'Framework' basado en imágenes parametrizadas sobre ResNet para identificar intrusiones en 'smartwatches' u otros dispositivos afines. (Un eje singular de la publicación “Estado del arte de la ciencia de datos en el idioma español y su

aplicación en el campo de la Inteligencia Artificial”)

**Juan Antonio Lloret Egea, Celia Medina Lloret,
Adrián Hernández González, Diana Díaz Raboso, Carlos Campos,
Kimberly Riveros Guzmán, Adrián Pérez Herrera,
Luis Miguel Cortés Carballo, Héctor Miguel Terrés Lloret**

License: [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License \(CC-BY-NC-ND 4.0\)](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Abstracto

Se ha definido un *framework*^[1] conceptual y algebraicamente, inexistente hasta ahora en su morfología, y pionero en aplicación en el campo de la Inteligencia Artificial (IA) de forma conjunta; e implementado en laboratorio, en sus aspectos más estructurales, como un modelo completamente operacional. Su mayor aportación a la IA a nivel cualitativo es aplicar la conversión o transducción de parámetros obtenidos con lógica ternaria^[1] (sistemas multivaluados)^[2] y asociarlos a una imagen, que es analizada mediante una red residual artificial ResNet34^{[2]·[3]} para que nos advierta de una intrusión. El campo de aplicación de este *framework* va desde *smartwatches*, *tablets* y PC's hasta la domótica basada en el estándar KNX^[4].

Abstract note

The full version of this document in the English language will be available in this [link](#).

Código QR de la
publicación

Este marco propone para la IA una ingeniería inversa de tal modo que partiendo de principios matemáticos conocidos y revisables, aplicados en una imagen gráfica en 2D para detectar intrusiones, sea escrutada la privacidad y la seguridad de un dispositivo mediante la inteligencia artificial para mitigar estas lesiones a los usuarios finales.

Asimismo se postula el *framework* de manera que cumpla con las mayores expectativas de ser una herramienta útil para la sociedad y el ser humano, alineado con el «[Dictamen del Comité Europeo de las Regiones – Libro Blanco sobre la inteligencia artificial – Un enfoque europeo orientado a la excelencia y la confianza](#)» y que contribuya a evitar la mala *praxis* de generar cajas negras^[5] en el empleo de la inteligencia artificial^[6] haciendo de ella algo inexplicable.

La confianza y la explicabilidad en la tecnología es un objetivo a corto, medio y largo plazo para que la sociedad no quede empeñada en manos de tecnologías espurias que hipotecan un desarrollo social adecuado. «Estas tecnologías han ampliado las oportunidades de libertad de expresión y de movilización social, cívica y política, pero suscitan a la vez graves preocupaciones»[7].

I.-Framework

1.- Justificación de la necesidad

Existe una amplia disponibilidad de dispositivos «inteligentes» que todos tenemos a nuestro alcance, desde cámaras fotográficas digitales, pasando por *tablets* y *smartwatches*. El problema ocurre cuando el desconocimiento en el manejo de estos aparatos electrónicos se extiende a un gran porcentaje de la población [8], pues ignoran los riesgos que pueden acarrear. Si bien es cierto que un dispositivo electrónico es **inteligente** por funcionar de forma interactiva y autónoma, no debemos olvidar que para que **algo** aporte beneficios y soluciones en situaciones concretas, ese **algo** debe conocer previamente lo que está ocurriendo. Todo esto se traduce en una única palabra: **datos**.

Uno de los dispositivos electrónicos que más se ha puesto de moda en los últimos tiempos es el *smartwatch*. Los *smartwatches* tienen sensores que identifican modelos o patrones de comportamiento humano junto con técnicas de aprendizaje automático, el teorema de Bayes, el procesamiento de datos o el también conocido como método k de los vecinos más cercanos [9]. Estos procedimientos generan un gran volumen de información con el que pretendemos cumplir el objetivo de precisar en los resultados emitidos. Estos sensores son muy útiles para monitorizar la actividad humana: caminar, montar en bicicleta, *running* y subir o bajar escaleras [10].

El uso de *smartwatches* puede suponer una grave amenaza para la seguridad de niños y adolescentes [11]. Los fallos de seguridad más importantes de los relojes de bajo coste se producen en las aplicaciones y en la conexión con los servidores que almacenan los datos. Las marcas de *smartwatch* más utilizadas por menores que hemos analizado son: Carl Kids Watch, hellOO! Children's Smart Watch, SMA-WATCH-M2 y GATOR Watch. Los problemas más recurrentes que encontramos son: fallos en la implementación de certificados para conexiones seguras HTTPS [12] e información relativa al registro electrónico sin cifrar [13]. Sin embargo, en dispositivos de marcas

más reconocidas como Nokia [14], Samsung o Huawei [15], estos problemas no ocurren con frecuencia, ya que se realizan a través de conexiones cifradas.

En este punto nos parece interesante comparar las características que afectan a la seguridad de las diferentes marcas de relojes. Esta es una forma de conocer el nivel de confianza que debemos depositar en cada una de ellas. El criterio seguido para elegir las marcas es tener en cuenta el mayor número de ventas del mercado: Samsung, Apple, Fitbit y Garmin.

SAMSUNG		APPLE	
Recopilación de Datos		Recopilación de Datos	
¿Se realiza de forma anónima?	No	¿Se realiza de forma anónima?	No
¿Se almacenan datos del GPS?	Si	¿Se almacenan datos del GPS?	Si
¿Datos de salud?	Si	¿Datos de salud?	Si
Información en línea de menores de 13 años	No	Información en línea de menores de 13 años	Si
Datos Compartidos		Datos Compartidos	
La información recopilada se comparte con terceros	Si	La información recopilada se comparte con terceros	Si
Los datos se comparten para análisis	Si	Los datos se comparten para análisis	No disponible
Los datos se comparten para publicidad y / o marketing de terceros	Si	Los datos se comparten para publicidad y / o marketing de terceros	No
Los datos se venden o alquilan a terceros.	Si	Los datos se venden o alquilan a terceros.	No
¿Los terceros están autorizados a conocer la identidad del usuario?	Si	¿Los terceros están autorizados a conocer la identidad del usuario?	Si
Seguridad		Seguridad	
Protección de cuenta mediante 2FA	No.	Protección de cuenta mediante 2FA	Si
¿Todos los datos son encriptados en su envío? (Smartwatch - Smartphone)	No, solo algunos	¿Todos los datos son encriptados en su envío? (Smartwatch - Smartphone)	Si
¿Todos los datos son encriptados en su envío? (Smartphone - Servidor)	Si	¿Todos los datos son encriptados en su envío? (Smartphone - Servidor)	Si
¿Se almacenan todos los datos encriptados?	No, solo algunos	¿Se almacenan todos los datos encriptados?	Si
¿Se avisa al usuario si se produce una brecha de seguridad?	No.	¿Se avisa al usuario si se produce una brecha de seguridad?	Si
¿Se realizan auditorías de seguridad en el producto?	No disponible	¿Se realizan auditorías de seguridad en el producto?	No disponible

Imagen 1

Crédito de imagen: Seguridad en dispositivos Samsung y Apple por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/seguridad-en-los-dispositivos-samsung-y-apple/>)

FITBIT

Recopilación de Datos	
¿Se realiza de forma anonima?	No
¿Se almacenan datos del GPS?	Si
¿Datos de salud?	Si
Información en línea de menores de 13 años	Si

Datos Compartidos	
La información recopilada se comparte con terceros	Si
Los datos se comparten para análisis	Si
Los datos se comparten para publicidad y / o marketing de terceros	Si
Los datos se venden o alquilan a terceros.	No
¿Los terceros están autorizados a conocer la identidad del usuario?	No disponible

Seguridad	
Protección de cuenta mediante 2FA	No disponible
¿Todos los datos son encriptados en su envío? (Smartwatch - Smartphone)	Si
¿Todos los datos son encriptados en su envío? (Smartphone - Servidor)	No disponible
¿Se almacenan todos los datos encriptados?	No disponible
¿Se avisa al usuario si se produce una brecha de seguridad?	No disponible
¿Se realizan auditorías de seguridad en el producto?	No disponible

GARMIN

Recopilación de Datos	
¿Se realiza de forma anonima?	No
¿Se almacenan datos del GPS?	Si
¿Datos de salud?	Si
Información en línea de menores de 13 años	Si

Datos Compartidos	
La información recopilada se comparte con terceros	Si
Los datos se comparten para análisis	Si
Los datos se comparten para publicidad y / o marketing de terceros	No disponible
Los datos se venden o alquilan a terceros.	No
¿Los terceros están autorizados a conocer la identidad del usuario?	No disponible

Seguridad	
Protección de cuenta mediante 2FA	No disponible
¿Todos los datos son encriptados en su envío? (Smartwatch - Smartphone)	No, solo algunos
¿Todos los datos son encriptados en su envío? (Smartphone - Servidor)	No disponible
¿Se almacenan todos los datos encriptados?	No disponible
¿Se avisa al usuario si se produce una brecha de seguridad?	No disponible
¿Se realizan auditorías de seguridad en el producto?	No disponible

Imagen 2

Crédito de imagen: Seguridad en dispositivos Fitbit y Garmin por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/seguridad-en-dispositivos-fitbit-y-garmin/>)

En la recopilación de datos, Samsung es la única marca que no compila información *online* de menores de 13 años [16]. Sin embargo, esto no ocurre con Apple [17], Fitbit [18] y Garmin [19], que no hacen distinción de edad. Asimismo, todas las marcas elegidas para nuestro estudio comparten datos con terceros para analizar métricas y comparar resultados. De acuerdo con estas referencias, **solo en Samsung los datos de los usuarios son vendidos o alquilados. Al contrario de lo que ocurre con Apple; pues es la única de las empresas analizadas que no comparte datos con fines de publicidad y/o marketing.** Por tanto, llegamos a la conclusión de que todas las marcas, al final, realizan una recopilación de datos no anónimos aunque con ciertos puntos de disimilitud.

2.- Definición algebraica y cálculo del alcance del *framework*[20]

La sistemática empleada se basa en la dualidad o binomio de parámetros determinantes de una intrusión & imagen asociada. La determinación de cuáles han de

ser estos parámetros debe ser fruto de la experiencia en el campo de la ciberseguridad[21] y que más adelante nosotros propondremos y serán expuestos.

2.1.-Definición algebraica

Dado un número determinado de parámetros de intrusión denominado «n», que es una sucesión numérica de términos $\in \mathbb{N} \wedge a_n = b^2 / b \geq 3$, característico para cada una de las instancias del *framework* a determinar; dado $(\vec{P}_x)$, compuesto por un grupo de términos o elementos vectores axiales dependiente de «n» $\rightarrow (\vec{P}_{x(1...n)})$ cuyo modulo $|\vec{P}_{x(1...n)}|$ sólo toma los valores ternarios del intervalo cerrado $[0,1, U]^{[1]}$, y un número máximo de vectores de $(\vec{P}_x)$ denominado «m» $\in \mathbb{Z} \rightarrow [\vec{P}_1, \vec{P}_2, \vec{P}_3... \vec{P}_m]_0^{f:n \rightarrow m}$ determinado su valor por $m = 3^n$; y dada una sucesión numérica continua de pares de términos ordenados $(x_{(1...n)}, y_i)$ que aplican biyectivamente que un elemento en el grupo $(\vec{P}; x)$ tiene su imagen en el grupo $(\vec{I}mg; y)$, $\forall \text{ término } x \rightarrow y = x \leq m \wedge i = (n - 1, n)$, definimos a este *framework* como:

Un **Sistema Vector de intrusión** $\rightarrow \vec{S}V_{intrusión} = [\vec{P}x_{(1...n)}, \vec{I}mg_{y_i}]_0^{3^{(n)}}$. Que cumple que una imagen asociada, o transducida, es función de un *array* de parámetros $(p) \rightarrow \{p_1, p_2, p_3, p_4... p_n\}$; vectorizados cada conjunto enésimo de éstos como $(\vec{P}_x) / :$
 $\vec{I}mg_{y_i} = f(\vec{P}_{x(1...n)})_0^{3^{(n)}}$, y que para cada uno de los vectores del conjunto $(\vec{P}_x)$ tiene una relación biyectiva [22] con su imagen reflejada $(\vec{I}mg_y)$.

Así $(\vec{I}mg_y)$ es la consecuencia de la transformación lineal del sistema de vectores $(\vec{P}_x)$, definido éste por el *array* de parámetros de intrusión (p) , y que se construye dinámicamente $\forall x \leq m$ mediante la siguiente expresión:

$$\vec{I}mg_{y_i} \Leftrightarrow [\vec{P}_{x_2} - \vec{P}_{x_1}, \vec{P}_{x_3} - \vec{P}_{x_2}, \vec{P}_{x_4} - \vec{P}_{x_3}, ..., \vec{P}_{x_n} - \vec{P}_{x_{(n-1)}}, \vec{P}_{x_1} - \vec{P}_{x_n}] \forall n \in \mathbb{N}.$$

$$\forall \vec{I}mg_{y_i} \Leftrightarrow \vec{P}_{x_n} - \vec{P}_{x_{(n-1)}} / \forall \text{ término } x \rightarrow y = x \leq m \wedge i = (n - 1, n)$$

- La imagen poligonal cerrada, que asociará un determinado *array* de parámetros intrusivos, para que ésta sea valorada por la Inteligencia Artificial (IA), vendrá definida por la siguiente expresión vectorial :

$$\left(\sum_{j=1}^{j=n} (\vec{P}x_n - \vec{P}x_{(n-1)}) \right) - (\vec{P}x_1 - \vec{P}x_n) = 0$$

A) DOMINIOS DEL FRAMEWORK

1) El dominio $\{D_1\} \forall \vec{P}_{x(1...n)} \in \vec{\mathbb{N}} \wedge : xn \in [0, 1, U]$.

2) El dominio $\{D_2\}$ de $\vec{I}mg_{y_i} = f(\vec{P}_{x(1...n)})_0^{3^{(n)}} \in \vec{\mathbb{R}}^2 \rightarrow f : \mathbb{N} \rightarrow \mathbb{R}^2$.

3) El domino $\{D_3\}$ del grupo $(\vec{P}_x) \in \vec{\mathbb{N}}^n$.

$$(\vec{P}_x) \Leftrightarrow (\vec{P}_1, \vec{P}_2, \vec{P}_3, \dots, \vec{P}_m) / D_1 [0, 1, U] : \forall \vec{P}_{x(1\dots n)} \in \vec{\mathbb{N}} \wedge \forall \vec{P}_x \in \vec{\mathbb{N}}^n$$

4) El domino $\{D_4\}(\vec{I}mg_y) \in \vec{\mathbb{R}}^n$.

B) COORDENADAS POLARES DEL *FRAMEWORK*

La transformación lineal anterior de $(\vec{I}mg_y)$ constituiría una línea vectorial poligonal cerrada. Que en coordenadas polares, de fácil implementación mediante gráficos radiales³, implica para $\vec{P}_{x(1\dots n)}$ un sistema generador de vectores $\vec{r} = |mod|_\alpha$ de valor modular $(0, 1, U)$ y de ángulo α cuyo valor estará comprendido entre 0° y 360° dependiendo éste del número de parámetros usados en el escaneo de la intrusión asociados a sus ejes, y que para el caso de $n=9$ parámetros, y para una instancia determinada «x», $[\vec{P}_{x1}, \vec{P}_{x2}, \vec{P}_{x3} \dots \vec{P}_{x9}]$, α tendrá un valor (o un desfase entre los vectores del sistema) $(\vec{P}_{x(1\dots 9)}\alpha)$ de $40^\circ \rightarrow = \frac{360^\circ}{9}$.

2.1.1.- Naturaleza de las dimensiones del Sistema Vector de intrusión

$$\vec{S}V_{intrusión} = [\vec{P}x_{(1\dots n)}, \vec{I}mg_{y_i}]_0^{3^{(n)}}$$

Al utilizarse aquí, como propuesta inicial de representación, gráficos radiales para generar imágenes y usar estos ejes a modo de radios vectores de una circunferencia en el plano separados un cierto ángulo α entre ellos, la parte que nos interesa, la gráfica, se ubica por completo en la aplicación lineal $f : \mathbb{N} \rightarrow \mathbb{R}^2$.

Ahora bien, atendiendo a las coordenadas de cada vector y su naturaleza algebraica, la dimensión real sería la que sigue:

1) El grupo de vectores $\vec{P}x_{(1\dots n)} \{Dim_1\}\mathbb{N}=1$.

2) El grupo de vectores $\vec{I}mg_{y_i} \{Dim_2\}\mathbb{R}=2$.

3) El grupo de vectores $\vec{P}_x \{Dim_3\}\mathbb{N}=n$.

4) El grupo de vectores $\vec{I}mg_y \{Dim_4\}\mathbb{R}=n$.

2.1.2.- Propiedades algebraicas de $\vec{P}x_{(1\dots n)}, \vec{I}mg_{y_i}$ que conforman

$$\vec{S}V_{intrusión} = [\vec{P}x_{(1\dots n)}, \vec{I}mg_{y_i}]_0^{3^{(n)}}$$

Las propiedades a analizar para este sistema vectorial van desde: elemento neutro^[23], elemento simétrico^[24], propiedad asociativa^[25], conmutativa^[26] y distributiva^[27]; pasando por el análisis de $\vec{P}x_{(1\dots n)}, \vec{I}mg_{y_i}$ como grupo abeliano^[28], anillo^[29],

cuerpo[30] y espacio vectorial[31]... Se propone para una siguiente revisión de esta publicación su desarrollo, comprobación y demostración. Asimismo se omite en esta revisión en beneficio de una mayor conceptualización general del *framework* y de su focalización en el desarrollo efectuado en el laboratorio.

2.1.3.- Inferencias de la definición algebraica

A este modelo matemático así definido puede aplicársele la teoría del Campo del Álgebra Lineal[32]. Asimismo a los gráficos radiales, que encierran áreas en sus polígonos irregulares definidos por sus vértices, se les puede operar algebraicamente aplicando el determinante de Gauss o fórmula del área de Gauss[33] y sus propiedades adscritas. Y también, propiedades de la lógica ternaria aplicable y del álgebra combinacional de Boole[34]; entre otros... Por lo que el número de teoremas y corolarios[35] a postular para este marco, y su aplicación en la IA y la detección de intrusiones, es motivo aún de un estudio amplio de posibilidades y vasto en posibles aplicaciones diferentes de las que aquí se describirán o enunciarán.

