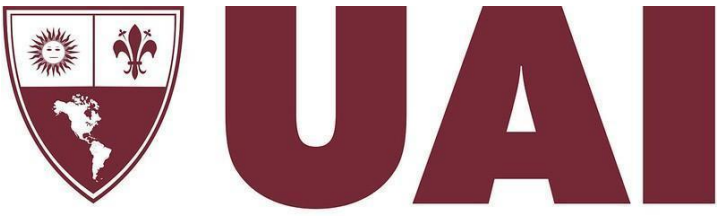


Universidad Abierta Interamericana

Sistema de Penalización basado en Datos Biométricos y Blockchain para Videojuegos

Tutoría técnica: Ing. Pablo Vilaboa

Profesora de trabajo final: Dra. Marcela Samela

Alumno: Kevin Miyashiro

Trabajo Final de Carrera presentado para obtener el título de Lic. en Gestión de Tecnología informática

Diciembre, 2023

Resumen

El presente Trabajo Final de Carrera tiene como objetivo ofrecer una solución conceptual alternativa, frente a las penalizaciones que se producen sobre los usuarios que infringen las normas en los videojuegos, obteniendo cierta ventaja ilegítima del resto de los usuarios y ocasionando un ambiente hostil en toda la comunidad en línea. Esta problemática se ha presentado de manera reiterada, sin poder lidiar con ello de raíz, y generando pérdidas económicas para las empresas de videojuegos.

Esta propuesta de intervención en el campo profesional busca complementar a los sistemas antitrampas actuales, centrándose en las sanciones hacia los usuarios reales de manera definitiva con el fin de mantener la comunidad limpia. Por tal motivo, en primer lugar, se explicará qué son las trampas en los videojuegos, como también así los diferentes sistemas antitrampas actuales y las penalidades que se aplican sobre los usuarios por infringir las normas. Luego se expondrán los diferentes conceptos de Blockchain, centrándose en los sistemas parcialmente descentralizados y en los contratos inteligentes (Smart-Contracts). Además, se utilizará la Biometría que servirá de base para obtener los datos de los usuarios de forma unívoca.

Se propone implementar un sistema de penalización parcialmente descentralizado que permita el almacenamiento de los datos biométricos, utilizando la tecnología Blockchain.

Por último, se elaborará un plan para llevar a cabo esta propuesta de intervención. También se mencionarán las limitaciones externas que condicionarán la ejecución exitosa del proyecto y se comentarán los trabajos futuros relacionados.

Palabras clave: *antitrampas, biometría, blockchain, penalizaciones, videojuegos.*

Abstract

The main objective of this final degree project is to offer a concept solution when facing punishments for cheaters in videogames. These users gain an illegal advantage against the rest of players making a really bad environment for the whole online gaming community.

Nowadays this problem stills happens frequently generating company economic losses and without having a solid solution to solve it.

To keep the community clean this solution proposal looks for complementing current anti-cheat systems and focusing on real user punishments in a permanent way.

First, it will be explained what a cheat is in videogames. Then the different types of existing anti-cheat systems will be seen, and which penalties are applied to cheaters for violating game rules.

After introducing the mentioned concepts Blockchain including partially decentralized systems and smart contracts will be exposed. Biometry will help to identify users and to obtain characteristics that are unique to an individual.

It is proposed to implement a partially decentralized punishment system based on Blockchain technology that allows biometrics data storage.

At the end of this written study a plan with the detailed steps to carry out this project will be made. Future related studies to this proposal and conditions that affect the correct deployment will be mentioned too.

Keywords: *anti-cheat, biometry, blockchain, punishments, videogames.*

Dedicatoria

A mis padres, a mi novia, y a mis amigos que me han acompañado durante este proceso tan importante para mí.

Reconocimientos

A la Dra. Marcela Rosalba Samela quien ha sido mi profesora del Trabajo Final de Carrera, por toda la ayuda que me ha brindado durante la realización de este escrito. Además, por haberme dado el coraje para presentar un resumen del escrito en el XXI CIITI 2023, donde obtuvimos el primer puesto.

Al Ing. Pablo Vilaboa quien aceptó ser mi tutor en el inicio de la tesis.

Al Dr. Jorge Kamlofsky por haberme ayudado con documentación muy valiosa para la escritura de este trabajo.

También a la universidad por promover mi crecimiento profesional.

A todos ellos, nuevamente, ¡muchas gracias!

Índice General

Índice de Gráficos

Índice de Tablas

Estructura General del Trabajo Final

Capítulos

El capítulo 1 abarcará la introducción del trabajo final, en el cual estaremos explicando la problemática a resolver, describiendo brevemente la propuesta de intervención y la justificación de esta. Se detallará el marco institucional a quien se encuentra dirigida la propuesta y los objetivos generales y específicos que desprenden de esta última. En el capítulo 2, se estará abordando el marco teórico y el estado del arte en el que se apoyará la propuesta de intervención. El capítulo 3, será destinado en su totalidad, al desarrollo de la propuesta de intervención.

Anexos

La sección de anexos estará compuesta por información adicional que ayudará a comprender la propuesta de intervención.

Acrónimos

Los acrónimos utilizados durante este trabajo estarán enumerados y explicados en este apartado.

Referencias

En la sección de referencias encontraremos toda la bibliografía consultada y citada que forma parte de este trabajo.

Capítulo 1 – Introducción

1.1 Descripción del problema

En la actualidad tanto los jugadores como los propietarios del software de videojuegos online son amenazados constantemente por los usuarios que presentan un comportamiento tramposo (Duh y Chen, 2009, p. 567), obteniendo cierta ventaja ilegítima sobre los demás usuarios. Esto afecta negativamente a toda la comunidad de videojuegos online.

Según Lehtonen (2022), cuando los usuarios perciben la presencia de numerosos usuarios tramposos, la jugabilidad se degrada a tal punto, que muchos de los usuarios deciden finalmente abandonar el juego u optar por jugar un juego de la competencia. El punto más importante por destacar es el problema que deben afrontar los stakeholders, debido a que perder usuarios en la plataforma, se traduce en grandes pérdidas económicas y deriva en un serio impacto en el negocio. Por lo mencionado anteriormente, es que se debe tener especial cuidado con este aspecto, para asegurar la continuidad del producto software y de la compañía en el tiempo.

Existen múltiples investigaciones acerca de las diferentes trampas que realizan los usuarios en los videojuegos, de los sistemas antitrampas actuales (Van de Ven, 2023) y también del comportamiento de los individuos en este mismo ámbito (Consalvo, 2007). Sin embargo, poco se habla de las penalizaciones, y más específicamente de la efectividad de cada una.

Hoy en día se dispone de diferentes métodos para sancionar a las personas que realizan trampa al jugar videojuegos en línea. Entre los que podemos enumerar la desconexión de la sesión del usuario, la suspensión de cuenta (permanente o temporal), el bloqueo de dirección IP, y el bloqueo de dirección MAC (prohibición por hardware), entre otros (Van de Ven, 2023, p. 17).

Todos estos métodos pueden ser implementados por las empresas propietarias del software, por los desarrolladores de los videojuegos o por empresas terceras desarrolladoras de software a través de un sistema antitrampas. Por ejemplo, a través del Valve Anti-Cheat System (VAC) (Valve Corporation, s.f.), con el Riot Vanguard (Riot Games, 2020), o mediante el Easy Anti-Cheat (Epic Games, 2023).

Los sistemas antitrampas mencionados se encargan de forma automática de la prevención, detección y penalización de los usuarios tramposos, ofreciendo diferentes

escenarios como la implementación del sistema del lado del cliente o en el servidor (Van de Ven, 2023, p. 7). Si bien cada tipo de sanción ofrece diferentes ventajas, lo que tienen en común estos métodos es que permiten de alguna manera la reincidencia de los usuarios infractores. Basta con crearse una cuenta nueva, cambiar la dirección IP utilizada, conseguir un nuevo hardware o computadora, para volver a conectarse y continuar haciendo uso del software de forma ilegítima.

1.2 Justificación de la propuesta

La propuesta de intervención busca ser un complemento a los sistemas antitrampas actuales. Sin ánimos de reemplazarlos, debido a que en primer lugar estos sistemas deberán detectar a los usuarios infractores, para luego ser reportados en un sistema parcialmente descentralizado (Zheng et al., 2017, p. 559). Buscando desalentar cualquier actividad en los usuarios que esté relacionada estrechamente con el uso de trampas, el uso de software de terceros, la explotación de bugs, o cualquier otro aspecto que permita obtener una ventaja ilegítima en el videojuego, sobre el resto de los usuarios.

El sistema de penalización parcialmente descentralizado resguardará los datos biométricos de los usuarios de las plataformas adheridas y contará con información adicional acerca de los antecedentes. El conjunto de esta información será oportuno y de utilidad para la toma de decisiones de los stakeholders, asegurando la continuidad del producto software, manteniendo a los usuarios activos y evitando posibles pérdidas económicas.

El hecho de que un usuario sea reportado en esta plataforma implica que todas aquellas entidades que se encuentren vinculadas a este consorcio disponen de información acerca del usuario. De esta forma, las empresas asociadas a este sistema podrán consultar el historial de las personas y si han sido marcadas por otro sistema antitrampas en el pasado, en otras palabras, si los usuarios han realizado algún tipo de trampa. En caso afirmativo, los propietarios del software podrán reservarse el derecho de admisión de los usuarios en su plataforma.

1.3 Marco Institucional

Este trabajo está destinado principalmente a las empresas desarrolladoras de videojuegos y a otros stakeholders, que busquen frenar el abuso por parte de los usuarios tramposos y de mantener la comunidad limpia.

1.4 Objetivos del Trabajo Final

1.4.1 *Objetivo General*

Proponer un sistema de penalización parcialmente descentralizado para los videojuegos online, basado en la tecnología Blockchain y en el resguardo de los datos biométricos.

1.4.2 *Objetivos Específicos*

Para poder satisfacer el objetivo general, será necesario cubrir los siguientes aspectos:

- Indicar cómo el sistema se integrará a los sistemas antitrampas actuales.
- Presentar un plan para implementar el sistema de penalización.
- Explicar las limitaciones que dificultarían la implementación y el funcionamiento de la propuesta mencionada.
- Promover a los stakeholders a adoptar este sistema basado en Blockchain, para reportar y castigar de forma permanente a usuarios conflictivos.
- Desalentar a los usuarios a realizar trampas en los videojuegos online.

1.5 Contribuciones Principales

Esta propuesta brindará una herramienta para mantener la comunidad limpia, ayudando a los propietarios del software a detener a los usuarios tramposos, y minimizando el impacto en el negocio por las posibles pérdidas económicas. Implementar este sistema, resultará en un espacio colaborativo donde diferentes empresas podrán apoyarse mutuamente para lidiar con este tipo de usuarios.

Capítulo 2 – Marco Teórico

2.1 Los Sistemas Antitrampas en Videojuegos

2.1.1 *Introducción*

Según Lehtonen (2020), los *cheats* (trampas) en los videojuegos online y más específicamente las personas que hacen trampa al momento de jugar en línea han sido siempre un problema que aqueja a toda la comunidad. En la actualidad existen diferentes sistemas de software que se encargan de la prevención, detección y penalización de estas prácticas. Estos sistemas son mejor conocidos como los sistemas *anti-cheat,* o sistemas antitrampas. El desafío principal que tienen los sistemas antitrampas, es que deben poder detectar y neutralizar continuamente las nuevas trampas que los usuarios realizan en el juego. Para tal fin, los desarrolladores de los sistemas anti-cheat deben proveer actualizaciones de manera periódica (p. 1).

2.1.2 *Resumen*

En este capítulo estaremos abordando el concepto de trampa, los diferentes tipos de trampas conocidos, y también comprenderemos cómo funcionan los sistemas antitrampas que buscan solventar estos inconvenientes. Para culminar, se revisarán los diversos tipos de penalidades que existen, y que los propietarios del software aplican sobre los usuarios.

2.1.3 *¿Qué se considera Trampa o Cheat en Videojuegos?*

Cuando en este trabajo nos referimos a las trampas en videojuegos, nos referimos específicamente a los videojuegos multijugador en línea. Tal como lo comentan Duh y Chen (2009), hoy en día la mayoría de las personas optan por jugar en línea, en comparación a como lo era en el pasado, que solamente competían contra la computadora (contra la "IA" o inteligencia artificial). Esto se debe a que las personas encuentran mucho más interesante el juego cooperativo y competitivo online, en comparación con el viejo paradigma local (p. 567). Por el motivo mencionado anteriormente, es que abordaremos los diferentes escenarios que se presentan en el ámbito online.

Cada compañía desarrolladora de videojuegos puede adoptar diferentes posturas con respecto a lo que se considera o no, comportamiento tramposo. Según Duh y Chen (2009) esta falta de consistencia se debe a tres razones, primero que se trata de una problemática que no ha sido abordada en profundidad por los investigadores, segundo que diferentes tipos de juegos implican diferentes formas de hacer trampa, y tercero que nuevas trampas son desarrolladas en

el momento que los sistemas antitrampas logran neutralizar a las anteriores (p. 568). Por otro lado, los jugadores se ven influenciados por diferentes factores subjetivos y por sus propios valores para definir el concepto de trampa (Consalvo, 2007, p. 87).

En este trabajo en particular vamos a considerar como trampa, a cualquier uso de software externo, modificación del software cliente o servidor para obtener una ventaja desleal sobre el resto de los jugadores, como también así cualquier aprovechamiento de alguna falla en el software (*bug*) o manipulación de la red. El tipo de trampa más conocido por los usuarios es el uso de software externo, que permite modificar el comportamiento del cliente del juego en cuestión (Lehtonen, 2020, p. 11). Del punto de vista técnico, hay dos elementos computacionales que se pueden manipular con estas herramientas: la memoria RAM *(Random Access Memory*) o la red. (Van de Ven, 2023, p. 6).

La relevancia de cada trampa está relacionada con el tipo de juego que el usuario esté ejecutando (Lehtonen, 2020, p. 8). Por ejemplo, en los juegos de disparos en primera persona (*First Person Shooters o FPS*), el llamado *wallhack*, se refiere a la modificación de las texturas gráficas del juego permitiendo ver a través de las paredes y pudiendo detectar enemigos, cuando en realidad no debería suceder (Van de Ven, 2023, p. 35). Según Chen y Duh (2009), el aumento en los reflejos consiste en reemplazar mediante un programa o software externo, las reacciones humanas para producir resultados superiores. Un ejemplo de lo que mencionan los autores, es el *aimbot* o *aimhack*, un software que permite apuntar a los enemigos de forma automática (p. 568).

Por otro lado, tal como comenta Khalifa (2016), los servidores de videojuegos online y en particular los que corresponden a los juegos de disparos de primera persona, disponen de un mecanismo para compensar aquellos retardos en la comunicación entre el cliente y el servidor, esto es conocido como la predicción del lado del cliente (*client-side prediction)*, en la que no es necesario esperar la respuesta del servidor, ante cualquier *input* o entrada de información por parte del usuario. Además, el autor indica que, al recibir la respuesta del servidor, es posible corregir aquellas predicciones erróneas (p. 11). Esto permite que se registre un evento que tuvo lugar en el pasado y que no debió ser registrado, pero que sirve para dar una sensación de fluidez al videojuego. Dentro del paquete de información que viaja hacia el servidor, existe un campo que determina la latencia del jugador. Un usuario avanzado podría manipular estos valores e indicarle al servidor que está experimentando latencia alta o retardo, para registrar valores que no debería. Esto es conocido como latencia falsa (*Fake-Lag*) (Van de Ven, 2023,

p. 6). Volviendo al ejemplo de los juegos FPS y más específicamente al Counter-Strike de Valve Corporation (Counter Strike, 2023), el servidor registra de todas formas los disparos que han impactado sobre un jugador que, en realidad, ya se encuentra escondido detrás de una pared.

Es importante destacar que, las trampas mencionadas anteriormente, no son de gran importancia en otros juegos, como por ejemplo en los juegos por turnos. En este género de juegos, espiar la red para modificar paquetes podría ser mucho más beneficioso del punto de vista del atacante (Lehtonen, 2020, p. 9).

Además, es necesario establecer un criterio de clasificación para cada una de las trampas según el impacto en la plataforma online. Según Lehtonen (2020) se establecen dos categorías para indicar el grado de severidad de las trampas realizadas por los usuarios. Se distinguen las trampas livianas y las trampas duras. Cuando se habla de trampas livianas, se refiere a la explotación de las mecánicas del juego para obtener una ventaja ilegal sobre los demás jugadores. Estas características mencionadas no han sido programadas de forma intencional por los desarrolladores del videojuego (p. 8). Un ejemplo de lo mencionado anteriormente podría ser aprovecharse de un *bug* (falla) de forma deliberada, generar dinero dentro del juego rápidamente por algún error en el software, o realizar transacciones con dinero real fuera del juego para obtener beneficios dentro, como la compra de ítems.

Por otro lado, según Lehtonen (2020) las trampas duras (*hard cheats)* son aquellas trampas por las cuales los sistemas antitrampas han sido desarrollados. Se refiere específicamente al uso de programas externos para manipular el software cliente, o a la modificación de los paquetes de red que el cliente envía al servidor. Es importante destacar que los programas externos pueden inyectarse por sí mismos en la memoria, para crear nueva funcionalidad invocando a las funciones estándar del juego (p. 11).

2.1.4 *¿Qué es un Sistema Antitrampas?*

Van de Ven (2023) menciona que, con el fin de detener a los usuarios que buscan obtener una ventaja ilegítima sobre los demás, los sistemas antitrampas se encargan de la detección y penalización de los usuarios tramposos de forma automática. Dependiendo de donde operen estos sistemas, es necesario determinar una clasificación. En la actualidad existen diferentes tipos de sistemas antitrampas, pero en general son agrupados en dos grandes

categorías, los que funcionan en el servidor (*server-side*) y los que se encuentran en el cliente (*client-side*) (p. 7).

