock_url(@resource, :unlock_token =>
@resource.unlock_token) %></p>
```

### Edit.html.haml

```
.container.registration-content.new-object-form-layout
  .row-fluid.container
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header CÁMBIA TU CONTRASEÑA
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

    = simple_form_for(resource, :as => resource_name, :url =>
password_path(resource_name), :html => { :method => :put }) do |f|
      = f.input :reset_password_token, as: :hidden

      .input-rower.row-fluid
        = f.input :password, :autofocus => true, label: "Contraseña"

      .input-rower.row-fluid
        = f.input :password_confirmation, label: "Confirmar Contraseña"

      .input-rower.row-fluid.submit-registration{style: "width:466px;}
        = f.submit "Enviar", :class => "pull-right submit-btn"
```

### New.html.haml

```
.container.forgot-password-content.new-object-form-layout
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  %h2.forum-header ¿Olvidaste tu contraseña?
  = image_tag "Linea_Separadora.png", class: "linea-separadora"
  .row-fluid.container
    .span7

    = simple_form_for(resource, :as => resource_name, :url =>
password_path(resource_name), :html => { :method => :post }) do |f|
      .input-rower.row-fluid
        = f.input :email, label: "Correo", :autofocus => true
```

```

    = f.submit "Enviar las instrucciones", :class => "pull-right submit-btn"

    .span5
    = render "users/shared/links"

```

## Edit.html.haml

```

.container.registration-content.new-object-form-layout.profile-container
  .row-fluid.container
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      CUENTA
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid
    .span4
      .well.profile-image
        .avatar
          - if resource.avatar
            = image_tag resource.avatar.file_url(:avatarize).to_s
          - else
            = image_tag "Avatar.png", class: "avatar-image"
          = render "avatars/avatar_upload"

    .span8
      = simple_form_for(resource, :as => resource_name, :url =>
        users_profile_path(resource), :html => { :method => :put }) do |f|
        .row-fluid
          .span6
            %h3 Datos Personales

            = f.input :sex, as: :radio_buttons, collection: User::SEX, label: "Sexo"

            = f.input :birthday, label:"Cumpleaños", :input_html => {:class =>
              "datepicker"}

            = f.input :occupation, as: :select, collection: User::OCUPATIONS, label:
              "Ocupación"

            = f.input :interests, label: "Intereses"

            - if devise_mapping.confirmable? && resource.pending_reconfirmation?
              .input-rower.row-fluid

```

```

    Actualmente esperando confirmación para:
    = resource.unconfirmed_email

- if current_user.provider.nil?
  .input-rower.row-fluid

    = f.input :password, :autocomplete => "off", label: "Nueva
Contraseña"
    %i (Deja en blanco si no deseas cambiar tu contraseña)
    %br

    = f.input :password_confirmation, label: "Confirmar contraseña"

  .input-rower.row-fluid

    = f.input :current_password, label: "Contraseña actual"
    %i (Necesitamos tu contraseña actual para verificar tu identidad)
    %br

  .span6
    %h3 Datos de contacto
    = f.input :nickname, label: "Nombre de Usuario"
    = f.input :email, :autofocus => true, label: "Correo"
    = f.input :phone, label: "Teléfono (Sin guiones)"
    = f.input :address, label: "Dirección"
  .input-rower.row-fluid.send-update-profile
    = f.submit "Actualizar", :class => "pull-right submit-btn"

= image_tag "Linea_Separadora.png", class: "linea-separadora"

  .input-rower.row-fluid.cancel-account
    ¿Insatisfecho?
    = link_to "Cancelar mi cuenta", registration_path(resource_name), :data => {
:confirm => "¿Estas seguro?" }, :method => :delete , :class => "submit-btn"

```

New.html.haml

```

.container.registration-content.new-object-form-layout
  .row-fluid.container

    %div{class: "span#{ devise_error_messages!.blank? ? '6' : '8' }}"
      %h2.forum-header Registrarse

```

```

    = simple_form_for(resource, :as => resource_name, :url =>
registration_path(resource_name)) do |f|
      .input-rower.row-fluid
        = f.input :email, label: "Correo", :autofocus => true
      .input-rower.row-fluid
        = f.input :password, label: "Contraseña"
      .input-rower.row-fluid
        = f.input :password_confirmation, label: "Confirmar contraseña"
      .input-rower.row-fluid.submit-registration{style: "width:466px;"}
        = f.submit "Enviar", :class => "pull-right submit-btn"
        = raw recaptcha_tags

%div{class: "span#{ devise_error_messages!.blank? ? '6' : '4'}"}
  = render "users/shared/links"

```

#### New.html.haml

```

.container.session-content.new-object-form-layout
  .row-fluid.container
    .span6
      %h2.forum-header Iniciar sesión
      = simple_form_for(resource, :as => resource_name, :url =>
session_path(resource_name)) do |f|
        .input-rower.row-fluid
          = f.input :email
        .input-rower.row-fluid
          = f.input :password, label: "Contraseña"
        - if devise_mapping.rememberable?
          .input-rower.remember.row-fluid
            = f.input :remember_me, as: :boolean, label: "Recordarme"
        .input-rower.row-fluid.submit-session= f.submit "Enviar", class: "pull-right
submit-btn"
    .span6
      = render "users/shared/links"

```

`_links.html.haml`

