



*Universidad Nacional de La Matanza*

**Departamento de Ingeniería e Investigaciones  
Tecnológicas**

# **Sistemas Operativos Avanzados**

## **Trabajo Práctico Sistemas embebidos y Android IoT**

### **Estacionamiento Automatizado**

1°er cuatrimestre 2017

Miércoles Turno Noche

Integrantes:

<b>Nombre y Apellido</b>	<b>DNI</b>
Esquer Juan Matias	35137733
Soria Sheila	35537624
Villca Raúl	34370611

<b>Objetivo del trabajo práctico</b>	<b>3</b>
<b>Descripción del entorno de trabajo</b>	<b>4</b>
Sistema Embebido	4
Componentes:	4
Sensores:	4
Actuadores:	4
IoT – Aplicación Android	4
Software utilizado	4
Hardware utilizado	4
Sensores en la aplicación Android	5
<b>Descripción del alcance del sistema (SE + IoT)</b>	<b>5</b>
Sistema embebido:	5
Aplicación Android:	6
Comunicación SE + Aplicación Android:	10
<b>Avance del proyecto en imágenes:</b>	<b>10</b>
<b>Conclusiones:</b>	<b>13</b>

## Objetivo del trabajo práctico

Implementar los conocimientos obtenidos sobre Sistemas Embebidos e IOT a lo largo de la cursada desarrollando un sistema de estacionamiento automatizado que servirá para gestionar los espacios disponibles en un estacionamiento informando a los clientes sobre la disponibilidad del lugar a través de un display y una aplicación Android que permitirá además la reserva de espacio de estacionamiento.

# Descripción del entorno de trabajo

## Sistema Embebido

### Componentes:

- Microcontrolador Raspberry pi 3 b
- 1 Protoboard
- Cables para conexión
- Resistencias para ultrasonido
- Representación del estacionamiento en maqueta

### Sensores:

- 2 Sensores de ultrasonido
- 2 Sensores infrarrojos
- 1 Foto celda

### Actuadores:

- 2 Servomotor
- 2 Buzzers
- 3 Luces (Leds)
- Display LCD 16x2

## IoT – Aplicación Android

### Software utilizado

- Android Studio (aplicación) y PyChar (placa raspberry)

### Hardware utilizado

- Dispositivo Android, Placa Raspberry, Sensores y Actuadores

## Sensores en la aplicación Android

- Lector de huella
- Giroscopio
- Sensor de proximidad

## Descripción del alcance del sistema (SE + IoT)

### Sistema embebido:

La funcionalidad principal del sistema es gestionar los espacios disponibles en un estacionamiento informando a los clientes sobre la disponibilidad del lugar a través de un display en la entrada del estacionamiento. La gestión se realizara utilizando los sensores ultrasónicos que informaran si el espacio está ocupado o no. Además, estos sensores controlaran una posible colisión del automóvil al estacionar y en caso de ser necesario se alertara al conductor mediante la activación de una bocina (buzzer).

[El **display** es un LCD (Liquid Crystal Display) de 16x2 (esto quiere decir que dispone de 2 filas de 16 caracteres cada una) donde el proceso de visualización está gobernado por un micro controlador el cual dirige todo su funcionamiento.

Los **sensores ultrasónicos** están compuestos por un transmisor y receptor ambos en forma de cilindro. El transmisor se encarga de emitir la señal ultrasónica para luego ser rebotada a través de un objeto llegando al cilindro receptor. Se tienen tres canales: el canal del disparo, el canal donde se emite la señal desde el módulo y el canal del eco respectivamente. El módulo inicia su trabajo enviando un pulso TTL (típicamente 5 V) con una duración de 10 us. Una vez que el módulo haya enviado ocho pulsos ultrasónicos de 40 kHz aproximadamente, el pin de eco enviará la distancia a la cual está el objeto en forma de pulsos con nivel TTL de amplitud más baja. Dependiendo del ancho del pulso que emita el receptor (va desde aproximadamente 150 us. a 25 ms.) se tendrá distintos valores de distancias y a partir de ellos se determinara la presencia/ausencia de un objeto.

Un **buzzer** (zumbador) es un transductor que se encarga de transformar energía eléctrica en acústica. Posee dos terminales una positiva donde se envía la señal eléctrica y otra negativa por lo general puesta a tierra. Para generar un sonido de un tono, es decir de una sola frecuencia, basta con

aplicar una onda cuadrada a la terminal positiva del buzzer para que éste transforme la onda cuadrada a un sonido con su correspondiente frecuencia.]

