

Algoritmo para corregir la posición de matrices de datos (2D)

---

Número Publicado el 20 de junio de 2017

<http://dx.doi.org/10.23857/dom.cien.pocaip.2017.3.3.jun.812-832>  
[URL: http://dominiodelasciencias.com/ojs/index.php/es/index](http://dominiodelasciencias.com/ojs/index.php/es/index)

Ciencias Informáticas

Artículo Científico

## Algoritmo para corregir la posición de matrices de datos (2D)

*Algorithm to right the position of data matrix (2D)*

*Algoritmo para corrigir a posição da data matrix (2D)*

Darwin R. Caina-Aysabucha<sup>I</sup>  
Universidad Central del Ecuador  
Quito, Ecuador  
[drcaina@uce.edu.ec](mailto:drcaina@uce.edu.ec)

Jorge S. García-Espinoza<sup>III</sup>  
Universidad Central del Ecuador  
Quito, Ecuador  
[jsgarcia@uce.edu.ec](mailto:jsgarcia@uce.edu.ec)

Rene A. Carrillo-Flores<sup>II</sup>  
Universidad Central del Ecuador  
Quito, Ecuador  
[rcarrillo@uce.edu.ec](mailto:rcarrillo@uce.edu.ec)

Marcelo P. Carrillo-Flores<sup>IV</sup>  
Laboratorios Siegfried  
Quito, Ecuador  
[mpcarillo@siegfried.com.ec](mailto:mpcarillo@siegfried.com.ec)

**Recibido:** 30 de enero de 2017 \* **Corregido:** 20 de febrero de 2017 \* **Aceptado:** 20 junio de 2017

- <sup>I.</sup> Magister en Ciencias en Tecnologías de la Información; Ingeniero en Electrónica y Telecomunicaciones; Docente de la Facultad de Ingeniería, Ciencias Físicas y Matemática.
- <sup>II.</sup> Magister en Ingeniería Industrial; Ingeniero en Informática; Docente de la Facultad de Ingeniería, Ciencias Físicas y Matemática.
- <sup>III.</sup> Magister en Docencia Matemática; Licenciado en Ciencias de la Educación Profesor de Enseñanza media en la Especialización de Matemáticas y Física; Doctor en Ciencias de la Educación mención Gerencia Educativa; Docente de la Facultad de Ingeniería, Ciencias Físicas y Matemática.
- <sup>IV.</sup> Magister en Sistemas de Gestión Integrados; Diploma Superior en Auditoria de Gestión de la Calidad; Bioquímico Farmacéutico opción Bioquímica de Alimentos; Laboratorios Siegfried.

### **Resumen.**

Las Matrices de Datos o códigos 2D, son ampliamente empleados en los diferentes sectores de la industria, generalmente para identificación y etiquetado. En el presente trabajo se aborda el problema del código 2D, cuando este no se encuentra debidamente en una posición vertical con respecto al dispositivo lector, esto puede ocasionar una demora en la lectura o incluso, el no reconocimiento del código por parte del dispositivo lector; por este motivo se plantea un algoritmo para corregir la posición del Datamatrix, basado en técnicas del procesamiento digital de imágenes, con las cuales se determinan el ángulo de desviación con respecto a la vertical, para su posterior rotación y corrección, la determinación del valor del ángulo es independiente de la orientación que tiene el código, que puede ser hacia la izquierda o hacia la derecha. Las herramientas de software libre y gratuitas OpenCV y Python permiten el desarrollo del algoritmo de una forma rápida y económica. Finalmente, las pruebas del algoritmo con datos reales, muestra buenos resultados.

**Palabras Clave:** Matriz de datos; código 2D; procesamiento digital de imágenes; openCV; python.

**Abstract.**

Data Matrix or 2D codes are widely used in different industry sectors, generally for identification and labeling. In this paper, the problem of the 2D code is addressed, when it is not properly in a vertical position with respect to the reading device, this could cause a delay in the reading or even the non-recognition of the code by the reading device; for that reason an novel algorithm is proposed to right the position of the Datamatrix, based on digital image processing techniques, with which the angle of deviation with respect to the vertical could be calculated on an easy way, to right and correct the position later, the value of the angle is independent of the orientation of the 2D code, which can be left or right. Free and open source tools like OpenCV and Python allow the development of the algorithm in a quick and economical way. Finally, testing the algorithm with real data shows good outcomes.

