TFG



PRODUCTORA AUDIOVISUAL

Patricia Villanueva Polo

Índice

[Índice de Ilustraciones 3](#_Toc71484869)

[1. Glosario de términos 4](#_Toc71484870)

[2. Introducción 5](#_Toc71484871)

[3. Módulos formativos aplicados 6](#_Toc71484872)

[4. Objetivos del proyecto 7](#_Toc71484873)

[5. Tecnologías y herramientas empleadas. 8](#_Toc71484874)

[6. Fases del Proyecto 9](#_Toc71484875)

[ Modelo de datos utilizado 9](#_Toc71484876)

[ Diagrama de clases y casos de uso 10](#_Toc71484877)

[ La Web “Neveritas Producciones” 10](#_Toc71484878)

[ El panel de Administración 15](#_Toc71484879)

[ Xampp y el Despliegue de la aplicación 25](#_Toc71484880)

[7. Conclusiones y mejoras del proyecto 29](#_Toc71484881)

[8. Bibliografía 30](#_Toc71484882)

[ Sobre javax.mail: 30](#_Toc71484883)

[ Recursos: 30](#_Toc71484884)

[9. Anexos (optativos) 31](#_Toc71484885)

[ Scripts de cargas de BBDD 31](#_Toc71484886)

[10. Cuenta Github y enlace al TFG 35](#_Toc71484887)

# Índice de Ilustraciones

[1 Modelo de datos 9](#_Toc71484794)

[2. Casos de Uso de Visitantes 10](#_Toc71484795)

[3 Casos de Uso Administrador 10](#_Toc71484796)

[4 Wireframes web y administración 11](#_Toc71484797)

[5 Pantone y Logo 12](#_Toc71484798)

[6 Slide de la web 12](#_Toc71484799)

[7 Flecha navegación arriba 13](#_Toc71484800)

[8 Menú galería de trabajos 13](#_Toc71484801)

[9 Ejemplo reproducción videos 13](#_Toc71484802)

[10 Muestra llamadas a archivos externos desde el HTML 14](#_Toc71484803)

[11 iconos redes sociales 14](#_Toc71484804)

[12 iconos para mostrar los servicios 14](#_Toc71484805)

[13 Ejemplos Pantone cliente 15](#_Toc71484806)

[14 Enlace Administrar 15](#_Toc71484807)

[15 Página Login 16](#_Toc71484808)

[16 Error en el Login 16](#_Toc71484809)

[17 Página de Bienvenida 17](#_Toc71484810)

[18 Objeto Usuario 17](#_Toc71484811)

[19 Objeto Video 18](#_Toc71484812)

[20 Objeto Empresa 18](#_Toc71484813)

[21 Objeto Conexion 19](#_Toc71484814)

[22 Objeto SendEmail 19](#_Toc71484815)

[23 Interfaz Usuario 19](#_Toc71484816)

[24 Interfaz Video 20](#_Toc71484817)

[25 Interfaz Empresa 20](#_Toc71484818)

[26 Enlace salir 20](#_Toc71484819)

[27 Formulario de contacto 21](#_Toc71484820)

[28 Servlet Administración 21](#_Toc71484821)

[29 Añadir Galeria 22](#_Toc71484822)

[30 Añadir Empresa 22](#_Toc71484823)

[31 Eliminar Galeria 23](#_Toc71484824)

[32 Eliminar Empresa 23](#_Toc71484825)

[33 PopUp Aviso Legal 24](#_Toc71484826)

# Glosario de términos

**Reel:** Es un currículo, portfolio o presentación de los mejores trabajos que se han realizado en formato vídeo. Una muestra, en pequeños fragmentos, de todos los trabajos realizados por: directores de fotografía, directores de cine, animadores y todo aquel que esté involucrado en el mundo audiovisual.

# Introducción

El TFG consistirá en el desarrollo de la web para la productora audiovisual Neveritas Producciones, junto con la creación de un panel de administración de contenidos.

Tanto la web como la aplicación estarán desplegadas en un servidor por lo que resultarán accesibles desde cualquier equipo con conexión a internet sin grandes requerimientos o recursos para ello. El panel de administrador será accesible desde la propia web, pero únicamente a aquellas personas que posean un usuario y contraseña válidos.

# Módulos formativos aplicados

**Lenguaje de Marcas**: para la realización de la página web, usaremos HTML5 y CSS3.

**Programación y Entornos de Desarrollo**: Java y JavaScript

**Base de datos**: para el mantenimiento de la web

**Desarrollo de interfaces**: Diseñar el wireframe y el tratamiento de fuentes, colores y toda la experiencia visual del usuario.

**Despliegue de aplicaciones web y Sistemas informáticos**: Para montar la red local y desplegar la aplicación java y la web.

# Objetivos del proyecto

Neveritas Producciones es una productora audiovisual granadina cuyo catálogo de servicios cubre aspectos muy amplios del mundo audiovisual. Dados los problemas que debe afrontar en la situación actual debida al COVID-19, su propietaria ha solicitado realizar un nuevo diseño para su página web, con la intención de dar a conocer su trabajo y poder darse a conocer digitalmente a nuevos clientes.

Esta web estará enfocada, mayoritariamente, en proporcionar información a los usuarios. Es por ello que se plantea incluir los siguientes epígrafes o puntos en el menú:

* La productora
* Galería de trabajos
* Servicios que ofertamos
* Nuestros Clientes

Además, en la cabecera de la web, bien visibles, aparecerán el teléfono de contacto, redes sociales y logotipo. A continuación, una slider con diferentes imágenes de la productora en plena realización de su trabajo, a modo de presentación de la web. Por último, en el pie se incluirá un formulario de contacto y los datos empresariales.

Puesto que la propietaria de la productora necesita poder actualizar de manera autónoma los contenidos de su web. Se realizará una aplicación para que pueda añadir nuevas imágenes y detalles a la galería y ampliar los logotipos de los clientes con los que trabaja.

# Tecnologías y herramientas empleadas.

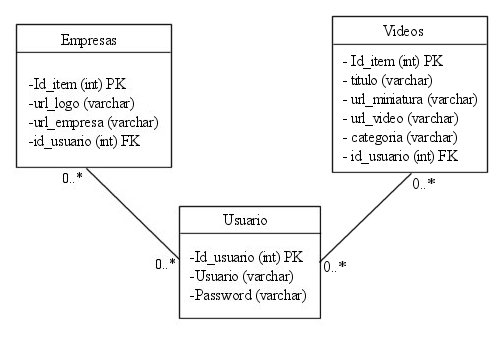
Las Tecnologías necesarias para la implementación de la web y la aplicación que permitirá su mantenimiento serán las siguientes:

* Visual Studio code: Para el desarrollo de HTML5, CSS3 y JavaScript
* Eclipse: Para el desarrollo de Java
* Photoshop: Para los diseños que requerirá la web, como el logo, iconos, imágenes y similares. Y también para cualquier documento visual que requiera el trabajo (wireframes, mockups, modelo de datos…etc).
* Los vídeos estarán alojados en la cuenta de Youtube de la productora y serán proporcionados por la misma.
* Xampp para realizar el despliegue de la aplicación en local (la cual incluye, apache, tomcat, mysql, que serán necesarios para realizar el despliegue).
* WireframeSketcher: Para el diseño del wireframe del anteproyecto.

