"Protección de datos con inteligencia artificial en el sector financiero"

TESIS

QUE PARA OBTENER EL GRADO DE:

Maestro En Tecnologías De La Información

PRESENTA:
Ing. Marco Antonio Corchado Reyes

DIRECTORA DE TESIS
Mtra. Elva Bernal Rodriguez

CUAUTITLÁN IZCALLI, EDO. DE MÉXICO. AGOSTO, 2022

"2022. Año del Quincentenario de Toluca, Capital del Estado de México".

Cuautitlán Izcalli, Estado de México a 08 de julio de 2022
TESCI/DIDT/062/VI/22

DIRECCIÓN ACADÉMICA
DEPARTAMENTO DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
COORDINACIÓN DE POSGRADO

INGENIERO
MARCO ANTONIO CORCHADO REYES
P R E S E N T E

Por este conducto me permito informarle que puede proceder a la digitalización del Trabajo de Tesis titulado:

"PROTECCIÓN DE DATOS CON INTELIGENCIA ARTIFICIAL EN EL SECTOR FINANCIERO"

Ya que la comisión encargada de revisar el trabajo que se presenta para efectos de titulación, han dado su autorización conforme a lo estipulado en el Lineamiento para la operación de los Estudios de Posgrado en el Sistema Nacional de Institutos Tecnológicos.

Sin nada más que agregar, quedo a sus órdenes para cualquier aclaración.

A T E N T A M E N T E

Recibí Original
Marco Antonio Corchado Reyes
08/07/22

MTRA. ROCÍO ORTEGA JIMÉNEZ
DEPARTAMENTO DE
INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
COORDINACIÓN DE POSGRADO

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CUAUTITLÁN IZCALLI
UNIDAD DE POSGRADO

c.c.p. Archivo
Departamento de Titulación
Expediente del alumno

Índice

Agradecimientos

Agradezco a mis padres el Sr. Roberto Corchado Ibañez y la Sra. Adriana Alicia Reyes Mundo, por siempre motivarme a seguir adelante, brindarme su apoyo incondicional ante cualquier situación, por llenarme de amor, sabiduría y siempre creer en mí.

A mis hermanos Roberto y Rosaura, por siempre estar en los momentos buenos y malos, por enseñarme que en la vida por muy difícil que sean los obstáculos no hay que rendirse, por enseñarme a que la familia siempre estará presente a pesar de la distancia.

A mis profesores, por compartir sus conocimientos ya que a pesar de la pandemia sanitaria dieron todo de ellos para seguir con el curso.

A la Mtra. Consuello Macias y al Ing. Enrique Martínez Negrete QEPD, que desde la licenciatura me brindaron sus enseñanzas no solo a nivel académico, sino también a nivel personal, me enseñaron a no rendirme, a siempre tratar de dar lo mejor de mí, y sobre todo a enfrentar los problemas con madurez, que ahora como padre de familia sus consejos me han ayudado a crecer como persona.

A la Mtra. Elva Bernal Rodriguez, por su liderazgo, su apoyo, paciencia, enseñanzas y entusiasmo en desarrollar este trabajo de tesis.

Y agradezco en especial a mi esposa la Lic. Carina Salgado, e hijos José Miguel y Natalia, por ser mi inspiración, por llenarme de amor día a día, su amor incondicional me motiva a ser un mejor padre, un mejor esposo y una mejor persona, y que sin el apoyo de ellos no me hubiera sido posible estar pasando por esta etapa de mi vida académica.

Estado del arte

Desarrollo del estado del arte en investigación: una herramienta basada en inteligencia artificial

El trabajo describe el desarrollo de un prototipo de software basado en inteligencia artificial para la construcción semi-asistida del estado del arte en un proceso de investigación (SASR). La herramienta permite obtener información con valor agregado y mapas, visualizaciones, para la escritura de un estado del arte de un tema de investigación. Los resultados son verificados mediante una encuesta a 50 estudiantes de maestría y doctorado, quienes reportan la utilidad de la herramienta.

En el documento se presenta la arquitectura y la implementación SASR. Finalmente se presenta un estudio comparativo con las herramientas existentes, se discuten las potencialidades de la herramienta y el impacto positivo que puede tener SASR en la investigación, principalmente en el contexto colombiano. Así también se discuten las técnicas de inteligencia artificial implementadas, la escalabilidad de la herramienta y la facilidad de integrar nuevos análisis y visualizaciones.

Investigación y Desarrollo de Técnicas de Scraping

Se elaboran 3 patrones básicos de búsqueda: recursivo, buscadores y e-commerce, para estandarizar diferentes tipos de búsqueda. Utilizando patrones de diseño se elabora un sistema eficaz, modular y totalmente escalable. Se implementa, además, un buscador para filtrar las direcciones según el contenido deseado.

Se define el Web Scraping como un conjunto de técnicas que nos permiten, de forma automática, extraer datos de sitios web. En definitiva, se trata de obtener, analizar y conocer el código HTML devuelto por cualquier sitio tras una petición HTTP:GET.

El web scraping no es, en sí, sólo un software. Es mucho más allá de una simple herramienta. La personalización y la posibilidad de implementar nueva funcionalidad son claves en un mercado cada vez más exigente. Se ha visto cómo las librerías más potentes para elaborar bots no tienen valor (de hecho, son gratuitas), mientras que las empresas más caras son aquellas que separan totalmente la capa de recolección de datos del servicio ofrecido: Data As A Service.

Utilizando patrones de diseño se han elaborado los cimientos de lo que podemos definir como un buen crawler, que permite seccionar targets estandarizando diferentes tipos de motores para obtener información según qué tipo de página y, además, de una forma sencilla permite modificar y ampliar la funcionalidad respetando los principios básicos del buen software, obteniendo, por tanto, un concepto de sistema modulable, escalable, personalizable y estandarizado.

Además, se ha definido un método de 3 pasos que permite:

- Obtener URLs personalizando el algoritmo.
- Buscar y filtrar los resultados según el contenido deseado.
- Tratamiento final del resultado y exportación a formatos estandarizados comunes.

Sistema para el monitoreo y gestión de datos, servicios y archivos con notificaciones electrónicas basado en PHP y Shell Script en servidores con sistema operativo GNU/LINUX Ubuntu Server caso: Inversiones INTRAWEB, C.A.

Durante el levantamiento de información en el servidor se enumeraron los servicios, archivos de configuración y datos que en ellos se ejecutan, se destacó que presentan fallas semanales en los servidores, que son solventadas mediante la contratación de personal externo a la empresa, especialistas en servidores Ubuntu Server, generando un gasto en honorarios profesionales.

El análisis de datos recolectados reveló que es necesario contar con un sistema de monitoreo y gestión de datos que informe periódicamente el estado operativo del servidor de computación, además que no solo permite avisar cuando ocurra un problema, sino que también solvente la problemática presente. Cuando un servicio se detenga, permita reiniciarlo.

El diseño de la aplicación se realizó tomando como base la metodología de desarrollo James Senn, esta metodología al ser tradicional, se efectuó la documentación que permite el diseño lógico del sistema, como diagramas de flujo de datos, diccionarios de datos, diccionarios de procesos, entre otros. Esta documentación sirve de insumo indispensable en la empresa para cualquier cambio y/o evolución del sistema que se requiera realizar en un futuro.

Capítulo I Introducción

1.1 Planteamiento del problema

Las aplicaciones WEB con información ya sea pública o privada están al alcance de la creación de “robots” que automatizan las consultas mediante web services, por lo que no garantiza que se le dé un uso correcto a la información anteriormente mencionada. Adicional a que se compromete la infraestructura de los sitios haciendo consultas en un tiempo menor al soportado por los servidores.

Actualmente el sector financiero compite para dar créditos de manera más rápida y autónoma por lo que utilizan herramientas tecnológicas que les facilita esta tarea, estas herramientas con el paso del tiempo forman parte de la creación de un Big Data, en donde la información que se recaba no solo puede ser para la obtención de un crédito, si no para múltiples productos o servicios que la financiera ofrezca.

Las Fintech en México no tienen las mismas regulaciones que un banco, por lo que pueden usar la tecnología para agilizar sus procesos de otorgamientos de crédito.

En México podemos acceder a información sensible en ciertos portales del gobierno, iniciando por el CURP, este dato se ha convertido en el identificador único de cada ciudadano, por lo que al tenerlo podemos entrar a otros portales del gobierno, como el historial laboral donde podemos tener acceso a la información financiera, antigüedad de empleo, e incluso dirección, si automatizamos esta consulta se puede hacer un análisis de datos que permita conocer la situación económica de cada persona, que en las manos equivocadas puede ser de alto riesgo.

1.2 Justificación

El uso de Internet se ha convertido en algo de lo más natural como parte de la rutina del día a día de las personas, la mayoría de las veces se comparte información personal y en la mayor de las veces esta información puede ser sensible, de igual manera la mayoría de los sitios se conforman con estrictas normas de seguridad para evitar que esta información sea expuesta.

La presente investigación surge de un hueco de seguridad que la mayor parte de los programadores no contempla; este hueco es la automatización ya que el robot o programa que tenga como finalidad una consulta puede hacerla de manera masiva para extraer la información simulando ser un usuario.

La investigación busca implementar un sistema de seguridad en el se analize el comportamiento del servidor de manera autónoma con inteligencia artificial y programación común, para poder determinar cuando se realizan consultas normales y cuando son automatizadas; cuando se dé el caso de una consulta automatizada se bloqueará el sitio web.

1.3 Objetivos

1.3.1 Objetivo general

Proteger de automatizaciones de consultas a portales WEB con servicios de inteligencia artificial instalados en la infraestructura de la empresa dueña de datos, para evitar el riesgo más grande de Internet: el robo de identidad de plataformas en donde pudieran encontrarse datos sensibles para fines lucrativos de las FIntech en México.

Las plataformas del Gobierno de México actualmente no tienen una validación que garantice que la persona que está consultando sus datos sea la dueña de los mismos, algunos ejemplos de estas plataformas son: Renapo, SAT, Consulta de Cedula profesional, Semanas cotizadas IMSS e ISSSTE, entre otros.

1.3.2 Objetivos específicos

- Evitar que en los sitios WEB con o sin “Captcha” se pueda automatizar la consulta y extracción de datos con Web Scraping, o alguna otra técnica que potencialmente exponga la información del usuario.
- Implementar un modelo de seguridad para verificar que quien consulta no sea un programa de automatización (ROBOT), este modelo de seguridad analiza el tráfico del servidor.
- Brindar confianza al usuario de que su información esté segura, denegando el acceso cuando se detecte el uso de robots, la mayor parte de la seguridad informática se enfoca a ataques directos, pero la automatización no es un ataque.

Capítulo II Marco teórico

2.1 Web scraping

Web scraping es el proceso de recolectar datos contenidos en páginas web mediante técnicas automatizadas. Lo distintivo del web scraping es que en principio los datos parecen poco estructurados. Corresponde por tanto al analista de datos identificar cuál es el patrón que siguen los datos, para luego crear y ejecutar un algoritmo de extracción y procesamiento de los mismos. En la práctica lo que se hace es escribir un programa que envía consultas a un servidor web, recibe las respuestas (usualmente en forma de páginas web) y examina los datos para extraer la información necesaria.

El web scraping es la solución intermedia entre la recolección manual de datos (marcando, copiando y pegando textos) y el acceso automatizado a los mismos con base en un protocolo predeterminado (una interfaz de programación, API). Se aplica cuando tales protocolos no están disponibles y la cantidad de datos que se desea extraer es demasiado grande para que pueda ser realizada en forma manual.

Como herramienta de investigación, el web scraping multiplica las posibilidades de recolectar información que, aunque está publicada en Internet, es inaccesible por no tener una estructura clara, estar dispersa o ser masiva. Ocurre así por ejemplo con la información de los gobiernos, que por lo general generan altos volúmenes de datos, con una estructura ajustada solo a las necesidades de la burocracia y dispersa a lo largo de diferentes páginas y sitios web. Haciendo uso del web scraping es posible extraer de forma automatizada la información y darle sentido. (Lopez, 2018)

2.1.1 Extracción

Técnicas para la extracción de información

Web bot, Spider, Crawler, Arañas y Rastreadores: Inspeccionan las páginas web de internet de forma metódica y automatizada. Se usan para rastrear la red. Lee la estructura de hipertexto y accede a todos los enlaces referidos en el sitio web. Son utilizadas la mayoría de las veces para poder crear una copia de todas las páginas web visitadas para que después puedan ser procesadas por un motor de búsqueda; esto hace que se puedan indexar las páginas, proporcionando un sistema de búsquedas rápido

Plataformas de agregación verticales: Existen plataformas que tienen el propósito de crear y controlar numerosos robots que están destinados para mercados verticales específicos. Mediante el uso de esta preparación técnica se realiza mediante el establecimiento de la base de conocimientos destinado a la totalidad de plataformas verticales y luego a crearla automáticamente. Medimos nuestras plataformas por la calidad de la información que se obtiene. Esto asegura que la robustez de nuestras plataformas utilizadas consiga la información de calidad y no sólo fragmentos de datos inútiles.

Reorganización de la anotación semántica.

El desarrollo de web scraping puede realizarse para páginas web que adoptan marcas y anotaciones que pueden ser destinadas a localizar fragmentos específicos semánticos o metadatos. Las anotaciones pueden ser incrustadas en las páginas y esto puede ser visto como análisis de la representación estructurada (DOM). Esto permite recuperar instrucciones de datos desde cualquier capa de páginas web.

Herramientas utilizadas en la extracción:

- **ScraperWiki:** Es una plataforma web que permite crear scrapers de forma colaborativa entre programadores y periodistas para extraer y analizar datos públicos contenidos en la web.

- **PHP**: Cuenta con librerías para realizar web scraping como cURL, el cual permite la transferencia y descarga de datos, archivos y sitios completos a través de una amplia variedad de protocolos, y Crawl, que contiene varias opciones para especificar el comportamiento de la extracción como filtros Content-Type, manejo de cookies, manejo de robots y limitación de opciones

- **Guzzle:** Es un framework que incluye las herramientas necesarias para crear un cliente robusto de servicios web. Incluye: descripciones de Servicio para definir las entradas y salidas de una API, iteradores para recorrer webs paginadas, procesamiento por lotes para el envío de un gran número de solicitudes de la manera más eficiente posible. Fué creado usando Symfony2 y emplea la librería cURL de PHP.

- **Jsoup de Java:** Es una librería para realizar web scraping. Proporciona una API muy conveniente para la extracción y manipulación de datos, utilizando lo mejor de DOM, CSS, y métodos de jQuery similares.

    - Raspa y analiza el código HTML de una URL, archivo o cadena
    - Encuentra y extrae los datos, utilizando el DOM o selectores CSS
    - Manipula los elementos HTML, atributos y texto.
    - Limpia el contenido enviado por los usuarios contra una lista blanca de seguridad, para evitar ataques XSS.
    - Salida HTML ordenada

Beautifulsoup: Es una biblioteca de Python diseñada para proyectos de respuesta rápida como screen scraping o web scraping. Ofrece algunos métodos simples y modismos de Python para navegar, buscar y modificar un árbol de análisis: una herramienta para la disección de un documento y extraer lo que necesita, además de que no se necesita mucho código para escribir una aplicación. Beautiful Soup convierte automáticamente los documentos entrantes a Unicode y documentos salientes a UTF-8, también trabaja con analizadores de Python populares como lxml y html5lib y permite realizar el recorrido del DOM. (Hernández, 2015)

2.2 Servicios WEB

El consorcio W3C (World Wide Web Consortium) define un servicio web como un sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable. Cuenta con una interfaz descrita en un formato procesable por un equipo informático (específicamente en WSDL), a través de la que es posible interactuar con el mismo mediante el intercambio de mensajes SOAP, típicamente transmitidos usando serialización XML sobre HTTP conjuntamente con otros estándares web.

2.2.1 Tipos de Servicios Web

Los servicios web que han sido más relevantes pueden ser clasificados de la siguiente manera:

- **Remote Procedure Calls (RPC, Llamadas a Procedimientos Remotos):** están basados en RPC y presentan una interfaz de llamada a procedimientos remotos y funciones distribuidas. Es una comunicación nodo a nodo entre cliente y servidor, donde el cliente solicita que se ejecute cierto procedimiento o función y el servidor envía la respuesta. Las primeras referencias de servicios web estaban basadas en esta versión, sin embargo, las numerosas problemáticas por el acoplamiento entre los sistemas y el nacimiento de nuevas tecnologías, han quedado a éste casi olvidado.

- **Service Oriented Architecture (SOA, Arquitectura Orientada a Servicios):** es una arquitectura de aplicación en la cual todas las funciones están definidas como servicios independientes con interfaces invocables que pueden ser llamados en secuencias bien definidas para formar los procesos de negocio. Al contrario que los Servicios Web basados en RPC, este estilo es débilmente acoplado, lo cual es preferible ya que se centra en los servicios proporcionados por el documento WSDL, más que en los detalles de implementación con los distintos sistemas. El más relevante sería SOAP (Simple Object Access Protocol).

- **REST (Representational State Transfer):** es un conjunto de principios de arquitectura para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc.) y sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo SOAP. (Ruiz, 2019)

- **SOAP (Simple Object Access Protocol)**: este protocolo permite a un WS (Web Service) comunicar dos objetos, de diferentes procesos, mediante XML; es muy recomendable para entornos formales y donde las funcionalidades de interfaz y datos estén claramente definidas. (Castro, 2013)

2.2.2 Características de REST

REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web. El término fue introducido en la tesis doctoral de Roy Fielding en 2000, quien es uno de los principales autores de la especificación de HTTP.

REST no es más que una colección de recursos definidos y diseccionados. El término a menudo es utilizado para describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional como hace SOAP.

REST es un estilo de arquitectura basado en estándares como son HTTP, URL, la representación de los recursos: XML/HTML/GIF/JPEG/etc. y los tipos MIME: text/xml, text/html, …

La motivación de REST es la de capturar las características de la Web que la han hecho tan exitosa.

REST se ha centrado en explotar el éxito de la Web, que no es más que el uso de formatos de mensaje extensibles, estándares y un esquema de direccionamiento global.

En particular, el concepto central de la Web es un espacio de URIs unificado. Las URIs identifican recursos, los cuales son objetos conceptuales. La representación de tales objetos se distribuye por medio de mensajes a través de la Web. Este sistema es extremadamente desacoplado.

Las características principales de un modelo REST serían las siguientes:

- **Escalabilidad**: La variedad de sistemas y de clientes crece continuamente, pero cualquiera de ellos puede acceder a través de la Web. Gracias al protocolo HTTP, pueden interactuar con cualquier servidor HTTP sin ninguna configuración especial.

- **Independencia:** Los clientes y servidores pueden tener puestas en funcionamiento complejas. Diseñar un protocolo que permita este tipo de características resulta muy complicado. HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs.

- **Compatibilidad:** En ocasiones existen componentes intermedios que dificultan la comunicación entre sistemas, como pueden ser los firewalls. Las organizaciones protegen sus redes mediante firewalls y cierran casi todos los puertos TCP salvo el 80, el que usan los navegadores web. REST al utilizar HTTP sobre Transmission Control Protocol (TCP) en el puerto de red 80 no resulta bloqueado. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.

- **Identificación de recursos:** REST utiliza una sintaxis universal como es el uso de URIs. HTTP es un protocolo centrado en URIs, donde los recursos son los objetos lógicos a los que se envían mensajes.

- **Protocolo cliente/servidor sin estado:** Cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión.
- **Operaciones bien definidas:** HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE. (Ruiz, 2019)

2.2.3 Postman

Postman nace como una herramienta que inicialmente nos permite crear peticiones sobre APIs de una forma muy sencilla y poder de esta manera probar las APIs. Todo basado en una extensión de Google Chrome. El usuario de Postman puede ser un desarrollador que esté verificando el funcionamiento de una API para desarrollar sobre ella o un operador el cual esté realizando tareas de monitorización sobre un API. (Gonzales Quevedo, 2022)

Postman dejó de ser una extensión de Google Chrome y se convierte en una aplicación independiente, y podemos descargarla directamente desde https://www.postman.com/

2.2.4 Certificados SSL

En el modelo de referencia TCP/IP, SSL se introduce como una especie de nivel o capa adicional, situada entre la capa de aplicación y la capa de transporte. Lo anterior hace que sea independiente de la aplicación que lo utilice, es decir, que no solo puede ser utilizado para encriptar la comunicación entre un navegador y un servidor Web, sino también en cualquier aplicación como IMAP, FTP, Telnet, etc. También puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 214 bytes, volviendo a reensamblarlos en el receptor. Además, SSL establece una comunicación segura a nivel de socket (nombre de máquina más puerto), de forma transparente al usuario y a las aplicaciones que lo usan.

