



Desarrollo de un lector de fuentes RSS online

Memoria de Proyecto Final de Máster

Máster Universitario en Aplicaciones multimedia

Itinerario Profesional

Autor: Pedro Uceda Martínez

Profesor: Sergio Schvarstein Liuboschetz

Fecha de entrega 09/01/2017

Copyright



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada

[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Abstract

Through this End-of-Master's Project there have been carried out the preliminary study, analysis, design and development of a responsive web application whose goal is to manage in a intuitive and easy manner a set of RSS and Atom feeds that nowadays are provided by most of the websites on the net, most of them, but not only, online journals, and other kind of websites such as blogs.

This way, through the use of this application we provide another alternative to the users to search and subscribe to this feeds so they can access to all of their content from an unique application and having the change to check their e-mail due to the functionality of this application of sending an email, if desired, with the relevant contents of one or several subscriptions to the user.

So we could say that the main functionalities of this product are, among other functionalities, provide a way to search new feeds using the data provided by Google Feed API, that is used as feed search engine; the option to subscribe to one or several feeds and the possibility to check their content later, and the previously mentioned functionality of receiving an e-mail with the most relevant contents of one or more subscriptions.

During the development of this application we have followed the waterfall model due to the short project duration, getting in the end the goal application.

Resumen

Durante la elaboración de este Trabajo de Fin de Master se ha llevado a cabo el estudio preliminar, análisis, diseño e implementación de una aplicación web *responsive* que permite gestionar de una manera sencilla e intuitiva las distintas fuentes RSS y Atom que hoy día nos ofrecen los distintos sitios web, mayoritariamente diarios online, pero también otro tipo de sitios web como por ejemplo blogs.

Así, con esta aplicación se ofrece al usuario una alternativa más de aplicación para buscar y suscribirse a estas fuentes, para así poder acceder fácilmente a todos los contenidos de las mismas desde un punto único, y poder además estar actualizado cuando se desee ya que la misma enviará, previa configuración por parte del usuario, un sencillo correo a la hora que se desee en el que se listan los titulares que encajen con la configuración introducida en la aplicación.

De este modo, las funcionalidades principales de esta aplicación son, entre otras, tener un punto único de acceso para buscar estas fuentes, usando Google Feed API como motor de búsqueda de feeds, poder suscribirse a estas fuentes de modo que pueda visualizarse desde la aplicación misma los contenidos de estas suscripciones por parte del usuario en todo momento y poder recibir en el correo personal del usuario bajo demanda alertas diarias para los contenidos que se desee.

Durante el desarrollo de este proyecto, debido a la corta duración del mismo, se ha seguido un desarrollo siguiendo el modelo en cascada, obteniendo finalmente la aplicación deseada.

Palabras clave

Lector, RSS, Agregador, Aplicación Web, dinámica, filtros, analítica web.

Índice

Capítulo 1: Introducción	9
1. Introducción	9
2. Descripción	10
3. Objetivos generales	11
4. Metodología y proceso de trabajo.....	12
5. Planificación.....	14
6. Presupuesto	16
7. Estructura del resto del documento	17
Capítulo 2: Análisis	18
1. Estado del arte	18
2. Público objetivo y perfiles de usuario	20
3. Análisis de requisitos	21
1.1.1 Requisitos de información.....	21
1.1.2 Requisitos funcionales.....	22
1.1.3 Requisitos no funcionales	23
1.1.4 Casos de uso.....	24
Capítulo 3: Diseño	31
1. Descripción general	31
2. Arquitectura detallada	32
3. Diseño gráfico e interfaces	34
3.1 Estilos.....	34
3.2 Usabilidad/UX	34
4. Diseño de la Base de datos.....	36
4. Lenguajes de programación y APIs utilizadas	37
Capítulo 4: Implementación	45
1. Requisitos de instalación	45
2. Instrucciones de instalación.....	46
Capítulo 5: Demostración	49
1. Instrucciones de uso	49
2. Prototipos.....	50
1. Página principal – Consideraciones generales.....	50

2. Mi perfil.....	52
3. Ajustes.....	52
4. Buscar feeds.....	54
5. Ver más tarde.....	55
6. Todas mis suscripciones.....	56
7. Entradas de un feed	57
8. Avisos	58
9. Crear/Editar aviso	59
10. Versión mobile	60
3. Tests.....	61
4. Ejemplos de uso del producto.....	62
1. Pantalla principal, registro y autenticación	62
2. Pantalla Principal / Hoy	64
3. Buscar feeds.....	66
4. Suscripciones	68
5. Ver más tarde.....	72
6. Avisos (Ver todos)	73
Capítulo 6: Conclusiones y líneas de futuro	75
1. Conclusiones	75
2. Líneas de futuro	76
Bibliografía	77

Figuras y tablas

Índice de figuras

Ilustración 1 Interfaz de Google Reader	18
Ilustración 2 Interfaz de Feedly.....	19
Ilustración 3 Arquitectura detallada de la aplicación.....	32
Ilustración 4 Logotipo de la aplicación	34
Ilustración 5 Barra superior de la aplicación	34
Ilustración 6 Menú lateral de la aplicación	35
Ilustración 7 Menú móvil de la aplicación.....	35
Ilustración 8 Diagrama de base de datos.....	36
Ilustración 9 Estructura del front-end de la aplicación	37
Ilustración 10 Estructura del back-end de la aplicación	40
Ilustración 11 Clase de configuración de la aplicación	41
Ilustración 12 Ejemplo de controlador REST.....	42
Ilustración 13 Ejemplo de DAO.....	42
Ilustración 14 Ejemplo de entity.....	43
Ilustración 15 Ejemplo de servicio	43
Ilustración 16 Configuración de correo en application.properties.....	46
Ilustración 17 Fichero startup.bat	47
Ilustración 18 Log de la aplicación en la consola de windows	48
Ilustración 19 Vista de la aplicación en dispositivo de escritorio	49
Ilustración 20 Vista de la aplicación en dispositivo móvil.....	49
Ilustración 21 Mockup - página principal.....	50
Ilustración 22 Mockup - crear una cuenta	51
Ilustración 23 Mockup - Iniciar sesión.....	51
Ilustración 24 Mockup - Página principal autenticado.....	52
Ilustración 25 Mockup - Mi perfil.....	52
Ilustración 26 Mockup - Mi perfil: Ajustes.....	53
Ilustración 27 Mockup - Buscar feeds.....	54
Ilustración 28 Mockup - Ver más tarde	55
Ilustración 29 Mockup - Suscripciones.....	56
Ilustración 30 - Mockup – suscripción.....	57
Ilustración 31 Mockup - Ver noticia en suscripción.....	57
Ilustración 32 Mockup – Alertas (Opciones de alerta)	58
Ilustración 33 Mockup - Alertas	58
Ilustración 34 Mockup - Crear aviso	59
Ilustración 35 Mockup - Editar aviso	59
Ilustración 36 Mockup - Version mobile 1	60
Ilustración 37 Mockup - Versión mobile 2	60
Ilustración 38 Pantalla principal sin autenticarse en la aplicación	62
Ilustración 39 Crear una cuenta.....	62
Ilustración 40 Crear una cuenta (registro realizado).....	63

Ilustración 41 Inicio de sesión	64
Ilustración 42 Hoy	65
Ilustración 43 Visualización de contenido de noticia.....	65
Ilustración 44 Guardar entrada.....	66
Ilustración 45 Buscar feeds - cargando.....	66
Ilustración 46 - Buscar feeds (con imagen).....	67
Ilustración 47 Buscar feeds (sin imagen)	67
Ilustración 48 Suscribirse a un feed.....	68
Ilustración 49 Suscripciones en menú lateral.....	69
Ilustración 50 Ver una suscripción.....	70
Ilustración 51 Ver noticia de una suscripción	71
Ilustración 52 Guardar entrada de una suscripción	71
Ilustración 53 Filtros en suscripción.....	72
Ilustración 54 Modificar datos de suscripción.....	72
Ilustración 55 Ver más tarde.....	72
Ilustración 56 Avisos	73
Ilustración 57 Nuevo aviso	73
Ilustración 58 Ejemplo de aviso por correo	74

Índice de tablas

Tabla 1. Planificación del proyecto.....	14
--	----

Capítulo 1: Introducción

1.Introducción

Mediante este Trabajo de Fin de Master se aborda la creación de un gestor o lector de fuentes de RSS, tipo de aplicación que es usada para mantenerse actualizado sobre los contenidos de distintos sitios web que interesan a un usuario en concreto.

Este tipo de aplicaciones es muy beneficioso para todos aquellos que suelen chequear a diario y de manera sistemática sitios web de diarios o blogs de su gusto, para ver qué contenidos o noticias han aparecido desde la última vez que visitó la web, puesto que ofrecen acceso a todos estos contenidos actualizados desde un único punto, el lector de fuentes RSS, fuentes que son ofrecidas por lo general por todos los sitios web que actualicen sus contenidos con frecuencia.

Así, en este TFM se desarrolla un lector de fuentes RSS al que se le ha dado el nombre de MyFeed. Mediante este desarrollo se espera aprender sobre cómo se gestionan estas fuentes RSS a la vez que gestionar y ejecutar el desarrollo de una aplicación web completa.

2. Descripción

En este Trabajo de Fin de Máster se llevará a cabo el análisis, diseño e implementación de una aplicación web responsive que permita gestionar de una forma interactiva, dinámica y eficiente las distintas fuentes RSS que los blogs, diarios y demás sitios webs suelen ofrecer.

Esta aplicación deberá ser usable e intuitiva, permitiendo gestionar todas esas fuentes y, consecuentemente, las noticias contenidas en cada una de ellas, de manera personalizable. Entre otras funcionalidades, se ofrecerá la posibilidad de gestionar las fuentes de una manera lógica como por ejemplo agruparlas por sitio web, o bien por categoría. Un ejemplo de ello sería agrupar en deportes la sección “fútbol” de El País y la de “Motor” de El Mundo.

Actualmente en el mercado encontramos que prácticamente todas las webs nos ofrecen sus contenidos, además de la manera tradicional, la web en sí, a través de fuentes RSS, ofreciéndonos sus noticias o eventos actualizados en este formato, que es un XML.

Así, si se desea leer estas fuentes, es necesario el uso de un lector o agregador de RSS. De entre este tipo de programas, el principal ha sido Google Reader que, por algún motivo, fue discontinuado o abandonado. Esto hizo que este nicho de mercado quedase libre, surgiendo otros lectores nuevos o quedándose otros pre-existentes como alternativas. Entre ellos, Outlook (tiene opción de añadir fuentes RSS), Feedly (considerado actualmente el mejor), Inoreader... Cada uno con sus pros y sus contras.

Así, si bien este proyecto no propone una idea rompedora y absolutamente nueva, encontramos que si realizamos un trabajo bien hecho sí que podríamos hacernos un hueco dentro de las aplicaciones de lectura de RSS actuales, dado que no existe un monopolio claro en este sector como ocurre en la mayoría de casos en el mundo de la informática, y podemos aprender de los errores de las aplicaciones existentes de este tipo y desarrollar una mejor.

Así, la motivación principal de este proyecto es, por una parte, realizar un sistema ideado por el alumno desde cero que permitirá aplicar diversos temas abordados durante este máster, entre ellos desarrollo web, usabilidad y gestión de proyectos; y por otra parte el poder desarrollar una aplicación web cuyo sector o nicho no tiene un monopolio claro, por lo que podría hacerse un hueco y competir con las aplicaciones ya existentes.

3. Objetivos generales

A continuación, se listan los objetivos del proyecto:

Objetivos de la aplicación/producto/servicio:

- Diseño y desarrollo de una aplicación web responsive que permita suscribirse a fuentes RSS y gestionar las suscripciones a las mismas.
- Que dicha aplicación web esté preparada para un posterior desarrollo de apps móviles.
- Desarrollar funcionalidades diferenciadoras de la competencia e.g. envío de avisos vía e-mail.

Objetivos para el cliente/usuario:

- Disponer de un punto único desde el que gestionar todas las fuentes
- Gestión de los feeds suscritos por parte del usuario por categorías y permitir almacenarlos para visualizarlos más tarde.
- Estar constantemente informado según las preferencias del usuario.

Objetivos personales del autor del TF:

- Conocer más a fondo y usar frameworks de la parte front-end para desarrollar un proyecto íntegro.
- Aplicar los estándares de marcado HTML5 y CSS3

4. Metodología y proceso de trabajo

Para llevar a cabo este proyecto se ha seguido una metodología de desarrollo en cascada, con el objetivo de completar tanto una memoria del proyecto como la aplicación web producto del mismo.

En primer lugar, se ha llevado a cabo un estudio preliminar para conocer el *estado del arte* en cuanto a lectores de RSS se refiere, de modo que se ha seguido un proceso de investigación de qué aplicaciones existen en el mercado actualmente y cuáles son sus puntos débiles y fuertes, tal y como puede verse en el apartado Descripción de esta memoria. Acto seguido se definieron los objetivos del proyecto, que sirvieron como guía durante el resto del desarrollo del mismo.

Posteriormente, se realizó una planificación de las tareas a realizar para la PEC2, que en la PEC3 se detalló tomando como guía los casos de uso que se identificaron en esta entrega, ya que las tareas de implementación en esta planificación preliminar se describieron a *grosso modo*. En cualquier caso, el tiempo estimado para la implementación permaneció invariable.

Una vez establecida la planificación, se llevó a cabo una toma de requisitos, con requisitos de tres tipos:

- Requisitos de información, que definen qué información ha de gestionar la aplicación.
- Requisitos funcionales, que esbozan el comportamiento que ha de cumplir la aplicación.
- Requisitos no funcionales, mediante los cuales se listan atributos deseables del sistema.
- Casos de uso, que definen detalladamente cómo se debe comportar el sistema, tanto a nivel interno como a nivel de interacción con el usuario.

Se ha llevado a cabo esta toma de requisitos debido a que facilita en gran medida qué funcionalidades han de desarrollarse y cuáles no, además de sentar las bases para el diseño de la base de datos del sistema tomando como entrada los requisitos funcionales. Además, tomando como base los casos de uso, se ha podido también realizar un testeo de la aplicación usándolos como guía.

Luego de haber realizado la toma o *elicitación* de requisitos, se ha procedido a realizar un prototipo de alta fidelidad desarrollado usando HTML5, CSS3, Bootstrap y jQuery. Se optó por un prototipo de alta fidelidad (Hi-Fi) en lugar de uno de baja fidelidad (Lo-Fi) debido a que de este modo se ha podido reusar el prototipo en sí para desarrollar la aplicación web real, que sería previsiblemente desarrollada usando HTML5 y CSS3, y efectivamente para el *fron-end* de la misma se usó AngularJS2, por lo que se pudo tomar el HTML5 y el CSS3 del prototipo íntegro como las diferentes plantillas de Angular usadas en la aplicación final, con las modificaciones pertinentes para incorporar funcionalidad.

Después de haberse realizado el prototipo, se procedió a establecer la arquitectura del sistema, es decir, qué tecnologías se usarán para desarrollar las distintas partes del sistema a nivel general y cómo interactuarán entre ellas. Esta arquitectura se ilustra en la sección Diseño de esta memoria.

Una vez definida la arquitectura del sistema, se comenzó la fase de Implementación del proyecto, con un previo levantamiento del esqueleto del mismo. Durante esta fase, se desarrollaron las distintas tareas del proyecto tomando como unidad a desarrollar un caso de uso cada vez. Una vez completada la implementación del caso de uso, y tras un breve testeo, se pasó al caso de uso siguiente.

Finalmente, se llevó a cabo el testeo completo de la aplicación, tomando una vez más como guía los anteriormente mencionados casos de uso, comprobando que la aplicación se comportase de acuerdo a lo definido en cada uno de ellos, y posteriormente se realizó el empaquetado de la misma y despliegue en un servidor externo para que sea accesible.

5. Planificación

A continuación, se presenta la planificación seguida a lo largo del desarrollo del proyecto:

Tabla 1. Planificación del proyecto

Tarea	Duración (días)	Inicio	Final
PEC1	27	25/10/2016	21/11/2016
Elaboración de la PEC1			
PEC2	27	25/10/2016	21/11/2016
Elaboración de la PEC2			
PEC3	27	25/10/2016	21/11/2016
Análisis de requisitos	6	25/10/2016	31/10/2016
Recopilación/uso de API de RSS	4	31/10/2016	03/11/2016
Arquitectura de la aplicación	3	04/11/2016	07/11/2016
Diagramas	1	07/11/2016	08/11/2016
Mockups	10'5	9/11/2016	20/11/2016
Redacción de documentación	1'5	20/11/2016	21/11/2016
PEC4	27	22/11/2016	19/12/2016
Diseño de la API	2	22/11/2016	24/11/2016
Diseño de la BB.DD	5	25/11/2016	29/11/2016
Estudio de tecnologías	5	29/11/2016	04/12/2016
Levantamiento del back-end	2	04/12/2016	06/12/2016
Levantamiento del front-end	2	06/12/2016	08/12/2016
UC-0001 Registro de usuario	3	08/12/2016	12/12/2016
UC-0002 Modificar datos de usuario	2	12/12/2016	14/12/2016
UC-0003 Recuperar contraseña	1	15/12/2016	15/12/2016
UC-0004 Buscar un feed	1	16/12/2016	16/12/2016
UC-0005 Suscribirse a un feed	1	17/12/2016	17/12/2016
UC-0006 Modificar una suscripción	1	18/12/2016	18/12/2016
UC-0007 Eliminar una suscripción	1	19/12/2016	19/12/2016
Cierre	20	20/12/2016	09/01/2017
UC-0008 Generación de nube de palabras	2	20/12/2016	21/12/2016
UC-0009 Almacenar noticias a ver más tarde	2	22/12/2016	23/12/2016
UC-0010 Visualizar suscripciones y noticias almacenadas	2	24/12/2016	25/12/2016
UC-0011 Guardar un aviso por correo	3	26/12/2016	28/12/2016
UC-0012 Realizar un aviso por correo	3	29/12/2016	01/01/2017

UC-0013 Filtrar suscripciones	3	02/01/2017	04/12/2016
Pruebas y resolución de incidencias	4	05/01/2017	09/01/2017
Memoria		26/09/2016	09/01/2017
Redacción y corrección de la memoria	76	25/10/2016	09/01/2017
Defensa del proyecto			
Preparacion de defensa	17	10/01/2017	27/01/2017

6. Presupuesto

Para la realización de este proyecto, se ha invertido un total de 76 días, con un tiempo medio de horas/día de 4,5h. Así, el desglose del coste aproximado de cada tarea, en horas, corresponde al siguiente:

- Idea, estudio preliminar y bases del proyecto: 120h
- Análisis: 45h
- Diseño: 108h
- Implementación: 130h
- Tests: 16h

Estas horas se refieren al trabajo de una sola persona. Además, hay que incluir en el presupuesto un PC portátil con un procesador AMD10 y 8Gb de Ram y el uso de Amazon AWS para el despliegue de la aplicación, que es gratuito durante los primeros 12 meses.

7. Estructura del resto del documento

A continuación, se listan qué capítulos componen el resto del documento y se describe brevemente su contenido:

- **Capítulo 2: Análisis.** En este capítulo se describe cuál es la situación actual en cuanto a lectores de RSS en el mercado y cómo se han obtenido y cuáles son los requisitos que deberá cumplir nuestra aplicación.
- **Capítulo 3: Diseño.** Se describe en términos generales la arquitectura usada y decisiones tomadas a la hora de implementar la aplicación.
- **Capítulo 4: Implementación.** Durante este capítulo se explica cuáles son los requisitos de instalación y cómo ha de instalarse la aplicación desarrollada.
- **Capítulo 5: Demostración.** En este capítulo se aporta la URL de la aplicación final y se muestran capturas de pantalla del prototipo desarrollado inicialmente, así como ejemplos de uso de la aplicación.
- **Capítulo 6: Conclusiones y líneas de futuro.** Durante este capítulo se exponen las conclusiones de la realización de proyecto; qué fue bien y que fue mal. Se describen también *grosso modo* los trabajos de mejora/continuación que podrían realizarse.

Capítulo 2: Análisis

1. Estado del arte

Un lector o recopilador de contenidos es un tipo de software que permite suscribirse a fuentes de noticias en formatos RSS, Atom y otros derivados del formato XML. Normalmente, este tipo de software reúne las noticias o historias publicadas en los sitios elegidos (fuentes web) y muestra las novedades y modificaciones de los mismos.

