

TFG Desarrollo web: Avalon

Oscar Castronuño Carmona

Plan de estudios del estudiante: [Grado de Ingeniería Informática](#)

Área de trabajo final: [Dessarollo web](#)

Nombre Consultor/a [Gregorio Robles Martínez](#)

Nombre Profesor/a responsable de la asignatura [Santi Caballé Llobet](#)

Data Entrega [09/01/2020](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Avalon, red Social</i>
Nombre del autor:	<i>Oscar Castronuño Carmona</i>
Nombre del consultor/a:	<i>Gregorio Robles Martínez</i>
Nombre del PRA:	<i>Santi Caballe Llobet</i>
Data de entrega (mm/aaaa):	<i>01/2020</i>
Titulación o programa:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>TFG Desenvolupament web</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Avalon, red social</i>
<p>Resumen del Trabajo (máximo 250 palabras): Con la finalidad, contexto de la aplicación, metodología, resultados y conclusiones del trabajo.</p> <p>El objetivo de este proyecto es realizar una red social donde se puedan realizar diferentes actividades de ocio y/o ámbito profesional. Las principales funciones que tiene prevista esta aplicación son:</p> <ul style="list-style-type: none"> - Buscar personas y grupos - Enviar mensajes entre personas - Seguir miembros y grupos - Publicar información, imágenes. - Crear grupos <p>Lo que quiero conseguir es unificar todas las redes sociales en una y mejorarlas.</p>	
<p>Abstract (in English, 250 words or less):</p> <p>The objective of this project is create a social network where people can use different social or business activities. The main functions for this application are:</p> <ul style="list-style-type: none"> - Search people and groups - Send messages - Follow members and groups - Publish information, images. - Create groups <p>My objective its unify all social networks in one and improve them.</p>	

Índex

1. Introducción.....	1
1.1 Contexto y justificación del trabajo.....	1
1.2 Objetivos del trabajo.....	1
1.3 Enfoque y método utilizado.....	2
1.4 Planificación del trabajo.....	2
2. Otros capítulos.....	3
3. Conclusiones.....	41
5. Bibliografía.....	42

1. Introducción

1.1 Contexto y justificación del Trabajo

La razón por la cual escogí realizar este trabajo fue la misma razón por la cual no utilizo ninguna red social. En las redes sociales les falta privacidad. Que todas las personas puedan ver que te gusta, que haces, a quien comentas... Esa fue la razón principal por la cual quise hacer esta red social, para poder compartir todo lo que te gusta y dar me gusta a todo lo que quieras sin preocuparte del que dirán.

En mi caso, siempre he sido muy fan de manga y anime, pero nunca seguía ninguna pagina relacionada con estos temas ya que mis amistades normalmente no les gustaban esos temas.

Para resolver este problema he decidido crear mi propia red social donde puedas comentar, seguir, hablar con quien quieras y que nadie pueda verlo. (solo podrán ver los comentarios que pongas y solo si tienen acceso a esa post).

1.2 Objetivos del Trabajo

- **Personas y grupos:** Cada usuario podrá publicar en su espacio personal fotos, información y buscar personas/grupos.

Los usuarios estarán divididos en categorías (Usuario, Empresa) donde se podrán cambiar siempre que el usuario quiera desde el menú de configuración de cuenta.

- **Buscar personas y grupo:** Una vez has iniciado sesión en la aplicación podrás buscar personas o grupos en un buscador en la parte superior de la pantalla siempre visible.

- **Enviar mensajes entre personas:** Para esta funcionalidad me gustaría que cada usuario pueda decidir de quien quiere recibir los mensajes.

- **Filtro por amistad:** Solo le permite recibir mensajes de sus amigos.

- **Filtro por sexo:** Solo permite recibir mensajes de un solo sexo (*Esto solo afecta a las personas que no están en tu lista de amistades.*)

- **Seguir miembros y grupos:** Esta funcionalidad tiene como objetivo hacer llegar las publicaciones de cada usuario a todos los usuarios que te tengan en sus amistades, y que en tu pagina principal aparezcan las ultimas publicaciones de las personas/grupo que conoces.

En caso de ser nuevo y no tener ninguna amistad o no llegar al mínimo de noticias se publicaran las publicaciones de personas/grupos con el perfil abierto.

- **Administración:** La función principal de estas cuentas es moderar el correcto funcionamiento de los usuarios. Por eso crearemos un apartado oculto solo para administradores donde se podrá ver todas las publicaciones con alguna denuncia, también podrán bloquear usuarios, eliminar publicaciones, avisar a

los usuarios. Estas cuentas recibirán todas las quejas de los usuarios y serán las encargadas de solucionar/contestar todos los mensajes de estos.

1.3 Enfoque y método utilizado

Para realizar este trabajo he decidido crear una pagina web desde cero con la versión 7 de PHP_[1] , ya que creo que era lo que mejor se ajusta a mis necesidades. Ya que utilizar plataformas existentes como pueden ser WordPress, Joomla o otros tipos de pagina, no ofrece la libertad que deseaba.

Para desarrollar esta pagina web he utilizado SCRUM_[2], es un sistema por fases, en ciclos temporales cortos y de duración fija (como iteraciones) y en cada iteración se tiene que proporcionar un resultado completo, donde al finalizar el ciclo se reflexiona en el trabajo relacionado y si se tiene que mejorar alguna parte.

1.4 Planificación del trabajo

Para realizar este proyecto se necesitan un servidor web, una base de datos y cualquier editor de textos.

Explicación del proyecto y planteamiento – 18/09/2019 – 01/10/2019

Creación base de datos – 02/10/2019 – 06/10/2019

Creación estructura de la página web – 07/10/2019 – 10/10/2019

- En esta parte crearemos como se vera la página web (no diseño) solo la estructura.

Crear lógica de la página – 10/10/2019 – 20/10/2019

- En esta parte del proyecto vamos a desarrollar toda la lógica del proyecto
 - Insertar datos en la base de datos.
 - Listar datos de la base de datos.
 - Modificar datos.
 - Eliminar datos en base de datos
 - Creación de control panel para administradores web.

Rellenar la base de datos para realizar pruebas del proyecto - 20/10/2019 - 05/11/2019

Diseño web y pruebas finales – 07/11/2019 – 20/12/2019

Memoria – 21/12/2019 - 09/01/2020

2. Otros capítulos

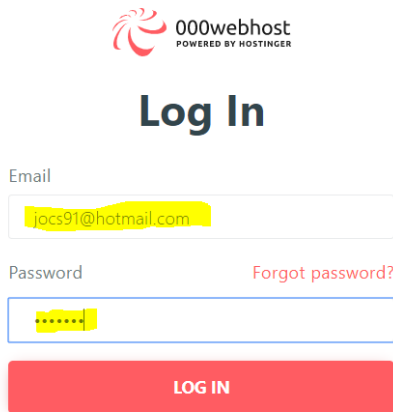
Inició pagina web:

Para realizar el proyecto he escogido un servidor gratuito para poder hacer mas real la implantación del proyecto, la idea de utilizar un servidor real, es para poder demostrar que mi proyecto se podría utilizar de forma inmediata en caso de quererlo.

Información hostweb:

Pagina donde tendremos los ficheros de Avalon:

<https://www.000webhost.com/members/website/tfgdesarrolloweboscar/dashboard>

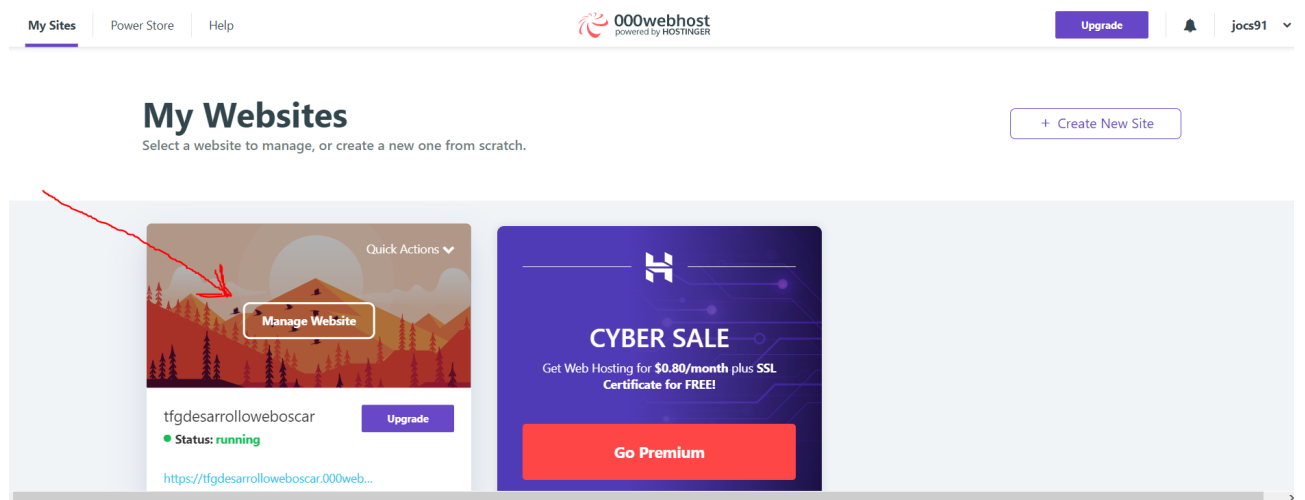


Email: jocs91@hotmail.com

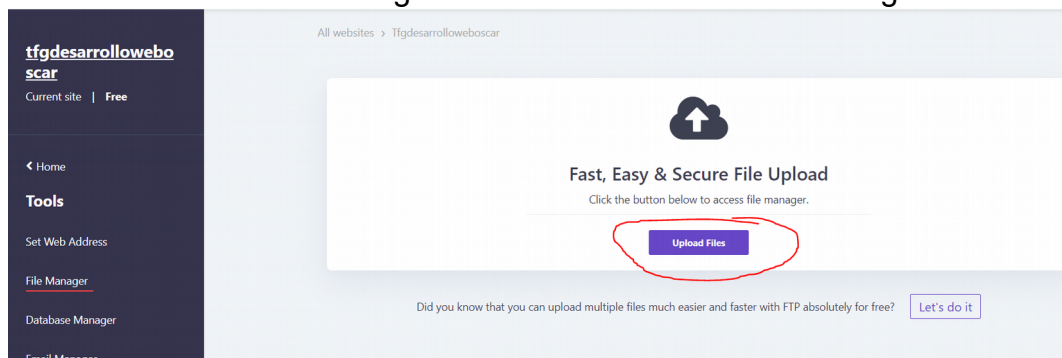
Al final he utilizado un correo persona, por problemas con el servidor, ya que es un servidor gratuito y las cuentas solo duran 1 mes.

password: oscar91

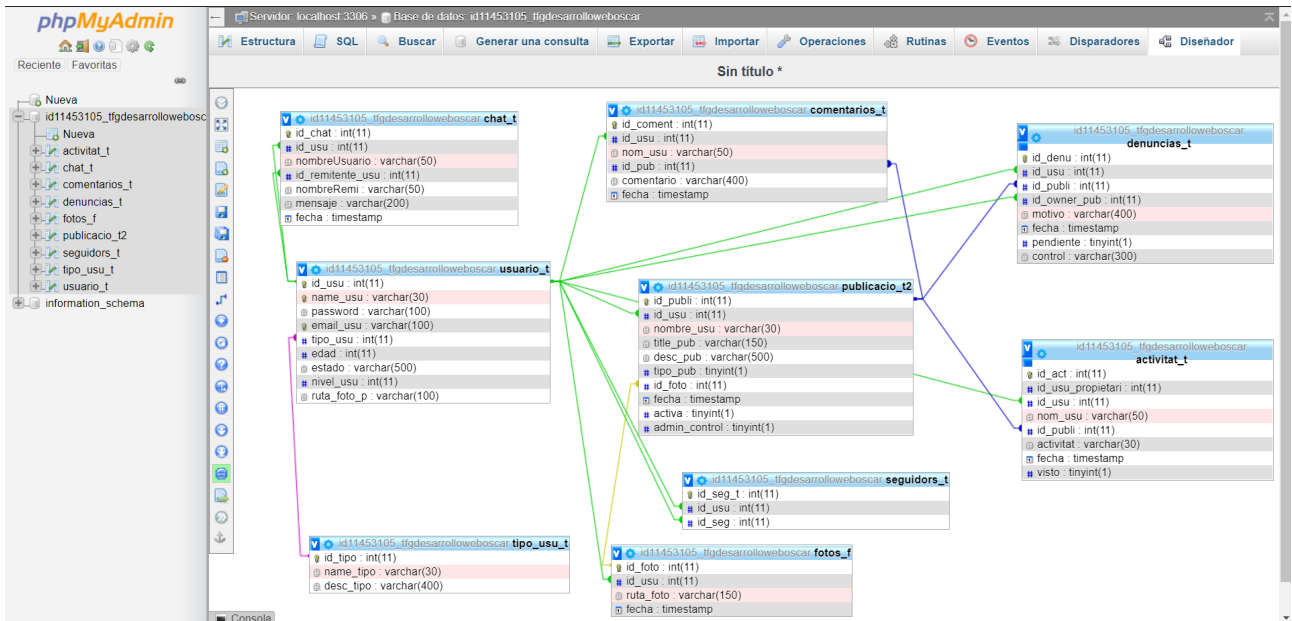
Una vez entramos en la pagina web podemos ver la siguiente pagina:



Ahora si entramos en “Manage Website” → tools → File Manager



File Manager: Es donde están alojados todos los documentos de la pagina web y donde guardaremos las imágenes de los usuarios.
 Por otra parte si en lugar de ir a “File Manager” vamos a tools → Database Manager



Database Manager: Nos permite administrar nuestra base de datos.

