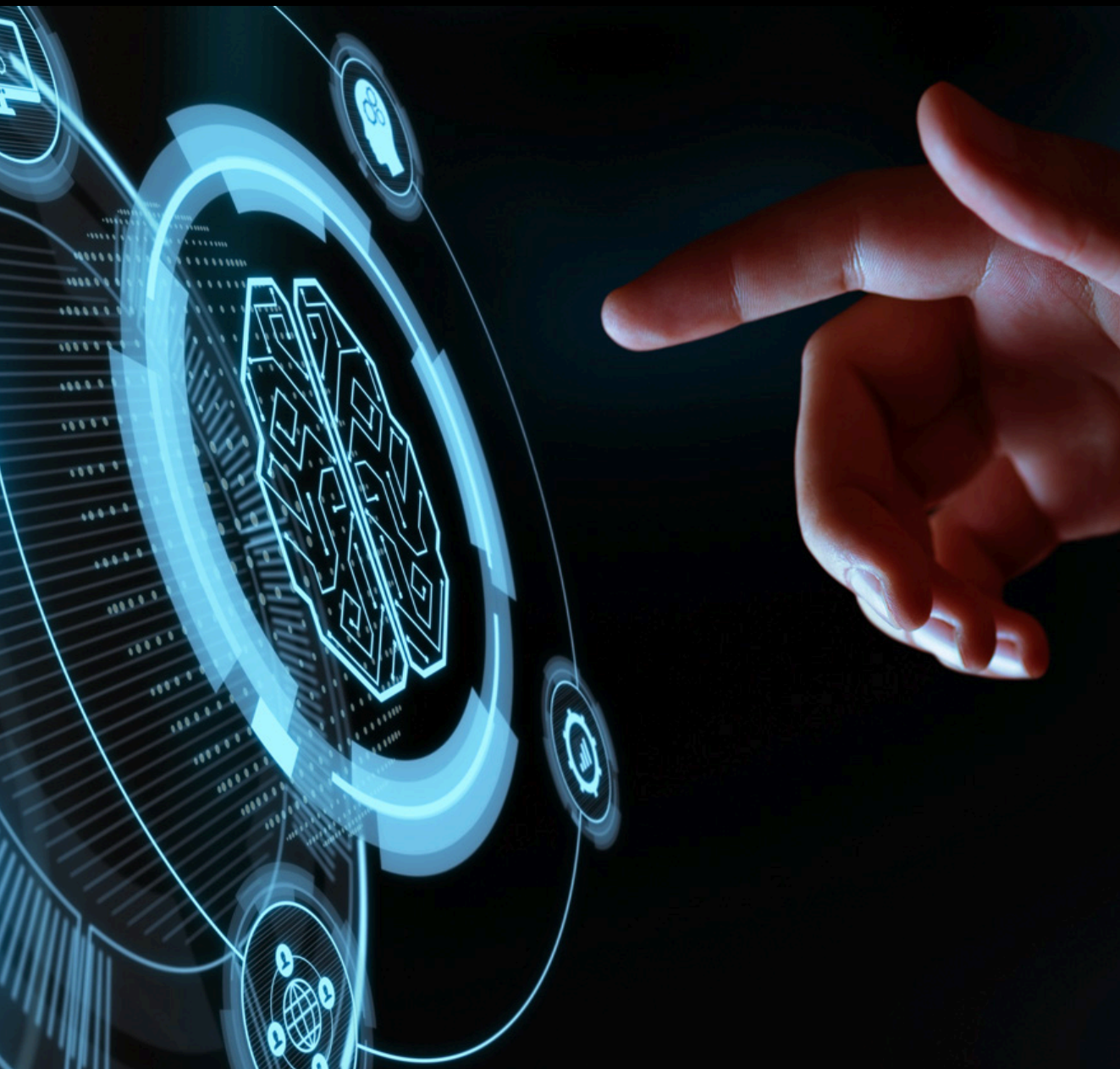
13 FONDO EDITORIAL Revista de la Carrera de Ingeniería de Sistemas Diciembre 2020

UNIVERSIDAD DE LIMA

# INTERFASES



# Revista digital de la Carrera de Ingeniería de Sistemas de la Universidad de Lima

Edición n.º 13 // Enero-diciembre 2020 // ISSN 1993-4912

## Objetivo de *Interfases*

*Interfases* es un espacio virtual, gestionado por la Universidad de Lima, para la publicación de investigaciones originales en áreas temáticas relacionadas con las ciencias de la computación, gestión de datos, aprendizaje automático, sistemas de información y áreas afines. Se publican artículos científicos y avances de investigación, siempre que cumplan con el proceso de revisión por pares. La revista *Interfases* está indizada en CrossRef, Dialnet, Latindex y DOAJ: Directory of Open Access Journals, y se publica una vez al año, en diciembre.

## Público objetivo

- Profesionales e investigadores relacionados con la ingeniería de sistemas y afines.
- Estudiantes y docentes del pregrado y posgrado de ingeniería de sistemas.
- Público interesado.

ISSN: 1993-4912 (electrónico)
doi: https://doi.org/10.26439/interfases2020.n013
Hecho el depósito legal en la Biblioteca Nacional del Perú n.° 2020-09967

Periodicidad: Anual
Edición: n.° 13
Fecha publicación: Diciembre 2020
Arbitraje editorial: Revisión por pares de doble ciego
Directorios y catálogos:
CrossRef, Dialnet, Latindex y DOAJ



Universidad de Lima

Av. Javier Prado Este 4600, Urbanización Monterrico Chico, Lima 33, Perú.
Apartado postal 852, Lima 100, Perú.
Teléfono: (511) 437-6767, anexo 30131
fondoeditorial@ulima.edu.pe
http://www.ulima.edu.pe/

Edición, diseño y diagramación:
Fondo Editorial
Imagen de portada:
Alexander Supertramp/Shutterstock.com

Carrera de Ingeniería de Sistemas
http://www.ulima.edu.pe/pregrado/ingenieria-de-sistemas

Contacto: interfases@ulima.edu.pe

# POLÍTICA EDITORIAL

## ENFOQUE Y ALCANCE

*Interfases* es un espacio virtual, gestionado por la Universidad de Lima, para la publicación de investigaciones originales en áreas temáticas relacionadas con las ciencias de la computación, gestión de datos, aprendizaje automático, sistemas de información y áreas afines. Se publican artículos científicos y avances de investigación, siempre que cumplan con el proceso de revisión por pares. La revista *Interfases* está indizada en CrossRef, Dialnet, Latindex y DOAJ: Directory of Open Access Journals, y se publica una vez al año, en diciembre.

## PROCESO DE REVISIÓN POR PARES

Los manuscritos originales e inéditos enviados a la revista *Interfases* siguen un proceso de evaluación en dos etapas.

En la primera, el editor examina el contenido para determinar si el manuscrito está alineado con el alcance y ha seguido las directrices para autores. Si el manuscrito no es aceptado, se devuelve al autor correspondiente con las razones detalladas que motivan la decisión adoptada por el editor.

Si el manuscrito es aceptado por el editor, este se envía a revisores externos expertos en el tema de investigación. Esta segunda evaluación corresponde a una revisión por pares doble ciego, donde el autor y revisores son anónimos.

El revisor evalúa el contenido del manuscrito y, basándose en su experiencia y conocimiento, adopta una de las siguientes recomendaciones:

1. El manuscrito es aceptado sin cambios o con cambios mínimos.
2. El manuscrito se acepta, a condición de realizar cambios importantes, de acuerdo con las observaciones del revisor. La versión corregida del manuscrito debe ser aprobada en una segunda revisión.
3. El manuscrito no se acepta por las contribuciones limitadas del estudio u otras consideraciones informadas por el revisor.

Con base en los comentarios de los revisores, el editor informa la decisión al autor correspondiente, quien tiene hasta 30 días para realizar los cambios al manuscrito (recomendación 1 y 2) o argumentar por qué no se acepta (recomendación 3).

Una vez que los revisores reciben el manuscrito corregido, tienen hasta 20 días para informar el resultado de la nueva evaluación; posteriormente, emiten su recomendación final. Una vez que el editor recibe la segunda ronda de revisiones, toma una decisión para publicar el manuscrito y luego se le notifica al autor correspondiente.

Cualquier objeción del autor respecto de la decisión del editor o hacia los comentarios de los revisores será resuelta por el Comité Editorial como instancia final.

La revista se adhiere a los criterios establecidos por el Guidelines on Good Publication Practice del Committee on Publication Ethics (COPE), el cual establece las sanciones en caso de plagio.

# DIRECTRICES PARA AUTORES/AS

## ENVÍO DEL MANUSCRITO

*Interfases* publica tres tipos de artículos: trabajos de investigación (hasta 5000 palabras), avances en investigación (hasta 2800 palabras) y revisiones (hasta 1500 palabras).

Todos los envíos se envían del mismo modo. Una vez que el editor verifique que el contenido del manuscrito pertenece al ámbito de *Interfases* lo remitirá a un proceso de revisión por pares. Este proceso (compuesto de dos rondas) toma aproximadamente 2-3 meses, pero dependiendo de la complejidad del manuscrito, podría extenderse.

Los manuscritos enviados a *Interfases* no deben haberse publicado previamente ni estar en consideración para su publicación en otra revista.

### Página del título

- La página del título debe incluir:
- Un título conciso e informativo (hasta 30 palabras).
- El nombre completo de cada autor, incluyendo la afiliación institucional, la dirección de correo electrónico y el código ORCiD.
- Resumen de 200-250 palabras. El resumen debe indicar la naturaleza y contribución del estudio. Evite las abreviaturas no definidas, las ecuaciones matemáticas o las referencias bibliográficas en el texto del resumen.
- Palabras clave (3-5) separadas por comas. Las palabras clave deben tomarse de la taxonomía de la IEEE Computer Society: https://www.computer.org/digital-library/journals/sc/tsc-taxonomy-list

### Texto

Los trabajos enviados deben haber sido redactados en un documento Word (.doc o .docx), y aquellos aceptados para ser publicados deben usar la plantilla de Interfases LATEX que estará disponible pronto.

Al redactar el manuscrito, usar la opción de numeración automática para enumerar las páginas. Por favor, evite el uso de funciones de campo. Utilice la función de tabla, no

una hoja de cálculo pegada, para hacer tablas. Si escribe su manuscrito con Word, use el editor de ecuaciones o MathType para las ecuaciones.

**Tablas**

Las tablas son el núcleo de los nuevos hallazgos reportados en la corriente principal de la ciencia, por lo tanto, incluya las tablas que considera son estrictamente necesarias. Todas las tablas se enumeran utilizando números arábigos (por ejemplo, Tabla 1, Tabla 2, ...) e incluyen un título que detalla la relevancia de los datos presentados.

Las tablas se mencionan en el orden en que aparecen en el manuscrito. Además del número, el título y los datos, las tablas pueden incluir una nota para detallar la fuente de información, así como explicaciones adicionales que no están incluidas en el manuscrito.

**Abreviaturas**

Use abreviaturas solo si son necesarias para mejorar la legibilidad de su documento. Debe definir cada abreviatura en la primera mención después de usarla de manera consistente.

**Conclusiones**

Recuerde que las conclusiones no son la versión narrativa y textual de las tablas incluidas en la sección Resultados. Por el contrario, las conclusiones reseñan y sintetizan los principales argumentos del artículo. Las conclusiones se extraen de los hallazgos y proporcionan una respuesta adecuada a la pregunta de investigación. Además, las conclusiones incluyen las limitaciones del estudio y sugieren nuevas preguntas y aplicaciones para futuros estudios.

**Referencias**

Las citas y las referencias deberán indicarse de acuerdo con las normas APA. Según la norma señalada, las referencias, enlistadas al final de la publicación, se realizarán de la siguiente forma:

a) Libros:

   Apellido del (los) autor(es), letra inicial del nombre del (los) autor(es). (Año de la publicación). *Título del libro* (en cursiva), (número de la edición). Lugar de publicación: Nombre de la editorial.

b) Artículos de revistas o capítulos de un libro:

   Apellido del (los) autor(es), letra inicial del nombre del (los) autor(es). (Año de publicación). Título del artículo o el capítulo. *Nombre de la revista o el libro* (en

cursiva), *número de la revista* (en cursiva), páginas en las que se encuentra el artículo o el capítulo.

c) Libros electrónicos:

Apellido del (los) autor(es), letra inicial del nombre del (los) autor(es). (Año de publicación). *Título del texto electrónico* (en cursiva). Recuperado de http://... (dirección web).

d) Artículos de revistas electrónicas:

Apellido del (los) autor(es), letra inicial del nombre del (los) autor(es). (Año de publicación). *Título del artículo. Nombre de la revista* (en cursiva), páginas en las que se encuentra el artículo. Recuperado de http://... (dirección web).

e) Ponencias en congresos o simposios:

Apellido del (los) expositor(es), letra inicial del nombre del (los) autor(es). (Año, X [indicar día] de XXX [indicar mes]). *Título de la ponencia* (en cursiva). Conferencia presentada en el XXXXXX [nombre del evento]. Recuperado de http://... (dirección web).

**Material suplementario electrónico**

Los autores pueden incluir archivos de texto (incluyendo tablas y figuras) y hojas de cálculo como material complementario. Sin embargo, para datos de investigación, es recomendable archivarlos en repositorios de datos. Para el código de *software*, los autores pueden usar plataformas como GitHub o similares.

Si los originales contienen fotografías o reproducciones de obras pictóricas, estas se entregarán aparte en archivos TIFF o JPG, con 300 píxeles de resolución (dpi). Si contienen gráficos, cuadros, dibujos, flujogramas u otros elementos, estos deben entregarse igualmente en un archivo aparte y en el programa original en que fueron creados (por ejemplo: Excel, Illustrator, etcétera).

**Lista preliminar para la preparación de envíos**

Los artículos deberán respetar el siguiente formato:

a) Página A4.

b) Título del artículo, centrado en negrita, con letra Times New Roman de doce puntos.

c) Títulos del texto, centrados en negrita, con letra Times New Roman de doce puntos, dejando dos líneas en blanco antes del párrafo.

d) Texto del cuerpo con letra Times New Roman de doce puntos, con espacio y medio de interlineado.

**Declaración de privacidad**

Los nombres y las direcciones de correo electrónico introducidos en esta revista se usarán exclusivamente para los fines establecidos en ella y no se proporcionarán a terceros o para su uso con otros fines.

# ÍNDICE

# PRESENTACIÓN

Dr. Julio Alejandro Padilla-Solís
Director de la revista *Interfases*

Nos complace presentar una nueva edición de la revista *Interfases* en este año 2020, marcado por el impacto que la pandemia del COVID-19 ha impuesto en nuestras formas habituales de convivencia e interacción social. Sin embargo, si existe un término que mejor puede caracterizar este 2020 es el aprendizaje. Hemos aprendido a reinventarnos, a adaptarnos a nuevas formas de vida que nos permiten reconstruir nuestros esquemas de desarrollo social y científico.

Los nuevos retos mundiales exigen a su vez soluciones creativas e innovadoras para superarlos. Es en este escenario que las tecnologías de la información han desempeñado un rol preponderante como una plataforma sobre la que se han llevado a cabo todos los procesos productivos, económicos y educativos en el 2020. Este hecho hizo más evidente que el impulso a la investigación y la innovación en tecnologías no es una opción sino un punto obligatorio en la agenda hacia un desarrollo sostenible. En ese contexto, y firmes en nuestro compromiso de contribuir al desarrollo científico, nos complace presentar un nuevo número de *Interfases* que en su treceava edición ha reunido aportes de investigadores cuyos trabajos se han enmarcado en las áreas de seguridad de la información, inteligencia artificial, simulación de sistemas, telecomunicaciones y gestión de las tecnologías de la información.

Es particularmente enriquecedora la experiencia adquirida en esta edición de *Interfases* no solo por haber cubierto distintas líneas de investigación, sino también por recopilar contribuciones de jóvenes y experimentados investigadores que han participado de la convocatoria de trabajos.

La sección de artículos de esta edición comienza con el aporte de los autores Joel Fernando Palomino Masco, Juan Antonio Paco Fernández y Michel Anyelo Zarzosa Rojas, quienes nos presentan una propuesta de solución de bajo consumo energético para la adquisición de datos de temperatura. El trabajo comprende el diseño y desarrollo de la solución respaldado en el protocolo I2C, así como la puesta en marcha de la solución localidades altoandinas del Perú, ubicadas entre los cuatro mil y cinnco mil metros sobre el nivel del mar. Es importante resaltar el esfuerzo por llevar a cabo esta investigación tanto por su impacto social respaldado por un proyecto del Gobierno del Perú como por su bajo costo, pues responde de forma directa a demandas de los sectores sociales menos favorecidos respecto del acondicionamiento de sus viviendas para mitigar el impacto de las bajas temperaturas a las que están expuestos.

La selección de artículos continúa con el aporte de Rubén Ahomed, quien comparte con los lectores un análisis sobre las barreras de la transformación digital relacionadas con el rendimiento financiero en las organizaciones. La transformación digital ha sido impulsada en los últimos años en organizaciones de toda escala con el fin de incorporar capacidades digitales a los procesos productivos y sus resultados en busca de una mayor eficiencia. El análisis efectuado en este artículo presenta al lector nueve barreras a la innovación y cinco al rendimiento financiero que explican de qué manera el crecimiento en términos de digitalización, así como el beneficio financiero esperado se ven influenciados, entre otros factores, por el sector industrial o área geográfica en la que una empresa ejerce su actividad. La definición de indicadores adecuados es además necesaria, pero a la vez compleja, en el contexto empresarial actual.

El tercer artículo presenta una interesante propuesta a cargo de Lennin Paul Quiroz Villalobos, quien nos presenta una plataforma de simulación para el desarrollo de un sistema de piloto automático para aeronaves de ala fija. La orientación de esta investigación hacia el sector industrial contribuye claramente en la consecución de dos objetivos que buscan las empresas en su carrera hacia la digitalización: la reducción de costos y del ciclo de desarrollo de nuevos productos. La plataforma Software in the Loop (SITL), presentada en este artículo, describe, así, los principios de diseño, proceso de construcción y validación de este simulador en la representación de la dinámica de vuelo en una aeronave ligera. SITL incorpora funcionalidades específicas para gestionar los procesos de control en un sistema de vuelo, así como el cálculo de distancias y opciones de geolocalización necesarias para caracterizar el comportamiento de un avión en vuelo.

El cuarto artículo de esta edición nos presenta una contribución en el área de la seguridad de la información a cargo de William-Rogelio Marchand-Niño y Edwin Jesús Vega Ventocilla. Los autores plantean un modelo de medición de los controles críticos de seguridad definidos por el Centro para la Seguridad de Internet (CIS por sus siglas en inglés) integrados en un Cuadro de Mando Integral (CMI) capaz de articular la visión de indicadores de nivel técnico u operativo con el nivel de gestión. Los autores plantean un enfoque

novedoso bajo el cual se definen diez dominios de controles de seguridad susceptibles de ser automatizados, los que son posteriormente integrados en los distintos cuadrantes del Cuadro de Mando integral propuesto con resultados prometedores. Esta investigación contribuye, así, a mejorar la gestión de la seguridad de la información, siendo esta una condición clave en las organizaciones actuales para asegurar la confidencialidad, integridad y disponibilidad de los datos.

La selección de artículos finaliza con el trabajo presentado por Andres Eduardo Moncada Vargas, cuya investigación está enfocada en la detección de páginas web de *phishing* asistida por métodos de aprendizaje automático. En este trabajo se destacan avances importantes alcanzados por la comunidad científica orientados a detectar y mitigar una de las amenazas más prevalentes de los últimos años: el *phishing*. Estos ataques consisten en intentos fraudulentos para robar información datos de un usuario (credenciales de acceso, cuentas bancarias, entre otras) y se han incrementado significativamente desde el inicio de la pandemia. En este artículo se analiza la efectividad de los mejores métodos de detección bajo un modelo de evaluación estructurado y objetivo que refleja las fortalezas y debilidades de cada método en cada uno de los casos de uso analizados. El comportamiento estudiado refleja también que estas soluciones requieren de una revisión y mejora constante, pues han de evolucionar en respuesta al dinamismo de sus adversarios.

Hacemos expreso nuestro agradecimiento a todos los investigadores que remitieron sus manuscritos para consideración en la presente edición de *Interfases.* Somos conscientes del esfuerzo que para cada uno ha significado el dedicar el tiempo necesario para producir resultados de investigación, particularmente en este 2020. Los aportes de nuestros autores nos dejan valiosas lecciones aprendidas junto con interesantes líneas de trabajo futuro. Agradecemos, del mismo modo, a los revisores de *Interfases*, quienes se encargan de asegurar la calidad de nuestra revista a través de un proceso de evaluación por pares conforme a las mejores prácticas de investigación.

Finalmente, nos complace poner a disposición de nuestros lectores esta decimotercera edición de *Interfases*, fruto de un esfuerzo colectivo aún más significativo en un año afectado por las circunstancias adversas que nos ha tocado vivir. Lo anterior no ha hecho más que renovar nuestro optimismo para iniciar un 2021 con las máximas expectativas por cumplir nuevos objetivos para nuestra revista. Asimismo, renovamos nuestro compromiso de realizar un trabajo de excelencia académica que cumpla con los estándares de calidad que nuestros lectores esperan en cada edición de *Interfases*.

# ARTÍCULOS DE INVESTIGACIÓN

Recibido: 27/8/2020 / Aceptado: 8/11/2020
doi: https://doi.org/10.26439/interfases2020.n013.4832

# Uso no estándar e implementación exitosa del protocolo I2C para un sistema de medición de temperatura en aldeas andinas a gran altitud

Joel Fernando Palomino Masco
j.palomino@pucp.edu.pe

Juan Antonio Paco Fernández
jpaco@pucp.edu.pe

Michel Anyelo Zarzosa Rojas
michel.zaro@gmail.com

Grupo de Telecomunicaciones Rurales de la Pontificia Universidad Católica del Perú

## Resumen

Este artículo describe el diseño y desarrollo de un sistema de adquisición de datos de temperatura compacto y de bajo consumo energético, el cual emplea el protocolo I2C con cables de hasta ocho metros de largo. Este sistema se ha utilizado en la implementación de un proyecto del Gobierno del Perú con el objetivo de validar el uso de muros Trombe en localidades ubicadas a gran altitud (4000-5000 m s. n. m.). En este artículo se explica la construcción del módulo, el subsistema de adquisición de datos y se ofrece una visión general de la implementación del sistema desarrollado.

PALABRAS CLAVE: protocolo I2C / sensores / temperatura / adquisición de datos

## Abstract

### NON-STANDARD USE AND SUCCESSFUL IMPLEMENTATION OF I2C PROTOCOL FOR A TEMPERATURE MEASUREMENT SYSTEM IN HIGH ALTITUDE ANDEAN SMALL TOWNS

This article describes the design and development of a compact low energy consumption temperature data acquisition system using the I2C protocol with cables up to eight meters long. This system has been used in the implementation of a Peruvian government project aimed at validating the use of Trombe walls in small towns located at high altitude (4000 - 5000 m.a.s.l.). This article explains the construction of the module, the data acquisition subsystem, and presents an overview of the implementation of the developed system.

KEYWORDS: I2C protocol / sensors / temperature / data acquisition

## 1. INTRODUCCIÓN

En zonas rurales de los Andes peruanos (por encima de los 4000 m s. n. m.) donde hay falta de energía eléctrica y las bajas temperaturas congelan el agua, el Gobierno, a través del Fondo de Cooperación para el Desarrollo Social (Foncodes), en su tarea de ayudar a las poblaciones vulnerables que viven en condiciones de pobreza o pobreza extrema, ha desarrollado el proyecto Mi Abrigo (Foncodes, 2017). Este proyecto tiene como objetivo "calentar" las casas mediante el uso de muros Trombe (Ana, Anabela e Ivo, 2016; National Renewable Energy Laboratory, 2017). Esta tecnología consiste en una pared orientada al sol, preferiblemente hacia el norte en el hemisferio sur y hacia el sur en el hemisferio norte, que va absorbiendo el calor del día y lo entrega lentamente a la vivienda durante la noche. El muro está construido con materiales comunes como piedras, hormigón, arcilla y agua, los cuales se encapsulan con vidrio o incluso plástico para formar un colector de energía solar.

Durante la primera etapa de ese proyecto se seleccionaron algunas casas para medir el impacto social de la instalación de esa solución. Una vez instalados los muros Trombe, se requería un sistema de medición de temperatura para verificar si estos muros, efectivamente, aumentaron la temperatura de las casas respecto a su estado anterior (figura 1). Por esta razón, el Gobierno solicitó el desarrollo de un módulo de adquisición de datos de temperatura.

Según lo anterior, el sistema de medición desarrollado tiene como objetivo registrar el incremento de temperatura al interior de las viviendas y compararlo con la temperatura externa durante el periodo crítico del año (temperaturas más bajas); la expectativa del Gobierno era alcanzar 10 °C de diferencia en el momento más frío, teniendo en consideración que la diferencia habitual es de menos de 3 °C en horas de la noche. Técnicamente, el sistema se basa en el uso del protocolo I2C en cables eléctricos de hasta ocho metros de longitud para lo cual fue necesario diseñar la electrónica adecuada para asegurar, bajo consumo energético, un correcto funcionamiento en condiciones extremas de altitud y la adecuada gestión de tres sensores de temperatura en un solo bus.

En ese sentido, el presente artículo describe, en la segunda sección, tanto los requisitos del sistema y el protocolo de comunicación usado como las características del *datalogger* y la placa de propósito específico (*shield*) diseñadas como parte del sistema. En la tercera sección se mencionan, brevemente, algunos aspectos relacionados con la instalación de los sistemas mientras que en la cuarta y última sección se detallan las conclusiones obtenidas.

*Figura 1.* Muro Trombe instalado

Fuente: foto tomada durante actividades de instalación (archivo fotográfico del autor)

## 2. SISTEMA DE MEDICIÓN DE TEMPERATURA

En atención a los plazos establecidos por Foncodes, el prototipo del sistema de medición se desarrolló utilizando las placas y sensores disponibles en el mercado local al momento del diseño. Anteriormente, nuestro grupo de investigación había elaborado un sistema de medición con el sensor LM35; al tener una salida analógica no es fácil transportarla por cables sin realizar una digitalización que evite perder información. Una termocupla es un sensor con mucho potencial, pero su costo elevado y dimensiones no eran los adecuados para nuestro proyecto. Por esta razón, el sensor de temperatura debe ser de tamaño reducido, con salida digital y de bajo costo.

Para este prototipo se utilizó la placa Arduino Due con el sensor de temperatura TMP112 de Texas Instruments. Sobre este primer desarrollo se realizaron algunas optimizaciones para obtener la versión a instalarse en el marco del proyecto.

Según lo dispuesto por el Gobierno, el esfuerzo principal se orientó en obtener un dispositivo funcional y eficiente usando los componentes disponibles en el mercado local; no fue posible, por ejemplo, optimizar el sistema mediante el diseño de una placa hecha a medida de muy bajo consumo energético.

### 2.1 Requisitos

El sistema de adquisición debía utilizar tres sensores de temperatura con una precisión de ± 0,5 °C, para registrar las variaciones de temperatura cada cinco minutos; además, tenía que trabajar de manera autónoma por un período de tres meses como mínimo. Los sensores debían instalarse en tres ubicaciones predeterminadas: el primero a 1,8 metros de altura, que representa la temperatura que siente la persona mientras realiza sus actividades diarias; el segundo debería ubicarse a 0,5 metros, correspondiente a la altura en la cual la persona está durmiendo y el último sensor estaría ubicado en el exterior de la casa para comparar las lecturas de temperatura.

Según la información recibida, la distancia entre los sensores y la caja de control podría ser de hasta ocho metros. En este proyecto, uno de los mayores desafíos fue desarrollar un sistema de medición confiable con sensores conectados por un cable de ocho metros de largo, al contrario de la mayoría de las aplicaciones que usan sensores con cables cortos o soldados en la misma placa con el microprocesador. El principal inconveniente de esta distribución física es que el voltaje no es estable cuando se transmite información con sensores analógicos. Una solución para este problema es realizar calibraciones individuales por cada juego de sensores y cable.

### 2.2 Protocolo de comunicaciones

Muchos sensores digitales poseen un puerto de comunicación serial para transmitir los datos de manera confiable; con ello se evita la pérdida de información por caída de voltaje. Para los microprocesadores ATMEL, los protocolos seriales disponibles son los siguientes: UART, SPI, I2C y One Wire. Los dos primeros protocolos se utilizan para la comunicación punto a punto. En el caso de SPI, si bien se podría usar en comunicaciones punto-multipunto bajo configuración maestro-esclavo. Para lograr esto se debía agregar un dispositivo que conmute la línea entre cada sensor con el maestro, como un *buffer* tri-estado. Esta alternativa es un poco más complicada pues requiere usar una línea de selección para cada esclavo (sensor) y el requerimiento pasaba por usar un único cable. Lo mismo ocurre en el caso del protocolo UART, se debe tener un cable por cada sensor. Por otro lado, los protocolos seriales I2C y One Wire están diseñados para comunicar varios dispositivos a través de un único cable.

Para el diseño se requería sensores de temperatura que poseyeran un protocolo serial y con la capacidad para medir temperaturas negativas, por ejemplo, el *datalogger* de temperatura de LabJack (LabJack Measurement & Automation, 2017; Sousa, 2017). Se encontraron dos sensores: DS18B20 de MAXIM y TMP112 de Texas Instruments, cada uno con una precisión de 0,5 °C, y capaces de medir temperaturas de -10 °C. Luego de un análisis, se eligió el TMP112, no solo por sus características sino también por su disponibilidad en el mercado local. El sensor DS18B20 tuvo que ser descartado porque no había

suficientes unidades disponibles para venta (se requerían al menos 160 sensores) y el tiempo de espera para la importación era extenso.

