Journal of Theoretical and Applied Information Technology
15th August 2021. Vol.99. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PUBLIC AND AUTOMATED FUNCTIONAL TESTING (CAPTCHA) WITH ANIMATIONS BASED ON FACIAL EXPRESSIONS

[1] RODOLFO ROMERO-HERRERA, [2] MANUEL OSWALDO LÓPEZ MARÍN, [3]JOSÉ FÉLIX SERRANO TALAMANTES

[1]IPN.ESCOM, Department of Postsgraduate, Ciudad de México

[2]IPN-CIDTEC, Department of Postgraduate, Ciudad de México

[3]IPN-CIDETEC, Department of Postsgraduate, Ciudad de México

E-mail: [2] rromeroh@ipn.mx

ABSTRACT

The growing need to avoid the generation of false information on the internet forces service providers to have options, the reverse test being completely automatic to distinguish computers from humans (CATPCHA Completely Automated Public Turing test to tell Computers and Humans Apart), one of the most varied and preferred methods to achieve this goal. In this research, a version of CAPTCHA is implemented that meets all design requirements, such as the simplicity in solving the challenge and the complexity necessary to avoid being solved by an automatic program. This article contributes by adding a procedure with which the CATPCHA is regenerated every 2 minutes, changing the location of the images randomly, including the keyboard where the answer is entered. Another contribution is the use of icons with facial expressions using animated gifs that have the cognitive advantage of identifying human beings concerning machines. In addition, the use of image segmentation techniques is avoided using techniques and distortions, which did not affect the use and identification of CATPCHA. To verify the effectiveness of the proposal, several tests were implemented that verified the effectiveness of the method.

Keywords: *Captcha, test de Turing, distortional, facial expression, Bots.*

1. INTRODUCTION

Attackers on the Internet use different methods and means to steal information; for example, bots are used to compromise and control computer systems and thus increase attack capacity [1]. Botnets are computer networks that are compromised and controlled by a computer attacker [2], used to carry out different types of attacks, such as DDoS [3], Spam, Keylogging, and Spreading malware. To slow down and in the best of cases disable the theft of information, captchas have been developed as a barrier that prevents this criminal activity. The challenge-response test, known as the Public Reverse Turing Test, is described and implemented as an automated variant to differentiate machines from humans (CAPTCHA) [4].

This work aims to develop a safe and simple option for the validation of a user when interacting with different systems, by creating a Turing test (captcha) with the advantage of contributing with the use of animated images based on facial expressions. In the related works section, it can be read how CAPTCHAS have evolved presenting alternative directions, without improving the cognitive process of recognition and use of facial expressions as an alternative to the Turin test. In addition, to strengthen the captcha, it regenerates every 2 minutes, changing the location of the images randomly; the same happens with the keyboard through which the answer is entered. It is worth highlighting as one of the main contributions the creation of a captcha based on animated images that use facial expressions in the form of an icon since in general computer programs have difficulties identifying human facial expressions. The human being, unlike machines, can quickly identify if a person is sad, happy, or angry; But for computers, it is a problem that even with the advancement of pattern recognition, it is very difficult for them to identify such affective states.

This is how this quality is used in favor of human beings in the development of the developed CATPCHA.
This approach is not presented in any of the references consulted. Mainly because science hardly considers affective states as part of human intelligence. Hence the novelty of the proposal.

For the development of the project and the research, a research protocol based on objectives or goals to be achieved was used, so the general design of the system is first proposed, as well as the resources required for its design, subsequently the technique is established. divide and win to obtain and develop the parts that make up the system. finally, several tests are carried out to verify the functionality and usability of the proposed captcha.

Generation and deployment time is improved compared to text, audio, and video-based ones.

