Universidad Andrés Bello

Facultad de Ingeniería

Ingeniería Civil Informática

PROYECTO DE SISTEMA ANTI-TRAMPAS DE VIDEOJUEGOS CON DEEP LEARNING

Memoria de título para optar al título de Ingeniería Civil Informática

Autor:

Alejandro Andrés Matus Silva

Profesor:

Ernesto Eduardo Vivanco Tapia

Santiago, Chile

Octubre,2024

Agradecimientos

Aquí van los agradecimientos.

Índice de Contenidos

Resumen

Aquí va el resumen en español. Este resumen debe incluir los objetivos, métodos, resultados y conclusiones principales del estudio en no más de 250 palabras.

Abstract

Aquí va el resumen en inglés. Este resumen debe incluir los objetivos, métodos, resultados y conclusiones principales del estudio en no más de 250 palabras.

1 Introducción

Los cheats o trampas en los videojuegos han evolucionado significativamente desde los primeros trucos y códigos incorporados en los juegos de un solo jugador hasta sistemas complejos de trampas que afectan la experiencia en juegos multijugador en línea. Estos cheats, que van desde códigos que permiten vida infinita hasta sistemas avanzados como aimbots y wallhacks, se han convertido en una amenaza para la experiencia justa de los jugadores y la reputación de las compañías de videojuegos.

En los juegos multijugador, la utilización de cheats afecta la competencia entre jugadores y representa una amenaza para la estabilidad de las plataformas de juego en línea, generando frustración en la comunidad de jugadores y pérdida de confianza en los desarrolladores. Según Mohr et al. (2011), el uso de trampas tiene un impacto considerable en la industria, afectando la confianza de los usuarios y las ganancias de los desarrolladores, quienes enfrentan costos adicionales para implementar soluciones de detección y sanción de trampas.

1.1. Contexto

Los videojuegos han dejado de ser una actividad exclusivamente recreativa para convertirse en un fenómeno cultural y una de las industrias de entretenimiento más rentables a nivel global. Juegos multijugador como Valorant, League of Legends, Fortnite, Call of Duty, y Counter-Strike congregan diariamente a millones de usuarios, fomentando comunidades competitivas e interacciones sociales en entornos virtuales. Sin embargo, este auge también ha traído consigo el crecimiento del uso de cheats o trampas, herramientas que otorgan ventajas injustas a ciertos jugadores y que alteran la experiencia del resto de la comunidad. Estos cheats incluyen desde manipulaciones simples como "wallhacks" y "aimbots", hasta trampas complejas que explotan vulnerabilidades del sistema operativo o del juego.

El impacto de estas trampas se extiende más allá de las partidas individuales. A nivel industrial, generan pérdida de confianza en los desarrolladores, reducen la popularidad de los juegos afectados y generan costos adicionales asociados a la implementación y actualización de sistemas de detección y prevención. Además, los desarrolladores de trampas también evolucionan constantemente, creando herramientas cada vez más sofisticadas que buscan evadir los sistemas tradicionales de detección. Esto plantea la necesidad urgente de soluciones innovadoras y adaptativas que garanticen un entorno de juego justo.

A la par, también se han popularizado sistemas anticheat que operan a nivel de kernel, como Vanguard, Ricochet y Easy Anti-Cheat (EAC). Si bien estos sistemas ofrecen un nivel de protección profundo al tener acceso a la capa más baja del sistema operativo, también han generado controversias debido a preocupaciones sobre la privacidad y la seguridad. La

posibilidad de que vulnerabilidades en estos sistemas sean explotadas por atacantes o que recopilen datos sensibles de los usuarios plantea un dilema entre seguridad y privacidad que debe ser considerado al desarrollar nuevas soluciones anticheat.

1.2. Importancia y Contribución

El desarrollo de un sistema anticheat eficaz y ético es esencial para proteger la integridad de los videojuegos multijugador y salvaguardar la experiencia de los jugadores. En el panorama actual, los sistemas de detección predominantes, como los basados en firmas o análisis estadístico, presentan limitaciones significativas:

- Requieren actualizaciones constantes para detectar nuevas trampas, lo que los vuelve reactivos más que preventivos.
- Pueden generar falsos positivos que penalizan injustamente a jugadores legítimos.
- Algunos sistemas, como los basados en el acceso a nivel de kernel, comprometen la privacidad de los usuarios, lo que ha generado controversias importantes.

El uso de inteligencia artificial (IA) representa una solución disruptiva en este ámbito. La IA permite analizar patrones de comportamiento complejos y detectar anomalías en tiempo real, adaptándose rápidamente a nuevas amenazas sin necesidad de depender de firmas predefinidas. Además, esta tecnología puede implementarse de manera ética, garantizando la privacidad de los jugadores mientras mejora significativamente la eficacia de los sistemas anticheat. Por ejemplo, sistemas como VACNet, desarrollado por Valve, utilizan redes neuronales profundas para analizar patrones de comportamiento en juegos como *Counter-Strike: Global Offensive* y *Counter-Strike 2*. Este enfoque ha demostrado ser efectivo para detectar trampas avanzadas como aimbots, estableciendo un precedente sobre cómo la IA puede transformar la seguridad en los videojuegos.