# Fases del Proyecto

## Modelo de datos utilizado

El modelo de datos necesario para la ejecución del proyecto es bastante sencillo, como se puede apreciar en la siguiente imagen:



1 Modelo de datos

Consiste en tres tablas:

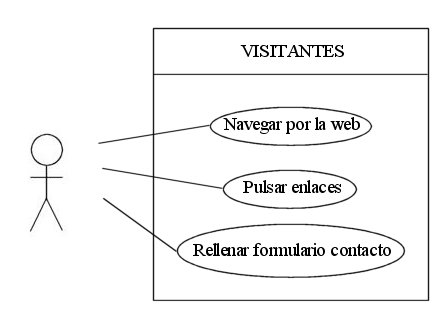
* **Usuarios**: Esta tabla contiene la información necesaria para logarse en el panel de Administración de la web. El usuario, la contraseña y un identificador único que consistirá en un número entero y auto incremental. En inicio, usuarios sólo contendrá los datos de la propietaria del sitio web.

Previendo que, a futuro, pueda ser necesario añadir nuevos usuarios y darles acceso solo a una parte de los contenidos, el *id\_usuario*, será clave foránea en el resto de tablas del modelo del dato.

* **Empresas**: En la tabla empresas almacenaremos la información necesaria para mostrar los logotipos en la web. La url dónde está subida la imagen del mismo, la url de la empresa a la que hace referencia, un identificador único y auto incremental (id\_item). Y la clave foránea que relaciona esta tabla con los usuarios.
* **Videos**: También incluye la información necesaria para construir la galería dinámica de trabajos que se muestra en la web. El título para poder identificarlo rápidamente, la url dónde se encuentra alojado (normalmente youtube), la url de la miniatura que se mostrará en la web y la categoría a la que pertenece y que será la que de dinamismo a la galería. Por último, el identificador único, también auto incremental y la clave foránea para relacionar estos ítems con los usuarios.

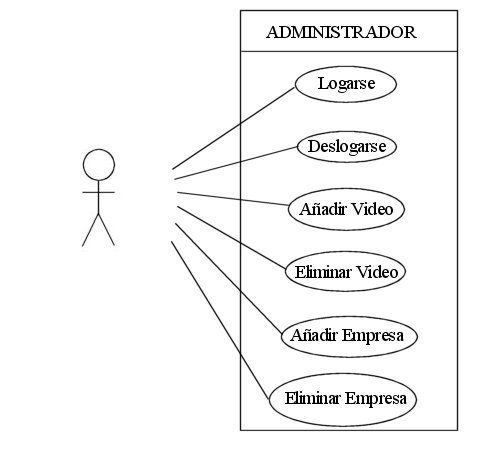
## Diagrama de clases y casos de uso

El visitante de la web puede realizar una serie de acciones muy limitadas, y tan solo en la página principal:



2. Casos de Uso de Visitantes

Mientras que la mayor parte de la acción corre a cargo del administrador o usuario del panel de administración:



3 Casos de Uso Administrador

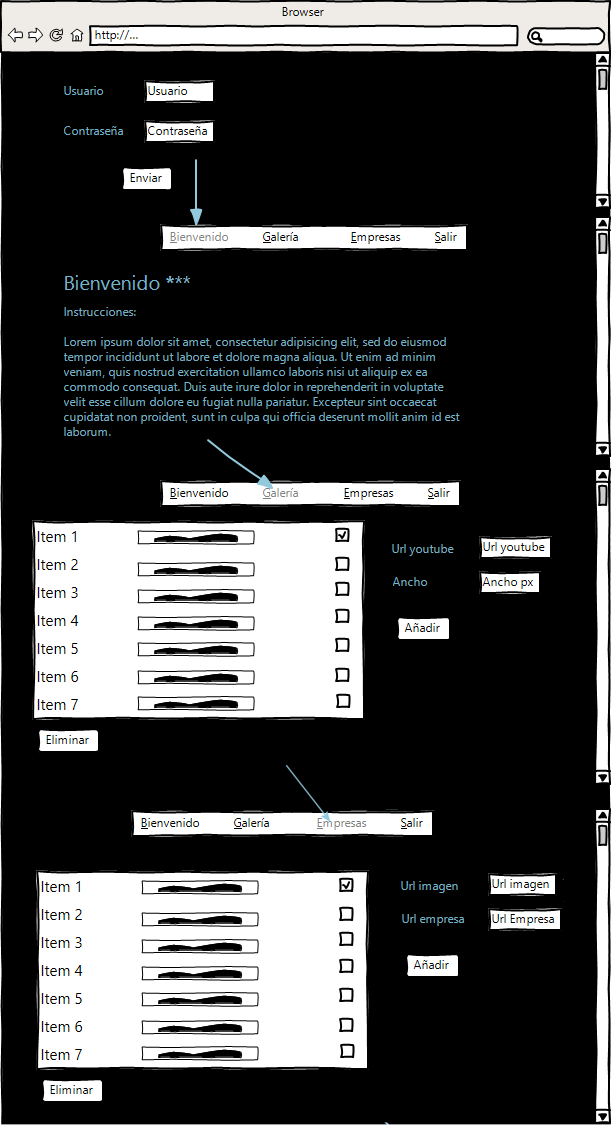
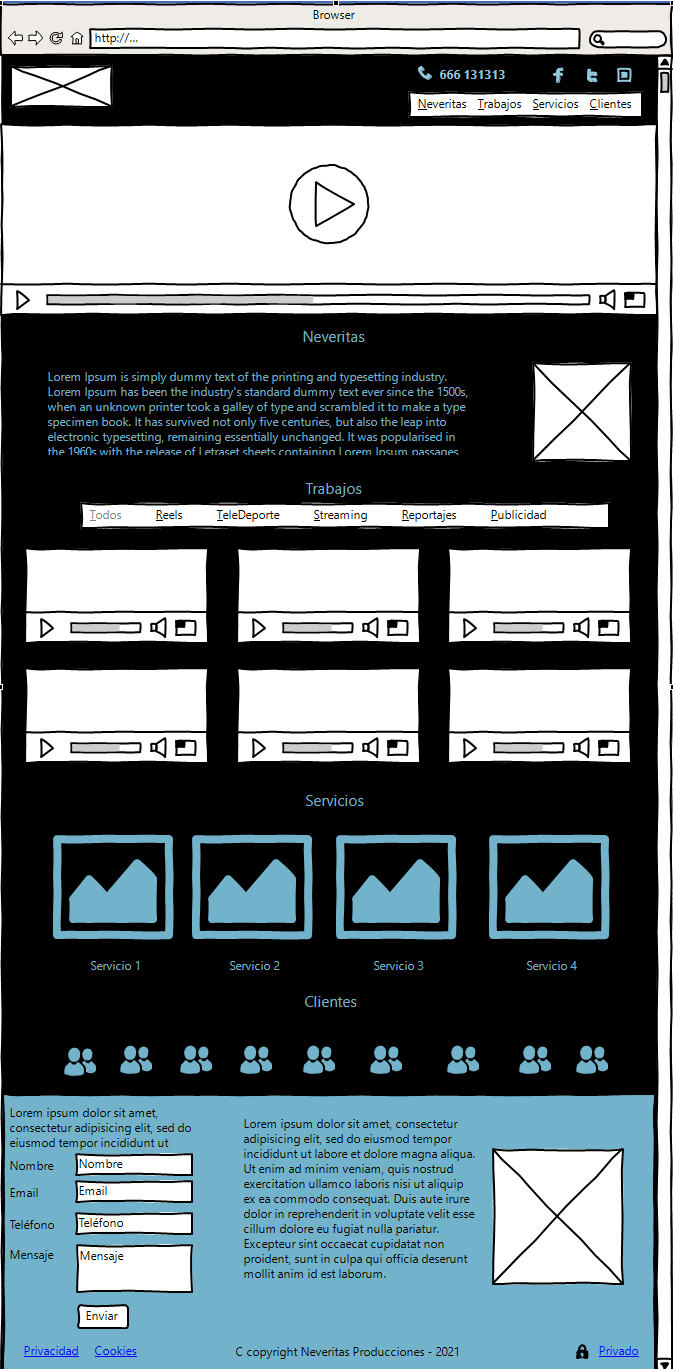
## La Web “Neveritas Producciones”

