

DISPOSITIVO DE REALIDAD AUMENTADA PARA INFORMACIÓN AMBIENTAL Y PERSONAL BÁSICA DEL USUARIO

Brandon Arturo Ruiz García, Ing.
Mauricio Olguín Carbajal, Dr.

Maria Areli Sanchez García, Ing.

Magalli Lora Dolores, Ing.
Israel Rivera Zarate, M. en C.

Instituto Politécnico Nacional
CIDETEC

brandon.ruizgarcia@gmail.com
magalli.lora@outlook.com

mariaareli137@hotmail.com
molguinc@ipn.mx irivera@ipn.mx

Resumen

Se presenta un dispositivo que trabaja con realidad aumentada para mostrar información ambiental e información personal básica de un usuario, todo esto a través de un lente y un micro display ubicados ambos en la parte frontal del dispositivo, utilizando Arduino como tarjeta para el procesamiento de los datos y una aplicación móvil desarrollada para el sistema Android. Con el propósito de ayudar a las personas a mantenerse informadas de sus signos vitales y condiciones climáticas.

Introducción

La realidad aumentada de forma general pretende cambiar el mundo real con el virtual mediante un sistema informático, añadiendo información de interés a nuestra experiencia visual normal.

Gracias a esta tecnología se pueden crear distintos tipos de experiencias interactivas como pueden ser catálogos de productos en 3D, diseño de interiores, video juegos y muchas cosas más. Tomando en cuenta esto se presenta la propuesta de un dispositivo de realidad aumentada que ayude a los usuarios a monitorear sus signos vitales y las condiciones climáticas del punto donde se encuentre, mientras realiza alguna actividad física.

Estos parámetros serán obtenidos por un sensor de frecuencia cardiaca y un sensor de temperatura que serán colocados en la oreja del usuario, estas mediciones estarán disponibles sin ninguna conexión a un Smartphone, y si se tiene alguna conexión con un dispositivo móvil se podrá observar la temperatura ambiente y la ubicación del usuario gracias al uso de la aplicación móvil que se desarrolló.

Desarrollo

El primer módulo que se desarrolló fue el Hardware, utilizando una tarjeta Arduino nano para el procesamiento de los datos obtenidos por los sensores. Para la medición temperatura corporal del usuario se utilizó un sensor LM35 con la salida calibrada a 1°C por cada 10 mV el cual requiere de poca corriente para funcionar por lo que es muy útil para el propósito del proyecto. Para la frecuencia cardiaca se utilizó un sensor de pulso amplificado ADA-1093, que internamente ya cuenta con circuitos de amplificación y de cancelación de ruido y que también tiene un tamaño reducido por lo cual es un elemento funcional para los fines que se buscan.

A continuación se muestra un diagrama de flujo del funcionamiento del código Arduino.

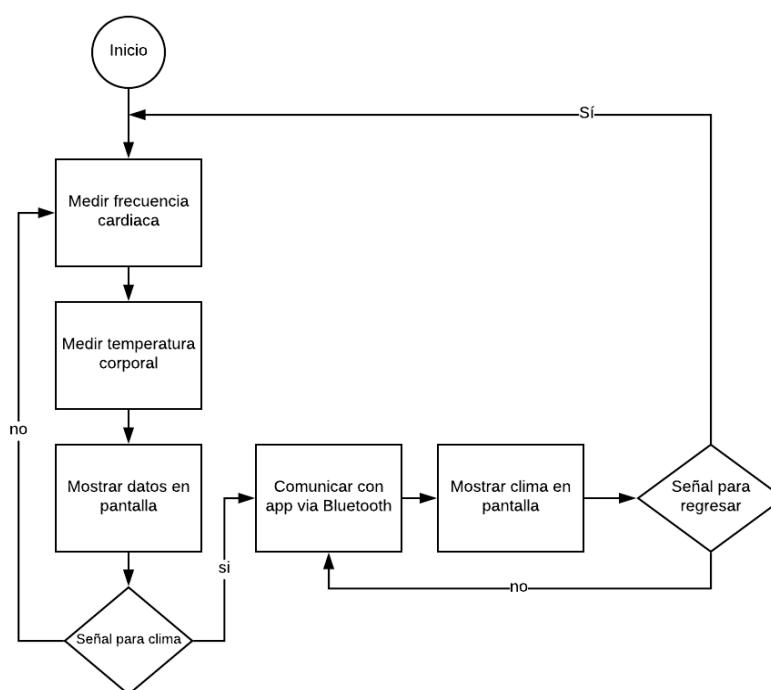


Figura 1. Diagrama de flujo del programa en Arduino

El segundo módulo desarrollado fue la aplicación en Android, donde primeramente es necesario indicar en el archivo **AndroidManifest.xml** los permisos necesarios para el correcto funcionamiento, siendo estos los siguientes: Internet, GPS y Bluetooth.

