

2016

# Memoria Soundcloud - IW

PRÁCTICA FINAL INGENIERÍA WEB

CRISTÓBAL JESÚS GONZÁLEZ – ALBERTO SAPIÑA MORA – CAMERON MCAINSH

## Índice

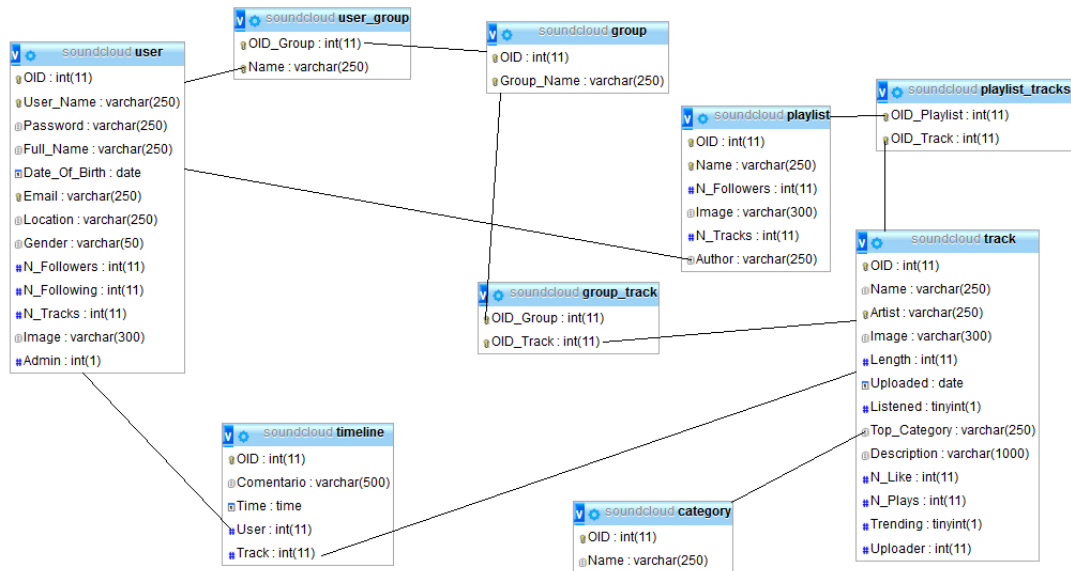
<b>Introducción</b>	2
<b>Diagramas</b>	3
<b>Wireframes y Mockups</b>	4
<i>Mapa del web. Wireframes:</i>	4
<b>Mockups</b>	6
<i>Un esbozo del sitio:</i>	6
<b>Descripción</b>	9
<i>Tecnologías</i>	9
<i>Arquitectura</i>	9
<b>Detalles de implementación</b>	10
<i>Encriptación:</i>	10
<i>Añadir canción a playlist:</i>	10
<i>Buscador:</i>	11
<b>Problemas encontrados</b>	13
<i>Redimensión del menú</i>	13
<i>Grocery CRUD no funcionaba correctamente</i>	13
<i>Paginador del buscador</i>	13
<i>Imágenes en la DB</i>	13
<i>Nombre de la tabla 'group'</i>	13
<b>Mejoras y ampliaciones</b>	14
<i>Ampliaciones:</i>	14
<i>Mejoras:</i>	14
<b>Referencias</b>	15

## Introducción

El trabajo empleado en esta práctica es la creación de una web, en nuestro caso una plataforma web de música en la que usuarios registrados pueden crear playlists y grupos así como añadir canciones a estos, hemos tomado como referencia la web soundcloud, que es una web de subida y reproducción de canciones, en nuestra web hemos implementado la creación de grupos/playlists, logueo de usuarios, añadido de canciones a grupos y playlists, unión a grupos, los cruds de canciones y categorías, así como la búsqueda de contenido en la web.

