

INFORME INGENIERÍA WEB

YOUTUBE

Miembros del equipo:

Encarna Amorós Beneite

Jesús Gallego Irles

Manuel José Verdú Ramón

Índice

Introducción/descripción general.....	5
Mockups y wireframes	6
Mockup.....	6
Pantalla de inicio.....	6
Pantalla de búsqueda	6
Pantalla video	7
Pantalla subir vídeo	8
Pantalla editar vídeo	9
Pantalla mi canal.....	9
Pantalla canal	10
Pantalla backoffice.....	11
Pantalla login	11
Pantalla registro	12
Wireframe.....	13
Búsqueda de vídeos	13
Ver mi canal.....	14
Ver canal usuario	14
Para subir vídeos	15
Para editar mi vídeo	16
Pantalla backoffice.....	16
Iniciar sesión y registro	17
Patrones utilizados y justificación	18
Patrones de pantalla principal	18
Patrones de navegación	19
Patrones de formulario.....	20
Patrón de identificación del usuario.....	22
Patrones de listados	22

Patrones de búsqueda en el sitio	23
Patrones RIA utilizados	24
Formulario inteligente	24
Autocompletado o sugerencias de campos.....	24
Consultas dinámicas.....	25
Animaciones	25
Diagramas	26
Diagrama de casos de uso	26
Diagrama de datos	27
Descripción técnica del sistema	29
PHP y CodeIgniter.....	29
Grocery CRUD	29
Javascript.....	30
jQuery.....	30
HTML5	31
CSS3.....	31
XAMPP.....	32
Detalles de implementación	33
Pantalla principal	33
Cabecera.....	33
Pantalla de búsqueda.....	33
Pantalla video.....	34
Pantalla subir vídeo	34
Pantalla editar vídeo.....	35
Pantalla canal.....	35
Pantalla backoffice	36
Pantalla login.....	36
Pantalla registro	37
Problemas encontrados y su solución.....	38

Mejoras y ampliaciones.....	40
Ampliaciones pantalla canal	40
Ampliaciones pantalla canal y video	40
Ampliaciones vídeo	40
Mejora en subir/editar vídeo	40
Referencias.....	42
Informe	42
Funcionalidades y diseño	42
Otras herramientas de trabajo	43

Introducción/descripción general

Hemos desarrollado la web Youtube. En ella cualquier usuario o invitado puede ver un listado de los videos más populares, buscar videos, ver un video en detalle (sus comentarios, el vídeo en sí, quien lo subió, vídeos relacionados...), ver canales en detalle (el canal de un usuario con sus vídeos subidos, comentarios y usuarios relacionados) y también puede registrarse o iniciar sesión si ya dispone de una cuenta.

Un usuario identificado o que ha iniciado sesión, tiene acceso a las mismas funciones, pero además puede subir videos y comentar canales de otros usuarios o su propio canal (pues al identificarse tiene acceso a su propio canal). En su canal puede añadir una serie de usuarios relacionados. También puede comentar los distintos vídeos que hay en la web.

Finalmente, hay cuentas especiales para los administradores de la página. Con estas cuentas dichos usuarios pueden añadir, ver, editar o eliminar los elementos de una serie de tablas (licencias, categorías, etiquetas, idiomas, lenguajes y calidades).

Mockups y wireframes

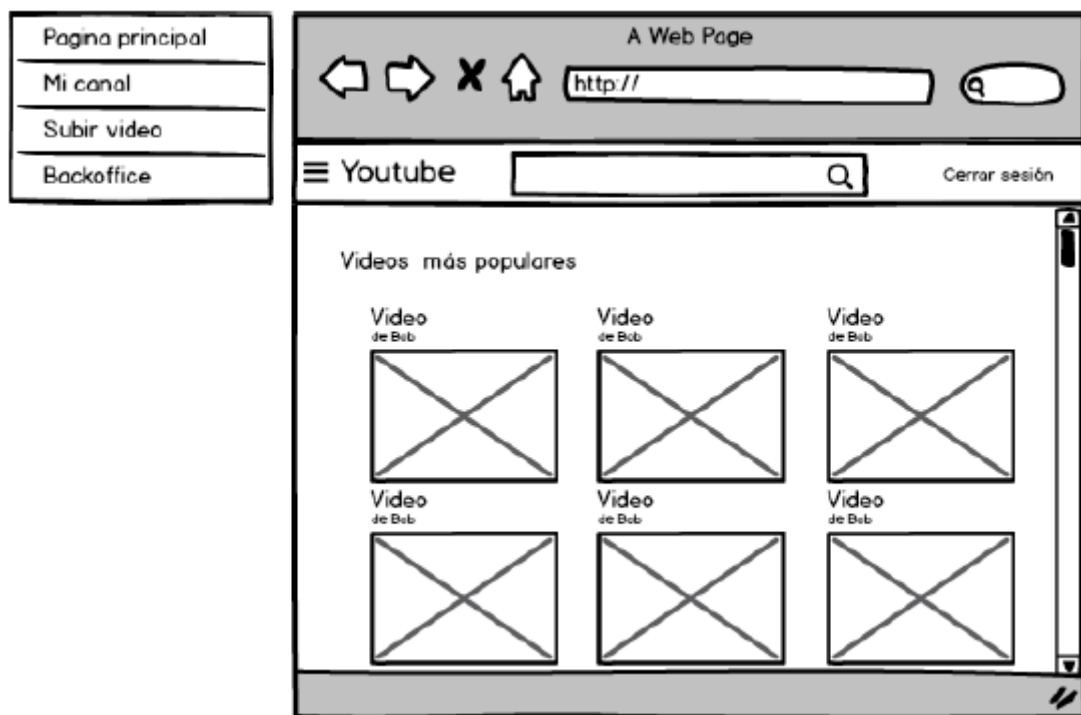
En este apartado mostraremos los mockups y wireframes realizados, así como los patrones de diseño web aplicados con su descripción y justificación.

Mockup

A continuación mostraremos los mockups implementados para el desarrollo de la web. Estos mockup fueron una fase inicial del wireframe que explicaremos después. En cada mockup, explicaremos de forma breve las funcionalidades que se podrán realizar en esa pantalla determinada.

Pantalla de inicio

Pantalla en la cual se mostrarán los primeros 21 vídeos más populares, es decir, con más visualizaciones/reproducciones.



Pantalla de búsqueda

Pantalla donde podemos ver paginados los resultados de una búsqueda de vídeos que hagamos. Buscaremos fijándonos en el título, descripción y etiquetas de los vídeos. También

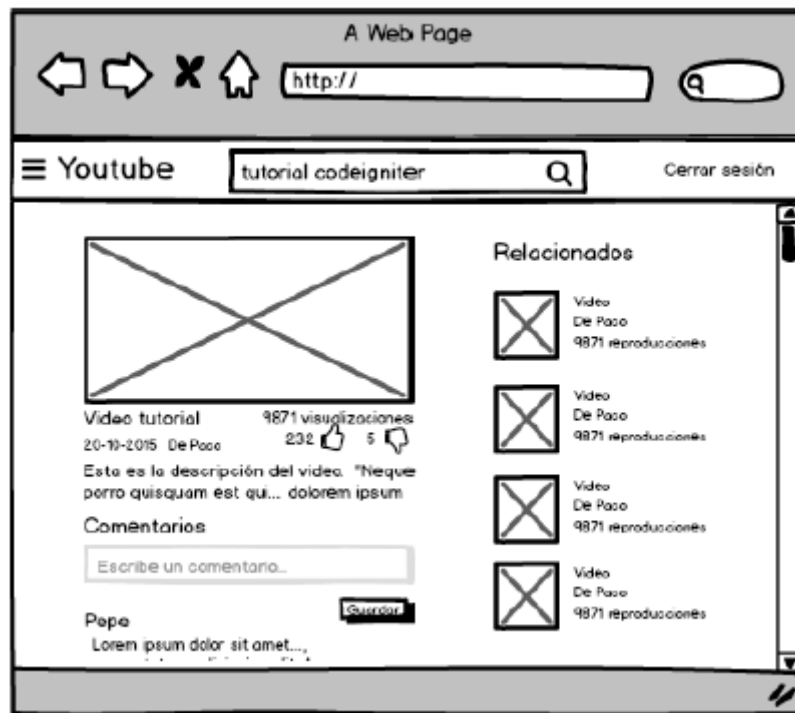
por el usuario que subió el vídeo. Se puede observar la barra de búsqueda en la cabecera, desde donde podemos utilizar esta funcionalidad.



Pantalla video

En esta pantalla vemos un vídeo en detalle: número de visualizaciones, de me gustas y no me gustas, el usuario que lo creó, su descripción, los comentarios del vídeo, vídeos relacionados con ese... Los relacionados son según etiquetas compartidas entre vídeos o títulos similares. Finalmente se verían más vídeos del usuario que subió el que vemos.

Podemos comentar el vídeo si hemos iniciado sesión (también eliminar los comentarios que hayamos hecho).



Pantalla subir vídeo

Si hemos iniciado sesión podemos acceder a este formulario para subir un nuevo vídeo. El título y la URL serán campos obligatorios que trataremos de señalar adecuadamente.

Pantalla editar vídeo

Si hemos iniciado sesión, desde nuestro canal y desde donde vemos todos nuestros vídeos subidos, podemos editar el que queramos. Este es el formulario para editar un vídeo que hayamos subido. Desde esta pantalla también podemos eliminar el vídeo.