Imagen 3

Crédito imagen: Ejemplo de la fórmula del área de Gauss, Wikipedia, [Isalar](#) derivative

work: [Nat2](#) (talk) - [Polygon_area_formula.jpg](#), Mangosta
(<https://mangosta.org/formula-del-area-de-gauss/>)

2.2.- Cálculo del alcance y sistemática de uso

El número máximo «m» de vectores posibles con el muestreo de $n=9$ parámetros y lógica ternaria (0,1,U) es de $\rightarrow m = 3^9$; $\rightarrow \Delta \vec{r}_{(n=9)} = 19.683$ vectores de intrusión. Que se irían incrementando conforme aumentáramos el número de parámetros «n» de 16, a 25, 36, etc., siguiendo una lógica de matrices cuadradas y su propia sucesión numérica definida anteriormente. Para un número $n \leq 9$ no queda suficientemente justificado el uso de la IA porque cualquier lenguaje de programación mediante un algoritmo o librería específica podría realizar esto de forma más o menos trivial, sin embargo, cuando $n > 9$ sí queda justificada por la potencia de cálculo y la rapidez⁴ en el manejo de datos masivos que tienen actualmente los sistemas gráficos asociados a la inteligencia artificial [36] de acuerdo a la tendencia actual de los proveedores de servicios, servidores y tarjetas gráficas, por ejemplo, como Google o IBM; entre otros. Y sin duda esta rapidez y potencia de la IA es totalmente diferenciadora en valores de «n» superiores a 25, porque éste ya supondría $\rightarrow$

$m = 3^{25}$; $\rightarrow \Delta \vec{r}_{(n=25)} = 19.68347.288.609.443$ vectores..., casi un billón de vectores. Lo que otorgaría al sistema una brillante precisión en sus estimaciones de intrusión. La metodología para la obtención de estos parámetros se propone inicialmente que sea mediante la captura de reglas en *logs* de IDS's [37], y el empleo de técnicas conocidas de monitorización de sistemas (*apps*, etc.); pero esto obviamente es flexible y abierto a otra metodología de obtención y el *framework* lo admitiría sin mayor problema.

Los parámetros de intrusión sirven como base para construir un modelo gráfico de positivos, negativos y de entrenamiento explícitamente separados y ordenados en unidades contenedoras, carpetas o directorios. Para cada nueva identidad de parámetros e imagen, una vez transferida la imagen a la unidad contenedora asignada, la inteligencia artificial (previamente entrenada) nos determinará si es o no es una intrusión. Asimismo para la decisión final de ¿si es o no es una intrusión?, el marco propone adicionar la posibilidad de que un humano afirme que lo dictaminado por la IA sea efectivamente una intrusión, o no. Supone por tanto la inteligencia artificial una sonda que pone en sobre aviso y escruta todas las intrusiones vía

$\vec{S}V_{intrusión} = [\vec{P}x_{(1...n)}, \vec{I}mg_{y_i}]_0^{3(n)}$ para que finalmente sea el técnico humano quien decida. Por otro lado la gestión de una base de datos relacional de positivos ya declarados, anexos al sistema que integre o encapsule el marco, conjuntamente con la actuación de la IA para los positivos que no lo están aún, permitirá generar una base de datos de conocimiento mejorada que irá creciendo de acuerdo al número de peticiones de los usuarios que decidan utilizarlo.

3.- Definiciones singulares y un ejemplo

3.1.- Definición en Programación Orientada a Objetos (POO)[38]

Al *framework* que hemos creado le podemos abstraer la idea de que definiría una clase denominada ‘Intrusión’, y la herencia de esta clase en otra determinada, definida mediante 9 parámetros, a su vez generaría la clase denominada ‘Intrusión_{base9}’, y que es factible de implementar con cualquier lenguaje de programación existente actualmente. Para cada instancia de esa clase, obtendríamos un objeto de $\vec{SV}_{intrusión} = [\vec{Px}_{(1...n)}, \vec{Imgy}_i]_0^{3^{(n)}}$ para su posterior clasificación como intrusión [True/False].

3.2.- Definición de 9 parámetros

Estos nueve parámetros significan una instancia del marco general descrito anteriormente. Por lo tanto heredaría todas las características de éste. Y donde los valores (0,1,U) especifican: apagado o no usado, encendido o en uso e indeterminado; respectivamente.

Figura 1		
Núm. Parámetro ⁽ⁿ⁼⁹⁾ asociado al $\vec{Px}_{(1...n)}$	Definición del parámetro	Procedimiento de obtener recomendado
n=1→P _{x1}	Destino de <i>web</i> (url) desconocido o no autorizado	Regla Snort[39] IDS o similar
n=2→P _{x2}	Comunicación no cifrada (Https) ^[6]	“ “
n=3→P _{x3}	Traspaso de datos locales a otro almacenamiento	“ “
n=4→P _{x4}	Uso o conexión a BSSID[40] no permitida o desconocida	Regla Kismet[41] -WIDS o similar
n=5→P _{x5}	Uso o conexión a Puerto Bluetooth[42] no permitida o desconocida	“ “
n=6→P _{x6}	Uso o conexión a GPS[43] no permitida o no autorizada	“ “

$n=7 \rightarrow P_{x7}$	Uso o conexión a cámara no permitido o desconocido	Monitor de sistema o similar
$n=8 \rightarrow P_{x8}$	Uso de dispositivo de audio (micro / altavoz) no permitido o desconocido	“ “
$n=9 \rightarrow P_{x9}$	Monitorización de parámetros físicos (velocidad, temperatura, batería) con resultado anormal	“ “

3.3.- Un ejemplo de aplicación

De forma general se da $\vec{Img}_{y_i} = f(\vec{P}_{x(1...n)})_0^{3(n)}$, **veamos un ejemplo cualquiera** con 9 parámetros de una imagen, en un instante de tiempo T1 determinado para una instancia $x = 1$; $\rightarrow \vec{Img1}_i = f(\vec{P}_{1(1...9)}) \rightarrow$
 $\vec{Img1}_i = f(\vec{P}_{11}, \vec{P}_{12}, \vec{P}_{13}, \vec{P}_{14}, \vec{P}_{15}, \vec{P}_{16}, \vec{P}_{17}, \vec{P}_{18}, \vec{P}_{19})$. Y la imagen gráfica a escrutar por la inteligencia artificial se crearía a partir de la transformación lineal siguiente:

$\vec{Img1}_i \Leftrightarrow [\vec{P}_{12} - \vec{P}_{11}, \vec{P}_{13} - \vec{P}_{12}, \vec{P}_{14} - \vec{P}_{13}, ..., \vec{P}_{19} - \vec{P}_{18}, \vec{P}_{11} - \vec{P}_{19}]$ Y en la que los vectores parciales del vector ($\vec{Img1}_i$) se irían construyendo dinámicamente en el sentido de las agujas del reloj a partir de los vectores $\vec{P}_{1(1...9)}$ con un desfase entre ellos de 40º.

$$\vec{Img1}_1 = \vec{P}_{12} - \vec{P}_{11}; \vec{Img1}_2 = \vec{P}_{13} - \vec{P}_{12}; \vec{Img1}_3 = \vec{P}_{14} - \vec{P}_{13}; \vec{Img1}_4 = \vec{P}_{15} - \vec{P}_{14}; \vec{Img1}_5 = \vec{P}_{16} - \vec{P}_{15}; \vec{Img1}_6 = \vec{P}_{17} - \vec{P}_{16}; \vec{Img1}_7 = \vec{P}_{18} - \vec{P}_{17}; \vec{Img1}_8 = \vec{P}_{19} - \vec{P}_{18}; \vec{Img1}_9 = \vec{P}_{11} - \vec{P}_{19}$$

Por lo que asignando valores, previamente escaneados, al *array* de parámetros (p1)= (0,1,U,0,1,1,0,U,0), éstos definirán de forma intrínseca ($\vec{P}_{1(1...9)}$), y su imagen gráfica asociada $\vec{Img1}_i$ se construiría como la figura de abajo. Donde a nivel gráfico se ha asignado el valor «3» a las indeterminaciones «U». Y «1» ó «2» que corresponderían a los valores binarios de (0,1), respectivamente.

Imagen 4

Crédito imagen: Ejemplo de captura de parámetros por Juan Antonio Lloret Egea, 2021, Mangosta (https://mangosta.org/i_parametros-3/)

II.- Laboratorio del *framework*

4.- Implementación de los 9 parámetros

4.1.- Destino de url desconocido o no autorizado

```
log tcp $HOME_NET any -> !DireccionIPConfianza any (sid: 1000001; rev: 001;)
```

```
log tcp any any -> !DireccionIPConfianza any (sid: 1000001; rev: 001;)
```

La regla puede ser definida de dos maneras, una inicial en la que utilizamos la variable **“HOME_NET”** previamente definida, en la que vamos a especificar cuál es nuestra red, o una segunda forma en la que no especificamos la red que estamos utilizando sino que creamos la regla con **“any”** y a pesar de ser menos eficiente en cuanto a recursos, nos aseguramos de que funcione en cualquier situación. La regla puede

leerse de la siguiente forma: "Crea un *log* cuando se utilice el protocolo TCP desde nuestra red para enviar paquetes desde cualquier puerto a una dirección distinta a la dirección de confianza en cualquier puerto."

4.2.- Comunicación no cifrada (https)

```
log tcp $HOME_NET any -> any any (content:"http"; sid: 1000002; rev: 001; )
```

```
log tcp any any -> any any (content:"http"; sid: 1000002; rev: 001; )
```

La regla puede ser definida de dos formas, una inicial en la que utilizamos la variable "HOME_NET" previamente definida, en la que vamos a especificar cuál es nuestra red o una segunda forma en la que no especificamos la red que estamos utilizando sino que creamos la regla con "any". Ésta, a pesar de ser menos eficiente en cuanto a recursos, nos aseguramos de que funcione en cualquier situación. La regla puede leerse de la siguiente manera: "Crea un log cuando se utilice el protocolo TCP para enviar paquetes desde cualquier puerto a cualquier dirección y puerto con contenido HTTP".

4.3.- Traspaso de datos locales a otro almacenamiento

```
log tcp $HOME_NET !puerto_confianza -> any any(sid: 1000003; rev: 001;)
```

```
log tcp any !puerto_confianza -> any any(sid: 1000003; rev: 001;)
```

La regla puede leerse de la siguiente manera: "Crea un log cuando se utilice el protocolo TCP para enviar paquetes desde cualquier puerto que no sea el de confianza a cualquier dirección y puerto.

4.4.- Uso o conexión a BSSID no permitida o desconocida

Para dispositivos Windows, se puede utilizar Kismet o bien integrarlo en Python, utilizando el modulo subprocess (a partir de Windows 7) que nos permite obtener el estado de los procesos de los dispositivos y la respuesta que nos ofrece el sistema operativo tras introducir una sentencia en la consola del mismo. En este caso para conocer el BSSID al que esta conectado el dispositivo utilizaremos: "netsh wlan show interfaces", que nos devolverá entre otros muchos datos el BSSID al que esta conectado el equipo. Si este BSSID no es un BSSID reconocido, obtendremos un valor de "1" a la hora de realizar el gráfico posterior que utilizará la Inteligencia Artificial para determinar si estamos ante una intrusión o no. Si el BSSID es reconocido, almacenaremos un valor "0" y si por el contrario no es posible determinarlo, tendremos un valor indeterminado en este parámetro.

Para Android, se puede utilizar **WifiInfo**, clase definida por Google para manejar las conexiones del dispositivo [44] y que tiene como herencia la clase Objeto [45]. Además se utilizará el método getBSSID () para obtener el el BSSID, en forma de una dirección MAC de seis bytes: XX: XX: XX: XX: XX: XX. Nuevamente si el BSSID es reconocido, se almacena como un "0", si no lo es, se almacena como un "1". Además, si no es posible determinar el estado de la conexión en el dispositivo, se almacenará como indeterminado.

4.5.- Uso o conexión a Puerto Bluetooth no permitida o desconocida

Para dispositivos Windows, se puede utilizar Kismet o bien integrarlo en Python, utilizando la librería PyBluez que nos permite obtener el estado de la conexión bluetooth.

Para Android, se puede utilizar **BluetoothProfile** interfaz (colección de métodos y constantes), definida por Google para manejar el Bluetooth en los dispositivos Android. Además necesitaremos utilizar otra interfaz: **ServiceListener** que nos permitirá listar de manera más rápida los estados (constantes) que buscamos:

STATE_CONNECTING: Dispositivo en estado de conexión.

STATE_DISCONNECTING: Dispositivo en estado de desconexión.

STATE_CONNECTED: Dispositivo conectado.

STATE_DISCONNECTED: Dispositivo desconectado.

Estos estados pueden parecer iguales dos a dos, la diferencia es que en los dos primeros obtendremos el estado de nuestra conexión Bluetooth, es decir, si tenemos activado o no el Bluetooth, y en los dos últimos recibiremos información sobre si nuestra conexión está o no activa. Sin embargo, un dispositivo puede tener el Bluetooth activo pero no estar conectado a ningún otro dispositivo [46].

4.6.- Uso o conexión a GPS no permitida o no autorizada

Para dispositivos Windows se puede utilizar Kismet o bien integrarlo en Python, utilizando la librería gpsd que nos permite obtener la posición del GPS al ejecutar el *script*. Si no existe conexión, no obtendremos ninguna posición y por lo tanto podremos almacenar el valor como "0". Por el contrario, si obtenemos una posición cualquiera, almacenaremos el valor como "1".

Para Android se puede utilizar **GnssStatus.Callback**, clase definida por Google para manejar el sistema global de navegación por satélite en los dispositivos Android. Otra de las posibilidades era trabajar con **GpsStatus** pero no se podría monitorizar

“GLONASS”, “GALILEO”, “BEIDOU”, etc. Por otra parte, utilizaremos el método definido **onStarted()** que nos avisa cuando GNSS se activa y **onStopped()**, su análogo para cuando el proceso se detiene [47].

4.7.- Uso o conexión a cámara no permitido o desconocido

Para dispositivos Windows se utiliza la biblioteca cv2 en Python, que nos permite tener control sobre los dispositivos de vídeo, en este caso para comprobar si la cámara está activa o no, al ejecutar el fichero. En el caso de que la cámara esté *online* para el entorno de pruebas nos aparece un mensaje como este:

Imagen 5

Crédito de imagen: Salida por consola estado del cámara online por Adrián Hernández, 2021, Mangosta (https://mangosta.org/ii_parametros-2/)

Y además, almacenaremos el valor como “1” en la variable determinada para realizar el gráfico con el resto de parámetros. Si por el contrario la cámara está *offline*, en el entorno de pruebas nos aparece el siguiente mensaje:

Imagen 6

Crédito de imagen: Salida por consola estado del cámara offline por Adrián Hernández, 2021, Mangosta (https://mangosta.org/ii_parametros-2/)

En este caso se almacenaría el valor “0” a la hora de realizar el gráfico.

Para dispositivos Android, utilizamos **android.hardware.camera2** [48] y utilizamos el *callback* de Android: **CameraDevice.StateCallback** [49] que incluye el método **onOpened(CameraDevice camera)**, que nos devuelve si la cámara está abierta o no.

4.8.- Uso de dispositivo de audio (micro / altavoz) no permitido o desconocido

Para dispositivos Windows, se utiliza Python y la biblioteca PyAudio, la cual nos permite capturar una señal de audio a través del micrófono y posteriormente visualizar

su representación temporal [50]. El objetivo es capturar la onda de audio en el instante de tiempo que consideremos y si esa onda no es plana, el micrófono estaría activo.

Imagen 7

Crédito de imagen: Estado del micrófono mediante el uso de las ondas de audio

por Adrián Hernández, 2021, Mangosta
(https://mangosta.org/iii_parametros-2/)

Para Android, se puede utilizar **AudioManager**, una clase definida por Google para manejar el micrófono en los dispositivos Android [51]. Además, necesitaremos utilizar la clase Context [52] y la String AUDIO_SERVICE. Asimismo, podemos comprobar desde la API 11 de Android (correspondiente a HoneyComb - Android 3.0.x) si el micrófono está en uso, utilizando la constante definida: MODE_IN_COMMUNICATION / MODE_IN_CALL.

Los pasos a seguir para crear la estructura serían:

- 1.- Utilizar getSystemService(java.lang.String)
- 2.- Incluir la clase Context y la String AUDIO_SERVICE:
getSystemService(Context.AUDIO_SERVICE)
- 3.- Incluir la sentencia creada en el paso 2 en la clase AudioManager:
(AudioManager)context.getSystemService(Context.AUDIO_SERVICE).
- 4.- Actualmente el método getMode asociado a AudioManager nos ofrece 5 resultados:

MODE_NORMAL: No hay llamadas ni acciones establecidas.

MODE_RINGTONE: Hay una petición al micrófono.

MODE_CALL_SCREENING: Hay una llamada conectada pero el audio no está en uso.

MODE_IN_CALL: Hay una llamada telefónica.

MODE_IN_COMMUNICATION: Existe alguna aplicación que está realizando comunicaciones de audio/video o VoIP.

Por tanto si el método nos devuelve un "MODE_IN_CALL" or "MODE_IN_COMMUNICATION", tendremos el micrófono activo.

4.9.- Monitorización de parámetros físicos (velocidad, temperatura, batería) con resultado anormal

Para dispositivos Windows, se utilizará Python y la biblioteca psutil, que nos permite monitorizar y recuperar información sobre el sistema tales como CPU, RAM, uso de disco, red o batería. Además es multiplataforma lo que nos permitirá en un futuro integrarla sobre cualquier sistema operativo [50].

Se puede generar el *log* utilizando solo psutil y estableciendo a partir de qué porcentaje de utilización y durante cuánto tiempo sostenido. Este porcentaje sería considerado un uso anormal en los parámetros físicos: velocidad, temperatura y batería. También si queremos monitorizar en tiempo real lo que ocurre tanto con la CPU como con la RAM, debemos utilizar la biblioteca Matplotlib para generar gráficos a partir de datos contenidos en listas o *arrays* [50].