# Diagramas

## Diagrama Entidad-Relacion:



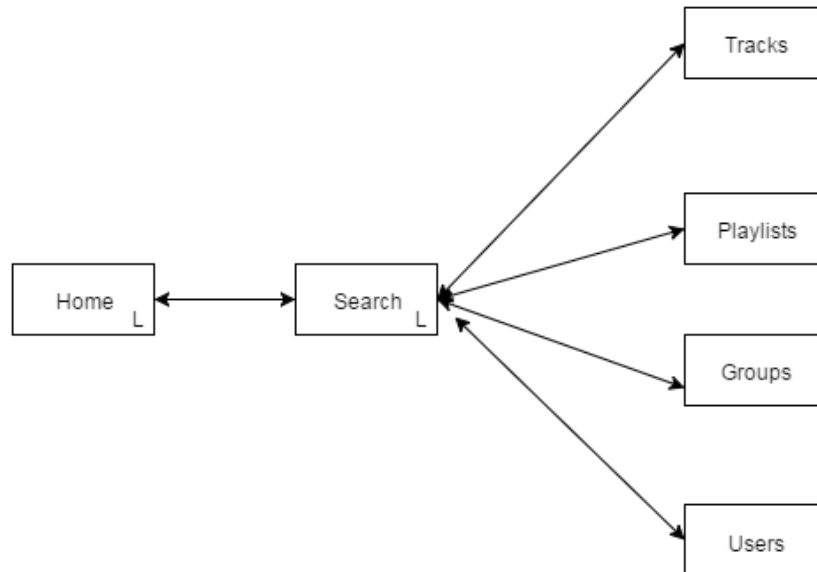
# Wireframes y Mockups

## Mapa del web. Wireframes:

Representación esquemática del sitio web Soundcloud y las páginas que lo forman.

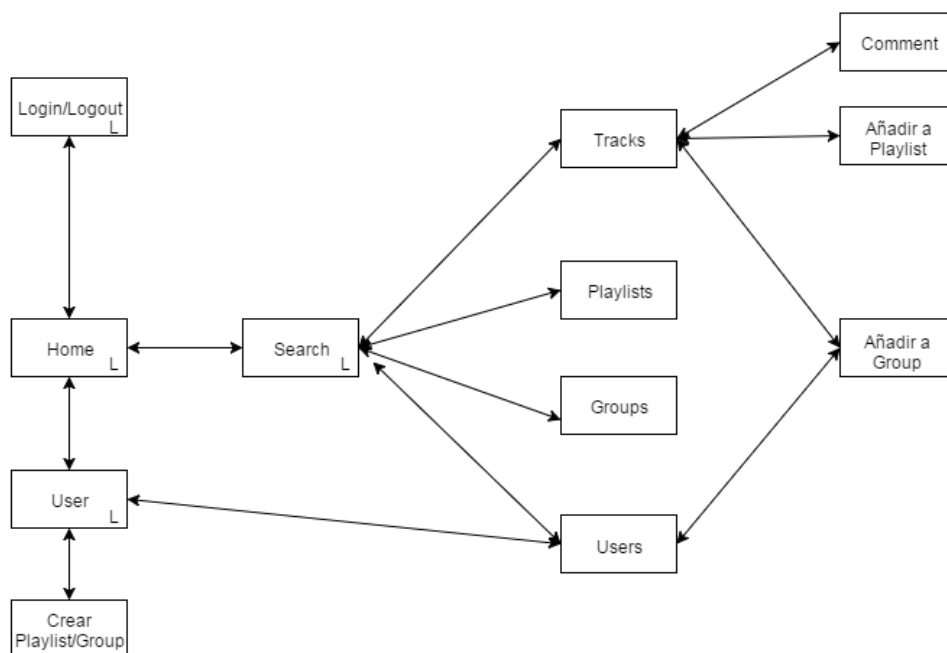
Para el **usuario no registrado** el sitio está limitado hasta el search.

El símbolo “L” significa **Landmark**, una opción visible desde cada view.