SSL es muy flexible con respecto a escoger el algoritmo de encriptación simétrico, la función de verificación de mensaje y el método de autenticación. La combinación de los elementos anteriores es conocida como suite de cifrado (Cipher Suite). Para la encriptación simétrica SSL puede usar los algoritmos DES (Data Encryption Standard), Triple DES, RC2, RC4, Fortezza e IDEA; para la verificación de mensajes puede usar MD5 (Message Digest Algorithm 5) o SHA-1 (Secure Hash Algorithm) como algoritmos de hashing y para la autenticación puede usar algoritmos RSA (Rivest, Shamir, Adelman) u operar en modo anónimo en donde se usa el intercambio de llaves de DiffieHellman.

Los algoritmos, longitudes de clave y funciones hash usados en SSL dependen del nivel de seguridad que se busque o se permita.

2.2.4.1 Versiones de certificados SSL

En 1994, Netscape Communications creó SSL v.2, la cual hacía posible mantener la confidencialidad en los números de las tarjetas de crédito y además autenticar al servidor web con el uso de encriptación y certificados digitales. En 1995, Netscape fortalece los algoritmos criptográficos y soluciona muchos de los problemas de seguridad existentes en SSL v.2 con la nueva versión SSL v.3, la cual soporta más algoritmos de seguridad que SSL v.2.

Por otra parte Internet Engineering Task Force (IETF) adoptó SSL para la creación de su protocolo Transport Layer Security (TLS) y WAP Forum adaptó este último para crear el protocolo inalámbrico equivalente, Wireless Transport Layer Security (WTLS). Conceptualmente, SSL, TLS y WTLS proveen el mismo servicio de seguridad: un canal seguro entre dos entidades, un cliente y un servidor.

Los navegadores más populares en la actualidad implementan SSL/TLS por defecto. Netscape Communicator (4.7) soporta solamente SSL y no TLS, mientras Netscape 6, MS Internet Explorer y Opera ofrecen soporte para ambos. Es importante señalar en este punto, además, que los navegadores mencionados anteriormente soportan todos los algoritmos simétricos RC2, RC4, DES y Triple DES y las funciones hash MD5 y SHA-1. Todos ellos brindan soporte total para RSA mientras que el establecimiento de llaves Diffie-Hellman todavía no se vislumbra.

Diferencias entre SSL v.2 y SSL v.3

La primera versión pública de SSL, versión 2, tenía un conjunto de desperfectos de seguridad que fueron corregidos en SSL v.3. Los navegadores en la actualidad todavía soportan SSL v.2 y en muchos otros sistemas todavía está en uso. He aquí un resumen de los principales problemas:

Las mismas llaves criptográficas son usadas para la autenticación de mensajes y para encriptar, lo cual significa que los MACs de los mensajes están innecesariamente debilitados (debido a las restricciones de exportación de los Estados Unidos, la longitud de la llave simétrica que puede usar Netscape e Internet Explorer fue limitada a 40 bits. Si la llave de encriptación es usada también para la autenticación de mensajes, la seguridad de los MACs es además afectada).

SSL v.2 no posee ninguna protección para la negociación (handshake) que tienen lugar entre cliente y servidor para establecer el intercambio de información, por tanto, un ataque persona-en-el-medio no puede ser detectado.

Finalmente, SSL v.2 simplemente utiliza el cerrado de conexión de TCP para indicar el fin del envío de datos, por tanto, un ataque puede sencillamente falsificar los TCP FINs y el receptor no puede decir que no es un fin de envío de datos legítimo (SSL v.3 soluciona este problema implementando una alerta de clausura explícita). Anterior a la propuesta de SSL v.3, estos problemas de seguridad fueron resueltos por Microsoft en su protocolo Private Communications Technology (PCT), el cual es muy similar a SSL/TLS y es aún soportado por los navegadores y servidores de Microsoft, pero SSL/TLS se ha convertido en un estándar y PCT no se ha podido imponer.

En adición al soporte para nuevos algoritmos de seguridad que ofrece SSL v.3, incorpora soporte para la carga de certificados en cadena. Esta característica permite al servidor pasar un certificado del servidor junto con los certificados de otros emisores al navegador. Según la especificación del protocolo, los objetivos de SSL v.3, en orden de prioridad, son los siguientes:

- **Seguridad criptográfica:** SSL debe ser usado para establecer una conexión segura entre dos partes.

- **Interoperabilidad:** Programadores independientes deben ser capaces de desarrollar aplicaciones utilizando SSL v.3 que puedan ser capaces de intercambiar satisfactoriamente parámetros criptográficos sin el conocimiento del código de otro.

- **Extensibilidad:** SSL v.3 pretende proveer un framework en el que nuevas llaves públicas y métodos de encriptación puedan ser incorporados si es necesario. Esto además conlleva dos objetivos: prevenir la necesidad de crear un nuevo protocolo (con el riesgo de introducir nuevas debilidades) y evitar la necesidad de implementar una nueva librería de seguridad completa.

- **Eficiencia relativa:** Las operaciones criptográficas tienden a hacer un uso intensivo del CPU, particularmente las operaciones de llave pública. Por esta razón, el protocolo SSL ha incorporado un esquema opcional de sesión oculta para reducir el número de conexiones que necesitan ser establecidas. Además, se han tomado precauciones para reducir la actividad en la red. Diferencias entre SSL v.3 y TLS El grupo de trabajo IETF adoptó el protocolo SSL v.3 y llevó a cabo pequeñas modificaciones para incrementar la seguridad, lo cual trajo como resultado el protocolo TLS. Algunas de estas modificaciones aparecen a continuación:

    - Las llaves criptográficas son ampliadas a partir de la mejora en el secreto intercambiado.

    - La construcción del MAC fue modificada ligeramente apareciendo un HMAC.

- Las implementaciones requirieron incluir soporte para el protocolo de intercambio de llaves Diffie-Hellman, el estándar de firma digital y para el algoritmo Triple-DES.

- Adiciona estandarización en el orden de los mensajes, más mensajes de alerta y más bloques de relleno a los bloques codificados.

Diferencias entre TLS y WTLS

El WAP Forum ha adaptado TLS para introducirlo en el entorno inalámbrico en dispositivos pequeños, los cuales tienen limitaciones en el ancho de banda, memoria y procesamiento. Las nuevas características son:

- WTLS incluye el uso de criptografía de curva elíptica por defecto.

- WTLS trabaja encima del datagrama en lugar de la capa de comunicación basada en conexión (comparado con UDP vs TCP en Internet).

- WTLS define su propio formato para los certificados optimizados por el tamaño, pero también soporta el certificado común X.509v3. (Martorell, 2006)

2.3 Fintech

El nombre proviene de las palabras inglesas financial technology y es la unión de las tecnologías digitales y los servicios financieros. Las empresas dedicadas a este ramo, las FinTech, utilizan nuevos modelos de negocios basados en el uso de estas tecnologías para brindar novedosos servicios financieros a personas, empresas y gobiernos (como sistemas de pagos móviles, préstamos de persona a persona, esquemas de financiamiento colectivo, etc.). Otorgan al usuario mayor control sobre sus finanzas y posibilitan nuevas formas de interacción económica y financiera, reduciendo la fricción (costos no monetarios, como por ejemplo, retrasos en tiempo al ejecutar transacciones financieras) y los costos de transacción.

FinTech promueve la inclusión financiera, ya que a través de las tecnologías digitales puede llegar a sectores de la población excluidos de los servicios financieros tradicionales.

2.3.1 Aplicaciones

FinTech ha introducido nuevos tipos de modelos de negocios y servicios financieros. Algunos de los que han tenido un mayor crecimiento e impacto son los siguientes:

Plataformas de pagos electrónicos: Existen diferentes modelos de negocios que usan este tipo de plataformas; algunos de los más comunes ofrecen al usuario una cuenta (no bancaria), llamada cartera digital, accesible a través de una plataforma en línea o una aplicación móvil. Esta refleja el saldo del usuario y le permite hacer depósitos, retiros o envíos directos a otros usuarios sin necesidad de un intermediario financiero tradicional. En México las plataformas de pagos han tenido gran crecimiento en los últimos años y actualmente existen más de cuarenta compañías incursionando en este sector.

Financiamiento colectivo o crowdfunding: Son plataformas que permiten a personas u organizaciones realizar campañas de financiamiento por internet para recaudar fondos de muchos individuos particulares. El crowdfunding permite el acceso a capital, a personas y organizaciones que están excluidas de los mecanismos de financiamiento tradicionales, o acceder a ellos a un menor costo.

Existen dos tipos de plataformas de crowdfunding: no financieras y financieras.

Crowdfunding no financiero: tienen por objeto apoyar compañías o causas altruistas y sociales (como para la elaboración de una película, el desarrollo de un nuevo producto o ayuda para construir viviendas), sin ofrecer ningún rendimiento o beneficio económico a las personas que contribuyen con el financiamiento. Algunas de estas, ofrecen recompensas o regalos simbólicos.

Crowdfunding financiero: las personas u organizaciones que solicitan financiamiento prometen a cambio algún rendimiento o beneficio económico, por lo que, desde el punto de vista de los que aportan los fondos, es un esquema de inversión.

2.3.2 Préstamos

Son plataformas en internet que otorgan préstamos a individuos y utilizan nuevas fuentes de información como redes sociales y sistemas de reputación (los usuarios se califican entre sí generando un puntaje o “reputación” que pretende medir la confiabilidad de cada persona), **junto con tecnologías de análisis de datos e inteligencia artificial**, para evaluar los riesgos crediticios. Estas plataformas permiten el acceso a créditos a nuevos segmentos de la población, por ejemplo, a las personas que no tienen un historial crediticio.

A diferencia del crowdfunding de deuda, es la propia empresa quien presta los fondos y no la comunidad.

2.3.3 Industria FinTech en México

Existen actualmente 238 empresas FinTech operando en México, con un crecimiento del 50% entre 2016 y 2017, y más de 540 mil usuarios activos. La Asociación FinTech México agrupa a cincuenta empresas FinTech, la Asociación de Plataformas de Fondeo Colectivo a más de veinte empresas dedicadas al crowdfunding y préstamos y la Asociación de Agregadores de Medios de Pago a más de treinta compañías de pagos. La Asociación Mexicana de Sociedades Financieras Populares agrupa a 24 empresas dedicadas al sector de ahorro y crédito popular.

2.3.4 Regulación

El Reino Unido y Singapur son países líderes en el sector financiero que están posicionando a su industria FinTech entre las más desarrolladas del mundo. Ambos gobiernos están favoreciendo el ecosistema FinTech con algunas prácticas regulatorias innovadoras como:

Ofrecer a empresas FinTech acceso a asesorías y retroalimentación constante de parte del regulador.

Sandbox regulatorio: Permite a nuevas empresas, bancos o instituciones financieras probar nuevos modelos de negocios FinTech por un periodo de tiempo. Es un permiso para operar en un marco regulatorio más flexible, dentro de ciertos límites en cuanto a número de clientes y cantidad de fondos que pueden manejar y bajo seguimiento por parte de la autoridad. De esta manera, la empresa puede comenzar a operar evitando los costos regulatorios.42

Plataformas regulatorias de código abierto (open-source): Permiten a las empresas compartir datos y herramientas de análisis en tiempo real con los reguladores. Este tipo de plataformas permiten la implementación de tecnologías regulatorias (RegTech) que utilizan sistemas de reportes, análisis de datos y riesgos, monitoreo de actividad y visualizaciones para facilitar a las compañías de servicios financieros tradicionales y empresas FinTech el cumplimiento de la norma. En México, hasta la fecha no existe una ley específica que regule a la industria FinTech, por lo que muchas de las empresas del sector han operado en áreas grises de las leyes existentes. Si bien esta situación ha permitido que se desarrolle un ecosistema de empresas FinTech, dentro de las cuales algunas han sido sumamente exitosas, la incertidumbre regulatoria les dificulta recaudar capital de inversión para desarrollarse.

Dado el vacío regulatorio en el que operan las empresas FinTech en México, éstas no cumplen con los estándares de riesgo de los bancos y servicios financieros tradicionales. (Ocampo, 2017)

2.4 Inteligencia artificial

Cuando se pretende establecer la relación entre la matemática y la IA aparecen los algoritmos como un puente entre ellos. Un algoritmo es una secuencia de instrucciones que representan un modelo de solución para cierto tipo de problemas. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

Una vez diseñado el algoritmo se realiza un programa cuyo desarrollo requiere un conocimiento de las técnicas de programación. El programador experto, Luis Joyanes, dice que los algoritmos son más importantes que los lenguajes de programación o las computadoras, ya que, un lenguaje de programación es sólo un medio para expresar un algoritmo y una computadora es solo un procesador para ejecutarlo.

La IA es una combinación de algoritmos planteados de manera que confieren a las máquinas capacidades como: aprender a partir de un conocimiento previo y/o conocimiento adquirido de la experiencia, ajustarse a nuevas aportaciones de datos y realizar tareas de una manera semejante a la forma en que las realizaría el ser humano.

En resumen, la IA se construye mediante algoritmos (con capacidades matemáticas de aprendizaje) y los datos para entrenarlos.

2.4.1 El aprendizaje automático (machine learning)

El machine learning o aprendizaje automático es una disciplina que pertenece al ámbito de la IA, cuyo objetivo es crear máquinas o sistemas que “aprenden automáticamente”. Aprender en este contexto significa identificar patrones complejos entre millones de datos.

El sistema que realmente aprende es un algoritmo que recibe los datos, los revisa, los procesa y es capaz de predecir comportamientos futuros, reconocer imágenes, etc. Aprender automáticamente significa que estos sistemas mejoran de forma autónoma con el tiempo, sin intervención humana.

Machine Learning resuelve situaciones por sí solo a partir de un análisis de datos y cuantos más datos tengan mejores resultados, además, para realizar el análisis se utilizan algoritmos que diseñan otros datos según las necesidades. A través de los datos de entrada, Machine Learning ejecuta un algoritmo y como

resultado, genera más información para el problema. El objetivo de generar más datos se basa en las siguientes técnicas:

- Regresión lineal y polinómica.
- Árboles de decisión.
- Redes neuronales.
- Red bayesiana.
- Cadenas de Markov.

Estas técnicas permiten a Machine Learning reconocer patrones, extraer conocimiento, descubrir información y hacer predicciones. Se considera que cada persona aprende de una manera particular, utiliza los sentidos, la experiencia y sus habilidades cognitivas, también puede confiar en estrategias personales y técnicas de aprendizaje, por ejemplo, tomar notas, resolver ejercicios, leer, memorizar, marcar libros. En el entorno informático, se pretende lograr que las computadoras alcancen la autonomía y de esta forma aprendan de forma automática sus propias habilidades que se definen con los algoritmos para el aprendizaje y la gestión de datos.

El Machine Learning no es auto programación, sino autoaprendizaje de datos y experiencia para generar patrones y resolver nuevas tareas. Este aprendizaje es la combinación de técnicas, datos, conceptualización de análisis de datos y algoritmos para generar nuevos patrones o modelos de predicción. (Fuentes, 2022)

2.4.1.1 Aplicaciones comunes de Machine Learning

Las aplicaciones del ML se pueden identificar en aplicaciones de uso cotidiano como Twitter o Facebook, por ejemplo, cuando agrega un amigo o una persona a través de las famosas recomendaciones, o incluso en YouTube es muy común viajar a través de los videos recomendados, muchos del usuario de estas herramientas y otras, han experimentado estas recomendaciones.

Para tener una mayor claridad sobre el Machine Learning, a continuación, se describen algunas áreas en las que este tipo de tecnología se usa comúnmente:

- Detección de correos electrónicos no deseados (spam): mediante la detección de texto en correos electrónicos recibidos, los algoritmos de Machine Learning los clasifican como no deseados.

- Detección de patrones en imágenes: esta función se encuentra en la cámara fotográfica en la que se detectan las sonrisas de las personas que se identifican y una vez que están sonriendo, se realiza la toma automáticamente. El desafío aquí es identificar que la persona está sonriendo dadas todas las características faciales que una persona puede tener .

- Otra Aplicación muy utilizada es AMAZON, la cual da recomendaciones de productos de acuerdo a los patrones de compra que identifica, es fascinante esta aplicación dentro del mundo del aprendizaje automático.

2.4.1.2 Tipos de aprendizaje de Machine Learning

Para comprender esta actividad, es importante conocer la clasificación de los tipos de algoritmos de aprendizaje de Machine Learning que se describirán a continuación:

Aprendizaje supervisado: se enseña al algoritmo cómo realizar su trabajo, con un conjunto de datos clasificados bajo una cierta apreciación o idea para encontrar patrones que puedan aplicarse en un análisis y producir una salida que ya se conoce.

El aprendizaje supervisado es la categoría más usada en el Machine Learning, donde se cuenta con ejemplos de pares entrada-salida. Por tal motivo, es implementado cuando se requiere hacer predicciones precisas a partir de nuevas entradas no vistas por el modelo. El desarrollo de técnicas a través de esta corriente opera a través del training data para realizar el ajuste del modelo, el cual contiene unos atributos o características bien definidas para realizar la predicción en etiquetas.

Es posible comprender el aprendizaje supervisado intuitivamente, con el siguiente ejemplo: un docente supervisa el proceso de formación a través de unos objetivos definidos al inicio del periodo académico (etiquetas), si el alumno no logra estas metas, el profesor realiza tutorías frecuentes (iteraciones en el desarrollo del modelo) hasta lograr el resultado esperado (predicciones precisas)

Se pueden presentar dos tipos de problemas en aprendizaje supervisado denominados clasificación y regresión *(Ver imagen 1)*.

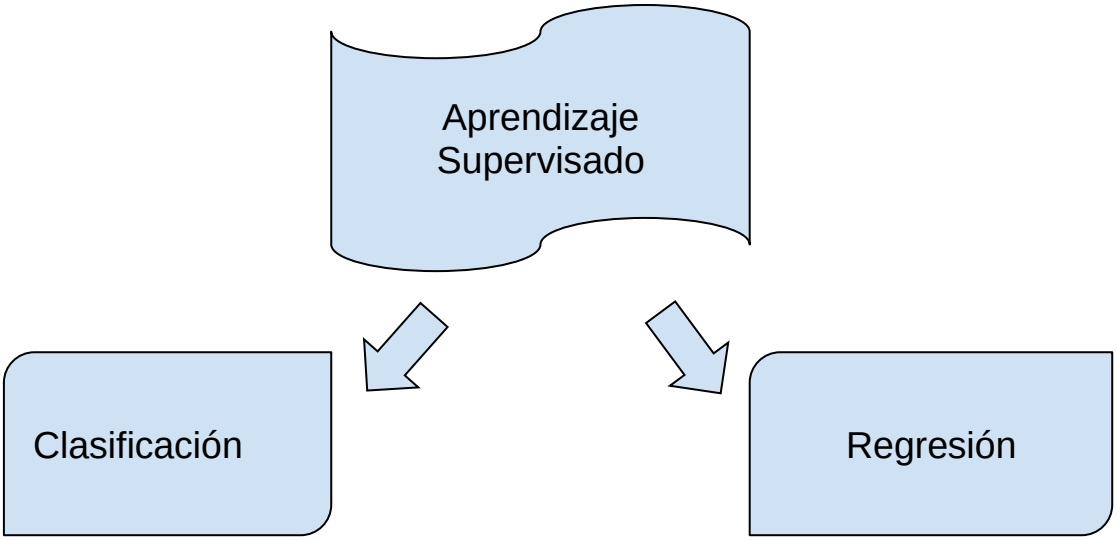


Imagen 1. Diagrama de Aprendizaje Supervisado (fuente propia).

Ante un problema de clasificación, la meta consiste en predecir una etiqueta de clases definidas en el entrenamiento, un ejemplo claro lo constituye la predicción de correos en las etiquetas spam o no spam, según parámetros de entrada definidos. Además, se cuenta con dos tipos de clasificación: binaria y multiclase. Como binaria se tiene la clasificación de correos mencionada, por su parte en el multiclase se cuenta con más de una clase para realizar la predicción.