El estacionamiento contará además con barreras automatizadas de ingreso y salida implementadas con sensores infrarrojos para detectar que un vehículo desea ingresar al estacionamiento y servomotores para el movimiento de las barreras. La apertura de la barrera para ingreso estará sujeta también a la disponibilidad del estacionamiento.

[El **sensor infrarrojo** es un dispositivo opto electrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión. Está formado por un Diodo emisor de luz infrarroja (LED IR - emite un tipo de radiación electromagnética (señales) llamada infrarroja, que es invisible para el ojo humano porque su longitud de onda es mayor a la del espectro visible.) y un Fototransistor (este dispositivo se diferencia de un transistor común porque su base ha sido sustituida por un cristal fotosensible que regula el flujo de corriente colector – emisor de acuerdo a la luz incidente sobre él.

Los **servomotores** son un tipo especial de motor con características especiales de control de posición. Para ello, el servomotor espera un tren de pulsos que se corresponde con el movimiento a realizar. La electrónica dentro del servomotor responderá al ancho de la señal modulada. Si los circuitos dentro del servomotor reciben una señal de entre 1 a 1,4 milisegundos, éste se moverá en sentido horario; entre 1,6 a 2 milisegundos moverá el servomotor en sentido anti horario; 1,5 milisegundos representa un estado neutro para los servomotores estándares.]

La iluminación del estacionamiento será a través de leds RGB (utilizamos el color rojo) y serán sensibles a la luz del día, para controlar esto utilizamos una fotocelda.

[Un LED RGB es un LED que incorpora en su mismo encapsulado tres LED, es RGB porque así se pueden formar miles de colores ajustando de manera individual cada color. Los tres LED están unidos por el negativo o cátodo. Una **fotocelda** es un dispositivo electrónico que es capaz de producir una pequeña cantidad de corriente eléctrica al ser expuesta a la luz, es una resistencia cuyo valor en ohmios cambia ante las variaciones de la luz. Esta resistencia está construida con un material sensible a la luz, de tal manera que cuando la luz incide sobre su superficie el material sufre una reacción química alterando su resistencia eléctrica, presenta bajo valor de su resistencia ante la presencia de luz y presenta un alto valor de la resistencia ante la ausencia de luz.]

## Aplicación Android:

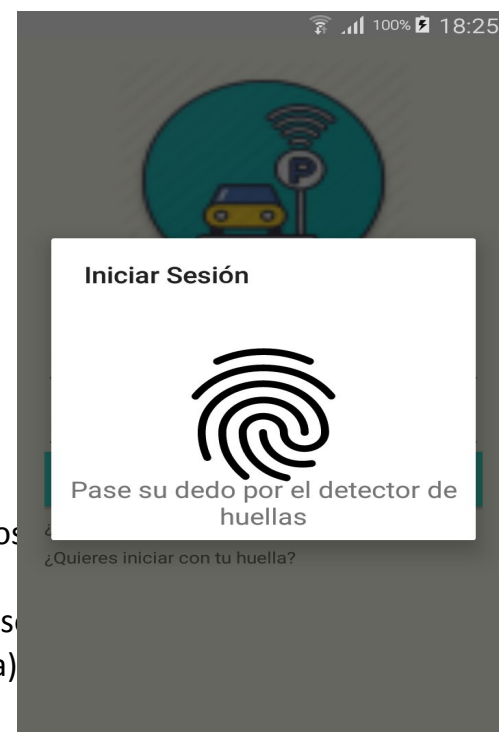
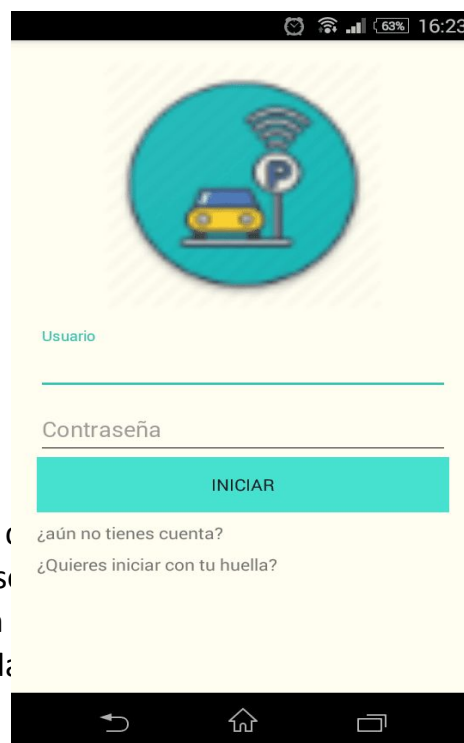
Desarrollamos una aplicación que funcione como intermediario para

