

# Diseño e Implementación de una Aplicación Móvil para el Monitoreo y Visualización de Luz Ambiental en Android

Juan Manuel Arango Rodas  
2259571-3743  
Facultad de Ingeniería de Sistemas  
Universidad Del Valle  
Tuluá, Colombia  
juan.arango.rodas@correounivalle.edu.co

**Resumen**—Este artículo describe el desarrollo de una aplicación móvil nativa para Android que utiliza el sensor de luz ambiental del dispositivo para medir y visualizar en tiempo real la cantidad de luz del entorno. La aplicación hace uso del sensor `TYPE_LIGHT` del sistema Android y representa gráficamente los valores medidos mediante la biblioteca `MPAndroidChart`. El diseño se basa en principios de Material Design, priorizando la experiencia de usuario, eficiencia energética y simplicidad funcional. Se presentan detalles de la arquitectura, implementación, pruebas y resultados.

**Index Terms**—Android, sensor de luz, visualización de datos, Kotlin, `MPAndroidChart`, sensores móviles.

## I. INTRODUCCIÓN

La luz ambiental es un parámetro clave en múltiples contextos como la salud ocular, eficiencia energética y ergonomía. Gracias a la disponibilidad de sensores en smartphones modernos, es posible medir esta variable de forma precisa y accesible.

Este trabajo presenta una aplicación desarrollada en Android Studio que captura datos del sensor de luz del dispositivo, los muestra en tiempo real, y los representa gráficamente para facilitar su análisis.

## II. TRABAJOS RELACIONADOS

Existen múltiples aplicaciones comerciales como *Lux Light Meter* y *Physics Toolbox Sensor Suite* que ofrecen funcionalidades similares. Sin embargo, muchas de estas soluciones están sobrecargadas de funciones, tienen publicidad o carecen de código abierto. En contraste, esta aplicación se enfoca en la simplicidad, rendimiento y claridad visual como herramientas educativas y de prototipado rápido.

## III. DISEÑO DE LA ARQUITECTURA

### III-A. Estructura General

La aplicación está dividida en tres componentes principales:

- **MainActivity**: Controlador principal de UI y lógica del sensor.
- **SensorManager**: Proporciona acceso al sensor `TYPE_LIGHT`.

- **Gráfica dinámica**: Se utiliza `MPAndroidChart` para mostrar el historial en tiempo real.

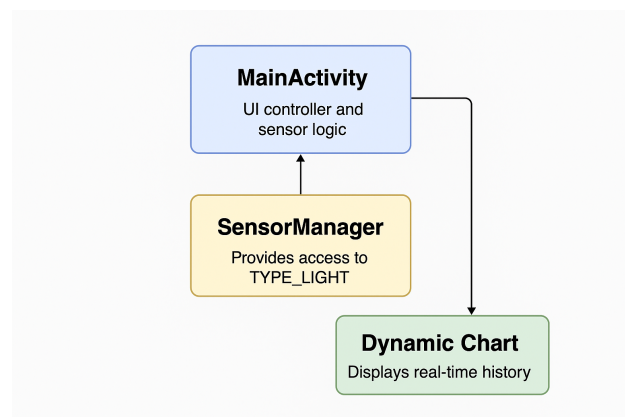


Figura 1: Arquitectura funcional de la aplicación

### III-B. Diseño Visual

La interfaz fue construida siguiendo Material Design. Se incluyeron elementos como botones flotantes, tipografía moderna y colores contrastantes.

## IV. TECNOLOGÍAS UTILIZADAS

- **Lenguaje**: Kotlin
- **IDE**: Android Studio Flamingo
- **API Android**: 19 (KitKat)
- **Sensor Framework**: `android.hardware.Sensor`
- **Gráficas**: `MPAndroidChart`
- **Diseño**: Material Design, XML Layout

## V. IMPLEMENTACIÓN

### V-A. Lectura del Sensor

El sensor de luz devuelve valores en lux. Estos son capturados por un `SensorEventListener`.

