



# **Desarrollo de una red social para el aprendizaje**

Memoria de Proyecto Final de Grado

***Grado Multimedia***

**Autor: Edgar Cumbreras Tirado**

Consultor: Ignasi Lorente Puchades

Profesor: Carlos Casado Martínez

Martes, 16 de Junio del 2015

## Copyright

La licencia usada para este proyecto es bajo creative commons siendo esta Reconocimiento – Compartir Igual Internacional 4.0

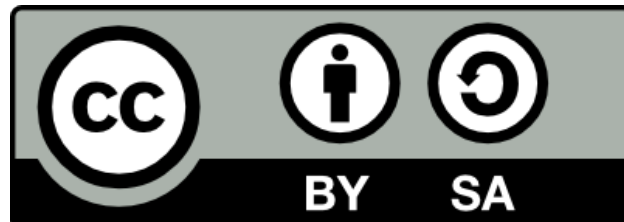


Figura 1. Imagen de la licencia 4.0

Más información en: <https://creativecommons.org/licenses/by-sa/4.0/>

## **Dedicatoria**

En especial a mi mujer *Eeva* por que sin su apoyo, su ayuda y comprensión no hubiera sido capaz de llegar tan lejos, ella está siendo la clave de mi éxito, gracias. A mis *padres* que aunque estén en la distancia siempre me han respaldado y a *David* que siempre ha estado ahí para compartir su conocimiento y experiencia.

Someone is sitting in the  
shade today because someone  
planted a tree a long time ago.

*Warren **Buffett***

*Desarrollo de una red social para el aprendizaje*  
*Grado Multimedia, Edgar Cumbreras Tirado*

## **Resumen**

Este trabajo final de grado se basa en el concepto de las redes sociales, creando una especializada para apoyar a aquellos usuarios que se estén embarcando en la idea de aprender un concepto. La idea de aprender una nueva tecnología que puede ser totalmente desconocida siempre genera dudas de cual es el mejor camino a tomar, por lo tanto, en esta red social se promoverá la idea de compartir caminos de aprendizaje. Estos caminos, estarán mostrados en la aplicación como listas de elementos que son recomendable leer, estudiar, ver o trabajar en orden de completar la comprensión y aprendizaje de dichos conceptos.

El proyecto creado para el trabajo tiene el concepto de generar un plan de negocio para la auto-sostenibilidad económica y financiera del proyecto, haciendo que una vez se ponga en producción para que el usuario sea capaz de usarlo, dispondrá de una manera de generar ingresos económicos básicos que puedan llegar en un cierto periodo de tiempo a pagar por los gastos de funcionamiento del mismo.

Cabe remarcar que todo el proyecto ha sido desarrollado usando metodologías ágiles de gestión de proyecto así como de desarrollo de software siendo estas Scrum y Extreme Programming. La plataforma ha sido creada completamente en JavaScript usando tecnologías como Node.js para el lado del servidor usando MongoDB como base de datos no relacional y haciendo uso de un MVC para la gestión del cliente como es AngularJS.

## **Anotaciones y convenciones**

Titulo: Georgia 20pt negrita

**The quick brown fox jumps over the lazy dog**

Subtitulo: Georgia 16pt negrita

**The quick brown fox jumps over the lazy dog**

Sección: Georgia 12pt cursiva y negrita

***The quick brown fox jumps over the lazy dog***

Párrafo: Georgia 10pt regular

The quick brown fox jumps over the lazy dog

Cita: Century School 10pt cursiva

*The quick brown fox jumps over the lazy dog*

Código: Monaco Monospace 10pt

The quick brown fox jumps over the lazy dog

Anotaciones: Georgia 10pt

EJEMPLO: [HTTP://WWW.URL-DE-EJEMPLO.COM](http://www.url-de-ejemplo.com)

# Índice

1. INTRODUCCIÓN	14
2. DESCRIPCIÓN	16
3. OBJETIVOS	18
3.1 PRINCIPALES	18
3.2 SECUNDARIOS	18
4. MARCO TEÓRICO	19
5. CONTENIDOS	21
5.1 TIPOLOGÍA Y FORMATO	21
5.2 ESTRUCTURA	21
5.2.1 HOME	21
5.2.2 MENÚ DEL USUARIO	22
5.2.3 PATHS	23
6. METODOLOGÍA	24
7. PLANIFICACIÓN	26
8. UX	28
9. PROTOTIPOS	33
9.1 PROTOTIPOS: BAJA RESOLUCIÓN	33
9.1.1 HOME	33
9.1.2 PATH: VISTA UNITARIA	34
9.1.3 PATH: CREAR Y EDITAR	35
9.1.4 PERFIL: SEGUIDORES Y SIGUIENDO	36
9.1.5 PERFIL: ME GUSTA	37
9.1.6 PATHS: PERFIL	38
9.1.7 OPCIONES Y AJUSTES	39
9.2 PROTOTIPO: ALTA RESOLUCIÓN	40
10. PERFILES DE USUARIOS	41
10.1 USUARIOS QUE BUSQUEN APRENDER UNA NUEVA TECNOLOGÍA	41
10.2 USUARIOS QUE CONOZCAN UNA NUEVA TECNOLOGÍA	41
11. ARQUITECTURA DE LA APLICACIÓN	44
12. PROCESO DE DESARROLLO	46
12.1 ESTRUCTURA DE LA APLICACIÓN	47

*Desarrollo de una red social para el aprendizaje*  
*Grado Multimedia, Edgar Cumbreras Tirado*

12.2 SECCIONES DE MENCIÓN	47
12.2.1 CONFIGURACIÓN DE MONGODB	47
12.2.2 CONFIGURACIÓN DEL MIDDLEWARE	48
12.2.3 SERVICIO DE ANGULARJS PARA THUMBS	48
13. DIAGRAMA UML: CLASES	50
14. APIS UTILIZADAS	51
15. PLATAFORMA DE DESARROLLO	52
16. SEGURIDAD	53
16.1 PROCESO DE IDENTIFICACIÓN	55
17. TESTS	57
17.2 TEST DE RENDIMIENTO	57
17.1 ANÁLISIS DEL STACK EN CARGA	58
18. PROCESO DE PRUEBAS	59
19. BUGS	60
19.1 LISTADO DE BUGS	60
20. VERSIONES DE LA APLICACIÓN	61
20.1 PAC 2	61
20.2 PAC 3	61
20.3 ENTREGA FINAL	61
21. REQUISITOS DE INSTALACIÓN	62
21.1 VERSIONADO	62
21.2 DEPENDENCIAS ESTÁNDARES	63
22. INSTRUCCIONES DE INSTALACIÓN	64
23. INSTRUCCIONES DE USO	66
23.1 CASOS DE USO	66
23.1 CREACIÓN DE UN USUARIO	66
23.2 CREACIÓN Y EDICIÓN DE UN PATH	68
23.3 PUBLICAR UN NUEVO COMENTARIO	70
23.4 VER LOS COMENTARIOS PREVIAMENTE PUBLICADOS	71
23.5 SEGUIR A UN USUARIO	72
23.6 VOTAR UN PATH	72
23.7 HACER UNA BÚSQUEDA	72
23.7 ACCESO AL MENÚ DE OPCIONES	73



*Desarrollo de una red social para el aprendizaje*  
*Grado Multimedia, Edgar Cumbreras Tirado*

24. CONFIGURACIÓN Y ADMINISTRACIÓN DE SERVIDORES	74
24.1 BANDERAS DE CONFIGURACIÓN	74
25. PLAN DE NEGOCIO	75
25.1 ELEVATOR PITCH	75
25.2 OBJETIVOS Y VISIÓN	75
25.3 OBSTÁCULOS	75
25.4 ESTRATEGIA DE MARKETING	76
25.5 ANÁLISIS DE MERCADO	76
25.5.1 TAMAÑO DEL MERCADO	76
25.5.4 TENDENCIAS DEL MERCADO	77
25.5.5 COMPETIDORES	77
25.6.1 PRESUPUESTO	77
25.6.2 PREVISIÓN FINANCIERA	78
CONCEPTO	78
VALOR BRUTO ANUAL	78
COMENTARIO	78
25.9 PROYECCIÓN DE FUTURO	79
28. CONCLUSIÓN	80
ANEXO 1. ENTREGABLES DEL PROYECTO	81
ANEXO 2. CÓDIGO FUENTE	82
ANEXO 2.1 CONTROLADOR DE RUTAS	82
ANEXO 2.2 CONFIGURACIÓN DEL MIDDLEWARE SOBRE EXPRESS.JS	83
ANEXO 2.3 CONFIGURACIÓN DE PASSPORT PARA AUTENTICACIÓN FACEBOOK	84
ANEXO 2.4 RUTAS PARA EL ENDPOINT DEL MODELO USER	85
ANEXO 2.5 ESQUEMA (MONGOOSE) PARA EL MODELO USER	86
ANEXO 2.6 ATRIBUTOS VIRTUALES PARA EL MODELO USER	87
ANEXO 2.7 ENDPOINT PARA LA SUBIDA DE IMÁGENES A AMAZON AWS S3	88
ANEXO 2.8 CONFIGURACIÓN DEL MODULO IMAGER	89
ANEXO 2.9 UNIT TEST PARA EL MODELO THUMB	90
ANEXO 2.10 ENDPOINT – CONTROLADOR PARA EL MODELO STEP	91
ANEXO 2.11 ENDPOINT – CONTROLADOR PARA LA BÚSQUEDA	92
ANEXO 2.12 SERVICIO ANGULAR PARA IDENTIFICACIÓN Y REGISTRO DEL MODELO USER	93
ANEXO 2.13 VISTA DE LOS COMENTARIOS	94
ANEXO 2.14 RUTAS DE ROUTER DE ANGULAR PARA LOS FOLLOWERS (GESTIÓN DE UNA SAP)	94

*Desarrollo de una red social para el aprendizaje*  
*Grado Multimedia, Edgar Cumbreras Tirado*

ANEXO 2.15 MODULO PRINCIPAL DEL FRONT-END (CONFIGURACIÓN DE ANGULARJS Y DEL ROUTER)	96
ANEXO 2.16 CONTROLADOR DE ANGULAR PARA LA SECCIÓN HOME	97
ANEXO 3. LIBRERÍAS EXTERNAS	98
ANEXO 4. CAPTURAS DE PANTALLA	99
ANEXO 4.1 HOME	99
4.2 SIGN UP (REGISTRO)	100
4.3 LOGIN (IDENTIFICACIÓN)	101
4.5 PATH EDIT (EDITAR PATH)	102
4.5 SETTINGS (AJUSTES)	103
4.6 PATH	104
4.7 COMMENTS (COMENTARIOS)	105
4.8 FOLLOWING Y FOLLOWERS (SIGUIENDO Y SEGUIDO)	106
4.9 PERFIL	107
ANEXO 5. GLOSARIO	108
ANEXO 6. BIBLIOGRAFÍA	109
ANEXO 7. BIOGRAFÍA DEL AUTOR	110

# Imágenes

FIGURA 1. IMAGEN DE LA LICENCIA 4.0	2
FIGURA 2. CONCEPTO VISUAL DEL DESARROLLO MODERNO VS CONVENCIONAL	25
FIGURA 3. DIAGRAMA DE GANTT	27
FIGURA 4. <i>SPINNER</i> DE CARGA AL SUBIR LAS IMÁGENES	29
FIGURA 5. VISTA DE LOS STEPS	29
FIGURA 6. ICONOS PRIMARIOS DE ACCIÓN	30
FIGURA 7. EJEMPLO DE ERRORES EN EL FORMULARIO DE REGISTRO	31
FIGURA 8. SECCIÓN DE BÚSQUEDA	32
FIGURA 9. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN HOME	33
FIGURA 10. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN VISTA UNITARIA	34
FIGURA 11. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN CREAR Y EDITAR PATH	35
FIGURA 12. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN SEGUIDORES Y SIGUIENDO	36
FIGURA 13. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN THUMB-UPS	37
FIGURA 14. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN PERFIL	38
FIGURA 15. PROTOTIPO DE BAJA RESOLUCIÓN: SECCIÓN OPCIONES Y AJUSTES	39
FIGURA 16. PROTOTIPO DE ALTA RESOLUCIÓN: SECCIÓN HOME	40
FIGURA 17. CASO DE USO PRIMARIO	43
FIGURA 18. ARQUITECTURA MVC	44
FIGURA 19. DIAGRAMA DE LA ARQUITECTURA WEB	45
FIGURA 20. DIAGRAMA DE CLASES	50
FIGURA 21. APLICACIÓN EN FACEBOOK DEVELOPERS	51
FIGURA 22. <i>PROFILING</i> DE LA CARGA DE ARCHIVOS	57
FIGURA 15. PROFILING DEL STACK DE LA APLICACIÓN	58
FIGURA 24. CAPTURA DEL ENLACE A SIGN UP	67
FIGURA 25. CAPTURA DE LA PANTALLA DE REGISTRO	67
FIGURA 26. CAPTURA DEL ACCESO RÁPIDO A LA CREACIÓN DE PATHS	68
FIGURA 27. CAPTURA DE UN PATH	68
FIGURA 28. CAPTURA DEL MODO EDICIÓN DEL PATH	69
FIGURA 29. CAPTURA DE LA SECCIÓN DE COMENTARIOS	70
FIGURA 22. CAPTURA DEL MENU PRINCIPAL	71
FIGURA 31. CAPTURA DE LA SECCIÓN INDIVIDUAL DE COMENTARIOS	71
FIGURA 32. CAPTURA DE LA CABEZERA MOSTRANDO EL BOTON DE SEGUIMIENTO	72
FIGURA 33. CAPTURA DE LA VERSIÓN REDUCIDA DE UN PATH MOSTRANDO LA OPCIÓN DE THUMB-UPS	72
FIGURA 34. CAPTURA DE LA CABEZERA MOSTRANDO LA OPCIÓN DE BÚSQUEDA	73
FIGURA 27. CAPTURA DEL MENÚ PRINCIPAL	73

FIGURA 36. CAPTURA DE LA CONFIGURACIÓN DE DEPLOYMENT DEL SERVIDOR EN HEROKU	74
FIGURA 37. CAPTURA DE LA SECCIÓN HOME	99
FIGURA 38. CAPTURA DE LA SECCIÓN DE REGISTRO	100
FIGURA 39. CAPTURA DE LA SECCIÓN DE IDENTIFICACIÓN	101
FIGURA 40. CAPTURA DE LA SECCIÓN DE EDICIÓN DE PATH	102
FIGURA 41. CAPTURA DE LA SECCIÓN DE LOS AJUSTES	103
FIGURA 42. CAPTURA DE LA SECCIÓN UNITARIA DEL PATH	104
FIGURA 43. CAPTURA DE LA SECCIÓN DE COMENTARIOS	105
FIGURA 44. CAPTURA DE LA SECCIÓN DE SEGUIENDO	106
FIGURA 45. CAPTURA DE LA SECCIÓN DEL PERFIL DE USUARIO CON PATHS	107

## **Tablas**

TABLA 1. PLANIFICACIÓN	27
TABLA 2. LISTA DE BUGS	60
TABLA 3. PREVISIÓN FINANCIERA	78

*Desarrollo de una red social para el aprendizaje*  
*Grado Multimedia, Edgar Cumbreras Tirado*

## **1. Introducción**

Desde hace unos años hemos visto como la tecnología ha invadido nuestro entorno día a día cambiando la manera en la que nos comunicamos hasta la manera en la que gastamos nuestro tiempo de ocio. Muchos son los sectores e industrias que han cambiado gracias a estas innovaciones, pero si he de elegir una que haya tenido un especial impacto en mí, esta ha de ser la de la educación. El sector educativo está cambiando de manera paulatina mientras usted lee este texto, nuevas plataformas y nuevas maneras de compartir conocimiento son desarrolladas en este mismo momento.

Es una gran época para todos aquellos que quieran compartir y hacer uso de esta fuente de conocimiento sin fin, como es internet, ya que el mero hecho de vivir a más de 1500 kilómetros a la sede más cercana hubiera supuesto que sacarse el grado fuera inviable para mí. Este tipo de cambios en la tecnología gracias a la red han sido una de las razones por las que actualmente valore más las posibilidades que he dispuesto durante estos cuatro años gracias a estudiar a distancia.

Durante estos años he sido capaz de experimentar una mejora a nivel profesional desde el primer semestre, tal ha sido la mejora que se ha producido en mí durante estos cuatro años que he sido capaz de cambiar de carrera completamente y en la actualidad trabajar como desarrollador web.

Gran parte de mi día a día como desarrollador web está enfocado en ganar conocimiento, expandir las posibilidades así como las herramientas con las que trabajo, por ello mismo me es necesario hacer uso de la red de manera habitual con dicho propósito. En la mayoría de las ocasiones se dispone de una idea en cuales son los lugares óptimos o cuales la metodología a usar para conocer el uso de dicha tecnología o herramienta, sin embargo existen otros tantos momentos que no es tan factible obtener el camino correcto para la comprensión y aprendizaje de la misma.

Por ello mismo quisiera que mi trabajo final de grado sea focalizado en solventar el problema que muchos estudiantes o trabajadores disponen cuando están

aprendiendo algo nuevo por si solos y no disponen de un apoyo, ayuda o guía para lograr el propósito como yo he disfrutado durante estos años gracias al apoyo de la universidad en la que he cursado este grado.

La necesidad personal de devolver a la comunidad parte de lo que he obtenido en ella, ha generado la idea que mi trabajo final de grado tenga una finalidad cien por cien dedicada a esta materia. Poder ofrecer a cualquier usuario de internet una plataforma que pueda llegar a ayudarles en el momento de emprender un nuevo viaje para aprender una nueva tecnología.

## **2. Descripción**

El concepto de este trabajo final de grado es desarrollar desde cero una red social focalizada en ayudar, promover y enfatizar el aprendizaje. Esta red social está ideada en el concepto del auto aprendizaje, un concepto que la mayoría de los ingenieros de software tienen muy presente debido a la cantidad de tecnologías y a las innovaciones que aparecen el mercado han de disponer de una formación continuada.

La necesidad y el deseo de aprender nuevas tecnologías de manera continua es un proceso metodológico que se obtiene con el tiempo y la experiencia, sin embargo, si se desconoce por completo la tecnología es necesario comenzar este proceso buscando toda la información y recursos que serán usados durante este proceso de aprendizaje.

Por lo tanto para poder incentivar el aprendizaje de nuevas tecnologías queremos crear una red social de modo que los usuarios sean capaces de compartir el proceso metodológico que han practicado durante los meses o quizá los años para aprender dicha tecnología. De esta manera sería posible disponer de diferentes recomendaciones de cual es la mejor manera de aprender esa nueva técnica ahorrando tiempo y despertando interés sobre alguna tecnología desconocida.

El proceso de consolidación de dichas recomendaciones es mediante un sistema de votos y comentarios que se harían sobre las listas que los usuarios han publicado. Si existen múltiples listas sobre el mismo lenguaje de programación los usuarios podrán ver las opiniones que esa lista ha obtenido.

La plataforma es desarrollada completamente en JavaScript tanto la parte del cliente como la del servidor usando el conocido MEAN stack<sup>1</sup>. Dicho conjunto de diferentes tecnologías está compuesto de una base de datos no relacional como es MongoDB, hace uso de Node.js en el lado del servidor en particular del

---

<sup>1</sup> MEAN STACK: [HTTP://MEAN.IO/](http://MEAN.IO/)



framework<sup>2</sup> Express.JS y para la parte del cliente usa el *framework* MVC<sup>3</sup> de Google, AngularJS. El proceso de trabajo será bajo metodologías ágiles como Scrum, así como el desarrollo del software será usando Extreme Programming.

En la sección de negocio quisiéramos generar un proyecto el cual sea auto-sostenible financieramente, ofreciendo un servicio a la comunidad de manera desinteresada. Apoyándose en conceptos del *software* abierto proyecto tendría una base conceptual que ha sido requerida por la comunidad solventando un problema común para los programadores más noveles como es el desconocimiento total de cómo empezar con un lenguaje de programación, herramienta o tecnología.

El proyecto acabado se puede visitar temporalmente en el siguiente enlace:

<https://learning-path.herokuapp.com/>

---

<sup>2</sup> FRAMEWORK: [HTTP://ES.WIKIPEDIA.ORG/WIKI/Framework](http://es.wikipedia.org/wiki/Framework)

<sup>3</sup> MVC: [HTTP://ES.WIKIPEDIA.ORG/WIKI/Modelo-Vista-Controlador](http://es.wikipedia.org/wiki/Modelo-Vista-Controlador)

## **3. Objetivos**

A continuación procedemos a enumerar los objetivos principales y secundarios de este proyecto.

### **3.1 Principales**

- Desarrollar una plataforma abierta para apoyar e incentivar el aprendizaje.
- Utilizar unificadamente conceptos aprendidos durante el grado.
- Producir un producto viable para el uso público.

### **3.2 Secundarios**

- Hacer uso de nuevas tecnologías relacionadas con el desarrollo web.
- Seguir metodologías y procesos ágiles de desarrollo de software.
- Disponer de un proporción cercana a 1:1 en cobertura de código<sup>4</sup>.
- Generar un plan de negocio para ofrecer una auto-mantenimiento financiera del proyecto.

---

<sup>4</sup> COBERTURA DE CÓDIGO: [HTTP://ES.WIKIPEDIA.ORG/WIKI/COBERTURA\\_DE\\_CÓDIGO](http://es.wikipedia.org/wiki/COBERTURA_DE_CÓDIGO)

## **4. Marco teórico**

El concepto base que hemos dispuesto para el desarrollo conceptual de esta red social ha sido basado en la iniciativa de software libre así como en el concepto y la filosofía en la que muchas de las nuevas plataformas de educación de manera gratuita se han formado.

El modelo argumental de la idea es un directo reflejo del movimiento creado por la el Instituto tecnológico de Massachusetts<sup>5</sup> (MIT) con *Open CourseWare*<sup>6</sup>. En sus comienzos este movimiento proveía acceso abierto y gratuito a materiales de universidades asociadas al programa. En el año actual, el sitio dispone de más de 1900 cursos ofreciendo cursos específicos en múltiples disciplinas de las cuales se pueden obtener, materiales educativos, proyectos y tareas así como exámenes online para cualquier individuo que quiera completarlos.

Existen en la actualidad un incipiente crecimiento de las plataformas que están basadas en la cultura del *Knowledge Sharing*<sup>7</sup>, plataformas multinacionales como Code.org apoyadas por gente tan importante en el mundo de la tecnología y las ciencias como Bill Gates, Stephen Hawking o Mark Zuckerberg ofrecen servicios educativos relativos a la programación e ingeniería de software, promoviendo el concepto de la hora de código, un concepto que se basa en mostrar los conceptos básicos de la programación haciendo uso de una plataforma que dispone de herramientas y actividades para todas las edades. Con movimientos culturales como este se promueve el interés por aprender a programar e incluso abrir una carrera para aquellos que desconocían totalmente este campo.

Como podemos ver estos tres conceptos son una base real que está ayudando a miles de personas a descubrir nuevas posibilidades de estudio y trabajo, por lo tanto con la creación y desarrollo de esta plataforma queremos unificar todos este tipo de concepto, culturas e ideas.

---

5 [HTTP://WEB.MIT.EDU/](http://web.mit.edu/)

6 [HTTP://OCW.MIT.EDU/DONATE/WHY-DONATE/](http://ocw.mit.edu/donate/why-donate/) - Q1

7 [HTTP://EN.WIKIPEDIA.ORG/WIKI/KNOWLEDGE\\_SHARING](http://en.wikipedia.org/wiki/Knowledge_Sharing)

Deseo ofrecer una plataforma de código libre la cual cualquier persona puede disponer de acceso al código fuente de la misma e inclusive ofrecer mejoras o cambios. Queremos abrir la posibilidad de disponer de una fuente de conocimiento en el que los usuarios puedan compartir maneras y enfoques para solventar problemas relativos al aprendizaje de nuevas tecnologías y deseamos que nuestro concepto abra la posibilidad de ofrecer soluciones para aquellas personas que no disponen de un conocimiento previo para poder iniciarse en el mundo de la ingeniería, el *software* y las tecnologías web.

## **5. Contenidos**

El concepto matriz de esta red social es ofrecer contenido de una manera ligera, creando un servicio de referencia para la adquisición de nuevos conocimientos. Facilitando la disposición de dichos contenidos en múltiples dispositivos así como la interacción de los mismos con las redes sociales.

### **5.1 Tipología y formato**

La red social que hemos desarrollado dispone de un contenido distribuido de una manera menos común de lo habitual, ya que la centralización del contenido e información asociada con la plataforma es mostrada en formato de lista ordinal. Este tipo de contenido directamente descendiente del texto, es la idea central de la plataforma, apoyándose en este tipo de estructura documental para poder facilitar un apoyo y directriz educativa de una manera sencilla, lógica y organizada; haciendo que secciones como Home, contiene tres diferentes categorías que de haber enfocado la página de otra manera deberían de ser secciones por si mismas.

### **5.2 Estructura**

Esta red social hace uso de parte del concepto de aplicación de una sola página, *Single page application*<sup>8</sup> (SAP) en muchas secciones de la misma, creando una estructura ágil, la cual dispone de la posibilidad de ofrecer contenido sin tener que ofrecer recargas de páginas completas. Gracias a este enfoque hemos podido organizar la estructura de la página en dos secciones principalmente que se sub-dividen en múltiples, siendo capaces de incrementar las características de la misma sin tener que cambiar el árbol de contenidos de manera drástica.

