

# Sistema de CAPTCHA por Medio de Cámara Web en HTML5 Canvas.

JAVIER SALVADOR VÉLEZ MONSIVÁIS

**Resumen—** El presente artículo describe el diseño, implementación, prueba y análisis de resultados de un sistema de CAPTCHA basado en información generada por una cámara web, implementado en el elemento Canvas de la tecnología Web HTML5. Por medio de una serie de algoritmos de procesamiento digital de imágenes y otros de validación, la aplicación pretende determinar si se trata de un ser humano quien se encuentra detrás de la computadora, con el fin de sustituir los sistemas de CAPTCHA utilizados actualmente, los cuales presentan varios inconvenientes mencionados más adelante.

## I. INTRODUCCIÓN

Como consecuencia de las poderosas capacidades de las computadoras y software actuales, así como de sus constantes mejoras en el campo de la Inteligencia Artificial, mejoras en los sistemas de autenticación de personas son requeridas continuamente. Uno de los sistemas de autenticación más populares, principalmente por su uso extendido en sitios y aplicaciones web es el sistema CAPTCHA.

Un sistema de CAPTCHA o Prueba de Turing Completamente Automática y Pública para Diferenciar Computadoras de Humanos (*Completely Automated Public Turing Test to tell Computers and Humans Apart*) es un sistema de autenticación de seres humanos que se usa en una extensa variedad de sitios y aplicaciones web para distinguir entre una persona y software automatizado que pretende hacerse pasar por un ser humano (*Bot*).

Si bien el sistema CAPTCHA representa una prueba relativamente sencilla, presenta varias desventajas o inconvenientes de los cuales se pueden listar los siguientes:

1. Requiere interpretar una imagen con caracteres estilizados, lo cual se puede ver afectado por las deficiencias en la visión de una persona.
2. La correcta deducción de los caracteres mostrados esta sujeta a la propia interpretación del usuario de caracteres similares como es el caso del “1” y “l”, “0” y “O”, “5” y “S” etc.
3. Existen casos en donde ciertos caracteres mostrados son completamente ilegibles (Figura 1).
4. Las personas con deficiencias graves en la vista requieren otros sistemas de autenticación como los del tipo auditivos,

sin embargo éstos no siempre se encuentran disponibles junto con un sistema de CAPTCHA convencional.

5. En algunos casos una computadora es capaz de superar las pruebas de autenticación CAPTCHA haciendo uso de algoritmos de Reconocimiento Óptico de Caracteres (OCR).

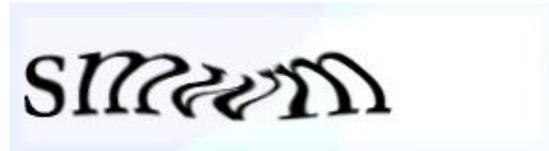


Figura 1. Caracteres de difícil interpretación.

Dentro de la literatura se pueden encontrar otros trabajos que proponen sistemas de CAPTCHA los cuales no están basados en la interpretación de texto. Un ejemplo de éstos es “*What’s Up CAPTCHA?*” (Gosswiler, Kamvar, Baluja) [1] el cual propone una autenticación basada en determinar la correcta orientación de una imagen. Otro ejemplo donde se propone un sistema basado en información generada por una cámara web se encuentra en el trabajo “*Advanced CAPTCHA technique using Hand Gesture based on SIFT*” (Srinivas, Kalyan, Koduganti) [2] en el cual se propone un modelo basado en la validación de movimientos hechos con la mano para representar números, por parte de el usuario ante una cámara web o teléfono celular.

En este trabajo se propone un modelo de un sistema de autenticación que supere los inconvenientes mencionados anteriormente, utilizando como información de prueba una imagen generada por la cámara web del usuario. El propósito de desarrollar un sistema de autenticación tiene como principal objetivo autenticar usuarios en sitios, aplicaciones y formas alojadas en la *World Wide Web*.

En el modelo propuesto, el sistema toma como información de entrada una imagen del usuario para posteriormente analizarla y determinar si se trata de un ser humano quien se encuentra frente a la computadora. El sistema propuesto puede ser utilizado en una amplia gama de aplicaciones, de manera directa, en todas aquellas en donde un CAPTCHA convencional es presentado al usuario, y se toma como una ventaja el hecho de que actualmente la gran mayoría de los dispositivos desde donde se accede a dichas aplicaciones cuenta con una cámara web.

