



Instituto Politécnico Nacional  
Escuela Superior de Cómputo



❖ **INTEGRANTES DEL EQUIPO:**

- Barrera Perez Carlos Tonatihu
  - López Higuera Antonio
- López Sánchez Gustavo Andrés
  - Monroy Martos Elioth

❖ **UNIDAD DE APRENDIZAJE:** Application Development for Mobile Devices

❖ **PROFESOR:** Alejandro Sigfrido Cifuentes Alvarez.

❖ **PROYECTO TERCER PARCIAL:** Visualización de una cámara remota desde Android..

❖ **FECHA:** 28 de mayo de 2019.

❖ **Versión del reporte:** 1.0

❖ **GRUPO:** 3CM8.

# Índice

<b>Índice</b>	<b>2</b>
<b>Objetivo</b>	<b>3</b>
<b>Conceptos</b>	<b>3</b>
<b>Desarrollo</b>	<b>4</b>
Descripción	4
Desarrollo del servidor web	4
Desarrollo de la aplicación móvil	4
Software y hardware utilizados	4
Código fuente	5
Pruebas	8
<b>Conclusiones</b>	<b>10</b>
Dificultades encontradas	10
Posibles aplicaciones	10
Conclusiones individuales	10
Barrera Perez Carlos Tonatihu	10
López Higuera Antonio	10
López Sánchez Andrés	11
Monroy Martos Elioth	11
<b>Bibliografía</b>	<b>11</b>

# Objetivo

Desarrollar una aplicación móvil, la cual permite observar de forma remota, lo que se graba en una cámara de un dispositivo distinto al que se usa para

# Conceptos

Para el entendimiento de esta proyecto es necesario conocer los siguientes conceptos los cuales fueron clave durante el desarrollo de esta aplicación.

- **Java.** Es un lenguaje de programación orientado a objetos, compilado e interpretado por su máquina virtual, Java puede ser utilizado en prácticamente cualquier tipo de proyecto debido a su portabilidad.[1]
- **Android.** Es un sistema operativo para dispositivos móviles, basado en el núcleo de linux, diseñado principalmente para dispositivos táctiles.[2]
- **Activity Android.** Una actividades es un punto de entrada para la interacción con el usuario. Representa una pantalla con una interfaz de usuario.[2]
- **Android Manifest.** Es un archivo que describe la información esencial de la aplicación a las herramientas de construcción de Android, al sistema operativo y a Google Play. En esta se describe el nombre del paquete, los componentes de la aplicación (incluye todas las actividades, servicios, *content provides*, entre otros), los permisos que requiere la aplicación, entre otras declaraciones esenciales. [2]
- **FPV.** El pilotaje por visión remota (FPV por sus siglas en inglés), consiste en dirigir un modelo teledirigido (avión, helicóptero, dron, coche, etc.) por medio de una cámara de vídeo a distancia y de una pantalla o unas gafas de vídeo.[3]
- **Python.** Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. [4]
- **OpenCV.** Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X, Windows y Android. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estérea y visión robótica. El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo. [5]

# Desarrollo

## Descripción

El desarrollo del proyecto se dividió en dos partes, la primera fue el desarrollo de un servidor web y un script en python los cuales pudieran tomar el video de la cámara de una laptop. La segunda parte consiste en el desarrollo de una aplicación móvil la cual permita acceder a la imagen generada por la cámara.

## Desarrollo del servidor web

Para el desarrollo del servidor, se usó flask y python, los cuales permiten servir archivos estáticos de manera fácil, además se desarrolló un script en python haciendo uso de open-cv, el cual permite capturar una imagen de la cámara de la laptop cada cierto tiempo.

## Desarrollo de la aplicación móvil

Para la aplicación móvil, se desarrolló una interfaz muy sencilla usando solamente un toolbar por defecto y un ImageView, en cual se iba actualizando la imagen recibida. Para lograr esto, se usó la librería ImageLoader, la cual permite descargar una imagen desde una uri especificada. Además de que se usó un handler para poder realizar esta acción cada determinado tiempo.