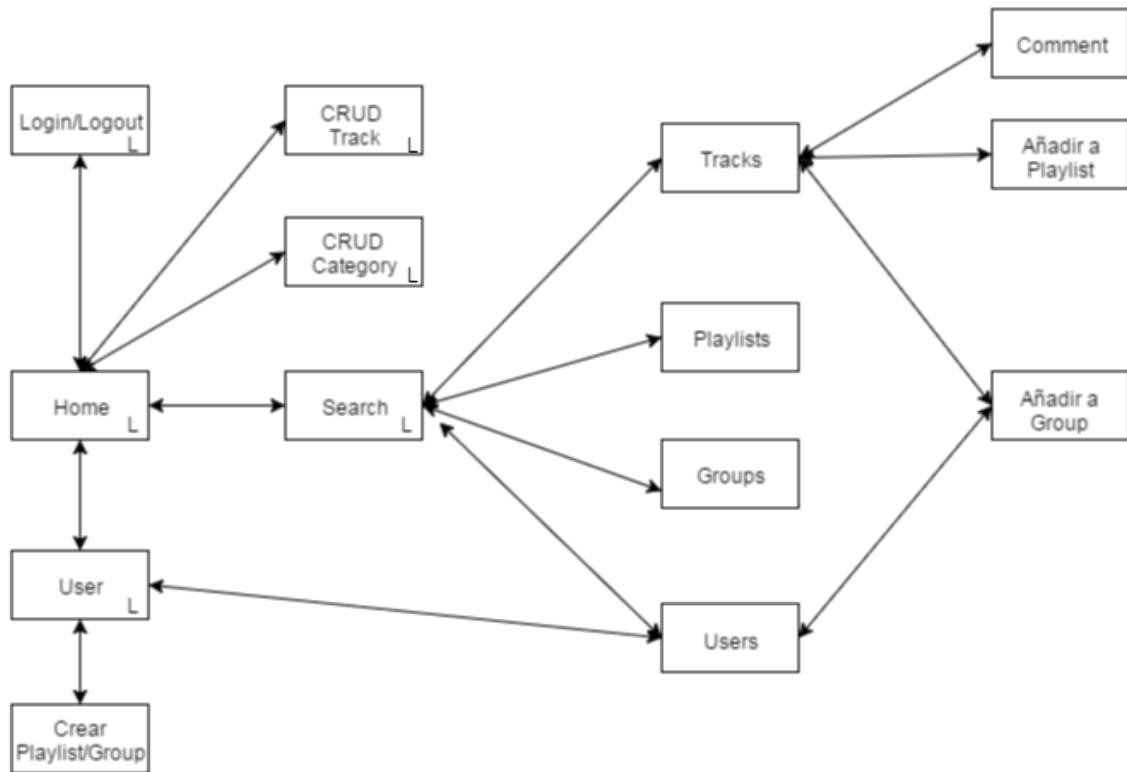


El **usuario normal o usuario registrado** dispone de más opciones disponibles. Después del **Login** el user puede por ejemplo añadir tracks a playlist o añadirse a un group. También

puede comentar un track en el minuto que quiera, crear playlists y grupos.



Un track puede ser subido solo desde **User Admin** y cada track y playlist pertenece a una categoría. Una categoría puede ser creado sólo desde un user con privilegios de **Admin**, entonces el wireframe de una cuenta de **User Admin** es el mismo que un user normal más los