## II. MARCO REFERENCIAL

Una prueba de CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) representa

un sistema de autenticación de seres humanos, generalmente compuesto por una serie de caracteres estilizados, donde la tarea de la persona consiste en interpretar e introducir los caracteres mostrados [3]. Este sistema es generalmente utilizado en sitios y aplicaciones web, como medio de filtración de software automático que pretende hacerse pasar por un usuario.

La plataforma utilizada para implementar este trabajo es la herramienta web **HTML5 Canvas**. El elemento Canvas es un componente de la tecnología web HTML5 que permite la renderización de formas en 2D de manera dinámica [5], y es compatible con las últimas versiones de los navegadores web Google Chrome, Mozilla Firefox y Opera.

Llamamos **procesamiento digital de imágenes** al conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar su calidad o facilitar la búsqueda de información [6]. Dentro del mismo contexto, el **color RGB** representa un modelo de composición del color conformado por diferentes intensidades de una componente roja (R), verde (V) y azul (B) [7].

Cabe mencionar que a lo largo de este trabajo se usa el término *pixel* para referirse a la estructura más básica por la cual está compuesta una imagen digital. Dicha estructura se compone a su vez por un elemento *Red, Green, Blue* y *Alpha*, los cuales están representados por un tipo de dato Entero y adquieren valores en el rango de 0 a 255.

### III. MODELO Y METODOLOGÍA

De forma general, el funcionamiento de la aplicación se compone de cuatro procesos principales (Figura 2): La generación de la imagen digital desde una cámara web, el procesamiento digital de la imagen, el algoritmo de validación y la emisión del resultado.

Debido a que se trata de una aplicación web, el desarrollo de la misma fue implementado utilizando los siguientes lenguajes: HTML5, JavaScript, CSS.

Así mismo todos los experimentos y resultados presentados en este trabajo fueron realizados en un sistema con las siguientes configuraciones: Navegador Google Chrome Versión 39.0.2171.71 (64-bit), Mac OSX 10.9.4.

#### A. Generación de la imagen digital desde una cámara web

Este proceso se comprende desde la captura de una imagen del usuario por medio de una cámara web hasta su representación digital dentro de la aplicación. La aplicación no requiere una cámara web específica, pues el sistema operativo funge como interfaz entre ésta y la obtención de datos digitales.

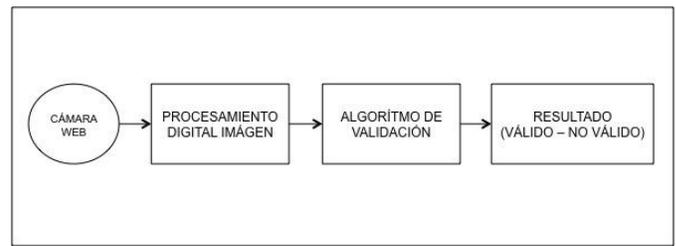


Figura 2. Diagrama de bloques del funcionamiento de la aplicación.

Para esta aplicación, se hace uso de la capacidad de los navegadores web modernos para obtener datos multimedia desde el hardware de la computadora del usuario, específicamente, haciendo uso del objeto **navigator** del DOM, invocando su método **getUserMedia**. Para este trabajo se generaron *frames* (imágenes) con una resolución de 375 x 500px cada uno.

De forma paralela, dentro del contenido del sitio web que aloja la aplicación, se cuenta un elemento de tipo `<canvas>` el cual tiene como función desplegar una secuencia de video obtenida desde la cámara web del usuario (Figura 3).



Figura 3. Secuencia de video en el elemento Canvas.

Posteriormente, haciendo uso del método **context.getImageData** del elemento Canvas, se obtiene un mapa de los píxeles de la muestra (*frame*) (Figura 4) contenido en un objeto de tipo **ImageData** el cual representa un cuadro de la secuencia de video en un momento dado (definido por la aplicación en la fase de Experimentación).

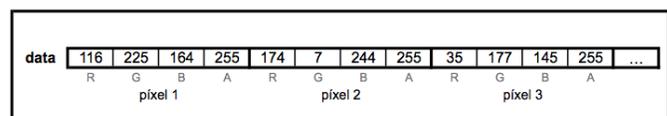


Figura 4. Mapa de los píxeles de la muestra.