El sensor TMP112 utiliza el protocolo I2C para transmitir información al microprocesador; este funciona con un suministro de 3,3 V, posee una resolución de 12 bits y una precisión de 0,5 °C. Su rango de medición va desde -40 °C a +125 °C. El protocolo I2C típicamente se emplea para transmitir información en cables cortos en la misma PCB; sin embargo, para el propósito previsto los sensores tenían que conectarse con un cable de hasta ocho metros de largo, pues esta era la distancia máxima de separación prevista para la instalación. El uso de integrados para extender el bus I2C como el P82B715 es una buena opción para este problema; sin embargo, como este sistema debía trabajar el mayor tiempo posible consumiendo la menor cantidad de energía posible, se optó por otra solución. De acuerdo con lo indicado por Truchsess (2010) sobre cómo alcanzar largas distancias con este protocolo en serie, se decidió, como solución, modificar el valor de las resistencias *pull-up* I2C, teniendo en cuenta que esta modificación implica un impacto despreciable en el consumo de corriente. De acuerdo con las ecuaciones (1) y (2), las cuales están descritas en el capítulo 7 de las especificaciones del protocolo I2C (NXP Semiconductors, 2014), las resistencias (máxima y mínima) son:

$$R_{p(MAX)} = \frac{t_r}{0.8473 * C_b} \quad (1)$$

$$R_{p(MIN)} = \frac{V_{DD} - V_{ol}}{I_{OL}} \quad (2)$$

Para el denominado Fast Mode, la frecuencia de la línea SCL está entre 10 KHz y 400 KHz, y la capacitancia del bus es la suma de la capacitancia del sensor TMP112 más la capacitancia del cable. Para el sensor de temperatura, de acuerdo con la hoja de datos (Texas Instruments, 2018), la capacitancia del pin digital es de 3 pF; el tiempo de subida es de 1 ms para una velocidad inferior a 100 KHz; el voltaje de suministro es de 3,3 voltios; el nivel de salida de voltaje es 0,4 voltios; la corriente de entrada es de 3 mA. Para el cable, la capacitancia es 33 pF / pie (General Cable/ Carol Brand, 2018); con una longitud de 8 metros la capacitancia del cable es 866 pF. Sumando ambos valores, la capacitancia total del bus, con 3 sensores, es 875 pF. Con estos valores, se define el nuevo rango como sigue:

$$\mathrm{R_{p(MAX)}} = \frac{1 * 10^{-6}}{0.8473 * 875 * 10^{-12}} = 1{,}348.8\Omega \quad (3)$$

$$\mathrm{R_{p(MIN)}} = \frac{3.3 - 0.4}{3 * 10^{-3}} = 966\Omega \quad (4)$$

Según las ecuaciones, la resistencia *pull-up* se estableció en un valor conveniente de 1 KΩ (ver resistencias R1 y R2 en figura 2). El cable usado es del tipo multifilar de cuatro hilos apantallado calibre 24. Adicionalmente, se tuvo en cuenta que en la mayoría de las placas Arduino el voltaje de funcionamiento del bus I2C está fijado en cinco voltios, mientras que el sensor de temperatura funciona a 3,3 voltios. Visto que las únicas placas que permiten este último nivel de voltaje son Arduino Due y Pro Mini, se eligió la placa Arduino Due, utilizando el segundo puerto I2C (SDA1, SCL1) que no tiene una resistencia *pull-up* fija. Debe indicarse que estas fórmulas son pertinentes debido a que la frecuencia usada es menor a 100 KHz.

### 2.3 *Datalogger*

Si bien los sensores son una parte fundamental del mismo sistema, por sí solos no pueden funcionar como un sistema de medición y adquisición de datos, por lo que es importante definir los requisitos necesarios para los demás componentes del *datalogger*:

- Reloj en tiempo real basado en el DS3231 de Maxim Integrated. Además de brindar la información de la fecha y hora de la medición, este dispositivo posee un pin de interrupción para informar al Arduino el momento en que debe medir.
- Tarjeta micro SD de 2 GB para almacenar un promedio de dos millones de lecturas de temperatura.
- Dos LED para mostrar el estado del sistema: uno que indica si el dispositivo está encendido ("DEVICE ON") y otro para indicar que está en operación ("DEVICE WORKING").
- Conector para unir el cable con los sensores.

Asimismo, se creó una librería especial que gestiona los componentes de la placa. Con ella el usuario podrá, además, establecer el nombre del *datalogger* de acuerdo con el lugar de instalación y elegir el tiempo de muestreo en minutos. Esta librería se implementó para trabajar con el entorno de Arduino IDE, y están implementadas en lenguaje C++. Se definieron las siguientes funciones para esta librería:

- Configurar las entradas y salidas digitales necesarias para cada periférico
- Leer la temperatura de cada sensor
- Almacenar las lecturas de temperatura en la tarjeta SD
- Obtener la marca de tiempo del reloj de tiempo real

El reloj de tiempo real también usa un puerto I2C para comunicarse. Si este dispositivo se hubiera conectado junto con el sensor de temperatura, la capacitancia del bus aumentaría y afectaría el valor de la resistencia *pull-up*. Además, el reloj DS3231 no

funciona bien en frecuencias inferiores a 100 KHz. Por esta razón, para evitar la pérdida de datos, el dispositivo se conectó al otro puerto I2C.

Para leer y escribir datos desde/hacia la tarjeta SD, el Arduino usa el puerto de SPI para realizar la acción. El *software* Arduino tiene una librería que permite administrar memorias de hasta 16 GB de memoria. Finalmente, los LED de estado se conectaron a dos pines digitales.

**2.4 Suministro de energía**

Para que el sistema funcione en forma autónoma, al menos durante tres meses, se debía implementar y optimizar el consumo de energía. Dado que la variación de temperatura es un proceso relativamente lento, no es necesario medir cada segundo o, en cambio, la medición se puede hacer después de algunos minutos. El Arduino Due ofrece dos modos de ahorro de energía: WAIT y BACKUP. Durante las pruebas, el modo WAIT no logró resultados positivos, mientras que el modo BACKUP sí logró los resultados esperados. En el modo BACKUP, el programa Arduino se reinicia al despertar, por lo que el programa tuvo que ajustarse para funcionar con esta condición. El pin de interrupción del DS3231 se usó para activar el sistema cada 1, 5, 10, 15 o 20 minutos, estas opciones se pueden elegir con la librería creada de acuerdo con los criterios del usuario. Esta interrupción se conectó al pin analógico A6 que corresponde a WKUP1 de acuerdo con la hoja de datos del SAM3X8E.

Para calcular la capacidad de batería requerida, el estado de funcionamiento del sistema se dividió en dos modos: activo e inactivo. Mientras está activo, el *datalogger* lee los datos del sensor de temperatura y los almacena en la tarjeta SD. Estas tareas tardan unos tres segundos en completarse y el consumo de energía es de 58 mA. En modo inactivo el *datalogger* entra en suspensión, deshabilitando todos los componentes y usando el modo BACKUP de ahorro de energía. En este modo, el consumo de energía cae a 13 mA. Para el cálculo de la capacidad de la batería requerida se calcula la cantidad de horas que está activo el dispositivo en un día y se multiplica por su consumo de corriente en amperios. Esto nos da la cantidad de AH por día. Por ejemplo, para el caso de tres muestras por horas, el dispositivo está activo por 0,06 horas y en modo inactivo 23,94 horas. El consumo es de 0,31 AH por día, esto se multiplica por la cantidad de días que va a estar prendido. Finalmente, el consumo obtenido se multiplica por 1,4 pues la batería se debe sobredimensionar con una capacidad adicional equivalente al 40 por ciento del total.

La tabla 1 resume la capacidad de batería requerida para una autonomía de 30, 60 y 90 días de acuerdo con el tiempo de muestreo del sistema.

Tabla 1
*Capacidad de batería en función a los días de autonomía*

| Muestras por hora | Capacidad requerida de batería (A-H) | | |
|---|---|---|---|
| | 30 días de autonomía | 60 días de autonomía | 90 días de autonomía |
| 60 | 15,69 | 31,37 | 47,06 |
| 12 | 13,83 | 27,67 | 41,50 |
| 6 | 13,60 | 27,21 | 40,81 |
| 4 | 13,53 | 27,05 | 40,58 |
| 3 | 13,49 | 26,97 | 40,46 |

Elaboración propia

El sistema propuesto debía trabajar por un periodo mínimo de tres meses. Debido a que el cambio de temperatura es un proceso lento, no es necesario tomar muestras muy seguido. Por ello se escogió un periodo de muestreo de doce muestras por hora con una autonomía de noventa días. La batería debía tener una capacidad mínima de 45 A-H. En el mercado se encontró la batería de plomo marca Ritar que trabaja con un voltaje de 12 voltios y una capacidad de 45 AH. La reducción de voltaje se realizó mediante un conversor DC que viene integrado en el mismo módulo Arduino.

### 2.5 Placa de propósito específico (*shield*)

Después de seleccionar y probar cuidadosamente cada componente del sistema de registro de datos, se diseñó y desarrolló una placa de propósito específico para integrar cada uno de estos componentes. Por otro lado, construir una PCB personalizada usando componentes integrados disponibles en el mercado local, permitió un ahorro significativo de espacio y brindó confiabilidad al sistema. Adicionalmente, esto permitió la simplificación del proceso de instalación, ya que el *datalogger* se reduce a una sola PCB, como se ve en las figuras 2 y 3. Las resistencias R1 y R2 son las resistencias *pull-up* calculadas en el apartado B.

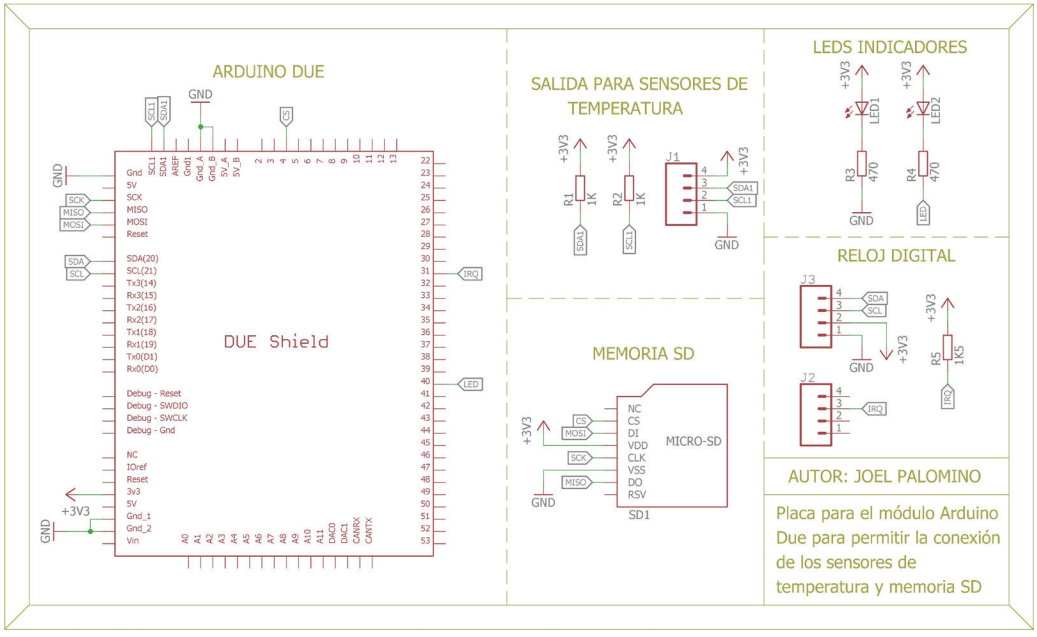



*Figura 2.* Diagrama esquemático de placa de propósito específico

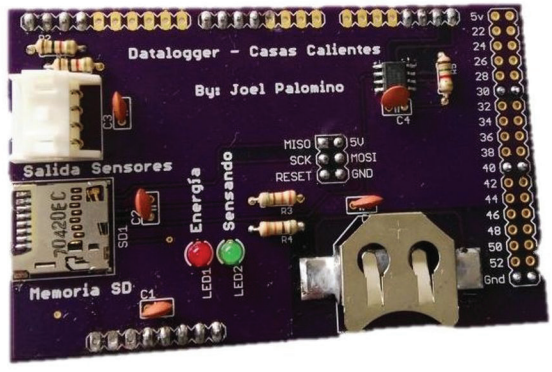



*Figura 3.* Placa de propósito específico

El *shield* fue diseñado para ubicarse encima del módulo Arduino Due, el cual tiene tres lados con borneras y solo tiene un borde libre para conexión USB y entrada de CC. En el diseño se requerían tres conectores para permitir la inserción de la tarjeta de memoria, el cable del sensor y el reloj de pila para el RTC. El espacio para montar estos conectores en el borde de la placa no era suficiente. Para resolver este problema, se suprimieron los conectores de los pines analógicos en la placa Arduino Due con el fin de permitir el montaje de la pila RTC.

Según se ha indicado, además del *shield*, también se desarrollaron tres placas para los sensores de temperatura (figuras 4 y 5). Estas placas fueron diseñadas para ser pequeñas y facilitar su instalación, pero también con el espacio mínimo indispensable para permitir la colocación de los conectores y dejar al sensor en el borde de la placa. Cada una de las placas desarrolladas fue rotulada para evitar confusiones en el proceso de instalación: fuera de la casa (EXTERIOR), dentro de la casa a 0,5 metros (INTERIOR 0,5) y dentro de la casa a 1,8 metros (INTERIOR 1,8)


*Figura 4.* Placa de sensores de temperatura

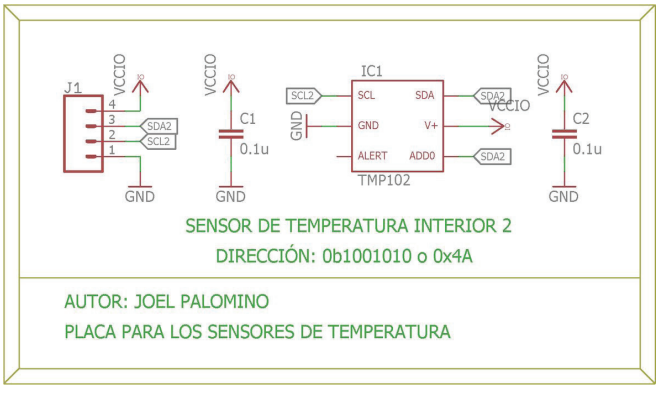



*Figura 5.* Diagrama esquemático de placa de sensores

De acuerdo con el diseño del *shield*, el RTC está conectado a los pines SDA y SCL del puerto I2C, en tanto que los pines SDA1 y SCL1 se conectan al sensor de temperatura. La razón, como se mencionó anteriormente, es que si el RTC está conectado junto con el sensor de temperatura, la capacitancia del bus aumentará y afectará el desempeño del bus I2C. Es necesario indicar que la precisión de los sensores fue validada mediante comparaciones con un equipo certificado (Fluke 971), según se muestra en la figura 6.

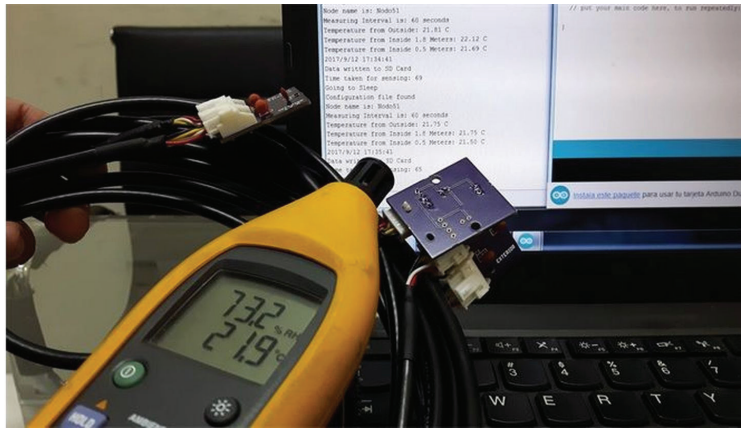



*Figura 6.* Validación de resultados de medición de sensores

Para almacenar los datos, se utilizó una tarjeta micro SD para ahorrar espacio y permitir su extracción y lectura como memoria externa en una computadora. Para este proyecto, la capacidad mínima de las tarjetas de memoria era de 4 GB. Sin embargo, solo se requiere una capacidad de memoria de 1 MB para los datos recopilados en un mes con una velocidad de muestreo de un minuto.

Finalmente, en la figura 7 se muestra el diagrama general del sistema desarrollado con todos los componentes y las conexiones entre ellos.

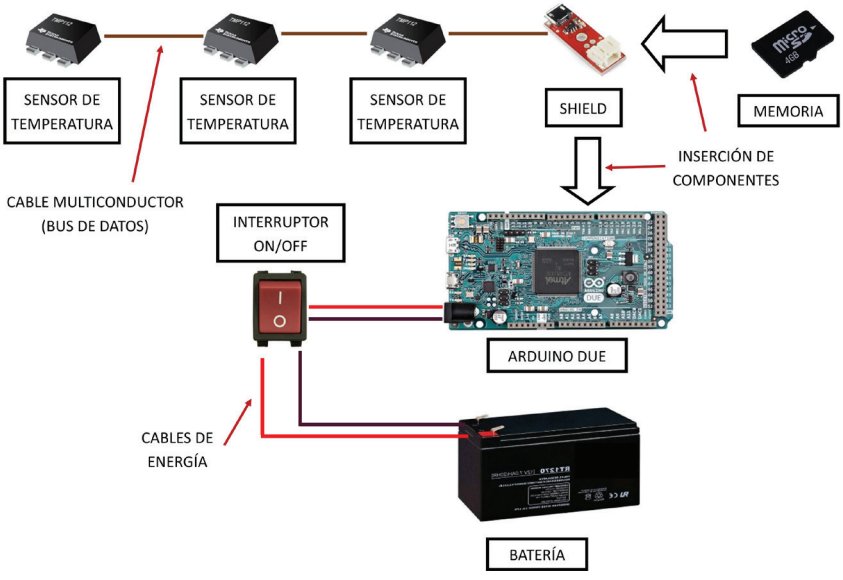



*Figura 7.* Esquema general del módulo desarrollado

## 3. INSTALACIÓN DEL SISTEMA DE MEDICIÓN

El *datalogger* desarrollado fue instalado en 54 casas que habían sido equipadas previamente con los muros Trombe. Estas casas estaban ubicadas en 18 comunidades en la región sur del Perú. Se eligieron las ubicaciones de los sensores según lo indicado en el apartado 2.1 (Requisitos) de este artículo.

Las zonas de instalación son áreas rurales sin electricidad, saneamiento, ni servicios de telecomunicaciones. Y en la mayoría de los casos, son áreas de difícil acceso (ver figura 8). Además, los pobladores en general solo hablan el idioma quechua.

*Figura 8.* Instalación del sistema de medición de temperatura

### 3.1 Tratamiento de los datos

Los datos se guardan en formato CSV para facilitar su visualización y gestión mediante un editor de hojas de datos. La figura 9 muestra los registros de medición durante un período de 24 horas, y se puede verificar que los valores de temperatura son coherentes. La totalidad de los datos recogidos son almacenados por Foncodes, cuyos funcionarios viajan periódicamente para cumplir la tarea de recopilar esta información. Finalmente, debe indicarse que estos registros están tabulados y resumidos.

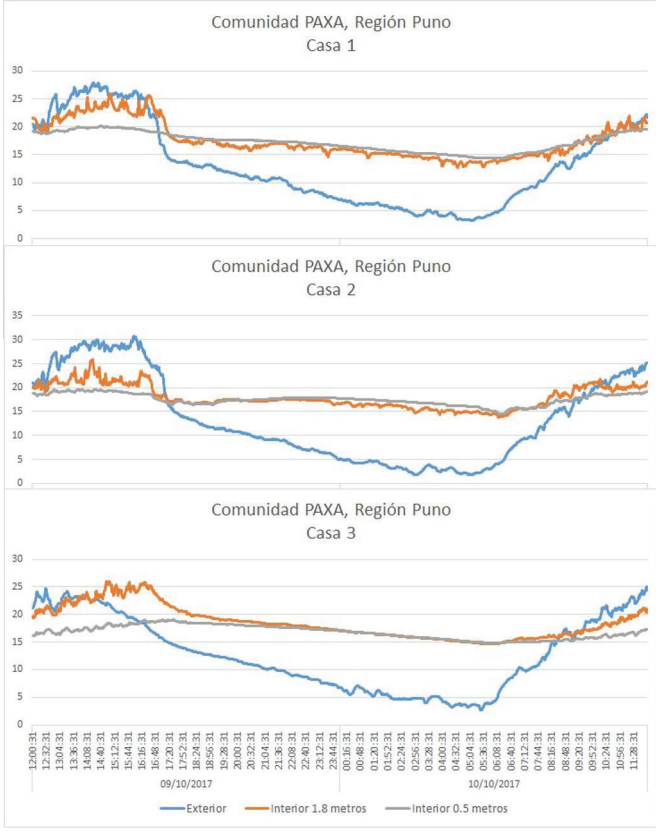



*Figura 9.* Registro de medición (24 horas) en tres casas

## 4. CONCLUSIONES

Este documento presenta la descripción y el proceso de implementación de un *shield* para el Arduino Due, para que trabaje como un *datalogger* de bajo consumo de energía para medición de temperatura. Este sistema se instaló en condiciones geográficas extremas

y se pudo comprobar su correcto funcionamiento (lecturas estables y consistentes) a pesar de las condiciones ambientales difíciles. El *datalogger* es empleado actualmente por el Gobierno para recopilar datos de temperatura. Además, se verificó también que los sensores de temperatura han medido valores negativos según lo requerido por Foncodes (el rango completo es de -45 a 12 °C).

Por otro lado, la implementación exitosa del sistema de medición de temperatura ha demostrado que el protocolo I2C puede usarse sobre cables relativamente largos, validando también el cálculo realizado para encontrar un valor adecuado para las resistencias *pull-up*. Esto abre la puerta a aplicaciones que pueden emplear el protocolo I2C en condiciones no estándar.

Este sistema de medición se utiliza actualmente en el proyecto Mi Abrigo para verificar que las casas acondicionadas con muros Trombe mejoran las condiciones de vida de las personas en zonas rurales aisladas de los Andes peruanos.

Finalmente, como un resultado colateral, se considera necesario indicar que, gracias al apoyo del Gobierno, se ha demostrado que es posible desarrollar soluciones confiables a medida y de bajo costo para cubrir necesidades específicas del sector público. Además, se ha probado que no en todos los casos la solución técnica pasa, necesariamente, por importar tecnología.

## REFERENCIAS

Ana, B., Anabela, P., e Ivo, P. (2016). Dynamic Simulation of the Trombe Wall Thermal Performance. *IAHS World Conference*. Algarbe.

Foncodes. (2017). *Ministerio de Desarrollo e Inclusión Social. Proyecto Mi Abrigo*. Recuperado de http://www.foncodes.gob.pe/portal/index.php/proyectos/miabrigo

General Cable/ Carol Brand. (2018). *4 Wire Multi-Conductor Foild Shield Computer Cable*. Recuperado de https://www.generalcable.com/assets/documents/LATAM%20Documents/Mexico%20Site/Cat%C3%A1logos/Electronics.pdf?ext=.pd

LabJack Measurement & Automation. (2017). *Digit Datasheet*. Recuperado de https://labjack.com/support/datasheets/digit

National Renewable Energy Laboratory. (2017). *NREL*. Recuperado de U.S. Deparment of Energy: https://www.nrel.gov/docs/legosti/fy98/22834.pdf

NXP Semiconductors. (2014). *I2C Bus Specification and User Manual*. Recuperado de https://www.nxp.com/docs/en/user-guide/UM10204.pdf

Sousa, J. (2017). *I2C Network for Home Automation*. Portugal: University of Porto.

Texas Instruments. (2018). *Digital Temperature Sensors With SMBus and TwoWire Serial Interface in SOT563*. Dallas.

Truchsess, W. (2010). *Effects on Varying I2C Pull Up Resistors*. Recuperado de https://www.thebackshed.com/forum/uploads/BobD/2013-01-26_175209_I2C_-_Effects_of_Varying_Pullup_Resistors.pdf

Recibido: 8/9/2020 / Aceptado: 7/11/2020
doi: https://doi.org/10.26439/interfases2020.n013.5026

# Revisión de literatura sobre las barreras a la transformación digital y su relación con el rendimiento financiero

Rubén Ahomed
Facultad de Comunicación, Universidad de Lima, Lima, Perú
aahomed@ulima.edu.pe
ORCID 0000-0002-7307-5375

Resumen

La transformación digital es una actividad que han venido adoptando las empresas para aprovechar los desarrollos tecnológicos digitales y mantenerse a la vanguardia empresarial rentablemente. El propósito del presente análisis es realizar una revisión de la literatura relacionada con la transformación digital, las barreras que existen para alcanzarla y su rendimiento financiero. A través de una revisión de 59 artículos, se definió un marco conceptual con nueve definiciones para la transformación digital, nueve para las barreras a la innovación y cinco para el rendimiento financiero. Los principales hallazgos son que existen barreras para el desarrollo de la transformación digital empresarial, pero difieren según la industria y país; por otro lado, el beneficio financiero no es claro y depende del contexto.

PALABRAS CLAVE: transformación digital / barreras / rentabilidad / rendimiento financiero

Abstract

## LITERATURE REVIEW ON BARRIERS TO DIGITAL TRANSFORMATION AND ITS RELATIONSHIP WITH THE FINANCIAL PERFORMANCE

Digital transformation is an activity that companies have been adopting to take advantage of digital technology developments and stay at the business forefront profitably. The purpose of this analysis is to review the literature related to digital transformation, the barriers to achieving it and its financial performance. Through a review of 59 articles, a conceptual framework was defined with nine definitions for digital transformation, nine for barriers to innovation and five for financial performance. The main findings are that there are barriers to the development of the digital business transformation, but they differ according to the industry and country. On the other hand, the financial benefit is not clear and depends on the context.

KEYWORDS: digital transformation / barriers / profitability / financial performance

## INTRODUCCIÓN

Una empresa en proceso de transformación digital utiliza nuevas tecnologías digitales como los medios sociales, el acceso móvil, el análisis o los dispositivos integrados para permitir importantes progresos empresariales como la mejora de la experiencia del cliente, la racionalización de las operaciones o la creación de nuevos modelos empresariales (Singh y Hess, 2017). Sin embargo, se advierte que la transformación digital es un proceso de cambio, que no se enfoca en la tecnología, sino que se sirve de ella para lograr sus objetivos (Westerman, 2018). Asimismo, en un entorno VUCA (abreviación, por sus siglas en inglés, de *volatility, uncertainty, complexity, ambiguity*), en donde la transformación digital se desarrolla dentro de un estado de bloqueo, producto de la pandemia por el COVID 19, es imperativo que las empresas adopten modelos digitales de gestión para alcanzar un cierto nivel de madurez digital, puesto que empresas con mayor nivel de madurez, son generalmente más flexibles. Y, por el contrario, las empresas menos maduras son más frágiles (Fletcher y Griffiths, 2020).

## 2. MARCO TEÓRICO-CONCEPTUAL: ESTADO DEL ARTE

### 2.1 Transformación digital

La transformación digital es un proceso que tiene por objeto mejorar una empresa promoviendo cambios en sus características mediante la combinación de tecnologías de la información, la computación, la comunicación y la conectividad (Legner, Eymann, Hess, Matt, Böhmann, Drews, Mädche, Urbach y Ahlemann, F., 2017; Vial, 2019). Es la transformación de las actividades, los procesos, las competencias y los modelos empresariales para aprovechar los cambios y las oportunidades que traen consigo las tecnologías digitales (Demirkan, Spohrer y Welser, 2016). La transformación digital es una forma de transformación empresarial, facilitada por los sistemas de información, que va acompañada de cambios económicos y tecnológicos tanto a nivel organizativo como industrial (Chanias, Myers y Hess, 2019); dependiendo de la industria, el desarrollo de capacidades dinámicas gerenciales y organizacionales es clave para que su desarrollo sea exitoso (Li, Su, Zhang y Mao, 2017). Finalmente, para lograr una transformación digital exitosa, se tiene que trabajar con el personal de la empresa dándoles autonomía para crear o participar en equipos de la manera que deseen, incluso con socios externos (Westerman, Soule y Eswaran, 2019).

### 2.2 Barreras a la innovación

Las barreras son definidas como obstáculos a la innovación (Borins, 2001) o como factores que inhiben la aceptación de la tecnología (Cenfetelli y Schwarz, 2011). Asimismo, muchas empresas pueden tomar diferentes rutas para mejorar su rendimiento cuando la transformación digital cambia sus modelos de negocios (Bouwman, Nikou, y de Reuver, 2019). Bajo

esta perspectiva, la inercia y la resistencia son las principales barreras organizacionales que limitan un proceso de transformación digital (Vial, 2019), aglutinando a muchas otras. Por ejemplo, en empresas de negocios electrónicos se encontraron cinco barreras para la adopción de innovación: a) percepción de riesgo, b) déficit de conocimiento, c) confianza, d) tamaño de la empresa y e) disposición organizativa (Johnson, 2010). En empresas con pocos años, se encontraron siete barreras: a) falta de financiación al interior de la empresa, b) falta de financiación de otras empresas, c) falta de personal calificado, d) falta de información en tecnología, e) dificultad para encontrar socios para innovación, f) mercado dominado por empresas establecidas y g) demanda incierta de bienes o servicios innovadores (Pellegrino, 2018). En educación, se encontraron cuatro barreras: a) el desafío de aprovechar la motivación intrínseca de los participantes, b) la creación de oportunidades de colaboración y comunicación entre los diferentes interesados, c) la búsqueda de tiempo suficiente para programar y aplicar una innovación y d) la falta de apoyo a los recursos digitales (Lanford, Corwin, Maruco y Ochsner, 2019).

### 2.3 Rendimiento financiero

El rendimiento financiero producto de la innovación digital es contradictorio y probablemente dependiente del contexto (Khin y Ho, 2019). Sin embargo, se encontró que las empresas con niveles de innovación digital superiores al promedio tienen mayores ingresos y márgenes de rentabilidad en 9 % y 26 % respectivamente (Westerman, Bonnet y McAfee, 2014). En una investigación hecha a 450 grandes empresas y una encuesta a 1559 personas en 106 países, se encontró que el 74 % de las organizaciones no han podido establecer indicadores clave para ayudarlos a medir el impacto financiero de la transformación digital (Fitzgerald, Kruschwitz, Bonnet y Welch, 2013). En Rusia se halló que las empresas con mayor actividad digital obtienen mayores ingresos, beneficios y valor de mercado que aquellas con una actividad pasiva (Veselovsky, Izmailova, Yunusov, y Yunusov, 2019). En Malasia encontraron que la innovación digital medía el efecto de la orientación tecnológica y la capacidad digital en los resultados financieros y no financieros (Khin y Ho, 2019). En un estudio longitudinal hecho en Bélgica los periódicos de la zona flamenca siguen buscando rentabilidad al apostar por la reducción de costos y la expansión a pesar de la penetración de la digitalización de los contenidos (Van der Burg y Van den Bulck, 2017).