**Keywords:** Datamatrix; code 2D; digital image processing; openCV; python.

### **Resumo.**

Data Matrix ou 2D códigos são amplamente utilizados em diferentes setores da indústria, geralmente para identificação e rotulagem. Neste artigo, o problema do código 2D é abordado, quando não está corretamente em uma posição vertical em relação ao dispositivo de leitura, isso pode causar um atraso na leitura ou mesmo o não reconhecimento do código pelo dispositivo de leitura ; Por esse motivo, um novo algoritmo é proposto para direcionar a posição da Datamatrix, com base em técnicas de processamento de imagens digitais, com as quais o ângulo de desvio em relação à vertical poderia ser calculado de maneira fácil, direto e correto posteriormente, O valor do ângulo é independente da orientação do código 2D, que pode ser à esquerda ou à direita. Ferramentas abertas e de código aberto como OpenCV e Python permitem o desenvolvimento do algoritmo de forma rápida e econômica. Finalmente, testar o algoritmo com dados reais mostra bons resultados.

**Palavras chave:** Matriz de dados; Código 2D; Processamento de imagem digital; OpenCV; Python.

## **Introducción.**

La Matriz de Datos (Datamatrix en inglés) es un sistema de codificación bidimensional (2D), que está regulado por el estándar ECC200 (Keyence, 2017), este tipo de código es empleado ampliamente en el sector industrial para etiquetar e identificar los diferentes productos, debido a sus ventajas técnicas en relación a los códigos de barras u otros, también se ha visto un uso creciente en el campo de los dispositivos móviles, debido a que estos cuentan con una cámara incorporada para lectura de este tipo de códigos (Roos, et al., 2010). La representación en la mayoría de los casos es en forma de cuadrado, aunque también tiene la forma de rectángulo (GS1 2011; Barcode Coder, 2017). Al ser un elemento que generalmente es ubicado sobre la superficie de un objeto, esta propenso a factores externos, como los factores climáticos, así como también de una mala manipulación durante su traslado de un lugar a otro; debido a lo mencionado un código 2D esta propenso a la pérdida de cierta información, sin embargo el Datamatrix emplea una corrección de errores basada en códigos de Reed-Solomon, el mismo que permite decodificar la información hasta con un daño aproximado del 30% del código completo (Shrivastava 2013; Datamark 2017). Para decodificar la información se necesita de un lector de dos dimensiones, que puede ser, un lector dedicado exclusivamente para esta tarea, o una cámara dentro de un dispositivo móvil con la respectiva aplicación para decodificar este tipo de códigos.

Un problema adicional a la pérdida de información, es la posición de la lectura del código 2D con respecto al dispositivo lector, ya que muchas veces el lector con el código no puede estar totalmente alineado para una lectura vertical, es por esto que es necesario que el lector del código cuente con un algoritmo para corrección de la posición, antes de tomar los datos para ser decodificados.

## Algoritmo para corregir la posición de matrices de datos (2D)

---

Los problemas con códigos 2D pueden presentarse de varias formas (Microscan 2017; GS1 2011), y para solucionar los mismos una de las herramientas a utilizar es el Procesamiento Digital de Imágenes (PDI), tanto para generar los códigos (Ghunawat 2016), como para el tratamiento en el lado de la decodificación, en donde se han aplicado algunas técnicas de PDI, algunas de las cuales se puede verificar en algoritmos para el reconocimiento de códigos 2D usando detección de bordes y operaciones morfológicas (Gaur & Tiwari, 2014), algoritmo para localizar, segmentar y decodificar los códigos 2D más usados (Ouaviani, et. al., 1999), también algoritmos más sofisticados para reconocimiento de códigos 2D usando características de textura combinando con redes neuronales (Gaur & Tiwari, 2014).

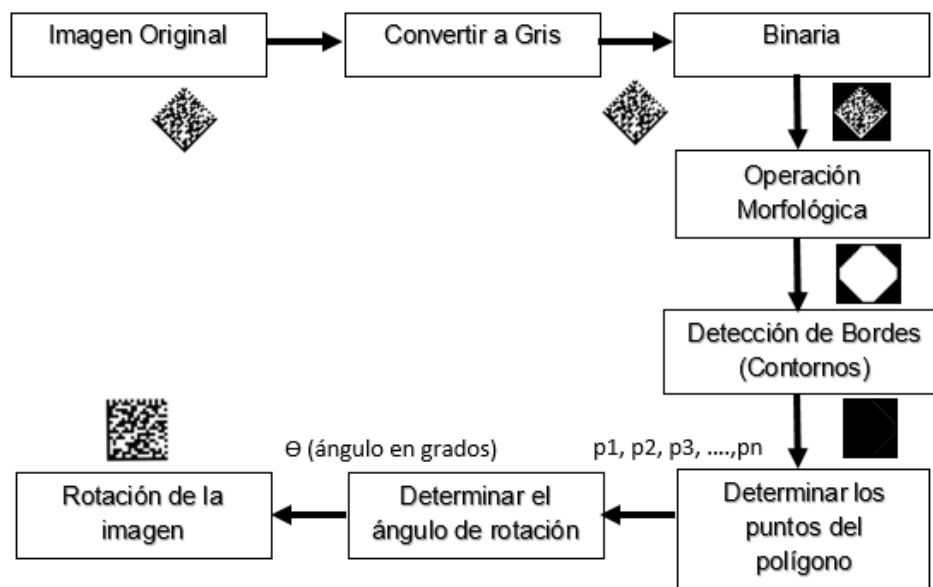
En base a los aspectos anteriormente mencionados, el objetivo del presente trabajo es implementar un algoritmo para corregir la posición de Datamatrix, utilizando técnicas de procesamiento digital de imágenes. El algoritmo propuesto calcula automáticamente el ángulo que deberá girar la imagen para quedar en posición totalmente vertical, y de esa manera facilite la lectura y decodificación de la información, independientemente si la orientación del código 2D está a la izquierda o a la derecha.