#### B. Procesamiento digital de la imagen

En esta esta fase de la aplicación se realiza un análisis de los píxeles obtenidos durante la fase anterior, por medio de una de

las funciones claves del algoritmo, la **detección de piel**, la cual analiza cada píxel de la muestra generada.

Así mismo, el algoritmo incluye una **ventana de exclusión** la cual tiene como propósito delimitar el área en donde se analizarán los píxeles de la muestra. Esta ventana es mostrada al usuario por medio de líneas guía (Figura 5).

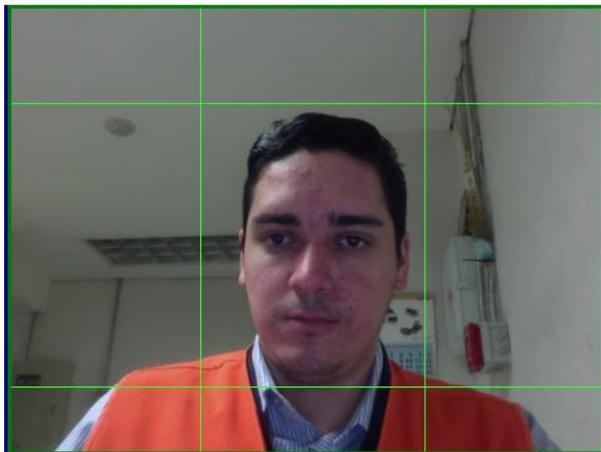


Figura 5. Ventana de Exclusión.

El proceso de detección de piel en la muestra funciona haciendo uso de una subrutina denominada **validarPixel** la cual funciona utilizando un *perfil de color*.

El perfil de color representa el rango de valores válidos para cada componente (Rojo, Verde, Azul) en un píxel. Este perfil de color deberá ser predefinido dentro de la aplicación. Cabe mencionar que para una detección de piel más flexible, varios perfiles de color deberán ser predefinidos en la aplicación, esto para igualar una variedad de tonos de piel.

Durante la fase de experimentación, únicamente un solo perfil de color (Perfil de color A)(Figura 6) ha sido definido. Para obtener este perfil, se hizo uso del software Adobe Photoshop y su herramienta *Color Range*.

Perfil de Color A		
	Inferior	Superior
R	40	168
G	30	98
B	18	88

Figura 6. Perfil de color A

### C. Algoritmo de validación

El algoritmo de validación representa la fase de la aplicación en donde se determina si una muestra es válida o no. Este algoritmo, realiza un conteo de los píxeles validados por la función **validarPixel** dentro del área delimitada por la ventana de exclusión. Se ha definido un parámetro de **tolerancia** el cual determina el criterio para que una muestra sea considerada como válida o no. Después, la función

**validarMuestra** (Figura 7) evalúa si la muestra proporcionada es válida o no en base a la cuenta total de píxeles válidos y el parámetro de tolerancia. En esta experimentación el parámetro de tolerancia definido es de **18000 píxeles** válidos por muestra.

```

función validarMuestra(contexto, datosImagen){
    si (datosImagen.cuenta > _TOLERANCIA){
        despliega("A C E P T A D O");
        regresa VERDADERO;
    }sino{
        regresa FALSO;
    }
}
    
```

Figura 7. Pseudocódigo de la función validarMuestra.

### D. Emisión del resultado

Este proceso representa la última fase de la aplicación y consiste únicamente en informar a componentes subsecuentes de la misma si la muestra analizada representa una prueba de autenticación válida.

Para este caso, el resultado de la muestra analizada representa una variable de tipo booleana con posibles valores de **true** cuando una muestra es válida y **false** en caso contrario. Así mismo, se despliega el mensaje de "[ A C E P T A D O ]" (Figura 8) en el caso de muestras que han sido aprobadas.



Figura 7. Muestra aceptada.

## IV. EXPERIMENTACIÓN

Para la fase de experimentación, se analizaron cuatro grupos conformados por 100 muestras cada uno, esto con el fin de analizar un número considerable de muestras, y al mismo tiempo evitar alargar demasiado el proceso de obtención de muestras. Se utilizaron diferentes configuraciones de iluminación y uso de la ventana de exclusión para cada grupo de muestras (Tabla 1).