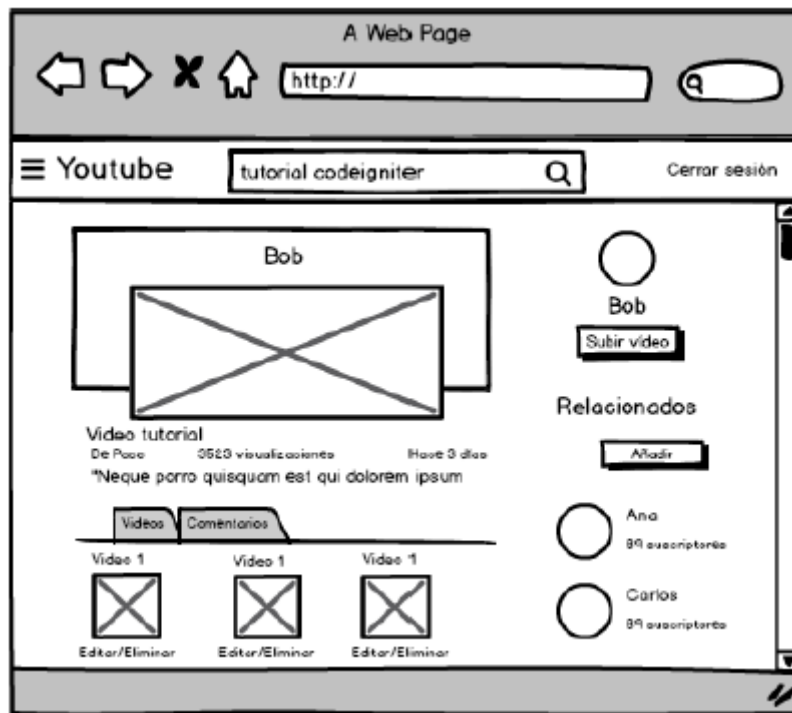
Nota: La URL no se puede modificar.



The image shows a browser window titled "A Web Page" with a navigation bar containing back, forward, and home icons, and a search bar with "http://". Below the browser window is a YouTube interface. The top bar includes the YouTube logo, a search bar with "tutorial codeigniter", and a "Cerrar sesión" link. The main content area is titled "Editar video" and contains several form fields: "(*): Título:" with a text box containing "Video ejemplo"; "(*): URL del video:" with a text box containing "https://www.youtube.com/watch?"; "Descripción del video:" with a text box containing "Una descripción del video que ahora estoy editando"; "Visibilidad:" with a dropdown menu showing "una de ej."; "Tipo de licencia:" with a dropdown menu showing "una de ej."; "Categoría:" with a dropdown menu showing "una de ej."; "Idiomas:" with a dropdown menu showing "una de ej."; "Calidades:" with a dropdown menu showing "Una de ej." and "Otra de ej."; and "Etiquetas:" with a text box containing "Ej: tag1, tag2". At the bottom of the form are two buttons: "Guardar" and "Eliminar".

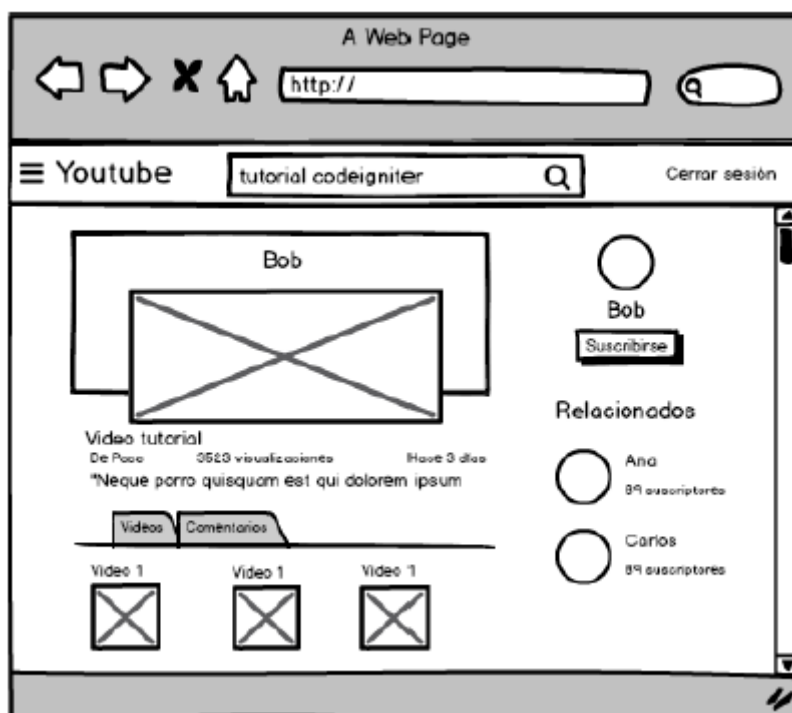
Pantalla mi canal

Si hemos iniciado sesión podremos ver la pantalla mi canal, desde donde vemos nuestra cuenta, los vídeos que hemos subido pudiendo eliminarlos o acceder a editarlos, etc. También podemos acceder a subir vídeo o añadir usuarios relacionados con nuestra cuenta. También podemos ver los comentarios que hagan en nuestro canal y comentar nosotros mismos, o eliminar los comentarios que hagamos.



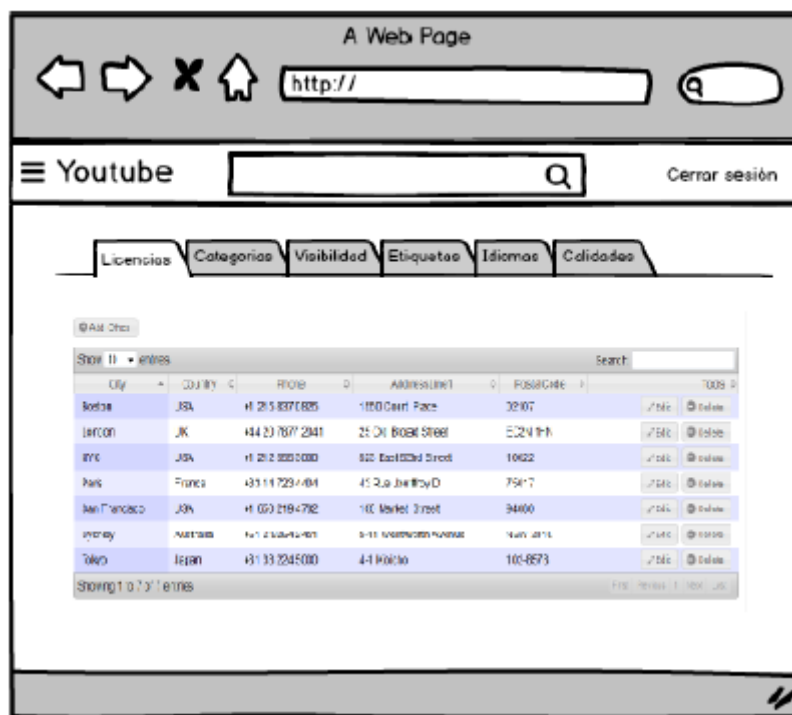
Pantalla canal

Habiendo iniciado sesión o no, podemos ver el canal de otros usuarios. En el canal de un usuario podemos ver sus vídeos subidos, su último vídeo subido como principal, sus usuarios relacionados, etc. Podemos comentar dicho canal (o eliminar los comentarios que hayamos hecho) solo si hemos iniciado sesión.



Pantalla backoffice

Si hemos iniciado sesión y nuestra cuenta de usuario es del tipo administrador, podemos acceder al CRUD del backoffice, hecho con Grocery CRUD. Podemos ver las distintas tablas en cada pestaña, ver filas en detalle, editar una fila o eliminarla, hacer búsquedas en una tabla por filtrado, ordenar las filas según una columna (de forma ascendente o descendente). También podemos exportar la tabla en un determinado formato u obtener una vista previa a imprimir desde el navegador.



Nota: Desde un principio tuvimos claro que haríamos el CRUD del backoffice con grocery CRUD. Como sabíamos también el tema a utilizar (datatables) escogimos para este mockup un ejemplo de la interfaz que grocery genera de forma automática y la insertamos en la pantalla.

Pantalla login

Es el formulario desde el que podemos acceder para identificarnos con la cuenta que tengamos en la web. También hay un enlace al registro.

A Web Page

http://

Youtube

Iniciar sesión | Registro

Iniciar sesión

Nombre de usuario:

Contraseña:

Iniciar sesión

Si no tienes cuenta, ¡regístrate aquí!

Pantalla registro

Formulario desde el que podemos crear una nueva cuenta de usuario con la que poder identificarnos después en la pantalla de iniciar sesión. Hay un enlace a la pantalla de iniciar sesión también.

A Web Page

http://

Youtube

Iniciar sesión | Registro

Registro

Nombre de usuario:

Email:

Contraseña:

Contraseña:

Regístrate

Si ya tienes cuenta, ¡inicia sesión aquí!

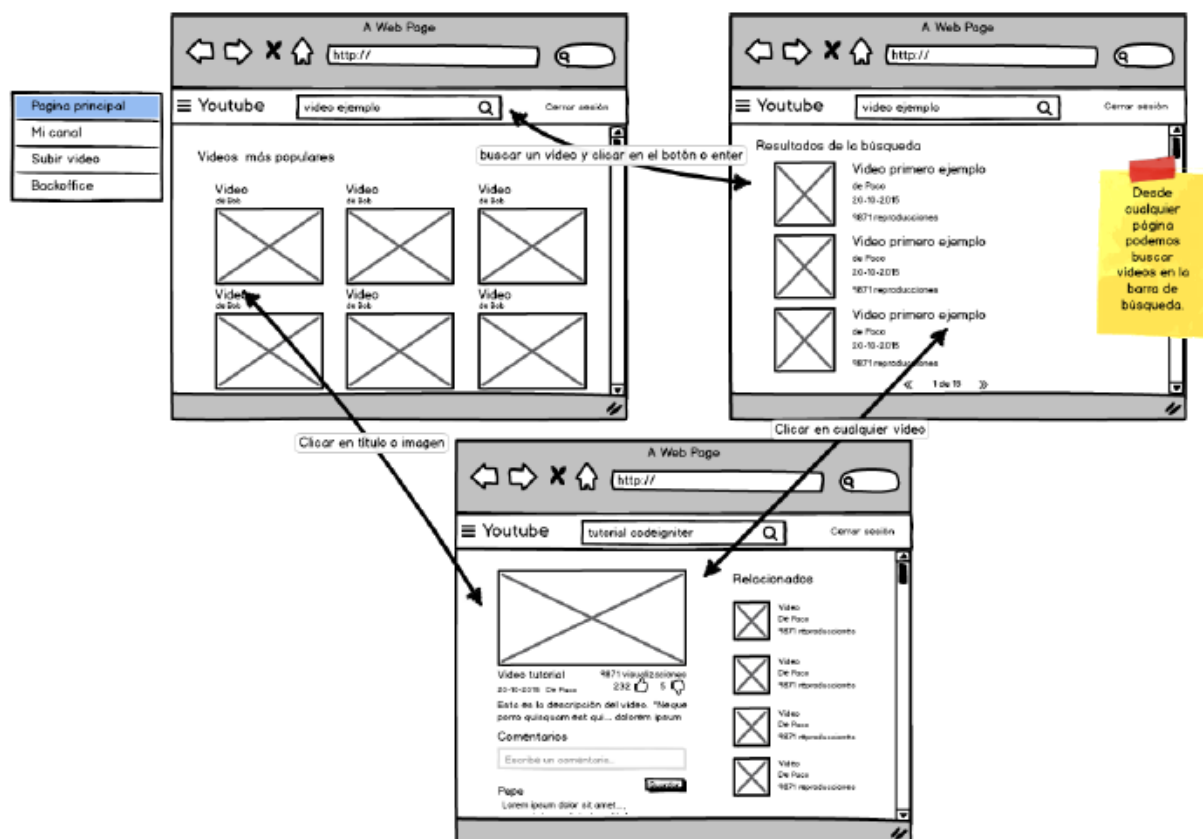
Nota: Podemos ver que la cabecera cambia según hayamos iniciado sesión o no. Si no, aparecen los links a iniciar sesión y registro. Si hemos iniciado sesión aparece la opción de cerrar sesión. Además, si somos un usuario del tipo administrador, en el menú de la cabecera se incluye el link a "backoffice".

