

Aplicación móvil para el cálculo del consumo eléctrico en dispositivos de medición digital.

Mobile application for the calculation of electricity consumption in digital meter devices

Bixbi Tapia Sueldo¹, <https://orcid.org/0000-0002-6819-3270>

Manuel J. Ibarra², <https://orcid.org/0000-0001-6711-4916>

¹Universidad Nacional Micaela Bastidas de Apurímac, bixbi_ts@gmail.com, Abancay, Perú

²Universidad Nacional Micaela Bastidas de Apurímac, manuelibarra@gmail.com, Abancay, Perú

Resumen - El objetivo de esta investigación fue desarrollar una aplicación móvil basada en Reconocimiento Óptico de Caracteres (OCR) para reducir los errores en las lecturas de los medidores de energía eléctrica que se presentan mensualmente en la empresa Electro Sur Este. Estos errores perjudican a los clientes, quienes no están satisfechos con sus facturas y deben incurrir en gastos innecesarios para corregirlos. La investigación fue experimental, con preprueba y posprueba, y se realizó con un diseño de grupo único. La metodología aplicada se basó en un proceso ágil iterativo e incremental. La muestra estuvo compuesta por 295 medidores de usuarios de la provincia de Grau, distrito de Chuquibambilla, que opera con el alimentador 01. Los resultados de la usabilidad de la aplicación móvil para la lectura de los medidores digitales muestran que se logró reducir los errores de 9 a 0, disminuyendo también el tiempo de lectura y procesamiento de datos; se redujo el número de lecturas anómalas y, finalmente, el número de quejas de los usuarios por consumo excesivo disminuyó de 18 a 9. En conclusión, la aplicación móvil basada en OCR es una herramienta eficaz para la reducción de errores en las lecturas de medidores eléctricos digitales.

Palabras clave: OCR, lectura medidor, aplicación móvil, Android, luz

Abstract - The purpose of this research was to develop a mobile application based on (Optical Character Recognition), to reduce errors in the readings of electric power meters that are presented monthly in Electro Sur Este company, which are harmful for customers, they are not satisfied with their invoices and have to make unnecessary expenses to correct the errors. The research was experimental with pre-test and post-test with a single group design. The methodology was applied based on an iterative and incremental agile process. The sample was of 295 meters of users of the Province of Grau, district of Chuquibambilla that work with 01 feeder. The results of Mobile Application usability for the reading the digital meters machine show that it was possible to reduce from 9 errors to 0 errors, and the time of reading and processing of data also decreased; the number of anomalous readings was also reduced, and finally the number of users's complaints reduced from 18 to 9 related to excessive consumption. Finally, the conclusion is that Mobile Application based on Optical Character Recognition is a tool for the reduction of errors of readings of digital electric meters.

Keywords: OCR, meter reading, mobile app, Android, light

1. INTRODUCCIÓN

La visión artificial ha evolucionado rápidamente en estos últimos años, gracias a sus técnicas y métodos basados en la captura de imágenes que posteriormente son procesadas de forma digital para así extraer información requerida.

Este trabajo de investigación reduce la cantidad de errores en la lectura de los medidores eléctricos analógicos usando una aplicación móvil basada en el reconocimiento óptico de caracteres (OCR) en Electro Sur Este S.A.A, obteniendo datos confiables y más seguros. OCR es un mecanismo de reconocimiento óptico de caracteres que permite reconocer los caracteres o números de

una imagen dada (1)(2)y tiene sus etapas definidas. (3)

El trabajo de investigación también se centra en reducir tiempo por lectura realizada de los medidores, detectar las lecturas anómalas que se encuentra fuera del rango de consumo de un cliente normal, para posteriormente dar una solución inmediata por los técnicos de mantenimiento, y por último disminuir los reclamos por excesivos consumos facturados realizado mensualmente por los clientes descontentos. (ISO 9001).

Esta investigación es aplicada de nivel experimental y diseño pretest post-test con un solo grupo, teniendo una muestra de 295 medidores de

la provincia de Grau, distrito de Chuquibambilla, este proyecto se desarrolla mediante la metodología ágil SCRUM.

En la primera etapa se identifican los requerimientos para el desarrollo de la aplicación móvil mediante un algoritmo de OCR denominado tesseract-ocr, en seguida se obtienen las lecturas mediante la aplicación móvil desde un celular o Tablet (4), la cual posteriormente será comparada con el padrón de lectura registrado manualmente, y mediante la comparación se obtiene la cantidad de lecturas erradas.