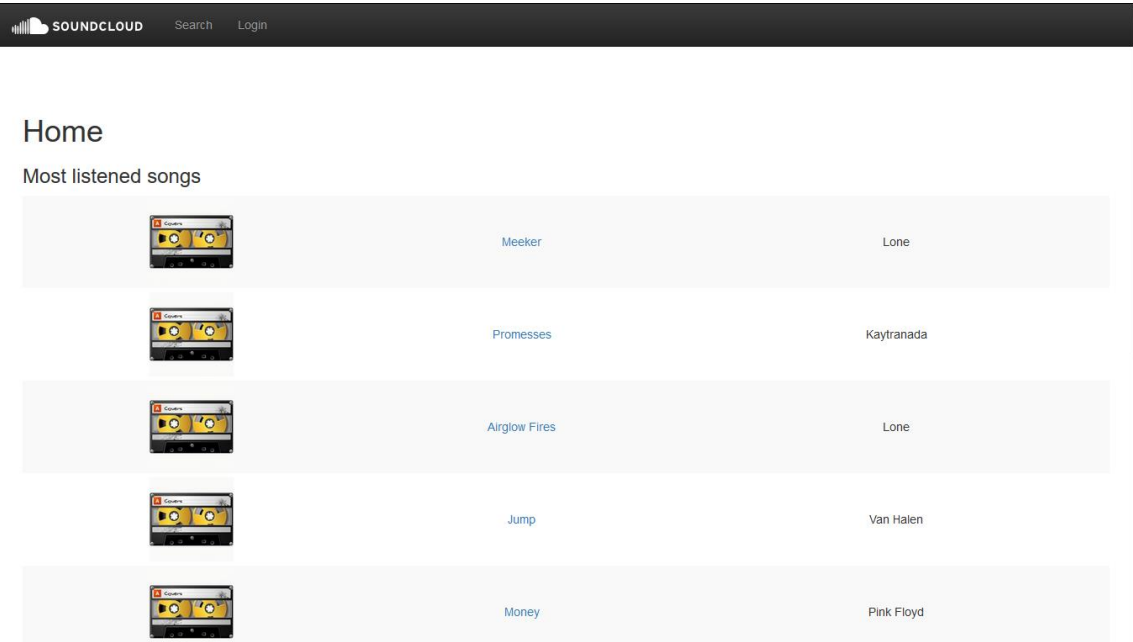


funciones *CRUD Category* y *CRUD Track*.

# Mockups

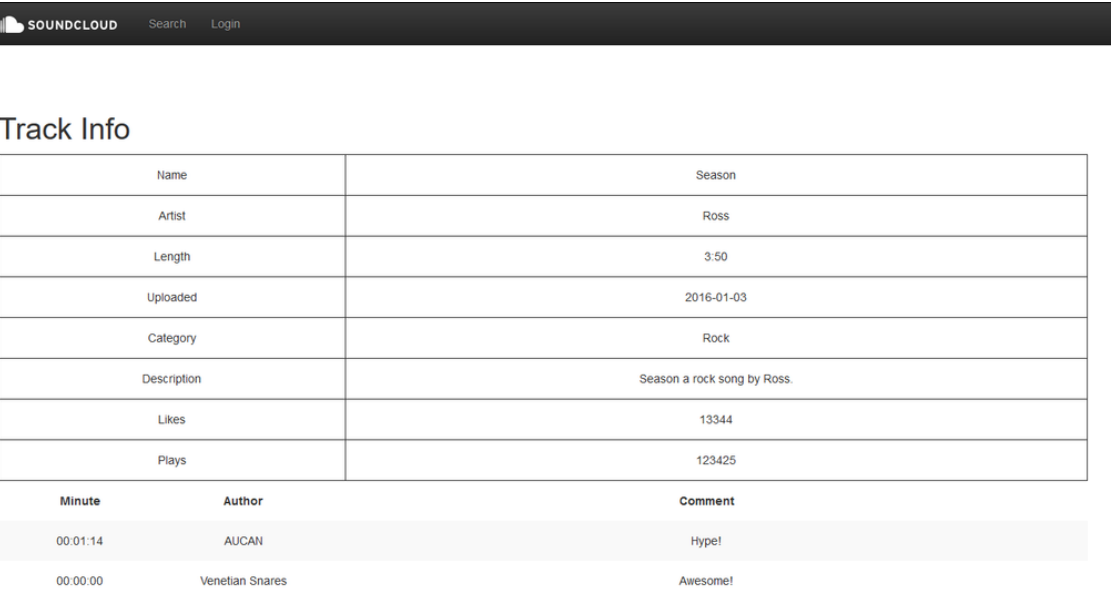
## Un esbozo del sitio:

En la página *Home* se ven las tracks más populares del sitio.



Con un click sobre el nombre del track sale la página de Track Info donde se puede ver todo sobre el track, incluidos los comments de los usuarios en un punto temporal específico.

Si un user quiere crear un Playlist o Group, puede hacerlo desde la página de su perfil (User Info).



Soundcloud	
Search Logout AUCAN	
User Info	
Username	AUCAN
Full name	Marco Ferrè
Date of birth	1990-01-12
Email	aucan@gmail.com
Location	Bergamo
Gender	male
Followers	23
Following	4
Tracks uploaded	4

La función *Search* busca en tiempo real y se puede filtrar por tracks, users, playlists o groups.

SOUNDCLOUD

Search

Login

Buscar...




Tracks

Users

Playlists

Groups

Tracks

Cover	Name	Artist	Length	Top Category	Likes	Plays
	Season	Ross	230	Rock	13344	123425
	Logic ft. Jhene	Louis Fouton	313	Eletronica	12	1234
	Valentino	Dj Snake & Aluna George	123	Pop	543	54326

SOUNDCLOUD

Search

Login

Buscar...