2. RELATED WORKS

There are the following related works: Kurt Alfred Kluever (2008) proposed a video captcha where the user provides three tags that identify the presented video [5]. Gossweiler, Kamvar, Baluja (2009) showed a text-type captcha that is based on the vertical orientation of an image [6]. Goswami, Singh, Vatsa, Powell, Noore (2012) proposed a captcha based on the recognition of faces of the same subject, the user distinguishes the faces between non-human images, backgrounds, distortions, and faces [7]. Goswami, Singh, Vatsa, Powell, Noore (2012) implemented a captcha where the user selects human faces from an image with distortions [8]. Kumarasubramanian, Ostrovsky, Pandey, Wadia (2013) explore how to use captchas cryptographically by proposing the creation of a protocol based on the resolution process [9]. Goswami, Singh, Vatsa, Powell, Noore (2014) proposed a captcha in which a couple of human faces must be found in an image in a set of distorted backgrounds and images [10] Fujita, Ikeya, Kani, Nishigaki (2015) created a captcha in which two three-dimensional images are mixed to form a third, the challenge is to identify an individual image [11]. Hernandez-Castro, R-Moreno, Barrero, Gibson (2017) created a captcha to be solved using machine learning to identify attack vectors [12]. Gao, Mohamed, Saxena, Zhang (2017) designed a video-type catpcha and implemented an automated attack framework to defeat this design using image processing [13] Osadchy, Hernandez-Castro, Gibson, Dunkelman, Pérez-Cabo (2017) Using deep learning techniques, they identified the main problems of captchas [14]. Lin, Lin, Lv, Cai, Cao (2018) solved text-type captchas based on the Chinese language [15]. In 2018, researchers from the Georgia Institute of Technology created a captcha specific for mobile devices based on face recognition [16]. Ogiela, Krzyworzeka, Ogiela (2018) proposed a proposal for cognitive captchas, in which their resolution is based on having special knowledge and perceptual skills [17]. Thawatwong Lawan (2018) with three different types of captcha differentiated which are the ideal usability patterns for creating a captcha, resulting in captchas with image patterns being the simplest to solve by users [18].
In the cited references it is observed how the captchas have evolved dealing with different problems, but none of them considers the advantage obtained if facial expressions are considered as icons in the usability for the recognition of CAPTCHAS, hence the importance of meeting the project objective.

3. METHODOLOGY

There are different types of CAPTCHA on the internet, which meet the objective of strengthening the security of the site where they are deployed. In this way, the captcha design must consider the ease of resolution and the security provided to the application that implements it.
Based on the conclusion of Ekman [19]; the project considered 6 emotions: Joy. Sadness, Fear, Anger, Disgust, and Surprise; to generate facial expressions in images as icons. There are different types of CAPTCHA on the internet, which meet the objective of strengthening the security of the site where they are deployed. In this way, the captcha design has to consider the ease of resolution and the security provided to the application that implements it.

3.1 TURING TEST

The Turing test proposes a method to indicate the existence of mentality in computers, and it consists in that a computer induces interrogators to believe that it is a person [4]. The inverse Turing test is a reorientation of the Turing test in which it is stated that human beings are the object of study and whether they are indistinguishable from machines [20].

3.2 CAPTCHA AND DISTORTIONS

The first captchas were designed to prevent a bot from performing automated tasks, such as:

Submitting forms with personal information, Internet payments, creating email accounts, and Sending spam. In these applications, the user was required to correctly enter alphanumeric characters contained in a distorted image [1]. Methods such as Blur, Warp, Change the shape of an image and add horizontal or vertical lines of different thicknesses were used to make it difficult for the attackers. Noise is another method used to hinder the recognition of images [3], it is introduced during its acquisition, transmission, or processing process. There are different types, among the most used are Gaussian, Multiplicative, Salt, and pepper.

3.3 TYPES OF CAPTCHA

1) Captcha based on text.

It is an image made up of different distorted characters. It is decrypted by optical character classification and is vulnerable because it separates the image from the background [3]. Language limits the variety of combinations [5].

2) Captcha based on images.

This type of captcha avoids language dependency, is accessible and easy to use. There is a high failure rate when interpreting the image [11], and it uses the cognitive abilities of the human presenting riddles, puzzles, pairs, sequences, etc [10].

3) Facial recognition in captchas.

The use of faces in the captcha provides several benefits: The user is familiar with a face, which gives accessibility, which takes advantage of the cognitive factor [6] [16].

4) Captcha based on video and audio.

Its little use is due to the need to require higher bandwidth for downloading and operation compared to text or image [16]. It has limited accessibility for users with vulnerabilities [14].

4 DESIGN

4.1 ARCHITECTURE

Web architecture is a complex system of interfaces that offer service and business processes implemented with specific design techniques, accessed through web browsers. The user sends requests to the server, where the application is hosted, which makes use of databases where the information is stored. The information is processed and updated to the server to present to the user. The pattern used in this project is the Model-View-Controller (MVC) [20]. Figure 1 shows the architecture and distribution of software involved in the captcha generation program.

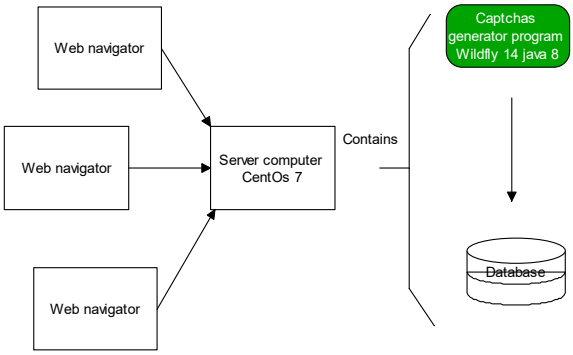

Figure 1. Architecture for the captcha generator

The Captcha program was developed in JAVA, by implementing Java Server Faces (JSF), which allows connection to the Java Persistence API (JPA) database, its view is developed in HTML5 and JQuery. MariaDB is used to manage the database.