Este TFG consta de dos partes bien diferenciadas. De un lado la web que será visible por el público general y proporcionará toda la información necesaria sobre el trabajo de la productora y los métodos de contacto. De otro, el panel de administración que permitirá a la propietaria de la web, realizar modificaciones en su contenido de forma autónoma, cuando lo crea necesario y de forma muy sencilla.

Para llevar a cabo este proyecto, se comenzó llevando a cabo una entrevista con la productora para recabar información sobre qué querían conseguir con la web, qué deseaban mostrar y qué servicios ofrecían.

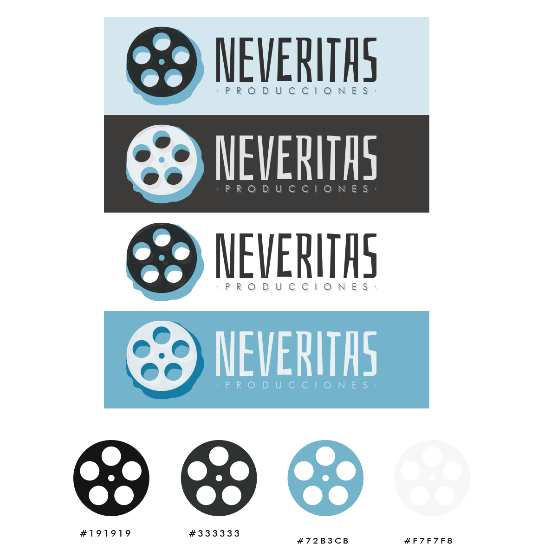
Esto llevó a indagar sobre el tipo de diseños más atractivos en este tipo de empresas, y hablar de qué es lo que buscan los posibles clientes, lo cual daba una idea de qué tipo de diseño y contenidos debían mostrarse.

Una vez realizada la captura de ideas, el siguiente paso era desarrollar un wireframe y presentarlo al cliente:



4 Wireframes web y administración

La productora ya tenía una marca creada que no deseaba modificar y un Pantone con una serie de colores predeterminados que se emplearían para el diseño web, mantenimiento de este modo la imagen que ya era conocida por varios de sus clientes y que deseaban expandir.



5 Pantone y Logo

Con el diseño aprobado, el siguiente paso consistió en la maquetación y desarrollo del código que debería dar vida a la web.

Se empleó Visual Studio Code para realizar una primera maquetación, desarrollando el código HTML, CSS y JavaScript necesario para su visualización y usabilidad.

En JavaScript se diseñaron ciertos recursos dinámicos como el movimiento del slide, atendiendo a una superposición de dos imágenes cambiantes gracias a un efecto de opacidad. Dichas imágenes se eligen desde una matriz o array, en el que se vuelcan las urls relativas de todas las fotos que deseen utilizarse para el visionado.



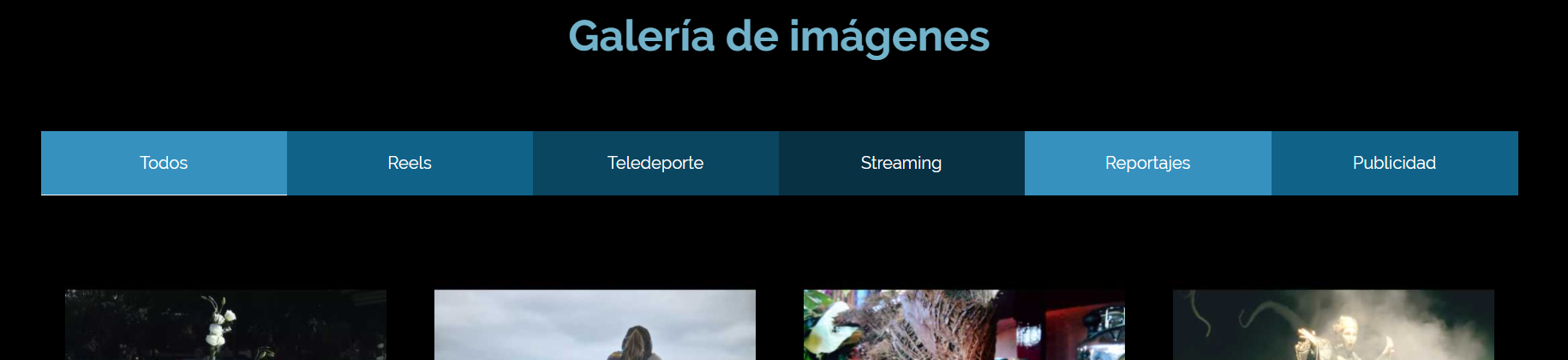
6 Slide de la web

Otro recurso, sencillo pero muy útil, consiste en una flecha que aparece al desplazarse hacia abajo en la web. Pulsando sobre dicha flecha se puede navegar rápidamente hacia la cabecera de la web en cualquier momento, sin necesidad de hacer scroll.