Actualmente existen multitud de programas con esta funcionalidad. A continuación se presentan los de mayor relevancia, con sus principales características:

- **Google Reader.** Si bien esta aplicación se abandonó, sí tuvo relevancia desde 2005 a julio de 2013, fecha de su cierre. Entre sus funcionalidades se ha de destacar que su interfaz era similar a la de un correo electrónico, permitía compartir contenidos por correo electrónico o bien a modo de “blog personal”. En cuanto la organización, permitía que se seleccionasen noticias como “favoritas”, registrándose y pudiéndose verse más tarde. Además, también permitió durante cierto tiempo que se usase offline.

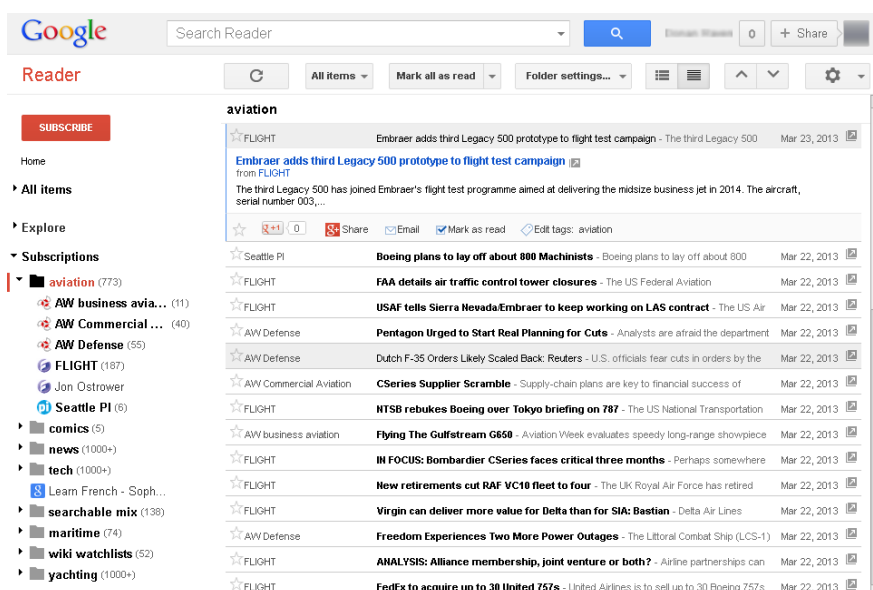


Ilustración 1 Interfaz de Google Reader

- **Feedly.** Actualmente es el más popular y usado. Su potencial está en que permite buscar los feeds que nos interesan desde su misma página, buscando por palabras clave y mostrándonos una lista de feeds o fuentes posibles, que podemos asociar a una lista. Nos permite organizar los feeds en Todos, Favoritos o bien en nuestras listas propias. Una vez suscritos a nuestros feeds, en “Today”, nos aparecerán listados de arriba abajo nuestras colecciones con las últimas actualizaciones.

Al igual que hacía Google Reader, nos permite guardar noticias para más tarde, compartir nuestras colecciones o listas además a través de redes sociales y también posee una nueva

característica a partir de octubre de 2016 llamada “Integración”, que permite enviar noticias a Google Docs, Trello, publicarlas en Wordpress...

Asimismo también tiene una versión Android, además de la web.

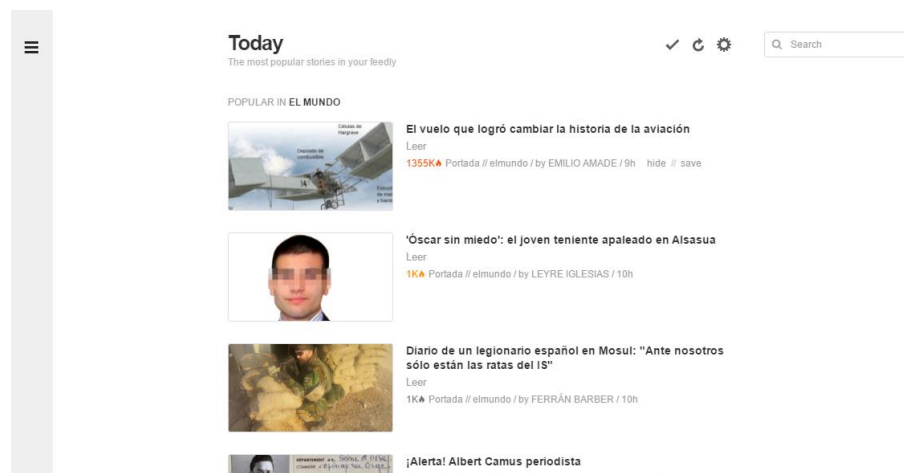


Ilustración 2 Interfaz de Feedly

- Otra aplicación es **Inoreader**. Posee una interfaz limpia y prácticamente la misma funcionalidad que Feedly, al que añade lo que en esta web se denomina “estadísticas” sobre los feeds y la posibilidad de trabajar en equipos.
- Existen otras webs con funcionalidad similar, como **NewsBlur**, **G2Reader** y **AOL Reader**. Comentar que todas las mencionadas actualmente tienen versión free y versión de pago.
- Comentar además el caso del diario El Mundo (www.elmundo.es), que propone retos técnicos relacionados con sus fuentes RSS, premiando a quien desarrolle aplicaciones que los usen. Ejemplos de los tales son una nube de términos, titulares para el escritorio, barras laterales para diversos navegadores...

Por lo tanto, puede decirse que a grandes rasgos las **funcionalidades representativas de este tipo de aplicaciones** actualmente son las siguientes:

- Una interfaz intuitiva, simple y limpia.
- Posibilidad de encontrar y suscribirse a feeds o fuentes desde la misma aplicación.
- Organización de los feeds, normalmente en categorías. Los mismos se pueden filtrar posteriormente según las etiquetas o categorías con las que se hayan clasificado por parte del usuario.
- Posibilidad de almacenar las noticias que se desee para visualizarlas más tarde.
- Posibilidad de compartir noticias por redes sociales.

Con el desarrollo de nuestra aplicación se ha intentado abordar todas estas funcionalidades, además de incluir otras funcionalidades extra que la diferencien del resto.

2. Público objetivo y perfiles de usuario

El público *target* de esta aplicación es toda aquella persona que haga uso de internet a diario para informarse sobre temas de actualidad visitando diarios online u otro tipo de sitios web como blogs, sea desde dispositivos móviles como de escritorio.

Concretamente, se identifican dos tipos de posibles usuarios:

- Personas que sólo acceden al tipo de sitios web mencionados, pero que nunca han usado un lector de fuentes RSS.
- Personas que ya hagan uso de los distintos lectores de fuentes RSS disponibles. Especialmente este último perfil es el que más nos interesa, dado que son usuarios que ya están familiarizados con este tipo de aplicaciones.

3. Análisis de requisitos

Se ha llevado a cabo la toma y análisis de los requisitos del proyecto, capturando requisitos de tipo funcional, no funcional, de información (datos que la aplicación ha de conservar) y se han plasmado en una serie de caso de usos que ilustran los comportamientos de la aplicación. Así, en el apartado que nos ocupa se describe de una manera formal, mediante esta *elicitación* de requisitos, qué se espera de nuestra aplicación.

1.1.1 Requisitos de información

A continuación se procede a describir la información que ha de gestionar la aplicación:

IRQ-0001	Información de usuario
Descripción	El sistema deberá almacenar la información correspondiente a la información sobre los usuarios. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre de usuario • Contraseña • Nombre • Apellido(s) • Correo electrónico o email

IRQ-0002	Feed (Suscripción)
Descripción	El sistema deberá almacenar la información correspondiente a un feed al que se haya suscrito un usuario. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Url • Etiqueta o etiquetas • Categoría

IRQ-0003	Avisos por correo
Descripción	El sistema deberá almacenar la información correspondiente a los avisos que el usuario desee recibir en su correo electrónico. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Feed/Suscripción o feeds • Usuario • Etiquetas • Palabras para filtrar • Hora de envió

IRQ-0004	Noticia a visualizar más tarde
Descripción	El sistema deberá almacenar la información correspondiente a las noticias que un usuario haya marcado para visualizar más tarde. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Fuente • Titular • Contenido • Fecha de publicación • Categorías

IRQ-0005	Cuentas de redes sociales
-----------------	----------------------------------

Descripción	El sistema deberá almacenar la información correspondiente a la información sobre las cuentas de redes sociales, si lo desea el usuario, para poder compartir fácilmente las noticias.
--------------------	--

1.1.2 Requisitos funcionales

Se describen a continuación los requisitos funcionales que debe cumplir la aplicación:

FRQ-0001	Registro de usuario
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información de usuario • [UC-0001] Registro de usuario
Descripción	El sistema deberá <i>permitir a un usuario su registro para que pueda almacenar los feeds</i> o suscripciones que desee.

FRQ-0002	Modificación de datos
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información de usuario
Descripción	El sistema deberá <i>permitir modificar todos los datos de usuario excepto el nombre de usuario, que es inmutable, si el usuario lo desea.</i>

FRQ-0003	Recuperación de contraseña
Dependencias	Ninguno
Descripción	El sistema deberá permitir al usuario recuperar la contraseña en caso de haberla olvidado.

FRQ-0004	Búsqueda de feeds
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir a los usuarios visualizar una lista de feeds según las palabras que introduzcan.</i>

FRQ-0005	Suscripción a un feed
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir a un usuario suscribirse a un feed, introduciendo una etiqueta para clasificarlo.</i>

FRQ-0006	Modificación y eliminación de suscripción
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir a un usuario editar una suscripción o bien eliminarla.</i>

FRQ-0007	Nube de palabras
-----------------	-------------------------

Dependencias	Ninguno
Descripción	El sistema deberá <i>generar y mostrar al usuario una nube de palabras con las palabras con más ocurrencias de los titulares de los feeds a los que está suscrito.</i>

FRQ-0008	Almacenar noticias a visualizar más tarde
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir a un usuario guardar una noticia para poder visualizarla más tarde.</i> <i>Dicha organización podrá ser en carpetas / forma de árbol.</i>

FRQ-0009	Avisos por correo
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Avisos por correo
Descripción	El sistema deberá <i>permitir al usuario almacenar la información de un aviso por correo, introduciendo feeds, categorías, palabras por las que filtrar y una hora.</i> <i>El sistema deberá recopilar la información y enviarla al usuario a la hora indicada.</i>

FRQ-0010	Filtrar suscripciones
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir a un usuario filtrar noticias de los feeds a los que esté suscrito según fuente, categoría, palabra, o fecha/hora de publicación.</i>

FRQ-0011	Compartir noticias en redes sociales
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario compartir una noticia en redes sociales.</i>

1.1.3 Requisitos no funcionales

Son deseables las siguientes características de funcionamiento:

NFR-0001	Visualización en navegadores
Dependencias	Ninguno
Descripción	El sistema deberá <i>visualizarse en los navegadores mayoritarios, tanto para dispositivos de escritorio como móviles</i>

NFR-0002	Desarrollo de API REST
Dependencias	Ninguno

Descripción	El sistema deberá <i>ofrecer una API REST que usará actualmente y posteriormente será usada para desarrollo de apps móviles.</i>
--------------------	--

NFR-0003	Seguridad de las contraseñas
Dependencias	Ninguno
Descripción	El sistema deberá <i>Las contraseñas han de almacenarse cifradas</i>

1.1.4 Casos de uso

Se espera que el sistema sea capaz de interactuar con el usuario tal y como se especifica en los siguientes casos de uso:

UC-0001	Registro de usuario	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario pulse en el botón para registrarse</i>	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El sistema <i>mostrará el formulario de registro.</i>
	2	El usuario <i>introducirá nombre de usuario, contraseña, nombre, apellidos y dirección de correo electrónico.</i>
	3	El sistema <i>comprobará si existen el nombre de usuario y/o email en el sistema.</i>
	4	Si <i>existen el nombre de usuario o correo en el sistema, el sistema informará al usuario y requerirá cambiar tales datos.</i>
5	Si <i>no se encuentra en el sistema el nombre de usuario ni contraseña y el email es correcto, el sistema salvará los datos y se enviará un correo al usuario. Informará al usuario de que ya está registrado y dará acceso de nuevo a la aplicación, ya logado.</i>	
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0002	Modificar datos de usuario
Dependencias	Ninguno
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El usuario pulse en el botón de modificar sus datos</i>

Precondición	El usuario se encuentra ya registrado y autenticado	
Secuencia normal	Paso	Acción
	1	El sistema <i>recuperará los datos del usuario (nombre de usuario, nombre y correo) y mostrará un formulario en el que permita cambiar los dos últimos datos.</i>
	2	El actor Usuario (ACT-0001) <i>cambiará los datos que desee y enviará el formulario</i>
	3	Si <i>el formulario es correcto</i> , el sistema <i>modificará tales datos e informará al usuario.</i>
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0003	Recuperar contraseña	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario seleccione la opción de recuperar la contraseña</i>	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El sistema <i>mostrará un formulario en el que se muestre un botón para enviar un correo al email del usuario.</i>
	2	El usuario <i>si lo desea, pulsará el botón.</i>
	3	El sistema <i>enviará un correo al usuario con un enlace.</i>
	4	El actor usuario <i>abrirá su email y pulsará en el enlace enviado</i>
	5	El sistema <i>abrirá otro formulario en el que aparezcan los campos contraseña y repetir contraseña</i>
	6	El actor usuario <i>rellenará los campos y pulsará enviar.</i>
	7	Si <i>los datos son correctos</i> , el sistema <i>modificará la contraseña.</i>
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0004	Buscar un feed	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso	

	cuando el usuario seleccione que desea buscar un feed.	
Precondición	El usuario haya accedido a la web	
Secuencia normal	Paso	Acción
	1	El usuario introducirá los términos a buscar para obtener un feed y pulsará buscar
	2	El sistema ejecutará una petición AJAX a la API de Google, parseará los términos al formato correcto, obtendrá los resultados y los listará bajo el elemento en el que se buscó, dando la posibilidad de seleccionarlos como suscritos.
	3	El usuario podrá visualizar un feed, haciendo click sobre el mismo.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0005	Suscribirse a un feed	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando tras buscar feeds, se selecciona uno de ellos para suscribirse.	
Precondición	Se ha ejecutado el caso de uso [UC-004] Buscar un feed.	
Secuencia normal	Paso	Acción
	1	El usuario seleccionará la opción 'suscribirse' sobre un feed listado.
	2	El sistema <i>requerirá al usuario introducir una etiqueta o categoría para el feed</i>
	3	El usuario introducirá la categoría o seleccionará una ya introducida y pulsará aceptar
	4	El sistema guardará la información del feed asociada al usuario y la categoría deseada
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0006	Modificar una suscripción	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario indique que desea modificar una suscripción.	

Precondición	-	
Secuencia normal	Paso	Acción
	1	El usuario seleccionará un feed con la opción modificar feed
	2	El sistema mostrará qué puede modificar del feed, de momento la categoría
	3	El usuario seleccionará la nueva categoría deseada
	4	El sistema reasignará el feed a la nueva categoría
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0007	Eliminar una suscripción	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee eliminar un feed o suscripción.	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El usuario seleccionará la opción eliminar un feed
	2	El sistema preguntará al usuario si está seguro de realizar la acción
	3	Si el usuario selecciona sí, el sistema el sistema eliminará el feed y, si es el único de una categoría, ésta última.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0008	Generación de nube de palabras	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se acceda a la aplicación, caso en el que se mostrará una nube de palabras.	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El usuario se autentique
	2	El sistema recuperará las noticias de las suscripciones del usuario y generará la nube de palabras, con el tamaño de las palabras proporcional a

		las ocurrencias de la misma.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0009	Almacenar noticias a ver más tarde	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee almacenar una noticia para poder verla en otro momento.	
Precondición	El usuario haya visualizado una noticia de un feed o suscripción.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona una noticia para guardarla.
	2	El sistema guardará la noticia junto con la hora y día de la misma y la fuente o feed.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0010	Visualizar suscripciones y noticias almacenadas	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee visualizar noticias para ver más tarde o suscripciones.	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El usuario hace inicio de sesión o ver noticias y suscripciones
	2	El sistema recuperará de BBDD las suscripciones y noticias del usuario y las mostrará.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0011	Guardar un aviso por correo	
Dependencias	Ninguno	

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee crear un aviso por correo.	
Precondición	El usuario tenga al menos una suscripción	
Secuencia normal	Paso	Acción
	1	El usuario seleccionará la opción generar un aviso por correo
	2	El sistema mostrará el formulario de avisos por correo
	3	El usuario seleccionará feeds suscritos, introducirá palabras clave y a qué hora desea recibir el aviso, así como un nombre para el mismo. Posteriormente pulsará guardar.
4	El sistema salvará la información en BB.DD.	
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0012	Realizar un aviso por correo	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se deba enviar un aviso por correo activo.	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El sistema tendrá que ir comprobando cada poco tiempo los avisos que hay salvados, en caso de coincidir alguno: Se realizará el filtrado de los feeds según la información del aviso. Se montará el mensaje con esas noticias que pasen el filtro, incluyendo el nombre del aviso. Se enviará a la dirección de correo del usuario asociado el aviso.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

UC-0013	Filtrar suscripciones	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desee filtrar noticias de una suscripción.	

Precondición	-	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) selecciona que quiere realizar filtrado en sus suscripciones
	2	El sistema <i>mostrará las opciones de filtrado.</i>
	3	El actor Usuario (ACT-0001) introducirá palabras clave y seleccionará los feeds sobre los que filtrar
	4	El sistema recorrerá las noticias de los feeds, los filtrará y los devolverá.
Postcondición	-	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-

Capítulo 3: Diseño

1. Descripción general

La aplicación a desarrollar será una aplicación web que estará compuesta por dos partes bien definidas:

- El front-end de la aplicación, con el que el usuario interactuará y que en última instancia estará compuesto por código HTML, CSS y Javascript que correrá en el navegador del usuario.
- El back-end de la aplicación, que recibe las peticiones del front end a través de una API de tipo REST que expone, procesando las mismas según la lógica necesaria y almacenando/actualizando información en base de datos cuando proceda.

El desacoplamiento de esta arquitectura se justifica principalmente en base a que el back-end de la aplicación sea siempre el mismo, único, pero que a futuros además de poderse interactuar con el back-end de la aplicación desde el front-end desarrollado, sea posible que otras aplicaciones, tipo móvil, consuman los servicios ofrecidos por el mismo.

2. Arquitectura detallada

A continuación, se presenta el diagrama de la arquitectura de la aplicación:

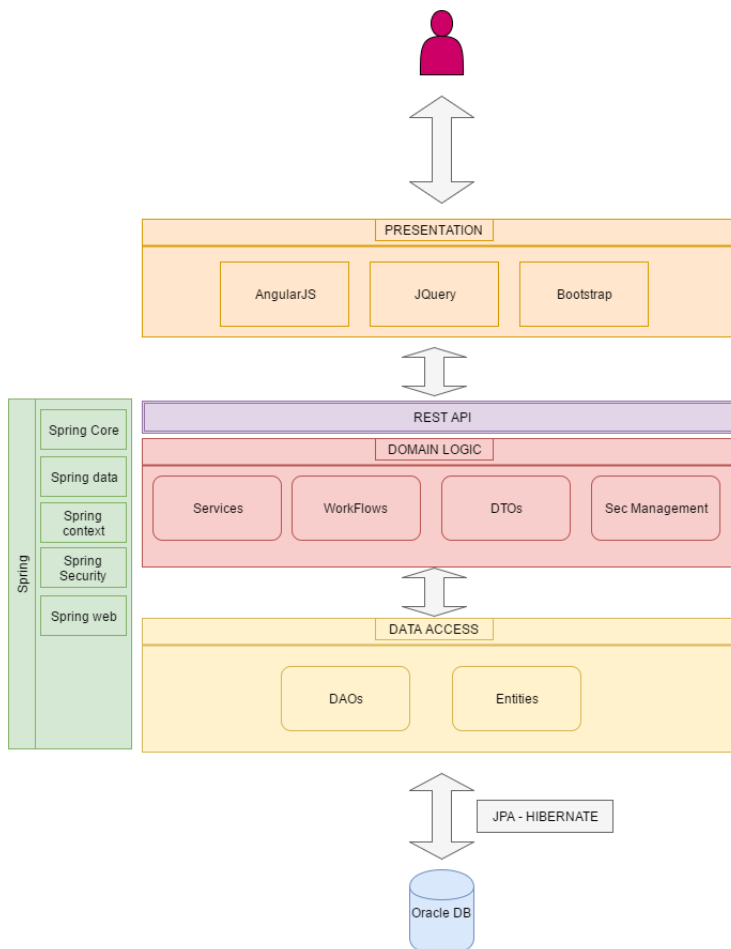


Ilustración 3 Arquitectura detallada de la aplicación

La arquitectura propuesta está compuesta por 3 capas, que son las siguientes, de abajo a arriba:

- **Capa de acceso a datos.** Será la encargada de actualizar y leer datos de nuestra base de datos, que será de tipo Oracle. Usa mayormente el módulo Spring-data-jpa de Spring, que se basa en la utilización de jpa con hibernate, y tiene dos tipos de objetos principalmente:
 - **Entities.** Entidades que mapearán las distintas tablas de nuestra base de datos, así como las relaciones entre ellas.
 - **DAOs (Data Acces Objects).** Clases que contendrán las distintas queries de consulta, creación, modificación y eliminación de datos en nuestra base de datos, usando las entidades anteriores.