#### **5.2.1 Home**

Al disponer de este estilo de contenido ligero y dinámico la plataforma dispondrá de una estructura centralizada a una lista global que aparecerá en la página de

---

<sup>8</sup> [HTTP://EN.WIKIPEDIA.ORG/WIKI/SINGLE-PAGE\\_APPLICATION](http://en.wikipedia.org/wiki/Single-page_application)

inicio. Esta sección comprenderá la página inicial la cual mostrará una serie de listas globales que serán compuestas por tres categorías:

- **Casa** (*Home*): La sección principal de la aplicación que se muestra al inicio de la visita al sitio web, mostrando todos los paths creados sin ningún tipo de filtro aplicado.
- **De moda** (*Trending*): En esta categoría de la página inicial podemos ver el contenido que más atención está obteniendo, es decir, las listas que mayor crecimiento de puntuación han obtenido en un reciente periodo de tiempo.
- **Punteros** (*Top*): En esta categoría se mostrarán las listas con mayor puntuación desde el comienzo de la red social.
- **Más comentados** (*Most-commented*): El contenido es organizado de manera que se muestra de mayor a menor en cantidad de comentarios.

### **5.2.2 Menú del usuario**

El usuario dispone de un menú específico para poder gestionar todas las secciones que este tiene disponible. En cada una de las opciones del menú se puede acceder a una vista determinada del contenido, ofreciendo así una vista detallada del contenido relativo al usuario.

Las opciones que ofrece el menú son las siguientes:

- **Tus paths** (*Your paths*): En esta opción podemos ver todos los paths creados por el dueño del perfil.
- **Tus votos** (*Your thumbsups*): Aquí podemos ver la recopilación de los paths a los que el usuario ha votado.
- **Tus seguidores** (*Your followers*): Lista identificativa de usuarios que siguen al usuario del perfil.
- **Estás siguiendo** (*You are following*): Lista de usuarios los cuales son seguidos por dicho usuario.
- **Tus comentarios** (*Your comments*): Lista de los comentarios creados por el usuario, mostrando un enlace al path relativo.
- **Tus ajustes** (*Your settings*): Sección para que el usuario pueda cambiar principalmente su contraseña y subir un nuevo avatar.

### **5.2.3 Paths**

Para la creación y edición de paths, disponemos de la ya nombrada sección predeterminada para esta gestión. Esta parte del perfil del usuario, está enfocada para generar dichas acciones de una manera rápida y sencilla.

Esta sección dispone de dos modos, una en la que el usuario ve los paths que dispone y dispone de un botón para añadir nuevos paths y otro modo de edición que se activa al pulsar el botón naranja con el icono del lápiz. Entrando en este modo el usuario verá como el path muestra las opciones pertinentes:

- Cambiar el título.
- Cambiar la descripción.
- Editar (añadir y eliminar) steps.
  - Cambiar el nombre, modificar la URL.
- Editar (añadir y eliminar) tags.
- Eliminar dicho path.

## 6. Metodología

Para el correcto desarrollo de este trabajo de fin de grado hemos querido optar por una de las **metodologías de gestión de proyectos** que están más comúnmente adheridas al desarrollo de *software* y es una de las metodologías ágiles por excelencia. Esta metodología comúnmente conocida como *Scrum*, es un proceso incremental e iterativo el que se focaliza en ejecutar secciones del proyecto en bloques marcados temporalmente, cortos y de una fija duración denominado iteración o *Sprints*. Estas iteraciones serán las secciones del proyecto en las cuales trabajaremos con el concepto de lograr un objetivo previamente acordado y ajustado para ser completamente pro-eficiente en dicho tiempo.

Hemos elegido este tipo de metodología ya que es bien conocida como ser realmente efectiva en proyectos complejos como los de desarrollo de *software*. Dicha efectividad es relativa a la capacidad de resolver imprevistos, cambios o problemas que puedan producirse a lo largo del proyecto.

Este tipo de **flexibilidad adaptativa** es principalmente relacionada a la manera en la que se estructura el proyecto. El proyecto conformado por *Sprints* hace que sea posible ver como el *software* evoluciona e incrementa en funcionalidad y tamaño siendo posible adaptar y posibilitar cambios de estructura que no serían posible ver sin hasta que el programa fuera completamente desarrollado.

En los últimos años hemos visto como el concepto de *Extreme Programming* ha ido evolucionando para llegar a ser comúnmente conocida como la metodología de **desarrollo de *software*** excelencia. Este tipo de metodología normalmente usada en conjunción *Scrum* ha ido ganando adeptos por la cantidad de prácticas de ingeniería de *software* que han creado una solidez estructural a la hora de desarrollar *software*.



La metodología de Extreme Programming está definida por cuatro principales áreas:

- Focalización en la retro-alimentación:
  - Desarrollo guiado por pruebas<sup>9</sup>: *Test driven development* (TDD).
  - Pre-plan iterativo.
- Proceso de creación continuo:
  - Integración continua.<sup>10</sup>
  - Diseño por iteración.
  - Pequeñas entregas.
- Conocimiento compartido de la aplicación:
  - Diseño simple.
  - Sistema metafórico.
  - Convenciones y estándares unificados en la creación del código.

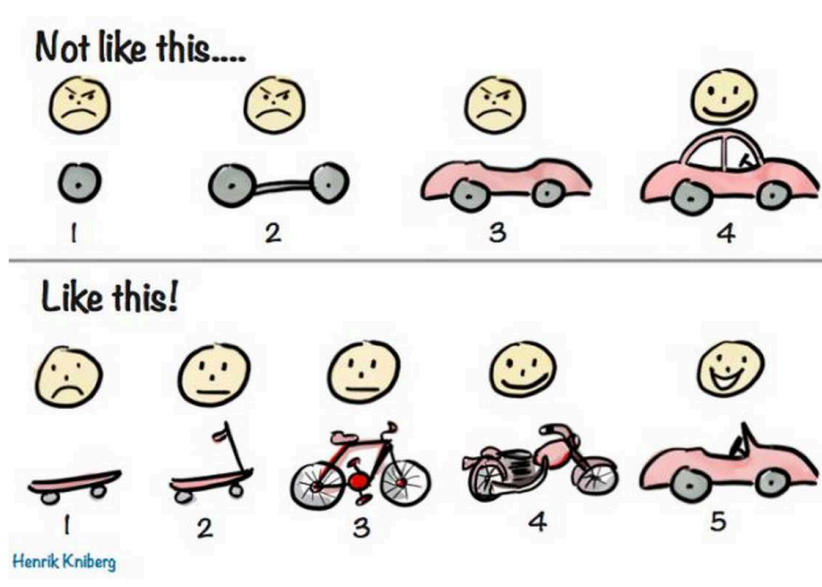


Figura 2. Concepto visual del desarrollo moderno vs convencional

Por lo tanto, nosotros haremos el uso de *Extreme Programming* como metodología de desarrollo de *software* mientras que el **proceso de gestión del proyecto** será basado en *Scrum*.

<sup>9</sup> TDD: [HTTP://ES.WIKIPEDIA.ORG/WIKI/DESARROLLO\\_GUIADO\\_POR\\_PRUEBAS](http://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas)

<sup>10</sup> INTEGRACIÓN CONTINUA: [HTTP://ES.WIKIPEDIA.ORG/WIKI/INTEGRACIÓN\\_CONTINUA](http://es.wikipedia.org/wiki/Integración_continua)

## 7. Planificación

La planificación de este proyecto ha sido basado en la estructura de estudio que dispondremos en la asignatura a la que el mismo está adherida. El proyecto será elaborado de una manera iterativa basado en los principios ágiles de metodología de proyectos. Al disponer de este tipo de enfoque de trabajo que tan bien nos ha funcionado durante toda el grado y al ser una manera de trabajo que encaja perfectamente con la filosofía de *Scrum* procederemos a encajar las secciones temporales iterativas, *Sprints*, en las que se ha dividido todo el trabajo necesario para poder finalizar este proyecto.

<b>Sprint</b>	<b>Sección</b>	<b>Posición</b>	<b>Descripción</b>	<b>Puntuación</b>
1	Desarrollo memoria	- 1	Paginas previas e introducción	1
	Desarrollo memoria	- 2	Descripción y objetivos	2
	Desarrollo memoria	- 3	Metodología y planificación	4
2	Desarrollo memoria	- 4	Presupuesto, análisis de mercado, financiero	5
	Diseño aplicación	- 5	Análisis de diseño del proyecto	3
	Diseño gráfico	6	Gráficos de baja calidad	2
3	Diseño gráfico	7	Test de usuarios	2
	Desarrollo aplicación	- 8	Diseño estructura del proyecto Modelaje UML	3
	Diseño gráfico	9	Gráficos de alta calidad	4
4	Desarrollo aplicación	- 10	Desarrollo del cliente	5
	Desarrollo aplicación	- 11	Desarrollo del servidor	5
	Desarrollo aplicación	- 12	Integración de cliente y servidor	5

*Desarrollo de una red social para el aprendizaje*  
*Grado Multimedia, Edgar Cumbreras Tirado*

5	Diseño gráfico	13	Test de usuarios 2	2
	Desarrollo aplicación	- 14	Mejoras necesarias	3
	Desarrollo memoria	- 15	Documentación sobre instalación y dependencias	4
6	Desarrollo memoria	- 16	Preparación final de la documentación necesaria	5
	Desarrollo aplicación	- 17	Puesta en marcha del proyecto	5
	Desarrollo memoria	- 18	Finalización de los entregables entrega final	5

Tabla 1. Planificación

La planificación del proyecto ha sido dividido durante la duración del proyecto haciendo que la finalización de los *Sprints* sea en la entrega de prácticas, haciendo que la gestión de dicho proyecto sea más factible.

Cabe remarcar que para poder comprender el desarrollo del proyecto de una manera más global nos es posible generar una planificación estructural de los *Sprint* de manera que pudiéramos llegar a interpretarlo como un gráfico de *Gantt*.

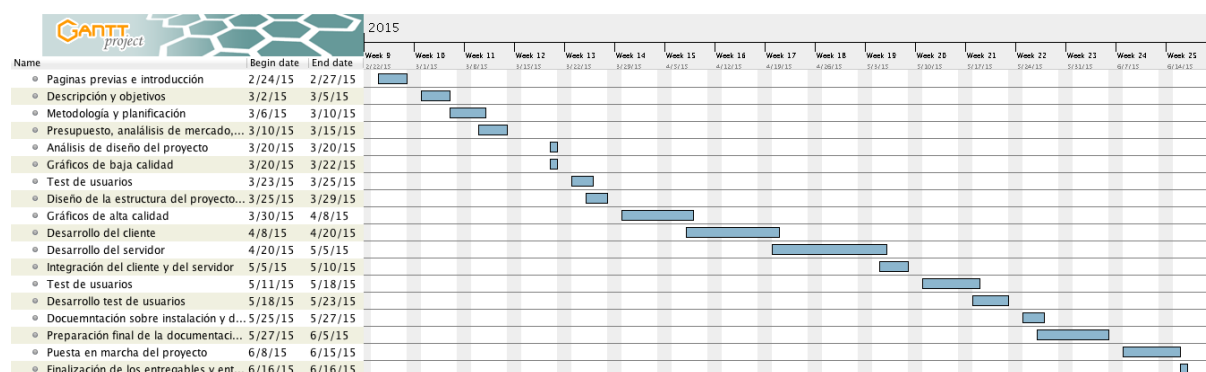


Figura 3. Diagrama de Gantt

## 8. UX

La creación de la interfaz gráfica así como el diseño de la plataforma ha sido foco de el concepto del diseño centrado en el usuario<sup>11</sup> (DCU). Esta filosofía del diseño se centra en disponer del usuario como centro del proceso de creación e integración de la interfaz gráfica. Este tipo de enfoque está dirigido en solventar y resolver problemas, cumplir con requerimientos y ofrecer un servicio al usuario objetivo de una manera directa siendo este el que tiene voz directa sobre los puntos de inflexión en la creación de la interfaz y de sus posibles usos.

Para poder generar la interfaz de la red social he pasado por tres diferentes etapas cíclicas:

1. **Investigación** cualitativa y cuantitativa de los usuarios finales de la plataforma.
2. **Diseño** de una plataforma que se ajusta a las expectativas del usuario.
3. **Prueba** incipientemente que se cumplan estos valores, con el proceso de creación cíclico por medio del test de usuarios. Generando secciones de interfaz para pasar a testearlos, con la revisión y *feedback* de los usuarios se aplican las mejoras continuando con una correcta dirección sobre el objetivo final.

Para poder completar el diseño y generar una lista de los requerimientos que nuestra aplicación debe de cumplir se han hecho múltiples evaluaciones heurísticas de páginas con similar objetivo a la plataforma que he creado.

Dichas evaluaciones heurísticas estaban basadas<sup>12</sup> en los 10 puntos de usabilidad para la creación de interfaz de usuarios:

1. **Visibilidad del estado del sistema:** El usuario debe de saber en todo momento en que estado se encuentra el sistema, conociendo si se encuentra

---

11 [HTTP://ES.WIKIPEDIA.ORG/WIKI/DISEÑO\\_CENTRADO\\_EN\\_EL\\_USUARIO](http://es.wikipedia.org/wiki/Dise%C3%B1o_centrado_en_el_usuario)

12 [HTTP://WWW.NNGROUP.COM/ARTICLES/TEN-USABILITY-HEURISTICS/](http://www.nngroup.com/articles/ten-usability-heuristics/)

algún proceso en marcha o si necesita completar alguna acción.

En la aplicación disponemos de múltiples lugares donde se mostrará un *spinner* de carga si fuese necesario.

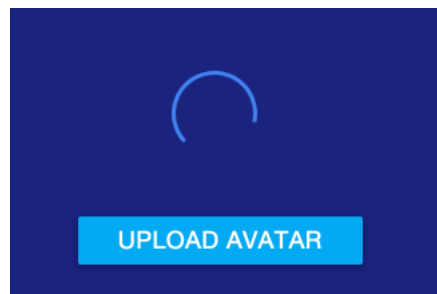


Figura 4. *Spinner* de carga al subir las imágenes

**2.Similitud entre el sistema y el mundo real:** Es necesario seguir las convecciones habituales que se usan para llegar a una comprensión sencilla de la plataforma.

Al editar cualquier path, se muestran como si de una lista estandar se tratase haciendo que sea mucho más sencillo comprender su uso.

**3.Control y libertad del usuario:** Se ha de ofrecer la posibilidad de rehacer ciertas acciones.

En cualquiera de las acciones relacionadas con la creación de steps, se habilita al usuario a borrar los elementos.

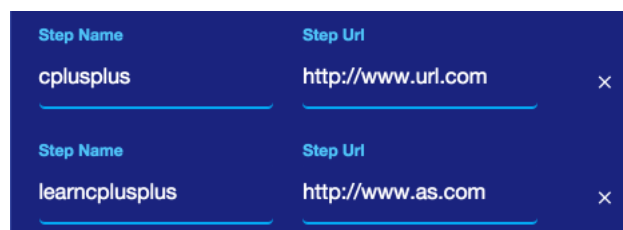


Figura 5. Vista de los steps

**4.Consistencia y cumplimiento de estándares:** La plataforma necesitará cumplir estándares y que usan los sitios web.

En nuestro caso hemos podido obtener dicha consistencia mediante la viabilidad de visitar la aplicación con diferentes navegadores obteniendo la misma experiencia de usuario.

**5.Prevencción de errores:** La prevención de errores evitará que el usuario disponga de una experiencia frustrante.

Este punto no es una característica determinada de la aplicación sino es más enfocado a la manera que se han desarrollado las acciones a lo largo de la aplicación, por ejemplo, si un usuario crea un nuevo path, verá que ya está rellenado con información de ejemplo haciendo que la comprensión para hacer dicha acción sea mucho más rápida.

**6.Preferencia al reconocimiento que a la memorización:** Es preferible que el usuario disponga de la información o acceso a los elementos de interacción con la plataforma antes que tener que hacer uso de la memoria para poder ejecutar una consecuente acción.

En la aplicación existen un solo tipo de botón para efectuar las acciones más importantes y habituales de la aplicación. Para crear y acceder a los paths, se usa el siguiente icono. Por lo tanto el usuario necesita menos tiempo y menos complejidad para comprender la mecánica para ejecutar las acciones más importantes.

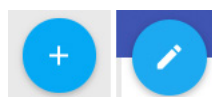


Figura 6. Iconos primarios de acción

**7.Flexibilidad y eficiencia de uso:** Revisión y mejora de la manera en la que se ofrecen los contenidos. Ofreciendo atajos de teclado y sistemas de control para usuarios avanzados.

En este caso hemos introducido el uso del botón intro como acceso a las acciones de guardar.

8. **Estética y diseño minimalista:** Los elementos que se ofrecen en pantalla han de ser solamente los necesarios, evitando distracciones o confusiones que puedan empeorar la experiencia del usuario.

En la aplicación podemos ver que las estructuras gráficas han sido reutilizadas para dar un efecto de simplicidad de esta manera el usuario reconoce las secciones posibilitando la rápida adaptación del mismo al sitio web.

9. **Reconocimiento de errores:** Los mensajes de error han de ser explicativos y entendibles para cualquier público, ofreciendo una causa y una solución al problema que acaba de ocurrir.

La aplicación está diseñada para reducir los errores al mínimo, pero en el caso que el usuario se encuentre con uno, se muestra directamente en el lugar que ha ocurrido con un lenguaje simple y directo.

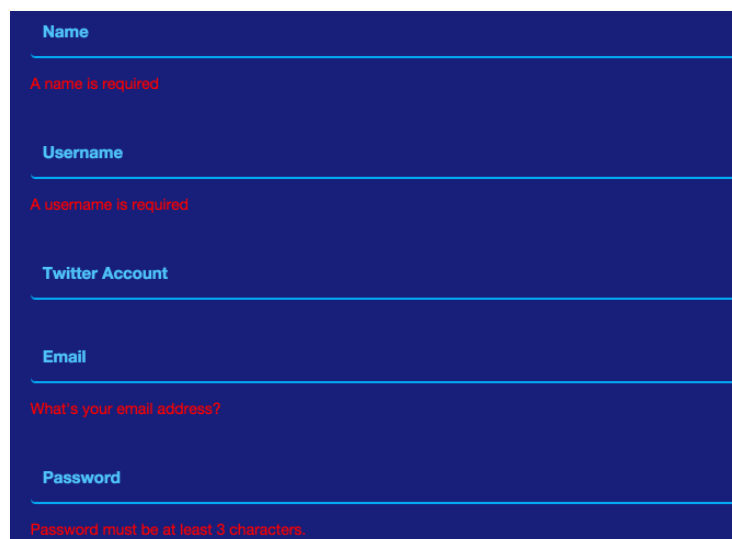
A screenshot of a registration form with a dark blue background and white text. The form has five input fields, each with a red error message below it. The fields and messages are: 'Name' with 'A name is required'; 'Username' with 'A username is required'; 'Twitter Account' with no message; 'Email' with 'What's your email address?'; and 'Password' with 'Password must be at least 3 characters.'

Figura 7. Ejemplo de errores en el formulario de registro

10. **Ayuda y documentación:** Uso de *FAQ*, secciones de búsqueda y ayuda así como rutas fáciles de aprendizaje son las bases para una robusta interfaz.

En la plataforma no disponemos de sección de ayuda, sin embargo, la sección de búsqueda se encuentra centrada en la cabecera de la página facilitando el uso de la misma.



Figura 8. Sección de búsqueda



## 9. Prototipos

### 9.1 Prototipos: Baja resolución

#### 9.1.1 Home

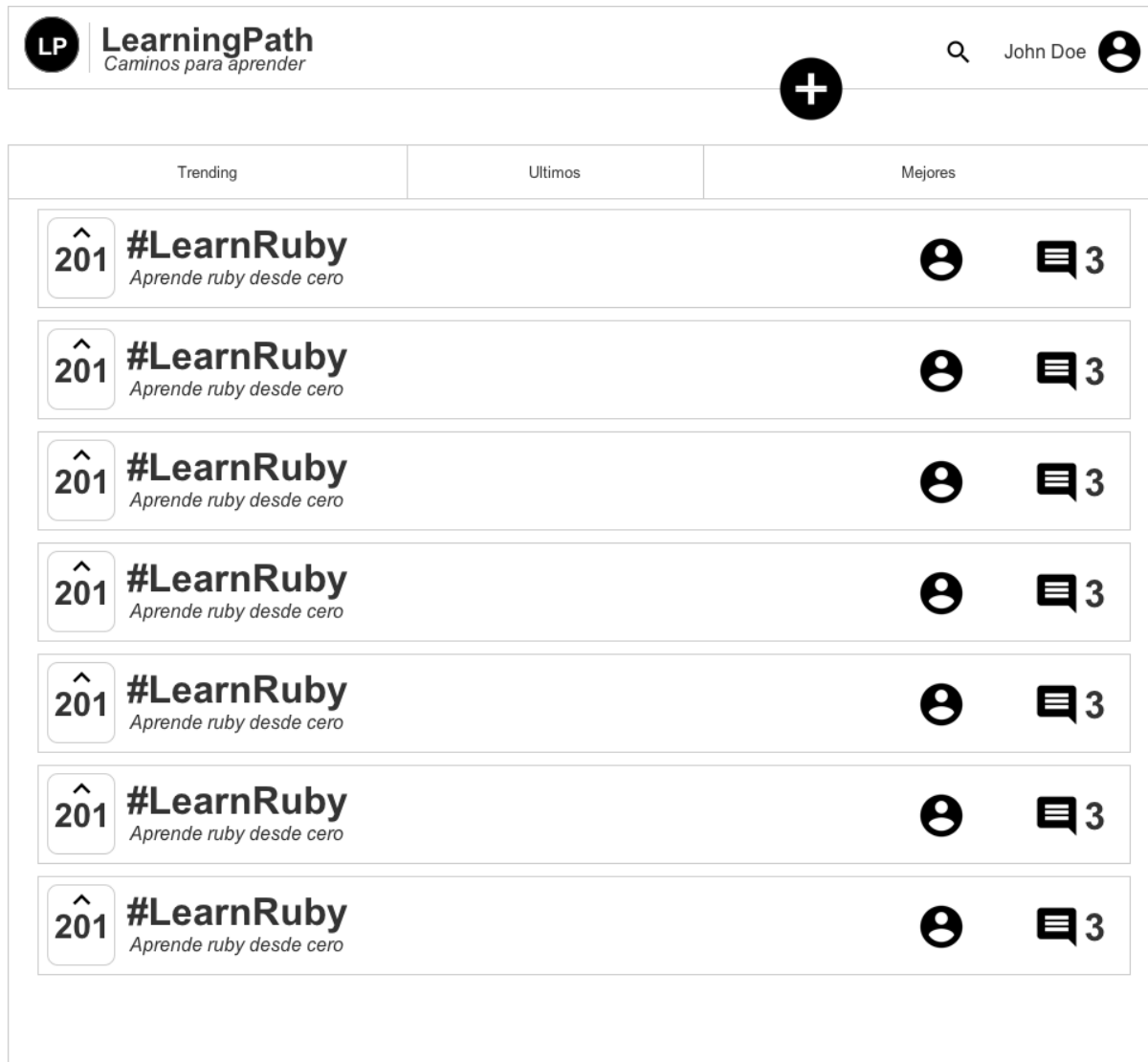


Figura 9. Prototipo de baja resolución: Sección home

En esta sección podemos ver la página principal de la aplicación donde disponemos acceso a todos los paths que hayan sido creados, el buscador, un acceso directo para editar los path del usuario identificado así como el menú del usuario.