```

- if controller_name != 'sessions'
  .input-rower.row-fluid
    = link_to "Iniciar Sesión", new_session_path(resource_name), class: "normal-link"
  %br/
- if devise_mapping.registerable? && controller_name != 'registrations'
  .input-rower.row-fluid
    = link_to "Registrarse", new_registration_path(resource_name), class: "normal-link"
  %br/
- if devise_mapping.recoverable? && controller_name != 'passwords'
  .input-rower.row-fluid
    = link_to "¿Olvidaste tu contraseña?", new_password_path(resource_name), class: "normal-link"
  %br/
- if devise_mapping.confirmable? && controller_name != 'confirmations'
  .input-rower.row-fluid
    = link_to "¿No recibiste la información para confirmar",
new_confirmation_path(resource_name), class: "normal-link"
  %br/
- if devise_mapping.lockable? && resource_class.unlock_strategy_enabled?(:email) &&
controller_name != 'unlocks'
  .input-rower.row-fluid
    = link_to "¿No recibiste la informacion para desbloquear tu cuenta?",
new_unlock_path(resource_name), class: "normal-link"
  %br/
- if devise_mapping.omniauthable?
  .input-rower.row-fluid
    %p Iniciar sesión con:
    - resource_class.omniauth_providers.each do |provider|
      = link_to omniauth_authorize_path(resource_name, provider), :class => "pull-left
social-media-sign-in" do
        - if provider == :facebook
          = image_tag "icono-facebook.png"
        - elsif provider == :twitter
          = image_tag "icono-twitter.png"
        - elsif provider == :google_oauth2
          = image_tag "icono-gplus.png"

```

`New.html.erb`

```
<h2>Resend unlock instructions</h2>
```

```

<%= form_for(resource, :as => resource_name, :url => unlock_path(resource_name), :html
=> { :method => :post }) do |f| %>
  <%= devise_error_messages! %>

  <div><%= f.label :email %><br />
  <%= f.email_field :email, :autofocus => true %></div>

  <div><%= f.submit "Resend unlock instructions" %></div>
<% end %>

<%= render "devise/shared/links" %>

```

### Show.html.haml

```

.container.registration-content.new-object-form-layout.profile-container
  .row-fluid.container
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      CUENTA
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

    - if @user.banned?
      %h3.banned-strip Usuario suspendido

  .row-fluid
    .span4
      .well.profile-image
        .avatar
          - if @user.avatar
            = image_tag @user.avatar.file_url(:avatarize).to_s
          - else
            = image_tag "Avatar.png", class: "avatar-image"
          = render "avatars/avatar_upload"
        %h4.media-heading.blue-text{style: "margin-top:10px;"}
          = @user.name.blank? ? @user.email : @user.name
    .span8
      .row-fluid
        .span6
          %h3 Datos Personales

          .input-rower.row-fluid.user-data
            %p.user-data-title Sexo
            - if !@user.sex.blank?
              %p= @user.sex
            - else

```



```

        %p.muted Sin información
    .input-rower.row-fluid.user-data
        %p.user-data-title Cumpleaños
        - if !@user.birthday.blank?
            %p= @user.birthday
        - else
            %p.muted Sin información
    .input-rower.row-fluid.user-data
        %p.user-data-title Ocupación
        - if !@user.ocupation.blank?
            %p= @user.ocupation
        - else
            %p.muted Sin información
    .input-rower.row-fluid.user-data
        %p.user-data-title Intereses
        - if !@user.interests.blank?
            %p= @user.interests
        - else
            %p.muted Sin información

    .span6
        %h3 Datos de contacto

        .input-rower.row-fluid.user-data
            %p.user-data-title Nombre de usuario
            - if !@user.nickname.blank?
                %p= @user.nickname
            - else
                %p.muted Sin información
        .input-rower.row-fluid.user-data
            %p.user-data-title Correo
            - if !@user.email.blank?
                %p= @user.email
            - else
                %p.muted Sin información
        .input-rower.row-fluid.user-data
            %p.user-data-title Teléfono
            - if !@user.phone.blank?
                %p= @user.phone
            - else
                %p.muted Sin información
        .input-rower.row-fluid.user-data
            %p.user-data-title Dirección
            - if !@user.address.blank?
                %p= @user.address
            - else

```