## 3. METODOLOGÍA

La revisión de la literatura se realizó utilizando la base de datos Web of Science (WoS), con la siguiente fórmula: *digital transformation AND barriers*; para ello se seleccionaron los artículos desde el año 2016 hasta agosto del 2020 obteniendo 89 artículos en 63 categorías. Se revisaron los artículos de las primeras diez categorías con un total de

59 artículos que representó el 67 % del total. Finalmente, se aplicó el criterio de definición de conceptos para escoger los artículos que explicaban detalladamente ambos constructos quedando nueve artículos por cada uno. Asimismo, se consideraron otros autores que fueron fuentes bibliográficas de varios de los autores que se investigó según los resultados en WoS, pero que no estaban dentro del criterio de tiempo que se usó para discriminar los años de evaluación de artículos.

## 4. RESULTADOS

Los resultados presentados a continuación se resumen en dos tablas con los conceptos de la transformación digital y las barreras a la innovación.

Tabla 1
*Conceptos de transformación digital*

| Autor | Concepto |
|---|---|
| Vial (2019) | Proceso que tiene por objeto mejorar una entidad desencadenando cambios significativos en sus propiedades mediante combinaciones de tecnologías de información, informática, comunicación y conectividad. |
| Demirkan, H., Spohrer, J. C. y Welser, J. J. (2016) | Es la transformación profunda y acelerada de las actividades, procesos, competencias y modelos comerciales para aprovechar los cambios y oportunidades que traen consigo las tecnologías digitales y su impacto en toda la sociedad de manera estratégica y prioritaria. |
| Chanias, S., Myers, M. D. y Hess, T. (2019) | El uso extendido de la informática avanzada, como la analítica, la computación móvil, *social media* o dispositivos inteligentes incorporados y la mejora del uso de tecnologías tradicionales, como la planificación de los recursos empresariales (ERP), para permitir grandes mejoras en los negocios. |
| Li, L., Su, F., Zhang, W. y Mao, J. Y. (2017) | La transformación digital pone de relieve el impacto de la tecnología de la información en la organización, estructura, rutinas y flujo de información. |
| Andriole, S. (2017) | La transformación digital no es una actualización de *software* o un proyecto de mejora en la cadena de suministro. Es un impacto digital planeado para lo que puede ser un sistema que funciona razonablemente. |
| Singh, A. y Hess, T. (2017) | Una empresa en proceso de transformación digital utiliza nuevas tecnologías digitales, como los medios sociales, el acceso móvil, el análisis o los dispositivos integrados para permitir importantes mejoras empresariales, como la mejora de la experiencia del cliente, la racionalización de las operaciones o la creación de nuevos modelos empresariales. |
| Westerman, G. (2018) | Es un proceso de cambio, que no se enfoca en la tecnología, sino que se sirve de ella para lograr sus objetivos. |
| Fitzgerald, M., Kruschwitz, N., Bonnet, D. y Welch, M. (2013) | El uso de nuevas tecnologías digitales (medios sociales, móviles, analíticas o dispositivos incorporados) para permitir importantes mejoras comerciales (como mejorar la experiencia del cliente, racionalizar las operaciones o crear nuevos modelos de negocio). |
| Legner *et al.* (2017) | Describe los cambios impuestos por las tecnologías de la información (TI) como un medio para automatizar (parcialmente) las tareas. |

Elaboración propia

Tabla 2
*Barreras a la innovación*

| Autor | Concepto |
|---|---|
| Johnson, M. (2010) | a) Percepción de riesgo b) Déficit de conocimiento c) Confianza, d) Tamaño de la empresa e) Disposición organizativa. |
| Fitzgerald, M., Kruschwitz, N., Bonnet, D. y Welch, M. (2013) | a) Falta de urgencia b) Falta de visión c) Falta de dirección d) Actitud de los trabajadores con mayor experiencia e) Tecnología tradicional f) Fatiga a la innovación g) Política interna h) Abogar por el cambio i) Falta de incentivos. |
| Pellegrino, G. (2018) | a) Falta de financiación al interior de la empresa b) Falta de financiación de otras empresas c) Falta de personal calificado d) Falta de información en tecnología e) Dificultad para encontrar socios para innovación f) Mercado dominado por empresas establecidas g) Demanda incierta de bienes o servicios innovadores. |
| Lanford, M., Corwin, Z. B., Maruco, T. y Ochsner, A. (2019) | a) El desafío de aprovechar la motivación intrínseca de los participantes b) La creación de oportunidades de colaboración y comunicación entre los diferentes interesados c) La búsqueda de tiempo suficiente para programar y aplicar una innovación d) La falta de apoyo a los recursos digitales. |
| Horváth, D. y Szabó, R. Z. (2019) | a) Recursos humanos y circunstancias de trabajo b) Escasez de recursos financieros c) Problemas de estandarización d) Preocupación por la seguridad cibernética y los problemas de propiedad de los datos e) Riesgo de fragilidad f) Integración tecnológica g) Dificultad de coordinación entre las unidades organizativas h) Falta de habilidades y actividades de planificación i) Resistencia organizativa. |
| Vial, G. (2019) | a) Inercia b) Resistencia. |
| Cichosz, M.,Wallenburg, C. M. y Knemeyer, A. M. (2020) | a) Complejidad de la red logística (en el negocio de los servicios logísticos) b) Falta de recursos. |
| Agrawal, P., Narain, R. y Ullah, I. (2020) | a) Alto sentido de urgencia b) Falta de directrices específicas para la industria c) Falta de habilidades y talento digital d) Alto costo de implementación y funcionamiento. |
| Stentoft, J., Adsbøll Wickstrøm, K., Philipsen, K. y Haug, A. (2020) | a) Falta de conocimiento sobre las nuevas tecnologías digitales b) Falta de normas c) Más atención a la operación a expensas del desarrollo de la empresa d) Falta de protección de datos (ciberseguridad) e) Falta de personal calificado f) Falta de preparación de los empleados g) Requerimiento de la educación continua de los empleados h) Falta de comprensión de la importancia estratégica de las nuevas tecnologías digitales i) Falta de comprensión de la interacción entre la tecnología y los seres humanos j) Pocos recursos financieros k) Pocos recursos humanos (mano de obra). |

Elaboración propia

## 5. DISCUSIÓN

Se encontró una variedad de definiciones del concepto de transformación digital vinculados a procesos, competencias y modelos comerciales, relacionados con la informática

o al impacto de la tecnología de la información. Por otro lado, existe una variedad de barreras dependiendo de las industrias y los países pero que se resumen dentro de los conceptos de inercia y resistencia. Finalmente, los resultados financieros son aún difíciles de identificar (Fitzgerald *et al*., 2013); sin embargo, existe evidencia de que es positiva, no obstante, siguen prácticas tradicionales para encontrar el beneficio.

## 6. CONCLUSIONES

La transformación digital empresarial en un proceso que se viene implementando hace varios años, su objetivo fundamental es llevar a las organizaciones a un estado que haga que sus negocios se mantengan rentables. Para lograr esto se valen del uso de la tecnología digital en sus diferentes manifestaciones, superando una serie de barreras. Sin embargo, encontrar el rendimiento financiero sigue siendo una labor compleja de lograr puesto que no se cuenta con el desarrollo correcto de los indicadores o en otros casos depende del contexto en que se desarrolla la empresa. Finalmente, en el entorno VUCA de la actualidad, las empresas tienen que buscar aceleradamente la digitalización de sus negocios, puesto que les proveerá de la agilidad necesaria para hacer frente a los cambios producto de factores exógenos como el del COVID 19.

## REFERENCIAS

Agrawal, P., Narain, R. y Ullah, I. (2020). Analysis of Barriers in Implementation of Digital Transformation of Supply Chain Using Interpretive Structural Modelling Approach. *Journal of Modelling in Management*, *15*(1), 297-317. https://doi.org/10.1108/JM2-03-2019-0066

Andriole, S. (2017). Five Myths About Digital Transformation. *MIT Sloan Management Review*, *58*(3), 20-22.

Borins, S. (2001). Innovation, Success and Failure in Public Management Research: Some Methodological Reflections. *Public Management Review*, *3*(1), 3-17. https://doi.org/10.1080/14616670010009423

Bouwman, H., Nikou, S. y de Reuver, M. (2019). Digitalization, business models, and SMEs: How do business model innovation practices improve performance of digitalizing SMEs? *Telecommunications Policy*, *43*(9), 101828. https://doi.org/10.1016/j.telpol.2019.101828

Cenfetelli, R. T. y Schwarz, A. (2011). Identifying and Testing the Inhibitors of Technology Usage Intentions. *Information Systems Research*, *22*(4), 808-823. https://doi.org/10.1287/isre.1100.0295

Chanias, S., Myers, M. D., y Hess, T. (2019). Digital Transformation Strategy Making in Pre-Digital Organizations: The Case of a Financial Services Provider. *Journal of Strategic Information Systems*, *28*(1), 17-33.

Cichosz, M., Wallenburg, C. M. y Knemeyer, A. M. (2020). Digital Transformation at Logistics Service Providers: Barriers, Success Factors and Leading Practices. *International Journal of Logistics Management*, *31*(2), 209-238.

Demirkan, H., Spohrer, J. C. y Welser, J. J. (2016). Digital Innovation and Strategic Transformation. *IT Professional*, *18*(6), 14-18. https://doi.org/10.1287/isre.1100.0295

Fitzgerald, M., Kruschwitz, N., Bonnet, D. y Welch, M. (2013). Embracing Digital Technology: A New Strategic Imperative. *MIT Sloan Management Review*, *55*(2), 1-12.

Fletcher, G. y Griffiths, M. (2020). Digital Transformation During a Lockdown. *International Journal of Information Management, ahead-of-p(ahead-of-print)*. https://doi.org/10.1016/j.ijinfomgt.2020.102185

Horváth, D. y Szabó, R. Z. (2019). Driving Forces and Barriers of Industry 4.0: Do Multinational and Small and Medium-Sized Companies Have Equal Opportunities? *Technological Forecasting and Social Change*, *146*, 119-132. https://doi.org/10.1016/j.techfore.2019.05.021 h

Johnson, M. (2010). Barriers to Innovation Adoption: A Study of E-Markets. *Industrial Management & Data Systems*, *110*(2), 157-174. https://doi.org/10.1108/02635571011020287

Khin, S. y Ho, T. C. F. (2019). Digital Technology, Digital Capability and Organizational Performance: A Mediating Role of Digital Innovation. *International Journal of Innovation Science*, *11*(2), 177-195. h https://doi.org/10.1108/IJIS-08-2018-0083

Lanford, M., Corwin, Z. B., Maruco, T. y Ochsner, A. (2019). Institutional Barriers to Innovation: Lessons From a Digital Intervention for Underrepresented Students Applying to College. *Journal of Research on Technology in Education*, *51*(3), 203-216. https://doi.org/10.1080/15391523.2019.1576558 ps

Legner, C., Eymann, T., Hess, T., Matt, C., Böhmann, T., Drews, P., Mädche, A., Urbach, N. y Ahlemann, F. (2017). Digitalization: Opportunity and Challenge for the Business and Information Systems Engineering Community. *Business & Information Systems Engineering*, *59*(4), 301-308. https://doi.org/10.1007/s12599-017-0484-2

Li, L., Su, F., Zhang, W. y Mao, J. Y. (2017). Digital Transformation by SME Entrepreneurs: A Capability Perspective. *Information Systems Journal*, *28*(6), 1129-1157.

Pellegrino, G. (2018). Barriers to Innovation in Young and Mature Firms. *Journal of Evolutionary Economics*, *28*(1), 181-206. https://doi.org/10.1007/s00191-017-0538-0

Singh, A. y Hess, T. (2017). How Chief Digital Officers Promote the Digital Transformation of their Companies. *MIS Quarterly Executive*, *16*(1), 1-17.

Stentoft, J., Adsbøll Wickstrøm, K., Philipsen, K. y Haug, A. (2020). Drivers and Barriers for Industry 4.0 Readiness and Practice: Empirical Evidence from Small and Medium-Sized Manufacturers. *Production Planning and Control*, 1-18. https://doi.org/10.1080/09537287.2020.1768318

Van der Burg, M. y Van den Bulck, H. (2017). Why are Traditional Newspaper Publishers Still Surviving in the Digital Era? The impact of Long-Term trends on the Flemish Newspaper Industry's Financing, 1990–2014. *Journal of Media Business Studies*, *14*(2), 82-115. https://doi.org/10.1080/16522354.2017.1290024

Veselovsky, M. Y., Izmailova, M. A., Yunusov, L. A., y Yunusov, I. A. (2019). Quality of Digital Transformation Management on the Way of Formation of Innovative Economy of Russia. *Quality - Access to Success, 20*(169), 66-71.

Vial, G. (2019). Understanding Digital Transformation: A Review and a Research Agenda. *Journal of Strategic Information Systems*, *28*(2), 118-144. https://doi.org/10.1016/j.jsis.2019.01.003

Westerman, G. (2018). Your Company Doesn't Need a Digital Strategy. *MIT Sloan Management Review*, *59*(3), 1-5.

Westerman, G., Bonnet, D. y McAfee, A. (2014). *Leading Digital: Turning Technology into Business Transformation*. Boston, MA: Harvard Business Review Press.

Westerman, G., Soule, D. L. y Eswaran, A. (2019). Building Digital Ready Culture in Traditional Organizations. *MIT Sloan Management Review*, *60*(4), 59-68.

Recibido: 23/9/2020 / Aceptado: 7/11/2020
doi: https://doi.org/10.26439/interfases2020.n013.4857

# Software in the Loop para la implementación de un sistema de piloto automático para aeronaves de ala fija

Lennin Paul Quiroz Villalobos
Universidad de Lima, Lima, Perú
(lquirozv@ulima.edu.pe)
https://orcid.org/0000-0003-1517-8963

Resumen

Cuando se desea desarrollar y probar *softwares* para sistemas complejos, como aviones, transbordadores espaciales, satélites, automóviles o plantas nucleares, uno de los mayores inconvenientes es la gran complejidad, riesgos y costos que implica realizar pruebas sobre el sistema real. Software in the Loop (SITL) permite testear algoritmos, códigos fuente o estrategias de control para sistemas complejos dentro de una simulación que contiene el modelo matemático del sistema físico real. El presente trabajo propone el uso de la plataforma Software in the Loop con el fin de desarrollar un sistema de piloto automático para una aeronave de ala fija. Se muestra la arquitectura de la aplicación implementada, el proceso de diseño de *software*, los protocolos utilizados, las estrategias de control, técnicas de programación, los resultados obtenidos y las conclusiones.

PALABRAS CLAVE: SITL / piloto automático / aeronaves / ingeniería de *software* / Simulink / PID

Abstract

## SOFTWARE IN THE LOOP FOR THE IMPLEMENTATION OF AN AUTOPILOT SYSTEM FOR FIXED-WING AIRCRAFTS

When you want to develop and test software products for complex systems such as airplanes, space shuttles, satellites, automobiles or nuclear plants, one of the biggest drawbacks is the availability of the physical system, due to the great complexity, risks and costs involved in performing tests on the real system. Software in the Loop (SITL) allows us to test algorithms, source code or control strategies for complex systems within a simulation that contains the mathematical model of the real physical system. This research work proposes the use of a Software in the Loop platform for the development of an autopilot system for a fixed-wing aircraft. The architecture of the implemented application, the software design process, the protocols used, the control strategies, the programming techniques, the results obtained and the conclusions are shown herein.

KEYWORDS: SITL / autopilot / aircrafts / software engineering / Simulink / PID

## 1. INTRODUCCIÓN

La ingeniería de *software* es una disciplina que tiene como objetivo proporcionar teorías, métodos y herramientas para el desarrollo de *software* de calidad. El ciclo de desarrollo de un producto *software* comprende requisitos, diseño, implementación, pruebas y mantenimiento. Cuando se trata de sistemas complejos el desarrollo debe ser iterativo e incremental, agregando nuevas funcionalidades en cada ciclo. Antes de aceptar o dar por válido un nuevo módulo, este debe pasar por un proceso de verificación y validación. En la etapa de verificación se comprueba si se está cumpliendo con los requisitos establecidos; luego, en la etapa de validación se comprueba si el *software* hace realmente lo que se espera. La mejor manera de validar un *software* es sobre el sistema físico sobre el cual operará; sin embargo, cuando se trata de sistemas complejos, no es posible realizar el proceso de validación parcial o total sobre el sistema real, debido al costo, riesgos o tiempo empleado. En este tipo de proyectos se utiliza una plataforma Software in the Loop, la cual permite interactuar con el modelo matemático del sistema físico en un entorno virtual para probar algoritmos, código fuente, estrategias de control, observar sus efectos sobre el modelo, así como detectar y corregir errores a un costo mínimo, antes de que se propaguen al resto del sistema.

El presente trabajo de investigación, propone el desarrollo de un sistema de piloto automático para una aeronave de ala fija, usando una plataforma Software in the Loop. Inicialmente, se muestra el ambiente de desarrollo implementado y sus componentes, luego, los conceptos correspondientes a la dinámica de vuelo de la aeronave, los parámetros críticos de operación, las estrategias de control propuestas, las pruebas realizadas y las conclusiones.

## 2. ARQUITECTURA DE LA APLICACIÓN

A continuación, se muestra la arquitectura de la plataforma SITL implementada:

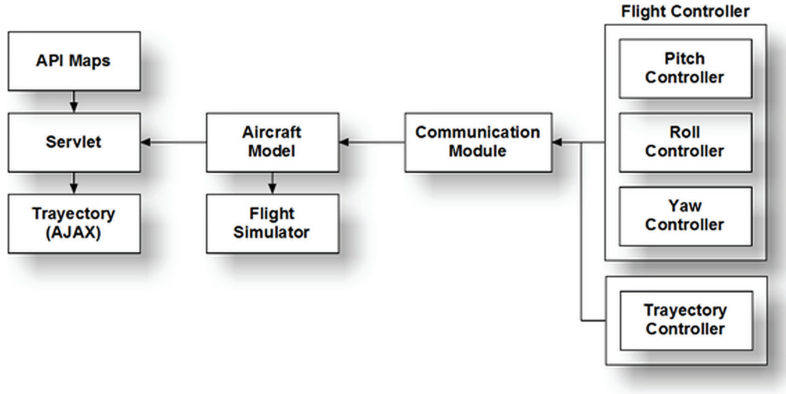



*Figura 1.* Arquitectura SITL
Elaboración propia

Tal como se muestra en la figura 1, los módulos de control de vuelo (Pitch, Roll y Yaw), así como el módulo de ajuste de trayectoria se comunican con los módulos centrales, conteniendo el modelo de la aeronave de ala fija y la salida del simulador, a través del módulo de comunicaciones, usando el protocolo UDP (User Datagram Protocol). Así mismo, el bloque *servlet*, a partir de las coordenadas latitud y longitud generadas por el simulador, utiliza la API de Microsoft Maps, para mostrar en tiempo real la posición actual y la trayectoria de la aeronave utilizando la tecnología AJAX (Asynchronous JavaScript and XML).

## 3. DISEÑO DE LA PLATAFORMA SITL

El sistema SITL implementado es un conjunto de componentes conectados entre sí a través de interfaces y protocolos. Un componente es una parte modular de la arquitectura física, cuya principal característica es que oculta su implementación tras un conjunto de interfaces externas, esto permite que las interfaces requeridas y ofertadas puedan ser reemplazables por nuevas versiones, implementadas en diferentes lenguajes o accedidas directamente para propósitos de *testing*.

Durante el proceso de implementación, tener una vista arquitectónica de alto nivel de los componentes del sistema fue fundamental para poder programar y priorizar las tareas de implementación, así como para poder determinar en qué componentes del sistema sería necesario realizar pruebas unitarias y pruebas de integración. Para lograr este objetivo el Diagrama de Componentes del sistema fue necesario.

La figura 2 muestra el diagrama de componentes del sistema implementado. Los módulos del componente Controller fueron desarrollados en Simulink (Matlab R2013b), los módulos del componente Simulator fueron acondicionados usando el simulador FlightGear (Ver 2.10) y los módulos del componente Maps fueron desarrollados en Java (Ver. 8.0.65). Se utilizó el sistema operativo Windows 7.

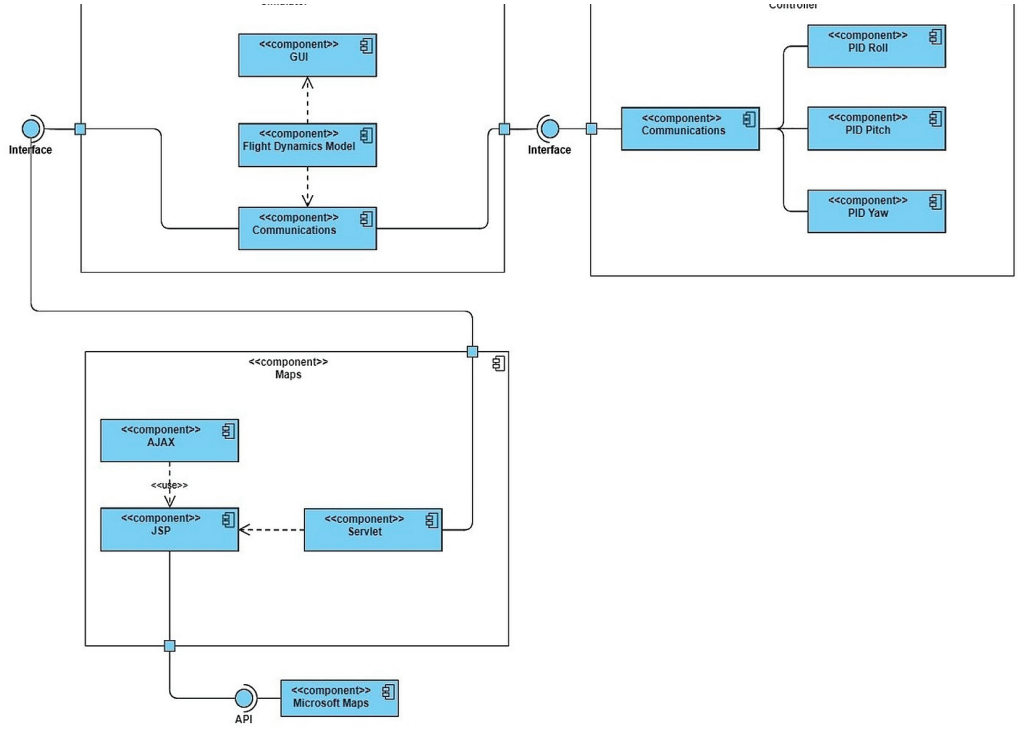



*Figura 2.* Diagrama de componentes de la plataforma SITL

Elaboración propia

Así mismo, para poder modelar la manera en que los componentes interactúan entre sí a través del tiempo para completar la funcionalidad de la plataforma SITL, el Diagrama de Secuencia del sistema también fue necesario. La principal ventaja del diagrama de secuencia es que nos permite representar la lógica de operación entre los diferentes módulos que componen el sistema en orden cronológico.

Tal como se muestra en la figura 3, el diagrama de secuencia de la plataforma SITL nos permite describir cómo los elementos que componen el sistema intercambian mensajes para llevar a cabo las acciones de control y el trazado de la trayectoria sobre el mapa.

Las líneas de vida (líneas verticales) para los elementos del simulador y los elementos del sistema de control, se van intercalando en un bucle infinito para detectar las posiciones relativas de las superficies de control y aplicar las acciones correctivas necesarias para mantener la aeronave en vuelo.

Así mismo, los parámetros latitud y longitud generados por el simulador son informados periódicamente a los elementos encargados de describir la trayectoria de la aeronave. Utilizando el modelo MVC (Modelo-Vista-Controlador) el *servlet* Controller actualiza permanentemente los mapas desde la API de Microsoft Maps de acuerdo con el desplazamiento de la aeronave; luego, la posición exacta dentro del mapa debe ser especificada mediante un marcador, sin perder el histórico de posiciones previas, para lo cual se utiliza la tecnología AJAX.

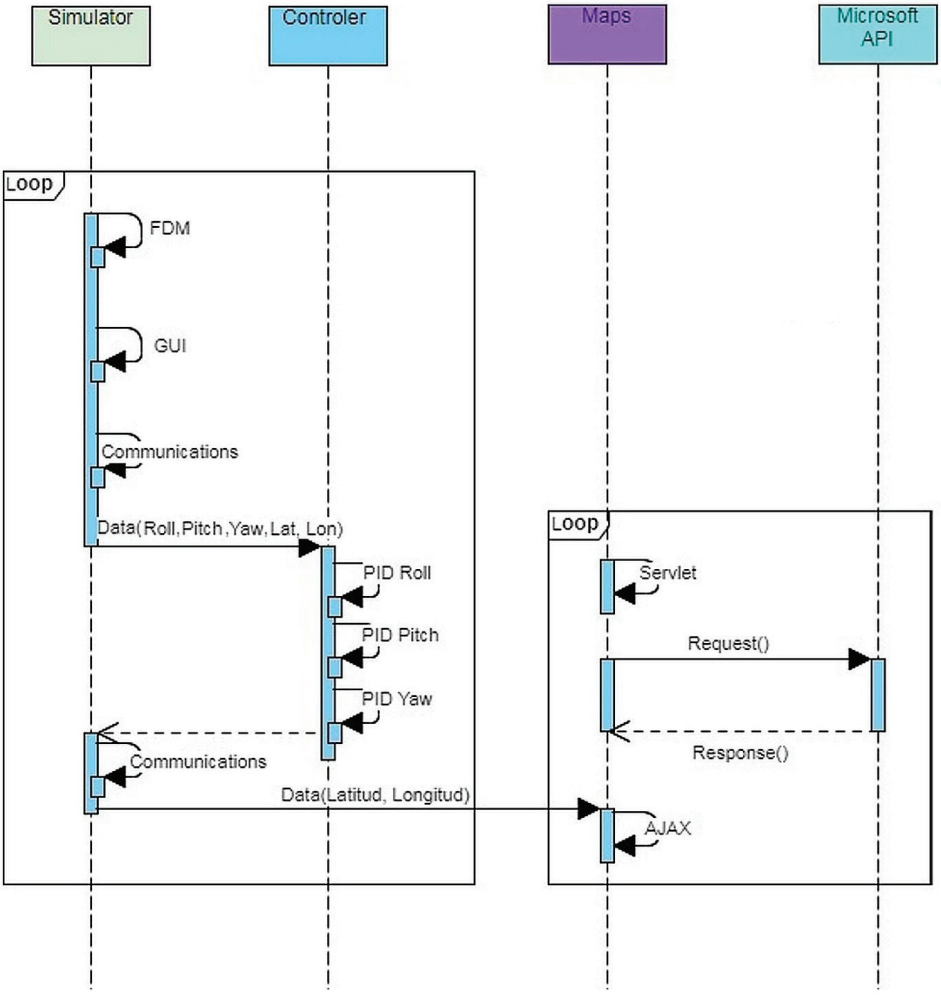



*Figura 3.* Diagrama de secuencia para la plataforma SITL

Elaboración propia

Finalmente, el ciclo de vida de desarrollo del sistema ha sido iterativo e incremental.

En un ciclo de vida iterativo el desarrollo de un componente empieza desde la definición de requisitos, planificación de tareas, ejecución y evaluación; luego, se pasa a otro componente donde se repite el mismo proceso. El ciclo de vida iterativo se aplicó para el desarrollo de los módulos controladores de Pitch, Roll y Yaw.

En un ciclo de vida incremental se busca el crecimiento progresivo de la funcionalidad del sistema. El procedimiento consiste en dividir el proyecto en módulos bien diferenciados, cada uno con una función específica. Cada módulo se va construyendo de acuerdo a su prioridad dentro del proyecto. Un aspecto sumamente importante en el ciclo de vida incremental son las pruebas de integración entre módulos para verificar que dos módulos que funcionan correctamente de forma separada lo sigan haciendo al integrarse. Así mismo, podría suceder que un módulo que antes funcionaba correctamente ahora se vea afectado por un nuevo módulo, por lo cual las pruebas de regresión también son sumamente importantes. El orden de prioridad en que fueron construidos los módulos de la plataforma SITL son: Simulator, Controller, Maps.

## 4. CONSTRUCCIÓN DE LA PLATAFORMA SITL

### 4.1 Dinámica de vuelo

Para un mejor entendimiento del sistema implementado, a continuación, se presenta una versión simplificada de la dinámica de vuelo de un avión. El comportamiento (*attitud*) de un avión en vuelo está determinado por sus ángulos Roll, Pitch y Yaw (Koks, 2008), tal como se muestra en la figura 4:

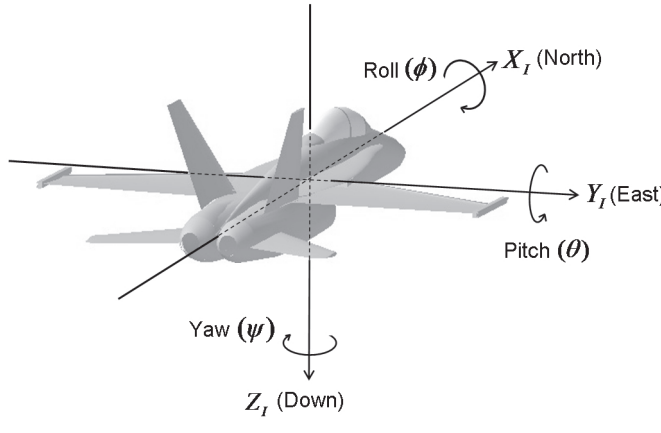



*Figura 4.* Ángulos de vuelo

Fuente: http://www.chrobotics.com/library/understanding-euler-angles

El Roll (alabeo), se considera como una rotación angular respecto al eje x. El Pitch (cabeceo), se considera como una rotación angular respecto al eje y. El Yaw (guiñada), se considera como una rotación angular respecto al eje z. Para una aeronave en vuelo, al cumplirse la segunda Ley de Newton, F=m.a, será un sistema de referencia no inercial y su orientación estará determinada por los valores de los ángulos Roll, Pitch y Yaw en cada instante de vuelo (Peet, 2010).