comunicarse con el estacionamiento automatizado a través del celular. Para ello creamos en primer lugar las interfaces de usuario. La primera activity que se muestra al iniciarse la aplicación verifica si el usuario ya está ingresado en el sistema o no. En caso de no estarlo se mostrará una pantalla que permite el ingreso de clave y contraseña o el logueo a partir del uso de huella digital. Si el usuario no está registrado en la aplicación puede seleccionar la opción de registro desde la misma pantalla.

Una vez registrado e ingresado en la aplicación se mostrará una pantalla, podrá ver su ubicación en un mapa, donde el usuario visualizara los estacionamientos cercanos. Para ello utilizamos un servicio de google maps. Esta activity contendrá toda la lógica de reserva del sistema y es donde se utilizan los fragmentos para mostrar las reservas activas y generar nuevas. Si pasamos la mano por el frente de la pantalla se mostrarán las reservas. Para generar una reserva simplemente se debe seleccionar el lote que no esté completo y se ingresa el horario deseado. Se identifica si un lote está completo o no por los colores: ROJO (sin disponibilidad) AMARILLO (poca disponibilidad) VERDE (amplia disponibilidad).

Cuando el tiempo de reserva este por llegar a su límite se dará aviso al usuario mediante una notificación, que será enviada por el sistema embebido. Utilizamos el acelerómetro para que al agitar 2 veces el dispositivo se realice un acercamiento o alejamiento en el mapa de ubicación. Además también lo podremos usar para enviarle un mensaje al sistema embebido, este mostrará por su pantalla el nombre del usuario que agito el celular.

## Activities:

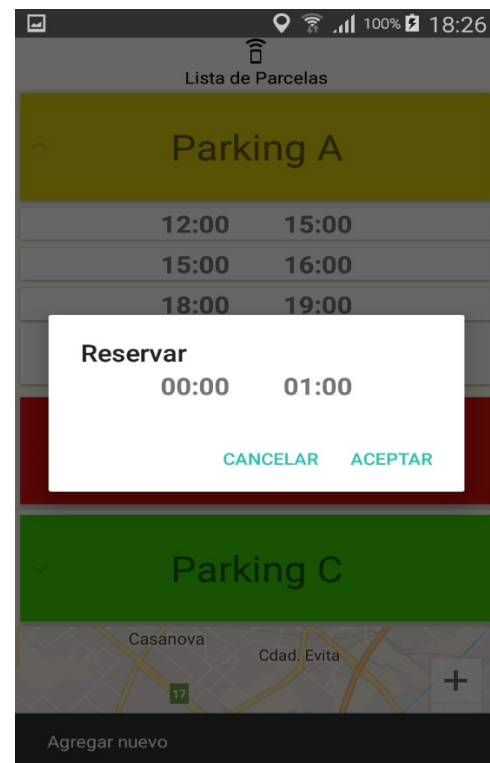
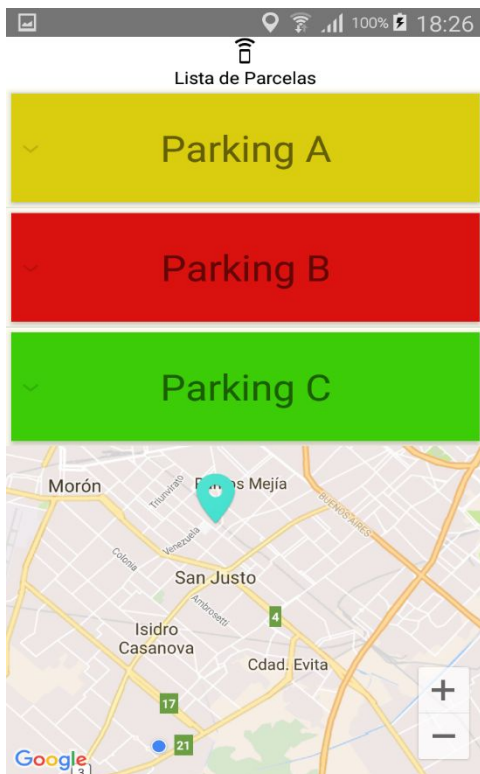


opcionalmente con su huella digital que deberá estar registrada en el sistema, la idea es que el acceso sea fácil e inmediato. Alternativamente se puede realizar el registro para nuevos usuarios si es que no se posee una cuenta y no quiere ingresar su huella digital, o porque no tiene una registrada en el sistema.