Según Lehtonen (2020) aquellos que funcionan únicamente en el servidor, se encargan de revisar los paquetes de red provenientes del cliente y aseguran que los datos y el estado del juego sean manipulados correctamente en el servidor. Por otro lado, los sistemas antitrampas que se encuentran en el cliente, operan en la computadora del usuario y envían información al servidor. Dentro de los sistemas antitrampas que se encuentran dentro del servidor, podemos enumerar la revisión de los datos enviados por el cliente, el diseño de aplicaciones utilizando un protocolo resistente a las manipulaciones, la ofuscación del tráfico de la red y el análisis de datos estadísticos (p. 13).

Como regla general, Lehtonen (2020) indica que es importante que el servidor no confíe ciegamente en los datos que le son enviados por el cliente del juego. De no ser así, se podría estar procesando información proveniente de un cliente modificado, con el objetivo de realizar acciones en el servidor de forma malintencionada. Es responsabilidad del servidor realizar los diferentes controles en los datos recibidos, y los desarrolladores deben tener en cuenta diferentes decisiones de diseño para que estos datos no sean manipulados. En ciertas ocasiones el cliente del juego es el encargado de procesar datos y de determinar estados del juego, sin involucrar al servidor. Si bien es cierto que se evita la sobrecarga del servidor y otros problemas de performance, los inconvenientes de seguridad que se producen no justifican la utilización de este modelo de procesamiento. Alguna de las razones por las cuales el desarrollador no implementa lógica de control exhaustiva en el servidor y la delega mayormente al cliente, puede deberse a problemas de conectividad en la región en la que el software es utilizado, o por la confianza desmedida que se les otorga a los dispositivos que ejecutan el software. (p. 13). Con respecto a los problemas de conectividad, el autor se refiere a que en la región que se utiliza el juego, es sabido de antemano que no se tiene una infraestructura de Internet de alta velocidad que garantice el correcto funcionamiento del software, por lo tanto, el desarrollador toma la decisión en la etapa de diseño de implementar la mayor parte de la lógica del juego en el cliente para asegurar un nivel de jugabilidad y performance aceptables.

Con lo mencionado anteriormente, es posible limitar el tráfico de información desde el cliente al servidor y viceversa, con solamente aquellos datos necesarios para el proceso involucrado. Además, cuando se refiere a la confianza de dispositivos, debemos recordar que los celulares o tablets también se consideran computadoras, y que pueden ejecutar software de

la misma manera. El problema es que algunos desarrolladores creen que estos dispositivos son más difíciles de manipular, cuando en realidad es posible desarrollar trampas mediante aplicaciones de terceros que se ejecuten en el dispositivo involucrado (Lehtonen, 2020, p. 13). A continuación, se revisarán los beneficios de cada protocolo de comunicación.

Según Lehtonen (2020) es importante determinar el protocolo a utilizar para el envío y recepción de datos entre el cliente y el servidor. Por un lado, TCP (*Transmission Control Protocol*) (Postel, 1981) fue diseñado para enviar datos en bloques, estableciendo una secuencialidad en los paquetes, por lo tanto, no pueden enviarse dos paquetes con la misma secuencia. Además, en cuanto un paquete no puede ser enviado, los siguientes paquetes se bloquean sin poder ser enviados hasta que se envíe el paquete anterior. Esto podría resultar en un problema de performance en los videojuegos que involucran un tráfico constante entre el cliente y el servidor. En el caso de UDP (*User Datagram Protocol*) (Postel, 1980), esta secuencialidad no existe, y la lógica de control debe ser implementada por el programador. De todas formas, el protocolo TCP brinda mayor seguridad antitrampas en comparación con el protocolo UDP (p. 19).

Además, de optar por TCP o por UDP, la característica más importante de una aplicación inteligente es que la información que viaje en los paquetes, a través de la red, debe ser la mínima necesaria para establecer ciertos estados del juego y nada más. El hecho de incluir más información de la necesaria podría dar lugar a ser manipulada por un usuario tramposo para anticipar ciertos eventos que en realidad, no debería conocerlos (Lehtonen, 2020, p. 20).

Para que los paquetes que viajan por la red no sean capturados y leídos fácilmente por cualquiera que lo desee, es necesario establecer alguna técnica de encriptación en la información que viaja del servidor hacia el cliente. Luego el cliente deberá incluir internamente la función inversa para poder desencriptar los datos y hacer uso de ellos. El concepto explicado anteriormente, se lo denomina ofuscación del tráfico de la red (Lehtonen, 2020, p. 23).

Finalmente, el análisis de los datos estadísticos es otra alternativa de sistema antitrampas en los videojuegos online, que es establecida en el servidor. Según Lehtonen (2020) consiste en la recolección y revisión de los datos generados por los diferentes eventos en el servidor, que aportan información significativa para determinar si un usuario se encuentra o no realizando trampa, basándose en la desviación del puntaje del jugador en comparación con el promedio global. En caso de que el sistema haya detectado una desviación significativa,

el usuario es marcado para la posterior evaluación por parte de un revisor humano (p. 28). Una de las características más importantes por destacar, es que no se trata de un método intrusivo, sino que solamente se basa en la información generada por los eventos del videojuego. Por ejemplo, en los juegos de disparos como en el Counter-Strike, se analiza la cantidad promedio de enemigos abatidos por partida, el promedio del daño recibido o realizado, y demás datos relacionados con el puntaje del jugador. El conjunto de estas variables conlleva a marcar como sospechoso a un usuario, el cual es derivado a revisión manual por parte de algún miembro del equipo de revisión. Finalmente, el encargado de analizar al usuario es quien dictamina si el usuario se trata de un tramposo o no, y emite un juicio en el que prohíbe o no al usuario, de continuar utilizando la plataforma.

Hasta ahora se han comentado los diferentes tipos de sistemas antitrampas que se ejecutan del lado del servidor, a continuación, se hará una revisión bibliográfica de los sistemas antitrampas que se encuentran del lado del cliente.

Hoy en día los sistemas antitrampas que se ejecutan en el cliente son los más utilizados por las empresas desarrolladoras de videojuegos para detectar trampas en la computadora del usuario, ya que, con los métodos mencionados anteriormente del lado del servidor, resulta difícil detectar ciertos tipos de trampas. Tal como comenta Van de Ven (2023), al delegar la responsabilidad de detección de usuarios tramposos a la computadora cliente, el servidor cuenta con más recursos disponibles, mejorando la performance general del servicio. El problema que presentan estos sistemas es que las revisiones de seguridad se realizan completamente en el equipo que ejecuta el videojuego, por lo tanto, un usuario experimentado podría analizar y burlar estos controles con el fin de ejecutar programas externos o realizar modificaciones en el cliente (p. 7).

Según Lehtonen (2020) dentro de los métodos antitrampas que se ejecutan del lado del cliente, podemos enumerar la encriptación del código fuente, la verificación de archivos mediante la codificación criptográfica (*hash)*, la detección de trampas conocidas, la ofuscación de la memoria y los sistemas antitrampas basados en controladores de kernel (p. 33). El uso o combinación de alguno de los métodos mencionados ayuda a prevenir el uso de trampas en el cliente. La encriptación del código fuente se refiere a disponer del cliente del juego de forma protegida y solamente desencriptar las partes de este, a medida que el juego es ejecutado (Lehtonen, 2020, p. 34). Esto hace el programa más difícil de ser interpretado y manipulado por el usuario. La verificación de archivos mediante alguna técnica criptográfica sirve para

verificar si alguno de los archivos del cliente sufrió alguna modificación y ya no se encuentra en su estado original (Lehtonen, 2020, p. 44). Algunos usuarios podrían realizar alteraciones en el cliente para modificar bibliotecas (*DLLs*) manipulando la memoria y el comportamiento del juego. Esta revisión se realiza en primer lugar en cuanto se ejecuta el juego. En caso de que el cliente detecte alguna anomalía, detendrá la ejecución del juego inmediatamente y solicitará revisar los archivos o descargar nuevamente el cliente del juego.

Tal como lo indica Lehtonen (2020), la detección de trampas conocidas consiste en la revisión de la computadora del usuario en busca de programas externos, basados en cierta firma, que se encuentren ejecutándose del lado del cliente, y que se traten de programas para realizar trampas en los videojuegos. La manera más simple de aplicar este método es la comparación de los nombres de los procesos y los hashes obtenidos a partir de estos (p. 48). Es importante destacar que las empresas desarrolladoras de videojuegos no hacen público el completo funcionamiento de sus sistemas antitrampas, para no otorgar información sensible a los hackers y demás usuarios tramposos que busquen denegar a estos sistemas (Van de Ven, 2023, p. 8). Por el motivo comentado anteriormente es que estos sistemas antitrampas podrían ser más intrusivos dependiendo de cada desarrollador, ya que se podría compartir más información que la necesaria para la detección de trampas. En el pasado, el sistema antitrampas del tipo client-side utilizado por Blizzard Entertaiment, llamado *The Warden*, fue descubierto por realizar diferentes tipos de escaneos en la computadora del usuario (Ward, 2005), en la que se recopilaba información acerca de los procesos en ejecución para verificar si se trataban de programas para realizar trampas conocidas, pero además se analizaban los títulos de las ventanas o de cualquier otro programa. Aun conociendo cómo el sistema implementado por Blizzard funcionaba, muchos de los usuarios han estado de acuerdo con este método para detener a los usuarios tramposos. Valve Corporation también ha tenido problemas con la implementación del sistema VAC (Valve Anti-Cheat System), en el 2014 la empresa fue acusada por escanear y recopilar la información del caché del DNS (Domain Name System), un registro con todas las direcciones web accedidas desde las computadoras de los usuarios (Amazon Web Services, s.f.) y de enviarla a los servidores de Valve. Gabe Newell, el CEO de la empresa, tuvo que explicar que la información recopilada estaba destinada solamente a la verificación del uso de trampas que funcionaban a nivel kernel (núcleo) del sistema operativo. Más específicamente lo que Valve buscaba era la dirección de los servidores DRM (Digital Rights Management) consultados. Según Lehtonen (2020) los servidores DRM son utilizados

para validar la compra y el licenciamiento de cierto software en particular. En este caso se buscaba si habían sido contactados servidores que tenían que ver con la venta de software destinado al uso de trampas, aunque en este mismo escaneo, también se revisaba toda aquella información referida al historial de navegación del usuario (p. 50). Por el motivo mencionado es que Valve ha tenido algunas disputas en cuanto a la privacidad de los usuarios (IEEE Spectrum, 2010).

Muchos de los sistemas antitrampas actuales funcionan como cualquier programa estándar y se ejecutan en el espacio del usuario, restringiendo la habilidad de proteger cualquier videojuego (Lehtonen, 2020, p. 58). Esto quiere decir que solamente disponen de permiso para acceder a ciertas áreas de la memoria y del sistema operativo en el que se desenvuelven. Tal como se visualiza en la Tabla 1, los sistemas antitrampas que funcionan en el espacio del usuario se encuentran mucho más limitados en comparación con aquellos que operan como controladores del kernel o en el núcleo del sistema operativo. Además, el autor indica que, los sistemas antitrampas que trabajan a nivel de kernel, tienen acceso a cualquier sector de la memoria permitiendo detectar cualquier anomalía que esté relacionada con la manipulación y el uso de trampas.

Tabla 1

Comparación de privilegios entre el espacio del usuario y el kernel.

Privilegio	Espacio del Usuario	Espacio del Kernel
Acceso a la memoria	Acceso Limitado	Acceso Completo
Acceso al hardware	Sin acceso directo	Acceso Completo
Acceso a las instrucciones del CPU	Solo a instrucciones sin privilegios	Todas las instrucciones
Acceso a estructuras críticas del SO	Sin acceso	Acceso Completo

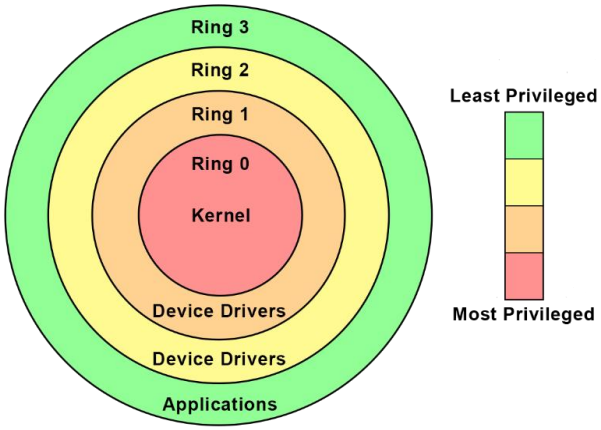
Nota. Adaptado de Lehtonen (2020, p. 58)

Según Lee et al. (2018), los sistemas operativos modernos en la arquitectura x86 adaptan dos modelos de niveles de privilegio, en los cuales los programas del usuario se ejecutan en el anillo 3 (espacio del usuario) y en el anillo 0 (núcleo o kernel). La arquitectura x86, de hecho, soporta cuatro capas de privilegios, desde la 0 hasta la 3, siendo el anillo 0 el de

mayor privilegio y el anillo 3 el de menor (p. 1442). Como podemos ver a continuación en la Figura 1, se muestran los anillos de privilegio en la arquitectura x86.

Figura 1

Imagen de los anillos de privilegio del sistema operativo de la arquitectura x86.



Nota. Obtenido de */dev/null: Anti-cheat kernel driver* por Riot Games, s.f, ContentStack (https://images.contentstack.io/v3/assets/blt731acb42bb3d1659/blt28df09c60292cc6c/5e3101fd14bf1c024132f20e/Kernel_Drivers_Image.png)

Un ejemplo de sistema antitrampas que opera en el nivel cero como un controlador de kernel, es el Vanguard de Riot Games (Riot Games, 2020).

2.1.5 *¿Qué son y cómo se aplican las Penalizaciones en los Videojuegos?*

Las penalizaciones se refieren a las medidas que toman las empresas desarrolladoras de videojuegos o el operador del juego, contra los usuarios que realizan trampas. Según Van de Ven (2023) el factor disuasorio más obvio de todos es recibir un castigo por realizar trampas en los videojuegos online. Los castigos determinados abarcan desde pequeñas penalidades como cuentas silenciadas o prohibiciones temporales, hasta bloqueos de dirección IP y prohibiciones por hardware, incrementando la severidad según el número de ofensas cometidas anteriormente (p. 17). El problema que deben afrontar los stakeholders además de las pérdidas económicas producidas por el abandono de los usuarios en la plataforma, es la reincidencia de los usuarios tramposos debido a la deficiencia de los métodos mencionados.

La suspensión de cuenta se refiere a una prohibición temporal o permanente de acceso del usuario a la plataforma mediante un *flag* (bandera o marca) a nivel base de datos, para indicar que el usuario está suspendido. Para volver a ingresar a la plataforma, los usuarios

necesitan crearse una cuenta nueva asociado a otro email, o esperar a que el tiempo de penalización acabe. El bloqueo de dirección IP, se refiere a prohibir las conexiones entrantes desde cierta ubicación (IP) otorgada por el ISP (Internet Service Provider). Los usuarios pueden utilizar una VPN (Virtual Private Network) para saltar este inconveniente o solicitar al ISP para renovar esta dirección. La prohibición por hardware hace referencia al bloqueo de alguno, o de todos los componentes utilizados en la computadora origen de la infracción. La más común se refiere a la prohibición de MAC Address (Media Access Control) que identifica unívocamente a la placa de red (Ohaio State University, [OSU], 2021). Finalmente, la desconexión del usuario siendo la más leve, se refiere al cierre de la conexión entre el cliente y el servidor. En este último caso, es necesario reejecutar el software para hacer uso de él nuevamente.

2.2 La Tecnología Blockchain

2.2.1 *Introducción*

Como se ha comentado anteriormente en este trabajo, las penalizaciones que los desarrolladores de los sistemas antitrampas o los operadores del juego determinan sobre los usuarios, permiten de alguna manera la reincidencia de ellos en las diferentes plataformas de videojuegos online. Por lo que se necesita de una solución que permita neutralizar esta problemática. La propuesta de intervención en el campo profesional que se desarrollará en el capítulo número tres estará basada en la tecnología Blockchain, aprovechando las ventajas de las características que otorgan estos sistemas.

2.2.2 *Resumen*

Para poder comprender cómo Blockchain puede apoyar a los stakeholders en la toma de decisiones, en primer lugar, es necesario comprender qué es y cómo funciona esta tecnología. Por otro lado, se diferenciarán los diferentes tipos de sistemas Blockchain que existen, las características y ventajas de cada uno haciendo énfasis en los sistemas parcialmente descentralizados (*consortium blockchain*), y los contratos inteligentes que permitirán la ejecución de lógica de programación, acorde al modelo a utilizar.

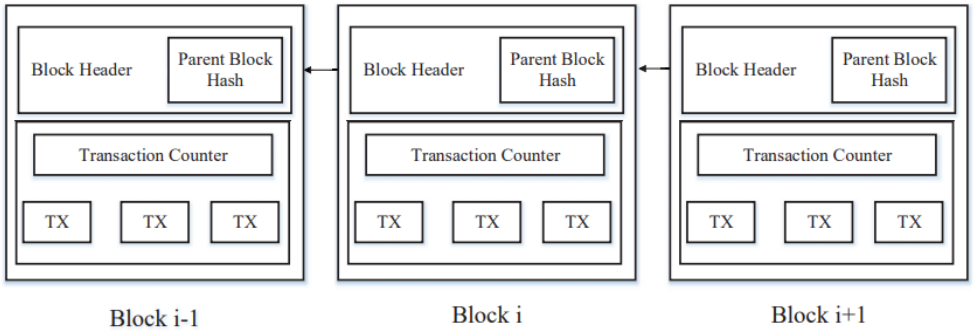
2.2.3 *Conceptos Principales*

Según Nakamoto (2008), el concepto de Blockchain fue introducido por primera vez como una solución entre pares (*peer-to-peer*), para poder enviar y recibir pagos directamente sin necesidad de la participación de un intermediario, como lo es el banco o cualquier institución financiera. La primera implementación de esta tecnología fue la criptomoneda Bitcoin (p. 1). El blockchain puede comprenderse como un gran libro contable público y

distribuido, que guarda todas las transacciones asentadas en una lista de bloques. La cadena crece a medida que nuevos bloques son agregados a la misma constantemente (Zheng, et al., 2017, p. 557). En la Figura 2, se muestra la arquitectura Blockchain.

