2021

IES San Vicente

Sergio Peña Ruiz
Christian López Carretero

[PROYECTO FINAL DE CICLO: ROUTAD]

RoutAd es una aplicación que mostrara publicidad en función del trayecto que realizara un VTC en función de las empresas vinculadas a la empresa.

Índice

1. Introducción....3
1.1 Objetivos....3
2. Antecedentes....4
2.1 Aplicaciones que usan este tipo de tecnología o similares....4
Yelp....4
Swarm....5
Waze....5
3. Análisis....6
4. Diseño....9
4.1 Diseño base de datos....10
4.2 Diagramas de actividad de funciones principales....12
4.2 Descripción detallada de la aplicación....14
Back office....14
Backend para API....15
Frontend Tablet....17
5. Resultados....19
6. Conclusiones....24
6.1 Descripción del desarrollo del proyecto....24
6.2 Conclusiones del trabajo realizado....25
6.1 Trabajos futuros....25
7. Bibliografía....25
8. Apéndice....26
Librerías utilizadas....26

1. Introducción

La aplicación se ha desarrollado para que empresas del sector VTC, puedan ofrecer publicidad a los pasajeros que contraten sus servicios, estos se visualizaran a través de una tablet que se encontrara instalada en el vehículo, se mostraran anuncios publicitarios de las empresas adscritas a la empresa VTC. Los anuncios que se visualizaran se definen en el momento del inicio del trayecto, enviando un aviso al servidor y este a su vez enviara el itinerario a seguir y la tablet mostrara los anuncios según la ruta, que se definirá a través de las coordenadas de inicio y fin.

1.1 Objetivos

La aplicación estará compuesta principalmente por dos tecnologías, el backend se comunicara a través de API Rest y la parte visual sera una tablet con Android, desarrollando una aplicación con Ionic.

En el inicio del desarrollo vamos a definir la estructura de nuestro backend , su función principal sera la de realizar la discriminación de los anuncios en función de la ruta definida al inicio del servicio a realizar, para su realización deberemos de completar los siguientes objetivos.

- Capturar el inicio del servicio donde se reciba los parámetros necesarios para el inicio de la ruta, la cual contendrá información general, y ciertos parámetros del GPS.
- En función de los parámetros recibidos crear un nuevo *Servicio* en el que dispondrá de una planificación de los anuncios a visualizar.
- Para la definición de los anuncios en función de la ruta se utilizara Mapbox, calculando una distancia entre la localización del anunciante y los puntos que define mapbox en la ruta de destino.
- Enviar una notificación push a la tablet de la planificación a seguir.

Por el lado del cliente que es quien consumirá nuestra API, debernos de completar los siguientes servicios.

- Recibir la notificación push para poder iniciar el itinerario a seguir, las tablets trabajaran con los videos o imágenes almacenados en local.
- Disponer de un botón para actualizar los anuncios, una vez este se encuentre en una zona con conexión Wi-Fi, para no consumir los datos móviles.

Finalmente se ha decidido optar por una ampliación en el proyecto y hemos desarrollado un backoffice, para facilitar la gestión de la aplicación y poder ofrecer información, que recibirá a través de la base de datos.

Esta ofrecerá información sobre los anuncios, tablets y usuarios en el caso de ser administrador. En anuncios podemos subir y modificar los mismos a través de una sencilla interfaz y con la implementación de datatables, para una óptima carga de las tablas de los distintos espacios. Posteriormente se explicara la metodología y ciclos de vida.

2. Antecedentes

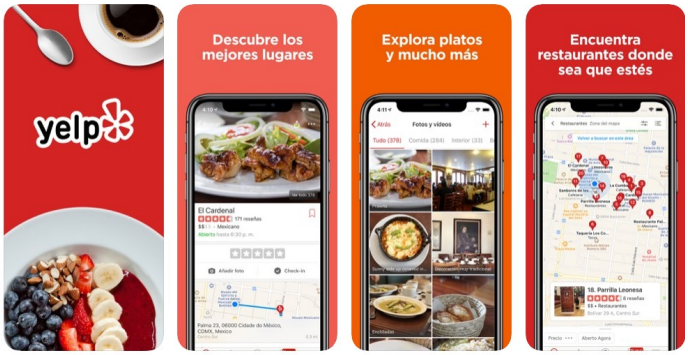
Nos podemos encontrar con innumerables servicios de publicidad por ubicación, aquí únicamente la hemos realizado para el inicio y fin de el trayecto enviado, aplicaciones que puedan realizar esto son muchas, nos encontramos como los servicios de Google, que creo *Eddystone* que es un perfil de baliza que utiliza Bluetooth de baja energía, Apple utiliza esta tecnología bajo el nombre de Ibeacon, pero es limitada.

Google Beacon seria la versión más avanzada que utiliza también WiFi para comunicarse entre dispositivos.

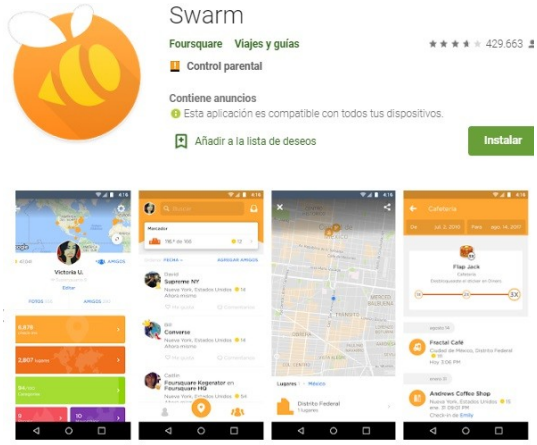
No podemos usar la misma tecnología ya que esta es de pago, posteriormente veremos una posible adaptación tratando de emular dicha funcionalidad.

2.1 Aplicaciones que usan este tipo de tecnología o similares

Yelp



Swarm



Estas aplicaciones utiliza lo que denomina GeoValla, lo que hace es fijar un perímetro virtual alrededor del negocio físico, y esta interactuá con el móvil de las personas que estén dentro de la zona, enviándoles avisos sobre promociones.