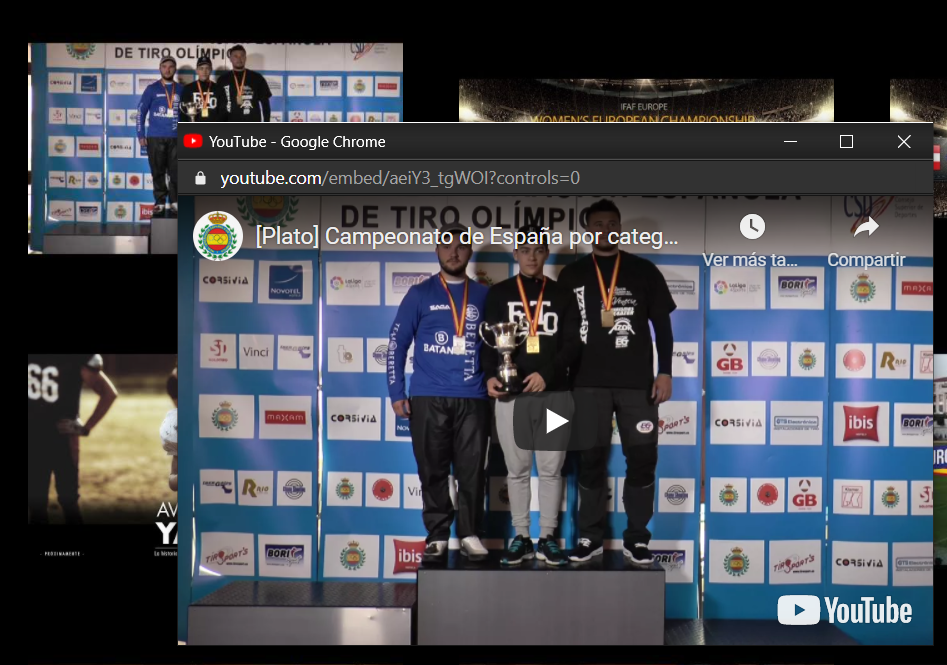
7 Flecha navegación arriba

El menú de la galería de trabajos, es otra utilidad sencilla pero que aporta valor a la usabilidad. Permite mostrar únicamente los vídeos pertenecientes a una categoría en concreto, lo cual facilita la navegación en caso de que el número de trabajos mostrados sea muy elevado, o que el cliente sólo esté interesado en una categoría en concreto.



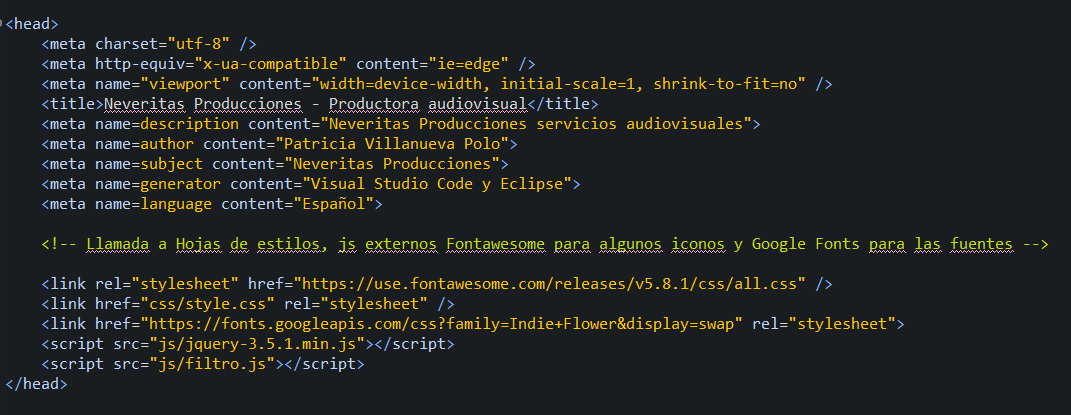
8 Menú galería de trabajos

También sobre la galería de imágenes hay un segundo JavaScript que permite que, al pinchar sobre una de las miniaturas, se abra un pop-up con el vídeo para poder visualizarlo.



9 Ejemplo reproducción videos

Se ha procurado, en todo momento, no incluir JavaScript ni código CSS en el propio HTML, para dejar más limpio el código, simplemente llamadas a los archivos necesarios desde el head del documento.



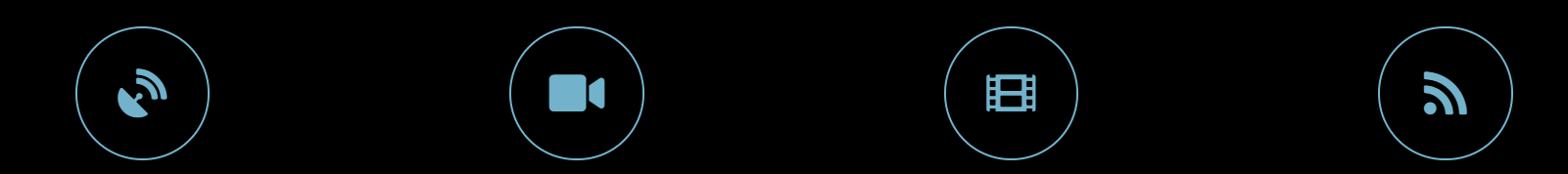
10 Muestra llamadas a archivos externos desde el HTML

Para poder ampliar el abanico de fuentes utilizadas, se ha recurrido a Google Fonts, que, con un simple script de llamada, proporciona acceso a una gran variedad de fuentes de distintas formas, tamaños y grosores, que pueden emplearse gratuitamente. Siempre, claro está, eligiendo aquellas que permitan una visualización lo más clara posible de los textos.

En el caso de algunos de los iconos que se muestran, como los de las redes sociales o la sección servicios, se ha empleado FontAwesome, también en su versión gratuita. Estos iconos renderizan muy bien, se reconocen con facilidad y no aumentan el peso de la web con archivos de imagen innecesarios.



11 iconos redes sociales



12 iconos para mostrar los servicios

También se ha incluido una llamada al script jQuery, para poder emplear este recurso a la hora de programar nuestro archivo JavaScript.

En el CSS, aparte de tratar de ser lo más simple posible, también se ha incluido el Pantone del cliente para dar color a los distintos aspectos de la web. El color de fondo, las letras, la sección de clientes o el formulario de contacto, son ejemplos de ello.



13 Ejemplos Pantone cliente

Los pasos para maquetar todo lo referente a la web principal han sido los siguientes:

* Diseño
* Generar código HTML (index.HTML)
* Generar el código CSS para darle forma y color (style.CSS)
* Generar el JavaScript para darle dinamismo a la web (filtro.js)

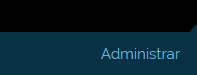
Una vez concluida esta parte de la programación, se han podido trasladar todos estos recursos al programa Eclipse y modificar las extensiones de la mayoría de los archivos HTML, por JSP, para poder llevar a cabo las funcionalidades del panel de administración que veremos a continuación. Y que forman parte de la siguiente parte del proyecto.

## El panel de Administración

Para que la web y sus contenidos fueran fácilmente administrables por la propietaria, era necesario realizar el diseño y maquetación del que sería el panel de administración.

En este panel de administración se ha mantenido la misma imagen y diseño que en la web principal, para dar sensación de continuidad al pasar de una a otra.

El acceso se realiza a través de un enlace en el pie de la web principal, denominado “Administrar”.

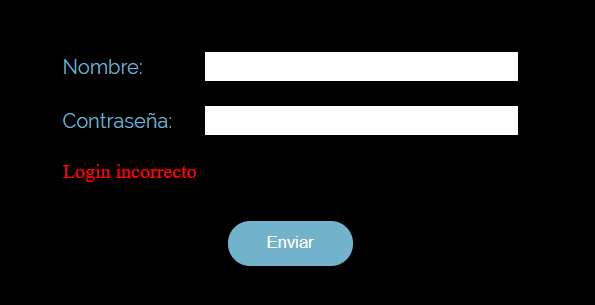


14 Enlace Administrar

Al pulsar sobre él redirige a una página de login. Tan solo se podrá acceder si se tiene un usuario y contraseña válidos. Estos datos son validados contra la base de datos, concretamente la tabla usuarios de neveritas. Si ambos datos, recogidos por el Servlet “Login”, son comparados con los que se encuentran en la tabla y coinciden, entonces se habilita en acceso, si no, muestra un mensaje de error en la propia página de Login.