## Software y hardware utilizados

A continuación, se presenta una lista del software y hardware usado para la elaboración del proyecto:

- **Android Studio 3.4.** Android IDE para el desarrollo de aplicaciones para el sistema operativo Android.
- **Teléfono con Android 5+.** Usado para realizar la pruebas correspondientes.
- **Java.** Lenguaje de programación multipropósito con el cual fue programada la aplicación móvil y el servidor web.
- **Java Development Kit 8.** Necesario para el desarrollo de aplicaciones Java.
- **Cámara de laptop.** Se utiliza como cámara remota.
- **Python 3.** Lenguaje de programación interpretado en el cual fue desarrollado el script para la obtención de las imágenes de la cámara de la laptop y también el servidor web.
- **Flask.** Framework para el desarrollo de aplicaciones web en python, fue usado para crear el servidor web que sirve el archivo estático.

- **OpenCV.** Librería que permite trabajar con imágenes de diversas formas, aplicando filtros. reconocimiento de patrones y otras cosas, fue utilizada para capturar imágenes desde la cámara de la laptop.

## Código fuente

A continuación, se presenta el código más importante elaborado, para el desarrollo del proyecto.

El siguiente código, muestra el script usado para obtener cada .8 segundos un frame de la cámara de la laptop.

camara.py

```
import cv2
import threading, time
import pickle
cap=cv2.VideoCapture(0);
while(True):
    leído, frame=cap.read()
    if leído==True:
        cv2.imwrite("static/image.jpg", frame)
    else:
        pass
    time.sleep(.8)
cap.release()
```

El siguiente archivo, muestra el código usado para ejecutar el servidor que sirve los archivos estáticos y por ende, permite que pueda ser accesada la imagen desde otro dispositivo en la misma red, en este caso, el dispositivo móvil android.

servidor.py

```
from flask import Flask, request, render_template, send_from_directory
app=Flask(__name__)
@app.route('/', methods=['GET', 'POST', 'HEAD', 'PUT', 'DELETE', 'CONNECT', 'OPTIONS', 'TRACE', 'PATCH'])
def hola():
    return "Hola mundo"
if __name__ == '__main__':
    app.debug=False
    app.run(debug=False, port=5000, host='0.0.0.0')
```

A continuación, se muestra el archivo manifest realizado para que la aplicación tuviera acceso a Internet.

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.webcam">
    <uses-permission android:name="android.permission.INTERNET" />
```

```

<application
    android:usesCleartextTraffic="true"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>
</manifest>

```

La clase principal de la aplicación Android es el siguiente:  
 ActivityMain.java:

```

package com.example.webcam;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Handler;
import android.os.StrictMode;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import com.nostra13.universalimageloader.core.ImageLoader;
import com.nostra13.universalimageloader.core.ImageLoaderConfiguration;
import com.nostra13.universalimageloader.core.assist.FailReason;
import
com.nostra13.universalimageloader.core.listener.ImageLoadingListener;
import
com.nostra13.universalimageloader.core.listener.SimpleImageLoadingListene
r;
import java.io.InputStream;
import java.net.Socket;
public class MainActivity extends AppCompatActivity {
    public final String url="http://10.100.64.27:5000/static/image.jpg";
    ImageView imagen;
    ImageLoader imageLoader;
    public Socket cliente;
    public String HOST="10.100.64.27";
    public int PUERTO=8485;
    InputStream socketInputStream;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle("Proyecto 2");

        imageLoader = ImageLoader.getInstance(); // Get singleton instance

```

```

        imagen=findViewById(R.id.image);

        ImageLoader.getInstance().init(ImageLoaderConfiguration.createDefault(MainActivity.this));
        imageLoader.displayImage(url, imagen);
        continuar();
    }
    public void continuar() {
        final Handler handler = new Handler();
        final Runnable runnable[] = new Runnable[1];
        runnable[0] = new Runnable() {
            @Override
            public void run() {
                try {
                    imageLoader.displayImage(url, imagen);
                } catch (Exception e) {
                    e.printStackTrace();
                }
                handler.postDelayed(runnable[0], 100);
            }
        };
        handler.post(runnable[0]);
    }
}

```

La vista para la clase anterior, es la siguiente:

activity\_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/image"/>
</android.support.constraint.ConstraintLayout>

```

## Pruebas

A continuación se muestran algunas capturas de pantalla de la aplicación móvil. En la cual se muestran imágenes de lo que se encontraba capturando la cámara de la laptop. Las Figuras 1, 2 y 3 muestran las ya mencionadas capturas de pantalla.

