Prueba de concepto de interfaz *touchless* en teclado numérico aleatorio para mitigación de *shoulder surfing* en cajeros automáticos

Bruno Fabrizio Ríos Villegas
20163498@aloe.ulima.edu.pe
https://orcid.org/0009-0009-7832-5589
Universidad de Lima, Perú

Carlos Martin Torres Paredes
cmtorres@ulima.edu.pe
https://orcid.org/0000-0002-7464-5392
Universidad de Lima, Perú

Recibido: 27 de julio del 2023 / Aceptado: 23 de octubre del 2023
doi: https://doi.org/10.26439/interfases2023.n018.6557

RESUMEN. La inclusión financiera en el Perú está en aumento, pues ya el 56 % de los adultos tiene productos financieros. Esto ha incrementado el uso de cajeros automáticos y los riesgos asociados a ellos, como el *shoulder surfing*. Buscando mitigar el riesgo de este ataque, se hizo una prueba de concepto de interfaz *touchless* que permite a los usuarios ingresar su PIN de manera segura, proponiendo un ejemplo para que sea usado por entidades bancarias o fabricantes de cajeros automáticos. Para esto, se generaron secuencias desordenadas aleatoriamente con los números del 0 al 9 sin que estos se repitan. Luego, se implementan sensores infrarrojos para ingresar los números del PIN. Se realizaron pruebas de mitigación y usabilidad con un grupo de 16 personas. La primera prueba mostró resultados alentadores, pues a los atacantes se le dificulta identificar los dígitos ingresados por los usuarios y solo lograron registrar el 25 % correctamente. Asimismo, en las pruebas de usabilidad se obtuvo un promedio general de usabilidad de 78.4375, situando a la interfaz en un rango B +, por encima del umbral de 68 puntos. Considerando esto, se concluye que la propuesta cumple con el objetivo de permitir al usuario ingresar su PIN de manera segura ante ataques de *shoulder surfing*.

PALABRAS CLAVE: interfaces *touchless* / cajeros automáticos / *shoulder surfing* / teclado numérico aleatorio

PROOF OF CONCEPT OF TOUCHLESS INTERFACE ON RANDOM KEYPAD FOR ATM SHOULDER SURFING MITIGATION

ABSTRACT. Financial inclusion in Peru is on the rise, with 56% of adults already having financial products. This has increased the use of ATMs and the risks associated with them, such as shoulder surfing. To mitigate the risk of this attack, a proof of concept of a touchless interface that allows users to enter their PIN securely was developed, proposing an example for use by banking institutions or ATM manufacturers. For this purpose, randomly disordered sequences of numbers from 0 to 9 were generated without repeating them. Then, infrared sensors were implemented to enter the PIN numbers. Mitigation and usability tests were performed with a group of 16 people. The first test showed encouraging results, as the attackers found it difficult to identify the digits entered by the users and only managed to register 25% correctly. Likewise, in the usability tests, an usability average of 78.4375 was obtained, placing the interface in a B+ range, above the threshold of 68 points. Considering this, it is concluded that the proposal meets the objective of allowing the user to enter his PIN securely against shoulder surfing attacks.

KEYWORDS: touchless interfaces / automated teller machines / shoulder surfing / random keypad

1. INTRODUCCIÓN

Los cajeros automáticos son una infraestructura de tecnología computarizada que provee a clientes de instituciones financieras el acceso a transacciones en un espacio público sin la necesidad de personal humano (Edem Udo Udo et al., 2017). Según la Superintendencia de Banca y Seguros y AFP (2020), se estimó que la cantidad de cajeros automáticos en el Perú en el año 2019 era de 28 407. Además, según datos de Statista (2023a), en el año 2021 había 117,24 cajeros automáticos por cada 100 000 adultos peruanos. Esto es relevante puesto que, según Toledo y León (2023), la inclusión financiera en el país alcanzó el 56 %, lo cual significa que ese mismo porcentaje de personas adultas tienen, al menos, un producto financiero.

Lamentablemente, el uso de estos dispositivos también conlleva riesgos. En el ámbito de la seguridad, uno de estos riesgos está asociado al PIN. Un inconveniente es que las personas tienen por costumbre utilizar el mismo código para autenticar diversos dispositivos o cuentas (celular, laptop, tarjeta, etcétera) (Rajarajan et al., 2014). Asimismo, se suele elegir secuencias fácilmente recordables, como fechas de nacimiento o números de casa. Paralelamente, los delincuentes han ideado diversos métodos para obtener el PIN de las personas como el *shoulder surfing*, el *keylogging* y el *phishing* (Rajarajan et al., 2014). Según Abhishek et al. (2019), el número de robos y fraudes relacionados con cajeros automáticos crece cada día porque más del 90 % son vulnerables a ataques como el *shoulder surfing*. De hecho, una encuesta realizada por Ipsos (2019) revela que ya son 400 000 los peruanos que han sido víctimas de algún tipo de robo o fraude financiero.

A pesar de que esta problemática ha tratado de ser mitigada por investigadores en años previos, haciendo uso de teclados aleatorios, se encuentra un vacío en la adición de nuevas tecnologías a las propuestas presentadas. Por esto, el presente trabajo tuvo como objetivo la creación de una prueba de concepto de interfaz *touchless* en teclado numérico aleatorio para la mitigación del riesgo de *shoulder surfing* en cajeros automáticos.

El artículo inicia explorando los antecedentes. Después, se presenta la sección del marco conceptual. Luego, se describe la metodología utilizada, así como la experimentación y resultados. Finalmente, se discuten los resultados mostrados, se presentan las conclusiones, se indican las limitaciones de la investigación y se exploran trabajos futuros.

2. ANTECEDENTES

En este apartado se exploran investigaciones realizadas por diversos autores en las dos dimensiones de la solución a implementar.

2.1. Soluciones ante ataques de acceso a información confidencial

Para prevenir el ataque de *shoulder surfing*, Adithya et al. (2018) implementaron un método llamado *key shuffling method*, el cual hace que los dígitos de un teclado numérico (o *keypad*)

aparezcan de manera aleatoria cada vez que un usuario ingresa su PIN. Los investigadores señalaron que esta forma particular de mostrar el *keypad* evitará que se pueda descifrar el PIN que se está ingresando, basándose en la posición usual que las teclas ocupan. Siguiendo esta misma línea, Shukla et al. (2018) implementaron un teclado numérico aleatorio y agregaron tecnología de reconocimiento facial. Los autores indican que esta combinación ayudará a evitar el *shoulder surfing*, pero que la tecnología de reconocimiento facial está asociada con altos costos. Por otro lado, Agarwal et al. (2011) implementaron un método que ayude a mitigar el riesgo de *shoulder surfing y* también el de *key logging* Para esto crearon un teclado dinámico que cambia su disposición cada vez que se da clic en él; además, las teclas únicamente eran visibles durante unos segundos antes de tocarlas, luego eran cubiertas por una capa de color. Finalmente, Maiti et al. (2017) complementaron la idea de un teclado aleatorio con realidad aumentada. Los autores inician explicando diferentes técnicas de aleatorización que pueden ser utilizadas y, una vez que la técnica ha sido elegida, el usuario debe colocarse unas gafas que muestran la nueva disposición del teclado. En la Tabla 1 puede observarse un resumen de estas propuestas.

Tabla 1

Resumen de propuestas de teclado

Autores	Propuesta de teclado	Conclusiones
Adithya et al. (2018)	"*Key shuffling method*". La disposición del *keypad* será diferente para cada usuario.	Se disminuirá la posibilidad de identificar los dígitos del PIN basándose en la posición usual de las teclas.
Shukla et al. (2018)	*Keypad* dinámico virtual que cambia con cada usuario y reconocimiento facial.	Confunde a las personas que tratan de adivinar la contraseña que se está digitando y evita que puedan robarla.
Agarwal et al. (2011)	Teclado dinámico virtual que cambia con cada usuario y cada vez que se da clic en él, más una capa de colores que cubre las teclas antes de presionarlas.	El *software* de *key logging* solo capturó colores y los atacantes solo lograron identificar entre el 33 % y 67 % de teclas presionadas.
Maiti et al. (2017)	Teclado aleatorio visible mediante gafas de realidad aumentada.	Es virtualmente imposible para un atacante adivinar las teclas que están siendo presionadas. Sin embargo, aumenta el tiempo y se reduce la precisión del digitado.