Este proyecto busca contribuir al desarrollo de la industria de videojuegos mediante la implementación de un sistema anticheat basado en IA. Su principal contribución radica en la combinación de eficiencia operativa, capacidad adaptativa y consideraciones éticas, estableciendo un nuevo estándar en la seguridad de los juegos multijugador.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema anticheat basado en inteligencia artificial para detectar y prevenir trampas en juegos multijugador, garantizando una experiencia de juego justa y respetuosa con la privacidad de los usuarios.

1.3.2. Objetivos Específicos

1. Explicar el contexto de los antitrampas en los videojuegos y la problemática que estos suponen para un entendimiento claro del proyecto.

2. Adjuntar la documentación adecuada para desarrollar de forma correcta el proyecto.

3. Diseñar un modelo de aprendizaje automático capaz de detectar patrones anómalos relacionados con trampas.

4. Desarrollar una interfaz intuitiva y eficaz para implementación del sistema, cumpliendo con la ISO 27001, 27002 y 27701 para garantizar la seguridad y privacidad de la información. Esto permitirá a los jugadores no correr riegos de intrusión en sus datos.

5. Implementar una arquitectura cliente-servidor que permita la detección de trampas en videojuegos en tiempo real.

1.4. Limitaciones del Proyecto

Considerando las características técnicas y la implementación del sistema anti-trampas, se han identificado varias limitaciones, entre las que destacan:

1. **Disponibilidad y Acceso a Datos:** La calidad del modelo de la IA depende de la disponibilidad de datos representativos sobre trampas. La recopilación de estos datos puede verse limitada por cuestiones legales y técnicas.

2. **Variabilidad de las Trampas o Cheats:** La adaptabilidad del sistema es crucial, dado que los desarrolladores de cheats crean constantemente nuevas formas de evadir las medidas de detección.

3. **Compatibilidad de Hardware y Software**: Asegurar que el sistema funcione en diferentes dispositivos de gama baja y media, sin dañar la compatibilidad entre los archivos y el sistema.

4. **Infraestructura de los videojuegos:** El Proyecto está condicionado por la infraestructura de los videojuegos, dado que estos puedes estar manejados de cierta forma que se vuelva incompatible o cause problemas de rendimiento.

5. **Falsos positivos en detección:** Penalizar a jugadores legítimos puede dañar la reputación del servicio y generar descontento en comunidades de videojuegos.

6. **Riesgos asociados a sistemas de tipo Kernel:** Aunque ofrecen ventajas en la detección, generan controversias debido a los problemas de privacidad y vulnerabilidades.

2 Marco Teórico

En este capítulo se presentarán las herramientas utilizadas para el desarrollo del sistema y se describirá la metodología a utilizar para llevar a cabo cada etapa del proyecto.

2.1. Metodología de Trabajo

La metodología de un proyecto es el conjunto de técnicas, herramientas y procesos que guían la gestión de un proyecto de inicio a fin, permitiendo planificar, ejecutar y controlar proyectos de manera efectiva.

2.1.1. Calidad de Software y Norma ISO

La Organización Internacional de Normalización (ISO) establece estándares que garantizan la calidad, seguridad y eficiencia de productos, servicios y sistemas. En el desarrollo de este proyecto, las normas ISO relevantes abarcan tanto aspectos específicos de las interfaces como principios generales del software. A continuación, se describen las principales normas aplicables:

1. Norma ISO 27001:

   Esta norma es el estándar internacional más reconocido para gestionar la seguridad de la información. Proporciona un marco estructurado para implementar, mantener y mejorar continuamente un Sistema de Gestión de Seguridad de la Información (SGSI). Este estándar se centra en la protección de la confidencialidad, integridad y disponibilidad de los datos mediante un conjunto de controles específicos y la revisión constante de su eficacia.

   Implementar la ISO 27001 permite proteger la confidencialidad y seguridad de los datos recolectados del cliente tanto como logs de trampas o patrones de comportamiento, para garantizar que las conexiones entre el cliente y el servidor sean seguras, minimizando riesgos de interceptación de datos.

2. Norma ISO 27002:

   Esta norma detalla los controles de seguridad descritos en la ISO 27001 y sirve como una guía para implementar las mejores prácticas de la protección de la información, ayudando a las organizaciones a prevenir accesos no autorizados, manipulación de datos y perdida de la

información. Este estándar pone énfasis en áreas como la seguridad de redes, autentificación, control de accesos y monitoreo constante de sistemas. Fundamental para garantizar que las medidas técnicas se mantengan alineadas con las amenazas actuales y emergentes.

La implementación de la ISO 27002 asegura solo el personal autorizado puede acceder a datos sensibles y las configuraciones de detección y registros de trampas, implementando cifrado en las comunicaciones para proteger datos en transito entre el cliente y el servidor.

3. Norma ISO 27701:

Esta norma amplia las normas 27001 y 27002 con un enfoque mas especifico en la privacidad de los datos personales. Esta norma esta diseñada para garantizar que las organizaciones cumplan con las regulaciones como el Reglamento General de Protección de Datos (GDPR). Este estándar establece los principios para recopilar, almacenar y procesar datos de forma ética y legal, reduciendo riesgos de exposición y abuso de la información personal. También define los roles y responsabilidades para los controladores y procesadores de datos.