- **Capa de lógica de negocio.** Esta capa se encarga de gestionar la lógica de negocio del sistema. Para ello, hace uso de la capa anterior, añadiendo la lógica necesaria en los servicios que la usan, y gestionando todos los *workflows* o flujos de trabajo que existan, así como la seguridad. Además, provee de una **API REST**. Usará principalmente Spring Context y Spring web.

- **Capa de presentación.** Es la capa que se encargara de renderizar en el navegador la aplicación web, la parte que interactúa con el usuario. Para ello, hace uso de la API REST que

proporciona la capa de la lógica de negocio. Usará frameworks tales como AngularJS, Bootstrap y JQuery.

Cabe indicarse que se ha utilizado maven para construir el proyecto y gestionar dependencias en la parte del back.

3. Diseño gráfico e interfaces

3.1 Estilos

En cuanto a la línea gráfica seguida en el desarrollo de la aplicación, se han de destacar:

- El logotipo, creado en base al nombre de la aplicación e incluyendo el símbolo que representa una fuente de tipo RSS o Atom. Se ha utilizado el color naranja para que destaque sobre el resto de la aplicación:



Ilustración 4 Logotipo de la aplicación

- En cuanto a la paleta de colores, se han elegido tres colores principales:
 - Negro para el texto y el menú
 - Gris como fondo
 - Dos tonalidades de azul, una para los títulos y otra que es la de por defecto de bootstrap para los elementos 'primary'.

Nótese que los estilos generales de la aplicación, a nivel visual hablando, son los de Bootstrap3 con distintas modificaciones realizadas.

3.2 Usabilidad/UX

En cuanto a usabilidad se refiere, se ha de destacar que se han usado:

- Una barra superior, en la que solamente se identifique la aplicación mediante su logo (que al clickarse, devuelve a la página principal) y, en caso de encontrarse un usuario autenticado, su nombre de usuario y la opción de acceder mediante un menú *dropdown* a las distintas opciones para visualizar/editar el perfil, así como cerrar sesión.



Ilustración 5 Barra superior de la aplicación

- Una barra lateral, a la izquierda, en la que se listan las posibles acciones a realizar. En el caso de dispositivos móviles, esta barra desaparece y es accesible desde un botón en el menú superior.

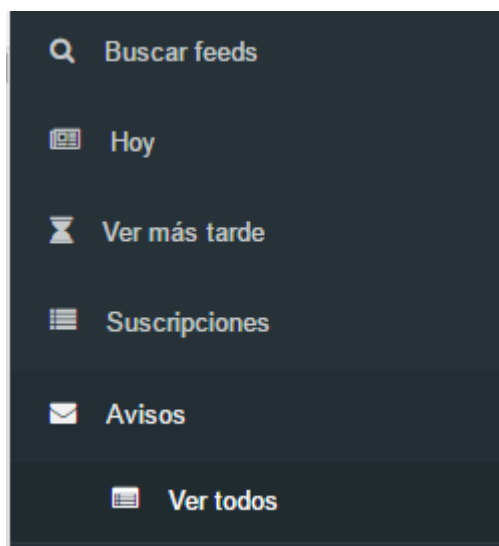


Ilustración 6 Menú lateral de la aplicación



Ilustración 7 Menú móvil de la aplicación

- Un panel de contenido, en el que se muestre en la esquina superior izquierda el nombre de la pantalla en la que se encuentra actualmente, así como el resto del contenido de la página seleccionada en el menu.

Cabe indicarse que debido a la simplicidad de la navegación, no se ha incluido mapa del sitio web y que en la versión móvil en determinadas ocasiones se prescinde de algunos elementos, como imágenes al listar noticias debido al poco espacio disponible.

4. Diseño de la Base de datos

El diseño de la base de datos es el siguiente, que ilustra cómo será a nivel físico las tablas de nuestra base de datos, partiendo de los requisitos de información definidos:

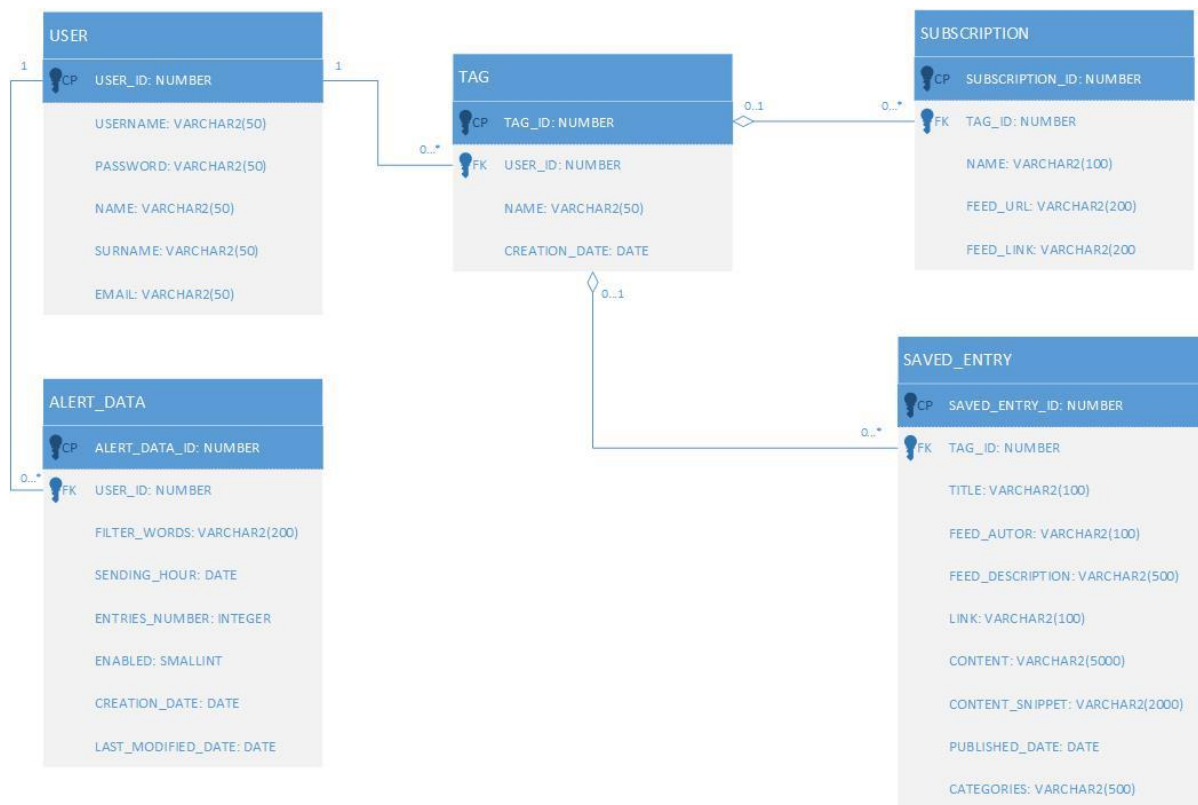


Ilustración 8 Diagrama de base de datos

Esta base de datos se ha desarrollado en Oracle SQL. El script completo para crearla se adjunta a este TFM como archivo separado.

4. Lenguajes de programación y APIs utilizadas

Anteriormente, en el artículo Diseño de esta memoria, se ha descrito la arquitectura propuesta para la aplicación. En este apartado se describirá con más detalle cómo se ha implementado la aplicación siguiendo esta arquitectura.

En cuanto al front-end se refiere, se ha utilizado el framework AngularJS 2.0, cuya *release* fue lanzada recientemente, usando en los distintos ficheros TypeScript, una especie de Javascript de tipado fuerte a diferencia de javascript, que es de tipado débil. Para levantarlo y compilarlo se ha usado npm junto a webpack. Además, se ha usado el IDE IntelliJ Webstorm para el desarrollo del proyecto.

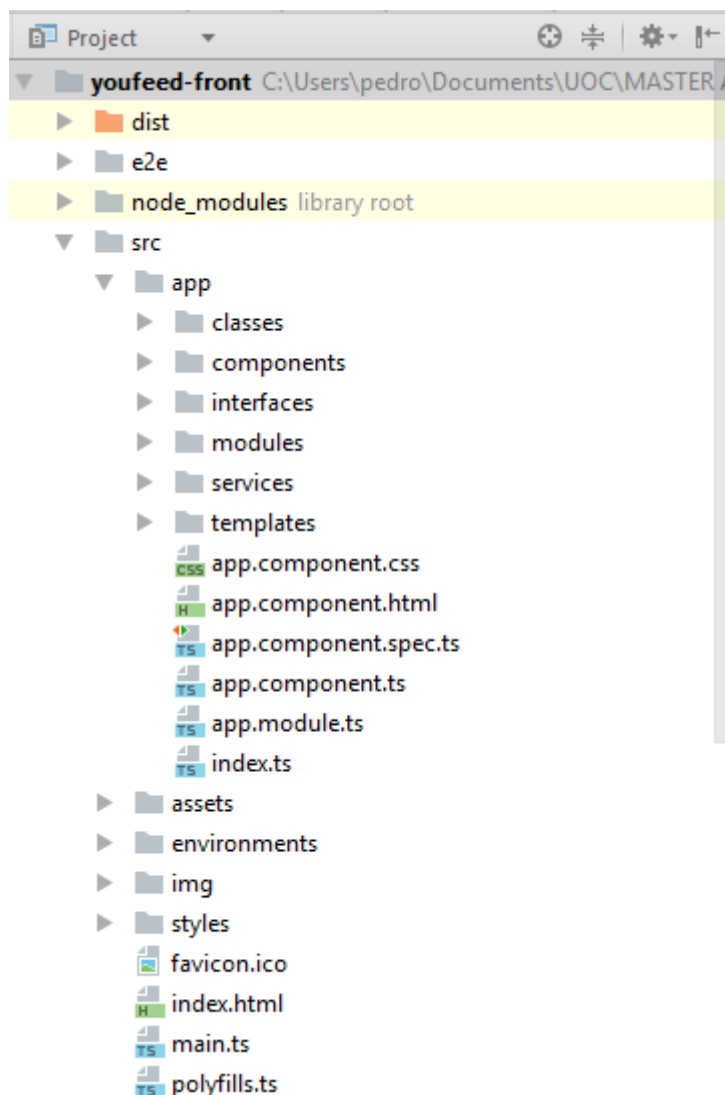


Ilustración 9 Estructura del front-end de la aplicación

Comentar que tenemos varios elementos dentro de nuestra aplicación con Angular, entre ellos:

- Servicios. Usados para mantener el estado general de la aplicación, por ejemplo ofrecer servicios de autenticación de usuarios. Otros servicios pueden ser encargados de hacer llamadas al back-end.

- Componentes. “Partes de nuestra aplicación”. Por ejemplo, se han tomado como componentes independientes el menú superior, el menú de la izquierda, y cada uno de los elementos que vayan mostrándose en el panel central.
- Templates. Plantillas en las que se basan los componentes para generar el HTML resultante. Normalmente contienen bindings, es decir, referencias a las distintas variables de los componentes asociados.

En lo que al back respecta, éste corre en un servidor Tomcat8, elegido en contraposición a JBoss debido a su más sencilla configuración respecto a este, y ha sido desarrollado usando Java 8 junto a distintos módulos del framework Spring, dado que el ejecutor de este proyecto es desarrollador back-end y trabaja a diario con estas tecnologías. Incidir en que para levantar el proyecto se ha usado Spring Boot, que nos permite desarrollar de manera rápida y eficaz proyectos Spring con una mínima configuración. Para gestionar las dependencias del proyecto y construirlo se ha usado Maven.

El fichero POM (Pom Object Model) describe nuestro proyecto y las dependencias que necesita, siendo inicialmente el de la aplicación desarrollada el siguiente:

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
  </parent>

  <groupId>com.uoc</groupId>
  <artifactId>youfeed</artifactId>
  <version>1.0.0-SNAPSHOT</version>

  <dependencies>
    <!--Spring web-->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!--Spring data-->
    <dependency>
      <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<!--Spring security-->
<dependency>
  <groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>
</dependency>

<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>10.2.0.3.0</version>
</dependency>

</dependencies>

</project>
```

Como puede verse, nuestro proyecto se llama youfeed, y usa spring boot con sus módulos spring-boot-starter-data-jpa para acceder a la base de datos y spring-boot-starter-security para securizar la aplicación. Además, también es necesario disponer del driver de ojdbc para poder trabajar con la base de datos Oracle.

A continuación, se presenta una captura del proyecto obtenida del IDE usado, IntelliJ IDEA:

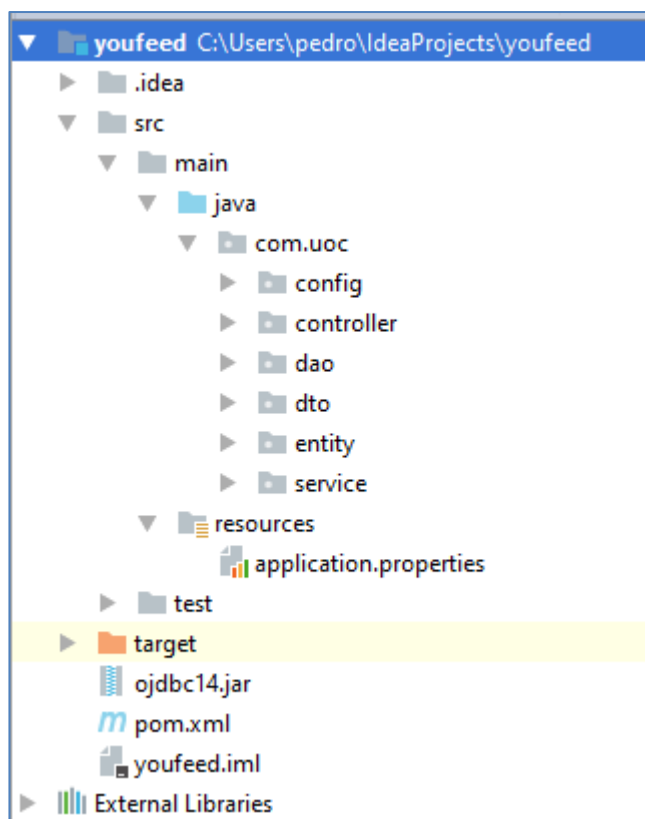


Ilustración 10 Estructura del back-end de la aplicación

Como puede observarse, el código en la aplicación se ha separado en distintos paquetes, según el funcionamiento de las clases que contengan. A continuación, listaremos los mismos con un ejemplo para cada uno:

- **Config:** Contiene las clases de configuración del proyecto.


```

package com.uoc.config;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;

/**
 * Created by pedro on 12/12/2016.
 */
@SpringBootApplication
@EnableWebMvc
@ComponentScan("com.uoc")
@EntityScan(basePackages = "com.uoc.entity")
@EnableJpaRepositories("com.uoc.dao")
public class MyFeedConfiguration {

    Logger logger = LoggerFactory.getLogger(this.getClass());

    public static void main(String[] args) throws Exception {
        SpringApplication.run(MyFeedConfiguration.class, args);
    }

}

```

Ilustración 11 Clase de configuración de la aplicación

La clase `MyFeedConfiguration`, es la “clase de entrada” de la aplicación, la primera que se ejecuta, y como puede verse, mediante etiquetas definimos qué paquetes serán escaneados por spring (`com.uoc` en este caso), dónde estarán nuestras entidades y dónde los repositorios o daos.

- **Controller:** Contiene las clases que ofrecen la API tipo REST de nuestro proyecto. Todas ellas están anotadas con `@RestController` de spring y contienen métodos de entrada que llamarán a los servicios correspondientes.

```

@RestController
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;

    @RequestMapping(value = "/login", method = RequestMethod.POST, consumes = "application/json")
    public ResponseEntity createUser(@RequestBody LoginDto loginRequest) {
        return getUserService().loginUser(loginRequest);
    }

    @RequestMapping(value = "/create", method = RequestMethod.POST, consumes = "application/json")
    public ResponseEntity createUser(@RequestBody UserDto userRequest) {
        return getUserService().createUser(userRequest);
    }

    @RequestMapping(value = "/update", method = RequestMethod.PUT, consumes = "application/json")
    public ResponseEntity updateUser(@RequestBody UserModel userRequest) {
        return getUserService().updateUser(userRequest);
    }
}

```

Ilustración 12 Ejemplo de controlador REST

En la imagen precedente se muestra parte de la clase UserController, que ofrece métodos para gestionar los usuarios en la aplicación. Por ejemplo, el método /user/login que recibe una petición POST con un json que contiene un username y una password.

- **Dao:** Son los Data Acces Object o repositorios que usamos para realizar consultas en base de datos. Este paquete contiene interfaces que extienden de JpaRepository<G,T> de spring data, ofreciéndonos los métodos CRUD (CREATE, RETREIVE, UPDATED y DELETE) para cada tabla a la que hagan referencia y además definir métodos propios de manera sintáctica sin tener que escribir consultas. En caso de ser necesario realizar alguna consulta “complicada”, se usará en estos daos el lenguaje hql para realizarlas.

```

@Repository
public interface UserDao extends JpaRepository<UserModel, Integer> {

    UserModel findByUsername(String username);

    Collection<UserModel> findByUsernameOrEmail(String username, String email);
}

```

Ilustración 13 Ejemplo de DAO

En el caso de la interfaz UserDao, tenemos al momento de desarrollar esta documentación dos métodos definidos de manera sintáctica, uno que busca usuarios dado un username y otro que los busca dados un username y un email.

- Dto: Data Transfer Objects. Son objetos que funcionan como simple portadores de información, ya sea dentro de la aplicación como fuera de ella.
- Entity: Clases que mapean las distintas tablas de base de datos.

```

@Entity
@Table(name = "MYFEED_USER")
public class UserModel implements Serializable {

    private static final long serialVersionUID = -5197614923009792351L;

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "USER_SEQ")
    @SequenceGenerator(name = "USER_SEQ", sequenceName = "MYFEED_USER_SEQ")
    @Column(name = "USER_ID")
    private Integer id;

    @Column(name = "USERNAME")
    private String username;

    @Column(name = "USER_PASSWORD")
    private String password;

    @Column(name = "NAME")
    private String name;

    @Column(name = "SURNAME")
    private String surname;

    @Column(name = "EMAIL")
    private String email;

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "user")
    private Collection<TagModel> tags;

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "user")

```

Ilustración 14 Ejemplo de entity

Como podemos ver en el caso de la clase UserModel, esta define los atributos que tiene un usuario en base de datos y define el mapeo de los mismos con la tabla MYFEED_USER.

- Service: Clases que ofrecen funcionalidad a los controladores o a otros servicios y hacen uso de los daos.