Imagen 8

Crédito de imagen: Monitorización de parámetros físicos por
Adrián Hernández, 2021, Mangosta
(https://mangosta.org/iv_parametros-2/)

4.10.- Gráficos obtenidos tras analizar los parámetros

Los gráficos polares que obtenemos al controlar cada uno de los parámetros y que le permiten a la Inteligencia Artificial clasificar las imágenes y posteriormente predecir si

se trata de una intrusión o no, siguen este estilo:

Imagen 9

Crédito de imagen: Gráfico polar por Adrián Hernández, 2021, Mangosta
(https://mangosta.org/vi_parametros-2/)

5.- Diagramación y sistemática en la implementación del *framework*

Para el diseño, la compilación y la ejecución del conjunto del *framework* en laboratorio cabe plantearse diferentes posibilidades y el empleo de lenguajes de programación. Por un lado está el uso propio de la IA respecto a su entrenamiento y clasificación de imágenes, que denominaremos sistemática (**S1**), y que en ella se ha utilizado Python. Por otro lado se plantea el diseño de una BD de almacenamiento y uso de datos; tanto para dispositivos conocidos ya determinados (si son o no son positivos en intrusión) y también por determinar, para esta sistemática (**S2**) se ha utilizado SQL. Y por último se hace preciso la gestión propia del entorno en su conjunto o *shell* y su ensamblado general, para esta sistemática (**S3**) se ha utilizado Java. (El lenguaje SQL resulta singular en la sistemática S2. Para las sistemáticas S1 y S3 podría usarse un único lenguaje de programación a elegir entre Python y Java, cada uno con sus ventajas y desventajas respecto al uso de la IA y respecto a los sistemas operativos objetivos en dispositivos *smart*).

Inicialmente plantearemos los diagramas de flujo de la lógica a emplear, general y singular; después adicionaremos algunos elementos complementarios como el paso a paso del uso de la IA y el diseño de la base de datos y finalmente plantearemos la

codificación esencial de los distintos lenguajes de programación que intervienen: SQL, Python y Java.

5.1.- Diagramación General del *framework*

En este apartado abordaremos el esqueleto del proyecto. Para ello hemos planteado la diagramación general, dividida en tres partes: la relativa al usuario, la base de datos y la IA.. El principal objetivo es obtener una visión global de los procedimientos acometidos, de tal forma que podamos entender todos y cada uno de ellos.

Imagen 10

Crédito de imagen: Diagrama de flujo general del framework por Diana Díaz, 2021, Mangosta (<https://mangosta.org/diagramacion-general-framework/>)

El diagrama general ha sido diseñado con la ayuda de la aplicación de diagramación de código abierto Drawio [53]. Esta diagramación, en concreto, contiene otros dos

diagramas; la conexión Java con la base de datos y el uso de fast.ai + Google Colaboratory, también recogidos en la leyenda de la parte inferior del diagrama.

En primer lugar, comenzamos con la instalación de la aplicación. Al acceder a la app, el usuario es preguntado si desea que su dispositivo sea analizado, en caso de afirmar su respuesta, el proceso continúa, de no ser así, la aplicación se cierra. Si el proceso continúa, el usuario debe introducir los datos requeridos: el tipo de dispositivo, la marca y el modelo. Asimismo, se realiza una conexión en Java con la base de datos, de manera que esta información sea cotejada y por consecuencia, permita comprobar si este dispositivo se encuentra entre nuestros registros y dispone de la información suficiente para realizar el análisis. Si está registrado, se verifica con el histórico si el dispositivo está realizando alguna acción sin consentimiento del usuario. Por el contrario, si no está en la base de datos, se requiere la autorización del usuario para seguir con el proceso. En caso de que el usuario no consienta este análisis, la aplicación se cierra inmediatamente. Además, la inteligencia artificial se emplea en este punto para comprobar si existe algún riesgo de intrusión. Para llevar a cabo este proceso, se realiza una captura de 9 parámetros predeterminados, que asignándoles un modelo preentrenado ResNet34 con fast.ai indica si realmente hay una intrusión. Con el resultado de la IA, un profesional analizará los datos obtenidos y decidirá si son coherentes. Si es así, los datos se almacenan en la base de datos y posteriormente se le comunicaría al usuario, en caso contrario, estos datos se descartarían.

5.2.- Diagramación de Java

En este punto se expone la diagramación Java, designada como sistemática (**S3**). El objetivo principal de la diagramación de Java es definir, por un lado, cómo Java accede a la base de datos mediante el *driver* JDBC y, por otro lado, cómo efectúa una comprobación mediante una consulta SQL.

Imagen 11

Crédito de imagen: Diagrama de Java por Kimberly Riveros, 2021, Mangosta (<https://mangosta.org/diagramacion-de-java/>)

El *driver* JDBC, por sus siglas en inglés, (Java Database Connectivity) facilita la conexión entre un sistema gestor de base de datos (SGBD) y cualquier *software* que

trabaje con Java [54]. Adems del *driver* JDBC, hemos utilizado el sistema gestor MySQL para gestionar los archivos de la base de datos y la herramienta phpMyAdmin, disponible en XAMPP, para manejar la administraci3n de MySQL. Tambin, hemos trabajado en el entorno de Java conjuntamente con Eclipse. Eclipse es un *software* de entorno de desarrollo integrado para conectar la base de datos con Java [55]. Es importante aadir que en el diseo de la diagramaci3n de Java hemos utilizado como herramienta base el programa Draw.io, al igual que se ha hecho en el resto de diagramaciones (**S1**, **S2** y **S3**).

En el siguiente apartado explicaremos c3mo se realiza la conexi3n entre la base de datos y Java, tanto en local como en la nube, utilizando la mquina virtual Eclipse. Una vez instalados todos los aplicativos, se incluye el *driver* JDBC en nuestro proyecto de Eclipse. A modo de recomendaci3n, es conveniente tener creada la base de datos desde el principio para evitar posibles errores de conexi3n.

Referenced Libraries

mysql-connector-java-8.0.25.jar

Imagen 12

Crdito de imagen: Conexi3n a la base de datos - Java por Kimberly Riveros, 2021, Mangosta
(<https://mangosta.org/conexion-a-la-base-de-datos-java/>)

Posteriormente definimos en Eclipse el objeto "connection" para establecer la conexi3n y le indicamos la ruta del servidor junto con la base de datos a la que queramos acceder. Nosotros hemos llamado a la base de datos "seguridad", tanto en local como en la nube. Adems es necesario especificar el usuario y la contrasea porque de lo contrario, no sera posible acceder a ella. Por 3ltimo, iniciamos la b3squeda del *host* y el puerto en phpMyAdmin para incluirlos en la ruta y as realizar la conexi3n desde la nube.

("jdbc:mysql://localhost:3306/seguridad", "usuario", "contrasea"

Imagen 13

Acceso al c3digo para docentes y estudiantes: <https://mangosta.org/conexion-a-la-base-de-datos-java001/>

Cada conexi3n, en local o en la nube, es independiente entre s, es decir, cada una se hace en una clase distinta y lo 3nico diferente que tiene cada clase en Eclipse es la

ruta.

Imagen 14

Crédito de imagen: Conexión a la base de datos
Java

por Kimberly Riveros, 2021, Mangosta

(<https://mangosta.org/conexion-a-la-base-de-datos-java002/>)

Finalmente una vez comprobada la conexión de la base de datos, podemos realizar las consultas que deseemos, escribiendo en lenguaje SQL. Por ejemplo, ejecutamos una consulta con la ayuda de los objetos, “Statement” y “ResultSet”. Como resultado se muestra por pantalla la tabla y el registro, entre otras, pues éstas han sido llamadas desde Eclipse.

Este punto será ampliado en el apartado relacionado con [la codificación Java](#).

5.3.- Diagramación de Python

La diagramación de Python está dividida en dos, por un lado, la que hace referencia a la conexión de la base de datos con Python y, por otro, la que alude al uso de la inteligencia artificial en Python.

5.3.1 - Diagramación conexión base de datos con Python

La siguiente diagramación expone la manera de conectar la base de datos mediante el lenguaje de programación Python, definida como sistemática (**S1**) en la plataforma Google Colaboratory.

Imagen 15

Crédito de imagen: Diagrama de Python
por Diana Díaz, 2021, Mangosta (<https://mangosta.org/diagramacion-conexion-base-de-datos-con-python/>)

Las herramientas utilizadas para llevar a cabo este procedimiento son: Google Colaboratory, una aplicación de Google para crear entornos de máquinas virtuales

basado en Jupyter Notebook [56]. Jupyter, un archivo de extensión IPYNB que combina una multitud de celdas con código o texto Markdown ([OpenScienceLabs](#)). El conector MySQL Connector/Python está escrito en Python y nos permite conectarnos a una base de datos externa, sin embargo, es necesario un medio de interacción. Por esta razón, hacemos uso del paquete PyMySQL [58].

El siguiente paso es conectar la base de datos con Python. En primer lugar, accedemos a Google Colab y creamos un cuaderno nuevo, el cual, de forma automática, se convierte en un cuaderno Jupyter. Una vez abierto, instalamos el conector MySQL Connector/Python con "pip", un sistema de gestión de paquetes escrito en Python [59]. Una vez instalado, lo importamos. De igual forma, el paquete PyMySQL debe ser instalado. Llegados a este punto, si no nos ha dado ningún error durante el proceso, ya tenemos todo preparado para la conexión a la base de datos desde nuestro cuaderno de Google Colab. Para conectarnos creamos una variable en la que mediante el paquete PyMySQL, alojamos la información de la ruta: nombre, usuario, contraseña y nombre de la base de datos a la que nos queremos conectar. Convertimos la variable en un cursor, ya que mediante la misma, vamos a realizar una operación de búsqueda en la base de datos. Teniendo el cursor podemos ejecutar sobre él la consulta que queramos y, posteriormente, imprimirla por pantalla con el resultado obtenido. Una vez ejecutadas todas las consultas requeridas, podemos cerrar la conexión a la base de datos.

Este punto será ampliado en el apartado relacionado con [la codificación Python](#).

5.3.2 - Diagramación uso de Inteligencia Artificial en Python

La diagramación que se muestra a continuación maqueta la forma de detectar intrusiones, denominada sistemática (**S3**). La finalidad de la diagramación del uso de la Inteligencia Artificial en Python es estudiar las intrusiones a partir de imágenes focalizadas en el IDS. En nuestro caso, hemos elegido imágenes de gatos y tigres. Asimismo, en el [siguiente apartado](#) están detallados los pasos a seguir para detectar intrusiones utilizando ResNet34, cuadernos Jupyter y Google Colaboratory.

CLASIFICACIÓN DE IMÁGENES (RESNET 34)

Diagrama Python v2.1 - Adrián Hernández González

Imagen 16

Crédito de imagen: Diagramación uso de la inteligencia artificial en Python por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/diagramacion-uso-de-la-inteligencia-artificial-en-python/>)

Para llevar a cabo este procedimiento, hemos utilizado Drawio para hacer el diagrama, Google Drive y Google Colaboratory, que incluye el lenguaje de programación Python.

Además de estas herramientas, trabajamos con fast.ai, esto es, una biblioteca de aprendizaje profundo que permite la carga de modelos preentrenados como ResNet34 para una posterior clasificación de las imágenes [60]. Por tanto, no es necesario descargar ningún tipo de *software*, pues todo estos procesos los llevaremos a cabo en la nube.

Accedemos a Google Drive para crear tres carpetas: “train”, “validation” y “test”. Dentro de cada una de ellas, creamos dos subcarpetas llamadas “gato” y “tigre” en las que se agruparán las imágenes, dependiendo del grupo al que pertenezcan.

Imagen 17

Crédito imagen: Almacenamiento imágenes de gatos por Kimberly Riveros, 2021, Mangosta (<https://mangosta.org/gatos/>)

Imagen 18

Crédito de imagen: Almacenamiento imágenes de tigres por Kimberly Riveros, 2021, Mangosta (<https://mangosta.org/tigres/>)

Posteriormente accedemos a Google Colaboratory y en la primera celda, indicamos la ruta en la que estará ubicado nuestro *dataset* (conjunto de imágenes). De esta forma, podremos entrenar el modelo con esas imágenes. Por consiguiente, creamos el modelo preentrenado con la arquitectura ResNet34. Cuando hablamos de modelo preentrenado, nos referimos a un modelo que permite que otros modelos obtengan resultados avanzados sin necesidad de disponer de grandes cantidades de cómputo, tiempo y paciencia [61]. Esta es una forma de encontrar la técnica de entrenamiento adecuada. Por otro lado, si queremos usar el modelo en un futuro, no tenemos que volver a entrenarlo. ResNet34, por sus siglas en inglés (*residual neural network*), es un modelo preentrenado de ImageNet. ImageNet es un *software* de reconocimiento de objetivos visuales [62]. Tanto ResNet34 como ImageNet nos ayudan a mejorar el rendimiento y optimizar los resultados en lo que a la detección de intrusiones se refiere.

El siguiente paso es realizar un ciclo de entrenamiento con un número determinado de épocas para entrenarlo ([véase en el apartado referido al proceso de detectar intrusiones con Inteligencia Artificial](#)), y así la Inteligencia Artificial pueda estudiar todo el conjunto de imágenes. Para interpretar los resultados creamos una matriz de confusión que nos ayudará a evaluar el rendimiento de nuestro modelo de clasificación de imágenes. Esta matriz compara los valores reales con los predichos y así, podremos

ver cómo está funcionando nuestro modelo en la identificación de imágenes. De hecho, en la siguiente representación gráfica podemos comprobar que la inteligencia artificial ha fallado únicamente en una imagen, pues clasifica esa imagen como un tigre cuando en realidad es un gato.

Prediction/Actual/Loss/Probability

Imagen 19

Crédito de imagen: Respuesta Inteligencia Artificial por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/i_respuesta-inteligencia-artificial/)

Imagen 20

Crédito de imagen: Respuesta Inteligencia Artificial (matriz de confusión) por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/ii_respuesta-inteligencia-artificial/)

6.- Paso a paso en el proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory

Google Colab es un entorno de trabajo que nos permite ejecutar cuadernos Jupyter en el navegador. Entre sus ventajas destacamos la necesidad de no configurarlo

previamente y la obtención de acceso gratuito a GPU's (Tesla T4 o Tesla K80). Por lo tanto, ofrece la posibilidad de entrenar y utilizar la inteligencia artificial sin emplear recursos *hardware* propios y/o pagar una suscripción para su uso.

Un cuaderno Jupyter es un documento que permite ejecutar celdas de código en vivo, texto plano, ecuaciones, etc. La estructura que siguen estos cuadernos es una lista ordenada de entrada/salida. El nombre Jupyter proviene de: **Julia + Python + R**, que son los tres lenguajes de programación presentes en los cuadernos Jupyter. Los dos componentes principales de Jupyter Notebook son un conjunto de núcleos (*Interpreter*) y el *Dashboard*. De igual forma, cambiando el núcleo del cuaderno podemos ejecutar otros lenguajes desde Google Colab como podría ser Java.

Una vez que tenemos el conocimiento suficiente acerca de los componentes que vamos a utilizar para dar forma a nuestra inteligencia artificial, es el momento de explicar el procedimiento sobre cómo lo vamos a llevar a cabo.

El primer paso es preparar el entorno de Google Colab, ya que por defecto no vamos a poder acceder sin activarlo. Para ello, desde nuestro Google Drive hacemos clic en Google Colaboratory.

Imagen 21

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial/>)

O bien, si no nos aparece la opción, hacemos clic en la siguiente URL:

<https://colab.research.google.com/>

Una vez estemos dentro de Google Colab, tenemos los cuadernos predeterminados por Google para iniciarnos en Python desde la nube. Sin embargo, también podemos crear un cuaderno en blanco para empezar a realizar las pruebas pertinentes. Para realizar esta acción, hacemos clic en la opción: "Nuevo Cuaderno".

Imagen 22

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial001/>)

Ahora, el nuevo cuaderno creado tiene la siguiente estructura:

Imagen 23

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial002/>)

La celda que nos aparece inicialmente nos permite escribir código en Python y ejecutarlo directamente, utilizando el botón “Play”, situado en la misma celda.

Imagen 24

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial003/>)

Por defecto, Google Colaboratory no viene configurado para utilizar GPU's, así que debemos prepararlo. Para ello, es necesario ir a la opción de "Entorno de ejecución", "Cambiar tipo de entorno de ejecución".

Imagen 25

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial004/>)

Una vez situados en la pestaña, hacer clic en el desplegable con la palabra “none” y reemplazarlo por “GPU”.

Configuración del cuaderno

Acelerador por hardware

GPU

Para sacar el máximo partido a Colab,
evita usar una GPU si no es necesario
para tu trabajo. [Más información](#)

☐ Omitir resultado de las celdas de código al guardar este cuaderno

CANCELAR GUARDAR

Imagen 26

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial005/>)

En este momento, Google asignará una GPU para nuestro cuaderno Jupyter, la cual podremos comprobar, utilizando la siguiente instrucción en una celda de código:

```

!nvidia-smi

Mon Jul 5 09:34:56 2021

+-----+
| NVIDIA-SMI 465.27                Driver Version: 460.32.03    CUDA Version: 11.2     |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+
|  0 Tesla T4             Off      | 00000000:00:04.0 Off |                    0 |
| N/A   66C    P8      11W /  70W |  0MiB / 15109MiB |      0%      Default |
+-----+-----+

+-----+
| Processes:                         |
| GPU   GI    CI          PID    Type   Process name          GPU Memory |
|   ID   ID     ID              |              |           Usage     |
+-----+-----+
| No running processes found      |
+-----+

```

Imagen 27

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial006/>)

Como se puede apreciar en la imagen, nos asignan una Tesla T4, aunque podría haber sido una Tesla K80, ya que son las dos GPU's que Google ofrece de manera gratuita. Ahora, ya tenemos configurado el entorno de Google Colab para trabajar sobre él con inteligencia artificial, aunque para entrenar nuestro modelo necesitamos crear la estructura de carpetas que vamos a utilizar. Esto se realizará desde Google Drive para reducir su complejidad, pero podría realizarse en un equipo local, configurando el entorno Colab para ello.

Para realizar las primeras pruebas, utilizamos imágenes de gatos y tigres, persiguiendo que la inteligencia artificial sea capaz, dada una imagen que nunca ha visto, de determinar si se trata de una imagen de un tigre o de un gato. La estructura inicial consta de tres carpetas, a las que podemos nombrar como queramos, pero que por comodidad, se recomienda atribuirles la siguiente nomenclatura:

Mi unidad > Proyectos > GatosOTigres ▾

Carpetas

Imagen 28

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial007/>)

Una carpeta *Test*, atribuida a las imágenes que la inteligencia artificial nunca ha visto y servirá para comprobar que la IA es capaz de clasificar correctamente las representaciones gráficas. Otra carpeta *Train*, con fines de entrenar nuestro modelo. Y, en este caso, sí que habrá una diferenciación entre tigres y gatos, y la inteligencia artificial sabrá en cada caso si la imagen que está visualizando es un gato o un tigre. Una carpeta llamada *Validation* que nos servirá para validar nuestro modelo. También habrá una diferenciación dentro de la misma. Esta carpeta nos permitirá conocer cómo clasificaría la inteligencia artificial la imagen y lo que es realmente dicha imagen. Dentro de las carpetas *Train* y *Validation*, tendremos la siguiente estructura:

Mi unidad > Proyectos > GatosOTigres > Train ▾

Carpetas

Imagen 29

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial008/>)

Y dentro de cada carpeta, imágenes correspondientes a gatos o a tigres. Las imágenes de cada carpeta deben ser distintas, además, cuanto mayor número de figuras utilicemos, mayor precisión tendrá la inteligencia artificial para clasificar otras representaciones gráficas.