```

        %p.muted Sin información

    .input-rower.row-fluid.send-update-profile
      - if user_signed_in? && current_user.role == User::COLLABORATOR && current_user
        != @user
          - if @user.banned?
            = link_to "Quitar Suspensión", unban_users_profile_path(@user), class:
"submit-btn pull-right", method: :post
          - else
            = link_to "Suspende usuario", ban_users_profile_path(@user), class:
"submit-btn pull-right", method: :post
          - if user_signed_in? && @user == current_user
            = link_to "Editar", edit_user_registration_path, class: "submit-btn pull-
right"
          - if !@user.blog && @user.role == User::COLLABORATOR
            = link_to "Hacer mi blog", new_blogger_blog_path, class: "submit-btn pull-
right"
          - if @user.blog
            = link_to "Ir al blog", blogger_blog_path(@user.blog), class: "submit-btn
pull-right"
          - if user_signed_in? && current_user == @user
            = link_to "Editar blog", edit_blogger_blog_path(@user.blog), class: "submit-
btn pull-right"
            = link_to "Borrar blog", blogger_blog_path(@user.blog), method: "delete",
class: "pull-right red-submit-btn"

_video.html.haml

.well.media.bordered-well
  .forum-track{style: "max-width:674px; padding:0;}
    %h3.pull-left.forum-header.search-content{style: "max-width:614px; padding-left:0;
padding-right:0; "}
      = @client.video_by(video.url).title
      - if user_signed_in? && current_user == video.user
        .btn-group.pull-right
          = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
            %i.icon-cog
            %span.caret
            %ul.dropdown-menu.pull-right
              %li= link_to "Borrar Video", workshop_video_path(video), method: :delete,
data: { confirm: "Esta seguro" }
              %li= link_to "Editar Video", edit_workshop_video_path(video)

```

```
%iframe#ytplayer{style:"margin-top:10px;",":allowscriptaccess => "always" , :name =>
"window", :src => "http://www.youtube.com/embed/#{video.unique_id.to_s}", width:
"674", height: "430", :frameborder => "0", :allowfullscreen => "allowfullscreen"}
```

### Edit.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      EDITAR VIDEO
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid{style: "margin-top:5px;"}
    = link_to 'Regresar', :back, :class => 'normal-link pull-right'
    = simple_form_for @video do |f|
      = f.input :url, label: "Enlace (debe contener el prefijo http:// y ser de
youtube)"
      .row-fluid
        = f.button :submit, 'Enviar', class: "submit-btn"
```

### Index.html.haml

```
.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      VIDEOS
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
  .row-fluid{style: "margin-top: 5px; height:35px;"}
    - if user_signed_in? && current_user.role == User::COLLABORATOR
      .btn-group.pull-right
        = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
          %i.icon-cog
          %span.caret
        %ul.dropdown-menu.pull-right
          %li= link_to "Nuevo Video", new_workshop_video_path

  .row-fluid.blog-container{style: "margin-top: 5px;"}
    .span9
      = render @videos

      = paginate @videos
```

```

    .span3
      .well.bordered-well
        %h4.blue-text
          Archivo
        %ul.archive-ul

      - Workshop::Video.select("*").order("created_at DESC").group_by { |m|
m.created_at.beginning_of_month}.each do |date , videos|
        .accordion-group
          %li.closedli.accordion-heading
            = link_to l(date, format:"%B %Y") , "#Collapse#{date.strftime("%B")}",
class: "accordion-toggle black-bold-link"
          .accordion{id: "Collapse#{date.strftime("%B")}" }
            %ul.post-list-archive
              - videos.each do |innervideo|
                %li.closedli.post-li.word-wrapping-break
                  = link_to @client.video_by(innervideo.url).title,
workshop_video_path(innervideo), class: "black-bold-link"
New.html.haml

```

```

.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      NUEVO VIDEO
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

  .row-fluid{style: "margin-top:5px;"}
    = link_to 'Regresar', :back, :class => 'normal-link pull-right'
    = simple_form_for @video do |f|
      = f.input :url, label: "Enlace (debe contener el prefijo http:// y ser de
youtube)"
    .row-fluid
      = f.button :submit, 'Enviar', class: "submit-btn"

```

Show.html.haml

```

.container.new-object-form-layout.profile-container
  .row-fluid
    = image_tag "Linea_Separadora.png", class: "linea-separadora"
    %h2.forum-header
      VIDEO
    = image_tag "Linea_Separadora.png", class: "linea-separadora"

```

```

.row-fluid.blog-container{style: "margin-top: 5px;"}
  .media
    .forum-track{style: "max-width:960px; padding:0;"}
      %h3.pull-left.forum-header.search-content{style: "max-width:840px; padding-
left:0; padding-right:0; "}
        = @client.video_by(@video.url).title
      - if user_signed_in? && current_user == @video.user
        .btn-group.pull-right
          = link_to "#", class: "btn dropdown-toggle", data:{toggle:"dropdown"} do
            %i.icon-cog
            %span.caret
          %ul.dropdown-menu.pull-right
            %li= link_to "Borrar Video", workshop_video_path(@video), method:
:delete, data: { confirm: "¿Esta seguro?" }
            %li= link_to "Editar Video", edit_workshop_video_path(@video)

    .row-fluid{style: "margin-top:5px;"}
      = link_to 'Regresar', workshop_videos_path, :class => 'normal-link pull-right'

    .container{style: "width:920px;"}
      %iframe#ytplayer{style:"margin-top:10px;",:allowscriptaccess => "always" ,
:name => "window", :src => "http://www.youtube.com/embed/#{@video.unique_id.to_s}",
width: "920", height: "586", :frameborder => "0", :allowfullscreen =>
"allowfullscreen"}