4.2 CAPTCHA

The captcha creation program is separated into modules:

Common components. This module is made up of elements of business logic, for the creation of objects and classes of information storage that is transferred through the different modules. See figure 2.

Graphic interface. Developed with html5 and jquery technology, this module oversees allowing user interaction with the system, shows the generated captcha, and enables navigation in the application to capture data. See figure 2.

Controller component. It distributes the loads in the system, organizes the captcha generation requests, establishes transactions that start with the capture of user data with data persistence, sends the captcha to the graphical interface for its display; Also, it regenerates the view and the captcha now in which the validity of this is fulfilled. See figure 2.

Captcha generator core. It is the center of business logic. Where the selection of the facial expression to be identified and the additional ones that will serve to hide the correct face is made. See figure 2. Once the images are joined, the nucleus applies different distortions and geometric transformations to make identification difficult. By having the images distorted, the program creates the sequence of images in a gif-type image and is sent to the graphical interface for presentation.

Journal of Theoretical and Applied Information Technology
15th August 2021. Vol.99. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

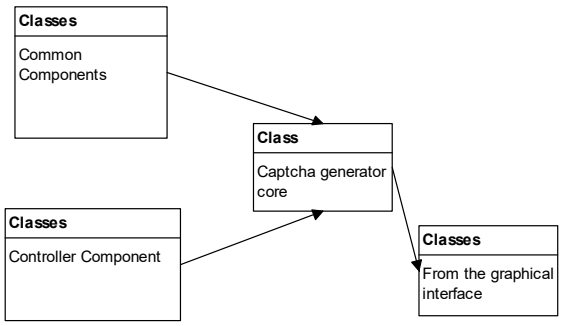

Figure 2. Clase para componentes comunes.

4.3 DATA FLOW

The user logs in to the captcha generator system; for the software test you are asked to capture your data, and it is stored in the dedicated database, then the captcha is presented; every attempt the user makes is logged. See figure 3.

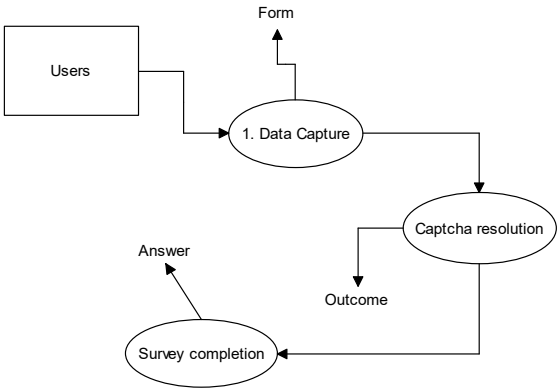

Figure 3. - Diagrama de flujo de datos

4.4 CAPCHA GENERATION

The generation of the captcha begins with the selection of the facial expression and is carried out randomly on a preset set. Then, a randomly selected set of face images is created, and a subset is taken, with which an image is created that concentrates 9 individual images arranged in rows and columns. A series of geometric distortions and transformations are applied to the image composed of sub-images to make it difficult to identify patterns. See Figure 4.

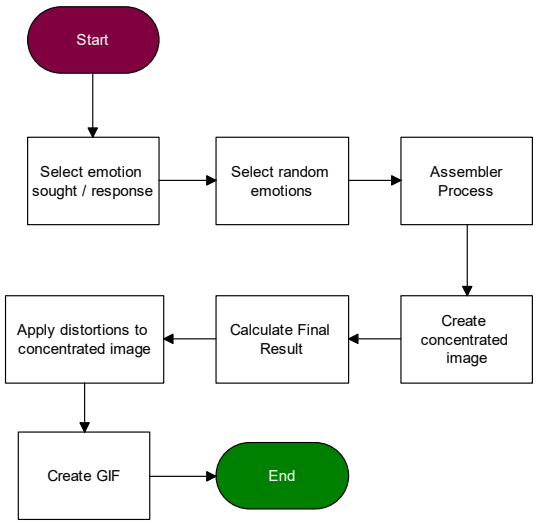

Figure 4. Captcha generation flow diagram.

4.5 CAPTCHA RESOLUTION

Once the captcha has been generated, the user data is requested and stored, and it is presented on the screen. The user has 2 minutes to enter the correct answer in a text field, using a numeric keyboard, where the digits are randomly arranged. At the end of the time, a new captcha is generated and the order of the keyboard with which the answer is entered is changed. See Figure 5. If the captcha is not resolved correctly, a new one will be generated; the cycle is concluded by entering the correct answer.