Gracias a las investigaciones realizadas en años previos, puede notarse que el uso de teclados aleatorios es una técnica útil para mitigar el riesgo del ataque de *shoulder surfing*. Asimismo, dentro de los métodos complementarios no se ha encontrado una propuesta *touchless*.

2.2. Propuestas de interfaz touchless

Montanaro et al. (2016) implementan una interfaz para el control de un elevador. Los investigadores basaron la interfaz en el movimiento de las manos del usuario y propusieron tres implementaciones distintas, las cuales posteriormente se evaluaron utilizando la prueba

UMUX. Este es un cuestionario diseñado para medir la usabilidad percibida; tiene cuatro ítems que son evaluados desde el 1 (muy en desacuerdo) hasta el 7 (muy de acuerdo) (Lewis et al., 2013). En primer lugar, se tiene una interfaz lineal basada en el seguimiento de la mano del usuario en los ejes X e Y. Luego, una interfaz circular en la que el usuario selecciona el piso moviendo su mano circularmente. Finalmente, una interfaz de botones que replica el panel de botones usual de un elevador.

Por otro lado, Chakraborty et al. (2016) implementan una interfaz llamada *Sporshohin*, la cual consiste en cuatro LED y cuatro fototransistores ubicados en pares, en cuatro lados de un cubo. Para que funcione, el usuario debe acercar su mano a uno de los lados para que la luz del LED rebote en ella, sea recibida por el fototransistor y se ejecute una acción en pantalla. En la Tabla 2 se presenta un resumen de las mencionadas propuestas.

Tabla 2

Resumen de propuestas touchless

Autores	Propuesta *touchless*	Ventajas	Limitaciones
Chakraborty et al. (2016)	*Sporshohin*	Dispositivo versátil que puede adaptarse a diversas situaciones; de bajo costo, bajo consumo energético, robusto y fácil de usar. Permite que dispositivos puedan ser usados en espacios públicos de manera limpia y segura, pues estos no son tocados.	Rango de reconocimiento de los gestos no muy amplio. Se menciona que un extraño podría memorizar los gestos y saber qué datos se ingresaron, por lo que se propone que los dígitos sean dinámicos en vez de estáticos.
Montanaro et al. (2016)	*Widget linear*	Puntaje promedio UMUX de 84,27 y desviación estándar de 12,62. No desorienta a los usuarios.	Necesidad de suficiente espacio para que sea usado correctamente.
	Widget de botones.	Puntaje promedio UMUX de 83,40 y desviación estándar de 19,15.	
	Widget circular.	Puntaje promedio UMUX de 70,30 y desviación estándar de 24,81.	Dificultad por parte de los usuarios para comprender cómo usarlo.

3. MARCO CONCEPTUAL

En esta sección se explican, con rigurosidad académica, algunos conceptos necesarios para entender la investigación realizada.

3.1. PIN

Uno de los métodos de autenticación más comunes es el número de identificación personal (PIN, por su sigla en inglés), el cual usualmente consta de cuatro o seis dígitos. Este código es usado para desbloquear un teléfono inteligente, retirar dinero de un cajero automático,

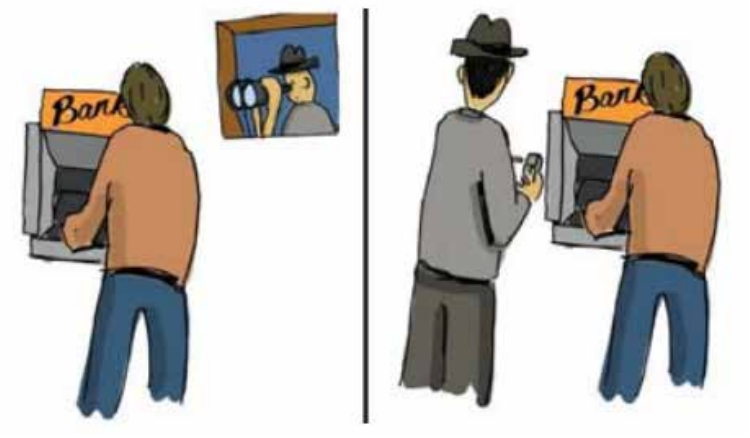
abrir una puerta bloqueada, etcétera. Este tipo de autenticación es conocida por ser rápida y fácil de utilizar; pero puede ser peligroso usarla bajo ciertas circunstancias. Por ejemplo, un atacante podría obtener el PIN de una persona al observarla cuando lo ingresa al utilizar un cajero automático. Esto es algo conocido como *shoulder surfing* (Bultel et al., 2018).

3.2. *Shoulder surfing*

El *shoulder surfing* es un ataque que no requiere de tecnología, pues utiliza técnicas de observación directa como el seguimiento de los movimientos del hombro o mano del usuario para obtener información (Adithya et al., 2018). Asimismo, Abhishek et al. (2019) lo definen como el robo de contraseñas, códigos o del PIN directamente, observando los gestos de la víctima mientras esta se encuentra ingresando los datos en un teclado. En la Figura 1 puede observarse un ejemplo de este ataque.

Figura 1

Ejemplo de shoulder surfing



Nota. De Abhishek et al. (2019).

3.3. Pruebas de mitigación

La detección de vulnerabilidades estudia métodos para detectar los problemas de seguridad conocidos basándose en las características de estos. La mitigación de vulnerabilidades son los métodos que se utilizan para solucionarlas o mitigar el impacto que tienen (Yu et al., 2020). Las pruebas de mitigación se realizan para medir el efecto del método de mitigación que se haya aplicado.

3.4. Pruebas de usabilidad

La usabilidad percibida es un componente importante del constructo de alto nivel de la usabilidad, la cual es una parte fundamental de la experiencia de usuario (UX). Motivados

por la afluencia de psicólogos experimentales en este campo a inicios de la década del ochenta, los primeros cuestionarios de usabilidad estandarizados aparecieron a finales de la mencionada década. Dos de los cuestionarios estandarizados más populares usados para medir este componente son el *Computer System Usability Questionnaire* (CSUQ) y el *System Usability Scale* (SUS); no obstante, existen otros como el *Usability Metrics for User Experience* (UMUX) y su variante UMUX-LITE (Lewis, 2018).

3.5. SUS

El SUS se ha vuelto bastante popular para la evaluación de la usabilidad percibida y es considerado el estándar de la industria debido a su excelente fiabilidad, validez y sensibilidad a diferentes variables. Este cuestionario cuenta con diez ítems. Cinco de ellos (los impares) están expresados en un tono positivo, y los otros cinco (los pares), en un tono negativo. Todos los ítems pueden ser respondidos en una escala del 1 (muy en desacuerdo) al 5 (muy de acuerdo) (Lewis, 2018).

Para calcular la puntuación global, se realizan los siguientes pasos. Primero, cada ítem se convierte a una escala del 0 al 4, donde los números más altos indican una mayor cantidad de usabilidad percibida. Luego, estos puntajes son sumados. Finalmente, la suma se multiplica por 2.5. Este proceso produce puntajes en un rango entre 0 y 100. Usando datos de 446 estudios y más de 5000 respuestas individuales se determinó que la puntuación media general del SUS es de 68 puntos, con una desviación estándar de 12.5 (Borsci et al., 2015). Asimismo, se establecieron grados a los rangos de puntaje obtenidos usando SUS, los cuales van desde F (absolutamente insatisfactorio) hasta A + (absolutamente satisfactorio). Estos rangos son: F (0-51.7), D (51.8-62.6), C- (62.7-64.9), C (65-71), C+ (71.1-72.5), B- (72.6-74), B (74.1-77.1), B+ (77.2-78.8), A- (78.9-80.7), A (80.8-84), A+ (84.1-100).

3.6. UMUX

El *Usability Metric for User Experience* es un cuestionario diseñado para medir la usabilidad percibida de un sistema. Es consistente con el cuestionario estándar de la industria, SUS, pero utilizando únicamente cuatro ítems, en vez de los diez de SUS. Los ítems de este cuestionario varían en tono y, a diferencia de SUS, van en una escala del 1 (muy en desacuerdo) al 7 (muy de acuerdo). En las pruebas realizadas entre SUS y UMUX, se pudo encontrar que hay una correlación muy alta entre los puntajes obtenidos mediante ambos cuestionarios, lo cual sugiere fuertemente que UMUX es estadísticamente equivalente a SUS (Lewis et al., 2013).