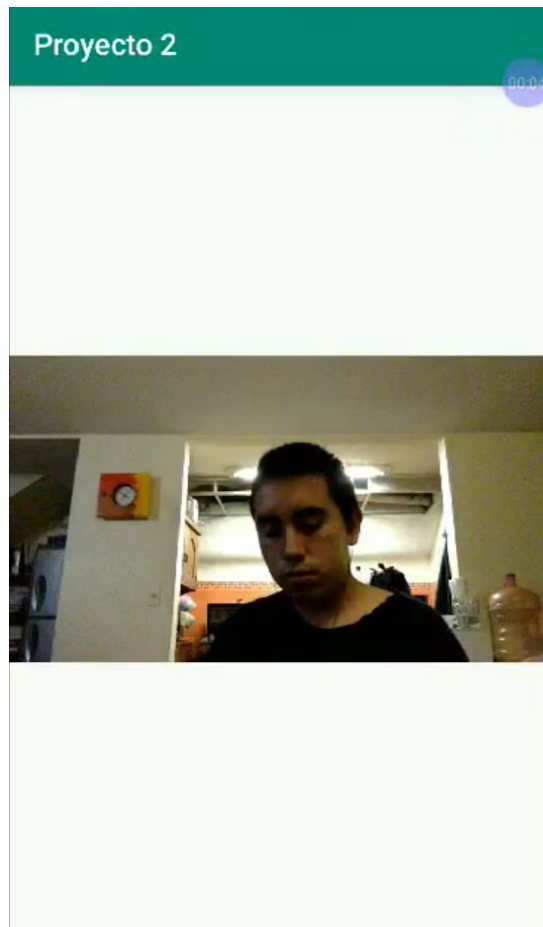


Figura 1. Captura de pantalla de la aplicación (1).



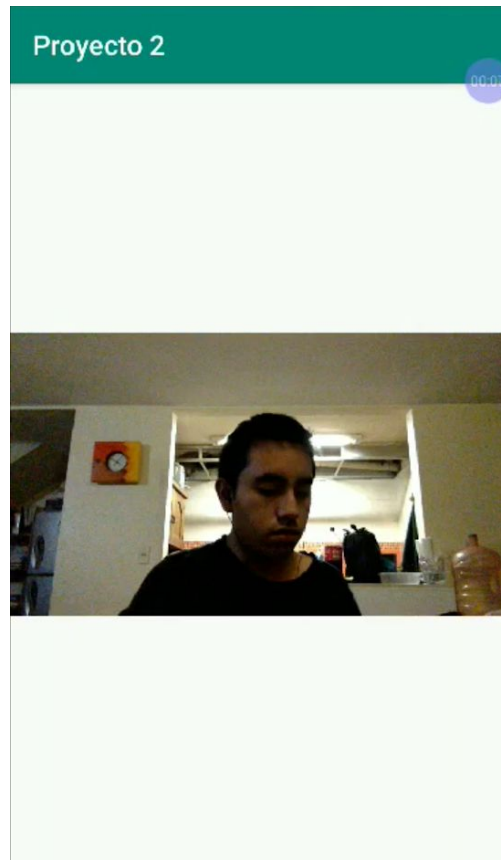


Figura 2. Captura de pantalla de la aplicación (2).

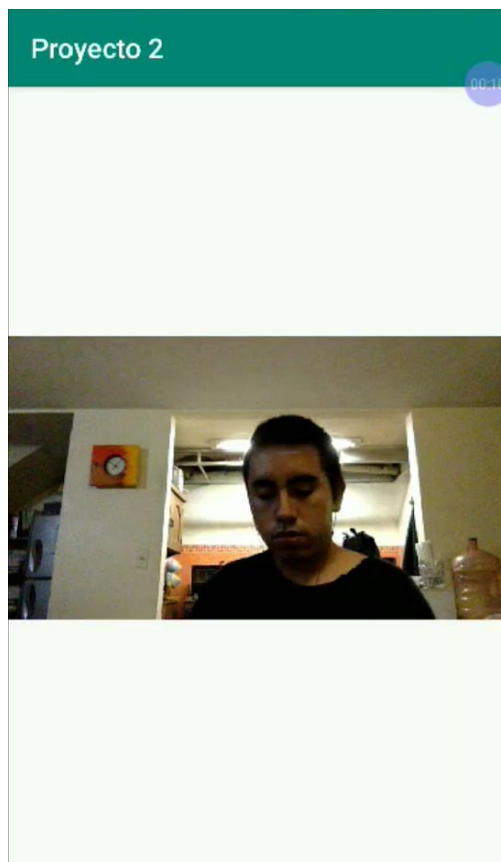


Figura 3. Captura de pantalla de la aplicación (3).