Sin embargo, para poder determinar las ecuaciones que gobiernan las leyes de vuelo del sistema, es necesario expresar las fuerzas que actúan en coordenadas de un sistema de referencia inercial. Para cumplir este objetivo, es necesario definir las matrices de rotación respecto a cada uno de los ángulos Roll (ecuación 1), Pitch (ecuación 2) y Yaw (ecuación 3), (Slabaugh, 2017), tal como se muestra:

$$R(x,\phi)=\begin{bmatrix}x_1\\y_1\\z_1\end{bmatrix}=\begin{bmatrix}1&0&0\\0&\cos\phi&-sen\,\phi\\0&sen\,\phi&\cos\phi\end{bmatrix}\begin{bmatrix}x_B\\y_B\\z_B\end{bmatrix} \quad (1)$$

$$R(y,\theta)=\begin{bmatrix}x_2\\y_2\\z_2\end{bmatrix}=\begin{bmatrix}\cos\theta&0&sen\,\theta\\0&1&0\\-sen\,\theta&0&\cos\theta\end{bmatrix}\begin{bmatrix}x_1\\y_1\\z_1\end{bmatrix} \quad (2)$$

$$R(z,\psi)=\begin{bmatrix}x_B\\y_B\\z_B\end{bmatrix}=\begin{bmatrix}\cos\psi&-sen\psi&0\\sen\psi&\cos\psi&0\\0&0&1\end{bmatrix}\begin{bmatrix}x_2\\y_2\\z_2\end{bmatrix} \quad (3)$$

Luego, la matriz de rotación completa se obtiene mediante tres rotaciones sucesivas de los ángulos (Jia, 2020), tal como se muestra:

$$R(\phi,\theta,\psi)=\begin{bmatrix}\cos\theta\cos\psi&\cos\psi\,sen\theta sen\phi-sen\psi\cos\phi&\cos\psi\,sen\theta\cos\phi+sen\psi\,sen\,\phi\\sen\psi\cos\theta&sen\psi\,sen\theta sen\phi+\cos\psi\cos\phi&sen\psi\,sen\theta\cos\phi-sen\phi\cos\psi\\-sen\,\theta&\cos\theta\,sen\,\phi&\cos\theta\cos\phi\end{bmatrix} \quad (4)$$

La ecuación 4, obtenida previamente, servirá para expresar la orientación, posición y movimiento de la aeronave respecto al sistema de referencia inercial.

### 4.2 Estrategia de control

Un PID (Proporcional Integral Derivativo) es un tipo de control cuya principal característica es generar una señal de corrección para el error que se obtiene al comparar el valor actual de un parámetro y el valor deseado o valor objetivo de dicho parámetro. Esta señal

de corrección se aplicará a los mecanismos actuadores del sistema; el nuevo valor será informado por los sensores y comparado nuevamente, funcionando así de forma realimentada. En un controlador PID la acción proporcional, determina la reacción o magnitud de la respuesta al error, la acción integral genera una señal de corrección proporcional a la integral del error y la acción derivativa determina la reacción con relación al tiempo en que se produce. En algunos casos, en función de la aplicación específica del controlador, alguno de los parámetros del PID podría ser cero (Astrom, 2002; Cova, 2005).

La figura 5 muestra cómo interactúan cada una de las tres acciones de control en un controlador PID.

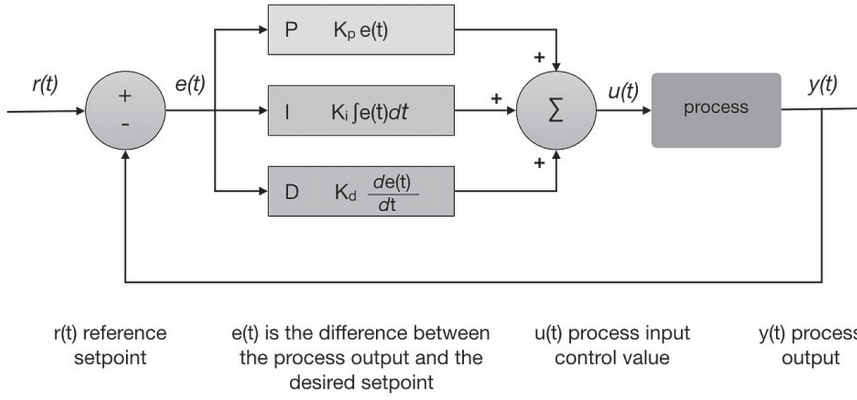



*Figura 5*. Control PID

Fuente: https://mjwhite8119.github.io/Robots/pid-control

Para la implementación del sistema de control de piloto automático, se han utilizado tres controladores PID, tal como se muestra en la figura 5, uno para cada grado de libertad de la aeronave. Los valores óptimos para los parámetros P, I, D de cada controlador se obtuvieron experimentalmente.

### 4.3 Flight Dinamics Models (FDM)

Un FDM contiene las ecuaciones matemáticas que son usadas para calcular las fuerzas físicas que actúan sobre la simulación de vuelo de una aeronave. Para la implementación del presente trabajo se usó el simulador FlightGear. Los FDM que incorpora Flight Gear son YASim, JSBSim y UIUC.

FlightGear (https://www.flightgear.org/) es un simulador *open source* y multiplataforma. En la instalación por defecto incluye muchos modelos de aeronaves; sin embargo, modelos adicionales pueden ser agregados desde la web de soporte o creados por el

mismo usuario en aplicaciones 3D externas y cargadas mediante un archivo XML que especifica las características de la aeronave. FlightGear está basado en OpenGL.

### 4.4 Sistema de control

El sistema de control implementado se muestra en la figura 6. Este está desarrollado utilizando la herramienta Simulink de Matlab, la cual utiliza los bloques de la librería *net_ctrl* packet para realizar la comunicación UDP con el simulador, y los bloques PID del *Control System Toolbox* para la implementación de los controladores. Los valores de latitud y longitud del destino deseado se setean manualmente antes de iniciar cada misión.

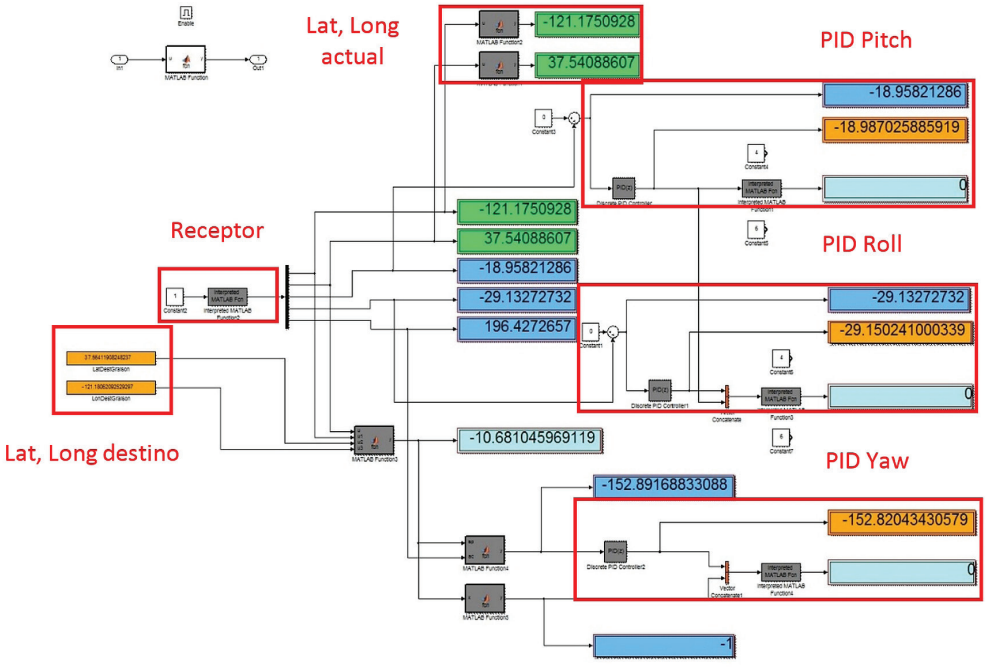



*Figura 6.* Bloques del sistema de control

Elaboración propia

### 4.5 Cálculo de distancia

Para que la aeronave pueda ir desde un punto origen a un punto destino, es necesario proporcionar las coordenadas Latitud y Longitud de la ciudad hacia donde queremos ir (destino). Las coordenadas Latitud y Longitud del aeropuerto donde se encuentra actualmente la aeronave se tomarán como coordenadas de origen. Para hacer el cálculo de la distancia entre origen y destino, se debe tener en cuenta que, debido a la curvatura de la Tierra, la distancia más corta entre dos puntos es un arco, tal como se muestra en la

figura 7. Por lo tanto, para realizar este cálculo, el uso de la fórmula Haversine fue necesario (Hartanto, Furqan, Putera, Siahaan y Fitriani, 2017).


*Figura 7.* Distancia más corta entre dos puntos

Elaboración propia

La longitud del sector circular será: d=θ.r, donde θ es el ángulo central formado por los puntos de origen, destino y r será el radio de la Tierra. La fórmula Haversine hav θ viene dada por: (Surowski, 2011).

$$\text{hav}(\theta) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\,\text{hav}(\lambda_2 - \lambda_1) \quad (5)$$

Donde

$$\varphi_1, \varphi_2 : \text{latitud del punto 1 y latitud del punto 2}$$
$$\lambda_1, \lambda_2 : \text{longitud del punto 1 y longitud del punto 2}$$

Luego, la función Haversine de un ángulo θ es:

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2} \quad (6)$$

Para obtener la distancia d, aplicamos Haversine inverso al ángulo central θ

$$d = r\,hav^{-1}(h) = 2r\arcsin(\sqrt{h}) \tag{7}$$

Donde h=hav *(θ).* Reemplazando, tenemos la distancia d a partir de las coordenadas de Latitud y Longitud de los puntos de origen y destino:

$$\begin{aligned} d &= 2r\arcsin\left(\sqrt{hav(\varphi_2-\varphi_1)+\cos(\varphi_1)\cos(\varphi_2)hav(\lambda_2-\lambda_1)}\right) \\ &= 2r\arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)+\cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \end{aligned} \tag{8}$$

La ecuación 8 muestra la expresión matemática necesaria para calcular la distancia entre dos puntos sobre la Tierra a partir de sus coordenadas de latitud y longitud.

### 4.6 Cálculo de orientación

Para calcular la orientación correcta, el sistema de control se encargará de ir calculado el vector direccional necesario para poder ajustar el rumbo de la aeronave hacia el destino. Sin embargo, el sistema de control requiere convertir las coordenadas Latitud y Longitud a señales de corrección que se puedan aplicar a los ángulos de la aeronave (Yaw). Para lograr este objetivo, el cálculo del ángulo Bearing también fue necesario.

El Bearing es el ángulo que se forma entre un meridiano y la línea que conecta hacia la posición actual del objeto. Así mismo, el ángulo Heading (Yaw) es el ángulo actual de la aeronave respecto al sistema de referencia no inercial; por lo tanto, para tener la certeza de que la aeronave está en el rumbo correcto, el ángulo Bearing y el Heading deben coincidir.

El cálculo del ángulo Bearing se muestra en la ecuación 9:

$$B = \text{atan}\left((\text{Cos}\,\theta_b * \text{Sin}(\Delta L)), (\text{Cos}\,\theta_a * \text{Sin}\,\theta_b - \text{Sin}\,\theta_a * \text{Cos}\,\theta_b * \text{Cos}\,\Delta L)\right) \tag{9}$$

Donde:

$\theta_a, \theta_b$ : latitud del punto $A$ y latitud del punto $B$

$\Delta L$ : diferencia de longitud entre el punto A y el punto B

El ángulo Bearing se calcula para cada punto donde se encuentre el avión; luego, el sistema de control se encarga de ir ajustando el Yaw para que los ángulos Bearing y Heading coincidan.

### 4.7 Mapas

Para la implementación de los mapas en tiempo real, se usó la API de Microsoft Maps mediante una función Java Script. Sin embargo, debido a que la información de Latitud y Longitud actual de la aeronave debe ser extraída desde el simulador, la implementación de un *servlet* en Java fue necesario. La aplicación web implementada está basada en el patrón MVC (Modelo-Vista-Controlador) donde el controlador está implementado en Java, la vista es un JSP (Java Server Pages) y el modelo es generado a través de un archivo CSV.

Así mismo, para la implementación de la gráfica de la trayectoria de la aeronave, manteniendo los registros previos de ubicación, el uso de la tecnología AJAX sobre JSP fue necesario. A continuación, en la figura 8, se muestra la función Ajax que se utilizó para graficar la trayectoria de la aeronave.

```
function Ajax()
{

        if( window.XMLHttpRequest )
                ajax = new XMLHttpRequest();
        else
                ajax = new ActiveXObject("Microsoft.XMLHTTP");

        ajax.open( "GET", "Servidor?"+0, true );
        ajax.send( "" );

        ajax.onreadystatechange = funcionCallback;

        setTimeout("parametros()",500);
}
```

*Figura 8.* Función AJAX implementada

Elaboración propia

## 5. PRUEBAS REALIZADAS Y RESULTADOS

Para las pruebas se utilizó la aeronave Piper J3 Cub, la cual se caracteriza por ser simple y ligera. Por esta razón, muchas veces es utilizada como aeronave de entrenamiento. La figura 9 muestra vistas desde diferentes ángulos de esta aeronave.

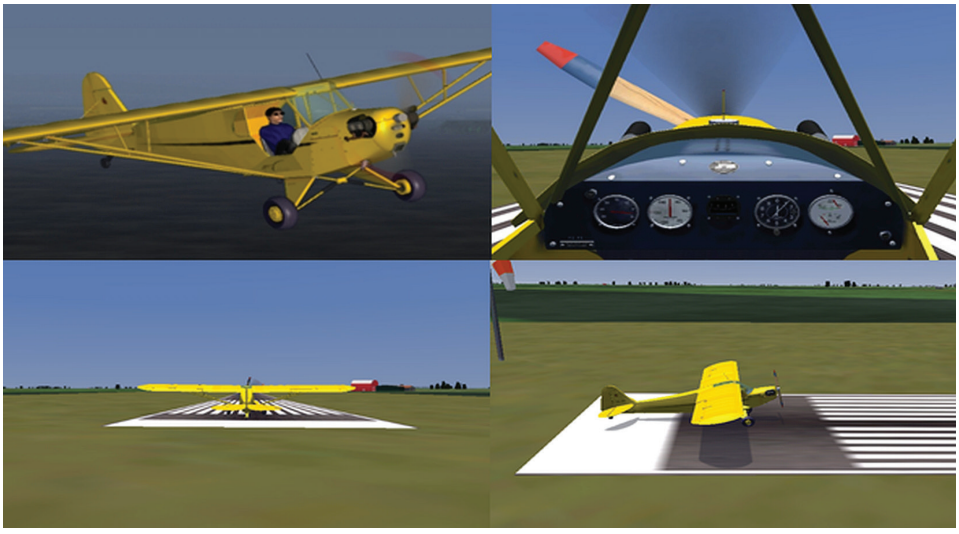


*Figura 9.* Vistas de la aeronave Piper J3 Cub

Elaboración propia

La figura 10 muestra el mapa de la trayectoria seguida por la aeronave Piper J3 Cub para ir desde el aeropuerto de Westley hacia la ciudad de Grayson (USA).

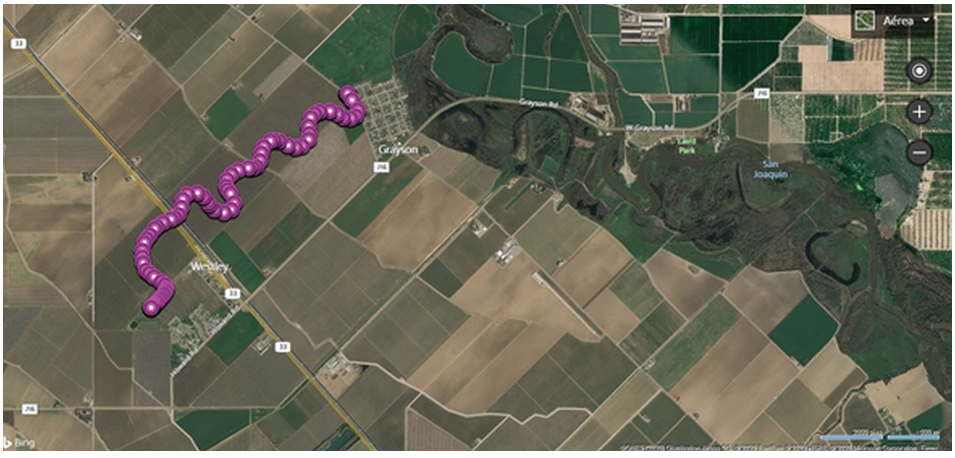



*Figura 10.* Trayectoria de la aeronave

Elaboración propia

Así mismo, utilizando la herramienta Scope, se graficaron los valores obtenidos para los ángulos Roll, Pitch, Yaw para esta trayectoria, tal como se muestra en la figura 11:

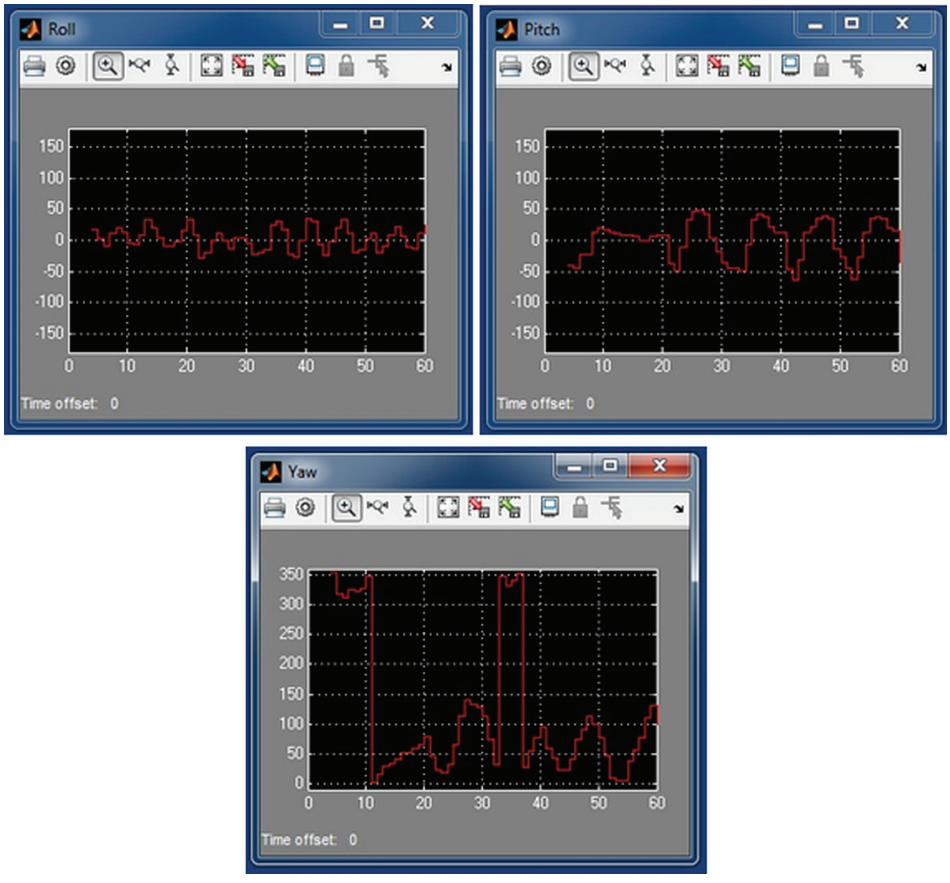



*Figura 11.* Gráficas de Roll, Pitch, Yaw

Elaboración propia

Tal como se observa en la figura 11, los ángulos Roll y Pitch han tenido variaciones menores, durante toda la trayectoria; sin embargo, el ángulo Yaw tiene variaciones drásticas cada cierto tiempo, esto se debe a la acción de control que se ejerce sobre este ángulo para orientar la aeronave hacia la dirección correcta.

Las pruebas realizadas, también están disponibles en video en el siguiente link: https://www.youtube.com/watch?v=TgjSFvtcUFE

## 6. CONCLUSIONES

- Software in the Loop ha demostrado ser una técnica válida para el diseño de sistemas complejos como *software* de aviónica, donde por limitaciones de costo, tiempo y seguridad no es posible realizar pruebas directamente sobre el sistema físico real.
- Los diagramas de componentes y secuencia son fundamentales en el proceso de diseño de *software* de sistemas complejos ya que proporcionan una vista de alto nivel de los módulos y componentes del sistema. Así mismo, nos permiten conocer cómo interactúan entre sí los diversos componentes a través del tiempo para completar la funcionalidad deseada.
- El ciclo de vida iterativo e incremental, se ajusta adecuadamente a sistemas con muchos módulos interactuando entre sí, ya que nos permite aplicar el proceso de desarrollo a cada componente individual y de esta manera ir agregando funcionalidades de forma gradual. Sin embargo, para que cada nuevo módulo sea validado será necesario aplicar pruebas de integración y pruebas de regresión con el resto del sistema, generando al final de este proceso una nueva versión.
- Para especificar la dinámica de vuelo de una aeronave de ala fija se necesita un sistema de referencia relativo a la misma aeronave, donde normalmente se miden los ángulos Roll, Pitch y Yaw, así como un sistema de referencia relativo a tierra de tal manera que se pueda determinar la posición de la aeronave en cualquier punto en términos de latitud y longitud.
- A pesar de que una aeronave de ala fija se considera un sistema no lineal, la estrategia de control PID brinda una aproximación aceptable para controlar la dinámica de vuelo de la aeronave durante las etapas de despegue, vuelo crucero y aterrizaje.
- El Modelo Dinámico de Vuelo (FDM) de una aeronave es el componente fundamental en un simulador, ya que al controlar la exactitud con que las fuerzas involucradas son representadas, determina la fiabilidad que tendrá el sistema cuando sea aplicado al sistema físico real.
- Simulink ofrece una plataforma confiable para comunicaciones en tiempo real con aplicaciones externas basadas en el Protocolo de Datagramas de Usuario.
- La fórmula Haversiana es una técnica precisa para calcular el arco de distancia entre dos puntos de la tierra. Así mismo, para determinar la correcta orientación de una aeronave hacia un destino, el cálculo del ángulo Bearing será necesario.

## REFERENCIAS

Aboelela, M. A. S., Ahmed, M. F., y Dorrah, H. T. (2012). Design of aerospace control systems using fractional PID controller. *Journal of Advanced Research*, *3*(3), 225–232. http://doi.org/10.1016/j.jare.2011.07.003

Akyürek, Ş., Kürkçü, B., Kaynak, Ü., y Kasnakoğlu, C. (2016). Control Loss Recovery Autopilot Design for Fixed-Wing Aircraft. *IFAC-PapersOnLine*, *49*(9), 117-123. http://doi.org/10.1016/j.ifacol.2016.07.509

Astrom, K. J. (2002). PID Control. *Control System Design*, 216–251. Recuperado de https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf

Attya, S. M., y Abdulla, A. I. (2018). PID Controller Design and Simulation for Aircraft Roll Control Based on Evolutionary Technique Using MATLAB, JET, *8*(3), 5-9. Recuperado de http://doi.org/10.17605/OSF.IO/UT8FB

Bansal, H. O. (2009). Tuning of PID Controllers using Simulink. *International Journal of Mathematical Modeling, Simulations and Applications*, *2*(3), 337–344. Recuperado de https://www.researchgate.net/publication/268802558_Tuning_of_PID_Controllers_using_Simulink

Coopmans, C., Podhradský, M., y Hoffer, N. V. (2016). Software- and hardware-in-the-loop verification of flight dynamics model and flight control simulation of a fixed-wing unmanned aerial vehicle. *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS 2015*, 115-122. http://doi.org/10.1109/RED-UAS.2015.7440998

Cova, W. J. D. (2005). Control PID, un Enfoque Descriptivo. *Universidad Tecnológica Nacional, Facultad Regional La Rioja, Departamento de Electrónica*. Recuperado de http://www.frlr.utn.edu.ar/archivos/alumnos/electronica/catedras/38-sistemas-de-control-aplicado/Publicaciones/Control_PID_Enfoque_Descriptivo.pdf

De Castro, D. F., y dos Santos, D. A. (2016). A software-in-the-loop simulation scheme for position formation flight of multicopters. *Journal of Aerospace Technology and Management*, *8*(4), 431–440. http://doi.org/10.5028/jatm.v8i4.612

Ellingsen, G., y McLain, T. (2017). ROSplane: Fixed-wing autopilot for education and research. *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017*, 1503–1507. http://doi.org/10.1109/ICUAS.2017.7991397

FlightGear. (2020). FlightGEar Flight Simulator. Recuperado de https://www.flightgear.org/

Gouthami, E., y Rani, M. A. (2016). Modeling of an Adaptive Controller for an Aircraft Roll Control System using PID, Fuzzy-PID and Genetic Algorithm, *11*(1), 15-24. http://doi.org/10.9790/2834-11121524

Hartanto, S., Furqan, M., Putera, A., Siahaan, U., y Fitriani, W. (2017). Haversine Method in Looking for the Nearest Masjid. *International Journal of Engineering Research*, (agosto). http://doi.org/10.23883/IJRTER.2017.3402.PD61H

Islam, M. T., Alam, M. S., Laskar, M. A. R., y Garg, A. (2016). Modeling and simulation of longitudinal autopilot for general aviation aircraft. *2016 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016*, (diciembre del 2017), 490–495. http://doi.org/10.1109/ICIEV.2016.7760051

Jia, Y.-B. (2020). Rotation in the Space. *Iowa State University*, 1-14. Recuperado de http://web.cs.iastate.edu/~cs577/handouts/rotation.pdf

Khalid, A., Zeb, K., y Haider, A. (2019). Conventional PID, adaptive PID, and sliding mode controllers design for aircraft pitch control. *2019 International Conference on Engineering and Emerging Technologies, ICEET 2019*, (julio), 1-6. http://doi.org/10.1109/CEET1.2019.8711871

Koks, D. (2008). Using Rotations to Build Aerospace Coordinate Systems. *Electronic Warfare and Radar Division Systems Sciences Laboratory*. Recuperado de https://apps.dtic.mil/dtic/tr/fulltext/u2/a484864.pdf

Korkmaz, H., Ertin, O. B., Kasnakoğlu, C., y Kaynak, Ü. (2013). Design of a Flight Stabilizer System for a Small Fixed Wing Unmanned Aerial Vehicle using System Identification. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *1*(PART 1), 145-149. http://doi.org/10.3182/20130916-2-tr-4042.00012

Peet, M. M. (2010). Spacecraft and Aircraft Dynamics. *Illinois Institute of Technology*. Recuperado de http://control.asu.edu/Classes/MMAE441/Aircraft/441Lecture1.pdf

Qays, H. M., Jumaa, B. A., y Salman, A. D. (2019). Design and Implementation of Autonomous Quadcopter using SITL Simulator. *Iraqi Journals of Computers, Communications, Control & Systems Engineering*, *20*(1), 1-16. http://doi.org/10.33103/uot.ijccce.20.1.1

R., R., M., C., S., C., Kumar, P., y N., P. (2020). PID Controller Design for Dynamic Motion of an Aircraft. *SSRN Electronic Journal*, (mayo), 859-862. http://doi.org/10.2139/ssrn.3511417

Redshift Labs. (2020). Understanding Euler Angles. *Chrobotics*. Recuperado de http://www.chrobotics.com/library/understanding-euler-angles

Slabaugh, G. G. (2017). Computing Euler Angles from a Rotation Matrix. *Digital Environment Research Institute (DERI) Queen Mary University of London*, 1-7. Recuperado de https://www.gregslabaugh.net/publications/euler.pdf

Sudha, G., y Deepa, S. N. (2016). Optimization for PID Control Parameters on Pitch Control of Aircraft Dynamics Based on Tuning Methods. *Applied Mathematics and Information Sciences*, *10*(1), 343-350. http://doi.org/10.18576/amis/100136

Surowski, D. (2011). Distance between Points on the Earth's Surface. *Kansas State University, Department of Mathematics*. Recuperado de https://www.math.ksu.edu/~dbski/writings/haversine.pdf

Wahid, N., Hassan, N., Rahmat, M. F., & Mansor, S. (2011). Application of Intelligent Controller in Feedback Control Loop for Aircraft Pitch Control. *Australian Journal of Basic and Applied Sciences*, *5*(12), 1065–1074. Recuperado de http://www.ajbasweb.com/old/ajbas/2011/December-2011/1065-1074.pdf

White, M. (2019). PID Control for Robotics. *Programming.* Recuperado de https://mjwhite8119.github.io/Robots/pid-control

Recibido: 5/10/2020 / Aceptado: 10/11/2020
doi: https://doi.org/10.26439/interfases2020.n013.4876

# Modelo Balanced Scorecard para los controles críticos de seguridad informática según el Center for Internet Security (CIS)

William-Rogelio Marchand-Niño
william.marchand@unas.edu.pe / ORCID: 0000-0003-2650-4226

Edwin Jesús Vega Ventocilla
edwin.vega@unas.edu.pe / ORCID: 0000-0002-3628-9016

Universidad Nacional Agraria de la Selva, Tingo María, Perú

## Resumen

En diversos sectores de las actividades humanas, las organizaciones están adoptando con mayor intensidad las tecnologías de la información (TI). De este modo, exponen datos sensibles y confidenciales de empleados y clientes, lo cual genera que las entidades públicas y privadas desarrollen normas y regulaciones para proteger estos activos y asegurar su confidencialidad, integridad y disponibilidad. Como resultado del estudio, se formula un modelo de Cuadro de Mando Integral que vincula a los controles críticos de seguridad del CIS, soportado además por un aplicativo de ofimática como una herramienta preliminar que facilite la presentación de resultados. Dichos resultados resaltan que sobre la aplicación preliminar que se dio en cinco instituciones, la mayor proporción (80 %) está de acuerdo con el modelo propuesto y su utilidad para el monitoreo y gestión de los controles de seguridad.