### **Metodología.**

En el presente trabajo se ha propuesto un algoritmo basado en el procesamiento digital de imágenes, el cual se muestra en la Figura 1, en donde se puede observar el tratamiento que tiene la imagen, desde la entrada pasando por cada uno de los bloques del algoritmo propuesto, hasta la obtención de imagen final, en donde se encuentra el Datamatrix rotado y corregido a una posición vertical, en donde el dispositivo lector podrá decodificar la información del código 2D sin mayor inconveniente.

## Herramientas.

Para el desarrollo del algoritmo se utilizó el lenguaje de programación Python y las librerías OpenCV (Howse, 2013), en vista que ambas herramientas son gratuitas y de libre distribución, lo cual, permite lograr un sistema confiable con muy pocos recursos económicos. Para la lectura de los códigos 2D corregidos, se utilizó la aplicación “QR Code Reader” instalada en un teléfono Samsung modelo SM-J111M, la que está disponible en una versión gratuita en la tienda de Google Play, la cual permite decodificar y obtener la información del Datamatrix, de una forma rápida y confiable, una vez que su posición del código 2D ha sido solucionada.



*Figura N° 1. Algoritmo para rotar una imagen a la posición vertical.*

## OpenCV

OpenCV (*Open Source Computer Vision*) desarrolladas por Intel mediante código C/C++, estas librerías proporcionan varios algoritmos para el procesamiento de imágenes y de visión por

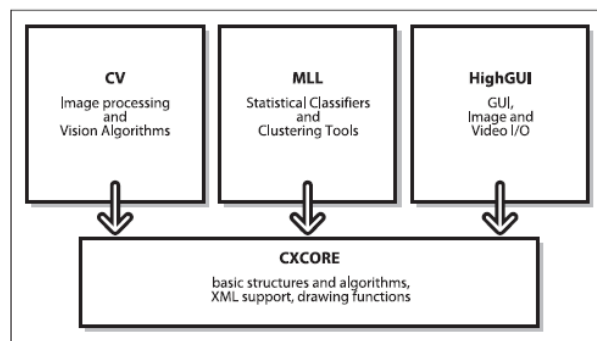
## Algoritmo para corregir la posición de matrices de datos (2D)

---

computadora. La fortaleza de OpenCV está en la rapidez con la que funcionan sus algoritmos y en la gran cantidad de algoritmos disponibles, que día a día se van incrementando y compartiendo como resultado de un mejoramiento y desarrollo continuo de esta comunidad. La Figura 2, nos muestra la estructura de OpenCV. La versión de OpenCV 3.0, es la empleada para el desarrollo del presente trabajo.

### *Python*

Es un lenguaje de programación interpretado o de script, usa tipado dinámico, es multiplataforma y orientado a objetos. Fue creado por Guido van Rossum a finales de los años ochenta y tiene una licencia de código abierto denominada *Python Software Foundation Licence*. La versión de python utilizada en el presente trabajo es la 2.7.11.



**Figura N° 2.- Estructura básica de OpenCV (Bradski & Kaehler, 2008).**

### *Datos*

Para seleccionar los datos a utilizar, es importante conocer la estructura de un Datamatrix (Fuwa, 2017), el mismo que está compuesto de un borde más la información en sí mismo, esto se puede observar en la Figura 3.



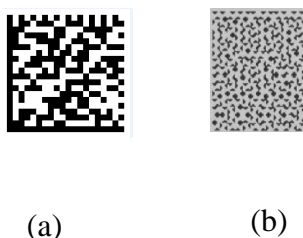
Algoritmo para corregir la posición de matrices de datos (2D)

---



**Figura N° 3.- Estructura de un Datamatrix (Sivart, 2017).**

En este sentido los datos a utilizar en este trabajo son de dos tipos, por un lado, un Datamatrix completo, y por otro lado datos que contienen solo la parte correspondiente a la información, estos tipos de datos se pueden apreciar en la Figura 4.