15 Página Login



16 Error en el Login

El acceso nos redirige hacia una web de Bienvenida, en la que se detallan las instrucciones de uso de la aplicación.

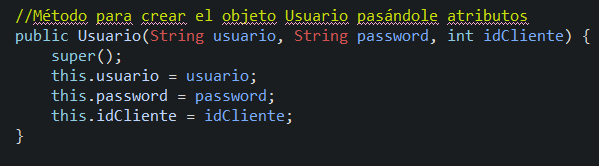


17 Página de Bienvenida

Cada una de las páginas de que consta el panel de administración, incluyendo el login y la web de bienvenida, fueron desarrolladas previamente desde Visual Studio Code en HTML, y poseen sus propios archivos CSS. Sin embargo, para la funcionalidad requerida, finalmente se transformaron a archivos JSP en Eclipse, para añadir el resto de código necesario y la conexión con los Servlets y el resto de clases.

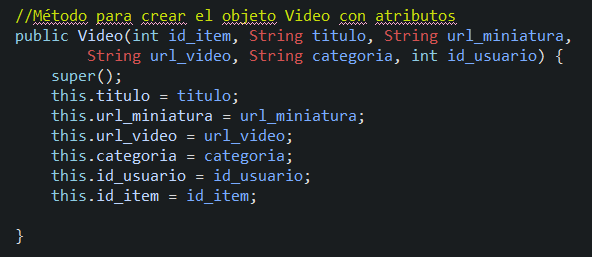
A continuación, se detallan el conjunto de archivos y sus funcionalidades:

* 1. **Clases**:
     + Usuario: Clase de la tabla usuarios, nos permite crear un objeto usuario cuyos atributos son los campos de dicha tabla.



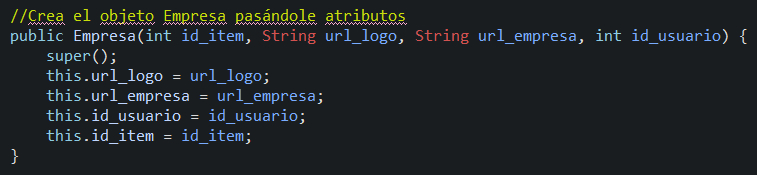
18 Objeto Usuario

* + - Video: Clase de la tabla videos, nos permite crear un objeto video cuyos atributos son los campos de dicha tabla.



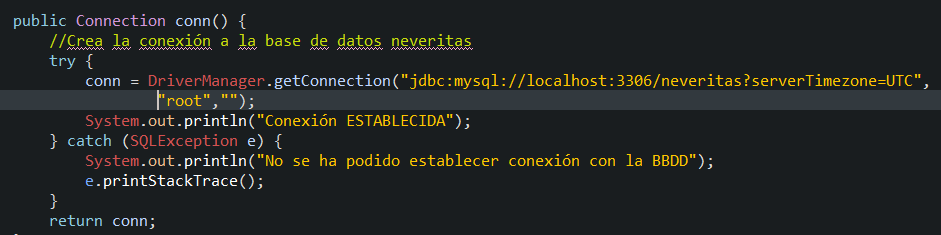
19 Objeto Video

* + - Empresa: Clase de la tabla empresas, nos permite crear un objeto empresa cuyos atributos son los campos de dicha tabla.



20 Objeto Empresa

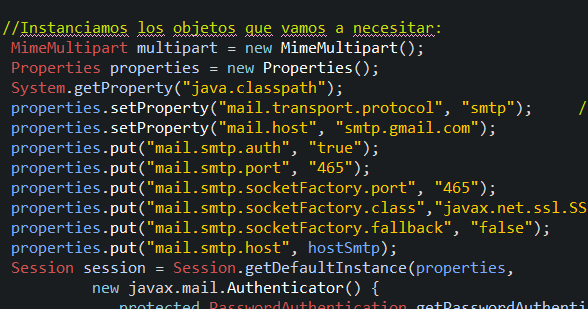
* + - UsuarioDaoImpl: Clase que implementa la interfaz IntUsuarioDao. En esta clase se desarrolla el método para buscar un usuario en la tabla usuarios, por su “usuario” y “password”.
    - VideoDaoImpl: Clase que implementa la interfaz IntVideoDao. En esta clase se desarrollan los métodos necesarios para añadir un nuevo vídeo a la tabla videos, eliminar uno, varios o todos los de dicha tabla, buscar un vídeo por el id del vídeo y buscar todos los vídeos asociados a un id\_usuario concreto y almacenarlos en un objeto Lista.
    - EmpresaDaoImpl: Clase que implementa la interfaz IntEmpresaDao. En esta clase se desarrollan los métodos necesarios para añadir una nueva empresa a la tabla empresas, eliminar una, varias o todas los de dicha tabla, buscar una empresa por el id de la empresa y buscar todas las empresas asociadas a un id\_usuario concreto y almacenarlas en un objeto Lista.
    - Conexion: Esta clase sirve para realizar la conexión con la base de datos. En ella se implementa el código y los parámetros de acceso a mysql, usuario y contraseña. Para funcionar debe estar cargada la librería del conector mysql en tomcat.



21 Objeto Conexion

* + - SendEmail: Esta clase permite realizar la conexión con el servidor de correo elegido y enviar un email desde la web. Más concretamente desde el formulario de contacto que se encuentra a pie de página. Para funcionar debe estar cargada la librería javax.mail en tomcat.

En esta clase se especifica el host que da el servicio, el puerto de acceso, el tipo de conexión, si el mensaje será de tipo HTML o tipo texto, el asunto del email, el from y otra serie de datos necesarios para realizar la conexión y envío.



22 Objeto SendEmail

* 1. **Interfaces**:
     + IntUsuarioDao: Interfaz con los métodos que deberán desarrollarse en UsuarioDaoImpl.



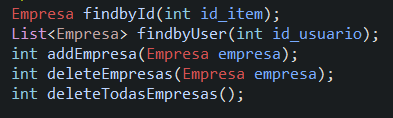
23 Interfaz Usuario

* + - IntVideoDao: Interfaz con los métodos que deberán desarrollarse en VideoDaoImpl.



24 Interfaz Video

* + - IntEmpresaDao: Interfaz con los métodos que deberán desarrollarse en EmpresaDaoImpl.



25 Interfaz Empresa

* 1. **Servlets**:
     + Login: Este Servlet o controlador se encarga de recoger la petición enviada desde el formulario de login.JSP. Recoge los datos del formulario y crea un objeto UsuarioDaoImpl para ejecutar los métodos necesarios para extraer la información de la base de datos y compararla con la del formulario. Si coincide, proporciona acceso a Bienvenida.JSP, sino muestra un mensaje de error en login.JSP.