2. TRABAJOS RELACIONADOS

Barragán del Pozo Edgar Edison realizó un trabajo de investigación titulado "Diseño e Implementación del Sistema de Control Vehicular Utilizando Reconocimiento Óptico de Caracteres en el Laboratorio de Automatización Industrial de la E.I.S.". La inseguridad ha sido un problema que ha afectado a toda la sociedad, solo en la ciudad de Riobamba las estadísticas de los últimos 4 años indica que existe un promedio de 25 robos de vehículos anualmente. El objetivo del trabajo fue diseñar e implementar un sistema de control vehicular mediante el Reconocimiento Óptico de Caracteres (OCR), que permita obtener el número de placa de un vehículo con alto grado de rapidez y confiabilidad. Se tomó una muestra de 10 estudiantes de la E.I.S., para que observen las 10 placas vehiculares, obteniendo una confiabilidad del 90%, el tiempo empleado que se tarda el sistema de Control vehicular en reconocer el número de placa y emitir la señal de alarma está en un tiempo promedio de 8.3435 segundos.

La conclusión fue que el Sistema de Control Vehicular utilizando Reconocimiento Óptico de Caracteres en el Laboratorio de Automatización de la E.I.S., asegura la rapidez y confiabilidad. El prototipo implementado asegura el correcto reconocimiento de caracteres de forma eficiente y confiable cuando la placa del vehículo adquirida tiene un ángulo de 0 grados y así la imagen tomada es precisa para aplicar la técnica de OCR.

Laine Mikael, Nevalainen Olli S. (1) realizaron una investigación titulada "Un Sistema Autónomo de OCR para Teléfonos con Cámara Móvil". Universidad de Turku, Finlandia 2006. El propósito de este trabajo fue demostrar que las cámaras móviles actuales son capaces de ejecutar el software de OCR sin depender de hardware o instalaciones proporcionadas por la red dedicada.

El sistema se limita a reconocer las letras mayúsculas en inglés impreso en negro, sobre un fondo blanco. Las oportunidades y las dificultades relacionadas con traer OCR para funcionar en plataformas móviles que tienen la capacidad de captura de imágenes también se discuten en términos más generales. Sin embargo, los resultados fueron alentadores en que binarización, segmentación y OCR se podría hacer en un teléfono con cámara estándar con un solo procesador.

Loh Zhi Chang, Zhon ZhiYing Steven (6), realizaron un trabajo de investigación titulado "Las Técnicas de preprocesamiento robusto para aplicaciones de OCR en Dispositivos Móviles" en la Universidad Nacional de Singapur". En este trabajo, se proponen dos técnicas de preprocesamiento para la aplicación OCR móvil. La primera técnica de ayuda para localizar una señal en una imagen natural de manera eficiente, mientras que la segunda técnica implementa algoritmo de umbral de Otsu para convertir las imágenes en una imagen binaria. Estas técnicas fueron implementadas en una aplicación móvil de OCR que se desarrolla utilizando el escritorio de la biblioteca de fuentes abiertas. Después de la implementación, la aplicación de OCR se prueba con 50 imágenes de signos para verificar la exactitud. Los resultados experimentales han demostrado que estas técnicas pueden mejorar significativamente la precisión de OCR y disminuir el tiempo total de cálculo.

3. DESARROLLO DE LA APLICACIÓN MÓVIL

3.1 Módulos de la aplicación

Esta aplicación resuelve las necesidades del área comercial, optimizando el proceso de lectura de medidores y obteniendo datos confiables. Cuenta con un diseño agradable de fácil utilización y con una gama de herramientas que permite realizar las tareas rápidamente.

La aplicación móvil consta de cuatro módulos principales que son: Módulo de Inicio de Sesión, Módulo de Importación, Módulo de Lectura y Módulo Exportación.

Módulo de inicio de sesión: Para la autenticación de los usuarios que acceden a la aplicación móvil, desde el administrador hasta los usuarios.

Módulo de importación: Para realizar el proceso de lectura de los medidores eléctricos es

necesario contar con el archivo.xls que se exporta del sistema SIELSE y se comparte en el Dropbox

Una vez que el usuario se loguee, e ingrese a la aplicación, se activa este módulo donde indica el tipo de ejecutor, nombre del trabajador, la ruta asignada a lecturar y la opción de importar el marco de trabajo.