Wireframe

En este apartado explicaremos sin extendernos demasiado, cómo planeamos que sea el esquema de navegación; es decir, la relación entre las distintas pantallas.

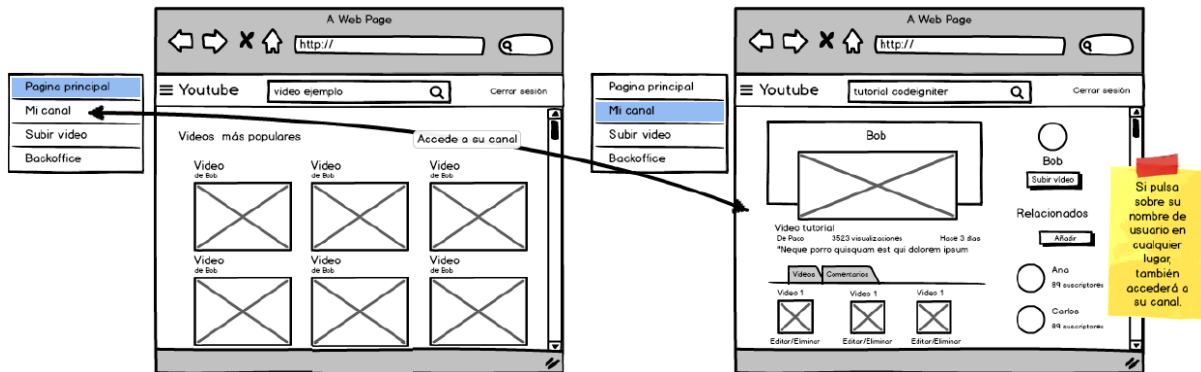
Búsqueda de vídeos

En todas las pantallas de la web aparece la cabecera con la barra de búsqueda. En cualquier momento un usuario puede escribir en dicha barra y clicar en el botón con el icono de la lupa o pulsar enter en el teclado y accederá a la pantalla donde se le mostrarán los resultados de su búsqueda. Estos resultados se refieren a vídeos subidos en la web por los distintos usuarios.



Ver mi canal

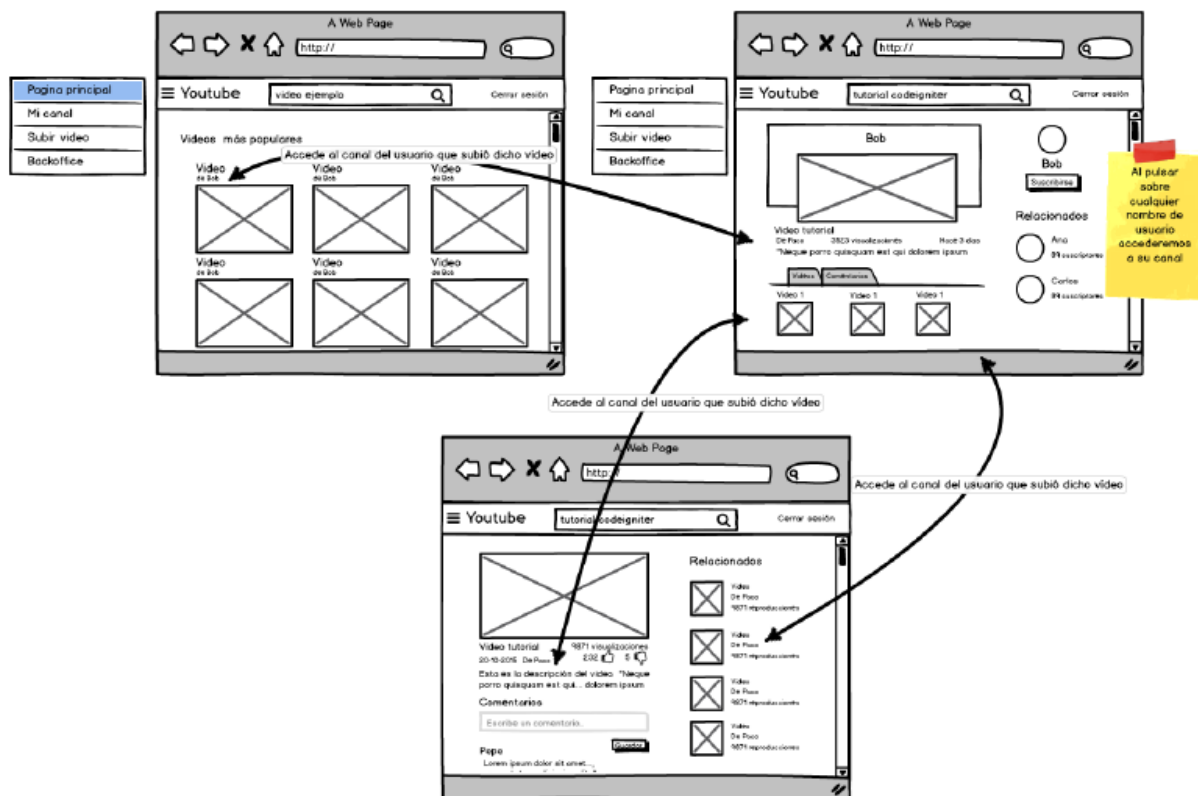
Si hemos iniciado sesión, en el menú de la cabecera podremos acceder a nuestro canal. También si clicamos en nuestro nombre de usuario en cualquiera de las pantallas de la web.



Ver canal usuario

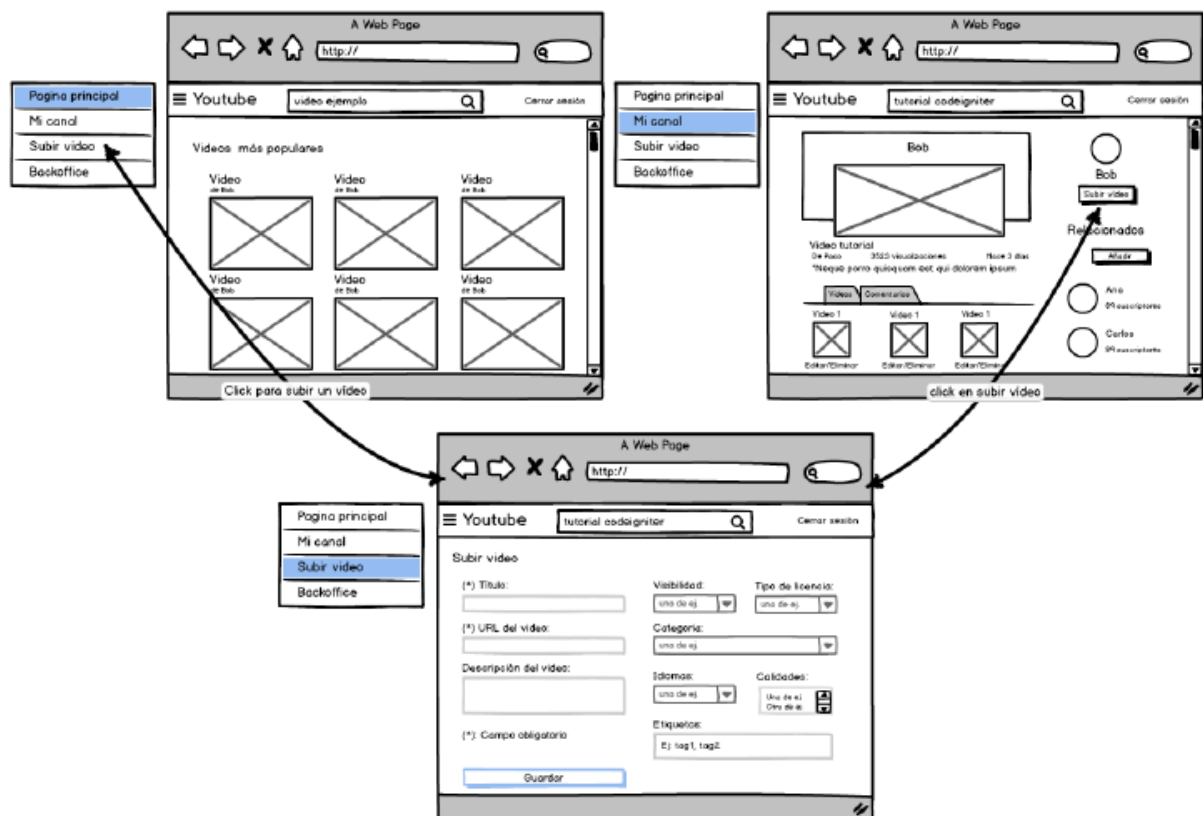
Al clicar en un nombre de usuario, ya sea en uno que aparezca asociado a un vídeo en la pantalla principal (imagen superior izquierda), o en un vídeo en detalle que estemos viendo, o un nombre que aparezca junto a su vídeo en una lista de vídeos relacionados, etc (imagen inferior) accederemos al canal de dicho usuario (imagen superior derecha). En cualquier parte, como ya hemos dicho, que aparezca el nombre de un usuario y pulsemos en él.