```

@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserDao userDao;

    @Override
    public ResponseEntity loginUser(LoginDto loginDto) {
        ResponseEntity res;
        UserModel userModel = userDao.findByUsername(loginDto.getUsername());
        if(userModel == null){
            res = new ResponseEntity( body: "No se encontró el usuario", HttpStatus.INTERNAL_SERVER_ERROR);
        }else{
            res = new ResponseEntity(userModel, HttpStatus.OK);
        }
        return res;
    }
}

```

Ilustración 15 Ejemplo de servicio

Finalmente, cabe mencionar, que en un principio se iba a usar RSS Google Feed API como motor de búsqueda. En un principio, en el prototipo de la aplicación, era gestionado desde el front-end usando una librería javascript aportada por google. Dada la incompatibilidad de esta librería con Angular y la necesidad de acceder a los feeds desde el back-end para realizar envíos de alerta, finalmente fue desplazada al back-end de la aplicación y se habilitaron métodos para acceder a las funcionalidades desde la API REST propia cuando sea necesario. Posteriormente, la API se deprecó, por lo que se optó por cambiar de proveedor a yahoo, usando yahoo YQL.

Capítulo 4: Implementación

1. Requisitos de instalación

Para poder ejecutar correctamente el proyecto, es necesario en primer lugar un PC con acceso a Internet, ya sea Windows o Linux, aunque a continuación se describirá el proceso para Windows.

Además, este PC deberá tener instalado el SDK de Java 8, tal como se indica en el siguiente enlace: https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html.

Asimismo, deberá descargarse y descomprimirse el servidor de aplicaciones Tomcat 8, disponible en <https://tomcat.apache.org/download-80.cgi>.

2. Instrucciones de instalación

En primer lugar ha de abrirse el fichero war adjunto a este TFM, con un programa como winrar, y editar el fichero \WEB-INF\classes\application.properties en su interior para establecer un servidor de correos. Durante el desarrollo se ha usado Gmail. Un ejemplo es el siguiente:

```
#EMAIL
spring.mail.host=smtp.gmail.com
spring.mail.username=pedrouceda92@gmail.com
spring.mail.password=
spring.mail.properties.mail.smtp.auth = true
spring.mail.properties.mail.smtp.socketFactory.port = 465
spring.mail.properties.mail.smtp.socketFactory.class = javax.net.ssl.SSLSocketFactory
spring.mail.properties.mail.smtp.socketFactory.fallback = false
spring.mail.properties.mail.smtp.ssl.enable = true
```

Ilustración 16 Configuración de correo en application.properties

El fichero que se entrega tiene los mismos datos de esta captura con la contraseña (spring.mail.password) vacía.

En primer lugar, ha de copiarse el fichero war al directorio donde se descomprimió el Tomcat 8 descargado, concretamente a {dir.tomcat.8}/webapps .

Luego de haberse copiado el war, ha de ejecutarse el fichero {dir.tomcat.8}/bin/startup.bat

The screenshot shows a Windows Explorer window with the following path: > Este equipo > Disco local (C:) > apache-tomcat-8.5.9 > bin. The main pane displays a list of files and folders with columns for Name, Date modified, Type, and Size. The 'startup.bat' file is highlighted in grey.

Nombre	Fecha de modifica...	Tipo	Tamaño
bootstrap.jar	04/01/2017 18:37	Executable Jar File	32 KB
catalina.bat	04/01/2017 18:37	Archivo por lotes ...	15 KB
catalina.sh	04/01/2017 18:37	Shell Script	22 KB
catalina-tasks.xml	04/01/2017 18:37	Archivo XML	2 KB
commons-daemon.jar	04/01/2017 18:37	Executable Jar File	24 KB
commons-daemon-native.tar.gz	04/01/2017 18:37	Archivo GZ	201 KB
configtest.bat	04/01/2017 18:37	Archivo por lotes ...	2 KB
configtest.sh	04/01/2017 18:37	Shell Script	2 KB
daemon.sh	04/01/2017 18:37	Shell Script	8 KB
digest.bat	04/01/2017 18:37	Archivo por lotes ...	3 KB
digest.sh	04/01/2017 18:37	Shell Script	2 KB
setclasspath.bat	04/01/2017 18:37	Archivo por lotes ...	4 KB
setclasspath.sh	04/01/2017 18:37	Shell Script	4 KB
shutdown.bat	04/01/2017 18:37	Archivo por lotes ...	2 KB
shutdown.sh	04/01/2017 18:37	Shell Script	2 KB
startup.bat	04/01/2017 18:37	Archivo por lotes ...	2 KB
startup.sh	04/01/2017 18:37	Shell Script	2 KB
tomcat-juli.jar	04/01/2017 18:37	Executable Jar File	41 KB
tomcat-native.tar.gz	04/01/2017 18:37	Archivo GZ	383 KB
tool-wrapper.bat	04/01/2017 18:37	Archivo por lotes ...	4 KB
tool-wrapper.sh	04/01/2017 18:37	Shell Script	5 KB
version.bat	04/01/2017 18:37	Archivo por lotes ...	2 KB
version.sh	04/01/2017 18:37	Shell Script	2 KB

Ilustración 17 Fichero startup.bat

Acto seguido se abrirá un terminal de Windows en el que se indicará que el servidor inició “Server startup in xxxx ms” tras haberse desplegado el war suministrado.

```

Tomcat
09-Jan-2017 21:20:43.998 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor: Oracle Corporation
09-Jan-2017 21:20:43.999 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE: C:\apache-tomcat-8.5.9
09-Jan-2017 21:20:43.999 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME: C:\apache-tomcat-8.5.9
09-Jan-2017 21:20:44.000 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command Line argument: -Djava.util.logging.config.file=C:\a
pache-tomcat-8.5.9\conf\logging.properties
09-Jan-2017 21:20:44.000 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.util.logging.manager=org.apac
he.juli.ClassLoaderLogManager
09-Jan-2017 21:20:44.001 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djdk.tls.ephemeralDHKeySize=2048
09-Jan-2017 21:20:44.001 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.protocol.handler.pkgs=org.apa
che.catalina.webresources
09-Jan-2017 21:20:44.002 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.base=C:\apache-tomcat-8.5
.9
09-Jan-2017 21:20:44.003 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.home=C:\apache-tomcat-8.5
.9
09-Jan-2017 21:20:44.004 INFORMACIEN [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.io.tmpdir=C:\apache-tomcat-8
.5.9\temp
09-Jan-2017 21:20:44.005 INFORMACIEN [main] org.apache.catalina.core.AppLifecycleListener.lifecycleEvent La biblioteca nativa de Apache Tomcat basada en ARP
que permite un rendimiento optimo en entornos de desarrollo no ha sido hallada en java.library.path: C:\Program Files\Java\jdk1.8.0_112\bin;C:\Windows\Sun\Ja
va\bin;C:\Windows\system32;C:\Windows;C:\ProgramData\Oracle\Java\javapath;C:\oraclexe\App\oracle\product\11.2.0\server\bin;C:\Windows\system32;C:\Windows;C:
\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Skype\Phone\;C:\Program Files\Node.js\;C:\Program Files\Git\cmd;C:\P
rogram Files\Java\jdk1.8.0_112\bin;C:\apache-maven-3.3.9\bin;C:\Program Files\MySQL\MySQL Utilities 1.6\;C:\Users\pedro\AppData\Roaming\npm;.
09-Jan-2017 21:20:44.247 INFORMACIEN [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["http-nio-8080"]
09-Jan-2017 21:20:44.517 INFORMACIEN [main] org.apache.tomcat.util.net.NioSelectorPool.getSharedSelector Using a shared selector for servlet write/read
09-Jan-2017 21:20:44.522 INFORMACIEN [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["ajp-nio-8009"]
09-Jan-2017 21:20:44.527 INFORMACIEN [main] org.apache.tomcat.util.net.NioSelectorPool.getSharedSelector Using a shared selector for servlet write/read
09-Jan-2017 21:20:44.532 INFORMACIEN [main] org.apache.catalina.startup.Catalina.load Initialization processed in 1163 ms
09-Jan-2017 21:20:44.596 INFORMACIEN [main] org.apache.catalina.core.StandardService.startInternal Arrancando servicio Catalina
09-Jan-2017 21:20:44.603 INFORMACIEN [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/8.5.9
09-Jan-2017 21:20:44.672 INFORMACIEN [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Despliegue del archivo C:\apache-tomcat-8.5.9\w
ebapps\youfeed-1.0.0-SNAPSHOT.war de
la aplicaci3n web
09-Jan-2017 21:20:59.133 INFORMACIEN [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars Al menos un JAR, que se ha explorado buscando TLDs
, a n no contenYa TLDs. Activar historial de depuraci3n para este historial para una completa lista de los JARs que fueron explorados y de los que nos se h
all3n TLDs. Saltarse JARs no necesarios durante la exploraci3n puede dar lugar a una mejora de tiempo significativa en el arranque y compilaci3n de JSP .

=====
:: Spring Boot :: (v1.4.2.RELEASE)

