

# Servicios Multimedia Avanzados

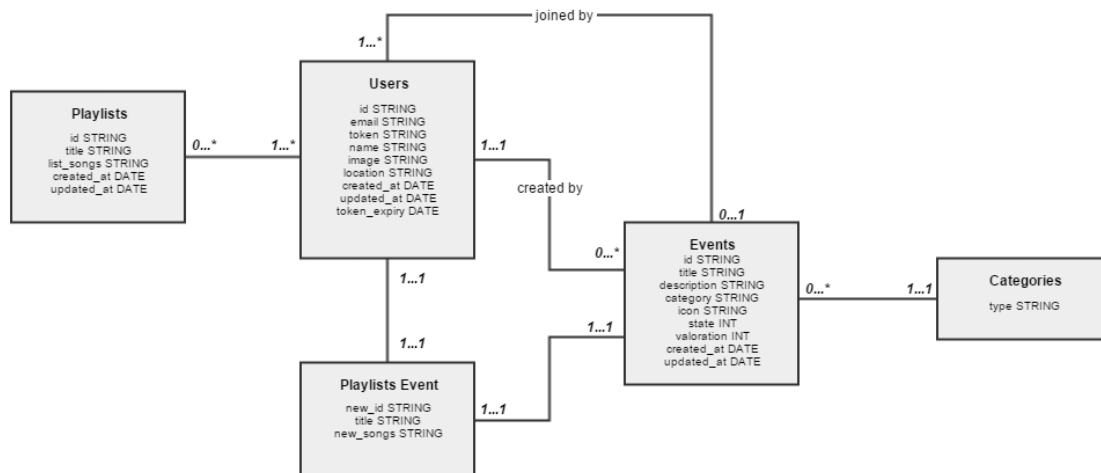
2015-2016

## Informe de Resultados de Aprendizaje

|                                      |  |
|--------------------------------------|--|
| Nombre grupo:                        | Roxanne  |
| Integrantes del grupo (solo de SMA): | Roxanne López van Dooren   |
| Título del proyecto:                 | Mixeet   |
| Descripción breve proyecto:          | Aplicación de navegador y móvil que permite al usuario crear eventos a los que puede asociar una lista de reproducción que se moldeará en tiempo real gracias a la participación activa de otros usuarios invitados. |

**NOTA:** Entregar este informe en formato docx

### 1.- Implementación del sistema de persistencia.



Como podemos ver en la imagen superior, nuestro esquema de base de datos es muy simple ya que consta tan sólo de **cinco colecciones** de por sí bastante ilustrativos. La existencia de una de estas tablas, "Playlists Event", es necesaria porque a la hora de crear un evento, uno o varios usuarios podrán modificar la lista de reproducción añadiendo más canciones por lo que habría que crear una nueva lista asociada tanto al evento en cuestión como a los usuarios implicados.

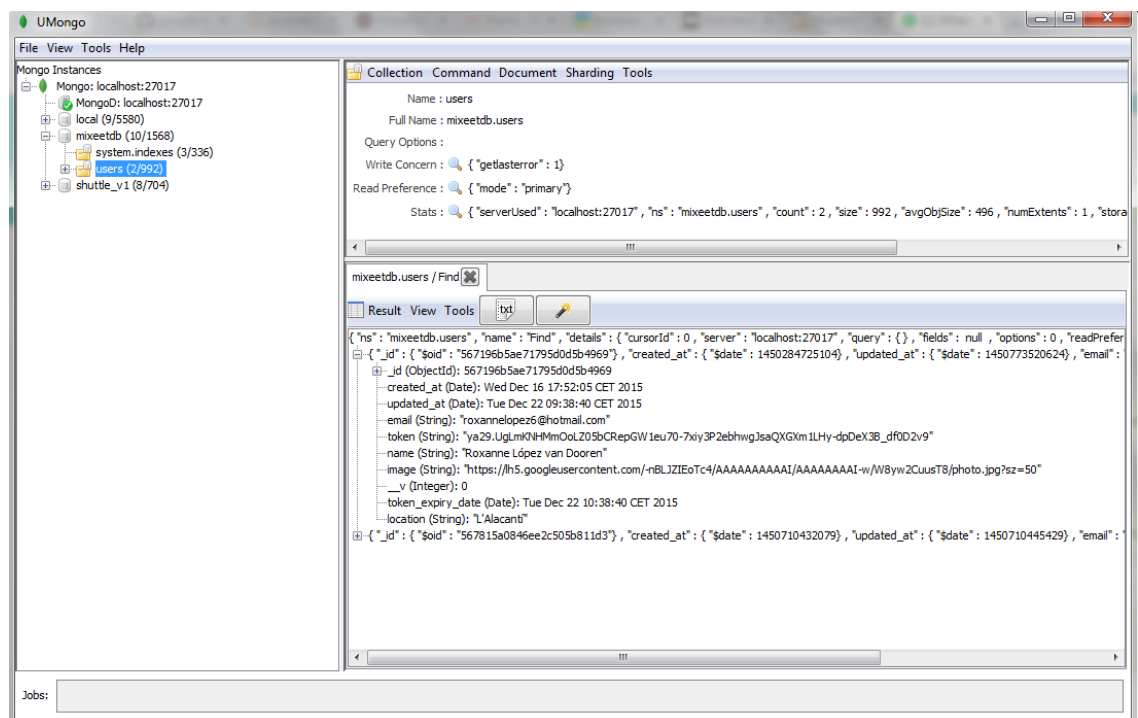
A continuación, se va a explicar por encima las relaciones creadas:

- **Users-Playlists:** Al obtener la información de las listas de reproducción de youtube, un usuario puede o bien no tener ninguna o muchas relacionadas a su cuenta. Asimismo, una lista de reproducción debe tener como mínimo una asociación con un usuario (su creador) y como máximo \*.
- **Events-Categories:** Al crear un evento tendremos que asociarlo a una categoría obligatoriamente de una lista de categorías disponibles en nuestra base de datos.
- **Users-create-Events:** Un usuario puede no haber creado ningún evento o muchos eventos, mientras que un evento sólo puede y debe pertenecer a un usuario.
- **Users-join-Events:** Además, un usuario sólo puede unirse a máximo un evento activo, y a su vez un evento puede tener al menos un usuario o muchos usuarios unidos a él.
- **Users-PlaylistsEvents-Events:** Cuando un usuario crea un evento, se crea una lista de reproducción nueva asociada a estos dos elementos y pertenecerá solamente a este usuario en cuestión. Es una relación de dependencia total (o una ternaria).

El sistema elegido para este proyecto ha sido el de MongoDB, es decir, una base de datos NoSQL – orientada a documentos. A diferencia de las bases de datos que son relacionales, este sistema permite que en una misma colección puedan existir diferentes esquemas, lo cual nos facilita en Mixeet la gestión de nuestros recursos para que escalen fácilmente en el futuro.

Otro de los motivos por el que se ha optado por este sistema no relacional, es que trabaja con estructuras llamadas JSON que resultan muy cómodas y flexibles para nuestras consultas.

Uno de los programas de gestión de bases de datos que se ha usado se denomina UMongo (<http://edgytech.com/umongo/>) que nos permite manejar y visualizar nuestras colecciones cómodamente.



## 2.- API de servicios Backend

### Usuarios

|              |  |
|--------------|--|
| Ruta:        | /users/signin  |
| Descripción: | Comprueba la existencia del usuario obtenido por parámetro para, o bien, registrarlo como uno nuevo o loguearlo. |
| Métodos:     | GET  |
| Parámetros:  | -  |
| Devuelve:    | Token del usuario creado y/o logueado.   |

|              |  |
|--------------|--|
| Ruta:        | /users/me  |
| Descripción: | Método privado que busca al usuario en la colección existente para devolver toda su información. |
| Métodos:     | GET  |
| Parámetros:  | - (REQUIERE AUTH)  |
| Devuelve:    | Email, name, image y location.   |
| Ruta:        | /users/info  |
| Descripción: | Método público que busca al usuario a partir de su email para ver si existe o no.                |
| Métodos:     | GET  |
| Parámetros:  | email  |
| Devuelve:    | Toda la información del usuario si existe o el mensaje "User not found" si no.                   |

|              |  |
|--------------|--|
| Ruta:        | /users/me/modify   |
| Descripción: | Captura la información de los inputs para actualizar la información de usuario (nombre e imagen) |
| Métodos:     | POST   |
| Parámetros:  | - (REQUIERE AUTH)  |
| Devuelve:    | Mensaje de éxito "ok" o de error según el caso.  |

|              |  |
|--------------|--|
| Ruta:        | /users/location/new  |
| Descripción: | Captura la ubicación recibida por el API de Google Maps para actualizar la información de usuario (localización) |
| Métodos:     | POST   |
| Parámetros:  | - (REQUIERE AUTH)  |
| Devuelve:    | Mensaje de éxito "ok" o de error según el caso.  |