Ahora bien, en los problemas de regresión se busca predecir un número continuo (punto flotante en programación). Como casos prácticos se tiene la predicción de ingresos anuales por núcleo familiar, según el nivel educativo de sus intrigantes, edad y experiencia laboral o la producción agropecuaria cuyas características pueden ser el estimado de rendimientos históricos, condiciones climáticas y colaboradores, es claro que el valor previsto en estos casos es un número dentro de un intervalo definido.

Gracias a lo expuesto, es posible concluir que el caso que ocupa el presente trabajo deber ser abordado como un problema de clasificación multiclase al contar con las clases bien definidas de tipos de planeta, a continuación, se exponen las técnicas de aprendizaje supervisado a ser modelados.

Aprendizaje no supervisado: se define como un modelo predictivo entrenado de manera similar al aprendizaje supervisado, pero la diferencia es que la comprensión se da en datos no clasificados o etiquetados y descubre patrones de ejemplos similares entre grupos de datos.

Aprendizaje reforzado: es un tipo de aprendizaje automático en el que no hay capacitación con datos clasificados o no clasificados; el sistema aprende en un entorno donde no hay información sobre la posible salida, a través de acciones y los resultados obtenidos, además, el modelo se refuerza al resolver el problema de la mejor manera. (Solórzano Guerrero, 2021)

2.4.2 Big Data

El Big Data es un término que describe la ingente cantidad de datos, tanto estructurados como no estructurados, que pueden obtenerse de los equipos o procesos cada día. El gran volumen de datos, su variabilidad, velocidad, veracidad y valor provocan problemas para extraer datos reales y de alta calidad; en ese sentido, los softwares tradicionales no son aparentes para estas exigencias.

El Big Data puede generar un gran volumen de datos, sin embargo, uno de los aspectos más importantes es la calidad de los mismos para tomar decisiones que permitan obtener ventajas comparativas; en efecto, si los datos no son de buena calidad se puede incurrir en graves errores.

Tareas importantes en esta disciplina son:

- La recopilación y almacenamiento de datos que se debe automatizar.
- El algoritmo que extraiga la información de utilidad lo cual implica un recurso humano, el científico de datos (data scientist) cuyo trabajo es extraer conocimiento a partir de los datos para tomar las mejores decisiones.

Algunas aplicaciones del Big Data pueden ser: por ejemplo: prever crisis económicas, prevenir enfermedades, prever fallas de equipos, trazar la ruta idónea de transporte público o acertar al abrir un local en determinada zona.

2.4.2.1 Benefcios

En este contexto, es claro que las oportunidades que genera el big data son enormes, y estas oportunidades, son ya hoy en día, en muchos casos, un beneficio tangible.

El universo digital es un área empresarial absolutamente en alza, que tendrá un enorme valor en el futuro. Algunos de los beneficios más relevantes del big data son poder ofrecer una visión cada vez más precisa de las fluctuaciones y rendimientos de todo tipo de recursos, permitir realizar adaptaciones experimentales a cualquier escala de un proceso y conocer su impacto en tiempo casi real, ayudar a conocer mejor la demanda y así realizar una segmentación mucho más ajustada de la oferta para cada bien o servicio, o acelerar la innovación y la prestación de servicios cada vez más innovadores y más eficientes

Las grandes empresas supieron ver el valor potencial de las técnicas del big data y la minería de datos hace años, y así Axciom, Google, IBM o Facebook llevan años invirtiendo en descubrir nuevos usos de los datos, cómo tratarlos y cómo transformarlos en valor.

Siguiendo a los grandes pioneros, en la mayor parte de los sectores, tanto compañías maduras como nuevas entrantes están poniendo en marcha estrategias para innovar y capturar valor. Por ejemplo, en el sector sanitario, algunas empresas pioneras están analizando los resultados que determinados medicamentos ampliamente prescritos tienen sobre la salud, y están descubriendo beneficios y riesgos que no fueron descubiertos durante los ensayos clínicos. Otras empresas están recolectando datos provenientes de sensores integrados en productos tales como juguetes para niños o bienes industriales, para determinar cómo se están utilizando estos productos en la práctica. Con este nuevo conocimiento, las empresas son capaces de generar nuevos servicios y diseñar productos futuros. De este modo, el análisis de datos se convierte en una importante ventaja competitiva para las empresas.

2.4.2.2 Riesgos

En efecto, el big data debe hacer frente a determinados retos o limitaciones. En concreto, algunos de los retos más importantes (dejando de lado las dificultades técnicas de almacenamiento o investigación computacional) son:

1) El riesgo de caer en conclusiones erróneas que nadie revisa.

2) El riesgo que para las personas pueda tener tomar decisiones, automatizadas sin un sesgo humano.

3) El riesgo para la privacidad de las personas.

2.4.2.2 Datos de carácter personal

Se entiende por dato de carácter personal "cualquier información concerniente a personas físicas identificadas o identificables". Una persona es identificable cuando su identidad pueda determinarse, directa o indirectamente, mediante cualquier información referida a su identidad física, fisiológica, psíquica, económica, cultural o social, salvo que dicha identificación requiera actividades o plazos desproporcionados.

Los datos de carácter personal no se limitan únicamente a nombres y apellidos, sino que son una lista amplia y abierta, que va creciendo, y que incluye datos como nuestra voz, número de la Seguridad Social, nuestra dirección o datos económicos. Pero también son datos de carácter personal nuestros "likes" en Facebook, nuestro ADN o nuestra forma de caminar. Ni siquiera nosotros mismos somos conscientes de las formas en las que nuestro propio día a día nos hace identificables

Así, por ejemplo, aunque no nos hayamos registrado en un sitio web, éste puede utilizar técnicas analíticas para rastrear las huellas digitales que nuestras actividades han ido dejando hasta terminar identificándonos.

Mención señalada merecen los llamados datos especialmente protegidos, puesto que son aquellos datos que, de divulgarse de manera indebida, podrían afectar a la esfera más íntima del ser humano, tales como ideología, afiliación sindical, religión, creencias, origen racial o étnico, salud y orientación sexual. Estos datos requieren un nivel de protección mayor y la Ley les reserva un tratamiento especial. (Gil, 2016)

2.4.3 Sistemas expertos

Los sistemas expertos se definen en forma general como los sistemas de computación (incluyen hardware y software) que recopilan y simulan el pensamiento de expertos humanos en un área específica del conocimiento. Estos sistemas son capaces de procesar y memorizar información, aprender y razonar en situaciones determinísticas e inciertas, comunicarse con humanos y/o sistemas expertos, tomar decisiones apropiadas y explicar por qué estas decisiones han sido tomadas. De esta forma, los sistemas expertos actúan como un consultor que puede proporcionar ayuda a un experto humano con un grado razonable de credibilidad.

Existen dos enfoques para la construcción de los sistemas expertos:

- El primer enfoque permite la introducción del conocimiento acumulado de expertos humanos a lo largo de su vida profesional, obteniéndose de esta forma lo que se conoce como sistema experto. El principal problema de este enfoque se relaciona con el proceso de captación de la información, la cual se ha de hacer mediante entrevista al experto en un dominio específico o mediante la observación de su comportamiento a través de un análisis de protocolo. Esto trae bloqueos o cuellos de botellas en el desarrollo de la aplicación.

- El segundo enfoque busca la elaboración de programas de ordenador capaces de generar conocimiento a través del análisis de los datos empíricos y posteriormente se usa ese conocimiento para hacer inferencias sobre nuevos datos. Como resultado de este enfoque surgen procedimientos conocidos como Machine Learning (Aprendizaje Automático) o Data Mining (explotación de datos), los cuales permiten transformar una base de datos en base de conocimiento (más adelante en este artículo se menciona con más detalle este concepto). Este segundo enfoque es uno de los más utilizados para el diseño de sistemas expertos.

Los sistemas expertos se constituyen en la herramienta de la Inteligencia Artificial más utilizada desde sus inicios y, como se dijo anteriormente,

corresponden a programas de ordenador que recopilan en un programa informático el conocimiento de especialistas en una materia. Existen distintos tipos de sistemas expertos, teniendo en cuenta la forma como los sistemas expertos representan el conocimiento incluido en ellos, y los sistemas expertos basados en reglas son los más comúnmente utilizados en el ámbito financiero. Los componentes principales de un sistema experto basado en reglas son los siguientes:

- Base de conocimiento: Contiene el conocimiento y las experiencias de los expertos en un determinado dominio representado por medio de símbolos. Dentro de ella se puede distinguir el conocimiento declarativo (hechos) y procedimental (reglas).

- Base de Datos, Memoria de trabajo o Modelo situacional: Es una memoria auxiliar que contiene la información relacionada con el problema que se va a resolver, es decir, los datos iniciales y los datos intermedio que corresponden al estado del sistema a lo largo del proceso.

- Motor de Inferencias o Estructura de control: Es la parte del sistema experto que se encarga de realizar los procesos de inferencia entre la información contenida en la base de datos o memoria de trabajo y la base de conocimiento, con el fin de obtener las conclusiones que sean necesarias.

- Interfaz de usuario o Subsistema de consulta: Es la parte del sistema experto que permite la comunicación entre el usuario y el motor de inferencias. Adicionalmente, permite introducir la información que necesita el sistema y comunica las respuestas del sistema experto al usuario.

- Modelo de justificación o Subsistema de explicación: Esta parte del sistema experto explica los pasos realizados por el motor de inferencias para llegar a las conclusiones esperadas, indica también por qué se utilizan ciertas reglas y no otras, y por qué se planteó determinada pregunta en el diálogo con el usuario.

- Subsistema de Adquisición del Conocimiento: Es una interfaz que facilita la introducción del conocimiento en la base de datos y de los mecanismos de inferencia. Esta parte del sistema experto también se encarga de comprobar la veracidad y coherencia de los hechos y reglas que se introducen en la base de conocimiento. (Sierra, 2007)

2.4.4 Implicaciones sociales

El desarrollo de tecnologías inteligentes impacta profundamente en la sociedad. En el sector productivo, las oficinas gerenciales incorporan métodos automáticos para la toma de decisiones. En la manufactura, usan robots con capacidades de desplazamiento y localización de objetos. En la agricultura, se desarrollan tecnologías para diagnosticar oportunamente enfermedades en cosechas, así como sistemas de vigilancia del suelo, utilizando sensores, imágenes satelitales y registros históricos para predecir la productividad de los plantíos. Debido a su alto costo, estas tecnologías sólo son accesibles para grandes empresas en la actualidad.

En el sector salud, sistemas inteligentes, sensores de bajo costo y ambientes virtuales están transformando la prevención, el diagnóstico y el tratamiento de enfermedades de alto riesgo como cáncer, obesidad, hipertensión y diabetes. Por ejemplo, en la prevención, algunas aplicaciones móviles permiten utilizar los sensores del teléfono celular para vigilar la cantidad de azúcar ingerida y el ritmo cardíaco del usuario. Gracias a su bajo costo, estas tecnologías pueden llevarse a zonas rurales, incrementando la cobertura y evitando gastos de traslado de pacientes. Por otro lado, el uso de ambientes virtuales (espacios intangibles creados por computadoras, accesibles a través de redes, como internet o equipos de realidad virtual), puede cambiar el tratamiento y la rehabilitación de padecimientos motrices o cognitivos que requieren ejercicios físicos o mentales.

En la educación, la IA será central en las escuelas del futuro, cambiando radicalmente el papel del profesor en el aula. Actualmente ya hay sistemas inteligentes capaces de dar asesoría personalizada a cada alumno en reportes y ensayos, lo cual permite a los profesores identificar áreas de oportunidad con mayor eficacia. Asimismo, también ha crecido el número de plataformas que ofrecen tutorías por internet para todos los grados educativos. Sin embargo, el INEGI reportó que en 2016 sólo el 59.5% de los mexicanos mayores de 6 años tenían acceso a Internet y que estos se encuentran distribuidos en el 47% de los hogares, lo que sugiere la necesidad de mejorar el acceso a este tipo de tecnologías.

Finalmente, en materia de seguridad, se pueden analizar eficientemente días enteros de grabación de cámaras de circuito cerrado, así como rastrear la ubicación de individuos. También se podrían emplear sistemas inteligentes en drones para detectar actividades criminales, aunque este uso es controvertido, ya que suscita preocupaciones sobre el control que el gobierno podría ejercer sobre la población y su privacidad.

2.4.5 Implicaciones económicas

La IA traerá cambios importantes en el ámbito laboral tanto nacional como internacional. Por un lado, los empleos requerirán de conocimientos en computación y análisis de datos y por otro, disminuirá la oferta de aquellos empleos que pueden ser automatizados, como la albañilería, manufactura, o las ventas por teléfono.

Varias actividades se benefician del desarrollo de la IA, como la de los médicos, que cuentan con tecnología eficiente para obtener información sobre enfermedades y medicamentos usando lenguaje natural. Los estudiantes pueden enfocarse mejor en desarrollar habilidades analíticas y computacionales en vez de memorizar contenidos. La educación para desarrollar las áreas de inteligencia artificial desde temprana edad es fundamental, en particular para fortalecer las habilidades que tienen que ver con el análisis de datos, abstracción, desarrollo de algoritmos y solución de problemas. Estas habilidades se conocen como pensamiento computacional y en la época actual, son tan importantes como las habilidades matemáticas o de comunicación. Es por ello que países como Estados Unidos, Reino Unido y Finlandia fomentan estas habilidades desde la educación básica y media.

Con este cambio de paradigma, la competitividad internacional favorecerá a egresados con habilidades de razonamiento computacional, abstracción de conceptos y capaces de trabajar en ambientes multidisciplinarios. (Ocampo M. &., 2018)

2.4.6 Lenguajes de programación

Existen varios lenguajes de programación que se utilizan para construir aplicaciones de Inteligencia artificial y Machine Learning. Cada aplicación tiene sus propios requisitos y restricciones, y algunos lenguajes podrían ser mejores que otros según los problemas específicos. Los lenguajes de programación han evolucionado, y otros se han creado en función de los requisitos únicos de las aplicaciones de Inteligencia artificial.

La creación de un modelo de Machine Learning no se limita a usar un algoritmo de aprendizaje o una biblioteca de Machine Learning; es un proceso que generalmente implica al menos 6 pasos:

1. Recolectar los datos: Los datos se pueden recolectar de fuentes tal como un sitio web, utilizando una API o una base de datos. Este paso es uno de los más complicados y requiere un tiempo determinado.

2. Preprocesamiento de los datos: Con los datos disponibles, se debe asegurar que todos tengan un formato correcto para alimentar el algoritmo de aprendizaje. Por lo general se tiene que realizar varias tareas de preprocesamiento antes de poder usar los datos.

3. Explore los datos: Se realiza un análisis previo para corregir los casos de valores faltantes o tratar de encontrar a primera vista cualquier patrón en ellos que facilite la construcción del modelo. En este punto, se deben detectar valores atípicos; o encuentre las características que tienen más influencia para hacer una predicción.

4. Entrena el algoritmo: los algoritmos de aprendizaje se alimentan con los datos que se procesaron en las etapas anteriores. La idea es que los algoritmos pueden extraer información útil de los datos iniciales y luego hacer las predicciones.

5. Evaluar el algoritmo. Se realizan las pruebas de la información que genera el conocimiento del entrenamiento previo que se obtuvo a través del algoritmo.

6. Uso del modelo.

2.4.6.1 R

R es un lenguaje de programación especialmente orientado al análisis estadístico y a la representación gráfica de los resultados obtenidos. Es un proyecto de GNU por lo que los usuarios son libres de modificarlo y extenderlo. R se distribuye como software libre bajo la licencia GNU y es multiplataforma (hay versiones para plataformas Windows, Mac y Linux, y de hecho algunas distribuciones de Linux lo tienen incorporado), lo que también ha facilitado su adopción y la existencia de Una comunidad muy activa su entorno, con constantes desarrollos de nuevas funcionalidades y versiones mejoradas de las existentes. Es un lenguaje basado en comandos, en lugar de hacer clic y arrastrar iconos o menús con el mouse, escribe comandos o instrucciones que se ejecutan. Una secuencia de instrucciones o comandos R que implementa un flujo de trabajo para realizar una tarea se denomina script o script R.

R tiene algunas características especiales que lo hacen versátil para el manejo de elementos estadísticos, específicamente para operaciones con matrices y vectores, lo que facilita la manipulación de bases de datos. Por lo tanto, R le permite manipular datos muy rápidamente. En cuanto al aprendizaje automático, R ha implementado una gran cantidad de algoritmos, como consecuencia de las diferentes líneas de investigación de grupos que llevaron a su creación, debido precisamente al hecho de que R nació en el campo académico. Frente a R, es su curva de aprendizaje, que generalmente es más lenta y complicada en comparación con Python.

2.4.6.2 PHYTON

Python es un lenguaje donde su código se ejecuta en el navegador al cargar la página, es independiente de la plataforma y orientado a objetos, está listo para realizar cualquier tipo de programa desde aplicaciones de Windows hasta servidores de red o incluso páginas web. Es un lenguaje interpretado, lo que ofrece ventajas como la velocidad de desarrollo e inconvenientes como una velocidad más baja al ser ejecutado. En los últimos años, este lenguaje se ha vuelto muy popular y algunas de las razones son las siguientes:

- El número de bibliotecas que contiene, los tipos de datos y las funciones incorporadas en el lenguaje.
- Python es gratis, importante: incluso para fines comerciales.
- La simplicidad y velocidad con la que se crean los programas. Un programa en Python tiene menos líneas de código que su equivalente en Java o C.
- • El número de plataformas en las que se puede desarrollar, como Unix, Windows, OS/2, Mac y otras.

Características de Python para el desarrollo de Machine Learning:

1. La asociación Python de ML: se ha visto favorecida por aplicaciones que van desde el desarrollo web hasta la automatización de scripts y procesos

2. Amplia selección de bibliotecas y marcos: Uno de los aspectos que hace que Python sea una opción tan popular en general es su abundancia de bibliotecas y macros que facilitan la codificación y ahorran tiempo en el desarrollo.

3. Código legible y conciso: facilidad de uso y simplicidad, especialmente para los nuevos desarrolladores. El aprendizaje profundo se basa en algoritmos extremadamente complejos y flujos de trabajo de múltiples etapas.

4. Agilidad: La sintaxis simple de Python significa que también es más rápido en desarrollo que muchos lenguajes de programación y permite al desarrollador probar algoritmos rápidamente sin tener que implementarlos.

5. Colaboración: fácil de leer es de gran valor para la codificación cooperativa, o cuando los proyectos de Python de Deep Learning o ML cambian de manos entre los equipos de desarrollo.

6. Python es un lenguaje de programación de código abierto y está respaldado por una gran cantidad de recursos y documentación de alta calidad. (Rojas, 2020)

2.5 Shell Scripting

El intérprete de mandatos o "shell" es la interfaz principal entre el usuario y el sistema, permitiéndole a aquél interactuar con los recursos de éste. El usuario introduce sus órdenes, el intérprete las procesa y genera la salida correspondiente.

Por lo tanto, un intérprete de mandatos de Unix es tanto una interfaz de ejecución de órdenes y utilidades, como un lenguaje de programación, que admite crear nuevas órdenes denominadas guiones o “shellscripts”, utilizando combinaciones de mandatos y estructuras lógicas de control, que cuentan con características similares a las del sistema y que permiten que los usuarios y grupos de la máquina cuenten con un entorno personalizado.

2.5.1 Características

Las principales características del intérprete GNU BASH son:

- Ejecución síncrona de órdenes (una tras otra) o asíncrona (en paralelo).
- Distintos tipos de redirecciones de entradas y salidas para el control y filtrado de la información.
- Control del entorno de los procesos.
- Ejecución de mandatos interactiva y desatendida, aceptando entradas desde teclado o desde ficheros.
- Proporciona una serie de órdenes internas para la manipulación directa del intérprete y su entorno de operación.
- Un lenguaje de programación de alto nivel, que incluye distintos tipos de variables, operadores, matrices, estructuras de control de flujo, entrecomillado, sustitución de valores y funciones.
- Control de trabajos en primer y segundo plano.
- Edición del histórico de mandatos ejecutados.
- Posibilidad de usar una "shell" para el uso de un entorno controlado.

2.5.2 Cuándo utilizar el intérprete de mandatos.

Una “shell” de Unix puede utilizarse como interfaz para ejecutar órdenes en la línea de comandos o como intérprete de un lenguaje de programación para la administración del sistema.

El lenguaje de BASH incluye una sintaxis algo engorrosa, pero relativamente fácil de aprender, con una serie de órdenes internas que funcionan de forma similar a la línea de comandos. Un programa o guion puede dividirse en secciones cortas, cómodas de depurar, permitiendo realizar prototipos de aplicaciones más complejas.