2017-01-09 21:21:03.213 INFO 9000 --- [ost-startStop-1] com.uoc.config.MyFeedConfiguraton : Starting MyFeedConfiguraton v1.0.0-SNAPSHOT on DESKTOP-S
81B388 with PID 9000 (C:\apache-tomcat-8.5.9\webapps\youfeed-1.0.0-SNAPSHOT\WEB-INF\classes started by pedro in C:\apache-tomcat-8.5.9\bin)
2017-01-09 21:21:03.226 INFO 9000 --- [ost-startStop-1] com.uoc.config.MyFeedConfiguraton : No active profile set, falling back to default profiles:
default
2017-01-09 21:21:03.380 INFO 9000 --- [ost-startStop-1] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.Anno
tationConfigEmbeddedWebApplicationContext@496cc9ba: startup date [Mon Jan 09 21:21:03 CET 2017]; root of context hierarchy
2017-01-09 21:21:12.690 INFO 9000 --- [ost-startStop-1] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTra
nsactionManagementConfiguraton' of type [class org.springframework.transaction.annotation.ProxyTransactionManagementConfiguraton$EnhancerBySpringGLIB$$a2
453a9c] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2017-01-09 21:21:12.924 INFO 9000 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 9
545 ms
2017-01-09 21:21:15.239 INFO 9000 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'errorPageFilter' to: [/*]
2017-01-09 21:21:15.240 INFO 9000 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/*]
2017-01-09 21:21:15.245 INFO 9000 --- [ost-startStop-1] .s.DelegatingFilterProxyRegistrationBean : Mapping filter: 'springSecurityFilterChain' to: [/*]
2017-01-09 21:21:15.246 INFO 9000 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Mapping servlet: 'dispatcherServlet' to [/]

```

Ilustraci3n 18 Log de la aplicaci3n en la consola de windows

Una vez encendido el servidor, podremos acceder a la aplicaci3n en <http://localhost:8080/youfeed/index.html>

Capítulo 5: Demostración

1. Instrucciones de uso

La aplicación desarrollada ha sido subida a la URL que se indica a continuación, de acceso público:

- <http://ec2-54-213-215-103.us-west-2.compute.amazonaws.com:8080/youfeed/>

En el apartado 4. Ejemplos de uso del producto de este capítulo se hace un *tour* por la aplicación, que también podrá realizarse desde la web en cuestión en vivo.

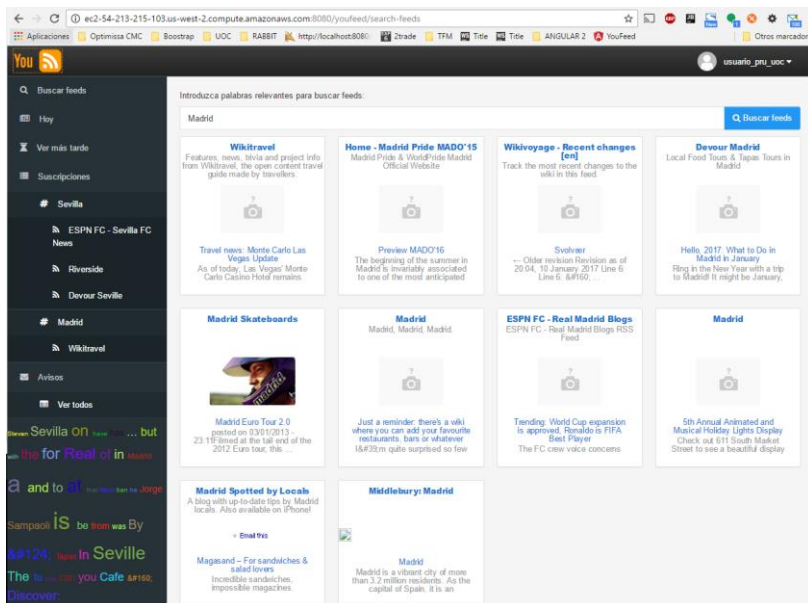


Ilustración 19 Vista de la aplicación en dispositivo de escritorio

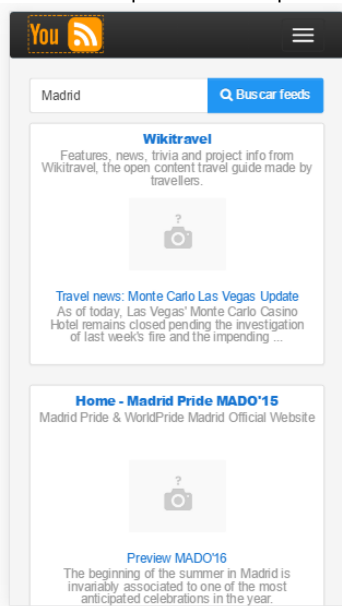


Ilustración 20 Vista de la aplicación en dispositivo móvil

2. Prototipos

Como parte del diseño de la aplicación, se desarrolló un prototipo de alta fidelidad, Hi-Fi por sus siglas en inglés, usando HTML5, CSS3 y jQuery, que se usó como base para el desarrollo del front-end del proyecto.

Con motivo de la PEC3, se subió el mismo a <http://eimtcms.uoc.edu/~pucedam/tfm/mockups/index.html>, donde puede ser consultado. Nótese que es sólo un prototipo, por lo que tiene poca funcionalidad. A continuación se describirán los mock-ups que se realizaron.

1. Página principal – Consideraciones generales

Como se puede observar en la siguiente ilustración, se presenta el layout general de la aplicación: Una barra de navegación superior en la que se presentan las opciones del usuario, y una lateral en la que se presentan todos los elementos disponibles en la aplicación y se sitúa al usuario.

Nótese que se prescinde de breadcrumbs, usándose colores en esta barra vertical para indicar al usuario en qué parte de la aplicación se encuentra y no se desorienta.

La ilustración siguiente muestra la página de inicio, en el caso de no haberse iniciado sesión (en la esquina superior izquierda debería aparecer un “iniciar sesión” también).

Así, se presentaría la aplicación y se invitaría al usuario a registrarse o iniciar sesión en la misma.

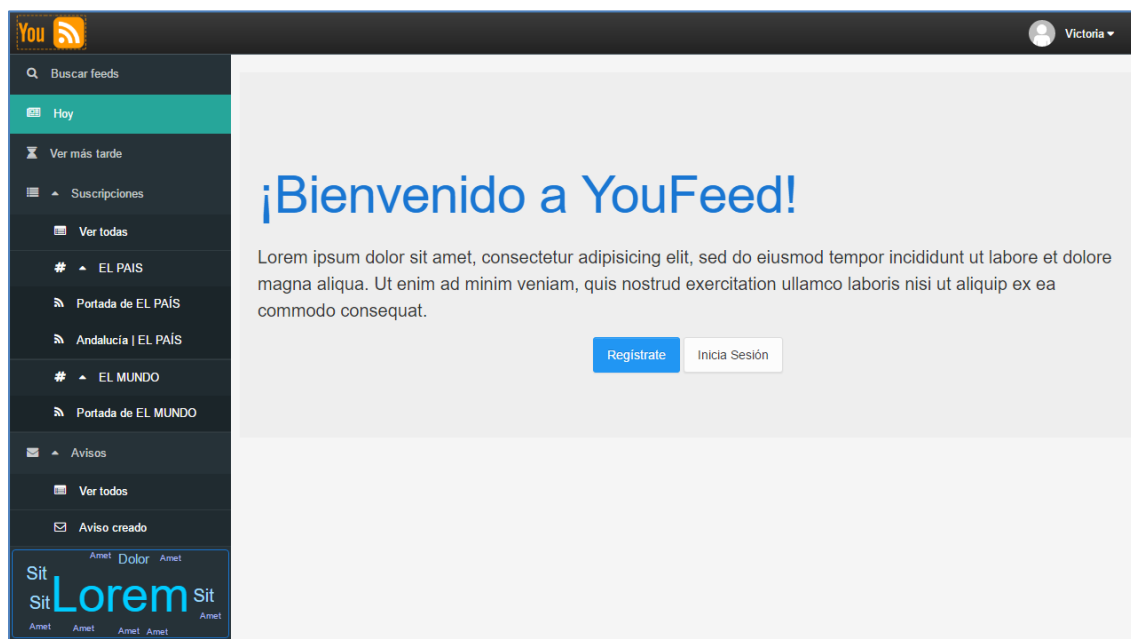


Ilustración 21 Mockup - página principal

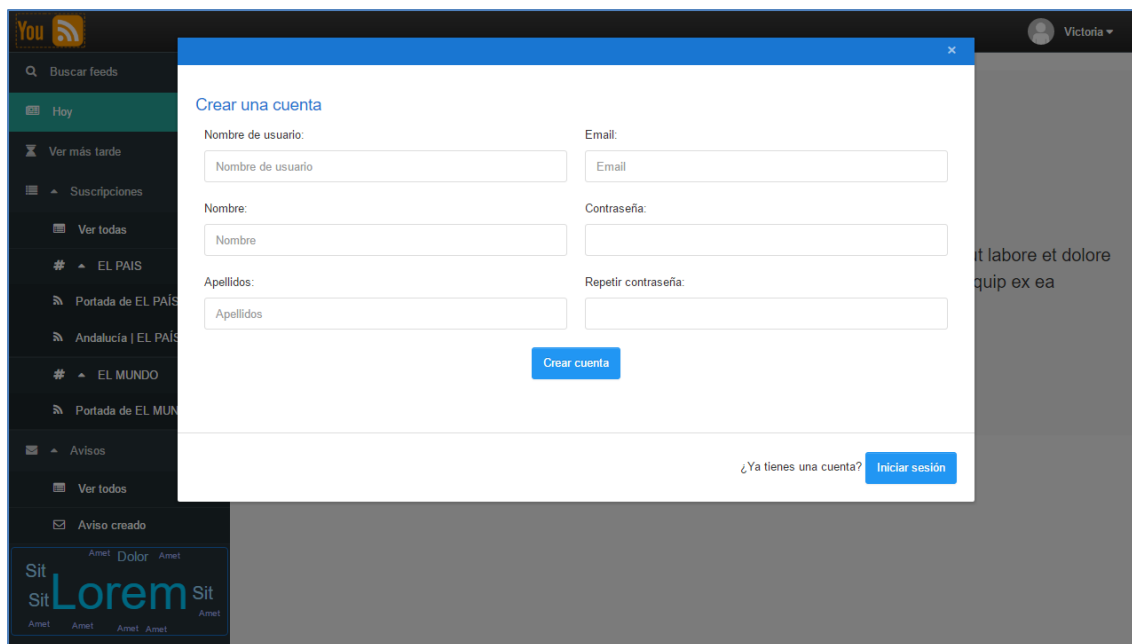


Ilustración 22 Mockup - crear una cuenta

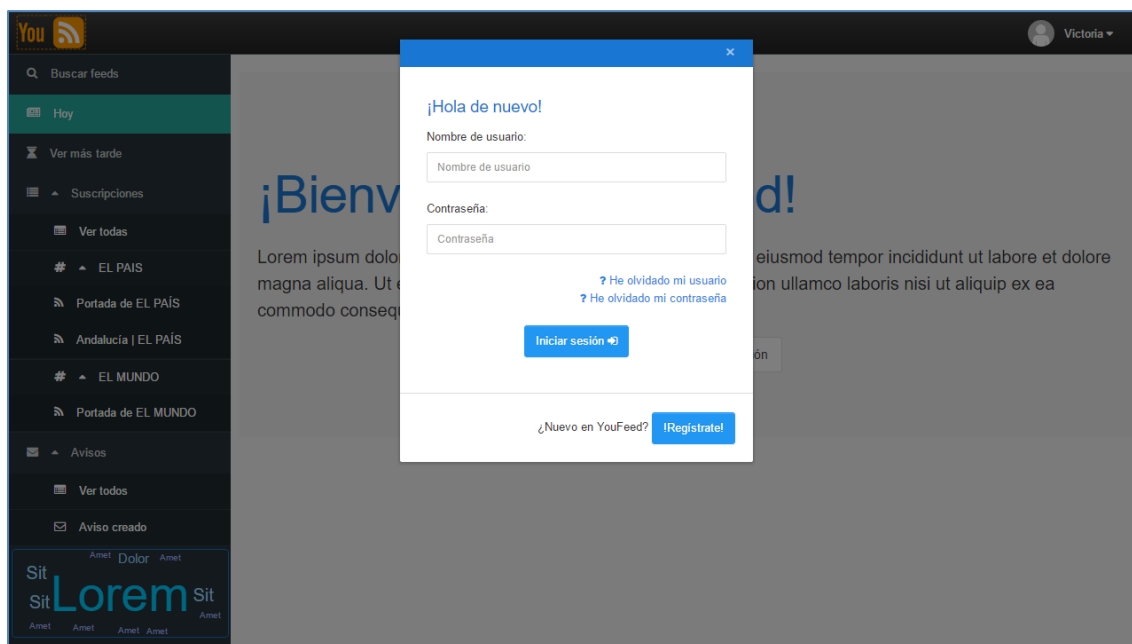


Ilustración 23 Mockup - Iniciar sesión

Una vez se haya iniciado sesión, se podrá acceder a “mi perfil”, para visualizar los datos del usuario y editarlos si se necesita, además de “ajustes”, donde se permite cambiar la contraseña. Además, nótese que en la página principal del mockup se encuentra la nube de palabras generada en la esquina inferior izquierda, en este caso con texto autogenerado del tipo “Lorem ipsum”.

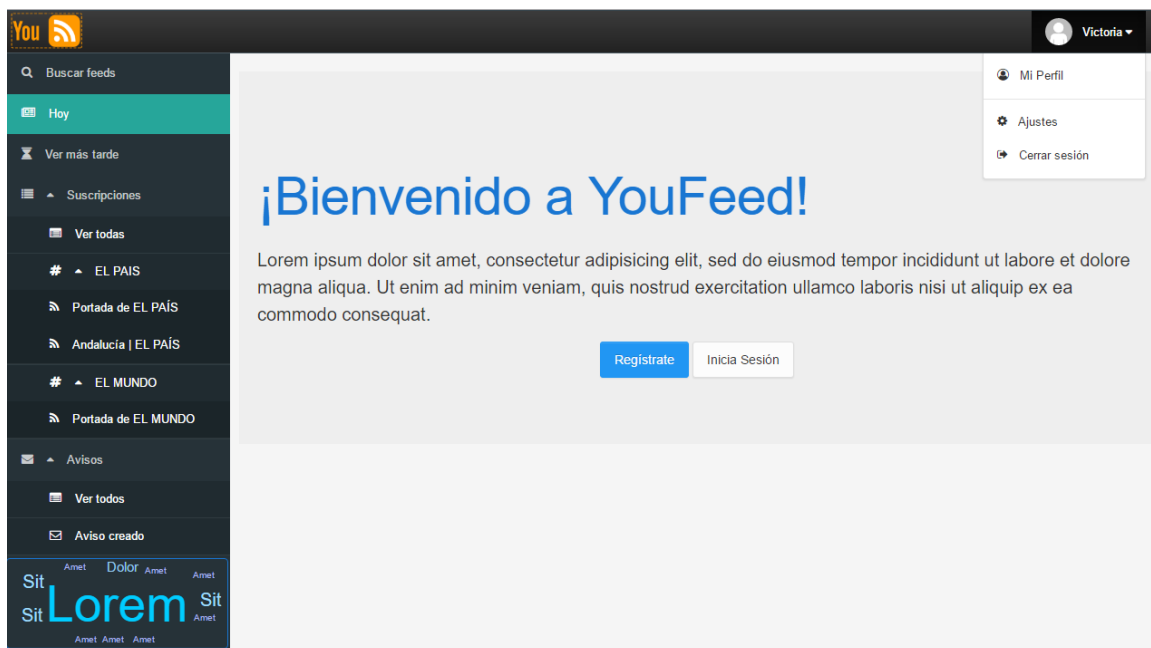


Ilustración 24 Mockup - Página principal autenticado

2. Mi perfil

El usuario, pulsando sobre “Victoria” en la imagen y “Mi Perfil”, puede ver cuáles son sus datos de usuario y cambiarlos si desea, excepto el nombre de usuario. Además, se le da la opción explícitamente a acceder a “Ajustes”.

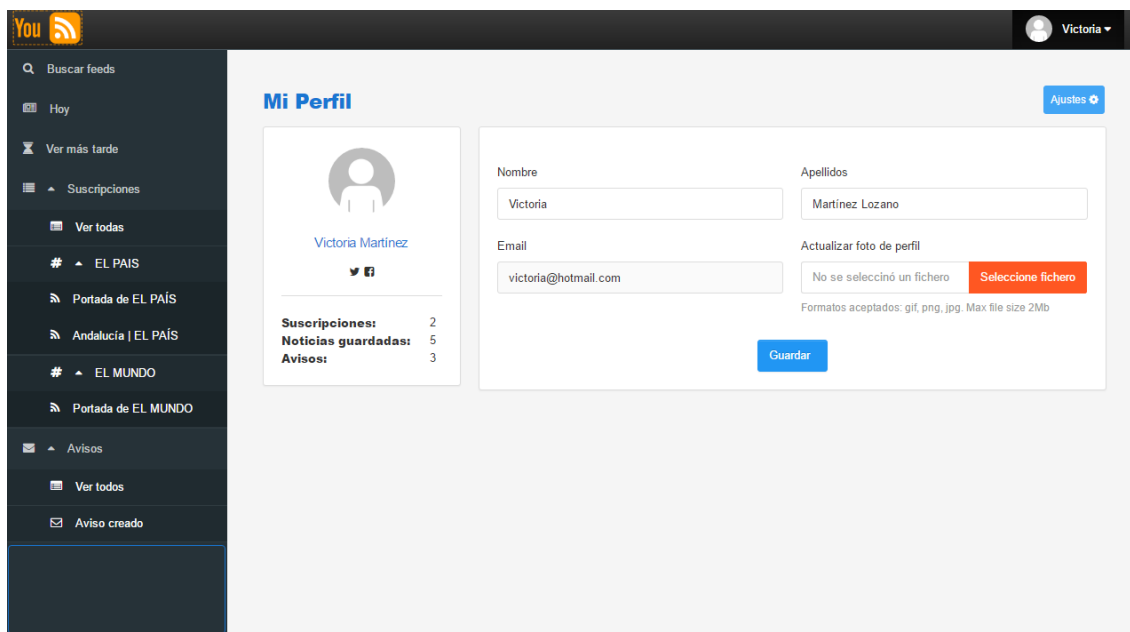


Ilustración 25 Mockup - Mi perfil

3. Ajustes

En la ventana de ajustes, al usuarios e le permite cambiar tanto su correo (de nuevo) como su contraseña.

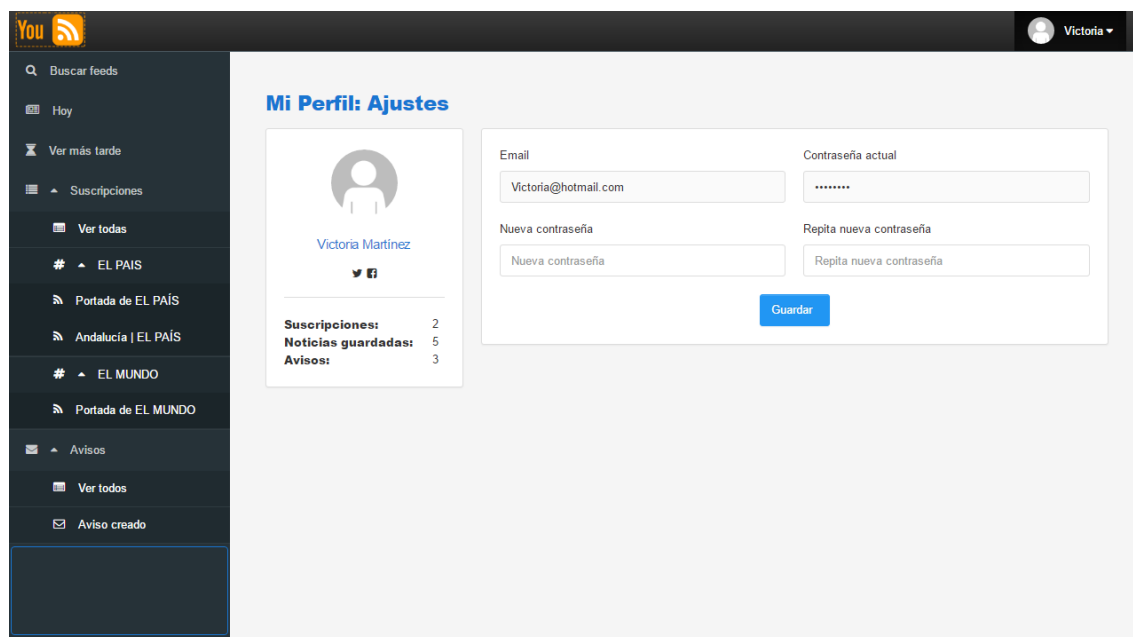


Ilustración 26 Mockup - Mi perfil: Ajustes

4. Buscar feeds

En la siguiente pantalla se permite al usuario buscar feeds por sitio o contenido. Tras pulsar el botón “buscar feeds”, el sistema realiza la búsqueda y devuelve una lista con los mismos, a los que el usuario podrá suscribirse, ver o abrir en otra ventana.

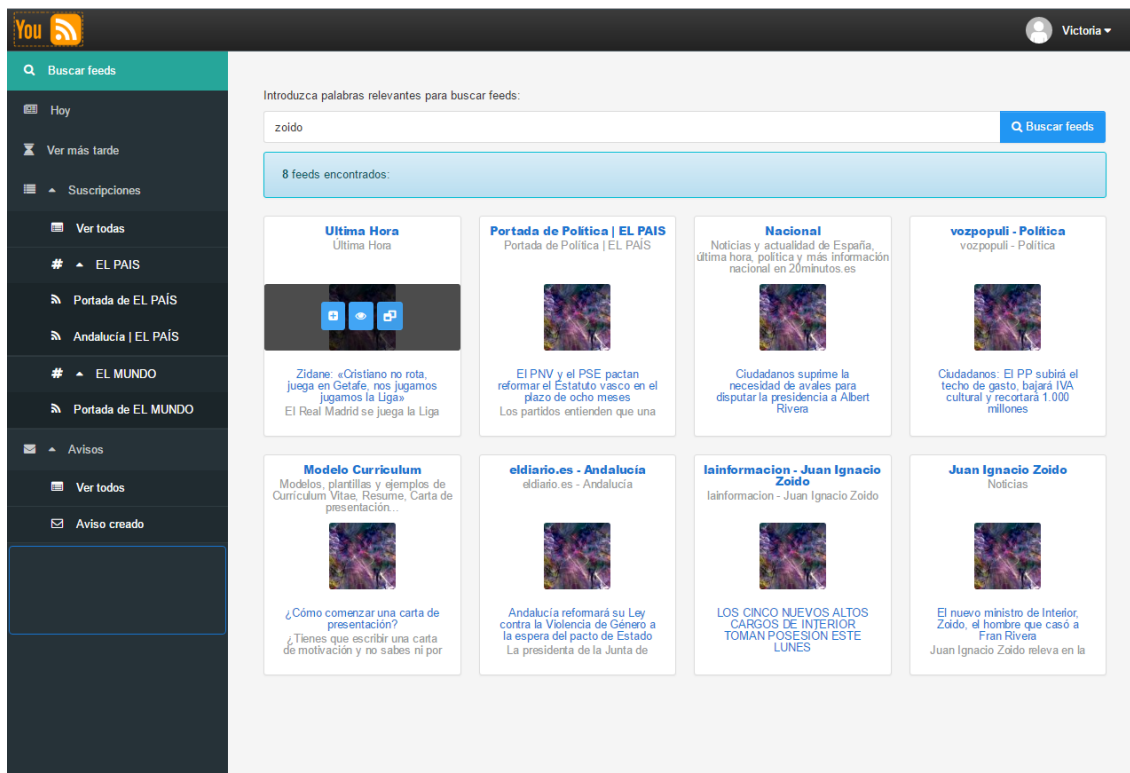


Ilustración 27 Mockup - Buscar feeds

5. Ver más tarde

En este caso se muestra un ejemplo de la vista de noticias guardadas para ver más tarde. Como puede verse, se da la opción de filtrar por número de entradas que se quieren ver, fecha (Hoy, Ayer, Anterior) feed y palabras que se quieran introducir.

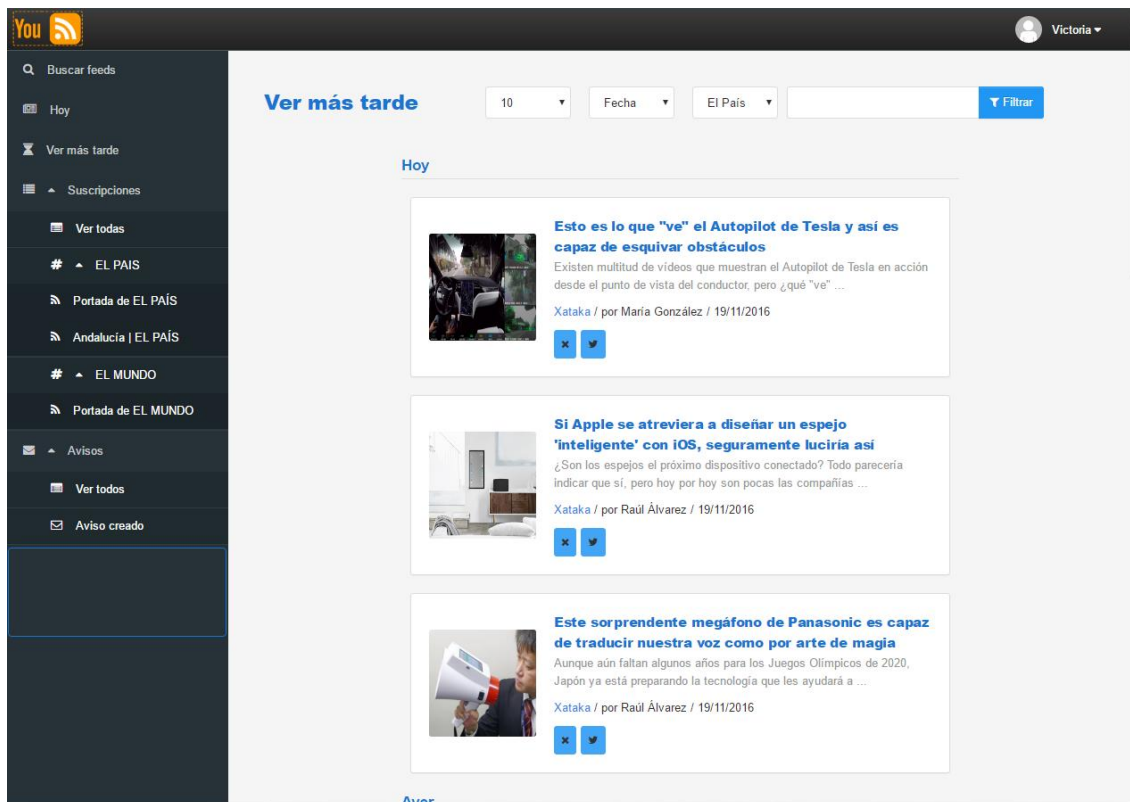


Ilustración 28 Mockup - Ver más tarde

6. Todas mis suscripciones

En esta página, que será parecida a la que haya de mostrarse cuando un usuario con sesión acceda a la aplicación, se listarán todas las noticias disponibles para todos los feeds, por lo que se indican las suscripciones, el titular y la hora de las noticias o entradas.

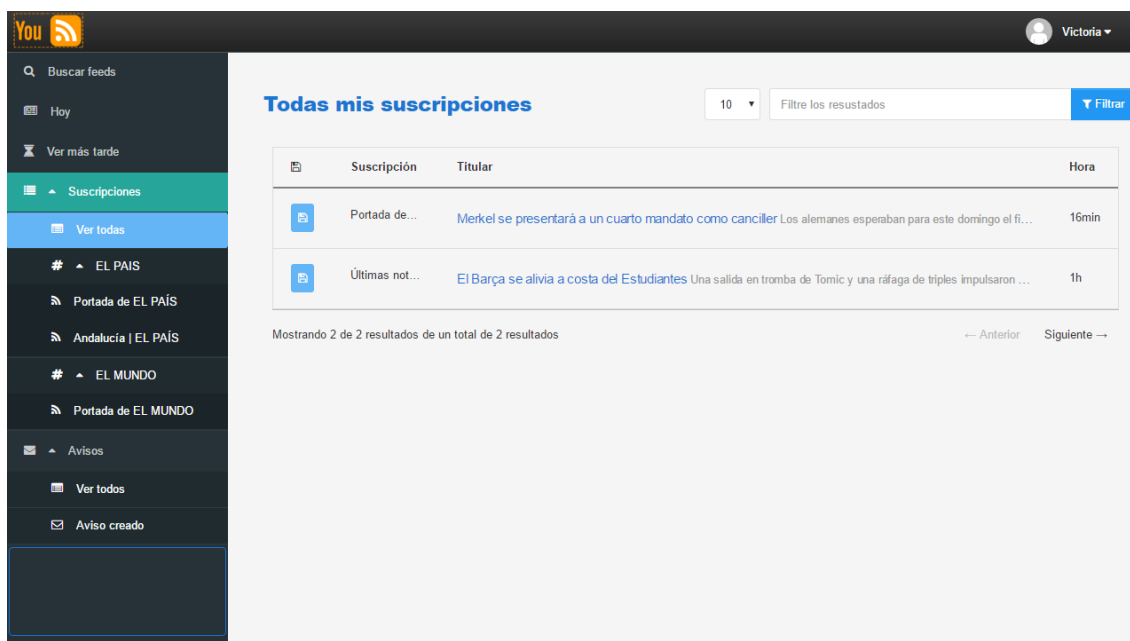


Ilustración 29 Mockup - Suscripciones

7. Entradas de un feed

Al pulsar sobre un feed, se mostrarán las entradas del mismo. Asimismo, se dará la opción a suscribirse (si es que no se ha suscrito) o modificar o eliminar la suscripción e caso contrario.

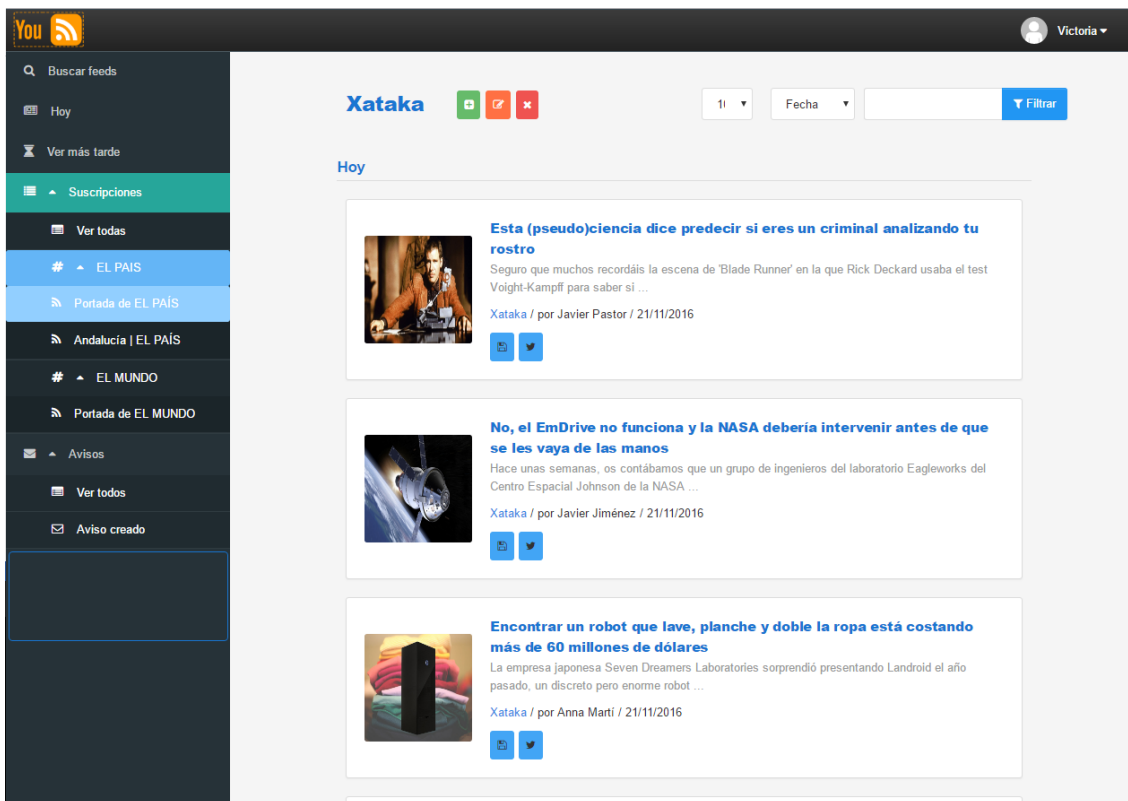