Waze

Waze aparte de ofrecer grande servicios de navegación, el que nos atañe es el sistema de publicidad Waze Ads, se utiliza para realizar campañas geolocalizadas y segmentadas, teniendo una gran cantidad de opciones. Esta seria un poco la referencia en el tema de publicidad.

Por ejemplo esta la opción Zero-Speed Takeover, que muestra el local que tienes próximo, en esta aplicación aparece al estar parado, nuestra idea es que aparezca cuando pases cerca del establecimiento, e inicie la reproducción del anuncio.

3. Análisis

La aplicación esta diseñada para que en la tablet que este instalada en un vehículo, pueda reproducir los anuncios que tiene ligado al usuario que inicia sesión, esta al recibir que inicia el servicio debe de reproducir los videos que se encuentran en la ruta, una vez llegue al destino debe de parar la reproducción y estar a la espera de recibir otro servicio.

Esta dividida en 3 partes claramente definidas, disponemos de una página web, en la cual siendo usuario, puedes editar, borrar y subir nuevos anuncios. Otro panel donde se pueden visualizar las tablets enlazadas al usuario, en ella se puede ver los videos que tiene descargados la tablet en ese momento y un historial de servicios realizados. Si el usuario es administrador tiene acceso a toda la información y a un panal de administración de los usuarios.

Por otra parte nos encontramos con la parte de la tablet, esta al iniciar sesión consulta si se ha registrado en el sistema, si no esta registrada, procede a realizar el registro y consulta los videos que tiene asignados a esa cuenta de usuario, la descarga de los videos solo se producirá si dispone de conexión Wi-Fi, por cada video descargado actualizara la base de datos para tener un control de los videos descargados en el dispositivo. Cuando no este realizando un servicio esta podrá realizar gestión de los videos de forma manual, para su borrado o ver si falta algún video por descargar. Finalmente esta se encontrara a la espera de recibir un nuevo servicio, cuando sea recibido esta recibirá el listado de video a reproducir, cuando termine el video estará disponible una imagen publicitaria de la empresa, esta se encontrara disponible hasta el inicio del siguiente video.

Por último nos encontramos con el servicio API Rest, que interconectará lo explicado anteriormente, gestionando todos los servicios de registro e interoperabilidad de los mismos, guardando la información en la base de datos.

Requisitos a cumplir en primera instancia:

- Back Office.
  - Inicio de sesión y registro.
  - Panel gestión de usuarios.
    - Ver perfil.
    - Borrar perfil.
    - Activar/desactivar
  - Panel de gestión de anuncios.
    - CRUD de anuncios.
    - Subida de video e imagen del anuncio.
    - Definir URL para publicidad extra.
    - Determinar la ubicación del anuncio.
    - Activar/desactivar.
  - Panel de gestión de tablets.
    - CRUD de tablets.
    - Modificación de matricula e imagen corporativa.
    - Listado de anuncios disponibles en la tablet especifica.
    - Listado de servicios realizados en la tablet especifica.
- Servicio API Rest.
  - Anuncios.
    - Visualizar todos los anuncios.
    - URL de anuncio a través de ID.
    - Borrar anuncio.
    - Visualizar anuncio por ID.
    - Revisión de anuncios.
  - Autenticación.
    - Registro e inicio de sesión.
  - OneSignal
    - Listado de dispositivos registrados.
    - Registrar nuevo dispositivo.
    - Enviar notificación push.
  - Tablet
    - Listado de tablets.
    - Visualizar tablet por ID.
    - Borrar tablet.
    - Cambiar el registro de matricula.
    - Recibir el video que se ha descargado de la tablet.
  - Usuarios
    - Listado de usuarios.
    - Visualizar usuario por ID.

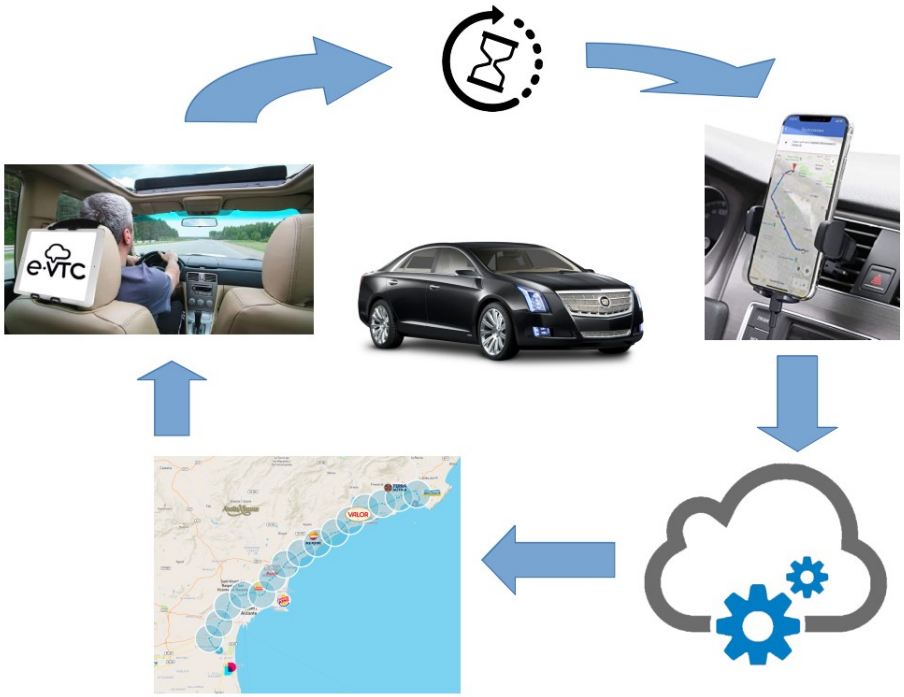
        - Borrar usuario.
        - Cambiar rol del usuario ADMIN/USER
    - Servicios
        - Listado de servicios.
        - Visualizar servicio por ID.
        - Borrar servicio.
        - Crear un nuevo servicio.
    - Reproducciones
        - Reproducciones de un anuncio.
        - Guardar reproducción.
- Dispositivo de reproducción (Tablet).
    - Servicios.
        - Autenticación.
            - Inicio de sesión.
        - Descargas.
            - Estado de la red Wifi/4G.
            - Creación de directorio.
            - Descargar anuncios/Listado de descarga a la espera.
            - Verificación de hash
            - Enviar confirmación de descarga.
            - Borrar anuncio.
        - OneSignal.
            - Registro del dispositivo en oneSignal.
            - Tratar notificación push recibida.
        - Reproductor.
            - Inicio y parada de reproducción.
            - Establecer imagen de fondo.
    - Vistas.
        - Inicio de sesión.
        - Gestión de descargas.