Sin embargo, hay ciertas tareas que deben ser resueltas con otros intérpretes más complejos o con lenguajes compilados de alto nivel, tales como:

- Procesos a tiempo real, o donde la velocidad es un factor fundamental.
- Operaciones matemáticas de alta precisión, de lógica difusa o de números complejos.
- Portabilidad de código entre distintas plataformas.
- Aplicaciones complejas que necesiten programación estructurada o proceso multihilo.
- Aplicaciones críticas para el funcionamiento del sistema.
- Situaciones donde debe garantizarse la seguridad e integridad del sistema, para protegerlo contra intrusión o vandalismo.
- Proyectos formados por componentes con dependencias de bloqueos. · Proceso intensivo de ficheros, que requieran accesos directos o indexados.
- Uso de matrices multidimensionales o estructuras de datos (listas, colas, pilas, etc.).
- Proceso de gráficos.
- Manipulación de dispositivos, puertos o “sockets”.
- Uso de bibliotecas de programación o de código propietario. (Labrador, 2014)

2.6 La Nube

2.6.1 Antecedentes

El desarrollo de la computación en la nube comenzó a través de grandes empresas de servicios de Internet como Google y Amazon los cuales construyeron su propia infraestructura. A partir de allí surgió una arquitectura: un sistema de recursos distribuidos de manera horizontal, introducidos como servicios virtuales de tecnologías de información (TI) escalados masivamente y manejados como recursos agrupados y configurados continuamente.

El modelo de esta arquitectura tiene como base a “Las granjas de servidores” , estas eran similares en su arquitectura al procesamiento en red (*grid*), sin embargo, mientras que las redes se utilizan para aplicaciones de procesamiento técnico con un acoplamiento más bien débil (consistentes en un sistema compuesto de subsistemas con cierta autonomía de acción que mantienen una interrelación continua entre ellos formando una “supercomputadora virtual” para realizar grandes tareas), la nube orientó sus aplicaciones a los servicios de Internet.

Aunque la implementación es reciente, la idea no es nueva ya que se ha discutido en el medio desde hace algunos años con distintos nombres tales como: “utility computing”, computación en demanda, computación elástica, o “grid computing” (no confundir con el procesamiento en red mencionado anteriormente).

Haciendo una comparación de ideas y tecnologías entre las décadas de 1960 y 1970 con la época actual se tiene lo siguiente:

Década de 1960

- Uso de “terminales tontas” que dependían de un sistema central más potente.
- La información se guardaba en el servidor.
- Se necesitaba conexión constante con el sistema central para funcionar correctamente.
- Imposibilidad de instalar aplicaciones

Siglo XXI

- Terminales poco potentes pero autosuficientes (Netbooks, tablets, smartphones)

- La información se aloja en los servidores del proveedor de servicios, aunque hay posibilidad de guardar información en la terminal del usuario.

- Se necesita conexión constante con el sistema central para hacer uso de todos los recursos.

- Dependiendo de la terminal, es posible instalar aplicaciones, aunque la idea es ejecutarlas a través de Internet.

2.6.2 Características

No es necesario disponer de un equipo potente, tan solo de un aparato con conexión a internet; esto debido a que el dispositivo del usuario no realizará ningún proceso complejo y los ficheros pueden guardarse en la nube. Los servidores en donde se hallan los programas que se utilicen son los encargados de las tareas complicadas que antes se realizaban localmente. Con el uso del Cloud Computing no hay necesidad por parte del usuario de conocer la infraestructura detrás de esta, ya que pasa a ser una abstracción, “una nube” donde las aplicaciones y servicios pueden fácilmente crecer, funcionar rápido y con pocas fallas. Este tipo de servicio se puede pagar según alguna métrica de consumo, no por el equipo usado en sí, sino por uso de CPU/hora como en el caso de Amazon EC2.

Entre otras características podemos mencionar:

- Es auto reparable: En caso de surgir un fallo, el último respaldo (backup) de la aplicación se convierte automáticamente en la copia primaria y a partir de esta se genera uno nuevo.

- Es escalable: Todo el sistema y su arquitectura es predecible y eficiente. Si un servidor maneja 1000 transacciones, 2000 transacciones serán manejadas por 2 servidores. Se establece un nivel de servicios que crea nuevas instancias de acuerdo a la demanda de operaciones existente de tal forma que se reduzca el tiempo de espera y los cuellos de botella

- Virtualización: las aplicaciones son independientes del hardware en el que corran, incluso varias aplicaciones pueden correr en una misma máquina o una aplicación puede usar varias máquinas a la vez. El usuario es libre de usar la plataforma que desee en su terminal (Windows, Unix, Mac, etc.), al utilizar las aplicaciones existentes en la nube puede estar seguro de que su trabajo conservará sus características bajo otra plataforma.

- Posee un alto nivel de seguridad: El sistema está creado de tal forma que permite a diferentes clientes compartir la infraestructura sin preocuparse de ello y sin comprometer su seguridad y privacidad; de esto se ocupa el sistema proveedor que se encarga de cifrar los datos.

- Disponibilidad de la información: No se hace necesario guardar los documentos editados por el usuario en su computadora o en medios físicos propios ya que la información radicará en Internet permitiendo su acceso desde cualquier dispositivo conectado a la red (con autorización requerida) (Mejıa, 2011)

2.6.2 Arquitectura serverless

Serverless es un tipo de arquitectura donde los servidores (físicos o en la nube) dejan de existir para el desarrollador y en cambio el código corre en “ambientes de ejecución” que administran proveedores como Amazon, Google, IBM, etc

Las funciones serverless son sencillas de usar cuando no se requiere guardar estado en memoria. Debido a que no se tiene control acerca de cuándo los ambientes de ejecución son creados o destruidos, no se puede asumir que, al guardar un dato en la memoria de la función, este se mantenga allí cuando la función sea nuevamente invocada.

Ventajas:

- Completamente Administrada
  - Sin aprovisionamiento
  - Cero Administración (a nivel de hardware)
  - Alta disponibilidad
- ▪ Productividad del Desarrollador
  - Enfoca en el código
  - Reduce el tiempo al mercado
  - Innova rápidamente
- Escalamiento Continuo
  - Automáticamente
  - Aumenta o disminuye

Desventajas:

- Si no se desarrolla con cuidado, su código puede terminar bastante acoplado al proveedor.

- Al ser un servicio tan reciente, los lenguajes que se pueden usar para implementar las funciones están limitados por lo que esté soportado por el proveedor.

- Desplegar y monitorear el comportamiento de múltiples funciones es mucho más complicado que monitorear un monolito.

- Se requiere esfuerzo extra para poder desarrollar localmente sin necesidad de desplegar el código a los ambientes de ejecución cada que se realice un cambio, ya que puede ser demorado y tedioso.

- Las herramientas alrededor de la automatización del despliegue de funciones serverless son aún muy inmaduras. (Moreno, 2020)

2.7 Redes neuronales

2.7.1 Conceptos

Un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades medidas y las relaciones entre ellas (vector de características). Por ejemplo, un patrón puede ser una señal sonora y su vector de características el conjunto de coeficientes espectrales extraídos de ella (espectrograma). Otro ejemplo podría ser una imagen de una cara humana de las cuales se extrae el vector de características formado por un conjunto de valores numéricos calculados a partir de la misma. El reconocimiento automático, descripción, clasificación y agrupamiento de patrones son actividades importantes en una gran variedad de disciplinas científicas, como biología, psicología, medicina, visión por computador, inteligencia artificial, teledetección, etc.

2.7.2 Reconocimiento Estadístico de Patrones (REP)

El REP es una disciplina relativamente madura hasta el punto de que existen ya en el mercado un cierto número de sistemas comerciales de reconocimiento de patrones que emplean esta técnica. En REP, un patrón se representa por un vector numérico de dimensión n. De esta forma, un patrón es un punto en un espacio n-dimensional (de características). Un REP funciona en dos modos diferentes: entrenamiento y reconocimiento. En modo de entrenamiento, se diseña el extractor de características para representar los patrones de entrada y se entrena al clasificador con un conjunto de datos de entrenamiento de forma que el número de patrones mal identificados se minimice. En el modo de reconocimiento, el clasificador ya entrenado toma como entrada el vector de características de un patrón desconocido y lo asigna a una de las clases o categorías.

2.7.3 Redes Neuronales Artificiales.

La neurocomputación es una aportación más al viejo objetivo de crear sistemas inteligentes, considerando como tales a máquinas capaces de llevar a cabo tareas que exhiben alguna de las características asociadas a la inteligencia humana. En las dos últimas décadas, los avances en este campo han sido espectaculares; en particular el desarrollo de las redes neuronales artificiales (RNA. Originalmente, los trabajos en RNA surgen de la idea de que para que las máquinas puedan llevar a cabo dichas tareas inteligentes, sería conveniente que el modelo de computación se asemejara más a la fisiología del cerebro humano que al modelo computacional vigente por aquellas fechas: modelo von Neumann. Sin embargo, el auge de estos sistemas se debe más al éxito obtenido en aplicaciones reales (reconocimiento de patrones, predicción, optimización, etc.) que a la semejanza con el modelo biológico. Por ejemplo, el perceptrón multicapa, que es una de las redes más utilizadas, es criticada por su escaso parecido con el funcionamiento de las neuronas dentro del cerebro humano especialmente en todo lo referente a su algoritmo de aprendizaje.

En cualquier caso, lo que se plantea es un modelo computacional alternativo a la máquina von Neumann o a los ordenadores paralelos actuales, los cuales no están dotados de forma global de las siguientes características:

- Masivamente paralelos
- Computación y representación distribuida
- Aprendizaje
- Generalización
- Adaptabilidad
- Procesamiento de información inherente al contexto
- Tolerante a fallos
- Bajo consumo de energía

(Alonso Romero, 2001)

2.8 Crontabs

Cron es el nombre del programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts (grupos de comandos) a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado para ejecutar cualquier cosa. Como se define en las páginas del manual de cron (#> man cron) es un demonio que ejecuta programas agendados.

En prácticamente todas las distribuciones de Linux se usa la versión Vixie Cron, por la persona que la desarrolló, que es Paul Vixie, uno de los grandes gurús de Unix, también creador, entre otros sistemas, de BIND que es uno de los servidores DNS más populares del mundo.

Cron es un demonio (servicio), lo que significa que solo requiere ser iniciado una vez, generalmente con el mismo arranque del sistema. El servicio de cron se llama crond. En la mayoría de las distribuciones el servicio se instala automáticamente y queda iniciado desde el arranque del sistema, se puede comprobar de varias maneras:

#> /etc/rc.d/init.d/crond status
#> /etc/init.d/crond status Usa cualquiera de los dos dependiendo de tu distro crond (pid 507) is running...

o si tienes el comando service instalado:
#> service crond status crond (pid 507) is running...

se puede también revisar a través del comando ps:
ps -ef | grep crond

si por alguna razón, cron no está funcionando:
#> /etc/rc.d/init.d/crond start
Starting crond: [OK]

Si el servicio no estuviera configurado para arrancar desde un principio, bastaría con agregarlo con el comando chkconfig:

#> chkconfig --level 35 crond on

Hay al menos dos maneras distintas de usar cron:

La primera es en el directorio /etc, donde muy seguramente encontrarás los siguientes directorios:

- cron.hourly
- cron.daily
- cron.weekly
- cron.monthly

Si se coloca un archivo tipo script en cualquiera de estos directorios, entonces el script se ejecutará cada hora, cada día, cada semana o cada mes, dependiendo del directorio.

(Durán, 2012)

2.9 Captchas

Captcha son las siglas de Completely Automated Public Turing test to tell Computers and Humans Apart (prueba de Turing completamente automática y pública para diferenciar ordenadores de humanos). Este test es controlado por una máquina, en lugar de por un humano como en la prueba de Turing. Por ello, consiste en una prueba de Turing inversa. Consiste en que el usuario introduzca correctamente un conjunto de caracteres que se muestran en una imagen distorsionada que aparece en pantalla. Se supone que una máquina no es capaz de comprender e introducir la secuencia de forma correcta, por lo que solamente el humano podría hacerlo.

Esta herramienta permite identificar si quien hace uso de la página es un humano y no un robot, minimizando así los eventos de ataques informáticos con el fin de colapsar los servicios web, bloqueo de servidores, o inserción de basura en bases de datos.

Algunas ventajas de usar captcha en las páginas web con interacción de usuarios en sesiones no seguras son las siguientes:

- Evita que un sitio web sea dañado o vulnerado
- Evita el acceso a información privada
- Evita el spam en blog y foros.

Aunque no todo es beneficio, el captcha también tiene desventajas, como, por ejemplo:

- Puede ser molesto para usuarios principiantes o sin mucha experiencia en sistemas.
- En ocasiones los números, letras o imágenes que hay que resolver no son lo suficientemente claros para solucionar el captcha.

Hoy en día existen muchas opciones en el mercado, de pago y gratuitas que permiten agregar seguridad a las páginas web mediante este tipo de herramientas, donde dependiendo de la herramienta, se utiliza una estrategia diferente para identificar a quien usa el recurso, por ejemplo a través de preguntas, presentándole palabras distorsionadas al usuario que sería difícil a una máquina comprender, o preguntándole acerca de una imagen que la aplicación le presente al usuario o permitiéndole elegir entre diferente imágenes las que corresponden a una categoría.

(Muñoz Luna, 2017)

Capítulo III Metodología

3.1 Origen programación extrema (XP)

Nace de la mano de Kent Beck en el verano de 1996, cuando trabajaba para Chrysler Corporation. Él tenía varias ideas de metodologías para la realización de programas que eran cruciales para el buen desarrollo de cualquier sistema. Las ideas primordiales de sus sistemas las comunicó en las revistas C++ Magazine en una entrevista que ésta le hizo el año 1999.

3.2 ¿Qué es programación extrema o XP?

Es una Metodología ligera de desarrollo de aplicaciones que se basa en la simplicidad, la comunicación y la realimentación del código desarrollado.

La programación extrema se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ligeras, que son aquéllas en la que se da prioridad a las tareas que dan resultados directos y que reducen la burocracia que hay alrededor tanto como sea posible.

La programación extrema, dentro de las metodologías ágiles, se puede clasificar dentro de las evolutivas.

3.3 Objetivos de XP

- La Satisfacción del cliente.
- Potenciar el trabajo en grupo.
- Minimizar el riesgo actuando sobre las variables del proyecto: costo, tiempo, calidad, alcance.

3.4 Características

- Metodología basada en prueba y error para obtener un software que funcione realmente.
- Fundamentada en principios.
- Está orientada hacia quien produce y usa software (el cliente participa muy activamente).

- Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema.
- Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo.
- Cliente bien definido.
- Los requisitos pueden cambiar.
- Grupo pequeño y muy integrado (2-12 personas).
- Equipo con formación elevada y capacidad de aprender

3.5 Roles de la metodología XP

Programador: Es el Responsable de implementar las historias de usuario por el cliente. Además, estima el tiempo de desarrollo de cada historia de usuario para que el cliente pueda asignarle prioridad dentro de la iteración. Cada iteración incorpora nueva funcionalidad de acuerdo a las prioridades establecidas por el cliente. El Programador también es responsable de diseñar y ejecutar los test de unidad del código que ha implementado o modificado.

Cliente: Determina la funcionalidad que se pretende en cada iteración y define las prioridades de implementación según el valor de negocio que aporta cada historia. El Cliente también es responsable de diseñar y ejecutar los test de aceptación.

Encargado de pruebas (TESTER): Es el encargado de ejecutar las pruebas regularmente, difunde los resultados dentro del equipo y es también el responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (TRACKER): Una de las tareas más importante del tracker, consiste en seguir la evolución de las estimaciones realizadas por los programadores y compararlas con el tiempo real de desarrollo. De esta forma, puede brindar información estadística en lo que refiere a la calidad de las estimaciones para que puedan ser mejoradas.

Entrenador (COACH): Es responsable del proceso en general. Se encarga de iniciar y guiar a las personas del equipo en poner en marcha cada una de las prácticas de la metodología XP.

Consultor: Es un Miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (BIG BOSS): Es el vínculo entre el cliente y programadores. Experto en tecnología y labores de gestión. Construye el plantel del equipo, obtiene los recursos necesarios y maneja los problemas que se generan. Administra a su vez las reuniones (planes de iteración, agenda de compromisos, etc). Su labor fundamental es la coordinación.

Capítulo IV Desarrollo y aplicación

4.1 Análisis

El análisis realizado se realiza utilizando la metodología XP, por lo que haremos una herramienta que incluya todos los sistemas globalizados, desde el portal donde vamos a obtener los datos, el api que será consumido por un segundo portal y los sistemas de detección de robots *(Ver tabla 1)*.

Los resultados serán medidos de manera individual, ya que la metodología XP se basa en tener identificados los alcances en todos los sistemas involucrados.

Fases	**Descripción de cada fase**	**Herramientas**	**Productos o entregables**
Implementación de Infraestructura	Se montará y configura un servidor en la nube con AWS	AWS	Servidor Operativo
Creación sitio web 1	Sitio web con datos de persona	HTML, PHP, MYSQL, AWS	Sitio web con datos de persona
Creación de API	Mediante Scraping se van a extraer los datos del Sitio web 1, con métodos de consulta GET	PYTHON, PHP	API Funcional
Creación de sitio web 2	Sitio web para solicitud de crédito automatizada, en base a los datos extraídos con el API	HTML, PHP	Sitio web con respuesta a solicitud
Análisis de logs autónomo	Cada minuto se realizará un barrido autónomo para la creación de un archivo nuevo de logs, con indicadores	Python , shell scripting (linux), AWS	Archivo .log con el resumen de archivos.

	de ip en exceso.		
Programa de validación de logs	Se valida la información del archivo de logs creado para determinar si la página muestra un error, o si muestra la información solicitada.	PHP, AWS	Mensaje de error o imprimir información

Tabla 1. Descripción de etapas de desarrollo (fuente propia)

4.1.1 Implementación de Infraestructura

La arquitectura es uno de los elementos más importantes para el desarrollo de herramientas que trabajan en la nube, en este caso se usará como proveedor a Amazon Web Services, ya que cuenta con los siguientes servicios:

- Sistema Operativo Linux
- Acceso Root a la consola
- Sistemas escalables
- Reglas de Firewall de entrada y salida personalizadas
- Primer año gratis en instancias seleccionadas

Usaremos el servicio EC2 con un sistema operativo Ubuntu Server para la implementación de nuestra herramienta, ya que nos permitirá tener el control total del sistema operativo con el usuario root, dependemos de los logs del sistema para realizar análisis de datos de nuestro tráfico.

Para las bases de datos se usará MYSQL como gestor, debido a su práctico manejo y compatibilidad con nuestro sistema operativo.

Usaremos la herramienta Webmin para administrar nuestro servidor con una interfaz gráfica, esto ayudará al monitoreo de nuestro servidor y las herramientas implementadas en él.

4.1.2 Creación sitio web 1

El sitio web denominado “Sitio web 1” será la simulación de una empresa dueña de datos, en la que con llenar un formulario se puede visualizar la información de la persona a buscar, este formulario se conectará a una base de datos que es donde consultará dicha información.

El código del formulario se escribe en HTML *(Ver imagen 2)* en el que con un método GET enviará la información a un archivo nombrado “buscar.php” *(Ver imagen 3)* que es el que recibirá la información a consultar.

```
<html>
<head>
    <title>Busqueda de personas</title>
</head>

<body>
<h1>Empresa dueña de los datos</h1>
<br>
    <form action="buscar.php" method="GET">
Buscar Persona: <input type="text" name="buscar" id="buscar">
    <input type="submit">
    </form>
</body>
</html>
```

Imagen 2. Código HTML para el formulario (fuente propia).

```php
$buscar=$_GET['buscar'];

$servername = "localhost";
$username = "dev_tec";
$password = "xxxxxxxxxx";
$dbname = "TESCI";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM MUESTRAS where TELEFONO='$buscar'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
    echo "Nombre: <a class='Nombre'>" . $row["NOMBRE"]."</a><br>";
    echo "Apellido Paterno: <a class='paterno'>" . $row["APATERO"]."</a><br>";
    echo "Apellido Materno: <a class='materno'>" . $row["AMATERNO"]."</a><br>";
    echo "Calle: <a class='calle'>" . $row["CALLE"]."</a><br>";
    echo "Numero: <a class='numero'>" . $row["NUMERO"]."</a><br>";
    echo "Colonia: <a class='colonia'>" . $row["COLONIA"]."</a><br>";
    echo "Municipio: <a class='municipio'>" . $row["MUNICIPIO"]."</a><br>";
    echo "Teléfono: <a class='telefono'>" . $row["TELEFONO"]."</a><br>";
    echo "Empresa: <a class='empresa'>" . $row["EMPRESA"]."</a><br>";
    echo "SDI: <a class='sdi'>" . $row["SDI"]."</a><br>";
```

Imagen 3. Archivo PHP que consulta la BD (fuente propia).