- usuario_t:** Tabla donde guardamos los usuarios.
- id_usu:** *int* → Valor numérico auto-incremental, valor único.
- name_usu:** *String* → Nombre del usuario, valor único.
- password:** *String* → Guardar la password.
- email_usu:** *String* → Email del usuario, valor único.
- tipo_usu:** *int* → (**Foreign Key tipo_usu_t**) Guarda id del tipo usuario.
- edad:** *int* → Guardamos la edad del usuario.
- estado:** *String* → Guardamos como se siente el usuario actualmente.
- nivel_usu:** *int* → nos dice si es administrador/moderador/persona normal (100 admin, 75 moderador, 40 usuario normal, 0 baneado) (*por ejemplo el 60 usuario de pago, 20 usuario de prueba...*)
- ruta_foto_p:** *String* → Tiene la ruta donde el usuario guarda sus fotos de perfil.

La tabla usuario se usa para poder acceder a la página y para ver si tiene permisos para acceder a diferentes zonas, de esta tabla solo destacaría el campo **nivel_usu** ya que si un administrador determina que este usuario no es apto para la utilizar nuestra página se marcará como un 0 y nunca más podrá utilizar ese usuario.

- seguidors_t:** Tabla donde guardamos amigos/seguidores.
- id_seg_t:** *int* → Valor numérico auto-incremental, valor único.
- id_usu:** *int* → (**Foreign Key usuario_t**) id usuario principal
- id_seg:** *int* → (**Foreign Key usuario_t**) id usuario o grupo a cual sigue

Esta tabla tiene como propósito guardar de cada usuario todas las personas/grupos a las que sigue. Esto solo funciona en una dirección, ya que si usuario A sigue a B, no quiere decir que B siga a A.

tipo_usu_t: Tabla donde guardamos tipos de usuarios posibles en la aplicación.

id_tipo: *int* → valor auto-incremental y único

name_tipo: *String* → Nombre del tipo. (Persona, Empresa ..)

desc_tipo: *String* → Pequeña descripción del tipo.

publicacio_t2: Tabla que guarda las publicaciones.

id_publi: *int* → Valor auto-incremental y único

id_usu: *int* → (**Foreign Key usuario_t**) Id usuario/grupo responsable de la publicación

nombre_usu: *String* → Nombre del responsable publicación.

title_pub: *String* → Título de la publicación.

desc_pub: *String* → Guarda la descripción de la publicación.

tipo_pub: *boolean* → 0 es considerada una publicación normal, 1 oferta de trabajo.

id_foto: *int* → (**Foreign Key fotos_t**) Id del grupo de fotos que están relacionadas con la publicación

fecha: *timestamp* → Fecha y hora de la publicación.

activa: *boolean* → 0 es considerado falso, cualquier otro valor verdadero.

admin_control: *boolean* → 0 es considerado normal, cualquier otro valor no se mostrara en la pagina.

Esta tabla es la encargada de guardar toda la información de las publicaciones de todos los usuarios, dar especial mención al campo **nombre_usu** ya que guardamos este nombre para no tener que hacer un select extra para mostrar el nombre, ya que si tenemos que hacer este mismo select para todas las publicaciones esto podría llegar a ralentizar la pagina web. Otro campo al que me gustaría mencionar es el campo **activa** ya se usa para saber si el usuario quiere mostrar esta publicación o no. Y con la misma función el campo **admin_control** ya que es el campo que decide si se muestra a la página o no.

denuncias_t: Tabla donde guardamos las denuncias de las publicaciones

id_denu: *int* → Valor auto-incremental y único

id_usu: *int* → (**Foreign Key usuario_t**) Id usuario/grupo responsable de la denuncia.

id_pub: *int* → (**Foreign Key publicacio_t**) Id publicación denunciada.

id_owner_pub: *int* → (**Foreign Key usuario_t**) Id usuario/grupo que recibe la denuncia

motivo: *String* → Texto explicativo de la denuncia.

fecha: *timestamp* → Fecha y hora de la denuncia.

pendiente: *boolean* → 0 es considerado falso (**no revisada**), 1 verdadero (**revisada**).

control: *String* → Administrador escribe resultado después de valorar la denuncia.

La tabla denuncias guarda las denuncias que recibe una publicación con toda la información importante, **id_usu** quien ha realizado la denuncia, **id_pub** que es la identificación de la publicación denunciada, y el **id_owner_pub** de la persona responsable de la publicación.

Estos son los campos mas importantes para los administradores y para el funcionamiento de la página web tenemos el campo **pendiente** que nos dice si ha sido revisada o no, y en caso de haber sido revisada nos dice cual ha sido la decisión del administrador.

fotos_t: Guarda todas las fotos de la aplicación.

id_foto: *int* → Valor auto-incremental y único.

id_usu: *int* → (**Foreign Key usuario_t**) id usuario responsable de la fotos.

ruta_fotos: *String* → Ruta de donde tenemos las fotos guardada.

fecha: *timestamp* → Fecha y hora de la foto.

activitat_t: Tabla donde guardamos todas las interacciones de cada publicación.
id_act: *int* → Valor auto-incremental y único.
id_usu_propietari: *int* → (**Foreign Key usuario_t**) Usuario propietario de la publicación
id_usu: *int* → (**Foreign Key usuario_t**) id usuario/grupo que ha interactuado.
nom_usu: *String* → Guardamos el nombre del usuario que realizo la actividad.
id_publi: *int* → (**Foreign Key publicacio_t**) Id publicación.
activitat: *String* → Guarda información del tipo de acción (Me gusta, comentario..).
fecha: *timestamp* → Fecha y hora de la interacción.
visto: *boolean* → 0 es considerado falso, cualquier otro valor verdadero.

comentarios_t: Tabla para los comentarios de las publicaciones.
id_coment: *int* → Valor auto-incremental y único.
id_usu: *int* → (**Foreign Key usuario_t**) Id usuario/grupo que ha comentado
nom_usu: *String* → Nombre usuario del comentario.
id_publi: *int* → (**Foreign Key publicacio_t**) Id publicación comentada
comentario: *String* → Contenido del comentario.
fecha: *timestamp* → Fecha y hora del comentario

Sistema de ficheros web.

connect.php: Pagina encargada de realizar todas las conexiones de la base de datos.

close.php: Cierra la session del usuario actual.

index.php: login

createUser.php : Creación de usuarios.

mainPage.php: Cuerpo principal del web.

usuario.php: Pagina para modificar datos personales del usuario.

publicaciones.php: Pagina que crea y lista todas las publicaciones. (la lista cambia dependiendo del modo en el que entres).

chat.php: Pagina de chat.

chat_aux.php: Pagina que hace el refresco del chat automático.

buscador.php: Pagina que muestra los usuarios.

buscadorphp.php: Pagina que hace las peticiones sql.

busc.php: script que hace que el buscador de usuarios funcione instantáneamente.

activitats.php: Muestra las notificaciones.

accion.php: Cuando un usuario hace una acción en una publicación llama a esta pagina y se encarga de comprobar si es posible o no, y realizar la acción si hace falta.

controlPanel.php: Pagina donde permite administrar la pagina web.

cssweb.css: Contiene todos los estilos de la pagina web.

Este proyecto ha sido diseñado para continuar creciendo sin ninguna limitación, por eso he creado el fichero "Connect.php" que contiene casi todas las llamadas a la base de datos. Gracias a este fichero podemos añadir, modificar, y eliminar una funcionalidad de la pagina sin tener que modificar todos los ficheros.

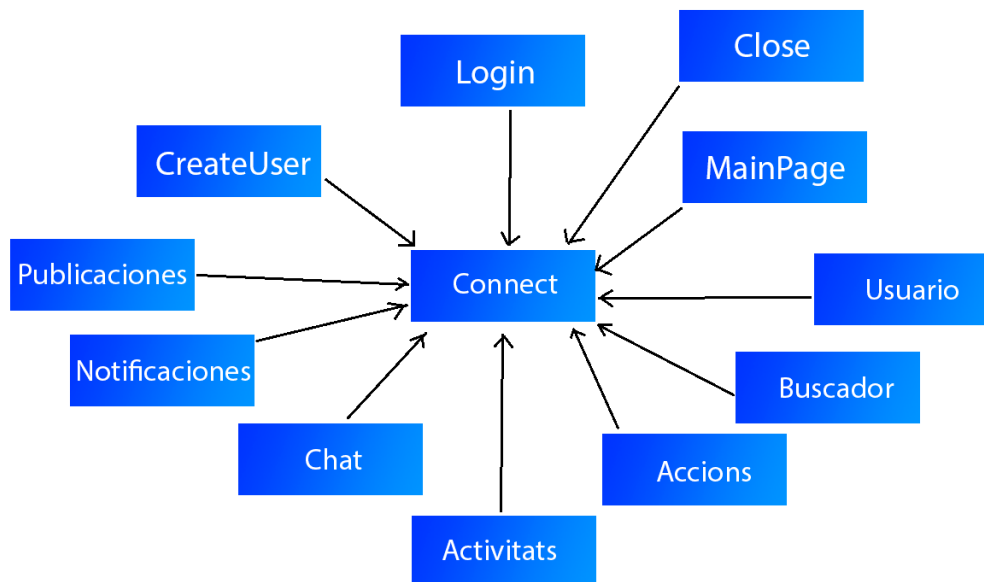


Imagen 0.1

Si en un futuro queremos implementar una nueva función, o una nueva pagina solo tendremos que diseñar la nueva pagina y añadir la funcionalidad a "Connect.php". O en caso contrario, que se quiera modificar toda la lógica del proyecto, solo se tendrá que modificar este fichero.

Así bien no todos son ventajas, también hace que este fichero sea el mas importante de la página.

Como todos las llamadas a la base de datos están concentradas en un solo fichero también he querido hacer lo mismo para la pagina web principal, de tal forma que ha quedado un diseño como se muestra en la imagen de a continuación:

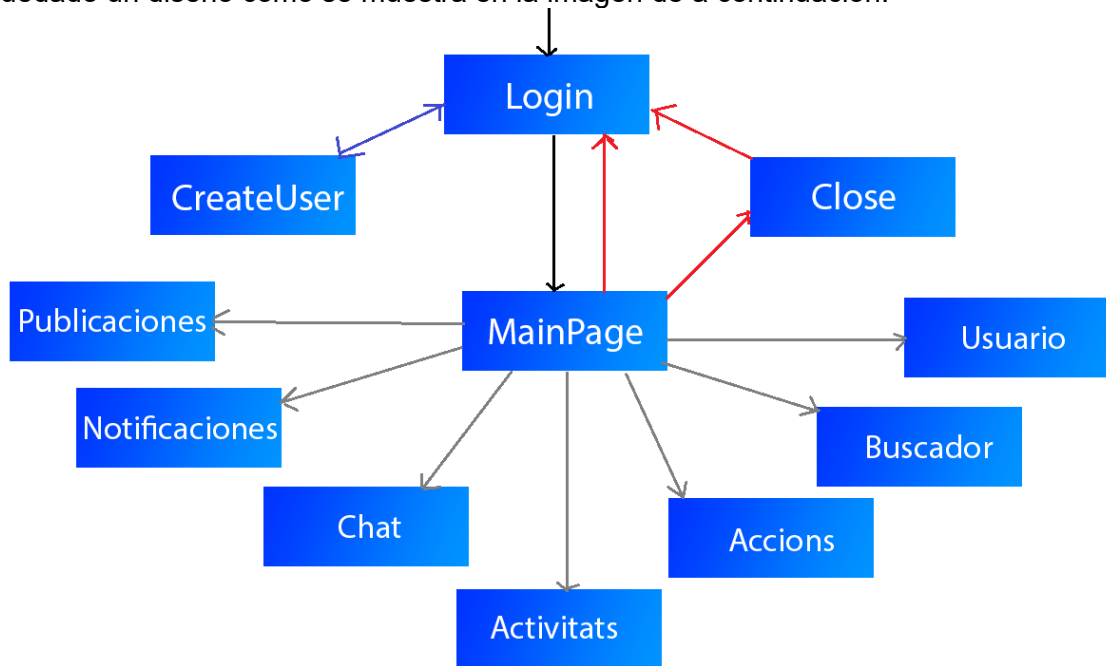


Imagen 0.2

Los usuarios que quieran entrar a nuestra aplicación siempre tendrán que hacer login primero, en caso de no hacer login y escribir la URL (<https://tfgdesarrolloweboscar.000webhostapp.com/mainPage.php>) directamente la página te redireccionará directamente al login.

Cuando un usuario entra a nuestra aplicación por primera vez se le mostrará el “login.php” donde podrá escoger entre hacer login con usuario que ya tenga, o crear un nuevo usuario “CreateUser.php”.

Cuando entramos con un usuario valido, se muestra la página web “MainPage.php” que sera la página encargada de mostrar todas las demás páginas. Para lograr hacer esto utilizamos una etiqueta de html <iframe> que nos permite mostrar otras páginas web dentro de una nuestra y donde se mostraran todas las funcionalidades de la página.