También recoge la llamada que se invoca desde la opción del menú del panel de administración, “Salir”. Si se pulsa dicho enlace, el Servlet realiza una función para finalizar la sesión del usuario logado y le redirige a la web principal, fuera del panel de administración.



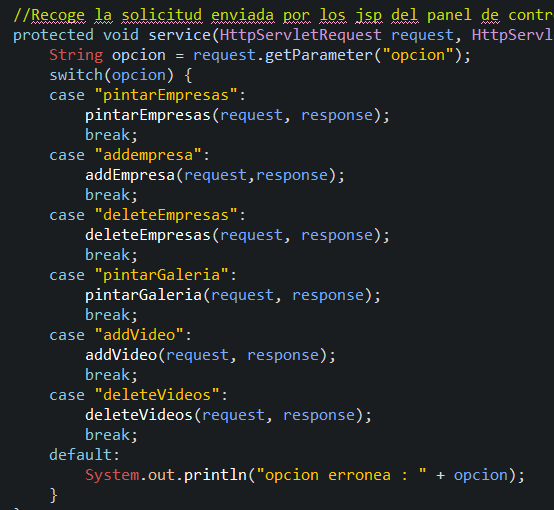
26 Enlace salir

* + - FormSender: Este controlador se encarga de recoger la solicitud del formulario de envío de la web. Recoge los datos del formulario y los valida, mostrando mensajes de error si alguno de ellos está vacío o no es correcto. En caso de estar todo bien, y usando un objeto de SendEmail, realiza la llamada al gestor de correo y envía el email al destinatario.



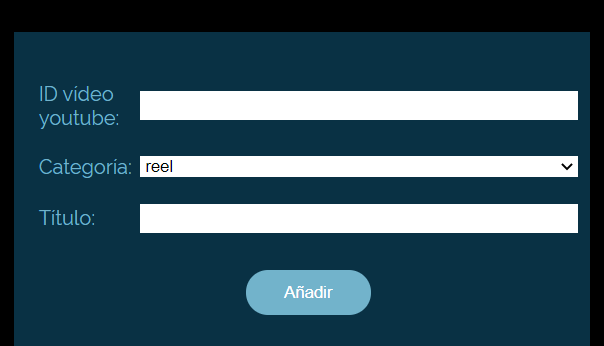
27 Formulario de contacto

* + - Administración: Este controlador se encarga de la mayor parte de las acciones del panel de administración, recogiendo las solicitudes provenientes de galería.JSP y empresas.JSP. En función del enlace pulsado “Añadir” o “Eliminar” y de qué página JSP provenga la solicitud, el método service llamará al método correspondiente a la acción que deba realizar. Dichos métodos son:



28 Servlet Administración

* + - * añadirEmpresa o añadirVideo: Creando un objeto Empresa o Video y objetos de tipo EmpresaDaoImpl o VideoDaoImpl, y recogiendo los datos proporcionados en los formularios de galería.JSP o empresa.JSP. Estos métodos añaden un nuevo vídeo o logo de empresa a la web.



29 Añadir Galeria



30 Añadir Empresa

* + - * deleteEmpresa o deleteVideo: funcionan de forma parecida, sólo que, en esta ocasión, dependiendo de los checks marcados en la página JSP, borrar uno, varios o todos los registros de la tabla correspondiente.



31 Eliminar Galeria



32 Eliminar Empresa

* + - * pintarEmpresa o pintarVideo: los métodos anteriores no redirigen de nuevo a galería o empresas, sino que llaman a los métodos pintar. Este tipo de método, se encarga de leer la tabla videos o empresas y volcar la información en un objeto List. De este modo, pueden enviar dicha lista a la página JSP y esta, podrá pintar una tabla con las empresas o videos que se encuentran en la tabla, en tiempo real.

La web principal también debe mostrar estos logos y videos, es por eso que ha habido que convertirá de HTML a JSP, para poder emplear métodos java que conecten con la base de datos y extraigan esa información para pintarla en la web. Sin embargo, no acceden en ningún momento al Servlet Administracion.

* 1. **WebContent**:
     + JS:
       - Filtro: En cuanto al panel de administración, este js tiene una función para hacer que los checkboxes de galería y empresas funcionen correctamente. Es decir, se encarga de marcar todos los checks si se marca el general, desmarcarlos si se desmarca e igualmente, si están todos marcados y se quita uno, que se desmarque el general. El resto de funciones se han visto en el epígrafe anterior.
       - jQuery-3.5.1.min: JavaScript descargado de jQuery.org que nos permite trabajar con JQuery en filtro.js.
     + CSS:
       - Style.CSS: Estilos generales de la web principal
       - Style-empresas: estilos específicos empresas.JSP
       - Style-galeria: estilos específicos galería.JSP
       - Style-login: estilos específicos login.JSP
       - Style-bienvenido: estilos específicos bienvenido.JSP
       - Style-legal: estilos específicos de aviso y privacidad.
     + HTML:
       - Aviso: web HTML con el Aviso Legal de la web, se abre como un popup.



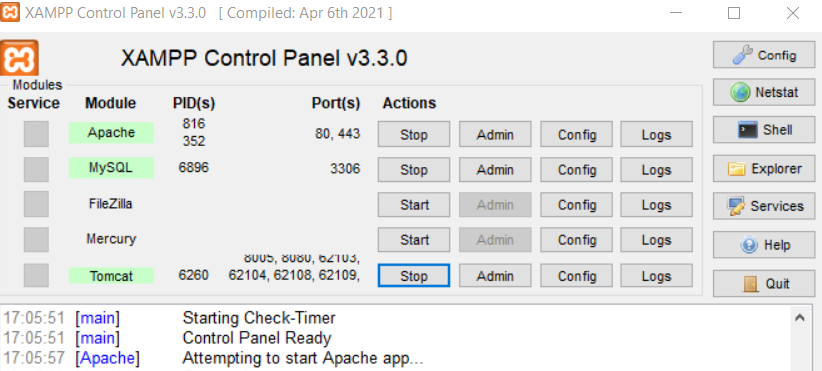
33 PopUp Aviso Legal

* + - * Privacidad: web HTML con el contenido de privacidad legal de la web, se abre como un popup.
    - JSP:
      * Index: página de inicio de la web principal
      * Login: página de login que da acceso al panel de administración.
      * Bienvenido: página de bienvenida con el manual para usar el panel de administración
      * Galería: página para administrar la galería de videos de la web principal.
      * Empresas: página para administrar los logos de empresa con sus correspondientes enlaces, de la web principal.

## Xampp y el Despliegue de la aplicación

Para desplegar la aplicación en un servidor local, se ha elegido instalar la distribución de Apache, Xampp. Dicha distribución ya viene provista de los tres recursos que son necesarios para desplegar una aplicación java en un entorno. Apache para los contenidos, Tomcat para los recursos java y MySQL para crear y gestionar la base de datos.

Una vez instalado Xampp, con un wizard muy fácil de seguir, el siguiente paso consiste en configurar la base de datos y tomcat. Para ello, desde el panel de aministración de Xampp, y tras inicializar mysql, accedemos a la bbdd, pulsando sobre el botón “Admin”.