Tracks

Users

Playlists

Groups

Users

Avatar	User	Full Name	Birth date	Email	Location	Gender	Followers	Following	Tracks
	AUCAN	Marco Ferrè	1990-01-12	aucan@gmail.com	Bergamo	male	0	0	0
	Venetian Snares	Vincenzo Twin	1985-12-24	vincennzo.1@gmail.com	Verana	male	0	0	0
	Aphex Jauz	Alfonso Ziq	1990-02-25	aphex.jauz@gmail.com	New York	male	0	0	0



La página de Group Info es donde un user se puede añadir a un dado Group.

En este caso el user “AUCAN” quiera añadirse al Group que se llama “Rock Fan Club”.



# Group Info

Name	Rock fan club
------	---------------

User

Location

AUCAN

Bergamo

Usted ya está unido a este grupo, para salir pulse aquí

Ahora “AUCAN” es el primer y único miembro de “Rock fan club”.

# Descripción

## ***Tecnologías***

- Bootstrap.
- Codeigniter.
- Html 5.
- Javascript.
- Jquery.
- Php.

## ***Arquitectura***

La web se alojará en un servidor web con las siguientes características:

Procesador: E5-2640 v3

Núcleos: 8

Velocidad/núcleo: 2,6 GHz

Memoria RAM: 64 GB

Disco duro: 2 x 1TB SATA

Sistema Operativo: Linux CentOS 7

Correrá sobre un servidor apache con php y mysql.

## Detalles de implementación

### *Encriptación:*

Se ha utilizado una función de encriptación con hash usando la función crypt() de php

```
$cost = 10;

$salt = strtr(base64_encode(mcrypt_create_iv(16, MCRYPT_DEV_URANDOM)), '+', '.');

$salt = sprintf("$2a$%02d$", $cost) . $salt;

$hash = crypt($password, $salt);
```

Primero se define \$cost que es simplemente para la cadena aleatoria que generaremos luego, después se genera el \$salt que es una cadena aleatoria para dificultar las descriptaciones no deseadas, finalmente se genera el \$hash con la contraseña encriptada.

Para la descriptación en el login

```
function verifyPass($passInput, $pass)
{
    if(hash_equals($pass, crypt($passInput, $pass)))
    {
        return true;
    }else{
        return false;
    }
}
```

Los argumentos \$passInput (sería la contraseña que el usuario ha introducido al intentar loguear) y \$pass (sería el hash que tenemos guardado en la base de datos), la función hash\_equals de php lo que hace es comparar el hash de los dos argumentos y en el segundo argumento pasamos la contraseña introducida encriptada con clave el hash anterior.

### *Añadir canción a playlist:*

```
public function add_to_playlist()
{
    $oidTrack = $this->input->post('oidTrack');
    $oidPlaylist = $this->input->post('oidPlaylist');
    $message = "";
    if($this->verify_track($oidTrack, $oidPlaylist))
    {
        $message = "<p style='color:blue'>Track added to the playlist</p>";
        $this->db->query("INSERT INTO playlist_tracks (OID_Playlist, OID_Track) VALUES($oidPlaylist, $oidTrack)");
        $this->Track_info_model->update_N_Tracks_playlist($oidPlaylist);
    }else{
        $message = "<p style='color:blue'>Track already in the playlist</p>";
    }

    $this->index2($oidTrack, $message);
}
```

Con esta función añadimos una canción(track) a una playlist, primero recogemos las variables mediante post, \$oidTrack y \$oidPlaylist son los identificadores de track y playlist respectivamente inicializamos la variable \$mensaje que tan solo se usará para avisar al usuario de la acción final, se comprueba la canción con la función verifyTrack que verifica que la

canción no esté ya en la playlist y, si no está en la playlist se añade la canción mediante un insert, con la función `update_N_Tracks_playlist` simplemente se suma 1 al número de tracks de la playlist, finalmente se llama a `index2` que recarga la página mostrando además el mensaje.