4. Diseño

Procederemos a ver las distintas herramientas y lenguajes que hemos decido utilizar para la realización de esta aplicación, veremos el flujo de la aplicación de una forma simplificada con un pequeño ejemplo gráfico, la definición de las clases, y estructura de la base de datos.

Comenzaremos hablando de las tecnologías utilizadas, por la parte del backoffice y la API se utilizara Symfony 5, en la cual se integraran servicios de la API de oneSignal, para poder enviar información a los dispositivos móviles, en la sección de panel de gestión el motor de plantillas Twig. Para la base de datos utilizaremos MariaDB y finalmente por parte de la tablet la interfaz sera desarrollada con Ionic 5 y compilada con Android Studio.

A continuación veremos un grafico en el cual se puede ver el ciclo de vida a grandes rasgos de la aplicación.

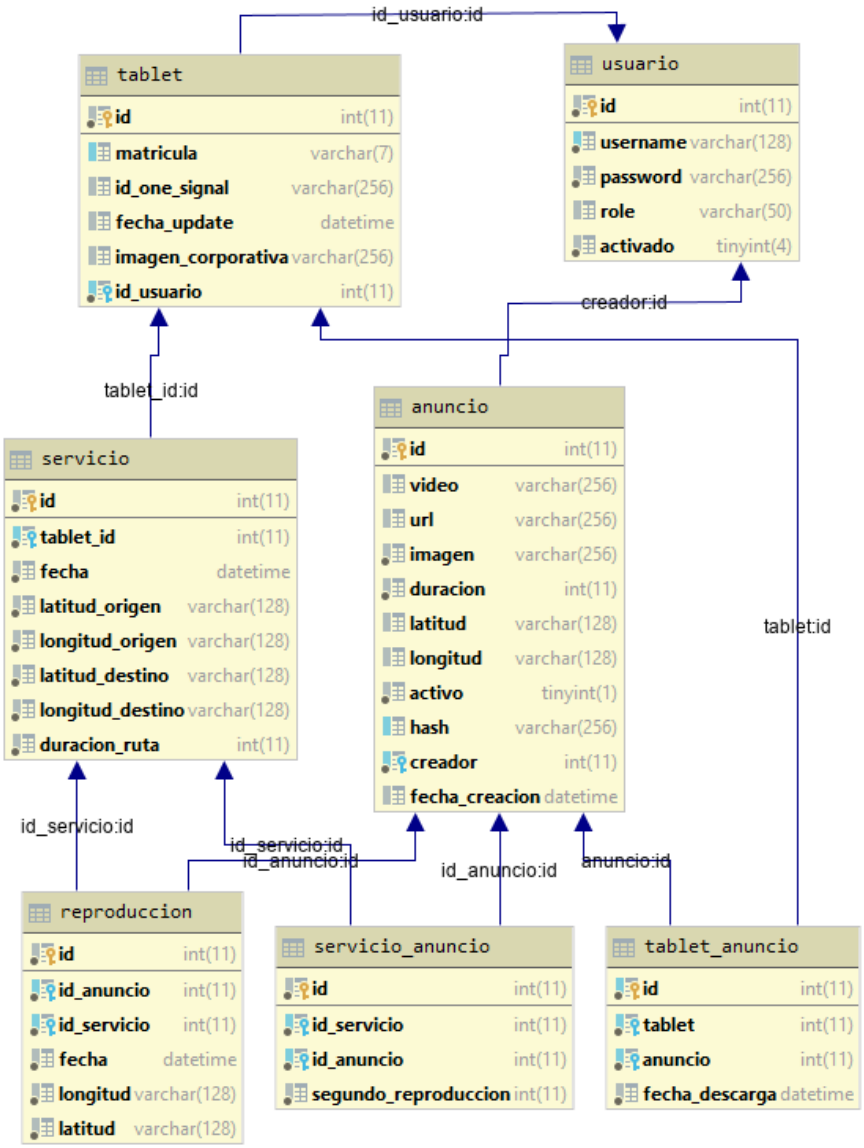


4.1 Diseño base de datos

Para ver la estructura de la base de datos vamos a ver y explicar las tablas de forma independiente y posterior veremos la estructura en conjunto.

- *Usuario*: Contendrá el nombre de usuario, contraseña, rol y si esta activado el usuario.
- *Anuncio*: Esta tabla albergara toda la información relativa al anuncio, como el nombre del video, nombre de imagen, url para posible ruta para publicidad, la duración del mismo, coordenadas, estado del anuncio (activo o no), hash para la integridad en la descarga desde la tablet, id del creador que tendrá como clave ajena la id del usuario que ha subido el video y la fecha de subida.
- *Tablet:* Contendrá la matricula del vehículo que sera única, ID de oneSignal, para poder identificar la tablet dentro del sistema, fecha de la ultima modificación, imagen corporativa y como clave ajena la id del usuario.
- *Tablet anuncio*: Sera la tabla que interconectará las tablas tablet y anuncio como clave ajena y con la fecha de descarga del anuncio.
- *Reproducción*: Se almacenara la id de anuncio y de servicio como clave ajena, se guardara la fecha en la que se realizo la reproducción y las coordenadas.
- *Servicio*: Como clave ajena tendremos la id de la tablet, para poder identificar de quien es el servicio realizado, fecha del mismo, latitud y longitud tanto de origen como destino y la duración total de la ruta mostrada en minutos.
- *Servicio anuncio*: La tabla que interconectará el servicio con el anuncio e indicara el tiempo de espera para la siguiente reproducción.

En la siguiente pagina podremos ver la imagen con el diseño la base de datos, de esta forma podemos ver la estructura con una mejor claridad.