Módulo de lectura: Una vez importado el archivo.xls, el usuario podrá ver todo el marco de trabajo asignado, y podrá realizar la operación de captura de imagen del medidor eléctrico, para luego procesar dicha imagen y obtener la información requerida

Módulo de exportación: Al finalizar el proceso de lectura de medidores, el usuario podrá exportar la data y compartirlo en el Dropbox.

3.2 Arquitectura lógica de la aplicación

Lógicamente el sistema funciona con un servidor de base de datos en Electro Sur Este, al cual acceden los usuarios que trabajan en el área de informática, mientras que los usuarios de campo utilizan la aplicación móvil y guarda los datos localmente y luego se sincroniza mediante dropbox con la base de datos central. La figura muestra la Arquitectura lógica de la aplicación



Figura 1. Arquitectura Lógica de la aplicación

3.3 Programación en Android

Se utiliza como gestor de base de datos SQLite. SQLite no es una base de datos cliente-servidor, sino que enlaza con el propio programa, integrándose como una parte más del mismo.

El modelo vista controlador, es un patrón que divide la aplicación en tres niveles distintos, uno que presenta la interfaz gráfica (vista), otro que presenta el tratamiento de datos (modelo) y otro que se encarga de toda la lógica que se tiene que llevar a cabo por la aplicación (controlador).

En Android se tiene un MVC peculiar, las vistas se crean en XML, y cada una de estas vistas tiene asociado una Activity que la gestiona.

Por lo tanto, el esquema real del MVC en Android se muestra en la Figura 2.

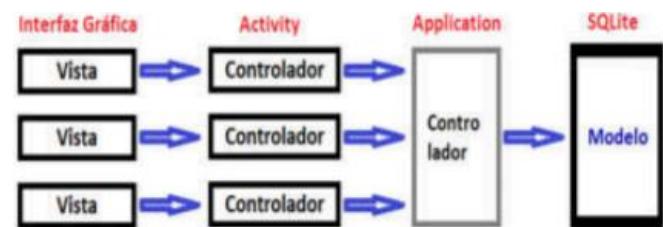


Figura 2. MVC en Android

3.4 Sprint Back Log

La lista de requisitos se muestra en la tabla 1

Tabla 1. lista de requisitos o Sprint BackLog del sistema

ID	Nombre de la historia	Importancia	Estimación	Sprint
1	Ingreso a la aplicación (inicio de sesión)	1	3	1
2	Importar padrón de lectura (marco de trabajo), a la aplicación móvil.	5	5	2
3	Administrar marco de trabajo (padrón de lectura digital)	4	3	2
4	Proceso de captura de la imagen del medidor eléctrico	6	8	3
5	Proceso OCR (Reconocimiento Óptico de Caracteres)	7	13	4
6	Generación del nuevo padrón de lectura	3	5	5
7	Exportar padrón de lectura	2	2	5

3.5 Interfaces de usuario

Los dispositivos móviles están cautivando cada vez más el interés de las personas, en el área educativa, salud y otras áreas (7), (8). El diseño de interfaces para dispositivos móviles es cada vez más complejo y tiene sus dificultades al momento de realizar proyectos (9).

La Figura 3 muestra un ejemplo de la interfaz de usuario para registrar una nueva lectura del medidor, Figura 3.



Figura 3. Interfaz para lectura de medidor

La aplicación móvil muestra el código del cliente y el código del medidor, la lectura anterior y el consumo promedio (cargado desde una base de datos) y se debe ingresar la lectura actual.

3.6 Código Fuente

A continuación, se muestra el código fuente de la aplicación móvil.

La Figura 4 muestra el código fuente para cargar el hardware de la cámara

```

Camera.AutoFocusCallback autoFocusCallback = new Camera.AutoFocusCallback() {
    @Override
    public void onAutoFocus(boolean success, Camera camera) {}
};

private MotorCamara(SurfaceHolder surfaceHolder) {
    this.surfaceHolder = surfaceHolder;
}

static public MotorCamara New(SurfaceHolder surfaceHolder) {
    Log.d(TAG, "Creación de un motor de cámara");
    return new MotorCamara(surfaceHolder);
}

public void requestFocus() {
    if (camera == null)
        return;
    if (isOn()) {
        camera.setAutoFocus(autoFocusCallback);
    }
}

public void start() {
    Log.d(TAG, "Ingresando MotorCamara - start()");
    this.camera = UtilCamara.getCamera();

    if (this.camera == null)
        return;
    Log.d(TAG, "Hardware de la cámara");

    try {
        this.camera.setPreviewDisplay(this.surfaceHolder);
        this.camera.setDisplayOrientation(90);
        this.camera.startPreview();
        on = true;
        Log.d(TAG, "Se inició la previsualización de MotorCamara");
    } catch (IOException e) {
        Log.e(TAG, "Error en setPreviewDisplay");
    }
}

public void stop() {
    if (camera != null) {
        camera.release();
        camera = null;
    }
    on = false;
    Log.d(TAG, "Deteniendo MotorCamara");
}

public void takeShot(Camera.ShutterCallback shutterCallback,
                    Camera.PictureCallback rawPictureCallback,
                    Camera.PictureCallback jpegPictureCallback) {
    if (isOn()) {
        camera.takePicture(shutterCallback, rawPictureCallback, jpegPictureCallback);
    }
}