En este <iframe> lo primero que se muestra es “publicaciones.php” en modo = tendencias, donde se mostraran todas las publicaciones de todos los usuarios en orden por ultima publicación. La idea de mostrar todas las publicaciones sin importar si son amigos o no, es para los nuevos usuarios, ya que seria un poco triste que nada mas entrar encontraran una página sin ninguna publicación.

Aparte el usuario puede escoger entre todas las opciones que nos ofrece nuestro sitio web.

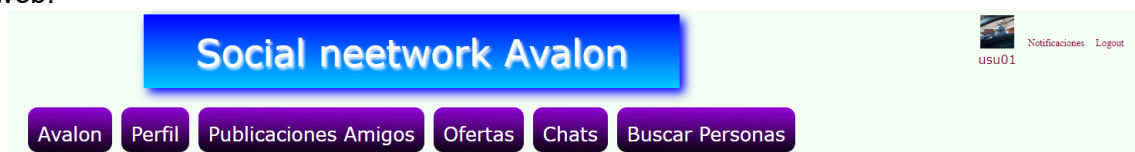


Imagen 0.3

Cuando vamos a **Avalon** “Imagen 0.3” muestra todas las publicaciones de todos los usuarios sin importar si son amigos o no, si vamos a **Perfil** “Imagen 0.3” muestra todas las publicaciones que hemos publicado en el sitio web y si vamos a **Publicaciones Amigos** “Imagen 0.3” muestra todas las publicaciones que pertenecen a nuestros amigos o las nuestras.

Todas las publicaciones siempre se muestran por orden de publicación de mas nueva a mas vieja.

El apartado de **Ofertas** “Imagen 0.3” esta diseñado para mostrar las ofertas de trabajo de las empresas, ya que si eres una empresa también puedes crear ofertas de trabajo. En este apartado también se puede filtrar por empresas a las que sigues, o todas las empresas.

En cuanto a **Chats** “Imagen 0.3” no tiene el mejor nombre, ya que es un nombre poco intuitivo, pero nos permite ver todos tus amigos/seguidores y chatear con ellos.

Buscar Personas “Imagen 0.3” nos permite buscar nuevos usuarios a los que no tienes en tu lista de amigos, una vez le das a seguir a un usuario desaparecerá de la lista de buscar personas.

En cuanto en la “Imagen 0.3” arriba a la derecha podemos ver la foto de nuestro usuario y nuestro nombre, en caso de la “Imagen 0.3” nuestro nombre es “usu01”, y si entramos a **Usuario** Puedes modificar los datos del usuario.

Notificaciones “Imagen 0.3” Puedes ver todo lo que este relacionado con tu usuario (me gustas, comentarios..).

Logout: sales de la pagina y te reenvía a “index.php”.

Restricciones de la aplicación:

CreateUser.php

- Al crear un usuario, no se puede utilizar un correo que exista en la aplicación, o un nombre de usuario que exista.
- La edad del usuario tiene que ser superior a 1 y inferior a 120 años.
- Los passwords tienen que coincidir.
- Se tienen que informar el nombre de usuario, Email, password, re-password.

Usuario.php

- No podemos modificar el nombre de usuario.
- Tiene las mismas restricciones que “CreateUser.php”.

publicaciones.php

- Como máximo se pueden subir 10 imágenes por publicación.

Funcionamiento de la aplicación

- Crear usuarios sin foto:

	id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
	1	Admin_dios	Admin_dios	Admin_dios@hotmail.com	1	28		100	img/chica01/profile/1.jpg
	3	empresa01	empresa01	empresa01@hotmail.com	5	50	Somos los mejores	40	img/empresa01/profile/1.jpg
	4	usuario01	usuario01	usuario01@hotmail.com	3	25	usuario01	40	img/usuario01/profile/1.jpg
	6	usuario02	usuario02	usuario02@hotmail.es	3	25	usuario02	40	img/usuario02/profile/1.jpg
	7	usuario03	usuario03	usuario03@hotmail.com	3	35		0	img/no_photo.jpg

Figura 1.

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Usuario creado [Ir al login](#)

Figura 2.

Figura 3.

	id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
	1	Admin_dios	Admin_dios	Admin_dios@hotmail.com	1	28		100	img/chica01/profile/1.jpg
	3	empresa01	empresa01	empresa01@hotmail.com	5	50	Somos los mejores	40	img/empresa01/profile/1.jpg
	4	usuario01	usuario01	usuario01@hotmail.com	3	25	usuario01	40	img/usuario01/profile/1.jpg
	6	usuario02	usuario02	usuario02@hotmail.es	3	25	usuario02	40	img/usuario02/profile/1.jpg
	7	usuario03	usuario03	usuario03@hotmail.com	3	35		0	img/no_photo.jpg
	8	nophotouser	nophotouser	nophotouser@hotmail.com	3	28	No tengo foto	40	img/no_photo.jpg

Figura 4.

Figura 1: Muestra como en la base de datos no existe ningún usuario con el nombre "nophotouser".

Figura 2: Muestra como rellenamos el formulario para crear un nuevo usuario en la aplicación.

Figura 3: Muestra como el usuario ha sido creado satisfactoriamente.

Figura 4: Muestra como el usuario ya aparece en la base de datos.

- Crear usuario con foto:

Partiendo "Imagen 4" de la prueba anterior creamos un nuevo usuario con imagen:

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: **art04.jpg**

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Usuario creado [Ir al login](#)

Imagen 5.

		id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
<input type="checkbox"/>		1	Admin_dios	Admin_dios	Admin_dios@hotmail.com	1	28		100	img/chica01/profile/1.jpg
<input type="checkbox"/>		3	empresa01	empresa01	empresa01@hotmail.com	5	50	Somos los mejores	40	img/empresa01/profile/1.jpg
<input type="checkbox"/>		4	usuario01	usuario01	usuario01@hotmail.com	3	25	usuario01	40	img/usuario01/profile/1.jpg
<input type="checkbox"/>		6	usuario02	usuario02	usuario02@hotmail.es	3	25	usuario02	40	img/usuario02/profile/1.jpg
<input type="checkbox"/>		7	usuario03	usuario03	usuario03@hotmail.com	3	35		0	img/no_photo.jpg
<input type="checkbox"/>		8	nophotouser	nophotouser	nophotouser@hotmail.com	3	28	No tengo foto..	40	img/no_photo.jpg
<input type="checkbox"/>		9	test01	test01	test01@hotmail.com	3	25	si tengo foto	40	img/test01/profile/1.jpg

Imagen 6.

Imagen 7.

En esta prueba partimos de la prueba anterior "Imagen 4" donde podemos ver que no hay ningún usuario con el nombre test01. Una vez hemos visto que no existe el usuario vamos a crear el nuevo usuario "Imagen 6". En esta prueba podemos ver como en la "Imagen 5" se ha seleccionado una imagen y en la "Imagen 7" esta guardada en la base de datos.

- Crear usuario tipo empresa:



Imagen 8.



Imagen 9.

Imagen 10.

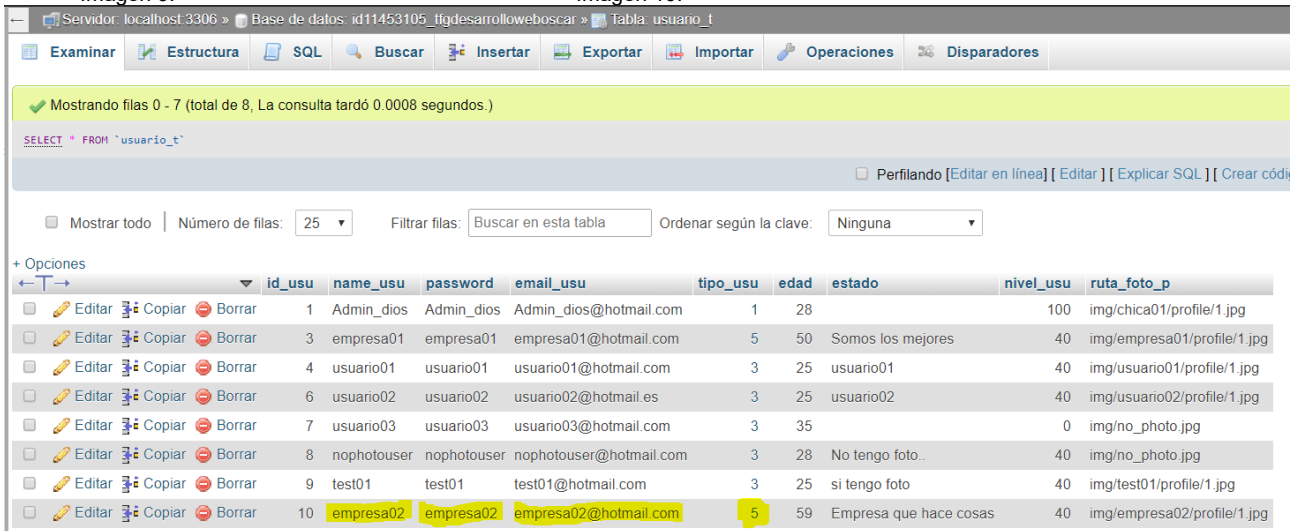


Imagen 11.

En la “Imagen 8” podemos ver todos los tipos de usuarios que se pueden crear actualmente en nuestra página web. Como podemos ver vamos a crear un usuario de tipo Empresa “Imagen 9”, y en la “Imagen 10” muestra como el usuario ha sido creado correctamente y guardado en la base de datos como tipo empresa (tipo_usu = 5) “Imagen 11”.

- Crear usuario ya registrado:

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Imagen 12.

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

El nombre ya existe, escoge otro nombre.

Imagen 13.

Para esta prueba no importa el tipo de usuario, no puede haber dos usuarios con el mismo nombre. En esta prueba vemos como en la “Imagen 12” estamos intentando crear el usuario “test01”, el cual hemos creado a la prueba anterior (mirar Imagen 11 de la prueba anterior) y en la “Imagen 13” nos salta un error “El nombre ya existe, escoge otro nombre”.

- Crear usuario Password no coincide:

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: album02-4.JPG

Imagen 14.

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Los password no coinciden.

Imagen 15.

En esta prueba podemos ver que en la “Imagen 14” los passwords no coinciden, así que en la “Imagen 15” nos muestra un error “Los password no coinciden”.

- Crear usuario con Email ya registrado:

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: album02-4.JPG

Imagen 16.

Nuevo Usuario de Avalon

Username:

Password:

re-Password:

Email:

Edad:

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

El Email ya existe.

Imagen 17.

En esta prueba podemos ver que en la “Imagen 16” el Email registrado ya ha sido registrado anteriormente (mirar Imagen 11), y en la “Imagen 17” nos muestra un error “El Email ya existe”.

- Crear usuario con otros errores:

Nuevo Usuario de Avalon
*Campos con * son obligatorios*

Username*:

Password*:

re-Password*:

Email*:

Edad (1 - 120):

Escribe como

Estado: ! El valor debe ser inferior o igual a 120

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Imagen 18.

Nuevo Usuario de Avalon
*Campos con * son obligatorios*

Username*:

Password*:

re-Password*:

Email*:

Edad (1 - 120):

Escribe como te sientes

Estado:

Tipo usuario:

Foto perfil: Ningún archivo seleccionado

Faltan campos para rellenar!

Imagen 19.

En esta prueba mostramos los otros posibles errores, como puede ser al poner una edad superior a 120 años “Imagen 18” o dejar algún campo sin rellenar (Username*, Password*, re-Password*, Email*) “Imagen 19”.

El error de la “Imagen 19” saldrá siempre que falte uno de los campos marcados con “*”.

- Login No existe usuario:

Login

Username:

Password:

[Crear cuenta](#)

Imagen 20.

Datos Incorrectos. [Try again!!](#)

Login

Username:

Password:

[Crear cuenta](#)

Imagen 21.

En esta prueba intentamos hacer login con usuario “Imagen 20” que no existe (o las credenciales no coinciden) para generar el error de la “Imagen 21”.

- Sistema login correcto:

Login

Username:

Password :

[Crear cuenta](#)

Imagen 22.



Imagen 23.

En esta prueba mostramos como utilizando un usuario correcto “**Imagen 22**” podemos acceder a la aplicación “**Imagen 23**”, en caso de escribir la URL a mano, sin haber realizado el login, la página te enviaría a la página de login otra vez.

Aparte de que usuario sea correcto también verificamos que el usuario no este suspendido por los administradores, en caso de estar suspendido por los administradores salta el error de login incorrecto.