Figura 2

Arquitectura Blockchain

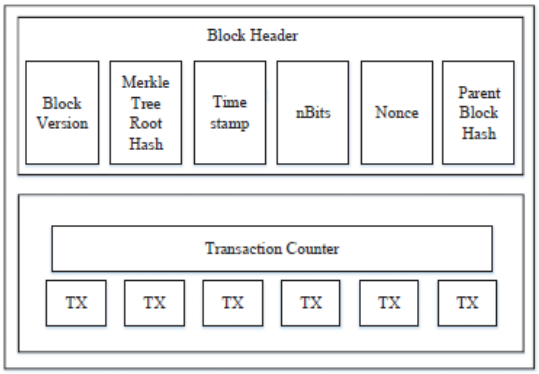


Nota. Obtenido de *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*, (p. 558), por Zheng et al., 2017.

Las ventajas del uso de la tecnología Blockchain está determinada por características específicas, entre las que podemos nombrar, la descentralización, persistencia, el anonimato y la auditabilidad (Zheng et al, 2017, p. 557). A continuación, se muestra la Figura 3, con la estructura de un bloque en blockchain.

Figura 3

Estructura de un bloque en Blockchain



Nota. Obtenido de *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*, (p. 558), por Zheng et al., 2017.

Según mencionan Zheng et al. (2017) el bloque actual tiene en su cabecera el hash del bloque anterior, esto resulta en que cada bloque tiene solamente un bloque padre. El primer bloque de la cadena se denomina bloque génesis, este no tiene antecesor. Un bloque está compuesto por su cabecera y su cuerpo, dentro de la cabecera podemos enumerar los siguientes componentes: la versión del bloque, el hash del árbol de Merkle, la fecha, una cantidad de bits "n", un número para una única utilización (*nonce*), y el hash del bloque padre. El cuerpo del bloque está compuesto por las transacciones y por un contador (p. 558). El autor se refiere al hash como al resultado de una operación criptográfica. Tal como lo menciona Pierro (2017), un hash puede comprenderse como una versión encriptada de una cadena original, desde la cual no es posible calcular el valor que tenía originalmente (p. 93).

Tal como lo indican Zheng, et al. (2017), dentro de las características principales que provee la tecnología Blockchain, se tiene la descentralización, donde no se requiere involucrar a terceros en las transacciones, contrariamente a como sucede en los sistemas centralizados. La persistencia, que permite que las transacciones puedan ser validadas rápidamente y las que son inválidas no sean admitidas por los mineros honestos, siendo casi imposible eliminar o reversar transacciones una vez que estas han sido incluidas en la cadena de bloques. Además, el autor agrega que cada usuario puede interactuar con el sistema blockchain, con una dirección generada, la cual no revela la identidad real del usuario, promoviendo la anonimidad. Finalmente, blockchain permite ser auditable, ya que cada transacción está relacionada con una transacción anterior (p. 559).

2.2.4 *Los Algoritmos de Consenso*

En blockchain para poder establecer consenso entre los diferentes nodos sin confianza, es necesario que ellos estén de acuerdo con respecto a las validaciones que debe cumplir la información. Este inconveniente se lo considera un problema bizantino, tal como comentan Zheng et al. (2017) en el que un ataque fallará solamente si ciertos generales atacan la ciudad, mientras que otros deciden no hacerlo, de forma análoga en blockchain si los nodos no llegan a un consenso mediante la validación de las transacciones, éstas no son adheridas al bloque (p. 559). Existen diferentes algoritmos para lograr consenso entre los participantes de un sistema blockchain, a continuación, se mencionan las más importantes.

PoW (Proof-of-Work). El protocolo PoW (*Proof-of-Work*) o prueba de trabajo, es el utilizado por el sistema blockchain de Bitcoin. Según Zheng et al. (2017) en una red descentralizada, alguien debe ser el encargado de registrar las transacciones. Si bien la forma

más simple de realizarlo es seleccionando un nodo al azar, esto haría que la red se encuentre vulnerable a ataques. Por tal motivo, si un nodo desea publicar un bloque de transacciones, una gran cantidad de trabajo es requerida, para probar que el nodo no busca atacar la red. Este trabajo se refiere a cálculos computacionales (p. 560). Como menciona Nakamoto (2008), el protocolo involucra el escaneo de un valor que cuando es sometido al proceso de conversión a hash, utilizando el algoritmo SHA-256, el valor obtenido del hash debe comenzar con un número de cero bits. El esfuerzo necesario es exponencial al número de bits cero requeridos y puede ser verificado ejecutando una operación hash simple (p. 3)

PoS (Proof-of-Stake). Según Zheng et al. PoS (*Proof-of-Stake*) o Prueba de Participación, es una alternativa más costo-eficiente en comparación con PoW, en la que los mineros deben probar ser dueños de cierta cantidad de divisas. El autor menciona que se cree que las personas con la mayor cantidad de divisas dentro del sistema blockchain son menos propensos a atacar la red, aunque esta selección sea injusta, ya que una única persona está destinada a ser quien domine la red. Por tal motivo, además indica que varias soluciones han sido propuestas combinando esta alternativa para decidir quién de los nodos será el que genere el próximo bloque en la cadena (p. 560).

PoA (Proof-of-Authority). Según Asad et al. (2020) PoA *(Proof-of-Authority)* o Prueba de Autoridad es un algoritmo de consenso en blockchain que propone una solución efectiva y eficiente para resguardar la privacidad de los datos en la red de la cadena de bloques. El algoritmo PoA utiliza el valor de la identidad para imponer seguridad, mediante la selección de antemano de nodos confiables. El sistema es altamente escalable ya que depende de un número restringido de validadores de bloques (p. 35).

En la Tabla 2, a continuación, se realiza una comparación entre los diferentes algoritmos de consenso y sus características principales.

Tabla 2

Comparación de PoA con PoW y PoS

Característica	Proof Of Work	Proof Of Stake	Proof of Authority
Velocidad	La más lenta	Promedio	La más rápida
Consumo de Energía	Ineficiente	Eficiente	Eficiente
Seguridad	Sin permisos, no confiable	Sin permisos, no confiable	Mediante Permisos, confiable
Madurez	Probado	Sin Probar	Seguro
Costo	Costoso	Menos Costoso	Sin Costo

Nota. Adaptado de Asad et al (2020, p. 37).

Por otro lado, existen otros algoritmos de consenso que utiliza el framework de Hyperledger Fabric, para los sistemas parcialmente descentralizados o permisionados (Hyperledger, s.f).

PBFT (Practical Byzantine Fault Tolerance). Según Zheng et al. (2017) PBFT es una replicación del algoritmo de tolerancia de fallas bizantinas, que fue presentado originalmente por Miguel Castro y Barbara Liskov en el año 1999. Actualmente el framework de Hyperledger Fabric utiliza este mecanismo [además del mecanismo Raft] para establecer consenso en los sistemas blockchain de consorcio, ya que puede soportar hasta un tercio de nodos maliciosos en la red. Un nuevo bloque es determinado por ronda. En cada ronda, un nodo primario o líder, es seleccionado de acuerdo con ciertas reglas, siendo este nodo el responsable de ordenar la transacción. El proceso completo se divide en tres fases: prepreparado (*pre-prepared*), preparado (*prepared*) y comprometido (commited). En cada etapa, un nodo puede pasar a la siguiente, solamente si ha recibido los votos de más de dos tercios de la totalidad de nodos que conforman la red. Por tal motivo, en PBFT se requiere que todos los participantes sean conocidos en la red (p. 560).

Raft. Según Ongaro y Ousterhout (2014), el algoritmo Raft implementa el consenso basándose en la elección de un líder distinguido, y luego otorgándole la completa responsabilidad de gestionar la replicación del registro. El líder acepta diferentes modificaciones para replicarlas en los demás servidores, y les indica que es seguro aplicar los registros en su base local. El hecho de tener un líder, simplifica la gestión de la replicación del

registro. Por ejemplo, un líder puede decidir dónde asentar nuevos registros sin consultar otros nodos, y el flujo de la información se realiza de una forma simple desde el líder a los demás nodos. En caso de que un líder falle o se desconecte de la red, uno nuevo es elegido (p. 307).

Para que el algoritmo sea capaz de resolver la funcionalidad que se mencionó en el párrafo anterior, es necesario comprender como se compone el método. Tal como mencionan Ongaro y Ousterhout (2014), Raft está constituido por un grupo de servidores o nodos, siendo cinco un número típico, en el que se permiten hasta dos fallas del sistema. En un momento dado cada nodo se encuentra en uno de los tres estados: líder (*leader*), seguidor (*follower*) o candidato (*candidate*). Durante la operación normal, existe un solo líder y todos los demás nodos pasan a ser seguidores. Estos seguidores adoptan un carácter pasivo, es decir, no emiten transacciones por sí mismos, y solamente responden a las peticiones del líder. El líder gestiona todas las peticiones, y en caso de que un cliente reciba una petición, la reenvía al líder vigente. El estado candidato es únicamente utilizado para la etapa en la que se necesite elegir un nuevo líder (p. 309).

2.2.5 *¿Qué es un Sistema Parcialmente Descentralizado?*

Para poder optar por un tipo de implementación blockchain, es necesario comprender primero cómo funciona cada una, junto con sus características específicas. Existen diferentes tipos de implementaciones de Blockchain, cada una de ellas denotan características con respecto a la descentralización, la eficiencia, inmutabilidad, los permisos, y al mecanismo de consenso (Zheng, et al., p. 559). Se puede definir tres tipos de categorías de sistemas blockchain, las públicas, las privadas y las de consorcio. Como mostrado en la Tabla 3 a continuación, con un resumen de las características de cada una de las implementaciones.

Tabla 3

Comparación entre blockchain pública, de consorcio y privada.

Propiedad	Blockchain Pública	Blockchain de Consorcio	Blockchain Privada
Determinación del consenso	Mediante todos los mineros	Nodos seleccionados	Una sola organización
Permiso de lectura	Pública	Pública o restringida	Pública o restringida
Inmutabilidad	Casi imposible de manipular	Puede ser manipulable	Puede ser manipulable
Eficiencia	Baja	Alta	Alta
Centralizado	No	Parcialmente	Si
Proceso de consenso	Sin Permisos	Mediante Permisos	Mediante Permisos

Nota. Adaptado de Zheng et al., (2017, p. 559).

En este trabajo en particular, la implementación se basará en un sistema blockchain de consorcio, en el que las empresas interesadas podrán asociarse para cooperar.

Según Zheng et al. (2017) con respecto a la taxonomía, en las implementaciones de blockchain públicas, la información se encuentra visible para cualquiera que quiera acceder a los datos, y todos pueden formar parte del proceso del consenso. Es diferente en el caso de las privadas, ya que un único ente regula el funcionamiento de esta y solo un grupo perteneciente de nodos de cierta organización pueden participar en el proceso de consenso y de acceso a la información. Finalmente, los blockchain de consorcio (*consortium blockchains*), corresponden a un grupo selecto de nodos que pueden participar en el proceso de consenso y acceso a datos, pero estos nodos corresponden a diferentes organizaciones (p. 559). Tal como se mostró en la Tabla 3 anteriormente, la diferencia principal entre estos tres tipos de sistemas es que los sistemas blockchain públicos son completamente descentralizados, los privados al ser operados por una única empresa son centralizados y los de consorcio son parcialmente descentralizados, ya que un grupo de organizaciones participa de este consenso. De la misma forma los permisos de lectura también varían entre implementaciones, por ejemplo, en los sistemas públicos cualquiera puede acceder a los datos, en los privados solamente los nodos pertenecientes a la organización, y en los de consorcio aquellas organizaciones que forman parte de este (Zheng et al, 2017, p. 559). Al hablar de inmutabilidad de los datos, es importante destacar que en los sistemas blockchain públicos resulta casi imposible manipular las transacciones, debido a la

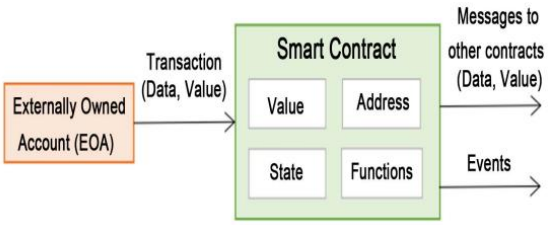
gran cantidad de usuarios que pueden participar del proceso de consenso. Por otro lado, en las privadas y de consorcio, lo mencionado anteriormente puede suceder al tratarse de un grupo reducido de nodos, en el que la mayoría decide corromperse (Zheng et al, 2017, p. 559). En cuanto a la eficiencia de cada una de las implementaciones, en las públicas se demora un gran tiempo en que las transacciones se propaguen a través de la red para ser validadas, debido a la inmensa cantidad de nodos que existen, resultando en una eficiencia baja. De forma contraria en las implementaciones privadas o de consorcio, el tiempo de propagación y validación es menor debido a la cantidad reducida de miembros, por lo tanto, se tiene una latencia baja y una eficiencia alta, como mostrado en la Tabla 3.

2.2.6 *¿Qué es un Contrato Inteligente?*

Con el surgimiento de la tecnología blockchain, se ha demostrado sus aplicaciones en múltiples áreas. La integración de blockchain con los contratos inteligentes, brinda gran flexibilidad para diseñar, desarrollar e implementar varios problemas de la vida real en menos tiempo y costo sin involucrar a un sistema tradicional basado en terceros (Mohanta et. al, 2018, p. 1). La propuesta mencionada en este trabajo requiere de la generación de estos contratos para poder interactuar con el sistema blockchain, sin necesidad de una entidad central controladora. El concepto de contrato inteligente fue mencionado por primera vez en el año 1994 por Mark Szabo para almacenar lógica de programación de forma descentralizada. Un contrato inteligente es un programa informático, que puede verificarse por sí mismo, que se ejecuta automáticamente y es resistente a la manipulación (Mohanta et al., 2018, p. 1). Por el motivo mencionado, es que no es necesario involucrar a terceros en la transacción. Ver Figura 4 a continuación con un ejemplo.

Figura 4

Estructura básica de un contrato inteligente



Nota. Obtenido de *A basic structure of Smart Contract* (p. 1), por Mohanta et al., 2018.

2.3 La Biometría

2.3.1 *Introducción*

Para poder identificar de forma unívoca a los usuarios de las diferentes organizaciones asociadas al sistema blockchain de consorcio, es necesario apoyarse en la ciencia de la biometría. Puntualmente existen diferentes rasgos que pueden ser analizados, como las manos, las huellas dactilares, los rostros, la retina o el iris. “La Biometría es una ciencia que analiza las distancias y posiciones entre las partes del cuerpo para poder identificar o clasificar a las personas” (Serratosa, 2012, p. 5).

2.3.2 *Resumen*

A continuación, se explicarán los conceptos fundamentales de la Biometría, los tipos de sistemas biométricos que existen, los rasgos que deben ser analizados, las cuestiones de ética y privacidad de los datos y los usos frecuentes en los diferentes campos de estudio.

2.3.3 *¿Cómo funciona La Biometría?*

Según Serratosa (2012) Alphonse Bertillon (1853-1914) fue un policía francés que en 1882 presentó el primer sistema de identificación de personas, basándose en las características físicas, al cual denominó antropometría. Este fue el primer sistema científico utilizado por la policía para identificar a los criminales (p. 7).

La antropometría utiliza once medidas de la cabeza y del cuerpo para identificar a las personas. Según Serratosa (2012) para poder conocer la similitud entre dos personas, se debe calcular la distancia euclídea entre los vectores formados por los once componentes de Bertillon. Estas corresponden a la altura, anchura de los brazos extendidos, altura de la persona sentada, largura de la cabeza, anchura de la cabeza, anchura de la oreja derecha, largura de la oreja derecha, largura del pie izquierdo, largura del dedo corazón izquierdo, largura del dedo meñique izquierdo y largura del antebrazo izquierdo (p. 8). Siendo A y B dos vectores que corresponden a las once medidas de Bertillon, y se desea saber si pertenecen a la misma persona, se calcula de la forma que se muestra en la figura a continuación y se considera que las medidas corresponden a la misma persona si el valor obtenido es menor o igual al umbral establecido.

$$D_{\text{Bertillon}}(A, B) = \sqrt{\sum_{i=1}^{11}(A_i - B_i)^2} \quad (1)$$

$$D_{\text{Bertillon}}(A, B) \leq Llindar_{Bertillon} \quad (2)$$

Nota. Fórmula para calcular la distancia euclídea entre dos vectores. Adaptado de *La Biometría para la identificación de personas* (p. 8), por F. Serratosa, 2012. Universitat Oberta de Catalunya.

El método de Bertillon o Bertillonaje fue desprestigiado posteriormente por el caso policial de William y Will West en Kansas (Estados Unidos) en el año 1903, ya que, debido a las similitudes en las características físicas de ambas personas, dio lugar a una confusión en la que se creía que Will West se trataba de un delincuente que se había cambiado el nombre y era William West, pero en realidad Will era una persona diferente. Ver Anexo II con una tabla comparativa y las fotografías de ambos individuos. Sin embargo, hoy en día la distancia euclidiana se sigue utilizando como parte de otros algoritmos de reconocimiento biométrico.