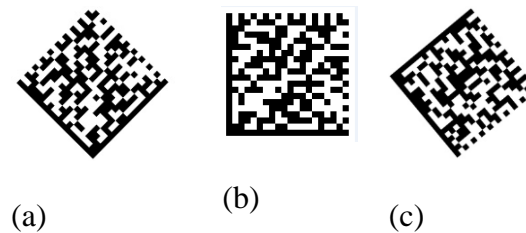


**Figura N° 4.- (a) Datamatrix (b) Información de un Datamatrix.**

La posición ideal de lectura de estos códigos 2D, es cuando están en posición vertical con respecto al dispositivo de lectura como se indica en la Figura 5(b), sin embargo, algunos códigos por diferentes circunstancias no pueden estar en esa posición vertical, y adoptan una posición de inclinación con respecto al dispositivo lector, bien sea al lado izquierdo o al lado derecho como se indica en la Figura 5(a) y 5(c) respectivamente; estos dos casos especiales son los que se toma como referencia para resolver con el algoritmo planteado en la Figura 1.

Algoritmo para corregir la posición de matrices de datos (2D)

---



**Figura N° 5.- Posiciones de un Datamatrix. (a) Inclinado a la izquierda (b) Vertical (c) Inclinado a la derecha.**

### **Procedimiento.**

En este punto es importante destacar los principales fundamentos y tareas realizadas en cada uno de los bloques propuestos en el algoritmo de corrección de la posición de un Datamatrix.

### **Toma de Datos.**

El primer paso dentro del algoritmo es cargar una imagen de códigos 2D, previamente almacenados y convertidos a imágenes en formato de JPEG<sup>1</sup>: La imagen en general es una imagen de color, la cual está formada por los componentes RGB, correspondientes al color rojo, verde y azul. Las dimensiones de las imágenes son de 400x300 píxeles para imágenes de Datamatrix, y de 1600x1200 píxeles para imágenes parciales del código 2D, de acuerdo lo indicado en la Figura 4.

### **Conversión a Grises.**

El siguiente bloque tiene por objetivo convertir la imagen que tiene 3 canales a una imagen con 1 solo canal, lo que permite eliminar información irrelevante de la imagen y reducir el tamaño

---

<sup>1</sup> JPEG, es un estándar de compresión y codificación de archivos e imágenes.

Algoritmo para corregir la posición de matrices de datos (2D)

---

de la imagen, para el siguiente paso del procesamiento. Una representación matemática para este procedimiento se muestra en la ecuación 1, el cual es denominado como método ponderado.

$$I = ((0,3)R + (0,59)G + (0,11)B) \quad [1]$$

### **Binarización.**

En la etapa de binarización, el propósito es convertir una imagen de grises en un imagen binaria, es decir, una imagen con valores comprendidos de 0 a 255, a una imagen con dos tipos de valores 0 o 1, esto permitirá identificar claramente los pixeles del Datamatrix con el valor 1 (color blanco), y el resto de los elementos con valor 0 (color negro). Para este propósito se ha utilizado la técnica basada en un Umbral T de acuerdo a la ecuación 2, en donde experimentalmente se determinó que T tiene un valor de 100.

$$B(x, y) = \begin{cases} 1 & \text{si } I(x, y) > T \\ 0 & \text{si } I(x, y) \leq T \end{cases} \quad [2]$$

### **Operaciones Morfológicas.**

Debido a la binarización tenemos regiones que no están definidas completamente o no muestran una región sólida (con huecos), para mejorar este aspecto se utiliza las operaciones morfológicas, las cuales ayudan a cambiar la forma y estructura de las regiones de pixeles, una de estas operaciones es la dilatación, la cual rellena los “huecos” y permite obtener regiones bien definidas y sólidas. Matemáticamente esta operación se representa en la ecuación 3 (Jaehne, 2002).

$$B \oplus E = \{p: E_p \cap B \neq \emptyset\} \quad [3]$$

Algoritmo para corregir la posición de matrices de datos (2D)

---

Donde se expresa la convolución de una imagen binaria B con un elemento estructurante E, permiten obtener la nueva imagen mejorada, para que sea la entrada del bloque de detección de bordes.

### **Detección de Bordes.**

Luego de la etapa de binarización, se procede a identificar los bordes en la imagen, lo que significa que identificamos píxeles alrededor de los cuales la imagen presenta una brusca variación, y que con los pasos previos se garantiza que los píxeles son parte de la escena y no corresponden a elementos de ruido, en sí mismo, el borde no es más que una cadena de puntos de borde. Uno de los algoritmos para este fin es el algoritmo de Canny (Canny, 1986), sin embargo, estos bordes no son expresados como contornos. Para recuperar en términos de contornos, a partir de la imagen binaria, OpenCV cuenta con una función `cv2.findContours( )`, la cual está basada en el algoritmo propuesto en el trabajo denominado “*Topological Structural Analysis of Digitized Binary Images by Border Following*” (Suzuki & Abe, 1985), los contornos son de mucha utilidad para el análisis de la forma de un objeto.