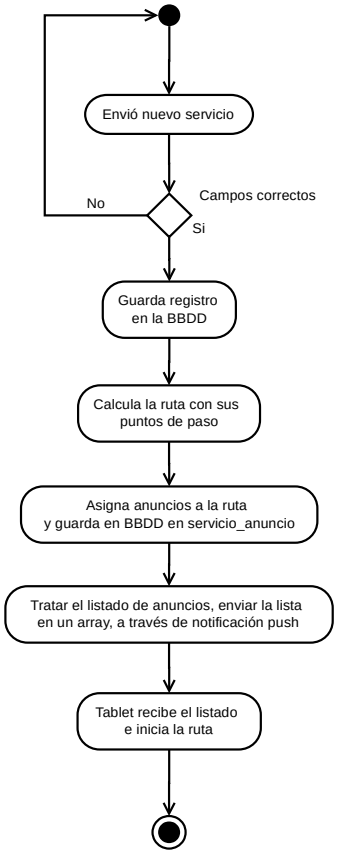

id_usuario:id
tablet
id int(11)
matricula varchar(7)
id_one_signal varchar(256)
fecha_update datetime
imagen_corporativa varchar(256)
id_usuario int(11)
usuario
id int(11)
username varchar(128)
password varchar(256)
role varchar(50)
activado tinyint(4)
creador:id
tablet_id:id
anuncio
id int(11)
video varchar(256)
url varchar(256)
imagen varchar(256)
duracion int(11)
latitud varchar(128)
longitud varchar(128)
activo tinyint(1)
hash varchar(256)
creador int(11)
fecha_creacion datetime
tablet:id
servicio
id int(11)
tablet_id int(11)
fecha datetime
latitud_origen varchar(128)
longitud_origen varchar(128)
latitud_destino varchar(128)
longitud_destino varchar(128)
duracion_ruta int(11)
id_servicio:id
id_servicio:id
id_anuncio:id
id_anuncio:id
anuncio:id
reproduccion
id int(11)
id_anuncio int(11)
id_servicio int(11)
fecha datetime
longitud varchar(128)
latitud varchar(128)
servicio_anuncio
id int(11)
id_servicio int(11)
id_anuncio int(11)
segundo_reproduccion int(11)
tablet_anuncio
id int(11)
tablet int(11)
anuncio int(11)
fecha_descarga datetime

4.2 Diagramas de actividad de funciones principales

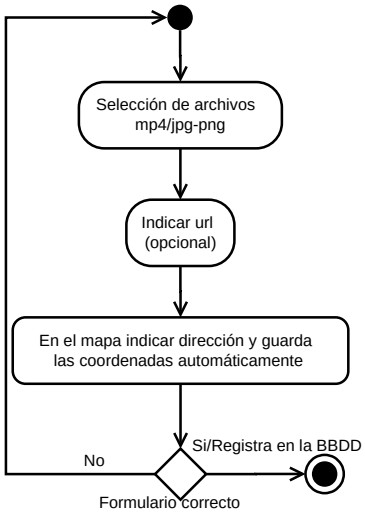
A través de diagramas de actividad, veremos el flujo que siguen para la realización de las tareas más habituales.

A continuación veremos el inicio de un servicio, este en principio se debería de ejecutar desde una aplicación móvil, pero por no sobre dimensionar el proyecto, sera simulando el inicio de un servicio que se realizara a través de la API que esta a la espera de recibir nuevos servicios.

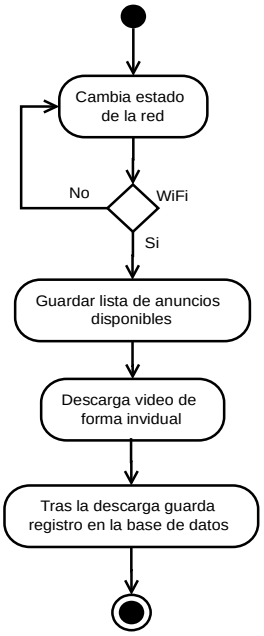
Los datos que debe de recibir deben de ser los siguientes; matricula, datos de origen y destino e id de oneSignal.



El siguiente caso de actividad podremos ver, como seria la publicación de un nuevo anuncio.



Para la descarga de un video desde una tablet, veremos los pasos a realizar.



4.2 Descripción detallada de la aplicación

Back office

Comenzaremos por el backoffice, se hará también referencia a la interfaz grafica la cual utilizara Bootstrap. Para todos los listados se utilizara la librería Datatables, para una paginación automática y una carga mucho mas optima de los registros, a la vez que podremos buscar con un campo de búsqueda y ordenar las columnas.

El controlador de *seguridad* se ha definido a través del paquete security. Este mostrara la pantalla de inicio, nos encontraremos con un formulario de inicio de sesión y opciones de registro en el navbar. Una vez se inicia sesión sera redirigido al listado de anuncios.

- Security
  - Controlara el acceso de usuarios mediante inicio de sesion, discriminando según su rol, para permitir ver unas opciones u otras.
  - Cerrar sesión.

El controlador de *usuario* dispondrá de las siguientes funciones y nos mostrara en la vista.

- Usuarios
  - Rol usuario y administrador
    - Formulario de registro de un nuevo usuario.
    - Editar tu usuario.
  - Rol administrador
    - Ver listado de usuarios.
    - Poder eliminar el usuario.
    - Cambiar el rol de activo o inactivo.
    - Ver los detalles del usuario, sus anuncios y tablets registradas

En el controlador de *anuncios*, nos encontraremos el listado de tus anuncios, si fueras administrador, están disponible todos los anuncios. En el listado tendremos las opciones de ver, editar y borrado. En la parte superior de la tabla nos encontraremos con el botón de subir un nuevo anuncio, y nos encontraremos con el siguiente formulario.

- Video
- Imagen
- URL
- Mapa

Para el mapa se ha utilizado Mapbox, de esta forma, en la barra de búsqueda se indica la ubicación de donde se quiere reproducir el anuncio, de forma interna guardaremos las coordenadas de latitud y longitud en el formulario, antes de su envió, todos los campos serán obligatorios a excepción de la URL.