3.7. Arduino

Arduino es una plataforma de computación física de código abierto, basada en una placa microcontroladora y un entorno de desarrollo. Usando Arduino se pueden crear diseños fácilmente, con conocimientos sencillos de electrónica y programación. Los proyectos realizados de esta manera pueden ser autónomos o pueden comunicarse con un *software*

ejecutado mediante una computadora. La placa Arduino UNO cuenta con catorce pines de entrada/salida y seis pines de entrada analógica, un puerto USB, un puerto de energía, un botón de reinicio, entre otras partes. El entorno de desarrollo puede ser desplegado en Windows, Linux o MacOS y es llamado Arduino IDE. En este se puede escribir código, el cual es llamado *Sketch* y está basado en lenguaje C y C + + (Ahmad, 2013).

4. METODOLOGÍA Y EXPERIMENTACIÓN

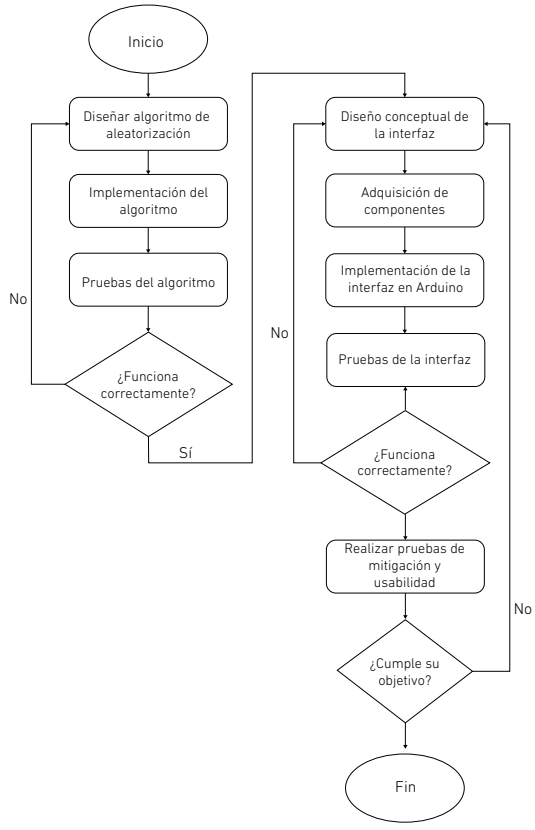
En este apartado se presentan los pasos de la metodología para la implementación de la prueba de concepto de interfaz, así como la descripción detallada de las pruebas de mitigación y usabilidad aplicadas.

4.1. Resumen de la metodología

Con intención de dar un panorama general de la investigación, en la Figura 2 se muestra un diagrama que resume el proceso que la metodología siguió.

Figura 2

Diagrama resumen



La metodología inicia con el diseño e implementación de un algoritmo para la generación de números aleatorios. Estos son usados para producir la secuencia que se le presenta en pantalla a cada usuario. Una vez que el algoritmo fue implementado se realizaron pruebas para medir su efectividad. Estas pruebas consistieron en comparar las secuencias aleatorias que el algoritmo va generando, de forma que se obtuvo el número de secuencias iguales en un determinado número de ejecuciones. Se estableció como límite que el promedio de secuencias repetidas en 1500 usos (ejecuciones) de la interfaz sea menor a uno.

Luego, se procedió a adquirir los componentes necesarios para construir la interfaz de manera física. Finalizado este proceso, se validó que los datos puedan ser ingresados correctamente. Para realizar esta validación, se generaron secuencias aleatorias y se colocaron diferentes códigos PIN, revisando que cada valor ingresado correspondiera con el dígito del PIN que se desea colocar. Al corresponder los valores en su totalidad, entonces se está logrando la funcionalidad deseada y se pasó a la siguiente fase.

En la etapa final se aplicaron dos tipos de pruebas: de mitigación y de usabilidad. La primera tuvo como objetivo comprobar que la interfaz cumple su misión de mitigar el riesgo de ataque de *shoulder surfing*. En esta prueba se reunió a un grupo de personas, el cual se subdividió en dos grupos de igual tamaño: usuarios y atacantes. La prueba consistió en que los atacantes intentan descubrir los dígitos del PIN de los usuarios mientras estos los ingresan en la interfaz; el grupo de usuarios no supo que estaba siendo observado. Para medir la efectividad de esta prueba, se recogieron y analizaron los resultados, para posteriormente compararlos con los de la literatura actual, de forma que se conozca si la implementación obtuvo resultados similares a otras implementaciones. La segunda prueba buscó conocer la experiencia de los usuarios hacia el uso de la interfaz implementada. Para esto, se le solicitó al grupo previamente reunido que haga uso de la interfaz y que, al terminar, completen el cuestionario SUS. Para medir la efectividad de esta prueba, se recogieron y analizaron los datos y se compararon con la puntuación media general de usabilidad SUS, la cual es de 68 puntos.

4.2. Diseño del algoritmo de aleatorización

El algoritmo que se desarrolló tiene como objetivo la generación de un arreglo que contenga los números del 0 al 9 y que los presente en un orden aleatorio. Para esto se creó un arreglo de diez elementos. Luego, se generaron los números del 0 al 9 de manera pseudo aleatoria y se agregaron al arreglo. Para que ninguno de los números se repita, se implementó una función que compara los números generados con los que están dentro del arreglo, de forma que únicamente agregue los que no están presentes.

4.3. Implementación del algoritmo de aleatorización

La implementación del algoritmo generador de secuencias aleatorias será realizada en Arduino IDE, el entorno de programación para Arduino. Véase la Figura 3.

Figura 3

Algoritmo generador de secuencias aleatorias

La función **GeneraciónSecuencias** realiza las operaciones necesarias para generar secuencias aleatorias que le serán presentadas a cada usuario cuando utilice la interfaz.

Entrada: arreglo: arreglo de 10 elementos.

Salida: arreglo: el arreglo con los números ya ingresados que forman la secuencia aleatoria.

Variables: randNumber: entero que almacena un número aleatorio entre el 0 y 9.
iniciado: indica que el algoritmo ha funcionado y la secuencia ha sido generada.
pos: indica la posición en el arreglo.

```
GeneraciónSecuencias(arreglo):
1.    arreglo ← VaciarArreglo()
2.    Mientras pos < 10:
3.        randNumber ← random(10)
5.        Si ValorEnArreglo(randNumber, arreglo) = falso Entonces
6.                arreglo[pos] ← randNumber
7.            pos ← pos + 1
8.        Fin Si
10.   Fin Repetir
11.   pos ← 0
13.   Imprimir arreglo
```

El Algoritmo 1 se encarga de generar e imprimir las secuencias aleatorias y utiliza otros dos algoritmos para funcionar correctamente. El primero es «VaciarArreglo()», el cual se encarga de quitar los elementos del arreglo para que el proceso pueda volver a iniciar. El segundo algoritmo usado es el encargado de verificar si los números generados se encuentran en el arreglo. Véase la Figura 4.

Figura 4

Algoritmo encargado de verificar si los números aleatorios generados se encuentran en el arreglo

La función **ValorEnArreglo** se encarga de comprobar si el número generado aleatoriamente se encuentra o no en el arreglo.

Entrada: arreglo: arreglo de 10 elementos.
randNumber: número entre el 0 y 9 generado aleatoriamente.

Salida: estado: indica verdadero si el número ya está en el arreglo o falso si no se encuentra en él.

Variables: i: contador para el bucle Desde.

ValorEnArreglo(randNumber, arreglo):

```
1.  i ← 0
2.  Desde i = 0 hasta 10 Hacer
3.      Si arreglo[i] = randNumber Entonces
4.          estado ← verdadero
5.      Sino
6.          estado ← falso
7.      Fin Si
8.  Fin Repetir
9.  Retornar estado
```

4.4. Diseño conceptual de la interfaz en Arduino

Con base en la investigación de Montanaro et al. (2016), se decidió usar una pantalla lineal, pues fue la que obtuvo un mayor promedio de usabilidad. Esto se tradujo, en el contexto de la presente investigación, en el uso de una pantalla que muestra la secuencia para cada usuario de manera lineal. Del mismo modo, el método de entrada de datos (es decir, lo que el usuario utilizará para ingresar los números presentados en la pantalla para formar su PIN) está dispuesto de manera horizontal. Finalmente, se agregó un botón que el usuario debe presionar para que aparezca su secuencia aleatoria en pantalla y otro botón para confirmar su PIN. La Tabla 3 muestra un resumen de los resultados de la mencionada investigación.