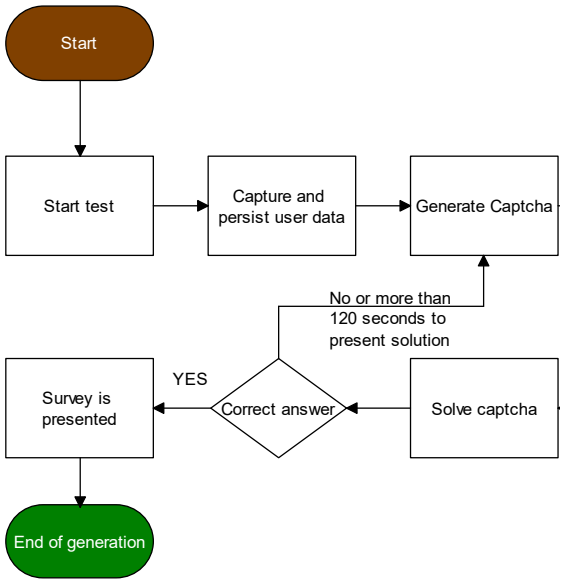

Figure 5. Captcha resolution flow chart

5. CAPTCHA RESOLUTION

1. The user is presented with the home screen. In this part, all the variables are initialized, for their later use in the different modules. see figure 6.

Figure 6. Home screen and information capture

2.1 With data capture, the captcha "creation process" proceeds to:

a. A random number from 1 to the maximum number of facial expressions set. This number will indicate which expression is being searched for.
b. The number of grouped images that will make up the GIF.
c. The number of images that will make up a grouped image.
d. Height and width of the gif.
e. The cadence of the gif, that is, the time between the change of images.

2.2 Generate a sequence of random numbers that will be distributed in the image, to indicate what their numbering is.

2.3 Generate random individual images, following the process in step 2.1.

2.4 Take 9 individual images to create a grouped image, and a series of distortions are applied to it, such as: resizing, blurring dilation, erosion, contrast change, and geometric transformation.

2.5 Assign the 6 images an identifying number from 1 to 9, obtained in point 2.2, this number is attached to the image as a watermark. See figure 7.

2.6 With the set of images, the GIF is created, according to the parameters of time, size, and members. See figure 7.

3 A new sequence of random numbers from 1 to 9 is generated, with which the keyboard is created, with which the answer is entered; It is noted that this sequence is different from that of point 2.2. See figure 8.

4 Present the user with the gif image and the keyboard where the answer is captured, establishing the 2-minute countdown for the resolution of the challenge.

5 Finish if the challenge was answered.

Figure 7. Captcha generated.

Figure 8. Panel for the introduction of answer

6 TEST AND RESULTS

As a result, a captcha was obtained that generates icons with easy expressions. The developed system was tested in a dedicated and exclusive environment, with the Linux Centos 7 operating system. Access to it was made with wireless network connections and a dedicated router.

6.1 FACE RECOGNITION TESTS

To strengthen the implemented captcha, different distortions were applied to the images, including a watermark. Once the image was processed, it was subjected to face identification by a face detection program, OpenCV and MatLab with the Viola & Jones method [21].

The test consisted first in verifying that the program can detect faces, as can be seen in Figure 9 and Figure 10.

Journal of Theoretical and Applied Information Technology
15th August 2021. Vol.99. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 9 Image before face identification

Figure 10. Image after Face Identification Processing

The Viola & Jones method could not detect faces in the created image [21] (See figure 11 & 12).

Figure 11. The image that makes up the captcha after being processed with the face identification program.

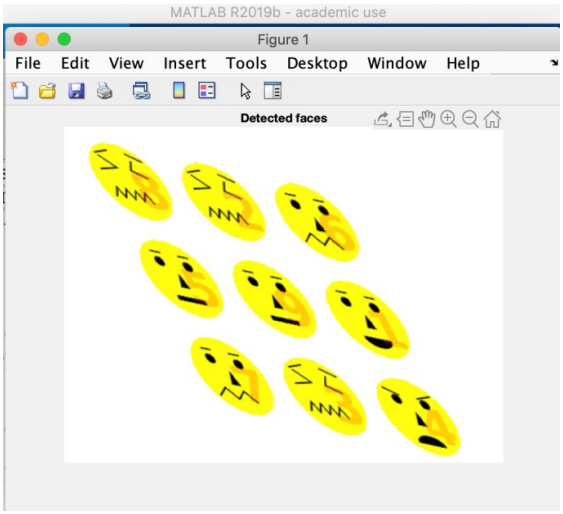

Figure 12. Face identification using Viola & Jones.

6.2 CODE TESTING

The applied code tests were implemented to detect coding defects. To carry out this process, the FindBugs tool was used, which is a plug-in for the Eclipse development environment.

As a result of the tests, a code without defects was obtained.