- Publicar con una imagen:

Para esta prueba vamos a revisar la tabla de publicaciones y ver que el ultimo ID es el 17. “Imagen 24”

	id_publi	id_usu	nombre_usu	title_pub	desc_pub	tipo_pub	id_foto	fecha	activa	admin_control
<input type="checkbox"/>	2	4	usuario01	Primera publicacion de la pagina	Esta publicacion estara activa para todos.	0	13	2019-12-18 03:24:29	1	0
<input type="checkbox"/>	3	4	usuario01	Segunda publicacion	Esta publicacion solo sera para mi	0	8	2019-12-18 05:08:08	0	0
<input type="checkbox"/>	4	4	usuario01	Tercera publicacion	Esta publicacion sera para todos, no foto	0	8	2019-12-20 10:40:28	1	0
<input type="checkbox"/>	5	3	empresa01	Publicacion oculta	oculta sin foto	0	8	2019-12-18 03:30:45	0	0
<input type="checkbox"/>	6	3	empresa01	Publicacion para todos	para todos con foto	0	15	2019-12-18 03:31:06	1	0
<input type="checkbox"/>	8	3	empresa01	Primera oferta	Buscamos Programador de php, pagamos 10.000€	1	8	2019-12-18 04:55:16	1	0
<input type="checkbox"/>	9	3	empresa01	No oferta	Publicacion normal sin foto	0	8	2019-12-18 04:56:36	1	0
<input type="checkbox"/>	10	3	empresa01	Publicacion normal foto	foto	0	18	2019-12-18 04:57:01	1	0
<input type="checkbox"/>	11	3	empresa01	oferta con foto	foto 100€	1	19	2019-12-18 04:57:21	1	0
<input type="checkbox"/>	12	7	usuario03	publicacion con denuncia falsa	admins dian que todo ok	0	25	2019-12-19 04:09:54	1	1
<input type="checkbox"/>	13	7	usuario03	publicacion denuncia	admin dira que si	0	8	2019-12-18 16:06:19	1	1
<input type="checkbox"/>	14	9	test01	Publicación con una imagen	En esta prueba vamos a subir una sola imagen	0	26	2019-12-20 11:14:36	1	1
<input type="checkbox"/>	15	9	test01	Publicación 2Imagen	Publicación con mas de una imagen	0	27	2019-12-20 11:14:36	1	1
<input type="checkbox"/>	16	9	test01	publicacion para denunciar	texto	0	28	2019-12-20 10:53:17	1	1
<input type="checkbox"/>	17	12	usu01	hello	how are you??	0	29	2020-01-02 11:18:37	0	0

Imagen 24.

Publicar en Avalon

Titulo:

Descripción:

Foto:

Ningún archivo seleccionado

Publicacion con imagen usu01 [Ocultar publicacion](#)

Me gusta (0) [2020-01-08 11:40:01](#) [Denunciar](#)

Publicacion con imagen

Añade tu comentario:

Imagen 25.

Imagen 26.

Una vez hemos visto que la ultima publicación tiene el ID = 17 podemos crear la nueva publicación con las dos imágenes “Imagen 25” y “Imagen 26”.

	id_publi	id_usu	nombre_usu	title_pub	desc_pub	tipo_pub	id_foto	fecha	activa	admin_control
<input type="checkbox"/>	2	4	usuario01	Primera publicacion de la pagina	Esta publicacion estara activa para todos.	0	13	2019-12-18 03:24:29	1	0
<input type="checkbox"/>	3	4	usuario01	Segunda publicacion	Esta publicacion solo sera para mi	0	8	2019-12-18 05:08:08	0	0
<input type="checkbox"/>	4	4	usuario01	Tercera publicacion	Esta publicacion sera para todos, no foto	0	8	2019-12-20 10:40:28	1	0
<input type="checkbox"/>	5	3	empresa01	Publicacion oculta	oculta sin foto	0	8	2019-12-18 03:30:45	0	0
<input type="checkbox"/>	6	3	empresa01	Publicacion para todos	para todos con foto	0	15	2019-12-18 03:31:06	1	0
<input type="checkbox"/>	8	3	empresa01	Primera oferta	Buscamos Programador de php, pagamos 10.000€	1	8	2019-12-18 04:55:16	1	0
<input type="checkbox"/>	9	3	empresa01	No oferta	Publicacion normal sin foto	0	8	2019-12-18 04:56:36	1	0
<input type="checkbox"/>	10	3	empresa01	Publicacion normal foto	foto	0	18	2019-12-18 04:57:01	1	0
<input type="checkbox"/>	11	3	empresa01	oferta con foto	foto 100€	1	19	2019-12-18 04:57:21	1	0
<input type="checkbox"/>	12	7	usuario03	publicacion con denuncia falsa	admins dian que todo ok	0	25	2019-12-19 04:09:54	1	1
<input type="checkbox"/>	13	7	usuario03	publicacion denuncia	admin dira que si	0	8	2019-12-18 16:06:19	1	1
<input type="checkbox"/>	14	9	test01	Publicación con una imagen	En esta prueba vamos a subir una sola imagen	0	26	2019-12-20 11:14:36	1	1
<input type="checkbox"/>	15	9	test01	Publicación 2Imagen	Publicación con mas de una imagen	0	27	2019-12-20 11:14:36	1	1
<input type="checkbox"/>	16	9	test01	publicacion para denunciar	texto	0	28	2019-12-20 10:53:17	1	1
<input type="checkbox"/>	17	12	usu01	hello	how are you??	0	29	2020-01-02 11:18:37	0	0
<input type="checkbox"/>	18	12	usu01	Publicacion con imagen	Publicacion con imagen	0	30	2020-01-08 11:40:01	1	0

Imagen 27.

Una vez creada la nueva publicación podemos ver que ha creado la publicación 18 con el id_foto = 30 “Imagen 27”.

id_foto	id_usu	ruta_foto	fecha
8	1		2019-12-17 15:43:46
13	4	img/usuario01/publicaciones/1	2019-12-18 03:24:29
15	3	img/empresa01/publicaciones/1	2019-12-18 03:31:06
18	3	img/empresa01/publicaciones/2	2019-12-18 04:57:01
19	3	img/empresa01/publicaciones/3	2019-12-18 04:57:21
25	7	img/usuario03/publicaciones/1	2019-12-18 14:42:26
26	9	img/test01/publicaciones/1	2019-12-20 07:59:29
27	9	img/test01/publicaciones/2	2019-12-20 08:02:36
28	9	img/test01/publicaciones/3	2019-12-20 10:50:07
29	12	img/usu01/publicaciones/1	2020-01-02 11:17:10
30	12	img/usu01/publicaciones/2	2020-01-08 11:40:01

Imagen 28.

También comprobamos la tabla de imágenes con el ID_foto = 30 “Imagen 28” que es el mismo id que teníamos en la publicación “Imagen 27”. Podemos ver que la ruta de las imágenes es “img/usu01/publicaciones/2”:



Name	Size	Date	Permissions
0.jpg	19.8 kB	2019-12-20 08:02:00	-rw-r--
1.jpg	20.2 kB	2019-12-20 08:02:00	-rw-r--

Podemos ver que tiene dos imágenes dentro de la carpeta, entonces por código contamos el numero de imágenes que tiene y las mostramos con la url: “img/usu01/publicaciones/2/X.jpg” donde X es el un numero dentro de un bucle que va desde 0 a numero de fotos.

- Presionar una imagen:



Imagen 29.

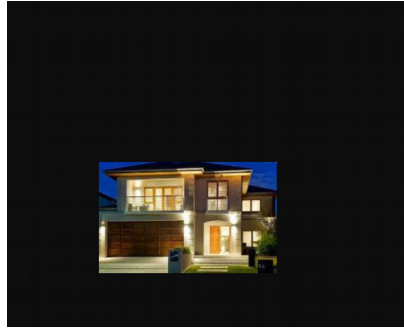


Imagen 30.

En esta prueba, podemos ver que cuando un usuario presiona una imagen “Imagen 29” abre una nueva ventana con la imagen que ha seleccionado “Imagen 30”.

- Ver todas las publicaciones: (Avalon)

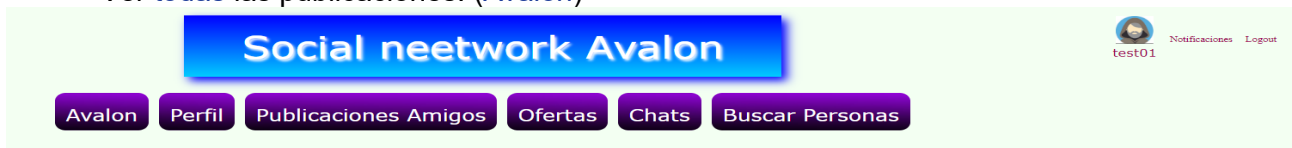


Imagen 31.

Imagen 31: Muestra por orden todas las publicaciones realizadas por todos los usuarios, sin importar si lo conoces o no.



Imagen 32.

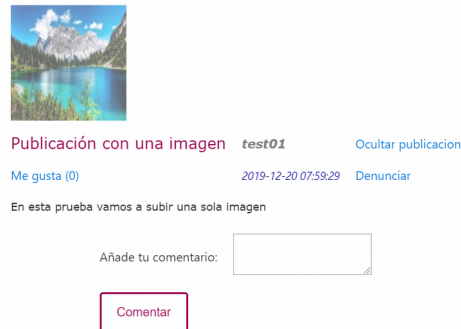


Imagen 33.



Imagen 34.

En Lila se muestra el titulo de la publicación.
En Gris se muestra el nombre del responsable de la publicación.
El botón para decir si te gusta o no (solo se puede dar me gusta una vez por persona en una misma publicación).
En azul la fecha de la publicación con su hora.
El botón para denunciar (solo se puede denunciar una vez por persona en una misma publicación).
Y luego puedes añadir un comentario.
Justo debajo si tiene comentarios se muestran por orden.

- Ver todas **tus** las publicaciones: (Perfil)

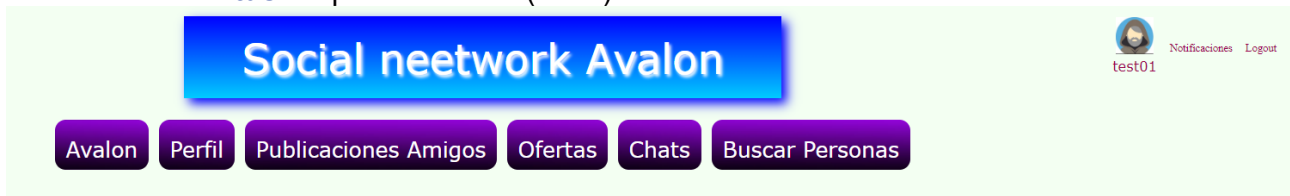


Imagen 35.



Imagen 36.

Imagen 37.

En esta prueba se puede ver como si entramos en “Perfil” se muestran nuestras todas las publicaciones realizadas por el usuario que ha accedido a nuestra web, en este caso test01 “Imagen 35”, en las “Imagen 36” y “Imagen 37” podemos ver las diferentes publicaciones de este usuario.

- Ver todas las publicaciones de tus **amigos/followers**: (Publicaciones Amigos)



Imagen 38.

En la “Imagen 38” podemos ver como se muestran las publicaciones de nuestro amigo “usuario01” y nuestras publicaciones “test01” ya que cuando entramos en “Publicaciones de amigos” también podemos ver las publicaciones que hemos realizado nosotros mismos.

- Buscar personas:

			11	usu	usu	3	25	ahahs	40	img/no_photo.jpg
			12	usu01	usugsgssh	3	24	hshs	40	img/usu01/profile/1.jpg

Imagen 39.



Imagen 40.



Imagen 41.

Entramos a “Buscar Personas” donde esta funcionalidad solo muestra las personas que no tenemos en nuestra lista de amistades, antes de hacer nada comprobamos los Id de nuestro usuario “usu01” = 12 y el usuario que vamos a seguir “usu” = 11 “Imagen 39”. En la “Imagen 40” podemos ver como se muestra el usuario “usu” y en la “Imagen 41” podemos ver como ya no se muestra, eso es por que ya le hemos dado a seguir.

id_seg_t	id_usu	id_seg
6	4	3
7	3	4
8	9	4
9	4	9
10	4	8
11	11	6
12	12	3

Imagen 42.

id_seg_t	id_usu	id_seg
6	4	3
7	3	4
8	9	4
9	4	9
10	4	8
11	11	6
12	12	3
13	12	11

Imagen 43.

En la “Imagen 42” podemos ver como aun no lo estamos siguiendo, y en la “Imagen 43” ya lo estamos siguiendo.

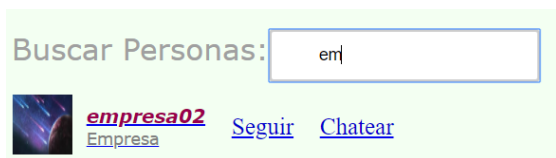


Imagen 44.

Esta “Imagen 44” simplemente muestra como el buscador funciona en tiempo real.

- Dar me gusta a una publicación:

	id_act	id_usu_propietari	id_usu	nom_usu	id_publici	activitat	fecha	visto
	19	3	4	usuario01	11	me gusta	2019-12-18 09:36:32	1
	20	3	4	usuario01	9	me gusta	2019-12-18 09:36:32	1
	21	3	4	usuario01	8	Comentario	2019-12-18 09:36:32	1
	22	4	4	usuario01	2	me gusta	2019-12-18 09:36:22	1
	23	3	3	empresa01	11	me gusta	2019-12-18 09:37:15	1
	24	7	7	usuario03	13	Comentario	2019-12-18 14:45:48	1
	25	7	7	usuario03	13	Comentario	2019-12-18 14:45:48	1
	26	7	4	usuario01	12	me gusta	2019-12-18 14:45:48	1
	27	3	4	usuario01	11	Comentario	2019-12-19 07:52:50	0
	28	4	4	usuario01	3	me gusta	2019-12-20 07:14:09	1
	29	4	4	usuario01	4	Comentario	2019-12-20 07:16:50	1
	30	4	4	usuario01	4	Comentario	2019-12-20 07:25:25	1

Imagen 45

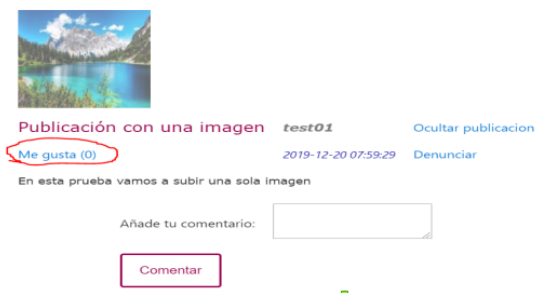


Imagen 46.