Ilustración 30 - Mockup – suscripción

Las noticias del mockup son reales, todas de xataka a modo de ejemplo. En caso de pulsar un titular, se abrirá un panel modal del siguiente tipo:



Ilustración 31 Mockup - Ver noticia en suscripción

8. Avisos

La pantalla de avisos de un usuario tendrá la siguiente estética:

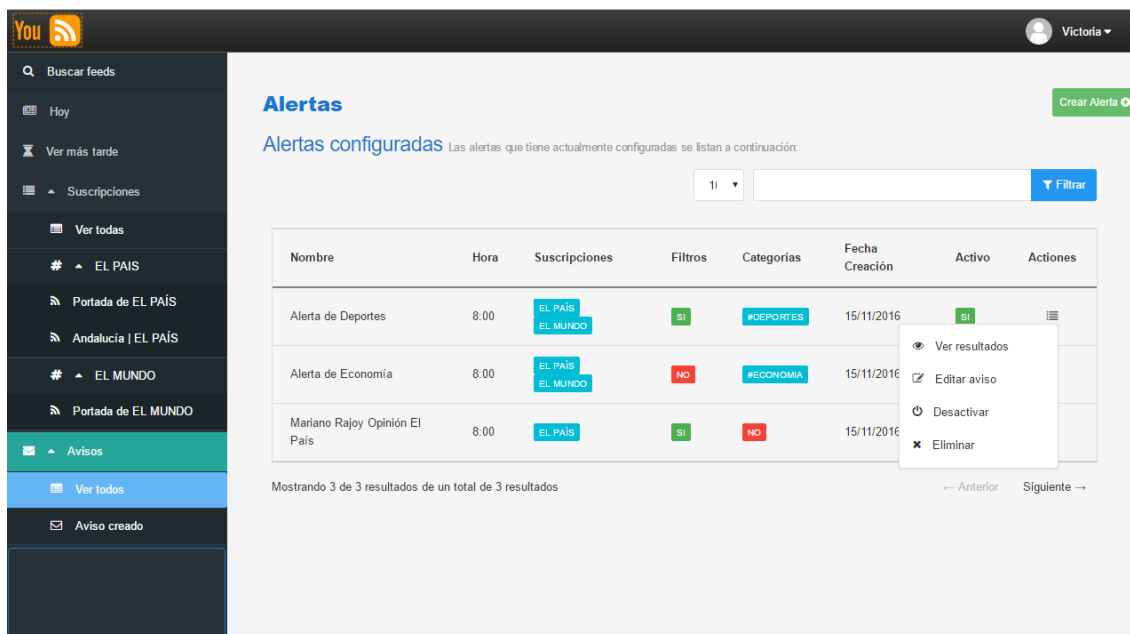


Ilustración 32 Mockup – Alertas (Opciones de alerta)

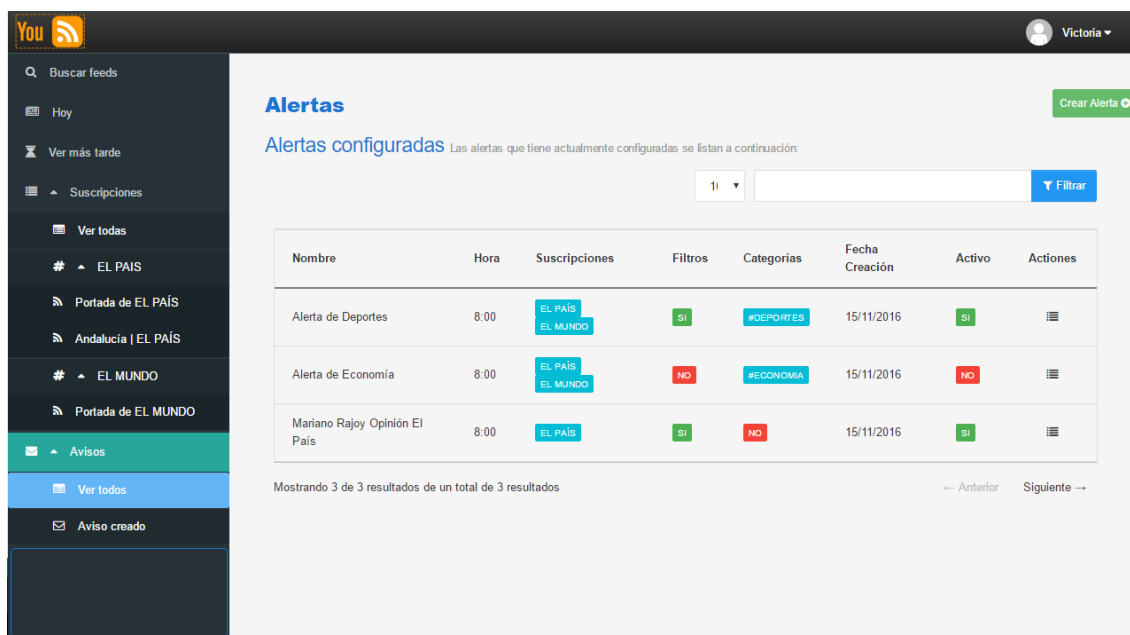


Ilustración 33 Mockup - Alertas

9. Crear/Editar aviso

The screenshot shows the 'Nuevo Aviso' (New Alert) form. On the left is a dark sidebar with navigation options: 'Buscar feeds', 'Hoy', 'Ver más tarde', 'Suscripciones', 'Ver todas', 'EL PAÍS', 'Portada de EL PAÍS', 'Andalucía | EL PAÍS', 'EL MUNDO', 'Portada de EL MUNDO', 'Avisos', 'Ver todos', and 'Aviso creado'. The main content area is titled 'Nuevo Aviso' and is divided into two columns: 'DATOS DE LA ALERTA' and 'FILTROS'. The 'DATOS DE LA ALERTA' column contains fields for 'Nombre de la alerta' (with placeholder 'Nombre de la alerta'), 'Email' (with 'pedro_2492@hotmail.com'), 'Hora' (with two empty input boxes), and 'Feeds' (with a dropdown menu showing 'El País, El Mundo'). The 'FILTROS' column contains a 'Categorías' dropdown set to 'Deportes', a 'Términos a buscar' text area with placeholder 'Introduzca términos por los que filtrar', and an 'Activo' toggle set to 'Si' with an 'Activar alerta' button. A 'Crear Alerta' button is located at the bottom right of the form.

Ilustración 34 Mockup - Crear aviso

The screenshot shows the 'Editar Aviso' (Edit Alert) form. The layout is identical to the 'Nuevo Aviso' form, but the 'Nombre de la alerta' field is pre-filled with 'Alerta de deportes' and the 'Hora' field is pre-filled with '8' and '15'. The 'Activar alerta' button is now disabled. A blue 'Editar Aviso' button is located at the bottom right of the form.

Ilustración 35 Mockup - Editar aviso

10. Versión mobile

La aplicación ha de ser responsive, por lo que el prototipo inicial también lo es, si bien queda hacer algunas mejoras. Téngase a modo de ejemplo la pantalla de inicio:

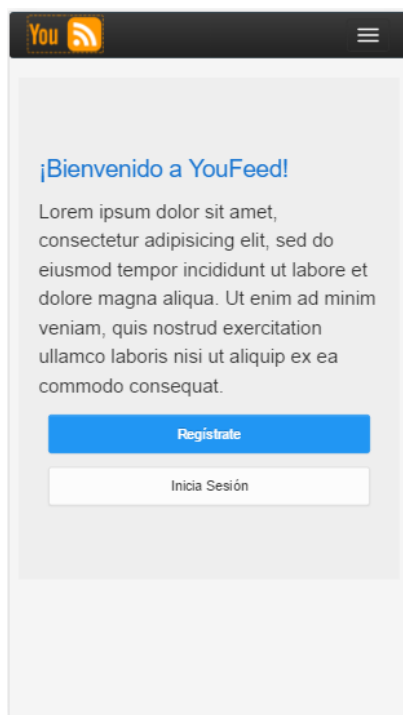


Ilustración 36 Mockup - Version mobile 1

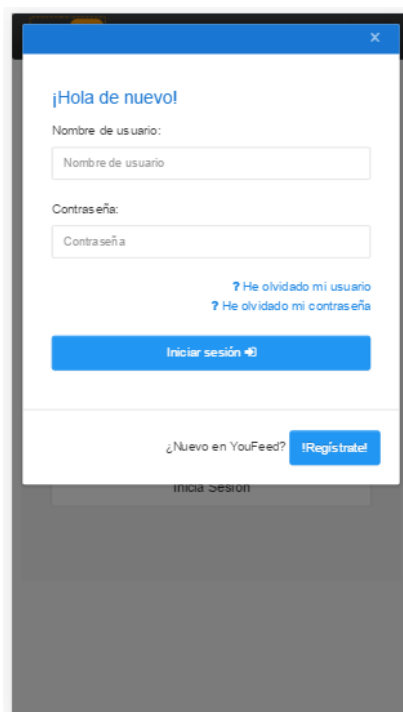


Ilustración 37 Mockup - Versión mobile 2

3. Tests

A la hora de realizar los tests, se han tomado los casos de uso uno por uno y seguido los pasos marcados en los mismos.

Durante el desarrollo de los tests, indicar que fue problemático especialmente el registro de usuario, ya que el cifrado de las contraseñas no se realizaba correctamente, por lo que se decidió para la primera versión de la aplicación prescindir de la validación de contraseña a la hora de autenticar, es decir, ha de mandarse un texto cualquiera pero este no se validará a la hora de hacer login, simplemente se tendrá en cuenta el usuario.

Por lo demás, en esta fase se han resuelto pequeños bugs, exceptuando además de la gestión de las contraseñas de los usuarios la gestión de los usuarios en sí misma, ya que los componentes que componen la aplicación acceden en su mayoría al usuario en sesión, cuya actualización por parte de un componente no era inmediatamente visible por parte de los demás, siendo necesaria el refresco de la aplicación (pulsar F5). Se hubo de cambiar pues la manera en la que se gestionaba el usuario en el servicio UserService para que, una vez se alterase el mismo, fuera emitido al resto de componentes de la aplicación, que están suscritos a un atributo de este componente, que además tuvo que ser modificado para comportarse como Singleton, es decir, que la misma instancia de este servicio fuese usada en toda la aplicación.

4. Ejemplos de uso del producto

Durante este apartado mostraremos y explicaremos algunas funcionalidades relevantes de nuestra aplicación mediante capturas de pantalla. La versión usada es la de escritorio.

1. Pantalla principal, registro y autenticación

Al acceder a la aplicación, se mostrará en el caso de no encontrarse autenticado la siguiente pantalla, en la que se nos da la opción de iniciar sesión o bien registrarnos:

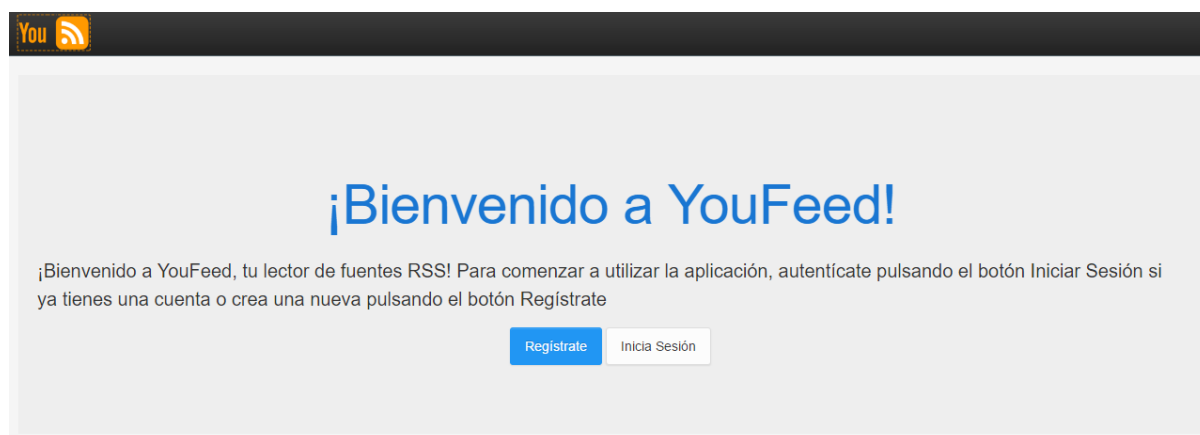


Ilustración 38 Pantalla principal sin autenticarse en la aplicación

En el caso de no encontrarse registrado, pulsando el botón "Regístrate", se mostrará un panel en el que el usuario deberá introducir los datos de su cuenta:

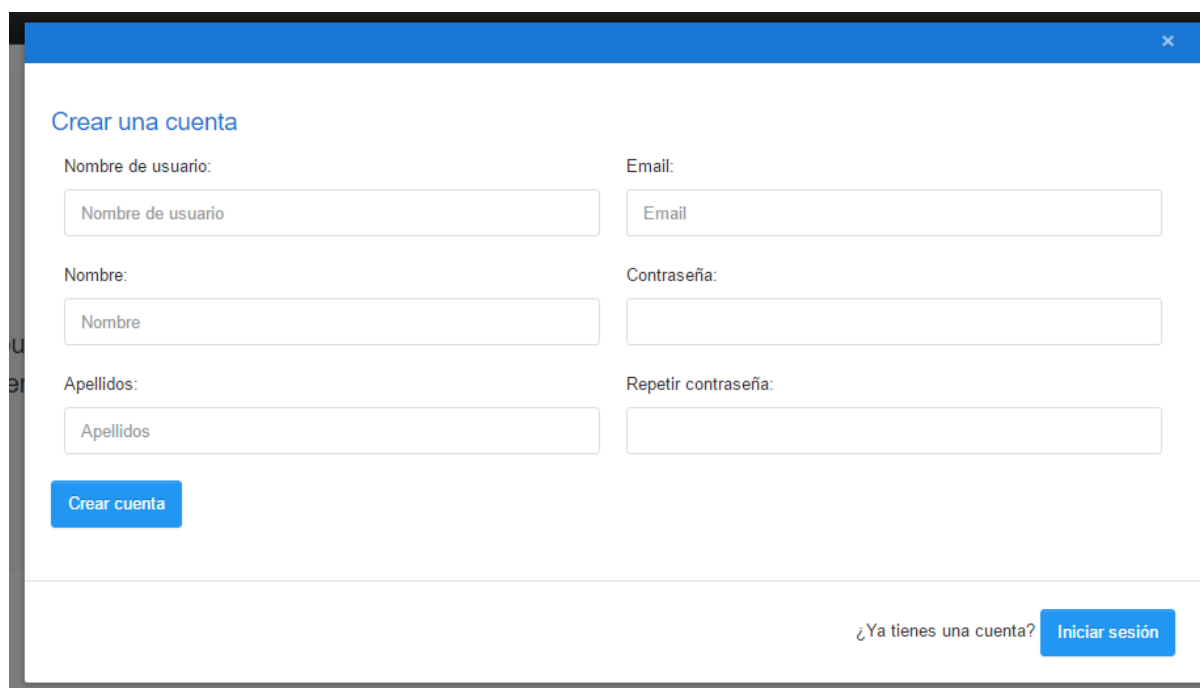
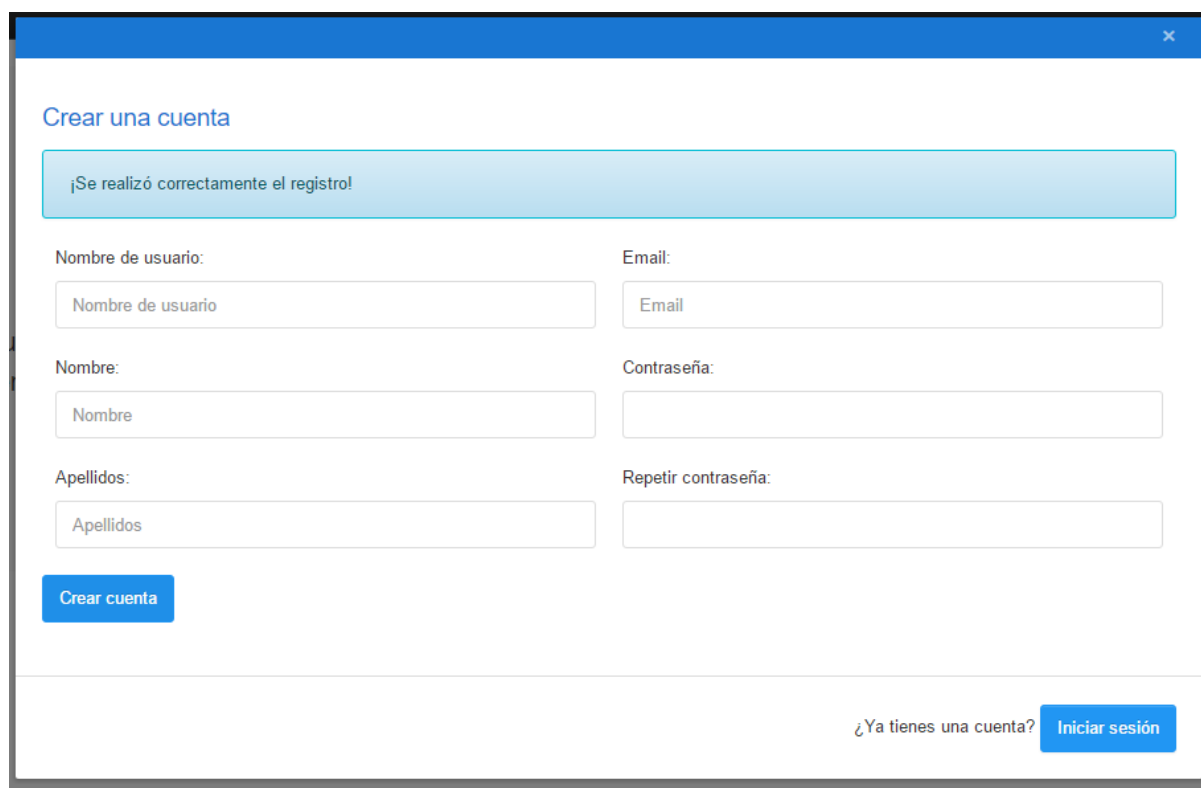


Ilustración 39 Crear una cuenta

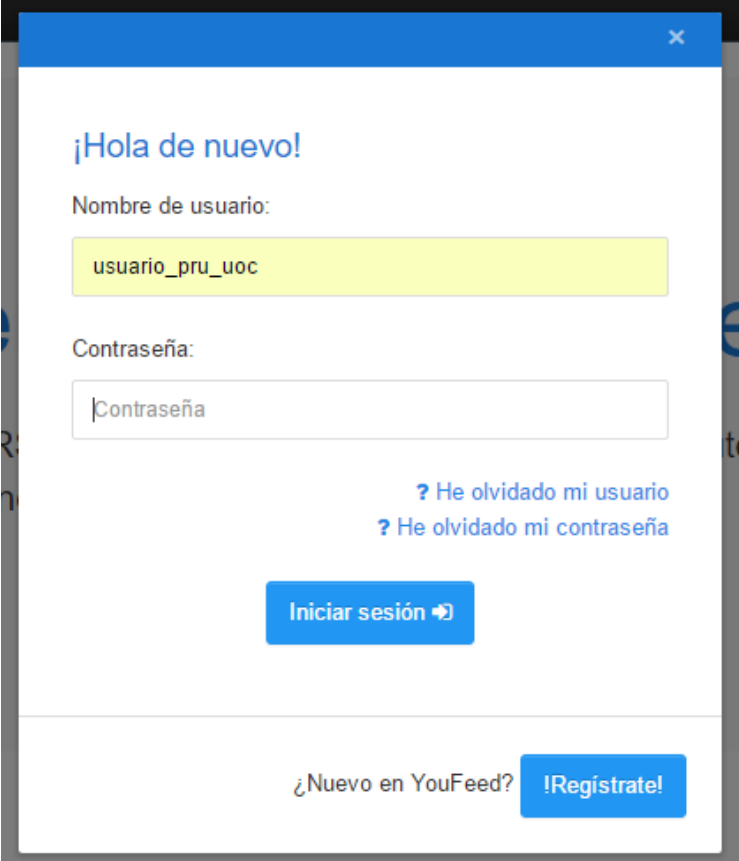
Tras haber introducido datos válidos (en caso de no serlos, se indicará), se mostrará el panel de la ilustración inferior. Nótese que la contraseña en esta versión no se gestiona, por lo que puede introducirse 12345678 para que el formulario permita el registro, ya que exige que sea como mínimo de una longitud de 5 caracteres.



The screenshot shows a web form titled "Crear una cuenta" (Create an account) with a blue header bar. A light blue success message box at the top reads "¡Se realizó correctamente el registro!". Below this, the form is organized into two columns. The left column contains three input fields: "Nombre de usuario" (User name), "Nombre" (First name), and "Apellidos" (Last name). The right column contains three input fields: "Email", "Contraseña" (Password), and "Repetir contraseña" (Repeat password). A blue "Crear cuenta" (Create account) button is positioned below the left column. At the bottom right, there is a link "¿Ya tienes una cuenta?" (Do you already have an account?) next to a blue "Iniciar sesión" (Log in) button.

Ilustración 40 Crear una cuenta (registro realizado)

Tras lo cual ya podrá accederse a la aplicación, obviando la contraseña. En nuestro caso hemos usado el usuario usuario_pru_uoc, dado de alta y con datos iniciales establecidos:



The image shows a login window with a blue header bar containing a close button (X). The main content area is white and features the following elements:

- Greeting: "¡Hola de nuevo!" in blue text.
- Label: "Nombre de usuario:" in grey text.
- Input field: A yellow-highlighted text box containing the text "usuario_pru_uoc".
- Label: "Contraseña:" in grey text.
- Input field: A white text box with a grey border containing the text "Contraseña".