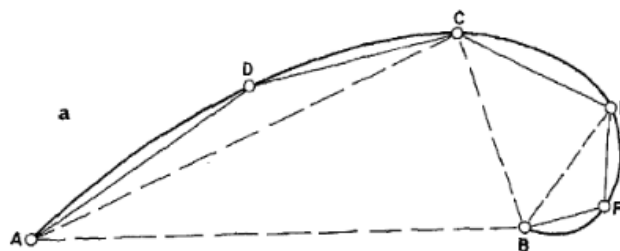
### **Determinar los Puntos del Polígono.**

Los bordes determinados en términos de contornos, son curvas que contienen muchos puntos, pero la idea es reducir ese número de puntos para lograr aproximar a un polígono más simple, que permita determinar sus vértices, los cuales permitirán al mismo tiempo conocer de una forma aproximada los vértices del código 2D; con los vértices encontrados ya se puede determinar la orientación del polígono, y paralelamente nos permite conocer la orientación del Datamatrix, lo cual ayuda a determinar el ángulo que deber rotar el Datamatrix para ubicarse en posición vertical.

Algoritmo para corregir la posición de matrices de datos (2D)

---

Para hacer esta aproximación de curva a polígono, uno de los algoritmos más empleados es el de Douglas-Peucker (Ramer, 1972), este algoritmo utiliza el criterio de la distancia de un punto a la recta, conjuntamente con un umbral de error determinado, con esto permite eliminar los puntos que están dentro del margen de error, y conserva los puntos que están fuera del valor del error; la aproximación de una curva a un polígono se puede apreciar de mejor manera en la figura 6.



**Figura N° 6.- Aproximación de una curva a un polígono (Ramer, 1972).**

En OpenCV se cuenta con la función `cv2.approxPolyDP( )`, que está basada en el algoritmo de Douglas-Peucker, y la cual permite determinar los vértices del polígono en la imagen de contornos.

### **Determinar el Ángulo de Rotación.**

Determinado los puntos que conforman el polígono, se procede a conocer la orientación del polígono respecto a la vertical; esto se puede indagar aplicando el concepto de la pendiente de una recta, y a través de la pendiente se puede conocer el ángulo que tiene el polígono con respecto a la vertical. La ecuación 4 muestra la expresión matemática para calcular el ángulo de un segmento, basado en el concepto de la pendiente.

$$\alpha = \tan^{-1} \left( \frac{y_2 - y_1}{x_2 - x_1} \right) \quad [4]$$

El ángulo de inclinación del polígono, permite también conocer cuál es el ángulo de inclinación del Datamatrix, ya que el polígono se obtuvo a partir de la imagen de entrada.

### **Rotar la Imagen.**

Para rotar la imagen se acude a la transformada de rotación, la cual es una transformación geométrica, cuyo rol es modificar la relación espacial entre pixeles, reubicando los pixeles en el plano de la imagen. Expresando matemáticamente esta transformada, y asumiendo como punto de origen el (0,0) en el plano de la imagen, las ecuaciones 5 y 6 muestran esta operación.

$$I' = R(\theta)I \quad [5]$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad [6]$$

Finalmente, basado en el fundamento matemático de las ecuaciones 5 y 6, se procede a rotar la imagen con el ángulo calculado, con el fin de ubicarla el código 2D en forma vertical, teniendo en cuenta que si el giro es de izquierda a derecha el ángulo es negativo, y de derecha a izquierda el ángulo es positivo, el sentido de giro es determinado por el algoritmo de forma automática; una vez colocado el código 2D en posición vertical, se facilita la decodificación de la información por medio de dispositivo lector, al tener en la posición adecuada de lectura al Datamatrix.

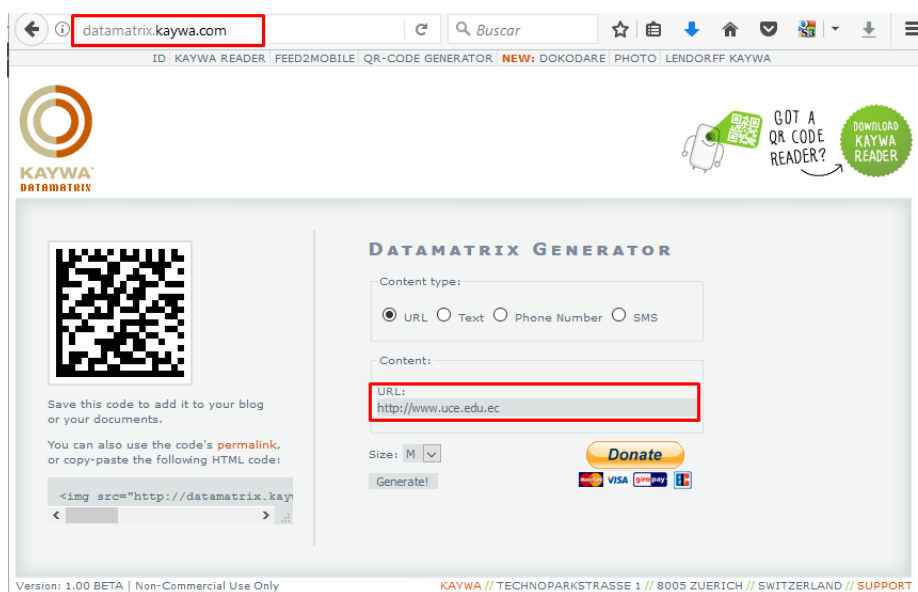
### **Resultados y Discusión.**

Para proceder a validar el algoritmo propuesto, se genera un código Datamatrix mediante la herramienta disponible en la dirección Web <http://datamatrix.kaywa.com>, en donde se genera un