```

## Anexo D Código Hojas de Estilo

active\_admin.css.scss

```
// SASS variable overrides must be declared before loading up Active Admin's styles.
//
// To view the variables that Active Admin provides, take a look at
// `app/assets/stylesheets/active_admin/mixins/_variables.css.scss` in the
// Active Admin source.
//
// For example, to change the sidebar width:
// $sidebar-width: 242px;

// Active Admin's got SASS!
@import "active_admin/mixins";
@import "active_admin/base";

// Overriding any non-variable SASS must be done after the fact.
// For example, to change the default status-tag color:
//
// body.active_admin {
//   .status_tag { background: #6090DB; }
// }
//
// Notice that Active Admin CSS rules are nested within a
// 'body.active_admin' selector to prevent conflicts with
// other pages in the app. It is best to wrap your changes in a
// namespace so they are properly recognized. You have options
// if you e.g. want different styles for different namespaces:
```

```
//  
  
// .active_admin      applies to any Active Admin namespace  
  
// .admin_namespace  applies to the admin namespace (eg: /admin)  
  
// .other_namespace  applies to a custom namespace named other (eg: /other)
```

## application.css

```
/*  
  
 * This is a manifest file that'll be compiled into application.css, which will  
 * include all the files  
  
 * listed below.  
  
 *  
 * Any CSS and SCSS file within this directory, lib/assets/stylesheets,  
 * vendor/assets/stylesheets, or vendor/assets/stylesheets of plugins, if any, can be referenced here using a  
 * relative path.  
  
 *  
 * You're free to add application-wide styles to this file and they'll appear at the  
 * top of the  
 * compiled file, but it's generally better to create a new file per style scope.  
  
 *  
 *= require_self  
  
 *= require token-input-facebook  
  
 *= require jquery.ui.datepicker  
  
 *= require_tree .  
  
*/
```

**articles.css.scss**

```
// Place all the styles related to the articles controller here.
// They will automatically be included in application.css.
// You can use Sass (SCSS) here: http://sass-lang.com/

#new_article{
  #article_title{
    width:300px;
  }
  .submit-btn{
    margin-right:0;
  }
}

.edit_article{
  .submit-btn{
    margin-right:0;
  }
}
```

**avatars.css.scss**

```
// Place all the styles related to the avatars controller here.
// They will automatically be included in application.css.
// You can use Sass (SCSS) here: http://sass-lang.com/

.avatar-upload-faker{
  display:block;
```



```
position:absolute;
top:225px;
left:17px;
width:221px;
height:25px;
background-color:rgba(78,78,78,0.4) !important;
line-height:25px !important;
color:rgba(255,0,0,0) !important;
&:hover{
    color:white !important;
    font-size:16px;
    font-weight:bold;
}
&:focus{
    color:#589AD9 !important;
    font-size:20px;
    font-weight:bold;
    text-shadow: 1px 1px white !important;
}
}
#new_avatar{
    margin:0;
}
.user-marquee{
    position:absolute;
}
```

bootstrap\_and\_overrides.css

```
/*  
  
=require twitter-bootstrap-static/bootstrap  
  
Use Font Awesome icons (default)  
  
To use Glyphicons sprites instead of Font Awesome, replace with "require twitter-  
bootstrap-static/sprites"  
  
=require twitter-bootstrap-static/fontawesome  
  
*/
```

#### companies.css.scss

```
// Place all the styles related to the companies controller here.  
// They will automatically be included in application.css.  
// You can use Sass (SCSS) here: http://sass-lang.com/
```

```
#company_company_logo, #company_name{  
  width:400px;  
}
```

```
.company-logo-style{  
  width: 300px;  
  height:300px;  
}
```

#### contacts.css.scss

```
// Place all the styles related to the contacts controller here.  
// They will automatically be included in application.css.  
// You can use Sass (SCSS) here: http://sass-lang.com/  
  
.contact-cicy-title{  
  font-weight:bold;  
  padding:20px 30px 20px 20px;  
}  
  
.contact-cicy-address{  
  padding:20px 30px 20px 20px;  
}  
  
#contact_name, #contact_email, #contact_occupation, #contact_contact_text{  
  width: 650px;  
}  
  
#contact_contact_text{  
  height:200px;  
}
```

investigators.css.scss

```
// Place all the styles related to the investigators controller here.  
// They will automatically be included in application.css.  
// You can use Sass (SCSS) here: http://sass-lang.com/  
  
.investigator-profile-box{  
  font-family: 'AvenirNextLTPro-Regular';  
  position:relative;  
  width:300px;
```

```
height:300px;

margin-left:8px;

margin-right:8px;

display:inline-block;

background-position: center;

background: url(cuadro_investigador.png) no-repeat;

}

.investigator-profile-picture{

    position:absolute;

}

.investigator-image-box{

    position:relative;

    margin-left:15px;

    margin-top:15px;

    img{

        width:120px;

        height:120px;

    }

}

.investigation-black-title{

    color:black;

    font-family: 'AvenirNextLTPro-Regular';

    font-weight:bold;

}

.investigator-name{

    padding-top:10px;

    max-width:120px;

}
```

```
.investigator-text{
  color:#1F93FF;
}
```