Tabla 1. Configuración de las muestras analizadas

Grupo	Configuración
1	Iluminación media, sin Ventana de Exclusión
2	Iluminación media, con Ventana de Exclusión
3	Iluminación alta, sin Ventana de Exclusión
4	Iluminación alta, con Ventana de Exclusión

## V. RESULTADOS

En esta sección se presentan los resultados obtenidos de los distintos grupos de muestras descritos en la sección de Experimentación (Tabla 1).

Tabla 2. Resultados bajo distintas configuraciones.

Grupo	Ventana de Exclusión	Iluminación	Tolerancia	Eficiencia
1	No	Media	18 000	<b>52/100</b>
2	Si	Media	18 000	<b>91/100</b>
3	No	Alta	18 000	<b>36/100</b>
4	Si	Alta	18 000	<b>66/100</b>

Si se observa la columna de 'Eficiencia' para cada uno de los cuatro grupos de muestras en la Tabla 1, se puede apreciar que la ventana de exclusión representa un atributo decisivo para alcanzar un valor superior a sesenta por ciento (60%) en la evaluación de eficiencia de reconocimiento por grupo.

Otra información extraíble de la Tabla 2, es que una iluminación alta no necesariamente representa una mejoría en la evaluación final de las muestras, lo cual se puede observar si se comparan los grupos 2 y 4.

Para esta fase de experimentación, se puede identificar en la Tabla 2 que la configuración más adecuada para obtener una eficiencia de reconocimiento elevada, son aquellas muestras que fueron realizadas utilizando la ventana de exclusión, y una iluminación media.

## VI. CONCLUSIONES

En este trabajo se pudo constatar como el procesamiento digital de imágenes, permite extraer información sumamente útil de las muestras, ya que una imagen de tamaño promedio contiene una cantidad de información bastante grande. De esta información se pueden extraer muchos rasgos y patrones que nos permiten implementar algoritmos basados en el procesamiento de este tipo de muestras.

Observando la Tabla 2 de la sección de Resultados se puede concluir que se obtiene un mejor análisis de las muestras utilizando la ventana de exclusión. Del mismo modo se puede observar que en este caso el uso de una iluminación alta no representó mejoras en la eficiencia de la aplicación.

La obtención de resultados fidedignos, esta en función de muchos factores internos y externos, como son: el tono de piel del usuario, la iluminación sobre el usuario en el momento de la muestra, el nivel de ruido producido por elementos externos al usuario al momento de generar la imagen, entre otros.

Si bien la implementación del modelo mostrado en este trabajo representa un enfoque simple, la etapa de experimentación y los resultados obtenidos sugieren que la aplicación descrita resulta viable para su uso en futuros sistemas de autenticación de usuarios. Esto no sin antes, desarrollar algoritmos de detección y validación más robustos.

## REFERENCIAS

- [1] Gossweiler R, Kamvar M, Baluja S. "What's Up CAPTCHA? A CAPTCHA Based On Image Orientation". World Wide Web Conference 2009, pages 841-850.
- [2] Srinivas, Kalyan, Koduganti, "Advanced CAPTCHA technique using Hand Gesture based on SIFT", International Journal of Computer Applications, Volume 31– No.11, October 2011
- [3] Walsh, Eric. "CAPTCHA he cracked by artificial intelligence". mybroadband.co.za. Reuters. Retrieved 27 November 201
- [4] Lin R, Huang S, Bell G, Lee Y, A new CAPTCHA interface design for mobile devices, AUIC '11 Proceedings of the Twelfth Australasian User Interface Conference - Volume 117, pages 3-8.
- [5] Fulton, Fulton, "HTML5 Canvas, 2nd Edition", Chapter 2 Drawing on the Canvas, O'REILLY, 2013.
- [6] Gonzalez Rafael, "Digital Image Processing", 2.4.2 Representing Digital Images, pp 54, 2002.
- [7] Charles A. Poynton. "Digital Video and HDTV: Algorithms and Interfaces", 2003.
- [8] Koprowski, Wilczyński, Wróbel, BBońska-Fajfrowska, "Calibration and segmentation of skin areas in hyperspectral imaging for the needs of dermatology", BioMedical Engineering OnLine 2014, 13:113 (August 2014).