## Algoritmo para corregir la posición de matrices de datos (2D)

código con la URL<sup>2</sup> de la Universidad Central del Ecuador (UCE) y un segundo código con el URL de la Universidad Técnica Equinoccial (UTE), en la figura 7 se muestra este procedimiento.

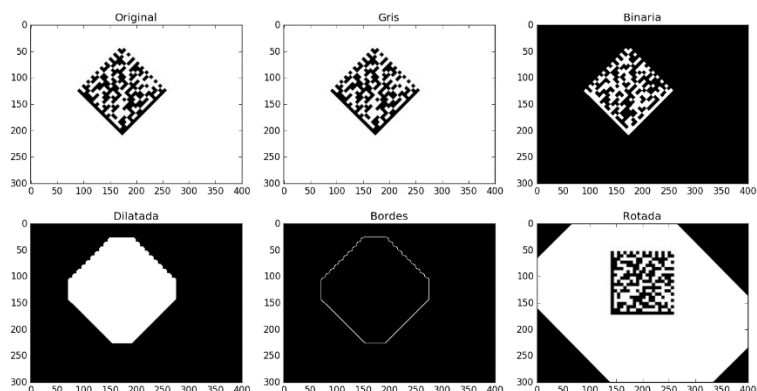
Luego de obtener los respectivos códigos, se altera la posición del código 2D, en el caso del código de la UCE se orienta hacia la izquierda, y en el caso del código de la UTE se lo hace hacia la derecha. Con la orientación alterada se procede a introducir en el algoritmo, con el objetivo de validar el mismo, es así que se obtienen los resultados que se pueden observar en la figura 8.



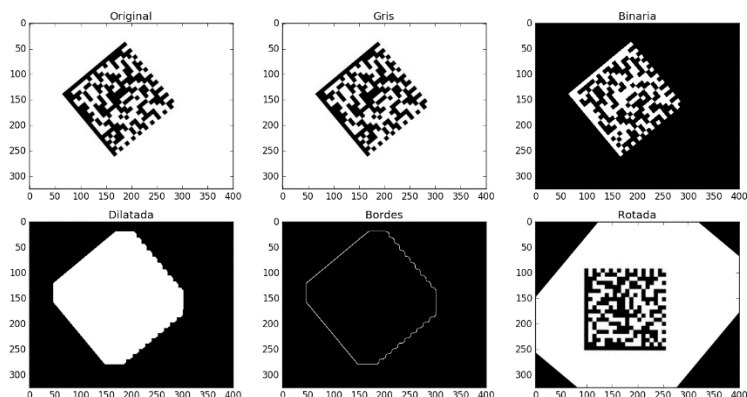
**Figura N° 7.- Imagen de captura de la página Web donde se genera un código Datamatrix.**

<sup>2</sup> URL (Uniform resource locator) en español Localizador de Recursos Uniforme.

Algoritmo para corregir la posición de matrices de datos (2D)



(a)



(b)

**Figura N° 8.- Resultado obtenido para un Datamatrix que contiene una dirección URL (a) Código para la UCE con orientación a la izquierda (b) Código para la UTE con orientación a la derecha.**

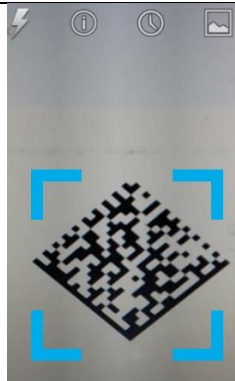

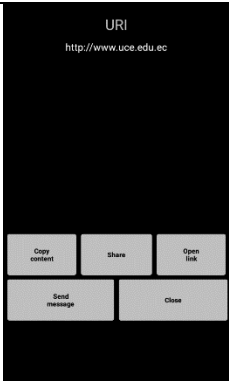



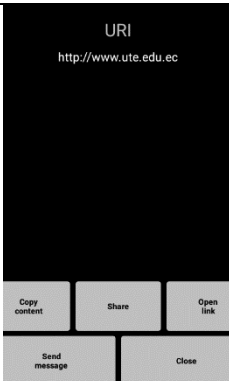

Para la realización de la decodificación de la información contenida en los códigos Datamatrix, se utilizó la aplicación “QR Code Reader” para Android, disponible en una versión gratuita en la tienda de Google Play. En la tabla 1, se muestra la posición original del código 2D, en la cual muchas veces hay dificultades para ser decodificada, debido a que el dispositivo lector no la reconoce por la mala ubicación del código, o a veces toma mucho tiempo en ser leído el código



Algoritmo para corregir la posición de matrices de datos (2D)

debido también a una posición incorrecta; en la misma tabla se indica los resultados obtenidos luego de aplicar la corrección de la posición del Datamatrix mediante el algoritmo desarrollado, en donde la decodificación se realiza sin mayor contratiempo.