Imagen 47.

			31	9	9	test01	14	me gusta	2019-12-20 08:39:08	0
--	--	--	----	---	---	--------	----	----------	---------------------	---

Imagen 48.

En esta prueba podemos ver como en la “Imagen 45” no hay ningún “me gusta” del usuario *test01*. Así que vamos a dar “me gusta” en la “Imagen 46” y en la “Imagen 47” se muestra que ya tiene un me “me gusta(1)”.

Una vez le das a “me gusta” se inserta a la base de datos con tu usuario “Imagen 48”. esta información luego servirá para comprobar que un usuario no intenta dar “me gusta” a una publicación consecutivas veces, ya que solo se permite dar “me gusta” una sola vez por usuario en una publicación.

En caso de que un usuario intente dar “me gusta”, mas de una vez a una misma publicación, la aplicación va a ignorarlo.

- Comentar una publicación:

			id_coment	id_usu	nom_usu	id_pub	comentario	fecha	
<input type="checkbox"/>				3	4	usuario01	8	Quantas horas se tiene que trabajar?	2019-12-18 09:36:08
<input type="checkbox"/>				4	7	usuario03	13	hola	2019-12-18 14:44:09
<input type="checkbox"/>				5	7	usuario03	13	adeu	2019-12-18 14:44:20
<input type="checkbox"/>				6	4	usuario01	11	de que trata la oferta?	2019-12-19 07:52:50
<input type="checkbox"/>				7	4	usuario01	4	prueba de notificacion	2019-12-20 07:16:46
<input type="checkbox"/>				8	4	usuario01	4	Prueba notificacion2	2019-12-20 07:25:22
<input type="checkbox"/>				9	9	test01	2	Comentario para probar	2019-12-20 08:44:54
<input type="checkbox"/>				10	9	test01	8	Me gustaria ganar ese dinero, que tengo que hacer?	2019-12-20 08:46:52
<input type="checkbox"/>				11	8	nophotouser	2	que casa tan chula	2019-12-20 09:34:54
<input type="checkbox"/>				12	4	usuario01	4	hola	2019-12-21 12:25:46

Imagen 49.

En esta prueba podemos ver como en la base de datos no tiene ningún comentario "Imagen 49".



Imagen 50.



Imagen 51.

				id_coment	id_usu	nom_usu	id_pub	comentario	fecha
<input type="checkbox"/>				3	4	usuario01	8	Quantas horas se tiene que trabajar?	2019-12-18 09:36:08
<input type="checkbox"/>				4	7	usuario03	13	hola	2019-12-18 14:44:09
<input type="checkbox"/>				5	7	usuario03	13	adeu	2019-12-18 14:44:20
<input type="checkbox"/>				6	4	usuario01	11	de que trata la oferta?	2019-12-19 07:52:50
<input type="checkbox"/>				7	4	usuario01	4	prueba de notificacion	2019-12-20 07:16:46
<input type="checkbox"/>				8	4	usuario01	4	Prueba notificacion2	2019-12-20 07:25:22
<input type="checkbox"/>				9	9	test01	2	Comentario para probar	2019-12-20 08:44:54
<input type="checkbox"/>				10	9	test01	8	Me gustaria ganar ese dinero, que tengo que hacer?	2019-12-20 08:46:52
<input type="checkbox"/>				11	8	nophotouser	2	que casa tan chula	2019-12-20 09:34:54
<input type="checkbox"/>				12	4	usuario01	4	hola	2019-12-21 12:25:46
<input type="checkbox"/>				13	12	usu01	18	Comentario de prueba	2020-01-08 12:30:50

Imagen 52.

Así que vamos a comentar una publicación que no tiene comentarios "Imagen 50" y vemos como instantáneamente sale el comentario debajo de la publicación "Imagen 51" y se guarda a la base de datos "Imagen 52".

Todos los comentarios se muestran en orden inverso al comentario, así que el primer comentario que se mostrara es el primer comentario y el ultimo, el ultimo comentario.

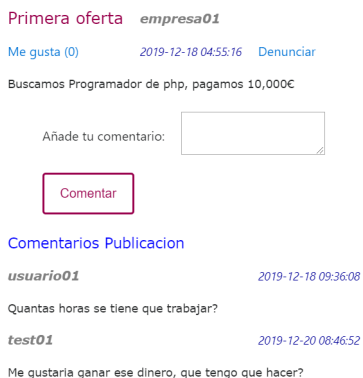


Imagen 53.

En la "Imagen 53" podemos ver un ejemplo de mas de un comentario, donde todos los comentarios de esa misma publicación se muestran por orden de publicación.

- Denunciar una publicación:

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0.0014 segundos.)

```
SELECT * FROM `denuncias_t`
```

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

	id_denu	id_usu	id_publi	id_owner_pub	motivo	fecha	pendiente	control
<input type="checkbox"/>	7	4	13	7	asdas	2019-12-19 03:56:24	2	conducta poco apropiada
<input type="checkbox"/>	8	4	12	7	falsa alarma	2019-12-18 16:05:38	1	falsa alarma1
<input type="checkbox"/>	9	3	2	4	prueba	2019-12-19 03:42:41	0	

Imagen 54.

En esta prueba vamos a denunciar una publicación, pero primero vamos a comprobar el estado de la base de datos “Imagen 54”.



Imagen 55.

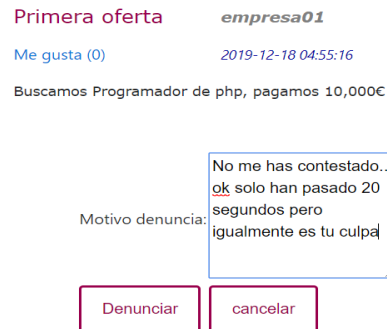


Imagen 56.

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0.0007 segundos.)

```
SELECT * FROM `denuncias_t`
```

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

	id_denu	id_usu	id_publi	id_owner_pub	motivo	fecha	pendiente	control
<input type="checkbox"/>	7	4	13	7	asdas	2019-12-19 03:56:24	2	conducta poco apropiada
<input type="checkbox"/>	8	4	12	7	falsa alarma	2019-12-18 16:05:38	1	falsa alarma1
<input type="checkbox"/>	9	3	2	4	prueba	2019-12-19 03:42:41	0	
<input type="checkbox"/>	10	9	8	3	No me has contestado... ok solo han pasado 20 seg.	2019-12-20 08:49:32	0	

Imagen 57.

Una vez comprobado el estado de la base de datos podemos proceder a denunciar una publicación “Imagen 55”, donde nos lleva a otra página para poner la denuncia “Imagen 56”. Una vez rellenamos el formulario de denuncia y le damos a denunciar aparecerá en la base de datos de denuncias “Imagen 57”, y esta denuncia quedara a la espera de que un administrador la verifique.

- Ocultar o mostrar publicaciones nuestras:

+ Opciones		id_publici	id_usu	nombre_usu	title_pub	desc_pub	tipo_pub	id_foto	fecha	activa	admin_control
<input type="checkbox"/>	Editar	18	12	usu01	Publicacion con imagen	Publicacion con imagen	0	30	2020-01-08 11:40:01	1	0

Imagen 58.

Three screenshots of a web application showing different publication states:

- Imagen 58:** Shows a publication titled "Publicacion con imagen" by user "usu01". A yellow highlight is on the "Ocultar publicacion" button.
- Imagen 60:** Shows the same publication, but with a yellow highlight on the "Mostrar publicacion" button.
- Imagen 61:** Shows a different publication titled "oferta con foto" by "empresa01". A red circle highlights the "Denunciar" button.

Imagen 59.

Imagen 60.

Imagen 61.

+ Opciones		id_publici	id_usu	nombre_usu	title_pub	desc_pub	tipo_pub	id_foto	fecha	activa	admin_control
<input type="checkbox"/>	Editar	18	12	usu01	Publicacion con imagen	Publicacion con imagen	0	30	2020-01-08 12:44:30	0	0

Imagen 62.

En esta prueba tengo que remarcar que solo es posible si eres el propietario de la publicación, en la "Imagen 61" podemos observar como no nos aparece el botón de mostrar o ocultar una publicación, en caso de ser el propietario de la publicación aparece un botón que te permite ocultar "Imagen 59" o mostrar "Imagen 60".

Cuando esta visible para todo el mundo quiere decir que en la base de datos tiene el campo activa a 1 "Imagen 58" y si esta oculto, tiene el campo activa igual a 0 "Imagen 62", para que sea mas fácil diferenciar las publicaciones que no están activas se muestran con un borde rojo y una opacidad al 70% "Imagen 60".

- Chat:

id_chat	id_usu	nombreUsuario	id_remitente_usu	nombreRemi	mensaje	fecha
1	4	usuario01	3	empresa01	hola	2019-12-19 08:33:18
6	3	empresa01	4	usuario01	que tal?	2019-12-19 08:54:30
11	4	usuario01	3	empresa01	muy bien y tu?	2019-12-19 10:15:33

Imagen 63.

Primero vemos como no hay ningún mensaje entre *test01* y *usuario01* “Imagen 63”.



Imagen 64.

Para poder hablar con otro usuario hay dos formas, hablar con una persona que sigues (chats → *usuario01*) o (Buscar persona → *usuairoXXX*). Para esta prueba hemos mandado un mensaje a un usuario al cual seguimos “Imagen 64”.

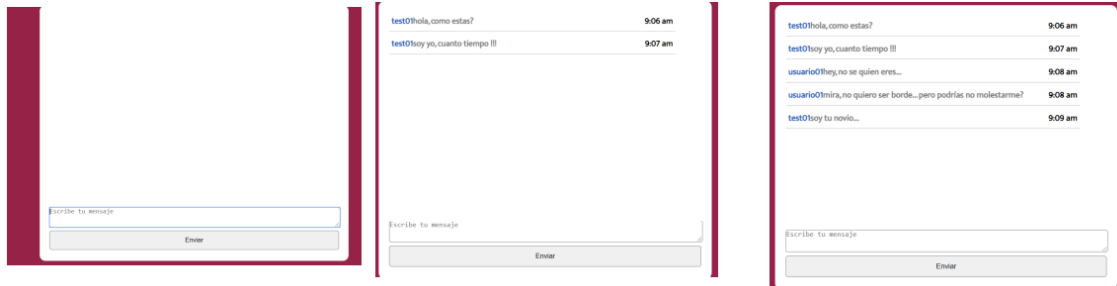


Imagen 65.

En esta secuencia de imágenes podemos ver una pequeña conversación entre dos usuarios “Imagen 65”. (el receptor tiene que actualizar la página para ver el mensaje).

id_chat	id_usu	nombreUsuario	id_remitente_usu	nombreRemi	mensaje	fecha
1	4	usuario01	3	empresa01	hola	2019-12-19 08:33:18
6	3	empresa01	4	usuario01	que tal?	2019-12-19 08:54:30
11	4	usuario01	3	empresa01	muy bien y tu?	2019-12-19 10:15:33
12	9	test01	4	usuario01	hola, como estas?	2019-12-20 09:06:50
13	9	test01	4	usuario01	soy yo, cuanto tiempo III	2019-12-20 09:07:05
14	4	usuario01	9	test01	hey, no se quien eres...	2019-12-20 09:08:22
15	4	usuario01	9	test01	mira, no quiero ser borde... pero podrias no moles...	2019-12-20 09:08:40
16	9	test01	4	usuario01	soy tu novio...	2019-12-20 09:09:01

Imagen 66.

y finalmente podemos ver como guardamos el chat entre los usuarios “Imagen 66”.

- Creación de carpetas dinámicas para imágenes:

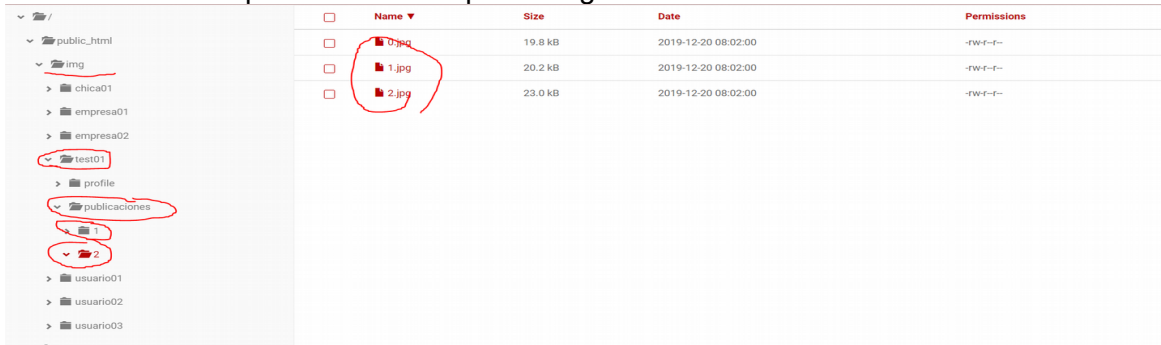


Imagen 67.