Estos son agregados utilizando la siguiente sitaxis:

```
<uses-permission Android:name="Android.permission.recurso"/>
```

Los datos obtenidos por el GPS del teléfono inteligente son muy importantes ya que serán los parámetros para consumir la API que brindara la información del clima. Para poder conocer las coordenadas de latitud y longitud dentro la aplicación es necesario hacer uso de un LocationManager, que es la clase que dará acceso al servicio de localización del dispositivo móvil. En el siguiente código se muestra el método para obtener dichos parametros.

```
public void miDireccion(Location loc) throws IOException {  
    double longg, latt;  
    latt = loc.getLatitude();  
    longg = loc.getLongitude();  
    if(latt != 0.0 && longg != 0.0){  
        Geocoder geocoder = new Geocoder(this, Locale.getDefault());  
        List<Address> lista = geocoder.getFromLocation(loc.getLatitude(), loc.getLongitude(), 1);  
        if(!lista.isEmpty()){  
            Address address = lista.get(0);  
            direccion = "Direccion : " + address.getAddressLine(0);  
            tDir = direccion;  
        }  
    }  
}
```

Una vez conocidas las coordenadas del usuario se puede proceder a hacer uso del API de servicio web que brinda la compañía OpenWeather, la cual es gratis y es de tipo REST (Transferencia de Estado Representacional), por lo que son necesarios archivos JSON para poder conocer la información. A continuación se muestra el método para poder utilizar este servicio.

```
public static JSONObject getWeatherJSON(String lat, String lon){  
    try {  
        URL url = new URL(String.format(OPEN_WEATHER_MAP_URL, lat, lon));  
        HttpURLConnection connection =  
            (HttpURLConnection)url.openConnection();  
        connection.setRequestProperty("x-api-key", OPEN_WEATHER_MAP_API);  
        BufferedReader reader = new BufferedReader(  
            new InputStreamReader(connection.getInputStream()));  
        StringBuffer json = new StringBuffer(1024);  
        String tmp="";  
        while((tmp=reader.readLine())!=null)  
            json.append(tmp).append("\n");  
        reader.close();  
    }  
}
```

```
JSONObject data = new JSONObject(json.toString());
if(data.getInt("cod") != 200){
    return null;
}
return data;
}catch(Exception e){
    return null;
}
}
```

La comunicación de la aplicación con Arduino se hizo a través de Bluetooth con un módulo HC-06 de la siguiente manera:

```
private class ConnectAsyncTask extends AsyncTask<BluetoothDevice, Integer, BluetoothSocket>{
    private BluetoothSocket mmSocket;
    private BluetoothDevice mmDevice;
    @Override
    protected BluetoothSocket doInBackground(BluetoothDevice... device) {
        mmDevice = device[0];
        try {
            String mmUUID = "00001101-0000-1000-8000-00805F9B34FB";
            mmSocket =
mmDevice.createInsecureRfcommSocketToServiceRecord(UUID.fromString(mmUUID));
            mmSocket.connect();
        } catch (Exception e) { }
        return mmSocket;
    }
    @Override
    protected void onPostExecute(BluetoothSocket result) {
        btSocket = result;
        //Enable Button
        //btToggle.setEnabled(true);
    }
}
```

De forma general se puede observar la estructura de la aplicación en el siguiente diagrama de clases