La implementación de la ISO 27701 asegura proteger los datos personales de los jugadores, garantizando el anonimato cuando sea necesario y asegurar que las actividades de detección no invalidan la privacidad del usuario.

4. Norma ISO 25010:

Esta norma pertenece a la familia ISO 25000 y proporciona un modelo de calidad mas amplio que abarca no solo interfaces, sino todo el software desarrollado y que cumpla con los requisitos del cliente y opere de manera eficiente en diferentes entornos.

Funcionalidad: El sistema debe cumplir correctamente con los requisitos especificados, incluyendo la recolección y análisis de datos del anti-trampas y generar reportes detallados de los patrones analizados por la IA

Fiabilidad: El software debe ser estable y consistente, asegurando que tanto la IA como los dispositivos con el anti-trampas funcionen adecuadamente en diferentes plataformas.

Usabilidad: Diseñar una interfaz amigable que permita a los administradores visualizar patrones de IA. Proveer herramientas visuales, como gráficos y dashboards, para asegurar que las herramientas del software sean fáciles de usar y accesibles.

Eficiencia de Rendimiento: El sistema debe operar de manera óptima, gestionando grandes volúmenes de datos en tiempo real con una latencia mínima y operando eficientemente en entornos con recursos limitados.

Compatibilidad: La solución debe integrarse sin problemas en los servidores de juegos sin ningún problema de software o hardware e interoperabilidad con APIs externas que complementen capacidades de detección.

Seguridad: Debe proteger los modelos de IA y datos utilizados para el entrenamiento y la inferencia contra manipulaciones externas o intentos de ingeniería inversa. Garantizar la seguridad de los datos y privacidad de usuario.

Mantenibilidad: Facilita la actualización y corrección de errores, garantizando que el sistema evolucione según las necesidades actuales.

Portabilidad: El software debe adaptarse a diferentes entornos tecnológicos, permitiendo su uso en diversos dispositivos y plataformas.

Implementar la ISO 25010 asegura que todos los componentes del software, desde la IA hasta la gestión de datos, cumplan con estándares internacionales de calidad y funcionalidad integral.

5. Norma ISO 15408:

Esta norma ISO, conocida como Common Criteria, proporciona un marco para evaluar y certificar la seguridad de sistemas de TI. Se utiliza para analizar componentes clave, identificando vulnerabilidades y asegurando que el sistema sea robusto frente a ataques. Este estándar incluye la evaluación de algoritmos, mecanismos de autentificación y medidas de protección física y lógica.

Implementar la ISO 15408 asegura que la validación de los algoritmos de detección sea resistente a trampas avanzadas y certificar que el sistema cumple con estándares internacionales de seguridad.

6. Norma ISO 21500:

Esta Norma ofrece un marco para la planificación, ejecución y controlar proyectos de manera eficiente. Es útil para organizar equipos, asignar recursos y medir el progreso del proyecto. Este estándar divide la gestión de proyectos en procesos principales como inicio, planificación, ejecución, monitoreo y cierre, asegurando que los objetivos se logren dentro del tiempo y presupuesto establecidos.

Implementar la ISO 21500 asegura que la planificación de Sprints para garantizar el desarrollo continuo del sistema y el monitoreo del progreso mediante métricas claras y revisiones regulares para la gestión de los plazos predefinidos del proyecto.

7. Norma ISO 31000:

Esta Norma proporciona los principio y directrices para identificar, analizar y gestionar riesgos en cualquier organización o proyecto de manera sistemática, mejorando la toma de decisiones y garantizando la resiliencia frente a incertidumbres.

La implementación de la ISO 31000 asegura un enfoque estructurado para anticipar y manejar los riesgos asociados con el desarrollo e implementación del sistema. Al adoptar esta norma, se asegura que el sistema sea resiliente frente a desafíos técnicos, éticos y operativos, lo que contribuye a crear un entorno de juego seguro y confiable.

2.1.2. Metodología Scrum

Scrum es un marco de trabajo diseñado para gestionar y desarrollar proyectos, facilitando a individuos, equipos y organizaciones la creación de valor mediante soluciones flexibles para problemas complejos (Schwaber & Sutherland, 2020). Este enfoque se basa en un proceso iterativo e incremental, caracterizado por entregas frecuentes y programadas en intervalos de tiempo cortos y constantes, lo que lo convierte en una metodología eficaz para abordar proyectos de alta complejidad.

2.1.3. Equipo Scrum

El equipo Scrum es el núcleo del marco y debe ser lo suficientemente ágil para adaptarse a las necesidades del proyecto. Está compuesto por tres roles fundamentales: el Scrum Master, el Product Owner (propietario del producto) y los desarrolladores. Este equipo es responsable de diversas actividades, como la colaboración, verificación, mantenimiento, operación, investigación, experimentación, desarrollo y cualquier tarea necesaria para el éxito del proyecto (Schwaber & Sutherland, 2020).

- **Scrum Master:** Es el líder del equipo encargado de garantizar que los principios y eventos de Scrum se entiendan y se cumplan. Este rol incluye capacitar a los miembros, facilitar las reuniones, asegurar el cumplimiento de los plazos establecidos y eliminar 12 impedimentos que puedan obstaculizar el flujo de trabajo. También apoya en la planificación tanto del proyecto como de las reuniones.