Tabla 3

Resultados de usabilidad utilizando el cuestionario UMUX

Tipo	Puntaje promedio	Desviación estándar
Lineal	84,27	12,62
Circular	70,30	24,81
Botones	83,40	19,15

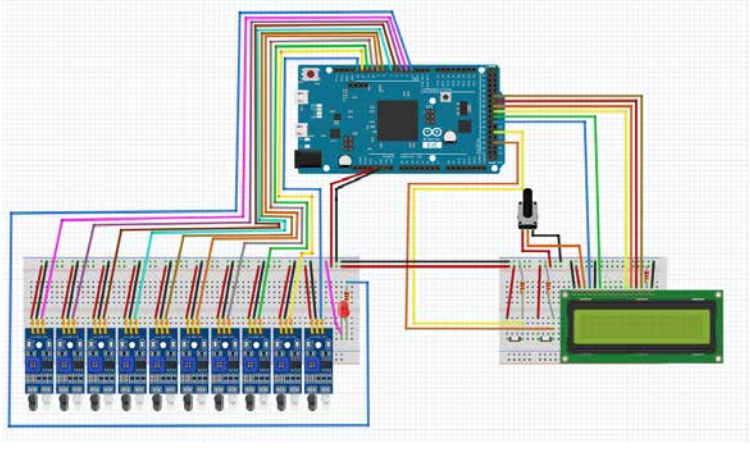
Nota. Adaptado de Montanaro et al. (2016).

4.5. Implementación de la interfaz en Arduino

La placa que se usó es un Arduino DUE. Para visualizar la secuencia generada, se usó una pantalla LCD1602A. En esta misma pantalla se imprimen los números del PIN que el usuario ingrese. Además, se cuenta con un pulsador que el usuario deberá presionar para que el algoritmo inicie y muestre la secuencia. Para el ingreso del PIN se usaron diez sensores infrarrojos FC-51. Cuando un usuario utiliza la interfaz, se generan los diez números del 0 al 9 en un orden aleatorio y se imprimen en pantalla. Luego, el usuario es capaz de ingresar su PIN de cuatro dígitos usando estos sensores FC-51. Cada sensor sirve para ingresar uno de los números mostrados en pantalla, estando asignada la posición del número en pantalla a la posición del sensor. Por ejemplo, si la secuencia fuera 9013674258 y el PIN del usuario fuera 1234, el usuario debería acercarse en orden al tercer sensor, luego al octavo sensor, después al cuarto sensor y, finalmente, al séptimo sensor para ingresar su PIN. Debido a que la secuencia cambia con cada usuario, el valor que se ingresa con cada sensor cambia dinámicamente en cada uso. Para mayor orden se usaron dos *protoboards* distintas. Por un lado, en una *protoboard* pequeña se colocó la pantalla LCD, el pulsador de inicio y el pulsador de confirmación. Por otro lado, en una *protoboard* más larga, se hizo la implementación de los diez sensores y también se incluyó un LED, el cual se enciende cuando se ingresa un número del PIN y busca otorgarle retroalimentación al usuario, de forma que sepa que el dígito de su PIN fue ingresado satisfactoriamente. Cabe destacar que, debido a que un PIN cuenta con cuatro dígitos, la interfaz solo permite ingresar cuatro números. En caso de que el usuario cometa un error al digitar uno de los números de su PIN, deberá presionar nuevamente el pulsador de inicio para que se le genere otra secuencia aleatoria. También es importante indicar que el pulsador de fin únicamente funcionará cuando el usuario haya ingresado cuatro dígitos. En la Figura 5 puede observarse el diagrama de conexión.

Figura 5

Diagrama de conexión



Para que el *hardware* de sensores funcionara correctamente, se desarrolló el algoritmo de control de sensores, el cual puede verse en la Figura 6.

Figura 6

Algoritmo de control de sensores

La función ControlDeSensores se encarga de implementar las sentencias necesarias para el correcto funcionamiento de los sensores.

Entrada:	arreglo: arreglo de 10 elementos. iniciado: booleano que indica cuando la secuencia aleatoria ha sido creada. sensor (K): los 10 sensores, del 0 al 9
Salida:	ninguna
Variables:	contador: entero encargado de almacenar la cantidad de sensores presionados.

```
ControlDeSensores(iniciado, arreglo, sensor):
1.      contador ← 0
2.      Si sensorK es activado & iniciado = verdadero & contador < 4 Entonces
3.              Imprimir arreglo [K]
4.              contador ← contador + 1
5.      Fin Si
```

4.6. Pruebas de mitigación y usabilidad

A continuación, se brinda una explicación detallada de las dos pruebas que fueron realizadas: de mitigación y de usabilidad.

4.6.1. Pruebas de mitigación

Para esta primera prueba, se reunió a un grupo de personas que cumplen tres condiciones: ser mayores de edad, tener una cuenta bancaria y haber usado un cajero automático previamente. La primera acción fue explicarles cómo se utiliza la interfaz. Seguidamente, el grupo de personas fue dividido de manera aleatoria en dos: al primer grupo se le asignó la función de usuarios y al segundo, la de atacantes. El grupo 1 tuvo como labor interactuar con la interfaz, para lo cual se les asignó un PIN aleatorio al momento de la prueba, para asegurar que nadie de los presentes lo haya conocido con antelación. Luego, se les dispuso en fila y, uno a uno, debieron ingresar el PIN que les fue asignado. El segundo grupo debió observar a los usuarios mientras ingresaban el PIN, para lo cual se les indicó que se sitúen en la fila, como si también estuvieran esperando su turno. Cabe destacar que no se le comunicó al grupo de usuarios que el otro grupo estaba allí para observarlos, a fin de que ingresaran su PIN con la mayor naturalidad posible.

4.6.2. Pruebas de usabilidad

Para realizar la prueba de usabilidad percibida, se les pidió a ambos grupos (los usuarios y los atacantes) que utilicen la interfaz e ingresen un PIN que se les brindó en ese mismo momento. El primer grupo, el de previos usuarios, ya había usado la interfaz en la prueba de mitigación, por lo que se encontró más familiarizado con ella y pudo ayudar brindando datos sobre cómo los usuarios más experimentados se sienten al utilizar la interfaz. Además, para acentuar esta diferencia, se realizaron tres rondas con este grupo; es decir, cada uno de los integrantes ingresó tres PIN distintos, uno por cada ronda. Por el contrario, el grupo que fungió como atacante en la prueba de mitigación, ahora, en una única ronda, tuvo su primera interacción con la interfaz y, por lo tanto, proveyó datos sobre usuarios totalmente nuevos con su uso. Para obtener resultados cuantitativos en esta prueba, se usó la métrica SUS (*System Usability Scale*), que es el cuestionario utilizado por expertos en la industria. A pesar de haber sido desarrollados por separado y tener distintos formatos, los diferentes cuestionarios (CSUQ, SUS, UMUX y UMUX-Lite) están fuertemente correlacionados y, al ser transformados a una escala común de 0 a 100 puntos, sus puntuaciones medias tienen magnitudes y calificaciones similares. Sin embargo, debido a la enorme cantidad de investigaciones realizadas con SUS, esta métrica es probablemente la mejor opción por defecto (Lewis, 2018). Una vez que los grupos concluyeron con el ingreso del, o de los, PIN, se les solicitó que completen el cuestionario individualmente y se registraron sus puntajes. Estos puntajes fueron analizados posteriormente, para obtener los resultados de la prueba de usabilidad. La Tabla 4 contiene las preguntas del cuestionario SUS validadas al idioma español.

Tabla 4

Preguntas del cuestionario SUS validadas al español

Número	Pregunta
1	Me gustaría usar esta herramienta frecuentemente.
2	Considero que esta herramienta es innecesariamente compleja.
3	Considero que la herramienta es fácil de usar.
4	Considero necesario el apoyo de personal experto para poder utilizar esta herramienta.
5	Considero que las funciones de la herramienta están bien integradas.
6	Considero que la herramienta presenta muchas contradicciones.
7	Imagino que la mayoría de las personas aprenderían a usar esta herramienta rápidamente.
8	Considero que el uso de esta herramienta es tedioso.
9	Me sentí muy confiado al usar la herramienta.
10	Necesité saber bastantes cosas antes de poder empezar a usar esta herramienta.