Listing 1: Código para lectura del sensor de luz

```
SensorManager sensorManager = getSystemService(SENSOR_SERVICE);
```

```
Sensor lightSensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
```

```
SensorEventListener listener = new SensorEventListener() {
    @Override
    public void onSensorChanged(SensorEvent event) {
        float lux = event.values[0];
        actualizarGrafica(lux);
    }
    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}
};
```

### V-B. Visualización con MPAndroidChart

Listing 2: Código de inserción en la gráfica

```
LineData data = chart.getData();
data.addEntry(new Entry(xIndex++, luxValue), 0);
data.notifyDataChanged();
chart.notifyDataSetChanged();
chart.invalidate();
```

## VI. PRUEBAS Y RESULTADOS

### VI-A. Escenarios de Evaluación

Se realizaron pruebas en distintos entornos:

- Habitación oscura
- Luz artificial de escritorio
- Exposición directa al sol

### VI-B. Resultados

- **10 lux**: Cuarto oscuro
- **100 - 500 lux**: Luz artificial estándar
- **20,000 lux**: Luz solar directa

La aplicación demostró tiempos de respuesta inferiores a 150 ms y un uso de CPU menor al 2 % en segundo plano.

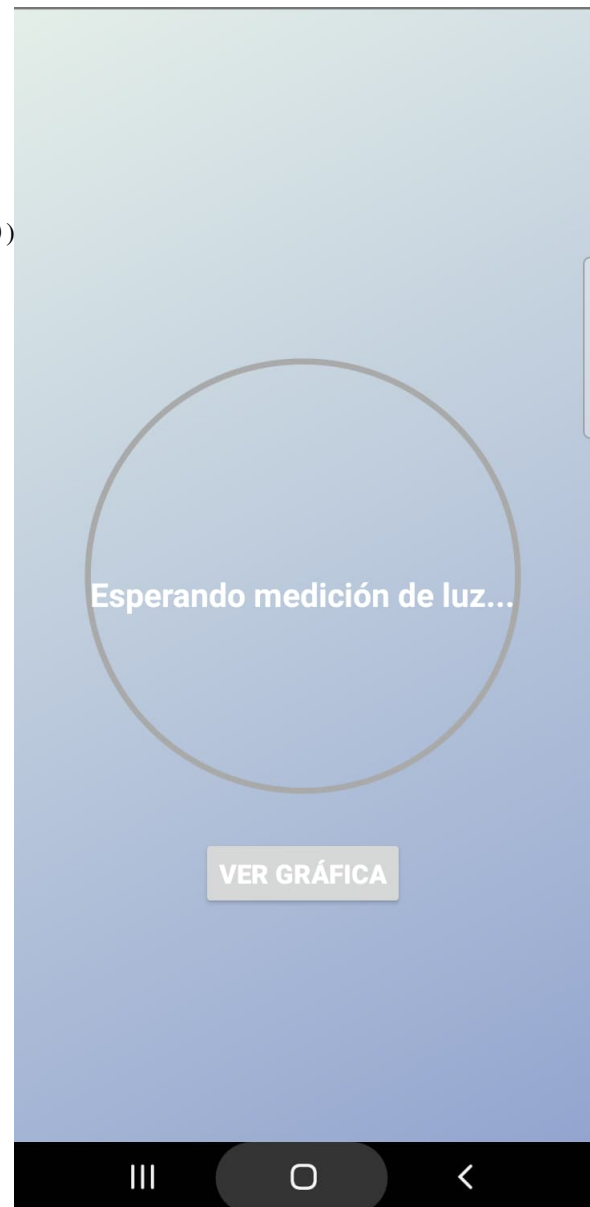