6.3 TEST OF PERFORMANCE

Performance tests are performed on computers, networks, software, or other devices; are used to determine the speed and efficiency of these. This procedure can involve both quantitative tests (measure response times or quantity in millions of lines of code) and qualitative tests (reliability, scalability, and interoperability are evaluated). These performance tests can be carried out through automated tools, which allow determining the stability of the system [22], the tool used to carry out this test is JMeter [23]. The measurements within the test are:

- Average: the average or arithmetic mean of the time in milliseconds.
- Median: of the time in milliseconds.
- 90% line: maximum time used by 90% of the sample; to the rest of the
- itself took longer.
- Min: minimum time of the sample of a certain URL.
- Max: maximum time of the sample of a certain URL.

- % Error: percentage of requirements with errors.
- Performance: measured in request/second.
- KB/sec: speed measurement in Kbytes per second.
- The tests were performed to find the correct number of users after several incremental tests. It started with a small number of users, gradually increasing it until it found a limit case of users:

For 10 recurring users:

In table 1 the total mean was 967 ms, which means that the response time of the system was 0.9 seconds, a low time for 10 recurring users who made 430 requests.

As can be seen, the average time to access a page is 967 seconds, making a total of 430 requests to the server.

The total time used for the 10 threads can be calculated with equation (1):

Total Time = #Samples * Average = 430 * 967 = 415810 milliseconds (1)

The total average time required by each thread can be calculated with equation (2):

((Total Time / 1000) / 60) / Thread quantity = ((415810/1000) / 60) / 10

= 0.69301667 minutes (2)

In the same table 1, there were no errors during the requests, therefore all were accepted and attended correctly.

For 100 returning users:

In table 1 the total mean was 9627 ms, which means that the response time of the system was 9.63 seconds, a low time for 100 recurring users who made 4300 requests.

As can be seen, the average time to access a page is 9627 seconds, making a total of 4300 requests to the server. The total time used for the 10 threads can be calculated with equation (3):

Total time = #Samples ∗ Average = 4300 ∗ 9627 = 41396100 milliseconds. (3)

The total average time required by each thread can be calculated with equation (4):

((Total Time / 1000) / 60) / Thread quantity = ((41396100/1000) / 60) / 100

= 6.89935 minutes (4)

In the same table, no request had an error, therefore all were accepted and attended to correctly.

For 1000 recurring users:

In table 1 the total mean was 3366 ms, which means that the response time of the system was 3.36 seconds, a very low time for 1000 recurring users, who made 41020 requests.

As can be seen, the average time to access a page is 3366 seconds, making a total of 41020 requests to the server.

The total time used for the 100 threads can be calculated by (5):

Total Time = #Samples * Average = 41020 * 3366 = 138073320 milliseconds (5)

The total average time required by each thread can be calculated with (6):

((Total Time / 1000) / 60) / Thread quantity = ((138073320/1000) / 60) / 1000 = 2.301222 minutes (6)

In table 1 no request had an error, therefore all were accepted and attended to correctly.

Summarizing, for the Average one must:
• 10 users 967 ms
• 100 users 9627 ms
• 1000 users 3366 ms

It is observed that comparing the results of 10 users for 1000 users, the difference is 2.4 seconds, which is a minimum time increase compared to the number of users being served.
It is important to note that in the measured times the access time to the database and the image processing are already included, in this way the system can be classified as fast.

Table 1 Performance tests for total

Users	Samples	Mean	Median	90% line	Min	Max	%Error	Performance (seg)	Kb/seg
10	430	967	242	1845	0	25730	0	12.2	1658.05
100	4300	9627	2339	19224	0	202575	0	18.3	2503.54
1000	41020	3366	53	5327	0	704944	0	4.1	362.91