Según Serratosa (2012) el reconocimiento biométrico se refiere al uso de diferentes características anatómicas (como huellas dactilares, cara o iris) y de comportamiento (como habla, firma o la forma de teclear). Estas características se denominan identificadores biométricos o rasgos y sirven para reconocer automáticamente a los individuos. (p. 14). Es importante destacar que existen diferentes técnicas para el reconocimiento biométrico de los individuos basándose en diferentes características físicas. “Sin embargo, incluso con la diversidad de técnicas existentes, a la hora de desarrollar un sistema de identificación biométrica, se mantiene un esquema totalmente independiente de la técnica empleada” (Marín et al., 2009, p. 30). En el próximo subpunto se detallan las etapas comunes en todo sistema biométrico.

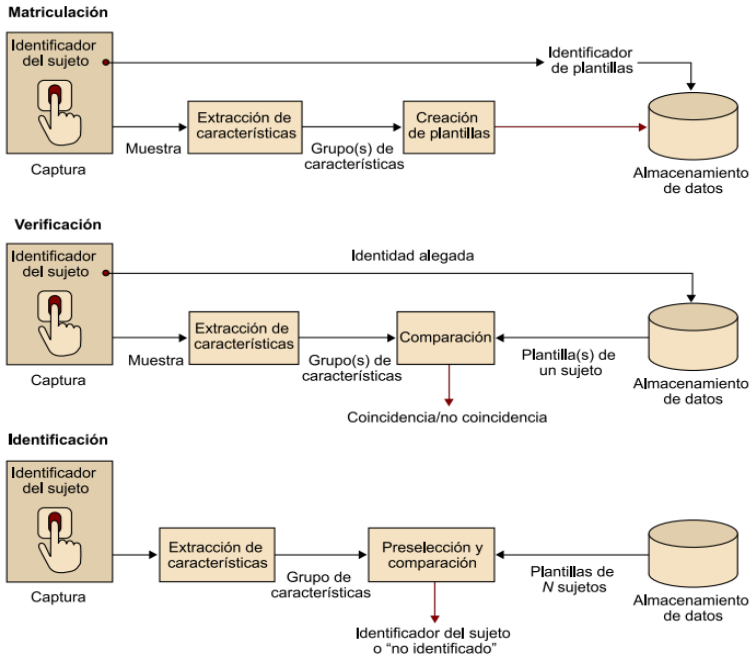
2.3.4 *Los sistemas biométricos*

Los sistemas biométricos se basan en dos fases o etapas diferentes, el reclutamiento y la utilización. Según Marín et al. (2009), en la fase de reclutamiento se toma una serie de muestras del usuario, y se procesan para obtener un patrón que luego se almacena. Este conjunto de datos será el que caracterizará al usuario. En caso de capturar más de una muestra, el patrón será el resultado de obtener la media de la característica obtenida. Este proceso se realiza de forma supervisada, es decir, una persona se encarga de controlar la captura de los

datos y de asegurar la identidad de la persona de la cual se está reclutando los datos biométricos. Por otro lado, los autores definen la etapa de la utilización (verificación o identificación), como la comparación entre las características tomadas en el momento, con aquellas que ya fueron almacenadas en la base de datos durante la etapa de reclutamiento (p. 31). Como mostrado en la Figura número 5.

Figura 5

Etapa de Matriculación, Verificación e Identificación.



Nota. Obtenido de *La Biometría para la identificación de personas* (p. 20), por Serratosa F., 2012, UOC.

Las fases mencionadas anteriormente se descomponen en diferentes procesos que abarcan desde la captura hasta el almacenamiento para poder convertir las muestras en un elemento que sirva para la identificación. Adicionalmente es importante destacar que hoy en día es posible realizar la matriculación a distancia y en forma automática utilizando como dispositivo de captura, la cámara de los teléfonos celulares o de cualquier notebook o computadora.

Según Serratosa (2012), la captura se refiere a la representación digital del rasgo biométrico a capturar. Por lo general el sensor que se utiliza en este proceso, es un sistema de captura de imagen a excepción de la identificación por voz. La información que se captura se llama muestra y en ocasiones se involucran otros periféricos para la introducción de datos no

biométricos o para simplemente mostrar información. La extracción de las características se refiere al procesamiento de la información capturada, para generar un registro de identificación. El objetivo de este registro de información es obtener una representación compacta que facilite la comparación, reducir el ruido, y aumentar la información de la representación original. Luego de la extracción de los datos se procede a la creación de plantillas, que corresponde a la representación resumida de un conjunto de muestras de un tipo de característica biométrica. Este proceso se da únicamente en la matriculación o reclutamiento de características (p. 18).

Una vez tomadas las características, éstas ya se encuentran en condiciones de ser comparadas. Según Marín et al. (2009) las características de la muestra capturada se deben comparar con el patrón o con la plantilla almacenada previamente. Es importante destacar que esta comparación no se trata de una comparación binaria, sino que se trata de una aproximación en base a un puntaje obtenido, que se refiere a la probabilidad de semejanza. Esto se debe a la variación de las muestras por las diferencias en las capturas o por la leve variación de las características del usuario. Por lo mencionado anteriormente, es que se debe establecer un umbral que se refiere a la tolerancia de la probabilidad (p. 31). Durante la etapa de comparación, en algunos casos es necesario realizar algún tipo de filtrado o selección sobre la información almacenada. “En los sistemas de identificación con muchos datos (podemos hablar de 50 millones de huellas dactilares), el filtrado es un método para aumentar el tiempo de respuesta del sistema.” (Serratosa, 2012, p. 19). El autor se refiere a utilizar técnicas conocidas de las bases de datos para realizar consultas rápidamente y evitar los problemas de performance. Hoy en día los sistemas gestores de bases de datos permiten realizar este proceso fácilmente.

Según Serratosa (2012) el almacenamiento de los datos es el proceso necesario para guardar la información del usuario. La información se compone de un identificador único, como el DNI, la plantilla biométrica y datos personales adicionales. La información se guarda en sistemas centralizados o en tarjetas inteligentes. Finalmente es necesario aplicar técnicas de encriptación sobre los datos para que el registro conformado por el DNI y los rasgos biométricos sea inseparable (p. 19). En próximo capítulo de este trabajo, se mostrará la aplicación de la encriptación sobre cada vector de características, para poder ser almacenados en la cadena de bloques.

La Identificación Biométrica puede realizarse mediante dos esquemas, el reconocimiento y la autenticación. Según Martin et al. (2009) el reconocimiento se basa en

identificar a un usuario dentro de una base de datos que contiene todos los usuarios capturados. Por lo tanto, se debe comparar las características extraídas con todos los patrones o plantillas que fueron reclutados en la base de datos. Además, los autores mencionan la autenticación, como un método que responde a la pregunta, ¿es este el sujeto que alega ser? Mediante la autenticación, el usuario también comunica al sistema la identidad que indica ser, además de que se capturan las características biométricas. Luego el sistema realiza la comparación de las características extraídas con la plantilla o patrón señalado. En caso de que la comparación supere el umbral de similitud establecido, se considera que el usuario es quien dice ser. Si la comparación no supera dicho valor, el proceso de autenticación es rechazado (p. 33).

2.3.5 *Los rasgos biométricos*

Los rasgos por considerar para los procesos de reconocimiento y autenticación deben respetar ciertas características. Según Serratosa (2012) el rasgo debe ser universal, es decir, que cada persona debe disponer de ese rasgo biométrico. Al mismo tiempo debe ser lo suficientemente particular en cada individuo y permanecer invariable frente a otros factores, como el tiempo, edad y enfermedades (p. 22). Otros autores como Marín et al. (2009) se refieren a esta última característica como la estabilidad del rasgo (p. 36). Además, Serratosa (2012) indica, que el rasgo requiere ser medible de forma cuantitativa, respondiendo a un rendimiento que se encuentra vinculado con la robustez y precisión del sistema. Finalmente, tiene que ser aceptado por los usuarios para el uso en la identificación biométrica y la falsificación debe ser dificultosa (p. 22). En este trabajo en particular no vamos a profundizar en todos los rasgos biométricos que pueden tomarse, sino que solamente vamos a hacer foco en algunos de los más relevantes.

Según Marín et al. (2009) la huella dactilar, es la técnica más estudiada y probada que se conoce. Se cuenta con diferentes estudios que confirman la unicidad de la huella de un individuo, como también así, la característica de la permanencia en el tiempo. Debido a su antigüedad es una técnica que se encuentra más desarrollada que cualquier otra. La captura puede realizarse con diferentes dispositivos, entre los que podemos enumerar, los ópticos, de estado sólido y ultrasonido. Por otro lado, los elementos que se evalúan para la extracción de características de la huella dactilar son las crestas (*rides*), los valles (*valleys*), y las singularidades como las curvas (*loops*), bifurcaciones (*deltas*) y espirales (*whorls*) (p. 33).

Si bien la huella dactilar se considera la técnica más estudiada y probada, existen otros métodos que facilitan la medición, con más aceptabilidad por parte de los usuarios, aunque en

algunos casos con menor rendimiento del punto de vista del sistema biométrico utilizado. “La textura visual del iris humano se determina por el proceso caótico y morfogenético durante el proceso embrional. Se ha postulado ser distintivo para cada persona y cada ojo” (Serratosa, 2012, p. 24). Una de las características más valiosas que presenta este método está dada por la permanencia en el tiempo. Según Marín et al. (2009) teniendo en cuenta las características en las que esta técnica está basada, el patrón de la textura del iris permanece sin alteraciones durante la vida de la persona debido a la protección que le ofrece la córnea (p. 34).

Los métodos mencionados hasta ahora exigen cierto grado de cooperación por parte de los usuarios y se requiere disponer de los recursos necesarios para llevar a cabo el proceso de la captura. En la propuesta de intervención que se mencionará en el capítulo siguiente, estaremos utilizando el método de reconocimiento basado en el rostro, debido a las bondades de las características que ofrece, como el bajo nivel de intrusión y la aceptación por parte de los usuarios. Según Serratosa (2012) la cara es uno de los rasgos biométricos más aceptables, ya que, es el más utilizado por los humanos para reconocer a otras personas en las interacciones visuales que se dan en la vida cotidiana. Además, menciona que el método para adquirir imágenes del rostro no es intrusivo y no requiere de una interacción por parte del sujeto (p. 23).

Volviendo a los intereses de este escrito, uno de los problemas principales que tienen las empresas desarrolladoras de videojuegos online o los operadores del juego, es que cuando los usuarios son registrados en su plataforma, solamente se les solicita un e-mail y algunos datos que pueden ser falsificados o reutilizados. Por el motivo mencionado es que las plataformas de videojuegos online podrían optar por un método de identificación unívoco para poder tomar decisiones en el futuro, adoptando alguno de los estándares de reconocimiento biométrico, ya sea el rostro, el iris o el que se prefiera. En la propuesta de intervención que mencionaremos más adelante, se optará por utilizar el reconocimiento facial como una alternativa posible, junto a una solución integral para estos inconvenientes.

2.3.6 *Detección y reconocimiento facial*

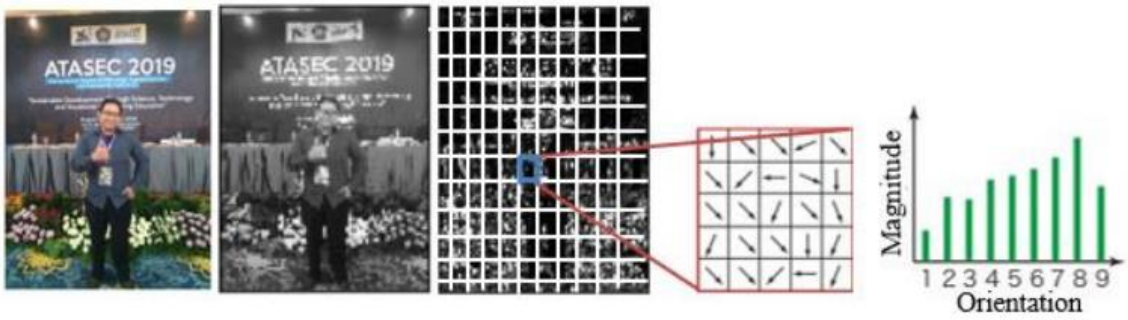
Existen diferentes algoritmos para lograr la detección de rostros dentro de una imagen y extraer las características fundamentales, como por ejemplo el algoritmo de Viola-Jones, o el Histograma de Gradientes Orientados (HOG) de Dalal y Triggs. El algoritmo de Viola-Jones fue propuesto por Paul Viola y Michael Jones en el año 2001. Se basa en cuatro elementos, las características de tipo Haar, las imágenes integrales, el algoritmo AdaBoost, y los

clasificadores en cascada. El marco de trabajo fue demostrado y en parte a la vez motivado por la detección facial (Viola y Jones, 2001, p. 1).

Por otro lado, los descriptores HOG o Histogramas de Gradientes Orientados (*Histogram of Oriented Gradiants*) se basan en la orientación del gradiente en áreas locales de una imagen. Fueron presentados por primera vez en el año 2005 por Dalal y Triggs. Según Rahmad et al. (2020), han tenido éxito en la detección de objetos y de peatones, representando un objeto como un único vector opuesto a un conjunto de vectores de características donde cada uno representa una región de la imagen, mediante un detector de subregión que se desplaza sobre una imagen. El descriptor HOG permite aprovechar de forma eficiente la información del gradiente a partir de combinar esta información en forma de histogramas orientados locales, que se calculan en celdas de tamaño pequeño, las cuales se distribuyen de forma uniforme por toda la imagen (p. 4). Ver Figura 6 a continuación con la representación de los gradientes e histogramas de una imagen.

Figura 6

Extracción de características basadas en el histograma de gradientes orientados



Nota. Obtenido de Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection (p. 4), por Rahmad et al. (2020)

En la propuesta de intervención utilizaremos un algoritmo basado en HOG pre-entrenado para detectar rostros y obtener un vector de 128 características de cada uno.

2.3.7 *Ética y Privacidad de los datos*

Los sistemas de reconocimiento biométricos traen aparejados una serie de cuestiones sociales, legales y culturales que se deben tener en cuenta al implementar los procesos de captura de rasgos y almacenamiento. Si por alguna razón se ignora alguno de estos elementos, la eficacia del sistema podría verse comprometida y podría traer serias consecuencias no intencionadas (Pato y Millet, 2010, p. 85). Estas consideraciones se refieren a la participación

de los individuos con los sistemas biométricos, a los potenciales impactos que estos sistemas pueden tener sobre la sociedad, a los asuntos legales que derivan de la biometría y a las políticas de recolección de datos y uso de ellos.

Según Pato y Millet (2010) la performance de los sistemas biométricos puede verse afectada si los factores sociales no son considerados correctamente. Estos factores sociales se dividen en dos tipos, aquellos que motivan el uso por parte de los usuarios y los que facilitan la participación (p. 86). Cuando se mencionan a los factores que motivan el uso de los sistemas, se refieren a aquellos que se encuentran estrechamente relacionados con los beneficios que pueden brindar los sistemas. La participación de los individuos también podría verse motivada por las consecuencias negativas de la no participación (Pato y Millet, 2010, p. 87). En el capítulo siguiente, al comentar la propuesta de intervención en el campo profesional, se mencionará que la decisión final la tendrá el stakeholder permitiendo o no, el ámbito multijugador para aquellos usuarios infractores. De la misma forma, aquellos usuarios que se reúsen a registrar sus datos biométricos podrían ser privados de los servicios multijugador en línea por parte de los operadores del videojuego.

Aquellos factores que facilitan la participación de los individuos se refieren al nivel de facilidad con que los usuarios pueden interactuar con los sistemas y a la gestión de aquellos errores que podrían producirse durante el proceso de captura o verificación (Pato y Millet, 2010, p. 87).

Según Pato y Millet (2010) el deseo de participación puede verse influenciado por la preocupación de que los usos del sistema cambiarán con el paso del tiempo. Los autores ponen como ejemplo a un sistema que inicialmente fue implementado para permitir el fácil acceso de sus empleados a la zona de trabajo, podría luego ser usado para rastrear la asistencia, las horas trabajadas, e incluso el movimiento dentro del lugar de trabajo (p. 87). Por lo comentado anteriormente es que se deben establecer claramente y desde el inicio, los fines por los cuales los datos biométricos serán utilizados dentro del sistema.

Es importante tener en cuenta que la solución de reconocimiento biométrico a implementar sea acorde a las necesidades y a la problemática a resolver. Según Pato y Millet (2010) incluso aquellas soluciones técnicas de gran efectividad podrían tornarse inapropiadas debido a los efectos secundarios percibidos o reales que los sistemas producen en los individuos, siendo la proporcionalidad, es decir, cómo los individuos perciben al sistema y los

efectos secundarios que producen, una característica a considerar primeramente cuando se está analizando el espacio de la solución (p. 89). Los efectos secundarios que podrían originarse están relacionados con la potencial privación de los derechos de aquellos usuarios que no pueden participar, con el impacto de la variación cultural desde diferentes perspectivas de los usuarios, y con demás inconvenientes vinculados con la privacidad de los datos.

Según Pato y Millet (2010), donde los sistemas biométricos son usados de forma exhaustiva, es probable que algunos miembros de la comunidad estén privados de sus derechos. Algunos individuos podrían no matricularse de forma correcta en el sistema o ser reconocidos, debido a una cuestión física, y otros usuarios podrían no tener características lo suficientemente distintivas para ser reconocidos por el sistema. Además, los autores indican que podrían existir cuestiones culturales o religiosas que dificultan el proceso de reconocimiento. Las creencias religiosas acerca del cuerpo y la jurisdicción sectaria sobre las características personales (como la barba o los pañuelos en la cabeza), o el contacto interpersonal (como la exposición de partes del cuerpo, contacto físico o toma de fotografías) podrían convertir un sistema biométrico en una intrusión inaceptable (p. 89). Por tal motivo, es probable que existan diferentes grupos que puedan verse discriminados al estar obligados a utilizar un sistema específico y violar sus creencias religiosas. Otra categoría de individuos podría elegir no participar en los sistemas biométricos debido a la preocupación relacionada con el uso inadecuado de los datos (Pato y Millet, 2010, p. 89).