### 9.1.2 Path: vista unitaria

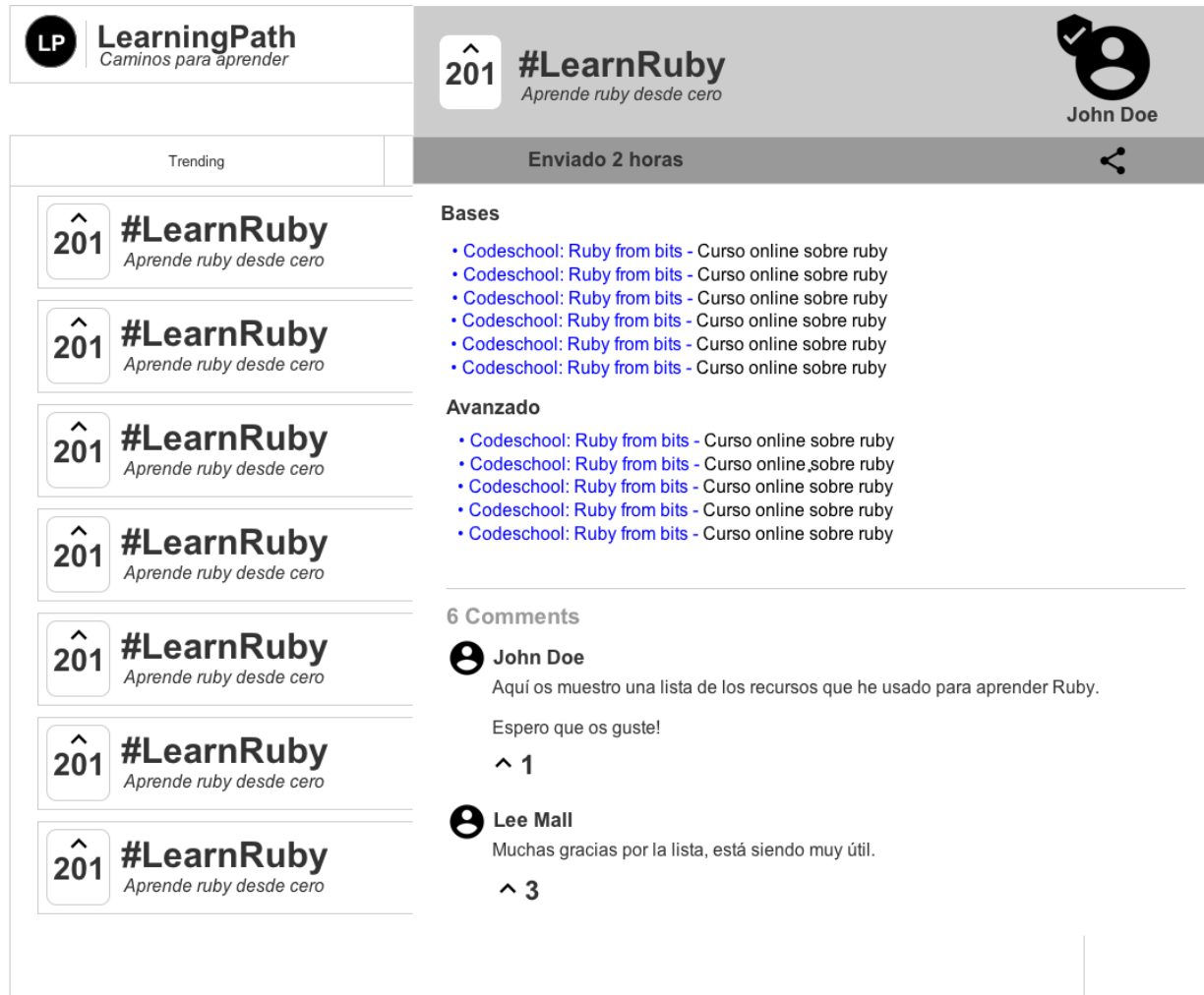


Figura 10. Prototipo de baja resolución: Sección vista unitaria

Para el path, disponemos de una vista unitaria en la que el usuario encontrará toda la información detallada sobre el path, los pasos o steps que conforman dicho elemento así como la posibilidad de añadir comentarios. En esta sección también dispone de la posibilidad de seguir al creador del path.

La sección de los comentarios, dispondrá del nombre de usuario, la fecha así como el comentario en sí del usuario.

En esta sección también se puede hacer thumb-up o me gusta de manera sencilla, disponiendo del total de los me gusta que dispone el path a la izquierda arriba de la página.

### 9.1.3 Path: Crear y editar

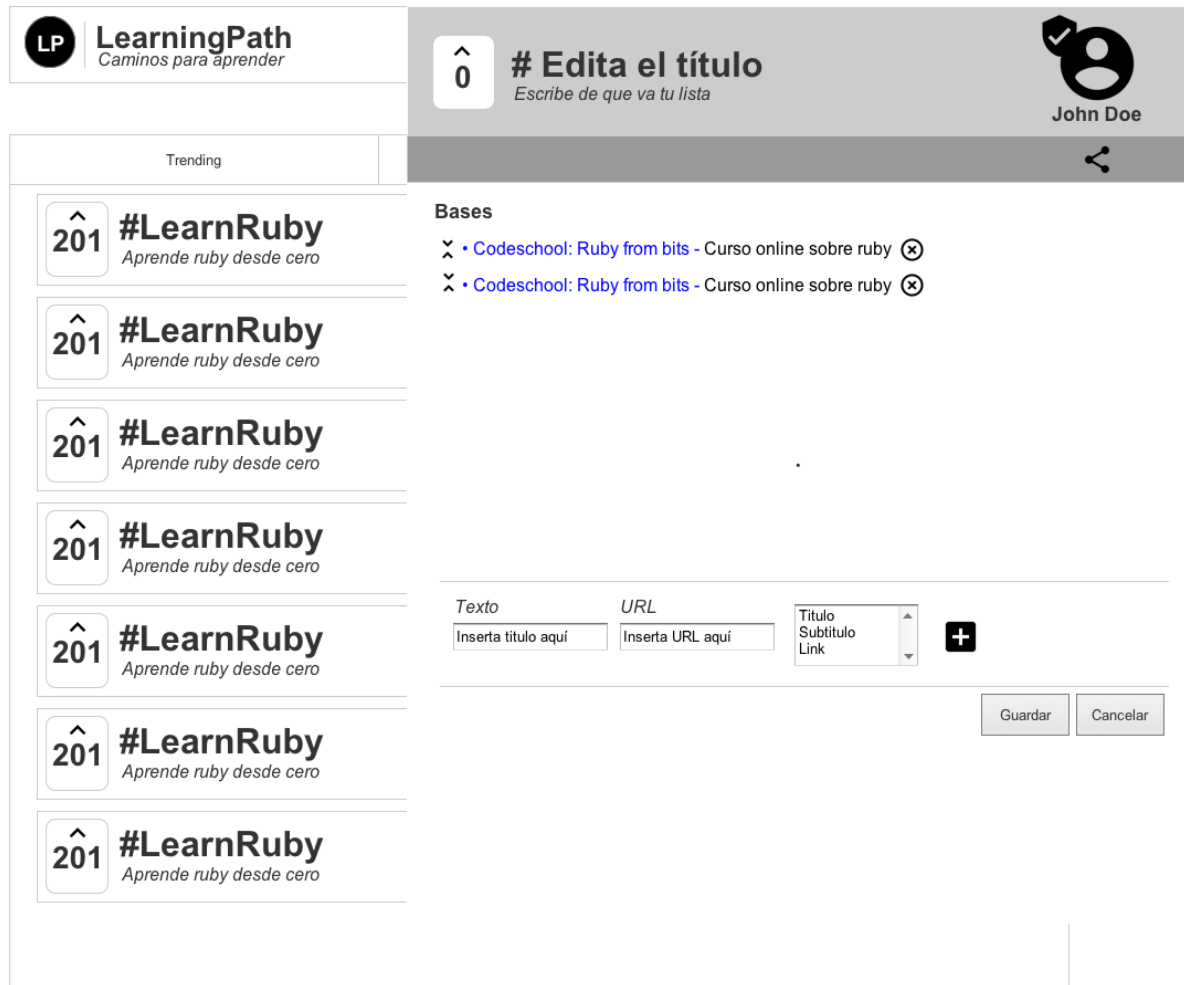


Figura 11. Prototipo de baja resolución: Sección crear y editar path

En esta sección se dispone de dos modos, la visual en la que el usuario puede ver los path que ha creado así como el modo de edición.

El modo de edición que mostramos en el *wireframe*, habilita la posibilidad de cambiar el nombre, la descripción, añadir steps y modificarlos así como la opción de añadir tantos tags como sea conveniente.

Dentro de esta página existe la posibilidad de añadir un nuevo path tanto como de borrar cualquier path crea conveniente que exista con anterioridad.

### 9.1.4 Perfil: seguidores y siguiendo

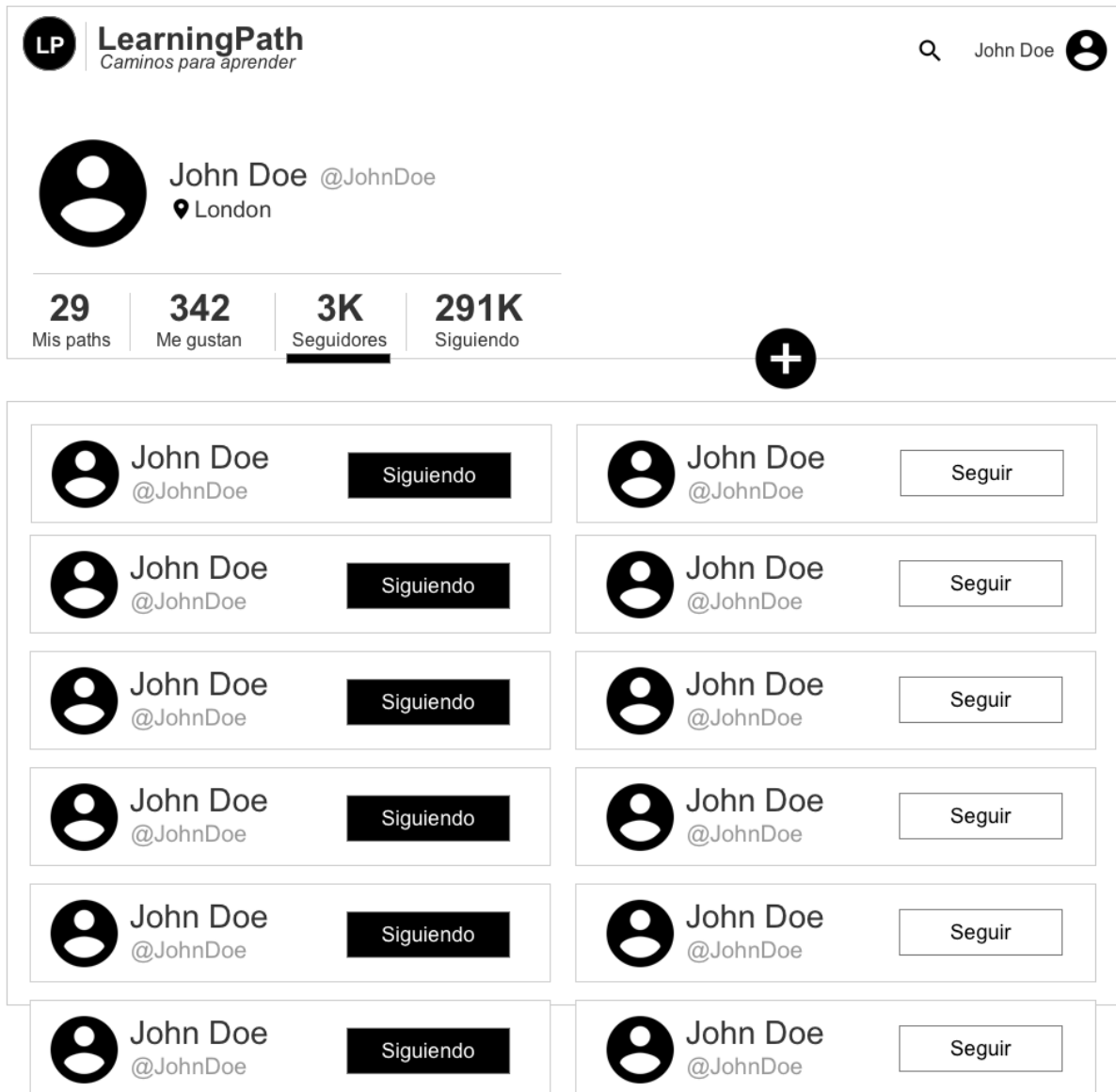


Figura 12. Prototipo de baja resolución: Sección seguidores y siguiendo

En esta sección disponemos de una lista de usuarios los cuales siguen al usuario principal, mostrando con facilidad la posibilidad de ver cual es el contenido que dichos usuarios han creado haciendo *click* sobre el nombre del usuario.

Existe la versión alternativa de la página en la que el usuario ve a los usuarios que está siguiendo, mostrando con facilidad la posibilidad de acceder al contenido generado por dichos usuarios.

### 9.1.5 Perfil: Me gusta

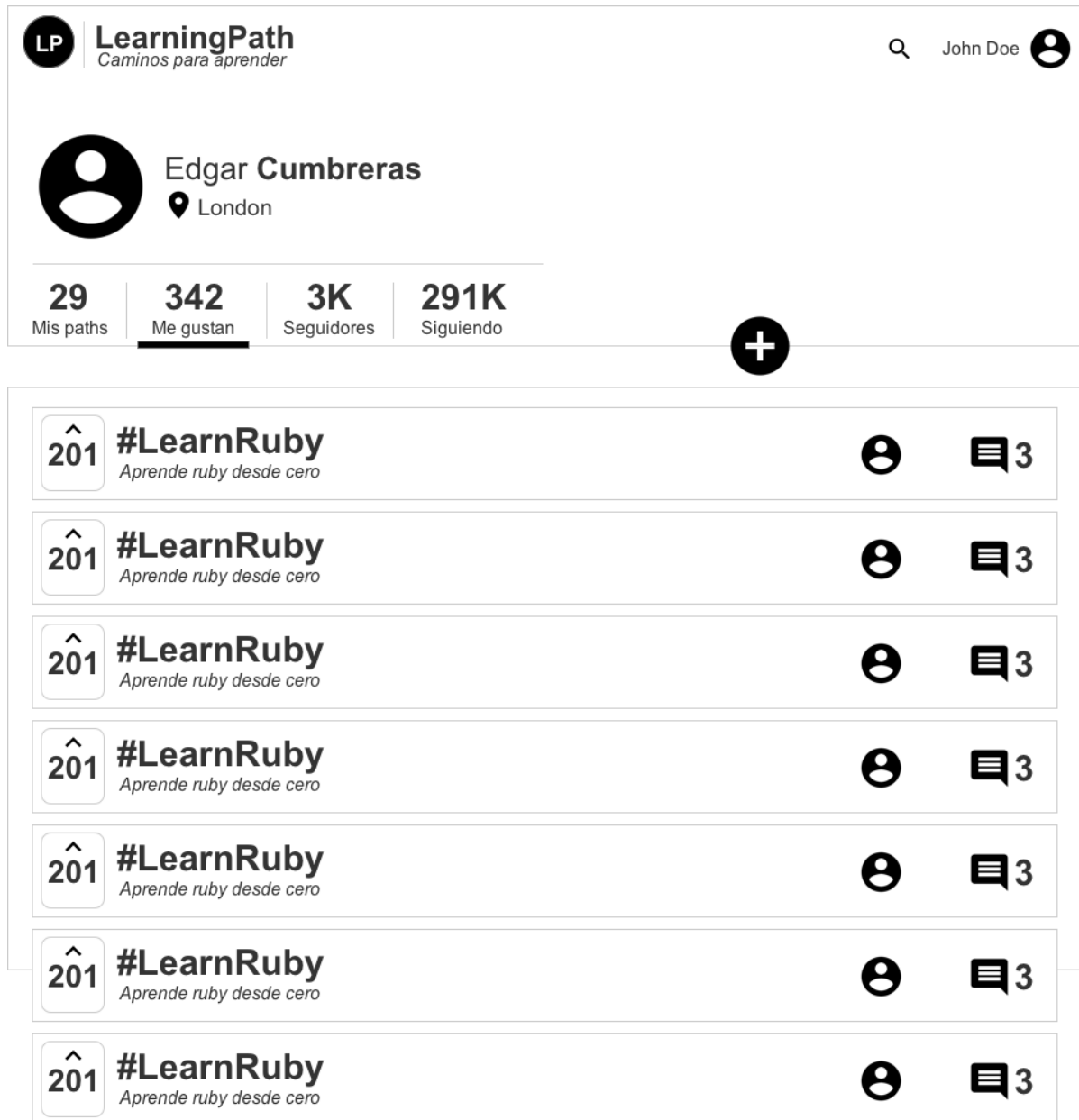


Figura 13. Prototipo de baja resolución: Sección thumb-ups

Esta sección muestra todos los paths que el usuario ha dado una valoración positiva, de esta manera puede usar esta funcionalidad para guardar todos aquellos paths que cree que le serán de ayuda o está usando como guía en el momento.

Esta funcionalidad es privada y otro usuario no puede ver los path que otro usuario haya añadido a dicha lista.

### 9.1.6 Paths: perfil

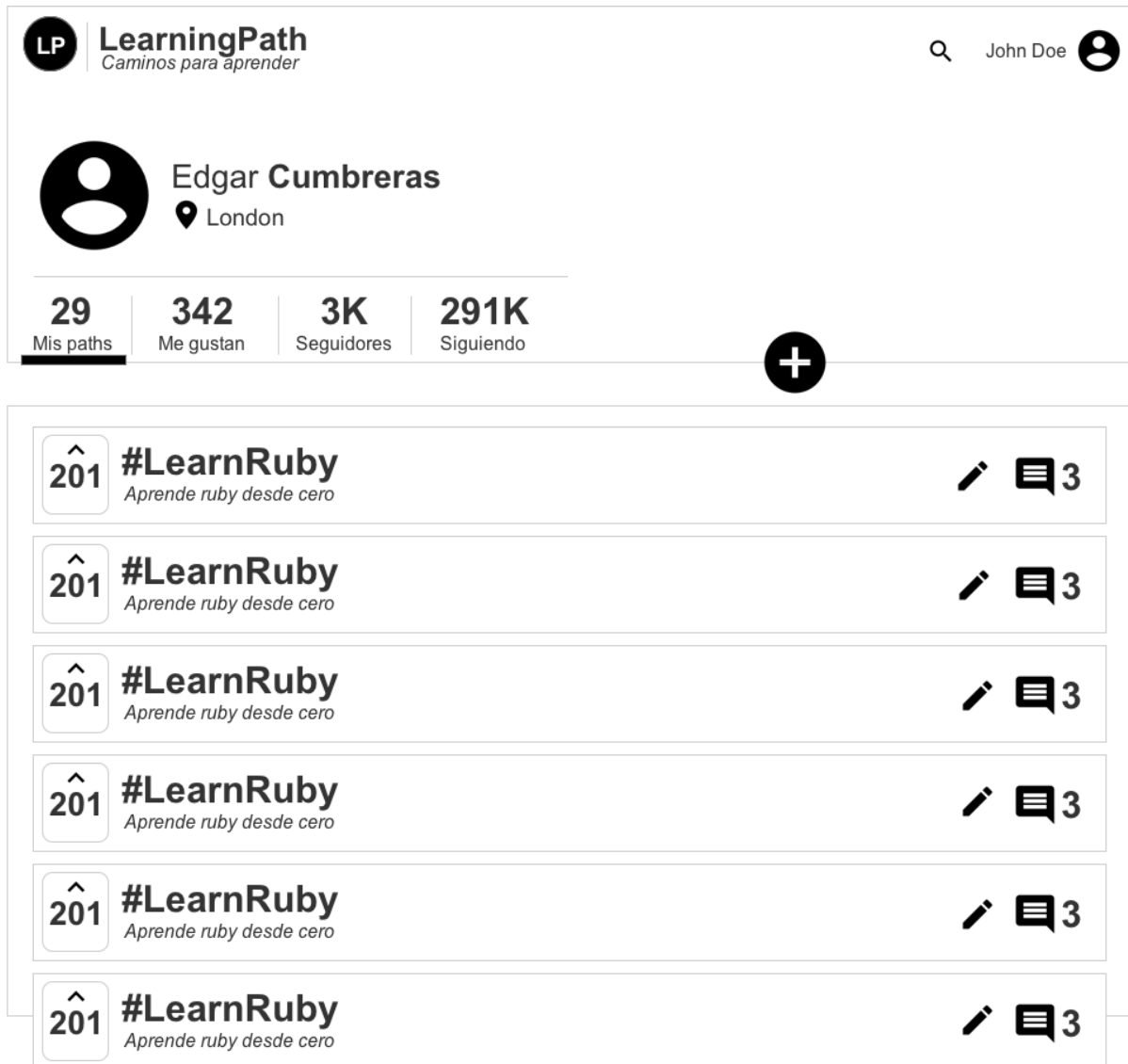


Figura 14. Prototipo de baja resolución: Sección perfil

Esta sección es la principal del usuario, en la cual podrá ver todos los path que haya generado, siendo una vista complementaria al punto 9.1.3, mostrando el modo visual.

El modo de edición será accionado cuando el usuario haga *click* en el icono con el lápiz.

### 9.1.7 Opciones y ajustes

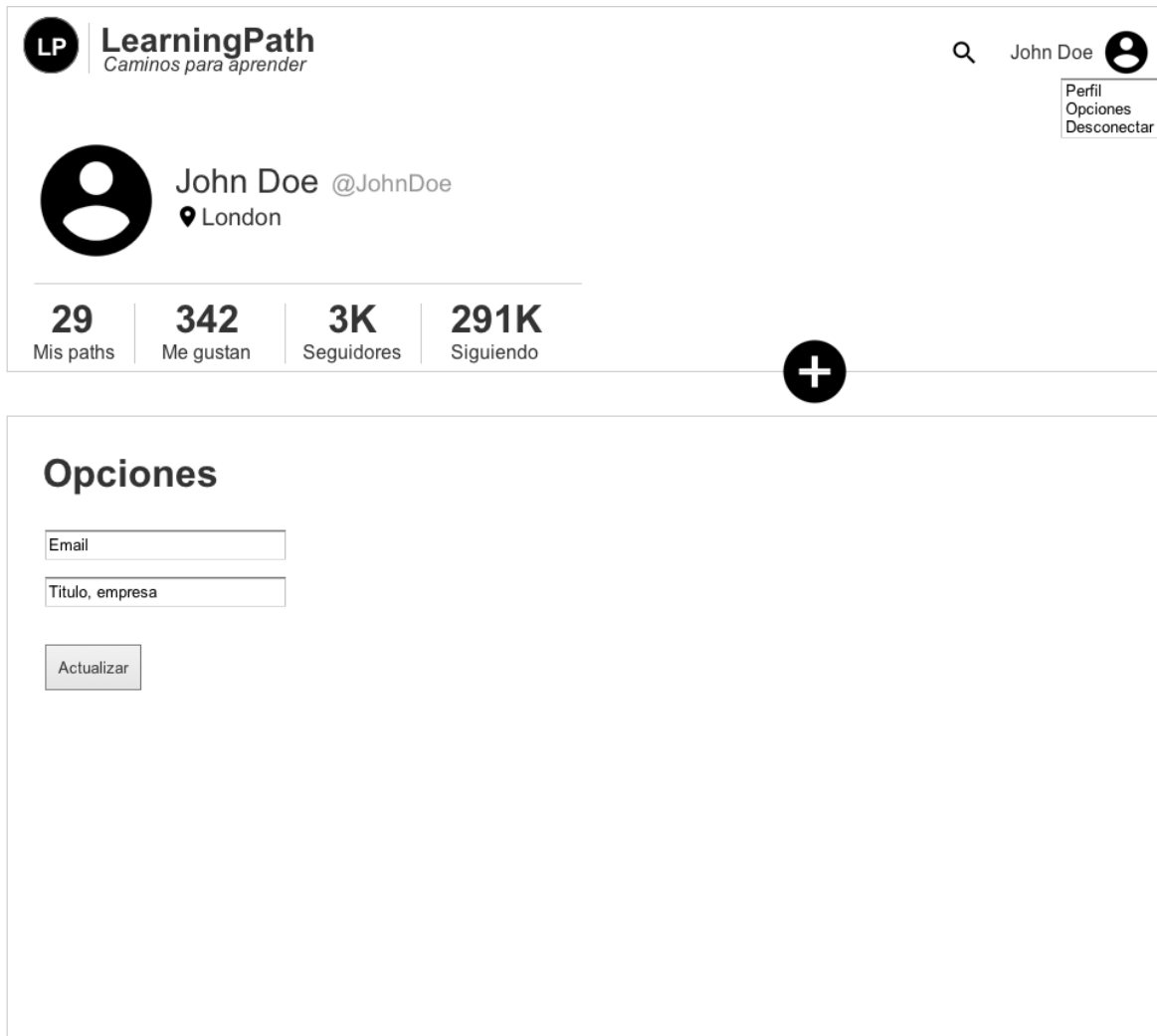


Figura 15. Prototipo de baja resolución: Sección opciones y ajustes

En esta sección disponemos acceso a los ajustes, configuración que el usuario dispone.

Para poder acceder a todas estas secciones el usuario ha de identificarse, así como para poder modificarlas es necesario que introduzca la contraseña en modo de seguridad.

El usuario también puede subir un nuevo avatar desde esta sección.