Graphics to show Data Mean Median Deviation Performance

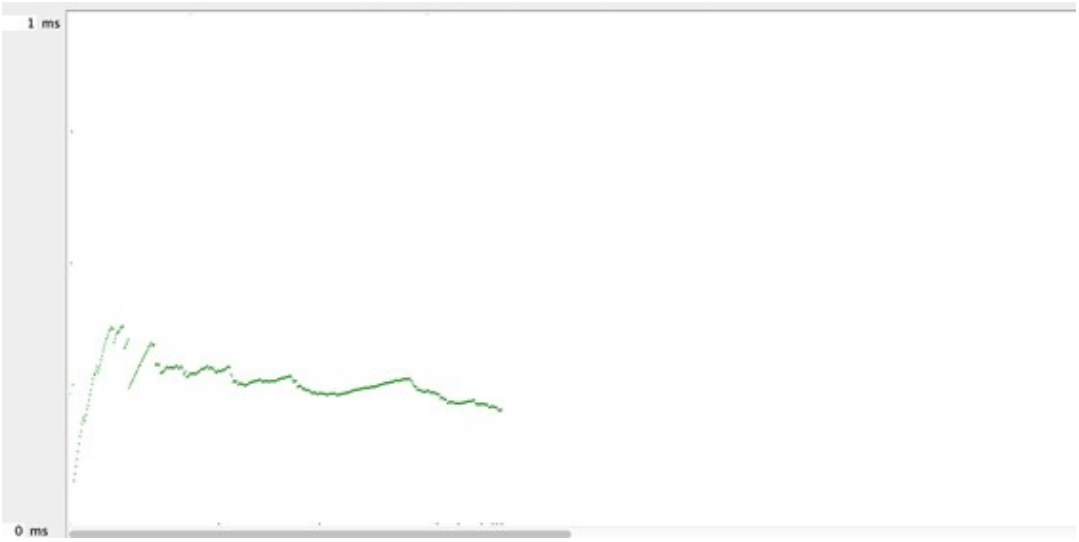

Figure 13. Graph of mean, median, dispersion, and performance (represented as the current number of requests/minutes that the server handles), this for the test of 10 recurring users.

Graphics to show Data Mean Median Deviation Performance

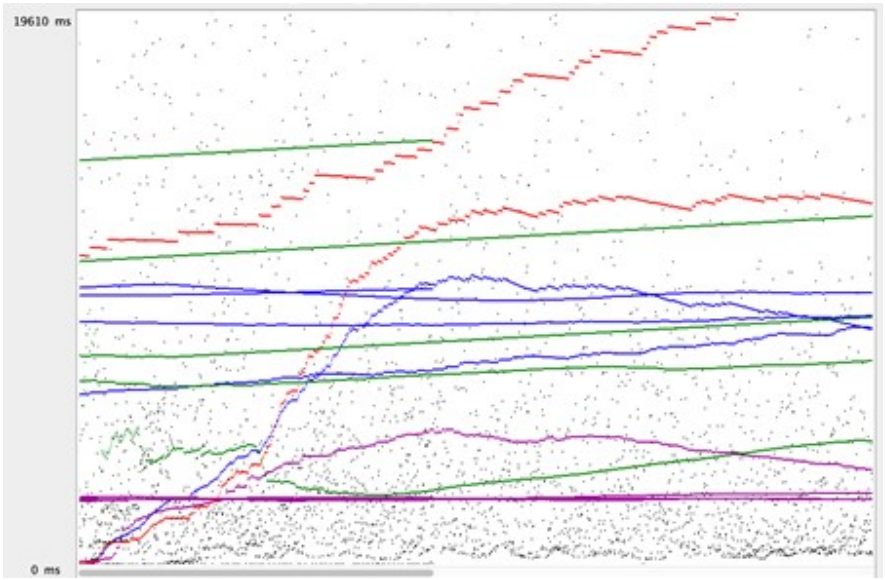

Figure 14. Graph of mean, median, dispersion, and performance (represented as the current number of requests/minutes that the server handles), this for the test of 100 recurring users.

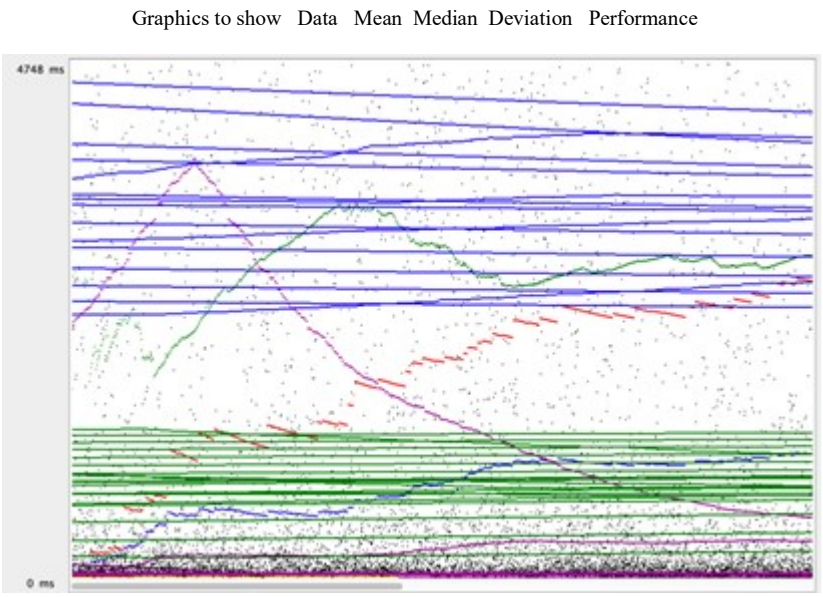

Figure 15. Graph of mean, median, dispersion, and performance (represented as the current number of requests/minutes that the server handles), this for the test of 1000 recurring users.

6.4 USABILITY TESTING

The captcha proposal was tested with 100 people.