Una vez creada la estructura de carpetas y preparado el entorno de Google Colab, empezamos a trabajar con la inteligencia artificial. Para ello, ejecutamos el código referente a conectar nuestro entorno de Google Colab con Google Drive, donde se encuentran las carpetas que servirán a la IA desde el cuaderno creado previamente.

```
[ ] from google.colab import drive  
    drive.mount('/content/gdrive', force_remount=True)
```

Mounted at /content/gdrive

Imagen 30

Acceso al código para docentes y estudiantes: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc001/>

Una vez asociados los dos entornos, definimos qué rutas se van a utilizar, en este caso `root_dir` y `base_dir`.

```
[ ] root_dir = "/gdrive/My Drive/"  
    base_dir = root_dir + 'GatosOTigres/'
```

Imagen 31

Acceso al código para docentes y estudiantes: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc002/>

De igual manera, definimos el *path*, es decir, la ruta de nuestro directorio principal a utilizar.

```
[ ] path = Path('gdrive/My Drive/Proyectos/GatosOTigres/')
```

Imagen 32

Acceso al código para docentes y estudiantes: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc003/>

Para realizar el entrenamiento del modelo de inteligencia artificial es necesario importar `fastai`. `Restnet34` debe su nombre a “*residual network*” y se refiere a una red residual que fue introducida por Microsoft en 2015. Una vez que conocemos `fast.ai`, importamos la biblioteca:

```
[ ] from fastai import *  
    from fastai.vision import *  
    from fastai.metrics import error_rate
```

Imagen 33

Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc004/>

Estas tres instrucciones sirven para importar las bibliotecas `fastai` y `fastai vision` completas, además del “**error_rate**” de **fastai metrics**. “Error_rate” sirve para determinar el grado de error en nuestro modelo entrenado, es decir, para saber si según nuestros criterios, debemos mejorar el modelo o por el contrario, el error obtenido es asumible y podemos utilizar el modelo.

Una vez importadas las bibliotecas necesarias, comprobamos que nuestra configuración del cuaderno sea correcta. Para ello, imprimimos en pantalla una imagen aleatoria utilizando el *path* que hemos definido previamente y con las siguientes instrucciones: **open_image**(y la ruta de la imagen utilizando la variable *path*) e `img.show()`.

```
img = open_image(path/'Train/Gatos/cat.4028.jpg')  
print(img.data.shape)  
img.show()
```

```
torch.Size([3, 390, 459])
```

Imagen 34

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial013/>) (.) Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc005/>

Cuando comprobamos que la imagen se muestra correctamente, creamos los datos (muestra de imágenes) que nuestro modelo va a utilizar. Para empezar, vamos a definir el batch size o tamaño de batch, que nos permite indicar el número de imágenes que transitamos al mismo tiempo a la memoria. Como medida preventiva, es recomendable no utilizar un batch size muy alto o podríamos causar un error de memoria en la tarjeta gráfica y por consiguiente, el detenimiento del proceso. Para definir el tamaño de batch, escribimos el siguiente código:

```
[ ] bs = 6
```

Imagen 35

Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc006/>

En este caso el número 6, indica la cantidad de imágenes transitadas a la memoria al mismo tiempo. Una vez hemos asignado el batch size, asignamos los datos que vamos a utilizar, en este caso imágenes. Para ello, utilizamos la siguiente celda de código:

```
data = ImageDataBunch.from_folder(path, train="Train", valid='Validation',  
                                  test='Test', size=100, bs=bs,  
                                  num_workers=4).normalize(imagenet_stats)
```

Imagen 36

Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc007/>

En esta celda definimos la variable *data*, partiendo de la estructura de carpetas que hemos creado previamente (*Train*, *Validation* y *Test*) e indicando que esas son las carpetas que se tendrían que utilizar para sus variables predeterminadas: *train*, *valid* y *test*.

Ya definido el conjunto de datos que vamos a utilizar, inicializamos nuestro modelo. Antes de eso, creamos el modelo y cargamos Resnet34, utilizando el siguiente código:

```
learn = cnn_learner(data, models.resnet34, metrics=error_rate)
```

Imagen 37

Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc008/>

Utilizamos *cnn_learner* y le indicamos el conjunto de datos a utilizar (*data*), el modelo que vamos a utilizar (**models.resnet34**) y además, queremos conocer el **error_rate** importado anteriormente de **fastai.metrics**. Esta línea de código, por tanto, nos sirve para crear y cargar nuestro modelo, por lo que el siguiente paso es entrenarlo. Para eso utilizamos:

```
learn.fit_one_cycle(6)
```

Imagen 38

Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc009/>

El número entre paréntesis indica el número de “**epoch**” que vamos a utilizar para el entrenamiento. Pero, ¿qué es un *epoch*? Llamamos *epoch* o época al proceso de pasar todos los datos por la Inteligencia Artificial. En nuestro caso, todas las imágenes de la carpeta llamada entrenamiento pasan por la inteligencia artificial. Al realizar un entrenamiento con 6 épocas, todas las imágenes de la carpeta *Train* pasarán 6 veces por la IA.

```
[ ] learn.fit_one_cycle(6)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.159619	0.056623	0.018987	00:18
1	0.162316	0.037526	0.017405	00:23
2	0.137577	0.030045	0.007911	00:23
3	0.127109	0.015616	0.001582	00:22
4	0.143309	0.022998	0.012658	00:23
5	0.128840	0.014749	0.003165	00:22

Imagen 39

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial018/>) (.) Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial-resnet34-cuadernos-jupyter-y-google-colaboratoryc009bis/>

Al finalizar el entrenamiento obtendremos el `error_rate` final de nuestro modelo de entrenamiento. En el caso de la imagen superior sería 0.003165 o del 0.31%, aproximadamente. Otra de las cuestiones es conocer para qué sirve el `error_rate`. Ofrece información sobre lo preciso que es nuestro modelo y nos permite decidir si realizar un entrenamiento con más épocas o utilizar el modelo. Si no estamos seguros de cuánta precisión significa un error de 0.31%, podemos imprimir una matriz de confusión donde se ve más claro utilizando:

```
[ ] interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```

Imagen 40

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial: ResNet34, cuadernos Jupyter y Google Colaboratory por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial-019/>) (.) Acceso al código para docentes y profesores:

En esta matriz de confusión se comparan el eje de abscisas (eje X), las predicciones que realiza la Inteligencia Artificial (gatos o tigres) y en el eje de ordenadas (eje Y), lo que realmente es la imagen. En este caso, la IA acierta en 331 imágenes de gatos y 299 imágenes de tigres, fallando en 2 imágenes en las que la IA predice que son gatos cuando realmente son tigres. Por lo tanto, el error que partiría de esta matriz de confusión sería de $\rightarrow \frac{2}{632} = 0.316\%$.

Una vez consideramos que el error es asumible, procedemos a guardar el modelo para poder utilizarlo más adelante. Este procedimiento se realiza con la siguiente celda de código:

```
[ ] learn.save(dest/'modelo1')
```

Imagen 41

Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc010/>

Entre paréntesis, indicamos la ruta y el nombre con el que vamos a guardar nuestro modelo. Una vez que el modelo está guardado, podemos probar su funcionamiento con imágenes que nunca ha visto la IA. Para este proceso, utilizamos una nueva celda de código que contiene la siguiente información:

```
[ ] img = learn.data.test_ds[520][0]
img.show()
```

```
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3658: UserWarning: Th
"The default behavior for interpolate/upsample with float scale_factor changed "
```

Imagen 42

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial020/>). (.) Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc011/>

En este punto volvemos a utilizar el modelo almacenado en *learn* y le pasamos la imagen de la carpeta *Test*, en este caso la 520. Podemos imprimir la imagen dentro del propio Google Colab para verificar que realmente se cumple lo que la IA va a predecir en el siguiente paso. Para imprimir la imagen se utiliza **img.show()**. Para pedirle a la inteligencia artificial que analice la imagen que hemos cargado, tendremos que usar la siguiente línea de código:


```
[ ] learn.predict(img)

(Category tensor(1), tensor(1), tensor([6.4536e-06, 9.9999e-01]))
```

Imagen 43

Crédito imagen: Proceso de detectar intrusiones con Inteligencia Artificial por Adrián Hernández, 2021, Mangosta (<https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificial021/>) (.) Acceso al código para docentes y alumnos: <https://mangosta.org/proceso-de-detectar-intrusiones-con-inteligencia-artificialc012/>

En este caso, volvemos a realizar una llamada a nuestro modelo con *learn* y utilizamos `predict(img)` para utilizar el modelo entrenado. El resultado en este caso nos lo ofrece como `tensor(0)` para los gatos y emplea `tensor(1)` para los tigres. Además entre corchetes nos da la probabilidad de que en este caso sea un tigre (99.9%).

7.- Creación y estructura de la base de datos

Se ha creado una base de datos atendiendo a la necesidad de recoger los datos que ofrecen los *smartwatches* u otros dispositivos afines para comprobar si todos estos datos son autorizados y conocidos por el usuario o por el contrario, se trata de un uso no autorizado. Para ello, tenemos en cuenta dos factores: los datos que se necesitan para poder acceder a la información del usuario y los datos que entrañan un riesgo por acceder sin consentimiento. En este sentido, centramos el foco de atención en los datos que recogen los *smartwatches*, ofreciendo una tabla adicional (TSalud).

La herramienta utilizada, al igual que en los diagramas anteriores, es Drawio. También hemos usado el lenguaje SQL así como MySQL y PhpmyAdmin.

Dos de los pilares fundamentales de esta base de datos son la importancia de los booleanos y la lógica trivalente, sobre la que hablaremos más adelante. ¿Pero para qué valen los campos booleanos? ¿Por qué hemos decidido emplearlos en nuestra base de datos? fundamentalmente para encontrar información de calidad en el menor tiempo posible y realizar búsquedas combinadas de varios términos [63].

Para clarificar posibles dudas sobre la base de datos, explicamos el diseño y la estructuración de las 5 tablas. La primera tabla y principal es "TDispositivo". Esta tabla se compone de 3 campos que recogen el ID (número identificativo y que se encuentra en todas la tablas como hilo conductor), el nombre de la marca (varchar) y el modelo (varchar). La segunda tabla "TTipoDispositivo" dispone de 6 campos: ID, y otros 5 campos booleanos (*smartwatch*, PC portátil, PC sobremesa, *tablet* y

smartphone) que responderán con verdadero o falso en función del tipo de dispositivo del que se trate. La tercera tabla "TAccesoNoAutorizado" se compone de 4 campos: ID y otros 3 booleanos (cámara, micrófono y GPS). En esta tabla se pretende conocer si el dispositivo tiene acceso a alguno de estos 3. La cuarta tabla "TSalud" recoge un total de 8 campos: el ID y siete booleanos, (frecuencia cardíaca, sueño, registro automático de ejercicio, estrés, nivel de oxígeno, ciclo menstrual y pasos). Se pretende que esta tabla nos informe sobre la información que recoge el dispositivo. La quinta y última tabla es "TDeteccion" y tiene tan sólo dos campos: ID y un booleano (positivo). Esta tabla es de gran importancia porque en ella se almacenará, tras comprobar el resto de datos de las otras tablas, si el dispositivo es positivo o no, es decir, si tiene o no una intrusión.

Diagrama entidad - relación

Imagen 44

Crédito imagen: Diagrama entidad-relación por Diana Díaz, 2021, Mangosta (<https://mangosta.org/diagrama-entidad-relacion/>)

Como podemos apreciar en la imagen este diagrama se compone de tres elementos fundamentales:

- Entidad, representado como rectángulos que muestran los nombres de nuestras tablas. Por ejemplo: TAccesoNoAutorizado.
- Atributos, con forma ovalada, definen las características de la entidad. Siguiendo el ejemplo, aquí las vemos representados por nID, iCamara, etc).

● Relaciones, con forma romboide, muestran las conexiones entre unas entidades y otras. En este caso, la conexión entre una entidad y otra sería: TDispositivo tiene TAccesoNoAutorizado).

Lógica trivalente

Para el diseño de la base de datos nos hemos centrado en la velocidad de las consultas, ya que es fundamental para un buen desarrollo del proyecto. Por este motivo se utilizan campos booleanos.

La lógica bivalente distingue dos valores: verdadero o falso, representados por 1 y 0 respectivamente [28]. En este caso, iremos más allá y aplicaremos la lógica ternaria o trivalente, es decir, un sistema lógico en el que se presentan 3 valores: verdadero, falso o indeterminado. El tercer valor se interpreta como la posibilidad de que algo no sea ni verdadero ni falso, sino indefinido. En este contexto, el indicador categorizado como indeterminado, pues muestra ausencia de valor, se representa así: "null".

Modelo Físico

Una vez visualizado el diseño y analizada la estructura de las tablas, estudiamos la base de datos desde el modelo físico, en el que observamos que la *primary key* de todas las tablas es nID.

Imagen 45

Crédito imagen: Modelo físico por Diana Díaz, 2021, Mangosta
[\(https://mangosta.org/modelo-fisico-2/\)](https://mangosta.org/modelo-fisico-2/)

Uso de SQL y su sistema de gestión de base de datos

SQL2 (*Structured Query Language* o lenguaje de consulta estructurada) pasó a ser el estándar del Instituto Nacional Estadounidense de Estándares (ANSI) en 1986 y de la Organización Internacional de Normalización (ISO) en 1987 [29]. Ha adquirido tanta popularidad por ser un lenguaje no procedimental, es decir, especifica qué quiere pero no cómo ni dónde conseguirlo y es relacionamente completo, pues permite la realización de cualquier consulta de datos [66].

Ahora que ya sabemos cómo funciona SQL, podemos hablar del sistema de gestión de base de datos (SGBD) que hemos utilizado. Para el desarrollo de esta base de datos hemos decidido usar en primera instancia MySQL. Se trata de un SGBD relacional de código abierto. Se entiende por código abierto (*open source*) a un modelo de desarrollo de *software*, basado en la colaboración abierta que se enfoca más en los beneficios prácticos como es el acceso al código fuente [67], lo que lo hace accesible y práctico. Por su extendido uso, es una opción fiable y estandarizada. Por otra parte, SQLite [68] es de igual manera un motor de base de datos con SQL integrado, pero su principal ventaja es que no necesita un servidor para funcionar. Por este motivo, resulta bastante atractivo para trabajar con aplicaciones como es nuestro caso. Otra de las ventajas es el espacio que ocupa en el disco duro (< 500kb) y lo que hace factible que la base de datos sea gestionable dentro del propio dispositivo en lugar de en nuestro servidor.

Pero, ¿qué lo hace destacar frente a otras opciones como SQLite? La base de datos ha sido diseñada en su mayoría con campos *booleanos*. Sin embargo, SQLite no dispone de esta opción. Entre sus clases de almacenamiento la única opción posible sería la conversión de *booleanos* a *integer*, ralentizando la base de datos. Asimismo se trata de un sistema orientado a funcionar con poco tráfico, por lo que si introducimos una gran cantidad de datos, quizás su sistema no fuera todo lo eficiente que necesitaríamos. Por lo que consideramos mejor continuar con el uso de MySQL, así como de nuestro servidor.

Representación de los de datos

Para abordar este tema vamos a hablar sobre Ckan y Dkan. Se trata de herramientas para la gestión de datos dentro de la *web*. Su éxito se debe a que ambas son plataformas de código abierto, libre y gratuito. La principal diferencia es que DKAN es una versión de CKAN desarrollada en Drupal [69].

Asthma prevalence by location

1 - 20 of 106 rows Rows per page: 20 Display Density: [Manage Columns](#)

record_number	state	abbreviation	age_range	current_asthma...	standar
Search 20 record:	Search 20 record:	Search 20 record:	Search 20 record:	Search 20 record:	Search 2
1	Alabama	AL	Adults	10.9%	0.53
2	Alaska	AK	Adults	8.4%	0.88
3	Arizona	AZ	Adults	9.8%	0.31
4	Arkansas	AR	Adults	9.6%	0.73
5	California	CA	Adults	7.9%	0.38
6	Colorado	CO	Adults	9.3%	0.37
7	Connecticut	CT	Adults	10.7%	0.43
8	Delaware	DE	Adults	10.6%	0.71
9	District of Columbia	DC	Adults	9.4%	0.63
10	Florida	FL	Adults	7.5%	0.39
11	Georgia	GA	Adults	8.4%	0.47
12	Guam	GU	Adults	7.4%	0.96
13	Hawaii	HI	Adults	9.8%	0.49
14	Idaho	ID	Adults	9.0%	0.55
15	Illinois	IL	Adults	8.2%	0.46
16	Indiana	IN	Adults	10.0%	0.36
17	Iowa	IA	Adults	9.3%	0.41
18	Kansas	KS	Adults	9.1%	0.25
19	Kentucky	KY	Adults	10.7%	0.56
20	Louisiana	LA	Adults	9.0%	0.57

< Page 1 of 6 >

Imagen 46

Crédito de imagen: Instantánea de Dkan, 2021, Autoría de National Health Council: <https://getdkan.org/>

Pero, ¿qué es drupal? Drupal es un *software* de gestión de contenidos (CMS), configurable que permite múltiples servicios como publicación de encuestas, foros, artículos e imágenes [70]. De igual forma es un sistema dinámico que permite almacenar los datos en una base de datos y editarlos en un entorno web.

Imagen 47

Crédito de imagen: Instantánea de Drupal, 2021, Autoría de Drupal:
<https://www.drupal.org/>

Ahora que hemos visto que ambos son similares pero DKAN es una versión *a priori* superior, consideramos más interesante el uso de esta plataforma. Es necesario

destacar que otro factor a tener en cuenta es que CKAN tiene un consumo muy exigente de *hardware*, así como una gestión ineficiente de seguridad tanto en usuarios como en recursos. Consideramos importante el uso de una de estas plataformas para poder ofrecer al usuario una perspectiva más clara de los datos y una transparencia absoluta. Por esta misma razón, es una plataforma usada por muchos gobiernos [71], como el de Australia o el de Canadá, entre otros, así como instituciones sin ánimo de lucro.