Nuestra base de datos no va ser un esquema relacional debido a que solo se usará como muestra, por lo que solo cuenta con una tabla *(Ver Imagen 4)*, tomando en consideración que para realizar un scraping y protección del mismo, tomamos como punto de partida los datos mostrados en pantalla independientemente de la estructura generada del lado del servidor.

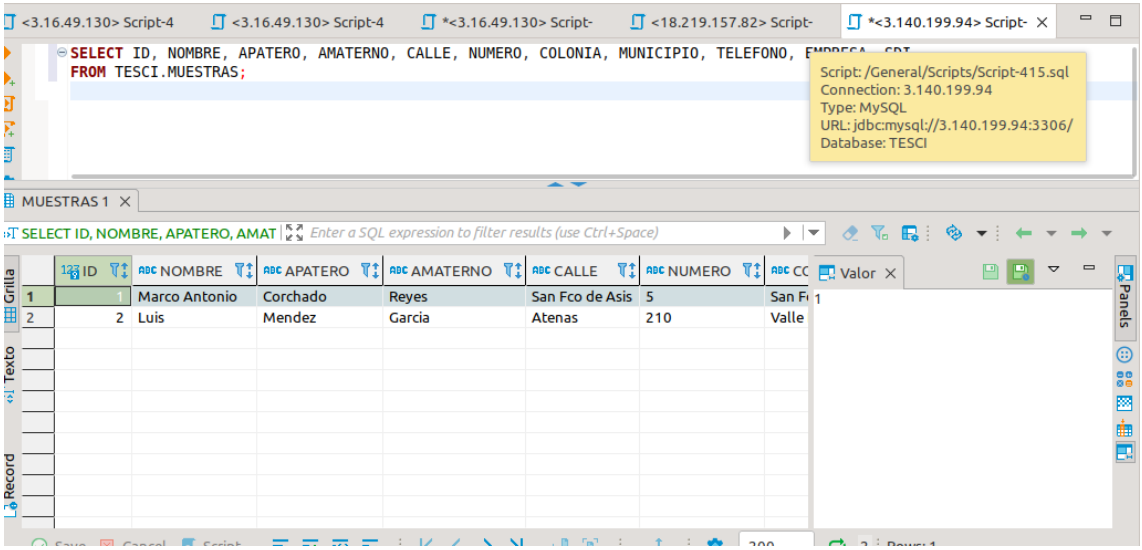


Imagen 4. Base de datos con información (fuente propia).

4.1.3 Creación de API

Tenemos dos opciones de desarrollo del API un tipo REST o un tipo SOAP, lo primero que debemos considerar es que el API solo tendrá métodos de consulta, y la información que obtengamos será de campos variables o desconocidos, lo ideal es descartar un modelo XML para la lectura de esta información, si utilizamos un objeto JSON tenemos la ventaja de no desarrollar un esquema XSD para la parametrización de datos.

Nuestra API la trabajaremos fuera del servidor, ya que será un externo el que intente extraer la información de nuestro sitio WEB, para esto vamos a crear 2 archivos, uno en PHP y uno en PYTHON. El archivo PHP nos servirá para dar un formato de salida de tipo JSON mediante los headers que interpreta el navegador, y mediante una variable de tipo GET ejecutará en la consola del sistema operativo el programa en python que es el que va realizar el Scraping a nuestro "sitio web 1" *(Ver imagen 5)*.

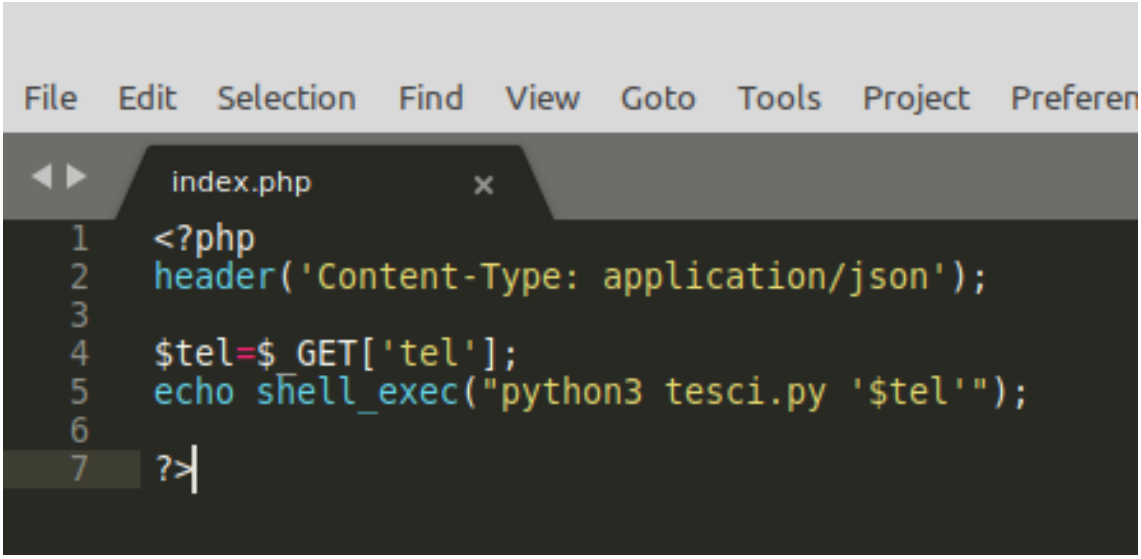


Imagen 5. Ejecución del programa de scraping mediante PHP (fuente propia).

El programa "tesci.py" hace la extracción de datos de nuestra primer página web, aquí mostramos que con unas simples línea de código podemos hacer una consulta y con un ciclo for ir almacenando cada dato de nuestro HTML*(Ver imagen 6)*, si la página tuviera n cantidad de registros sería el mismo proceso de extracción y almacenamiento de los mismos en un servidor externo.

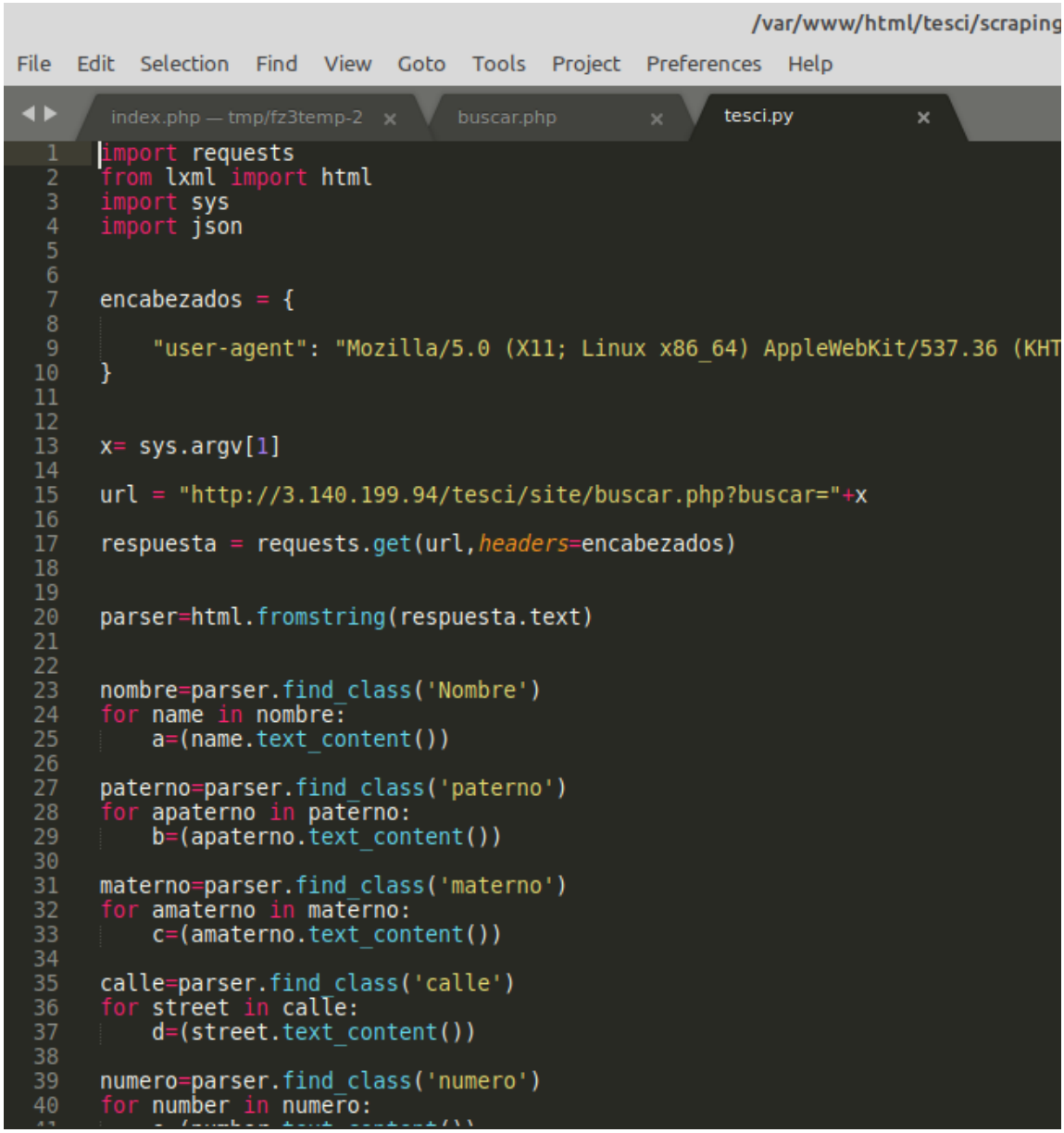


Imagen 6. Primera parte de scraping (fuente propia).

En la segunda parte del código mostramos el la continuación del barrido de datos y como con las variables recolectadas comenzamos el armado de un arreglo para una salida JSON *(Ver Imagen 7)*

Imagen 7. Segunda parte de scraping, armado de JSON (fuente propia).

Al ejecutar el PHP desde Postman podremos ver la respuesta que nos regresa, enviaremos una variable GET que será el identificador del cliente,y una consulta de tipo GET *(Ver Imagen 8)* y con esto el API estará lista para ser consumida por cualquier sistema.

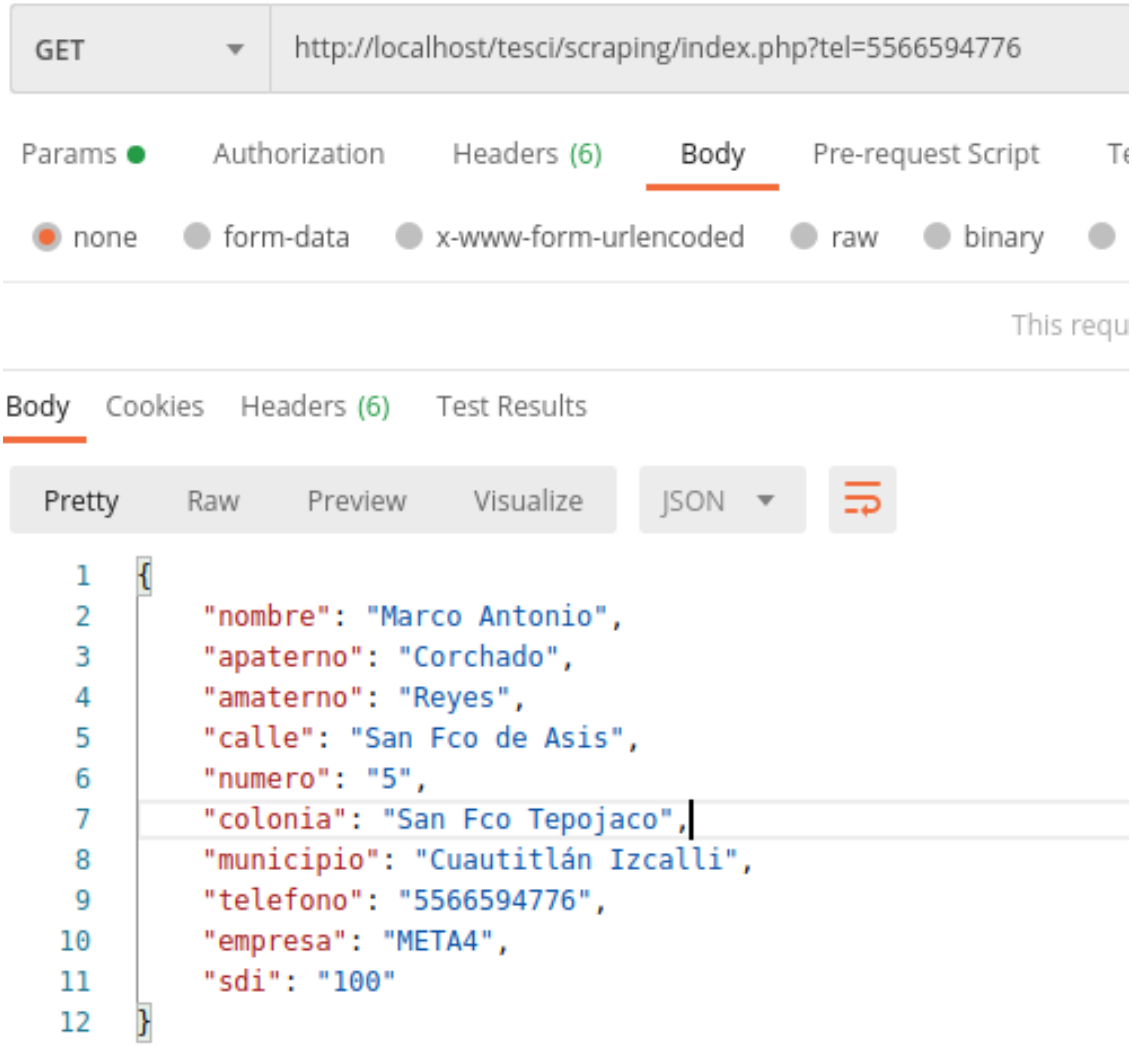


Imagen 8. API consumida por POSTMAN (fuente propia).

4.1.4 Creación de sitio web 2

En ese segundo sitio web vamos a simular ser la empresa otorgadora de créditos, la cual al consumir el API creada en el punto anterior se realiza una toma de decisiones *(Ver imagen 9)*, en este caso toma como referencia al campo SDI que es el salario diario integrado, sin ser necesario realizar una consulta en buro de credito es posible conocer el perfil salarial de la persona, sin tener acceso a la base de datos de manera directa.

```php
<?php

$busca=$_POST['buscar'];

$url="http://localhost/tesci/scraping/index.php?tel='$busca'";


$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => $url,
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'GET',
));

$response = curl_exec($curl);

curl_close($curl);
//echo $response;



$json = json_decode($response);
    $nombre  = $json->nombre;
    $sdi  = $json->sdi;

if($sdi>=20){

  echo "Estimado ".$nombre." FELICIDADES eres apto a obtener un credito";
}else{

  echo "Estimado ".$nombre." lo sentimos, tu solicitud no puede ser aprobada";
}
```

Imagen 9. Consumiendo el API para toma de decisiones (fuente propia).

4.1.5 Análisis de logs autónomo

El análisis de log se considera el “core” del sistema y trabaja con el método de “aprendizaje supervisado” del Machine Learning, podemos realizar la agrupación de información en base a modelos predictivos *(Ver imagen 10)*, desarrollamos el concepto de clasificación y regresión; la clasificación es utilizada al momento de extraer los datos que nos son útiles para la limpieza de nuestro archivo de logs general pues en él contamos con todo lo que pase por el protocolo HTTP, y la regresión es utilizada en cada una de las interacciones que se le da al archivo para leer nuevamente los datos dentro de él, haciendo de este ejercicio un bucle infinito para descartar o recopilar los datos necesarios para el uso de nuestra herramienta.

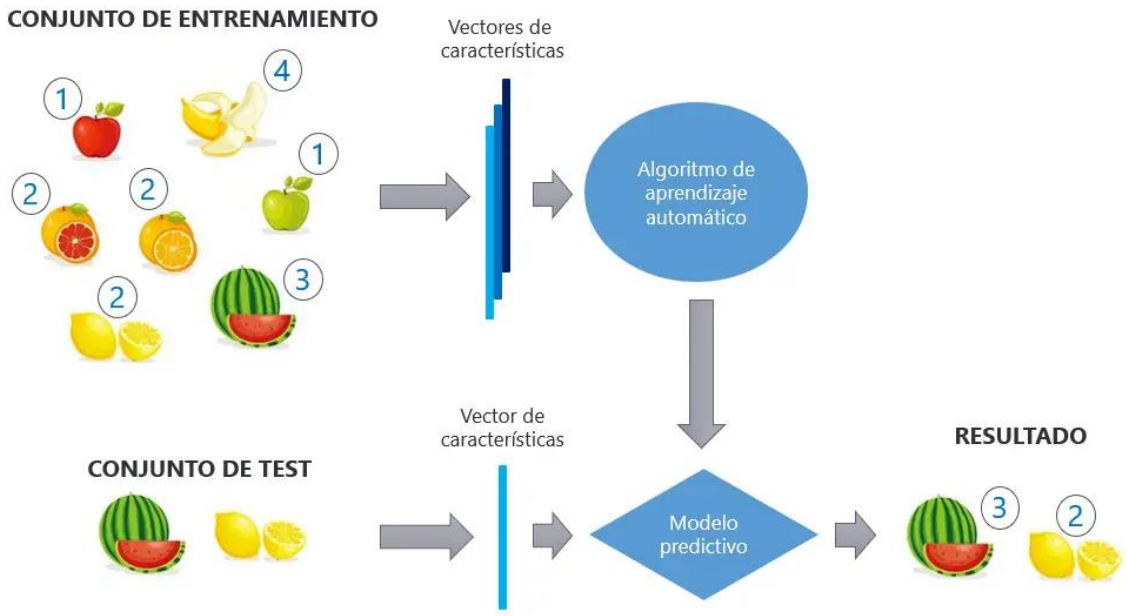


***Imagen10. Aprendizaje Supervisado** (Diego Calvo | Mar 23, 2019 | Aprendizaje automático)*

De esta manera podemos predecir cuando los datos están teniendo una saturación de búsqueda en el servidor y dejando listo el archivo de logs nuevo, es decir un archivo depurado, una vez listo el archivo será alcanzable poder analizarlo en otra etapa para tomar la decisión de bloquear o no los portales WEB.

El análisis se ejecuta minuto a minuto en el servidor para poder tener la información más real posible, debemos considerar que los sistemas de aprendizaje supervisado pudieran tener un margen de error, y parte del Machine Learning es “enseñarle” a los sistemas como distinguir de manera correcta los datos, es cuando ejecutamos una regresión en su análisis.

4.1.6 Programa de validación de logs

El programa que realiza el barrido de los logs creados de manera autónoma tiene una sencilla pero importante función, la cual es encender o apagar una bandera para avisar al sitio web si muestra o no la información, busca la IP del servidor que esté ejecutando la petición y hace un conteo creando un nuevo archivo con este resumen *(Ver imagen 11)*, si el número de intentos coincide o supera el límite parametrizado es cuando la bandera cambia su estatus.