Nota: Si accedemos al canal de un usuario el cual somos nosotros estando logeados, aparecerá la pantalla de mi canal. Las diferencias son que en la pantalla de mi canal puedo editar mis vídeos, eliminarlos, subir un nuevo vídeo, añadir un canal relacionado, etc.



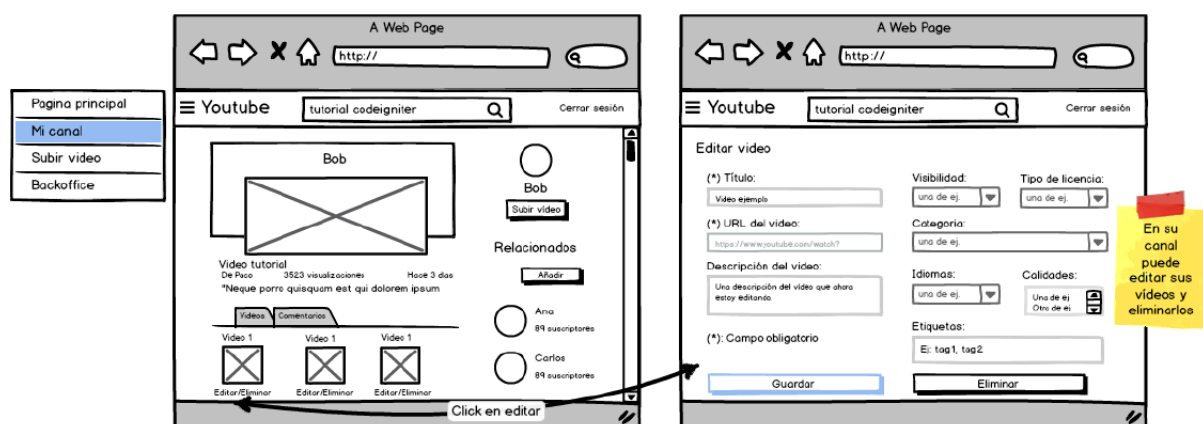
Para subir vídeos

Si hemos iniciado sesión podremos acceder al formulario de subir un vídeo clicando en la opción del menú llamada "Subir vídeo". También podremos acceder desde el botón "subir vídeo" que aparece en la pantalla de nuestro canal. Si no hemos iniciado sesión, nos enviará a la pantalla de login.



Para editar mi vídeo

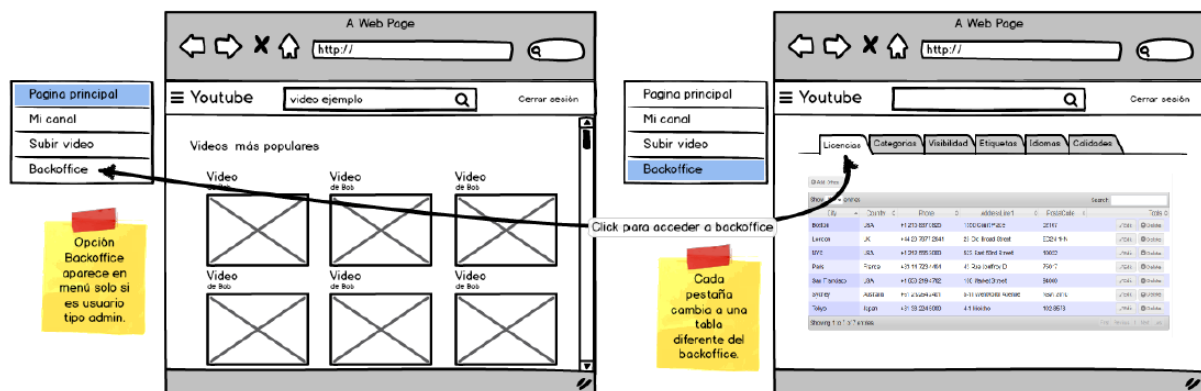
En la pantalla de nuestro canal podemos ver todos los vídeos que hemos subido a la web. Sobre cada uno de ellos podemos eliminarlo o editarlo. Si pulsamos en editar accedemos a un formulario para poder realizar esta acción. Cabe señalar que desde el formulario de edición de uno de nuestros vídeos también podemos eliminarlo.



Pantalla backoffice

Si un usuario está logeado y es un usuario del tipo “administrador” podrá ver un link a backoffice desde el menú de la cabecera. Solo los usuarios que son administradores pueden

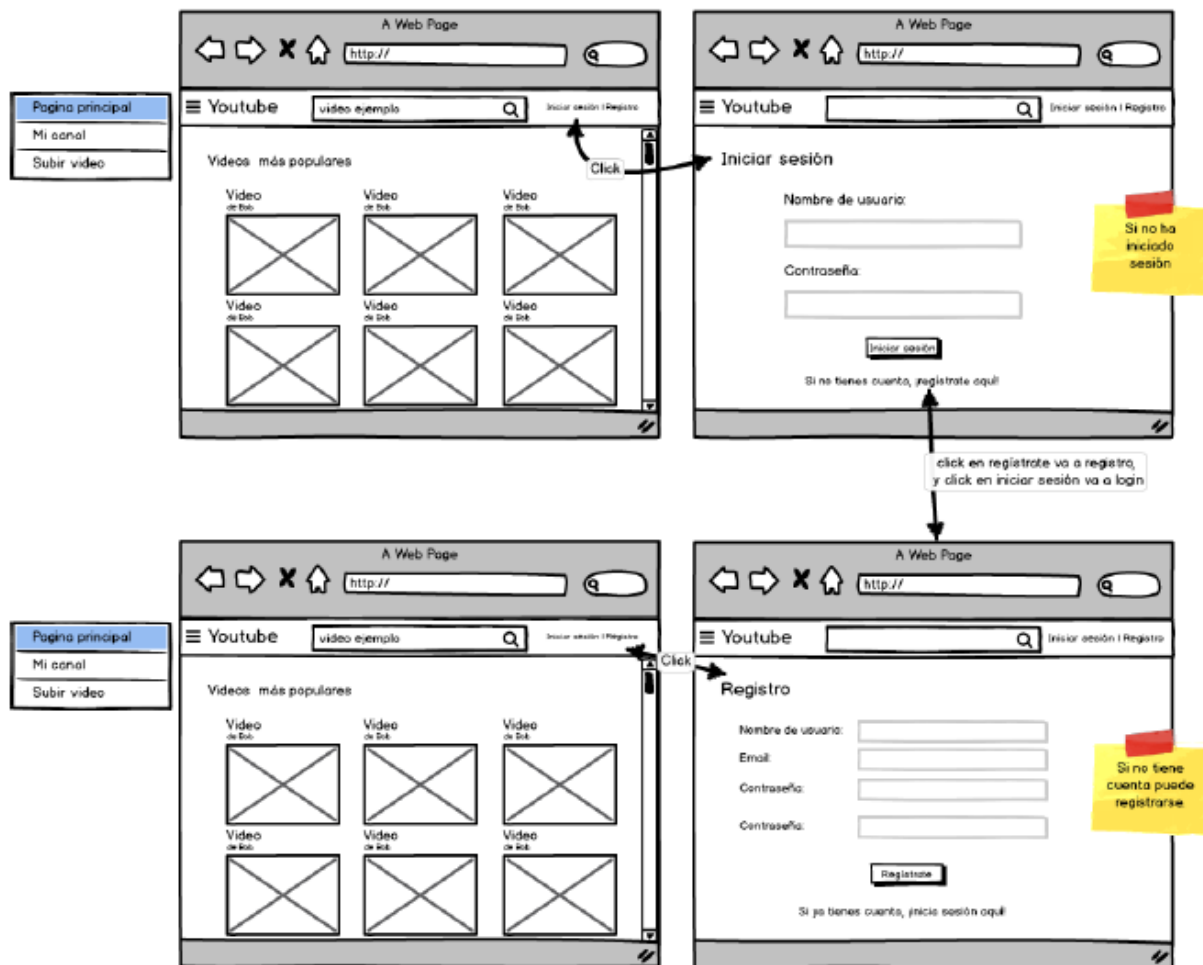
acceder al CRUD del backoffice. Una vez acceda, puede ver las tablas del backoffice, una en cada pestaña de la página que se presenta. En cada tabla puede realizar búsquedas con filtrado, ordenar las filas según una columna de forma ascendente o descendente, ver en detalle los datos de una fila, editarlos, o eliminar dicha fila. También exportar la tabla en cierto formato o obtener una vista previa para luego imprimirla. Si el usuario no es del tipo administrador e intenta iniciar sesión, se le redirige a la pantalla principal.



Iniciar sesión y registro

Cuando un usuario no ha iniciado sesión en la cabecera aparecen las opciones “iniciar sesión” y “registro”. Si clicla en iniciar sesión accederá al formulario para poder logearse en la web. Sii clicla en registro aparecerá el formulario de registro, con el que, si los datos son correctos, se creará su nueva cuenta (un nuevo usuario) y se le redireccionará a la pantalla de iniciar sesión.

Por último, cabe señalar que desde la pantalla de inicio de sesión puede acceder desde un link al registro. Y viceversa en la pantalla de registro, donde aparece un enlace que llevaría al usuario a la pantalla de iniciar sesión.



Patrones utilizados y justificación

Una vez explicados y mostrados los mockups y wireframes, explicaremos en qué patrones nos hemos basado con su correspondiente definición y justificación.

Patrones de pantalla principal

Desde la página principal podemos acceder a cualquier funcionalidad desde el menú que aparece en la cabecera. En ella aparecen los vídeos más populares subidos por los distintos usuarios. Como se puede observar no está ni demasiado llena ni demasiado vacía.

En esta pantalla hemos utilizado los siguientes patrones:

Bandeja de entrada

Mostramos los elementos más interesantes en un listado, es decir, los videos más visitados de la página.

Panel de control

Disponemos de una cabecera, la cual está incluida en todas las pantallas de la página, desde la cual podemos acceder a todas las funcionalidades de la página. En ella hay un menú en la esquina superior izquierda desde donde podemos ir a la página principal, subir un vídeo, ver nuestro canal y ver el backoffice (según los permisos de nuestro tipo de usuario y si hemos iniciado sesión claro). En la cabecera también hay una barra de búsqueda para la búsqueda de vídeos y finalmente, desde esta misma cabecera podemos acceder a iniciar sesión, al registro; o a cerrar sesión si el usuario ya ha iniciado sesión con una cuenta.