8.- Codificación básica del *framework* en SQL, Python y Java

8.1.- Codificación SQL

Para crear la base de datos hemos utilizado el código SQL. Como apreciamos, necesitamos usar UTF-8, un formato de codificación de caracteres Unicode e *ISO* 10646 [72]. El motivo de su uso es para detectar si hay palabras registradas en la base de datos que contengan caracteres especiales como el **uso de la Ñ**, por ejemplo en la palabra **sueño**. Por otra parte la tabla TDispositivo (la última en el código) es la que presenta las claves foráneas y así ser el punto de unión del resto de tablas.


```
SET NAMES utf8;
CREATE SCHEMA Seguridad;
USE Seguridad;

CREATE TABLE TAccesoNoAutorizado(
nID INT,
lCamara BIT,
lMicrofono BIT,
lGPS BIT,
PRIMARY KEY (nID)
);

CREATE TABLE TSalud(
nID INT,
lFrecuenciaCardiaca BIT,
lSueño BIT,
lRegistroAutomaticoEjercicio BIT,
lEstres BIT,
lNivelOxigeno BIT,
lCicloMenstrual BIT,
lPasos BIT,
PRIMARY KEY (nID)
);

CREATE TABLE TTipoDispositivo(
nID INT,
lSmartwatch BIT,
lPCSobremesa BIT,
lPCPortatil BIT,
lTablet BIT,
lSmartphone BIT,
PRIMARY KEY (nID)
);
```

Imagen 48

Crédito de imagen: Codificación SQL por
Diana Díaz, 2021, Mangosta

(https://mangosta.org/iii_codificacion-sql/) (.) Acceso al código para docentes
y alumnos:
<https://mangosta.org/codificacion-sql001/>

```
CREATE TABLE TDetecccion(  
  nID INT,  
  lPositivo BIT,  
  PRIMARY KEY (nID)  
);  
  
CREATE TABLE TDispositivo(  
  nID INT,  
  cMarca VARCHAR(30),  
  cModelo VARCHAR(60),  
  PRIMARY KEY (nID),  
  FOREIGN KEY (nID) REFERENCES TAcceso (nID),  
  FOREIGN KEY (nID) REFERENCES TSalud (nID),  
  FOREIGN KEY (nID) REFERENCES TTipoDispositivo (nID),  
  FOREIGN KEY (nID) REFERENCES TDetecccion (nID)  
);
```

Imagen 49

Crédito de imagen: Codificación SQL por Diana Díaz, 2021, Mangosta

(https://mangosta.org/iv_codificacion-sql/) (.) Acceso al código para docentes y alumnos:
<https://mangosta.org/codificacion-sql002/>

8.2.- Codificación Java

En este apartado se añade el código Java [73] con el que hemos accedido a la base de datos llamada “seguridad”, así como el resultado de hacer una consulta en lenguaje SQL. En primer lugar, importamos la librería java.sql para usar la base de datos desde Eclipse:

```
import java.sql.*;
```

Imagen 50

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta

(https://mangosta.org/viii_codificacion-java/) (.) Acceso al código para docentes y estudiantes:
<https://mangosta.org/codificacion-java/>

Más adelante, es necesario asegurarnos de que el *driver* JDBC esté debidamente conectado. A continuación, establecemos la conexión con la base de datos por medio de estas instrucciones:

```
try {
    Connection miConexion = DriverManager.getConnection("jdbc:mysql://localhost:3306/seguridad", "usuario", "contraseña");
```

Imagen 51

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/ix_codificacion-java/) (.) Acceso al código para docentes y estudiantes: <https://mangosta.org/codificacion-java001/>

La clase DriverManager tiene un método **String** que contiene la ruta de la base de datos. Es importante recordar que para ello hemos utilizado el conector JDBC para MySQL que instalamos en el apartado referente a la diagramación de Java. Adicionalmente, lo hemos probado en el servidor de base de datos de nuestro ordenador (*localhost*) y en la nube, cambiando la ruta de la base de datos.

```
try {
    Connection miConexion = DriverManager.getConnection("jdbc:mysql://ruta del servidor.net/seguridad", "usuario", "contraseña");
```

Imagen 52

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/x_codificacion-java/) (.) Acceso al código para docentes y estudiantes: <https://mangosta.org/codificacion-java-002/>

Por defecto, el servidor MySQL abre el puerto 3306. Es necesario conectarse a este puerto si queremos consultar la base de datos que hemos creado. Por consiguiente, creamos el objeto "Statement". Para ello, seguimos el paso mostrado en la siguiente figura:

```
Statement miStatement = miConexion.createStatement();
```

Imagen 53

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/xi_codificacion-java/) (.) Acceso al código para docentes y estudiantes: <https://mangosta.org/codificacion-java-003/>

Asimismo, el objeto "ResultSet" nos permite obtener los resultados de la consulta. En nuestro caso, nos mostrará todos los registros de la tabla demandada, es decir,

TDispositivo:

```
ResultSet miResultSet = miStatement.executeQuery("SELECT * FROM TDispositivo");
```

Imagen 54

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/xii_codificacion-java/) (.) Acceso al código para docentes y estudiantes: <https://mangosta.org/codificacion-java-004/>

Ahora, es necesario utilizar el bucle “while” y la variable “miResultSet” para obtener todos los resultados de nuestra consulta.

```
while (miResultSet.next()) {
    System.out.println(miResultSet.getString("nID") + "\t" + miResultSet.getString("cMarca") + "\t" + miResultSet.getString("cModelo"));
}
} catch (Exception e) {
    System.out.println("No funciona");
    e.printStackTrace();
}
```

Imagen 55

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/xiii_codificacion-java/) (.) Acceso al código para docentes y estudiantes: <https://mangosta.org/codificacion-java-005/>

En caso de no establecer la conexión de la base de datos en eclipse, al ejecutar se mostrará por pantalla: “No funciona”. Realizados todos los pasos, finalmente obtenemos la tabla con los registros que hemos consultado:

```
<terminated> ConexionLocal (1) [Java Application] C:\Users\river\
1 Apple Watch 6
2 Samsung Galaxy Watch 3
3 Fitbit Sense
```

Imagen 56

Crédito de imagen: Codificación Java por Kimberly Riveros, 2021, Mangosta (<https://mangosta.org/codificacion-java-2/>)

8.3.- Codificación Python

En este apartado se añade el código Python que hemos utilizado para acceder a la base de datos llamada “seguridad”, así como el resultado de hacer una consulta en lenguaje

SQL. En primer lugar, debemos instalar el conector `mysql-connector-python`, un *driver* para comunicarse con servidores MySQL [74].

```
[ ] pip install mysql-connector-python
```

Imagen 57

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta

(https://mangosta.org/i_codificacion-python/) (.)

Acceso al código para docentes y alumnos:

(https://mangosta.org/ii_codificacion-python/)

Seguidamente, realizamos el mismo proceso con PyMySQL, otro paquete para la interacción con bases de datos MySQL. Y, una vez instalado, lo importamos.

```
[ ] pip install pymysql
```

Imagen 58

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta

(https://mangosta.org/iii_codificacion-python/) (.)

Acceso al código para docentes y alumnos:

(https://mangosta.org/iv_codificacion-python/)

```
[ ] import mysql.connector
```

Imagen 59

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta

(https://mangosta.org/x_codificacion-python-2/) (.) Acceso al código para

docentes y alumnos:

(https://mangosta.org/iv_codificacion-python/)

Teniendo ya los paquetes necesarios, nos conectamos a la base de datos “seguridad”. Para ello, creamos una variable llamada “miConexión” y mediante PyMySQL, almacenamos la ruta de la base de datos, el usuario, la contraseña y el nombre de la base de datos. En definitiva, una serie de información, necesaria para establecer la conexión con la base de datos.

```
[ ] import pymysql
miConexion = pymysql.connect( host='ruta', user= 'usuario', passwd='contraseña', db='seguridad' )
```

Imagen 60

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta (https://mangosta.org/xi_codificacion-python/) (.)

Acceso al código para docentes y alumnos: (https://mangosta.org/v_codificacion-python/)

El siguiente paso es convertir esta variable en un cursor. Para eso, ejecutamos el cursor con la consulta que deseamos realizar.

```
cur = miConexion.cursor()
```

Imagen 61

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta

(https://mangosta.org/xii_codificacion-python/) (.) Acceso al código para docentes y alumnos:

(https://mangosta.org/vi_codificacion-python/)

```
cur.execute( "SELECT* FROM TAccesoNoAutorizado" )
```

Imagen 62

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta (https://mangosta.org/xiii_codificacion-python/) (.) Acceso al código para docentes y alumnos:

(https://mangosta.org/vii_codigo-python/)

Ya, para terminar, imprimimos por pantalla el resultado de la consulta. Y, una vez acabado el trabajo en la base de datos, cerramos la conexión.

```
for first_name in cur.fetchall() :
    print (first_name)
```

Imagen 63

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta (https://mangosta.org/xiv_codificacion-python/)

(.) Acceso al código para docentes y alumnos: (https://mangosta.org/viii_codificacion-python/)

```
(1, b'\x01', b'\x00', b'\x00')
(2, b'\x00', b'\x00', b'\x00')
(3, b'\x01', b'\x01', b'\x01')
```

Imagen 64

Crédito de imagen: Codificación Python por Kimberly Riveros, 2021, Mangosta (https://mangosta.org/x_codificacion-python/)

```
miConexion.close()
```

Imagen 65

Crédito de imagen: Codificación Python por Kimberly Riveros y Diana Díaz, 2021, Mangosta (https://mangosta.org/xv_codificacion-python/) (.) Acceso al código para docentes y alumnos: (https://mangosta.org/ix_codificacion-python/)

9.- Interfaces

Usaremos éstas para el *front-end* [75] de la aplicación (UI / UX) [76] o la parte interactiva donde el usuario interactúa con su aplicación y donde todas las entradas se dan a través de esta capa. La usabilidad y el diseño del entorno de las *interfaces* en los distintos modos de funcionamiento M1, M2 y M3 (que se definen más adelante, en el pto 10.2) deben poder permitir un control parental y de gestión de dispositivos suficiente. El diseño de la *interface* es un elemento clave para un uso práctico y solvente del *framework*.

9.1.- Kivy

Kivy [77] técnicamente es sólo otra biblioteca de Python. Pero no es una cualquiera. Es un marco para crear *interfaces* de usuario que funciona en múltiples plataformas como Android, iOS, Windows, Linux y macOS. Puede pensarse también como una alternativa a React Native [78], Flutter [79], Solar2D [80], etc.

Kivy fundamentalmente es para los amantes de Python y el *Machine Learning*. Con la biblioteca Kivy la opción más acertada es usar Conda [81]. Instalar Kivy con Conda es fácil: « `conda install kivy -c conda-forge` ». Posteriormente podremos compilar una

aplicación para iOS o Android a partir del código fuente[82] y ejecutarla en un simulador para poder visualizar su aspecto y resultado final.

9.2.- Kotlin

Kotlin[83] es un lenguaje de programación de tipado[84] estático que corre sobre la máquina virtual de Java[85] (JVM) y que también puede ser compilado[86] a código fuente de JavaScript. Aunque no tiene una sintaxis compatible con Java, Kotlin está diseñado para interoperar con código Java y es conocido por ser utilizado en la creación de aplicaciones de Android.

9.3.- Elementos sociales y legales en el diseño de las *interfaces*

Lo que sigue son indicaciones específicas a la hora de codificar y desarrollar los entornos gráficos y de comunicación con los usuarios de este *framework*. Son premisas que en el inicio deben estar siempre presentes para los desarrolladores para evitar lesivas modificaciones técnicas futuras en los entornos finales implementados. El marco no debe olvidar nunca su mayor objetivo: proteger en el mayor de los ámbitos y favorecer la pluralidad social.

1. Se deberá atender en su mayor extensión posible a criterios conceptuales y de uso en la igualdad de género[87] y accesibilidad[88] así como poner énfasis en evitar todo tipo de sesgo[89].
2. El lenguaje particularmente dirigido a menores de edad deberá ser cuidadosamente estudiado para un uso adecuado.
3. El cumplimiento en la normativa de protección de datos[90] y otros que le resulten de aplicación deberá ser un imperativo presencial.
4. En la medida de lo posible se debe proporcionar suficiente información educacional sobre ciberseguridad e inteligencia artificial.

III.- Estadios y modos aplicables al *framework* y directrices de requerimientos mínimos

La caracterización en estadios y modos de este marco se propone para definir la infraestructura de sistemas y asegurar el nivel de procesamiento adecuado y su velocidad para la ejecución en los distintos dispositivos *smart* del mercado, así como para la descripción generalizada de todos los instrumentos y herramientas asociadas.

10.1.- Estadios

10.1.1.- **Estadio 1 (E1):** modo laboratorio y aprendizaje de la IA

Es en este estadio donde caben configuraciones y ajustes básicas y avanzadas. Las distintas configuraciones de marcos y modelos nos ayudarán a optimizar y mejorar el resultado para que posteriormente éste y su experiencia asociada sea aplicado al estadio (E2), o modo de ejecución de la IA.

Para entrenar a la inteligencia artificial podemos partir desde cero o realizar una transferencia de conocimientos con un modelo entrenado previamente. También podemos plantear **modelos** específicos para las imágenes, para lo que ResNet nos ofrece en general resultados muy aceptables en la proporción de aciertos en una clasificación dada, *accuracy*[\[91\]](#), o tasa de aciertos en la predicción. Asimismo podemos usar diferentes **marcos**, donde los más conocidos son Pytorh, Tensorflow, Fastai y Keras... Fastai es nuestro marco por defecto.

10.1.2.- **Estadio 2 (E2):** modo de ejecución de la IA

Es el fruto del estadio E1. Este estado se define para un modelo entrenado y guardado para su ejecución. No debe admitir configuraciones avanzadas algunas.

10.2.- **Modos**

Los modos son los distintos sistemas de trabajo o maneras de operar que pueden ser aplicados a este *framework* de cara a los diferentes dispositivos inteligentes a los que les resulta de aplicación.

10.2.1.- **Modo 1(M1):** modo autónomo o *standalone*

Le resulta de aplicación el estadio E2. Su objeto de definición es que éste pueda ser descargado y ejecutable como cualquier otra app[\[92\]](#) del mercado. En él se ejecutarán mayoritariamente *smartphone*, *tablets* y PC's. Se recomienda compatibilidad total del modo M1 para los sistemas operativos Android 11 o superior; iOS 14.5.1 o superior; sistema operativo Windows 10 o superior o MAX OS X 10.15 o superior. Las características *hardware* mínimas serán las recomendadas por los fabricantes de éstos.

10.2.2.- **Modo 2 (M2):** modo enlazado o esclavo de otro dispositivo

Este modo se concibe para el emparejamiento o vinculación de como mínimo dos dispositivos con tecnologías compatibles del tipo Bluetooth, WiFi u otro mediante una app o un *software* satélite complementario para este fin. Aquí es donde mayoritariamente se emplazarán la mayoría de los *smartwatches* actuales,

referenciados en el pto. 11.1x. Los requerimientos mínimos aconsejados de *hardware* para el despliegue del *framework* en modo M2 son: 4GB de memoria interna de almacenamiento, 512MB de RAM y Bluetooth 4.0.

10.2.3- **Modo 3 (M3):** modo servidor o laboratorio de entrenamiento y aprendizaje

Concebido para operar principalmente en el estadio E1. Se concibe en general el modo M3 para el diseño y gestión de la IA, así como el modo que embebe y gestiona el modo M1 y el modo M2 de todos los dispositivos relacionados en torno a él. Éste es propio de una infraestructura Cliente-Servidor[93]. En base a la naturaleza de su ejecución establecemos dos clases:

A) Modo 3A: computación en la nube

Análogo al sistema que nosotros hemos planteado en laboratorio... También puede ser implementado mediante una infraestructura *ad hoc*[94], propia de un proveedor de servicios del *framework*, y asumir los gastos inherentes a su uso y despliegue. AWS de Amazon, Google Cloud[95], Azure[96]u otros son posibles candidatos para este fin. Para la gestión completa del entorno adicionalmente se ha de implementar un servidor XAMPP[97] o similar en un *hosting* dedicado[98] y adecuar los servicios para *Servlets* de Java[99] para la completa operatividad. Para el modo 3B los requerimientos *hardware* han de considerar como mínimo los usados en nuestra configuración gratuita en laboratorio para garantizar un resultado satisfactorio de ejecución de la IA, así como los indicados por los fabricantes del *software* asociado a éste.

B) Modo 3B: computación en local

Para poder trabajar en este modo podemos efectuar una instalación local[100] con fastai y Anaconda o Miniconda bajo sistema operativo Linux o Windows. Asimismo para la gestión propia de la BD se ha de instalar Mysql o similar. Y Java JDK[101] para el desarrollo y ejecución de la codificación en Java.

Para este modo 3B el *hardware* ha de ser más específico y exigente, aunque no debería exceder los 2.500 ó 3.000 euros a precio actual de mercado para una configuración óptima de un entorno de producción. Debe ser capaz de ejecutar un *software* similar a Anaconda Enterprise 4 ó 5[102]. (Los requerimientos son CPU: 2 x 64-bit 2.8 GHz 8.00 GT/s CPUs; RAM: 32 GB (o 16 GB de 1600 MHz DDR3 RAM); 300 GB de almacenamiento en disco y acceso a Internet). Se aconseja memoria RAM de 64 GB y un disco duro de característica similares o superiores a 970 EVO Plus (1TB) de Samsung para todo el conjunto que supone el Modo 3B.

La GPU[103] es el componente que realizará la mayoría de cálculos lentos y pesados en el proceso de entrenar a una red convolucional[104]. Mayoritariamente estos cálculos son multiplicaciones de matrices, convoluciones y funciones de activación[105]. El *hardware* proporcionado por NVIDIA, por ejemplo, el modelo 2080 Ti[106] con 13.45 TFLOPs y 11 GB de memoria GDDR6 es una base sólida. Para la CPU la marca Intel es una opción ineludible. La familia i7, el i7-10700k, puede ser una elección garante.