Otro de los problemas que deben ser analizados con respecto al impacto social que los sistemas biométricos producen, es la privacidad como una consideración cultural. Según Pato y Millet (2010), casi ninguna discusión popular acerca de los sistemas y las tecnologías biométricas se dan sin referirse a las preocupaciones de privacidad, la potencial vigilancia, y a las grandes bases de datos con información personal puestas a disposición para usos desconocidos (p. 90).

Hoy en día la información de los individuos almacenada en bases de datos está relacionada con otros sistemas con fines comerciales o incluso por cuestiones legales. De la misma forma los datos biométricos pueden ser potencialmente asociados con otros sistemas y se deberá responder ante las cuestiones surgidas por las asociaciones indeseadas. Dependiendo en los usos anticipados de los datos personales, los mecanismos técnicos y políticos deben ser definidos para prevenir la asociación de datos no autorizados (Pato y Millet, 2010, p. 91). Por otro lado, los sistemas de vigilancia encubierta también podrían traer aparejadas implicancias

con respecto a la privacidad. Según Pato y Millet (2010), algunos sistemas de reconocimiento pueden funcionar a distancia, haciendo posible la asociación de datos con una persona, sin que esta participe de forma explícita. Si estos sistemas fuesen implementados de forma global, podría existir una potencial desconfianza de las instituciones que adoptan estas tecnologías (p. 92).

La suposición de los asuntos técnicos en el contexto informático es muy diferente con respecto a los que se tienen en el campo legal. Se debe entender el contexto global y el legal en simultáneo, para que los sistemas biométricos operen de forma efectiva. La biometría trae aparejado una serie de asuntos legales como la remediación, fiabilidad y privacidad. Tal como mencionan Pato y Millet (2010), la remediación se refiere a los pasos legales para combatir el uso fraudulento de la biometría, tales como el fraude de identidad por alteración o encubrimiento de rasgos biométricos, la modificación de las referencias biométricas o el uso de muestras biométricas falsas para personificar a un individuo. También se incluye las circunstancias en donde se le niegan incorrectamente los derechos a un individuo debido a una no coincidencia falsa durante el proceso de reconocimiento (p. 95). Por tal motivo, es necesario que los sistemas cuenten con medidas de seguridad contra fraudes y que aquellos usuarios que no puedan ser autenticados incorrectamente, se les otorgue un método de identificación secundario con celeridad ya que ningún sistema está libre de errores o fraudes. Además, es necesario que mediante las políticas y las leyes se accione contra los perpetradores de fraude y se incentive a los propietarios de los sistemas biométricos a implementar ambientes seguros para proteger las muestras biométricas (Pato y Millet, 2010, p. 96).

La privacidad y la fiabilidad son elementos clave para que los sistemas biométricos sean ampliamente aceptados. Según Pato y Millet (2010) a largo plazo, las aplicaciones biométricas que han sido foco de errores frecuentes perderán el soporte público, aun cuando los medios oficiales y televisivos hayan promocionado la idea de que los datos biométricos son casi infalibles. Además, los autores agregan como ejemplo para remarcar la importancia del asunto, que, en un proceso judicial, los jueces deben de estar seguros de que las pruebas de las huellas dactilares pertenecen a cierta persona y no a otra, para poder emitir un juicio de forma correcta (p. 96).

Es importante establecer una política de datos para poder responder a cuestiones que tienen que ver con el compartimiento, almacenaje, integridad y confidencialidad de la información, asociados a los sistemas biométricos. De acuerdo con Pato y Millet (2010), los

datos biométricos pueden estar correlacionados a través de diferentes sistemas de identificación para reconocer individuos. Los datos recolectados asociados con un individuo por diferentes organizaciones y utilizando la misma modalidad biométrica pueden ser similares, pero casi ciertamente no idénticos, porque la adquisición de la muestra para la matriculación puede variar (p. 111).

Por otro lado, es importante destacar que, para poder almacenar los datos biométricos de los usuarios, se deberá realizar la correspondiente disociación de la información. De ninguna manera se debe almacenar datos de los usuarios en un formato plano e interpretable por cualquiera, sino que deben estar encriptados bajo un algoritmo criptográfico confiable. Se debe tener en cuenta que cada país tiene sus regulaciones en cuanto a la ética y privacidad de los datos. En Europa, por ejemplo, a través del Reglamento general de protección de datos (RGPD), los usuarios pueden solicitar el borrado de sus datos personales de cualquier base de datos (Reglamento (UE) 2016/679 Del Parlamento Europeo y del Consejo, 2016).

2.3.8 *Usos frecuentes de La Biometría*

El uso de la biometría abarca un gran abanico de aplicaciones como por ejemplo el control de acceso a un lugar de trabajo, el otorgamiento de beneficios, aplicaciones bancarias, forenses, e incluso en los dispositivos móviles para poder hacer uso de ellos (Serratosa, 2012, p. 31). Por tal motivo es necesario comprender los contextos y categorías de las aplicaciones. A continuación, haremos una revisión de los diferentes contextos existentes.

Sistemas cooperativos contra no cooperativos. Se distingue entre aquellos sistemas que requieren o no la cooperación del usuario para el reconocimiento biométrico. Según Serratosa (2012) durante la captura del iris, el usuario debe cooperar para centrar dicho rasgo en el medio de la imagen. En el caso de los sistemas no cooperativos, se refiere a aquellos que no requieren la ayuda del usuario para ser reconocidos, como lo es el reconocimiento facial (p. 31).

Sistemas habituados contra no habituados. Es importante comprender la frecuencia con que ciertos usuarios utilizan el sistema de reconocimiento biométrico. “Es un factor importante en el diseño del sistema puesto que se ha demostrado que, si el usuario está habituado a interactuar con el sistema, la precisión de este aumenta claramente” (Serratosa, 2012, p. 31).

Supervisados y no supervisados. La supervisión se refiere a la necesidad de control por parte de una persona durante los procesos de matriculación o de verificación de los datos recolectados. “Los sistemas supervisados son aquellos en los que en el proceso de adquisición de los datos está supervisado, observado o guiado por un humano (por ejemplo, un encargado o un funcionario de seguridad)” (Serratosa, 2012, p. 31).

Entorno de funcionamiento estándar frente a no estándar. El entorno de funcionamiento estándar se refiere a las condiciones normales que los seres humanos estamos acostumbrados, con respecto a temperatura, presión, luz, humedad y ruido, entre otros (Serratosa, 2012, p. 31). Podemos intuir que los entornos no estándares, son aquellos que operan bajo condiciones especiales, fuera de los parámetros a los que los humanos estamos habituados.

Sistemas privados frente a públicos. Según Serratosa (2012), los usuarios que hacen uso de los sistemas privados son los clientes o los empleados de una empresa u organización que ha implementado un sistema biométrico. Además, el autor indica que, estos usuarios son especiales ya que están acostumbrados a utilizar el sistema biométrico y facilitan el proceso de matriculación, verificación e incluso identificación, siendo la tasa de error muy baja. Los sistemas públicos hacen referencia a todos los otros sistemas restantes (p. 32).

Abiertos o Cerrados. Es importante distinguir entre los sistemas abiertos y cerrados. Este contexto se refiere a la posibilidad de utilizar los datos recolectados en otras aplicaciones. “En los sistemas abiertos, la plantilla de un usuario almacenada en la base de datos se puede usar en varias aplicaciones” (Serratosa, 2012, p. 32). Por lo mencionado anteriormente también tendremos variaciones con respecto a la arquitectura de la solución y particularmente de la distribución de la base de datos. Según Serratosa (2012) en los sistemas abiertos se dispone de una única base de datos y de un solo proceso de matriculación por usuario. En cambio, en los sistemas cerrados, se deberá realizar otra matriculación y se tendrá otra base de datos asociado a este nuevo proceso. Además, el autor menciona que en los sistemas abiertos se deberá utilizar formatos estándares para todo el proceso, siendo este punto un desafío para los implementadores (p. 32).

Declarado o Encubierto. Los sistemas pueden ser declarados o encubiertos. Se refiere al grado de conciencia que tiene el usuario de ser partícipe de un sistema biométrico. “Los sistemas declarados son aquellos en los que el usuario se da cuenta y acepta la interacción con

el sistema biométrico" (Serratosa, 2012, p. 32). El concepto de sistema encubierto está relacionado con lo explicado anteriormente acerca de vigilancia encubierta. Son aquellos en los que el usuario no sabe que está siendo parte de un sistema biométrico para su reconocimiento a distancia (Pato y Millet, 2010, p. 92).

Por otro lado, las aplicaciones biométricas, pueden ser categorizadas en horizontales y verticales. En la categorización horizontal, las aplicaciones tienen un objetivo o entorno en común. La categorización vertical se basa en las necesidades de cada sector industrial o gubernamental (Serratosa, 2012, p. 32).

2.4 Estado del Arte

Con el avance de la tecnología han surgido nuevas propuestas de solución y aportes innovadores en el campo de los sistemas antitrampas, dejando de lado aquellos métodos tradicionales utilizados hasta ahora, como, por ejemplo, la aplicación de la inteligencia artificial (IA) para la clasificación de usuarios tramposos (Spijkerman y Ehlers, 2020). Este estudio menciona la utilización de árboles de decisión, máquina de vectores de soporte y el uso de clasificadores de Naive Bayes basado en un modelo probabilístico. El modelo propuesto analiza los movimientos y las dinámicas del ratón (*mouse*), las diferentes teclas presionadas y además realiza un análisis de datos estadísticos.

De forma similar otro trabajo menciona el uso de Deep Learning para el análisis del comportamiento desde la perspectiva del usuario, en el juego Counter-Strike: Global Offensive de Valve Corporation (Zhang, 2021). Este modelo menciona que cierta información es privada y accesible únicamente desde el servidor, por lo tanto, se evalúa la información desde la perspectiva local del usuario, para determinar si se está utilizando trampas.

Luego otros estudios mencionan la implementación de agentes de aprendizaje basados en Machine Learning, para diferenciar entre un humano y una máquina, utilizando el framework de Unity (Lukas et al., 2022). Con respecto a Blockchain, investigadores de IBM han estado trabajando en un sistema antitrampas para controlar los diferentes estados por los que pasa el cliente de un videojuego, a través del consenso y de los contratos inteligentes (Kalra et al, 2018). La solución mencionada propone restringir cualquier modificación que un usuario realice sobre el software cliente y esté relacionada con el uso de trampas.

Algunas plataformas externas como GamersClub han implementado recientemente un método de verificación complementario en el cual se solicita un documento de identidad para

acceder a ciertos servidores privados (GamersClub, 2023). Esta plataforma utiliza un aplicativo llamado MeuID (Idwall, 2021) que se encarga de verificar el documento ingresado y de validar la cuenta utilizada. En caso de que un individuo fuese prohibido de la plataforma por tratarse de un usuario tramposo o conflictivo, no podría volver a crear una cuenta bajo el documento ya utilizado. GamersClub es una plataforma privada de origen brasilera, ajena a Valve Corporation, que utiliza sus propios servidores de Counter-Strike bajo un modelo de suscripción, y un sistema antitrampas dedicado (GamersClub, s.f.).

Capítulo 3 – Propuesta De Intervención

3.1 Descripción

Durante el XIV Certamen de Trabajos Estudiantiles del XXI Congreso Internacional en Innovación Tecnológica, Miyashiro (2023) presentó la siguiente propuesta:

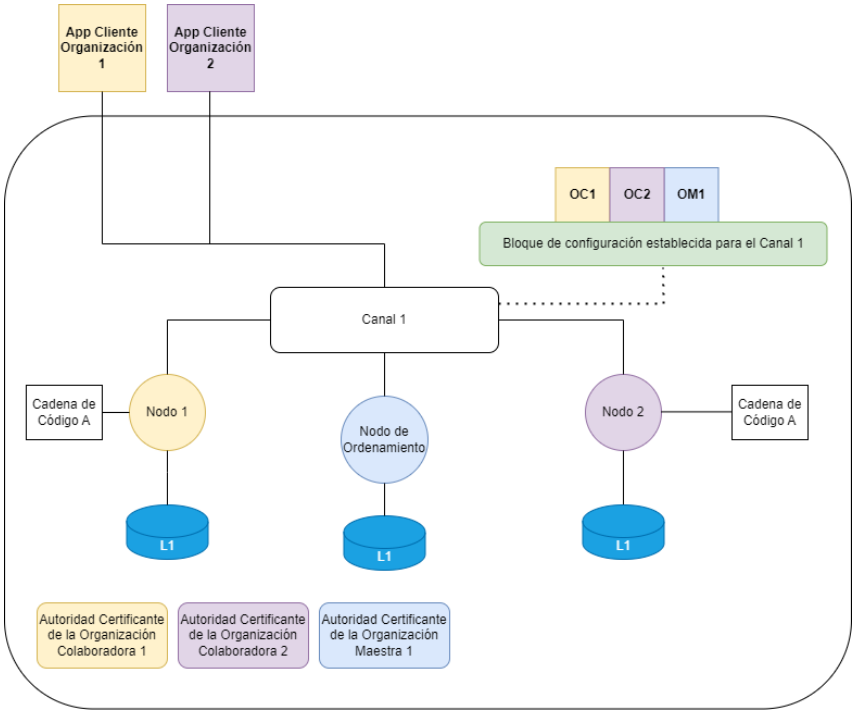
> El sistema de penalización propuesto en este trabajo expone diferentes ventajas frente a los métodos tradicionales de castigo, el hecho de que un usuario sea identificado de forma unívoca permite tomar decisiones más sólidas con respecto a qué hacer con el usuario. Extender el registro a la toma de datos biométricos y no solamente a datos falsificables, habilita a las organizaciones a poder detener a las personas que busquen hacer trampas en los videojuegos de forma permanente y no solamente castigar a un ID de usuario. Además, la implementación de un sistema blockchain de consorcio, provocará que las empresas asociadas puedan reportar a los usuarios de sus plataformas y automáticamente los demás nodos o empresas, puedan tomar decisiones en cascada (p. 10).

En el desarrollo de la propuesta se llamará organización maestra (OM) a aquella primera impulsora de la implementación de este sistema. Por otro lado, se denominará organización colaboradora (OCx) a todas las demás organizaciones que se encuentren asociadas a este consorcio de empresas, donde la letra “x” se refiere a un número asignado a cada una. Si bien, durante este trabajo nos hemos referido a los sistemas parcialmente descentralizados en el campo de estudio de Blockchain, debemos tener en cuenta que el proceso de matriculación de los datos biométricos es recomendable que se realice en un ambiente centralizado, aplicando las medidas de seguridad pertinentes. Como comentamos anteriormente en este trabajo (ut supra, 2.3.6, p. 24) los datos recolectados y asociados a cierto individuo por diferentes plataformas y utilizando la misma modalidad biométrica pueden ser similares, pero no idénticos, por lo tanto, podría dar lugar a reconocimientos incorrectos o a falsos no reconocimientos. La tecnología y los diferentes métodos utilizados en la captura, procesamiento y extracción de características podrían ser completamente diferentes entre organizaciones, imposibilitando la correcta identificación de los individuos. Por todos los motivos mencionados anteriormente, es que la matriculación es recomendable realizarla solamente por la OM, y por única vez. Las empresas que formen parte del consorcio asociarán los datos biométricos obtenidos durante la matriculación, con las cuentas de usuario de cada plataforma en particular. Además, es sumamente importante comunicarle al usuario los

términos y condiciones, indicando explícitamente la finalidad de esta matriculación para el uso autorizado, tanto por el sistema de matriculación de datos biométricos, como por todas las empresas adheridas al blockchain de consorcio. Durante el registro de los datos biométricos (matriculación), los términos y condiciones deberán ser aceptados por los usuarios, brindando el debido consentimiento del uso de la información y acceso a dispositivos, como la cámara web.

Figura 7

Esquema del Sistema Blockchain de consorcio de la propuesta



En la Figura 7 se muestra la estructura que deberá tener el sistema blockchain para poder funcionar en forma de consorcio de empresas. Las organizaciones OC1, OC2 y OM1 trabajarán en conjunto para reportar y nutrirse de la información generada de la cadena de bloques. Estas organizaciones son ficticias y deberán corresponderse a empresas que busquen asociarse para llevar adelante la implementación del sistema. Adicionalmente como puede verse en el gráfico, la base de datos (también conocida como *ledger* o L1 en la Figura 7) se encuentra replicada en cada nodo que forma parte de este sistema, por lo tanto, se encuentra dentro de la infraestructura local o cloud de cada organización. Cada nodo forma parte de una organización diferente, por ejemplo, el Nodo 1, pertenece a la OC1 y el Nodo 2 a OC2.

Existe un nodo particular que se denomina Nodo de Ordenamiento, este es el responsable de establecer el control de ordenamiento de las transacciones al momento de escribir en la cadena de bloques. El conjunto de estos nodos de ordenamiento se lo conoce como el servicio de ordenamiento (Hyperledger Fabric, 2022, párr. 3), y en este caso es parte de la Organización Maestra (OM1). En la Figura 7, solamente se expone un nodo de ordenamiento, pero en la práctica debemos contar con por lo menos tres nodos para establecer consenso y garantizar la alta disponibilidad del servicio.