- **Product Owner:** Se enfoca en maximizar el valor del producto gestionando y organizando el Product Backlog. Este rol asegura que las ideas del cliente estén estructuradas, detalladas y priorizadas adecuadamente para guiar al equipo de desarrollo.

- **Desarrolladores:** Compuesto por un equipo de entre 3 y 9 profesionales, su responsabilidad principal es crear el producto y generar incrementos basados en los

elementos del Product Backlog. Además, se encargan de planificar cada Sprint y ajustar el plan diariamente para alcanzar los objetivos establecidos.

2.1.4. Eventos de Scrum

Los eventos en Scrum están diseñados para mantener la regularidad y minimizar la necesidad de reuniones adicionales, creando una estructura eficiente para el proyecto (Schwaber & Sutherland, 2020). Estos son cinco:

- **Sprint:** Es el núcleo del marco Scrum y contiene todos los demás eventos. Tiene una duración fija y, al finalizar, el siguiente Sprint comienza inmediatamente. No se permiten cambios que comprometan el objetivo del Sprint.

- **Planificación del Sprint:** Es la reunión inicial donde el equipo de desarrolladores y el Product Owner responden a tres preguntas clave: ¿Por qué este Sprint es valioso?, ¿Qué se puede hacer en este Sprint?, y ¿Cómo se realizará el trabajo elegido?

- **Scrum Diario:** Son reuniones diarias breves (máximo 15 minutos) donde se revisa el progreso y se hacen ajustes necesarios en el Sprint Backlog.

- **Revisión del Sprint:** Se realiza al final del Sprint para inspeccionar los resultados y determinar ajustes para futuros Sprints. En esta reunión se evalúa lo logrado y se definen próximos pasos.

- **Retrospectiva del Sprint:** Es la última reunión del Sprint, llevada a cabo tras la revisión del Sprint y antes de la planificación del siguiente. Se identifican áreas de mejora para aumentar la eficacia, calidad y planificación del equipo.

2.1.5. Artefactos de Scrum

Scrum utiliza tres artefactos principales que buscan maximizar la transparencia del trabajo y proporcionar información clave:

- **Product Backlog:** Es una lista priorizada de todo lo necesario para el desarrollo y mejora del producto. Incluye los objetivos valorados por el cliente y el costo estimado de cada elemento**.**

- **Sprint Backlog:** Contiene las tareas seleccionadas del Product Backlog que serán desarrolladas durante un Sprint. Ayuda a subdividir el proyecto en partes más pequeñas, facilitando la identificación de problemas o tareas pendientes**.**

- **Incremento**: Es la suma de los elementos completados del Product Backlog durante un Sprint. Cada incremento es acumulativo y debe ser funcional junto con los

anteriores. Estos se presentan durante la revisión del Sprint, y pueden generarse varios incrementos dentro de un mismo Sprint.

(Schwaber & Sutherland, 2020)

2.1.6. Modelo de Vistas 4+1

El modelo de vistas 4+1 es una metodología utilizada para describir la arquitectura de sistemas de software, basada en múltiples perspectivas para abordar diferentes aspectos del sistema. Su nombre deriva de las cuatro vistas principales: lógica, física, de procesos y de desarrollo, más una vista adicional, denominada "+1", que integra las cuatro anteriores y se conoce como la vista externa.

2.1.6.1. Vista Lógica

Esta vista representa las funciones y la estructura del sistema, definiendo las operaciones que el sistema debe realizar y los servicios que ofrece. Su representación común se realiza mediante diagramas de secuencia UML (Moya, 2012).

2.1.6.2. Vista de Procesos

La vista de procesos detalla los procesos internos del sistema y cómo interactúan entre sí. Considera aspectos clave como disponibilidad, tolerancia a fallos y desempeño. También se representa mediante diagramas UML (Moya, 2012).

2.1.6.3. Vista de Desarrollo

Esta vista aborda la organización del sistema desde la perspectiva del programador, mostrando cómo se dividen los componentes del sistema y las dependencias entre ellos. Es comúnmente representada mediante diagramas UML (Moya, 2012).

2.1.6.4. Vista Física

Desde el punto de vista del ingeniero de sistemas, esta vista describe los componentes físicos o de hardware involucrados en el sistema. Los diagramas UML son útiles para complementar esta vista (Moya, 2012).

2.1.6.5. Vista Externa

La vista externa actúa como un integrador de las cuatro vistas principales, describiendo las interacciones entre objetos y procesos del sistema. Ofrece una visión general de cómo se usará el sistema, y se representa típicamente mediante diagramas UML de casos de uso (Moya, 2012).

2.1.7. UML (Lenguaje Unificado de Modelado)

El UML (Unified Modeling Language) es un lenguaje estándar utilizado para modelar, visualizar y construir la arquitectura de sistemas. Este lenguaje permite representar elementos como actividades, procesos de negocio y esquemas de bases de datos.

2.1.8. BPMN (Business Process Model and Notation)

El BPMN, o Notación de Modelado de Procesos de Negocio, es un estándar diseñado para la representación gráfica de procesos de negocio. Permite mapear todas las etapas, decisiones, actividades y flujos involucrados desde el inicio hasta el final de un proceso, proporcionando una visión completa de su ejecución.