# Conclusiones

## Dificultades encontradas

Resultó complicado lograr que tanto la aplicación móvil y el script de la cámara estuvieran sincronizados, para así evitar que se llegaran a descargar imágenes truncas o incompletas, por lo cual se utilizó un tiempo de ejecución del hilo de cada una de ellas similar, para así intentar solucionar el problema, lo cual funcionó siempre que este tiempo de espera fuera aumentando acercándose al segundo.

## Posibles aplicaciones

Poder monitorear diferentes dispositivos vía remota, ya que con solo conocer la ip del dispositivo es posible acceder a la cámara del mismo, por lo cual se pueden crear sistemas que permitan monitorear lugares como casas, teniendo así un sistema mínimo de seguridad. Además, también puede ser usado para implementar sistemas que muestren la imagen que manda un dispositivo como un dron o un robot.

## Conclusiones individuales

### Barrera Perez Carlos Tonatihu

En un inicio no se tenía claro cómo realizar este proyecto por lo cual a la hora de investigar se encontraron varias formas de poder realizarlo, se eligió utilizar openCV debido a que es una biblioteca bastante robusta al trabajar con imágenes y videos por lo que facilitó bastante el trabajo. Además, lo que se realizó en este proyecto se puede aplicar con cualquier dispositivo que cuente con cámara y que tenga alguna forma de comunicación y que se pueda programar como es el caso de cámaras de seguridad o drones.

### López Higuera Antonio

Los dispositivos móviles actualmente pueden ayudarte a realizar todo tipo de acciones y la interconectividad que desees es posible, desde poder transmitir datos básicos, hasta imágenes o vídeo de manera instantánea. El poder realizar estos proyectos te hacen aplicar conocimientos adquiridos durante toda la carrera, que en ocasiones creemos que jamás son aplicados, pero distintos conceptos son necesarios y bien aplicados.

## López Sánchez Andrés

La realización del proyecto resultó un reto pues el enviar información de un dispositivo a otro de manera constantes tiene una complejidad por diferentes factores, ya sea la latencia, interferencia, etc. es por ello se decidió utilizar openCV ya que proporciona una biblioteca fácil de integrar y que proporciona la comunicación entre los dispositivos de manera intuitiva, sin embargo, fue necesario también implementar los conocimientos previamente adquiridos para poder interpretar la información en una aplicación y consumir el servicio en segundo plano.

## Monroy Martos Elioth

Android nos permite hacer casi todo tipo de proyectos, para muestra este proyecto el cual combina aspectos de programación Android, servidores web, consumo de servicios web, e inclusive hace uso del hardware de un laptop, en específico de la cámara. por lo cual con este proyecto pudimos implementar conocimientos de diversas áreas de nuestra carrera, para así crear una pequeña aplicación la cual tiene una funcionalidad mínima, pero muy interesante. Como lo es, la posibilidad de monitorear una cámara remota de un dispositivo externo a nuestro teléfono.

## Bibliografía

- [1] M. Lozano Ortega and B. Bonev, "Introducción a Android", *Curso de Android y Java para Dispositivos Móviles*, 2010. [Online]. Disponible en: <http://www.jtech.ua.es/apuntes/ajdm2010/sesiones/sesion09-apuntes.html>. [Consultado: 10- Abril- 2019].
- [2] "Application Fundamentals", Android Developer, 2019. [Online]. Disponible en: <https://developer.android.com/guide/components/fundamentals>. [Consultado: 10-Abril-2019].
- [3] "Pilotaje con visión remota", Wikipedia, 2017. [Online]. Disponible en [https://es.wikipedia.org/wiki/Pilotaje\\_con\\_visi%C3%B3n\\_remota](https://es.wikipedia.org/wiki/Pilotaje_con_visi%C3%B3n_remota). [Consultado: 28-Mayo-2019].
- [4] "Python" Wikipedia, 2017. [Online]. Disponible en: <https://es.wikipedia.org/wiki/Python> [Consultado: 29-mayo-2019].
- [5] "OpenCV" Wikipedia, 2017. [Online]. Disponible en: <https://es.wikipedia.org/wiki/OpenCV> [Consultado: 29-mayo-2019].