Según Hyperledger Foundation (s.f), el framework Hyperledger Fabric es una plataforma de libro mayor distruído, de nivel empresarial, con carácter modular y flexible para diferentes casos de usos de la industria. La arquitectura modular está determinada por el uso de componentes plug-and-play, como el consenso, la privacidad y los servicios de membresía.

Como el framework a utilizar será Hyperledger Fabric, es importante destacar que los contratos inteligentes, deben ser encapsulados en cadenas de código. Esta característica es propia del framework utilizado. “Con respecto al framework utilizado, no necesariamente debe tratarse de Hyperledger Fabric. En esta propuesta en particular, mencionamos a este framework por el grado de madurez alcanzado, en materia de generación de blockchains de consorcio” (Miyashiro, 2023, p. 11). Las cadenas de códigos son programas que pueden incluir uno o más smart contracts embebidos en ellas, conteniendo toda lógica de negocio acordada por los nodos (Hyperledger Fabric, 2023, párr. 2), y como hemos comentado anteriormente, son necesarios para poder interactuar y realizar consultas en el sistema blockchain.

El bloque de configuración del canal es el responsable de establecer las políticas de consenso y los roles que cada una de las organizaciones posee dentro del sistema. Todas las organizaciones que forman parte del sistema deben estar de acuerdo con la configuración establecida en el canal. El canal es una subred privada necesaria para poder establecer la comunicación entre las organizaciones que forman parte y definido por los miembros (Hyperledger Fabric, 2023b, párr. 1). Por otro lado, cada organización posee una autoridad certificante que se encarga de la emisión de los certificados, con el fin de diferenciarse e identificar a otros nodos y organizaciones que formen parte de la red (Hyperledger Fabric, 2017, párr 1).

Finalmente, para poder interactuar con el sistema blockchain, cada organización posee una aplicación cliente, que se comunica con el canal y permite ejecutar diferentes consultas a

través de los contratos inteligentes que han sido instalados en cada uno de los nodos (Hyperledger Fabric, 2022b, párr. 1). Es importante recordar que estos contratos se implementan mediante las cadenas de código que son aprobadas en el canal.

La política de consenso, mejor conocida como endorsamiento en Hyperledger Fabric (Hyperledger Fabric, 2023c, párr. 1), podrá establecerse mediante el algoritmo práctico de tolerancia a fallas bizantinas (PBFT) o mediante el método Raft. En esta propuesta de intervención usaremos este último. Recordemos que Raft se basa en el mecanismo de líder y seguidor, en el que un líder es elegido de forma dinámica dentro de los nodos de ordenamiento del canal y luego replica los mensajes a los seguidores. El método Raft admite la pérdida de nodos en caso de cualquier falla técnica, siendo necesario que la mayoría de ellos deben estar disponibles para poder establecer el consenso.

Con respecto a los datos biométricos, en este ejemplo se tomarán las características del rostro para poder identificar de forma unívoca a cada usuario asociado, a través de un algoritmo de reconocimiento facial basado en el método de histogramas de gradientes orientados, con un vector de 128 características faciales. Ver Figura 8 con el esquema de la aplicación biométrica. Es importante destacar que podría utilizarse cualquier otro algoritmo o modalidad biométrica en esta etapa, dependiendo de las necesidades y de las preferencias de las organizaciones que implementen el sistema.

Figura 8

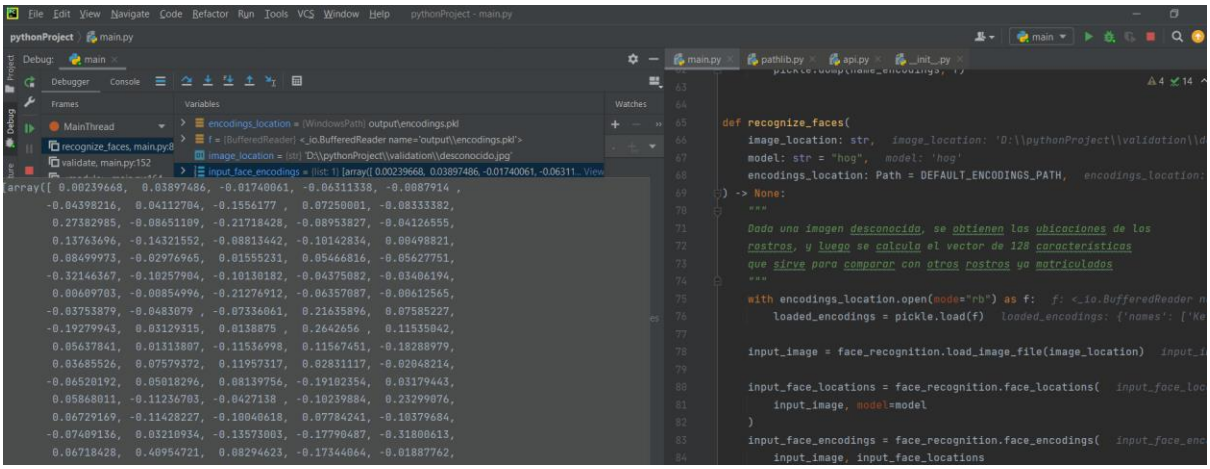
Esquema del Sistema de Captura de Datos Biométricos

Nota. Gráfico realizado utilizando la aplicación draw.io con los elementos de IBM / Security

Para lograr lo comentado en el párrafo anterior, se hará uso de la biblioteca preelaborada, *face-recognition (*reconocimiento facial) escrita en el lenguaje de programación Python (Geitgey, 2020), y que hereda funcionalidad de la biblioteca *dlib* (http://dlib.net/) que forma parte del estado del arte en reconocimiento facial y machine learning. Ver siguiente figura con un ejemplo del vector obtenido en la línea 83.

Figura 9

Fragmento de código que calcula el vector de 128 características



Nota. La IDE PyCharm Community Edition 2023, fue utilizada para mostrar el código.

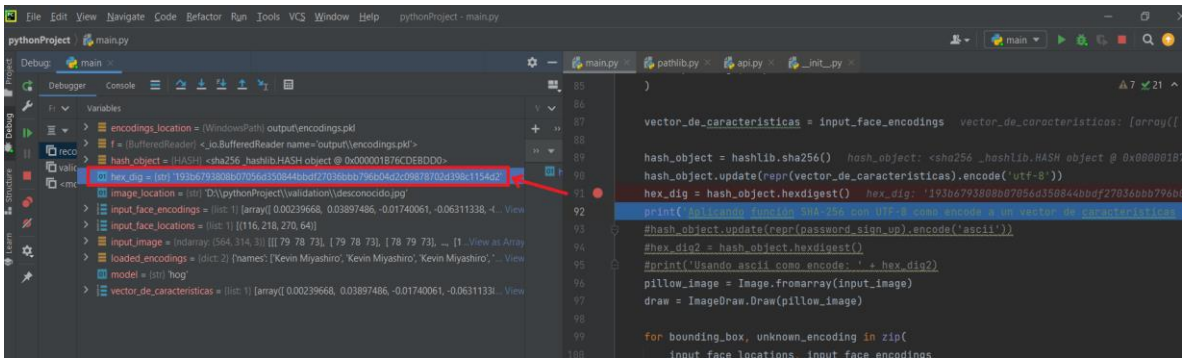
El método *face_locations*, en primer lugar, localiza las coordenadas del rostro dentro de la imagen, y las delimita dentro de un cuadro (bounding box). Ver Anexo I con un ejemplo del cuadro delimitador. Posteriormente, se calcula el vector de las características en base a la ubicación del punto anterior, con el método *face_encodings*.

Durante la etapa de entrenamiento, se requieren al menos dos capturas del rostro de un individuo. Cuantas más imágenes se aporten en esta etapa, mayor es la precisión lograda con el método *recognize_faces*, que sirve para el reconocimiento facial a partir de una imagen desconocida. La organización maestra deberá implementar una aplicación para llevar a cabo todo el proceso de matriculación y luego respaldar esta información en una base de datos centralizada. Dentro de la información respaldada acerca de los datos biométricos, también se incluye el resultado de la aplicación de una función criptográfica hash de 256 bits (SHA-256). Por cada imagen de entrenamiento se calcula el resultado de la función y se almacena en un vector. Estos hashes calculados serán los únicos datos con los que se podrá interactuar en el sistema blockchain y entre las empresas adheridas. Los vectores de 128 características

solamente estarán almacenados en la base de datos de la aplicación biométrica. Ver figura 10 a continuación con un ejemplo del cálculo de la función criptográfica SHA-256 sobre un vector de características.

Figura 10

Cálculo de la función SHA-256 sobre un vector de características



En resumen, los datos que el sistema biométrico deberá respaldar estarán conformados por la siguiente tupla:

ID, Nombre, Apellido, Documento, CaracterísticasBiométricas[2], HashBiométrico[2].

La dimensión del vector de hashes biométricos (HashBiometrico) deberá tener la misma longitud que el vector que almacena las características biométricas (en el ejemplo, se tomó un vector de dimensión 2). A la vez, cada elemento del vector “CaracterísticasBiométricas”, corresponde a un vector de 128 características faciales.

Por último, esta aplicación de captura de datos biométricos deberá exponer APIs de consulta y asociación con las bases de datos de usuarios en las plataformas adheridas al sistema. Estas APIs se encontrarán restringidas y solamente podrán ser invocadas por las empresas que formen parte del consorcio. De esta forma, los usuarios podrán vincular sus datos biométricos con las cuentas utilizadas en las diferentes plataformas de videojuegos online adheridas.

Cuando una organización colaboradora, asocia los IDs de cuenta de los usuarios locales, con el registro del sistema biométrico, podrá acceder a los hashes guardados en este sistema. Esta asociación podrá realizarse una única vez, por cada cuenta generada en la plataforma. Es importante tener en cuenta que no se puede tener más de una asociación que con un registro en el sistema biométrico, de lo contrario, una cuenta podría pertenecer a más de una identidad.

Los contratos inteligentes dentro del sistema blockchain, deberán manipular solamente el vector de hashes biométricos (HashBiométrico). La dimensión del vector estará relacionada directamente con la cantidad de vectores biométricos capturados en la etapa de matriculación. Además se necesitarán otros datos para emitir reportes e identificar a los usuarios. A continuación, detallaremos un ejemplo del smart contract utilizado para interactuar con el sistema blockchain, en el que se puede generar reportes de usuarios, y además se pueden obtener todos los reportes por usuario, utilizando el ID o mediante el vector de hash biométrico. Los campos utilizados en el reporte podrían variar en el contrato inteligente, dependiendo de los intereses de las organizaciones implementadoras.

Figura 11

Ejemplo de Estructura de Reporte en el Sistema Blockchain.

Nota. Fragmento de código escrito en el lenguaje de programación Golang.

De forma periódica, las empresas asociadas al sistema blockchain de consorcio, deberán realizar una consulta para buscar aquellas coincidencias con los usuarios locales, en base a cualquier campo que deseen utilizar, como el ID o con el vector HashBiométrico. Si existe alguna coincidencia entre la base de datos local y la cadena de bloques, se podrá tomar una decisión con respecto a la continuidad del usuario en la plataforma involucrada. Esto quiere decir que otra empresa ya ha reportado al usuario en el sistema blockchain de consorcio por actividades ilegales en su plataforma, al haber sido detectado por el sistema antitrampas de dicha organización. En caso de haber establecido una política de tolerancia cero con respecto a los usuarios tramposos, podría tomarse una decisión en cascada y también prohibir al usuario involucrado en la plataforma que realiza la consulta a la cadena de bloques.

Todo el detalle de las transacciones realizadas se podrá obtener mediante el método GetAllReportes del Smart Contract generado. Esta consulta es recomendable que sea realizada

durante un día en la semana, en que la organización involucrada esté dispuesta a establecer una ventana de tiempo de mantenimiento y a sincronizar las prohibiciones que han sido reportadas en la cadena de bloques. Ver código a continuación, con la función que retorna los registros de la cadena de bloques dentro del Smart Contract, escrito en Golang con Visual Studio Code 1.8.3 (Miyashiro, 2023b).

```
// El método GetAllReportes retorna todos los elementos encontrados en el
estado mundial.
func (s *SmartContract) GetAllReportes(ctx
contractapi.TransactionContextInterface) ([]*Reporte, error) {
	// range query with empty string for startKey and endKey does an
	// open-ended query of all assets in the chaincode namespace.
	resultsIterator, err := ctx.GetStub().GetStateByRange("", "")
	if err != nil {
		return nil, err
	}
	defer resultsIterator.Close()
	var reportes []*Reporte
	for resultsIterator.HasNext() {
		queryResponse, err := resultsIterator.Next()
		if err != nil {
			return nil, err
		}
		var reporte Reporte
		err = json.Unmarshal(queryResponse.Value, &reporte)
		if err != nil {
			return nil, err
		}
		reportes = append(reportes, &reporte)
	}
	return reportes, nil
}
```

A continuación, podemos ver el resultado de la ejecución del método dentro del Smart Contract, siendo quién invoca al método, la OC1 autorizada. En este caso, la cadena de bloques contiene un registro con un reporte de prohibición, originado el 11 de octubre de 2023.

Figura 12

Ejemplo de consulta a la cadena de bloques

Nota. El resultado arroja los reportes de usuarios generados en la cadena de bloques.

De la misma forma, para generar reportes se deberá utilizar el método CreateReporte. A continuación, se puede ver la definición del método para crear un reporte dentro del Smart Contract, escrito en Golang con Visual Studio Code 1.8.3 (Miyashiro, 2023c).

```
// Método utilizado para crear un reporte
func (s *SmartContract) CreateReporte(ctx
contractapi.TransactionContextInterface, id string, nombre string,
apellido string, estado string, hashbiometrico [3]string, source string,
fecha string) error {
	exists, err := s.ReporteExists(ctx, id)
	if err != nil {
		return err
	}
	if exists {
		return fmt.Errorf("el reporte %s ya existe", id)
	}
	reporte := Reporte{
		ID:             id,
		Nombre:          nombre,
		Apellido:           apellido,
		Estado:                        estado,
		HashBiometrico:   hashbiometrico,
		Source: source,
		Fecha: fecha,
	}
	reporteJSON, err := json.Marshal(reporte)
	if err != nil {
		return err
```

```
    }
    return ctx.GetStub().PutState(id, reporteJSON)
}
```

3.2 Acciones a realizar

En esta sección se detallará el plan necesario para llevar a cabo la propuesta de intervención. Es importante destacar que estará dividido en dos etapas, una destinada a la implementación de la aplicación de matriculación biométrica y la otra, a la generación del sistema blockchain respaldándose en las mejores prácticas de Hyperledger Fabric, junto con las organizaciones interesadas. Es importante tener en cuenta, que algunas de las acciones mencionadas la etapa de la creación del sistema blockchain, son abordadas desde el punto de vista de una única organización, aunque en la práctica, deberán repetirse las tareas de configuración en cada organización involucrada.

3.2.1 *Plan para la implementación del Sistema de Matriculación Biométrico*

Análisis de viabilidad para el desarrollo de una aplicación para la matriculación biométrica. Es necesario desarrollar un sistema para poder lograr la matriculación, por tal motivo se deberá realizar un análisis de las tecnologías disponibles y de los recursos funcionales y no funcionales para llevar adelante la implementación. El responsable de esta tarea es la organización maestra (OM).

Investigación y determinación de estándar biométrico a utilizar. Debe seleccionarse una modalidad biométrica accesible y un método para la matriculación asociada a la misma, como por ejemplo el reconocimiento facial utilizando HOG. El responsable de esta tarea es la organización maestra (OM) y debería ser discutida en conjunto con las organizaciones colaboradoras (OCx).

Asignación de equipo de proyecto para la implementación. Se deberá conformar un equipo de programadores y otros colaboradores funcionales para desarrollar el sistema biométrico. En caso de no contar con ellos, se deberá contratar externamente al capital humano adecuado. El responsable de esta tarea es la organización maestra y podría contar con apoyo de las demás organizaciones colaboradoras, si es que han llegado a un acuerdo en concreto en esta etapa.

Desarrollo de aplicación de matriculación biométrica. En base a diferentes especificaciones funcionales, los programadores podrán comenzar con las tareas de desarrollo

del sistema biométrico. La organización maestra, y más específicamente el equipo de proyecto serán los responsables de llevar a cabo esta tarea.

Pruebas unitarias. Se deberá probar cada módulo de la aplicación biométrica por separado, con el fin de corroborar el correcto funcionamiento. Los responsables de esta tarea son los consultores funcionales y programadores involucrados.

Pruebas integrales. En esta etapa, se deberá probar de forma completa el sistema de matriculación biométrico, desde el registro hasta la captura y resguardo de los datos. De la misma forma, los responsables de esta tarea son los consultores funcionales y programadores asignados al proyecto.

3.2.2 *Plan para la implementación del Sistema de Blockchain*

Asociación con organizaciones para la conformación del consorcio. Se debe establecer un acuerdo entre organizaciones con intereses similares, con respecto a las decisiones y medidas a tomar contra los usuarios infractores. Estas organizaciones formarán parte del sistema de consorcio y podrán apoyarse mediante el intercambio de reportes de usuarios en la cadena de bloques.

Generación de equipo de proyecto blockchain. Para esta etapa se requiere conformar un equipo de desarrolladores para la generación del sistema blockchain. Si bien inicialmente, la generación del ambiente estará a cargo de la organización maestra, las demás organizaciones deberán contar con su equipo de desarrolladores para poder realizar la implementación inicial y posterior mantenimiento necesario.

Determinación de consenso entre organizaciones. Con el fin de reportar usuarios se deberá establecer un algoritmo de consenso, basado en la tecnología blockchain utilizada. En el caso de Hyperledger Fabric, se podrá optar entre Raft y PBFT (Practical Byzantine Fault Tolerance). En los ejemplos mencionados en la descripción de la propuesta se utilizó Raft, determinando un nodo de ordenamiento para establecer consenso entre las diferentes organizaciones que forman parte del sistema blockchain. La organización maestra y las colaboradoras podrán contar con diferentes nodos de ordenamiento para garantizar la alta disponibilidad del servicio. Estas configuraciones deberán establecerse y ser aprobadas en el canal constituido por estas organizaciones, dependiendo de las políticas de endorsamiento establecidas.