Dentro de la vista de ver el anuncio, podremos reproducir el video, ver la imagen, y si esta activo, la url y el número de reproducciones del anuncio. En edición podremos ver la ubicación actual y sera posible modificarla, al igual que el resto de campos.

La vista de tablet, es una de las vistas mas interconectadas,en ella se utilizara el controlador de *tablet*, *servicio* y *reproducciones*, en la vista podremos ver nuestro listado de tablets, con sus respectivas opciones como las de anuncios. Ha diferencia de los anuncios, el registro de una tablet se realiza a través de la API, el funcionamiento seria que la tablet, se registrara automáticamente la primera vez que el usuario inicia la sesión desde ella, luego desde el apartado de edición,esta la opción de asignarle la matricula y se podrá incluir la imagen corporativa.

En la vista de la tablet podremos ver los siguientes datos

- Datos de la tablet
  - ID.
  - Matricula.
  - Fecha de última actualización.
  - Enlace a servicios realizados.
    - Este nos llevara a otro listado indicando cada trayecto con los siguiente campos.
      - ID.
      - Fecha de realización.
      - Coordenadas de origen y destino.
      - Duración de la ruta.
  - Anuncios actualmente disponibles en la tablet.
  - Visualización de imagen corporativa.

Backend para API

Veremos como haremos la comunicación de nuestro backend con la tablet. Para poder enviar datos de forma que sea cuando se produzca el evento, utilizaremos las notificaciones push, a través de oneSignal, de esta forma la tablet siempre estará disponible para recibir información, al igual que gestiona el registro de la tablet.

Aquí explicaremos los controladores y la lógica de negocio que pertenece a cada uno, las opciones globales se definieron en los requisitos, vamos a omitir las funciones básicas de CRUD, para no alargar excesivamente el apartado.

Controlador de autenticación se encargara del registro e inicio de sesión, para controlar los accesos al API, implementando el estándar JWT.

Controlador de anuncios contendrá la funciones de traer la url del video, por si fuera necesario su uso, para acceder a la ruta del video, que esta alojado en el servidor. La función que cabe destacar en este controlador seria la revisión de anuncios, para discernir que los anuncios sean de tu propiedad, este recibe la id de oneSignal que pertenece a la tablet, realizara una subconsulta en la base de datos, utilizando las tablas tablet_anuncio y anuncio, de esta forma devolverá los anuncios pertenecientes a la tablet.

Controlador de oneSignal nos proporcionara la comunicación con con las tablets, a continuación veremos las funciones utilizadas.

- Devolver el listado con todos los dispositivos registrados en el sistema.
- Dar de alta el dispositivo, este se realiza una vez se inicia sesión por primera vez en al tablet, este recibirá parámetros para el registro y se deberán de tratar para realizar el alta en oneSignal a través de la API que proporcionan ellos, a su misma vez instanciaremos una nueva tablet y esta realizara el registro en la base de datos.
- Enviar una notificación push, esta recibirá tanto el id de oneSignal que pertenece al dispositivo, como el listado de anuncios que va a enviar a la tablet. De esta forma podremos enviar el listado al dispositivo especifico.

Controlador de Reproducciones este controlador únicamente es para guardar los registros de reproducción de cada anuncio, este realizara dos acciones, que serán guardar un nuevo registro, que recibirá de la tablet los datos, que serian id del anuncio , servicio, fecha de reproducción y coordenadas del momento de la reproducción. También sera posible que devuelva el listado de todas las reproducciones enviando la id del anuncio.

Controlador de Servicio contiene las opciones comunes de borrar, devolver listados, pero la más importante es la de iniciar un nuevo servicio. Expondremos los pasos que debe realizar en detalle.

En primer lugar recibirá una petición por POST con los siguientes parámetros.

- Matricula.
- latitud y longitud de origen.
- latitud y longitud de destino.
- Id del dispositivo.

Dentro de la función de crear un nuevo servicio tras comprobar que los campos son correctos instanciamos un nuevo servicio, con los datos recibidos, uno de los campos sera la distancia de la ruta, esta la podemos definir gracias a una función creada, que

nos calcula con las coordenadas la distancia de la ruta, que sera devuelta en metros, también guardara la fecha en que se recibe el servicio. Una vez se introducen todos los campos se guarda el registro en la base de datos.

Dentro de la misma función deberemos de instanciar un objeto de Mapbox que vamos a utilizar para recibir la ruta, utilizando la API de Mapbox, nos devolverá la ruta completa con el itinerario de la ruta a realizar y en ella hay checkpoints, que serán en los que nos apoyaremos para definir que anuncios pasaran cerca de la ruta, con otra función nos comprobara si el anuncio que dispone la tablet, dentro de esta instanciara varios *servicio_anuncio* que guardara cada anuncio correctamente en la base de datos. Utilizando otra función con se comprobara la duración de la ruta y que la cantidad de anuncios no exceda de esta, nos devolverá la lista de videos a reproducir y se enviara a través de una notificación push a la tablet especifica que solicita el servicio.

Controlador de Tablet dispondrá de las siguientes funcionalidades.

- Poder registrar en la base de datos el video descargado correctamente.
- Borrar el anuncio del registro de la base de datos a petición de la tablet.
- Cambiar la matricula asociada a la tablet.

El guardado se realiza para que cuando la tablet inicie sesión, pueda saber que videos debe descargar o están almacenados en ella.

Controlador de usuario nos encontramos con el cambio de rol, para poder realizar la modificación, ademas de traer el listado de usuarios y el borrado de los mismos.

Frontend Tablet

Nos encontramos con la parte más visual de la aplicación, explicaremos cada uno de los módulos y el funcionamiento global de los pasos para la realización de un servicio, a la vez explicaremos el funcionamiento de la interfaz, que tendrá como particularidad, ya que es una tablet para visualizar anuncios, mantendrá la orientación horizontal

*Modulo de autenticación e*s lo primero que encontramos al abrir la aplicación, nos encontraremos con la vista de inicio de sesión, al acceder se pone en funcionamiento el modulo de oneSignal, este recogerá los datos de oneSignal y conectara con la API del backend, enviándole la ID que nos proporciona oneSignal, si es la primera vez se realiza el registro en el sistema, como hemos explicado anteriormente y seremos redirigidos a la vista principal.