PALABRAS CLAVE: cumplimiento / seguridad y privacidad / modelamiento organizacional

## Abstract

### BALANCED SCORECARD MODEL FOR CRITICAL COMPUTER SECURITY CONTROLS ACCORDING TO THE CENTER FOR INTERNET SECURITY (CIS)

In different sectors of human activities, organizations are adopting information technology (IT) more intensively, exposing sensitive and confidential information of employees and customers. This situation makes public and private entities to develop standards and regulations to protect these information assets, ensuring confidentiality, integrity and availability. As a result of the study, a Balanced Scorecard model that links the critical security controls of the CIS is formulated and supported by an office IT application as a preliminary tool that facilitates the presentation of the results. Such results highlight that the highest proportion (80%) of the preliminary application that occurred in five institutions agrees with the proposed model and its usefulness for monitoring and managing security controls.

KEYWORDS: compliance / security and privacy / organizational modeling

## 1. INTRODUCCIÓN

El hecho de que las organizaciones estén adoptando con más intensidad las tecnologías de la información (TI) y aspectos de seguridad, se está convirtiendo en algo más relevante porque se expone información sensible y personal en las soluciones tecnológicas como sitios web, aplicaciones de escritorio, aplicaciones web, aplicaciones móviles, y aplicaciones para internet de las cosas (IoT). Del mismo modo, las regulaciones gubernamentales y estándares relacionados se han estado desarrollando, quizá no a la velocidad del desarrollo tecnológico, pero sí tratando de cubrir la mayor parte de aspectos como el financiero, salud, contable, datos personales, etcétera. Sobre esto mencionan por ejemplo que en Estados Unidos se tiene la Ley SOX para prevenir fraudes contables y financieros incluyendo los registros que son creados y mantenidos con TI (Herath, T., Herath, H. y Bremser., 2010), la Ley de Responsabilidad y Portabilidad del Seguro de Salud (HIPPA) para información relacionada a la salud de las personas y otras regulaciones internacionales como International Financial Reporting Standards (IRFS). Asimismo, el sector bancario cuenta con el Payment Card Industry Data Security Standard (PCI DSS) (PCI Security Standards Council, 2016) para la administración de tarjetas de crédito. También los conocidos estándares de la familia ISO/IEC 27000 y de NIST (National Institute of Standards and Technology).

Una de las dificultades de estos estándares y regulaciones es el número y diversidad de controles que contienen; por ejemplo, la familia ISO/IEC 27000 (ISO/IEC, 2013) que define 114 controles, el NIST (National Institute of Standards and Technology, 2014) que define 444 controles de seguridad. Y en un intento de resumirlos, el Centro para la Seguridad de Internet (CIS, 2018) (CIS, Center for Internet Security) ha establecido los veinte controles críticos de seguridad. Asimismo, mantiene manuales denominados CIS Benchmarks orientados a plataformas específicas como sistemas operativos, dispositivos de red, motores de bases de datos, entre otros, cuyo volumen de controles e indicadores pueden superar fácilmente la centena.

Considerando la cantidad de controles que se deben aplicar y evaluar, además de las exigencias por las regulaciones o estándares, las organizaciones están frente a la tarea complicada de realizar el seguimiento, monitoreo y evaluación de los controles de seguridad de forma efectiva y eficiente.

Ante ese escenario, realizar un adecuado seguimiento y control de los indicadores de seguridad definidos en una organización es sumamente importante, sea por cumplimiento regulatorio o no. Por otro lado, hay propuestas de Balanced Scorecard (BSC) para seguridad informática, como la propuesta por DeLooze (2006), bajo los mismos principios del BSC tradicional como aquella que define cuatro grupos de *stakeholders* para un programa de seguridad y su aseguramiento. Los cuatro grupos propuestos son: usuarios, administradores, administradores de sistemas o dueños de los sistemas, y los

auditores o reguladores (DeLooze, 2006). Según el autor de la mencionada propuesta, el enfoque propuesto permite responder preguntas del tipo "¿cómo ve nuestro programa de seguridad a nuestros usuarios?", "¿cómo ve nuestro programa de seguridad a los propietarios del sistema?", "¿cómo ve nuestro programa de seguridad a nuestros administradores de sistemas? y "¿cómo ve nuestro programa de seguridad a los auditores?".

La adaptación de un modelo de BSC para seguridad de la información también fue abordada por algunos autores, quienes le dan un enfoque estratégico a la seguridad afirmando, por ejemplo, que uno de los aspectos importantes respecto a la perspectiva de valor para el negocio es proteger la reputación y generar confianza. Y esto se puede lograr con una efectiva aplicación de controles y el seguimiento a estos (Groš, 2019). Para efectos de esta investigación se consideran los controles CIS por la disponibilidad de la documentación detallada de forma libre, además de una oportunidad de profundizar el análisis de la aplicación de estos controles y sus beneficios. Por lo tanto, la pregunta de investigación se define de la siguiente manera: ¿de qué forma el Cuadro de Mando Integral se adecúa para el monitoreo del cumplimiento de los veinte controles críticos de seguridad del Center for Internet Security? El objetivo es, por lo tanto, determinar la manera en que el Cuadro de Mando Integral constituye una plataforma apropiada para el monitoreo del cumplimiento de los veinte controles críticos de seguridad propuestos por el CIS.

La hipótesis sobre la cual gira la investigación está formulada del siguiente modo: "Un modelo de Balanced Scorecard para seguridad informática ofrece una estructura de monitoreo efectivo y eficiente al cumplimiento de los veinte controles críticos de seguridad del Center for Internet Security". La investigación se justifica técnicamente porque se aplicarán conceptos sobre controles de seguridad informática y el Cuadro de Mando Integral (Balanced Scorecard). Desde el punto de vista organizacional, la investigación proporciona una herramienta a nivel estratégico para el monitoreo de los controles de seguridad informática que se deben implementar por exigencias regulatorias o por política institucional. La principal característica y contribución de esta investigación es la de ofrecer una alternativa que integre las mediciones operativas con la herramienta de gestión estratégica en términos de seguridad.

El artículo se compone de seis secciones siendo esta introducción la primera de ellas. En la sección 2 se realiza la revisión de la literatura que considera los principales antecedentes y principios. En la sección 3 se detalla la metodología para la construcción de la propuesta del modelo de BSC adaptada al monitoreo de los controles CIS, mientras que la sección 4 está dedicada a la presentación y discusión de los resultados obtenidos en la fase de evaluación. En la sección 5 se brindan las conclusiones del trabajo y, finalmente, en la sección 6 se describen las posibles líneas de trabajo futuro.

## 2. REVISIÓN DE LITERATURA

### 2.1 Cuadro de Mando Integral (Balanced Scorecard)

El Balanced Scorecard o Cuadro de Mando Integral (CMI), desarrollado por Kaplan y Norton en 1992, es más que solo un instrumento de medición. Es un sistema de gestión estratégica con una visibilidad y comprensión de los objetivos y métodos para alcanzarlos. Esto implica que se deben traducir en indicadores que reflejen la evaluación del desempeño de las estrategias implementadas (R. Kaplan y Norton, 2002, R. S. Kaplan y Norton, 2005).

El Cuadro de Mando Integral (CMI) orienta su uso a la alineación de indicadores financieros y no financieros para la gestión y control del rendimiento de las organizaciones (Caudle, 2008, R. S. Kaplan y Norton, 1996). Además, el CMI está pensado para la gestión de las estrategias formuladas en un plan estratégico institucional y el soporte para la materialización de los inductores y acciones que lo contienen (Marchand-Niño, 2013). En la figura 1 se pueden observar las cuatro perspectivas clásicas del CMI: financiera, cliente, procesos internos y la perspectiva de aprendizaje y crecimiento.

La evolución de los procesos en las organizaciones hace que se adopten las tecnologías de la información (TI) que suman complejidad, y la necesidad de desarrollar nuevos enfoques para el seguimiento y control del rendimiento organizacional. Sin embargo, el CMI también resulta útil para tales propósitos, es así que se formula un Balanced Scorecard (BSC) adaptado para sistemas de información (SI) (Martinsons, Davison y Tse, 1999). Las perspectivas consideradas en este CMI son orientación al usuario, valor para el negocio, procesos internos y preparación futura (ver figura 2). Como se puede deducir, los sistemas de información son un componente interno, por lo que las perspectivas consideradas en el CMI también son de carácter interno. Sin embargo, no pierde el enfoque estratégico, puesto que los sistemas de información también deben estar alineados a la misión y visión del negocio.

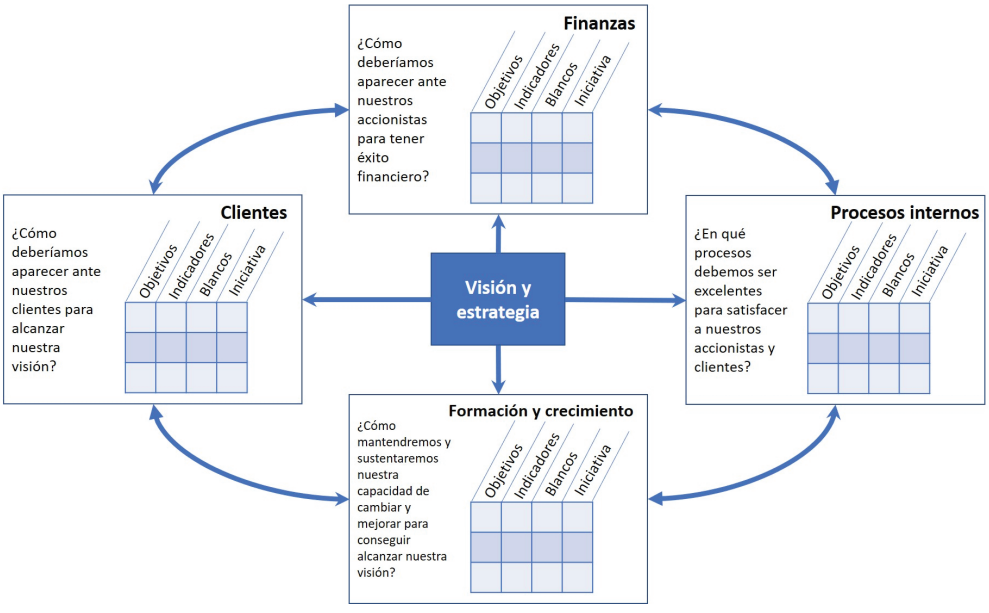



*Figura 1.* Perspectivas originales del Balanced Scorecard

Fuente: R. Kaplan y Norton, 2002

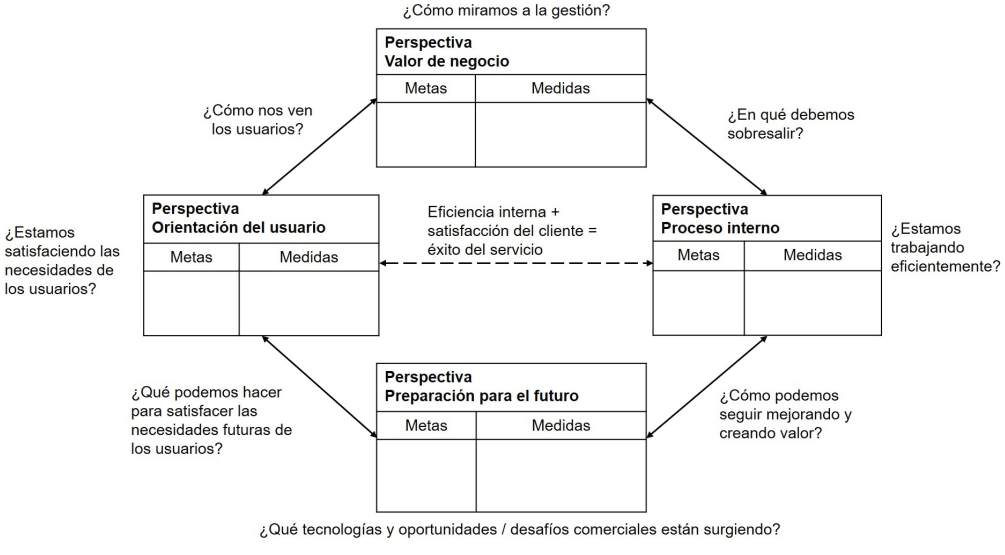



*Figura 2.* Perspectivas del Balanced Scorecard para sistemas de información

Fuente: Martinsons, Davison y Tse, 1999

Los administradores de TI buscan herramientas que les faciliten alinear los recursos de las tecnologías de información con el negocio y realizar un control de cómo estos contribuyen al soporte y rendimiento de la organización (Keyes, 2005). Una de esas herramientas de alto valor es el Cuadro de Mando Integral que debe poseer los atributos, como sencillez de presentación, enlaces explícitos a la estrategia de TI, amplio compromiso ejecutivo, definiciones de métricas estándar de empresa, capacidad de desglose y contexto disponible. Además, las métricas que se deben considerar se agrupan en siete categorías: rendimiento financiero, rendimiento del proyecto, rendimiento operacional, gestión del talento, satisfacción del usuario, iniciativas empresariales y seguridad de la información. Se evidencia que las métricas relacionadas con la seguridad de la información hacen su aparición en estos tipos de Balanced Scorecard.

Otro trabajo importante que propone un CMI para TI es el desarrollado con perspectivas consideradas que son similares a las de Martinsons (1999), tal como se muestran en la figura 3. Estas perspectivas son: orientación al usuario, excelencia operacional, contribución empresarial y orientación al futuro (Grembergen, 2005).

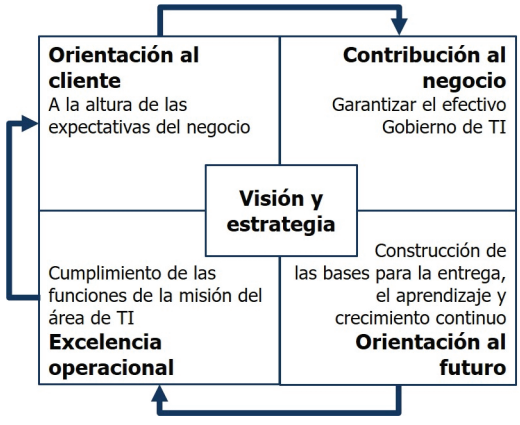



*Figura 3.* BSC para TI

Fuente: Grembergen, 2005

Como se puede observar en la figura 3, la adaptación de Grembergen (2005) establece la correspondencia con las cuatro perspectivas originales, considerando, además, que el componente de TI es interno en una organización y desde esa visión la perspectiva financiera pasa a conceptuarse como el aporte que realizan las tecnologías de la información al negocio en términos de eficiencia que pueden repercutir en aspectos financieros (ahorros, optimización de costos, etc.); la perspectiva original del cliente se transforma en perspectiva hacia el usuario de las TI. La perspectiva de procesos

internos se orienta a la excelencia operacional en el sentido de cómo las tecnologías de la información soportan el cumplimiento de las funciones del área de TI alineadas a los objetivos estratégicos de la organización; y finalmente la perspectiva de aprendizaje y crecimiento denominada en el BSC para TI como orientación al futuro es un concepto que para abordar las acciones de aprendizaje continuo de acuerdo con el crecimiento y madurez organizacional.

### 2.2 Controles de seguridad

Para efectos de la investigación, los conceptos y teoría están asociados a la seguridad de la información y seguridad informática.

El NIST en su publicación especial 800-53, define que un control de seguridad es una "salvaguarda o contramedida para proteger la confidencialidad, integridad y disponibilidad de la información de las organizaciones" (National Institute of Standards and Technology, 2014) functions, image, and reputation. Estos controles son necesarios para satisfacer los requerimientos de seguridad definidos por las entidades con el fin de mitigar los riesgos asociados a los activos de información. Definición parecida la realiza el Committee on National Security Systems (CNSS), una fuente autorizada de definiciones en los Estados Unidos que establece como controles de seguridad a aquellos "controles de gestión, operativos y técnicos (es decir, salvaguardas o contramedidas) prescritos para un sistema de información para proteger la confidencialidad, integridad y disponibilidad del sistema y su información" (CNSS, 2015).

Aunque en textos de otros estándares no se describe explícitamente el concepto de control de seguridad, la acepción definida por el CNSS será la que predomine en el desarrollo del proceso de investigación; además es útil ampliar el concepto en el sentido que detallar si existe alguna clasificación para los controles de seguridad y cuáles son las fuentes de datos para realizar el seguimiento o monitorización.

Los controles de seguridad se pueden clasificar en tres tipos:

- **Controles de administración**, acciones tomadas para administrar el desarrollo, mantenimiento y uso de los sistemas; por ejemplo, políticas y procedimientos.
- **Controles operativos**, mecanismos y procedimientos cotidianos utilizados para proteger los sistemas operacionales y su entorno; por ejemplo, formación de conciencia, la gestión de la configuración y la respuesta a incidentes.
- **Controles técnicos**, controles de *hardware / software* utilizados para proteger los sistemas de TI y la información que se almacena, procesa o transmite. Por ejemplo, los controles de acceso, los mecanismos de autenticación y el cifrado (Johnson, 2015).

La cantidad de controles definidos por diversos estándares y entidades reguladoras en el mundo puede convertirse en un problema para la implementación o adopción por las organizaciones, tal es así por ejemplo, que la basada en la ISO/IEC 27001:2013 (Indecopi, 2014) incluye 114 controles agrupados en 35 objetivos de control y 14 dominios, la norma NIST 800-53 revisión 4 (National Institute of Standards and Technology, 2014) establece 444 controles agrupados en 18 familias y el Centro para la Seguridad de Internet (CIS, Center for Internet Security) define los 20 controles críticos de seguridad con 171 subcontroles. Tomando en cuenta el artículo *SIEM-based framework for security controls automation* (Montesino, Fenz y Baluja, 2012), que postula la posibilidad de automatizar algunos controles, se puede facilitar su evaluación. Estos controles de seguridad susceptibles de automatizar son:

- Inventario de activos (*hardware* y *software*)
- Gestión de cuentas
- Gestión de *logs*
- Monitoreo de sistemas
- Protección contra *malware*
- Gestión de actualizaciones y escaneo de vulnerabilidades
- Verificación del cumplimiento y evaluación de seguridad
- *Backup* de información
- Seguridad física
- Gestión de incidentes

Entiéndase por "automatizar controles" al uso de herramientas que permitan recopilar datos en tiempo real o diferido con intervención humana mínima; herramientas como IDS/IPS, sensores, *sniffers*, *software* especializado, SIEM como le desarrollado por Splunk (Splunk Enterprise Security) (Splunk, 2020) o el desarrollado por IBM (IBM QRadar) (IBM, 2020), o como las herramientas *open source,* como el *framework* OSSIM (Our Open Source SIEM) (AT&T Cybersecurity, 2020).

Asimismo, se definen las evaluaciones de los controles de seguridad como "las pruebas o evaluaciones de los controles de seguridad de gestión, operacionales y técnicos para determinar la medida en que los controles se implementan correctamente, operan según lo previsto, y si realmente están produciendo el resultado deseado con respecto al cumplimiento de los requisitos de seguridad para un sistema de información u organización" (CNSS, 2015).

### 2.3 Evaluación de controles CIS

Entre algunos trabajos relacionados con la evaluación y monitoreo de los controles críticos de CIS se mencionan los siguientes:

Una forma de automatizar controles de CIS respecto a *firewall* de Palo Alto, en la que se propuso una metodología para diseñar una herramienta que automatizara la verificación de los controles CIS en los dispositivos de red de Palo Alto Networks. Algunos de los resultados son correspondientes al consumo de tiempo para las inspecciones manuales y automatizadas, y la carga de trabajo que eso representa para el personal involucrado en ese tipo de servicios (Perminov, Kosachenko, Konev, y Shelupanov, 2020). Otro factor revelado es sobre la capacidad de rastreo y detección de violaciones de seguridad en estos dispositivos lo que permite un tiempo de respuesta más eficaz.

Otro análisis que se realiza sobre los controles de CIS hace referencia a la existencia de múltiples estándares y modelos de controles para la seguridad de la información como ISO, NIST, entre otros, donde CIS nace como una alternativa más práctica y con un número de controles que sean manejables. Sin embargo, con el tiempo se ha convertido también en una solución parecida a las mencionadas sobre gestión de riesgos, aunque con menos controles, pero presentando un catálogo tal cual las demás soluciones (Groš, 2019). La crítica que se plantea es que el CIS de SANS (SysAdmin Audit, Networking and Security Institute) no se convierta finalmente en una opción más, que se realicen análisis más profundos con el propósito de mejorar este modelo de controles.

### 2.4 BSC para seguridad de la información

La aplicación adecuada del concepto de Balanced Scorecard puede contribuir a mejorar la gestión orientada a las tecnologías de la información y comunicación otorgando un enfoque estratégico a la seguridad. Uno de los aspectos importantes respecto a la perspectiva de valor para el negocio es el proteger la reputación y generar confianza. La perspectiva de los *stakeholders* se orienta hacia el comportamiento de los empleados que puede repercutir en la seguridad en sí misma así como en las necesidades de seguridad de estos que deben ser cubiertas y controladas (Herath *et al.*, 2010). En la perspectiva de procesos internos, los autores plantean que se deben medir de forma similar a otras aplicaciones de TI, considerando tres procesos generales: (1) la planificación y priorización de iniciativas de seguridad, (2) la implementación de servicios y productos de seguridad y (3) las operaciones y mantenimiento de los servicios de seguridad. Finalmente, en la perspectiva de preparación para el futuro, se afirma que un aspecto importante sería la capacitación continua del personal de seguridad de TI y los usuarios sobre diferentes tipos de amenazas y sus formas de evitarlas. En la figura 4 se muestra el modelo propuesto para seguridad de información.

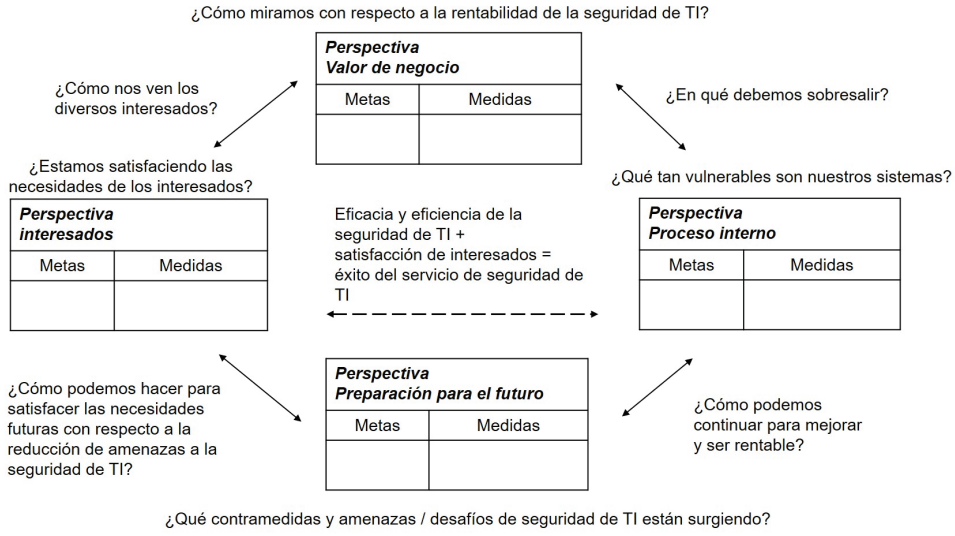



*Figura 4.* Modelo Balanced Scorecard para seguridad de la información

Fuente: Herath, T., Herath, H. y Bremser, 2010

Una de las preguntas que se formulan en trabajos como el de Herath (2010) es "¿cuáles son las medidas comunes que se utilizan en la seguridad de TI y cómo han cambiado en los últimos años?" Y como parte de una posible respuesta a esa pregunta, este trabajo determinará la forma de incluir los indicadores considerados en estándares difundidos ampliamente en la industria como son los veinte controles del CIS (Center for Internet Security).

## 3. METODOLOGÍA

En una primera etapa se adaptó el BSC original a un BSC para seguridad informática y posteriormente se realizó un mapeo de los controles del CIS con los cuadrantes del tablero de mando integral.

### 3.1 BSC para seguridad informática

En esta fase se desarrolla la propuesta del tablero de mando integral (BSC) para seguridad informática, en la cual se considera la analogía presentada en la figura 5.

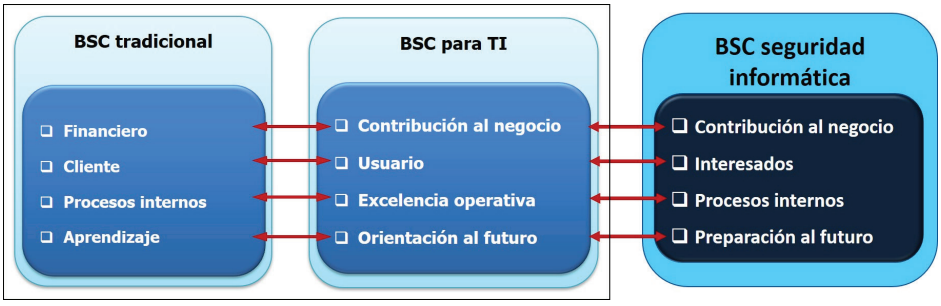



*Figura 5.* Correspondencia entre el BSC tradicional, BSC para TI y el BSC para seguridad informática

Elaboración propia

El componente de contribución al negocio tiene como función agregar valor al mismo negocio y valor a la función de seguridad de TI. Los interesados deben recibir los servicios de seguridad adecuados en función de sus roles y responsabilidades.

Los procesos internos están orientados a entregar productos y servicios de seguridad considerando aspectos de costos, eficiencia y recursos. Y finalmente, el componente de preparación al futuro implica un proceso de mejora continua en seguridad informática que permita hacer frente a los desafíos (vulnerabilidades, tipos de ataques, etc.) del futuro.

## 3.2 CONTROLES CIS

Los 20 controles críticos de CIS abordan los aspectos básicos, técnicos y de gestión de la seguridad informática y pueden tener una analogía con otros *frameworks* o estándares como ISO/IEC 27002 y NIST 800-53.

Para el estudio es necesario establecer qué controles se pueden automatizar para que sirvan como entrada para el tablero de mando integral de seguridad informática. Como referencia inicial se considera que pueden automatizar ciertos controles de NIST 800.53 (Montesino *et al.*, 2012), tal como se muestra en la figura 2. Si se toma esta información, se puede realizar un *benchmark* con los controles de CIS y determinar qué controles específicos pueden automatizarse.

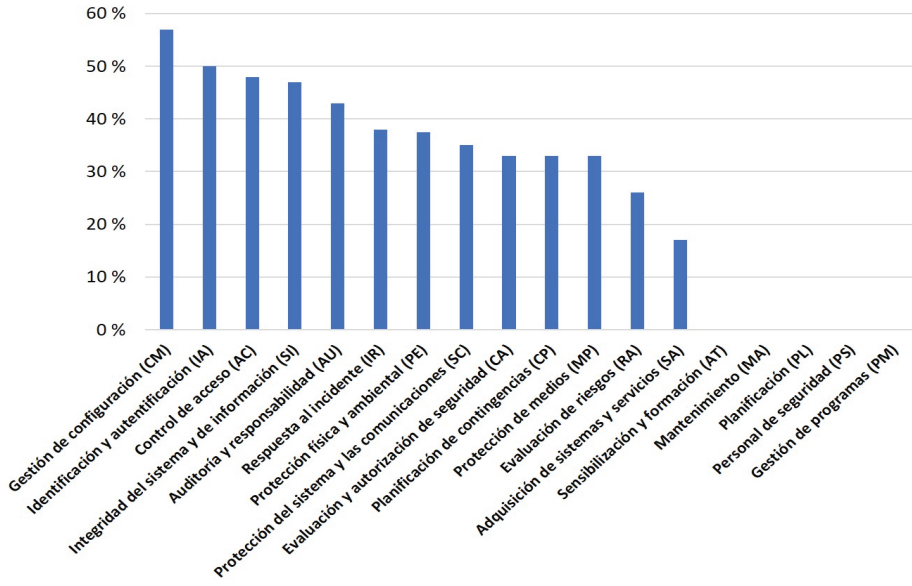



*Figura 6.* Porcentaje de controles automatizables en las diferentes familias de NIST SP 800-53

Fuente: Montesino, Fenz y Baluja, 2012

Tomando como referencia la figura 6 y el análisis comparativo se determinan los controles que son susceptibles de ser automatizados en la información de entrada, es decir, que para efectos de la alimentación de datos en el tablero de mando integral por cada indicador se puede hacer uso de herramientas cuyos reportes o datos pueden ser transferidos de forma automática. Los controles CIS que pueden ser automatizados se muestran en la tabla 1. Cabe recalcar que para el proceso de automatización se pueden considerar las herramientas mencionadas en la descripción de los controles de seguridad. Para la aplicación en cada caso se debe elaborar la declaración de aplicabilidad considerando la pertinencia y capacidades organizacionales.