In general, the individuals participating in this test have the characteristics of the table 2:

Table 2. Usability by school grade

Age range	Individuals	Minimum scholarship	Maximum scholarship	# Men	# Women
11 a 20	10	Primary school	High school	10	0
21 a 30	20	Degree	Master´s Degree	15	5
31 a 40	30	Degree	Master´s Degree	20	10
41 a 50	15	Master´s degree	Master´s Degree	10	5
51 a 60	10	Master´s degree	Master´s Degree	10	0
61 a 63	15	High school	PhD	15	0

At the end of the test, they were asked two questions, to know their experience of use:

From the first question, which can be seen in the graph in figure 16, 48% find it easy to use the application and only 7% find it difficult.

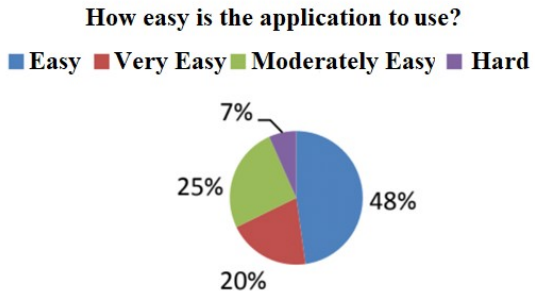

Figure 16. Application usage percentage.

Figure 17 is about identifying facial expressions. 43 percent were able to easily identify the face icons, while only 10 % found it difficult to recognize them.

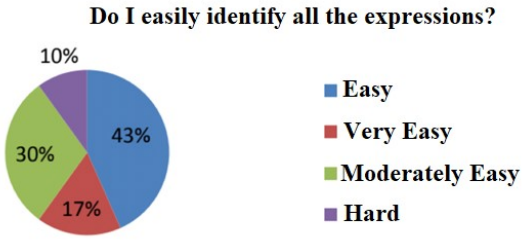

Figure 17. Percentage of identification of facial expressions.

Analyzing the results of these tests, it can be mentioned that:

- From the universe of respondents, and concentrating those who are at different levels, the easy answer (90%), is located the ages of 11, 18, 25, 28, 31, 37, 38, 40, 49, 51, 62. The range of studies is observed in table 3.

Table 3. Schooling of people who consider the proposal easy.

Scholarship	# individuals
Primary school	5
High school	10
Degree	40
Master´s Degree	25
PhD	10

- Some changed their mind after making more attempts to solve the captcha, that is, when they became familiar, they changed their answer from difficult to easy.
The range of studies is observed in table 4.

Table 4. Schooling of the people who changed their opinion about the proposal, from difficult to easy.

Scholarship	# individual
Primary school	1
High school	4
Degree	9
Master´s Degree	6
PhD	4

- From the universe of respondents and concentrating those that are considered at different levels and the answer as difficult (10%), the ages understood by them are: 25, 28, 31, 33, 51, 40, 52, 60, 63 With a range of studies of 5 for bachelor's degrees and 5 for masters.

7. DISCUSSION

From the different tests carried out, it can be observed that the icons generated with facial expressions are simple but unrecognizable by techniques such as Viola & Jones, which prevents the final user from being impersonated by a computer program.

As for the execution tests, there are no significant changes in implementation that impair performance even if the number of users increases; As can be seen in the statistics in Table 1 and Figures 13, 14, and 15.

A highlight in this article is the usability of the captcha that uses facial expressions. The tests carried out show that usability is high, but that it changes depending on the degree of studies and the age of the person. See table 2 and figures 16 and 17. These results show that it is easier for a person with more studies and younger age to renew a face into an icon, perhaps due to its relationship with technology. It is also important to note the influence of the use of facial expressions related to universal emotional states such as joy, sadness, etc.

More similar works, proposed by [Goswami, Singh, Vatsa, Powell, Noore] support the use of captchas with human faces, where it is necessary to identify among its components: different features, parts of the face, and a set of similar characteristics. Unlike this work, we use watermarks to identify the answer. Instead of just pointing the answer in the captcha, it is requested to capture it through a keyboard that changes randomly to generate a double challenge in the user, with a regeneration every 2 minutes, which allows limiting the attacks.

8. CONCLUSIONS AND FUTURE WORK

The captcha developed is based on the recognition of facial expressions that humans make naturally.

Distortions were applied to the created captcha, such as blurring, resizing, dilation, erosion, contrast change, salt & pepper noise, and geometric transformation, to prevent malware computer programs from being able to access the image generation process safely, in such a way that only the human being could direct a face on the icon.

As observed in the tests, a captcha with images of facial expressions is easy to resolve, since all people can relate it with emotions. The animated GIF maintains the cognitive difficulty that prevents it from being hacked; therefore, the inverse Turing test holds.