Imagen 67: En esta imagen se muestra la estructura interna cuando un usuario crea una publicación. `Img` → `Usuario` → `Publicaciones` → carpetas numeradas donde dice la publicación que es.

Las imágenes se guardan dentro del directorio como números, por que en el momento de mostrar la publicación (en este caso) del usuario `img` → `test01` → `publicaciones` → `2` → `0,1,2` imágenes, calculamos que en la carpeta `"img/test01/publicaciones/2` → contiene 3 imágenes así que desde código hacemos un bucle desde `0` a `< 3` de tal forma que mostraremos las imágenes `0`, `1` y `2`.

- Modificar imagen Perfil de un usuario.

	id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
	1	Admin_dios	Admin_dios	Admin_dios@hotmail.com	1	28		100	img/chica01/profile/1.jpg
	3	empresa01	empresa01	empresa01@hotmail.com	5	50	Somos los mejores	40	img/empresa01/profile/1.jpg
	4	usuario01	usuario01	usuario01@hotmail.com	3	25	usuario01	40	img/usuario01/profile/1.jpg
	6	usuario02	usuario02	usuario02@hotmail.es	3	25	usuario02	40	img/usuario02/profile/1.jpg
	7	usuario03	usuario03	usuario03@hotmail.com	3	35		0	img/no_photo.jpg
	8	nophotouser	nophotouser	nophotouser@hotmail.com	3	28	No tengo foto	40	img/no_photo.jpg
	9	test01	test01	test01@hotmail.com	3	25	si tengo foto	40	img/test01/profile/1.jpg
	10	empresa02	empresa02	empresa02@hotmail.com	5	59	Empresa que hace cosas	40	img/empresa02/profile/1.jpg

Imagen 68.



Imagen 69.

Para esta prueba he escogido un usuario que no tiene foto de perfil "Imagen 68" y "Imagen 69", para ver todo el proceso de creación de carpeta de imágenes para un usuario, y como se muestra la imagen.

Imagen 70.

Así que vamos a poner una imagen al usuario *nophotouser* "Imagen 70", y actualizar la pantalla para ver como se ha insertado la imagen "Imagen 71".

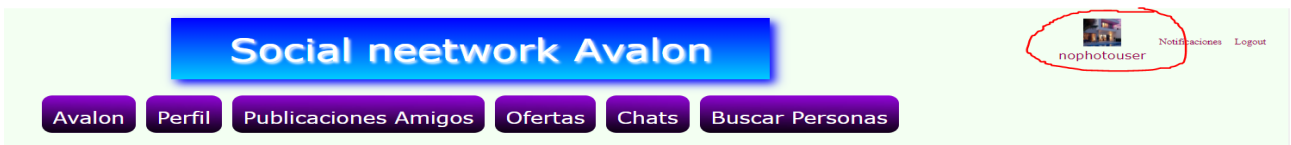


Imagen 71.

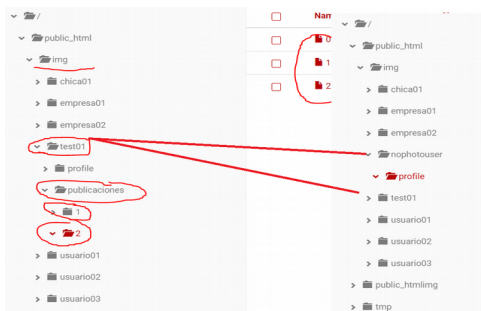


Imagen 72

Imagen 73.

También comprobamos que este usuario no tenia carpeta de imágenes "Imagen 72" y después de insertar la imagen se ha creado la carpeta para este usuario "Imagen 73".

+ Opciones		id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
		1	Admin_dios	Admin_dios	Admin_dios@hotmail.com	1	28		100	img/chica01/profile/1.jpg
		3	empresa01	empresa01	empresa01@hotmail.com	5	50	Somos los mejores	40	img/empresa01/profile/1.jpg
		4	usuario01	usuario01	usuario01@hotmail.com	3	25	usuario01	40	img/usuario01/profile/1.jpg
		6	usuario02	usuario02	usuario02@hotmail.es	3	25	usuario02	40	img/usuario02/profile/1.jpg
		7	usuario03	usuario03	usuario03@hotmail.com	3	35		0	img/no_photo.jpg
		8	nophotouser	nophotouser	nophotouser@hotmail.com	5	28		40	img/nophotouser/profile/3.jpg
		9	test01	test01	test01@hotmail.com	3	25	si tengo foto	40	img/testo1/profile/1.jpg
		10	empresa02	empresa02	empresa02@hotmail.com	5	59	Empresa que hace cosas	40	img/empresa02/profile/1.jpg

Imagen 74.

y finalmente podemos verificar que la imagen ha sido insertada en la base de datos con la ruta que hemos creado auto-matemáticamente “Imagen 74”.

- ofertas

Cuando entramos en ofertas aparecen las publicaciones que son ofertas. Estas publicaciones solo las pueden publicar los usuarios que son empresas, marcando un checkbox para que esa publicación sea una oferta, en caso de no marcar el checkbox sera tratada como una publicación normal.

En este apartado se puede filtrar por todas las ofertas o por las ofertas de tus amigos/followers.

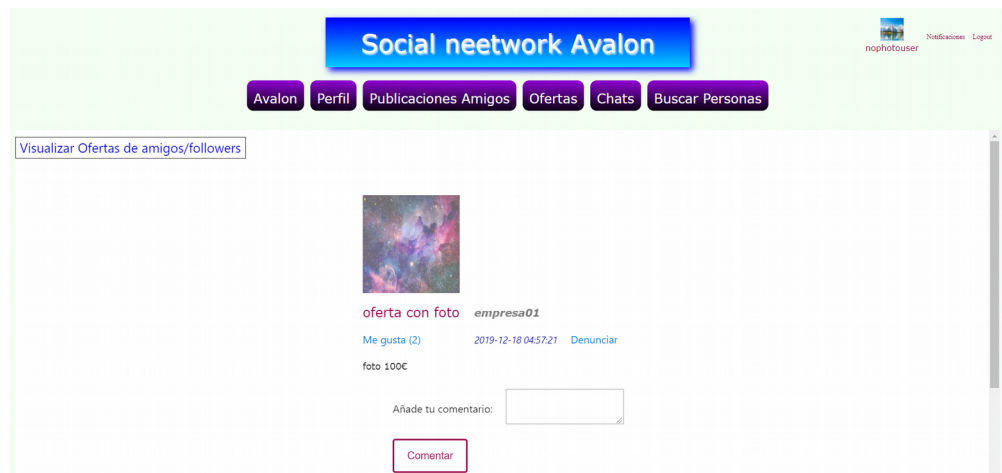


Imagen 75.

El funcionamiento es exactamente igual que el de publicaciones, puedes dar “me gusta”, comentar, o denunciar, simplemente se muestran en otra parte de la aplicación.

- **Notificaciones:**

Lo primero que tenemos que tener en cuenta al hablar de notificaciones es que las denuncias no se notifican, pero los “me gusta” o los “comentarios” si. Cuando entramos en las notificaciones podemos ver de forma inmediata las notificaciones vistas anteriormente o las que no.



Imagen 76.

En la “**Imagen 76**” podemos ver que los 5 primeros elementos se ven perfectamente y que los 4 siguientes elementos se ven con una opacidad del 70%, esa diferencia que podemos ver a simple vista es la que nos dice si una notificación la hemos visto anteriormente o no.

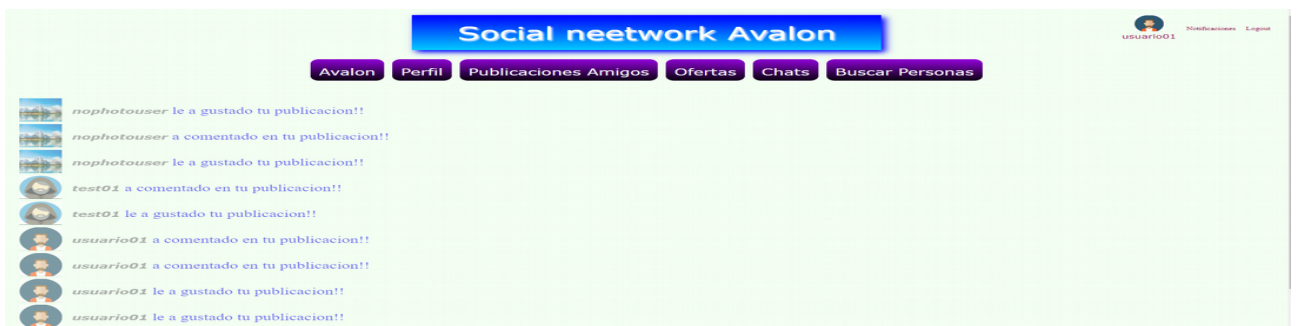


Imagen 77.

Una vez sales de las notificaciones y vuelves a entrar se marcan en la base de datos como vistas y luego cuando vuelves a entrar a la zona de notificación se ven todas como en visto “**Imagen 77**”.



Imagen 78.

Si el usuario decide presionar sobre cualquier notificación “**Imagen 78**” esta lo enviara a ver de que se trata la notificación, mostrando la publicación que toca.



Imagen 79.

Entonces después de presionar sobre una notificación se muestra la publicación que hace referencia dicha notificación “Imagen 79”.

- Control panel

Para acceder a esta parte del aplicativo no hay ningún botón que nos lleve al control panel, así que lo tenemos que escribir en la URL a mano. Si un usuario que no tiene permisos intenta entrar al Control Panel entonces es expulsado automáticamente y tiene que volver a hacer login. El hecho de expulsarlo de la página web es para intimidar, hacer pensar al usuario que se tiene en cuenta y así lograr que no lo pruebe mas veces.

<https://tfgdesarrolloweboscar.000webhostapp.com/controlPanel.php>

- Prueba con usuario no permitido:

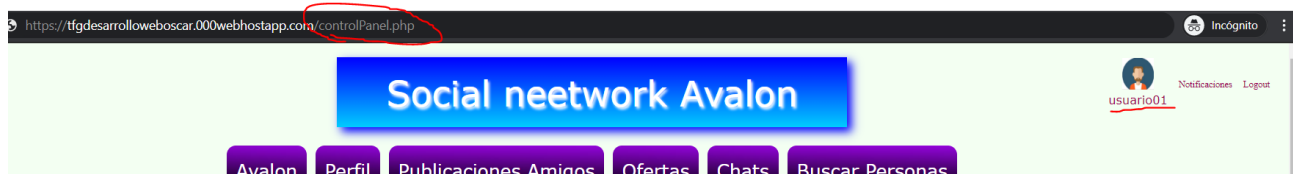


Imagen 80.

Actualmente estamos con usuario común “usuario01” así que probamos de escribir en la URL controlPanel.php “Imagen 80”.

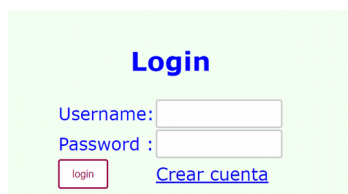


Imagen 81.

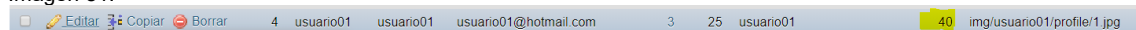
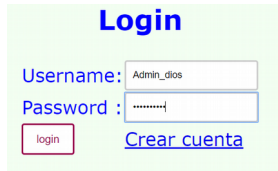


Imagen 82.

Como este usuario no tiene permisos nos enviá directamente a la página de login “Imagen 81”. Para poder acceder a la zona de administrador tienes que tener un usuario con nivel_usu < 39, y “usuario01” tiene nivel_usu = 40 “Imagen 82”

- Prueba con usuario permitido:



Login

Username:

Password:

[Crear cuenta](#)

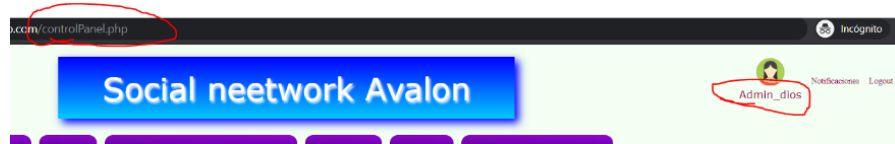
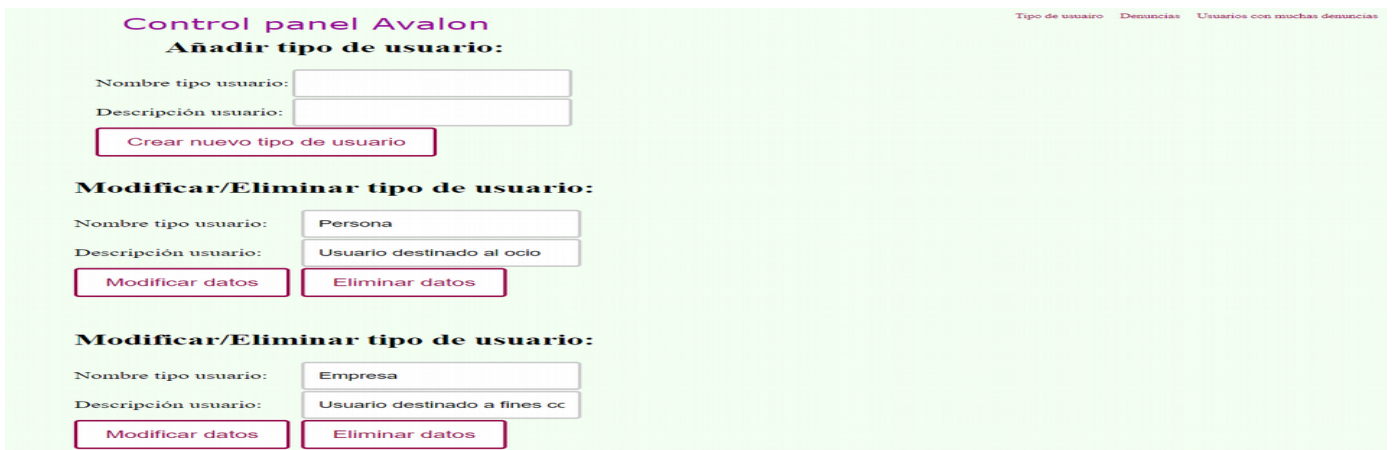


Imagen 83.