- Links: Two blue links with question marks: "? He olvidado mi usuario" and "? He olvidado mi contraseña".
- Button: A blue button with white text "Iniciar sesión" and a right-pointing arrow.
- Footer: A grey bar containing the text "¿Nuevo en YouFeed?" and a blue button with white text "¡Regístrate!".

Ilustración 41 Inicio de sesión

2. Pantalla Principal / Hoy

Al autenticarse en la aplicación, se mostrará la pantalla principal, que muestra los contenidos obtenidos de las suscripciones realizadas por el usuario anteriormente en una tabla que es ordenable. En caso de no haber suscripciones o no recuperarse noticias, se indicará.

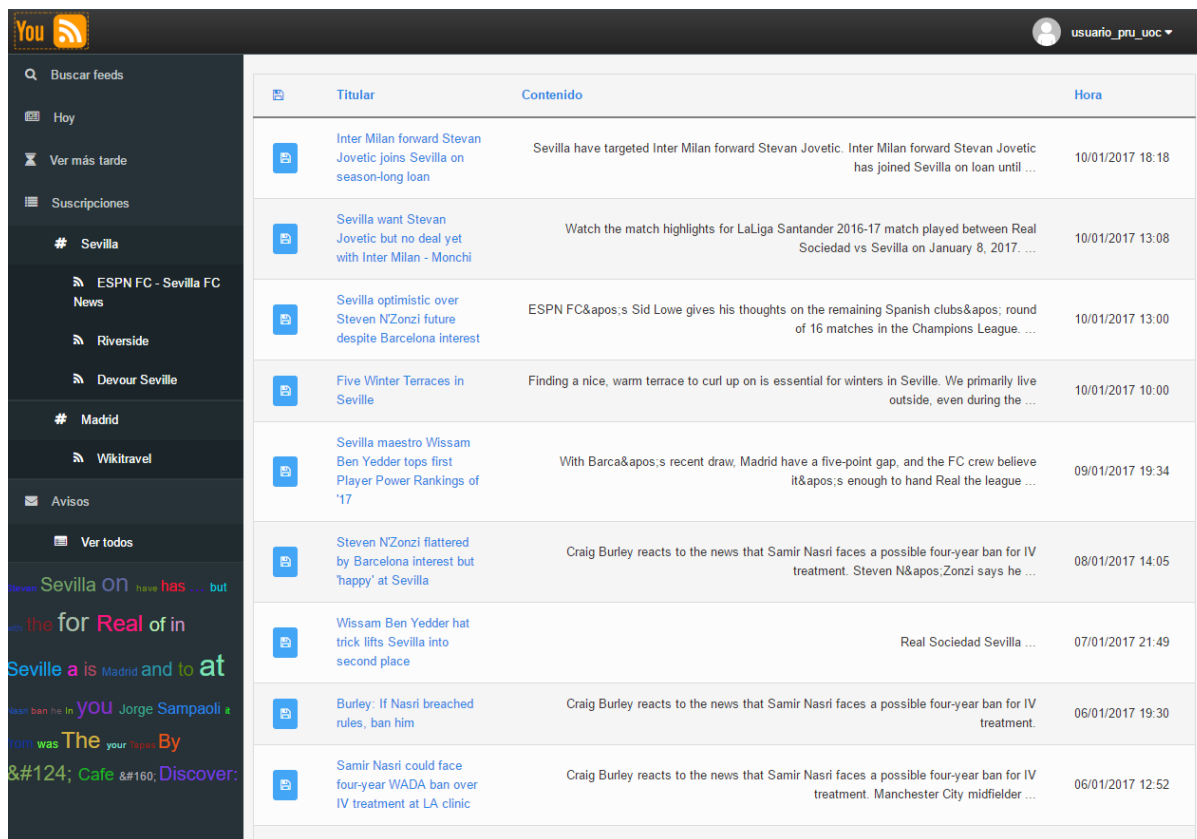


Ilustración 42 Hoy

Además, como puede verse en la ilustración anterior, se muestra una barra superior en la que el usuario puede gestionar sus datos y una barra lateral que permite el acceso a los contenidos de la aplicación y que muestra una nube de palabras con las palabras con más ocurrencias en los contenidos obtenidos de las suscripciones.

En caso de desear visualizar una noticia, bastará con pulsar sobre el titular de la misma:

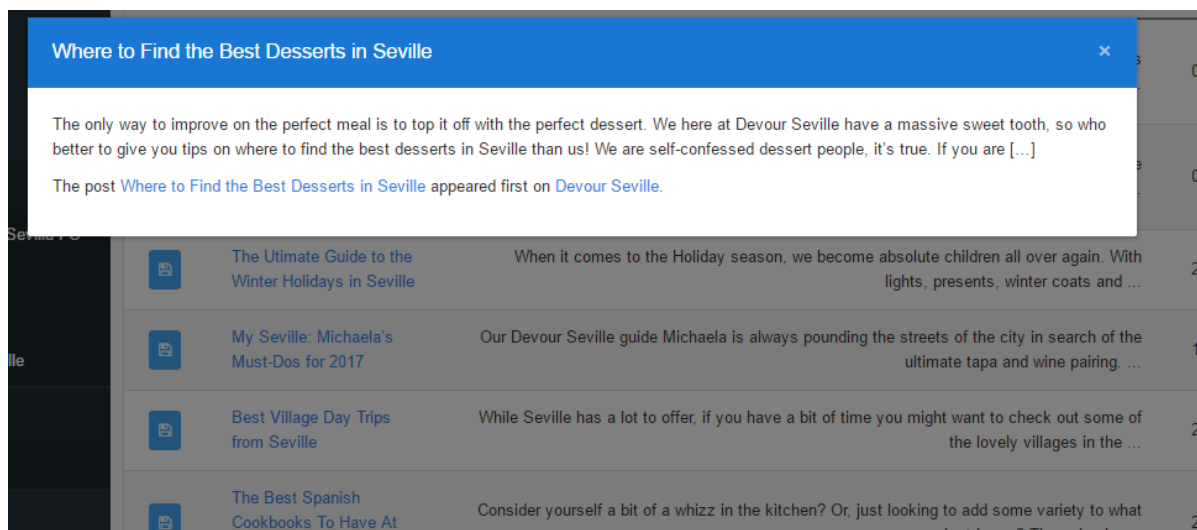



Ilustración 43 Visualización de contenido de noticia

Además, al pulsar sobre el botón  a la izquierda de una noticia, se dará la opción de guardarla tal cual en el sistema para posterior visualización en el apartado "Ver Más Tarde".

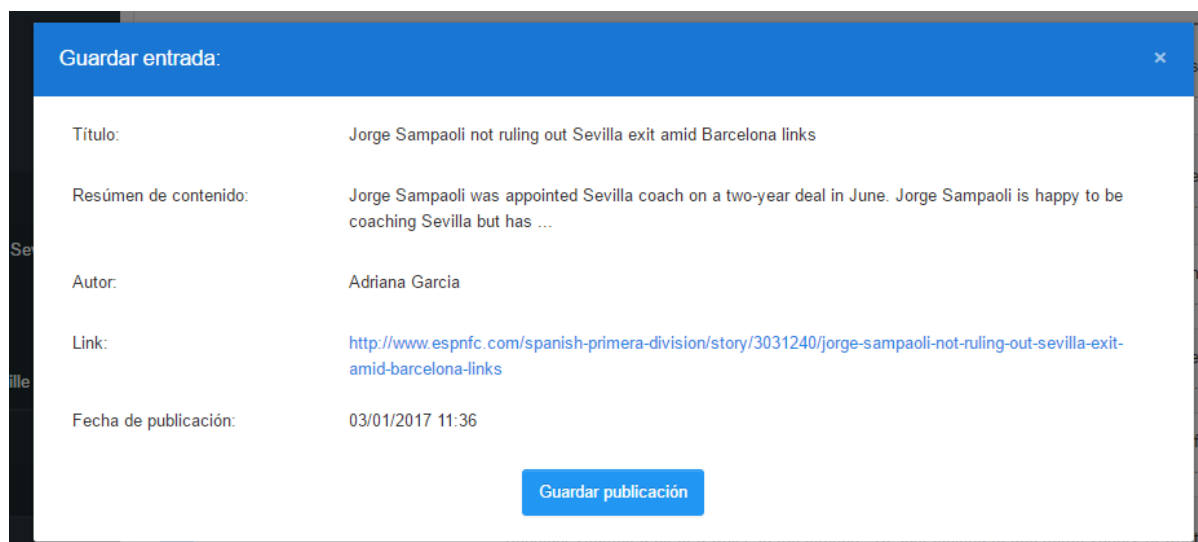


Ilustración 44 Guardar entrada

3. *Buscar feeds*

Al pulsar en el apartado “Buscar feeds” de la barra lateral, se abrirá la pantalla correspondiente, que inicialmente muestra un input en el que introducir un contenido a buscar y un botón para ejecutar la búsqueda:

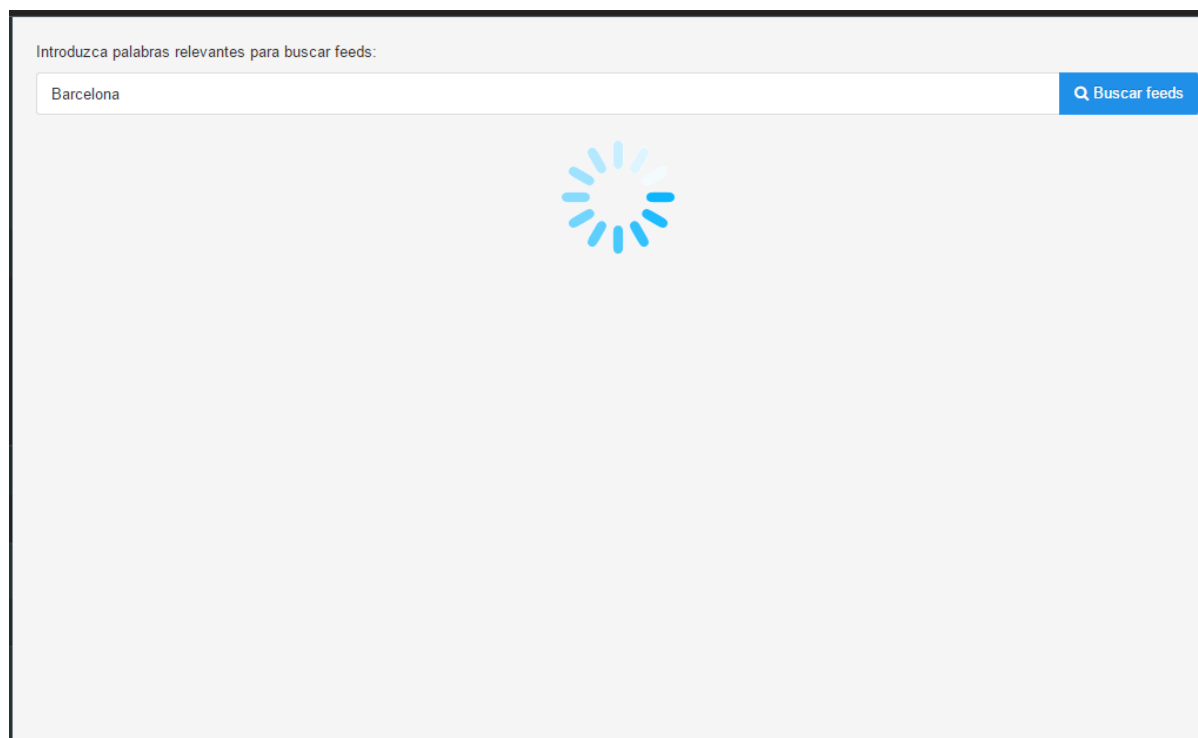


Ilustración 45 Buscar feeds - cargando

Una vez completada la búsqueda, se mostrarán todas las fuentes encontradas, con imagen que las represente si se encuentra alguna en sus contenidos:

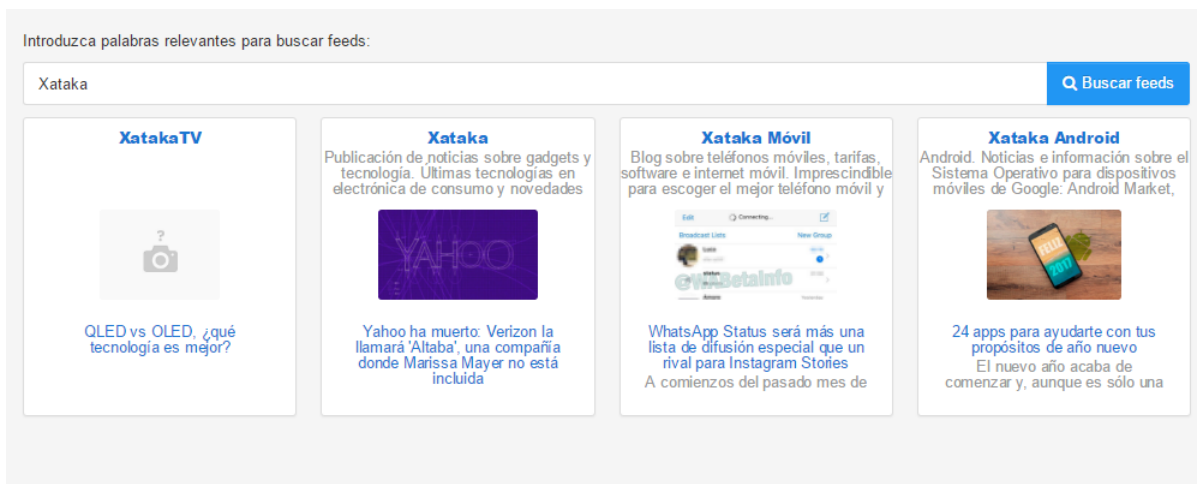


Ilustración 46 - Buscar feeds (con imagen)

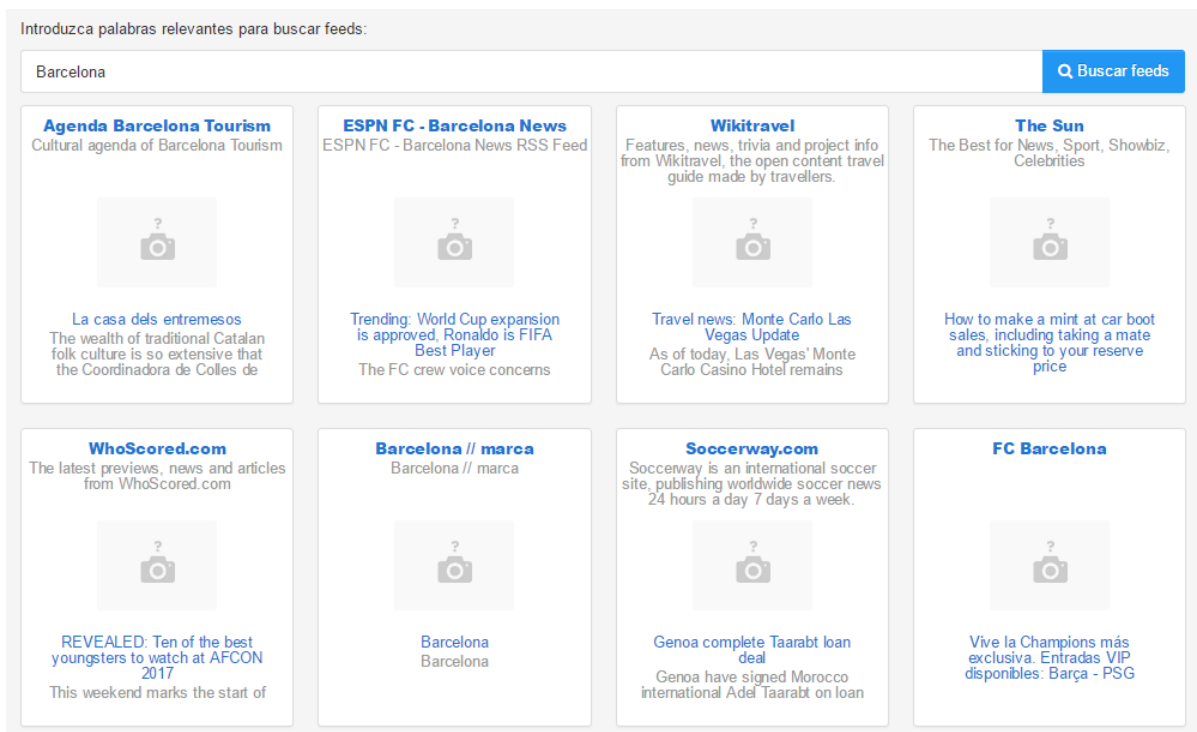
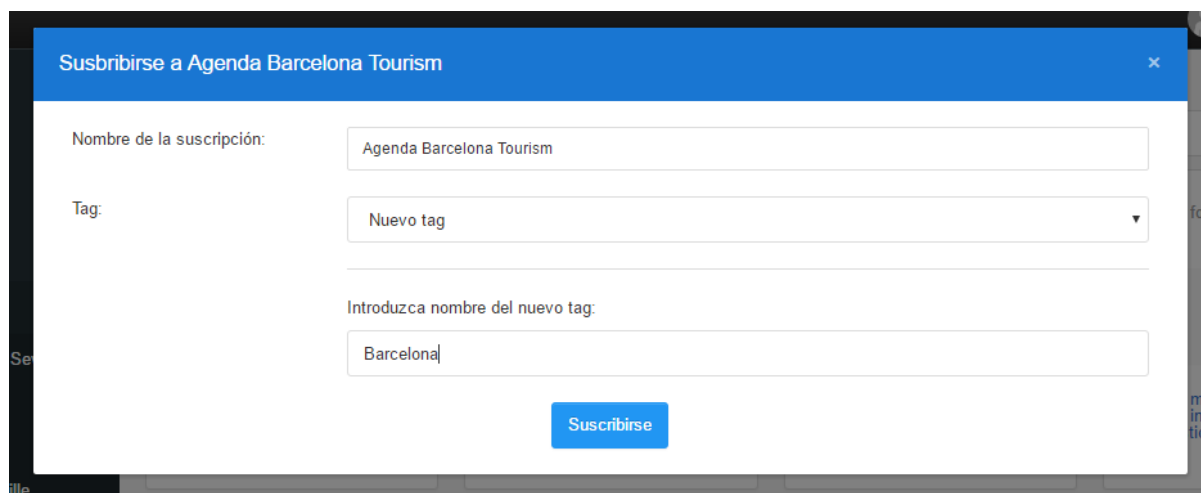


Ilustración 47 Buscar feeds (sin imagen)

Al situar el ratón sobre una de las imágenes, se dará la opción a suscribirse a la fuente, ver la última publicación de la misma o abrir en una nueva pestaña la url de la publicación.



The image shows a web form titled "Suscribirse a Agenda Barcelona Tourism" with a close button (x) in the top right corner. The form contains three input fields: "Nombre de la suscripción:" with the value "Agenda Barcelona Tourism", "Tag:" with a dropdown menu showing "Nuevo tag", and "Introduzca nombre del nuevo tag:" with the value "Barcelona". A blue "Suscribirse" button is located at the bottom center of the form.

Ilustración 48 Suscribirse a un feed

En la ilustración anterior, tras pulsar que se deseaba suscribir a la primera fuente encontrada, hemos creado la suscripción “Agenda Barcelona Tourism” (es decir, hemos usado el mismo nombre de la misma, pero podríamos haberlo cambiado) y la hemos asignado a un nuevo tag llamado Barcelona.

4. Suscripciones

En la barra lateral se presentan las suscripciones, junto al tag al que se asignaron, y se puede acceder a los contenidos de las mismas haciendo click en su nombre:

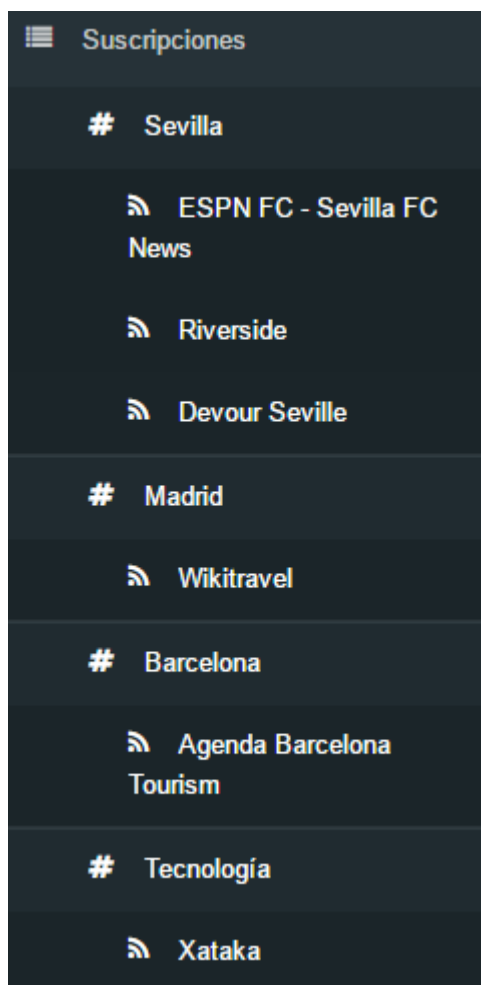


Ilustración 49 Suscripciones en menú lateral

Así, accederemos a la pantalla de la suscripción, que listará los contenidos y nos permitirá paginarlos de 5 a Todos, ver aquellos con fecha de hoy y hasta ayer y filtrar por contenido:

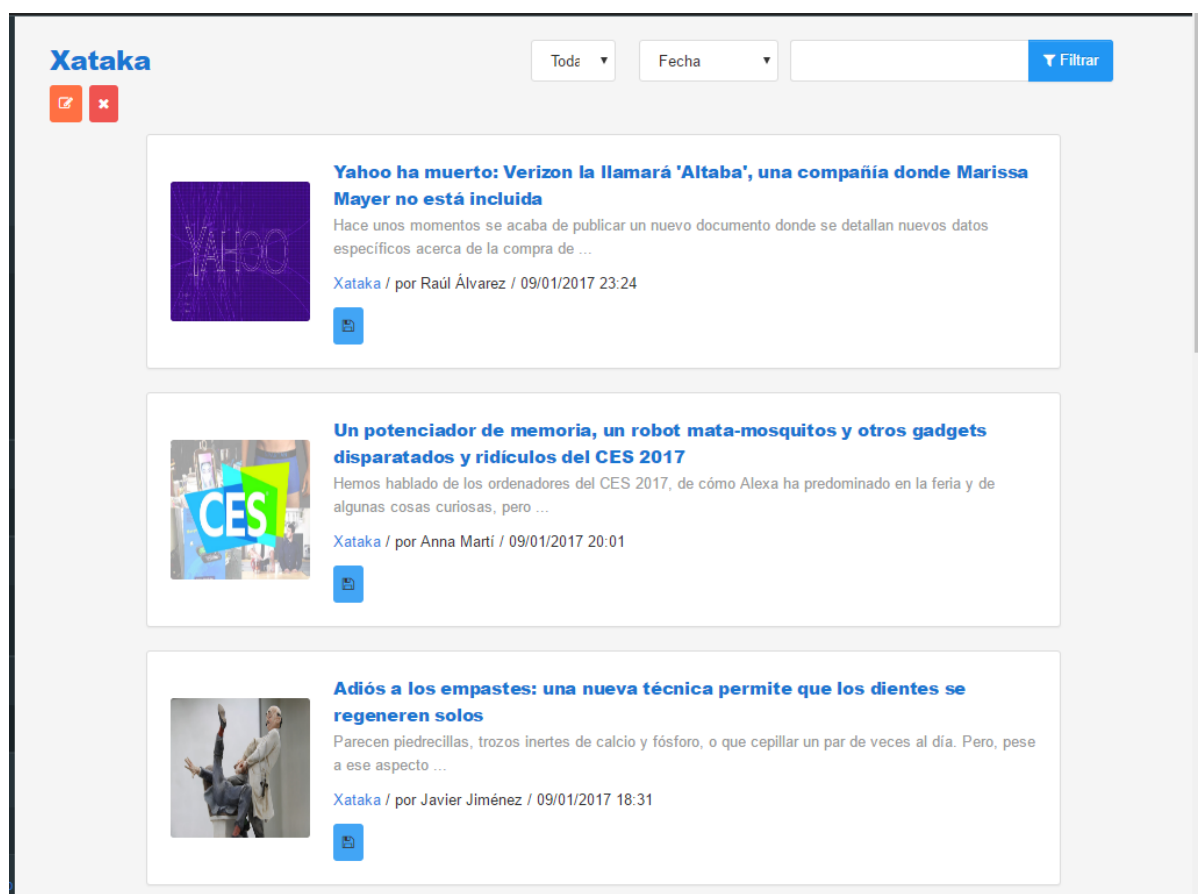


Ilustración 50 Ver una suscripción

Además, también se da la opción a guardar estas noticias, y de la misma manera que en la pantalla principal, se podrá visualizar el contenido completo de una noticia al hacer click sobre el titular de la misma:

Yahoo ha muerto: Verizon la llamará 'Altaba', una compañía donde Marissa Mayer no está incluida

Hace unos momentos se acaba de publicar un [nuevo documento](#) donde se detallan nuevos datos específicos acerca de [la compra de Yahoo por parte de Verizon](#). Aquí tenemos un primer acercamiento a lo que piensa hacer el operador y su estrategia a plazo medio, lo cual suena lógico hasta cierto punto después de los problemas que ha enfrentado la compañía de Marissa Mayer.

En el breve documento se destacan dos puntos que nos llaman la atención: por un lado la reducción de la junta directiva de la compañía, donde se anuncia la [renuncia de Marissa Mayer](#) como CEO, y por otro lado el cambio de nombre de Yahoo, que pasaría a ser conocida como [Altaba Inc.](#)

Los primeros cambios de una dañada Yahoo

La marca Yahoo está profundamente dañada, esto después de dos grandes [filtraciones de información](#) que expusieron los datos de [más de 1.500 millones de usuarios](#), lo que se coloca como el más grande en la historia de internet y la seguridad informática. Y ahora empezamos a ver los primeros cambios y movimientos que tratan de sanar los daños, sobre todo a la marca, donde estaba el rumor de que Verizon buscaba [pagar menos por la compañía](#), e incluso [cancelar la compra](#).

La junta directiva de Yahoo está informando a sus inversores algunos cambios que se aplicarán a partir del cierre de la compra por parte de Verizon. El primero de ellos es la reducción de dicho consejo a sólo cinco directores, lo que [dejará fuera a los otros seis](#) que integraban el consejo: David Filo, Eddy Hartenstein, Richard Hill, Jane Shaw, Maynard Webb, y la actual CEO Marissa Mayer.

Ilustración 51 Ver noticia de una suscripción

Guardar entrada:

Título:	Yahoo ha muerto: Verizon la llamará 'Altaba', una compañía donde Marissa Mayer no está incluida
Resumen de contenido:	Hace unos momentos se acaba de publicar un nuevo documento donde se detallan nuevos datos específicos acerca de la compra de ...
Autor:	Raúl Álvarez
Link:	http://feeds.weblogssl.com/~r/xataka2/~3/KIRPSAYSrsc/yahoo-ha-muerto-verizon-la-llamara-altaba-una-compania-donde-marissa-mayer-no-esta-incluida
Fecha de publicación:	09/01/2017 23:24

Guardar publicación

Ilustración 52 Guardar entrada de una suscripción



Ilustración 53 Filtros en suscripción

Además, nótese que mediante los botones debajo del nombre de la suscripción, se permite modificar o eliminar la misma.