Tabla 1
*Controles de CIS que pueden ser automatizados*

| Control | Subcontrol | |
|---|---|---|
| [01] Inventario y control de activos de *hardware* | 01.1 | Utilizar una herramienta de descubrimiento activo |
| | 01.2 | Utilizar una herramienta de descubrimiento pasivo de activos |
| | 01.3 | Utilizar DHCP Logging para actualizar el inventario de activos |
| | 01.4 | Mantener un inventario de activos detallado |
| | 01.5 | Mantener la información del inventario de activos |

(*continúa*)

(*continuación*)

| Control | Subcontrol |
|---|---|
| [02] Inventario y control de activos *software* | 02.1 Mantener un inventario de *software* autorizado<br>02.3 Utilizar herramientas de inventario de *software*<br>02.4 Rastrear información del inventario de *software*<br>02.5 Integrar los inventarios de activos de *hardware* y *software*<br>02.7 Utilizar lista blanca de aplicaciones |
| [03] Gestión continua de vulnerabilidades | 03.1 Ejecutar herramientas de escaneo automatizados de vulnerabilidades |
| [04] Uso controlado de privilegios administrativos | 04.1 Mantener un inventario de cuentas administrativas<br>04.4 Usar contraseñas únicas |
| [05] Configuración segura para *hardware* y *software* en dispositivos móviles, computadoras portátiles, estaciones de trabajo y servidores | 05.2 Mantener imágenes seguras |
| [07] Protección de correo electrónico y navegador web | 07.1 Asegurar el uso de navegadores y clientes de correo electrónico que cuenten con soporte |
| [08] Defensa contra *malware* | 08.1 Utilizar *software antimalware* de gestión centralizada<br>08.2 Asegurar que el *software antimalware* y las firmas estén actualizadas |
| [12] Defensa de borde | 12.11 Requerir autenticación multifactor en todos los inicios de sesión remotos<br>12.12 Gestionar todos los dispositivos remotos que se conectan a la red interna |
| [13] Protección de datos | 13.1 Mantener un inventario de información sensible<br>13.3 Monitorear y bloquear el tráfico de red no autorizado<br>13.5 Monitorear y detectar cualquier uso no autorizado de cifrado |
| [14] Control de acceso basado en la necesidad de conocer protección de datos | 14.5 Utilizar una herramienta de descubrimiento activo para identificar datos sensibles |
| [15] Control de acceso inalámbrico | 15.1 Mantener un inventario de puntos de acceso inalámbrico autorizados<br>15.2 Detectar puntos de acceso inalámbricos conectados a la red cableada |
| [16] Monitoreo y control de cuentas | 16.1 Mantener un inventario de sistemas de autenticación<br>16.2 Configurar un punto de autenticación centralizado<br>16.3 Requerir autenticación multifactor<br>16.6 Mantener un inventario de cuentas<br>16.12 Monitorear los intentos de acceso a cuentas desactivadas<br>16.13 Alertar sobre desviación de comportamiento de inicio de sesión de cuentas |
| [18] Seguridad del *software* de aplicación | 18.3 Verificar que el *software* adquirido aún tiene soporte<br>18.9 Sistemas separados de producción y no producción |

Elaboración propia

## 4. RESULTADOS

Para la formulación del modelo de evaluación de controles se realizó la alineación del Balanced Scorecard (BSC) con los controles de CIS, quedando distribuido tal como se muestra en la figura 7. Los 20 controles CIS han sido distribuidos en cada cuadrante del Balanced Scorecard predominando en procesos internos la mayor cantidad de controles (ocho en total), mientras que en el otro extremo el cuadrante Preparación para el futuro solo incluye el control 17 de CIS ("Implementar un programa de concienciación y entrenamiento de seguridad"); del mismo modo, los cuadrantes Interesados y Contribución al negocio tienen 5 y 6 controles CIS respectivamente.


*Figura 7.* Modelo de BSC con los controles CIS

Elaboración propia

Cada control CIS está compuesto por subcontroles que determinan los indicadores a monitorizar y medir. Estos indicadores poseen unidades propuestas por el modelo, pero cuyos valores deben ser ajustados a cada caso donde se aplique el modelo.

Los controles CIS se agrupan en tres dominios: básicos, fundamentales y organizacionales, que en la tabla 2 se observa en un grado de correspondencia entre los controles de cada grupo o dominio de CIS con los controles considerados en los cuadrantes del tablero de mando. De esta forma, se evidencia que los considerados en el cuadrante Procesos internos corresponde a la mayoría de los controles básicos y algunos del grupo de fundamentales de CIS; del mismo modo, los controles considerados en el cuadrante de Interesados corresponden en su mayoría a los controles fundamentales de CIS; y finalmente los controles de los cuadrantes Contribución al negocio y Preparación para el futuro corresponden mayoritariamente a los controles organizacionales de CIS. Esto supone una alineación lógica entre el tablero de mando integral y los dominios de CIS.

Tabla 2
*Correspondencia entre los controles CIS y los cuadrantes del BSC*

| Cuadrantes de BSC | Dominios de controles CIS |
|---|---|
| **Procesos internos**<br>01.0 Inventario y control de activos de *hardware*<br>02.0 Inventario y control de activos *software*<br>05.0 Configuración segura para *hardware* y software en dispositivos móviles, computadoras portátiles, estaciones de trabajo y servidores<br>06.0 Mantenimiento, monitoreo y análisis de logs de auditoría<br>08.0 Defensa contra *malware*<br>09.0 Limitación y control de puertos de red, protocolos y servicios<br>10.0 Capacidad de recuperación de datos<br>11.0 Configuración segura de los equipos de red, tales como cortafuegos, enrutadores y conmutadores. | **Básicos**<br>01.0 Inventario y control de activos de *hardware*<br>02.0 Inventario y control de activos *software*<br>03.0 Gestión continua de vulnerabilidades<br>04.0 Uso controlado de privilegios administrativos<br>05.0 Configuración segura para *hardware* y *software* en dispositivos móviles, computadoras portátiles, estaciones de trabajo y servidores<br>06.0 Mantenimiento, monitoreo y análisis de *logs* de auditoría |
| **Interesados**<br>04.0 Uso controlado de privilegios administrativos<br>07.0 Protección de correo electrónico y navegador web<br>12.0 Defensa de borde<br>15.0 Control de acceso inalámbrico<br>16.0 Monitoreo y control de cuentas | **Fundamentales**<br>07.0 Protección de correo electrónico y navegador web<br>08.0 Defensa contra *malware*<br>09.0 Limitación y control de puertos de red, protocolos y servicios<br>10.0 Capacidad de recuperación de datos<br>11.0 Configuración segura de los equipos de red, tales como cortafuegos, enrutadores y conmutadores<br>12.0 Defensa de borde<br>13.0 Protección de datos<br>14.0 Control de acceso basado en la necesidad de conocer protección de datos<br>15.0 Control de acceso inalámbrico |
| **Contribución al negocio**<br>03.0 Gestión continua de vulnerabilidades<br>13.0 Protección de datos<br>14.0 Control de acceso basado en la necesidad de conocer protección de datos<br>18.0 Seguridad del *software* de aplicación<br>19.0 Respuesta y manejo de incidentes<br>20.0 Pruebas de penetración y ejercicios de *red team* | **Organizacional**<br>16.0 Monitoreo y control de cuentas<br>17.0 Implementar un programa de concienciación y entrenamiento de seguridad<br>18.0 Seguridad del *software* de aplicación<br>19.0 Respuesta y manejo de incidentes<br>20.0 Pruebas de penetración y ejercicios de *red team* |
| **Preparación para el futuro**<br>17.0 Implementar un programa de concienciación y entrenamiento de seguridad. | |

Elaboración propia

Se generó una primera versión del tablero de mando integral cuya aplicación se realizó en cinco instituciones educativas durante el año 2019 (ver figura 8) y por medio de una encuesta de satisfacción de pudo comprobar el grado de efectividad en la evaluación de los controles de seguridad informática que corresponden a este caso, además de facilitar la visualización del estado de los mecanismos de seguridad implementados y tomar las decisiones necesarias para reforzar o corregir los niveles de seguridad en la organización.

**Balanced Scorecard - Tablero de mando integral de seguridad informática**

**Institución:** **Año:** 2019

| Cuadrantes | Control CIS | Nombre del control | Esperado | Estado actual | Detalle |
|---|---|---|---|---|---|
| Procesos internos | 01 | Inventario y control de activos de *hardware* | 1 | 0,49 | Detalle CIS01 |
| | 02 | Inventario y control de activos *software* | 1 | 0,35 | Detalle CIS02 |
| | 05 | Configuración segura para *hardware* y *software* en dispositivos móviles, computadoras portátiles, estaciones de trabajo y servidores | 1 | 0,00 | Detalle CIS05 |
| | 06 | Mantenimiento, monitoreo y análisis de logs de auditoria | 1 | 0,00 | Detalle CIS06 |
| | 08 | Defensa contra *malware* | 1 | 0,00 | Detalle CIS08 |
| | 09 | Limitación y control de puertos de red, protocolos y servicios | 1 | 0,00 | Detalle CIS09 |
| | 10 | Capacidad de recuperación de datos | 1 | 0,00 | Detalle CIS10 |
| | 11 | Configuración segura de los equipos de red, tales como cortafuegos, enrutadores y conmutadores | 1 | 0,00 | Detalle CIS11 |
| Interesados | 04 | Uso controlado de privilegios administrativos | 1 | 0,00 | Detalle CIS04 |
| | 07 | Protección de correo electrónico y navegador web | 1 | 0,00 | Detalle CIS07 |
| | 12 | Defensa de borde | 1 | 0,00 | Detalle CIS12 |
| | 15 | Control de acceso inalámbrico | 1 | 0,00 | Detalle CIS15 |
| | 16 | Monitoreo y control de cuentas | 1 | 0,00 | Detalle CIS16 |
| | 03 | Gestión continua de vulnerabilidades | 1 | 0,00 | Detalle CIS03 |

*Figura 8.* Aplicativo del Modelo de BSC con los controles CIS

Elaboración propia

En la tabla 3 se muestran los resultados de la encuesta de satisfacción de cinco instituciones de educación como casos de estudio, y de acuerdo con estos resultados se evidencia que la herramienta tiene que mejorar la calidad de las interfaces gráficas (presentación visual) y la facilidad de manejo. Sin embargo, la contribución hacia la representación de los controles CIS para el monitoreo y seguimiento de los indicadores resulta positivo, sumando a ello que, según lo aplicado, sí refleja los controles que deben ser medidos en la organización.

Tabla 3
*Encuesta de satisfacción sobre el uso de la herramienta*

| Enunciado | Totalmente de acuerdo | De acuerdo | Ni de acuerdo ni en desacuerdo | En desacuerdo | Totalmente en desacuerdo |
|---|---|---|---|---|---|
| El tablero es fácil de manejar. | | | 4 (80 %) | 1 (20 %) | |
| Los cuadrantes del tablero de mando integral reflejan los procesos de seguridad de la organización. | | 3 (60%) | 2 (40 %) | | |
| El tablero de mando integral de seguridad ayuda a realizar el seguimiento del estado de los controles de seguridad de la organización. | | 4 (80 %) | 1 (20 %) | | |
| Se pueden adaptar los controles a la realidad específica de la organización. | | 4 (80 %) | 1 (20 %) | | |

Elaboración propia

A diferencia de los modelos de BSC relacionados con la seguridad de la información, la propuesta presentada es un esfuerzo por integrar y articular los datos y mediciones desde los niveles operativos como insumos directos para la herramienta del sistema de gestión estratégica en términos de seguridad alineado a los objetivos institucionales del negocio.

La propuesta formulada en esta investigación trata de mantener una representación del Cuadro de Mando Integral en su forma esencial; no obstante, se pretende que el enfoque sea más amplio, involucrando no solo los controles definidos por el CIS, sino también nuevos controles orientados a la identificación, comprensión y proyección de las necesidades de los usuarios y organizaciones, en línea con el modelo de ciberconciencia situacional (Gutzwiller, Hunt y Lange, 2016). Asimismo, un modelo integrado de controles en seguridad puede incorporar técnicas de OSINT (inteligencia de fuentes abiertas) para el reconocimiento de la información organizacional expuesta relacionada con los controles de gestión de incidentes de seguridad que puedan derivar en mecanismos automatizados de notificación a los equipos de respuesta ante incidentes de seguridad (CSIRT, por sus siglas en inglés) haciendo uso del vocabulario para el registro de eventos y el intercambio de incidentes (VERIS, por sus siglas en inglés). Se prevé, asimismo, que los programas de entrenamiento y concienciación se articulen alrededor de la retroalimentación obtenida por una adecuada medición de los controles y tengan un efecto positivo en el proceso de gestión de seguridad de la información.

## 5. CONCLUSIONES

Los controles definidos por el CIS (Center of Internet Security) son susceptibles de ser categorizados en alguno de los cuadrantes del tablero de mando integral (Balanced Scorecard) y en general la mayoría de los controles básicos CIS se alinean con el cuadrante de Procesos internos del BSC, los controles de Fundamentales CIS se alinean con el cuadrante de Interesados del BSC y los controles Organizacionales CIS se alinean con el cuadrante de Contribución al negocio y Preparación para el futuro.

El desafío de disponer de los datos oportunamente requiere de un ordenamiento en los procedimientos para obtener, transmitir y almacenarlos. En general, los datos tienen que ser obtenidos de forma manual, por lo que puede conllevar a márgenes de error y oportunidad. Asimismo, se ha identificado qué tipo de datos se pueden automatizar en su proceso de recolección y alimentación para el tablero de mando integral de seguridad. Entre los controles que pueden automatizarse se mencionan los siguientes: inventario de activos (*hardware* y *software*), gestión de cuentas, gestión de *logs*, monitoreo de sistemas, protección contra *malware*, Gestión de actualizaciones y escaneo de vulnerabilidades, *backup* de información, y gestión de incidentes.

El modelo de Balanced Scorecard para el monitoreo de los 20 controles críticos de seguridad informática del CIS (Center for Internet Security) es un acercamiento inicial para ofrecer a las organizaciones una herramienta efectiva para el seguimiento y control de indicadores de seguridad. Para facilitar la aplicación del modelo se ha construido un aplicativo con macros junto al proceso de ingreso de datos y actualización sencilla. En vista de los resultados obtenidos, se ha validado favorablemente la aplicación de BSC y se han identificado diversas oportunidades para beneficiarse de un cuadro de control unificado en la gestión de la seguridad de la información.

## 6. TRABAJOS FUTUROS

Un aspecto importante que queda pendiente en la investigación es realizar pruebas con alimentación de datos de forma automatizada haciendo uso de herramientas existentes como los SIEM mencionados en este trabajo, de tal forma que los controles que son susceptibles de implementarse basados en estas entradas sean evaluados en términos de efectividad.

Asimismo, se debe extender el análisis respecto al alcance del modelo para cubrir aspectos de ciberconciencia situacional, inteligencia de amenazas y los procesos de intercambio de información con organismos públicos o equipos de respuesta a incidentes de seguridad. También, analizar la inclusión en el modelo para la medición de exposición de datos sensibles y confidenciales las técnicas de OSINT (inteligencia de fuentes abiertas).

Es preciso, también, desarrollar la herramienta basada en una plataforma estandarizada para el intercambio de datos (tecnologías web, bases de datos, entre otras) de tal forma que las consultas y reportes históricos sean accesibles y organizados de forma más eficiente.

## REFERENCIAS

AT&T Cybersecurity. (2020). AlienVault OSSIM. Recuperado de https://cybersecurity.att.com/products/ossim

Caudle, S. (2008). The Balanced Scorecard: A Strategic Tool in Implementing Homeland Security Strategies. *Homeland Security Affairs*, *4*(3).

CIS (Center for Internet Security). (2018). Homepage. Recuperado de https://www.cisecurity.org/

CNSS. (2015). Committee on National Security Systems (CNSS) Glossary. *CNSS Instruction*. https://doi.org/10.1016/0020-7292(88)90192-0

DeLooze, L. L. (2006). Creating a Balanced Scorecard for Computer Security. *2006 IEEE Information Assurance Workshop*, 15-18. https://doi.org/10.1109/IAW.2006.1652071

Grembergen, W. Van. (2005). *Strategies for information technology governance*. (J. Travers, M. Khosrow-Pour y A. Appicello, Eds.). Londres: Idea Group Inc. https://doi.org/10.4018/978-1-59140-140-7

Groš, S. (2019). A Critical View on CIS Controls. *Cornell University*. Recuperado de http://arxiv.org/abs/1910.01721

Gutzwiller, R. S., Hunt, S. M. y Lange, D. S. (2016). A Task Analysis toward Characterizing Cyber-Cognitive Situation Awareness (CCSA) in Cyber Defense Analysts. *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, CogSIMA 2016*, (Marzo), 14-20. https://doi.org/10.1109/COGSIMA.2016.7497780

Herath, T., Herath, H. y Bremser, W. G. (2010). Balanced Scorecard Implementation of Security Strategies: A Framework for IT Security Performance Management. *Information Systems Management*, *27*(1), 72-81. https://doi.org/10.1080/10580530903455247

IBM. (2020). Security information and event management (SIEM). Recuperado de https://www.ibm.com/security/security-intelligence

Indecopi. (2014). *Norma Técnica Peruana NTP-ISO/IEC 27001-2014. Tecnología de la Información*. Lima: Indecopi.

Industria de tarjetas de pago (PCI). Norma de seguridad de datos. Requisitos y procedimientos de evaluación de seguridad. Versión 3.2. (2016). *PCI Securiry Standards Council*. Recuperado de https://es.pcisecuritystandards.org/_onelink_/pcisecurity/en2es/minisite/en/docs/PCI_DSS_v3-2_es-LA.pdf

ISO/IEC. (2013). International Standard ISO/IEC-27002-2013. Switzerland.

Johnson, L. (2015). *Security Controls Evaluation, Testing, and Assessment Handbook*. (C. Katsaropoulos, Ed.), Security Controls Evaluation, Testing, and Assessment Handbook. Walthman: Elsevier. https://doi.org/10.1016/C2013-0-13416-2

Kaplan, R. y Norton, D. (2002). *Cuadro de Mando Integral* (The Balanced Scorecard). Barcelona: Ediciones Gestion 2000.

Kaplan, R. S. y Norton, D. P. (1996). The Balanced Scorecard: Translating Strategy Into Action. Proceedings of the IEEE. https://doi.org/10.1109/JPROC.1997.628729

Kaplan, R. S. y Norton, D. P. (2005). *Cómo utilizar el Cuadro de Mando Integral*. Barcelona: Gestión 2000.

Keyes, J. (2005). *Implementing the IT Balanced Scorecard. Auerbach Publications (first)*. Florida: Auerbach Publications. Recuperado de http://doi.wiley.com/10.1002/jcaf.20198

Marchand-Niño, W. R. (2013). Metodología de implantación del modelo Balanced Scorecard para la gestión estratégica de TIC. Caso: Universidad Nacional Agraria de la Selva. *PIRHUA-Universidad de Piura*. Recuperado de https://hdl.handle.net/11042/1842

Martinsons, M., Davison, R. y Tse, D. (1999). The Balanced Scorecard: a Foundation for the Strategic Management of Information Systems. *Decision Support Systems*, *25*(1), 71-88. https://doi.org/10.1016/S0167-9236(98)00086-4

Montesino, R., Fenz, S. y Baluja, W. (2012). SIEM-based framework for security controls automation. *Information Management & Computer Security*, *20*(4), 248-263. https://doi.org/10.1108/09685221211267639

NIST Special Publication 800-53 Revision 4 Security and Privacy Controls for Federal Information Systems and Organizations. Sp-800-53Ar4, 462. (2014). *National Institute of Standards and Technology* https://doi.org/10.6028/NIST.SP.800-53Ar4

Perminov, P., Kosachenko, T., Konev, A., & Shelupanov, A. (2020). Automation of Information Security Audit in the Information System on the Example of a Standard "cis Palo Alto 8 Firewall Benchmark." *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(2), 2085–2088. https://doi.org/10.30534/ijatcse/2020/182922020

Splunk® Enterprise Security (2020). *Splunk*. Recuperado de https://www.splunk.com/en_us/software/enterprise-security.html

Recibido: 21/10/2020 / Aceptado: 14/11/2020
doi: https://doi.org/10.26439/interfases2020.n013.4886

# Comparación de técnicas de *machine learning* para detección de sitios web de *phishing*

Andres Eduardo Moncada Vargas
Universidad de Lima, Lima, Perú
20152102@aloe.ulima.edu.pe

## Resumen

El *phishing* es el robo de datos personales a través de páginas web falsas. La víctima de este robo es dirigida a esta página falsa, donde se le solicita ingresar sus datos para validar su identidad. Es en ese momento que se realiza el robo, ya que al ingresar sus datos, estos son almacenados y usados por el *hacker* responsable de dicho ataque para venderlos o ingresar a las entidades y realizar robos o estafas. Para este trabajo se ha investigado sobre distintos métodos de detección de páginas web *phishing* utilizando técnicas de *machine learning*. Así, el propósito de este trabajo es realizar una comparación de dichas técnicas que han demostrado ser las más efectivas en la detección de los sitios web *phishing*. Los resultados obtenidos demuestran que los clasificadores de árboles, denominados Árbol de Decisión y Bosque Aleatorio, han alcanzado las mayores tasas de precisión y efectividad, con valores de entre 97 % y 99 % en la detección de este tipo de páginas.

PALABRAS CLAVE: *phishing* / *machine learning* / ciberseguridad / clasificador / Bosque Aleatorio / *dataset* / *grid search*

## Abstract

### A comparison of machine learning techniques for detection of phishing websites

Phishing is the theft of personal data through fake websites. Victims of this type of theft are directed to a fake website, where they are asked to enter their data to validate their identity. At that moment, theft is carried out, since entered data are stored and used by the hacker responsible for said attack to sell them or enter to websites and perform a fraud or scam. In order to conduct this work, we researched different methods for detecting phishing websites by using machine learning techniques. Thus, the purpose of this work is to compare machine learning techniques that have demonstrated to be the most effective methods to detect phishing websites. The results show that decision tree classifiers such as Decision Tree and Random Forest have achieved the highest accuracy and efficacy rates, with values between 97% and 99%, in detecting these types of websites.

KEYWORDS: phishing / machine learning / cybersecurity / classifier / Random Forest / dataset / grid search

## 1. INTRODUCCIÓN

El ataque cibernético conocido como *phishing* constituye una de las mayores amenazas en la actualidad, ya que es el medio por el cual se han realizado la mayoría de los robos y estafas cibernéticas en los últimos años. El *phishing* es comúnmente conocido como el robo de datos personales por medio de una página web falsa, denominada página *phishing*, creada para que las víctimas de este ataque ingresen sus datos para "validar" su identidad. Estas páginas *phishing* usualmente suelen imitar el diseño y configuración de páginas legítimas de distintas organizaciones, comúnmente bancos y empresas con servicios bancarios. El medio más usado para difundir estas páginas *phishing* es el correo *spam*. Sin embargo, no es el único medio por el cual se realizan ataques *phishing*, debido a que existen otros métodos como, por ejemplo, el *pharming*, que es el redireccionamiento de un usuario que se encuentra en una página legítima a una página *phishing* a través de enlaces directos implantados en dicha página legítima (Abu-Nimeh, Nappa, Wang y Nair, 2007, p. 1).

Según ESET (2017), Ecuador, Perú y México fueron los tres países latinoamericanos con más ataques de *phishing* en el 2016, teniendo un porcentaje del 20,9 %, 16,6 % y 16,1 % respectivamente. Según estudios realizados, en el 2017 hubo un incremento de ataques en un 15 % respecto del año anterior. En el año 2018 el incremento de ataques fue mayor, ya que se detectaron alrededor de 500 millones de ataques *phishing* en todo el mundo. De todos ellos, el número de ataques *phishing* financieros en el 2018 estuvo cerca del total de ataques *phishing* detectados en el 2017. Estos estudios dejan evidencias del porqué se considera al *phishing* como una de las ciberamenazas más comunes que existe en la actualidad.

Ante los ataques de *phishing*, los datos personales se encuentran en peligro, ya que vivimos en una época donde la internet es usada para la mayoría de las actividades diarias. Por este motivo, en este trabajo se han examinado varias metodologías, propuestas a lo largo de los últimos años, que han demostrado ser capaces de detectar y prevenir los ataques *phishing*. Lamentablemente, los métodos propuestos no son capaces de detectar nuevas variantes de estos ataques, debido a que las métricas más importantes para detectar páginas *phishing* derivan de experiencias humanas (Mao, Bian, Tian, Zhhu, Wei, Li y Liang, 2018, p. 2). Por dicho motivo, en la actualidad se recurre a la inteligencia artificial para poder identificar páginas *phishing* de manera dinámica y automática utilizando para ello diferentes métricas (Abu-Nimeh, 2007; Al-Janabi, 2017; Bulakh, 2016; Chen, 2010; Hota, 2018; Jain, 2016; Mao, 2018; Medvet, 2008; Mourtaji, 2017; Rajab, 2018; Sanglerdsinlapachai, 2010).

Anteriormente, Abu-Nimeh, Nappa, Wang y Nair (2007); Abdelhamid, Thabtah, y Abdel-jaber (2017) compararon diversas técnicas de *machine learning* para poder encontrar la mejor manera de detectar los ataques *phishing*. El objetivo de dichos trabajos fue

comprobar la precisión de dichas técnicas y sus resultados, y aunque la metodología de este trabajo podía ayudar a determinar a la técnica más eficiente entre las implementadas, los resultados y la forma de presentarlos no fueron suficientemente concluyentes para conocer qué métodos y bajo qué supuestos y condiciones alcanzaban los mejores resultados de manera objetiva.

La motivación de este trabajo es encontrar la técnica más eficiente para detectar los ataques *phishing*. Por ese motivo, la labor se enfoca en investigar trabajos que hayan usado técnicas de *machine learning* para la detección de *phishing*, replicarlos, revisar los criterios de calibrado y compararlos para determinar cuál de esas técnicas es la más efectiva para detectar páginas *phishing*. El análisis se conduce sobre un conjunto de características sugeridas en la literatura científica revisada en esta investigación. Las contribuciones que ofrece el presente trabajo son las siguientes:

- Realizar una revisión del uso de técnicas de *machine learning* para la detección de *phishing*. De esta forma, podemos valorar la efectividad de los algoritmos de clasificación estudiados.
- Buscar y seleccionar los *datasets* empleados por autores de investigaciones similares para contrastar los resultados de experimentación aplicados.
- Desarrollar una experimentación extendida sobre los criterios mencionados anteriormente.
- Comparar los resultados de la experimentación tomando en cuenta métricas objetivas como la exactitud, precisión, recuperación y valor F. Esto se concreta realizando una parametrización detallada de los algoritmos de clasificación a usar y se discuten los hallazgos de este trabajo en comparación con los resultados de investigaciones anteriores.

Este trabajo se compone de seis secciones; la primera es la presente introducción. En la sección 2 se presenta el estado del arte, en donde se revisa la literatura existente sobre detección de *phishing* basada en aprendizaje automático. La metodología de trabajo y la experimentación se describen en la sección 3. En la sección 4 se analizan los resultados obtenidos en esta investigación y se debate desde la comparativa con propuestas similares. Finalmente, en la sección 5 se presentan las conclusiones de este trabajo.

## 2. ESTADO DEL ARTE

Con el paso del tiempo se han diseñado varias herramientas como medida de seguridad para detectar el *phishing*; y, desafortunadamente, las habilidades de los *hackers*, en constante evolución, han sido un obstáculo para las herramientas tradicionales. Esto ocasionó que las antiguas técnicas de detección de *phishing* no funcionen contra los

ataques más recientes (Bulakh, 2016, p. 1). Como consecuencia de esta amenaza evolutiva, se ha decidido usar técnicas de *machine learning* que permitan encontrar nuevas páginas fraudulentas en un periodo de tiempo largo o indefinido (Hota, 2018; Medvet, 2008; Chen, 2010; Bulakh, 2016; Rajab, 2018).

Basándose en los enfoques de distintos autores para detectar correos *phishing* en el pasado, los autores Hota, Shrivas y Hota (2018) decidieron desarrollar un modelo de identificación de correos *phishing* ensamblando dos técnicas de árboles de decisiones, CART y C4.5, en un modelo de clasificación robusto con la intención de reducir el error que estos algoritmos pueden tener por separado. CART es un clasificador que construye un árbol de decisión binario dividiendo el registro de cada nodo en base a una función de un atributo. C4.5 es un árbol de decisión capaz de manejar atributos continuos y discretos, incluyendo registros con valores desconocidos en el entrenamiento de este.

Los autores Mao *et al*. (2018) analizaron varias soluciones anti-*phishing* y descubrieron que las similitudes de diseño dan un indicio alto de detección de este fraude. Lamentablemente, estas similitudes no podían ser comprendidas para detectar nuevos ataques. Decididos a superar esta dificultad, los autores crearon una herramienta capaz de determinar similitudes de diseño con base en reglas determinadas por un mecanismo de análisis de agregación. Para desarrollar esta herramienta, se evaluaron los clasificadores denominados Máquina de Soporte Vectorial y Árbol de Decisión. En el caso de Máquina de Soporte Vectorial, se configuraba el parámetro *gamma* (gama) y se comparaban los resultados obtenidos de varias pruebas realizadas con diferentes valores para este parámetro. En el caso de Árbol de Decisión, se hizo la misma comparación, pero configurando el parámetro *max depth* (máxima profundidad).

Jain y Gupta (2016) observaron que una de las soluciones más efectivas es integrar funciones de seguridad en los navegadores de internet. Por ese motivo, decidieron realizar una *whitelist* autoactualizable. De esta manera, los navegadores podrán mandar una alerta si es que se ingresa a una página web que no se encuentrara en la lista, evitando que el usuario ingrese a una página *phishing*. Además de eso, es posible detectar y agregar a la lista una página nueva gracias al sistema de detección implementado, el cual funciona con los hipervínculos de las páginas a las cuales ingresa el usuario. La ventaja de esta herramienta es que al agregarle más características de detección puede mejorar la exactitud de detectar una página *phishing*. Sin embargo, al hacer eso, el sistema de detección tomará más tiempo en correr para determinar si una página es *phishing* o no.

Las técnicas convencionales pueden ser muy útiles, pero si no están implementadas con un algoritmo que permita un actualizado automático, como el

mostrado anteriormente, entonces dichas técnicas no serán capaces de tener un funcionamiento continuo, a diferencia de las técnicas con *machine learning* que permiten su uso continuo para una detección de *phishing* a largo plazo. Por el motivo mencionado es que se propusieron varias técnicas de *machine learning*, para ser usadas en las herramientas de detección *phishing* más recientes.

Los métodos de detección de páginas *phishing* basados en similitudes visuales diseñados por Medvet, Kirda y Kruegel (2008), así como por Chen, Dick y Miller (2010) fueron planteados tras notar que las personas asociaban directamente las apariencias de las páginas *phishing* con las páginas legítimas. De este modo, las personas con poco o nulo conocimiento sobre el *phishing* pueden ser víctimas fáciles de robo de datos personales. Por ese motivo, estos métodos fueron diseñados para detectar similitudes visuales y evitar el ingreso de datos personales en páginas web fraudulentas. En el caso de Medvet *et al*. (2008), implementaron un complemento para la herramienta AntiPhish, la cual analiza si una página de ingreso de datos sensibles es segura a través del análisis de información enviada. Esta herramienta puede tener una mala clasificación si el uso de información usada en distintas cuentas es repetido. El implemento realizado por los autores ayuda a evitar la clasificación errónea de esta herramienta, ayudando a verificar si la página sospechosa es idéntica a una página legítima de una empresa o entidad bancaria. Sin embargo, este complemento no se restringe a AntiPhish, sino a toda herramienta que pueda otorgar al complemento una imagen de las páginas legítimas de empresas.