Imagen 84.

Para esta prueba si que utilizamos un usuario con nivel suficiente para entrar en el apartado de administradores “Imagen 83”, así que volvemos a escribir en la URL controlPanel.php y verificamos que podemos entrar “Imagen 84”.



Control panel Avalon Tipo de usuario Denuncias Usuarios con muchas denuncias

Añadir tipo de usuario:

Nombre tipo usuario:

Descripción usuario:

Modificar/Eliminar tipo de usuario:

Nombre tipo usuario:

Descripción usuario:

Modificar/Eliminar tipo de usuario:

Nombre tipo usuario:

Descripción usuario:

Imagen 85.

y como es un usuario permitido nos deja entrar al control Panel “Imagen 85”.

Control Panel, nos permite crear/modificar/eliminar tipo de usuario, administrar denuncias de los usuarios y administrar los usuarios problemáticos.

Los usuarios problemáticos son aquellos que tienen mas de diez denuncias.

- **Crear** nuevo tipo Usuario:



Imagen 86.

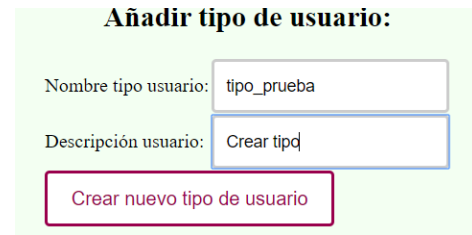


Imagen 87.



Imagen 88.

En la “Imagen 86” podemos ver que solo hay 5 tipos de persona, y en la “Imagen 87” creamos un nuevo tipo. Después de crearlo podemos verlo en la base de datos “Imagen 88”.

- **Modificar** nuevo tipo Usuario:

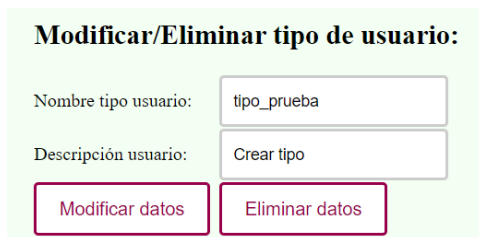


Imagen 89.

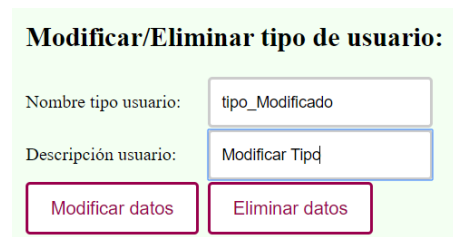


Imagen 90.

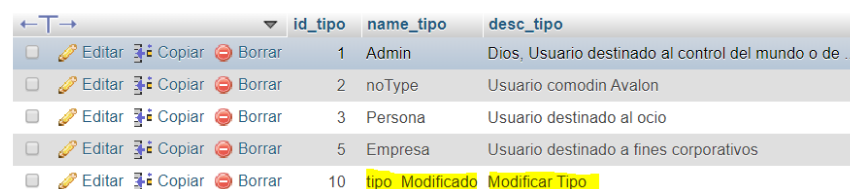


Imagen 91.

En esta prueba podemos ver como tenemos el tipo que hemos creado anteriormente “Imagen 88” y lo podemos modificar o eliminar “Imagen 89”. Así que vamos a modificar el tipo de persona “Imagen 90” y en la base de datos podemos ver como se ha modificado “Imagen 91”.

- Asignamos un usuario con este tipo para luego eliminar-lo:

	id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
	12	usu01	usu01	usugsgssh	3	24	hshs	40	img/usu01/profile/1.jpg

Imagen 92.

Para realizar esta prueba vamos a coger el usuario "usu01" "Imagen 92" que actualmente tiene el tipo de usuario = 3 y le vamos a asignar el nuevo tipo de usuario creado en la prueba anterior "Imagen 91".



Imagen 93.

Así que vamos al usuario "usu01" y vamos a la zona de modificar datos personales "Imagen 93".

Formulario de edición de usuario:

Username: usu01
Password:
re-Password:
Email: usugsgssh
Edad: 24
Estado: no funciona
Tipo usuario: tipo_Modificado
Foto perfil: Seleccionar archivo Ningún archivo seleccionado

Modificar datos usuario

Imagen 94.

Cambiamos el tipo de usuario al nuevo tipo que hemos creado anteriormente "Imagen 94".

	id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
	12	usu01	usu01	usugsgssh	10	24	no funciona	40	img/usu01/profile/1.jpg

Imagen 95.

Una vez cambiado por la página, podemos verificarlo en la base de datos "Imagen 95".

Modificar/Eliminar tipo de usuario:

Nombre tipo usuario: tipo_Modificado
Descripción usuario: Modificar Tipo

Modificar datos Eliminar datos

Imagen 96.

Una vez tenemos ya un usuario con el nuevo tipo, vamos a eliminarlo "Imagen 96".

	id_usu	name_usu	password	email_usu	tipo_usu	edad	estado	nivel_usu	ruta_foto_p
	12	usu01	usu01	usugsgssh	2	24	no funciona	40	img/usu01/profile/1.jpg

Imagen 97.

Una vez eliminado el tipo usuario "10" podemos ver que el usuario que tenia ese tipo ahora tiene el tipo usuario = 2 "Imagen 97".



	id_tipo	name_tipo	desc_tipo
<input type="checkbox"/> Editar Copiar Borrar	1	Admin	Dios, Usuario destinado al control del mundo o de ...
<input type="checkbox"/> Editar Copiar Borrar	2	noType	Usuario comodin Avalon
<input type="checkbox"/> Editar Copiar Borrar	3	Persona	Usuario destinado al ocio
<input type="checkbox"/> Editar Copiar Borrar	5	Empresa	Usuario destinado a fines corporativos

Imagen 98.

y vemos que ahora ya no existe en la base de datos el tipo=10 "Imagen 98".

Cuando eliminamos un tipo de persona, le asignamos a todos los usuarios que tengan el tipo eliminado el tipo 2 que seria como ser persona.

No le asignamos persona (tipo = 3) por que si lo eliminamos algún día nos quedaríamos sin ningún tipo que asignar, así que tenemos el tipo 2 que es un estado comodín que actuá en todos los efectos como el de una persona.

Para el usuario es un cambio que pasa transparente-mente y si ese usuario decidiera cambiar algún dato personal, se le asignaría directamente el tipo de persona 2 (a no ser que el decida cambiarlo a empresa).

- Denuncias:

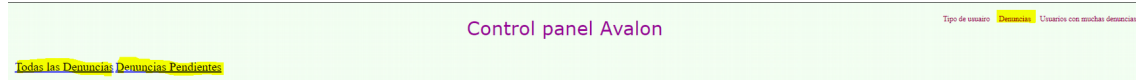


Imagen 99.

Para ir a las denuncias tenemos que entrar a Panel Control → denuncias “Imagen 99” en este apartado se puede filtrar por “Todas las denuncias” o “Denuncias pendientes”.
Todas las denuncias: Muestra todas independientemente del estado.
Denuncias pendientes: Muestra solo las denuncias que no han sido revisadas.

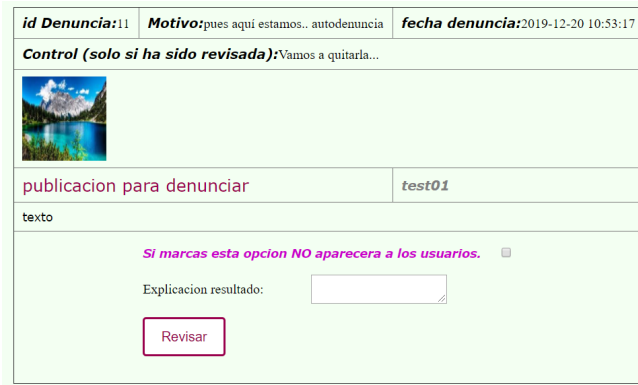


Imagen 100.



Imagen 101.

En la “Imagen 100” podemos ver que ya ha sido revisada, ya que tiene el apartado control activo, en cambio en la “Imagen 101” no ha sido controlada ya que no tiene el apartado control activo.

Como administradores podemos decidir si esta denuncia es buena o no, marcando la casilla y escribiendo una razón por la cual la marcamos. Si marcamos la casilla esta publicación dejara de ser visible para todos los usuarios, incluido el propietario de la publicación.

Para poder probar esta parte vamos a coger el usuario “test01” y vamos a crear varias publicaciones, y nos vamos a denunciar una (no importa que usuario sea) para que aparezca en el apartado de administración de denuncias.

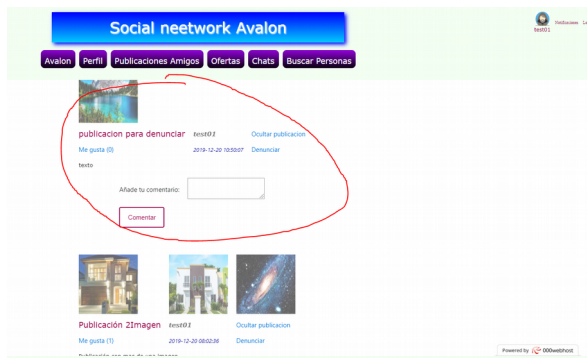


Imagen 102.

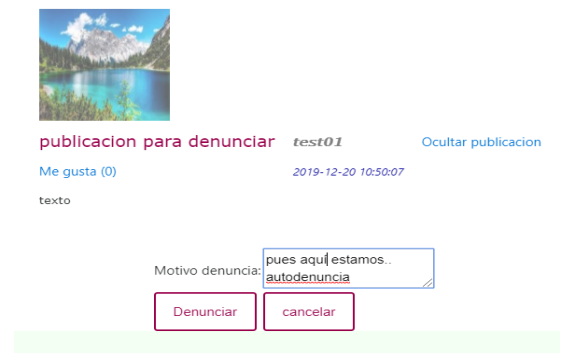


Imagen 103.

Ya hemos creado la publicación para denunciarnos “Imagen 102” y también hemos realizado la denuncia “Imagen 103”.



Imagen 104.

Volvemos a entrar a la parte de administrador con un usuario valido y vamos a catalogar a esa denuncia como valida "Imagen 104".

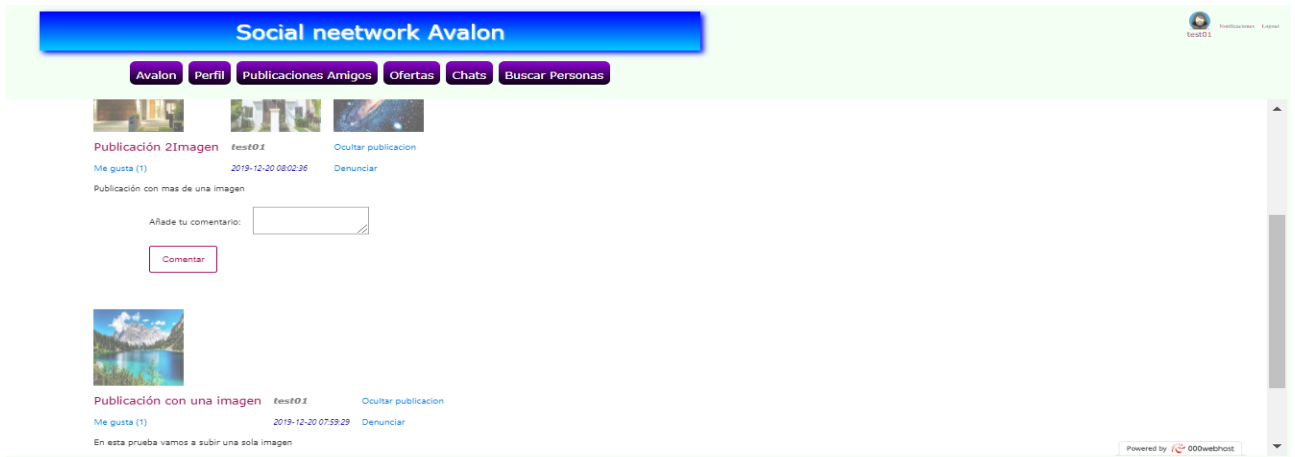


Imagen 105.