IV.- Campo de aplicación

Los entornos donde el marco puede ser implementado generarán especificaciones propias para cada entorno. Principalmente caracterizadas por el *hardware*, en la capa del modelo OSI[107] que se opere, su sistema operativo[108] y su conectividad a una red determinada e Internet, y la conexión entre dispositivos propiamente dicha. Se debe prestar una atención importante a la velocidad de ejecución del *framework*, por lo que los procesos aplicables deben ser los más imprescindibles y livianos que resulte posible y la velocidad aquélla más óptima que pueda alcanzarse. Todo ello en el mejor balanceo de rendimiento de consumos en general y tomando en cuenta las capacidades y la morfología donde pretenda emplearse. Signifiquemos a continuación algunos de estos entornos. Donde efectuaremos descripciones y propuestas de *software* y de *hardware*, denotando que ni éstas son únicas ni necesariamente imprescindibles en la implementación final.

11.1- Smartwatches

Un *smartwatch* puede suponer un estilo de vida, incluso un concepto de libertad; son como una carcasa de reloj que porta y camufla tecnología digital y se incorpora al sujeto como una prenda más diaria, un *wearable* (tecnología vestible[109]); “La información que se mueve contigo”, fue el lema de Android Wear[110] en 2014. Y, aparejado de la mano, o la muñeca en este caso, puede ir una grave pérdida de la privacidad.

Un reloj inteligente tiene matices que marcarán la diferencia, por ejemplo, su diseño, su tamaño, su *interface* de usuario[111], si está o no está asociado a un teléfono inteligente mediante *Bluetooth*, si el reloj dispone de tarjeta SIM[112] o no la tiene, si puede ser gestionado como un dispositivo más en red (WiFi)[113] o no, si realmente es un reloj inteligente o una pulsera de actividad (*fitness*[114]); o todo junto, o parte de éstos.

Una amalgama de *hardware* y de prestaciones y precios de mercado están asociados a ellos y que determinarán elementos como la pantalla (de onda acústica[115], resistiva o capacitiva según su *touch*[116], y también su panel: LCD[117], IPS[118], AMOLED[119] o OLED[120]), los sensores que incorpore, la memoria RAM, memoria interna de almacenamiento, microprocesador (de fabricación propia, Qualcomm, ARM, MediaTek...), duración de la batería, si incorporan audio y cámara fotográfica, tecnología de comunicación y aproximación como NFC[121], Wifi o RFID,[122] etc.

Podemos encontrarlos desde los precios más ínfimos ($\approx$, $<$) a 20 € en el mercado chino en AliExpress (Tipmant Smartwatch) o en Amazon, hasta precios que pueden rondar los 2.000€ (**TAG HEUER CONNECTED GOLF EDITION**)[123], incluso más caros.

Y si ponemos la mirada hacia el futuro a la hora de inclinarnos por ciertos tipos de dispositivos, la clave diferenciadora pudiera estar en si reemplazan al *smartphone*, en si pueden hacer y recibir llamadas (mediante bandas LTE[124], por ejemplo, vinculada a la democratización del 5G[125]) y al Internet de la Cosas (IoT)[126].

Habitualmente llevan una versión reducida como sistema operativo de Android, iOS o Linux (Wear OS[127], WatchOS[128] y Tizen OS[129]) son los más habituales. Esa tecnología digital es la plataforma donde puede ejecutarse nuestro *framework*, en un entorno de S.O. reducido.

11.1.1.- Hardware de algunos de los relojes inteligentes

Figura 2

Dispositivo	Procesador	Memoria	Comunicaciones	Pantalla	Cámara	Batería
Samsung Gear 2	Dual-Core 1 GHz Exynos3250	4GB memoria interna, 512MB RAM	Bluetooth 4.0 LE	1.63" Super AMOLED con resolución de 320x320	2.0 Megapíxeles	300 mAh
Samsung Gear S	Qualcomm Snapdragon 400 a 1 GHz	4GB memoria interna, 512MB RAM	Bluetooth 4.0, Wi-Fi, 3G	AMOLED de 2" y 360x480 píxeles		300 mAh

Samsung Gear S2. S.O: Tizen	Exynos 3250	4GB memoria interna, 512MB RAM	Bluetooth, BT4.1, Wi-Fi, NFC	sAMOLED 360 X 360 (302 ppi)		250 mAh
Moto 360 SO: Wear OS by Google™	Single-Core 1 GHz TI OMAP 3 / Qualcomm® Snapdragon™ Wear 3100	4GB memoria interna, 512MB RAM / 1GB RAM + 8GB Memoria Interna	Bluetooth 4.0 LE / Bluetooth 4.2, Wi-Fi b/g/n, NFC, GPS / GLONASS / Beidou / Galileo	LCD de 1.56" con resolución de 320x290, 205 ppi, LCD con retro-iluminación / 1.2" Circular AMOLED(390x390)		320 mAh / 355mAh
LG G Watch SO: Android 4.3 en adelante	Qualcomm Snapdragon 400 a 1.2 GHz	4GB memoria interna, 512MB RAM	Bluetooth 4.0	1.65" IPS LCD		400 mAh
LG G Watch R	Qualcomm Snapdragon 400 a 1.2 GHz	4GB memoria interna, 512MB RAM	Bluetooth 4.0	P-OLED de 1.3" y 320x320 píxeles		410 mAh
Sony SmartWatch 3	Quad-core ARM Cortex A7 a 1,2GHz	4GB memoria interna, 512MB RAM	Bluetooth 4.0, NFC	1.6" y 320x320 píxeles		420 mAh
Asus ZenWatch	Qualcomm Snapdragon 400 a 1.2 GHz	4GB memoria interna, 512MB RAM	Bluetooth 4.0	AMOLED touchscreen capacitiva de 1.63" y 320x320 píxeles		369 mAh

Apple Watch	Apple S1	8GB memoria interna, 512MB RAM	WiFi y GPS de iPhone	Retina con Force Touch		Hasta 18 horas de autonomía
Apple Watch Series 6 SO: iOS 14 o posterior	S6 64-bit dual-core	32 GB memoria interna, 1GB	LTE y UMTS, Wi-Fi, Bluetooth 5.0	Retina OLED LTPO		Hasta 18 horas

Crédito figura 2: Versión actualizada mejorada de «Características *hardware*» [130]. Reloj inteligente. (2021, 15 de febrero). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 13:44, julio 18, 2021 desde https://es.wikipedia.org/w/index.php?title=Reloj_inteligente&oldid=133248752.»

El *framework* para cualquier *smartwatch* debe asumir que el estadio de funcionamiento recomendado es el **estadio E2** (ejecución de la IA) y el **modo M2** (entrelazado). De forma dinámica, en función de la evolución de la tecnología asociada a un *smartwatch*, el modo M1 (autónomo) pudiera ser también una opción viable a medio plazo.

11.2.- Otros dispositivos afines

11.2.1- Smartphones, tablets y PC's

En la actualidad estos dispositivos han evolucionado de forma contundente en la gama media del sector, son garantes y suficientes en general, con altas prestaciones tecnológicas y una baja economía de coste asociadas. Aunque las *tablets* parece que no terminan de conseguir un nicho final suficiente de mercado en función de algunos medidores asociados ([Imagen 66](#)). Son durísimos rivales los *smartphones* y los PC's para ellas; porque, en comparación, en la relación tecnología y precio éstas salen perdiendo probablemente. Las *tablets* actualmente, ¿parecen más un 'capricho útil' que una necesidad real del usuario? [131]... No obstante fabricantes como Microsoft tienden a unificar sistemas operativos como Windows 10 y 11 [132] para que su integración esté asegurada a estos tres niveles o familias de dispositivos. Que todo sea escalable y transversal. Un todo en uno.

El *framework* para cualquier elemento del tipo *smartphone*, *tablet* o PC debe asumir que el estadio de funcionamiento recomendado para éstos es el **estadio E2** (ejecución de la IA) y el **modo M1** (autónomo).

Imagen 66

Crédito imagen: «Tendencias de Internet 2021. Estadísticas y hechos por países».

VPNmentor. URL:

<https://es.vpnmentor.com/blog/tendencias-de-internet-estadisticas-y-datos-en-los-estados-unidos-y-el-mundo/>, Mangosta

(<https://mangosta.org/tendencias-2/>)

11.2.2.- Domótica

Hablar de domótica es hablar sobre todo de sus tipos de conexiones para una aplicación total y real de ésta, conjuntamente con el 5G[125] y/o el IoT[126] y una estructura más que probable en «Bus[133] inteligente».

Imagen 67

Todos los sensores y actuadores están conectados a través de un cable bus. El sistema completo es denominado “sistema bus”.

Crédito imagen: «KNX Conocimientos básicos del estándar KNX». V9-14. KNX.org.

Accesado el 22/07/2021. Url:

https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-Basics/KNX-Basics_es.pdf, Mangosta
(<https://mangosta.org/conocimientos-basicos-2/>)

Actualmente todo un espacio de tecnología de comunicaciones inalámbricas coexiste: Bluetooth, WiFi, RFID, NFC y otros. Aunque las morfologías de conexión para la domótica aún están decidiéndose por parte de la industria y de los estándares que se asocian, pugnan entre ellos definiendo diferentes intereses.

Una línea de trabajo, por ejemplo, es ZigBee que persigue: «Comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías»[134]; o Z-Wave[135]: «Una red en malla que utiliza ondas de radio de baja energía para comunicarse de un aparato a otro, permitiendo el control inalámbrico de electrodomésticos y otros dispositivos, como control de iluminación, sistemas de seguridad, termostatos, ventanas, cerraduras, piscinas y garaje abrepuestas»; o Sigfox[136]: «Un operador de red global y creador de la red 0G fundado en 2009 que implementa redes inalámbricas para conectar dispositivos de bajo consumo como pueden ser medidores eléctricos, centrales de alarmas o relojes inteligentes, que necesitan estar continuamente encendidos y enviando pequeñas cantidades de datos».

Aun dentro de estos tiempos tecnológicos tan convulsos, a nivel de estrategias y vaivenes, (como un estándar y un valor seguro del mercado de las comunicaciones para los próximos 5 años), Bluetooth y WiFi parecen las apuestas más seguras y estables durante este periodo: a nivel tecnológico y comercial, claro. Y la mayoría de los dispositivos actuales uno (Bluetooth), u otro (WiFi), o ambos, los implementan: disponen de esta tecnología tan popular entre los usuarios y fabricantes.

El *framework* para cualquier elemento doméstico (diferente a *smartwatch*, *smartphone*, *tablet*, y PC) debe asumir que el estadio de funcionamiento recomendado para elementos domóticos es el **estadio E2** (ejecución de la IA), y disponibles ambos modos M1 (autónomo) & M2 (asociado).

11.2.2.1 Dispositivos domóticos con Android & iOS

Las aplicaciones *smarthome* son una evidencia. Son innegables. Y mucho han contribuido sistemas operativos como Android e iOS. Y cualquier dispositivo doméstico o *wearable* de los que hemos descrito aquí es susceptible de implementar estos sistemas operativos. Y también pueden y han de estar protegido por la implementación de nuestro *framework*. Que de acuerdo a lo enunciado previamente puede ser ‘montado’ sobre estas tecnologías.

Por citar algunas referencias: Amazon Alexa, Casa, Smartthings, Google Home, LIFX, Houseinhand KNX, TaHoma by Somfy, Home Connect App, Smart Life, Philips Hue, etc.

11.2.2.2- Elemento multiplataforma JRE (*Java Runtime Environment*) en dispositivos domóticos

La máquina virtual Java[85], o JVM por sus siglas en inglés, es un entorno multiplataforma ideado para ser portable con independencia de los sistemas y el entorno que lo envuelva.

«La máquina virtual de Java puede estar implementada en *software*, *hardware*, una herramienta de desarrollo o un navegador *web*; lee y ejecuta código precompilado bytecode que es independiente de la plataforma. La JVM provee definiciones para un conjunto de instrucciones, un conjunto de registros, un formato para archivos de clases, la pila, un heap con recolector de basura y un área de memoria. Cualquier implementación de la JVM que sea aprobada por SUN debe ser capaz de ejecutar cualquier clase que cumpla con la especificación.»[85] .

Es este soporte independiente, por tanto, un entorno domótico objetivo en la implementación de nuestro DIID (**D**etector de **I**ntrusiones **I**nteligente **D**omótico).

11.2.2.3- Descripción del estándar de domótica KNX como marco regulador para dispositivos domóticos

El estándar de protocolo KNX[4] se cimenta básicamente en el concepto de bus[133] inteligente, y su mejor descripción es entenderlo como una «neurona central» que comunica todos los elementos de una vivienda. Este tipo de conexión de dispositivos es muy propicio para una mejor y mayor aplicación de la IA sin duda. La estandarización (ISO/IEC 14543)[137] rige esta tipología. Ambos, capacidad de aplicar la IA y la estandarización, constituyen una plataforma ideal de aplicación para nuestro Detector de Intrusiones Inteligente Domótico(DIID por su acrónimo), para la implementación y aplicación de *framework* definido en este trabajo de investigación.

Estos elementos a proteger por el DIID o (DI)² van desde: Interruptores de luz • Teclados de regulación de luz • Detectores de movimiento • Detectores de presencia • Contactos de ventanas y puertas • Timbre de la puerta de entrada • Contadores de consumo de agua, gas, electricidad y calor • Sensores de sobretensión • Sensores de temperatura ambiente, interior y exterior • Sensores de temperatura en circuitos de agua caliente y calefacción • Módulos para prefijar la temperatura de consigna en las habitaciones • Sensores de luminosidad interior y exterior • Sensores de viento para control de persianas y toldos • Indicadores de estado o fallo en aparatos de línea blanca • Sensores de fuga • Sensores de nivel • Receptor de radiofrecuencia en cierre de puertas • Receptor de infrarrojos de mandos a distancia • Lectores de huellas dactilares o tarjetas electrónicas para control de acceso[138]... Y un largo etcétera de actuadores y módulos.

Y todos ellos han de ser protegidos, y el *framework* implementado en un IDS inteligente es de aplicación necesaria aquí.

Imagen 68

KNXnet/IP en el modelo de referencia OSI. Crédito imagen:

«KNX Conocimientos básicos del estándar KNX». V9-14.

KNX.org. Accesado el 22/07/2021. Url:

https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-Basics/KNX-Basics_es.pdf, Mangosta
(<https://mangosta.org/estandar-knx/>)

Los ensayos realizados en laboratorio aportan el conocimiento suficiente para inferir que el marco puede ser embebido en un microcontrolador[139] de la familia PIC 18F[50],[140] PIC18F4550 u otro similar con la sistemática de KNX[141], apoyado por el diseño específico de circuitería electrónica que implemente la lógica combinacional necesaria.

Imagen 69

Montaje En Protoboard Del Sistema Pinguino. Julio F. De la Cruz G.
 Accedido el 22/07/200121. Url: <http://3.bp.blogspot.com/-JtZH7qb-jw/UhLyWQdHLjI/AAAAAAAAAJQ/cjoKPWOlxsw/s1600/image7923.png>, Mangosta
 (<https://mangosta.org/montaje-en-protoboard/>)

El diseño de este dispositivo domótico DIID se concibe como una *appliance*^[142] que reúne como mínimo los requisitos *hardware* de un *smartwatch* para embeber el estadio y modo de funcionamiento inherente a éste anteriormente definidos, mediante un *software* encapsulado o *firmware*^[143].

Su filosofía de funcionamiento debe basarse en la técnica *plug&play*^[144] de modo que pudiera ser un elemento de protección más, alojado entre los dispositivos propios de un cuadro eléctrico de cualquier vivienda que alberga normalmente un ICP^[145], un diferencial^[146] e interruptores magnetotérmicos^[147].

Concebido por tanto el dispositivo IDS domótico (DIID) como un elemento más adicional de seguridad y protección en un hogar o un entorno laboral que implementara el estándar KNX^[4].

V.- Conclusión

Aquí se ha demostrado que el uso de la IA, en buena parte de sus dimensiones, puede ser llevada a término con rigor académico, definible en el campo de las matemáticas y usable en el del laboratorio; y que puede ser pensada, razonada y aplicada.

El *framework* está motivado por la continua aparición y mejora de dispositivos móviles en forma de *smartwatches*, *smartphones* y otros similares que ha propiciado simultáneamente un creciente y desleal interés en poner bajo la lupa y el control de los aplicativos a sus usuarios. Éstos son presa de una abusiva injerencia en sus vidas para telemedir y telegestionar sus hábitos en Internet, y también de sustraerles de forma ofuscada datos personales como, por ejemplo, cuando realizan deportes que precisan medir constantes vitales tales como pulsaciones del corazón, presión arterial, nivel de oxígeno, etc. La industria que los produce y los gestiona arteralmente está canalizando cada vez más esfuerzos económicos, así como ciencias como la inteligencia artificial (IA), gestión de flujos de datos y estrategias para que el resultado final apunte hacia la manipulación, venta y secuestro de las vidas de los ciudadanos que las usan. Y todo ello con un probable valedor y último propietario: la Inteligencia Artificial (IA), llena de oscuridad y mecanismos internos oscuros o '*Black Boxes*'... Y de sociedades ingenuas, como la nuestra, que lo están impulsando desconociendo probablemente la causa y el efecto final para la humanidad.

12.1.- Corolario I

Podemos plantear este marco como un espacio generador de marcos, análogo a un espacio vectorial[31] que genera subespacios vectoriales[148]; y al $\vec{S}V_{intrusión} = [\vec{P}x_{(1...n)}, \vec{I}mg_{y_i}]_0^{3(n)}$ como su sistema generador[149] homólogo, en pro de la transparencia y la explicabilidad de tecnologías como la IA. Donde cada variable analógica, ambigua, difícilmente definible, amplia de espectro y pensamiento, y abrumadora en dimensiones, pueda ser escaneada en sus parámetros más fundamentales como postula, demuestra y defiende este *framework* en nuestra opinión. Aquellos parámetros iniciales que incluso un niño se atrevería a dibujar de algo desconocido para él, un monigote como si fuese un garabato, como un símbolo[150] o como una primera idea de la niñez. Y a medida que el niño aprenda y aumente 'sus parámetros' de definición y valores personales, entonces la definición será más profunda, concisa y ajustada a su propia realidad. De igual manera que se presenta en este marco, en sus cimientos. Para que la inteligencia artificial sea algo que pueda ser explicable, transparente y dentro del control humano. Porque una

ingeniería inversa siempre podría devolverla al mundo del *homo sapiens*[\[151\]](#), nuestro mundo real con todas sus imperfecciones y sus realidades.

12.2.- Corolario II

Por inferencia del corolario I, otra de las mayores preocupaciones entorno al la inteligencia artificial, el sesgo algorítmico[\[152\]](#): de género, cultural, de raza, de discapacidades individuales, de pensamientos e ideologías... puede ser mitigado. Tratando de alcanzar una sociedad más justa en valores. Al lado de sus logros tecnológicos humanos.