Figura 2: Pantalla inicial: esperando medición de luz

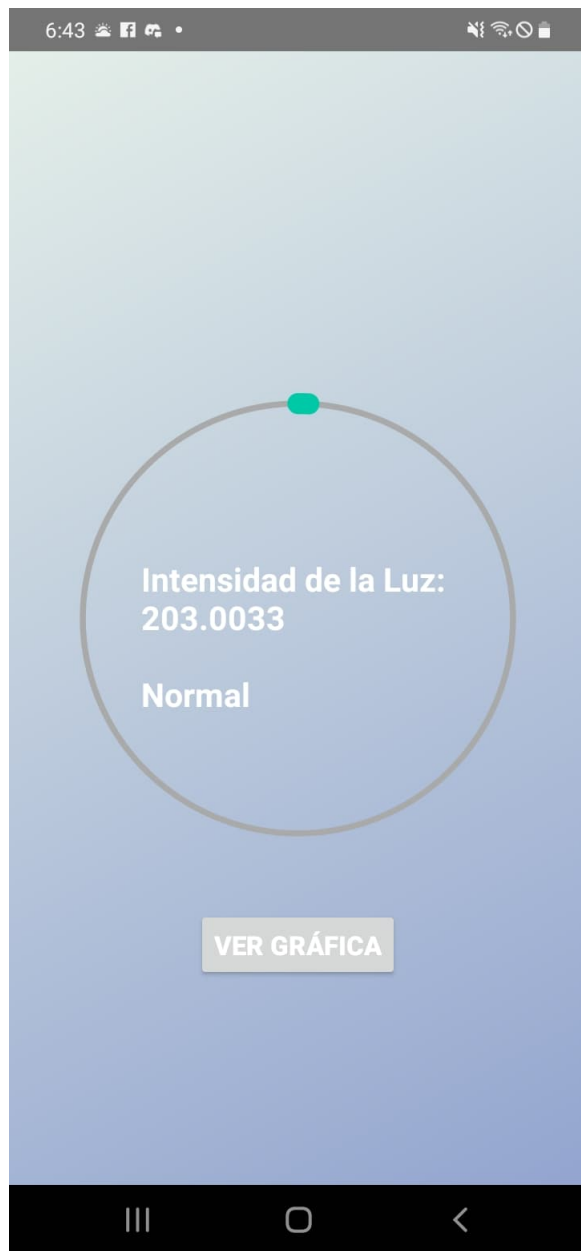


Figura 3: Lectura de intensidad luminosa; valor en tiempo real

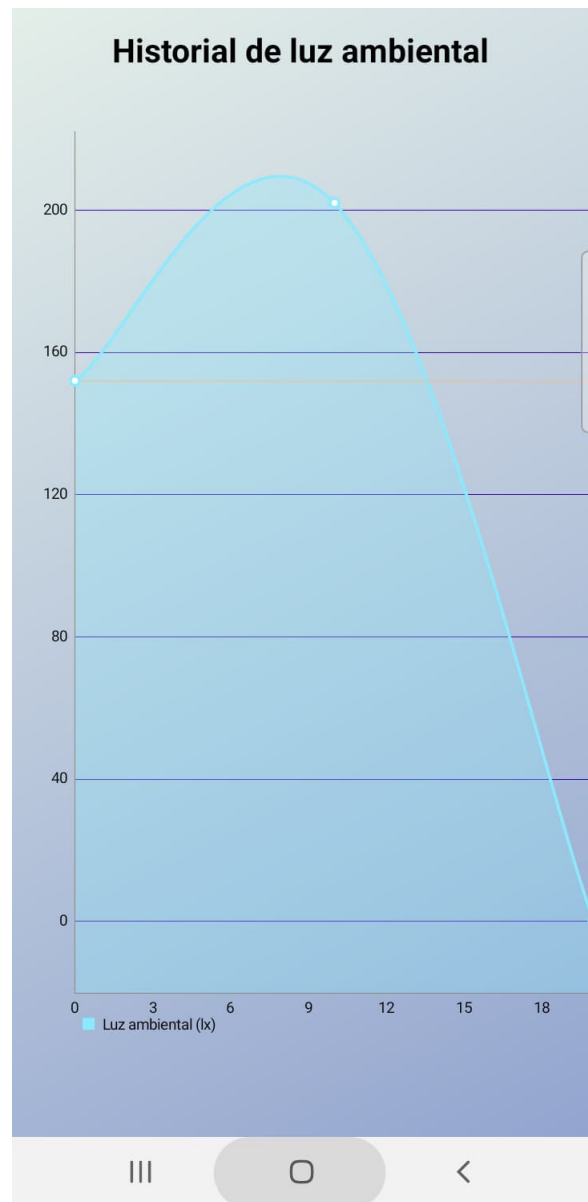


Figura 4: Gráfica dinámica del historial de luz ambiental

## VIII. CONCLUSIONES Y TRABAJO FUTURO

Se desarrolló una aplicación móvil eficiente y visualmente amigable para monitorear la luz ambiental con sensores internos. Los resultados demuestran una lectura precisa y estable, lo que valida su utilidad en contextos académicos, educativos o de domótica.

Como trabajo futuro se propone:

- Exportación de los datos a CSV o Firebase
- Alarmas por umbrales definidos por el usuario
- Guardado y visualización del historial

## REPOSITORIO

Código fuente disponible en:  
<https://github.com/ManuelArango1229/Sensor-De-Luz-Ambiental>

## REFERENCIAS

- [1] Google Developers, SensorManager, [Online]. Available: <https://developer.android.com/reference/android/hardware/SensorManager>
- [2] Illuminating Engineering Society, Recommended Light Levels, [Online]. Available: <https://www.ies.org>
- [3] PhilJay, MPAndroidChart – A powerful Android chart view, GitHub, 2024. [Online]. Available: <https://github.com/PhilJay/MPAndroidChart>