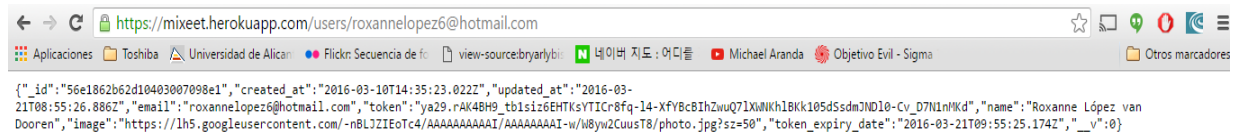
|              |  |
|--------------|--|
| Ruta:        | /users/me/playlists  |
| Descripción: | Método privado que a partir del usuario dado hace una petición al API de Youtube para capturar todas las listas de reproducción del canal de ese usuario |
| Métodos:     | GET  |
| Parámetros:  | - (REQUIERE AUTH)  |
| Devuelve:    | Información de las listas de reproducción  |

## Eventos

|              |  |
|--------------|--|
| Ruta:        | /events/new  |
| Descripción: | Captura los datos (título, descripción, icono, categoría y listas asociadas) y las añade a la colección de eventos de nuestra base de datos. |
| Métodos:     | POST   |
| Parámetros:  | - (REQUIERE AUTH)  |
| Devuelve:    | Mensaje de éxito "ok" o de error según el caso.  |

Teniendo en cuenta los objetivos previstos tanto para la asignatura como para el proyecto Mixeet, todavía quedan muchas rutas que no se han implementado por falta de esfuerzo y motivación, por lo que no hay mucho más que pueda describirse.

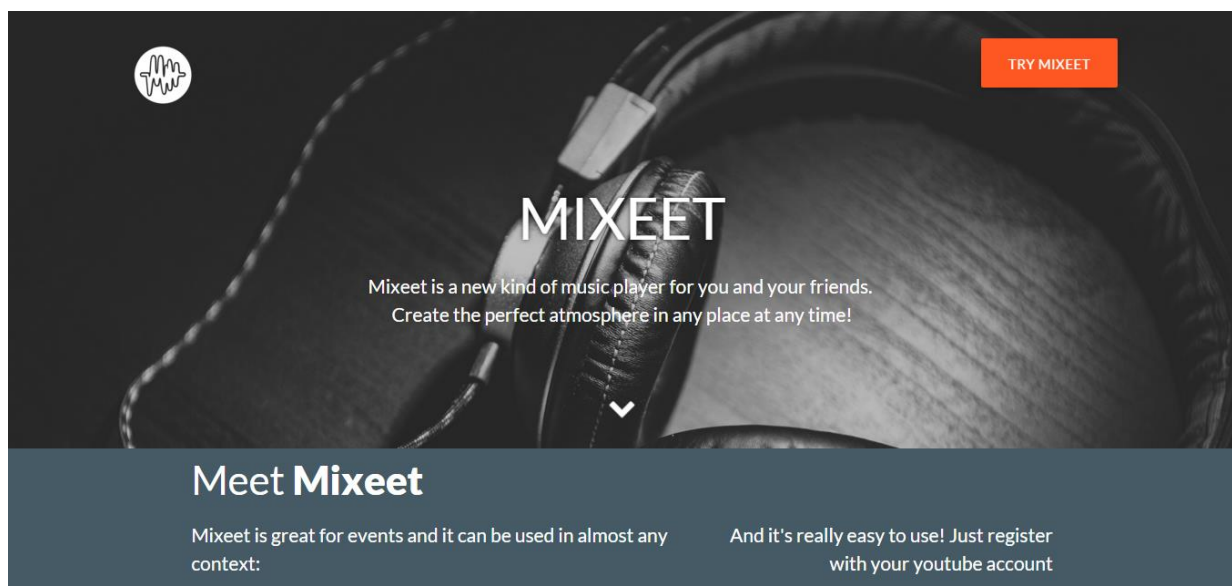
En la imagen siguiente podemos ver un ejemplo del API REST funcionando en nuestro navegador al hacer la petición /users/:roxannelopez6@hotmail.com (usuario ya existente en la base de datos):



### 3.- Implementación Frontend

Como ya mencionado anteriormente, en el hito 3 no se ha desarrollado lo suficiente la aplicación por lo que no se podrán mostrar todas las pantallas inicialmente previstas para el proyecto Mixeet. Las imágenes siguientes representan tanto la aplicación para navegador como la de para móvil.