```

Figura 4. Código Fuente para cargar el hardware de la cámara

En la Figura 5 se muestra el código fuente del OCR de la cámara.

```

public class OcrCamara extends FragmentActivity implements {

    static final String TAG = "DBG_" +
OcrCamara.class.getName();

    Button shutterButton;
    Button focusButton, btnok;
    Chronometer cronometro;
    VistCuadroEnfoque focusBox;
    SurfaceView cameraFrame;
    MotorCamara cameraEngine;
}

```

```

String value1, datoCliente;
TextView textocronometro;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.fm_ocr);
}

@Override
public void surfaceCreated(SurfaceHolder holder) {
    Log.d(TAG, "Entorno creado - cámara encendida");
    if (cameraEngine != null && !cameraEngine.isOn()) {
        cameraEngine.start();
    }

    if (cameraEngine != null && cameraEngine.isOn()) {
        Log.d(TAG, "Motor de la cámara se inicio");
        return;
    }
    cameraEngine = MotorCama-
ra.New(holder);
    cameraEngine.start();
    cronomet-
tro.setBase(SystemClock.elapsedRealtime());
    cronometro.start();
    Log.d(TAG, "Se inició el motor de la cámara");
}

@Override
public void surfaceChanged(SurfaceHolder holder, int format,
int width, int height) {}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {}

@Override
protected void onResume() {
    super.onResume();
    cameraFrame = (SurfaceView)
findViewById(R.id.camera_frame);
    shutterButton = (Button)
findViewById(R.id.shutter_button);
    focusBox = (VistCuadroEnfoque)
findViewById(R.id.focus_box);
    focusButton = (Button)
findViewById(R.id.focus_button);

btnok=(Button)findViewById(R.id.btnOk);
;

    cronomet-
tro=(Chronometer)findViewById(R.id.chr-
onometer2);
    textocronome-

```

```

tro=(TextView)findViewById(R.id.textVi-
ew2);
    shutterBut-
ton.setOnClickListener(this);
    focusBut-
ton.setOnClickListener(this);

btnok.setOnClickListener(this);
    Intent intent =
this.getIntent();
    Bundle extras = in-
tent.getExtras();

    if (extras != null) {//ver si contiene datos
        datoCliente = (String)
extras.get("codcliente");
        textocronome-
tro.setText(datoCliente);
    }

    SurfaceHolder surfaceHolder =
cameraFrame.getHolder();
    surfaceHol-
der.addCallback(this);
    surfaceHol-
der.setType(SurfaceHolder.SURFACE_TYPE_
_PUSH_BUFFERS);

    cameraFra-
me.setOnClickListener(this);
}

@Override
protected void onPause() {
    super.onPause();
    if (cameraEngine != null && cameraEngine.isOn()) {
        cameraEngi-
ne.stop();
    }

    SurfaceHolder surfaceHolder =
cameraFrame.getHolder();
    surfaceHol-
der.removeCallback(this);
}

@Override
public void onClick(View v) {
    if(v == shutterButton){
        if(cameraEngine != null && cameraEngine.isOn())){
            cameraEngi-
ne.takeShot(this,this,this);
        }
    }

    if(v == focusButton){
        if(cameraEngine!=null && cameraEngine.isOn())){
            cameraEngi-
ne.requestFocus();
        }
    }
}

```

```

        if(v == btnok){
            if(cameraEngine!=null &&
cameraEngine.isOn()){
                onBackPressed(); }
        }
        @Override
        public void onPictureTaken(byte[]
data, Camera camera) {
    Log.d(TAG, "Foto tomada\n");

    if (data == null) {
        Log.d(TAG, "Tiene datos
nulos");
        return;
    }
    Bitmap bmp = Herramien-
tas.getFocusedBitmap(this, camera,
data, focusBox.getBox());
    Log.d(TAG, "Tiene mapa de
bits");
    Log.d(TAG, "Inicialización de
TessBaseApi");
    new TessMotorAsin-
crono().executeOnExecutor(AsyncTask.SE
RIAL_EXECUTOR, this, bmp); }
}