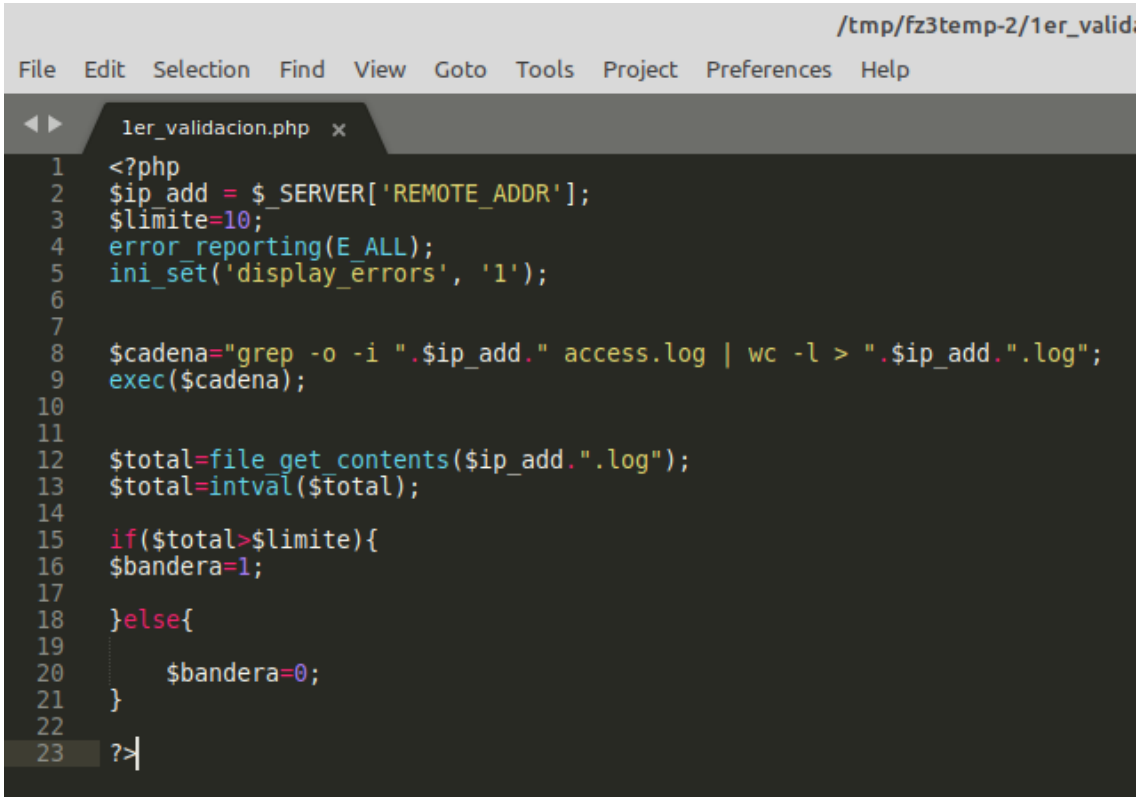


Imagen 11. Programa de asignación de bandera (fuente propia).

4.2 Desarrollo

4.2.1 Configuración de Arquitectura

En la consola de AWS podemos ver alguno de los servicios que nos ofrece *(Ver imagen 12)*, como se mencionó en el análisis tendremos que configurar una instancia con el servicio EC2, el cual nos permitirá montar un servidor web en la nube.

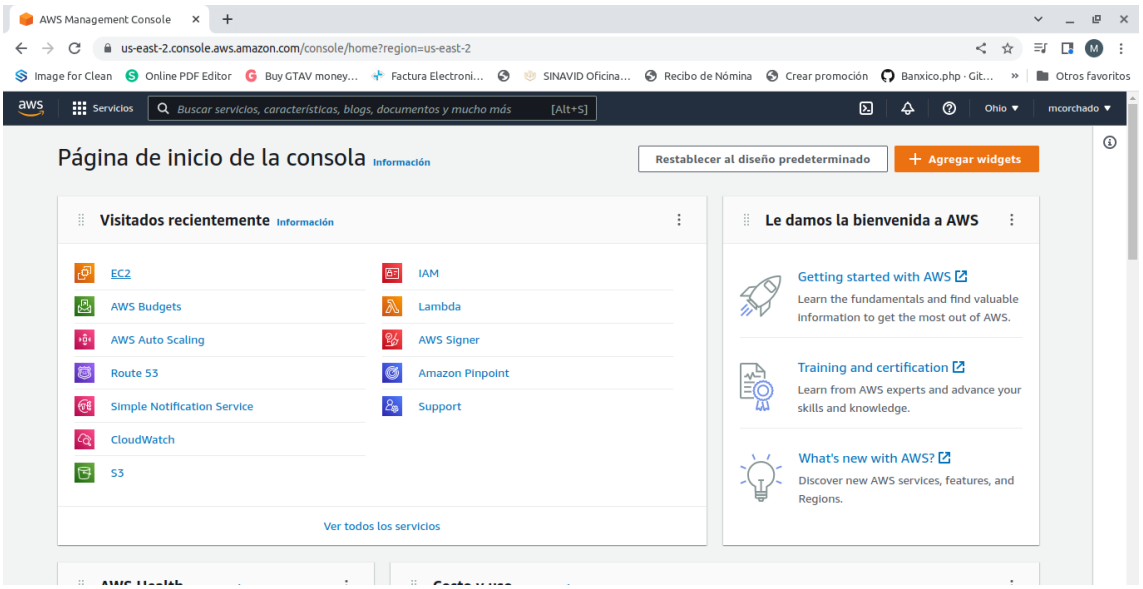


Imagen 12. Consola de administración de AWS (fuente propia).

Al iniciar la configuración de nuestro servidor con sistema operativo Ubuntu Server en su versión 22.04 *(Ver imagen 13)* el cual contiene los repositorios más actuales, sin embargo, se puede usar la versión deseada.

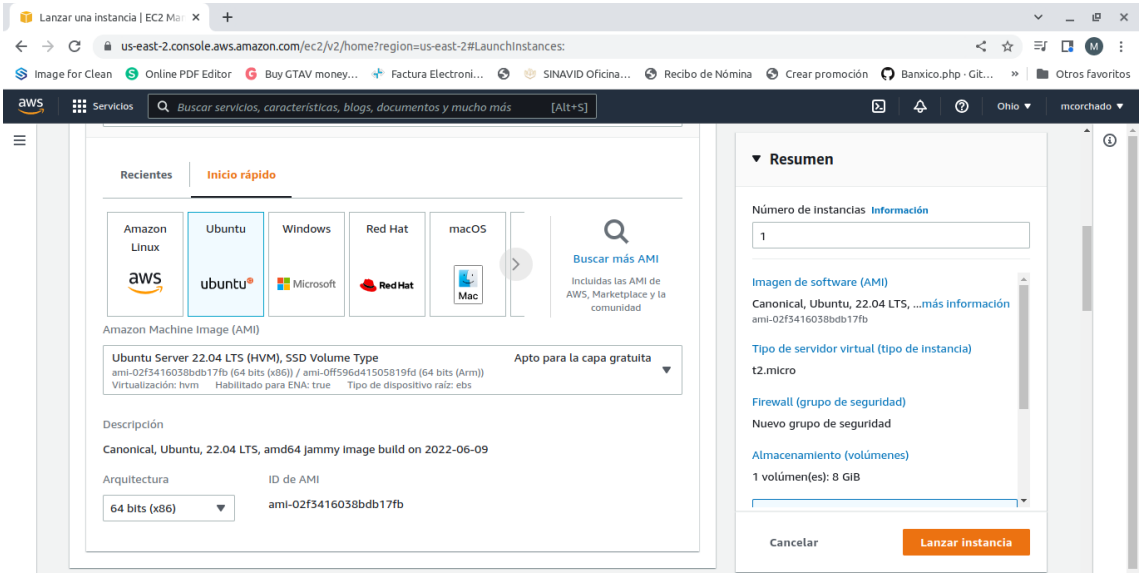


Imagen 13 Sistema Operativo Ubuntu (fuente propia).

Una vez que se configura la instancia a utilizar podremos observar los datos de nuestro servidor como la IP pública y privada *(Ver imagen 14)*, la IP privada es con la que se hacen configuraciones locales, en caso de tener más de una instancia con esta IP se realizará la comunicación local entre servidores.

La IP pública es con la que nos conectaremos vía SSH, FTP o para hacer nuestras consultas HTTP, es con la que tendremos entrada y salida de nuestro servidor.

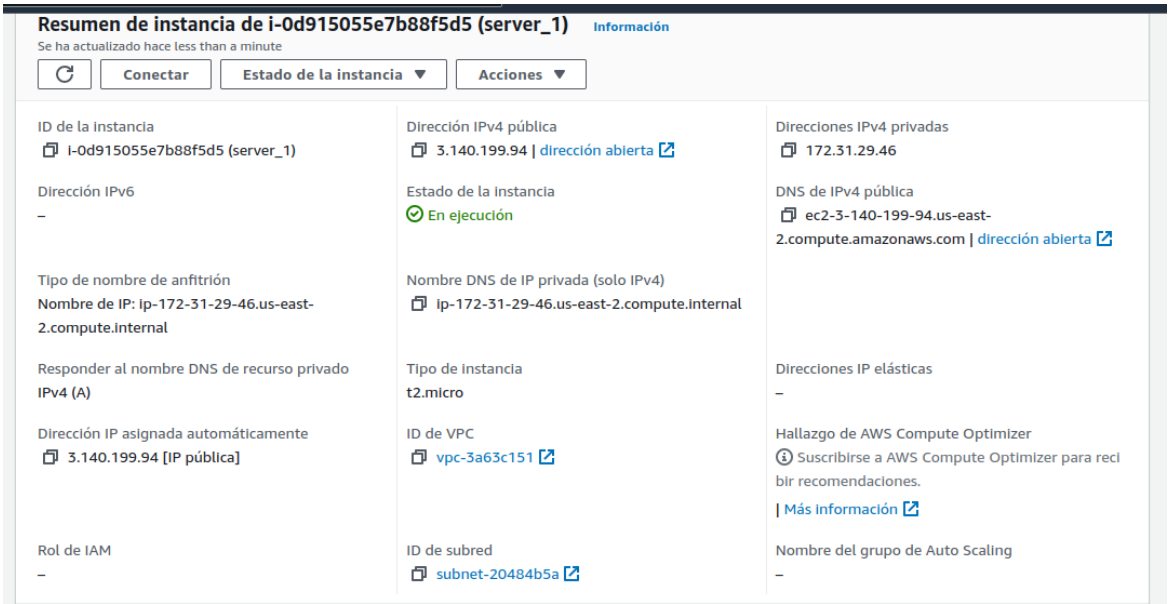
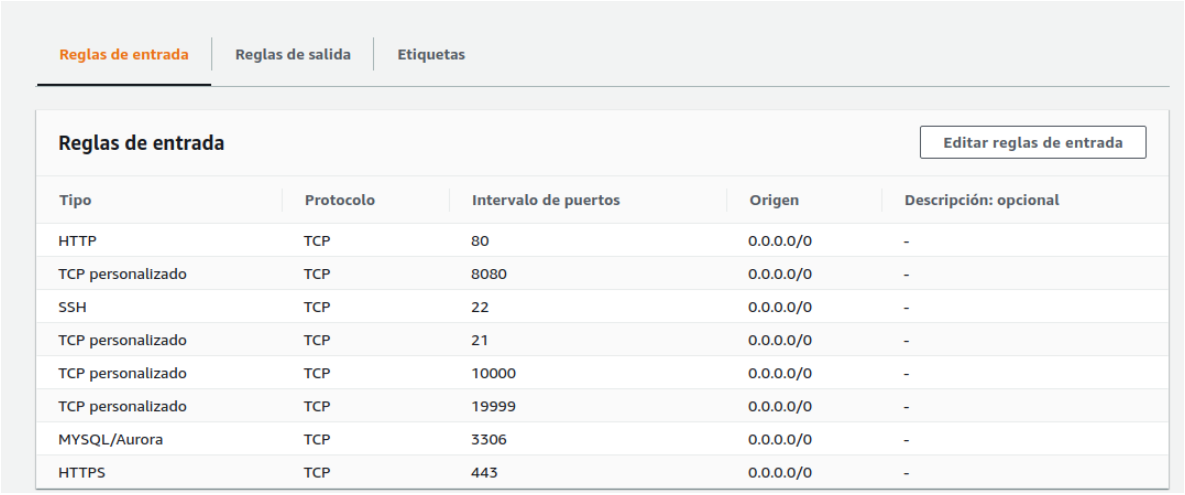


Imagen 14. Resumen de instancia (fuente propia).

El firewall que nos presenta la interfaz de Amazon Web Services es conocido como grupos de seguridad, y en él podremos dar de alta el puerto, protocolo y el origen mediante una IP *(Ver imagen 15)*, de igual manera se pueden configurar las reglas de salida, si es deseable solo tener respuesta de ciertas direcciones IP.



Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional
HTTP	TCP	80	0.0.0.0/0	-
TCP personalizado	TCP	8080	0.0.0.0/0	-
SSH	TCP	22	0.0.0.0/0	-
TCP personalizado	TCP	21	0.0.0.0/0	-
TCP personalizado	TCP	10000	0.0.0.0/0	-
TCP personalizado	TCP	19999	0.0.0.0/0	-
MYSQL/Aurora	TCP	3306	0.0.0.0/0	-
HTTPS	TCP	443	0.0.0.0/0	-

Imagen 15. reglas de entrada en el firewall (fuente propia).

Al configurar de manera correcta los puertos podremos comprobar que ya tenemos acceso a nuestro servidor mediante la IP pública, notaremos una breve descripción de la versión instalada de apache *(Ver imagen 16)*, debido a que en Linux viene incluido este servicio de manera nativa, también podemos comprobar con el comando CURL de la consola del sistema operativo *(Ver imagen 17)*

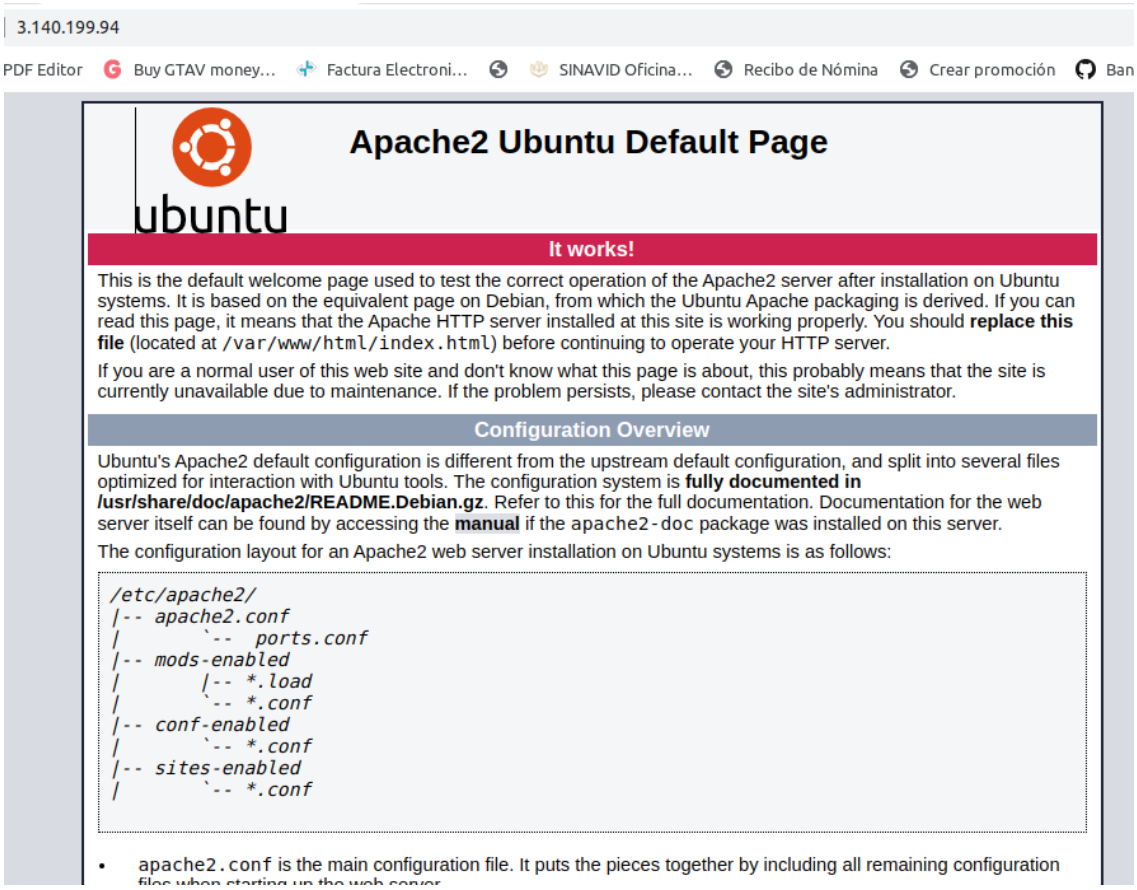


Imagen 16. Servidor Apache montado (fuente propia).

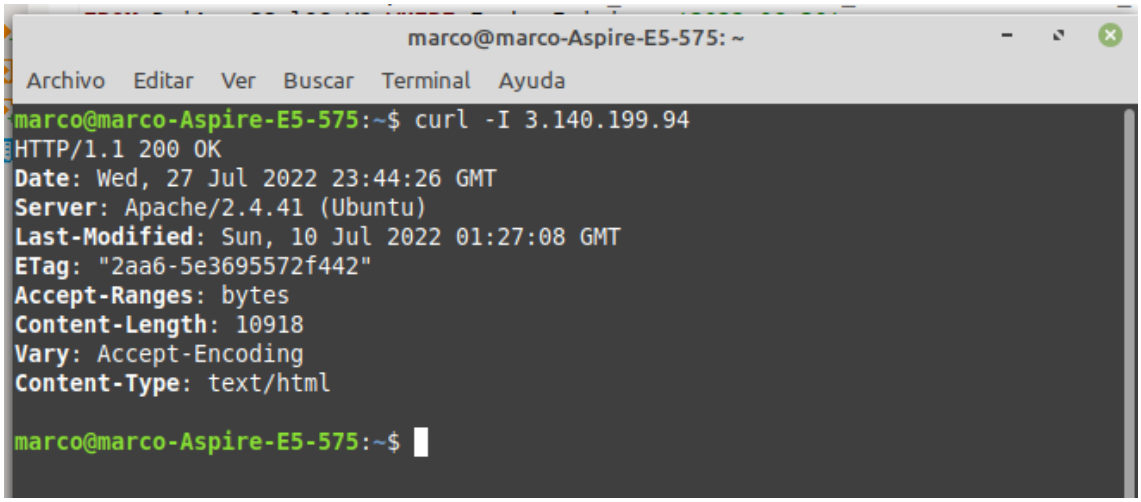


Imagen 17. comprobación con CURL (fuente propia).

Webmin nos va ayudar a gestionar mediante una interfaz todos los servicios que el servidor esté ejecutando, para poder entrar a la interface primero debemos asegurar que el servicio esté corriendo *(Ver imagen 18)*, este servicio corre en el puerto 10000 es por eso que en el firewall lo pusimos como una regla de entrada, para así poder acceder desde cualquier lado.

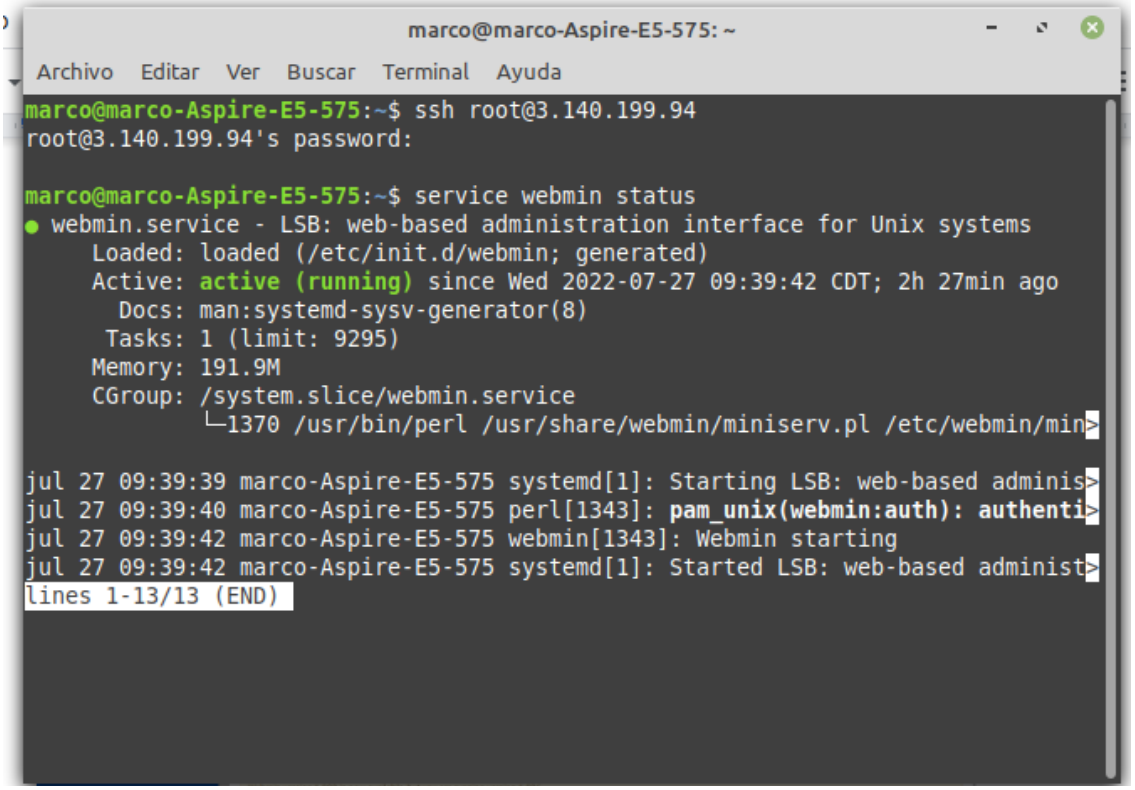


Imagen 18. Servicio Webmin corriendo (fuente propia).