## Landing



# Meet Mixeet

Mixeet is great for events and it can be used in almost any context:

*at a party with your friends a saturday night*

And it's really easy to use! Just register with your youtube account

SIGN IN WITH 



Mixeet obtains your playlists from youtube for you to manage them freely. You just have to create an **event** and invite your friends to your house, let them connect to your WiFi Network, and play the music!

Your friends will be able to **suggest** songs from their own playlists and Mixeet will customize immediately the music playing in your event!

**The perfect jukebox!**

Universidad de Alicante

INGENIERIA  
MULTIMEDIA

UA project

## Connect with your friends

Have a common WiFi network on and invite your friends to come by!

Roxanne López van Dooren © 2016 Copyright

More Links 



Rethink the way you listen to...  
Music in Company

SIGN IN WITH 

A Final Degree Project  
for the University of Alicante made by  
Roxanne López van Dooren © 2016 Copyright



Create events, like **#myplaylist!**


SIGN IN WITH 


A Final Degree Project  
for the University of Alicante made by  
Roxanne López van Dooren © 2016 Copyright


## Área de acceso

**Google**

Selecciona una cuenta



**Roxanne López van Dooren**  
roxanne.lopez.dooren@gmail.com >

**Roxanne López van Dooren**  
roxannelopez6@hotmail.com >

**GameJam Alicante**  
info@gamejamua.com >

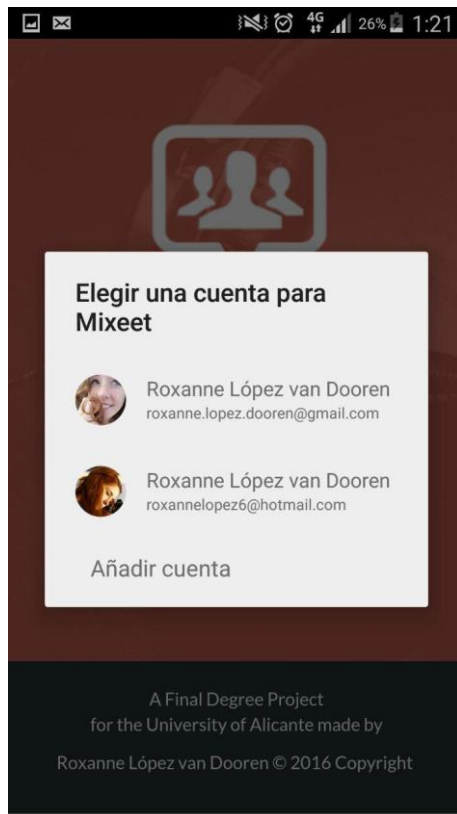
[Añadir cuenta](#)

▾ Mixeet quiere hacer lo siguiente:

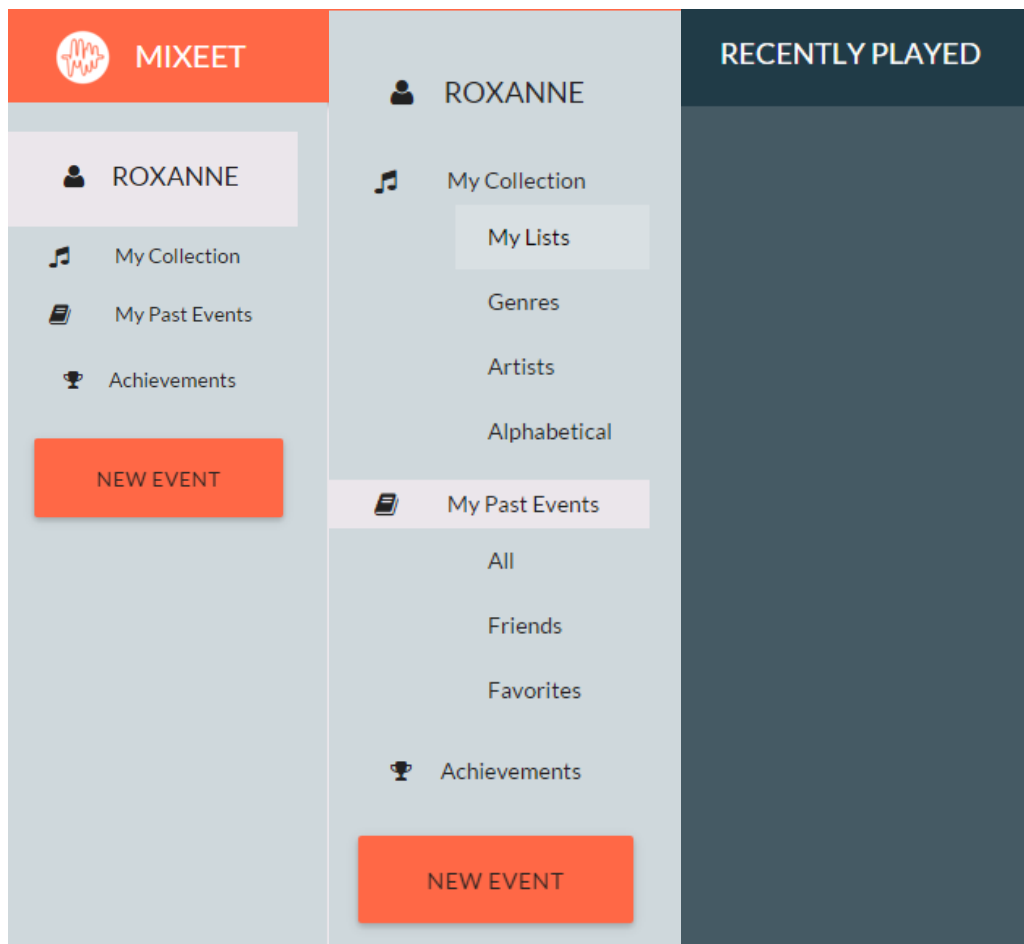
 Tener acceso sin conexión 

Denegar

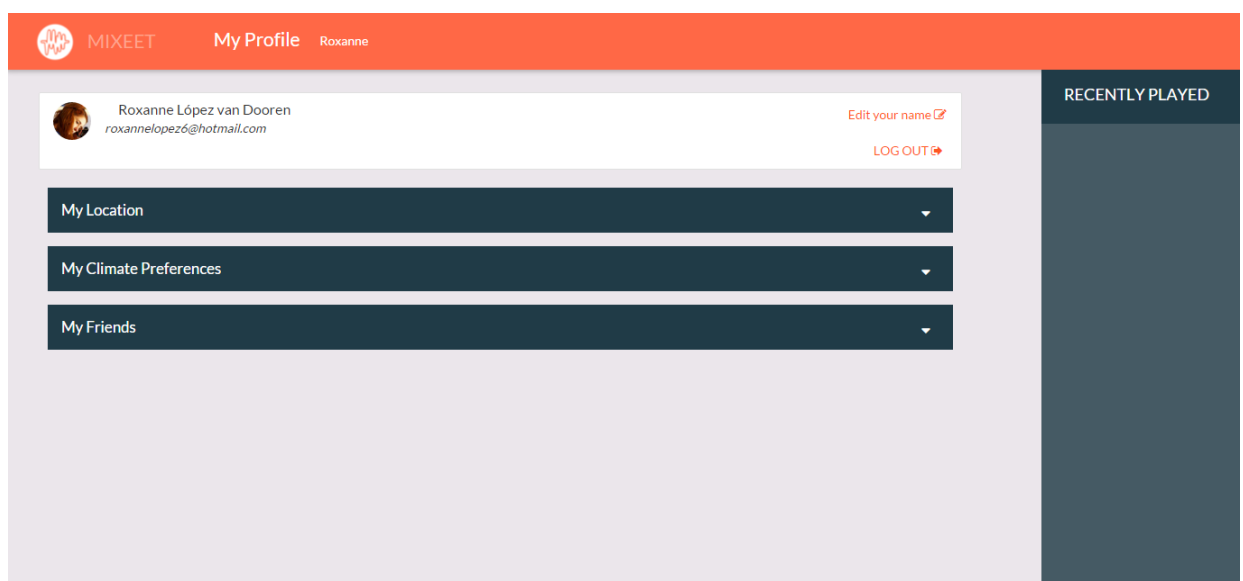
Permitir

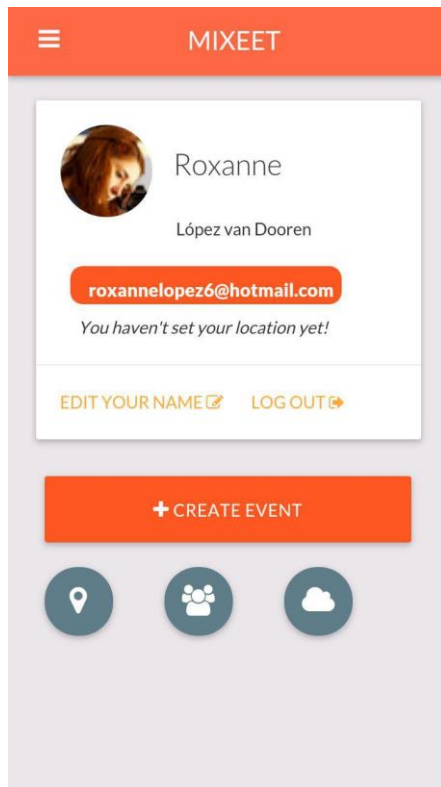


## Menús comunes

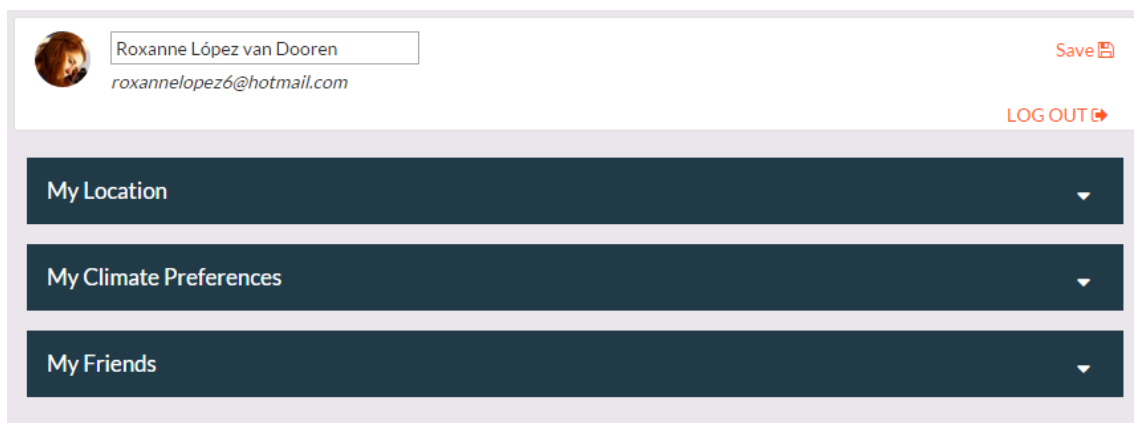


## Perfil de Usuario






## Edición información usuario





## Actualizar información ubicación usuario



**Roxanne López van Dooren**  
roxannelopez6@hotmail.com

Edit your name ✎  
LOG OUT ↗

My Location

You are currently in **L'Alacantí** **MODIFY LOCATION** 📍

My Climate Preferences

My Friends

### 4.- Servicios de integración para desarrolladores.

-

### 5.- Seguridad y Auditoría.

#### 5.1.- Seguridad.

En la aplicación de Mixeet la autenticación se hace por medio del API de Google Plus.

- **Mecanismo utilizado:** generación de un OAuth Refresh Token o Access Token.
- **Utilidad del mecanismo (describir qué se está protegiendo y ante qué):** a través de este token evitamos que un usuario no registrado e indeseado pueda acceder a las partes privadas de la aplicación. Asimismo, operaciones sobre la base de datos como modificaciones de información de un usuario o creaciones de eventos no se podrían realizar a fuerza bruta sin dicho token.
- **Cómo se ha implementado:** se ha utilizado la consola de desarrollador de Google para la generación de este token.

#### 5.2.- Auditoria.

-