- La tercera pantalla contendrá un mapa donde se puede ver todos los estacionamientos activos.

En esta sección podemos desplegar la opción para registrar una reserva a través de 2 formas, pasando la mano sobre el sensor infrarrojo del dispositivo o tocando el marcador en el mapa. Al registrar una reserva enviamos un **TOKEN** a la nube, para que nuestro sistema embebido pueda enviarle notificaciones cuando nuestro tiempo de reserva se este por agotar.

Al agitar el celular también enviaremos una notificación al sistema embebido para mostrar nuestro nombre en la pantalla, el sistema en este caso nos dará un mensaje de bienvenida



- Las imágenes corresponden a la activity principal y a los fragmentos utilizados para realizar las reservas donde implementamos **SensorEventListener** y mediante el método **onSensorChanged()** obtenemos información de los sensores acelerómetro y proximidad.



- o El acelerómetro obtendrá lecturas de las variaciones de movimientos en un lapso de tiempo, y realizar un zoom en el mapa encuadrando su ubicación y estacionamientos haciendo un mejor ajuste de enfoque.
  - o El sensor de proximidad desplegará una barra superior, fragmento que contiene una lista expandible, que mostrará las reservas del estacionamiento.
  - o Para el sensor de huellas utilizamos una SDK, sdk-v1 y pass-v1, de Samsung debido a que las pruebas las realizamos en ese dispositivo. Para comunicarnos con el sensor de huellas implementamos Handler.Callback y usamos el método handleMessage() del cual tendremos el resultado de la lectura realizada por el sensor dactilar
- **AndroidManifest.xml:** es importante destacar que en el archivo manifest, además de declarar todos los componentes utilizados por la aplicación, declaramos los servicios a utilizar como y declaramos los permisos necesarios para el funcionamiento de la misma. Estos fueron: "INTERNET", "USE\_FINGERPRINT", "ACCESS\_COARSE\_LOCATION", "READ\_GSERVICES", etc.
  - Para el manejo de notificaciones debemos crear un proyecto en firebase, luego bajar un archivo json llamado "google-services", este contiene configuración del proyecto para usar el servicio de firebase, y que debe integrarse a nuestro proyecto y agregarlo en nuestro gradle para poder utilizar el motor de firebase. Luego creamos 2 servicios FirebaseAuthService y NotificationService para el registro y comunicación con la nube. La conexión que se establece es remota (socket), cada dispositivo estará conectado al servidor de firebase y el sistema embebido enviará una petición REST a la nube para que él le envíe un mensaje al dispositivo en cuestión.
    - o NotificationService es utilizado para recibir los mensajes a través del método onMessageReceived(), esta conexión es asincronica y no requiere ser lanzada desde ninguna activity
    - o FirebaseAuthService se encarga de obtener el TOKEN de firebase notifications y vincular nuestra sesión con la nube, sin este servicio no podremos recibir mensajes ya que necesita vincularse con el servicio de firebase. Este servicio no requiere ser lanzado desde ninguna activity

- Para el mapa de GoogleMaps tuvimos que crear un proyecto en la consola de desarrollador de google, generar una key debug para poder registrar nuestro proyecto y así este nos generó una API\_KEY para poder utilizar el servicio de geolocalización. Luego de obtener la clave, la agregamos junto con el meta-data declarado en el archivo manifest, para decirle a la aplicación que vamos a utilizar los servicios de googleMaps, y también declaramos los permisos para geolocalización (ACCESS\_COARSE\_LOCATION, READ\_GSERVICES, ACCESS\_FINE\_LOCATION)