Nota. Tomado de Sevilla-González et al. (2020).

5. RESULTADOS

A continuación, se darán a conocer los resultados obtenidos al seguir los pasos de la metodología, así como los resultados de las pruebas de mitigación y usabilidad.

5.1. Resultados del algoritmo

Para validar los pasos 4.2 y 4.3 de la metodología, se realizó un algoritmo que utilice la misma lógica que el implementado en Arduino IDE. En este algoritmo se creó un arreglo que almacenará las secuencias generadas y las comparará con las nuevas que se vayan generando. Una vez que el algoritmo se haya detenido, se habrá obtenido un contador, el cual representa la cantidad de secuencias iguales dentro de un determinado número de repeticiones. Se midió usando 100, 500, 1000, 1500, 2000 y 5000 repeticiones. En la Tabla 5 puede observarse el resultado de las ejecuciones.

Tabla 5

Ejecuciones por conjunto de repeticiones

Rep.	Ejecuciones																
	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	Prom.	D.E.
5000	4	7	1	1	3	2	2	1	2	1	4	2	3	0	1	2,267	1,75
2000	2	0	0	2	0	0	1	1	1	1	0	1	0	0	1	0,667	0,723
1500	0	0	0	2	0	0	1	1	0	0	0	0	0	0	0	0,267	0,593
1000	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0,133	0,352
500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

La Tabla 5 muestra las 15 ejecuciones independientes realizadas para cada número de repeticiones (indicadas como E1 a E15). Se eligió un número amplio de ejecuciones para tener la posibilidad de obtener un promedio y una desviación estándar adecuadas. En los grupos que involucran 100 y 500 repeticiones, puede observarse que nunca se presentaron secuencias repetidas, por lo que tanto el promedio como la desviación estándar (D.E.) fueron de 0. En los grupos de 1000 repeticiones, se evidencia que en las ejecuciones E3 y E9 se encontró una secuencia repetida; debido a esto, el promedio obtenido es de 0,133 con una desviación estándar de 0,352. Por su parte, las ejecuciones con 1500 repeticiones mostraron poca repetición de secuencias; se calculó un promedio de 0,267 y una desviación estándar de 0,593. Respecto a las ejecuciones con 2000 repeticiones, empieza a notarse una cantidad ligeramente mayor de secuencias repetidas. Una vez obtenidos todos los datos, se calculó que en promedio se repiten 0,667 secuencias con una desviación estándar de 0,723. Finalmente, teniendo en cuenta las ejecuciones con 5000 repeticiones, puede observarse que en solo una ejecución, la E14, no se presentaron secuencias repetidas; en promedio, se repitieron 2,267 secuencias y

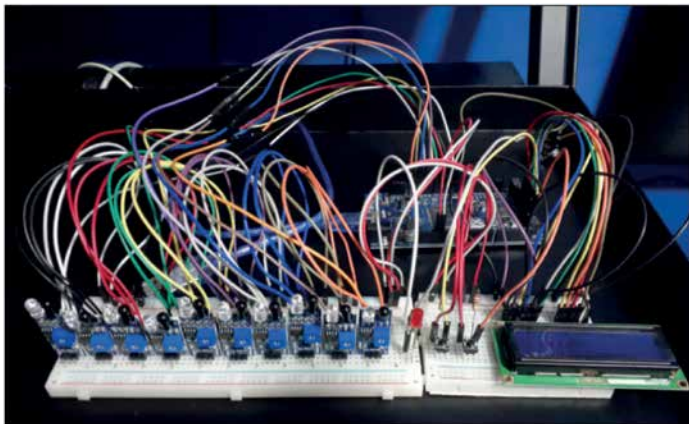
se tuvo una desviación estándar de 1,75. Posteriormente, en el apartado de discusión se hará un análisis de estos resultados.

5.2. Resultados de la implementación

Una vez que los componentes estuvieron disponibles, se procedió con la construcción de la interfaz. En la Figura 7 se muestra una imagen de la interfaz física.

Figura 7

Interfaz implementada



Se aprovechó que la pantalla LCD1602A cuenta con dieciséis columnas y dos filas para imprimir las letras «Sec:» en la fila superior, para que el usuario sepa dónde aparecerá la secuencia aleatoria generada. Además, se imprimieron las siglas «PIN:» en la fila inferior, para señalar al usuario en dónde aparecerán los dígitos que ingrese con los sensores. Además, la pantalla también es usada para mostrar la frase «PIN aceptado» cuando el usuario haya ingresado los cuatro dígitos de su PIN y haya presionado el pulsador de finalización. La Figura 5 muestra la implementación de la pantalla.

Figura 8

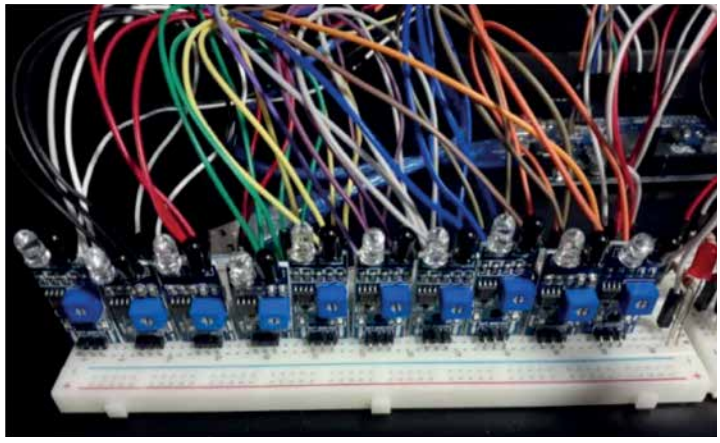
Implementación de la pantalla

Nota. En la foto de la izquierda puede observarse la secuencia en pantalla; al centro se observa que se ha ingresado el dígito de la primera posición (7); a la derecha se visualiza la pantalla de confirmación.

Del mismo modo, se hizo la implementación de los sensores para el ingreso del PIN del usuario. En la Figura 9 se puede observar una imagen con los diez sensores implementados.

Figura 9

Implementación de los sensores infrarrojos FC-51



Con la interfaz implementada y funcional, se procedió a realizar los diseños de la caja que contendrá el cableado de la interfaz. En la Figura 10 puede observarse la vista isométrica de esta caja, así como una vista superior. Asimismo, se realizó el modelado 3D de la caja, el cual puede visualizarse en la Figura 11.

Figura 10

Vista isométrica (izquierda) y superior (derecha) de la caja

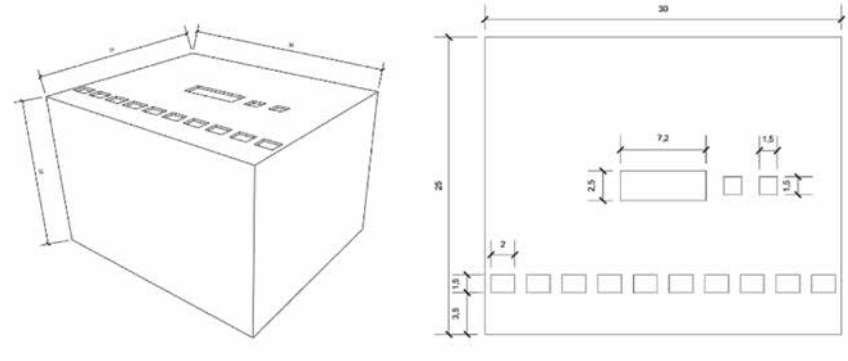
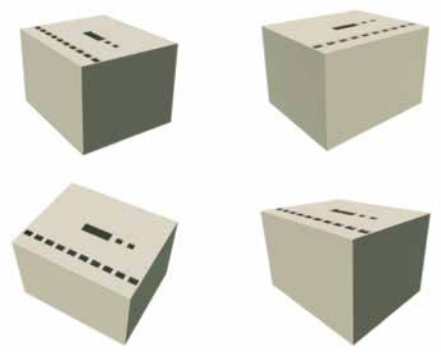