3 Toma de Requerimientos

Los requerimientos representan las necesidades o expectativas que el sistema debe satisfacer, definiendo tanto las funcionalidades que debe cumplir como las características que debe poseer. Estos requisitos son fundamentales para llevar a cabo el diseño y desarrollo adecuado del sistema.

Tabla 3.1: *Requerimiento 1, Entrenamiento de la IA*

Código	N.º 1
Nombre	Entrenamiento de la IA
Descripción	La IA debe ser entrenada utilizando un conjunto datos extenso y variado que cumplan con estándares de Big Data.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.2: *Requerimiento 2, Detección de Trampas*

Código	N.º 2
Nombre	Detección de Trampas
Descripción	El sistema debe detectar las trampas utilizadas y recolectar datos en tiempo real sobre estas para identificar cambios.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.3: *Requerimiento 3, Análisis de Datos*

Código	N.º 3
Nombre	Análisis de Datos
Descripción	El sistema debe contar con una plataforma que procese los datos recolectados y los analice mediante IA para generar acciones necesarias.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.4: *Requerimiento 4, Interfaz de Usuario*

Código	N.º 4
Nombre	Interfaz de Usuario
Descripción	El sistema debe poseer una interfaz accesible y fácil de usar que permita a los usuarios visualizar datos y la acciones a realizar.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.5: *Requerimiento 5, Compatibilidad Multiplataforma*

Código	N.º 5
Nombre	Compatibilidad Multiplataforma
Descripción	El sistema debe ser compatible en diferentes entornos Windows, macOS, Linux y servidores. Tanto como en plataformas y dispositivos de gama baja y media
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.6: *Requerimiento 6, Sistema de Notificaciones*

Código	N.º 6
Nombre	Sistema de Notificaciones
Descripción	El sistema debe enviar notificaciones en tiempo real sobre cambios críticos en el sistema, actualizaciones, intrusiones o fallos de detección.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.7: *Requerimiento 7, Monitoreo en Tiempo Real*

Código	N.º 7
Nombre	Monitoreo en Tiempo Real
Descripción	El sistema debe permitir el monitoreo continuo de las variables críticas de los juegos.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.8: *Requerimiento 8, Monitoreo de Procesos Externos*

Código	N.º 8
Nombre	Monitoreo de Procesos Externos
Descripción	El sistema debe detectar programas que interactúen ilegalmente con el cliente del juego.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.9: *Requerimiento 9, Soporte a Múltiples Idiomas*

Código	N.º 9
Nombre	Soporte a Múltiples Idiomas
Descripción	El sistema debe adaptar la interfaz y los reportes para el idioma especifico del usuario.
Nivel de Prioridad	Baja

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.10: *Requerimiento 10, Seguridad de Datos*

Código	N.º 10
Nombre	Seguridad de Datos
Descripción	El sistema debe garantizar la seguridad de los datos recolectados y los datos del usuario, protegiéndolos contra accesos no autorizados.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.11: *Requerimiento 11, Implementación de Métricas Avanzadas*

Código	N.º 11
Nombre	Implementación de Métricas Avanzadas
Descripción	El sistema debe generar gráficos y estadísticas que presenten datos sobre trampas detectadas y patrones de comportamiento.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.12: *Requerimiento 12, Conexión con Servidores*

Código	N.º 12
Nombre	Conexión con Servidores
Descripción	El sistema debe facilitar la conexión con los servidores de videojuegos para realizar el proceso de detección.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.13: *Requerimiento 13, Actualización de Algoritmos*

Código	N.º 13
Nombre	Actualización de Algoritmos
Descripción	El sistema debe permitir la actualización periódica y automatizada de los algoritmos de IA para mejorar su precisión y componentes del software y adaptarse a nuevas condiciones.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.14: *Requerimiento 14, Monitoreo de Trafico de Red*

Código	N.º 14
Nombre	Monitoreo de Trafico de Red
Descripción	El sistema debe analizar la comunicación entre el cliente y el servidor para identificar actividades sospechosas.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.15: *Requerimiento 15, Controles de Acceso*

Código	N.º 15
Nombre	Controles de Acceso
Descripción	El sistema debe gestionar los permisos según los roles: Jugador, Administradores y Moderadores.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.16: *Requerimiento 16, Mitigación de Errores*

Código	N.º 16
Nombre	Mitigación de Errores
Descripción	El sistema debe optimizar su modelo de detección, software y hardware para evitar errores críticos como reportes de: Falsos Positivos, Interrupción de Conexión, Problemas con Sistemas OS.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.17: *Requerimiento 17, Generar Reportes Detallados*

Código	N.º 17
Nombre	Generar Reportes Detallados
Descripción	El sistema debe proveer reportes automáticos que incluyan logs, patrones detectados y estadísticas relacionadas con trampas.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.18: *Requerimiento 18, Implementando Bloqueos Automáticos*

Código	N.º 18
Nombre	Implementando Bloqueos Automáticos
Descripción	El sistema debe implementar una función de bloqueo de ID automáticamente a usuarios detectados utilizando trampas, luego enviando un reporte a la desarrolladora del juego.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.19: *Requerimiento 19, Crear Listas Negras*