searches.css.scss

```
// Place all the styles related to the searches controller here.
// They will automatically be included in application.css.
// You can use Sass (SCSS) here: http://sass-lang.com/
```

```
.search-transparent-button{
  background-image:none;
  background-color: transparent;
  position:absolute;
  margin-left:-240px;
  -o-margin-left:0;
  margin-top:4px;
  border:none;
}
.home-search{
  padding-left:28px !important;
}
.home-search-form{
  width:240px !important;
  margin:23px 2px 0 0;
}
.search-content{
```

```
max-height:200px;

overflow:hidden;

font-family: 'AvenirNextLTPro-Regular';

}
```

### social-share-button.scss

```
/*
```

*Creating a sprite from following images:*

```
app/assets/images/sprites/social-share-button/baidu.png (16x16)
app/assets/images/sprites/social-share-button/delicious.png (16x16)
app/assets/images/sprites/social-share-button/douban.png (16x16)
app/assets/images/sprites/social-share-button/email.png (16x16)
app/assets/images/sprites/social-share-button/facebook.png (16x16)
app/assets/images/sprites/social-share-button/flickr.png (16x16)
app/assets/images/sprites/social-share-button/google_bookmark.png (16x16)
app/assets/images/sprites/social-share-button/google_plus.png (16x16)
app/assets/images/sprites/social-share-button/kaixin001.png (16x16)
app/assets/images/sprites/social-share-button/plurk.png (16x16)
app/assets/images/sprites/social-share-button/qq.png (16x16)
app/assets/images/sprites/social-share-button/renren.png (16x16)
app/assets/images/sprites/social-share-button/tqq.png (16x16)
app/assets/images/sprites/social-share-button/tumblr.png (16x16)
app/assets/images/sprites/social-share-button/twitter.png (16x16)
app/assets/images/sprites/social-share-button/weibo.png (16x16)
```

*Output files:*

*app/assets/images/sprites/social-share-button.png*

*app/assets/stylesheets/social-share-button.scss*

*Output size:*

*64x64*

*\*/*

```
.social-share-button{
```

```
  width: 200px;
```

```
  line-height: 36px;
```

```
  height: 36px;
```

```
  a{
```

```
    margin: 3px 0 0 4px;
```

```
    width: 24px;
```

```
    height: 24px;
```

```
  }
```

```
}
```

```
.social-share-button-baidu { display:inline-block; width: 16px; height: 16px;  
background: url(image-path('sprites/social-share-button.png')) 0px 0px no-repeat }
```

```
.social-share-button-delicious { display:inline-block; width: 16px; height: 16px;  
background: url(image-path('sprites/social-share-button.png')) -16px 0px no-repeat }
```

```
.social-share-button-douban { display:inline-block; width: 16px; height: 16px;  
background: url(image-path('sprites/social-share-button.png')) 0px -16px no-repeat }
```

```
.social-share-button-email { display:inline-block; width: 16px; height: 16px;  
background: url(image-path('sprites/social-share-button.png')) -16px -16px no-repeat }
```

```
.social-share-button-facebook { display:inline-block; background: url('facebook-icon-  
blue.png') no-repeat }
```

```
.social-share-button-flickr { display:inline-block; width: 16px; height: 16px;  
background: url(image-path('sprites/social-share-button.png')) -32px -16px no-repeat }
```

```
.social-share-button-google_bookmark { display:inline-block; width: 16px; height:
16px; background: url(image-path('sprites/social-share-button.png')) 0px -32px no-
repeat }

.social-share-button-google_plus { display:inline-block; background: url('gplus-icon-
blue.png') no-repeat }

.social-share-button-kaixin001 { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) -32px -32px no-repeat }

.social-share-button-plurk { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) -48px 0px no-repeat }

.social-share-button-qq { display:inline-block; width: 16px; height: 16px; background:
url(image-path('sprites/social-share-button.png')) -48px -16px no-repeat }

.social-share-button-renren { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) -48px -32px no-repeat }

.social-share-button-tqq { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) 0px -48px no-repeat }

.social-share-button-tumblr { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) -16px -48px no-repeat }

.social-share-button-twitter { display:inline-block; background: url('twitter-icon-
blue2.png') no-repeat }

.social-share-button-weibo { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) -48px -48px no-repeat }

@media only screen and (-webkit-device-pixel-ratio: 2) {

    .social-share-button-baidu { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) 0px 0px no-repeat;
background-size:64px 64px; }

    .social-share-button-delicious { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -16px 0px no-repeat;
background-size:64px 64px; }

    .social-share-button-douban { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) 0px -16px no-repeat;
background-size:64px 64px;}

    .social-share-button-email { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -16px -16px no-
repeat; background-size:64px 64px;}

    .social-share-button-facebook { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -32px 0px no-repeat;
background-size:64px 64px;}
```



```
.social-share-button-flickr { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -32px -16px no-
repeat; background-size:64px 64px;}

.social-share-button-google_bookmark { display:inline-block; width: 16px; height:
16px; background: url(image-path('sprites/social-share-button@2x.png')) 0px -32px no-
repeat; background-size:64px 64px;}

.social-share-button-google_plus { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -16px -32px no-
repeat; background-size:64px 64px;}

.social-share-button-kaixin001 { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -32px -32px no-
repeat; background-size:64px 64px;}

.social-share-button-plurk { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button.png')) -48px 0px no-repeat;
background-size:64px 64px; }

.social-share-button-qq { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -48px 16px no-
repeat; background-size:64px 64px;}

.social-share-button-renren { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -48px -32px no-
repeat; background-size:64px 64px;}

.social-share-button-tqq { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) 0px -48 no-repeat;
background-size:64px 64px;}

.social-share-button-tumblr { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -16px -48px no-
repeat; background-size:64px 64px;}

.social-share-button-twitter { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -32px -48px no-
repeat; background-size:64px 64px;}

.social-share-button-weibo { display:inline-block; width: 16px; height: 16px;
background: url(image-path('sprites/social-share-button@2x.png')) -48px -48px no-
repeat; background-size:64px 64px;}

}
```