Figura 11

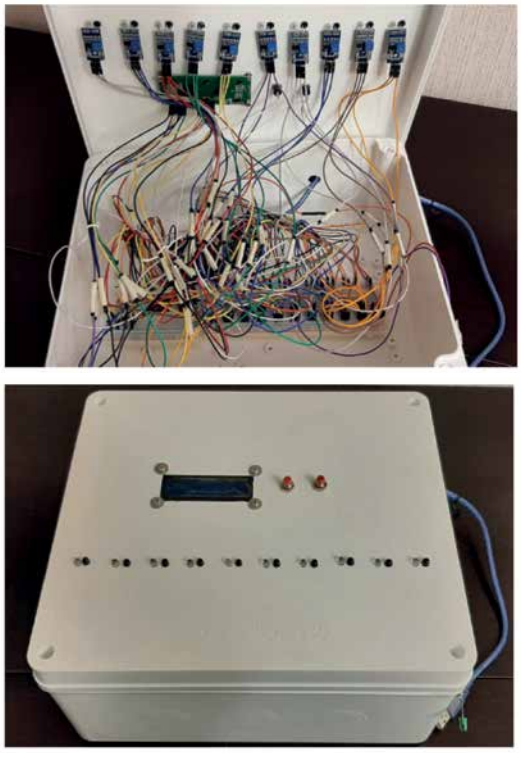
Modelado 3D de la caja



Como puede notarse, este contenedor permite que los usuarios únicamente vean los componentes necesarios para interactuar con la interfaz: la pantalla LCD, los pulsadores de inicio y fin, y los sensores. En la Figura 12 se muestra tanto la vista interna como la vista externa de la caja.

Figura 12

Vistas interna y externa de la caja final



5.3. Muestra para las pruebas

Para las pruebas se reunió a un grupo de 16 personas que cumplieran con las condiciones establecidas en la experimentación y se les dividió aleatoriamente en dos grupos. En la Tabla 6 pueden observarse características demográficas de la muestra.

Tabla 6

Demografía de la muestra

Usuario	Edad	Género	Nivel de estudios	Profesión/Ocupación	Grupo
1	67	Masculino	Superior universitaria incompleta	Jubilado	1
2	34	Masculino	Superior no universitaria completa	Ingeniero de Sistemas	1
3	34	Femenino	Superior universitaria completa	Abogada	1
4	58	Femenino	Superior universitaria completa	Ama de casa	1
5	53	Masculino	Superior universitaria completa	Militar	1
6	20	Femenino	Superior universitaria incompleta	Estudiante	1
7	22	Femenino	Superior universitaria incompleta	Estudiante	1
8	23	Femenino	Superior universitaria incompleta	Estudiante	1
9	55	Femenino	Superior universitaria completa	Abogada	2
10	69	Femenino	Secundaria completa	Ama de casa	2
11	23	Femenino	Superior universitaria incompleta	Estudiante	2
12	61	Femenino	Superior universitaria completa	Docente	2
13	19	Masculino	Superior universitaria incompleta	Estudiante	2
14	21	Masculino	Superior universitaria incompleta	Estudiante	2
15	23	Masculino	Superior universitaria incompleta	Estudiante	2
16	24	Masculino	Superior universitaria completa	Administrador	2

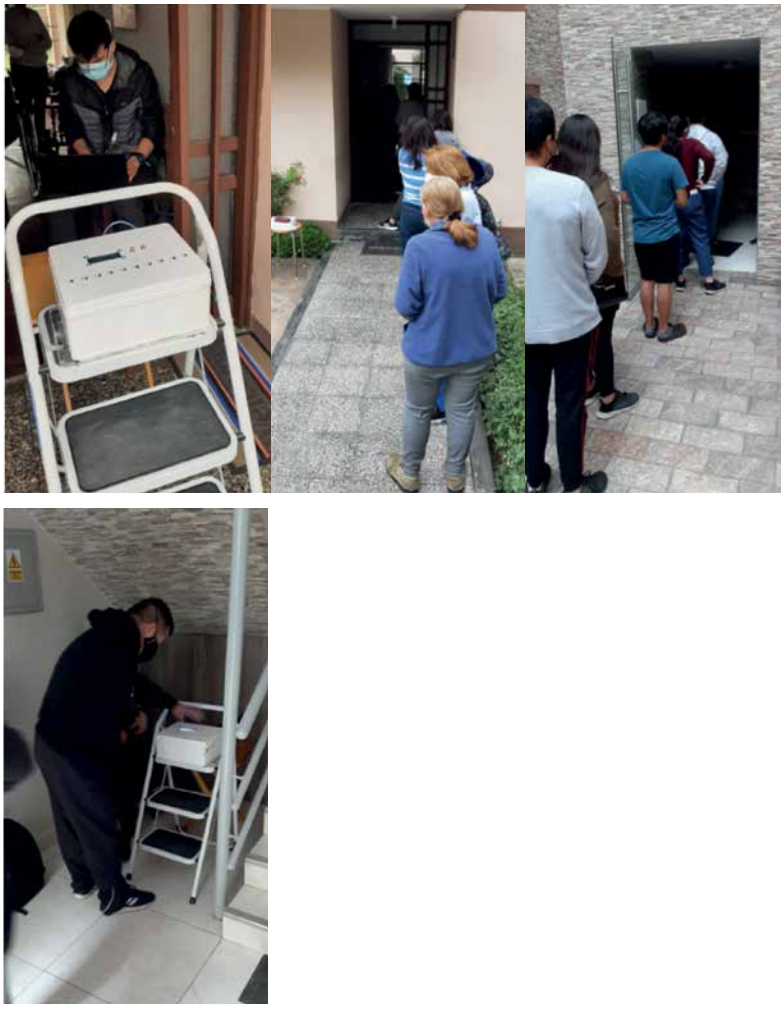
5.4. Resultados de las pruebas de mitigación

Se realizaron dos pruebas de mitigación en dos momentos del día, ambas con iguales condiciones y el mismo número de individuos: ocho personas. De este número de personas, en cada prueba, cuatro formaron parte del grupo 1 (usuarios) y otras cuatro integraron el grupo 2 (atacantes), dando un total de 16 personas distintas. Una vez que se tuvo a los grupos 1 y 2 separados, se procedió a explicar a cada uno la labor que debían realizar. Al grupo 1 se le volvió a instruir en el uso de la interfaz y se le brindó a cada integrante el PIN que deberán ingresar. Al grupo 2 se le reveló la verdadera naturaleza de la prueba y, en vez de brindarles un PIN que ingresar, se les encomendó la tarea de observar el PIN de la persona que tuvieran delante en la fila y memorizar los dígitos que lograran visualizar. Además, a los integrantes de este grupo se les solicitó que simularan hacer uso de la interfaz, pero sin ingresar ningún dígito realmente, pues la prueba de usabilidad a la que se los sometería después, requería que no hubieran usado antes la interfaz. Finalmente, se reunió a ambos grupos y se les colocó de forma intercalada

en una fila para iniciar la prueba. En la Figura 13 pueden visualizarse fotos tomadas en ambas pruebas de mitigación realizadas. Cabe destacar que, una vez finalizada la prueba, se volvió a separar a los grupos, se brindó otro PIN a cada usuario y se volvió a realizar la prueba, es decir, se realizaron dos rondas del experimento.

Figura 13

Fotos de las pruebas de mitigación



Al concluir cada ronda, se les pidió a los integrantes del grupo 2 que mencionen en privado los dígitos que habían podido observar. En la Tabla 7 se encuentra el PIN ingresado por los usuarios en cada ronda y, entre corchetes, los dígitos que los atacantes señalaron haber descubierto. La presencia de un guion «-» indica que los atacantes no pudieron identificar el número.