Código	N.º 19
Nombre	Crear Listas Negras
Descripción	El sistema debe registrar a los usuarios reincidentes que utilicen trampas, facilitando su monitoreo y sanción.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.20: *Requerimiento 20, Privacidad del Usuario*

Código	N.º 20
Nombre	Privacidad del Usuario
Descripción	El sistema debe cumplir con las regulaciones como GDPR, garantizando la anonimización de datos sensibles del usuario.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.21: *Requerimiento 21, Pruebas de Estrés*

Código	N.º 21
Nombre	Pruebas de Estrés
Descripción	Al sistema se le debe realizar pruebas bajo condiciones extremas para medir su capacidad y estabilidad.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.22: *Requerimiento 22, Sistema Anti-Ingeniería Inversa*

Código	N.º 22
Nombre	Sistema Anti-Ingeniería Inversa
Descripción	El sistema debe implementar medidas que dificulten la manipulación o descompilación del software anti-trampas.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.23: *Requerimiento 23, Diseño de Paneles de Control*

Código	N.º 23
Nombre	Diseño de Paneles de Control
Descripción	La interfaz debe permitir crear dashboards personalizados para la visualización de métricas.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.24: *Requerimiento 24, Ajuste de Modelos*

Código	N.º 24
Nombre	Ajuste de Modelos
Descripción	El sistema debe adaptar los modelos de IA según las características de cada videojuego.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.25: *Requerimiento 25, Integración de APIs Externas*

Código	N.º 25
Nombre	Integración de APIs Externas
Descripción	El sistema debe permitir la compatibilidad con APIs externas que complementen la detección de trampas.
Nivel de Prioridad	Baja

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.26: *Requerimiento 26, Pruebas de Usuario*

Código	N.º 26
Nombre	Pruebas de Usuario
Descripción	El sistema debe llevar a cabo pruebas con jugadores para evaluar la experiencia de uso y mejorar el sistema con base en su retroalimentación.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.27: *Requerimiento 27, Adaptarse a Nuevas Trampas*

Código	N.º 27
Nombre	Adaptarse a Nuevas Trampas
Descripción	El sistema debe implementar un mecanismo para actualizar automáticamente el sistema frente a nuevas amenazas detectadas.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.28: *Requerimiento 28, Impacto en Hardware*

Código	N.º 28
Nombre	Impacto en Hardware
Descripción	El sistema debe asegurar que no afecta significativamente el rendimiento de dispositivos de gama baja.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.29: *Requerimiento 29, Accesibilidad*

Código	N.º 29
Nombre	Accesibilidad
Descripción	La interfaz debe poseer un modo de accesibilidad para pueda ser usado por personas con discapacidad, cumpliendo las normativas de accesibilidad digital.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.30: *Requerimiento 30, Métricas en Tiempo Real*

Código	N.º 30
Nombre	Métricas en Tiempo Real
Descripción	El sistema debe proveer datos procesados en vivo para la detección de trampas y métricas de rendimiento.
Nivel de Prioridad	Alta

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.31: *Requerimiento 31, Latencia del Sistema*

Código	N.º 31
Nombre	Latencia del Sistema
Descripción	Se debe medir y optimizar el impacto del sistema en el rendimiento del juego en tiempo real.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.32: *Requerimiento 32, Compatibilidad Futura*

Código	N.º 32
Nombre	Compatibilidad Futura
Descripción	El sistema debe garantizar que pueda integrarse con nuevas tecnologías y plataformas emergentes.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Tabla 3.33: *Requerimiento 33, Recuperación Contra Fallos*

Código	N.º 33
Nombre	Recuperación Contra Fallos
Descripción	El sistema debe ser capaz de reiniciarse automáticamente y conservar datos en caso de fallos técnicos y crear respaldos de datos críticos y configuraciones.
Nivel de Prioridad	Media

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

A continuación, se presenta en la Tabla 3.34 que incluye el código de requerimiento, el nombre del requerimiento y el nivel de prioridad de cada uno de los 33 requerimientos identificados.

Tabla 3.34: Tabla Resumen de Requerimientos

Código de Requerimiento	**Nombre**	**Prioridad**
N.º 1	Entrenamiento de la IA	Alta
N.º 2	Detección de Trampas	Alta
N.º 3	Análisis de Datos	Alta
N.º 4	Interfaz de Usuario	Media
N.º 5	Compatibilidad Multiplataforma	Media
N.º 6	Sistema de Notificaciones	Alta
N.º 7	Monitoreo en Tiempo Real	Alta
N.º 8	Monitoreo de Procesos Externos	Alta
N.º 9	Soporte a Múltiples Idiomas	Baja
N.º 10	Seguridad de Datos	Alta
N.º 11	Implementación de Métricas Avanzadas	Media
N.º 12	Conexión con Servidores	Alta
N.º 13	Actualización de Algoritmos	Media
N.º 14	Monitoreo de Trafico de Red	Alta
N.º 15	Controles de Acceso	Alta
Continúa en la siguiente página		