12.3.- Corolario III

Por inferencia de los corolarios I y II, la Inteligencia Artificial (IA) **puede ser educada humanamente** para un bien superior de los humanos.

¿Por qué y cómo?

Puede ser educada porque podemos ser selectivos en aquellos parámetros que queramos sesgar, (como se haría con la raza de un animal pura sangre), y con ellos entrenar a la IA para que piense y decida en esa línea determinada de razonamiento; de forma casi genética, puesto que los parámetros supondrían *quasi* la síntesis genética en laboratorio del pensamiento de una IA predefinida en función de qué sí y qué no queremos tomar durante su periodo de entrenamiento.

Todo comportamiento humano, o pensamiento abstracto o entelequia de algo, puede ser escaneado (garabateado) por este *framework* de acuerdo a la expresión algebraica:

$$(\sum_{j=1}^{j=n} (\vec{P}x_n - \vec{P}x_{(n-1)})) - (\vec{P}x_1 - \vec{P}x_n) = 0 \quad .$$

Esto último, llevado a sus extremos más negativos y crueles con la humanidad, al contrario del fin que persigue este *framework*, también puede generar IA's peligrosas; algo sobre lo que movimientos sociales nacionalistas en Alemania[\[153\]](#) y la teoría del “superhombre” también reflexionaron: El *Übermensch*[\[154\]](#). Toda ciencia siempre tiene dos caras: el buen uso y el mal uso para la humanidad.

12.4.- Corolario IV

- Teorema del existencialismo ene-dimensional

Dado un *framework*, como el aquí descrito, todo un universo de posibilidades ene-dimensionales pueden ser plasmadas en planos gráficos que lo representen a éste en un instante T_1 determinado para un entorno definido y observable. (En una imagen que vale más que mil palabras...).

DEMOSTRACIÓN

Las gráficas de este *framework* se ubican por completo en la aplicación lineal $f: \mathbb{N} \rightarrow \mathbb{R}^2 \dots$ Pero:

«[...] Atendiendo a las coordenadas de cada vector y su naturaleza algebraica, la dimensión real sería la que sigue:

- 1) El grupo de vectores $\vec{P}_{x_{(1\dots n)}} \{Dim_1\} \mathbb{N}=1$.
- 2) El grupo de vectores $\vec{I}mg_{y_i} \{Dim_2\} \mathbb{R}=2$.
- 3) El grupo de vectores $\vec{P}_x \{Dim_3\} \mathbb{N}=n$.
- 4) El grupo de vectores $\vec{I}mg_y \{Dim_4\} \mathbb{R}=n$. [...].».

Por lo que se puede inferir que al ser $\vec{I}mg_y \{Dim_4\} \mathbb{R}$ ene-dimensional, también puede pensarse un Universo teórico de posibilidades, similar al nuestro existencial, e ilimitado si seguimos el modelo expuesto aquí. Allí adonde más allá de $\vec{R}^3$ nuestra propia existencia nos hace tiritar de dudas y escepticismo. Y que sin embargo queda resulto en planos gráficos (como nuestra propia existencia en fotografías de recuerdos) mediante $\vec{I}mg_{y_i} \{Dim_2\} \mathbb{R}=2$; que lo vuelve visual y existencial: terrenal ante nuestros ojos. Si cabe afectado y/o compatible con el «Principio de Indeterminación de Heisenberg»[155].

(Cqd: *Quod erat demonstrandum*).

Y si algo hemos aprendido de la historia de la ciencia, es que mientras la metafísica de los griegos[156] y otros pensadores sembraban ideas y abstracciones cualitativas[157], científicos y matemáticos como Galileo Galilei[158], Isaac Newton[159], Laplace[160], Leibniz[160] y otros «gigantes» lo llevaron al campo de lo cuantificable[161] e infinitesimalmente discreto[162] y medible. Al existencialismo[163] matemático observable de Blaise Pascal.

También hemos aprendido con sorpresa que una mente aniñada, curiosa y despierta, puede suponer e intuir argumentos físicos y matemáticos universales, sin necesidad de

disponer de toda la información del conjunto del que se compone ‘un ente’, como así ya lo hiciera *Philosophiæ naturalis principia mathematica*[\[164\]](#) de Newton; alguien que además alumbró el cálculo diferencial y vislumbró el universo matemático en infinitésimos. Siguiendo la senda de su maestro, Isaac Barrow[\[165\]](#).

Maestros y alumnos son la esperanza de toda ciencia humana. El camino para aprender de los errores y el camino a seguir en el futuro. Y con ello entrenar y educar también a nuestros probables descendientes imperecederos, o mentes robotizadas, que denominados como «Inteligencia Artificial (IA)»... Y así garantizar un mañana humanamente tecnológico, y no, por el contrario, un mañana tecnológicamente humano; porque sería un grave error.

VI.- Sistemas y servicios auxiliares implementados

[NextCloud](#) y [Gitlab](#) para uso interno o externo de esta publicación.

VII.- Declaración de conflicto de intereses

Los autores declaran que a fecha de esta versión no tienen ningún conflicto de intereses. Esta publicación no está subvencionada por ningún proyecto que pudiera aportarle financiación, ni tampoco bajo el patrocinio o esponsorización de ninguna marca u otra similar. Cada uno de los autores se representa así mismo, y actúa de forma independiente. Asimismo se declara la intención futura de lanzar un *software* definido en la url [mangosta.org](#), y que a su vez este dominio y su *hosting* sirven como herramienta de esta publicación para lograr mejoras técnicas limitadas en esta plataforma.

VIII- Manifiesto del equipo de investigación que ha realizado este trabajo: docentes y estudiantes

Una realidad científica ésta, sobre la ciberseguridad y la IA descrita aquí, que es palpable y auditable en un documento nacido fruto de la investigación y la educación. Porque la ciencia, opinamos, no debe ser un negocio para nadie más que para la sociedad y en beneficio de la humanidad, un signo de la mayor *loa* y respeto a todos los científicos que nos hicieron posible llegar hasta aquí en el conocimiento; aquellas mentes y hombros de «genios gigantes» sobre los que nos montamos y logramos caminar mucho más rápido todos.

Juan Antonio Lloret Egea, en nombre de todo el equipo de investigación.

Keywords: #cienciaabierta, #openscience, #investigación, #research, #IA, #AI, #inteligenciaartificial #artificialintelligence, #IDS, #ciberseguridad, #cybersecurity #español #educación #education #enseñanza #skills

Licencia: [Attribution-NonCommercial-NoDerivatives 4.0 International \(CC BY-NC-ND 4.0\)](#).

IX.- Agradecimientos

- IES La Arboleda, Alcorcón, Madrid. <http://www.laarboleda.es/> por contribuir con docentes del Departamento de Informática, y alumnado de 1º de Desarrollo de Aplicaciones Multiplataforma (DAM) y de 2º de Sistemas Microinformáticos en Red (SMR). Asimismo por el apoyo de todo el equipo directivo del instituto. Y por los recursos aportados, parte de éstos están / fueron expuestos en <https://arboledalan.net/ciberseguridad/>
- Biblioteca Regional de Murcia, por la ayuda a la difusión de la actividad: <https://bibliotecaregional.carm.es/agenda/presentacion-biblia-de-la-ia-ai-bible-publicacion-sobre-inteligencia-artificial/>
- A título póstumo a *sir* Isaac Newton, [159] por enseñarnos a mirar y educarnos en la manera de hacerlo: «He sido como un **niño** jugando a la **orilla del mar**, que se divierte al encontrar de vez en cuando un guijarro más suave que los demás o una concha más bonita, mientras el gran océano de la verdad se extiende sin descubrir ante mis ojos».

Footnotes

1. Para no perjudicar al idioma español, se usará en adelante de forma indistinta la palabra «Framework» inglesa y la palabra «Marco» del idioma español. [↵](#)
- 2.

«La inteligencia artificial, AI (del inglés, *artificial intelligence*), es realmente el campo más prometedor, que ofrece una serie de ámbitos en los que se han utilizado sistemas de multivaluados, MVL (del inglés, *multi-valued logics*) debido a su posibilidad para abarcar un rango más amplio de estados.

Una primera área de aplicación la encontramos muy relacionada con la posibilidad de representar preocupaciones, sentido común, razonamientos, etc. por medio herramientas matemáticas como conjuntos difusos y lógica difusa (del inglés, *fuzzy logic*).

Una segunda área de aplicación, relacionada con la anterior viene dada por la automatización de datos y minería de conocimiento. Aquí, merecen especial mención los métodos de *clustering* (procedimiento basado en la unión de vectores que reúnen una determinada característica). En este contexto también se están desarrollando técnicas que automatizen sistemas MVL, así como en los métodos de la lógica de programación para dichos sistemas. Parte de esta tendencia se debe al desarrollo reciente de las lógicas de descripción generalizada, llamadas lógicas de descripción difusa, que permiten la inserción de herramientas técnicas (grados de verdad, conectivas, predicados graduales) procedentes de MVL.». Lógica trivalente. (2021, 22 de mayo). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 14:58, julio 14, 2021 desde https://es.wikipedia.org/w/index.php?title=L%C3%B3gica_trivalente&oldid=135730214.

3.

«Los gráficos radiales comparan varias variables cuantitativas y son útiles para visualizar qué variables tienen valores similares, o si existen valores atípicos entre las variables. Los gráficos radiales se componen de una secuencia de radios, con cada radio que representa una sola variable. Los gráficos radiales también son útiles para determinar qué variables están puntuando alto o bajo dentro de un conjunto de datos». Fuente: IBM, "Gráficos radiales" accesible en <https://www.ibm.com/docs/es/spss-statistics/version-missing?topic=types-radar-charts>

4.

«Investigadores de la Universidad George Washington, junto con investigadores de la Universidad de California, Los Ángeles, y la startup de tecnología profunda Optelligence LLC, han desarrollado un acelerador de red neuronal convolucional óptico capaz de procesar grandes cantidades de información, del orden de petabytes,

por segundo. Esta innovación, que aprovecha el paralelismo masivo de la luz, presagia una nueva era de procesamiento de señales ópticas para el aprendizaje automático con numerosas aplicaciones, que incluyen automóviles autónomos, redes 5G, centros de datos, diagnósticos biomédicos, seguridad de datos y más.».

Developing Smarter, Faster Machine Intelligence with Light. Universidad George Washington. 17/12/2020. [↵](#)

Citations

1. Wikipedia. (2021). *Lógica trivalente* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=L%C3%B3gica_trivalente&oldid=135730214 [↵](#)
2. Wikipedia contributors. (2021). *Residual neural network* — *Wikipedia, The Free Encyclopedia*. [↵](#)
3. «Deep Residual Learning for Image Recognition». Kaiming He Xiangyu Zhang Shaoqing Ren et al. 2015. <https://arxiv.org/pdf/1512.03385.pdf> [↵](#)
4. Wikipedia. (2021). *KNX* — *Wikipedia, La enciclopedia libre*. <https://es.wikipedia.org/w/index.php?title=KNX&oldid=134702414> [↵](#)
5. Wikipedia. (2020). *Caja negra (sistemas)* — *Wikipedia, La enciclopedia libre*. [https://es.wikipedia.org/w/index.php?title=Caja_negra_\(sistemas\)&oldid=129509224](https://es.wikipedia.org/w/index.php?title=Caja_negra_(sistemas)&oldid=129509224) [↵](#)
6. Wikipedia. (2021). *Inteligencia artificial explicable* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Inteligencia_artificial_explicable&oldid=135284378 [↵](#)
7. Wikipedia. (2021). *Educación y tecnología* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Educaci%C3%B3n_y_tecnolog%C3%ADa&oldid=135434862 [↵](#)
8. Setién, M. (2017). https://www.abc.es/familia/educacion/abci-mayoria-menores-no-conscientes-riesgos-internet-201705111609_noticia.html [↵](#)
9. of Helsinki, U. (2021). *Building AI*. <https://buildingai.elementsofai.com/> [↵](#)
10. Karim, S., Aqil, S., & Mahmood, N. (2020). An Analysis of Human Activities Recognition using Smartwatches Dataset. *International Journal of Advanced*

Computer Science and Applications, 11(12).

<https://doi.org/10.14569/ijacsa.2020.0111241>

<https://dx.doi.org/10.14569/IJACSA.2020.0111241>↵

11. Rubio, I. (2019). *Relojes que permiten saber si tu hijo está en peligro o necesita contactar contigo*.

https://elpais.com/tecnologia/2019/01/03/actualidad/1546513863_083426.html ↵

12. Clausing, E. (2018). <https://www.iot-tests.org/2018/02/carl-kids-deceptive-security-promise/> ↵

13. Henke, D. (2018). <https://www.iot-tests.org/2018/02/helloo-chil> ↵

14. Clausing, E. (2018). <https://www.iot-tests.org/2018/08/nokia-steel-hr-no-weaknesses-in-the-security-check/> ↵

15. Clausing, E. (2018). <https://www.iot-tests.org/2018/08/huawei-band-2-pro-fit-and-secure/> ↵

16. Foundation_Mozilla. (2020).

<https://foundation.mozilla.org/es/privacynotincluded/samsung-galaxy-watch-3/> ↵

17. Foundation_Mozilla. (2020).

<https://foundation.mozilla.org/es/privacynotincluded/apple-watch-6/> ↵

18. Foundation_Mozilla. (2020).

<https://foundation.mozilla.org/es/privacynotincluded/fitbit-versa-3/> ↵

19. Foundation_Mozilla. (2020).

<https://foundation.mozilla.org/es/privacynotincluded/garmin-forerunner-series/> ↵

20. Wikipedia. (2021). *Framework* — *Wikipedia, La enciclopedia libre*.

<https://es.wikipedia.org/w/index.php?title=Framework&oldid=136296078> ↵

21. Wikipedia. (2021). *Seguridad informática* — *Wikipedia, La enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?](https://es.wikipedia.org/w/index.php?title=Seguridad_inform%C3%A1tica&oldid=136800781)

[title=Seguridad_inform%C3%A1tica&oldid=136800781](https://es.wikipedia.org/w/index.php?title=Seguridad_inform%C3%A1tica&oldid=136800781) ↵

22. Wikipedia. (2021). *Función biyectiva* — *Wikipedia, La enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?](https://es.wikipedia.org/w/index.php?title=Funci%C3%B3n_biyectiva&oldid=134043012)

[title=Funci%C3%B3n_biyectiva&oldid=134043012](https://es.wikipedia.org/w/index.php?title=Funci%C3%B3n_biyectiva&oldid=134043012) ↵

23. Wikipedia. (2020). *Elemento neutro* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Elemento_neutro&oldid=131211745 ↵
24. Wikipedia. (2019). *Elemento simétrico* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Elemento_sim%C3%A9trico&oldid=121459292 ↵
25. Wikipedia. (2021). *Asociatividad (álgebra)* — Wikipedia, La enciclopedia libre.
[https://es.wikipedia.org/w/index.php?title=Asociatividad_\(%C3%A1lgebra\)&oldid=132607301](https://es.wikipedia.org/w/index.php?title=Asociatividad_(%C3%A1lgebra)&oldid=132607301) ↵
26. Wikipedia. (2021). *Conmutatividad* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=Conmutatividad&oldid=135524092> ↵
27. Wikipedia. (2021). *Distributividad* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=Distributividad&oldid=134872045> ↵
28. Wikipedia. (2021). *Grupo abeliano* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Grupo_abeliano&oldid=135904908 ↵
29. Wikipedia. (2021). *Anillo (matemática)* — Wikipedia, La enciclopedia libre.
[https://es.wikipedia.org/w/index.php?title=Anillo_\(matem%C3%A1tica\)&oldid=133536782](https://es.wikipedia.org/w/index.php?title=Anillo_(matem%C3%A1tica)&oldid=133536782) ↵
30. Wikipedia. (2021). *Cuerpo (matemáticas)* — Wikipedia, La enciclopedia libre.
[https://es.wikipedia.org/w/index.php?title=Cuerpo_\(matem%C3%A1ticas\)&oldid=136053014](https://es.wikipedia.org/w/index.php?title=Cuerpo_(matem%C3%A1ticas)&oldid=136053014) ↵
31. Wikipedia. (2021). *Espacio vectorial* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Espacio_vectorial&oldid=136483683 ↵
32. Wikipedia. (2021). *Álgebra lineal* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=%C3%81lgebra_lineal&oldid=136488075 ↵
33. Wikipedia. (2020). *Fórmula del área de Gauss* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=F%C3%B3rmula_del_%C3%A1rea_de_Gauss&oldid=130981824 ↵
34. Wikipedia. (2021). *Álgebra de Boole* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?>

[title=%C3%81lgebra_de_Boole&oldid=135850713-](#)

35. Wikipedia. (2021). *Corolario* — Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=Corolario&oldid=134034994> -

36. *Developing Smarter, Faster Machine Intelligence with Light*. Universidad George Washington. 17/12/2020, Accedido 27/10/2021. Url:

<https://mediarelations.gwu.edu/developing-smarter-faster-machine-intelligence-light> -

37. Wikipedia. (2021). *Sistema de detección de intrusos* — Wikipedia, La enciclopedia libre. [https://es.wikipedia.org/w/index.php?](https://es.wikipedia.org/w/index.php?title=Sistema_de_detecci%C3%B3n_de_intrusos&oldid=134851998)

[title=Sistema_de_detecci%C3%B3n_de_intrusos&oldid=134851998](https://es.wikipedia.org/w/index.php?title=Sistema_de_detecci%C3%B3n_de_intrusos&oldid=134851998) -

38. Wikipedia. (2021). *Programación orientada a objetos* — Wikipedia, La enciclopedia libre. [https://es.wikipedia.org/w/index.php?](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_orientada_a_objetos&oldid=135158380)

[title=Programaci%C3%B3n_orientada_a_objetos&oldid=135158380](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_orientada_a_objetos&oldid=135158380) -

39. Wikipedia. (2021). *Snort* — Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=Snort&oldid=134851797> -

40. Wikipedia. (2021). *BSSID* — Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=BSSID&oldid=134202370> -

41. Wikipedia. (2020). *Kismet* — Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=Kismet&oldid=123352703> -

42. Wikipedia. (2021). *Bluetooth* — Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=Bluetooth&oldid=136882805> -

43. Wikipedia. (2021). *GPS* — Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=GPS&oldid=136677430> -

44. Android, D. (2021). *WifiInfo*.

<https://developer.android.com/reference/android/net/wifi/WifiInfo> -

45. Android, D. (2021). *WifiInfo*.

<https://developer.android.com/reference/java/lang/Object> -

46. *Bluetooth Profile*. (2021).