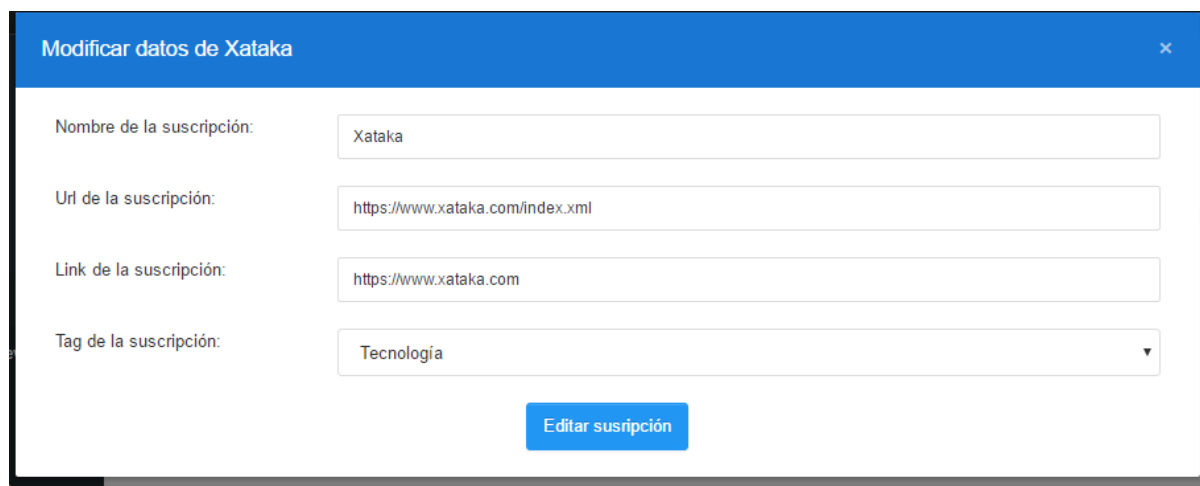


Ilustración 54 Modificar datos de suscripción

5. Ver más tarde

Es una pantalla muy similar a la anterior, con la diferencia de que en este caso se listan las noticias que se guardaron y que las mismas pueden ser eliminadas del sistema.

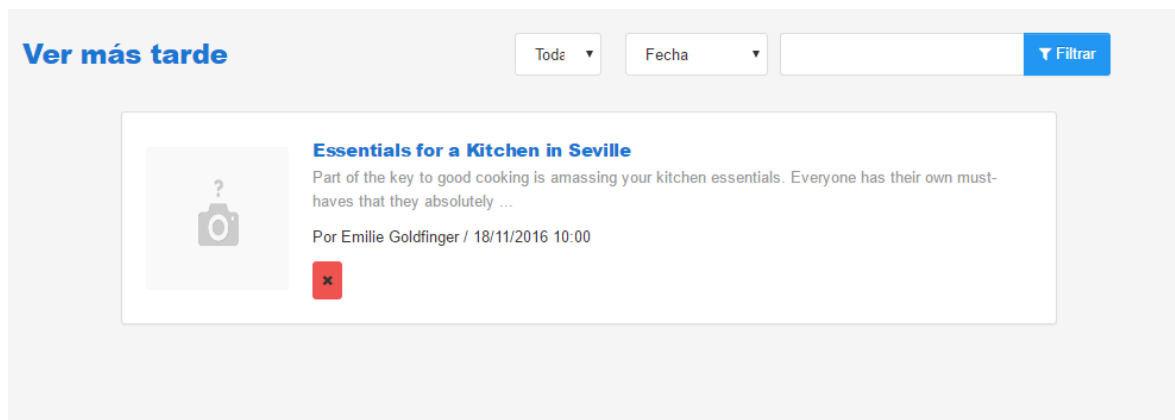


Ilustración 55 Ver más tarde

6. Avisos (Ver todos)

En esta pantalla se listan los avisos existentes, se permite activarlos, modificarlos y eliminarlos pulsando en la opción deseada en Acciones y se permite crear nuevos avisos.

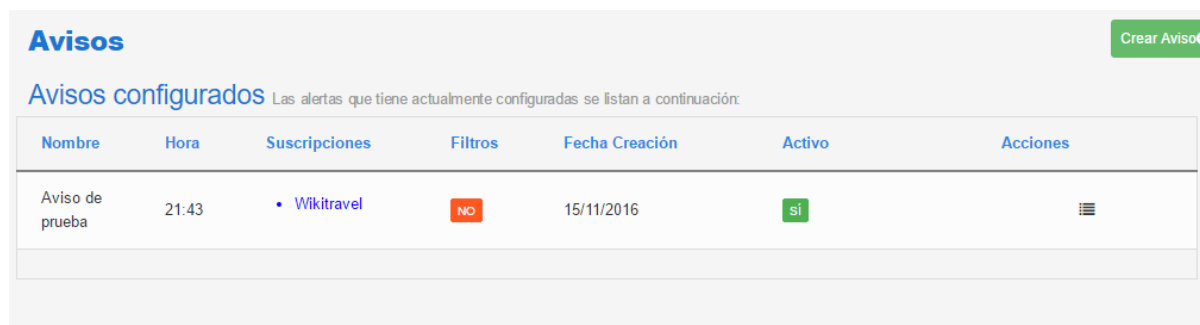


Ilustración 56 Avisos

En el ejemplo siguiente se ilustra la creación de un aviso para nuestra suscripción Xataka:

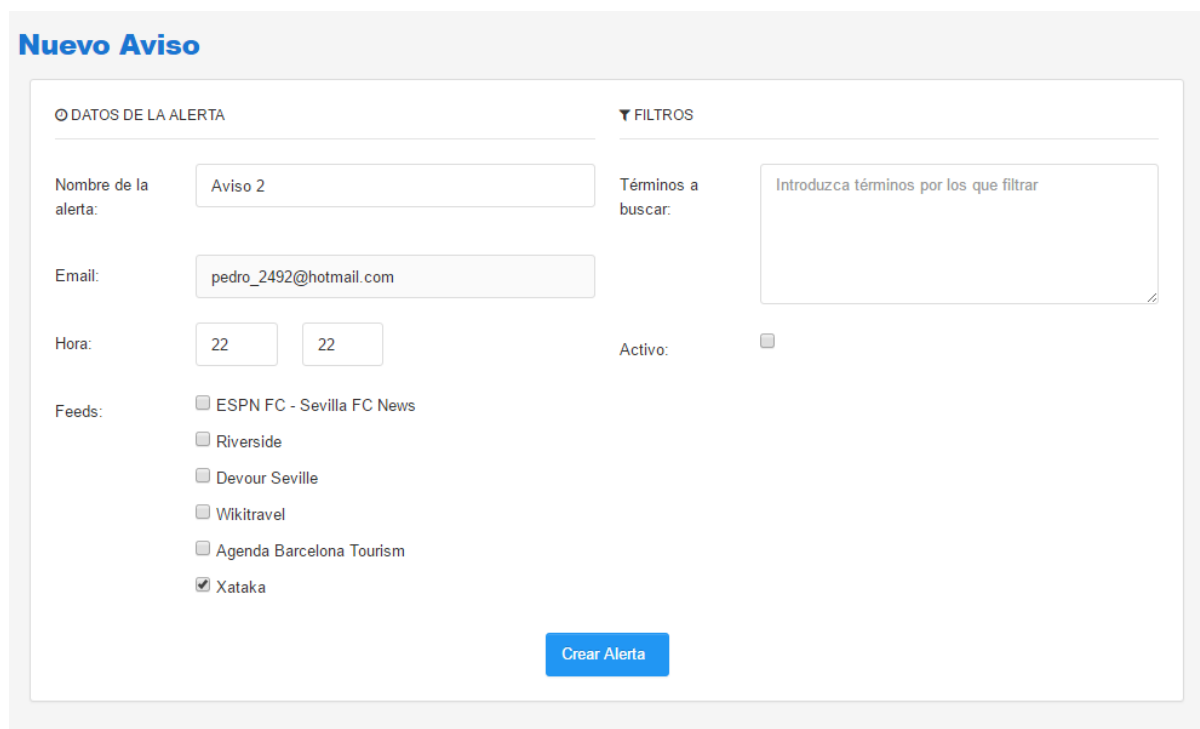


Ilustración 57 Nuevo aviso

A la hora deseada, y si el aviso está activo, se enviara un correo a la cuenta del usuario de este tipo:

A continuación se presentan las noticias correspondientes a la alerta Aviso 2:

Xataka

- [Esta centrifugadora cuesta 20 centavos y puede cambiar la medicina en medio mundo](#)
[Ver Noticia](#)
- [Los 38 juegos que más esperamos de 2017](#)
[Ver Noticia](#)
- [En los Países Bajos se anticipan un año y el 100% de sus trenes circula con energía eólica](#)
[Ver Noticia](#)
- [Un mes usando Smappee, el gadget para conocer al instante el consumo de cada electrodoméstico](#)
[Ver Noticia](#)
- [Los "avisos de cookies" no sirven y hasta la UE lo sabe: propone "reducirlos" e incluso eliminarlos](#)
[Ver Noticia](#)
- [El apéndice puede no ser tan inservible como pensábamos hasta ahora](#)
[Ver Noticia](#)
- [En Japón ya puedes reservar tu sala privada para disfrutar del porno en realidad virtual](#)
[Ver Noticia](#)

Ilustración 58 Ejemplo de aviso por correo

Capítulo 6: Conclusiones y líneas de futuro

1. Conclusiones

Mediante la realización de este proyecto o trabajo, se ha conseguido desarrollar desde cero una aplicación web, compuesta por una base de datos, un back-end y un front-end. Cabe destacar que a nivel profesional ha servido para conocer y coger cierta soltura con tecnologías del front-end, especialmente el framework AngularJS 2.0. Por otra parte, se ha tenido la oportunidad de poder planificar y gestionar un proyecto desde su inicio a su fin, enfrentándose a la complejidad que supone, con errores de estimación e imprevistos y aprendiendo en cierto modo a gestionarlos, aprendiendo la lección de que se debe ser más pesimista a la hora de estimar, dejando tiempo entre tareas que sirvan como 'colchón' ante los anteriormente mencionados errores de estimación y eventos inesperados, tanto internos como externos al proyecto.

No obstante, se ha conseguido desarrollar una aplicación con un nivel de calidad aceptable, si bien es cierto que dos de los objetivos inicialmente planteados quedaron en el tintero: Integrar Google Analytics para el análisis de la misma y la interacción con las redes sociales para poder compartir contenidos en las mismas. Ello es debido esencialmente al retraso producido en la PEC4 por desconocimiento inicial de tecnologías del front-end y por las exigentes circunstancias laborales que sufrió el realizador de este proyecto, que produjeron el ya mencionado retraso y que, pese a acelerarse el ritmo súbitamente en el cierre del proyecto, no se pudieron salvar.

Para concluir, volvemos a incidir en la metodología y planificación del proyecto. Si bien la metodología elegida no es inadecuada (desarrollo en cascada), como se ha mencionado anteriormente varias veces nuestra planificación ha sufrido varios imprevistos, por lo que ha sido necesario extender y acortar a veces partes del ciclo, o trasladar a sucesivas entregas funcionalidades planificadas.

Por ello, se ha tenido que replanificar en cada entrega el trabajo restante del proyecto, quedándose afuera como se ha indicado ya dos funcionalidades que habrán de completarse a futuros.

2. Líneas de futuro

En primer lugar, un desarrollo que sería muy útil y usable y que podría empezar a desarrollarse desde este momento es realizar una versión móvil de nuestra aplicación de manera que ésta pueda ser usada de manera nativa por dispositivos móviles ya sean Android o IOs, ya que durante la ejecución del proyecto se ha desarrollado una API REST que actualmente usa la aplicación web, es decir, el front-end, para interactuar con los datos. Indicar que se barajó desde un primer momento la posibilidad de realizar una aplicación móvil en paralelo, pero quedó fuera del alcance y para paliar este hecho se desarrolló una aplicación *responsive*.

Por otra parte, hablando de la aplicación actual en sí, sería pertinente exponerla a un grupo de usuarios, para así recibir un *feedback* y poder realizar cambios según las peticiones de los usuarios que prueben la aplicación. Nótese también que la autenticación en estos momentos sólo valida nombre de usuario (no controla la contraseña), por lo que inminentemente debería de realizarse este cambio antes de subir la aplicación a un entorno de producción.

Bibliografía

- Agregador o lector de RSS [Wikipedia] <https://es.wikipedia.org/wiki/Agregador>
- Alternativas a Feedly: los 7 mejores lectores de feeds RSS [genbeta.com].
<http://www.genbeta.com/a-fondo/alternativas-a-feedly-los-7-mejores-lectores-de-feeds-rss>
- Angular 2 services [Budacode blog]
<https://blog.budacode.com/2016/06/02/angular-2-services/>
- Angular tutorial [Angular.io] <https://angular.io/docs/ts/latest/tutorial/>
- Aplicación web Feedly [Feedly]. <https://feedly.com/>
- Aplicación web InoReader [Inoreader]. <http://www.inoreader.com/>
- Bootstrap [getbootstrap.com] <http://getbootstrap.com/>
- Google Feed API [Google] <https://developers.google.com/feed/v1/reference?hl=es>
- Google Feed API [Google] <https://developers.google.com/feed/v1/reference?hl=es>
- Google Reader [Google]. <https://www.google.com/reader/about/>
- Introducción a RSS [support.office.com].
<https://support.office.com/es-es/article/Introducci%C3%B3n-a-RSS-5d3b01d7-f143-4e2e-a307-b1f5c8b7c177>
- JQuery [jquery.com] <https://jquery.com/>
- Lectores RSS online [rss.nom.es] <http://www.rss.nom.es/lector-rss-online-web/>
- RSS [Wikipedia]. <https://es.wikipedia.org/wiki/RSS>
- Spring boot [Spring projects] <https://projects.spring.io/spring-boot/>