static\_pages.css.scss

```
// Place all the styles related to the static_pages controller here.
// They will automatically be included in application.css.
// You can use Sass (SCSS) here: http://sass-lang.com/

@font-face {
  font-family: 'AvenirNextLTPro-Bold';
  src:font-url('AvenirNextLTPro-Bold.eot');
  src:font-url('AvenirNextLTPro-Bold.eot?#iefix') format('embedded-opentype'),
  font-url('AvenirNextLTPro-Bold.svg#AvenirNextLTPro-Bold') format('svg'),
  font-url('AvenirNextLTPro-Bold.woff') format('woff'),
  font-url('AvenirNextLTPro-Bold.ttf') format('truetype');
  font-weight: normal;
  font-style: normal;
}

@font-face {
  font-family: 'AvenirNextLTPro-Demi';
  src:font-url('AvenirNextLTPro-Demi.eot');
  src:font-url('AvenirNextLTPro-Demi.eot?#iefix') format('embedded-opentype'),
  font-url('AvenirNextLTPro-Demi.svg#AvenirNextLTPro-Demi') format('svg'),
  font-url('AvenirNextLTPro-Demi.woff') format('woff'),
  font-url('AvenirNextLTPro-Demi.ttf') format('truetype');
  font-weight: normal;
  font-style: normal;
}

@font-face {
  font-family: 'AvenirNextLTPro-Regular';
  src:font-url('AvenirNextLTPro-Regular.eot');
```

```
src:font-url('AvenirNextLTPro-Regular.eot?#iefix') format('embedded-opentype'),
  font-url('AvenirNextLTPro-Regular.svg#AvenirNextLTPro-Regular') format('svg'),
  font-url('AvenirNextLTPro-Regular.woff') format('woff'),
  font-url('AvenirNextLTPro-Regular.ttf') format('truetype');
font-weight: normal;
font-style: normal;
}
```

```
html, body{
  background-color: #E2E2E2;
  height: auto;
  min-height: 100% !important;
}
```

```
//Clases Globales
.control-label{
  font-family: 'AvenirNextLTPro-Regular';
  color: rgba(77,77,77,1);
  font-weight:bold;
}
.home-search-form{
  width:285px;
  .home-search{
  }
}
```

```
}  
  
.red-submit-btn{  
  background-color:#a43c4b;  
  background-image: none;  
  font-family: 'AvenirNextLTPro-Demi';  
  color:white;  
  padding:7px 17px;  
  border-radius: 0 !important;  
  border:0;  
  font-size: 14px;  
  margin-right:42px;  
  &:hover{  
    text-decoration:none;  
    color:white;  
    background-color:#c1707b;  
  }  
  &:focus{  
    text-decoration:none;  
    color:white;  
    background-color:#c1707b;  
  }  
}  
  
.word-wrapping-break{  
  word-wrap: break-word !important;  
}  
  
.blue-text{
```

```
    color: #589AD9;
}

.avatar-thumb{
    width:100px;
    height:100px;
    img{
        width:100px;
        height:100px;
    }
}

.banned-strip{
    border-top: 2px solid #EED3D7;
    border-bottom: 2px solid #EED3D7;
    background-color: #F2DEDE;
    color: red;
    padding: 10px 5px;
}

.new-object-form-layout{
    width:960px;
    padding:0px 15px 0 15px;
}

.pattern-background{
    width:960px;
    padding:15px;
}
```

```

.only-pattern{
    background-image: url("trama-superior.png");
}

input[type="text"], input[type="url"], input[type="email"], input[type="password"],
#contact_contact_text{

    border:1px solid black;

    background: #ebebeb; /* Old browsers */

    /* IE9 SVG, needs conditional override of 'filter' to 'none' */

    background:
url(data:image/svg+xml;base64,PD94bWwgdmlVyc2lvcj0iMS4wIiA/Pgo8c3ZnIHhtbG5zPSJodHRwOi8v
d3d3LnczLm9yZy8yMDAwL3N2ZyIgd2lkdGg9IjEwMCUiIGhlaWdodD0iMTAwJSIgdmlld0JveD0iMCAwIDEgMS
IgcHJlc2VydmlVBC3BLY3RSYXRpbz0ibm9uZSI+CiAgPGxpbmVhckdyYWRpZW50IGlkPSJncmFkLXVjZ2ctZ2Vu
ZXJhdGVkIiBncmFkaWVudFVuaXRzPSJ1c2VyU3BhY2Vpb1VzZSIgeDE9IjAlIiB5MT0iMCUiIHgyPSIwJSIgeT
I9IjEwMCUiPgogICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3RvcC1jb2xvcj0iI2ViZWJlYiIgc3RvcC1vcGFjaXR5
PSIxIi8+CiAgICA8c3RvcCBvZmZzZXQ9IjE5JSIgc3RvcC1jb2xvcj0iI2ViZWJlYiIgc3RvcC1vcGFjaXR5PS
IxiI8+CiAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0b3AtY29sb3I9IiNmOWY5ZjkiIHN0b3Atb3BhY2l0eT0i
MSIvPgogIDwvbGluZWFyR3JhZG11bnQ+CiAgPHJlY3QgeD0iMCIgeT0iMCIgd2lkdGg9IjEiIGhlaWdodD0iMS
IgzmlsbD0idXJsKCNcmFkLXVjZ2ctZ2VuZXJhdGVkKSIGLz4KPC9zdmc+);

    background: -moz-linear-gradient(top, #ebebeb 0%, #ebebeb 19%, #f9f9f9 100%); /*
FF3.6+ */

    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#ebebeb),
color-stop(19%,#ebebeb), color-stop(100%,#f9f9f9)); /* Chrome,Safari4+ */

    background: -webkit-linear-gradient(top, #ebebeb 0%,#ebebeb 19%,#f9f9f9 100%); /*
Chrome10+,Safari5.1+ */

    background: -o-linear-gradient(top, #ebebeb 0%,#ebebeb 19%,#f9f9f9 100%); /* Opera
11.10+ */

    background: -ms-linear-gradient(top, #ebebeb 0%,#ebebeb 19%,#f9f9f9 100%); /* IE10+
*/

    background: linear-gradient(to bottom, #ebebeb 0%,#ebebeb 19%,#f9f9f9 100%); /* W3C
*/

    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#ebebeb',
endColorstr='#f9f9f9',GradientType=0 ); /* IE6-8 */

}

.normal-link{

    color:#1F93FF;
}