Por ello, como hemos visto, desde la cabecera podemos acceder a todas las funcionalidades de la página.

Tabla en blanco

Si la pantalla principal aparece en blanco, la web informa de que es debido a que no hay ningún vídeo subido y que el usuario puede empezar a registrar sus vídeos, los cuales aparecerán en la página principal si son de entre los vídeos más visitados por los usuarios. Se le proporciona un enlace desde el cual puede subir videos, es decir, le enviaría a la pantalla subir vídeo.

Lo mismo ocurre en la pantalla del canal de un usuario que ha iniciado sesión. Si no ha subido ningún vídeo, en su canal se le informa de que es debido a que aún no ha añadido ningún vídeo y que puede empezar dirigiéndose a un enlace que se le proporciona. Este enlace le llevará a la pantalla para subir un vídeo.

Patrones de navegación

En cuanto a la forma en la que se interconectan las distintas partes del sitio hemos utilizado los siguientes patrones:

Menú de primer nivel

Disponemos de un menú principal ubicado en la esquina izquierda de la cabecera, que nos da una visión global del sitio. Este menú nos permite ir a una pantalla u otra rápidamente. Podemos acceder a la pantalla principal, al canal del usuario (si ha iniciado sesión), a subir video, y finalmente al backoffice si se trata de la cuenta de un administrador. Desde la cabecera podemos iniciar sesión, registrarnos o cerrar nuestra sesión ya iniciada. Además, en cualquier momento podemos realizar una búsqueda de vídeos desde el buscador de la cabecera de la web.

El icono del menú de hamburger indica claramente que es un menú desplegable. Además según la opción elegida, dicha opción será la que se remarque en el menú.

También se incluye un icono junto a cada opción del menú con el fin de enfatizar la sección.

Navegación suplementaria

En la pantalla principal ofrecemos una serie de vídeos que son los más populares, animando así a los usuarios a ver dichos videos.

En la pantalla de canal de un usuario ofrecemos los canales relacionados (otros usuarios) y en la pantalla vídeo los vídeos relacionados del vídeo que se está viendo en detalle.

Patrones de formulario

Utilizamos una serie de formularios en las pantallas iniciar sesión, registro y subir vídeo. Hemos utilizado los siguientes patrones:

Formularios breves

Todos nuestros formularios tratan de ser breves para no hacer desistir al usuario de enviarlo.

En la pantalla de inicio de sesión tan solo pedimos el email y la contraseña, siendo los dos obligatorios.

En la del registro un nombre, un email y una contraseña que se ha de introducir dos veces por motivos de seguridad, todos ellos obligatorios.

Finalmente, en la pantalla de subir video tan solo se deben introducir un título y una URL de forma obligatoria. También se puede añadir una descripción y una serie de etiquetas, aunque no de forma obligatoria. Del mismo modo se puede calificar el vídeo según la visibilidad, el tipo de licencia, la categoría, los idiomas y las calidades del video, si no, se dejarán vacíos o con su valor por defecto. Estos últimos elementos se pueden introducir clicando tan solo en una serie de select para que el usuario no tenga que escribir.

En el formulario de editar vídeo ocurre lo mismo salvo que además la URL no es modificable.

Alineado de las etiquetas de los campos

En cada formulario la etiqueta se sitúa encima del campo y tanto la etiqueta como el campo están bien alineados.

Señalizar campos obligatorios

En cada formulario indicamos los campos obligatorios con el símbolo (*) en rojo. Además hay una leyenda con este símbolo indicando qué significa.

Pistas o sugerencias

Sobre las instrucciones del formato de un campo determinado o qué se está pidiendo en el campo, en cada formulario (excepto en iniciar sesión, por considerar que es demasiado obvio), incluimos un placeholder en cada campo indicando con un ejemplo o explicación que se debe introducir en dicho input.

Por ejemplo en el registro indicamos un ejemplo de cada campo. Además en el campo password se indica el rango de longitud permitida y los requerimientos mínimos (1 mayúscula, 1 minúscula y un número). En el formulario de subir vídeo también indicamos un ejemplo de cada campo que el usuario ha de escribir a mano.

Navegación con el teclado

El usuario se puede mover entre los inputs con el tabulador del teclado. No enfocamos ningún campo de forma automática al entrar en los formularios.

Botones de comando

Los botones para enviar los datos de los formularios iniciar sesión, registro y subir vídeo están remarcados en color azul, no habiendo ningún otro que pueda despistar al usuario. No utilizamos tampoco ningún botón de reset.

Tan solo en el formulario de editar vídeo aparecen dos botones, uno para guardar la edición y otro para eliminar el vídeo. El primero es de un color azul y el otro de un color rojo con sus respectivos mensajes, por lo que quedan suficientemente señalizados.

Mensajes de error

Cuando ocurre un error lo mostramos justo debajo del formulario en una sección de color rojo que se puede apreciar a simple vista. Especificamos el motivo del error para que el usuario entienda que debe modificar o añadir en el formulario, pero sin que estos mensajes sean demasiado largos.

Como ya hemos dicho, el mensaje de error se resalta claramente así como también el campo del error. Este o estos campos aparecen con un borde en rojo.

En caso de error, ningún campo del formulario se borra. Los datos que el usuario introdujo siguen estando, para que el usuario no tenga que volver a escribir u olvide lo que puso.

Patrón de identificación del usuario

Registro de usuarios

En cuanto a este patrón en el formulario de registro controlamos que el nombre de usuario y el email no se repitan con los de cualquier otro usuario ya registrado. También confirmamos la contraseña del usuario, es decir, el usuario ha de escribirla dos veces comprobando que se corresponde.

Log in

La web dispone de la pantalla login con la que el usuario puede iniciar sesión introduciendo su email único e irrepetible pudiendo así acceder a las partes privadas de la web.

Finalmente, cuando hay un error en el inicio de sesión los campos se borran.

Log out

Permitimos que el usuario pueda cerrar la sesión gracias a un botón que aparece en la cabecera (si el usuario está logeado). Este botón y funcionalidad al estar en la cabecera, están disponibles en todo momento.

Cuando el usuario cierra sesión aparece un mensaje que confirma que efectivamente se ha cerrado su sesión.

Patrones de listados

Lista tabular o tablas

En las pantallas del backoffice, gracias a la utilización de Grocery CRUD se muestran una serie de tablas que pueden ordenarse según las columnas de dichas tablas. También hay disponible un filtro en cada una de ellas.

Lista de imágenes o grid

En todos los lugares de la web donde aparecen listados de videos (vídeos más populares de la página principal, en un canal los vídeos del usuario, en una búsqueda los vídeos que coinciden con la búsqueda, los videos relacionados en el detalle de un vídeo...) representamos el vídeo con una imagen (thumbnails). Cuando el usuario clica en la imagen, es entonces cuando accede a la pantalla de ese vídeo en concreto, pudiendo visualizarlo sin problemas.

También representamos con imágenes los pequeños listados de canales relacionados de un canal que se esté viendo en detalle.

En la pantalla de búsqueda de vídeos, los vídeos que aparecen como resultado de una búsqueda aparecen con paginado, ya que pueden ser muchos los resultados encontrados.

Acciones de un listado

En las pantallas del backoffice, las filas de cada una de las tablas tienen 3 botones para ver dicho elemento en concreto, editarlo o eliminarlo.

También se muestran mensajes cuando una acción ha ejecutado correctamente, avisos de si realmente quiere abandonar una página sin guardar los datos y mensajes de error cuando estos ocurren. Cuando hay un error la acción no se llega a ejecutar.

Utilidades en un listado

En estas tablas del backoffice ya mencionado, podemos exportar (en formato CSV) e imprimir (ofreciendo una vista previa) la tabla que se está mostrando en pantalla. Estas acciones se pueden ejecutar clicando en sus respectivos botones, que aparecen en la esquina superior derecha de la tabla (no dentro de ella).

Patrones de búsqueda en el sitio

Búsqueda simple

Empleamos este patrón en cuanto a que la búsqueda la situamos en la cabecera, estando disponible siempre en cada pantalla que se muestra. La búsqueda es sobre vídeos del sitio web y se ordenan según número de visualizaciones.

Se puede buscar clicando en el botón a lado del input de la búsqueda o pulsando enter. Además el input tiene el placeholder “Buscar...”, por lo cual es intuitivo para los usuarios.

Resultados de la búsqueda

En la búsqueda de vídeos ya comentada no diferenciamos entre mayúsculas y minúsculas. Mostramos la cantidad de resultados encontrados.

Cada vídeo que se muestra como resultado de la búsqueda contiene la siguiente información: imagen del vídeo (thumbnail), título del vídeo, usuario que lo subió, número de visualizaciones y la descripción acortada hasta 112 caracteres. Esta información se ve en pequeño para no cargar demasiado la página.

Paginación

En la pantalla de resultados de búsqueda de vídeos mostramos los resultados paginados. Se muestran 20 resultados por página.

Hay botones de acceso a la anterior página a la actual y a su siguiente página. Clicando sobre cualquier número de página (enlaces numéricos) se mostrarán los resultados de esa página en concreto.

Con un color más oscuro resaltamos la página en la que el usuario se encuentra.

Patrones RIA utilizados

En el desarrollo de la web hemos compaginado en ocasiones tecnologías como son html, css3, javascript, DOM, http; es decir, AJAX. En cuanto a los patrones RIA hemos utilizado los siguientes.

Formulario inteligente

En cuanto a este patrón, lo usamos en los formularios cuando un campo es obligatorio y el usuario lo selecciona y acto seguido cambia el foco a otro elemento (onblur). Si el usuario no ha escrito nada en él, el borde de dicho input o campo adquiere un color rojo, como símbolo del error. Con el símbolo “(*)” y su respectiva leyenda, el usuario ya puede prever que el error es debido a que ha de rellenarlo.

Cuando el usuario vuelve a ese campo y empieza a escribir, el campo vuelve a su color original, haciendo ver que el error desaparece.

Autocompletado o sugerencias de campos

A la hora de subir un vídeo o editarlo, aparece en dichos formularios, un campo referido a las etiquetas del vídeo que se quiere añadir o editar. Cuando un usuario se posiciona en este determinado input, le aparecen etiquetas ya existentes pudiendo clicar en ellas sin tener que escribir. Conforme escribe una etiqueta, se van quedando tan solo las alternativas posibles (por ejemplo, si escribo co me seguirá apareciendo comida, pero desaparecerá por ejemplo las etiquetas gatitos, risa, etc).