Tabla 7

PIN brindado a cada usuario y dígitos descubiertos

Usuarios	Ronda	PIN asignado			
Usuario 1	1	3 [3]	6 [6]	2 [2]	8 [8]
	2	7 [7]	7 [7]	3 [3]	6 [6]
Usuario 2	1	4 [-]	3 [-]	1 [-]	6 [-]
	2	9 [-]	2 [-]	8 [-]	2 [-]
Usuario 3	1	5 [-]	3 [-]	2 [-]	1 [-]
	2	9 [-]	2 [-]	5 [-]	4 [-]
Usuario 4	1	0 [0]	1 [1]	7 [7]	8 [8]
	2	5 [5]	9 [9]	9 [9]	5 [5]
Usuario 5	1	9 [-]	6 [-]	6 [-]	4 [-]
	2	4 [-]	7 [-]	1 [-]	3 [-]
Usuario 6	1	2 [-]	4 [9]	5 [-]	6 [-]
	2	5 [-]	5 [-]	3 [-]	1 [-]
Usuario 7	1	3 [-]	7 [-]	4 [-]	8 [-]
	2	1 [-]	8 [-]	7 [-]	1 [-]
Usuario 8	1	8 [-]	8 [-]	0 [-]	8 [-]
	2	6 [-]	7 [-]	7 [-]	9 [-]

Como puede observarse, la mayoría de los atacantes no logró visualizar ninguno de los dígitos ingresados por el usuario que tenían enfrente e incluso cuando lograron observar uno (atacante 6, ronda 1), el número indicado no correspondía con el dígito ingresado. Estos resultados han sido comparados con los encontrados en la literatura actual en la sección de discusión, con el fin de determinar la efectividad del experimento.

5.5. Resultados de las pruebas de usabilidad

Para las pruebas de usabilidad se utilizaron los mismos grupos que en las pruebas de mitigación. Ahora, sin embargo, ambos grupos deberán interactuar con la interfaz e ingresar uno o varios PIN. El primer grupo, el de usuarios, ya había utilizado previamente la interfaz y ayudó brindando datos sobre cómo los usuarios más experimentados se sienten al usarla. Para acentuar esto, se realizaron tres rondas, en las que cada usuario debió ingresar un PIN diferente. Por el contrario, el grupo 2, el de atacantes, tuvo su primera interacción con la interfaz, lo cual sirvió para obtener datos sobre cómo los usuarios nuevos se sienten al usarla. Con el fin de medir cuantitativamente la usabilidad, se pidió a los integrantes de ambos grupos que llenaran una encuesta que contenía las diez preguntas del cuestionario SUS al terminar la prueba. En la Tabla 8 se muestra el promedio SUS de cada uno de los participantes, así como el promedio SUS de cada grupo.

Tabla 8

Promedio SUS de los participantes

Usuarios	Grupo	Promedio SUS	Promedio por grupo
Usuario 1	1	65	85
Usuario 2	1	85	
Usuario 3	1	90	
Usuario 4	1	97,5	
Usuario 5	1	85	
Usuario 6	1	97,5	
Usuario 7	1	77,5	
Usuario 8	1	82,5	
Usuario 9	2	75	71,875
Usuario 10	2	65	
Usuario 11	2	80	
Usuario 12	2	85	
Usuario 13	2	97,5	
Usuario 14	2	42,5	
Usuario 15	2	50	
Usuario 16	2	80	

6. DISCUSIÓN DE LOS RESULTADOS

La seguridad que brinda el algoritmo de generación de secuencias aleatorias es un punto clave de la investigación, por lo que se estableció como límite que el promedio de secuencias repetidas en 1500 usos de la interfaz sea menor a uno. Esto se determinó haciendo un análisis de los datos investigados respecto al uso de cajeros automáticos en el Perú. En primer lugar, según Statista (2023b), el número acumulado de usos de cajeros automáticos en el Perú durante el año 2019 fue de 495 millones. Como segundo dato importante, la Superintendencia de Banca y Seguros y AFP (2020) señala que el número de cajeros en el Perú para el 2019 era de 28 407. Al dividir estos números se obtiene un promedio de aproximadamente 17 425 usos de cajeros automáticos al año. Del mismo modo, para obtener el uso promedio mensual, se divide esta cifra entre 12, lo cual dio como resultado aproximadamente 1452 usos. Más adelante, en el apartado de resultados, se mostraron los datos recolectados al ejecutar el algoritmo en conjuntos de 100, 500, 1000, 1500, 2000 y 5000 repeticiones. Debido a que el promedio mensual de usos de un cajero es aproximadamente 1452, se propuso como objetivo obtener menos de una secuencia repetida en 1500 repeticiones del algoritmo. Al haber obtenido, en 1500 repeticiones, un promedio de repetición de 0.267 secuencias con una desviación estándar de 0.593, se evidencia que el algoritmo está cumpliendo con la finalidad requerida para su adecuado uso.

Pasando a las pruebas de mitigación, una vez analizados los datos se hace claro que a los atacantes se les dificulta en gran medida la tarea de identificar los dígitos ingresados por los usuarios, ya que, a excepción de dos casos, fallaron en la identificación de los dígitos. Indagando más profundamente en las dos excepciones, en la Tabla 6 se puede ver que ambos participantes eran adultos mayores, quienes, a pesar de entender que la prueba se estaba dando en un contexto de cajero automático, no utilizaron medidas de seguridad y permitieron que los atacantes observen libremente los dígitos que ingresaban. Se obtuvo que los atacantes lograron identificar en total el 25 % de los dígitos ingresados por el grupo de usuarios, una cifra menor a la señalada por Agarwal et al. (2011) en el resultado de su estudio utilizando un teclado aleatorio, puesto que dichos investigadores indicaron que su prueba de mitigación de riesgo de *shoulder surfing* obtuvo un rango entre 33 % y 66 % de dígitos descubiertos por los atacantes. Asimismo, Alsuhibany (2021) en su investigación realizó tres diferentes pruebas con ciertas variaciones en el método propuesto para la mitigación del riesgo de *shoulder surfing,* cada prueba difiere en la cantidad de texto ingresado previa y posteriormente al digitado de la contraseña del usuario para confundir al atacante. En el primer método, en el que el número de caracteres ingresados antes y después es igual, obtuvo entre un 17,6 % y 29,4 % de contraseñas descubiertas; en el segundo método, en el que la cantidad de dígitos es distinta, obtuvo entre 26,4 % y 44,1 % de contraseñas descubiertas; y en el último método, en el que usa un solo dígito previo y posterior, pero también una cantidad decidida por el usuario de caracteres extra, obtuvo entre 5,8 % y 8,8 % de contraseñas descubiertas por los atacantes. Tomando un promedio de sus resultados, se tiene que los atacantes obtuvieron una media de 22,02 % de todas las contraseñas utilizadas en el experimento, cifra similar al porcentaje de dígitos obtenidos por los atacantes en el presente artículo. Por otro lado, Roth et al. (2004) realizaron dos pruebas llamadas *Inmediate Oracle Choice (Inmediate OC)* y *Delayed Oracle Choice (Delayed OC)* para la mitigación de ataque de *shoulder surfing* en las que se reclutó 37 participantes de entre 20 y 30 años con educación académica. Debido a diversos motivos, cuatro participantes debieron ser excluidos, lo que dejó a la muestra con 33 participantes, a los cuales se les asignó aleatoriamente el método IOC o el método DOC y debieron completar 10 ciclos de ingreso de PIN. En la primera se logró identificar uno de los cuatro dígitos del PIN del usuario en el 8,75 % de los intentos, mientras que en 5 % de estos, se identificaron dos de los dígitos. Al usar *Delayed OC* los atacantes pudieron identificar una de las cifras del PIN en el 7,5 % de los intentos, mientras que en el 5 % de estos se pudieron reconocer 2 dígitos. Tanto con *Inmediate OC* como con *Delayed OC* no se lograron identificar tres o los cuatro dígitos del PIN en ningún intento. Finalmente, Still et al. (2018) en la investigación que realizaron sobre un método de selección de datos resistente al *shoulder surfing*, encontraron que el 30 % de los participantes que fungieron como atacantes fueron capaces de descubrir el PIN ingresado dados tres intentos.

Respecto a las pruebas de usabilidad, se obtuvo que el promedio SUS del grupo 1 es de 85 puntos, lo cual le da a la interfaz una calificación de A+ según los rangos SUS. Por otro lado, el promedio SUS del grupo 2 es de 71,875 puntos, ubicándose en el rango C+. Tal diferencia es una característica esperable de los resultados de la prueba, puesto que es común que los usuarios se sientan más cómodos con la interfaz conforme la utilicen más veces. No obstante, a pesar de la diferencia entre los promedios, ambos se encuentran sobre la puntuación media general, la cual es de 68 puntos.