Tabla 3.34 continuada de la página anterior

Código de Requerimiento	Nombre	Prioridad
N.º 16	Mitigación de Errores	Alta
N.º 17	Generar Reportes Detallados	Alta
N.º 18	Implementando Bloqueos Automáticos	Alta
N.º 19	Crear Listas Negras	Media
N.º 20	Privacidad del Usuario	Alta
N.º 21	Pruebas de Estrés	Media
N.º 22	Sistema Anti-Ingeniería Inversa	Alta
N.º 23	Diseño de Paneles de Control	Media
N.º 24	Ajuste de Modelos	Alta
N.º 25	Integración de APIs Externas	Baja
N.º 26	Pruebas de Usuario	Media
N.º 27	Adaptarse a Nuevas Trampas	Alta
N.º 28	Impacto en Hardware	Media
N.º 29	Accesibilidad	Media
N.º 30	Métricas en Tiempo Real	Alta

Continúa en la siguiente página

Tabla 3.34 continuada de la página anterior

Código de Requerimiento	Nombre	Prioridad
N.º 31	Latencia del Sistema	Media
N.º 32	Compatibilidad Futura	Media
N.º 33	Recuperación Contra Fallos	Media

4 Documentación del Product Backlog y el Sprint

En este apartado se presenta la documentación del Product Backlog, los Sprints realizados, y la información necesaria y pertinente para la implementación de Scrum.

4.1. Product Backlog

A continuación, en la Tabla 4.1, se detallan por mayor a menor relevancia las tareas realizadas en este proyecto, junto con el tiempo estimado de horas que se tardará en concretar. Se considera un total de 33 tareas y se espera terminar en un tiempo aproximado de 692 horas.

Tabla 4.1: *Product Backlog del Proyecto*

N° de Requerimi-ento	**Prioridad**	**Tarea**	**Estado**	**Estimado HH**
1	Alta	Entrenamiento de la IA con la base de datos	Pendiente	45
2	Alta	Implementar detección de trampas en tiempo real	Pendiente	40
3	Alta	Analizar datos recolectados	Pendiente	30
4	Media	Diseñar interfaz de usuario	Pendiente	30
5	Media	Garantizar compatibilidad multiplataforma	Pendiente	25
6	Alta	Configurar sistema de notificaciones	Pendiente	20
7	Alta	Implementar monitoreo en tiempo real	Pendiente	25
8	Alta	Detectar procesos externos ilegales	Pendiente	25
9	Baja	Implementar soporte para múltiples idiomas	Pendiente	15
10	Alta	Diseñar medidas de seguridad para datos	Pendiente	20

Continúa en la siguiente página

Tabla 4.1 continuada en la página anterior

N° de Requerimiento	Prioridad	Tarea	Estado	Estimado HH
11	Media	Crear métricas avanzadas	Pendiente	15
12	Alta	Conectar sistema con servidores	Pendiente	30
13	Media	Automatizar actualizaciones de algoritmos	Pendiente	15
14	Alta	Monitorear tráfico de red	Pendiente	25
15	Alta	Diseñar controles de acceso	Pendiente	15
16	Alta	Mitigar errores de detección	Pendiente	25
17	Alta	Generar reportes detallados	Pendiente	20
18	Alta	Implementar bloqueos automáticos	Pendiente	25
19	Media	Crear listas negras de usuarios infractores	Pendiente	15
20	Alta	Diseñar políticas para privacidad del usuario	Pendiente	20
21	Media	Realizar pruebas de estrés	Pendiente	15
22	Alta	Diseñar sistemas anti-ingeniería inversa	Pendiente	20
23	Media	Crear paneles de control personalizados	Pendiente	15
24	Alta	Ajustar modelos de IA según características de cada juego	Pendiente	20
25	Baja	Integrar APIs externas	Pendiente	10

Continúa en la siguiente página

Tabla 4.1 continuada en la página anterior

N° de Requerimiento	Prioridad	Tarea	Estado	Estimado HH
26	Media	Realizar pruebas de usuario	Pendiente	15
27	Alta	Adaptar el sistema a nuevas trampas	Pendiente	25
28	Media	Evaluar impacto en hardware	Pendiente	15
29	Media	Implementar accesibilidad en la interfaz	Pendiente	12
30	Alta	Proveer métricas en tiempo real	Pendiente	20
31	Media	Medir y optimizar latencia del sistema	Pendiente	15
32	Media	Diseñar compatibilidad futura	Pendiente	15
33	Media	Crear mecanismos de recuperación ante fallos	Pendiente	15

Fuente: Desarrollado por el estudiante encargado en este proyecto con fines académicos de la Universidad Andrés Bello (2024)

Total Estimado de Horas: 692

4.2. Sprint

En la Tabla 4.2 se presentan las tareas desarrolladas en el primer y único Sprint ordenadas según la fecha de inicio y término de cada una. Este Sprint se constituye de 33 tareas indicando la fecha en que se realizó cada tarea y el tiempo estimado en horas para su ejecución. En total, la realización de estas tareas tuvo una duración de aproximadamente 692 horas, lo que es equivalente a 28 días y 20 horas concretamente, trabajando entre 10 y 11 horas diarias son 59 días aproximadamente.