El usuario puede seleccionar una etiqueta que se muestra como alternativa clicando en ella, con enter o tab, etc.

Consultas dinámicas

En las pantallas referentes al backoffice, cuando un usuario consulta una tabla puede filtrar los resultados u ordenarlos de forma dinámica y sin tener que recargar la página o clicar en un botón de submit.

Animaciones

El menú que se abre desde la cabecera se abre con un efecto animado realizado con CSS y JQuery. También se cierra con el mismo efecto.

Además mostramos los mensajes, como los de correcto inicio de sesión, correcto cierre de sesión, error al comentario video o canal con transiciones CSS.

Diagramas

En este apartado mostraremos una serie de diagramas realizados para el diseño de la web.

Diagrama de casos de uso

Hemos diseñado los casos de uso diferenciando entre tres posibles tipos de usuarios.

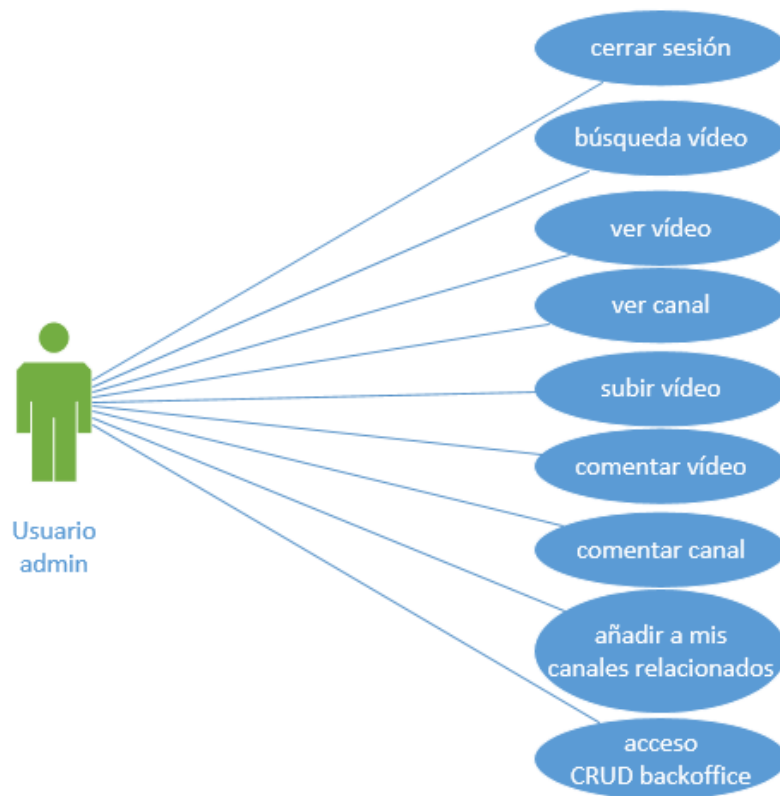
Cuando un usuario entra en la web pero no ha iniciado sesión:



Cuando un usuario entra en la web habiendo iniciado sesión:



Cuando un usuario ha iniciado sesión con una cuenta de administración:



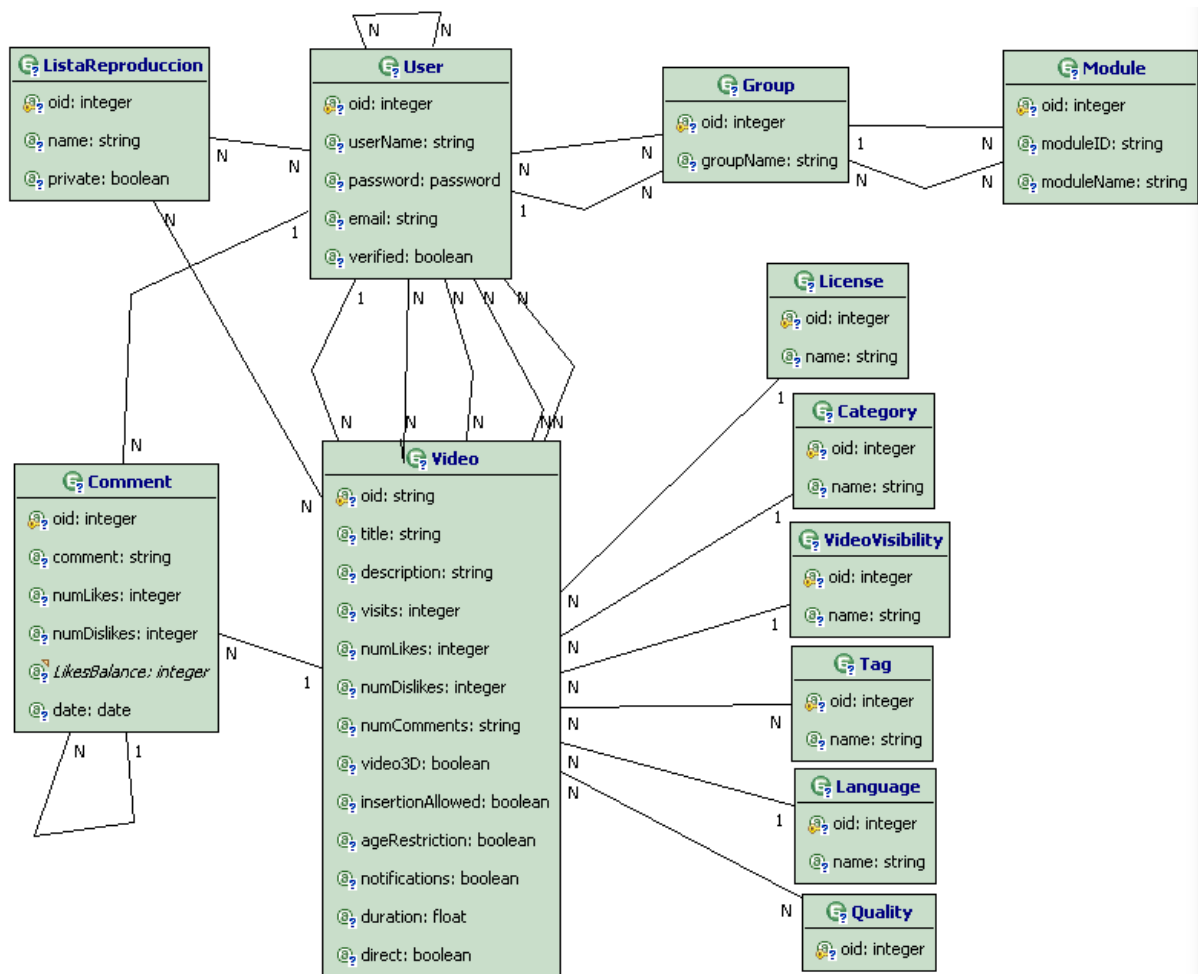
Nota: Un usuario que ha iniciado sesión puede eliminar sus comentarios realizados en un canal y en un vídeo, pero al ser una funcionalidad añadida no se incluyó en el diseño inicial de la web. Por ello no aparece en los casos de uso.

Algunas otras funcionalidades añadidas como “añadir a mis canales relacionados” aparece porque desde un principio ya pensamos en desarrollarla.

Diagrama de datos

Este es el diagrama de datos de nuestra web Youtube. Nos hemos centrado en desarrollar los siguientes aspectos: User, Video, Comment, License, Category, VideoVisibility, Tag, Language y Quality.

Nota: En la BD implementada VideoVisibility pasa a llamarse Visibility, para que la relación entre la tabla vídeo y visibility si pueda llamarse VideoVisibility.



Descripción técnica del sistema

En este apartado hablaremos de las tecnologías, arquitectura, servicios y servidores utilizados.

PHP y CodeIgniter

Para el desarrollo de la web hemos utilizado el lenguaje de programación PHP que permite la creación de webs dinámicas. También hemos utilizado un framework de PHP llamado codeigniter que permite utilizar el patrón MVC (modelo, vista, controlador). Es por ello que nuestra aplicación tiene 3 elementos básicos: los modelos, los controladores y finalmente las vistas.

Por un lado los modelos que hemos implementado, utilizando el lenguaje php, se comunican con una base de datos que tenemos creada con mySQL. También hemos desarrollado una serie de controladores, que son los que se comunican en lenguaje PHP con el modelo que corresponda para pedir datos, realizar inserciones en la BD, borrados, ediciones, etc. Estos controladores también son los responsables de cargar de datos las vistas html, es decir, las pantallas que el usuario ve finalmente. Las vistas además de estar escritas con html incluyen sentencias php para los contenidos dinámicos. Estas sentencias php por ejemplo, imprimen datos en el html, que su controlador correspondiente ya tiene al haber hecho llamadas al modelo.

Por tanto, el flujo normal de la web es el siguiente: El navegador hace una petición a nuestro servidor local con XAMPP, el cual explicaremos más adelante. Según la url de la petición se habrá llamado a un controlador u otro. El controlador llama al modelo si necesita pedirle datos que la vista se encargará de renderizar con PHP. También se realizan peticiones al controlador para insertar datos, editarlos, etc cuando se envía un formulario en el navegador (peticiones post...) y el controlador llama al modelo para realizar esas acciones. Quizás el controlador vuelva a renderizar los datos de la vista de un formulario mostrando errores o haga una redirección en el navegador hacia otra página.

Grocery CRUD

Grocery CRUD es una extensión de codeigniter que hemos usado para las pantallas del backoffice. Gracias a esta extensión hemos podido implementar fácilmente interfaces con las que poder llevar a cabo las operaciones básicas en una base de datos (create, read, update

y delete) de hasta 6 entidades de nuestra base de datos: licencias, categorías, etiquetas, visibilidad, idiomas y calidades.

Esta librería permite además el control y gestión de errores. Cuando una acción se ha ejecutado correctamente también informa al usuario. Cuando se visualiza una determinada tabla, gracias a Grocery podemos filtrar las filas según una cadena que busquemos, ordenar los datos según una determinada columna de la tabla, etc.

Para empezar a utilizar grocery tuvimos que añadir los elementos de la librería como archivos css, js, php... en nuestro proyecto.