Una vez un administrador a catalogado una denuncia como denuncia valida ya no se va a mostrar nunca mas en la página web, ya seas el propietario o no. "Imagen 105"

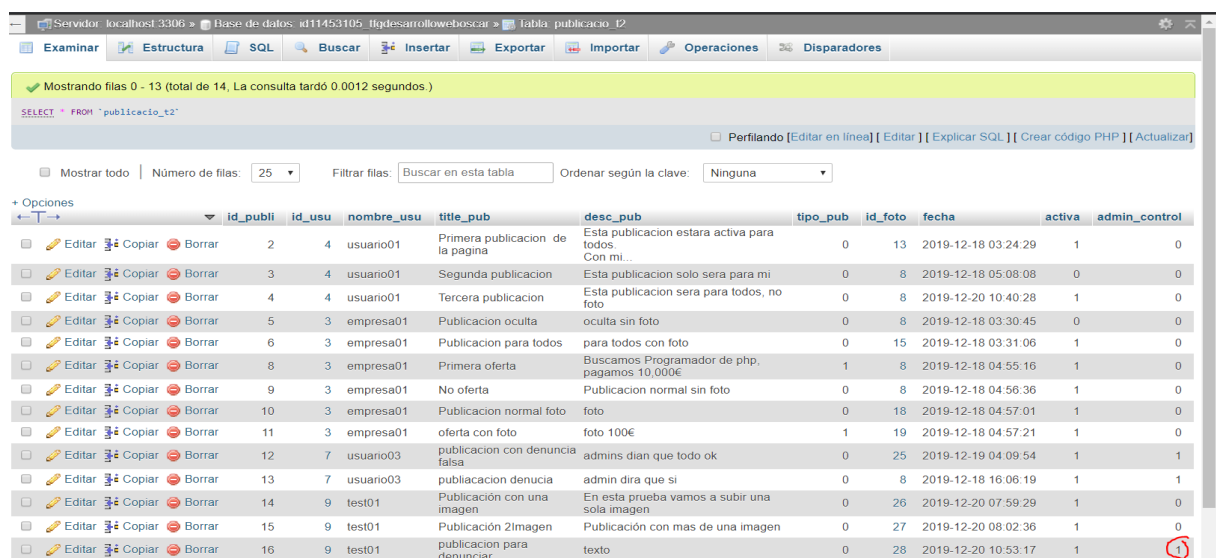


Imagen 106.

Como no se muestra en la página web, quiere decir que en la base de datos la publicación tiene `admin_control = 1` “Imagen 106”.



	id_denu	id_usu	id_publi	id_owner_pub	motivo	fecha	pendiente	control
	7	4	13	7	asdas	2019-12-19 03:56:24	2	conducta poco apropiada
	8	4	12	7	falsa alarma	2019-12-18 16:05:38	1	falsa alarma1
	9	3	2	4	prueba	2019-12-19 03:42:41	0	
	10	9	8	3	No me has contestado... ok solo han pasado 20 seg	2019-12-20 08:49:32	0	
	11	9	16	9	pues aqui estamos... autodenuncia	2019-12-20 10:53:17	2	Vamos a quitarla

Imagen 107.

Y en la base de datos de denuncias tenemos esa misma denuncia con un pendiente = 2, que quiere decir que ha sido validada de y vetada de la página y que tiene el campo control informado diciendo la razón por la cual esta publicación ya no se muestra en la página web “Imagen 107”.

Si Pendiente es = 0 quiere decir que no ha sido revisada, si es igual a 1 quiere decir que ha sido revisada y que se ha decidido que esa publicación cumple con todas las normas de la página web.

- Usuarios con muchas denuncias:

En este apartado aparecerán los usuarios que tienen mas de 10 denuncias (*para realizar la prueba lo he puesto a 1*)

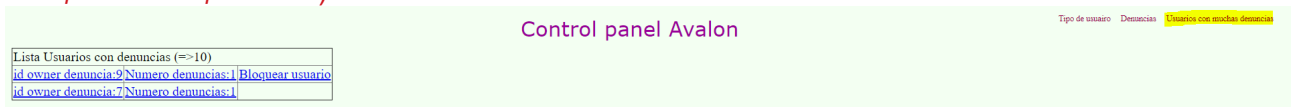


Imagen 108.

Imagen 108: Muestra como se muestran los usuarios con muchas denuncias. También nos muestra que id_owner_denuncia y numero denuncias son links que nos muestran todas las denuncias de ese mismo usuario "Imagen 109".

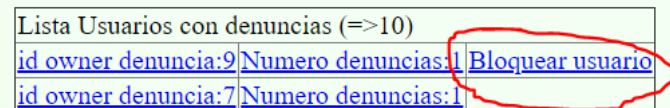


Imagen 109.

Imagen 110.

Imagen 110: Bloquear al usuario de la página. (Al bloquear a un usuario, se marcan todas sus publicaciones como bloqueadas y dejan de existir para la página y el usuario bloqueado no podrá volver a hacer login con ese usuario).

Pero antes de bloquear-lo vamos a mirar el estado de sus publicaciones "Imagen 111".

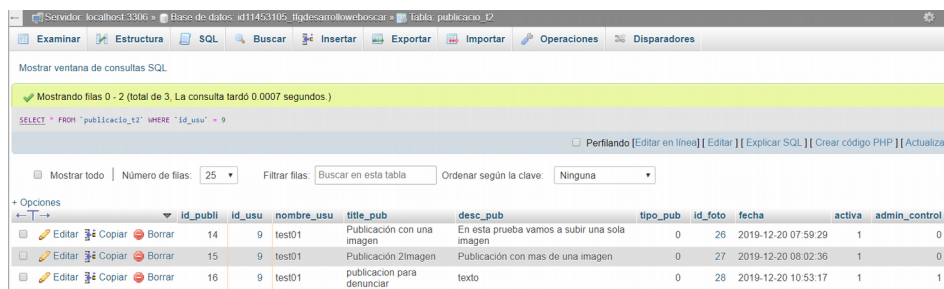


Imagen 111.

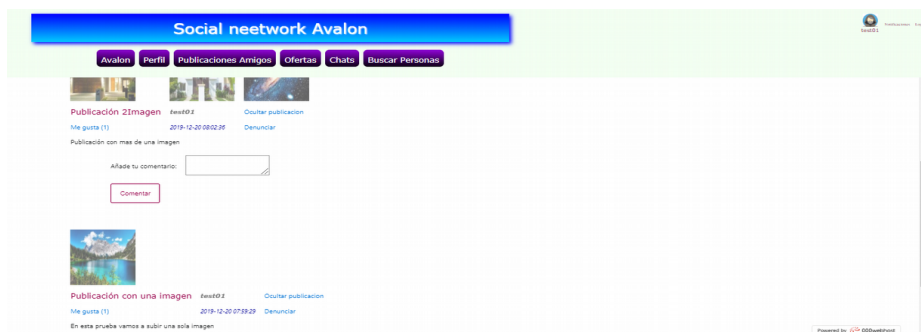


Imagen 112.

Imagen 112: Solo una de las publicaciones bloqueada por el administrador, Si vamos a su usuario podemos ver 2 publicaciones.



Imagen 113.



Imagen 114.

Imagen 113 y 114: Y si una persona es amigo suyo puede ver las dos publicaciones y le aparece en el chat, y puede ver tus publicaciones.

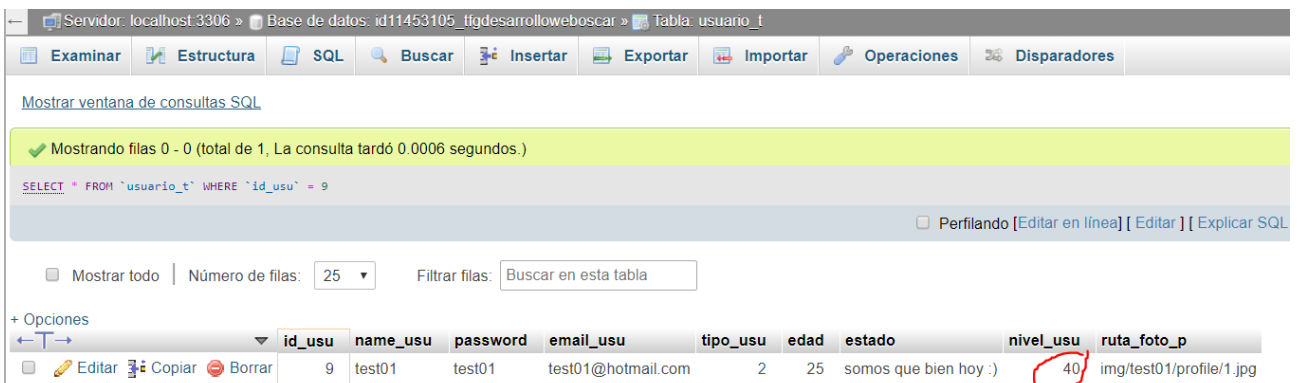


Imagen 115.

En la "Imagen 115" podemos ver que el usuario "test01" tiene nivel_usu = 40.

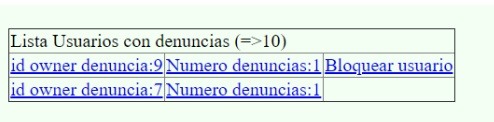


Imagen 116.

Así que ahora vamos a bloquear al usuario "test01" "Imagen 116".



Imagen 117.

Después de bloquear al usuario se le cambia el nivel a 0 "Imagen 117".

+ Opciones		id_publi	id_usu	nombre_usu	title_pub	desc_pub	tipo_pub	id_foto	fecha	activa	admin_control
Editar	Copiar	14	9	test01	Publicación con una imagen	En esta prueba vamos a subir una sola imagen	0	26	2019-12-20 11:14:36	1	1
Editar	Copiar	15	9	test01	Publicación 2Imagen	Publicación con mas de una imagen	0	27	2019-12-20 11:14:36	1	1
Editar	Copiar	16	9	test01	publicacion para denunciar	texto	0	28	2019-12-20 10:53:17	1	1

Imagen 118.

También se han cambiado el estado de admin_control a 1 de todas las publicaciones del usuario "test01" "Imagen 118"

Datos Incorrectos. [Try again!!](#)

Login

Username:

Password:

[Crear cuenta](#)

Imagen 119.

Social network Avalon

Buscar Personas:


[Chatear](#)


[Notificaciones](#)
[Logout](#)

Imagen 120.

Una vez vetado la entrada al usuario "test01" intentamos entrar a la página web con ese usuario y vemos que nos salta error de usuario no correcto "Imagen 119", y también podemos ver que los usuarios que tenían relación con el usuario "test01" les ha dejado de aparecer en su lista de amigos "Imagen 120".

3. Conclusions

El trabajo final me deja con un sabor de boca agri dulce, ya que no he podido finalizar todas las funcionalidades que tenia en mente. En gran parte acabo el proyecto con casi todas las funcionalidades acabadas al 100%, aun que hay un par de funcionalidades que me hubiera gustado perfeccionar ya no están completas al 100%. La primera funcionalidad que me gustaría mejorar es el mostrar las imágenes, ya que me hubiera gustado realizar un slider con ellas, donde el usuario podría ver todas las imágenes de una publicación de forma mucho mas bonita.

Otra funcionalidad que me hubiera gustado hacer como tenia pensado es realizar el chat dinámico en tiempo real, ya que ahora funciona pero el receptor del mensaje tiene que actualizar la pagina, o poder agrupar tus amistades en diferentes grupos, para poder darles permisos para ver según que publicaciones. Otra funcionalidad que me hubiera gustado finalizar es la de estados. (aun que la aplicación ya esta preparado para ello, ahora no puedes ver los estados de un usuario).

Pero también me voy con un buen sabor de boca, ya que he podido realizar las fotos dinámicas en tiempo real, también las notificaciones, puedes ver las que no has visto y las que si... y muchas mas funciones que han finalizado como yo lo tenia pensado.

La verdad después de analizar el proyecto, proponer un plan de trabajo, con el calendario y los tiempos teóricos marcados para el correcto funcionamiento del proyecto, me he dado cuenta de lo difícil que es realizar este trabajo de forma eficiente sin salirte del plan de trabajo (ya que he ido añadiendo nuevas funcionalidades en el trabajo), sin pasarte de los tiempos limites y también he visto que un contratiempo puede hacerte cambiar todo el plan de trabajo. Ya que con el chat he tenido muchos problemas y esto ha echo que los diferentes tiempos marcados se vieran afectados, sobretodo el tiempo de diseño. Que para mi gusto no esta ni de lejos como yo quería.

5. Bibliografía

[1] PHP: Shen, Z. (2018). A Gentler Introduction to PhP and Its Application in Database Programming.

[2] SCRUM : https://books.google.co.kr/books?hl=en&lr=&id=6pZCAwAAQBAJ&oi=fnd&pg=PT9&dq=SCRUM&ots=kbvVUZ8mhX&sig=bmIOiZ124WpNuhBspXKEINctr34&redir_esc=y#v=onepage&q=SCRUM&f=false

Páginas visitadas durante la creación de la página web.

Slideshow – Para mostrar imágenes diatónicamente

w3schools : https://www.w3schools.com/howto/howto_js_slideshow.asp

Chat interactivo – Para chatear de forma interactiva

webslesson: <https://www.webslesson.info/2018/07/live-chat-system-in-php-using-ajax-jquery.html>

Iframes – Para mostrar diferentes páginas web dentro de otra

w3schools: https://www.w3schools.com/html/html_iframe.asp

Php – Conectar con la base de datos

w3schools: https://www.w3schools.com/php/php_mysql_connect.asp