## **Comunicación SE + Aplicación Android:**

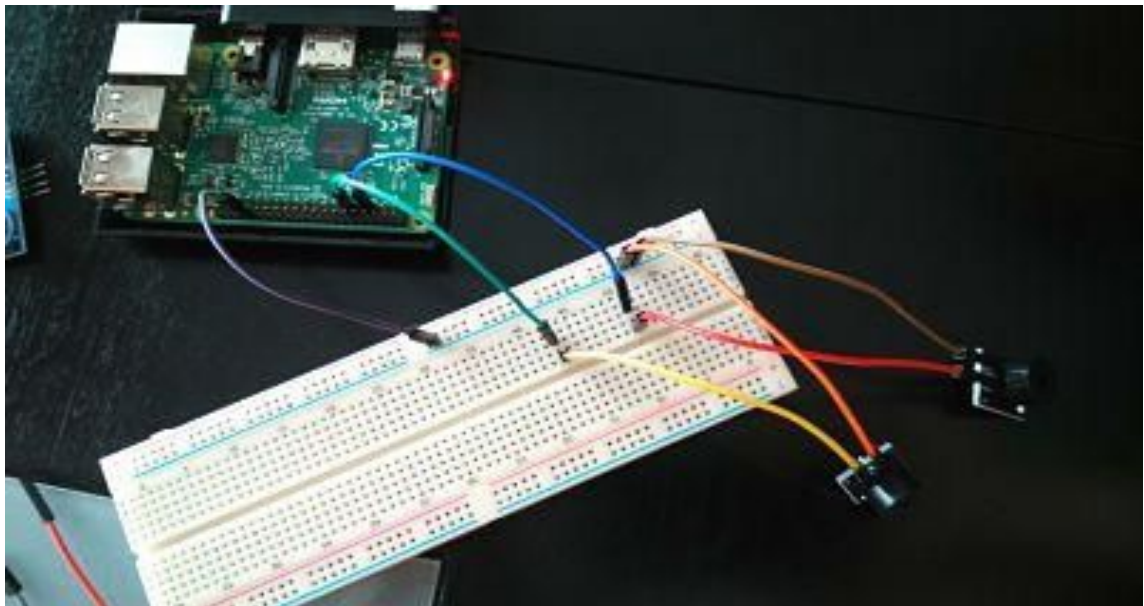
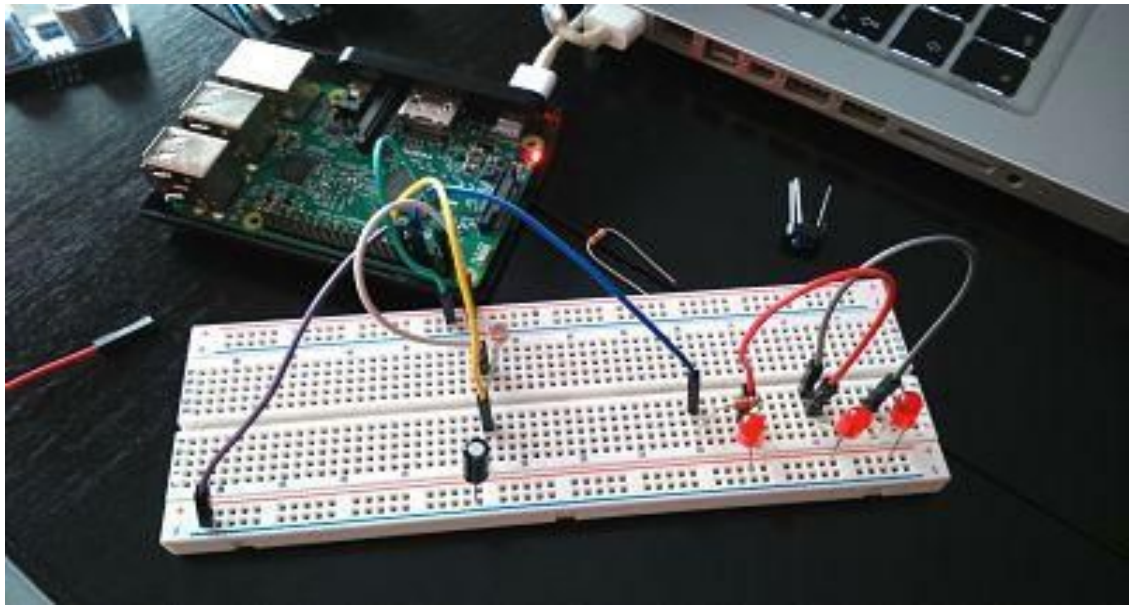
Al no contar con el tiempo adecuado para hacer un servidor dedicado optamos por utilizar la base de datos que nos provee Firebase (base de datos de tiempo real hosteada en la nube de google) y controlamos las altas de reserva desde la propia aplicación sincronizada con Firebase.