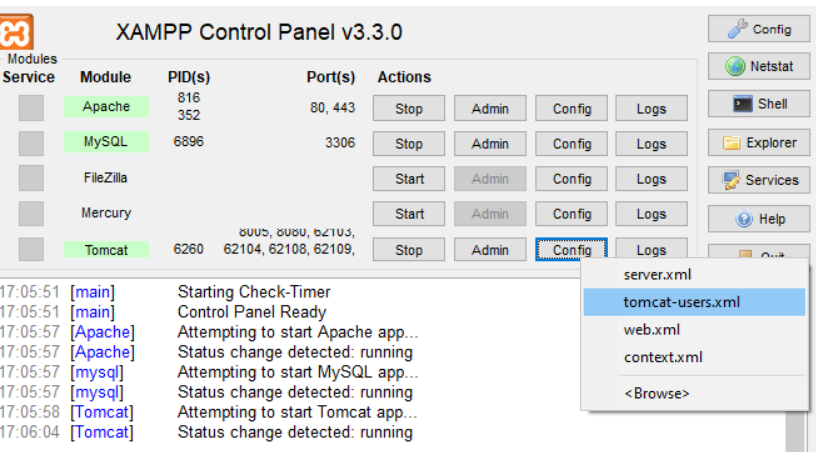
phpMyAdmin nos va a permitir ejecutar consultas de base de datos, gracias a lo cual podremos lanzar las instrucciones necesarias para generar la base de datos “neveritas” y las tablas que necesita nuestro proyecto: usuarios, videos y empresas. E, incluso, de insertar datos en dichas tablas. (Pueden consultarse las instrucciones en el anexo de este mismo documento).

El siguiente paso, será dotar a Tomcat de las librerías necesarias para la aplicación. En el caso de este proyecto, se requiere instalar:

* Javax.mail.jar
* mysql-connector-java-8.0.22.jar

La primera permitirá realizar conexiones con un servidor de correo para poder enviar emails desde la aplicación. La segunda es clave para realizar la conexión entre java y la base de datos.

Una vez instaladas las librerías, para lo que tan sólo es necesario guardar los jar en la carpeta lib de Tomcat, necesitamos tener permisos para acceder al Manager de Tomcat y poder realizar el despliegue de la aplicación .war que hemos generado anteriormente.



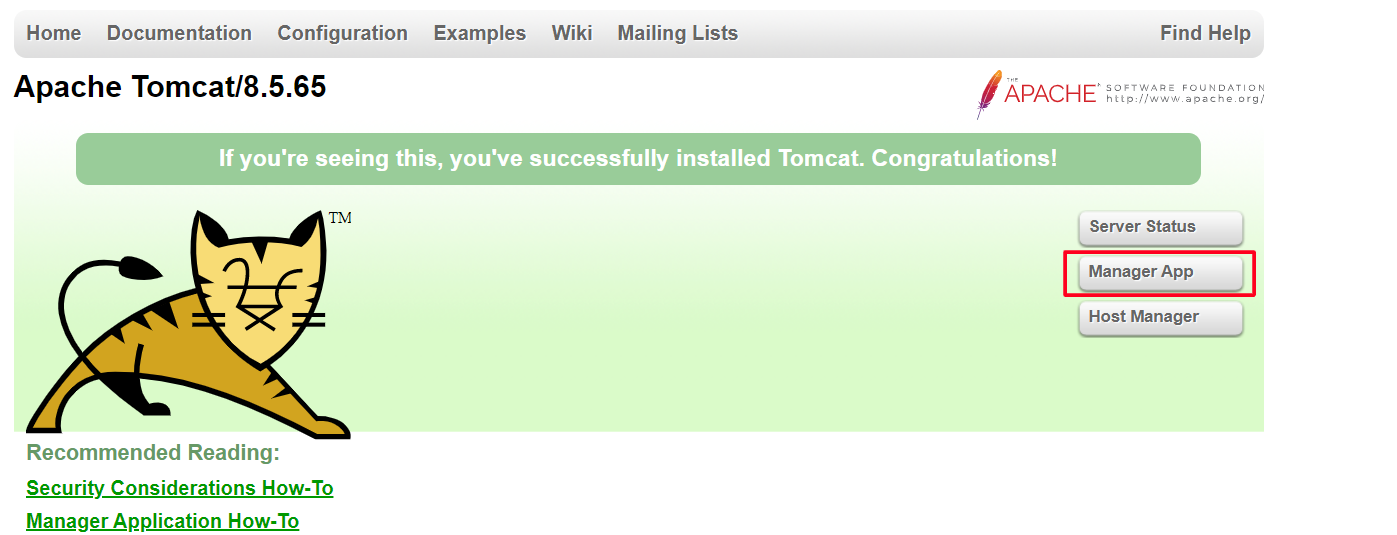
Accedemos al fichero tomcat-users.xml desde el panel de administración de Xampp. Una vez ahí, entre las etiquetas <tomcat-users> y </tomcat-users> habilitamos la siguiente configuración:

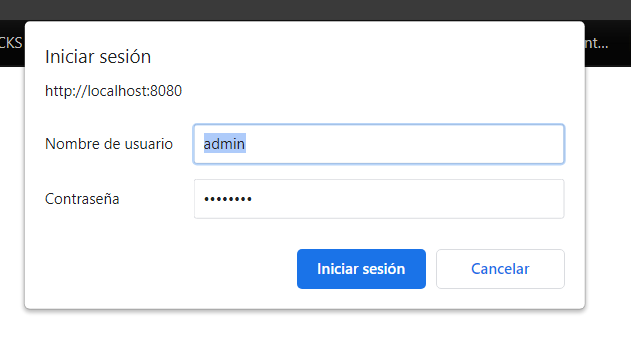
*<role rolename="manager-gui"/>*

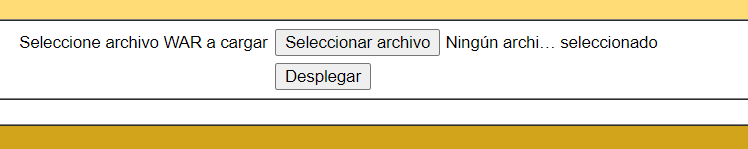
*<role rolename="manager-script"/>*

*<user username="****admin****" password="****cursodaw****" roles="manager-gui, manager-script"/>*

Dicho usuario y contraseña nos permitirá acceder al Manager de Tomcat y llevar a cabo el despliegue de la aplicación:









# Conclusiones y mejoras del proyecto

El proyecto en sí es bastante sencillo visualmente y en cuanto a los contenidos se refiere. No es una web que incluya muchos apartados ni cuyos contenidos vayan a variar con mucha frecuencia. Aún así, era necesario proveer al usuario de un pequeño panel de control, para que no tuviera que depender de un desarrollador, cada vez que quisiera hacer alguna modificación en la web.

Los contenidos con más probabilidad de ser modificados, ya que la productora se nutre de sus propios trabajos, era la galería y los logotipos de empresas, por lo que esas han sido las elegidas para desarrollar el panel de control en la aplicación java que incluye la web.

Dentro de las mejoras que podrían realizarse al proyecto, está, por supuesto, mejorar el desarrollo para que la visualización de todos los contenidos sea totalmente *responsive*, es decir, se ajuste a cualquier dispositivo.