Configuración de Autoridades Certificantes (CAs). El primer componente que debe desplegarse en una red blockchain es la CA (Certificate Authority) para cada organización involucrada. Este elemento permite identificar y distinguir a los diferentes componentes de la red, que pertenecen a diferentes organizaciones, mediante la emisión de certificados digitales.

Creación de nodos pares y nodos de ordenamiento. Luego de contar con la Autoridad Certificante para la identificación de los componentes de la red, ya es posible crear los nodos pertenecientes a cierta organización que mantendrán una copia del libro contable distribuido (ledger), y por otro lado la generación de los nodos de ordenamiento para poder establecer consenso y realizar la validación durante la ejecución de las transacciones.

Generación de Contratos Inteligentes. Luego de establecer el sistema parcialmente descentralizado o de consorcio, es necesario generar los Smart Contracts para facilitar la interacción con el sistema blockchain. Estos contratos, como se mostró en la descripción, deberán incluir la definición de la estructura de los reportes y los métodos necesarios para realizar las consultas y actualizaciones pertinentes. Además, los contratos deberán ser instalados en cada uno de los nodos interesados y aprobados en el canal, en forma de cadena de códigos.

Desarrollo de aplicación cliente blockchain. Para poder consultar y actualizar la cadena de bloques mediante la invocación de contratos inteligentes, será necesario desarrollar una aplicación cliente que pueda facilitar estas tareas. Además, esta aplicación será útil para poder asociar las bases de datos de usuarios locales de cada organización, con los hashes biométricos calculados a partir de los vectores de características faciales. Los responsables del desarrollo de esta aplicación serán los equipos de proyectos asignados y pertenecientes a cada organización.

Pruebas unitarias de la cadena de bloques. Se deberán realizar las diferentes pruebas necesarias para la interacción con el sistema blockchain, invocando los métodos de consulta y actualización definidos en el contrato inteligente, desde cada una de las organizaciones que formen parte del consorcio.

Pruebas integrales de la cadena de bloques. Se procede a probar toda la solución, partiendo desde la matriculación de los datos biométricos de los usuarios, hasta la generación de los reportes y la consulta de estos.

Configuración de proceso de fondo (batch) semanal. Luego de finalizar las pruebas integrales, cada organización ya se encuentra disponible para configurar el proceso de fondo semanal para sincronizar las bases de datos y tomar medidas en base a los nuevos reportes de usuarios infractores generados en la cadena de bloques, según las políticas de tolerancia de cada organización perteneciente.

3.3 Recursos a utilizar

En esta sección se detallarán los recursos humanos y materiales necesarios para llevar a cabo la propuesta. Es importante destacar que estos recursos son necesarios tanto del punto de vista de la organización maestra, como de todas las organizaciones colaboradoras que se asocien a la implementación.

Líder de Proyecto SR. Se requiere contar con un líder de proyecto senior, capaz de controlar y garantizar el cumplimiento de todas las tareas referidas al proyecto del sistema de matriculación biométrico como también al sistema blockchain de consorcio.

Programadores SR. La organización maestra deberá conformar un equipo de al menos tres desarrolladores con un alto nivel de conocimiento para desarrollar el sistema de matriculación biométrico y otro equipo destinado a la solución del sistema blockchain de consorcio. En total se requieren seis desarrolladores senior. Es importante tener en cuenta que, de la misma forma las organizaciones colaboradoras deberán conformar sus equipos para realizar las implementaciones locales. Es recomendable que estos desarrolladores cuenten con experiencia como full stack developers.

Administradores de Infraestructura Tecnológica SR. Para la puesta a punto inicial de los servidores necesarios para llevar adelante la propuesta, es necesario asignar un equipo de infraestructura responsable de estas configuraciones y de la seguridad de la información manipulada. Debido a la complejidad de la implementación se requiere un equipo de cuatro consultores de infraestructura tecnológica, con amplios conocimientos en administración de contenedores.

Consultores Funcionales. Se requieren tres consultores funcionales con un seniority intermedio, para elaborar las especificaciones técnicas que deberán llevar a cabo los programadores y además serán responsables de realizar las pruebas unitarias e integrales de toda la solución propuesta. Por otro lado, se requiere un cuarto recurso senior, para controlar

internamente las tareas funcionales. Estos recursos son necesarios desde el punto de vista de la organización maestra.

Soporte Técnico a las aplicaciones. Es necesario establecer un equipo para brindar soporte tanto durante la implementación de la solución integral, como también así luego de la salida en productivo. Para esto se requieren dos consultores SR (Senior), y dos consultores Ssr (Semi Senior).

Hasta ahora solamente se detallaron los recursos humanos necesarios para la implementación de la solución. A continuación, se indicarán los recursos materiales relacionados con todo el proyecto.

Servidor Linux de x64 bits para Blockchain. Cada nodo u organización deberá contar con un servidor Linux Ubuntu 22.04 o superior. Puede instalarse la versión Server (Servidor) o Desktop (Escritorio). Dentro de este servidor se deberá instalar Docker, Golang y el Framework de Hyperledger Fabric. La cantidad de servidores dependerá de la cantidad de nodos que se desee desplegar. Inicialmente para la organización maestra se requiere un servidor (para el nodo de ordenamiento), y para las organizaciones colaboradoras se requiere uno por cada nodo a instalar. La memoria RAM y el número de CPU por servidor, dependerá del volumen de transacciones a efectuar.

Storage para los nodos de Blockchain. Se recomienda asignar un storage inicial de 2 TB (Terabyte) del tipo SSD (Solid State Disk) para cualquier servidor involucrado en la red blockchain, aprovechando la tasa de I/O (Input/Outputs) de los discos sólidos y evitando problemas de performance.

Servidor Linux de x64 bits para el sistema biométrico. Adicionalmente la organización maestra deberá disponer de otro servidor destinado a la solución del sistema biométrico, también puede ser Linux Ubuntu 22.04 (Desktop o Server) o superior. En este servidor se deberá instalar Python 3.x o superior. Dependiendo del estándar biométrico a implementar se deberán instalar las herramientas adicionales necesarias.

Storage para el sistema biométrico. Inicialmente la organización maestra deberá contar con 2 TB (Terabytes) iniciales para almacenar la información biométrica capturada durante el proceso de matriculación. La dimensión del storage podrá ser mayor dependiendo de la cantidad de usuarios a someter por el proceso de captura de datos biométricos y de la modalidad biométrica empleada. Adicionalmente cada organización deberá disponer de

almacenamiento o storage local, para asociar las bases de datos de usuarios propias, con los hashes biométricos obtenidos durante el proceso de matriculación.

3.4 Cronograma de tareas

En este apartado se muestra el cronograma de tareas dividido en dos fases o etapas, a ejecutar durante un año. Al mismo tiempo el proyecto se encuentra determinado en tres cuartos de igual duración (de cuatro meses). Ver la Tabla 4 a continuación, con el detalle del plan.

Tabla 4

Cronograma de Tareas

	Primer Cuarto (Q" 1)				Segundo Cuarto (Q" 2)				Tercer Cuarto (Q" 3)			
Mes N°	1	2	3	4	5	6	7	8	9	10	11	12
Fase 1 - Plan de Implementación de Sistema de Matriculación Biométrico												
Análisis de viabilidad para el desarrollo de aplicación para la matriculación biométrica												
Investigación y determinación de estándar biométrico a utilizar												
Asignación de equipo de proyecto para la implementación												
Desarrollo de aplicación de matriculación biométrica												
Pruebas unitarias												
Pruebas integrales												
Fase 2 - Plan para la implementación del Sistema de Blockchain												
Asociación con organizaciones para la conformación del consorcio												
Generación de equipo de proyecto blockchain												
Determinación de consenso entre organizaciones												
Configuración de Autoridades Certificantes (CAs)												
Creación de nodos pares y nodos de ordenamiento												
Generación de Contratos Inteligentes												
Desarrollo de aplicación cliente blockchain												
Pruebas unitarias de la cadena de bloques												
Pruebas integrales de la solución												
Configuración de proceso de fondo (batch) semanal												

Durante la primera fase, se realizarán las tareas relacionadas con la implementación del sistema de matriculación biométrico, mientras que la segunda fase del plan estará dedicada al sistema blockchain.

3.5 Factores externos condicionantes

Existen factores externos que podrían condicionar la ejecución e implementación de este proyecto. En este subpunto se analizarán algunos de los más importantes para tener en cuenta.

La corrupción de los nodos. En el caso en que las empresas asociadas al sistema Blockchain, no reporten a todos los usuarios que han infringido las normas, podría resultar en un sistema sin utilidad. Es importante que las organizaciones que decidan asociarse compartan lineamientos similares con respecto a las decisiones que se deben tomar en contra de los usuarios, el hecho de reportar usuarios incorrectamente acabaría en un sistema no confiable.

Los datos biométricos de los usuarios. En caso de que los usuarios no deseen someterse al proceso de matriculación de datos biométricos, el proceso de identificación

unívoco sería afectado, y por lo tanto no se podría tener control sobre aquellos usuarios conflictivos. Por tal motivo sería conveniente, determinar que el proceso de captura y matriculación de datos sea un requerimiento mandatorio por parte de los usuarios, para hacer uso del ámbito multijugador en línea.

La caída de los nodos de ordenamiento. Es poco probable que todos los nodos de ordenamiento no se encuentren disponibles o sufran un ataque en simultaneo, pero en caso de suceder, el sistema no se encontraría habilitado para canalizar las consultas o actualizaciones que podría sufrir la cadena de bloques. Por lo mencionado anteriormente, es que es necesario determinar la cantidad de nodos de ordenamiento necesarios, para garantizar el consenso y la alta disponibilidad del servicio.

La necesidad de asociación entre organizaciones. Es sumamente importante que diferentes organizaciones se asocien e intercambien información significativa para la toma de decisiones. En caso de que la organización maestra no logre asociarse con ninguna otra organización, la propuesta no aprovechará las ventajas ofrecidas por la cadena de bloques, como por ejemplo la descentralización, y el compartimiento de la información. Cuantas más organizaciones logren asociarse, mayor es la probabilidad de que ocurran diferentes reportes de usuarios, donde posteriormente todos los miembros podrán consultar esta información y tomar decisiones en cascada.

3.6 Evaluación del Proyecto

Esta sección estará destinada a la evaluación del proyecto mencionado en la propuesta de intervención. Se construirá la matriz FODA y luego se procederá a realizar una evaluación de los costos asociados al proyecto.

3.6.1 *Matriz FODA de la propuesta de intervención*

Para mostrar en profundidad las características de la propuesta de intervención, se recurrirá a la elaboración de la matriz FODA, donde se detallarán las fortalezas, debilidades, oportunidades y amenazas que afronta el proyecto. Ver Tabla 5 con las características mencionadas.

Tabla 5

Matriz FODA de la propuesta de intervención

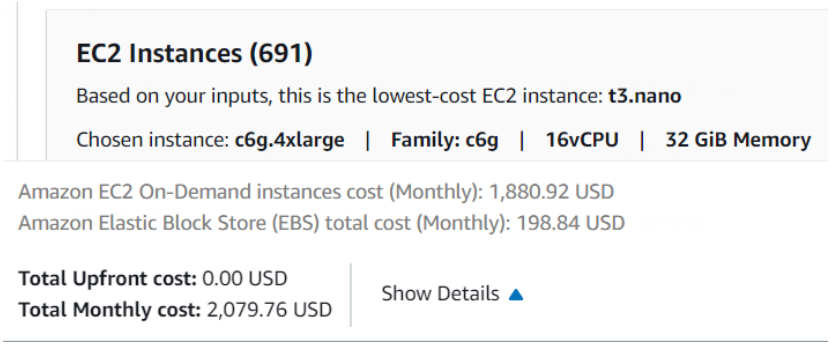
FORTALEZAS	OPORTUNIDADES
• La implementación complementa a los sistemas antitrampas actuales. • Los usuarios son disuadidos, previniendo la reincidencia de los infractores. • No requiere gran cantidad de recursos materiales, las herramientas y el framework son mayormente gratuitas.	• Nuevas alianzas y asociaciones con otras organizaciones con fines similares. • Se aprovechan todos los beneficios de la tecnología Blockchain. • Las empresas garantizan la continuidad del producto software en el tiempo.
DEBILIDADES	**AMENAZAS**
• El proyecto está dividido en dos fases, requiriendo mucho tiempo de implementación. • El estándar biométrico (modalidad y rasgo) seleccionado podría ser reemplazable por otro más preciso.	• En caso de que las organizaciones reporten incorrectamente, la cadena de bloques perdería utilidad. • Se requiere la colaboración de los usuarios en el proceso de matriculación biométrico. • Se precisa de capital humano especializado asignado al proyecto.

3.6.2 *Análisis de Costos*

Para poder estimar los recursos físicos o materiales, se tomará como ejemplo la cotización que ofrece Amazon Web Services, para el despliegue de los servidores dedicados (AWS Pricing Calculator, s.f). Los precios se encuentran en USD (dólares americanos).

Figura 13

Cálculo de costo del servidor AWS EC2 para Blockchain



Nota. Cotización generada utilizando la calculadora de Amazon Web Services.

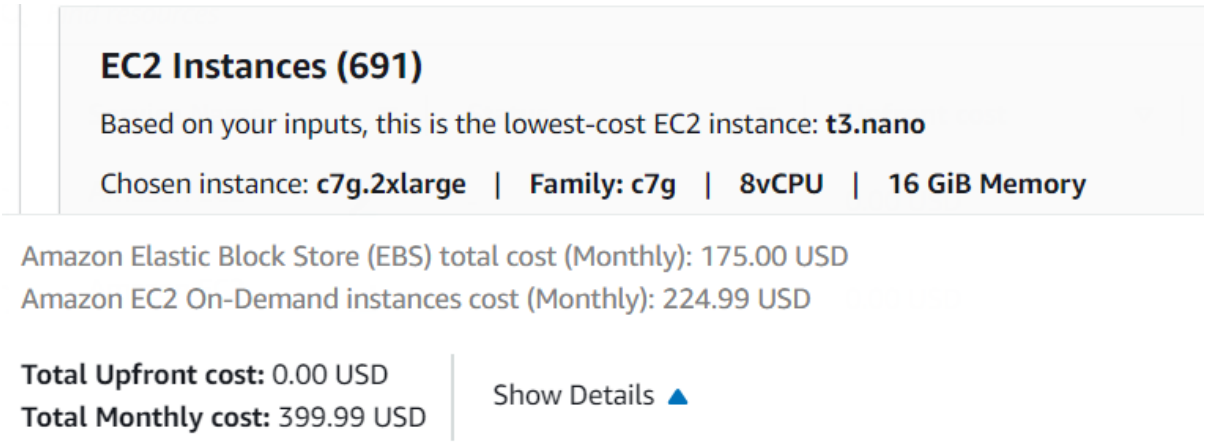
Como podemos ver en la Figura 13, un servidor en AWS EC2 de la familia c6g.4xlarge con 16 vCPU y 32 GB de RAM tiene un importe de 2079,76 USD (dólares americanos) por mes. Dentro de este importe también se tiene en cuenta el precio del storage SSD (2 TB). Este tipo de servidor sirve para generar el sistema blockchain inicial, desde el punto de vista de la organización maestra, o para disponibilizar cualquier tipo de nodo perteneciente a la cadena de bloques.

Dependiendo de cada organización, deberán contar con la infraestructura on-premise similar o se deberá contratar algún servicio dedicado de hosting para cumplir los requerimientos.

Con respecto a la instancia necesaria para la aplicación biométrica, un servidor con 8 vCPU, 16 GB de memoria RAM, y con 2 TB iniciales de storage, será necesario para disponibilizar el servicio de matriculación biométrico. De la misma forma, en caso de optar por la nube de AWS, la instancia tiene un importe a la fecha de 399.99 USD.- por mes. Ver Figura 14 con el detalle.

Figura 14

Cálculo de costos asociados al servidor para la aplicación biométrica



Nota. Cotización generada utilizando la calculadora de Amazon Web Services.

La diferencia de costos entre instancias se debe a que los servidores para la cadena de bloques necesitan satisfacer ciertos requerimientos de performance relacionados principalmente con el uso de CPU y con el storage. Con respecto al licenciamiento del sistema operativo, se utilizará Ubuntu Desktop o Ubuntu Linux Server, por lo tanto, al ser gratuito, no se requiere agregar costos adicionales. Solamente se agregan costos adicionales por el uso de

herramientas como gitlab (https://about.gitlab.com/) para administrar el versionado y para las IDE de desarrollo. A continuación, se muestra la Tabla 6 con los costos asociados al proyecto.

Tabla 6

Costos asociados al proyecto

Recursos Materiales			
Descripción	Cantidad Necesaria (unidades)	Precio por recurso (mensual)	Subtotal (anual)
Servidor Inicial para Blockchain	1	2079.76	24957.12
Servidor para aplicación biométrica	1	399.99	4799.88
Otras licencias de software (gitlab, y demás IDEs)	1	100	1200
Capital Humano			
Descripción	Cantidad Necesaria (horas)	Precio por hora (mensual)	Subtotal (anual)
Líder de Proyecto SR	1920	10	19200
Programadores SR	1600	8	12800
Administradores TI	1920	8	15360
Consultores Funcionales	1200	7	8400
Soporte de Nivel 1	1000	6	6000
		Total	92717

Tal como podemos ver en la Tabla 6, para los recursos materiales, se listan las unidades necesarias de cada recurso, junto con los precios unitarios y subtotales en USD (dólares americanos). Con respecto al capital humano necesario para llevar adelante el proyecto, se listan los puestos, las horas necesarias por cada uno de ellos, el precio por hora y los subtotales. Finalmente se calcula el monto total requerido para la implementación, que es de USD 92717.- al momento de haber realizado el cálculo de los servidores y recursos.