```

```
text-decoration:none;

font-family: 'AvenirNextLTPro-Regular';

font-weight:bold;

&:hover{

    text-decoration:none;

    color:#49a7ff;

}

&:focus{

    text-decoration:none;

    color:#49a7ff;

}

}
```

```
.black-link{

color:black;

text-decoration:none;

font-family: 'AvenirNextLTPro-Regular';

font-weight:normal;

&:hover{

    text-decoration:none;

    color:gray;

}

&:focus{

    text-decoration:none;

    color:gray;

}

}
```

```
.black-bold-link{
  color:black;
  text-decoration:none;
  font-family: 'AvenirNextLTPro-Regular';
  font-weight:bold;
  &:hover{
    text-decoration:none;
    color:gray;
  }
  &:focus{
    text-decoration:none;
    color:gray;
  }
}
```

```
.black-bold-dotted-link{
  color:black;
  text-decoration:none;
  font-family: 'AvenirNextLTPro-Regular';
  font-weight:bold;
  border-bottom:1px dotted black;
  &:hover{
    text-decoration:none;
    color:gray;
  }
  &:focus{
    text-decoration:none;
    color:gray;
  }
}
```



```
    }  
  }  
  
  .full-width{  
    width:100%;  
  }  
  
  .full-width-background{  
    width:960px;  
    background-image: url("trama-superior.png");  
  }  
  
  .yield-content{  
    min-height:100%;  
    margin: 0 auto;  
  }  
  
  .transparent-background{  
    background-image:url("transpBlack25.png");  
    background-repeat:repeat;  
  }  
  
  .main-container{  
    width:960px;  
  }  
  
  .no-link-styles{  
    text-decoration:none;  
    &:hover{  
      text-decoration:none;  
    }  
  }  
}
```

```
background-color:#6abcfd;

color:white;

-moz-opacity:0.5;

-webkit-opacity:0.5;

}

}
```

```
.gray-btn{

padding:6px 10px;

background-color: #4E4E4E;

color: white;

text-decoration:none;

border-radius:5px;

margin-left:5px;

&:hover{

text-decoration:none;

color:white;

}

&:focus{

text-decoration:none;

color:white;

}

}
```

```
.login-container{

width:400px;

}

}
```

```
//clases aplicadas en /views/layouts/application.html.haml

//Inicio de clases para el TOP (Logo, menu, barra busqueda y contador de visitas)

.top-container{ //Esta clase es la principal del top, dentro de ella van todos los
elementos

  padding: 0 0 15px 0;

  .logo-container{ //En esta clase se establecen las características del contenedor
que alojara el logo.

    padding: 24px 15px 0 37px;

    height:110px;

    .header-actions-sign-in-search{

      width:360px;

      height:100px;

    }

  }

}

.menu-container ul ul{

  display:none;

  position: absolute;

  top: 100%;

  width: 137px;

}

.menu-container ul ul li {

  float: none !important;

}

.menu-container ul li:hover > ul{

  display:block;

}

}
```

```
.menu-container{ //En esta clase se establecen el estilo para la barra de menu
  font-family: 'AvenirNextLTPro-Demi';
  height:60px;
  position:relative;
  background-color:#6abbfd;
  -webkit-box-shadow: 0px 5px 13px rgba(50, 50, 50, 0.37);
  -moz-box-shadow:    0px 5px 13px rgba(50, 50, 50, 0.37);
  box-shadow:        0px 5px 13px rgba(50, 50, 50, 0.37);
  ul{
    margin:0;
    padding:0;
    .menu-li{
      padding:0;
      float:left;
      display:inline-block;
      width:137.14px;
      background-image:url("fondo-boton-menu.png");
      z-index:1000;
      position:relative;
      .sub-menu-ul{
        .sub-menu-li{
          background-image:none;
          background-color:#589AD9;
          height:36px;
          line-height:36px;
          border-bottom: 2px solid #3D6DBB;
          width:190px;
          margin-left:25px;
        }
      }
    }
  }
}
```

```
a{
    height:36px;
    line-height:36px;
}
}
.sub-menu-li:first-child{
    border-top: 2px solid #3D6DBB;
}
.sub-menu-li:hover{
    background-color:#69BBFD;
}
}
a{
    font-size:14px;
    display:block;
    text-align:center;
    line-height:60px;
    color:white;
    padding:0;
    font-weight:normal;
    text-decoration:none;
    &:hover{
        text-decoration:none;
    }
    &:focus{
        text-decoration:none;
    }
}
```

```
    }  
    .menu-li:hover{  
        color:white;  
        background-image:url("fondo-boton-menu-hover.png");  
    }  
}  
}  
}
```

```
//Inicio clases del Footer
```

```
.footer{  
    width:960px;  
    height:70px;  
    line-height:71px;  
    font-size:12px;  
    padding-top:10px;  
    background:none;  
    ul{  
        list-style: none;  
        width:500px;  
        margin-top: 30px;  
  
        li{  
            float:left;  
            margin: 0 3px;  
            display: inline-block;  
        }  
    }  
}
```

```
}
```

```
.forgot-password-content{  
  .input-rower{  
    font-family: 'AvenirNextLTPro-Demi';  
    margin-top:10px;  
    p{  
      color:#4e4e4e;  
    }  
  }.normal-link{  
    color:#4e4e4e;  
    &:hover{  
      text-decoration:underline;  
    }  
  }  
  .social-media-sign-in{  
    display:inline-block;  
    margin-right: 15px;  
  }  
  .control-group{  
    margin-bottom:0;  
    line-height:40px;  
    width:300px;  
    display:inline;  
    label{  
      width:70px;  
      display:inline-block;
```





```

    }
  }
}
}
.submit-btn{
  display:inline;
  margin-top:5px;
}
}

// estilos para las vistas de sesion y registro -
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<

.new-blog-container{
  #blogger_blog_blog_name{
    width: 300px;
  }
  textarea{
    border: 1px solid black;
    width:905px;
    height:300px;
  }
  .submit-btn{
    margin-right:0;
  }
}

.session-content{