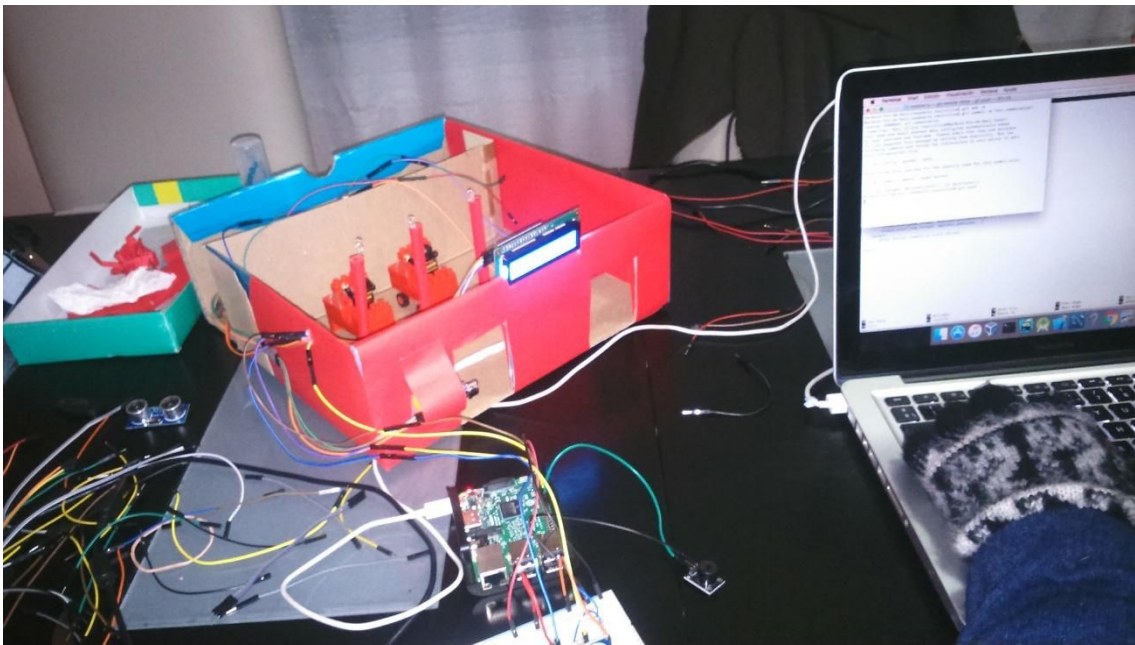
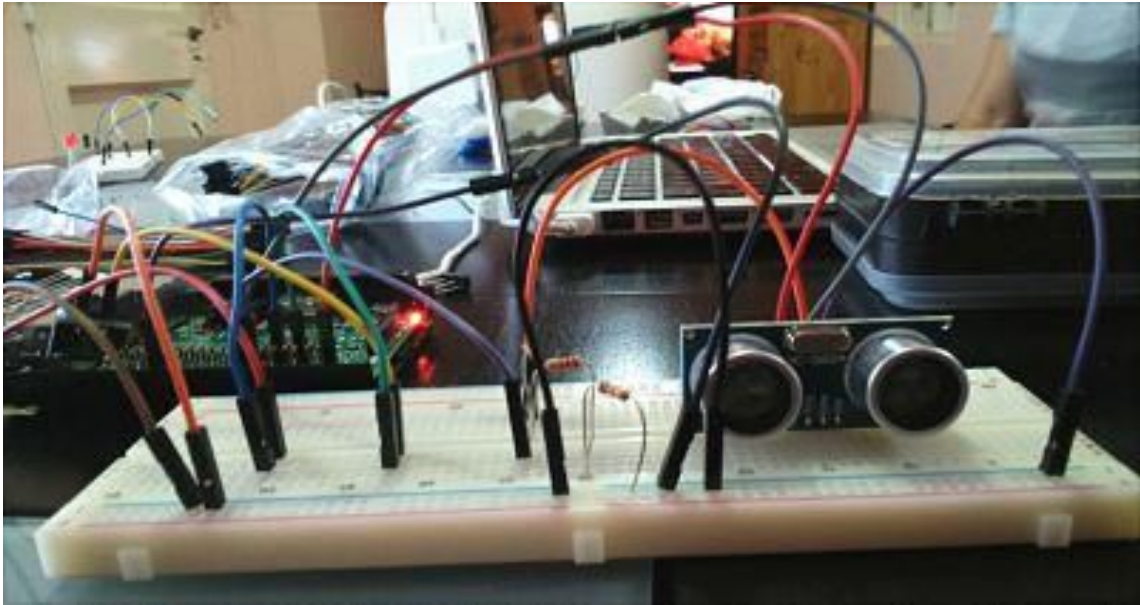
Desde el lado del servidor tuvimos que implementar la api de firebase en Python. Al principio tuvimos problemas al conectarlas debido a que una de las apis recomendadas en firebase no nos funcionaba e instalamos otra que funcionaba correctamente.