Conclusiones

Los usuarios tramposos han sido siempre un problema que aqueja a la comunidad en línea. Por esta razón, nuevas soluciones deberán pronto implementarse para asegurar la continuidad del producto software y de satisfacer los intereses de los stakeholders. Por tal motivo, la propuesta de intervención mencionada en este trabajo brinda una herramienta para disuadir y prohibir a los usuarios que degradan la experiencia de juego, e impactan en el ámbito multijugador en línea. El hecho de que un usuario sea reportado por infringir las normas, lo expone frente a todas las demás organizaciones que forman parte del consorcio, pudiendo acceder a la información del caso.

Dentro de los beneficios a obtener utilizando esta solución se tiene la continuidad del producto software y la permanencia de los usuarios en las plataformas, evitando la reducción de la base de usuarios por abandonos debido a la mala experiencia de juego. Por otro lado, la propuesta evita las pérdidas económicas, que surgen a partir de las continuas actualizaciones que requieren los sistemas antitrampas para garantizar la detección de usuarios conflictivos. Se busca disminuir la cantidad de usuarios tramposos debido a las consecuencias de ser expuestos y prohibidos globalmente en otras plataformas. En la actualidad existen soluciones antitrampas para combatir esta problemática, pero estas no terminan de ser completamente eficientes, ya que requieren continuas actualizaciones para cumplir con su objetivo, pueden ser intrusivos como los que funcionan a nivel kernel e incluso permiten la reincidencia.

Es importante recordar que, si bien la información obtenida a partir de esta solución es de gran utilidad para la toma de decisiones, el derecho de admisión se le delega a cada stakeholder en particular. Al identificar unívocamente a una persona mediante la biometría, es posible llevar un control de su comportamiento más preciso y luego tomar medidas en contra del uso de trampas online. A través de una red blockchain de consorcio, diferentes reportes son realizados, informando prohibiciones de los usuarios por el uso de trampas. Luego esta información es compartida, por todos los miembros que conforman la red. Los reportes mencionados, son generados a partir de los diferentes eventos que puedan producirse en los sistemas antitrampas actuales, y a través de una aplicación que funciona a modo de cliente con el sistema Blockchain. Por tal motivo, no se busca reemplazar a los sistemas antitrampas actuales, sino complementarlos.

Cada organización que forma parte del consorcio blockchain, es capaz de consultar periódicamente la red, para luego sincronizar las prohibiciones que empresas vecinas puedan compartir acerca de un usuario. La propuesta de intervención mencionada en este trabajo no solo busca aplicar a los videojuegos online, sino que también se encuentra abierto a cualquier otro ámbito en donde se requiera llevar un control de las acciones de los usuarios.

Trabajos futuros por realizar

En esta sección se revisarán algunos de los trabajos que podrían realizarse a partir de lo presentado en este escrito y que se relacionan con las limitaciones presentadas y con las mejoras que puedan producirse a partir de nuevas soluciones.

Es importante determinar un estándar y una modalidad biométrica alternativa para poder identificar unívocamente a los usuarios en caso de no poder utilizar el rostro por algún motivo o política en particular de las organizaciones. Aquellos interesados deberán revisar y optar por el algoritmo más confiable para el reconocimiento biométrico que se encuentre en el mercado.

Con respecto a la cadena de bloques y dependiendo de los intereses de los stakeholders, diferentes organizaciones podrían asociarse y establecer un canal con múltiples participantes. Se deberá estudiar en cada caso, los permisos a asignar a cada organización, si corresponde generar otro canal, o en definitiva si se requiere construir otra red blockchain nueva. Por el motivo mencionado, es necesario controlar el blockchain de consorcio a medida que nuevos miembros colaboradores son adheridos.

Por otro lado, se deberá establecer algún procedimiento para la revisión de los casos en que se haya reportado algún usuario incorrectamente en el blockchain, y para aquellos usuarios que deseen desasociarse del sistema biométrico.

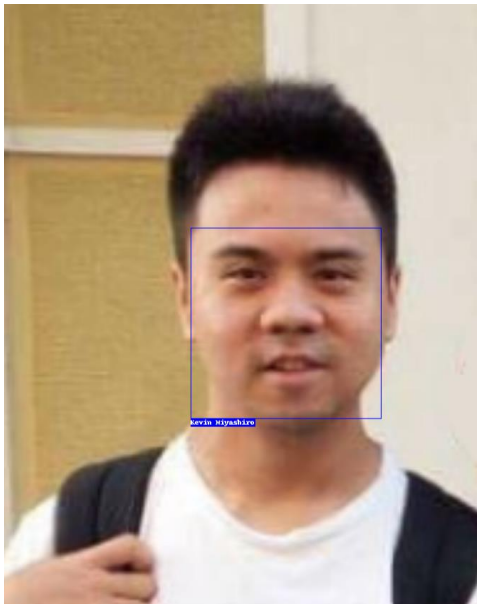
Como se comentó en las conclusiones, no necesariamente la propuesta aplica al ámbito de videojuegos multijugador en línea. Se deberán estudiar los casos en donde se requiera llevar un control de las acciones de los usuarios y amerite conformar un sistema Blockchain, junto con datos biométricos para aprovechar las características que brinda esta solución.

Anexo I

A continuación, se muestra el cuadro o bounding box, delimitando un rostro dentro de una imagen, para luego aplicar la técnica de histograma de gradientes orientados (HOG) en el reconocimiento facial.

Figura 15

Bounding Box o Cuadro Delimitador



Nota. Figura delimitando las coordenadas de un rostro para luego extraer las características.

Anexo II

En la Tabla 7, se puede apreciar la gran similitud entre Will y William West al aplicar el método de Alphonse Bertillon. Como se puede apreciar, la diferencia en la distancia euclidiana es mínima.

Tabla 7

Comparación de los componentes de Bertillon entre Will y William West

Will West	William West
178,5	177,5
187,0	188,0
91,2	91,3
19,7	19,8
15,8	15,9
14,8	14,8
6,6	6,5
28,2	27,5
12,3	12,2
9,7	9,6
50,2	50,3

Nota. Obtenido de *La Biometría para la identificación de personas* (p. 10), por Serratosa F., 2012, UOC.

En la Figura 16 a continuación, podemos apreciar la similitud entre las fotografías.

Figura 16

Fotografías de Will y William West durante su encarcelamiento

Nota. Obtenido de *La Biometría para la identificación de personas* (p. 11), por Serratosa F., 2012, UOC.

Acrónimos

API: Application Programming Interface, se refiere a la interfaz de programación de aplicación software que actúa de intermediario para poder establecer una comunicación e intercambiar datos.

CA: Certificate Authority, o autoridad certificante, corresponde al componente dentro de una red blockchain para la emisión de certificados pertenecientes a cierta organización.

DLL: Dynamic-Link Library, son aquellos archivos que poseen código ejecutable que se ejecuta según las necesidades del desarrollador.

FPS: First Person Shooters, corresponden a la categoría de videojuegos de disparos en primera persona.

HOG: Histogram of Gradients, se refiere al método de reconocimiento facial basándose en la orientación de los gradientes.

IDE: Entorno de desarrollo integrado, se refiere al software utilizado por los desarrolladores para escribir código de programación.

ISP: Internet Service Provider, es el proveedor del servicio de Internet doméstico.

OCx: Organización Colaboradora “x”, son todas aquellas organizaciones que formarán parte del consorcio, pudiendo consultar y reportar usuarios infringiendo las normas. Se refiere a “x”, un número asignado dentro del consorcio de empresas en el sistema blockchain.

OM: Organización Maestra, se refiere a aquella que tendrá el papel de impulsora de la implementación del sistema de consorcio.

SSD: Solid State Disk, se refiere a los discos de estado sólido, para almacenamiento de información.

SHA: Secure Hash Algorithm. Son un conjunto de funciones hash utilizadas durante el proceso de encriptación.

SO: Sistema Operativo. Hace referencia al conjunto de instrucciones, programas y servicios para controlar el hardware de la computadora.

SSr: Semi Senior, se refiere al grado de seniority de un recurso con conocimientos y experiencia intermedios dentro un campo de aplicación en particular.

SR: Senior, se refiere al grado de seniority de un recurso que forma parte de un equipo de trabajo. En este caso, se refiere a un recurso con un nivel avanzado de conocimientos en el campo.

VPN: Virtual Private Network, es una red privada desplegada sobre una red pública.

Referencias

Amazon Web Services. (s.f). *What is DNS*? [¿Qué es el DNS?]

https://aws.amazon.com/route53/what-is-dns/

Asad, A. N., Elahi T. M., Hasan A. A., Yousuf, A. M. (28-29 de noviembre de 2020).

Permission-Based Blockchain with Proof of Authority for Secured Healthcare Data Sharing [Blockchain basada en permisos con Prueba de Autoridad para Compartimiento de Datos Seguros de Salud]. *2nd ICA ICT*, Dhaka, Bangladesh.

AWS Pricing Calculator. (s.f). *Estimate the cost for your architecture solution*

[Estime el costo de su solución de arquitectura] Recuperado de https://calculator.aws/#/ el 2 de Noviembre de 2023.

Consalvo, M. (2007). *Cheating Gaining Advantage in Video Games*

[Las Trampas, obteniendo ventajas en Video Juegos]

Counter Strike 2 (2023). *The largest technical leap in Counter-Strike history.*

[El mayor salto técnico en la historia de Counter-Strike]. Recuperado de https://www.counter-strike.net/ el 3 de Noviembre de 2023.

Duh, H. B.-L. y Chen, V. H. H. (2009). *Cheating behaviors in online gaming*

[El comportamiento de los tramposos en juegos en línea]

Epic Games. (2023). *Don't bear with the cheaters*. [No tolere los tramposos]

https://www.easy.ac/en-us/#about

GamersClub (2023). *Conta Verificada Gamers Club*. [Cuenta Verificada Gamers Club]

https://verificado.gamersclub.com.br/

GamersClub (s.f.). ¡Gamers Club es la plataforma de CSGO más grande de América Latina!

Recuperado el 7 de Noviembre de 2023 de https://csgo.gamersclub.gg/

Geitgey, A. (20 de Febrero de 2020). *face-recognition 1.3.0*. Recognize faces from Python or

From the command line. [Reconocer rostros desde Python o desde la línea de comandos]. https://pypi.org/project/face-recognition/

Hyperledger Fabric. (2017). *Fabric CA User's Guide* [Guía del usuario de la Autoridad Certificante de Fabric]. https://hyperledger-fabric-ca.readthedocs.io/en/release-1.4/users-guide.html

Hyperledger Fabric. (2022). *The Ordering Service.* [El Servicio de Ordenamiento] https://hyperledger-fabric.readthedocs.io/en/release-2.2/orderer/ordering_service.html

Hyperledger Fabric (2022b). *Application.* [Aplicación]. https://hyperledger-fabric.readthedocs.io/en/release-2.2/developapps/application.html

Hyperledger Fabric. (2023). *What is a Chaincode?* [¿Qué es una cadena de código?] Recuperado el 4 de Noviembre de 2023 de https://hyperledger-fabric.readthedocs.io/en/latest/chaincode4ade.html

Hyperledger Fabric. (2023b). *Channels* [Canales] Recuperado el 4 de Noviembre de 2023 de https://hyperledger-fabric.readthedocs.io/en/latest/channels.html

Hyperledger Fabric. (2023c). *Endorsement Policies* [Políticas de Endorsamiento]. https://hyperledger-fabric.readthedocs.io/en/latest/endorsement-policies.html

Hyperledger Foundation. (s.f). *Hyperledger Fabric: Open, Proven, Enterprise-grade DLT*. [Hyperledger Fabric: Libro Mayor Abierto y Probado de Grado Empresarial].

Idwall (2021). *Aumente a segurança na validação online dos seus usuarios*. [Aumente la seguridad con la validación online de sus usuarios]. Recuperado el 4 de Noviembre de 2023 de https://idwall.co/meuid-empresas

IEEE Spectrum. (17 de Febrero de 2010). *Steamed: Valve Software Battles Video-game Cheaters. [Al vapor: El software de Valve Batalla contra los Tramposos en los Videojuegos]*. https://spectrum.ieee.org/steamed-valve-software-battles-videogame-cheaters

Kalra, S., Sanghi, R. y Dhawan M. (2018). Blockchain-based Real-time Cheat Prevention and Robustness for Multi-player Online Games. [Prevención de Trampas en Tiempo Real y robusto para Juegos Multijugador en línea basado en Blockchain]

Khalifa, S. (2016). *Machine Learning and Anti-Cheating in FPS Games*. [Tesis de Maestría, University College London]. ResearchGate.

Lee, H., Song, C., Kang, B. B. (2018). Lord of the x86 Rings: A Portable User Mode Privilege Separation Architecture on x86. [El señor de los x86 anillos: Un modo usuario portable]

Lehtonen, S. (2020). *Comparative Study of Anti-cheat Methods in Video Games* [Estudio Comparativo de Métodos Antitrampas en videojuegos]

Lukas, M., Tominic, I. y Bernik, A. (2022) Anticheat System based on Reinforcement Learning Agents in Unity [Sistema Antitrampas basado en el Reforzamiento de Agentes de Aprendizaje en Unity]

Marín, R. M., Uribe, R. C. J., Morales, O. C. J, (2009). *A glance to the biometric.* [Una mirada a la biometría]. *Universidad del Magdalena.*

Miyashiro, K. (2023). *Sistema de Penalización basado en Datos Biométricos y Blockchain Para Videojuegos*. XXI CIITI 2023, Congreso Internacional en Innovación Tecnológica Informática.

Miyashiro, K. (2023b). *Método GetAllReportes, para retornar todos los reportes generados en Blockchain. Blockchain.* Golang. Visual Studio Code 1.8.3.

Miyashiro, K. (2023c). *Smart Contract para la interacción con el sistema Blockchain.* Golang. Visual Studio Code 1.8.3.

Mohanta, B., Panda, S. y Jena D. (2018). An Overview of Smart Contract and Use cases in Blockchain Technology. [Una descripción general de los Contratos Inteligentes y los Casos de uso en la tecnología Blockchain]

Ongaro, D. y Ousterhout, J. (19-20 de junio de 2014). In Search of an Understandable Consensus Algorithm [En la Busqueda de un Algoritmo de Consenso Entendible]. *USENIX Annual Technical Conference*. Philadelphia, PA.

Pato, J. N. y Millett L. I. (2010). *Biometric Recognition: Challenges and Opportunities* [Reconocimiento Biométrico: Desafíos y Oportunidades]

Pierro, D. M. (2017). *What is Blockchain?* [¿Qué es Blockchain?] *Computing in Science & Engineering, vol. 19, n. 5.*

Postel, J. (28 de agosto de 1980). *RF 768 User Diagram Protocol. IETF Datatracker*. https://datatracker.ietf.org/doc/html/rfc768

Postel, J. (septiembre de 1981). *RF 793 Transmission Control Protocol. IETF Datatracker*. https://datatracker.ietf.org/doc/html/rfc793

Rahmad, C., Asmara, R. A., Putra, D. R. H., Dharma, I., Darmono, H., Muhiqqin, I. (2020). Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection [Comparación del Clasificador Haar en Cascada de Viola-Jones con el Histograma de Gradientes Orientados (HOG) para detección facial]

Reglamento (UE) 2016/679 Del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (reglamento general de protección de datos). Diario Oficial de la Unión Europea, L65 y L66, de 27 de abril de 2016. https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679

Riot Games (2020). *What is Vanguard?* [¿Qué es el Vanguard?] https://support-valorant.riotgames.com/hc/en-us/articles/360046160933-What-is-Vanguard-

Riot Games (s.f.). */dev/null: Anti-cheat kernel driver* [/dev/null: Antitrampas basado en el controlador del kernel]

https://images.contentstack.io/v3/assets/blt731acb42bb3d1659/blt28df09c60292cc6c/5e3101fd14bf1c024132f20e/Kernel_Drivers_Image.png

Serratosa, F (2012). *La biometría para la identificación de las personas*. Universidad Oberta De Catalunya.

Spijkerman, R y Ehlers, E. M. (2020) Cheat Detection in a Multiplayer First-Person Shooter Using Artificial Intelligence Tools [Detección de Trampas en un juego Multijugador en persona usando Inteligencia Artificial]

The Ohaio State University (2021). *What's a MAC Address and how do I find it?* [¿Qué es una Dirección MAC y cómo la encuentro?] https://slts.osu.edu/articles/whats-a-mac-address-and-how-do-i-find-it/

Valve Corporation. (s.f.). *What is VAC?* [¿Qué es el VAC?] https://help.steampowered.com/en/faqs/view/571A-97DA-70E9-FF74

Van de Ven, B. (2023). *Cheating and anti-cheat system action impacts on user experience* [Las trampas y la acción de los sistemas antitrampas impactan en la experiencia del usuario]

Viola, J. y Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features [Detección Rápida de Objetos utilizando una Cascada Potenciada de Características Simples]

Ward M. (31 de octubre de 2005). *Warcraft game maker in spying row*. BBC News. http://news.bbc.co.uk/2/hi/technology/4385050.stm

Zhang, Q. (2021). Improvement of Online Game Anti-Cheat System based on Deep Learning [Mejora del Sistema Antitrampas basado en Deep Learning]

Zheng, Z., Xie, S., Dai, H., Chen, X. y Wang, H. (2017). *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*. 2017 [Un Resumen de la Tecnología Blockchain: Arquitectura, Consenso y Futuras Tendencias]. IEEE 6th International Congress on Big Data.