```

```
padding-top:0;

font-family: 'AvenirNextLTPro-Regular';

.input-rower{

margin-top:10px;

p{
  color:#4e4e4e;
}

.normal-link{
  color:#4e4e4e;

  &:hover{
    text-decoration:underline;
  }
}

.social-media-sign-in{
  display:inline-block;
  margin-right: 15px;
}

.control-group{
  margin-bottom:0;
  line-height:40px;

  label{
    width:120px;
    display:inline-block;
    line-height:40px;
    margin-bottom:0;
    color:#4e4e4e;
  }
}
```



```
}  
}  
.registration-content{  
  padding:0 20px;  
  width:920px;  
  font-family: 'AvenirNextLTPro-Regular';  
.input-rower{  
  margin-top:10px;  
  font-family: 'AvenirNextLTPro-Regular';  
  p{  
    color:#4e4e4e;  
  }  
.normal-link{  
  color:#4e4e4e;  
  &:hover{  
    text-decoration:underline;  
  }  
}  
.social-media-sign-in{  
  display:inline-block;  
  margin-right: 15px;  
}  
.control-group{  
  margin-bottom:0;  
  line-height:40px;  
  label{  
    width:190px;  
    display:inline-block;
```





```
&:hover{
    text-decoration:none;
    color:white;
    background-color:#6abcfd;
}
&:focus{
    text-decoration:none;
    color:white;
    background-color:#6abcfd;
}
}

.alert.fade{
    margin-bottom:0;
}
```