Desde la placa se envía una notificación para consultar si existen nuevas reservas y a partir de esta información controlamos el vencimiento del tiempo de las mismas.

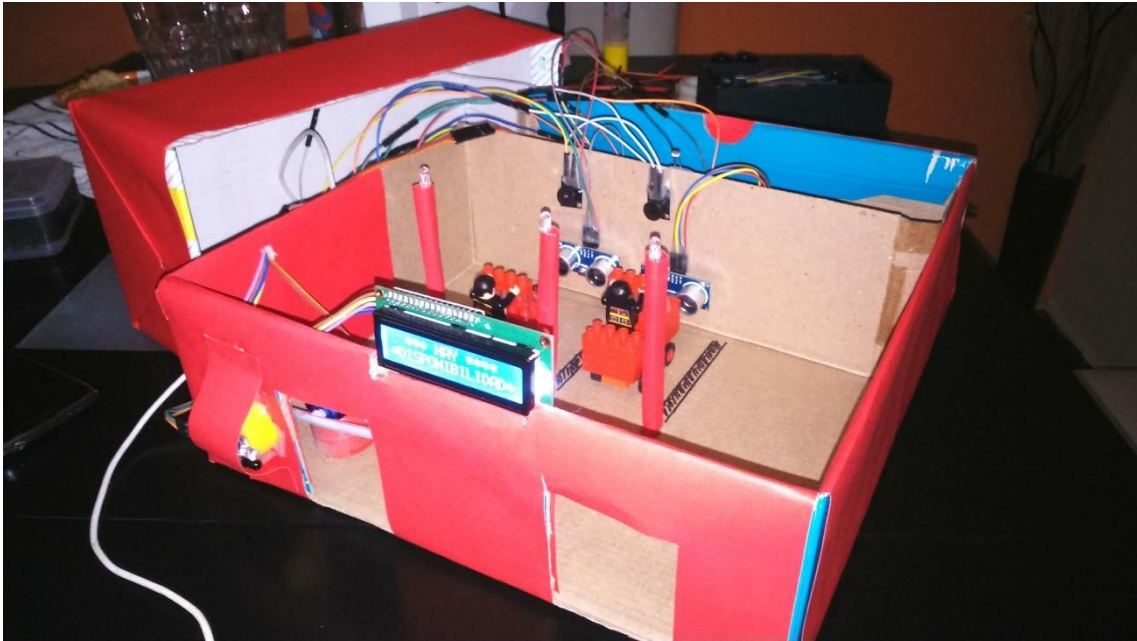
Cuando se vence una reserva las notificaciones se la enviamos a través de una request que nos ofrece la misma página de firebase. El servidor se fija cuál es el token que está registrado para esa request y envía la notificación al usuario.

## **Avance del proyecto en imágenes**









## Conclusiones

Al trabajar sobre nuevas tecnologías muchas veces estuvimos trabados con el proyecto avanzado lentamente por varios días. Todo el proceso del sistema embebido era nuevo para cada uno de los integrantes del equipo y eso nos limitó un poco, sin embargo nos esforzamos para cumplir con los objetivos del proyecto. Nos resultó interesante trabajar con la placa de raspberry y los sensores y ver como de a poco se conectaban los componentes y su funcionamiento.

Con respecto a la aplicación en Android contábamos con conocimientos básicos de desarrollo que nos benefició a la hora de investigar soluciones y trabajar.

Como experiencia, valoramos lo positivo de trabajar con nuevas tecnologías además de profundizar en el mundo de IoT, lo que nos permitió adquirir nuevos e interesantes conocimientos y finalmente tener en nuestras manos una aplicación funcional gracias al esfuerzo y trabajo de cada uno.