Modulo pages es modulo principal la cual se suscribe al servicio para controlara si el dispositivo dispone de wifi en ese momento, para poder iniciar la descarga del listado

de los videos, después veremos los detalles del servicio de descargas, donde veremos todo el proceso de gestión.

El botón para ver el listado de vídeos descargados sólo es visible si el dispositivo tiene conexión wifi (suponemos que si la tablet tiene wifi es que el vehículo se encuentra parado y sin ningún servicio activo). Al pulsar el botón se abrirá un modal que sera el modulo de *descargas* en el cual podremos ver la lista de videos que tenemos disponibles, tendremos las opciones de refrescar la lista, descargar de nuevo, opción de previsualización del video y borrado. Una vez salimos del modal la aplicación esta a la espera de recibir la notificación push con el listado.

Listado de servicios y su funcionamiento

Downloads este sera el servicio más importante ya que gestionara todo referente a los anuncios, a continuación veremos las funciones con las que contamos.
Podemos definir el estado de nuestra conexión para saber si disponemos de red Wifi, una vez se disponga de wifi se solicitara la petición de verificar los anuncios que pertenecen a la tablet, la lista que reciba si se da el caso se guardara en un array de descarga de anuncios pendientes y se empezarán a descargar uno a uno hasta vaciar la lista de anuncios pendientes de descargar. Este servicio también se encarga de la gestión local de los anuncios (comprobar checksum MD5, borrar vídeo, etc.). Cada vez que un vídeo se descarga correctamente o se borra de local, se realiza una llamada a la API del backend para que esta sepa en todo momento los vídeos que esa tablet tiene descargados.

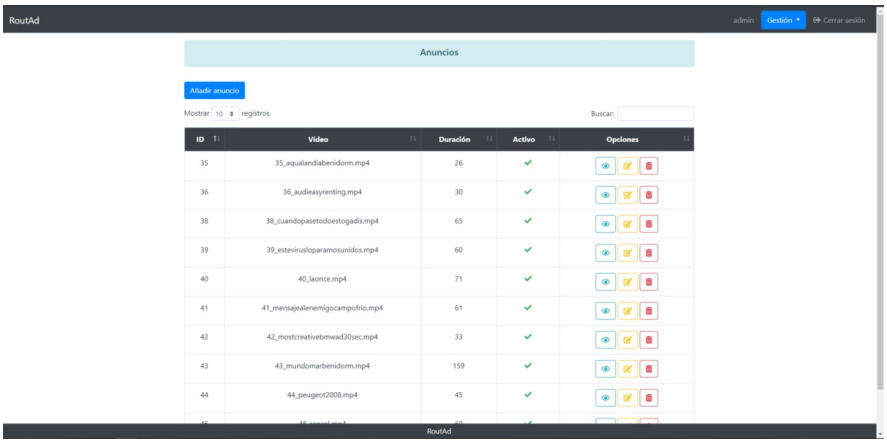
OneSignal iniciara en el momento que se realice el inicio de sesión, ya hemos indicado anteriormente como hace el registro, este también gestiona el inicio de servicio, que recibe a través de la notificación push, en ella vendrá la lista de videos que debe de reproducir el servicio, la id del mismo y el tiempo de espera entre los videos, este llamara al servicio del reproductor que es el que explicaremos a continuación.

VideoPlayer gestionara toda la reproducción del video, tendremos las funciones de definir la lista de reproducción, iniciarla o pararla. Una vez iniciada la lista empezará a reproducir todos los vídeos de la lista de reproducción uno a uno, dejando un tiempo de espera especificado en la notificación push. En dicho tiempo se mostrá una imagen publicitaria del anuncio que se acaba de reproducir hasta que empiece el siguiente. Una vez que se reproduzca cada vídeo se enviará a la API del backend toda la información de la reproducción de ese anuncio (coordenadas GPS, anuncio, servicio, momento).

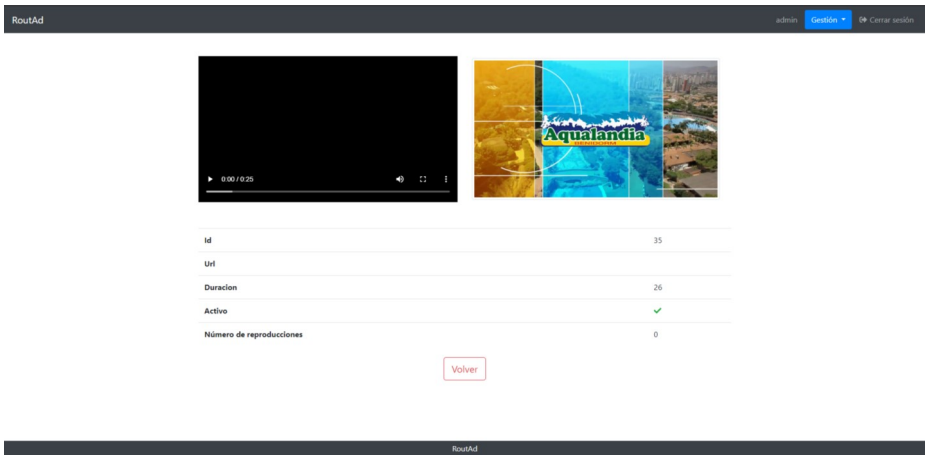
Para la compilación se ha utilizado Android Studio y sera compatible desde Android.