## 9.2 Prototipo: Alta resolución

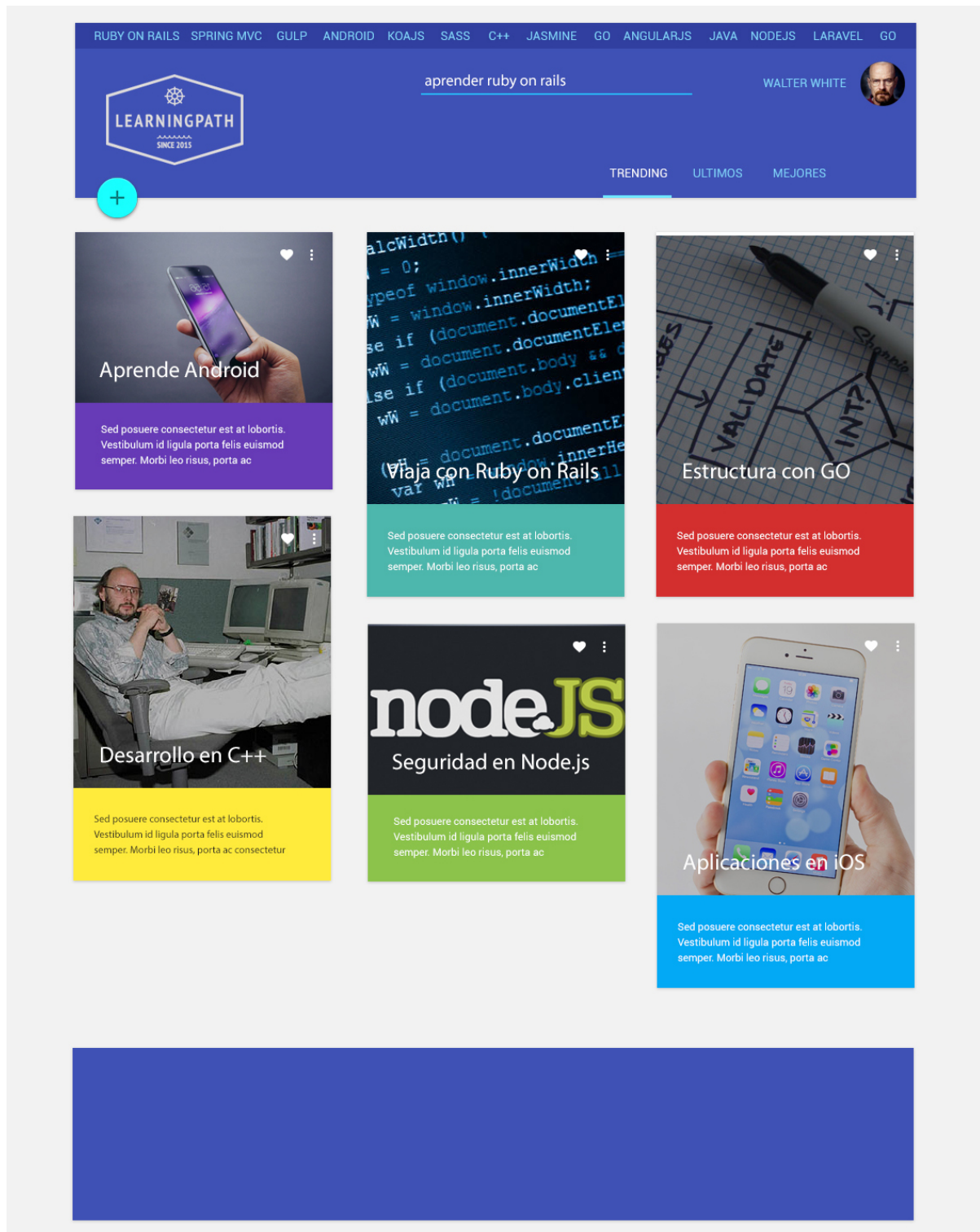


Figura 16. Prototipo de alta resolución: Sección home



## **10. Perfiles de usuarios**

La plataforma se compone principalmente de dos tipos de usuarios, los usuarios que buscan aprender una nueva tecnología y aquellos que conocen una tecnología y quieren compartir el proceso de aprendizaje de la misma.

### **10.1 Usuarios que busquen aprender una nueva tecnología**

La plataforma dispondrá de un perfil de usuario que busca aprender una nueva tecnología. Este usuario hará uso de la plataforma para buscar guía y recursos para aprender la nueva tecnología.

La interacción de estos usuarios con la plataforma será principalmente buscando información, generando comentarios, y aportando “me gusta” sobre las listas que más interés o beneficio les haya proporcionado.

No es viable conocer el nivel o tipo de educación que dispondrán todos los usuarios con este perfil, si que es posible adelantar que la mayoría serán usuarios relacionados con el desarrollo de software.

Este tipo de usuarios dispone de un conocimiento suficiente para interactuar en la red social con libertad sin la necesidad de un asesoramiento prioritario. Sin embargo podríamos disponer de especial atención y seguimiento en posibles dudas que se generen al momento del uso de la plataforma.

### **10.2 Usuarios que conozcan una nueva tecnología**

La plataforma se completa con este segundo tipo de perfil de usuario. Este usuario efectuará un uso focalizado en la creación de contenido. Publicará listas sobre conocimientos que disponga o metodologías que ha usado para obtener algún tipo de conocimiento.

Este tipo de usuarios tendrá una interacción más profunda y supondrá la base

sostenible de la red social al ser el perfil creador de contenidos.

Podemos vaticinar que la mayoría de los usuarios de este tipo dispondrán de experiencia en cualquiera de las disciplinas relacionadas con el desarrollo de *software* y *web*.

La capacidad de interacción de este tipo de usuarios no necesitará de apoyo para poder hacer uso del ciento por ciento de las características que la plataforma ofrece.

Para la aplicación disponemos de un caso de uso que será el más común en el cual encontraremos a un usuario (A) que es experto en un cierto campo relacionado con la tecnología o la ciencia creando un path mostrando la manera que cree oportuna para aprender una tecnología determinada, posteriormente dicho usuario compartirá dicho path con otros usuarios mediante cualquier red social.

En el mismo caso de uso dispondríamos de otro usuario menos experimentado (B) o totalmente desconocedor de dicha tecnología buscando un path que le pueda guiar como aprender la misma tecnología que el otro usuario haya mostrado.

El usuario (B) visitaría el path y podría crear un comentario indicando el nivel de utilidad que ha encontrado en dicho path. El usuario (B) al encontrar más útil contenido creado por el usuario (A) decide de seguirle, generando una posibilidad de asociación mutua, es decir, el usuario (A) también podría seguir al usuario (B). Por último al disponer de un resultado positivo usando (B) decide votar positivamente mediante un thumbs-up el path creado por el usuario (A).

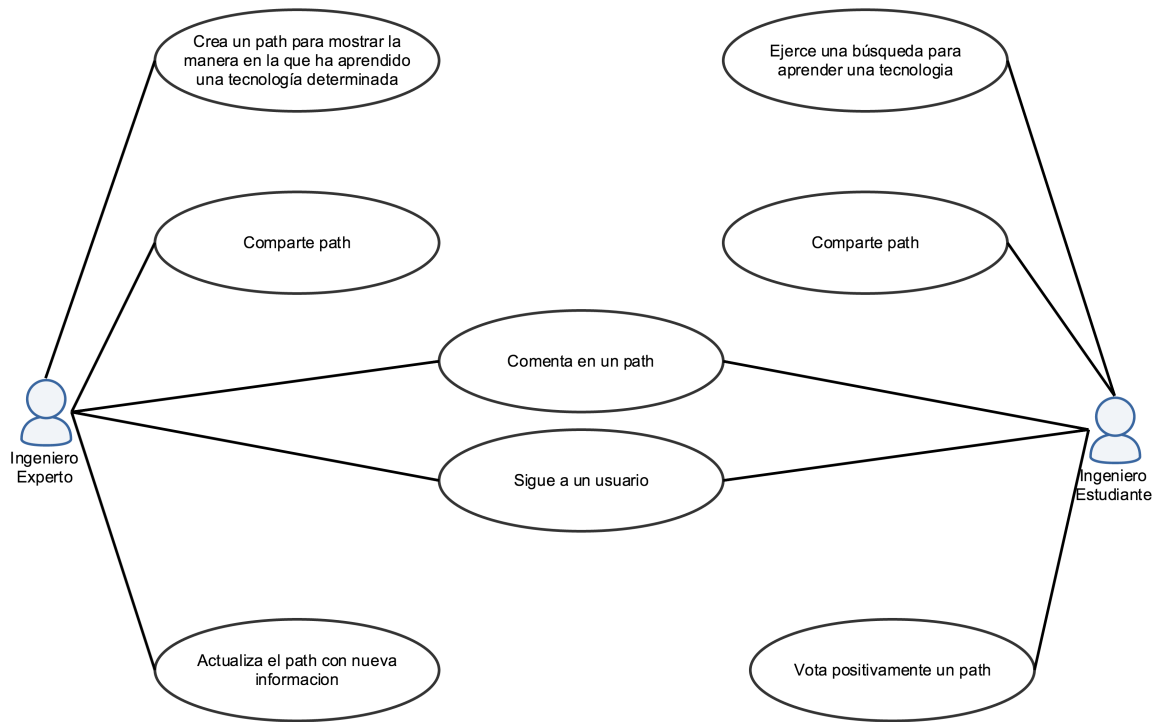


Figura 17. Caso de uso primario

## 11. Arquitectura de la aplicación

El desarrollo de la aplicación ha sido centrado en el uso de una arquitectura basada en el uso de JavaScript en todas las capas de la misma.

En el lado del cliente disponemos de AngularJS, un *framework* basado en la arquitectura MVC, esta capa de la aplicación es la encargada de gestionar la interacción del cliente con la aplicación; disponiendo de la lógica necesaria para mostrar y gestionar la interfaz así como el mecanismo para ejecutar la comunicación con el lado del servidor. En este lado del cliente, también disponemos de un *framework* visual como es Bootstrap, facilitando herramientas que aceleran el desarrollo de la interfaz.

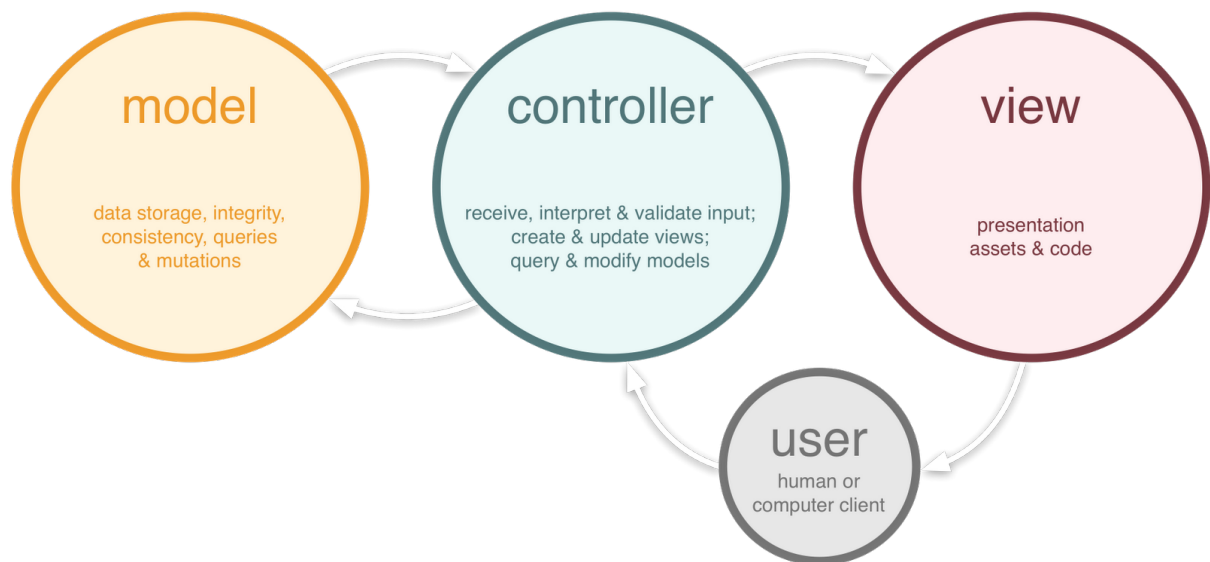


Figura 18. Arquitectura MVC

En la capa del servidor disponemos de Express.js, un *framework* de JavaScript basado en Node.js, que basado en middleware dispone de una REST<sup>13</sup> API para gestionar las solicitudes y las respuestas que el cliente, en nuestro caso la aplicación creada en AngularJS generará.

---

13 [HTTP://ES.WIKIPEDIA.ORG/WIKI/REPRESENTATIONAL\\_STATE\\_TRANSFER](http://es.wikipedia.org/wiki/Representational_State_Transfer)

La capa de persistencia de la aplicación dispone de una base de datos no relacional, el concepto de este tipo de arquitectura de base de datos es guardar la información en estructura de documentos de tipo JSON, es decir, en vez de guardar los datos en tablas como hacen las bases de datos relacionales, este tipo de arquitectura guarda los objetos con la anotación literal de JavaScript.

Como podemos ver, disponemos de JavaScript en el lado del servidor y en el lado del cliente, haciendo que el conocimiento necesario para poder aplicar cambios o mejoras en la aplicación sea más factible ya que abre todo el *stack* para cualquier conocedor del lenguaje.

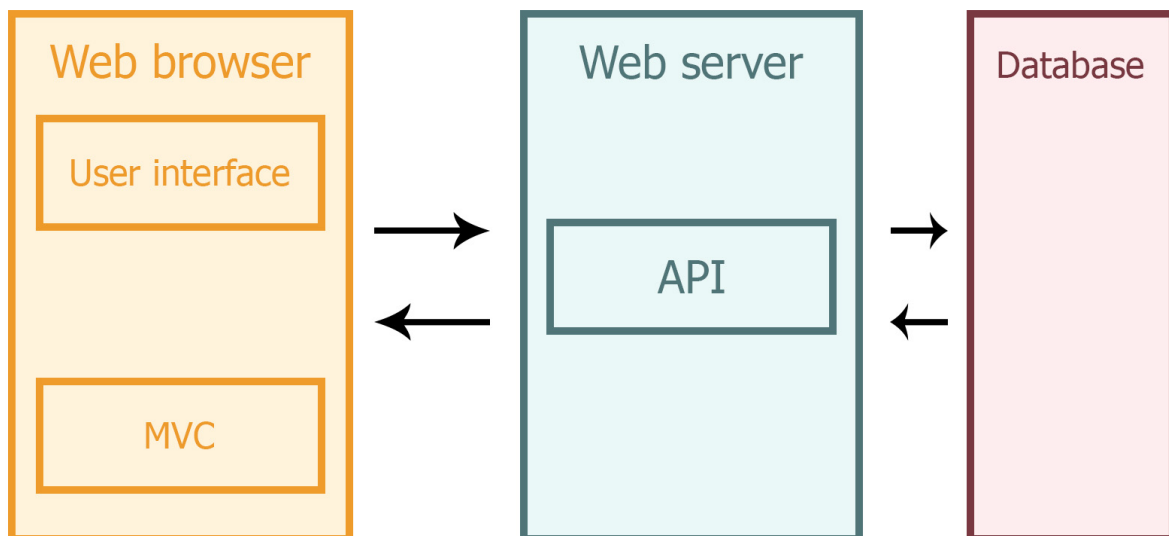


Figura 19. Diagrama de la arquitectura web

## **12. Proceso de desarrollo**

El proceso de desarrollo usado para la creación de esta aplicación está basado en *Extreme Programming*. Esta metodología de desarrollo de *software* se basa en el desarrollo incremental, haciendo que la creación de la plataforma haya sido un proceso incremental, publicando pequeñas partes de la aplicación.

El proceso de desarrollo que nos ofrece esta metodología es el siguiente:

Antes de que comiencen las iteraciones es necesario disponer de una fase de planificación, la cual genera el plan, las historias de usuarios y los requerimientos deseados. Una vez se disponen de las historias de usuarios, se generan múltiples iteraciones que finalizarán entregando versiones del producto con funcionalidad final. Es necesario planificación de la iteración, así como los elementos necesarios por parte de terceros.

Una vez se haya finalizado esta primera fase, se comienza el proceso cíclico de desarrollo que se compone de las siguientes fases:

1. **Diseño:** Se procede a generar todos los diseños y material relativo al mismo necesarios prioritariamente al desarrollo del código.
2. **Programación:** Esta es la fase que constituye la columna vertebral del proceso. Durante esta fase se desarrolla la sección de la iteración que sea pertinente.
3. **Testeo:** La última fase de la sección iterativa es la encargada de integrar las nuevas características con las antiguas así como hacer los cambios necesarios si son aplicables en la base de datos, o en cualquier otro elemento que sea necesario para la integración de la nueva sección. Pruebas automatizadas y manuales son llevadas a cabo en esta fase.
4. **Escucha:** Esta fase es la fase entre iteraciones, la cual es llevada a cabo tras

el *feedback* de cualquiera de los *stackholder*<sup>14</sup> o usuarios que están probando la aplicación. Esta retroalimentación será parte importante de la aceptación de esta iteración así como la sugerencia de cambios que sea necesario aplicar en próximas iteraciones.

## **12.1 Estructura de la aplicación**

La estructura de la aplicación dispone principalmente de dos secciones: *client* y *server*. Como se puede prever una carpeta es la que contiene todo el material para el cliente con los archivos de la aplicación de Angular, las vistas y las hojas de estilo. La carpeta del server dispone de todo lo necesario para hacer funcionar el servidor web así como las configuraciones que irán en el servidor en sí, tales como las claves de acceso para las API de Facebook, Twitter y Google plus.

## **12.2 Secciones de mención**

Dentro de la aplicación existen ciertas secciones que creo que deberían de ser mencionadas en detalle ya que son partes fundamentales para la aplicación.

### **12.2.1 Configuración de MongoDB**

En la siguiente sección de código podemos ver la configuración inicial de MongoDB. En esta configuración abrimos una conexión con MongoDB haciendo posible la petición a la base de datos sin tener que hacer múltiples peticiones de acceso.

```
app.use(session({
  secret: config.secrets.session,
  resave: true,
  saveUninitialized: true,
  store: new mongoStore({ mongoose_connection: mongoose.connection
}))
```

---

<sup>14</sup> [HTTP://ES.WIKIPEDIA.ORG/WIKI/STACKHOLDER](http://es.wikipedia.org/wiki/Stackholder)

### **12.2.2 Configuración del middleware**

En la siguiente sección de código podemos ver la configuración inicial de MongoDB así como el *middleware*<sup>15</sup> usado en la aplicación.

Entre algunos de los utilizados son el encargado de servir Jade como vistas, pudiendo hacer uso de este lenguaje de *templating* que es compilado a HTML. Un *middleware* para gestionar las galletas, los datos en las peticiones transformándolos directamente a JSON. En último lugar inicializamos passport, que será el encargado de gestionar el registro y la autenticación de usuarios.

```
module.exports = function(app) {  
  var env = app.get('env');  
  
  app.set('views', config.root + '/server/views');  
  app.set('view engine', 'jade');  
  app.use(compression());  
  app.use(bodyParser.urlencoded({ extended: false }));  
  app.use(bodyParser.json());  
  app.use(methodOverride());  
  app.use(cookieParser());  
  app.use(passport.initialize());  
}
```

### **12.2.3 Servicio de AngularJS para Thumbs**

AngularJS dispone de un tipo de objetos llamados Objetos Servicio, estos están directamente relacionados con la lógica de poder encapsular una parte de la aplicación que pueda ser reusada, por ello mismo he hecho uso de este tipo de Objeto encapsular la lógica de los thumbs-up, esto es debido a la necesidad de reusar dicha acción en múltiples partes de la aplicación.

---

<sup>15</sup> [HTTP://ES.WIKIPEDIA.ORG/WIKI/MIDDLEWARE](http://es.wikipedia.org/wiki/Middleware)



```
thumbsUpPath: function thumbUpPath(path, currentUser) {
  var deferred = $q.defer();
  // Same user can't thumbsUp itself
  if(path._user._id === currentUser._id) {
    deferred.reject('error');
    return;
  }

  var thumbsUp = {
    _user: currentUser._id,
    _path: path._id
  };

  var canThumbsUp = true;
  _.each(currentUser.thumbsUps, function(thumbsUp) {
    if(thumbsUp._path === path._id) {
      canThumbsUp = false;
    }
  });

  if(canThumbsUp) {
    $http.post('/api/thumbsups/', thumbsUp)
      .success(function(thumbsUp) {
        }).then(function() {
          $http.post('/api/users/' + currentUser._id +
'/addThumbsUp', thumbsUp);
          $http.post('/api/paths/' + path._id + '/addThumbsUp',
thumbsUp);
          deferred.resolve(true);
        });
  }
};
```

## 13. Diagrama UML: clases

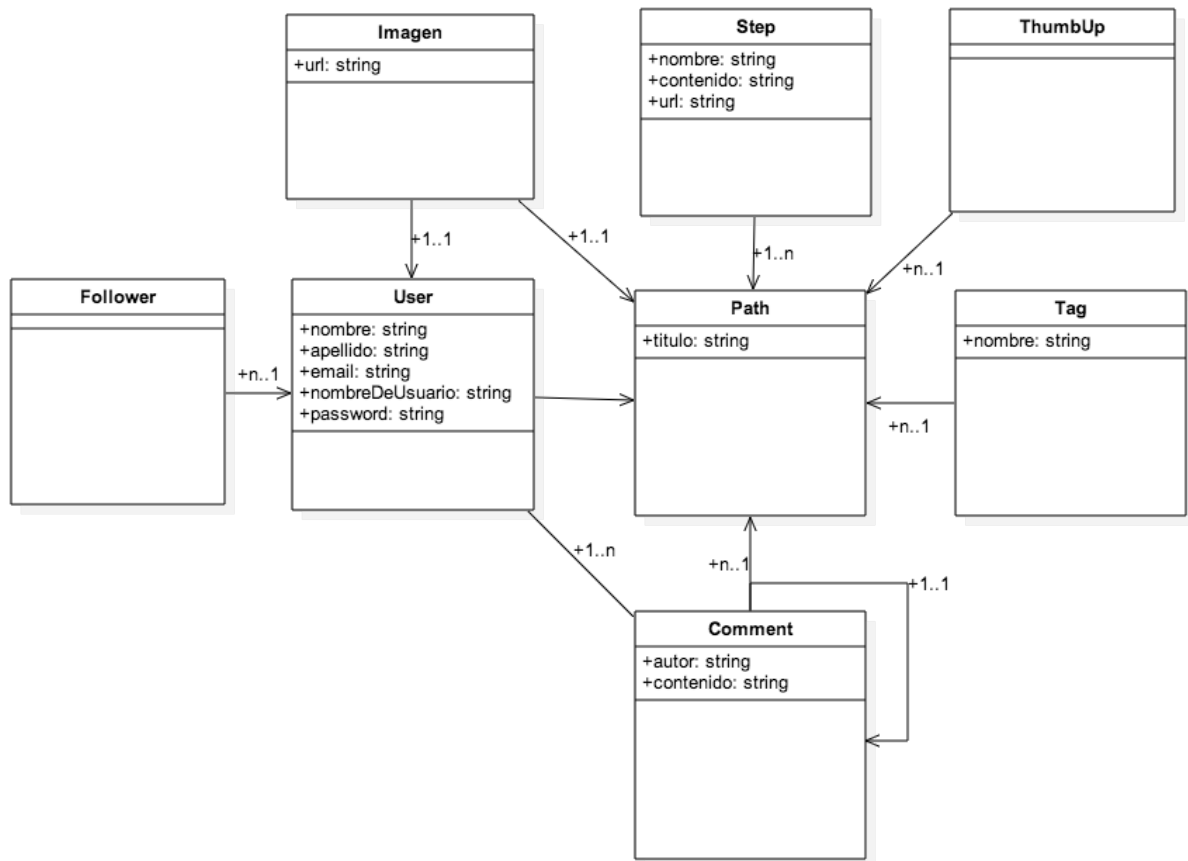


Figura 20. Diagrama de clases

En la aplicación disponemos de dos modelos primarios que son sobre los que se basan la mayoría de las acciones. Path y User disponen de una relación con la mayoría de las clases de la aplicación.

El uso de User es debido a que las acciones las ejecuta un usuario por lo tanto para que dichas acciones puedan ser asociadas con un perfil siempre hay que disponer de una relación con dicho modelo. Este modelo dispone de una relación directa con colección, seguidor, path, imagen y comentario.

La clase Path es usada a lo largo de la aplicación debido a que es la acción primaria focalizada de la aplicación, es decir, que las acciones como crear un comentario, añadir un path o modificar un tag, son todas creadas sobre el modelo Path.

## 14. APIs utilizadas

Para el registro y autenticación de usuarios en la plataforma, hemos hecho uso de la API de Facebook, Google y Twitter. Estas tres plataformas disponen de un protocolo de autenticación y autorización basado en el protocolo abierto OAuth<sup>16</sup>. Este protocolo permite la identificación de un usuario gracias al acceso del perfil del usuario en cualquiera de esas redes sociales, permitiendo que accedamos a información que dispone dicho usuario en su perfil.

Una vez disponemos de acceso a la información del usuario, generamos la autenticación o el registro del usuario en la red social haciendo uso de los datos que hemos sido autorizados como pueden ser el nombre completo, o el avatar que dispone el usuario.