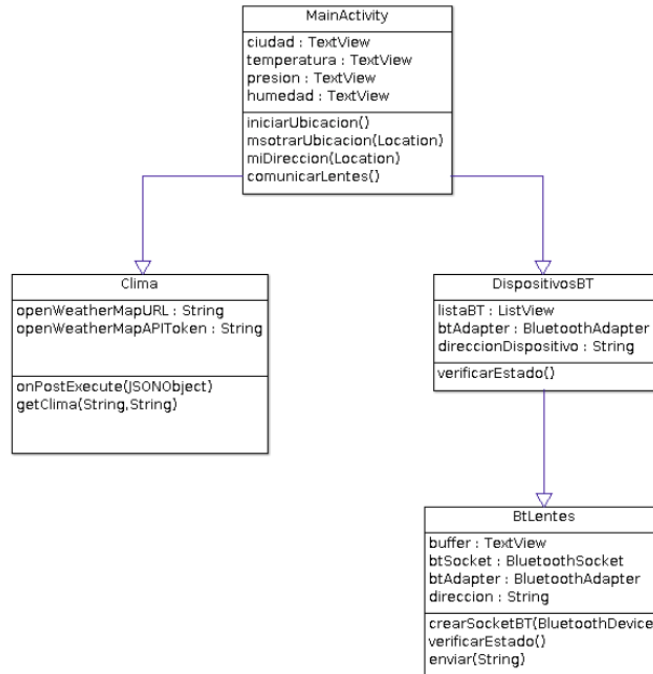


Figura 2. Diagrama de clases de la aplicación Android

Para el diseño de la carcasa donde se encuentra el circuito se utilizó Blender ya que este Software cuenta con las herramientas necesarias para poder exportar nuestro archivo a una impresora 3D. En la siguiente imagen se puede ver el diseño.

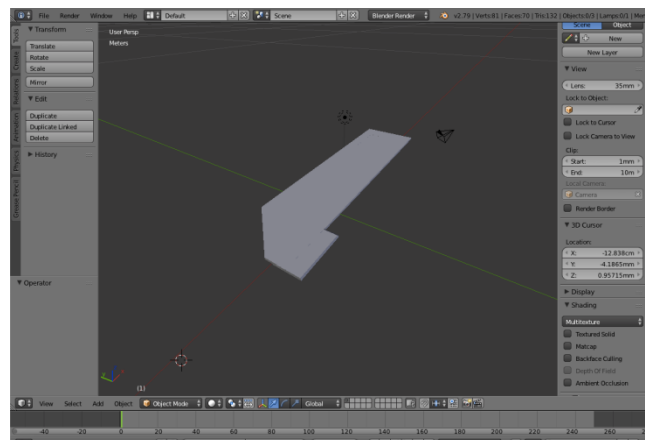


Figura 3. Diseño en blender de la carcasa

Por último el diseño final del dispositivo cuenta con un lente el cual está colocado a 45° con la finalidad de solventar el problema del reflejo del texto de manera invertida, A continuación se muestra una imagen del prototipo.



Figura 4. Diseño final de los lentes

Conclusiones

Arduino es una tarjeta basada en un microcontrolador ATmega328, y gracias a esto permite la elaboración de proyectos a un bajo costo y con buen rendimiento, además de que el tamaño de este permite crear dispositivos de volúmenes reducidos, lo cual es una parte importante para el desarrollo de este proyecto.

En este proyecto se utilizaron distintas tecnologías que ayudaron al correcto funcionamiento, en un artículo anterior se mostraron los avances que se tenían al momento, para este se muestra el proyecto en su totalidad.

Referencias

Innovae. (2015). ¿Qué es la realidad aumentada?, Recuperado: Marzo 20, 2019 from: <http://realidadaugmentada.info/tecnologia/>

S.Quiand, W. Jark. (1998). The penta-prism LTP_ long-trace-profiler with stationary optical fead and moving penta prism, AIP Review of Scientific Instruments

S.Inagaki, Yoshihisa Nishigori. (1998). Head-mounted display US5742264 A, Matsushita Electric Industrial Co. Ltd

Arduino. Arduino Nano Recuperado: Marzo 20, 2019 from: <https://www.arduino.cc/en/Main/arduinoBoardNano>

Texas Instruments. (1999). LM35 Precision Centigrade Temperature Sensors Recuperado: Marzo 20, 2019 from: <http://www.ti.com/lit/ds/symlink/lm35.pdf>

AG Electronica (2015). Sensor de pulso amplificado Recuperado: Marzo 20, 2019 from: <http://www.agspecinfo.com/pdfs/A/ADA1093.PDF>