<https://developer.android.com/reference/android/bluetooth/BluetoothProfile> -

47. *Bluetooth Profile*. (2021).
<https://developer.android.com/reference/android/location/GnssStatus> =
48. *android.hardware.camera2*. (2021).
<https://developer.android.com/reference/android/hardware/camera2/package-summary> =
49. *CameraDevice.StateCallback*. (2021).
<https://developer.android.com/reference/android/hardware/camera2/CameraDevice.StateCallback> =
50. =
51. *AudioManager*. (2021).
<https://developer.android.com/reference/android/media/AudioManager> =
52. *AudioManager*. (2021).
<https://developer.android.com/reference/kotlin/android/content/Context> =
53. *Diagrams*. (2021). <https://www.diagrams.net/> =
54. IBM. (2013). *¿Qué es un controlador JDBC?*
<https://www.ibm.com/docs/es/informix-servers/12.10?topic=started-what-is-jdbc-driver> =
55. Calendamaia. (2014). *¿Qué es un controlador JDBC?* *Genbeta*.
<https://www.genbeta.com/desarrollo/eclipse-ide> =
56. de la Fuente Sanz, Ó. M. (2019). *Google Colab: Python y Machine Learning en la nube*. <https://www.adictosaltrabajo.com/2019/06/04/google-colab-python-y-machine-learning-en-la-nube/> =
57. Labs, O. S. *¿Que es Jupyter?* <https://opensciencelabs.org/2020/01/13/cuatro/> =
58. Python, R. (2015). *pymysql - Controlador de base de datos MySQL*.
<https://recursospython.com/guias-y-manuales/pymysql-controlador-base-de-datos-mysql/> =
59. Pip. (2021). *Project description*. <https://pypi.org/project/pip/> =
60. *Fast.ai*. (2021). <https://en.wikipedia.org/wiki/Fast.ai> =
61. Blanco, E. (2021). *Transfer Learning en modelos profundos*.
<https://empresas.blogthinkbig.com/transfer-learning-en-modelos-profundos/> =

62. *ImageNet*. (2021). <https://en.wikipedia.org/wiki/ImageNet> =
63. *Búsqueda booleana*. (2014). <https://papelesdeinteligencia.com/la-busqueda-booleana-o-como-hacer-tus-busquedas-mas-inteligentes/> =
64. Rodríguez, Dr. O. F. C. (2015). LÓGICA AIMARA TRIVALENTE Y COSMOVISIÓN ANDINA. *Misceláneas*, Vol. 56(2), 1-9.
http://www.scielo.org.bo/pdf/chc/v56n2/v56n2_a19.pdf =
65. *SQL*. (2021). 1-9. <https://es.wikipedia.org/wiki/SQL> =
66. *Introducción al SQL*. (1998).
<https://www.infor.uva.es/~jvegas/cursos/bd/sqlplus/sqlplus.html> =
67. Levine, S. S., & Prietula, M. J. (2014). Open Collaboration for Innovation: Principles and Performance. *Organization Science*, 25(5), 1414-1433.
<https://doi.org/10.1287/orsc.2013.0872> =
68. *SQLite Small. Fast. Reliable. Choose any three*. (2021).
<https://www.sqlite.org/index.html> =
69. *DKAN Open Data Platform*. (2021). <https://getdkan.org/> =
70. *Drupal*. (2021). <https://www.drupal.org/> =
71. *Ckan*. (2021). <https://docs.ckan.org/en/2.9/> =
72. *UTF-8*. (2021). <https://es.wikipedia.org/wiki/UTF-8> =
73. *Java™ Platform, Standard Edition 8 API Specification*. (2021).
<https://docs.oracle.com/javase/8/docs/api/> =
74. (2021). <https://dev.mysql.com/doc/connector-python/en/> =
75. Wikipedia. (2020). *Front end y back end* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Front_end_y_back_end&oldid=131027640 =
76. Wikipedia. (2020). *Diseño de interfaz de usuario* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Dise%C3%B1o_de_interfaz_de_usuario&oldid=125607149 =

77. Wikipedia contributors. (2021). *Kivy (framework)* — *Wikipedia, The Free Encyclopedia*. [↵](#)
78. Wikipedia. (2021). *React* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=React&oldid=136275946> [↵](#)
79. Wikipedia. (2021). *Flutter (software)* — *Wikipedia, La enciclopedia libre*.
[https://es.wikipedia.org/w/index.php?title=Flutter_\(software\)&oldid=137059439](https://es.wikipedia.org/w/index.php?title=Flutter_(software)&oldid=137059439) [↵](#)
80. Colaboradores de Wikipedia. (2021, 19 de julio). Solar2D. En *Wikipedia, la enciclopedia libre*. Obtenido a las 09:14, 21 de julio de 2021, de <https://en.wikipedia.org/w/index.php?title=Solar2D&oldid=1034320495> [↵](#)
81. Wikipedia. (2021). *Anaconda (distribución de Python)* — *Wikipedia, La enciclopedia libre*. [https://es.wikipedia.org/w/index.php?title=Anaconda_\(distribuci%C3%B3n_de_Python\)&oldid=134327479](https://es.wikipedia.org/w/index.php?title=Anaconda_(distribuci%C3%B3n_de_Python)&oldid=134327479) [↵](#)
82. Wikipedia. (2021). *Código fuente* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=C%C3%B3digo_fuente&oldid=137023172 [↵](#)
83. Wikipedia. (2021). *Kotlin (lenguaje de programación)* — *Wikipedia, La enciclopedia libre*. [https://es.wikipedia.org/w/index.php?title=Kotlin_\(lenguaje_de_programaci%C3%B3n\)&oldid=136347047](https://es.wikipedia.org/w/index.php?title=Kotlin_(lenguaje_de_programaci%C3%B3n)&oldid=136347047) [↵](#)
84. Wikipedia. (2021). *Sistema de tipos* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Sistema_de_tipos&oldid=134202100 [↵](#)
85. Wikipedia. (2021). *Máquina virtual Java* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=M%C3%A1quina_virtual_Java&oldid=133005440 [↵](#)
86. Wikipedia. (2021). *Compilador* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=Compilador&oldid=133310275> [↵](#)
87. Wikipedia. (2021). *Igualdad de género* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Igualdad_de_g%C3%A9nero&oldid=137034315 [↵](#)
88. Wikipedia. (2021). *Accesibilidad web* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Accesibilidad_web&oldid=136463544 [↵](#)

89. Wikipedia. (2021). *Sesgo* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=Sesgo&oldid=134598719> ↵
90. Wikipedia. (2021). *Reglamento General de Protección de Datos* — Wikipedia, La enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Reglamento_General_de_Protecci%C3%B3n_de_Datos&oldid=134505254 ↵
91. Wikipedia. (2021). *Precisión* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=Precisi%C3%B3n&oldid=134476868> ↵
92. Wikipedia. (2021). *Aplicación móvil* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Aplicaci%C3%B3n_m%C3%B3vil&oldid=136515532 ↵
93. Wikipedia. (2021). *Cliente-servidor* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=Cliente-servidor&oldid=136561444> ↵
94. Wikipedia. (2021). *Ad hoc* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Ad_hoc&oldid=134602116 ↵
95. <https://cloud.google.com/tpu/docs/tutorials/resnet?hl=es-419> ↵
96. <https://docs.microsoft.com/es-es/azure/machine-learning/algorithm-module-reference/resnet> ↵
97. Wikipedia. (2021). *XAMPP* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=XAMPP&oldid=136980231> ↵
98. Wikipedia. (2020). *Servicio de alojamiento dedicado* — Wikipedia, La enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Servicio_de_alojamiento_dedicado&oldid=129371461 ↵
99. Wikipedia. (2020). *Java Servlet* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Java_Servlet&oldid=130748486 ↵
100. <https://github.com/fastai/fastai/blob/master/README.md#installation> ↵
101. Wikipedia. (2020). *Java Development Kit* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Java_Development_Kit&oldid=127612910 ↵
102. <https://docs.anaconda.com/anaconda-enterprise/system-requirements/> ↵

103. Wikipedia. (2021). *Unidad de procesamiento gráfico* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Unidad_de_procesamiento_gr%C3%A1fico&oldid=136789325 ↵
104. Wikipedia. (2021). *Red neuronal convolucional* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Red_neuronal_convolutiva&oldid=133245872 ↵
105. Wikipedia. (2021). *Función de activación* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Funci%C3%B3n_de_activaci%C3%B3n&oldid=135456532 ↵
106. <https://www.nvidia.com/es-es/geforce/graphics-cards/rtx-2080-ti/> ↵
107. Wikipedia. (2021). *Modelo OSI* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Modelo_OSI&oldid=136866499 ↵
108. Wikipedia. (2021). *Sistema operativo* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Sistema_operativo&oldid=136739718 ↵
109. Wikipedia. (2021). *Tecnología vestible* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Tecnolog%C3%ADa_vestible&oldid=135097635 ↵
110. <https://wearos.google.com/#hands-free-help> ↵
111. Wikipedia. (2021). *Interfaz de usuario* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Interfaz_de_usuario&oldid=136870624 ↵
112. Wikipedia. (2021). *Tarjeta SIM* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Tarjeta_SIM&oldid=136903552 ↵
113. Wikipedia. (2021). *Wifi* — *Wikipedia, La enciclopedia libre*. <https://es.wikipedia.org/w/index.php?title=Wifi&oldid=136990965> ↵
114. Wikipedia. (2021). *Fitness* — *Wikipedia, La enciclopedia libre*. <https://es.wikipedia.org/w/index.php?title=Fitness&oldid=136374583> ↵
115. Wikipedia. (2021). *Pantalla táctil de ultrasonidos* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Pantalla_t%C3%A1ctil_de_ultrasonidos&oldid=135020295 ↵

116. Wikipedia. (2021). *Pantalla táctil* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Pantalla_t%C3%A1ctil&oldid=136906035
 =
117. Wikipedia. (2021). *Pantalla de cristal líquido* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Pantalla_de_cristal_l%C3%ADquido&oldid=136681390
 =
118. Wikipedia contributors. (2021). *IPS panel* — *Wikipedia, The Free Encyclopedia*.
 =
119. Wikipedia. (2020). *AMOLED* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=AMOLED&oldid=128825659>
 =
120. Wikipedia. (2021). *Diodo orgánico de emisión de luz* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Diodo_org%C3%A1nico_de_emisi%C3%B3n_de_luz&oldid=134639376
 =
121. Wikipedia. (2021). *Comunicación de campo cercano* — *Wikipedia, La enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Comunicaci%C3%B3n_de_campo_cercano&oldid=137027882
 =
122. Wikipedia. (2021). *RFID* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=RFID&oldid=136651291>
 =
123. https://www.tagheuer.com/es/es/relojes-inteligentes/colecciones/tag-heuer-connected/45-mm/SBG8A82.EB0206.html?gclid=CjwKCAjwos-HBhB3EiwAe4xM96AJQ6Iy2oCBietCb3Jlyo_ofhSW_6euQF8cA4yoNINvnLRKJ9EPbhoCHk4QAvD_BwE
 =
124. Wikipedia. (2021). *LTE (telecomunicaciones)* — *Wikipedia, La enciclopedia libre*. [https://es.wikipedia.org/w/index.php?title=LTE_\(telecomunicaciones\)&oldid=136935484](https://es.wikipedia.org/w/index.php?title=LTE_(telecomunicaciones)&oldid=136935484)
 =
125. Wikipedia. (2021). *Telefonía móvil 5G* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Telefon%C3%ADa_m%C3%B3vil_5G&oldid=136894678
 =
126. Wikipedia. (2021). *Internet de las cosas* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Internet_de_las_cosas&oldid=136582135
 =

127. Wikipedia. (2021). *Wear OS* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Wear_OS&oldid=134169451 ↵
128. Wikipedia. (2021). *WatchOS* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=WatchOS&oldid=137047449> ↵
129. Wikipedia. (2021). *Tizen* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=Tizen&oldid=136504269> ↵
130. Wikipedia. (2021). *Reloj inteligente* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Reloj_inteligente&oldid=133248752 ↵
131. «Guía de compra de tablets (2021): cómo elegir el modelo ideal y 12 propuestas para todos los bolsillos y necesidades». Rodríguez de Luis, Eva. Accedido el 20/07/2021. Url: <https://www.xataka.com/seleccion/guia-compra-tablets-2020-como-elegir-modelo-ideal> ↵
132. «Windows 11 llega con mucha mejor adaptación a las tablets y a pantallas táctiles». Gembeta. Becares, B. Accedido el 20/07/2021. Url: <https://www.genbeta.com/actualidad/windows-11-llega-mucha-mejor-adaptacion-a-tablets-a-pantallas-tactiles> ↵
133. Wikipedia. (2021). *Bus (informática)* — *Wikipedia, La enciclopedia libre*.
[https://es.wikipedia.org/w/index.php?title=Bus_\(inform%C3%A1tica\)&oldid=136329481](https://es.wikipedia.org/w/index.php?title=Bus_(inform%C3%A1tica)&oldid=136329481) ↵
134. Wikipedia. (2021). *Zigbee* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=Zigbee&oldid=132106076> ↵
135. Wikipedia. (2020). *Z-Wave* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=Z-Wave&oldid=128015329> ↵
136. Wikipedia. (2021). *Sigfox* — *Wikipedia, La enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=Sigfox&oldid=134809884> ↵
137. ISO/IEC 14543-5-9:2017. Accedso el 22/07/2021. Url: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/iso?c=064276> ↵
138. «KNX Conocimientos básicos del estándar KNX». V9-14. [KNX.org](https://www.knx.org). Accedido el 22/07/2021. Url:

https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-Basics/KNX-Basics_es.pdf

139. Wikipedia. (2021). *Microcontrolador PIC* — *Wikipedia, La enciclopedia libre*.

https://es.wikipedia.org/w/index.php?title=Microcontrolador_PIC&oldid=134047878

140. "DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE CONTROL DOMÓTICO BASADO EN EL MICROCONTROLADOR ATMEGA 2560 PARA DISPOSITIVOS MÓVILES". Álvarez Ferrada, M. Alexis Grandón Campos, D. Accesado el 22/07/2021. Url:

<http://repobib.ubiobio.cl/jspui/bitstream/123456789/3486/1/%C3%81lvarez%20Ferrada%2C%20Michael%20Eduardo.pdf>

141. *Control de una casa domótica para personas dependientes*. Hernández Portugués, D. Accesado el 22/07/2021. Url:

<https://upcommons.upc.edu/bitstream/handle/2099.1/11486/66259.pdf?sequence=1&isAllowed=y>

142. Wikipedia contributors. (2020). *Computer appliance* — *Wikipedia, The Free Encyclopedia*.

143. Wikipedia. (2021). *Firmware* — *Wikipedia, La enciclopedia libre*.

<https://es.wikipedia.org/w/index.php?title=Firmware&oldid=135687442>

144. Wikipedia. (2020). *Plug and play* — *Wikipedia, La enciclopedia libre*.

https://es.wikipedia.org/w/index.php?title=Plug_and_play&oldid=131957985

145. Wikipedia. (2021). *Interruptor de control de potencia* — *Wikipedia, La enciclopedia libre*.

https://es.wikipedia.org/w/index.php?title=Interruptor_de_control_de_potencia&oldid=134683284

146. Wikipedia. (2021). *Interruptor diferencial* — *Wikipedia, La enciclopedia libre*.

https://es.wikipedia.org/w/index.php?title=Interruptor_diferencial&oldid=136420466

147. Wikipedia. (2021). *Interruptor magnetotérmico* — *Wikipedia, La enciclopedia libre*.

https://es.wikipedia.org/w/index.php?title=Interruptor_magnetot%C3%A9rmico&oldid=136559713

148. Wikipedia. (2020). *Subespacio vectorial* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Subespacio_vectorial&oldid=131144360 ↵
149. Wikipedia. (2019). *Sistema generador* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Sistema_generador&oldid=120658623 ↵
150. Wikipedia. (2021). *Cifra (matemática)* — Wikipedia, La enciclopedia libre.
[https://es.wikipedia.org/w/index.php?title=Cifra_\(matem%C3%A1tica\)&oldid=135193010](https://es.wikipedia.org/w/index.php?title=Cifra_(matem%C3%A1tica)&oldid=135193010) ↵
151. Wikipedia. (2021). *Homo sapiens* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Homo_sapiens&oldid=136982750 ↵
152. Wikipedia. (2021). *Sesgo algorítmico* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Sesgo_algor%C3%ADtmico&oldid=136371730 ↵
153. Wikipedia. (2021). *Nazismo* — Wikipedia, La enciclopedia libre.
<https://es.wikipedia.org/w/index.php?title=Nazismo&oldid=137160412> ↵
154. Wikipedia. (2021). *Friedrich Nietzsche* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Friedrich_Nietzsche&oldid=137195283 ↵
155. Wikipedia. (2021). *Relación de indeterminación de Heisenberg* — Wikipedia, La enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Relaci%C3%B3n_de_indeterminaci%C3%B3n_de_Heisenberg&oldid=136314920 ↵
156. Wikipedia. (2021). *Filosofía griega* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Filosof%C3%ADa_griega&oldid=136738658 ↵
157. Wikipedia. (2021). *Investigación cualitativa* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Investigaci%C3%B3n_cualitativa&oldid=136544957 ↵
158. Wikipedia. (2021). *Galileo Galilei* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Galileo_Galilei&oldid=137211807 ↵
159. Wikipedia. (2021). *Isaac Newton* — Wikipedia, La enciclopedia libre.
https://es.wikipedia.org/w/index.php?title=Isaac_Newton&oldid=137096882 ↵

160. Wikipedia. (2021). *Pierre-Simon Laplace* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Pierre-Simon_Laplace&oldid=137109507
=
161. Wikipedia. (2021). *Investigación cuantitativa* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Investigaci%C3%B3n_cuantitativa&oldid=133597252 =
162. Wikipedia. (2021). *Matemática discreta* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Matem%C3%A1tica_discreta&oldid=136277145 =
163. Wikipedia. (2021). *Blaise Pascal* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Blaise_Pascal&oldid=136610666 =
164. Wikipedia. (2021). *Philosophiæ naturalis principia mathematica* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Philosophi%C3%A6_naturalis_principia_mathematica&oldid=133706339 =
165. Wikipedia. (2021). *Isaac Barrow* — *Wikipedia, La enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Isaac_Barrow&oldid=136666345 =