Damos por terminada la configuración básica del servidor al ver que podemos entrar a webmin mediante la dirección URL conformada por la IP y el puerto 10000 *(Ver imagen 19)*. En esta pantalla notamos el rendimiento actual del servidor y podemos dar inicio a la programación de la herramienta.

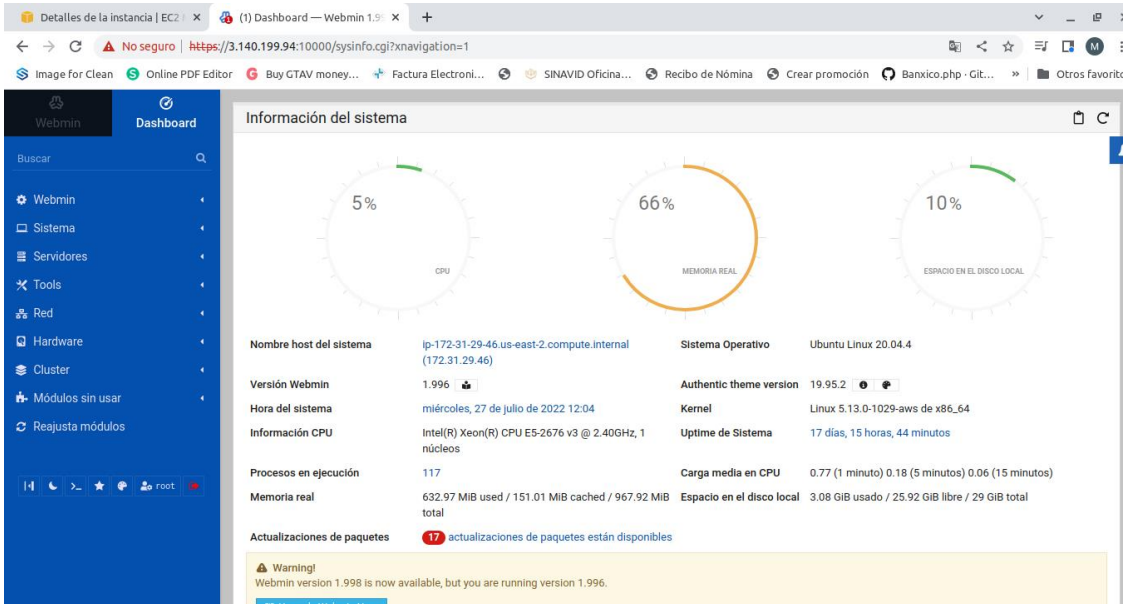


Imagen 19. Webmin corriendo en servidor (fuente propia).

4.2.2 Desarrollo de Herramientas

El desarrollo generado parte en el análisis de las consultas del servidor, en este caso se hace una búsqueda que coincida con la IP que está realizando múltiples consultas (*Ver imagen 20).*

```
root@ip-172-31-29-46:/var/log/apache2# tail -100f access.log | grep -i 187.190.186.198
187.190.186.198 - - [15/Jul/2022:12:33:37 -0500] "GET /tesci/site/ HTTP/1.1" 200 450 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:39 -0500] "GET /favicon.ico HTTP/1.1" 404 490 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:43 -0500] "GET /tesci/site/buscar.php?buscar= HTTP/1.1" 200 211 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:49 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 500 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:50 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:50 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:50 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:50 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:50 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:50 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:51 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:33:51 -0500] "GET /tesci/site/buscar.php?buscar=5566594776 HTTP/1.1" 200 499 "http://3.140.199.94/tesci/site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
187.190.186.198 - - [15/Jul/2022:12:34:40 -0500] "-" 408 0 "-" "-"
```

Imagen 20. Detección de consultas múltiples (fuente propia).

De manera autónoma el servidor minuto a minuto hace un análisis de las consultas realizadas, y guardando en un “access.log” los nuevos datos encontrados *(Ver imagen 21),* y a su vez nos crea un log con la IP que el servidor reconoció como sospechoso, este archivo nos servirá para determinar si se realiza un bloqueo o no.

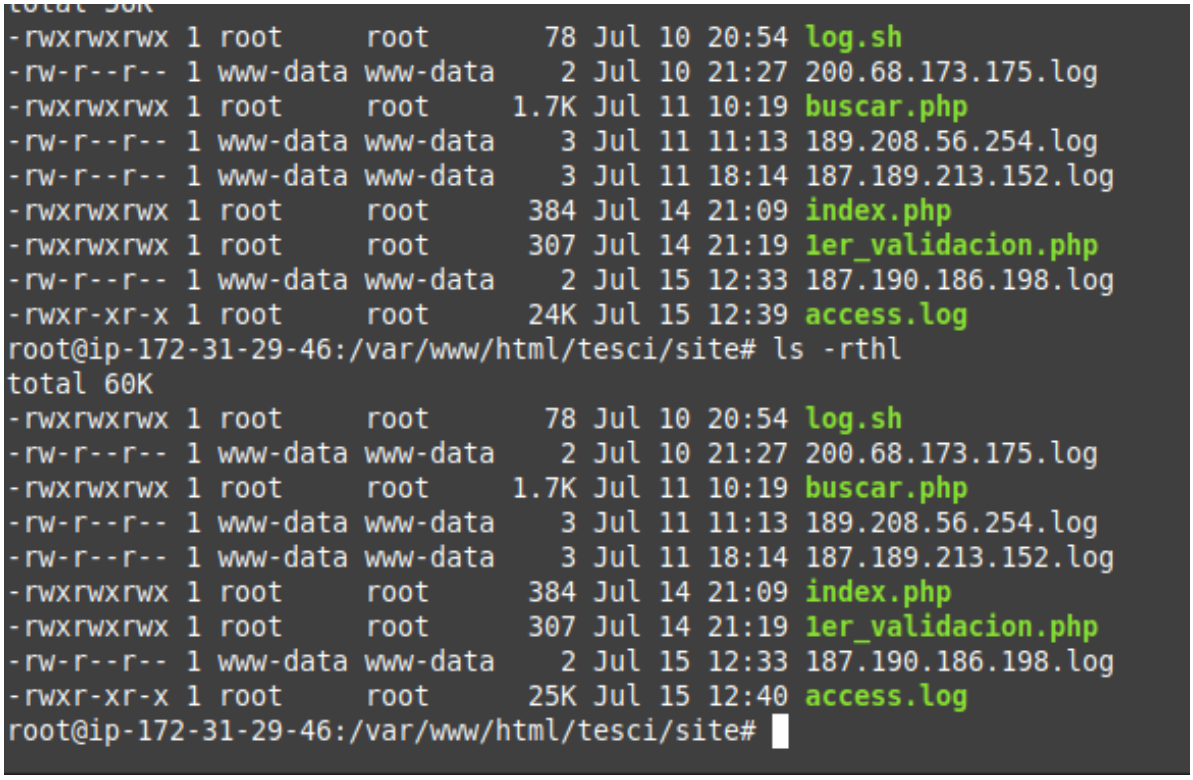


Imagen 21. Archivo access.log actualizado (fuente propia).

Una vez creado el archivo IP.log tenemos un programa PHP que tomará los datos de este archivo, sobre la IP directamente, se crea una bandera que activa o desactiva la información de la página web, en este ejemplo el límite es de 10 consultas a la IP *(Ver imagen 22)*.

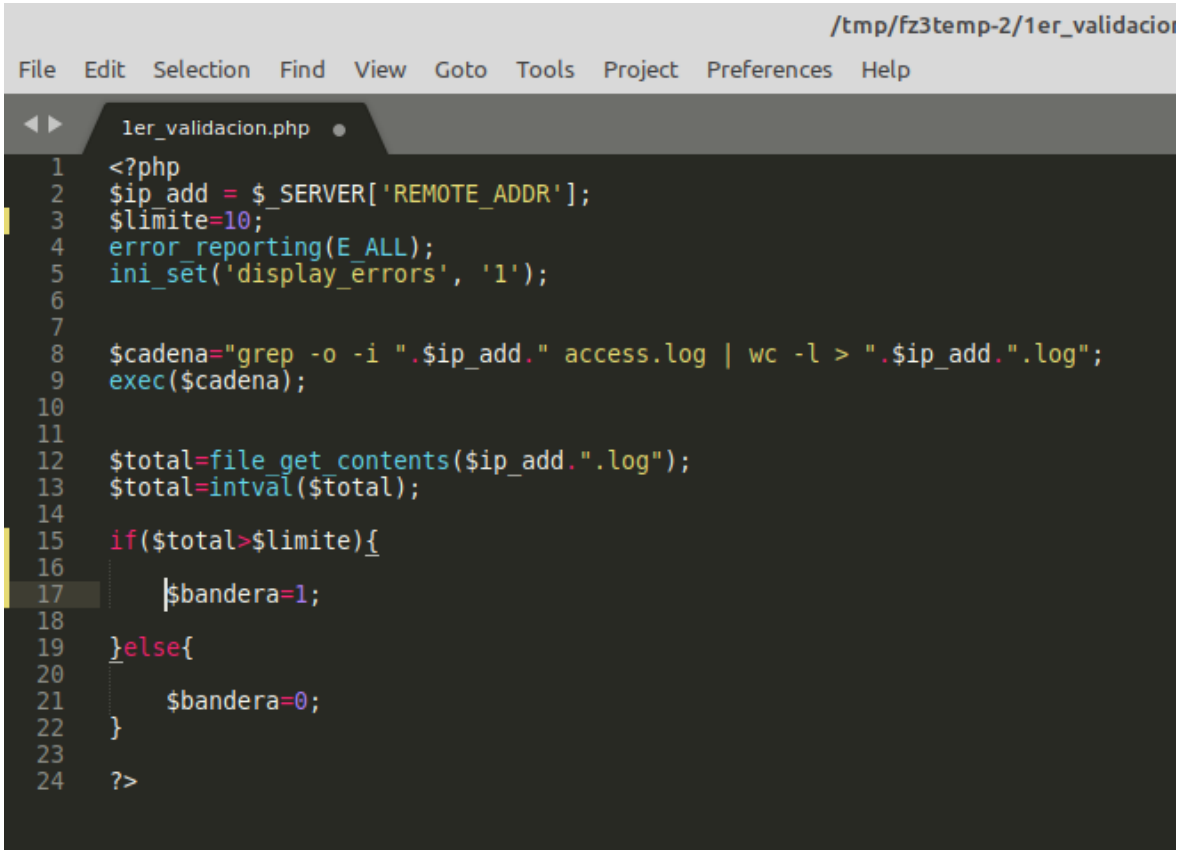


Imagen 22. Recopilación de análisis para bandera (fuente propia).

Finalmente, en un archivo PHP, llamamos el programa que hace el análisis de IPs, se realiza una condición sobre la variable bandera, en donde dependiendo el valor, muestra un error o los datos de la persona *(Ver imagen 23)*.

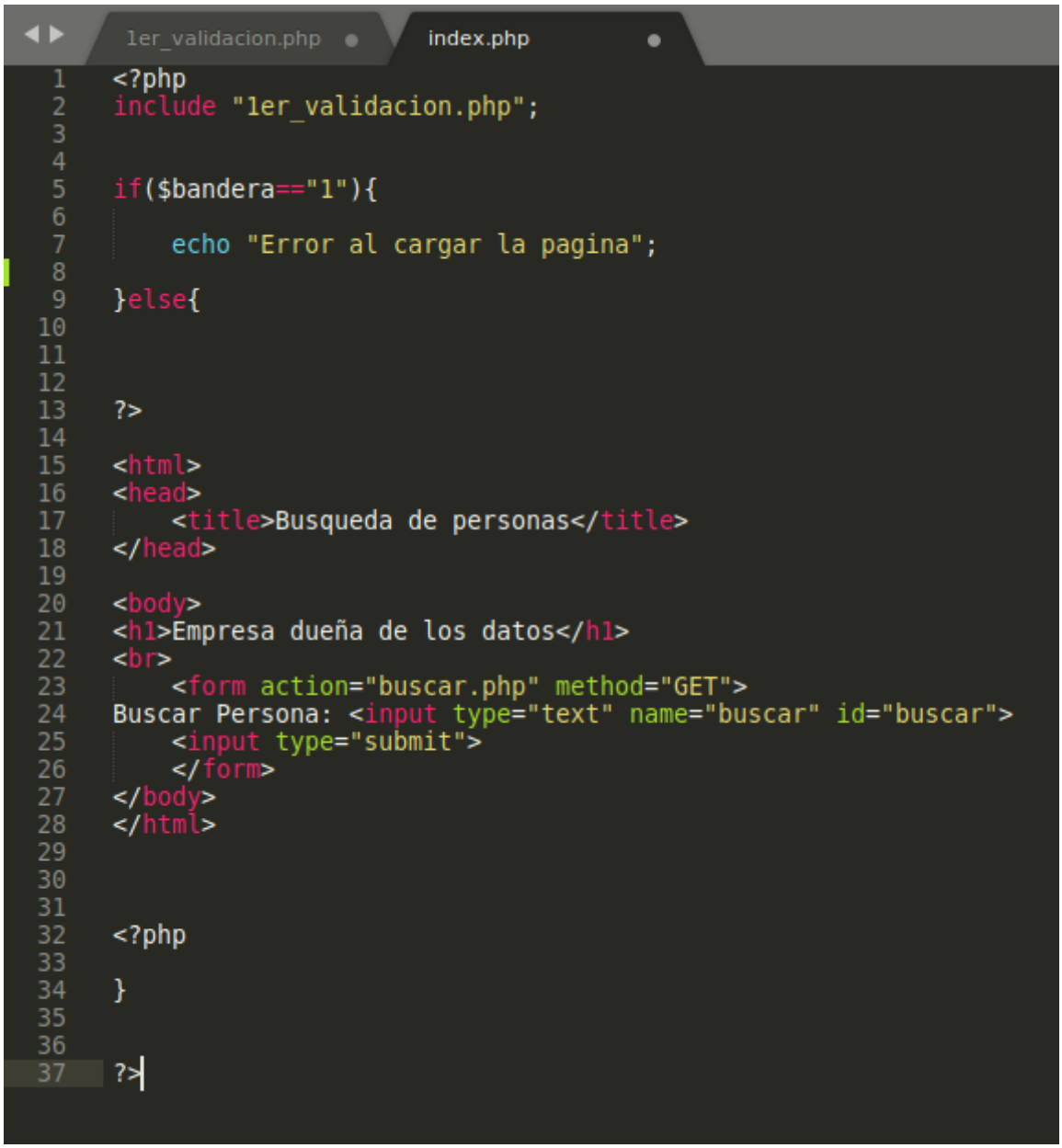


Imagen 23. Condición de variable para bandera (fuente propia).

4.3 Pruebas

En un entorno productivo, podemos asignar un límite diferente dependiendo de la gravedad de datos expuestos, esta variable puede ser almacenada en el código o en una base de datos, en este caso se integró directamente en el código *(Ver imagen 24)*, si la variable está en un rango permitido mostrará los datos a consultar *(Ver imagen 25)*, de lo contrario mostrará un error como parte del HTML *(Ver imagen 26)*.

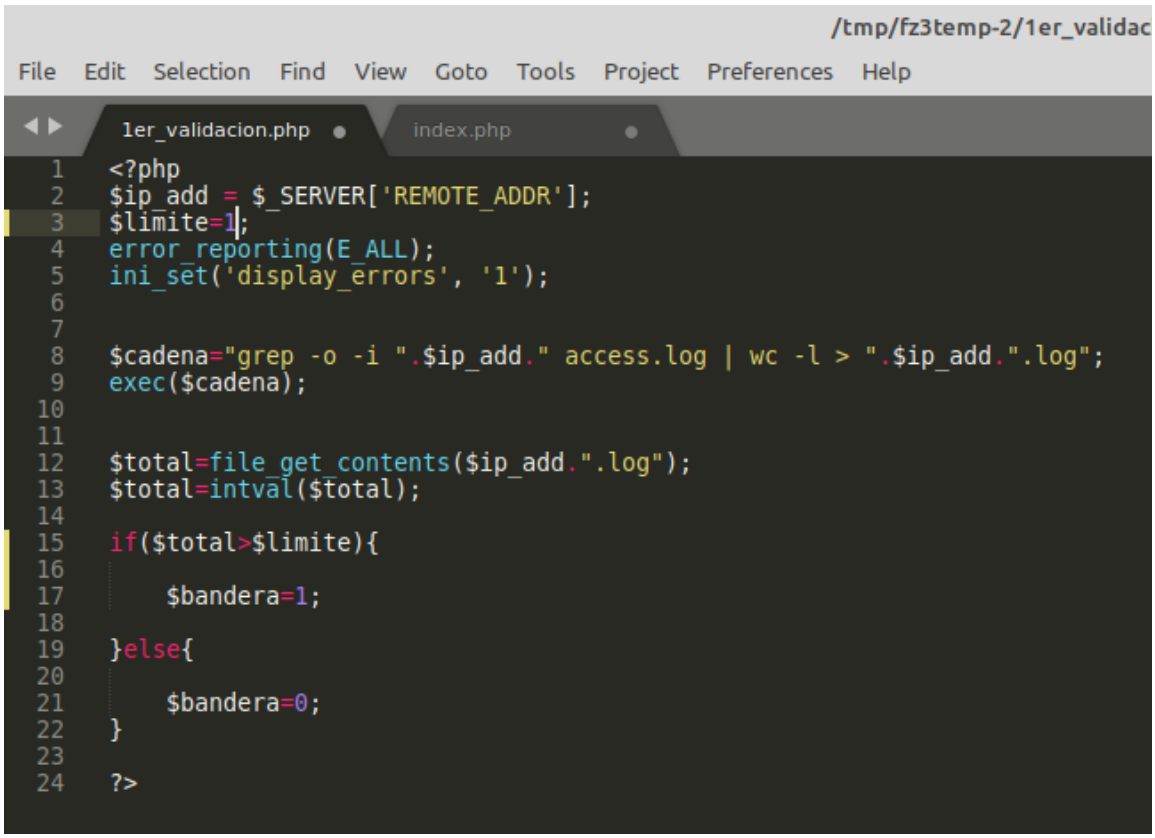


Imagen 24. Cambiando valores en variable límite (fuente propia).

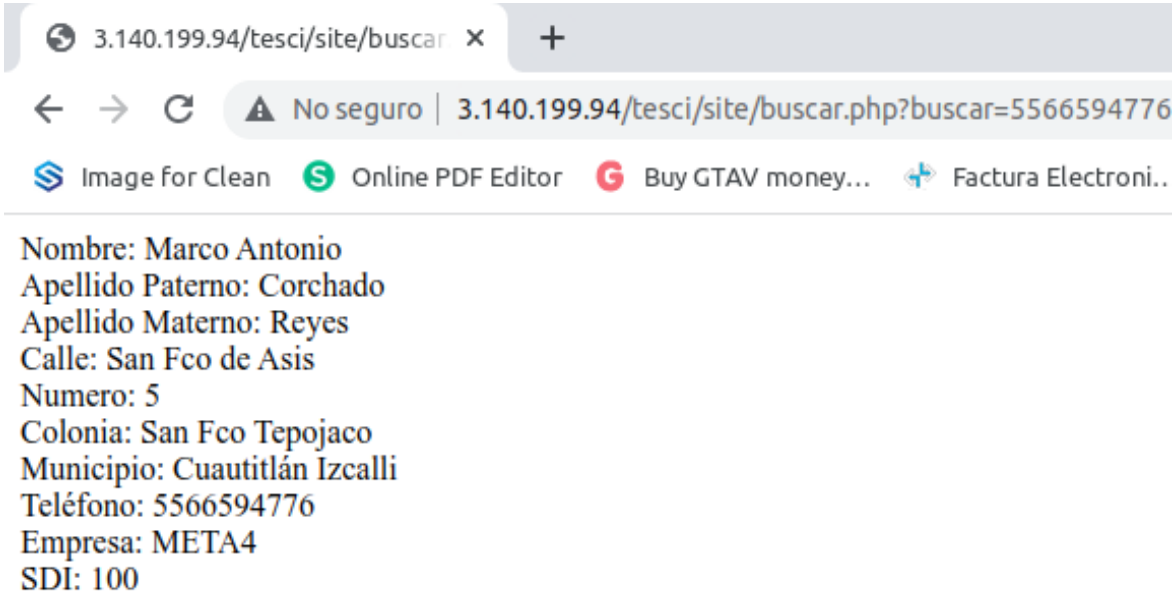


Imagen 25. Solicitud exitosa (fuente propia).