Tabla 4.2: *Planificación del Sprint*

N° Req.	Prioridad	Tarea	Inicio	Termino	Estado
1	Alta	Entrenar modelos de IA	29/01/2025	02/02/2025	Pendiente
3	Alta	Analizar datos recolectados	02/02/2025	05/02/2025	Pendiente
2	Alta	Implementar detección de trampas en tiempo real	05/02/2025	08/02/2025	Pendiente
7	Alta	Implementar monitoreo en tiempo real	08/02/2025	12/02/2025	Pendiente
14	Alta	Detectar procesos externos ilegales	12/02/2025	14/02/2025	Pendiente
16	Alta	Mitigar errores de detección	14/02/2025	16/02/2025	Pendiente
5	Media	Garantizar compatibilidad multiplataforma	16/02/2025	19/02/2025	Pendiente
10	Alta	Diseñar medidas de seguridad para datos	19/02/2025	21/02/2025	Pendiente
Continua en la siguiente pagina					

Tabla 4.2: *Planificación del Sprint*

N° Req.	Prioridad	Tarea	Inicio	Termino	Estado
12	Alta	Conectar sistema con servidores	21/02/2025	23/02/2025	Pendiente
6	Alta	Configurar sistema de notificaciones	23/02/2025	24/02/2025	Pendiente
4	Media	Diseñar interfaz de usuario	24/02/2025	28/02/2025	Pendiente
18	Alta	Implementar bloqueos automáticos	28/02/2025	01/03/2025	Pendiente
8	Alta	Detectar procesos externos ilegales	01/03/2025	03/03/2025	Pendiente
11	Media	Crear métricas avanzadas	03/03/2025	04/03/2025	Pendiente
15	Alta	Diseñar controles de acceso	04/03/2025	05/03/2025	Pendiente
9	Baja	Implementar soporte para múltiples idiomas	05/03/2025	06/03/2025	Pendiente
17	Alta	Generar reportes detallados	06/03/2025	08/03/2025	Pendiente
27	Alta	Adaptar el sistema a nuevas trampas	08/03/2025	11/03/2025	Pendiente
22	Alta	Diseñar sistemas anti-ingeniería inversa	11/03/2025	13/03/2025	Pendiente

Continua en la siguiente pagina

Tabla 4.2: *Planificación del Sprint*

N° Req.	Prioridad	Tarea	Inicio	Termino	Estado
20	Alta	Diseñar políticas para privacidad del usuario	13/03/2025	15/03/2025	Pendiente
19	Media	Crear listas negras de usuarios infractores	15/03/2025	16/03/2025	Pendiente
24	Alta	Ajustar modelos de IA según características de cada juego	16/03/2025	18/03/2025	Pendiente
28	Media	Evaluar impacto en hardware	18/03/2025	19/03/2025	Pendiente
26	Media	Realizar pruebas de usuario	19/03/2025	20/03/2025	Pendiente
25	Baja	Integrar APIs externas	20/03/2025	21/03/2025	Pendiente
30	Alta	Proveer métricas en tiempo real	21/03/2025	23/03/2025	Pendiente
31	Media	Medir y optimizar latencia del sistema	23/03/2025	24/03/2025	Pendiente
32	Media	Diseñar compatibilidad futura	24/03/2025	25/03/2025	Pendiente
13	Media	Automatizar actualizaciones de algoritmos	25/03/2025	26/03/2025	Pendiente
29	Media	Implementar accesibilidad en la interfaz	26/03/2025	27/03/2025	Pendiente
Continua en la siguiente pagina					

Tabla 4.2: *Planificación del Sprint*

N° Req.	Prioridad	Tarea	Inicio	Termino	Estado
33	Media	Crear mecanismos de recuperación ante fallos	27/03/2025	28/03/2025	Pendiente
21	Media	Realizar pruebas de estrés	28/03/2025	29/03/2025	Pendiente
-	Alta	Documentación y cierre del Sprint	30/03/2025	30/03/2025	Pendiente

Referencias

Mohr, S., & Rahman, S. S. (2011). IT security issues within the video game industry. *International Journal of Computer Science and Information Technology*, *3*(5), 1–16. https://doi.org/10.5121/ijcsit.2011.3501

International Organization for Standardization. (2022). *ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements*. https://www.iso.org/standard/27001

International Organization for Standardization. (2022). *ISO/IEC 27002:2022. Information security, cybersecurity and privacy protection — Information security controls*. https://www.iso.org/standard/75652.html

International Organization for Standardization. (2019). *ISO/IEC 27701:2019. Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines*. https://www.iso.org/standard/71670.html

International Organization for Standardization. (2023). *ISO/IEC 25010:2023. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model*. https://www.iso.org/standard/78176.html

International Organization for Standardization. (2022). *ISO/IEC 15408-4:2022. Information security, cybersecurity and privacy protection — Evaluation criteria for IT security. Part 4: Framework for the specification of evaluation methods and activities*. https://www.iso.org/standard/72913.html

International Organization for Standardization. (2021). *ISO 21500:2021. Project, programme and portfolio management — Context and concepts* https://www.iso.org/standard/75704.html

International Organization for Standardization. (2018). *ISO 31000:2018Risk management — Guidelines*. https://www.iso.org/standard/65694.html

Moya, R. (2012, marzo 31*). Modelo 4+1 de vistas de Kruchten para Dummies*. Jarroba. https://jarroba.com/modelo-41-vistas-de-kruchten-para-dummies/

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game*. https://www.scrum.org/resources/scrum-guide