Los autores Sanglerdsinlapachai y Rungsawang (2010) encontraron que, en la primera mitad del 2009, se detectaron más de 55 000 páginas *phishing* activas. Debido a esto, los dos autores detectaron que los dos mayores vectores de ataque son los correos y las páginas *phishing*. Después de decidir enfocarse en las páginas *phishing*, ambos decidieron hacer una herramienta de ensamble de clasificadores, el cual consiste en el ensamblado de distintas técnicas de *machine learning* junto con el uso de aplicaciones heurísticas de CANTINA. Este ensamble de clasificadores utiliza AdaBoost, J48 (árbol de decisión), Naive Bayes, Red Neuronal, Bosque Aleatorio o Máquina de Soporte Vectorial. Estos clasificadores fueron escogidos después de una comparación de clasificación entre distintos algoritmos; los seis mencionados son los que obtuvieron mejor resultado que un método heurístico implementado de la herramienta de detección de *phishing* CANTINA. Al final, el ensamble de clasificadores consistió en AdaBoost, Red Neuronal y Bosque Aleatorio, obteniendo el mejor resultado entre distintos ensamblados realizados con los clasificadores mencionados.

Tras revisar y analizar varios métodos de detección de *phishing*, Bulakh y Gupta (2016) decidieron usar una perspectiva nueva, la de las marcas usadas para el

*phishing*, para desarrollar un nuevo método de detección de *phishing*. Esta herramienta tiene como propósito permitir a las marcas defenderse de manera proactiva contra estos ataques. Tras analizar las páginas web *phishing*, se determinaron las características que les permiten tener un gran tráfico de usuarios y se desarrolló una solución que consiste en una etapa de prefiltro, seguido de un *whitelist* el cual contiene las páginas web legítimas. Dicho método finaliza con un clasificador de *machine learning* supervisado. Esta herramienta funciona cuando se filtran las páginas con formularios HTML; luego, al verificar si la página sospechosa está dentro de la *whitelist,* el cual es el *whitelist* autoactualizable mencionado anteriormente en esta sección y, en caso no se encuentre ahí, funciona mediante la utilización de un clasificador supervisado para clasificar si esta página es *phishing* o no. El clasificador elegido para esta herramienta es Bosque Aleatorio, después de pasar por una comparación con tres algoritmos de árboles de decisiones (J48, CART y PART), Red Neuronal, Regresión Logística y Naive Bayes.

En la actualidad, las redes sociales son el medio de entretenimiento y comunicación de las personas. Sin embargo, los usuarios promedio tienen poco o nulo conocimiento sobre ciberamenazas, por lo que son víctimas fáciles de las páginas maliciosas que aparentan ser páginas regulares. Estas páginas maliciosas pueden robar datos personales de la misma computadora, inyectar virus, descargar un *software* malicioso o ingresar a una página *phishing*. En las redes sociales, al ingresar a estas páginas a veces se activan códigos que hacen que el usuario comparta automáticamente el mismo enlace sin conocimiento alguno.

Para defender a los usuarios de las redes sociales de estas amenazas, los autores Al-Janabi, Quincey y Andras (2017) han planteado un clasificador de Bosque Aleatorio que analiza todas las publicaciones realizadas en una red social que contengan direcciones URL para el redireccionamiento a páginas fuera de dicha red social. Una vez que se hayan filtrado esas publicaciones, se analizan las características de la publicación, tanto del usuario que publicó el URL como el contenido de la publicación. El algoritmo usado para este trabajo fue Bosque Aleatorio, debido a que obtuvo los mejores resultados tras compararlo con otros algoritmos de clasificación, como Naive Bayes, K Vecinos Más Próximos y Regresión Logística.

Rajab (2018) descubrió que, al momento de ingresar a una página web *phishing*, muchas características pueden ser iguales a las de la página web real, pero otras pueden ser diferentes. Las características similares y diferentes en cada página web *phishing* no son las mismas siempre, lo cual llevó al autor a plantear la técnica del análisis de características. Gracias a ella, es posible determinar un conjunto de atributos de una página web sospechosa de *phishing* y compararlas con el mismo conjunto de características de la página web real. La forma en la que trabaja esta

técnica es analizando la correlación de las características con los atributos objetivos. De esta manera se determinan las características importantes, a través de la observación de una alta correlación con los atributos objetivos y baja correlación con características menos relevantes. Tras determinar estas condiciones, se puede ver el conjunto de características adecuado a comparar.

Las herramientas desarrolladas han ayudado en la detección de *phishing* gracias a su efectividad y capacidad de funcionamiento a largo plazo. Sin embargo, como todo instrumento creado por los humanos, tienen un tiempo de obsolescencia y, en este caso, utilidad. Esto se debe a que con el tiempo los hackers van a encontrar métodos para evadir estas detecciones de *phishing* y estas páginas volverán a ser difíciles de detectar, eventualmente. A pesar de esto, es posible que estas técnicas puedan ser mejoradas, al igual que varias aplicaciones y servicios en la red, y se pueda impedir su obsolescencia. Por lo tanto, uno de los propósitos que persigue este trabajo es ofrecer una visión actualizada sobre la efectividad del aprendizaje automático en la identificación de sitios web de *phishing*.

## 3. METODOLOGÍA Y EXPERIMENTACIÓN

Esta investigación propone una metodología de comparación objetiva entre los diferentes métodos de *machine learning* aplicados para la detección de *phishing*. El diagrama de bloques correspondiente a las fases de procesamiento se muestra en la figura 1 y en la siguiente subsección se describe en detalle cada fase, mientras que en las siguientes subsecciones se describen los *datasets* usados y la experimentación realizada.

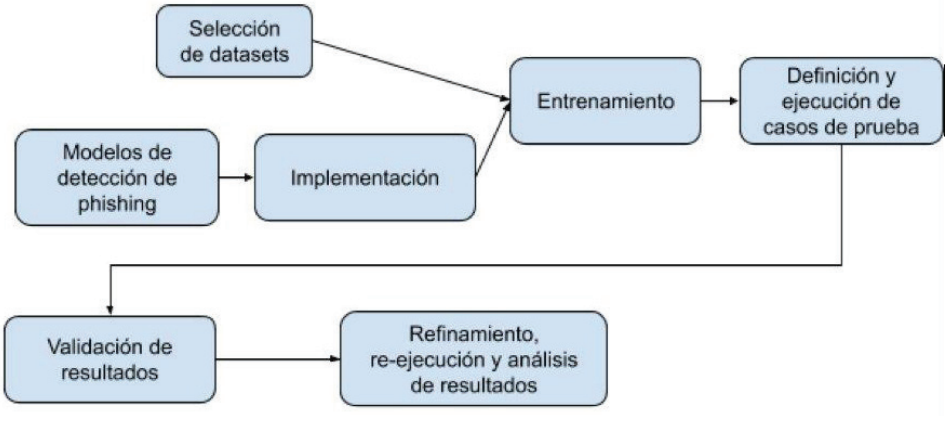



*Figura 1*. Diagrama de bloques de la metodología

Elaboración propia

**a. Metodología**

i. Modelos de detección de *phishing*

Se identifican metodologías usadas en los trabajos revisados en esta investigación para poder replicar las técnicas usadas por los autores y realizar una comparación exacta entre los resultados obtenidos en esta experimentación y la realizada por ellos.

ii. Implementación

Una vez que se determinen y se encuentre la manera de replicar las técnicas usadas o creadas por los distintos autores investigados, estas serán implementadas en ambientes de prueba.

iii. Selección de *datasets*

Al mismo tiempo en que se buscará la manera de replicar las técnicas de los autores, se realizará una búsqueda de los *datasets* usados por estos para llevar a cabo una experimentación comparativa.

iv. Entrenamiento

Tras encontrar todos los *datasets* públicos usados por los autores, estos serán divididos en dos conjuntos, de entrenamiento y de prueba, cada uno. En esta fase de la metodología se usan los conjuntos de entrenamiento con cada clasificador.

v. Definición y ejecución de casos de prueba

Después de entrenar los clasificadores con los conjuntos de entrenamiento, se definirá el caso de prueba de cada *dataset* usado en cada prueba. Posteriormente, se usarán los conjuntos de ejecución con cada clasificador implementado para la predicción y clasificación de estos.

vi. Validación de resultados

Tras ejecutar los clasificadores con los *datasets*, se observarán los resultados obtenidos. Si los resultados obtenidos son coherentes con los resultados de los autores, se procederá a su análisis. En caso contrario, se refinará la implementación de la técnica correspondiente y se volverá a ejecutar.

vii. Refinamiento, reejecución y análisis de resultados

En caso los resultados obtenidos no sean consistentes con los vistos en los trabajos investigados, se realizará un refinamiento de las técnicas en cada caso de prueba. En caso los resultados continúen siendo inconsistentes, se concluirá que los obtenidos son los resultados finales y se analizarán junto con los resultados de las demás pruebas y *datasets*. Al momento de analizar los resultados, se comparará la exactitud, precisión, recuperación y valor F obtenidos y se llegará a una conclusión respecto a los clasificadores.

**b. *Datasets***

i. ISCX-URL-2016: este es un conjunto de *datasets* creado por Islam *et al*. (2016) que contienen cuatro tipos de URL maliciosos, los cuales son *malware*, *phishing*, *spam* y desfiguración. Cada tipo de URL fue agrupado en cuatro *datasets*, combinando URL maliciosas correspondientes al grupo en el que se encuentran con URL benignas en cada uno. Entre los cuatro *datasets* por cada tipo de URL malicioso, se encuentra el primer *dataset* con más de cuarenta características, que luego de una evaluación fueron reducidos por los mismos autores. Debido al alcance de este trabajo de investigación, solo se utilizarán los *datasets* específicos para *phishing* y los que tengan las características reducidas tras la evaluación de los autores, los cuales contienen alrededor de 10 000 URL *phishing* extraídas de OpenPhish, un repositorio activo de páginas *phishing*, de las cuales se usaron 7586 URL *phishing* y 7781 páginas benignas. Al ser diferentes *datasets*, estos pueden tener características en común o únicas respecto a otros de los *datasets*. Por ejemplo:

1. Phishing_BestFirst: este *dataset* contiene catorce características, entre las cuales se encuentra la característica *class* que representa si dicha instancia es benigna o *phishing*, usando los valores *phishing* y *benign* para esta clasificación. Las características analizadas para determinar dicha clasificación están descritas en la tabla 1.

Tabla 1
*Descripción de características del* dataset *Phishing_BestFirst*

| Característica | Descripción |
|---|---|
| domain_token_count | Indica el conteo de *tokens* en el dominio. |
| tld | Indica el conteo de uso de un dominio de alto nivel en la URL. |
| urlLen | Indica el número de caracteres que tiene la URL. |
| domainlength | Indica el tamaño del dominio. |
| fileNameLen | Indica el tamaño del nombre del archivo al que corresponde la URL. |
| pathurlRatio | Indica la ratio de la ruta dividida por URL. |
| NumberofDotsinURL | Indica el número de puntos encontrados en la URL. |
| Query_DigitalCount | Indica el número de dígitos en la parte de consulta de la URL. |
| LongestPathTokenLength | Indica la longitud del *token* dentro de la ruta más larga en la URL. |
| delimiter_Domain | Indica el número de delimitadores de dominios que se encuentran en la URL. |

(*continúa*)

(*continuación*)

| | |
|---|---|
| delimiter_path | Indica el número de delimitadores de rutas encontrados en la URL. |
| SymbolCount_Domain | Indica el conteo de símbolos encontrados en el dominio. |
| Entropy_Domain | Indica el porcentaje de entropía encontrada en la URL. |
| class | Indica si la instancia es benigna o *phishing*. |

Elaboración propia

2. Phishing_Infogain: estos dos *datasets* son similares entre sí, con la única diferencia de que este tiene una característica más que la última mencionada. Dicha característica diferente es avgdomaintokenlen. Las características se explican con mayor detalle en la tabla 2.

Tabla 2
*Descripción de características del* dataset *Phishing_Infogain*

| Característica | Descripción |
|---|---|
| domain_token_count | Indica el conteo de *tokens* en el dominio. |
| avgdomaintokenlen | Indica la longitud promedio de los *tokens* de ruta dentro de la URL. |
| tld | Indica el conteo de uso de un dominio de alto nivel en la URL. |
| urlLen | Indica el número de caracteres que tiene la URL. |
| domainlength | Indica el tamaño del dominio. |
| fileNameLen | Indica el tamaño del nombre del archivo al que corresponde el URL. |
| pathurlRatio | Indica la ratio de la ruta dividida por URL. |
| NumberofDotsinURL | Indica el número de puntos encontrados en el URL. |
| Query_DigitalCount | Indica el número de dígitos en la parte de consulta del URL. |
| LongestPathTokenLength | Indica la longitud del *token* dentro de la ruta más larga en la URL. |
| delimiter_Domain | Indica el número de delimitadores de dominios que se encuentran en la URL. |
| delimiter_path | Indica el número de delimitadores de rutas encontrados en la URL. |
| SymbolCount_Domain | Indica el conteo de símbolos encontrados en el dominio. |
| Entropy_Domain | Indica el porcentaje de entropía encontrada en la URL. |
| Class | Indica si la instancia es benigna o *phishing*. |

Elaboración propia

ii. Phishing Dataset for Machine Learning: *Feature Evaluation* (Phishing_Legitimate_Full): Este *dataset* contiene 48 características extraídas de cinco mil páginas *phishing* y cinco mil páginas legítimas del 2015 y 2017. El archivo está en formato .arff. Este *dataset* fue creado por Tan (2019), uno de los autores del trabajo investigado por Chiew, Tan, Wong, Yong y Tiong (2019) y tiene un gran número de características para evaluar. Las características del *dataset* se encuentran descritas en la tabla 3.

Tabla 3
*Descripción de características del* dataset *Phishing_Legitimate_Ful*

| Característica | Descripción |
|---|---|
| NumDots | Cuenta el número de puntos en la URL. |
| SubdomainLevel | Cuenta el nivel de subdominio de la URL. |
| PathLevel | Cuenta la profundidad de la ruta de la URL. |
| UrlLength | Cuenta el número de caracteres que contiene la URL. |
| NumDash | Cuenta el número de "-" en la URL. |
| NumDashInHostname | Cuenta el número de "-" en la parte del *hostname* de la URL. |
| AtSymbol | Verifica si el símbolo "@" existe en la URL. |
| TildeSymbol | Verifica si el símbolo "~" existe en la URL. |
| NumUnderscore | Cuenta el número de "_" en la URL. |
| NumPercent | Cuenta el número de "%" en la URL. |
| NumQueryComponents | Cuenta el número de partes de consulta en la URL. |
| NumAmpersand | Cuenta el número de "&" en la URL. |
| NumHash | Cuenta el número de "#" en la URL. |
| NumNumericChars | Cuenta el número de caracteres numéricos en la URL. |
| NoHttps | Indica si hay HTTPS en la URL. |
| RandomString | Indica si hay un valor de cadena aleatorio en la URL. |
| IpAddress | Indica si la dirección IP es usada en la parte del *hostname* de la URL. |
| DomainInSubdomains | Indica si TLD o ccTLD es usado como parte del subdominio en la URL. |
| DomainInPaths | Indica si TLD o ccTLD es usado en la ruta de la URL. |
| HttpsInHostname | Indica si HTTPS es ofuscado en la parte de *hostname* de la URL. |
| HostnameLength | Cuenta el número de caracteres en la parte de *hostname* de la URL. |
| PathLength | Cuenta el número de caracteres en la ruta de la URL. |
| QueryLength | Cuenta el total de caracteres en la parte de consulta de la URL. |
| DoubleSlashInPath | Verifica si "//" existe en la ruta de la URL. |
| NumSensitiveWords | Cuenta el número de palabras sensibles en la URL. |
| EmbeddedBrandName | Indica si el nombre de la marca aparece en subdominios y la ruta de la URL siendo el nombre de la marca el nombre de dominio más frecuente en el contenido HTML. |
| PctExtHyperlinks | Cuenta el porcentaje de hipervínculos externos en el código fuente HTML. |
| PctExtResourceUrls | Cuenta el porcentaje de URL de recursos externos en el código fuente HTML. |
| ExtFavicon | Verifica si el favicon está siendo cargado desde un nombre de dominio que es diferente del nombre de dominio de la URL. |
| InsecureForms | Verifica si el atributo de formulario de acción contiene una URL sin protocolo HTTPS. |

(*continúa*)

(*continuación*)

| | |
|---|---|
| RelativeFormAction | Verifica si el atributo de formulario de acción contiene una URL relativa. |
| ExtFormAction | Verifica si el atributo de formulario de acción contiene una URL de un dominio externo. |
| AbnormalFormAction | Verifica si el atributo de formulario de acción contiene un "#" o "about:blank" o una cadena vacía o "javascript:true". |
| PctNullSelfRedirectHyperlinks | Cuenta el porcentaje de campos de hipervínculos que contienen valor vacío o un valor de autorredireccionamiento o un valor anormal. |
| FrequentDomainNameMismatch | Verifica si el nombre de dominio más frecuente en el código fuente HTML no concuerda con el nombre de dominio de la URL. |
| FakeLinkInStatusBar | Verifica si el código fuente HTML contiene un comando JavaScript onMouseOver para mostrar una URL falsa en la barra de estado. |
| RightClickDisabled | Verifica si el código fuente HTML contiene comando JavaScript para deshabilitar funciones de clic derecho. |
| PopUpWindow | Verifica si el código fuente HTML contiene comando JavaScript para lanzar ventanas emergentes. |
| SubmitInfoToEmail | Verifica si el código fuente HTML contiene la función HTML "mailto". |
| IframeOrFrame | Verifica si se usa iframe o frame en el código fuente HTML. |
| MissingTittle | Verifica si la etiqueta de título en el código fuente HTML está vacía. |
| ImagesOnlyInForm | Verifica si el alcance del formulario en el código fuente HTML contiene únicamente imágenes. |
| SubdomainLevelRT | Cuenta el número de puntos en la parte del *hostname* del URL. |
| UrlLengthRT | Cuenta el total de caracteres en el URL de la página y utiliza reglas y límites para generar el valor. |
| PctExtResourceUrlsRT | Cuenta el porcentaje de las URL de recursos externos en el código fuente HTML de la página. |
| AbnormalExtFormActionR | Verifica si el atributo de formulario de acción contiene un dominio extranjero o "about:blank" o se encuentra vacío. |
| ExtMetaScriptLinkRT | Cuenta el porcentaje de metaetiquetas, *scripts* y etiquetas de enlace que contienen URL externos en los atributos. |
| PctExtNullSelfRedirectHyperlinksRT | Calcula el porcentaje de hipervínculos en el código fuente HTML que usen un nombre de dominio diferente, empiecen con "#" o usen "JavaScript :void(0)". |
| CLASS_LABEL | Indica si la instancia es benigna o *phishing*. |

Elaboración propia

iii. Website Phishing Data Set (PhishingData): Este *dataset* fue usado por Cuzzocrea, Martinelli y Mercaldo (2018) para comparar algoritmos de *machine learning* basados en el árbol de decisiones (por ejemplo, Árbol de Decisión y Bosque Aleatorio). Debido a que los autores obtuvieron métricas únicamente de precisión, recuperación y valor F, estos son los valores por comparar en la

sección de resultados y discusiones. Este *dataset* presenta valores 1, 0 y -1, los cuales indican si las características presentan indicios de que la página es legítima, sospechosa o *phishing,* respectivamente. Las características de este *dataset* son descritas en la tabla 4.

Tabla 4
*Descripción de características del* dataset *PhishingDat*

| Característica | Descripción |
|---|---|
| SFH | Verifica si la información ingresada en la página se transfiere a un servidor o no, o si se transfiere a un servidor de un dominio diferente. |
| popUpWindow | Verifica si la página usa ventanas emergentes para redirigir a los usuarios a esta página. |
| SSLfinal_State | Verifica si el protocolo HTTPS de la página es de confianza o si es falso. |
| Request_URL | Verifica si la mayoría de los objetos en la página web son cargados desde un dominio diferente al de la URL o no. |
| URL_of_Anchor | Verifica si los enlaces dentro de la página web apuntan a un dominio diferente al de la URL o no. |
| web_traffic | Verifica si el tráfico web de la página es el de una página legítima, un *phishing* o si es sospechoso. |
| URL_Length | Verifica si la longitud de la URL es considerada *phishing*, sospechosa o legítima. |
| age_of_domain | Verifica si el tiempo de vida del dominio es de una página *phishing* o una legítima. |
| having_IP_Address | Verifica si la URL tiene la dirección IP en ella o no. |
| Result | Indica si la página es *phishing*, es sospechosa de *phishing* o si es legítima. |

Elaboración propia

### c. Experimentación

Siguiendo los pasos mencionados en la metodología, se ejecutaron los *datasets* mencionados anteriormente en el ambiente Spyder con las técnicas de clasificación mencionadas en los antecedentes. En los trabajos investigados se abordaron dos métodos de ejecución, el primero fue división por porcentaje (DP), el cual es la división del *dataset* en un conjunto de entrenamiento para los clasificadores y un conjunto de ejecución. El segundo es *cross-validation* de 10 pliegues (CV). La DP de los tres primeros *datasets* será explicada en la sección de resultados para cada uno. En el caso del CV, el último *dataset* fue dividido en diez grupos, usando nueve para el entrenamiento y el restante para la ejecución; luego, se repitió este proceso para que cada grupo haya sido usado para la ejecución y se obtengan resultados con todos estos.

En la fase de refinamiento, ejecución y análisis de resultados de la metodología, se realizó un refinamiento de los hiperparámentros usando *Grid Search* para la automatización de este proceso. Debido a que las ejecuciones en cada *dataset* se realizan de manera diferente y con un número de datos diferentes, cada *dataset* recibió un refinamiento único.

Los hiperparámetros de cada clasificador para cada *dataset* son indicados en las tablas 5, 6, 7, 8 y 9 en el orden en el que los clasificadores fueron mencionados anteriormente en la sección de antecedentes. Los hiperparámetros listados son los usados en el ambiente Spyder. En los casos de los clasificadores denominados Redes Neuronales y Máquina de Soporte Vectorial, debido al largo tiempo de entrenamiento y ejecución al usar *Grid Search*, se usaron los hiperparámetros por defecto de estos. Los resultados de esta experimentación se ven en la siguiente sección.

Tabla 5

*Valores de los hiperparámetros modificados del clasificador Árbol de Decisión para cada caso de prueba*

| Hiperparámetros | Phishing_ BestFirst | Phishing_Infogain | Phishing_ Legitimate_Full | PhishingData |
|---|---|---|---|---|
| criterion | gini | entropy | entropy | entropy |
| splitter | best | best | best | best |
| max_depth | None | None | None | None |
| min_samples_split | 2 | 2 | 2 | 2 |
| min_samples_leaf | 1 | 1 | 1 | 1 |
| min_weight_fraction_leaf | 0.0 | 0.0 | 0.0 | 0.0 |
| max_features | None | None | None | None |
| random_state | None | None | None | None |
| max_leaf_nodes | None | None | None | None |
| min_impurity_decrease | 0.0 | 0.0 | 0.0 | 0.0 |
| class_weight | None | None | None | None |
| ccp_alpha | 0.0 | 0.0 | 0.0 | 0.0 |

Elaboración propia

Tabla 6
*Valores de los hiperparámetros modificados del clasificador Regresión Logística para cada caso de prueba*

| Hiperparámetros | Phishing_ BestFirst | Phishing_Infogain | Phishing_ Legitimate_Full | PhishingData |
|---|---|---|---|---|
| penalty | none | none | none | l1 |
| dual | False | False | False | False |
| tol | 0.0001 | 0.0001 | 0.0001 | 0.0001 |
| C | 1.0 | 1.0 | 1.0 | 1.0 |
| fit_interceptor | True | True | True | True |
| intercept_scaling | 1 | 1 | 1 | 1 |
| class_weight | None | None | None | None |
| random_state | None | None | None | None |
| solver | newton-cg | newton-cg | newton-cg | liblinear |
| max_iter | 10000 | 1627 | 1627 | 1627 |
| multi_class | auto | auto | auto | auto |
| verbose | 0 | 0 | 0 | 0 |
| warm_start | False | False | False | False |
| n_jobs | None | None | None | None |
| l1_ratio | None | None | None | None |

Elaboración propia

Tabla 7
*Valores de los hiperparámetros modificados del clasificador Red Neuronal para cada caso de prueba*

| Hiperparámetros | Phishing_ BestFirst | Phishing _Infogain | Phishing_ Legitimate_Full | Phishing Data |
|---|---|---|---|---|
| hidden_layer_sizes | (100,) | (100,) | (100,) | (100,) |
| activation | relu | relu | relu | relu |
| solver | adam | adam | adam | adam |
| alpha | 0.0001 | 0.0001 | 0.0001 | 0.0001 |
| batch_size | auto | auto | auto | auto |
| learning_rate | constant | constant | constant | constant |
| learning_rate_init | 0.0001 | 0.0001 | 0.0001 | 0.0001 |
| power_t | 0.5 | 0.5 | 0.5 | 0.5 |
| max_iter | 200 | 400 | 400 | 1000 |
| shuffle | True | True | True | True |
| random_state | None | None | None | None |

(*continúa*)

(*continuación*)

| | | | | |
|---|---|---|---|---|
| tol | 0.0001 | 0.0001 | 0.0001 | 0.0001 |
| verbose | False | False | False | False |
| warm_start | False | False | False | False |
| momentum | 0.9 | 0.9 | 0.9 | 0.9 |
| nesterovs_ momentum | True | True | True | True |
| early_stopping | False | False | False | False |
| validation_fraction | 0.1 | 0.1 | 0.1 | 0.1 |
| beta_1 | 0.9 | 0.9 | 0.9 | 0.9 |
| beta_2 | 0.999 | 0.999 | 0.999 | 0.999 |
| épsilon | 1e-08 | 1e-08 | 1e-08 | 1e-08 |
| n_iter_no_change | 10 | 10 | 10 | 10 |
| max_fun | 15000 | 15000 | 15000 | 15000 |

Elaboración propia

Tabla 8

*Valores de los hiperparámetros modificados del clasificador Máquina de Soporte Vectorial para cada caso de prueba*

| Hiperparámetros | Phishing_ BestFirst | Phishing_Infogain | Phishing_ Legitimate_Full | PhishingData |
|---|---|---|---|---|
| C | 1.0 | 1.0 | 1.0 | 1.0 |
| kernel | rbf | rbf | rbf | rbf |
| degree | 3 | 3 | 3 | 3 |
| gamma | auto | auto | auto | auto |
| coef0 | 0.0 | 0.0 | 0.0 | 0.0 |
| shrinking | True | True | True | True |
| probability | False | False | False | False |
| tol | 0.001 | 0.001 | 0.001 | 0.001 |
| cache_size | 200 | 200 | 200 | 200 |
| class_weight | None | None | None | None |
| verbose | False | False | False | False |
| max_iter | -1 | -1 | -1 | -1 |
| decisión_function_shape | ovr | ovr | ovr | ovr |
| break_ties | False | False | False | False |
| random_state | None | None | None | None |

Elaboración propia

Tabla 9
*Valores de los hiperparámetros modificados del clasificador Bosque Aleatorio para cada caso de prueba*

| Hiperparámetros | Phishing_ BestFirst | Phishing_Infogain | Phishing_ Legitimate_Full | PhishingData |
|---|---|---|---|---|
| n_estimators | 100 | 100 | 100 | 100 |
| criterion | gini | entropy | gini | entropy |
| max_depth | None | None | None | None |
| min_samples_split | 2 | 2 | 2 | 2 |
| min_samples_leaf | 1 | 1 | 1 | 1 |
| min_weight_fraction_leaf | 0.0 | 0.0 | 0.0 | 0.0 |
| max_features | auto | auto | auto | auto |
| max_leaf_nodes | None | None | None | None |
| min_impurity_decrease | 0.0 | 0.0 | 0.0 | 0.0 |
| bootstrap | False | False | False | True |
| oob_score | False | False | False | False |
| n_jobs | None | None | None | None |
| random_state | Nonce | Nonce | Nonce | Nonce |
| verbose | 0 | 0 | 0 | 0 |
| warm_start | False | False | False | False |
| class_weight | balanced | balanced | balanced | balanced |
| ccp_alpha | 0.0 | 0.0 | 0.0 | 0.0 |
| max_samples | None | None | None | None |

Elaboración propia

## 4. RESULTADOS Y DISCUSIONES

### a. Resultados

Tras realizar la experimentación de la sección anterior, se obtuvieron los resultados vistos en las siguientes subsecciones. En cada subsección se menciona a los autores que usaron el *dataset* correspondiente. Los trabajos realizados por estos autores son mencionados como "Trabajos anteriores" en las tablas de comparación correspondientes.

Algo que se puede notar, en casi todos los casos, es que los resultados obtenidos por el clasificador Bosque Aleatorio son superiores a los demás, siendo Árbol de Decisión el siguiente clasificador con mejor resultado. El único caso en el que no ocurre esto es en el del *dataset* PhishingData que usa el método de entrenamiento CV de 10 *folds*, siendo Redes Neuronales el que obtiene los mejores resultados y Bosque Aleatorio los segundos mejores. Con base en esto podemos

confirmar lo mencionado por Chiew *et al.* (2019) acerca de Bosque Aleatorio, que es el clasificador más adecuado en comparación con los otros analizados.

i. Phishing_BestFirst

En el trabajo de detección de URL maliciosas realizado por Islam, Rathore, Lashkari, Stakhanova, y Ghorbani (2016), se creó este *dataset* junto con Phishing_Infogain. Al realizar la prueba de este *dataset*, se usó un 80 % de este para el entrenamiento y el 20 % restante para la ejecución de los clasificadores usados en su experimentación. En los resultados obtenidos, los cuales aparecen en la tabla 10, se puede observar que Bosque Aleatorio obtuvo un mejor rendimiento que los demás clasificadores con porcentajes de entre 97,69 % y 98,47 %, siendo seguido por Árbol de Decisión con una diferencia menor a 1% en cada métrica. También se observa que Regresión Logística es el clasificador menos eficiente de los cinco comparadores, con porcentajes de rendimiento entre 91,80 % y 91,81 %.