Finalmente creamos un controlador que hace uso de esta librería y una vista que renderiza los datos (las tablas, etc) que el controlador le proporciona. La vista también hace peticiones al controlador de forma interna, sin que se programen explícitamente, utilizando la librería de grocery.

Javascript

En las vistas utilizamos código html y php para contenidos dinámicos como ya hemos comentado, pero también utilizamos javascript para el lado del cliente.

En algunos formularios de la web como el registro de usuarios por ejemplo, utilizamos javascript para validar que un campo tenga una determinada longitud, haya mayúsculas y números, etc. También resaltamos los campos por los que se ha dado un error en un formulario utilizando javascript (le añadimos un estilo de un borde de color rojo, etc).

Para el cambio de pestañas en la cabecera del backoffice, con un script de Javascript según la URL en la que se está se selecciona una pestaña u otra. También realizamos redirecciones a otras páginas, añadimos clases CSS a determinados elementos del html, etc.

También lo utilizamos en la cabecera con un menú que se despliega y vuelve a desaparecer con cierto efecto, así como con los mensajes de confirmación de una acción que se ha realizado (mensajes con transiciones CSS).

JQuery

JQuery es una biblioteca de javascript que permite hacer más simple y sencilla la forma en la que interactuamos con los documentos HTML, podemos manipular los elementos del árbol DOM, crear animaciones...

En la web un ejemplo donde utilizamos JQuery es en la paginación de la búsqueda de videos (pantalla búsqueda). También todas las veces que usamos javascript y preferimos obtener un elemento del html de forma más rápida y sencilla. Y hemos llegado a utilizar jQuery para junto con AJAX realizar peticiones a un determinado controlador, lo que permitía hacer comprobaciones previas a la llamada al servidor, lo que se traduce en una liberación de carga del servidor.

HTML5

Como ya hemos explicado anteriormente las vistas están desarrolladas con HTML además de con PHP, Javascript, etc. Es un lenguaje ampliamente extendido de marcas de hipertexto, que utiliza etiquetas para definir sus elementos.

En nuestra web la mayor parte de una vista contiene html para definir listas, cabeceras, campos de formularios, botones, etc. Estos elementos son renderizados por el navegador que utilice el cliente, haciendo que el usuario finalmente vea el aspecto de la página con sus elementos, textos, colores, etc.

Hemos usado además algunos elementos de HTML5 que anteriores versiones de HTML no incluían. Por ejemplo elementos como header, main, footer, nav... que, aun siendo tratados como un simple div, ofrecen una mayor estructuración del documento.

CSS3

CSS son hojas de estilo en cascada, un lenguaje para definir la presentación de un documento HTML o XML. En nuestro caso, lo hemos usado para crear la presentación de las vistas de las que hemos hablado anteriormente que contienen elementos html.

CSS3 a diferencia de las anteriores versiones utiliza módulos y ejecuta algunas transiciones o efectos más rápidamente.

Usamos CSS para cuidar los márgenes de ciertos bloques en una página, para dar color a un input si hay error, para definir determinados formatos de letra, etc. Cada aspecto visual de las pantallas.

También nos hemos ayudado del framework CSS llamado Bootstrap. Con Bootstrap hemos podido dividir las páginas fácilmente en varias columnas de distintos tamaños (grids), que elementos de las pantallas sean responsive, crear formularios con su botón de submit ya con

el estilo que da bootstrap, etc. Así como la utilización de los iconos glyphicon para las opciones del menú por ejemplo.

XAMPP

Como servidor local en los ordenadores de cada uno de los miembros del equipo que hemos desarrollado la web, hemos utilizado XAMPP. Se trata de un servidor que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

En nuestro caso, hemos utilizado xampp para que el navegador pudiera hacer peticiones a nuestro propio ordenador, es decir, nuestro ordenador conseguía hacer el papel de servidor gracias a XAMPP funcionando.

También lo hemos utilizado para la persistencia de la web. Para ello, en XAMPP tenemos activos los servidores Apache (servidor web) y MySQL (servidor de bases de datos). En este último servicio, hemos creado la base de datos de la web con sus distintas tablas, atributos, etc.

Los modelos del programa se comunican con esta base de datos (iw-youtube en nuestro caso), para obtener datos, realizar inserciones, modificaciones o borrados.

Gracias a este programa hemos podido visualizar en todo momento las pantallas de la web, comprobar si las distintas funcionalidades o peticiones al controlador y el modelo funcionaban correctamente, que los datos de la web fueran persistentes alojados en algún sitio para poder ser recuperados después, etc.

Detalles de implementación

A continuación explicaremos por pantallas la implementación de las funcionalidades de la web de forma más detallada.

Pantalla principal

Es la página inicial al entrar en la web. La vista obtiene los datos de los primeros 21 vídeos con más visualizaciones por parte del controlador. El controlador los obtiene gracias al modelo que hace la consulta select que corresponde en la base de datos que tenemos creada.

Cabecera

La cabecera se muestra en todas las pantallas de la web. En ella se pueden buscar vídeos en la barra de búsqueda (funcionalidad que explicaremos más adelante) y tener acceso al menú principal.

Este menú aparece en la esquina superior izquierda de la cabecera. El menú tiene las siguientes opciones: página principal, mi canal y subir vídeo. Con PHP controlamos si un usuario ha iniciado sesión o no. Si no ha iniciado sesión y el cliente selecciona las opciones mi canal o subir vídeo, la web le redireccionará a la pantalla de iniciar sesión. Además también se controla que si un usuario ha iniciado sesión y su cuenta es del tipo administrador, aparezca la opción del menú backoffice.

Por último, si el usuario no ha iniciado sesión en la esquina superior derecha de la cabecera tendrá acceso a links a iniciar sesión y al registro. Si por el contrario, si ha iniciado sesión, aparecerá un link a cerrar sesión.

La cabecera se encuentra en un archivo separado de extensión .php y se incluye en todas las vistas ayudándonos de la función “load” de PHP.

Pantalla de búsqueda

Cuando un usuario escribe algún texto en la barra de búsqueda y pulsa enter o clic en el botón de la lupa que acompaña al input de la búsqueda, se le lleva a esta pantalla.

El controlador obtiene como parámetro el texto que se ha buscado (en la url aparecerá algo como “http://localhost/iw-ci/youtube/index.php/busqueda?search_query=video ejemplo”). El controlador llama entonces a una función del modelo pasándole dicho parámetro. El modelo

entonces busca en la tabla vídeos de la BD que tenga títulos, descripciones, etiquetas o usuarios con textos que contengan la cadena que se le ha pasado por parámetro. Se hace uso del “like=%...%” conocido de MySQL.

Como ya hemos dicho la vista representa todos los resultados encontrados que el controlador le devuelve pues el modelo se los habrá proporcionado, además muestra cuántos resultados se han encontrado en total. Los resultados se encuentran paginados con ayuda de una librería de JQuery.

Pantalla video

La pantalla de video muestra el vídeo cuyo id se le pasa al controlador mediante la url. Este último hará las llamadas correspondientes al modelo para conseguir la información relativa al vídeo (título, url, descripción, usuario que subió el vídeo, visitas, likes/dislikes) así como la lista de comentarios y una selección de vídeos relacionados con el actual.

Los vídeos relacionados se seleccionan mediante un algoritmo que comprueba el usuario que ha subido el vídeo, el título, y las etiquetas que éste contenga, priorizando por la cantidad de visitas que ha recibido el vídeo, ya que así es más probable que sea de interés..

Una vez el usuario está logueado, tendrá la posibilidad de indicar que le gusta o le disgusta un vídeo, así como enviar un nuevo comentario. Si no lo estuviese, la página le indica al usuario que es necesario hacerlo para continuar con esas acciones.

Pantalla subir vídeo

Esta pantalla es un formulario en la que un usuario que ha iniciado sesión utiliza para añadir un vídeo. Si el usuario no está identificado, se le redireccionará a la pantalla de inicio de sesión.

Para algunos de los campos en la vista utilizamos funciones de PHP como form_input, form_textarea, form_submit... y arrays de PHP. Otros simplemente con elementos de html.

Este formulario indica los campos que son obligatorios (título y URL) y si se pasa a otro campo sin haber escrito nada en ellos, el campo aparecerá con un borde rojo indicando el error. Si se envía el formulario con errores (campos obligatorios no rellenos) aparecerá el mensaje de error explicando el motivo. Esto lo hacemos con ayuda de funciones de PHP como form_validation y set_rules para el control de errores. Podemos así con PHP saber cuando un formulario traspasa las reglas que le indicamos.

Si todo va correctamente, la acción del formulario envía una petición http POST al controlador de dicha vista, en concreto a una función que se encarga de guardar el vídeo llamando a una función determinada del modelo. En esta petición POST irán los campos que el usuario ha seleccionado o escrito del vídeo que quiere subir.

Finalmente hacer mención a que si falla la validación los campos del formulario no se vuelven a poner en blanco porque resultaría molesto volver a rellenarlo.

Pantalla editar vídeo

Esta pantalla tiene un formulario en el que se pueden editar los datos de un vídeo subido con la excepción de que la url del vídeo debe seguir igual.

Cabe destacar que a este formulario sólo podrá acceder aquel que haya subido el vídeo a editar, de lo contrario se le redireccionará al inicio.

Obviamente se recogen por defecto los valores que ya tenía el vídeo (se hace una invocación get del modelo) para tener más claro lo que queremos cambiar y para que lo que no se modifique se mantenga en el mismo estado.

Al igual que en el resto de formularios si hay un error al validar, en este caso por ejemplo al dejar el título vacío, no se restaura el form, se deja como estaba al darle a enviar ayudándonos de la sesión.

Cuando todo está correcto se llama a los métodos del modelo para guardar el vídeo con un update, y también para registrar las etiquetas y las calidades del vídeo, que al ser relaciones N:N se tienen que rellenar en tabla aparte para relacionarlos. Aquí tenemos en cuenta también que si al editar se quita por ejemplo una etiqueta del vídeo, hay que borrar la relación de la muchos a muchos (pero no la etiqueta en sí, que puede estar presente en más vídeos).

Pantalla canal

La pantalla del canal muestra la información del usuario del id que se le pasa al controlador mediante la url, que hace las llamadas al modelo para obtener la lista de vídeos, la lista de comentarios al canal y la lista de usuarios relacionados.