```

Figura 5. Código Fuente del OCR

4. CONCLUSIONES

Se implementó la aplicación móvil basado en OCR para la reducción de los errores de lecturas de los medidores eléctricos de ELSE, en la fase de implementación se utilizó las herramientas: lenguaje de programación Java mediante la plataforma de Android Studio

Se redujo la cantidad de errores de lectura de medidores usando la aplicación móvil basada en OCR en un 100% que equivale a 9 lecturas erradas, validado mediante la prueba Z $z=3.0417>1.645$ por lo cual se rechaza la hipótesis nula (H_0), lo que significa que la aplicación móvil basada en OCR reduce la cantidad de errores de lecturas de los medidores de Electro Sur Este S.A.A.

Se redujo el tiempo por lectura realizada de cada medidor de un promedio de 13.14 segundos a 9.73 segundos, validado mediante la prueba de Z, $p=0.000<\alpha<0.05$, por lo tanto, se rechaza la hipótesis nula (H_0), lo que significa que la aplicación móvil basándose en OCR reduce el tiempo por lectura realizada.

Se detectó en un 1% las lecturas anómalas que se encuentra fuera del rango promedio de consumo

de un cliente, además 3% de lecturas refacturadas y 96% de lecturas normales.

Se redujo la cantidad de reclamos ingresados, validado mediante la prueba Z, $z_c=4.8>2.353$, por lo cual se rechaza la hipótesis nula (H_0), lo que significa que la aplicación móvil basada en OCR reduce el tiempo por lectura realizada.

Como trabajo futuro se pretende desarrollar una nueva versión para ALME, que haga uso de una conexión directa con el sistema SIELSE y actualice los datos en tiempo real.

AGRADECIMIENTOS

Agradecemos Electro Sur Este S.A., por permitir utilizar los datos de la empresa y a su vez por dar las facilidades del caso para la implementación de esta aplicación.

REFERENCIAS

- [1] M. Laine and O. S. Nevalainen, "A standalone OCR system for mobile cameraphones," in Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, 2006, pp. 1-5.
- [2] F. Cuevas, "Reconocimiento de caracteres con redes neuronales," Rev. Investig. y Tecnol., p. 30, 2012.
- [3] J. V. Pereira, Gráficos 3D en Interfaces Web, Tenerife: Universidad San Cristóbal de la Laguna, 2014.
- [4] R. Santiago, S. Trabaldo, M. Kamijo, and Á. Fernández, Mobile Learning: Nuevas realidades en el aula. Editorial Océano, 2015.
- [5] E. E. Barragán Del Pozo, "Diseño e Implementación de un Sistema de Control Vehicular Utilizando Reconocimiento Óptico de Caracteres en el Laboratorio de Automatización Industrial de la EIS.."2012.
- [6] L. Z. Chang and S. Z. ZhiYing, "Robust pre-processing techniques for OCR applications on mobile devices," in Proceedings of the 6th International Conference on Mobile Technology, Application & Systems, 2009, p.60.
- [7] C. C. Valero, M. R. Redondo, and A. S. Palacín, "Tendencias actuales en el uso de dispositivos móviles en educación," La Educ. Digit. Mag., vol. 147, pp. 1-21, 2012.
- [8] M. C. G. Mantilla, L. L. C. Ariza, and B. M. Delgado, "Metodología para el desarrollo de aplicaciones móviles," Tecnura Tecnol. y Cult. Afirmando el Conoc., vol. 18, no. 40, pp. 20-35, 2014.

[9] J. Cuello and J. Vittone, Diseñando apps para móviles. José Vittone-Javier Cuello, 2013.

BIOGRAFÍAS

Bixby Tapia, Se graduó en el año 2017 en Ingeniería Informática y Sistemas en la Universidad Nacional Micaela Bastidas de Apurímac y realizó sus prácticas pre profesionales en Electro Sur Este S.A

Manuel J. Ibarra, nacido en Cusco. Título obtenido como Ingeniero Informático y de Sistemas en la Universidad Nacional de San Antonio Abad del Cusco en el 2002. Magister en Ciencias, mención Computación, grado obtenido en la Universidad de Chile en el 201