5. Resultados

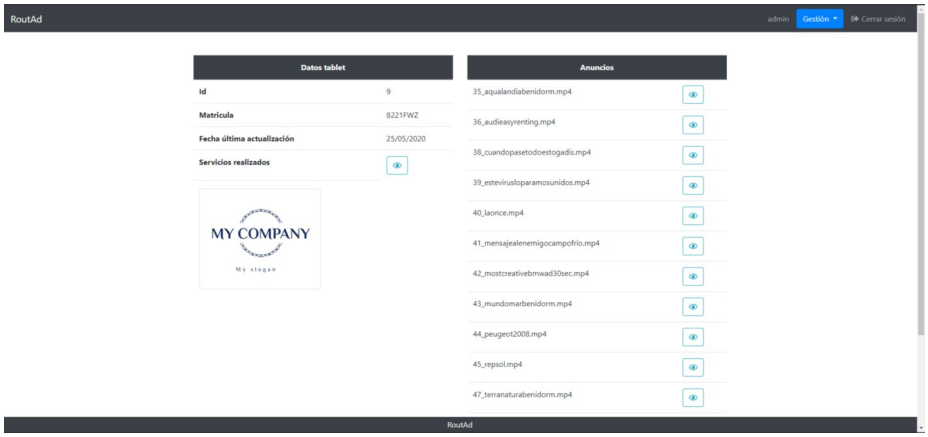
Veremos los resultados de nuestro proyecto, con algunas capturas de pantalla y algunos resultados que nos devuelve la API, tras el desarrollo de nuestra aplicación.



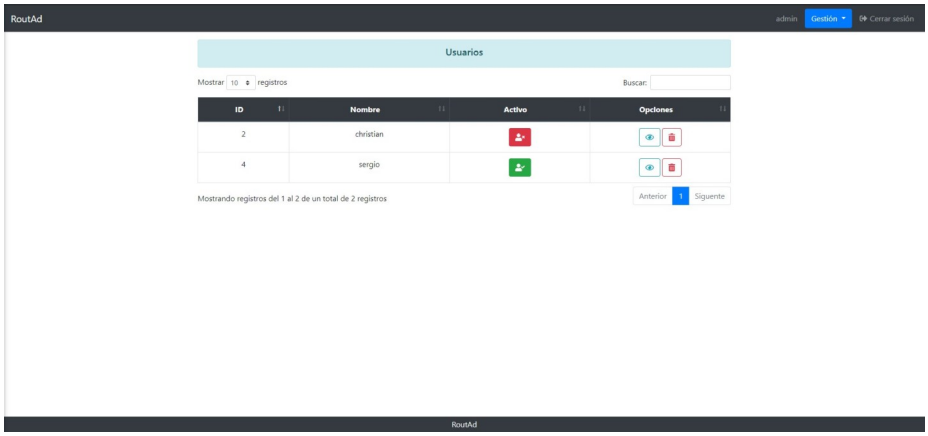
Aquí podemos ver la vista de listado de anuncios, donde podemos realizar las acciones descritas anteriormente. En esta veremos los detalles de un anuncio.



Gestión de las tablets registradas, con su listado de anuncios y enlace a servicios realizados, en la cual se vera la tabla de las rutas.



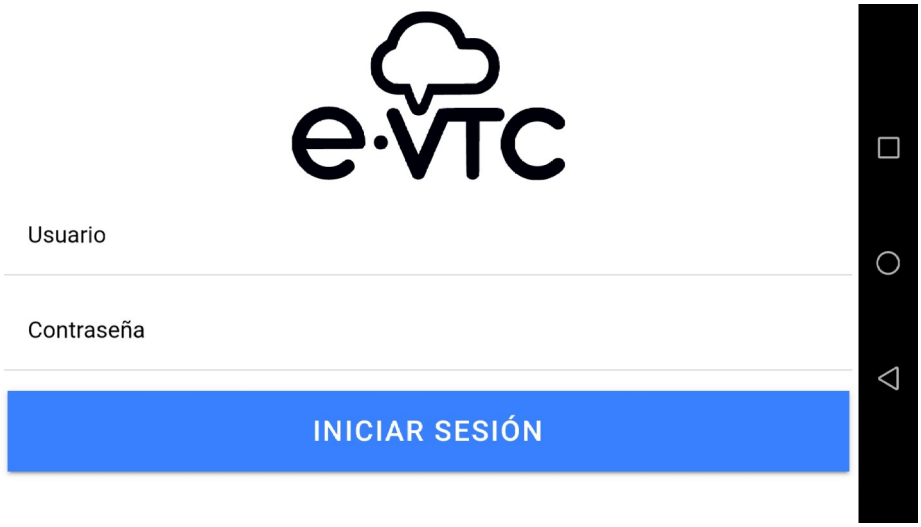
Finalmente veremos la sección de usuarios, que hay que se administrador para tener acceso a ella.



Lo siguiente que vamos a ver es un ejemplo completo, desde el panel de anuncios veremos cuales están activos y vamos ver todo el paso de la aplicación de forma visual, a la vez que vemos la interfaz de la tablet.

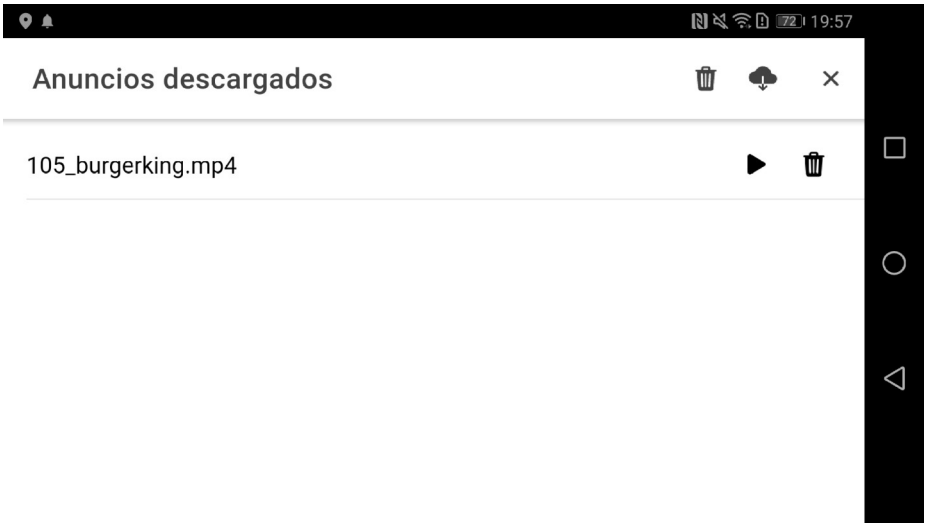
En primer lugar podemos ver los anuncios activos que seria el de Burger King.

Accedemos a la aplicación y vemos la pantalla de inició de sesión.