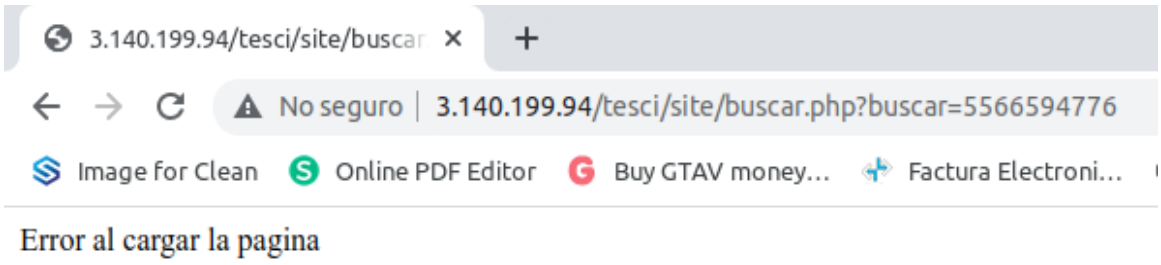


Imagen 26. Solicitud erronea (fuente propia).

Este error al ser controlado regresara un código HTTP 200, por lo que indica que el servidor no presenta ninguna inconsistencia o error en la infraestructura *(Ver imagen 27)* y demostrando que la herramienta implementada funciona ya que es un error generado de manera automática en base al análisis que el servidor realizó de manera autónoma.

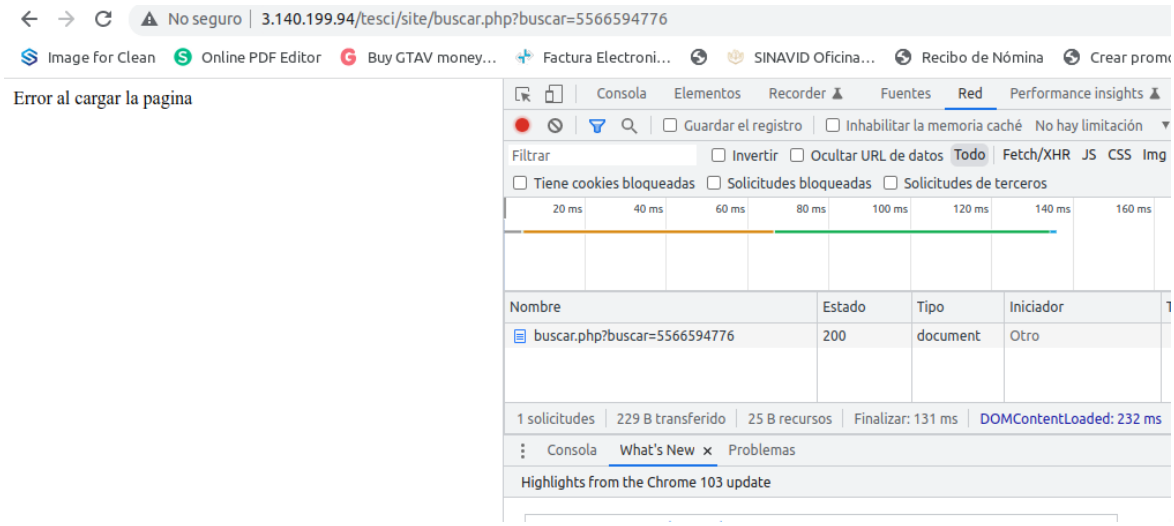


Imagen 27 Código HTTP 200 obtenido (fuente propia).

Capítulo V Conclusiones y/o resultados

5.1 Resultados

Los resultados del servicio parten del intento de automatizaciones de consultas mediante web scraping, actualmente el gobierno de México cuenta con varios portales en donde la consulta se basa en el llenado de un formulario y sin ningún tipo de validación adicional que compruebe que la consulta se lleva a cabo por el dueño de la información, algunos portales con estas características son la consulta de Cedula profesional, validación de INE/IFE, numero de seguridad social, semanas cotizadas IMSS/ISSSTE (con información salarial), IFT, entre otros.

Simulamos un portal web con el llenado de un formulario, el cual tiene acceso a una base de datos que contiene información de un usuario *(Ver imagen 28)*.

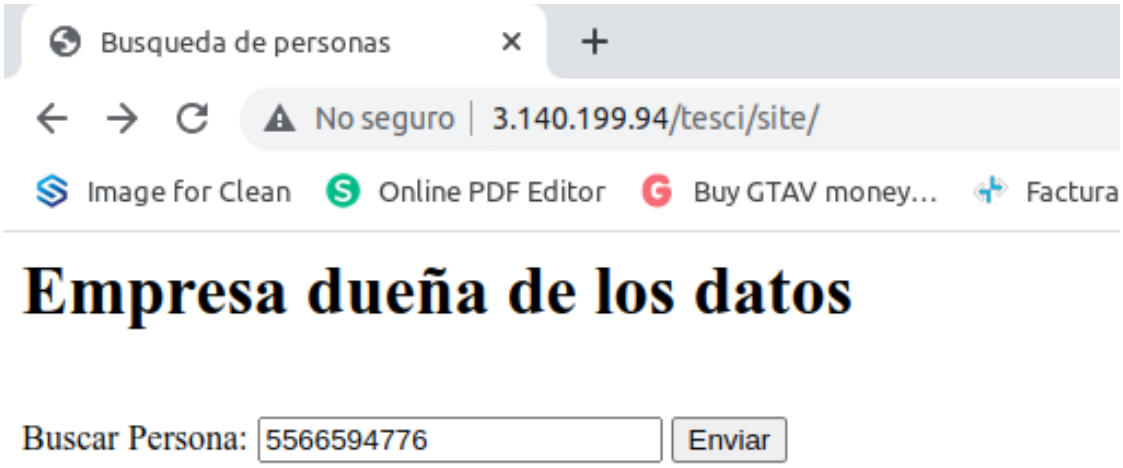


Imagen 28. Formulario de captura (fuente propia).

Al capturar los datos del formulario anterior se imprime en pantalla la información confidencial del usuario, como su dirección, su información salarial, laboral y nombre, comprobando que cualquier persona o empresa puede tener acceso a estos datos *(Ver imagen 29)*.

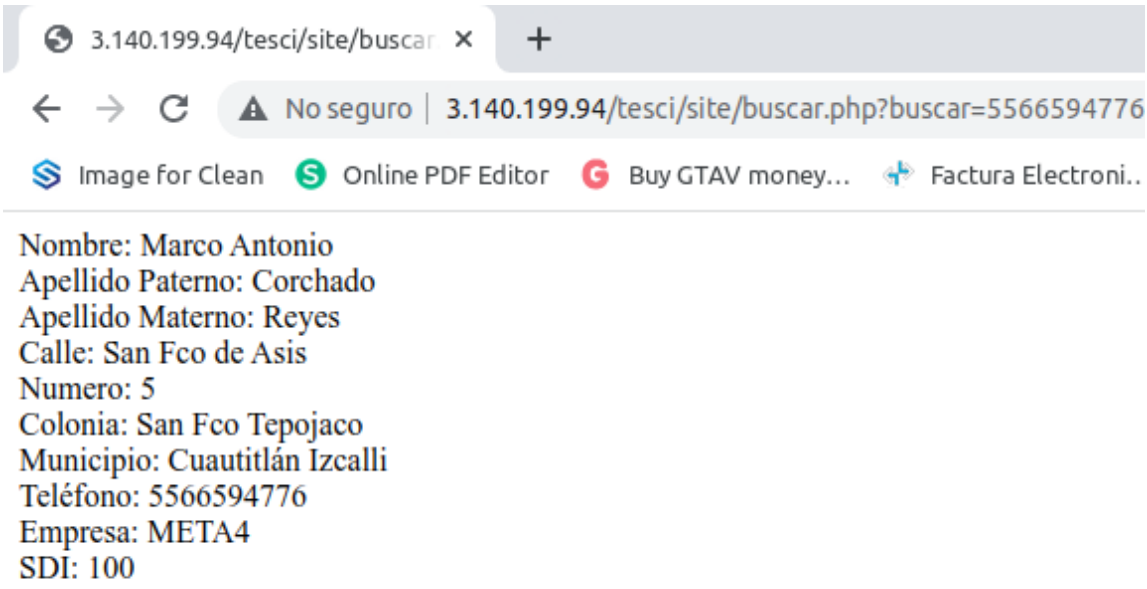


Imagen 29 Datos expuestos (fuente propia).

Un caso de uso de esta vulnerabilidad son las empresas financieras, en donde pueden hacer un API utilizando scraping o alguna otra técnica de análisis de contenido web *(Ver imagen 30)*.

Al crear un API de extracción de datos podemos integrarla prácticamente en cualquier sistema de consulta de datos, y con cualquier tipo de plataforma, incluso sistemas móviles como android o IOS, y que con un solo click podemos consultar la información que deseemos.

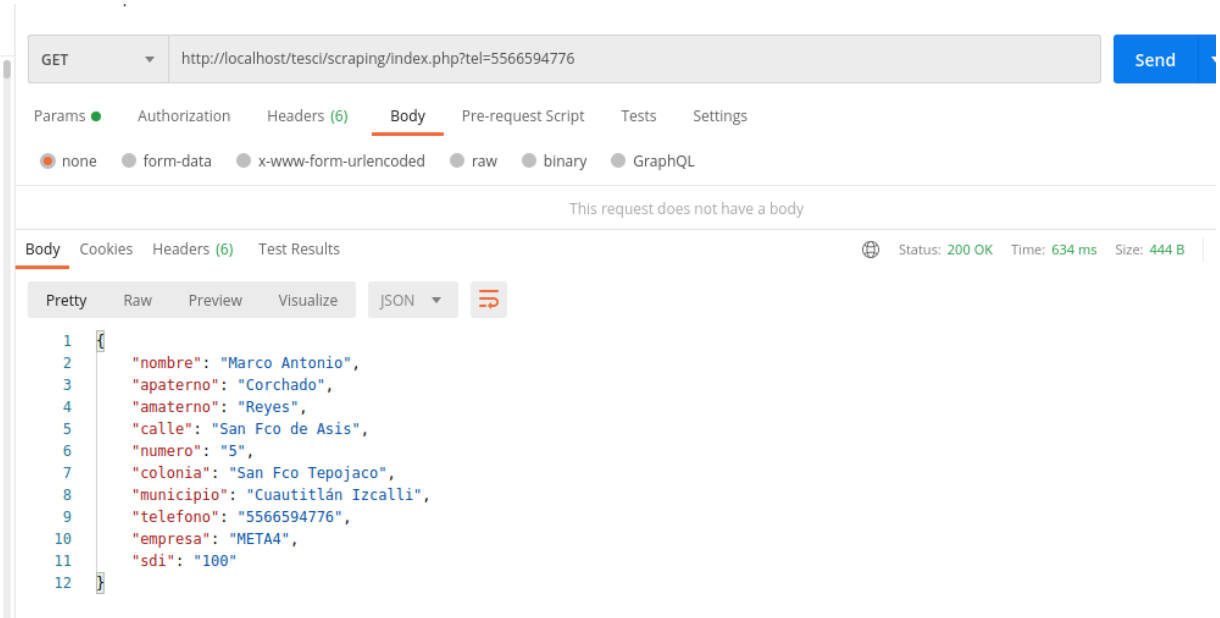


Imagen 30 scraping aplicado, API creada (fuente propia).

Al realizar un API pueden obtener de manera inmediata los datos requeridos para alguna solicitud y trámite integrándolos en sistemas propios *(Ver imagen 31)*, muchas financieras utilizan este modelo al otorgar créditos mediante aplicaciones, con la finalidad de acelerar los trámites.

Imagen 31. Empresa consume el API (fuente propia).

Al integrar el API en los sistemas de toma de decisiones, se puede otorgar o denegar el crédito de manera automática e inmediata, sin necesidad de consultar en buró de crédito o requerir papeleos como comprobantes de ingresos etc *(Ver imagen 32)*.

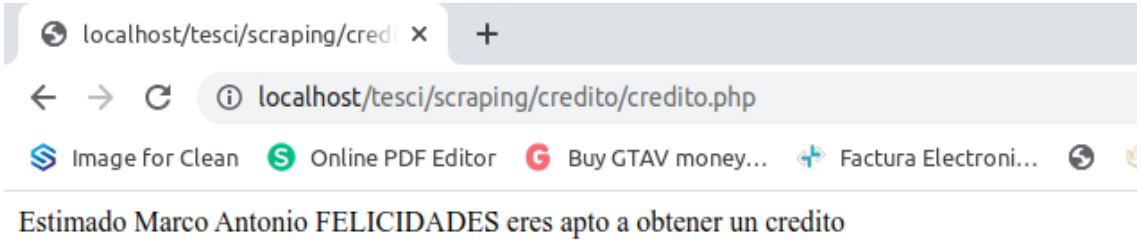


Imagen 32. Empresa obtiene datos automatizados (fuente propia).

El sistema implementado en esta investigación realiza un análisis de los datos solicitados al servidor, y al detectar que se usa un robot para la consulta, este bloquea la información a imprimir a pantalla *(Ver imagen 33)*, dando como resultado que el API realizada por las empresas externas deje de funcionar *(Ver imagen 34).*

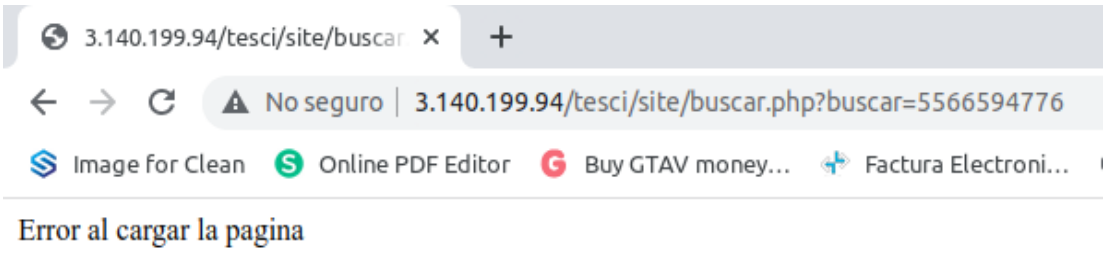


Imagen 33. Servicio Web bloqueado por servicio anti robots (fuente propia).

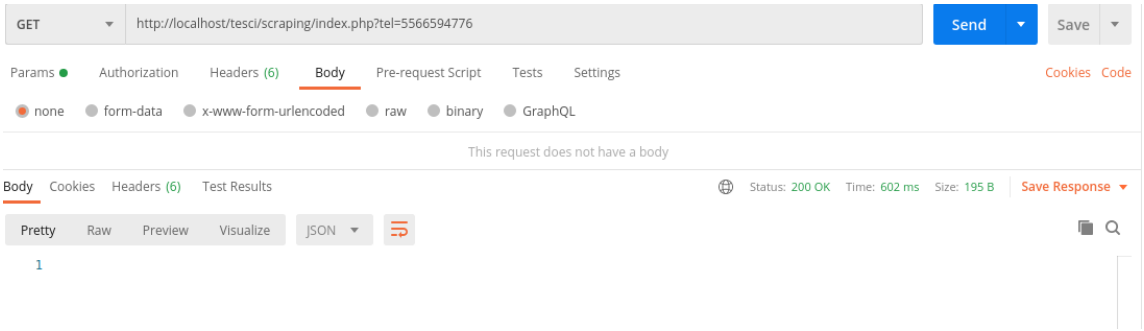


Imagen 34. Servicio scraping bloqueado por servicio anti robots (fuente propia).

5.2 Conclusiones

Al no tener estos datos disponibles las financieras no podrán acceder a los datos del cliente, dando como resultado un modelo de negocio tradicional y el cliente al entregar documentación física o digital, sabrá que es lo que está entregando.

En el sector financiero esto tendría un impacto importante en la otorgación de créditos, debido a que el armado del expediente digital del prospecto se tendría que llenar de manera tradicional o con el consentimiento del mismo.

Recordemos que aunque sea una eficiencia en el sector financiero el poder agilizar los trámites de solicitudes de crédito, se expone la información al no tener una regulación algunas financieras, más del tipo FINTECH, y si este tipo de técnicas pueden ser utilizadas en este sector, es prácticamente lo mismo para otros sectores.

Bibliografía

Guerrero, V. A. B. (2019). Desarrollo del estado del arte en investigación: una herramienta basada en inteligencia artificial. *Revista Politécnica*, *15*(30), 70-81.

Villanueva Rodríguez, U. J. (2019). Investigación y desarrollo de técnicas de scraping.

González, G., & Morales, L. (2017). *Sistema para el monitoreo y gestión de datos, servicios y archivos con notificaciones electrónicas basado en PHP y Shell script en servidores con sistema operativo GNU/Linux Ubuntu Server. Caso: Inversiones INTRAWEB, CA* (Doctoral dissertation).

López Gil, A. (2018). Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software.

Robles, G., & Ferrer, J. (2002). Programación eXtrema y Software Libre. *Universidad Politécnica de Madrid. España*.

Rivera Meza, I. D. (2017). Desarrollo e implementación de un sistema de código de barras con la metodología XP para optimizar el control de asistencia en la junta administradora de Servicios de Saneamiento Quilcas.

López, J. (2018). Web scraping.

Monago Ruiz, A. (2019). Servicio Web API REST sobre el Framework Spring, Hibernate, JSON Web Token y BBDD Oracle.

Ocampo, M., & Santa Catarina, C. (2017). Fintech: tecnología financiera. *Recuperado de https://www. foroconsultivo. org. mx/INCyTU/documentos/Completa/INCYTU_17-006. pdf*.

Ocampo, M., & Santa Catarina, C. (2018). Inteligencia artificial.

Labrador, R. M. G. CURSO 14146 SHELLSCRIPTS EN LINUX.

Tume Fuentes, M. G. (2022). Estado del arte de la inteligencia artificial y su aplicación en el mantenimiento.

Castro, M., Sánchez Rivero, D., Farfán, J., Castro, D., Cándido, A., & Vargas, A. (2013). Aplicación de Servicios Web SOAP/REST para funcionalidades existentes en sistemas informáticos provinciales. In VII Simposio Argentino de Informática en el Estado (SIE)-JAIIO 42 (2013).

Mejıa, O. (2011). Computación en la nube. ContactoS, 80, 45-52.

Alonso Romero, L., & Calonge Cano, T. (2001). Redes neuronales y reconocimiento de patrones.

Sierra, M. D. C. S. (2007). Inteligencia artificial en la gestión financiera empresarial. *Pensamiento & Gestión*, (23), 153-186.

Hernández, A. T., Vázquez, E. G., Rincón, C. A. B., García, J. M., Maldonado, A. C., & Ibarra-Orozco, R. (2015). Metodologías para análisis político utilizando Web Scraping. *Res. Comput. Sci.*, *95*, 113-121.

Gonzales Quevedo, S. M. (2022). Implementación del proceso de pruebas funcionales automatizadas aplicadas para una API en una empresa de telecomunicaciones.

Rojas, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E28), 586-599.

Gil, E. (2016). Big data, privacidad y protección de datos. *Madrid: Agencia Estatal Boletín Oficial del Estado*.

González Cangrejo, J. (2022). Algoritmos de Aprendizaje Supervisado en la Clasificación de Exoplanetas en Python.

Muñoz Luna, M. K., & Garcia Rodriguez, L. F. (2017). Análisis de riegos y prototipo de una página web mediante autenticación CAPTCHA.

Martorell, S. O., & Gutiérrez, L. C. (2006). Protocolo de seguridad SSL. *Ingeniería Industrial*, *27*(2-3), 57-62.

Moreno, J. (2020). Webinar: Herramientas Serverless para pentesting.