UNIVERSIDAD
COMPLUTENSE
MADRID

Proyecto de Innovación

Convocatoria 2023/2024

Proyecto 43

Estudio de la aplicación de herramientas de análisis de contratos inteligentes de Ethereum en las asignaturas de blockchain de las titulaciones de la Facultad de Informática

Responsable del Proyecto: Pablo Gordillo

Facultad de Informática

Departamento de Sistemas Informáticos y Computación

1. **Objetivos propuestos en la presentación del proyecto**

Las cadenas de bloques (blockchain) y los contratos inteligentes (smart contracts) son dos de las tecnologías más prometedoras de los últimos años. Aunque inicialmente se han destinado a la implementación de criptomonedas y aplicaciones financieras que no dependen de un supervisor centralizado, sus potenciales aplicaciones van mucho más allá, por ejemplo, en aspectos relacionados con la gestión de la propiedad, especialmente de bienes intangibles, y la gobernanza descentralizada: sistemas de votación, derechos de autor, títulos universitarios, registro de propiedad de bienes físicos (inmuebles, vehículos), etc.

En su aplicación tradicional en el ámbito financiero, estas tecnologías han permitido el desarrollo de plataformas descentralizadas como Bitcoin o Ethereum junto a sus respectivas criptomonedas: el Bitcoin y el Ether. Aunque los primeros sistemas que han implementado estas tecnologías adolecían de un alto consumo de energía, debido fundamentalmente al protocolo de consenso utilizado (basado en la técnica denominada "proof-of-work" o prueba de trabajo), estos problemas ya están técnicamente resueltos.

Más allá de las criptomonedas, la posibilidad de crear contratos inteligentes (smart contracts) en las plataformas de blockchain más avanzadas ha permitido el desarrollo de aplicaciones de todo tipo basadas en esta tecnología. Los contratos inteligentes son programas informáticos donde se recogen los términos de un contrato y son capaces de reaccionar de forma automática ante las distintas condiciones y cláusulas especificadas en el mismo. La utilización de la tecnología blockchain proporciona una plataforma confiable para hacer cumplir los términos del contrato entre participantes que no se conocen o no confían entre sí sin necesidad de un regulador central.

Los contratos pueden ser creados por cualquier usuario, y por tanto son susceptibles de contener errores de programación. El mal funcionamiento de los contratos puede tener consecuencias directas, pues manejan conceptos críticos para las organizaciones y los usuarios como son la propiedad de capitales económicos o bienes, tanto físicos como intelectuales. En la breve historia de este tipo de sistemas se han producido graves problemas causados por errores de programación, entre los que destaca el ataque DAO valorado en millones de dólares. En este contexto resulta fundamental aplicar técnicas rigurosas de programación para evitar cualquier tipo de error o vulnerabilidad y para optimizar el coste de ejecución de los contratos inteligentes. Es particularmente útil la aplicación de técnicas de análisis y verificación de programas que permitan asegurar el correcto funcionamiento de los contratos inteligentes. Estas técnicas han sido objeto de estudio desde hace décadas en diversas áreas de investigación teórica y aplicada, y su aplicación en la tecnología blockchain se han convertido en un tema de investigación de gran interés y relevancia.

El objetivo fundamental de este proyecto de innovación educativa consiste en estudiar la implantación de herramientas de análisis, verificación y optimización de contratos inteligentes en el proceso de aprendizaje de los estudiantes en las asignaturas de blockchain impartidas en los grados de la Facultad de Informática, así como en el Máster interuniversitario de Métodos Formales en Ingeniería Informática, impartido por las universidades UCM, UPM y UAM. En particular, se han considerado las asignaturas “Introducción a la Tecnología Blockchain y Smart Contracts”, optativa de grado, y “Análisis de Sistemas Concurrentes y Distribuidos”, optativa de máster.