En la tabla 11 se observa una comparación entre los resultados obtenidos en este trabajo y los que obtuvieron Islam *et al*. (2016) al momento de realizar su experimentación. Ellos solo usaron dos de los clasificadores que se usaron en este trabajo, por lo que esos dos son los únicos que se podrán comparar. Se puede observar que Bosque Aleatorio es mejor que Árbol de Decisión en ambos trabajos. Sin embargo, se puede notar que los resultados anteriores de Bosque Aleatorio pueden llegar a ser mejores que los obtenidos en este trabajo, mientras que en Árbol de Decisión ocurre lo contrario. No obstante, un punto a tomar en cuenta es que los resultados anteriores no presentan porcentaje con decimales, lo cual impide conocer si los resultados fueron redondeados o no. En conclusión, Bosque Aleatorio clasifica de manera más eficiente que Árbol de Decisión en esta prueba de características con valores exactos.

Tabla 10

*Resultados obtenidos para cada clasificador (porcentajes) (Phishing_BestFirst)*

| | Exactitud | Precisión | Recuperación | Valor F |
|---|---|---|---|---|
| Árbol de Decisión | 97,20 | 97,20 | 97,20 | 97,20 |
| Regresión Logística | 91,80 | 91,81 | 91,80 | 91,80 |
| Redes Neuronales | 96,16 | 96,98 | 95,19 | 96,08 |
| Máquina de Soporte Vectorial | 96,19 | 95,64 | 96,71 | 96,17 |
| Bosque Aleatorio | 98,11 | 98,47 | 97,69 | 98,08 |

Elaboración propia

Tabla 11
*Comparación de resultados anteriores y obtenidos (porcetantajes) (Phishing_BestFirst)*

| | Precisión | | Recuperación | |
|---|---|---|---|---|
| | Islam *et al.* (2016) | Propuesta DP | Islam *et al.* (2016) | Propuesta DP |
| Árbol de Decisión | 97 | 97,20 | 97 | 97,20 |
| Bosque Aleatorio | 99 | 98,47 | 99 | 97,69 |

Elaboración propia

ii. Phishing_Infogain

Este *dataset*, al igual que el anterior, fue creado por Islam *et al.* (2016). Por ese motivo, se utilizó el mismo método con el *dataset* anterior, se usó el 80 % del *dataset* para entrenamiento y el 20 % restante para la ejecución. Los resultados obtenidos se encuentran en la tabla 12. En este caso, se observa que los resultados obtenidos son similares a los resultados del *dataset* anterior, encontrándose una ligera diferencia entre ellos. En conclusión, el clasificador con el mejor rendimiento en los *datasets* creados por Islam *et al.* (2016) es Bosque Aleatorio, un resultado que se observó de igual manera en la investigación de estos autores.

En la tabla 13 se encuentra la comparación de los resultados anteriores, los mismos vistos en la tabla 11 y los resultados recientemente obtenidos. Lo observable aquí es que los resultados anteriores son mejores que los obtenidos en este trabajo. Esto da a concluir que este *dataset*, aunque tenga las mismas características y pueda ser usado para clasificación, no es el indicado para realizar una predicción formal y certera. Otra conclusión es que el mejor clasificador entre los dos vistos en esta tabla es Bosque Aleatorio, al igual que en el trabajo de Islam *et al.* (2016).

Tabla 12
*Resultados obtenidos para cada clasificador (porcentajes) (Phishing_Infogain)*

| | Exactitud | Precisión | Recuperación | Valor F |
|---|---|---|---|---|
| Árbol de Decisión | 97,33 | 97,33 | 97,20 | 97,20 |
| Regresión Logística | 91,44 | 91,52 | 91,44 | 91,44 |
| Redes Neuronales | 95,16 | 96,89 | 94,16 | 95,51 |
| Máquina de Soporte Vectorial | 95,74 | 96,59 | 94,75 | 95,66 |
| Bosque Aleatorio | 98,11 | 98,48 | 97,70 | 98,09 |

Elaboración propia

Tabla 13
*Comparación de resultados anteriores y obtenidos (porcentajes) (Phishing_Infogain)*

| | Precisión | | Recuperación | |
|---|---|---|---|---|
| | Islam *et al.* (2016) | Propuesta DP | Islam *et al.* (2016) | Propuesta DP |
| Árbol de Decisión | 97 | 97,33 | 97 | 97,33 |
| Bosque Aleatorio | 99 | 98,48 | 99 | 97,70 |

Elaboración propia

iii. Phishing_Legitimate_Full

En el trabajo de Chiew *et al.* (2019), donde fue creado y usado este *dataset*, se realizó la experimentación dividiendo el *dataset* en diez particiones y realizando la prueba con cada una de ellas; luego, se promedió la exactitud obtenida por cada partición. Para cada partición se usó el 70 % del *dataset* para entrenamiento y el otro 30 % para la ejecución. En este caso se realizó la división por porcentaje de igual manera que en el trabajo realizado, pero usando el *dataset* completo, en lugar de dividirlo en 10 partes.

Como se puede ver en la tabla 14, Redes Neuronales obtuvo un mejor rendimiento que con los *datasets* anteriores, logrando obtener un porcentaje de rendimiento cercano a Árbol de Decisión, superándolo ligeramente en Recuperación. Además de eso, se ve que Regresión Logística tuvo un mejor rendimiento de igual manera, logrando que Máquina de Soporte Vectorial quede como el clasificador menos eficiente para este caso de prueba. Pero, al igual que en los *datasets* anteriores, Bosque Aleatorio obtuvo el mejor rendimiento, siento el más eficiente entre los cincos clasificadores.

En la tabla 15 vemos una comparación entre los resultados anteriores de las pruebas realizadas con este *dataset* y los resultados obtenidos actualmente con una DP similar a la realizada en el trabajo de Chiew *et al*. (2019). En el trabajo anterior solo se evaluó la exactitud, por lo que será lo único a comparar en este caso de prueba. Se puede ver que, al igual que los casos anteriores, Bosque Aleatorio obtuvo el mejor rendimiento, seguido de Árbol de Decisión y, al final, Máquina de Soporte Vectorial. Sin embargo, a pesar de haber realizado el mismo experimento, los resultados anteriores y los obtenidos en este trabajo son diferentes. Eso lleva a suponer que los resultados obtenidos tras dividir el *dataset* en 10 partes en el trabajo anterior causaron que los resultados en cada uno no fueran tan buenos como una experimentación con el *dataset* entero.

En conclusión, es posible que utilizar el *dataset* entero y realizar un entrenamiento con este pueda aumentar las probabilidades de que se pueda

clasificar con mayor exactitud; asimismo, que usar Bosque Aleatorio es la mejor opción vista tanto por otros autores como en este trabajo.

Tabla 14
*Resultados obtenidos para cada clasificador (porcentajes) (Phishing_Legitimate_Full)*

| | Exactitud | Precisión | Recuperación | Valor F |
|---|---|---|---|---|
| Árbol de Decisión | 97,60 | 97,60 | 97,60 | 97,60 |
| Regresión Logística | 95,60 | 95,61 | 95,60 | 95,60 |
| Redes Neuronales | 97,45 | 96,71 | 98,18 | 97,44 |
| Máquina de Soporte Vectorial | 92,10 | 91,75 | 92,31 | 92,03 |
| Bosque Aleatorio | 98,80 | 98,79 | 98,79 | 98,79 |

Elaboración propia

Tabla 15
*Comparación de resultados anteriores y obtenidos (porcentajes) (Phishing_Legitimate_Full)*

| | Exactitud | |
|---|---|---|
| | Chiew *et al.* (2019) (DP) | Propuesta DP |
| Árbol de Decisión | 94,37 | 97,60 |
| Máquina de Soporte Vectorial | 92,20 | 92,10 |
| Bosque Aleatorio | 96,17 | 98,80 |

Elaboración propia

iv. PhishingData

En la tabla 16 se puede ver la comparación de los resultados obtenidos por este *dataset* usado por Cuzzocrea *et al.* (2018). Estos autores decidieron usar únicamente CV de 10 pliegues, por lo que, a diferencias de los anteriores casos de prueba, se están comparando estos resultados. Al observar bien la tabla, se puede deducir que este *dataset* no pude ser clasificado como los anteriores por los clasificadores, lo cual llevó a que los resultados en este caso de prueba fueran menores a los vistos en las tablas anteriores. En este caso, el clasificador con mejor resultado fue Redes Neuronales, por lo que se puede concluir que en un caso de prueba distinto a los anteriores se puede conseguir un clasificador diferente a Bosque Aleatorio como el de mejor rendimiento.

Algo similar se observa en la tabla 17, que es la comparación de los resultados anteriores y los resultados recientemente obtenidos de los clasificadores en común. En este caso vemos cómo los resultados de Árbol de Decisión del trabajo anterior son mayores a los de Bosque Aleatorio. Sin embargo, los resultados de este trabajo tienen un patrón diferente. Otra observación es que los resultados del trabajo anterior son mejores que los obtenidos en esta experimentación, lo cual indica que hace falta un mejor refinamiento al momento de hacer pruebas con este *dataset*. Se concluye además que los clasificadores pueden tener un mejor rendimiento dependiendo del caso de prueba, como se ve en los resultados de este *dataset*.

Tabla 16
*Resultados obtenidos para cada clasificador (porcentajes) (PhishingData)*

| | Exactitud | Precisión | Recuperación | Valor F |
|---|---|---|---|---|
| Árbol de Decisión | 87,08 | 87,36 | 87,10 | 87,52 |
| Regresión Logística | 81,92 | 75,39 | 81,92 | 78,44 |
| Redes Neuronales | 89,22 | 90,03 | 90,25 | 89,52 |
| Máquina de Soporte Vectorial | 86,26 | 85,66 | 86,26 | 84,44 |
| Bosque Aleatorio | 88,56 | 88,60 | 88,56 | 88,56 |

Elaboración propia

Tabla 17
*Comparación de resultados anteriores y resultados obtenidos (porcentajes) (PhishingData)*

| | Precisión | | Recuperación | | Valor F | |
|---|---|---|---|---|---|---|
| | Cuzzocrea *et al.* (2018) (CV) | Propuesta (CV) | Cuzzocrea *et al.* (2018) (CV) | Propuesta (CV) | Cuzzocrea *et al.* (2018) (CV) | Propuesta (CV) |
| Árbol de Decisión | 91,35 | 87,36 | 90,40 | 87,10 | 90,85 | 87,52 |
| Bosque Aleatorio | 90,40 | 88,60 | 90,20 | 88,56 | 90,30 | 88,56 |

Elaboración propia

**b. Discusión**

Como se observa en las tablas de la subsección anterior, desde la tabla 10 a la tabla 17, los experimentos llegan a tener resultados diferentes a los resultados vistos en los trabajos realizados por Islam *et al.* (2016), Chiew *et al.* (2019) y Cuzzocrea *et al.* (2018). Estas diferencias son atribuibles al escaso nivel de detalle aportado en la implementación de ciertos métodos de aprendizaje

automático. Debido a esto, la experimentación con cada *dataset* fue optimizada hasta obtener los mejores resultados posibles, Adicionalmente, estos autores solo usaron algunos de los clasificadores utilizados en la experimentación, los cuales son Árbol de Decisión, Bosque Aleatorio y, en el caso del *dataset* Phishing_Legitimate_Full, Máquina de Soporte Vectorial, por lo que la comparación entre los resultados de estos autores y de este trabajo se limita a estos métodos en común. No obstante, la inclusión de resultados con clasificadores adicionales contribuye a un mejor análisis de los resultados. En los casos de Bosque Aleatorio en los *datasets* usados en los trabajos de Islam *et al*. (2016) y Cuzzocrea *et al*. (2018), los resultados de dichas investigaciones fueron mejores que los obtenidos en este trabajo, por lo cual se presume que los clasificadores pueden ser refinados a mayor detalle, hipótesis que será contrastada en futuras investigaciones. En otros casos, como los de los resultados del *dataset* Phishing_Legitimate_Full del trabajo de Chiew *et al*. (2019) y los casos de Árbol de Decisión de los *datasets* del trabajo de Islam *et al*. (2016), los resultados de esta experimentación fueron mejores en la mayoría de los criterios comparados.

En los *datasets* Phishing_BestFirst y Phishing_Infogain, los resultados obtenidos en el trabajo de Islam *et al*. (2016) fueron redondeados, por lo que solo se muestran valores porcentuales enteros, siendo el 97 % para la precisión y recuperación del Árbol de Decisión y 99 % para Bosque Aleatorio. Se observa que los resultados obtenidos en Árbol de Decisión en la experimentación son mayores a 97 %, lo cual supone una mejora en estos resultados, mientras que en Bosque Aleatorio se observa un resultado menor a lo requerido para ser redondeado a 99 %, lo que indica que un mejor refinamiento de los hiperparámetros es capaz de dar mejores resultados que igualan o superan los obtenidos en el trabajo de Islam *et al*. (2016). Sin embargo, al analizar las tablas 10 y 11, podemos determinar que los clasificadores de árboles, Árbol de Decisión y Bosque Aleatorio son los más efectivos para este *dataset*, siendo el que da un mejor rendimiento entre estos dos Bosque Aleatorio. Este patrón también se observa en los resultados del trabajo de Islam *et al*. (2016), donde Bosque Aleatorio es el clasificador con mejor rendimiento, seguido por Árbol de Decisión.

En el *dataset* Phishing_Legitimate_Full, solo se realizó una comparación de la exactitud, debido a que fue la única métrica obtenida del trabajo de Chiew *et al*. (2019). En este caso, gracias a que el clasificador Máquina de Soporte Vectorial fue usado tanto en dicho estudio como en este trabajo, se puede hacer una comparación más amplia y que no engloba únicamente a clasificadores de árboles. En la tabla 15 se puede ver con claridad que los resultados obtenidos han sido mejores que los obtenidos en el trabajo de Chiew *et al*. (2019), a excepción de Máquina de Soporte Vectorial, lo que significa que en este

trabajo la configuración de los clasificadores ha sido mejor refinada o que el uso del *dataset* entero para el entrenamiento y ejecución de los clasificadores dio mejores resultados que dividiendo el *dataset* en diez partes, ejecutando los clasificadores con cada una de ellas y promediando los resultados.

Viendo primero el caso de Máquina de Soporte Vectorial, la diferencia entre los resultados de la parametrización propuesta y el trabajo investigado es de 0,1 %, lo cual no indica una baja de rendimiento significativa. En el caso de Árbol de Decisión, la diferencia entre el resultado obtenido en la experimentación y el resultado del trabajo investigado es de 3,23 %, teniendo un mejor resultado que el visto en el trabajo de Chiew *et al*. (2019). En el caso de Bosque Aleatorio, la diferencia de resultados respecto a este trabajo es de 2,63 %, siendo mejor el resultado obtenido en esta experimentación. Analizando las dos tablas se puede determinar que Bosque Aleatorio es el clasificador con mejor rendimiento del grupo para este *dataset*, al igual que para los anteriores.

Finalmente, en el caso del *dataset* PhishingData, ocurre un conjunto de hechos diferentes a los vistos en los *datasets* anteriores. En este caso se comparan las métricas de precisión, recuperación y valor F de los clasificadores Árbol de Decisión y Bosque Aleatorio, además de comparar resultados de ejecución de CV. Los resultados obtenidos en este trabajo son muy diferentes a los observados en el trabajo de Cuzzocrea *et al*., (2018). En este caso de prueba los resultados de las precisiones, las recuperaciones y los valores F obtenidos son menores que los vistos en el trabajo de estos autores. Además, en este caso en particular, se ve que el clasificador con mejor rendimiento es Redes Neuronales, un patrón diferente al visto en el trabajo de Cuzzocrea *et al*. (2018).

Tras revisar todos los resultados de la experimentación y de la comparación entre estos y los resultados de trabajos pasados, podemos concluir que los clasificadores de árboles de decisión siempre han dado los mejores resultados, siendo los más eficientes para la clasificación de páginas *phishing*. De entre los dos clasificadores de árboles de decisión, Árbol de Decisión y Bosque Aleatorio, se observa que Bosque Aleatorio obtiene mejores resultados en la mayoría de los *datasets*, a excepción del último, donde Redes Neuronales logra un mejor rendimiento. Pero, a pesar de eso, Bosque Aleatorio logró ser más eficiente que Árbol de Decisión. En los trabajos de Islam *et al*. (2016) y Chiew *et al*. (2019) se puede notar que el patrón de resultados es el mismo, a pesar de presentar indicadores de precisión diferentes. En estos casos, Bosque Aleatorio da mejores resultados que Árbol de Decisión. En el caso del *dataset* Phishing_Legitimate_Full, el clasificador de Máquina de Soporte Vectorial dio resultados menores a los otros dos clasificadores mencionados, de la misma manera que en este

trabajo. El único caso donde el patrón de resultados obtenidos en este trabajo no concuerda con el patrón del trabajo anterior es en el trabajo de Cuzzocrea *et al*, (2018), donde vemos que Redes Neuronales logró un mejor rendimiento, seguido por Bosque Aleatorio y posteriormente Árbol de Decisión, un patrón totalmente diferente al del trabajo de estos autores. Con esto podemos determinar que Bosque Aleatorio es el mejor clasificador para la detección de *phishing* y Árbol de Decisión es una segunda opción muy cercana.

## 5. CONCLUSIONES

El *phishing* es una de las ciberamenazas más grandes actualmente, debido al alto número de ataques reportados en Latinoamérica en los últimos años y la cantidad creciente de robos y estafas realizadas como consecuencia de estos ataques cada año. Debido a este problema, se han ideado varias herramientas capaces de detectar páginas *phishing*, que evitan el ingreso a estas y, por ende, el robo de datos personales. Sin embargo, estas herramientas deben ser actualizadas debido a que las páginas *phishing* tienen un tiempo de vida corto para evitar su detección y porque las técnicas de generación de estas páginas evolucionan para evadir las herramientas de detección. Por dicho motivo, se recurrió a técnicas de *machine learning* para detectar páginas *phishing* de manera efectiva. Los clasificadores de *machine learning* tienen un rendimiento variado debido a la ejecución que tienen y a las características de las cuales aprenden y luego comparan para determinar si una instancia es etiquetada como *phishing* o legítima. Tras investigar en distintos trabajos acerca de los clasificadores de *machine learning* más efectivos aplicados a la detección de *phishing*, se identificaron cinco principales: Árbol de Decisión, Regresión Logística, Redes Neuronales, Máquina de Soporte Vectorial y Bosque Aleatorio, Tras definir una metodología que permita hacer una comparación detallada y exhaustiva, se realizó una experimentación extendida. Los resultados experimentales fueron comparados entre sí y con los resultados obtenidos en investigaciones similares cuyos métodos de aprendizaje automático fueron aplicados a los mismos *datasets* analizados en este trabajo. Como resultado de este proceso, se concluye que los clasificadores de árboles son los más eficientes para la detección de Phishing, entre los cuales están Árbol de Decisión y Bosque Aleatorio. Los resultados entre estos dos clasificadores tienen diferencias ligeras, siendo Bosque Aleatorio el de mejor rendimiento en la mayoría de los casos, tanto en trabajos investigados como en esta experimentación. Árbol de Decisión, que es el clasificador de mejor rendimiento en el último *dataset* visto, es una segunda opción muy cercana y factible al momento de realizar una herramienta de detección de *phishing* usando *machine learning*. En investigaciones futuras se buscará extender el análisis a nuevas variantes de páginas *phishing*, realizar un análisis más profundo de las mejores estrategias de calibrado de parámetros, y ampliar la muestra de *datasets* a analizar. Se buscará también contemplar la detección frente a métodos de evasión.

## REFERENCIAS

Abdelhamid, N., Thabtah, F. y Abdel-jaber, H. (2017). Phishing Detection: A Recent Intelligent machine learning Comparison based on Models Content and Features. *IEEE Explorer*, *6*. doi:10.1109/ISI.2017.8004877

Abu-Nimeh, S., Nappa, D., Wang, X. y Nair, S. (2007). A Comparison of machine learning Techniques for Phishing Detection. *ACM Digital Library*, *10*. doi:10,.1145/1299015.1299021

Al-Janabi, M., De Quincey, E. y Andras, P. (2017). Using Supervised Machine Learning Algorithms to Detect Suspicious URLs in Online Social Networks. *ACM Digital Library*, *8*. doi:10.1145/3110025.3116201

Bulakh, V. y Gupta, M. (2016). Countering Phishing from Brands' Vantage Point. *ACM Digital Library*, *8*. doi:10.1145/2875475,2875478

Campo, D. (20 de noviembre de 2017). MachineLearningPhishing. *GitHub*. Recuperado de https://github.com/diegoocampoh/MachineLearningPhishing

Chen, T.-C., Dick, S. y Miller, J. (2010). Detecting Visually Similar Web Pages: Application to Phishing Detection. *ACM Digital Library*, *38*. doi:10.1145/3282373.3282422

Chiew, K. L., Tan, C. L., Wong, K. S., Yong, K. S. y Tiong, W. K. (2019). A New Hybrid Ensemble Feature Selection Framework for Machine Learning-Based Phishing Detection System. *Science Direct*, *14*. doi:10.1016/j.ins2019.01.064

Cuzzocrea, A., Martinelli, F., y Mercaldo, F. (2018). Applying Machine Learning Techniques to Detect and Analyze Web Phishing Attacks. *ACM Digital Library*, *5*. doi:10,1145/3282373,3282422

ESET Security Report Latinoamérica 2017. (2017). Recuperado de https://www.welivesecurity.com/wp-content/uploads/2017/04/eset-security-report-2017.pdf

Hota, H. S., Shrivas, A. K. y Hota, R. (2018). An Ensemble Model for Detecting Phishing Attack with Proposed Remove-Replace Feature Selection Technique. *Science Direct*, *8*. doi:10.1016/j.procs.2018.05.103

Islam Mamun, M. S., Rathore, M. A., Lashkari, A. H., Stakhanova, N. y Ghorbani, A. A. (2016). Detecting Malicious URLs Using Lexical Analysis. *Springer Link*, 16. doi:10,1007/978-3-319-46298-1_30

Jain, A. K. y Gupta, B. B. (2016). A novel Approach to Protect against Phishing Attacks at Client Side Using Auto-Updated White-List. *Springer Open*, *11*. doi:10.1186/s13635-016-0034-3

Mao, J., Bian, J., Tian, W., Zhu, S., Wei, T., Li, A., y Liang, Z, (2018), Detecting Phishing Websites via Aggregation Analysis of Page Layouts. *Science Direct*, 7, doi:10,1016/j,procs,2018,03,053

Medvet, E., Kirda, E. y Kruegel, C. (2008). Visual-Similarity-Based Phishing Detection. *ACM Digital Library*, *6*. doi:10.1145/1460877.1460905

Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill Science.

Mourtaji, Y., Bouhorma, P. y Alghazzawi, P. (2017). Perception of a New Framework for Detecting Phishing Web Pages. *ACM Digital Library*, 6. doi:10.1145/3175628.3175633

Rajab, M. (2018). An Anti-Phishing Method based on Feature Analysis. *ACM Digital Library*, *7*. doi:10.1145/3184066.3184082

Sanglerdsinlapachai, N. y Rungsawang, A. (2010). Web Phishing Detection Using Classifier Ensemble. *ACM Digital Library*, *6*. doi:10.1145/1967486,1967521

Tan, C. L. (2018). Phishing Dataset for Machine Learning: Feature Evaluation. *Mendeley.* doi:10.17632/h3cgnj8hft.1

URL dataset (ISCX-URL-2016). (2016). *UNB*. Recuperado de https://www.unb.ca/cic/datasets/url-2016.html

# PERFILES

**JOEL FERNANDO PALOMINO MASCO**

Correo electrónico: j.palomino@pucp.edu.pe

Ingeniero electrónico graduado de la Pontificia Universidad Católica, posee una Maestría en Ingeniería Mecatrónica otorgada por la Universidad de Nagasaki, en Japón. Ha participado en proyectos de investigación financiados por FONDECYT y ha desarrollado *hardware* electrónico para la captura de imágenes de mamíferos grandes con capacidad de transmisión inalámbrica. Tiene amplia experiencia en el desarrollo de aplicaciones con microprocesadores, sobre todo en la aplicación de sistemas operativos para dispositivos con memoria limitada. De ahí viene su principal interés en el estudio y aplicación de las redes de sensores para aplicaciones de vida inteligente y confort.

**JUAN ANTONIO PACO FERNÁNDEZ**

Correo electrónico: jpaco@pucp.edu.pe

Ingeniero electrónico colegiado con estudios de posgrado en Telecomunicaciones por INICTEL, UNI, actualmente cursa la Maestría en Gerencia de Tecnologías de Información en CENTRUM, PUCP. Tiene amplia experiencia en la gestión y ejecución de proyectos de telecomunicaciones, especialmente en redes inalámbricas; sistemas satelitales VSAT; sistemas de cableado estructurado; fibra óptica e infraestructura de red. Ha laborado en empresas del sector telecomunicaciones como Telefónica del Perú y TELEREP. En el ámbito académico, actualmente, colabora en un grupo de investigación en telecomunicaciones rurales (GTR, PUCP). Es instructor CISCO en el Programa CCNA de la Academia Local PUCP, profesor de cursos de formación continua en el Centro de Educación Continua de la PUCP, expositor o panelista en diversos eventos tecnológicos y autor de artículos científicos, así como de libros relacionados con sus actividades de investigación.

**MICHEL ZARZOSA ROJAS**

Correo electrónico: michel.zaro@gmail.com

Ingeniero electrónico por la Universidad Nacional Mayor de San Marcos con estudios de especialización en Seguridad de la Información en TECSUP. Cuenta con una amplia experiencia en proyectos de investigación aplicada I+D+I, especialmente en desarrollo de redes inalámbricas (Xbee, LoRa) y el diseño de *software* y *hardware* de sistemas embebidos. Ha participado en el diseño e instalación de un sistema de registro de temperatura y humedad en las zonas afectadas por las heladas a más de 3500 m s. n. m. en el marco del proyecto Mi Abrigo 1, 2 y 4 ejecutado por Foncodes. En el ámbito académico,

actualmente, colabora con el grupo de investigación en telecomunicaciones rurales (GTR PUCP) y cuenta con cuatro artículos científicos de los cuales es el autor principal de dos de ellos.

**RUBÉN AHOMED CHÁVEZ**

Correo electrónico: aahomed@ulima.edu.pe

Es licenciado en Ciencias de la Comunicación por la Universidad de Lima, MBA por la Universidad de Surrey de Inglaterra y actualmente es estudiante del programa doctoral en el Centrum PUCP Business School. Es docente en la Facultad de Comunicación de la Universidad de Lima y en la Maestría de Marketing en Esan Graduate School of Business. Actualmente labora como gerente general en Grupo La República Publicaciones. Sus áreas de interés son la transformación digital empresarial, los medios de comunicación y la comunicación publicitaria.

**LENNIN PAUL QUIROZ VILLALOBOS**

Correo electrónico: lquirozv@ulima.edu.pe

Es ingeniero electrónico e ingeniero de *software* por la Universidad Nacional Mayor de San Marcos y magíster en Ingeniería Mecánica y Aeroespacial. Profesionalmente, en el campo de la electrónica, se ha desempeñado como ingeniero de campo en el área de instrumentación, automatización y control en diversas empresas. En el campo del *software* se ha desempeñado como ingeniero de proyectos usando diferentes lenguajes y tecnologías. Actualmente se desempeña como docente e investigador en la Universidad de Lima. Sus áreas de interés son la ingeniería aeroespacial, robótica, inteligencia artificial, *computer vision*, ingeniería de *software*, automatización y control.

**WILLIAM-ROGELIO MARCHAND-NIÑO**

Correo electrónico: william.marchand@unas.edu.pe

Magíster en Dirección Estratégica de TI por la Universidad de Piura e ingeniero de sistemas por la Universidad Nacional del Centro del Perú, es *pentestester* senior en Open-Sec LLC/EIRL. Miembro senior de IEEE-Región 9, sección Perú. Docente en la Universidad Nacional Agraria de la Selva, es orador en diversos eventos de seguridad informática a nivel nacional (PeruHack, BSides Perú) e internacional (Ekoparty, EcuaHack, CIBSI-TIBETS, BSides Latam).

**EDWIN JESÚS VEGA VENTOCILLA**

Correo electrónico: edwin.vega@unas.edu.pe

Ingeniero en Informática y Sistemas de la Universidad Nacional Agraria de la Selva (UNAS), se desempeñó como Director del Centro de Tecnologías de la Información y Comunicación de la UNAS. Coordinador del área de gestión de soluciones de TI en la Universidad. Docente universitario en la Facultad de Ingeniería en Informática y Sistemas de la UNAS, con más de 10 años de experiencia.

**ANDRES EDUARDO MONCADA VARGAS**

Correo electrónico: 20152102@aloe.ulima.edu.pe

Egresado de la Carrera de Ingeniería de Sistemas de la Universidad de Lima. Ha laborado en el Departamento Universitario de Informática y Sistemas (DUIS) de la misma institución, en el área de Calidad, donde revisó y probó aplicaciones y procesos desarrollados para la universidad, orientados a los docentes y a los alumnos. Sus áreas de interés son el desarrollo de aplicaciones web y móviles, arquitectura de *software* y ciberseguridad.

UNIVERSIDAD DE LIMA

Uso no estándar e implementación exitosa del Protocolo I2C para un sistema de medición de temperatura en aldeas andinas a gran altitud

Revisión de literatura sobre las barreras a la transformación digital y su relación con el rendimiento financiero

Software in the Loop para la implementación de un sistema de piloto automático para aeronaves de ala fija

Modelo Balanced Scorecard para los controles críticos de seguridad informática según el Center for Internet Security (CIS)

Comparación de técnicas de *machine learning* para detección de sitios web de *phishing*

ISSN 1993-4912