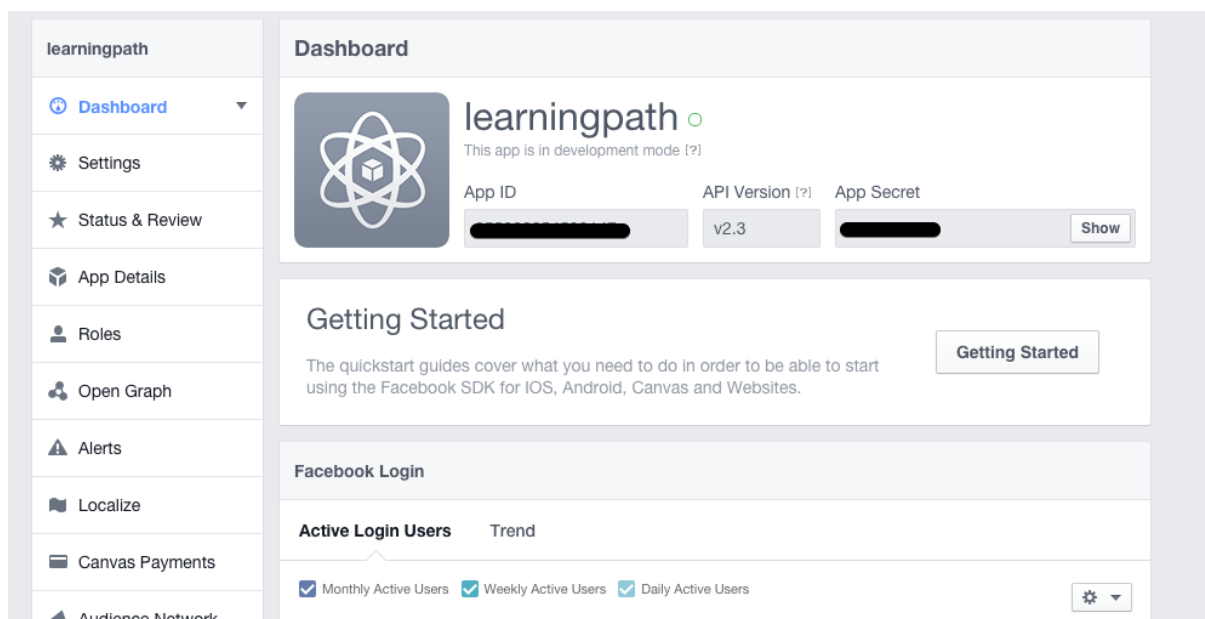


Figura 21. Aplicación en facebook developers

16 [HTTP://ES.WIKIPEDIA.ORG/WIKI/OAUTH](http://es.wikipedia.org/wiki/OAuth)

## **15. Plataforma de desarrollo**

El desarrollo de esta red social ha sido centralizada en dos ordenadores usando el sistema operativo OSX 10.9.5. El código ha sido alojado en Bitbucket.org como servidor de control de versiones Git<sup>17</sup>. El *software* de desarrollo en este caso ha sido Vim<sup>18</sup> en su versión MacVim para OSX.

El terminal es controlado desde la aplicación iTerm en su versión 2.0.0.20141103 usando Z-Shell (ZSH)<sup>19</sup> como Shell de Unix.

---

17 [HTTP://ES.WIKIPEDIA.ORG/WIKI/GIT](http://es.wikipedia.org/wiki/Git)

18 [HTTPS://GITHUB.COM/B4WINCKLER/MACVIM](https://github.com/b4winckler/macvim)

19 [HTTP://ES.WIKIPEDIA.ORG/WIKI/ZSH](http://es.wikipedia.org/wiki/Zsh)

## **16. Seguridad**

El desarrollo de cualquiera aplicación web dispone de ciertas tareas que son la espina vertebral de la plataforma. Seguridad y la prevención de ataques suele estar el epicentro de estas tareas debido a su importancia, ya que con el crecimiento exponencial del uso de la red también lo hacen las amenazas con las que las plataformas tienen que lidiar día a día.

Por ello mismo nosotros hemos dispuesto una serie de medidas que puedan paliar y mitigar estos posibles ataques que puedan abrir acceso al servidor en el que se encuentra alojada la plataforma y con el consecuente acceso a los datos personales de los usuarios.

Nuestra plataforma al no disponer de servicios, ni pasarelas de pago, no dispondrá de un certificado SSL<sup>20</sup> sin embargo si que centrará su atención en los puntos destacados de los certificados más habituales publicados por la Organización internacional para la Estandarización<sup>21</sup> (ISO) y la Comisión Electrotécnica Internacional<sup>22</sup> (IEC).

La aplicación ha sido desarrollada con un sistema de gestión de la seguridad de la información<sup>23</sup> (SGSI), este sistema de gestión es un proceso de gestión el cual se centraliza en gestionar eficientemente la accesibilidad de la información minimizando los riesgos en la circulación de la información. Este sistema dispone de un enfoque Planificar-Hacer-Revisar-Actuar, *Plan-Do-Check-Act* (PDCA), el cual se focaliza en cuatro puntos fases:

- **Planificar:** Evaluación de los riesgos así como selección de controles que atestigüen los riesgos.
- **Hacer:** Implementación del sistema y de los controles.

---

20 [HTTP://ES.WIKIPEDIA.ORG/WIKI/TRANSPORT\\_LAYER\\_SECURITY](http://es.wikipedia.org/wiki/Transport_Layer_Security)

21 [HTTP://ES.WIKIPEDIA.ORG/WIKI/ORGANIZACIÓN\\_INTERNACIONAL\\_DE\\_NORMALIZACIÓN](http://es.wikipedia.org/wiki/Organización_Internacional_de_Normalización)

22 [HTTP://ES.WIKIPEDIA.ORG/WIKI/COMISIÓN\\_ELECTROTÉCNICA\\_INTERNACIONAL](http://es.wikipedia.org/wiki/Comisión_Electrotécnica_Internacional)

23 [HTTP://ES.WIKIPEDIA.ORG/WIKI/SISTEMA\\_DE\\_GESTIÓN\\_DE\\_LA\\_SEGURIDAD\\_DE\\_LA\\_INFORMACIÓN](http://es.wikipedia.org/wiki/Sistema_de_Gestión_de_la_Seguridad_de_la_Información)

- **Controlar:** Revisar y evaluar los trabajos desempeñados
- **Actuar:** Implementar los cambios necesarios para disponer el SGSI a máximo rendimiento.

Hemos implementado ciertas medidas de seguridad estándares las cuales tienen el objetivo de darnos cierta robustez sobre ataques.

El desarrollo de esta aplicación ha estado focalizado en ciertos elementos característicos de JavaScript y de Node.js que tienden a crear vulnerabilidades.

El más común de los ataques a prevenirse es el llamado *Cross-site Scripting* (XSS), estos ataques son generados mediante inyecciones de código directamente en cualquier sección del sitio web al que el atacante tenga acceso. Este tipo de ataques es comúnmente generado a través de formularios o zonas de comentarios. La mejor manera de evitar este tipo de ataques es disponer de una metodología de saneamiento para todas las interacciones de contenido, no aceptando los datos de una fuente externa sin que hayan previamente sido validados.

El segundo ataque en el que hemos focalizado esfuerzos por mitigar es la falsificación de petición en sitios cruzados<sup>24</sup>, *Cross-Site Request Forgery* (CSRF). Este ataque fuerza al usuario a ejecutar unas acciones no deseadas en una plataforma en la que se encuentra actualmente autenticado. Esto ocurre porque las galletas se envían por cada petición inclusive si esas peticiones vienen de un sitio diferente. Express.js dispone de un *middleware* específico para evitar este tipo de ataques generando *tokens* en la primera petición *GET* que se crea al visitar el sitio, así se puede comprobar que las peticiones sucesivas provienen del sitio de origen.

JavaScript al ser un lenguaje de scripting dispone de ciertas vulnerabilidades al disponer de la posibilidad de evaluar e interpretar en tiempo real valores que hayan sido introducidos por el usuario, por ello mismo es necesario disponer de ciertas medidas de seguridad por defecto:

---

<sup>24</sup> CSRF: [HTTP://ES.WIKIPEDIA.ORG/WIKI/CROSS\\_SITE\\_REQUEST\\_FORGERY](http://es.wikipedia.org/wiki/Cross_Site_Request_Forgery)

Evitar hacer uso de la funcionalidad *eval*, esta propiedad de JavaScript abre la puerta a la interpretación en tiempo real de cadenas de texto las cuales pueden ataques por inyección.

Usando la bandera del modo estricto en JavaScript evita que se puedan generar ciertas acciones desde acceso en el navegador, como borrar propiedades de los objetos base, hace que las propiedades sean únicas, evita variables globales inintencionadas así como la congelación de los argumentos en las funciones. Este tipo de ayudas harán que nuestra aplicación sea más robusta así como evitará el concepto de errores silenciosos que el lenguaje dispone, en este caso los errores serán mostrados en vez de ser ignorados.

## **16.1 Proceso de identificación**

La identificación de los usuarios está ejecutada desde el lado del cliente, se utiliza el email y la contraseña introducida por el usuario para hacer una petición al servidor

```
$http.post('/auth/local', {  
  email: user.email,  
  password: user.password  
}).
```

Como podemos ver en el ejemplo de código es una petición post a /auth/local siendo un endpoint que he creado en el servidor.

```
router.post('/', function(req, res, next) {  
  passport.authenticate('local', function (err, user, info) {  
    var error = err || info;  
    if (error) return res.json(401, error);  
    if (!user) return res.json(404, {message: 'Something went wrong,  
please try again.'});
```

En este punto se verifica la contraseña haciendo uso del método que nos ofrece el modulo Passport. Este módulo verifica la contraseña que ha introducido y la que disponemos en el servidor descriptada gracias a un proceso de *de-hashificación*<sup>25</sup>.

Una vez se verifica si el proceso es satisfactorio se devuelve un *token* que será asociado en forma de *cookie* en el lado del cliente añadiendo dicha cookie al navegador del usuario, de esta manera el navegador recordará que el usuario está identificado haciendo uso de dicha galleta para identificar las peticiones y acciones que el usuario hará durante su visita al sitio web.

```
success(function(data) {
    $cookieStore.put('token', data.token);
    currentUser = User.get();
    deferred.resolve(data);
    return cb();
}).
error(function(err) {
    this.logout();
    deferred.reject(err);
    return cb(err);
}).bind(this));

return deferred.promise;
},
```

---

<sup>25</sup> AUTENTICACIÓN E IDENTIFICACIÓN: [HTTPS://CRACKSTATION.NET/HASHING-SECURITY.HTM](https://crackstation.net/hashing-security.htm)



## 17. Tests

Como parte de la fase de desarrollo y diseño de la plataforma se han efectuado diferentes pruebas las cuales tienen objetivo de asegurar la correcta funcionalidad, un rendimiento adecuado focalizado en la carga de usuarios y en la escalabilidad de la plataforma.

### 17.2 Test de rendimiento

Para concluir el test de la aplicación nos es necesario ver la carga que dispone la aplicación cuando está en producción ya que el servidor real de la aplicación dispone de mucha más latencia que la carga cuando la aplicación está siendo ejecutada por el servidor en red local. Este tipo de carga es directamente relacionada con la cantidad de elementos que tenga de dependencia, disponibilidad del servidor y la manera en la que el proceso renderización la página web sumando la cantidad de peticiones tenga que hacer para cargar los datos necesarios.

Como podemos ver en el siguiente gráfico, la aplicación dispone de una carga normal, disponiendo de 14 peticiones con una carga de 332KB transferidos. El tiempo finalizado para cargar los elementos necesarios para mostrar la aplicación es de 1.50s

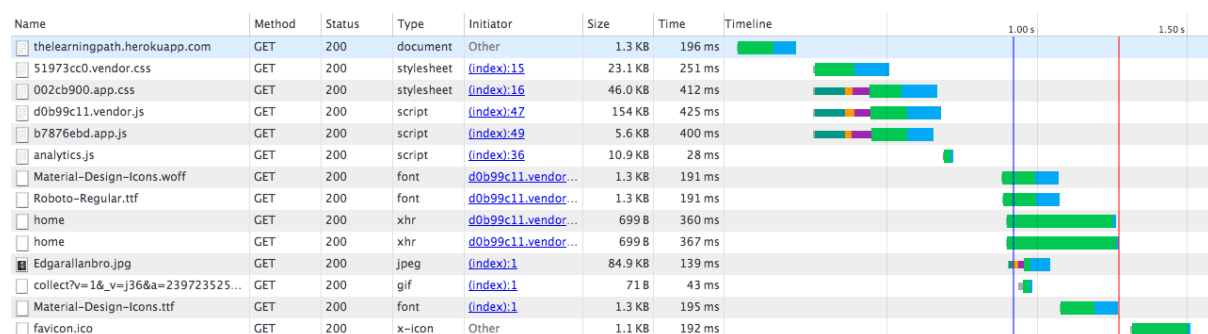


Figura 22. Profiling de la carga de archivos

## 17.1 Análisis del stack en carga

El concepto de la arquitectura de eventos está centrado en una acción ejecutará un evento y ese evento ejecutará otra acción, dependiendo de la manera en la que haya sido desarrollada esto puede determinar siendo un bucle ya que dicho receptor del evento puede ejecutar otro evento que haga ejecutar el evento del emisor, haciendo que se repita la acción indefinidamente, con la posible creación de común conocido *stack overflow*<sup>26</sup> o desbordamiento de pila, causando que la aplicación se interrumpa por completo.

Como podemos ver en el siguiente gráfico, disponemos de una carga normal del *stack*, inclusive disponiendo de una alta carga de eventos al momento de cargar el sitio web. Este tipo de gráficos que se disipan volumen hasta llegar a cero muestran que la aplicación no dispone de ningún problema de recursión, pérdida o bloqueo de memoria.

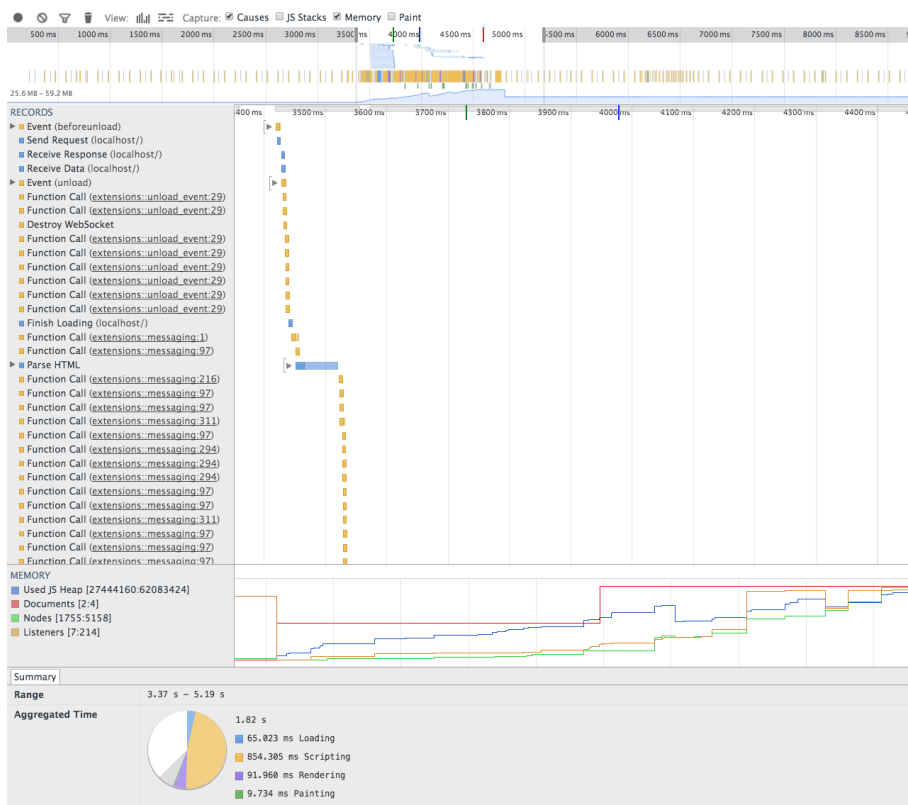


Figura 23. Profiling del stack de la aplicación

26 STACK OVERFLOW: [HTTP://EN.WIKIPEDIA.ORG/WIKI/STACK\\_OVERFLOW](http://en.wikipedia.org/wiki/Stack_overflow)

## **18. Proceso de pruebas**

Para poder asegurar la estabilidad técnica de la plataforma, habilitando una fácil implementación de nuevas características, mejoras de las actuales así como solventar posibles bugs y errores que aparezcan es necesario disponer de un banco de pruebas que asegure un estado óptimo del código fuente. Este tipo de banco de pruebas se compone de una serie de test unitarios en los que comprueba que la funcionalidad del programa es la correcta. Verificando estados, acciones y casos de uso estos test son ejecutados continuamente durante el proceso de desarrollo.

En nuestra plataforma, también disponemos de test de integración, estos test de integración son focalizados para poder comprobar que la sección del cliente funciona correctamente sin tener que hacer las comprobaciones de manera manual.

Esta plataforma ha sido creada usando un desarrollo guiado por pruebas de *software* o *test driven development* (TDD)<sup>27</sup>, este tipo de desarrollo involucra un enfoque en el que todo el código generado es precedido por unas pruebas que asegurarán que dicho código cumple el objetivo deseado.

Para la plataforma, disponemos de múltiples bancos de pruebas que se enfocan en el lado del cliente y el lado del servidor.

El lado del cliente se compone de una sección de test unitarios y de integración, mientras que el lado del servidor se compone de una serie de test unitarios.

---

<sup>27</sup> [HTTP://ES.WIKIPEDIA.ORG/WIKI/DESARROLLO\\_GUIADO\\_POR\\_PRUEBAS](http://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas)

## 19. Bugs

El proceso iterativo de trabajo, dividido en *Sprints*, nos ha ayudado a hacer múltiples revisiones del código, así como el desarrollo usando TDD, nos ha identificado múltiples bugs que han sido introducidos tras el desarrollo e implementación de otra sección de la aplicación.

### 19.1 Listado de bugs

A continuación mostramos los bugs que han sido solucionados desde la primera versión alfa de la aplicación hasta el momento de la entrega, así como el estado de los mismos.

Descripción	Estado
Angular router no identifica los parámetros por URL.	Resuelto
Grunt no pre-compila el SaSS a CSS.	Resuelto
Passport está usando OAuth 1.0 necesaria actualización y re-configuración del sistema.	Resuelto
Los modelos no aceptan sub-documentos insertados directamente en MongoDB.	Resuelto
La relación del modelo User con el modelo Path no permite guardar el documento.	Resuelto
Duplicados en el modelo Step cuando se guarda un modelo Path.	Resuelto
ThumbsUps no asociados directamente a su Path.	Resuelto
Usuario no puede hacer <i>login</i> , sesiones corruptas.	Resuelto
Imágenes no son públicas en el <i>bucket</i> de S3.	Resuelto
Imágenes corruptas tras envío de paquetes como <i>stream</i> .	Resuelto
Múltiples duplicados en comentarios tras una recarga de página.	Resuelto
El menú de tabs (top, trending y most-commented) no funciona en cargas aleatorias	Pendiente
Avatar no se muestra cuando se registra un usuario usando su cuenta de twitter	Pendiente

Tabla 2. Lista de bugs

## **20. Versiones de la aplicación**

### **20.1 PAC 2**

0.0.1: Instalación básica del sistema, configuración del servidor Express.js así como AngularJS.

0.0.2: Implementación del CRUD para modelo User, así como el sistema OpenGraph y OAuth para identificación y registro mediante redes sociales (Google, Facebook y Twitter).

0.0.3: Implementación del CRUD<sup>28</sup> para el modelo Path.

### **20.2 PAC 3**

0.0.4: Implementación del CRUD para el modelo Step y su relación con el modelo Path.

0.0.5: Implementación del CRUD para el modelo Comments, creando relación con Path y User.

0.1.0: Refactorización de todo visual e implementación de un nuevo sistema visual mediante Materialize, primera exposición de la plataforma en abierto.

<https://thelearningpath.herokuapp.com/>

### **20.3 Entrega final**

0.1.1: Implementación del sistema de votos, el modelo ThumbsUp y las relaciones pertinentes con el modelo Path y User.

0.1.2: Modularización del sistema, refactorización del Sistema de eventos.

0.1.3: Implementación del sistema de búsqueda.

0.1.4: Implementación de la versión responsiva para teléfonos y tablets.

0.1.5: Implementación del sistema de tags.

0.1.6: Adhesión de los tags al sistema de búsqueda.

0.1.7: Implementación del sistema de seguidores sobre el modelo User.

---

<sup>28</sup> CREATE, READ, UPDATE AND DESTROY (CREAR, BORRAR, ACTUALIZAR Y LEER)

## **21. Requisitos de instalación**

La aplicación está centralizada en el uso del *runtime*<sup>29</sup> de JavaScript que usa el navegador Chrome de Google. Este motor de JavaScript es la pieza central para la plataforma Node.js. Basado en el concepto de las aplicaciones isomorfas, esta aplicación hace uso de dicha plataforma para servir el servidor web que alimenta la aplicación.

La plataforma y servidor de base de datos es otro de los elementos necesarios para ejecutar el servidor web en modo local. Para esta aplicación será necesario disponer de la base de datos no relacional MongoDB así como de los procesos relativos a eventos que dispone la misma.

Es recomendable para el uso de esta plataforma hacerlo bajo cualquier sistema Unix<sup>30</sup>, ya que en la actualidad se han hecho uso de ciertas versiones de Node.js que no están disponibles en cualquier plataforma Windows.

El servidor deberá de disponer de una máquina con unos requerimientos mínimos:

- 700 MHz procesador (Intel Celeron o superior)
- 512 MiB RAM
- 1 GB espacio de disco duro.
- Video VGA con una resolución gráfica mínima de: 1024x768

### **21.1 Versionado**

Node.js: versión 0.12.2

Mongodb: versión 3.0.2

*\* Los requisitos de la aplicación principal serán instalados dependientemente del modulo de npm.*

---

<sup>29</sup> [HTTP://ES.WIKIPEDIA.ORG/WIKI/SISTEMA\\_EN\\_TIEMPO\\_DE\\_EJECUCI3N](http://es.wikipedia.org/wiki/Sistema_en_tiempo_de_ejecuci3n)  
<sup>30</sup> [HTTP://ES.WIKIPEDIA.ORG/WIKI/UNIX](http://es.wikipedia.org/wiki/Unix)

## 21.2 Dependencias estándares

Las dependencias de directas de la aplicación, son instaladas mediante el gestor de paquetes de Node.js. Este gestor de paquetes lee los paquetes necesarios para a la aplicación del archivo package.json. Este archivo es la configuración principal de la aplicación donde todos los cambios de requerimiento y versionados son guardados.

Aquí una sección del package.json de nuestra aplicación:

```
"dependencies": {  
  "aws-sdk": "^2.1.28",  
  "body-parser": "~1.5.0",  
  "composable-middleware": "^0.3.0",  
  "compression": "~1.0.1",  
  "connect-mongo": "^0.4.1",  
  "connect-multiparty": "^1.2.5",  
  "cookie-parser": "~1.0.1",  
  "errorhandler": "~1.0.0",  
  "express": "~4.0.0",  
  "express-jwt": "^0.1.3",  
  "express-session": "~1.0.2",  
  "flow": "^0.2.3",  
  "jade": "~1.2.0",  
  "jsonwebtoken": "^0.3.0",  
  "lodash": "~2.4.1",  
  "method-override": "~1.0.0",  
  "mongoose": "~3.8.8",  
  "mongoose-deep-populate": "^1.1.0",  
  "morgan": "~1.0.0",  
  "passport": "~0.2.0",  
  "passport-facebook": "latest",  
  "passport-google-oauth": "latest",
```

## **22. Instrucciones de instalación**

Previamente al inicio de la instalación de los servicios y aplicaciones necesarios, se debe de disponer del gestor de paquetes de OSX Homebrew<sup>31</sup> instalado, así como el sistema de control de versiones Git.

Las instrucciones de instalación son basadas en OSX 10.10.3 “Yosemite”:

1. Instalación de Node.js (incluye la instalación del gestor de paquetes npm).

```
$> brew update && brew doctor && brew node
```

2. Clonar el repositorio en el que se encuentra la aplicación

```
$> git clone giturl al repositorio
```

3. Instalación de MongoDB.

```
$> brew install mongodb
```

4. Instalación de los paquetes necesarios

```
$> npm install -g grunt-cli
```

5. Instalación de los paquetes dependientes de la aplicación

```
$> npm install
```

Una vez se hayan completados estos pasos será recomendable proceder a cargar una copia de seguridad de la base de datos o simplemente ejecutar la carga de semillas mediante grunt.

Para poder ejecutar la aplicación se ha de disponer de una instancia de MongoDB

---

<sup>31</sup> [HTTP://BREW.SH/](http://brew.sh/)



funcionando y ejecutar el siguiente comando:

```
$> npm install
```

Con ese comando se entra en modo de desarrollo en el cual el sistema cargará los archivos dinámicamente compilando las hojas de estilo de SaSS a CSS así como otro tipo de tareas relativas a la carga en tiempo real de modificación de archivos.

Para entrar en modo depuración es necesario disponer instalado el modulo de npm node-inspector.

```
$> npm install -g node-inspector
```

Una vez se haya instalado se ejecuta

```
$> node-inspector
```

Para hacer *debugging* en tiempo real hay que acceder a la siguiente Url con el servidor de desarrollo encendido.

```
$> open http://127.0.0.1:8080/debug?ws=127.0.0.1:8080&port=5858
```

Para ejecutar las pruebas tanto de integración como de funcionalidad simplemente se ejecuta el siguiente comando. Cabe la pena remarcar que se dispone de carga automática de cambio para todos los sistema de archivo relacionados con las pruebas.

```
$> grunt test
```

## **23. Instrucciones de uso**

Basado en el concepto de este proyecto, la idea principal es publicar o recabar información sobre la mejor manera o la más eficiente para poder disponer de un nuevo conocimiento, ya sea usando un framework o aprendiendo un nuevo lenguaje de programación.

Para poder hacer uso de la información que ofrece la aplicación no es necesario registrarse, ya que el concepto es que en una simple visita cualquier usuario pueda obtener información siendo ofrecida de manera desinteresada.

Sin embargo para poder añadir información, comentar o interactuar con la aplicación es necesario registrarse para disponer de un usuario.

### **23.1 Casos de uso**

Para poder disponer de mayor claridad sobre el funcionamiento de la aplicación vamos a mostrar ciertos casos de uso de la misma que puedan ser de ayuda.