**Tabla N° 1.- Pruebas realizadas con el lector de Datamatrix, para los códigos 2D de la UCE y UTE, antes y después de la ejecución del algoritmo que corrige la posición.**





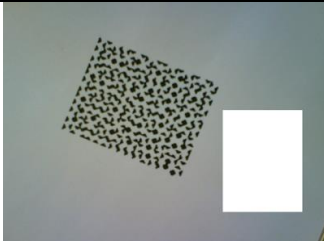
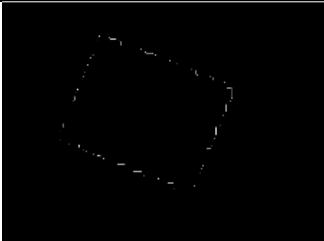
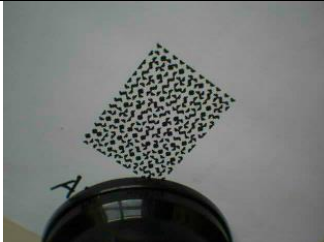

Orientación	Imagen Original		Imagen Rotada	Dirección Web	Página Web
Izquierda		Algoritmo			
Derecha		Algoritmo			

Finalmente se realiza las pruebas con un segundo tipo de datos, en donde se indica principalmente si es posible determinar un polígono adecuado, el mismo que es el elemento principal para determinar la orientación de la imagen y posterior corrección de la posición. En la tabla 2 se muestran diferentes escenarios, en la que se podría encontrarse un Datamatrix, y en base a

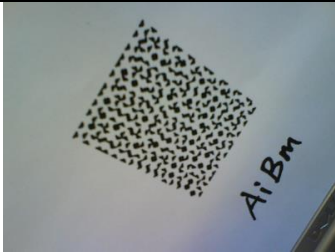

Algoritmo para corregir la posición de matrices de datos (2D)

ello se determina si es posible o no determinar un polígono, el cual sirve de base para la rotación del código 2D.

**Tabla N° 2.- Diferentes escenarios en donde se determina el polígono.**

Imagen Original	Rota	Polígono	Observación
	Si		La presencia de un elemento pequeño en relación al tamaño de la información, y que está ubicado en la parte inferior, no afecta a determinar el polígono.
	Si		La presencia de un elemento pequeño en relación al tamaño de la información, y que está ubicado en la parte inferior, no afecta a determinar el polígono.
	Si		La presencia de un objeto blanco, no influye en la definición de un polígono.
	No		La presencia del objeto negro, no permite definir un polígono.

Algoritmo para corregir la posición de matrices de datos (2D)

	Si		El ángulo de lectura no afecta mayormente a la determinación de un polígono, sin embargo, el resultado no es totalmente vertical.
---	----	--	---

### Conclusiones y Recomendaciones.

Los resultados experimentales obtenidos muestran la viabilidad de la propuesta, la cual permite corregir la posición vertical del Datamatrix antes de la decodificación, lo cual, junto al margen de error que soporta este tipo de código 2D, hace de este tipo de códigos una herramienta más fiable y más sólida para las aplicaciones en los diferentes segmentos de la industria. Sin embargo, como se indicó en la tabla 2, aún quedan aspectos a mejorar, en algunos casos excepcionales.

El uso de las diferentes técnicas de procesamiento digital de imágenes ha permitido lograr el objetivo de la propuesta, primero al tener una imagen de entrada, la cual contiene diferente tipo de información, las técnicas permiten eliminar cierta información de irrelevancia para el proceso, y permite quedarse solo con la información necesaria, de la cual se extrae los datos para rotar la imagen. Los algoritmos propuestos por Suzuki & Abe, así como por Douglas-Peucker, han sido los elementos centrales para resolver el problema, ya que el primero permite obtener los contornos, y el segundo permite aproximar ese contorno a un polígono. Sin embargo, en trabajos futuros se podría experimentar con otros algoritmos para determinar los bordes, como el de Canny, Sobel o Prewitt, y

## Algoritmo para corregir la posición de matrices de datos (2D)

---

con algoritmos basados en la transformada de Hough para reconocer líneas, círculos o elipses; de esta manera el procesamiento digital de imágenes ofrece alternativas para resolver este tipo de aplicaciones de la vida real.