En el canal se destaca el último vídeo subido por el usuario y, caso de que el usuario no haya subido aún ninguno, se le sugerirá que lo haga.

Si el usuario que accede al canal está logueado y es el propietario del canal, éste cambiará ligeramente la interfaz, añadiendo un botón para subir vídeos y la posibilidad de editar o borrar

los vídeos. Para esta última acción será necesario un paso de verificación, por motivos de seguridad por parte del usuario, en caso de que se presione el botón sin querer borrarlo realmente.

En la sección de usuarios relacionados el usuario propietario del canal podrá insertar las cuentas de otros usuarios que considere.

Pantalla backoffice

Para esta pantalla tan solo hemos necesitado una vista y un controlador, no ha hecho falta un modelo pues hemos utilizado Grocery CRUD para su implementación.

En la vista aparece una primera tabla y una cabecera de pestañas. Cada pestaña es un link que lleva al usuario a ver una tabla u otra del backoffice (licencias, categorías, visibilidad, etiquetas, idiomas o calidades).

Los datos que renderiza la vista vienen dados por el controlador. El controlador obtiene la librería de Grocery. Además obtiene los datos de la tabla que por defecto muestra, con la función `set_table`. A esta función le indicamos el nombre de la tabla que queremos mostrar. También indicamos el tema de grocery a utilizar con la función `set_theme`, las columnas que son obligatorias introducir al añadir una nueva fila a la tabla, las columnas que se mostrarán de la tabla, etc. En la vista se cargan todos los elementos `.js` para que grocery pueda realizar correctamente las funciones del CRUD, mostrar mensajes, etc. También se cargan los archivos `.css` para la representación visual de la tabla con el tema que se ha escogido, en nuestro caso `datatables`. Además en el controlador añadimos nuestros archivos `.js` y `.css` para la cabecera y las pestañas de las distintas tablas del backoffice.

Cuando se selecciona una pestaña de la pantalla backoffice, se hace una petición al controlador, a una determinada función del controlador, la cual carga los datos de la tabla que se pide, y de nuevo los archivos de grocery y nuestros propios archivos ya comentados (`.js` y `.css`). Entonces el controlador vuelve a cargar la vista, la cual renderiza los datos que se le hayan pasado por parámetro, es decir, la tabla que el controlador ha cargado esta vez, los archivos `.css` y `.js`, la cabecera, etc.

Pantalla login

En esta pantalla se muestra un pequeño formulario para rellenar el email y la contraseña del usuario que quiera iniciar sesión, además también se muestra una sugerencia de registro para quien no tenga cuenta aun en la web..

Al completar el formulario y enviarlo con petición POST se invoca desde el controller a un método del modelo que busca un usuario por su email y contraseña, de modo que si no se encuentra un usuario que coincida con ese mismo email y esa misma contraseña el login es incorrecto y se le informará con una alerta al usuario.

Si el login es correcto se le informará con una alerta de éxito y se le redirecciona a la pantalla de inicio directamente.

Pantalla registro

En esta pantalla se le presenta al usuario un formulario en el que se podrán rellenar los campos con el objetivo de crear una nueva cuenta.

Se controla con validaciones que no se escoja un email ni un “nick” de usuario repetido, además también se valida que se escriba dos veces seguidas la misma contraseña para poder proceder con el registro.

De un modo un poco más avanzado se controlan que los email y password sigan un patrón concreto. En el caso de la contraseña debe contener al menos una mayúscula, una minúscula y un número para que esta sea válida, por motivos de mínima seguridad. Y en el caso del email debe tener caracteres (sean letras, números o algún carácter como guiones o puntos, siempre y cuando el primer carácter sea una letra) en primer lugar, un “@” a continuación, seguidamente solo se pueden introducir de 2 a 6 letras, un punto y otra vez de 2 a 6 letras al acabar tras el punto. Es importante especificar que se permite después del punto volver a introducir 2-6 letras y otro punto más (como los email de la ua, ejemplo: mjvr4@alu.ua.es), pero esto último no es obligatorio, simplemente se permite.

Tras validar que todo es correcto (si algo falla no se limpia el form, se mantienen los campos ayudándonos de la sesión), se hacen las invocaciones al modelo desde el controller de modo que se realice el insert correspondiente para la nueva cuenta de usuario.

Problemas encontrados y su solución

Uno de los problemas que hemos tenido ha sido a la hora de compaginar Javascript con PHP, ya que el primero es del lado del cliente mientras que el otro lo es de parte del servidor. Por lo que era un poco difícil de juntar.

El problema en concreto fue el siguiente. Para mostrar las etiquetas en el selector correspondiente a la hora de subir o editar un vídeo, se utiliza una librería de JS llamada Select2, pero los datos los tenía que recibir del servidor mediante PHP.

- La solución ha sido hacer un “echo” de una variable JS asignándole los datos del array de PHP convertidos a JSON mediante la función `json_encode`. Así, una vez el servidor interpreta el código en PHP para generar el html, se creará la variable en JS que ya podrá leer el navegador y la correspondiente librería.

Otro problema que tuvimos fue cuando quisimos mostrar un mensaje de confirmación en cerrar sesión. El problema fue que dicho mensaje tan solo aparecía en la pantalla de inicio.

- La solución fue darnos cuenta que teníamos que hacer aparecer el mensaje en cada una de las vistas desde donde se podía acceder a cerrar sesión, es decir, en todas ya que cerrar sesión está en la cabecera la cual se incluye en todas las vistas. Para evitar duplicación de código dejamos todo el código necesario para hacer aparecer dicho mensaje en un archivo “`cerrarsesion.php`” y lo incluimos en la cabecera con la ayuda de la función de PHP llamada “`$this->load->view('...')`”. Como la cabecera está en todas las vistas donde puede aparecer la funcionalidad y mensaje “cerrar sesión”, va perfectamente.

Cuando quisimos pasar parámetros a la función `index` de un controlador vimos que no era posible de la manera en la que lo planteamos.

- Finalmente, en el controlador creamos funciones (a parte del `index`) que tomaban el parámetro que queríamos pasar para hacer una determinada función. Ejemplos en los que hemos necesitado aplicar este método son: `url-web/canal/ver/:id` y `url-web/video/watch/:id`, es decir, en ver un determinado canal o vídeo.

A la hora de implementar el CRUD del backoffice tuvimos un problema a la hora de obtener las tablas de la BD. Un miembro del equipo implementó la funcionalidad e hizo un pull request en el repositorio compartido Bitbucket. Cuando el git master del equipo (usamos parte de la metodología de gitFlow) comprobó la funcionalidad implementada tuvo errores del tipo “grocery no encuentra la tabla especificada”.

- Resultó ser que por algún motivo (quizás por los distintos sistemas operativos) un miembro del equipo creaba las tablas con las iniciales de los nombres de las tablas en mayúsculas. Los otros dos miembros hacían lo mismo copiando el script de sql en phpmyadmin pero el nombre de las tablas seguían todos en minúsculas, aunque las modificaran con un alter table. La solución fue que el miembro del equipo al que le salían en mayúsculas las cambiara a minúsculas. Por convenio a partir de ese momento trabajaríamos tan solo en la BD con nombres en minúsculas.

Mejoras y ampliaciones

Estas son las ampliaciones en cuanto a funcionalidades que hemos realizado en la web y también mejoras visuales o de otra índole.

Ampliaciones pantalla canal

Referente a los vídeos del usuario que está viendo su canal, dicho usuario puede editar los vídeos así como eliminarlos. También puede añadir usuarios relacionados a su canal.

Ampliaciones pantalla canal y video

Cuando un usuario comenta un canal o un vídeo, este también puede borrar los comentarios que haya hecho.

Ampliaciones vídeo

Cuando un usuario accede a un vídeo, el número de sus visualizaciones se incrementa en 1. Así cualquier usuario puede ver el número de reproducciones de los distintos vídeos de la web.

Además los usuarios pueden dar a me gusta o no me gusta en un determinado vídeo, una única vez. Si el usuario ya dio su “voto” y vuelve a seleccionar la misma opción, el voto será eliminado del vídeo, mientras que si se pulsa la opción contraria, ésta será ahora la opción del usuario. Con estas funcionalidades los usuarios pueden ver el número de “me gusta” y “no me gusta” de los diferentes vídeos.

Los usuarios también pueden ver de un vídeo en detalle, sus vídeos relacionados.

Mejora en subir/editar vídeo

A la hora de escribir las etiquetas de un vídeo que se está añadiendo o editando, cuando el usuario separa la etiqueta con una coma (tal y como se le indica en el ejemplo del placeholder del input) la tarjeta pasa a convertirse en un icono con una cruz. Al clicar en esta cruz se elimina dicha etiqueta fácilmente. Por otro lado, gracias a que el texto se convierte en una etiqueta visual al escribir la coma, ya no habrá posibilidad de que un usuario escriba etiquetas

con espacios después de la coma, etc. Además, visualmente, es mucho más cómodo e intuitivo para el usuario.

Las etiquetas pueden seleccionarse desde un desplegable con todas las disponibles o bien crear nuevas, que serán añadidas para que otros usuarios puedan utilizarlas.

Referencias

Estas son las referencias utilizadas para el desarrollo del informe, y el desarrollo de las funcionalidades de la web. También se incluyen las referencias de herramientas que nos han ayudado a la realización de la práctica.

Informe

Transparencias Parte 2, Tema 2 Patrones y Patrones RIA.

<https://es.wikipedia.org/wiki/JavaScript>

<https://es.wikipedia.org/wiki/Html5>

<https://es.wikipedia.org/wiki/Jquery>

<https://es.wikipedia.org/wiki/PHP>

<http://www.desarrolloweb.com/articulos/codeigniter.html>

<https://es.wikipedia.org/wiki/XAMPP>

Funcionalidades y diseño

<https://balsamiq.com/products/mockups/>

<http://www.desarrolloweb.com/manuales/manual-codeigniter.html>

<http://www.grocerycrud.com/documentation>

<https://secure.php.net/manual/es/index.php>

<http://getbootstrap.com/components/>

http://www.w3schools.com/html/html5_intro.asp

<http://api.jquery.com/>

<https://select2.github.io/examples.html#tags>

<http://web.enavu.com/js/jquery/jpaginate-jquery-pagination-system-plugin/>

Otras herramientas de trabajo

<https://trello.com>

<https://bitbucket.org/>

<https://es.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>