#### **23.1 Creación de un usuario**

Como ya hemos comentado, para poder hacer uso de las funcionalidades que ofrece la plataforma, es necesario disponer de un usuario, este proceso de registro o *Sign up* puede ser directo o mediante cualquiera de los tres proveedores de OAuth que disponemos: Twitter, Facebook y Google plus.

Para ello necesitamos hacer accede a la pantalla de registro haciendo *clic* en el enlace de la parte superior derecha de la pantalla, *Sign up*.

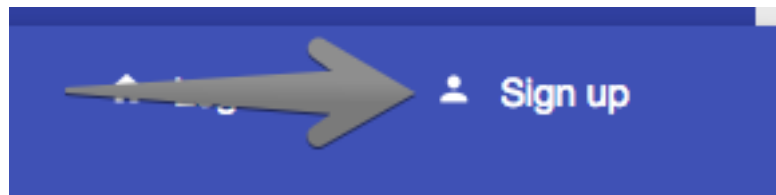


Figura 24. Captura del enlace a Sign up

Posteriormente nos valdría con rellenar el formulario con los datos necesarios, recordar que todos los campos son requeridos a excepción del *Twitter Account*.

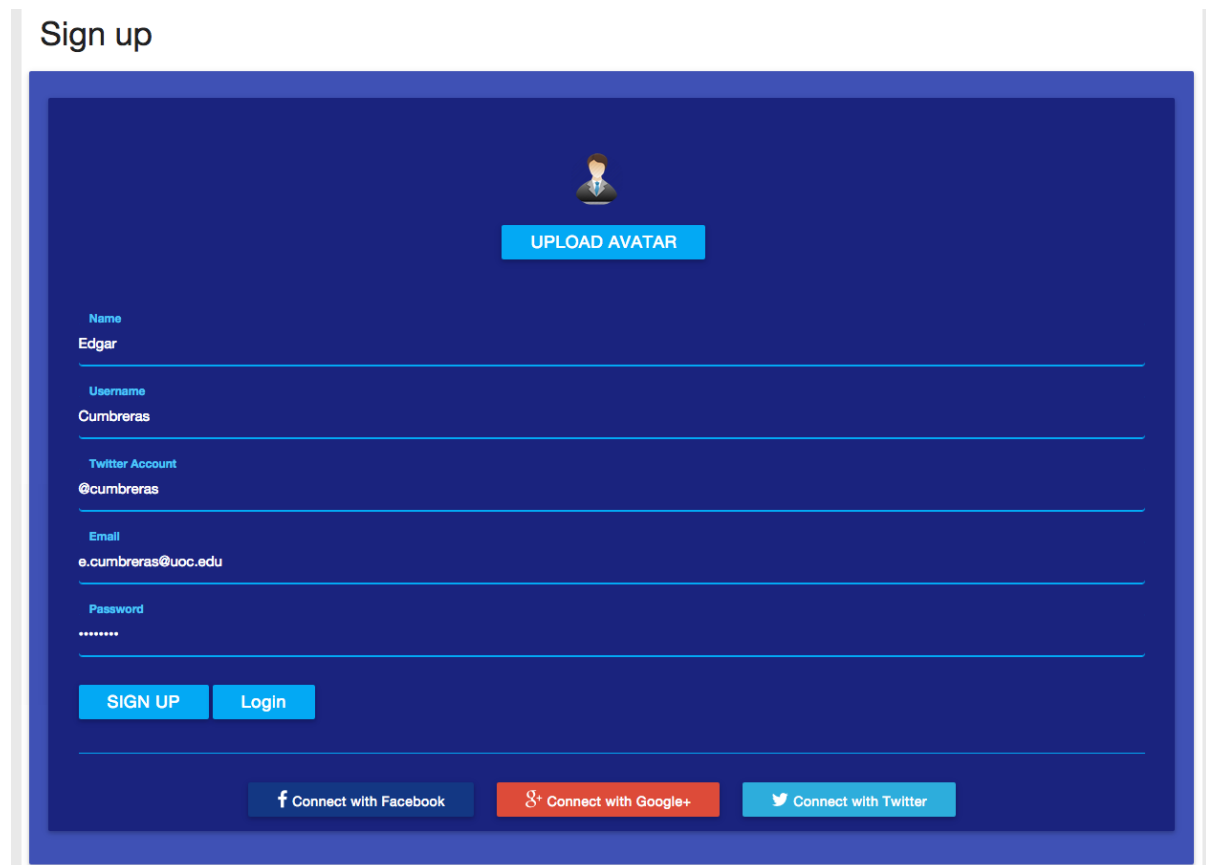


Figura 25. Captura de la pantalla de registro

Una vez se hayan añadido los datos, se puede subir un avatar en la parte central superior, pulsando en *Upload avatar* o subir avatar. Una vez se hayan completado estos pasos simplemente se procede a pulsar el botón *Sign Up* si estos son válidos la aplicación identificará al usuario directamente creando una sesión que durará un cierto periodo de tiempo.

## 23.2 Creación y edición de un path

Una vez disponemos nuestro usuario registrado e identificado podemos proceder a crear un path. Para poder crear un path es necesario que hagamos *click* en el icono de edición que ha aparecido en la parte superior izquierda de la pantalla.

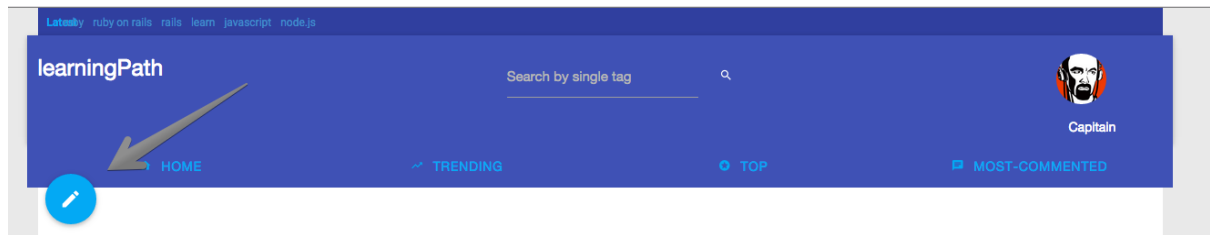


Figura 26. Captura del acceso rápido a la creación de paths

Una vez estemos en nuestro perfil, simplemente tenemos que hacer *click* en el icono de añadir path que está en la parte inferior derecha de la pantalla.



Al ejecutar dicha acción aparecerá un path con una información básica en nuestro perfil listo para ser editado. Para proceder a editar dicho path necesitamos hacer *click* en el botón de editar de color naranja que está posicionado en la parte inferior derecha del mismo.

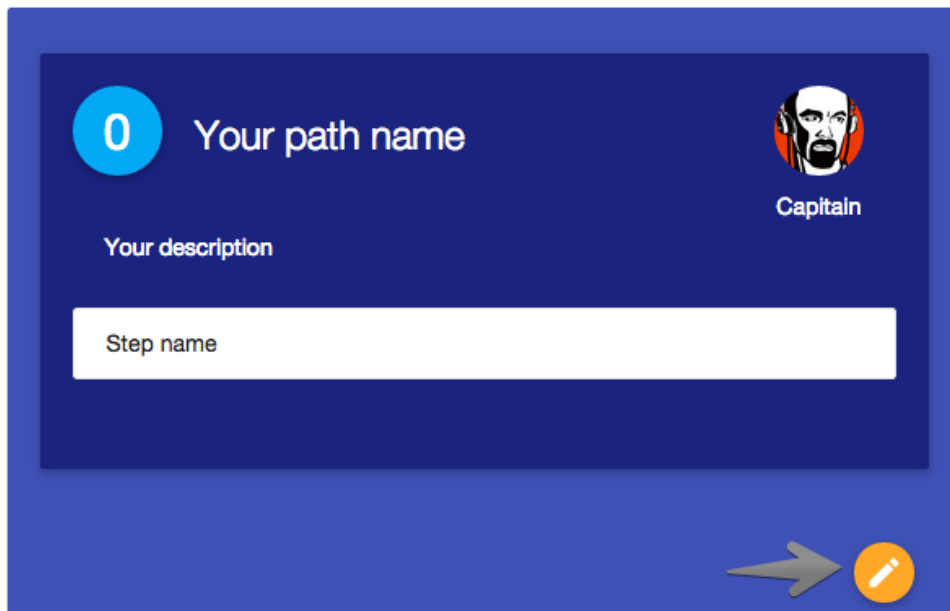


Figura 27. Captura de un path

Al hacer *clic* en ese botón veremos que el path cambia a modo edición y muestra todas las opciones posibles a editar.

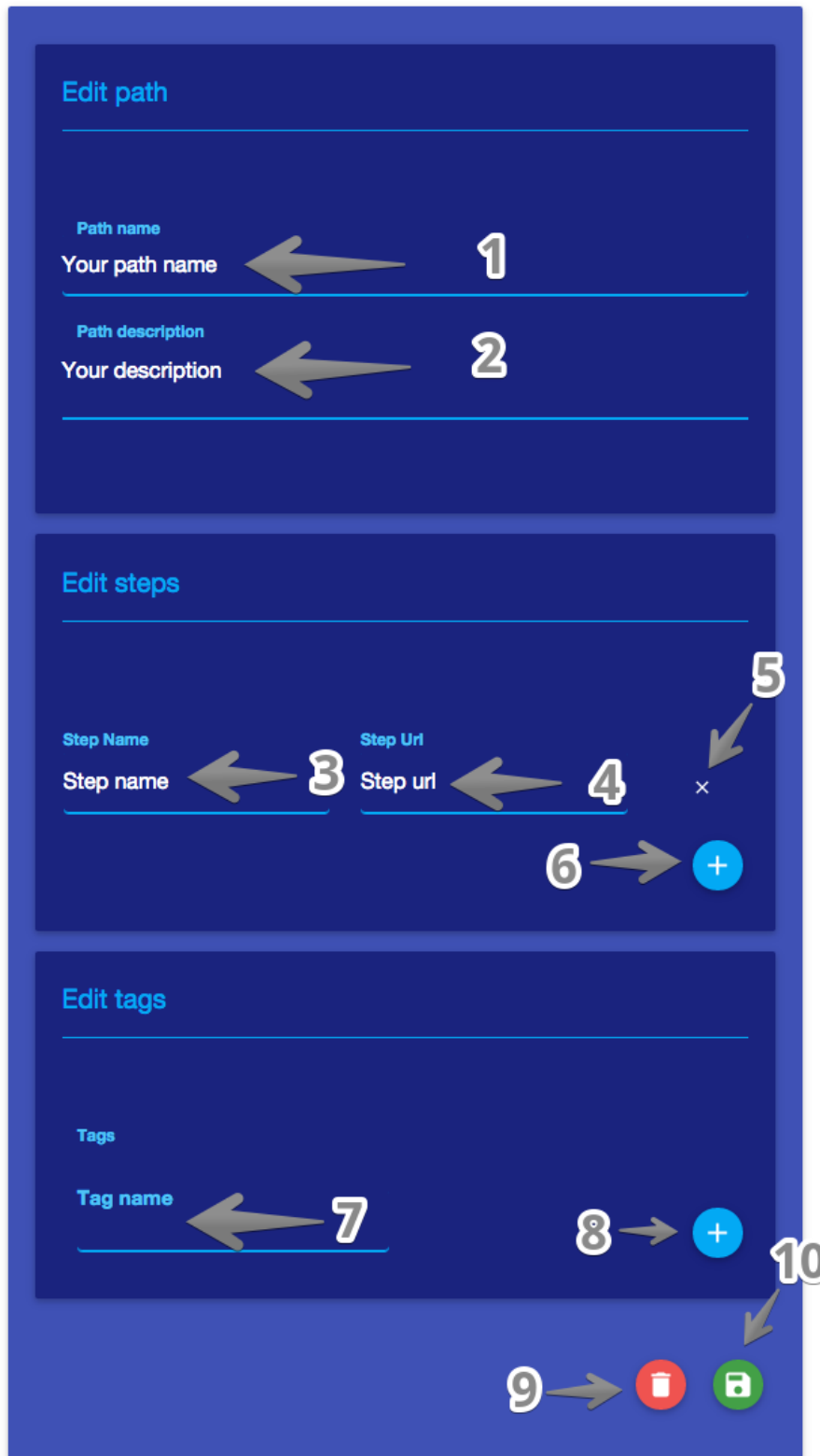


Figura 28. Captura del modo edición del path

En el modo de edición disponemos de múltiples opciones:

1. Nombre del path
2. Descripción del path
3. Nombre del step
4. Dirección web del step
5. Borrar el step
6. Añadir un nuevo step
7. Nombre del tag
8. Añadir el tag
9. Borrar el path
10. Guardar el path

Tras modificar las opciones como sea oportuno y guardar el path, este será público para todos los otros usuarios de la plataforma.

### **23.3 Publicar un nuevo comentario**

Para publicar un comentario en cualquiera de los paths de la plataforma, simplemente hay que estar identificado e ir a página que muestra el path. Allí se encuentra la sección de comentarios para dicho path.

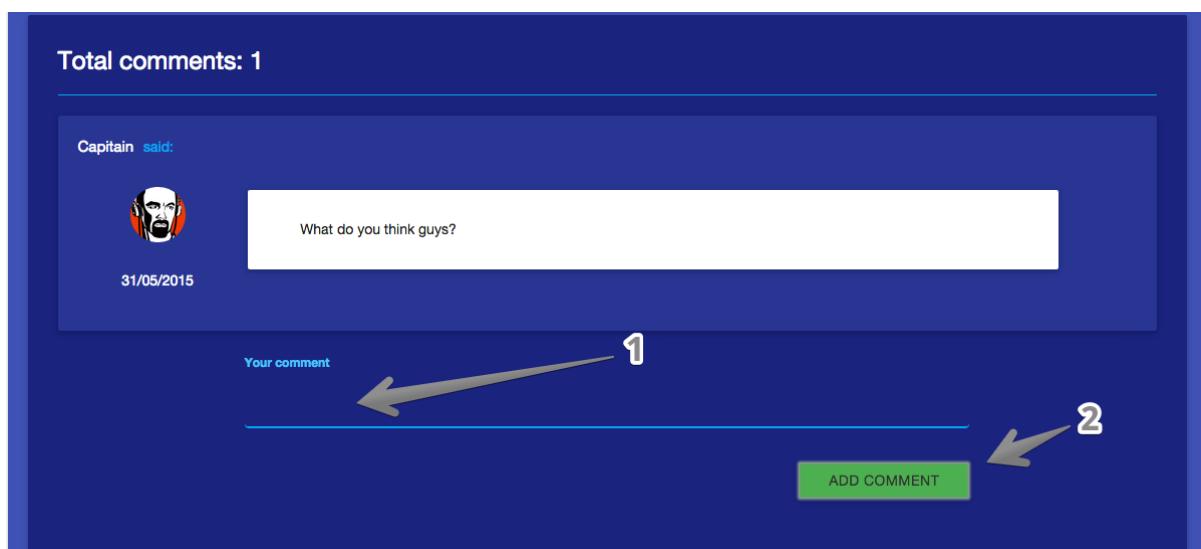


Figura 29. Captura de la sección de comentarios

Para poder añadir un nuevo path simplemente hay que rellenar el comentario (punto 1) y pulsar a publicar (punto 2).

### 23.4 Ver los comentarios previamente publicados

Cada usuario dispone de la posibilidad de ver los comentarios previamente publicados, para ello simplemente hay que acceder al menú principal y acceder mediante Tus comentarios, *Your comments*.

Una vez hayamos accedido al menú, dispondremos de una vista en la cual se muestran todos los comentarios que el usuario ha generado previamente, en esta vista también disponemos de un enlace al path sobre el que el comentario fue generado, así como la fecha en la que el comentario fue

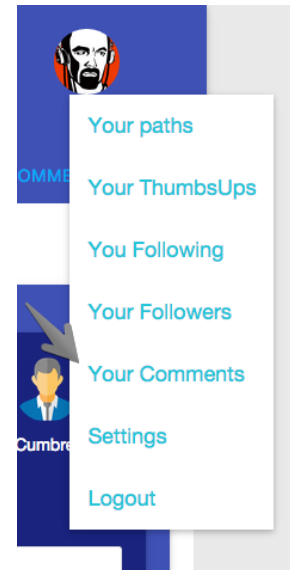


Figura 30. Captura del menú principal

generado. Este tipo de vistas son efectivas para el seguimiento de ayuda solicitada en los paths, ya que quizá alguna de las indicaciones que el path muestra es necesitada de aclaración o de mejora.

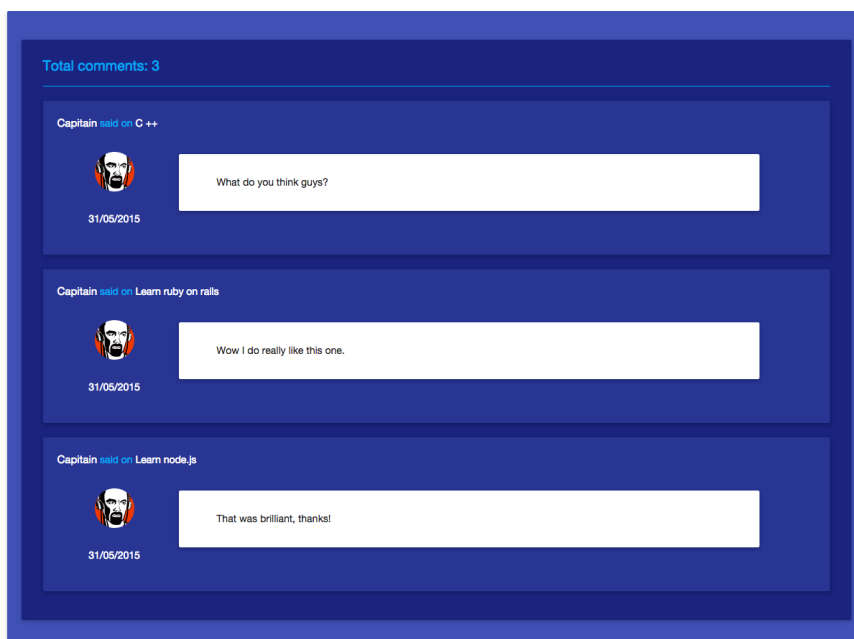


Figura 31. Captura de la sección individual de comentarios

### 23.5 Seguir a un usuario

Para seguir un usuario hay que entrar en el modo de vista unitaria del path, para esto es necesario pulsar en el título del path. Una vez estamos en esta vista, simplemente hay que pulsar el botón *Follow* para que se comience a seguir a este usuario.



Figura 32. Captura de la cabecera mostrando el boton de seguimiento

### 23.6 Votar un path

Para poder hacer un thumb-up o una votación positiva a un determinado path, es necesario hacer *clic* sobre el círculo con número en la parte superior derecho de cada path.



Figura 33. Captura de la versión reducida de un path mostrando la opción de thumb-ups

### 23.7 Hacer una búsqueda

El sistema ofrece un sistema de búsqueda basado en los tags que los usuarios añaden a los paths. Estos tags habitualmente son palabras clave que el usuario cree que determinan el uso, el contenido y el objetivo del path. Para poder ejercer una búsqueda es necesario introducir un termino en la sección de búsqueda que se encuentra en la parte superior de la página dentro de la cabecera, y pulsar el icono de búsqueda.



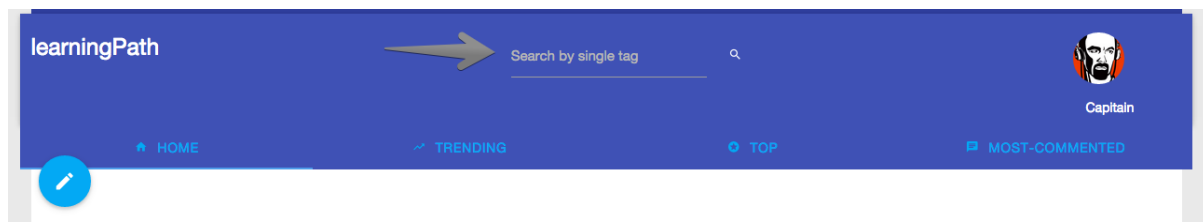


Figura 34. Captura de la cabecera mostrando la opción de búsqueda

### **23.7 Acceso al menú de opciones**

Para poder acceder a estas funcionalidades disponemos del menú principal, este menú principal es accesible al pasar el ratón sobre el nombre de usuario de la parte superior derecha de la pantalla.

En este menú disponemos de múltiples opciones, disponemos de acceso a:

1. Tus paths
2. Tus thumb-ups
3. Siguiendo
4. Seguidores
5. Comentarios
6. Ajustes
7. Desconectar

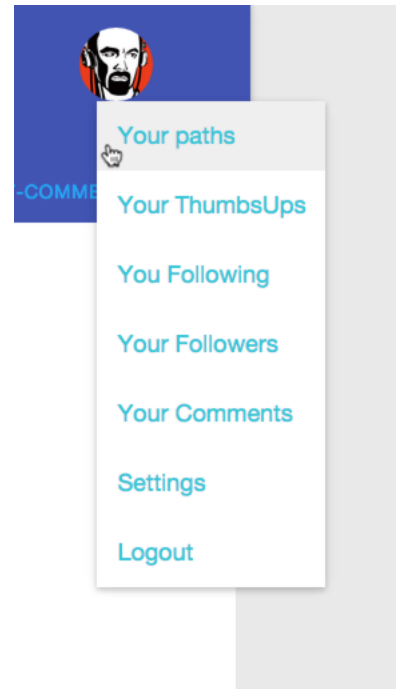


Figura 35. Captura del menú principal

## 24. Configuración y administración de servidores

La aplicación está actualmente alojada en los servidores de Heroku. Esta plataforma de servicio ofrece unos servidores de manera gratuita asociadas a ciertas condiciones. El servidor solamente dispone de una memoria de 512Mb y dispone de cierto tiempo de *timeout* en el que la aplicación se apagará si no hay ninguna petición en 30 minutos. También la aplicación ha de estar inactiva al menos 6 horas al día para entrar dentro de dicho plan, de lo contrario, habría que pasar al sistema de pago, induciendo un coste inicial para la aplicación.

### 24.1 Banderas de configuración

Para poder disponer del servidor en modo producción funcionando correctamente y hacer el *deployment* que disponga las configuraciones base adecuadas es necesario añadir los valores necesarios en la configuración que nos ofrece Heroku en su panel de control.

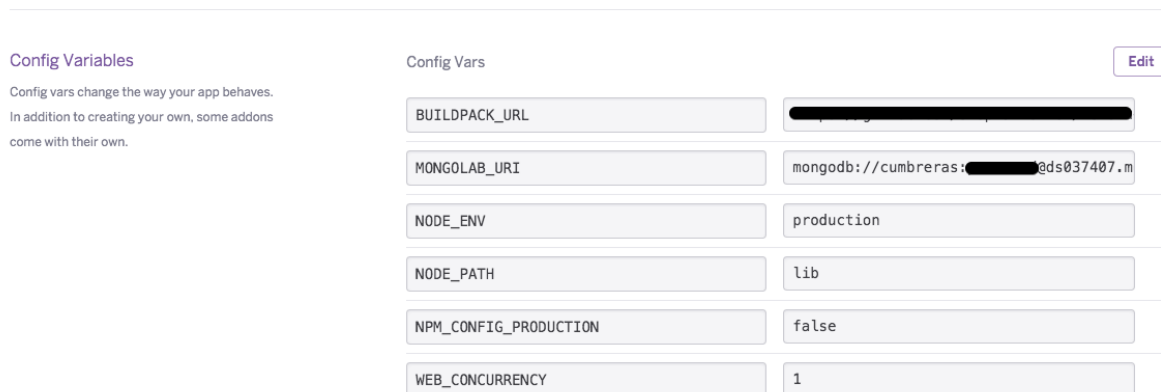


Figura 36. Captura de la configuración de deployment del servidor en Heroku

## **25. Plan de negocio**

### **25.1 Elevator Pitch**

LearningPath es una plataforma gratuita donde usuarios pueden obtener consejos de la mejor manera de aprender nuevas tecnologías, la plataforma les ofrece a los usuarios una manera de ayudar y distribuir su conocimiento así como participar en la comunidad del *open-source*.

### **25.2 Objetivos y visión**

Los objetivos que dispone esta plataforma son principalmente focalizados en incentivar la educación y el proceso de aprendizaje. La creación de guías materiales en las cuales los usuarios pueden obtener un punto de apoyo y referencia para empezar o continuar el proceso educativo.

La visión de la plataforma es ser pieza de referencia en la comunidad del código abierto, en la que los usuarios se apoyen mutuamente y sirva de punto de referencia como base y centro de ideas así como enfoques para la solución de problemas.

### **25.3 Obstáculos**

Existen ciertos obstáculos que pueden ser directamente relacionados con la falta de visitantes al sitio web, ya que la plataforma es principalmente dependiente de dicho contenido creador por los usuarios para poder ofrecer soluciones a los problemas que los visitantes vengan a solucionar.

Otro problema que puede encontrar la plataforma es aquel problema relacionado con la falta de personas relacionadas al desarrollo y mantenimiento de la aplicación, ya que como la plataforma será desarrollada en código abierto, se espera que el desarrollo de nuevas funcionalidades, solución de bugs y actualizaciones sean tareas realizadas por usuarios desinteresados que quieran apoyar la plataforma con su trabajo directo.