La utilización de estas herramientas de software libre y gratuitas, permiten que la aplicación sea multiplataforma y de bajo costo para un proyecto de esta índole, y que son de mucha ayuda para los diferentes sectores de la sociedad, entre ellos el productivo. La utilización de OpenCV permite avanzar más rápido en el desarrollo de los algoritmos, ya que muchos algoritmos parciales ya los tiene implementado dentro de sus librerías, lo cual permite ahorrar tiempo; y la utilización de Python permite una programación más rápida y transparente, debido a sus características de tipado dinámico y orientado a objetos.

### **Bibliografía.**

- BARCODE CODER. (25 de abril de 2017). DATA MATRIX BARCODE. Obtenido de <http://barcode-coder.com/en/datamatrix-specification-104.html>
- Bradski, G., & Kaehler, A. (2008). Learning OpenCV. Sebastopol, California (USA): O'Reilly Media, Inc.
- Datamark. (25 de abril de 2017). Identificación con Código Datamatrix 2D. Obtenido de <http://www.datamark.es/tecnologia/identificacion-codigos-datamatrix-2d/>
- Canny, J. (1986). A computational approach to edge detection. Transactions on pattern analysis and machine intelligence, 679-698.
- Fuwa Informationssysteme. (25 de abril de 2017). Grundlagen DataMatrix Code. Obtenido de <http://www.fuwa-it.de/lexika/>
- Gaur, P., & Tiwari, S. (2014, April). Recognition of 2D Barcode Images Using Edge Detection and Morphological Operation. International Journal of Computer Science and Mobile Computing, 3(4), 1277-1282.
- Gaur, P., & Tiwari, S. (2014, May). 2D QR Barcode Recognition Using Texture Features and Neural Network. International Journal of Research in Advent Technology, Vol.2, No.5. E-ISSN: 2321-9637.
- Ghunawat, M. (2016, July). Data Encoding and Decoding Using Data Matrix. Open Access International Journal of Science & Engineering, Volume 1, Issue 1.

Algoritmo para corregir la posición de matrices de datos (2D)

---

- GS1. (2011). GS1 DataMatrix: Una introducción y revisión técnica de la simbología más avanzada compatible con los identificadores de Aplicación GS1. Obtenido de <https://www.gs1.org.ar/documentos/DATAMATRIX.pdf>
- Howe, J. (2013). OpenCV Computer Vision with Python. Birmingham: Packt Publishing Ltd.
- Jaehne, B. (2012). Digital Image Processing (5<sup>th</sup>. Ed.). Heidelberg, Germany: Springer-Verlag.
- KEYENCE. (25 de abril de 2017). What is a Datamatrix code?. Obtenido de [http://www.keyence.com/ss/products/auto\\_id/barcode\\_lecture/basic\\_2d/datamatrix/](http://www.keyence.com/ss/products/auto_id/barcode_lecture/basic_2d/datamatrix/)
- MICROSCAN. (2017). Understanding 2D Verification. Obtenido de [http://files.microscan.com/whitepapers/wp\\_2Dverification.pdf](http://files.microscan.com/whitepapers/wp_2Dverification.pdf)
- Ouaviani, B., Bottazzi, M., Brunelli, E., Caselli, F., & Guerrero, M. (1999). A Common Image Processing Framework for 2D Barcode Reading. Seventh International Conference IEEE, Image Processing and Its Applications. Obtenido de [http://www2.informatik.uni-halle.de/agprbio/AG/Lehre/ABV\\_SS07/material/Ottaviani99.pdf](http://www2.informatik.uni-halle.de/agprbio/AG/Lehre/ABV_SS07/material/Ottaviani99.pdf)
- Ramer, U. (1972). An Iterative Procedure for the Polygonal Approximation of Plane Curves. Computer Graphics and Image Processing, 1, 244-256.
- Roos, A., Flegl, M., Wieland, S., Schwarzbacher, A., & Einsiedler, H. (2010). Semacode based Voucher Concept enabling flexible and user-friendly Hotspot Login on demand. Dublin Institute of Technology, School of Electronic and Communications Engineering. Obtenido de [http://www.electronics.dit.ie/staff/aschwarzbacher/research/2010\\_Mobilfunktagung\\_roos.pdf](http://www.electronics.dit.ie/staff/aschwarzbacher/research/2010_Mobilfunktagung_roos.pdf)
- Shrivastava, P., & Pratap, U. (2013, August). Error Detection and Correction Using Reed Solomon Codes. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 3. ISSN: 2277 128 X.
- Sivart. (25 de abril de 2017). EL Data Matrix. Obtenido de [http://sivartsl.com/descargas/data\\_matrix.pdf](http://sivartsl.com/descargas/data_matrix.pdf)
- Suzuki, S., & Abe, K. (1985). Topological Structural Analysis of Digitized Binary Images by Border Following. Computer Vision, Graphics, and Image Processing, 30(1):32-46.