Users with little relation to computer technology have difficulties handling Captcha, but it is mainly due to a lack of experience in its use and not to the use of animated GIF images with easy expressions in the CAPTCHA.

For future work, the number of images of facial expressions can be increased, as well as the use of affective expressions of animals.

REFERENCES:

[1] F. J. Elizondo Garca, «Enredándose. CAPTCHA.,» *Ingenierías, Enero - Marzo,* vol. XI, nº 38, 2008.

[2] P. Wang, L. Wu, R. Cunningham y C. Zou, «Honeypot detection in advanced botnet attacks,» *Int. J. Information and Computer Security,* vol. 4, nº 1, pp. 30-51, 2010.

[3] A. Ahmad El, J. Yan y N. Wai-Yin, «CAPTCHA Design. Color, Usability, and Security,» *IEEE INTERNET COMPUTING.*

[4] J. F. R. Buendía, Seguridad informática, España: McGraw-Hill, 2013.

[5] B. Kurt Alfred Kluever, Evaluating the Usability and Security of a Video CAPTCHA, Rochester, 2008.

[6] R. Gossweiler, M. Kamvar y S. Baluja, «What's Up CAPTCHA? A CAPTCHA Based on Image Orientation,» *In Proceedings of the 18th international conference on World wide web,* pp. 841-850, 2009.

[7] G. Goswami, B. M. Powell, M. Vatsa, R. Singh y A. Noore, «Face Recognition CAPTCHA,» *Future Generation Computer Systems,* nº 412-417, pp. 59-68, 2012.

[8] G. Goswami, B. M. Powell, M. Vatsa, R. Singh y A. Noore, «FaceDCAPTCHA: Face detection based color image CAPTCHA,» *Elsevier,* pp. 59-68, 2012.

[9] A. Kumarasubramanian, R. Ostrovsky, O. Pandey y A. Wadia, «Cryptography Using Captcha Puzzles,» *International Association for Cryptologic Research,* pp. 89-106, 2013.

[10] G. Goswami, B. M. Powell, M. Vatsa, R. Singh y A. Noore, «FR-CAPTCHA: CAPTCHA Based on Recognizing Human Faces,» *PLoS ONE,* vol. 9, nº 4, 2014.

[11] M. Fujita, Y. Ikeya, J. Kani y M. Nishigaki, «Chimera CAPTCHA: A Proposal of CAPTCHA Using Strangeness in Merged Objects,» *T. Tryfonas and I. Askoxylakis,* pp. 48-58, 2015.

[12] C. J. Hernández-Castro, M. d. R-Moreno, D. F. Barrero y S. Gibson, «Using machine learning to identify common flaws in CAPTCHA design: FunCAPTCHA case analysis,» *Computers & Security,* nº 70, pp. 744-756, 2017.

[13] S. Gao, M. Mohamed, N. Saxena y C. Zhang, «Emerging-image Motion CAPTCHAs: Vulnerabilities of Existing Designs, and Countermeasures,» *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,* 2017.

[14] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman y D. Perez-Cabo, «No Bot Expects the DeepCAPTCHA! Introducing Immutable Adversarial Examples, With Applications to CAPTCHA Generation,» *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,* vol. 12, nº 11, 2017.

[15] D. Lin, F. Lin, Y. Lv, F. Cai y D. Cao, «Chinese Character CAPTCHA Recognition and performance estimation via a deep neural network,» *Elsevier,* pp. 11-19, 2018.

[16] W. Lee y E. Uzun, «Captcha check could boost phone security,» *Biometric Technology Today,* p. 2, 2018.

[17] M. Ogiela, N. Krzyworzeka y L. Ogiela, «Application of knowledge-based cognitive CAPTCHA in Cloud of Things security,» *Concurrency Computat Pract Exper,* nº 30, 2018

[18] T. Lawan, «Application of Pattern for New CAPTCHA Generation Idea,» *Springer International Publishing AG, part of Springer Nature,* p. 257–264, 2018.

[19] P. Ekman, Emotions Revealed, New York: Times Books, 2003.

[20] S. C. Romaniz, «Seguridad de aplicaciones web: vulnerabilidades en los controles de acceso,» *In XIV Congreso Argentino de Ciencias de la Computación.,* 2008.

[21] Paul Viola, Michael J. Jones, "Robust Real-Time Face Detección", International Journal of Computer Visión 52(2), 137-154, Kluwer Academic Publishers, Netherlands 2004.

[23] F. J. Diaz, C. M. Tzancoff Banchoff, A. S. Rodríguez y V. Soria, «Usando Jmeter para pruebas de rendimiento,» XIV Congreso Argentino de Ciencias de la Computación, 2008.

[24] APACHE, «APACHE,» APACHE, 14 07 2020. [On line]. Available: https://jmeter.apache.org/. [Last access: 14 07 2020]