La financiación es otro de los problemas asociados a la plataforma, ya que el gol de dicha plataforma es mantenerse como una plataforma abierta sin coste para los usuarios, sin embargo, existen costes operacionales mensuales derivados de los servidores en los que la aplicación se encuentra.

## **25.4 Estrategia de marketing**

La promoción de la aplicación dispone de limitaciones directamente relacionadas con la viabilidad financiera del proceso promocional, por lo tanto todas las campañas promocionales deberán de ser basadas en la teórica de coste cero, en el cual se usarán todas las herramientas disponibles que no tengan un coste asociado para la plataforma. Al no disponer de coste y ofrecer un servicio a la comunidad podemos esperar que la gran parte de los usuarios que reciba la plataforma sea mediante el consumo del contenido que se ofrezca en la plataforma.

## **25.5 Análisis de mercado**

El concepto de una red social de aprendizaje y plataforma para compartir el enfoque en el momento de aprender un nuevo concepto no ha sido desarrollado en la actualidad, sin embargo existen ciertas aplicaciones y plataformas que han sido la base para crear el concepto en que nuestra red social está basada.

### **25.5.1 Tamaño del mercado**

En 2012 existían más de 2.65 millones de estudiantes estudiando a distancia y se conoce que más de 5 millones de estudiantes han estudiado a distancia de alguna manera a lo largo de su carrera educativa.

### **25.5.2 Crecimiento del mercado**

Aunque en los últimos tres años no se ha visto crecimiento alguno en el mercado a nivel mundial si que se puede ver una clara estabilidad en el medio disponiendo de una seguridad de que dicha estabilidad es un debido a un punto de madurez en el medio.

### **25.5.3 Rentabilidad del mercado**

El mercado de la educación a distancia y aprendizaje online tiene una expectativa de alcanzar los 107 billones de dólares para este mismo año.

### **25.5.4 Tendencias del mercado**

En el Mercado actual se está viendo incursión de ciertos sectores empresariales como usuarios base para este tipo de educación online, disponiendo de planes de re-educación de los empleados así como mejora educativa y formativa de las plantillas actuales.

Existen otra tendencia relativa a la reestructuración del mercado educativo, disponiendo de muchos sujetos requiriendo una educación a distancia debido a la necesidad de continuar o comenzar su vida laboral por motivos económicos.

### **25.5.5 Competidores**

Múltiples son las organizaciones que focalizan sus esfuerzos en la educación online. Pero en la actualidad no existe una plataforma que proponga o de una solución al problema que LearningPath soluciona. Por consiguiente es necesario añadir que el ámbito de la aplicación no tiene una dirección y foco comercial, lo que hace que en este apartado no contemos con competidores.

### **25.6.1 Presupuesto**

El apartado presupuestal del proyecto está enfocado a la perspectiva y enfoque de los modelos financieros basados en el software de código abierto.

Existen principalmente dos vertientes principales de canalización de fondos en orden de disponer el servicio sin costes directos ya que la focalización directa es no monetizar la aplicación evitando que los usuarios dispongan de un pago asociado al uso de esta.

Los principales modelos que podemos usar son los siguientes:

- **Concesiones:** Este modelo sostenible para el software libre es basado en la percepción de un fondo financiero por parte de una compañía o una sociedad
- **Recaudación de fondos:** Organizaciones como Wikipedia se basan en este modelo, en el cual disponen de una vía de financiación asociada a las aportaciones individuales o colectivas. Este tipo de aportaciones monetarias son las bases para el sustento organizativo.
- **Patrocinio:** Una empresa apoya el proyecto recibiendo publicidad a cambio del sostenimiento parcial o global de los costes asociados al proyecto.

Los costes iniciales del proyecto serán solicitados con carácter de devolución retroactiva a una empresa o individuo.

El presupuesto inicial se centra en los gastos del servidor, ya que el desarrollo no dispone de gasto asociado al desarrollo.

### 25.6.2 Previsión financiera

<i>Concepto</i>	<i>Valor bruto anual</i>	<i>Comentario</i>
<b>Dominio</b>	16.00€	Renovable bi-anual
<b>Servidor</b>	340.50€	Heroku, 2 dynos, valido para los primeros 12 meses
<b>CircleCI</b>	60€	Sistema de continua integración de test, valido para los dos primeros años.
<b>Github</b>	60€	Control de versiones.

Tabla 3. Previsión financiera

## **25.9 Proyección de futuro**

El concepto central del proyecto es poder crear una red social que llegue a ser referencia para poder adaptar nuevas tecnologías de una manera sencilla y guiada. Las características y funcionalidades que actualmente dispone la aplicación son aquellas que se han conceptualizado para generar un proyecto que dispone de una lógica funcional básica, ofreciendo una solución para un problema conocido y reconocido por la sociedad.

La futura planificación del proyecto es directamente relacionado con la aceptación y revisión que los usuarios ofrezcan. Existen ciertas directrices que el proyecto deberá de mantener en orden de disponer de su identidad intacta, sin embargo, la base de este proyecto es el usuario al disponer de posibilidad de aceptar cambios, mejoras en las implementaciones actuales y nuevas funcionalidades por parte de los mismos usuarios.

La red social a cierto punto dispondrá de una vía de acceso a generar funcionalidades por parte de cualquiera que lo desea y que disponga del conocimiento necesario para dicha tarea. Disponiendo de una plataforma cien por cien mantenida por usuarios de una manera desinteresada, de esta manera serán los propios usuarios los que dispongan de un gran control sobre que es lo que la aplicación les ofrece.

## **28. Conclusión**

La elaboración de esta proyecto ha sido uno de los mayores retos que he tenido de manera personal y estudiantil, ya que he intentando salir del ámbito en el que habitualmente me muevo e intentar dar un enfoque diferente basado en unos valores que preceden al nombre de la UOC y que desde ahora estarán conmigo para siempre.

Y es que la idea de crear una plataforma desde cero, de principio a fin con una tecnología puntera que he podido usar gracias a lo aprendido durante estos años, que se focaliza en ayudar semanera desinteresada, con un punto de vista totalmente diferente al que uno se habitúa a ver, es un gran logro que solo me hace mejorar las perspectivas que tengo sobre mi mismo y mi futuro.

Generar una herramienta que da valor desde el primer momento de uso, eliminando todas las barreras posibles para su uso e incentivando uno de los mayores bienes que tenemos como seres humanos como es la capacidad de aprender, es realmente un momento de realización personal.

La UOC ha ofrecido ayuda desde el primer momento en el que decidí inscribirme como alumno, me ha hecho aprender y ver múltiples disciplinas de la tecnología algo que ha sido y está siendo base para la nueva carrera laboral que he emprendido desde que tome mi primera lección como alumno.

Para mi esto solo es el comienzo aunque ya hayan pasado muchos años en esta organización, y es que el incentivo de ver resultados tras el trabajo duro es algo que enorgullece en gran manera, y es difícil de aplacar.

El objetivo con el que acabo este proyecto es que solo sea el comienzo de la multitud de oportunidades que tenga para devolver a la comunidad todo lo que he recibido de ella.

Gracias.



## **Anexo 1. Entregables del proyecto**

1. Repositorio conteniendo el código completo del sistema
2. Presentación para el público
3. Presentación (video) para el tribunal
4. Autoevaluación del proceso

## Anexo 2. Código fuente

### Anexo 2.1 Controlador de rutas

Sección del controlador de rutas del lado del servidor, ahí podemos ver principalmente cuales son las rutas añadidas al servicio.

También dispone del *middleware* específico para tratar con los errores HTTP 404.

```
module.exports = function(app) {
  app.use('/api/paths', require('./api/path'));
  app.use('/api/users', require('./api/user'));
  app.use('/api/steps', require('./api/step'));
  app.use('/api/tags', require('./api/tag'));
  app.use('/api/comments', require('./api/comment'));
  app.use('/api/thumbsups', require('./api/thumbsup'));
  app.use('/api/upload', require('./api/upload'));
  app.use('/api/search', require('./api/search'));

  app.use('/auth', require('./auth'));

  app.route('/:url(api|auth|components|app|bower_components|assets)/*')
  )
  .get(errors[404]);

  app.route('/')
  .get(function(req, res) {
    res.sendFile(app.get('appPath') + '/index.html');
  });
};
```

## **Anexo 2.2 Configuración del middleware sobre Express.js**

Sección del middleware de la aplicación donde todos los elementos del mismo quedan gestionados. También vemos una sección de la configuración de conexión con MongoDB así como la gestión de ambientes de producción.

```
module.exports = function(app) {
  var env = app.get('env');

  app.set('views', config.root + '/server/views');
  app.set('view engine', 'jade');
  app.use(compression());
  app.use(bodyParser.urlencoded({ extended: false }));
  app.use(bodyParser.json());
  app.use(methodOverride());
  app.use(cookieParser());
  app.use(passport.initialize());

  app.use(session({
    secret: config.secrets.session,
    resave: true,
    saveUninitialized: true,
    store: new mongoose.mongoStore({ mongoose_connection:
mongoose.connection })
  }));

  if ('production' === env) {
    app.use(favicon(path.join(config.root, 'public',
'favicon.ico')));
    app.use(express.static(path.join(config.root, 'public')));
  }
}
```

## **Anexo 2.3 Configuración de Passport para autenticación Facebook**

Sección de la autenticación y configuración del *passport* para el lado del servidor.

```
var passport = require('passport');
var FacebookStrategy = require('passport-facebook').Strategy;

exports.setup = function (User, config) {
  passport.use(new FacebookStrategy({
    clientID: config.facebook.clientID,
    clientSecret: config.facebook.clientSecret,
    callbackURL: config.facebook.callbackURL
  }),
  function(accessToken, refreshToken, profile, done) {
    User.findOne({
      'facebook.id': profile.id
    },
    function(err, user) {
      if (err) {
        return done(err);
      }
      if (!user) {
        user = new User({
          name: profile.displayName,
          email: profile.emails[0].value,
          role: 'user',
          username: profile.username,
          provider: 'facebook',
          facebook: profile._json
        });
        user.save(function(err) {
          if (err) {
            return done(err);
          }
          return done(null, user);
        });
      } else {
        return done(null, user);
      }
    });
  });
}
```

## **Anexo 2.4 Rutas para el endpoint del modelo User**

Estas son las rutas definidas para el *endpoint* del modelo User, como podemos ver hemos seguido una arquitectura REST.

```
'use strict';

var express = require('express');
var controller = require('./user.controller');
var config = require('../../config/environment');
var auth = require('../../auth/auth.service');

var router = express.Router();

router.get('/', auth.hasRole('admin'), controller.index);
router.delete('/:id', auth.hasRole('admin'), controller.destroy);
router.get('/me', auth.isAuthenticated(), controller.me);
router.put('/:id/password', auth.isAuthenticated(),
controller.changePassword);
router.get('/:id', auth.isAuthenticated(), controller.show);
router.get('/:id/thumbsups', auth.isAuthenticated(),
controller.thumbsUps);
router.post('/', controller.create);
router.post('/:id/addPath', auth.isAuthenticated(),
controller.addPath);
router.post('/:id/addThumbsUp', auth.isAuthenticated(),
controller.addThumbsUp);
router.post('/:id/addComment', auth.isAuthenticated(),
controller.addComment);
router.post('/:id/addImageUrl', auth.isAuthenticated(),
controller.addImageUrl);
router.post('/:id/addFollower', auth.isAuthenticated(),
```

## **Anexo 2.5 Esquema (mongoose) para el modelo User**

Aquí vemos el esquema mongoose definido para el la clase User, como podemos ver disponemos de asociaciones variadas con path, comments, followers, thumb-up.

```
var UserSchema = new Schema({
  name: String,
  handle: String,
  email: { type: String, lowercase: true },
  twitterAccount: { type: String, lowercase: true },
  imageUrl: { type: String, lowercase: true, default:
'https://cdn4.iconfinder.com/data/icons/pretty-office-part-4-shad
ow-style/256/Man1-.png' },
  role: {
    type: String,
    default: 'user'
  },
  hashedPassword: String,
  provider: String,
  salt: String,
  facebook: {},
  twitter: {},
  google: {},
  github: {},
  thumbsUps: [thumbsUpSchema],
  followers: [],
  followings: [],
  paths: [pathSchema],
  comments: [commentSchema]
});
```

## **Anexo 2.6 Atributos virtuales para el modelo User**

Mongoose nos ofrece la posibilidad de crear atributos virtuales, estos atributos son valores calculados en tiempo real que son asociados a un falso campo en el modelo.

```
UserSchema
  .path('email')
  .validate(function(email) {
    if (authTypes.indexOf(this.provider) !== -1) return true;
    return email.length;
  }, 'Email cannot be blank');

// Validate empty password
UserSchema
  .path('hashedPassword')
  .validate(function(hashedPassword) {
    if (authTypes.indexOf(this.provider) !== -1) return true;
    return hashedPassword.length;
  }, 'Password cannot be blank');

// Validate email is not taken
UserSchema
  .path('email')
  .validate(function(value, respond) {
    var self = this;
    this.constructor.findOne({email: value}, function(err, user) {
      if(err) throw err;
      if(user) {
```

## **Anexo 2.7 Endpoint para la subida de imágenes a Amazon AWS S3**

Este es el lado del servidor, en el cual disponemos de la conexión a Amazon para subir las imágenes, toda la configuración es basada en el SDK que Node.js nos ofrece.

```
var _ = require('lodash');
var AWS = require('aws-sdk');
var fs = require('fs');
var mime = require('mime');
var knox = require('knox');
var HTTPStatus = require('http-status');

var knoxClient = knox.createClient({
  key: '*****',
  , secret: '*****'
  , bucket: 'learningpath'
});

exports.create = function(req, res) {

  var file = req.files.file;
  var stream = fs.createReadStream(file.path);
  var mimetype = mime.lookup(file.path);
  var req;
  var imageUrl;

  req = knoxClient.putStream(stream, file.name, {
    'Content-Type': mimetype,
    'Cache-Control': 'max-age=604800',
    'x-amz-acl': 'public-read',
    'Content-Length': file.size
```



## **Anexo 2.8 Configuración del modulo imager**

Imager es el modulo usado para la configuración de los avatares usados por los usuarios, esta configuración es la que se usa para declarar los tamaños de imagen subidos a Amazon.

```
variants: {
  items: {
    // keepNames: true,
    resize: {
      mini : "300x200",
      preview: "800x600"
    },
    crop: {
      thumb: "200x200",
      // Sets the crop position, or "gravity". Default is NorthWest.
      // See
      http://www.graphicsmagick.org/GraphicsMagick.html#details-gravity
      for details
      thumb_center: "200x200 Center"
    },
    resizeAndCrop: {
      large: {resize: "1000x1000", crop: "900x900"}
    },
    thumbnail: {
      // "Cuts the thumbnail to fit"
      // See
      http://superuser.com/questions/275476/square-thumbnails-with-image
      magick-convert
      better_thumb: "100x100 NorthWest",
      better_thumb_center: "100x100 Center"
```

## **Anexo 2.9 Unit test para el modelo Thumb**

Una sección de un test unitario para el modelo Thumb.

```
'use strict';

var should = require('should');
var app = require('../..../app');
var request = require('supertest');

describe('GET /api/thumbsups', function() {

  it('should respond with JSON array', function(done) {
    request(app)
      .get('/api/thumbsups')
      .expect(200)
      .expect('Content-Type', /json/)
      .end(function(err, res) {
        if (err) return done(err);
        res.body.should.be.instanceof(Array);
        done();
      });
  });
});
```

## Anexo 2.10 Endpoint – Controlador para el modelo Step

Este código es el referente al *endpoint* del modelo step, el cual se usa para poder leer, borrar, actualizar y crear los steps que van asociados al modelo Path.

```
var _ = require('lodash');
var Step = require('./step.model');

// Get list of steps
exports.index = function(req, res) {
  Step.find(function (err, steps) {
    if(err) { return handleError(res, err); }
    return res.json(200, steps);
  });
};

// Get a single step
exports.show = function(req, res) {
  Step.findById(req.params.id, function (err, step) {
    if(err) { return handleError(res, err); }
    if(!step) { return res.send(404); }
    return res.json(step);
  });
};

// Creates a new step in the DB.
exports.create = function(req, res) {
  Step.create(req.body, function(err, step) {
    if(err) { return handleError(res, err); }
    return res.json(201, step);
  });
};
```

## **Anexo 2.11 Endpoint – Controlador para la búsqueda**

Para efectuar una búsqueda usamos una cadena de texto para encontrar aquellos Path que dispongan un tag que sea igual a dicha cadena.

```
'use strict';

var _ = require('lodash');
var Search = require('./search.model');
var Path = require('../path/path.model');

exports.index = function(req, res) {
  var searchTerm = req.query.searchTerm;
  Path
    .find()
    .where('tags.name').in([searchTerm])
    .populate('_user')
    .exec(function (err, paths) {
      if(err) { return handleError(res, err); }
      return res.json(200, paths);
    });
};
```

## **Anexo 2.12 Servicio Angular para identificación y registro del modelo User**

Para identificar a un usuario hacemos uso de galletas, por lo tanto una vez el usuario ha introducido la contraseña comprobamos que sea correcto y si lo es, añadimos una galleta así como le re direccionamos al lugar donde inició la petición.

```
login: function(user, callback) {
  var cb = callback || angular.noop;
  var deferred = $q.defer();

  $http.post('/auth/local', {
    email: user.email,
    password: user.password
  }).
  success(function(data) {
    $cookieStore.put('token', data.token);
    currentUser = User.get();
    deferred.resolve(data);
    return cb();
  }).
  error(function(err) {
    this.logout();
    deferred.reject(err);
    return cb(err);
  }).bind(this));

  return deferred.promise;
},
```

## **Anexo 2.13 Vista de los comentarios**

Este código es relativo a una de las vistas, en particular a la sección unitaria de comentarios. Como podemos ver todo el HTML creado en la plataforma es renderizado mediante el uso de Jade.

```
h4 {{pageTitle}}
.row
  .col.l12.s12.m12
    .card.indigo
      .card-content
        .card.indigo.darken-4.mvm
          .card-content
            .comments-container.phs
              .container--title
                h6 Total comments: {{userComments.comments.length}}
              hr
              .comments-panel(data-ng-repeat='comment in
userComments.comments')
                .card.indigo.darken-3.mvm
                  .card-content
                    .row
                      .col.l12.m12.s12
                        span.comments-title {{comment._user.handle}}
                        span.comments-subtitle=' said on '
                        a(href='/paths/show/{{comment._path._id}}')
                          {{comment._path.name}}
                    .row.valign-wrapper
                      .col.l2.center.mts.valign.hide-on-med-and-down
img.circle.avatar-image.tooltipped(data-ng-src='{{comment._user.
```

## **Anexo 2.14 Rutas de router de Angular para los followers (gestión de una SAP)**

Como hemos referido con anterioridad esta plataforma se basa en el concepto de *single page application* (SAP), la cual se centra en el desarrollo se ha centrado en que la aplicación no necesita refrescar la página para mostrar el nuevo contenido. Haciendo uso de peticiones para recibir la información. Este código muestra como está estructurada la plataforma en la parte del cliente para conseguir este gol.

```
angular.module('learningpathApp')
  .config(function ($stateProvider) {
    $stateProvider
      .state('followers', {
        url: '/followers/:userId',
        views: {
          'header': {
            templateUrl: 'components/header/header.html',
            controller: 'HeaderCtrl'
          },
          'navbar': {
            templateUrl: 'components/navbar/navbar.html',
            controller: 'NavbarCtrl'
          },
          'main': {
            templateUrl: 'app/followers/followers.html',
            controller: 'FollowersCtrl'
          },
          'footer': {
            templateUrl: 'components/footer/footer.html',
            controller: 'FooterCtrl'
          }
        }
      })
  })
```





## **Anexo 2.16 Controlador de Angular para la sección home**

Controlador (la C de MVC) del cliente que gestiona la sección de la página principal o HOME.

```
'use strict';

angular.module('learningpathApp')
  .controller('HomeCtrl', function ($scope, $http, Auth, $location,
  ThumbsUpService) {

    $scope.pageTitle = 'Home';

    $scope.currentUser = Auth.getCurrentUser();

    $scope.getPaths = function getPaths() {
      $http.get('/api/paths/home')
        .success(function(paths) {
          $scope.paths = paths;
        });
    }

    $scope.getPaths();

    $scope.$on('searchApplied', function(event, paths) {
      $scope.paths = paths;
      $scope.pageTitle = 'Search';
    });

    $scope.$on('searchRemoved', function(event, paths) {
      $scope.getPaths();
    });
  });
```

## **Anexo 3. Librerías externas**

Las librerías externas se han controlado desde dos puntos diferentes. Para el lado del servidor disponemos de npm<sup>32</sup> como gestión de paquetes y librerías, así como para el lado del cliente disponemos de bower<sup>33</sup> como gestor principal.

Algunas de las librerías usadas en el desarrollo de esta aplicación son:

- Cliente
  - AngularJS
  - Lodash.js
  - Json3
  - jQuery
  - bourbon
  - Materialize
  - Karma
  - Protractor
  - SASS
- Servidor
  - Express.js
  - Require.js
  - Mongoose
  - Passport
- Tooling
  - Grunt

Todas las librerías usadas se pueden encontrar en la aplicación en el archivo package.json y bowerfile.json.

---

<sup>32</sup> <https://www.npmjs.com/>

<sup>33</sup> <http://bower.io/>

## Anexo 4. Capturas de pantalla

### Anexo 4.1 Home

Vista de la página inicial de la plataforma.

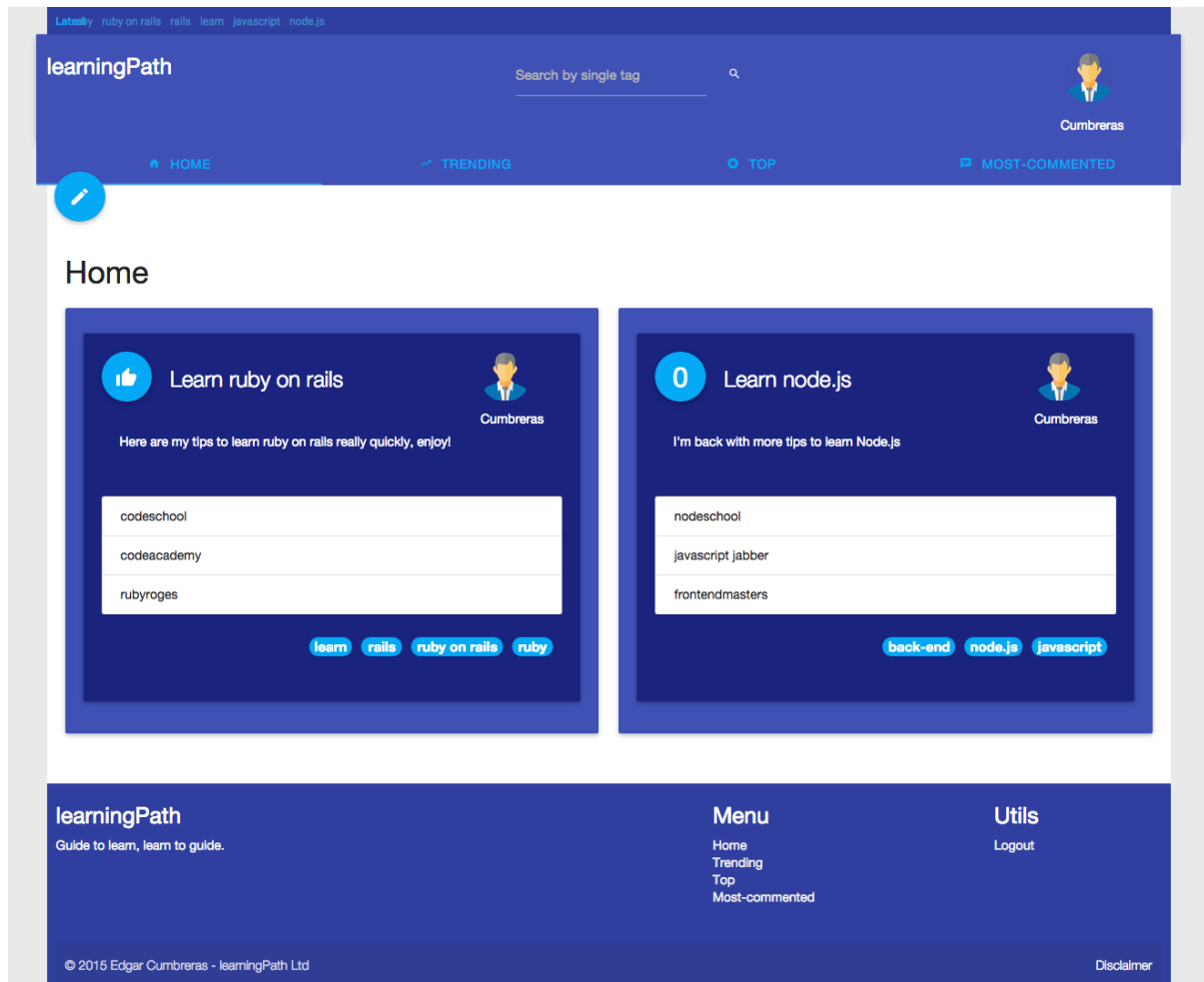
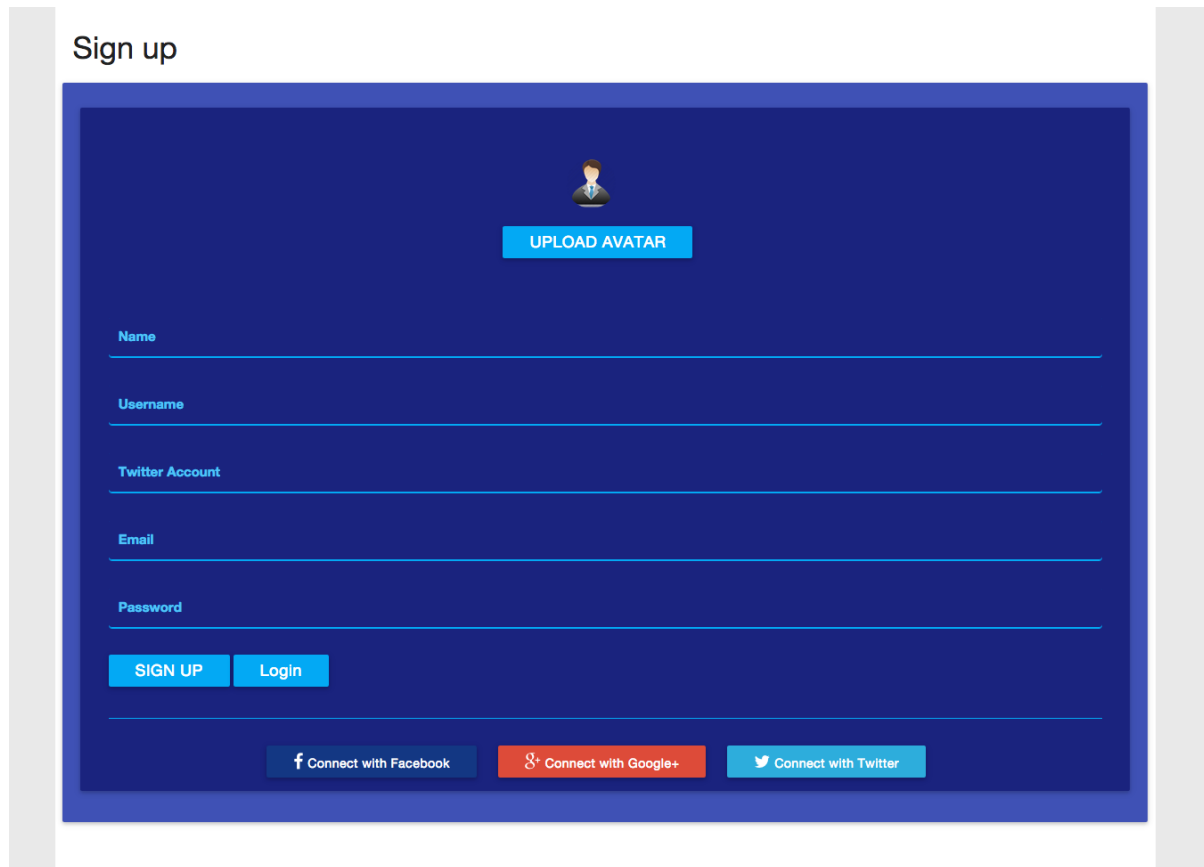


Figura 37. Captura de la sección Home

## 4.2 Sign Up (Registro)

Sección de registro y subida del avatar. También disponible el registro con Facebook, Twitter o Google Plus.