7. CONCLUSIONES

En el presente trabajo se implementó una prueba de concepto de interfaz *touchless* en un teclado numérico aleatorio para mitigar el riesgo de *shoulder surfing*. En primer lugar, gracias a las pruebas realizadas al algoritmo encargado de la generación de las secuencias, se pudo corroborar que, en promedio, no se obtienen secuencias repetidas en 1500 ejecuciones, por lo que se considera este objetivo cumplido. En segundo lugar, los resultados de las pruebas de mitigación revelaron resultados alentadores, puesto que únicamente se identificaron el 25 % de los dígitos ingresados por los usuarios. Finalmente, las pruebas de usabilidad mostraron una diferencia esperable entre los grupos 1 y 2, los cuales se sitúan en los rangos A+ y C+ de la escala SUS respectivamente, ambos sobre la media general de 68 puntos. Asimismo, en promedio global, la interfaz se encuentra en el rango B+ con 78,4375 puntos, por lo cual este ámbito se considera cumplido también.

8. LIMITACIONES DE LA INVESTIGACIÓN

La principal limitación de la investigación es la aplicación de la prueba de mitigación en un entorno que no es exactamente igual al de un cajero automático. En este sentido, se utilizó un ambiente semicerrado y se dispuso a las personas en una fila para simular un entorno real y reducir la implicancia de la limitación. Asimismo, otra limitación se encuentra en que los integrantes del grupo 2 en las pruebas de mitigación (es decir, quienes fungieron de atacantes), no cuentan con experiencia en ataques de *shoulder surfing*. Se buscó minimizar la implicancia de esta limitación al explicarles claramente las funciones que debían realizar y se hicieron dos rondas en las pruebas, de forma que se les dé una ventana más amplia.

9. TRABAJOS FUTUROS

Como trabajos futuros, se plantea reducir en mayor medida las limitaciones al realizar nuevas pruebas de mitigación en ambientes más semejantes a los de un cajero automático, además de contar con un mayor número de participantes en el que la variedad de atacantes brinde resultados en un contexto más cercano al real. Del mismo modo, se explorarán nuevas formas de generación de números aleatorios para reducir la posible predictibilidad del algoritmo y dar mayor seguridad a la aleatoriedad del sistema.

REFERENCIAS

Abhishek, K., Verma Kumar, M., & Prasad Singh, M. (2019). Automated random colour keypad. *International Journal of Information and Communication Technology*, *15*(2), 162-175. https://doi.org/10.1504/IJICT.2019.10018383

Adithya, P., Aishwarya, S., Megalai, S., Priyadharshini, S., & Kurinjimalar, R. (2018). Security enhancement in automated teller machine. *Proceedings of 2017 International Conference on Intelligent Computing and Control, I2C2 2017*, *2018-January*. https://doi.org/10.1109/I2C2.2017.8321773

Agarwal, M., Mehra, M., Pawar, R., & Shah, D. (2011). Secure authentication using dynamic virtual keyboard layout. *International Conference and Workshop on Emerging Trends in Technology 2011, ICWET 2011 - Conference Proceedings*, *Icwet*, 288-291. https://doi.org/10.1145/1980022.1980087

Ahmad, A. G. (2013). Arduino as a learning tool. *Sensing Technologies for Global Health, Military Medicine, and Environmental Monitoring III*, *8723*, 872313.

Alsuhibany, S. A. (2021). A Camouflage Text-Based Password Approach for Mobile Devices against Shoulder-Surfing Attack. *Security and Communication Networks*. https://doi.org/10.1155/2021/6653076

Borsci, S., Federici, S., Bacci, S., Gnaldi, M., & Bartolucci, F. (2015). Assessing user satisfaction in the era of user experience: Comparison of the SUS, UMUX, and UMUX-LITE as a function of product experience. *International Journal of Human-Computer Interaction*, *31*(8), 484-495. https://doi.org/10.1080/10447318.2015.1064648

Bultel, X., Dreier, J., Giraud, M., Izaute, M., Kheyrkhah, T., Lafourcade, P., Lakhzoum, D., Marlin, V., & Motá, L. (2018). Security analysis and psychological study of authentication methods with PIN codes. *Proceedings - International Conference on Research Challenges in Information Science*, *2018-May*, 1-11. https://doi.org/10.1109/RCIS.2018.8406648

Chakraborty, T., Nasim, M., Bin Malek, S. M., Sami, M. T. H. M., Saeef, M. S., & Al Islam, A. B. M. A. (2016). Sporshohin: A tale of devising visible light based low-cost robust touchless input device. *Proceedings of the 7th Annual Symposium on Computing for Development, ACM DEV-7 2016*. https://doi.org/10.1145/3001913.3001914

Edem Udo Udo, E., Abiso Kabir, A., Yusuff, A. M., & Bukola Simeon, A. (2017). Impact of automated teller machine on customer satisfaction and profitability of commercial banks. *IIARD International Journal of Banking and Finance Research*, *3*(2). http://www.iiardpub.org

Ipsos. (2019, 14 de octubre). *Hay 400,000 que sufrieron algún tipo de robo o fraude financiero*. Ipsos. https://www.ipsos.com/sites/default/files/ct/publication/

documents/2019-10/hay_400000_que_sufrieron_algun_tipo_de_robo_o_fraude_financiero.pdf

Lewis, J. R. (2018). Measuring perceived usability: The CSUQ, SUS, and UMUX. *International Journal of Human-Computer Interaction*, *34*(12), 1148-1156. https://doi.org/10.1080/10447318.2017.1418805

Lewis, J. R., Utesch, B. S., & Maher, D. E. (2013). UMUX-LITE - When there's no time for the SUS. *Conference on Human Factors in Computing Systems - Proceedings*, October, 2099-2102. https://doi.org/10.1145/2470654.2481287

Maiti, A., Jadliwala, M., & Weber, C. (2017). Preventing shoulder surfing using randomized augmented reality keyboards. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 630-635. https://doi.org/10.1109/PERCOMW.2017.7917636

Montanaro, L., Sernani, P., Dragoni, A. F., & Calvaresi, D. (2016). A touchless human-machine interface for the control of an elevator. *CEUR Workshop Proceedings*, *1746*, 58-65.

Rajarajan, S., Maheswari, K., Hemapriya, R., & Sriharilakshmi, S. (2014). Shoulder surfing resistant virtual keyboard for internet banking. *World Applied Sciences Journal*, *31*(7), 1297-1304. https://doi.org/10.5829/idosi.wasj.2014.31.07.378

Roth, V., Richter, K., & Freidinger, R. (2004). A PIN-entry method resilient against shoulder surfing. *Proceedings of the ACM Conference on Computer and Communications Security*, 236-245. https://doi.org/10.1145/1030083.1030116

Sevilla-Gonzalez, M. D. R., Moreno Loaeza, L., Lazaro-Carrera, L. S., Bourguet Ramirez, B., Vázquez Rodríguez, A., Peralta-Pedrero, M. L., & Almeda-Valdes, P. (2020). Spanish version of the system usability scale for the assessment of electronic tools: Development and validation. *JMIR Human Factors*, *7*(4), e21161. https://doi.org/10.2196/21161

Shukla, S., Helonde, A., Raut, S., Salode, S., & Zade, J. (2018). Random keypad and face recognition authentication mechanism. *IRJET*, *5*(3), 3685–3688.

Statista. (2023a). Number of automated teller machines (ATMs) per 100,000 adults in Peru from 2005 to 2021. https://www.statista.com/statistics/1079224/peru-automated-teller-machines-atm-penetration/

Statista. (2023b). Number of ATM transactions in selected countries in Latin America in 2019. https://www.statista.com/statistics/823923/number-atm-transactions-latin-america-country/

Still, J. D., & Bell, J. (2018). Incognito: Shoulder-surfing resistant selection method. *Journal of Information Security and Applications*, *40*, 1-8. https://doi.org/10.1016/j.jisa.2018.02.006

Superintendencia de Banca y Seguros y AFP. (2020). *Perú: indicadores de inclusión financiera de los sistemas financieros, de seguros y de pensiones - junio 2020*. https://intranet2.sbs.gob.pe/estadistica/financiera/2020/Junio/CIIF-0001-jn2020.PDF

Toledo Concha, E., & León Reyes, V. (2023). Financial inclusion in Peru: Appraisal and perspectives. *Quipukamayoc*, *31*(65), 73-84. https://doi.org/10.15381/quipu.v31i65.25882

Yu, M., Zhuge, J., Cao, M., Shi, Z., & Jiang, L. (2020). A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices. *Future Internet*, *12*(2), 1-23. https://doi.org/10.3390/fi12020027