## Buscador:

Para el buscador hemos utilizado un campo del texto, en el cual hemos capturado el evento `keyup` (soltar tecla del teclado) para realizar la búsqueda en tiempo real:

```
$(document).on("ready", function(){
    buscarTrack();
    buscarUsers();
    buscarGroups();
    buscarPlaylist();
    showTracks();
    //Evento para el campo de texto que hace de buscador
    $("#buscar").keyup(function(){
        buscar();
    });
```

La función `buscar()` llama a otras cuatro funciones (una por cada filtro que tiene el buscador) realiza una petición Ajax al servidor para actualizar la tabla de la vista con los nuevos resultados:

```
function buscarTrack(){
    valor = $("#buscar").val();
    $.ajax({
        url: "/soundcloud/index.php/search/buscarTrack",
        type: "POST",
        data:{buscar: valor, pag: pagTracks},
        success:function(respuesta){
            var resultados = eval(respuesta);
            html = "<h3>Tracks";
            if(resultados.length>0){
                //Comprobamos si hemos llegado a la ultima pagina (en cuyo caso ponemos la variable ultTracks a true)
                if(resultados.length<10){
                    ultTracks = true;
                }
                else{
                    ultTracks = false;
                }

                html+="

### 


```

```
    }  
    else{  
        html += " - No results</h3>";  
    }  
    $("#resultadosTracks").html(html);  
    mustDisablePageUp();  
    mustDisablePageDown();  
    mustShowPaginator();  
    }  
    });  
}
```

## Problemas encontrados

### *Redimensión del menú*

Al principio usando bootstrap el menú general que tenemos hecho al redimensionarse y no caber los elementos, estos desaparecían, la solución ha sido añadir un botón cuando eso ocurre que muestra los elementos del menú.

### *Grocery CRUD no funcionaba correctamente*

Al principio no funcionaban las operaciones y no cargaba los estilos, lo hemos solucionado cambiando el `base_url` de `config.php`, antes estaba vacía y ahora contiene `'localhost/soundcloud'`.

### *Paginador del buscador*

El paginador ha supuesto un problema ya que había que tener muchas cosas en cuenta, como por ejemplo que habíamos llegado a la última página para deshabilitar la página siguiente, si es la primera deshabilitar el link de anterior, si el usuario cambia de filtro tener en cuenta el cambio... etc. Al final hemos conseguido que sea funcional, pero visualmente no funciona como debe ya que los links se muestran deshabilitados aunque no deban, pero la funcionalidad es correcta.

### *Imágenes en la DB*

Cuando creamos la db pusimos los campos imagen como BLOB para poder almacenar las imágenes en ella. Sin embargo, vimos que era necesario realizar una codificación en base64 para poder enviar las imágenes vía http y mostrarlas correctamente en la web, por lo que hemos optado por cambiar dicho campo a tipo Varchar, en el cual almacenamos la ruta de dónde estarán las imágenes alojadas en el servidor web.

### *Nombre de la tabla 'group'*

Al intentar hacer consultas nos encontramos con que mysql confundía el nombre de la tabla con la instrucción **group by**. Para solucionarlo tuvimos que poner en las consultas las comillas para especificar que era un nombre de tabla ``group``.

## Mejoras y ampliaciones

### ***Ampliaciones:***

- Registro de usuarios.
- Subida de canciones por parte de usuarios y admins.
- Reproducción de canciones.
- Enlace con redes sociales.
- Versión móvil.

### ***Mejoras:***

- Añadir opciones a página home.
- Mejorar el diseño de la página.

## Referencias

Solución al post de starkoverflow que nos ayudó a solucionar el problema de que desaparecía el menú:

<http://stackoverflow.com/questions/30030642/bootstrap-3-responsive-menu-disappears>

Y <http://stackoverflow.com> en general.

Documentación donde buscamos cómo hacer un paginador con bootstrap:

[http://librosweb.es/libro/bootstrap\\_3/capitulo\\_6/paginadores.html](http://librosweb.es/libro/bootstrap_3/capitulo_6/paginadores.html)

Web de referencia tanto para HTML como para JQuery:

<http://www.w3schools.com/>

Documentación de CodeIgniter:

<http://www.codeigniter.com/userguide2/>

Más documentación sobre CodeIgniter:

<https://ellislab.com/codeigniter/user-guide/>

Documentación de Grocery Crud:

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

Documentación de JQuery:

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

Documentación de PHP:

<http://php.net/manual/es>

Referencia para la encriptación de contraseñas de usuarios:

<https://alias.io/2010/01/store-passwords-safely-with-php-and-mysql/>