The image shows a 'Sign up' form with a dark blue background. At the top, there is a 'Sign up' title. Below it, a small icon of a person is followed by an 'UPLOAD AVATAR' button. The form contains several input fields: 'Name', 'Username', 'Twitter Account', 'Email', and 'Password'. Below these fields are two buttons: 'SIGN UP' and 'Login'. At the bottom, there are three social media connection buttons: 'Connect with Facebook', 'Connect with Google+', and 'Connect with Twitter'.

Figura 38. Captura de la sección de registro

## 4.3 Login (Identificación)

Sección para la identificación de los usuarios. Aquí el usuario se puede identificar, solicitar la contraseña si la hubiese olvidado o identificarse con cualquiera de las plataformas que disponemos (Facebook, Twitter y Google Plus).

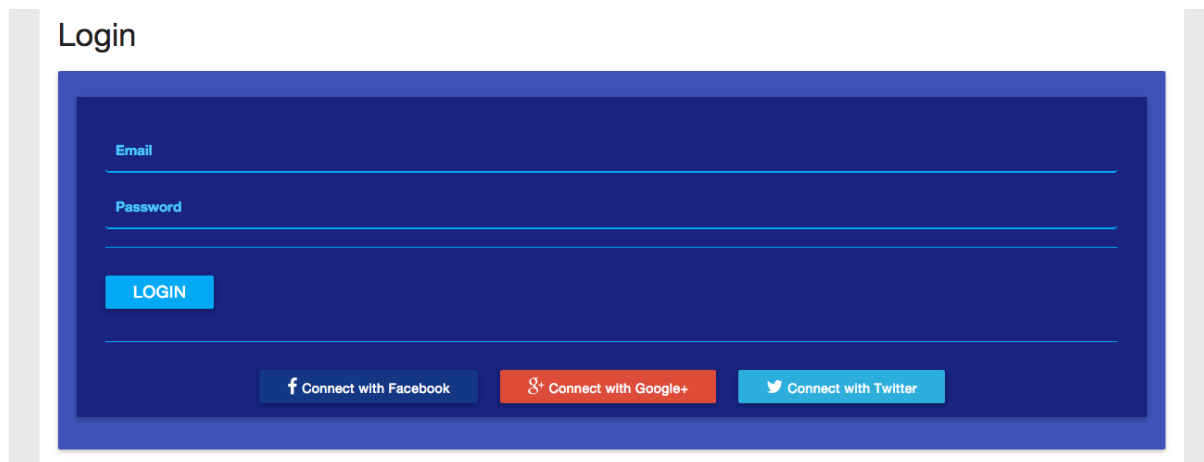


Figura 39. Captura de la sección de identificación

## 4.5 Path Edit (editar path)

Sección específica que aparece al momento de pulsar el modo edición en un path, el dueño del path podrá editar todas las opciones relativas con el mismo, así como añadir nuevos steps, tags o eliminar dicho path.

The screenshot displays a mobile application interface for editing a path. It is divided into three main sections: 'Edit path', 'Edit steps', and 'Edit tags'. At the bottom, there are two circular icons: a red one with a trash can symbol and a green one with a save symbol.

**Edit path**

Path name  
Learn ruby on rails

Path description  
Here are my tips to learn ruby on rails really quickly, enjoy!

**Edit steps**

Step Name	Step Url	
codeschool	http://www.codeschool.com	×

+

**Edit tags**

Tags  
learn rails ruby on rails ruby

Tag name  
+

Figura 40. Captura de la sección de edición de path

## 4.5 Settings (ajustes)

Esta es la página de ajustes, en la cual el usuario puede re-subir el avatar identificativo, así cambiar las opciones relativas con su perfil y contraseña.

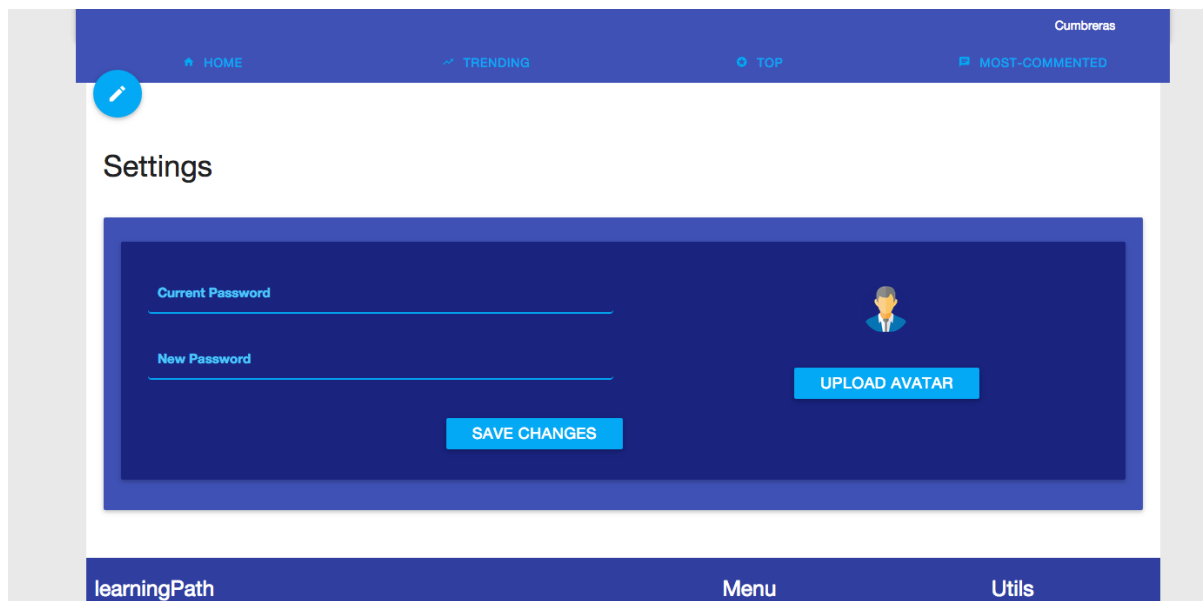


Figura 41. Captura de la sección de los ajustes

## 4.6 Path

En esta imagen podemos ver un path, mostrando quien lo ha creado, cuantos thumbs-up dispone, en que consiste (descripción y steps). Comentarios e indentificación de los usuarios que han hecho thumbs-up (votos positivos).

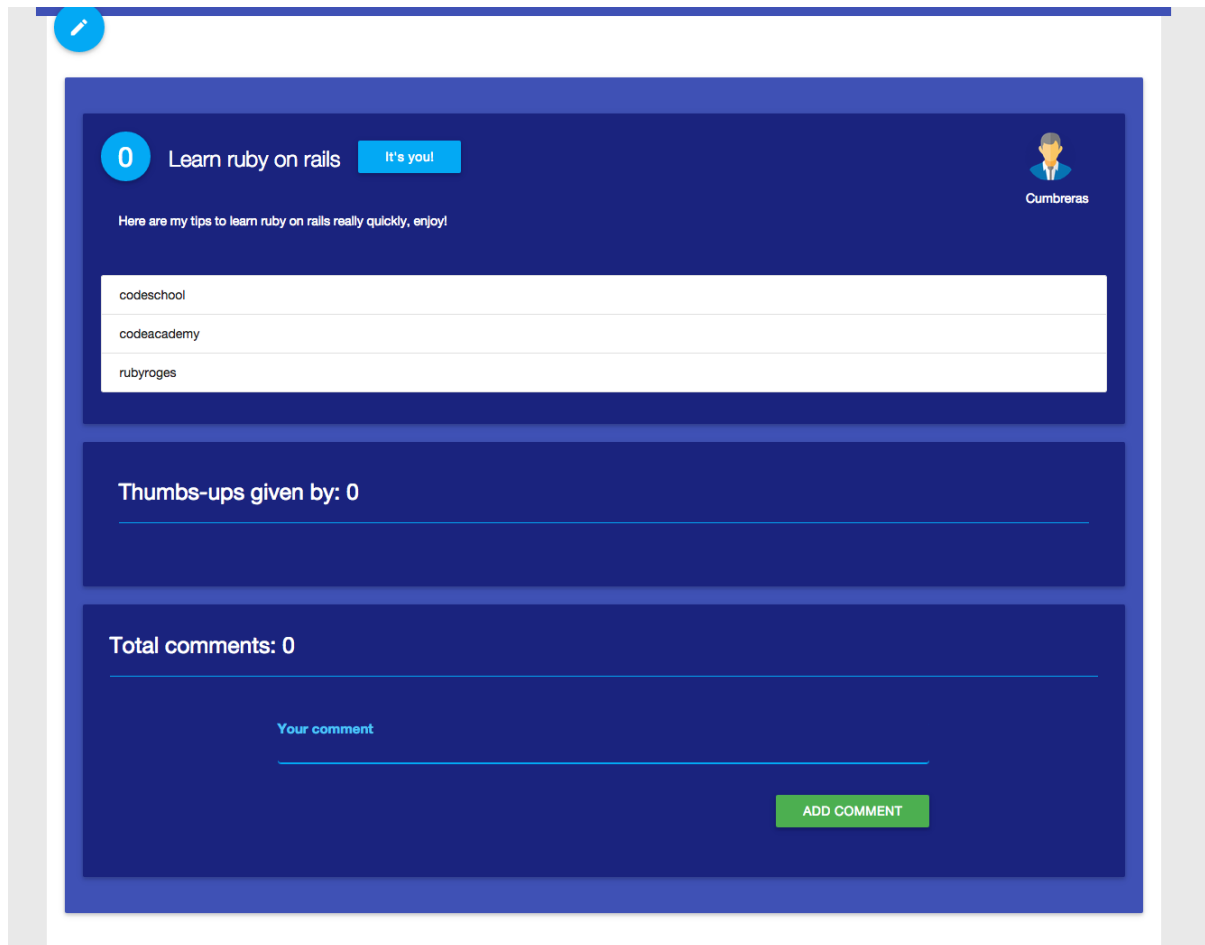


Figura 42. Captura de la sección unitaria del path



## **4.7 Comments (comentarios)**

Sección de comentarios específicos para un usuario, donde muestra todos los comentarios creados por un usuario desde su registro hasta el momento de visitar la página.

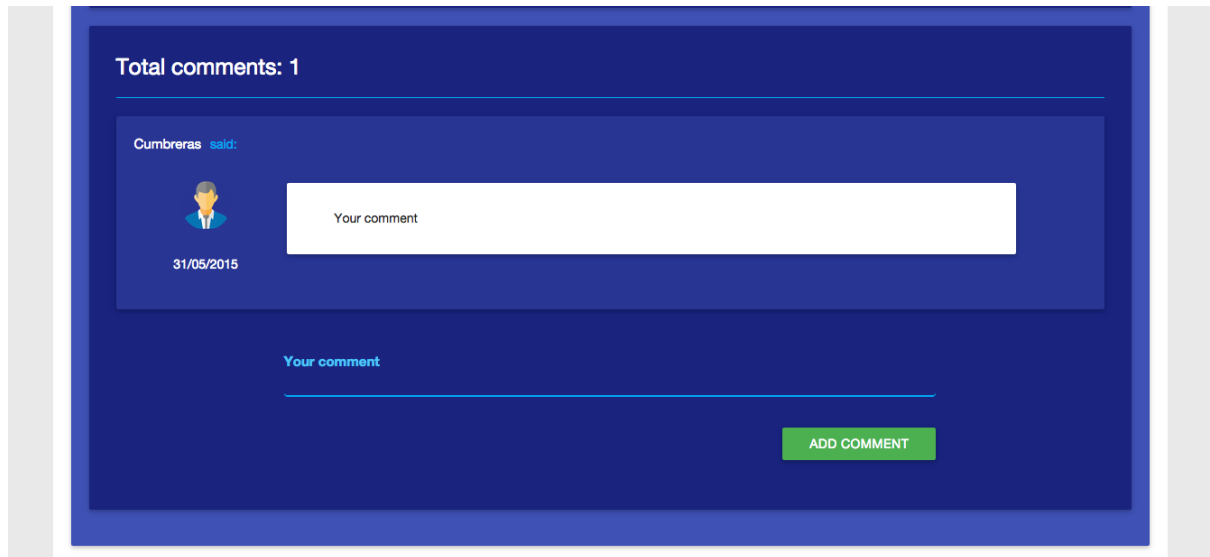


Figura 43. Captura de la sección de comentarios

## 4.8 Following y followers (Siguiendo y Seguido)

En esta sección se muestran los usuarios cuales el usuario dueño del perfil está siguiendo, existe otra sección similar a la misma la cual muestra la acción inversa, es decir, los usuarios que están siguiendo al usuario del perfil.

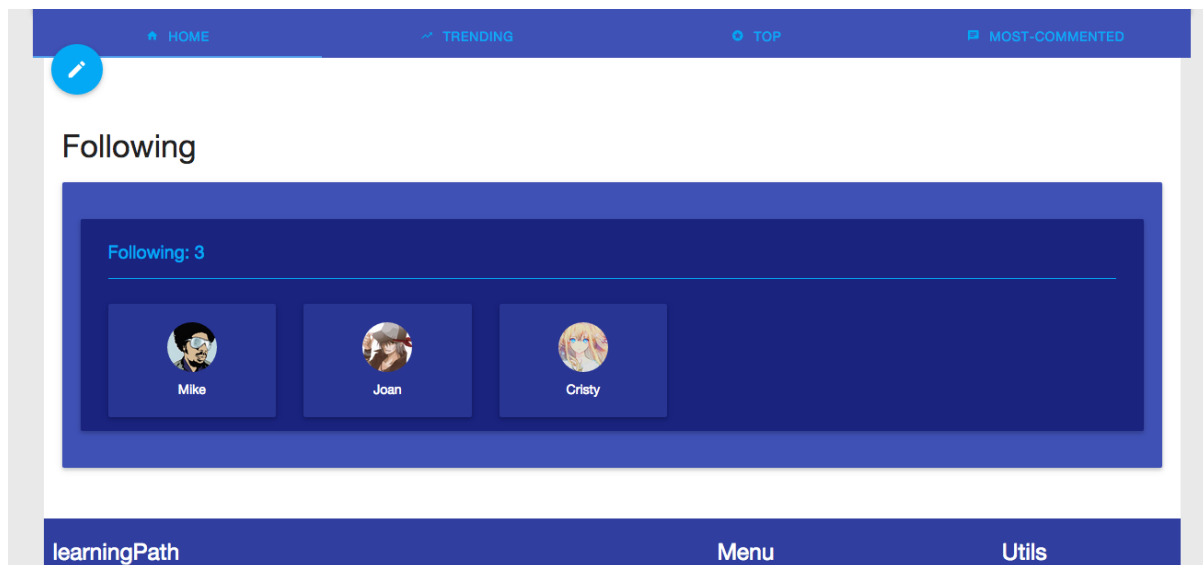


Figura 44. Captura de la sección de siguiendo

## 4.9 Perfil

En esta sección se muestran todos los paths generados por un usuario, así como la opción de editar si el dueño del perfil es el mismo que visita dicha página.

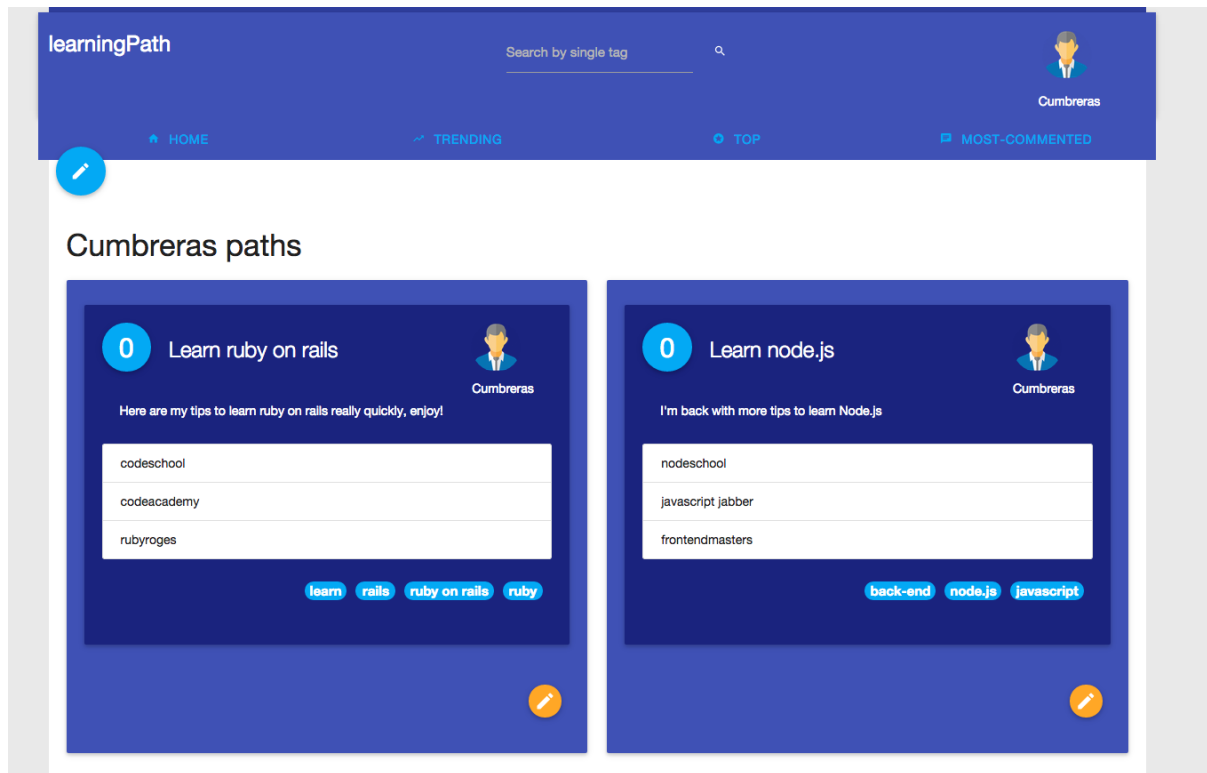


Figura 45. Captura de la sección del perfil de usuario con paths

## **Anexo 5. Glosario**

**API:** Interfaz de programación de aplicaciones. Representación del Sistema de comunicación para entre aplicaciones o sistemas de software.

**Bug:** Error o defecto de software, generalmente por un fallo en el desarrollo.

**Debugging:** Tarea programática para encontrar fallos y eliminarlos del sistema.

**Deployment:** Despliegue e implantación del código fuente al servidor encargado de ejecutarlo.

**FAQ:** Lista de pregunta y respuestas que surgen habitualmente dentro de un contexto determinado.

**Framework:** Estructura conceptual y tecnológica que sirve para la organización y desarrollo de software.

**Git:** Control de versiones, apoyo sistemático para el mantenimiento y gestión de archivos de código fuente.

**JSON:** Notación literal de objetos en JavaScript.

**REST:** Interfaz web con protocolos basados en patrones de intercambio de mensajes.

**MEAN Stack:** Infraestructura de la aplicación. Formada por MongoDB, Express.js, AngularJS y Node.js.

**Runtime:** Colección de funciones de utilidad las cuales son basadas en una arquitectura predefinida.

**Scrum:** Metodología de gestión de proyectos que se centra en el conjunto de prácticas, roles y tareas.

**Token:** Cadena de caracteres que sirve para identificar ciertas operaciones o conexiones en la aplicación.

**SSL:** Protocolo criptográfico que proporciona comunicaciones seguras por una red.

**TDD:** Práctica de ingeniería de software centrada el en modo de desarrollo del mismo, unificando criterios dirigido a la creación de pruebas primero.

**Wireframe:** Representación esquemática de una página web sin elementos gráficos que muestran su contenido.

## **Anexo 6. Bibliografía**

<http://aspiringcraftsman.com/>, “SOLID JavaScript: The Single Responsibility Principle”, [en línea], D. Greer, Dirección URL: [http://aspiringcraftsman.com/series/solid-javascript/], Consulta: [Abril, 2015]

<http://scrum.org>, “What is Scrum?”, [en línea], Desconocido, Dirección URL: [https://www.scrum.org/Resources/What-is-Scrum], Consulta: [Abril, 2015]

<http://extremeprogramming.com/>, “The rules of Extreme Programming”, [en línea], Desconocido, Dirección URL: [http://www.extremeprogramming.org/rules.html], Consulta: [Abril, 2015]

<http://agiledata.org/>, “Introduction to Test Driven Development [TDD]”, [en línea], Desconocido, Dirección URL: [http://agiledata.org/essays/tdd.html], Consulta: [Abril, 2015]

<http://webcredible.com/>, “User-centered design (UCD)”, [en línea], Clara Teoh, Dirección URL: [http://www.webcredible.com/blog-reports/web-usability/user-centered-design.shtml]Consulta: [Abril, 2015]

<http://acunetix.com/>, “XSS – The underestimated Exploit”, [en línea], Varios, Dirección URL: [https://www.acunetix.com/websecurity/xss/], Consulta: [Mayo, 2015]

<http://w3.org/>, “The principles of unobtrusive JavaScript”, [en línea], Varios, Dirección URL: [http://www.w3.org/wiki/The\_principles\_of\_unobtrusive\_JavaScript], Consulta: [Mayo, 2015]

## **Anexo 7. Biografía del autor**

Mi nombre es Edgar Cumbreras Tirado, nacido en Madrid en el año 1985 pero vivo y trabajo en Inglaterra desde el año 2009. Con anterioridad a esta carrera, estudié un grado medio en Imagen y sonido.

Desde hace dos años trabajo como desarrollador web para una pequeña *start-up* en Nothing Hill, Londres, anteriormente mi carrera ha sido principalmente enfocada en gerencia, administración de pequeñas y medianas empresas.

Mi pasión personal es el desarrollo web y la fotografía, por eso decidí enrolarme en la Universitat Oberta de Catalunya para estudiar este Grado de Multimedia.

En este trabajo fin de grado voy a desarrollar una aplicación para promocionar e incentivar el aprendizaje de nuevas tecnologías focalizado en el mundo de la informática y la programación. El concepto de dicha aplicación está ideado para poder devolver a la comunidad todo el apoyo que he recibido durante estos años estudiando en la UOC.

Mi idea es completar esta carrera para poder continuar estudiando en la UOC con un master en ingeniería de software.