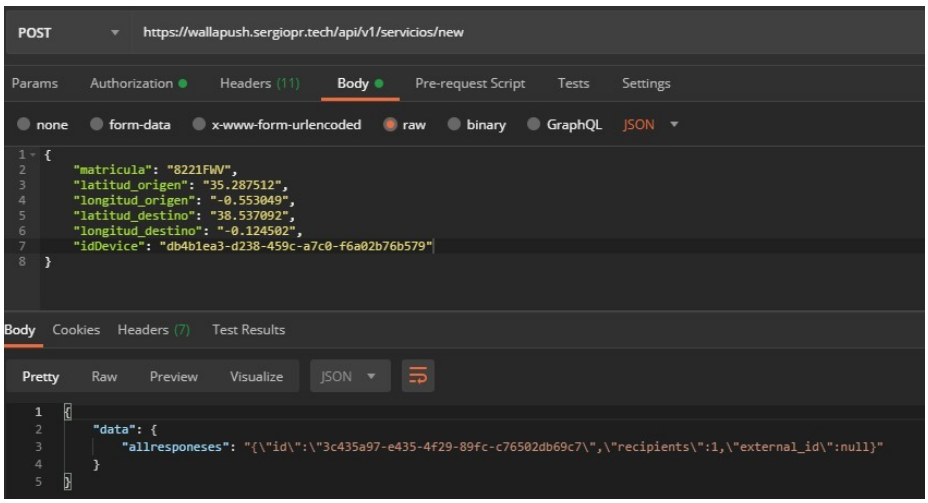


Tras el inicio de sesión en la pantalla principal si estamos conectados a la red wifi dispondremos del botón para la gestión de anuncios y descargas, si no este quedara oculto.

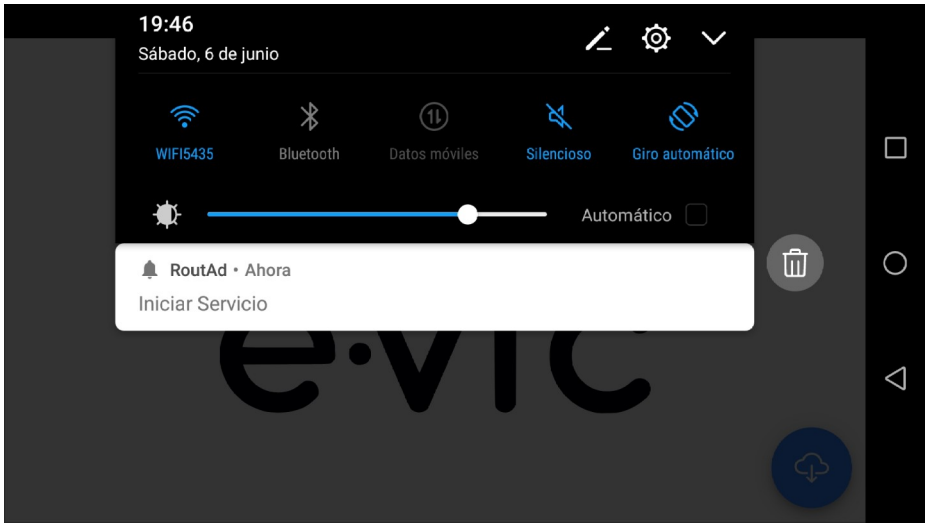
Al pulsar en el nos aparecerá el modal, para la gestión de los anuncios y vemos que unicamente esta el anuncio activado.



Ahora enviaremos los datos que son necesarios, para que inicie el servicio.



Como podemos ver nos devuelve una id interna de Postman, y los dispositivos que reciben la notificación al ser los servicios individuales vemos que indica 1 en recipients, y el móvil nos avisara con una notificación push e iniciaran los videos de forma automática.



Una vez terminan las reproducciones volverá a la vista de inicio a la espera de recibir nuevos servicios.

6. Conclusiones

6.1 Descripción del desarrollo del proyecto

Calculamos que el total de horas del proyecto (incluyendo investigación, correcciones de errores, etc) han sido unas 300.

En el proyecto nos hemos encontrado con los siguientes problemas que han derivado en la falta de algunas implementaciones planeadas en el inicio del proyecto:

- En el backend.
  - No hemos podido discriminar si el usuario está activo o no.
- En el frontend.
  - No hemos podido conseguir que se muestre la imagen publicitaria cuándo se acaba de reproducir un anuncio.
  - No hemos conseguido que la función de comprobar el checksum MD5 nos devuelva el valor del checksum generado.

6.2 Conclusiones del trabajo realizado

Durante el desarrollo del proyecto hemos podido comprobar cómo es el desarrollo de un proyecto muy similar a una situación laboral real. Dónde tuvimos que planificar los requisitos que tenía que tener la aplicación, planear la base de datos, tecnologías, y llamadas de API que tenía que tener para el correcto funcionamiento de la misma.

6.1 Trabajos futuros

- Implementación de geofences, para poder definir el momento exacto de iniciar la reproducción.
- Poder ver desde el backend que anuncios ha reproducido una tablet.
- Al ver el detalle de un anuncio en el backend, que se muestre el mapa de dónde está localizado dicho anuncio.
- Poner un contador del tiempo restante para el siguiente anuncio en el frontend.
- Evitar que la pantalla de la tablet se bloquee si la aplicación está abierta en primer plano.
- Bloquear todos los botones del dispositivo para que el cliente del servicio no pueda interactuar con la tablet.

7. Bibliografía

Para el desarrollo del backend con Symfony.

- Temario del modulo Desarrollo web en entorno servidor.
- Web oficial de Symfony.

Para el desarrollo del fronend con Ionic

- Temario del modulo Desarrollo web en entorno cliente.
- Web oficial de Ionic.
- Curso en Udemy de Ionic 5.

Búsquedas en Stakoverflow.

8. Apéndice

Librerías utilizadas

- Mapbox
- Datatables
- OneSignal
- SweetAlert
- Bootstrap
- FOSJsRoutingBundle
- Capacitor
- cordova-plugin-file-hash
- Video Player plugin for Cordova/PhoneGap
- Cordova Screen Orientation Plugin
- Full Screen Plugin for Cordova
- cordova-plugin-file-transfer