Modificar la aplicación para que funcione con más de un usuario, por si en un futuro hay nuevos socios que pueda administrar la web y filtrar los contenidos a los que cada uno pueda tener acceso.

Hacer mejoras en el contenido del slide. Actualmente es un carrusel muy sencillo con varias imágenes que se van mostrando una tras otra. En un futuro, lo deseables sería incorporar un vídeo con un reel resumen de trabajos, alta calidad de visualización y controles para detener la visualización y modificar el volumen de la grabación.

Por otro lado, en cuanto al desarrollo del formulario de contacto, en la actualidad requiere que se inserte el usuario y contraseña de acceso al servidor de correo en el código java. Lo ideal sería que dicho código sólo estuviera asequible accediendo a una base de datos para mayor seguridad (en el código subido a la plataforma no se incluye el email y contraseña de acceso, por lo que, para verlo en funcionamiento, será necesario añadirlo en el código correspondiente).

El reclamo inicial de esta web son los trabajos expuestos, las muestras de lo que la productora puede hacer, por lo que una mejora en la galería para hacerla más amena y visual sería otro buen cambio a futuro.

Además de incluir la opción de visualizar la web en varios idiomas, mínimo inglés y español, para captar posibles clientes de habla no española.

# Bibliografía

## Sobre javax.mail:

Para saber cómo funciona esta librería y qué datos y código debo implementar para poder enviar emails desde un formulario web.

<https://desarrolloweb.com/articulos/2243.php>

<https://www.baeldung.com/spring-email>

<https://www.oracle.com/java/technologies/javamail-releases.HTML>

<https://javaee.github.io/javamail/#Download_JavaMail_Release>

## Recursos:

<https://fonts.google.com/>

<https://fontawesome.com/>

<https://www.apachefriends.org/es/index.HTML>

# Anexos (optativos)

## Scripts de cargas de BBDD

create schema neveritas;

use neveritas;

create table usuarios(

id\_usuario int NOT NULL AUTO\_INCREMENT,

usuario varchar(255) NOT NULL,

password varchar(18) NOT NULL,

PRIMARY KEY (id\_usuario)

);

INSERT INTO usuarios(usuario, password) values('Cristina', 'neveritas13');

create table videos(

id\_item int NOT NULL AUTO\_INCREMENT,

titulo varchar(255),

url\_miniatura varchar(255) NOT NULL,

url\_video varchar(1000) NOT NULL,

categoria varchar(255) NOT NULL,

id\_usuario int NOT NULL,

PRIMARY KEY (id\_item),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario)

);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Reel Neveritas Boda','https://i.ytimg.com/vi/4h5QoSCtnzY/hqdefault.jpg','https://www.youtube.com/embed/4h5QoSCtnzY?controls=0','reel',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Reel Neveritas Tiro','https://i.ytimg.com/vi/30p1IIpR4wU/hqdefault.jpg','https://www.youtube.com/embed/30p1IIpR4wU?controls=0','reel',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Reel Neveritas Eventos','https://i.ytimg.com/vi/gcv3AOt7nQU/hqdefault.jpg','https://www.youtube.com/embed/gcv3AOt7nQU?controls=0','reel',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Reel Neveritas Teatro','https://i.ytimg.com/vi/84Xb7EVNK1s/hqdefault.jpg','https://www.youtube.com/embed/84Xb7EVNK1s?controls=0','reel',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Campeonato España','https://i.ytimg.com/vi/dr4H4Wh\_QKA/hqdefault.jpg','https://www.youtube.com/embed/dr4H4Wh\_QKA?controls=0','teledeporte',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Campeonato Plato','https://i.ytimg.com/vi/aeiY3\_tgWOI/hqdefault.jpg','https://www.youtube.com/embed/aeiY3\_tgWOI?controls=0','teledeporte',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Final WEC 2015','https://i.ytimg.com/vi/jjc2Ex8JFSY/hqdefault.jpg','https://www.youtube.com/embed/jjc2Ex8JFSY?controls=0','streaming',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Opening Game WEC','https://i.ytimg.com/vi/3XWGwlJd0dw/hqdefault.jpg','https://www.youtube.com/embed/3XWGwlJd0dw?controls=0','streaming',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Armas Históricas','https://i.ytimg.com/vi/LbU7WGhlKuM/hqdefault.jpg','https://www.youtube.com/embed/LbU7WGhlKuM?controls=0','streaming',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Avanzando Yardas','https://i.ytimg.com/vi/uPkNmHtlDk8/hqdefault.jpg','https://www.youtube.com/embed/uPkNmHtlDk8?controls=0','reportaje',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Flag Football','https://i.ytimg.com/vi/ZLU9dDVCccM/hqdefault.jpg','https://www.youtube.com/embed/ZLU9dDVCccM?controls=0','reportaje',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Entrevistas Football','https://i.ytimg.com/vi/wDvTek5mghE/hqdefault.jpg','https://www.youtube.com/embed/wDvTek5mghE?controls=0','reportaje',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Moda Arte 2015','https://i.ytimg.com/vi/aMPYwOd2ECk/hqdefault.jpg','https://www.youtube.com/embed/aMPYwOd2ECk?controls=0','publicidad',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Desfile Franela Gray','https://i.ytimg.com/vi/zZCVHW-Ieow/hqdefault.jpg','https://www.youtube.com/embed/zZCVHW-Ieow?controls=0','publicidad',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('Mar de Redín','https://i.ytimg.com/vi/KNCaeCtBp00/hqdefault.jpg','https://www.youtube.com/embed/KNCaeCtBp00?controls=0','publicidad',1);

INSERT INTO videos(titulo, url\_miniatura, url\_video, categoria, id\_usuario) values('El Retrato','https://i.ytimg.com/vi/pOiIrlAh3kg/hqdefault.jpg','https://www.youtube.com/embed/pOiIrlAh3kg?controls=0','publicidad',1);

create table empresas(

id\_item int NOT NULL AUTO\_INCREMENT,

url\_logo varchar(1000) NOT NULL,

url\_empresa varchar(1000) NOT NULL,

id\_usuario int NOT NULL,

PRIMARY KEY (id\_item),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario)

);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/Esero.svg.png','http://esero.es/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/laligasport.png','https://www.laligasportstv.com/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/ArtGest.png','https://artgest.es/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/JoomladayGranada.png','https://angelquintanar.com/experiencia-con-joomla/joomladays/joomladay-granada-16',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/FEFARRSS.png','https://www.fefa.es/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/Marsyluna.png','https://elvestidordefortuny.granadaimedia.com/2015/04/14/marsy-luna-blogger-granada/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/ParqueCienciasGranada.png','https://www.parqueciencias.com/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/Prodisa.png','https://prodisacomunicacion.com/es/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/Tve\_teledeporte.png','https://www.rtve.es/directo/teledeporte/',1);

INSERT INTO empresas(url\_logo, url\_empresa, id\_usuario) values('img/FederacionTiro.png','https://www.tirolimpico.org/',1);

# Cuenta Github y enlace al TFG

Toda la información de este TFG se encuentra alojada en la siguiente cuenta y repositorio de Github: <https://github.com/pvillanuevaGit/TFG>