Como resultado de este proyecto, los alumnos podrán practicar los conceptos más avanzados del desarrollo de sistemas de blockchain y contratos inteligentes con

herramientas punteras que todavía están disponibles exclusivamente en entornos académicos. De esta forma, los alumnos podrán profundizar de forma práctica en conceptos como la estructura del código de bytes de un contrato inteligente, permitiendo visualizar y recorrer su grafo de control de flujo entendiendo sus problemáticas y peculiaridades; analizar el consumo de recursos de todas las funciones públicas de un contrato aunque este no sea constante, analizar posibles vulnerabilidades en el código que conlleven pérdidas económicas, o el estudio de buenas prácticas de programación que permitan la optimización de los contratos inteligentes, lo que se traducirá en un menor consumo de recursos y como consecuencia un menor coste económico para el usuario. Como consecuencia, los alumnos tendrán la oportunidad de aprender los detalles técnicos más avanzados que solicitan las empresas y organizaciones punteras en estas tecnologías.

Desde el punto de visto del alumnado, el uso de estas herramientas, además de favorecer buenos hábitos de programación en Solidity, les introduce en el uso de herramientas cuya utilización es cada vez más habitual en proyectos de desarrollo de contratos inteligentes y les ayudará a realizar trabajos de mayor calidad. Desde el punto de vista del profesorado, la utilización de este tipo de herramientas fundamentalmente les ayudará en el desarrollo de clases tanto teóricas como prácticas.

De cara a abordar el objetivo fundamental presentado al inicio de la sección identificamos los siguientes sub-objetivos:

- Análisis y evaluación de las herramientas propuestas. Se deberá estudiar el posible uso de las herramientas en cada una de las asignaturas mencionadas, para poder concluir de qué manera y en qué momento resultaría más conveniente su uso (en qué temas, mediante ejercicios, mediante sesiones de prácticas, etc).

- Extensión y mejora de las herramientas seleccionadas. Puesto que las herramientas solo se han utilizado hasta el momento en el campo de la investigación, su aplicación en el ámbito docente requerirá desarrollar ciertas mejoras y adaptaciones que permitan tratar dichos programas con suficiente precisión y de forma lo más automática posible.

- Instalación y uso. Resultará importante que tanto la instalación de la herramienta como su interfaz sean lo más sencillas y amigables posible, de forma que en ningún caso estos aspectos puedan desanimar a los estudiantes a usarla. Será necesario también por tanto extender la herramienta en su estado actual en estos dos aspectos, el procedimiento de instalación y la interfaz de usuario. En su estado actual, muchas de las herramientas solo se pueden ejecutar en sistemas operativos basados en Linux a través de línea de comandos. Por ello se implementará en un servidor web y/o una extensión ("plugin") de un entorno de desarrollo como Eclipse que permita un uso sencillo por parte de los alumnos, y también por parte de profesores no iniciados en el uso de estas herramientas.

- Introducción a los alumnos en los conceptos de calidad de software y buenas prácticas de programación en el ámbito de los contratos inteligentes y Ethereum y utilización de herramientas automáticas dentro del proceso general de desarrollo de contratos inteligentes.

- Evaluación del impacto de estas herramientas en la formación de los estudiantes mediante la evaluación de las propiedades categorizadas antes y después de la introducción de estas herramientas en la formación de los alumnos.

2. **Objetivos alcanzados**

Las tareas desarrolladas y la evaluación de estas han permitido verificar la consecución del objetivo principal del proyecto, que era la implantación de herramientas de análisis y verificación de contratos inteligentes de Ethereum en el proceso de aprendizaje de los alumnos en las asignaturas impartidas en la Facultad de Informática relacionadas con esta materia. Así mismo, el uso de estas herramientas ha permitido a los alumnos profundizar en la noción de *gas* como medida del consumo de recursos, y ejercitar buenas prácticas de programación de contratos inteligentes en Solidity, permitiéndoles razonar sobre la complejidad del consumo de gas y optimizar su código.

En lo referente a los sub-objetivos identificados en la sección anterior también han sido logrados:

Análisis y evaluación de las herramientas propuestas. Se estudiaron las herramientas disponibles y los planes de estudio de las asignaturas involucradas. Se decidió utilizar la herramienta GASTAP en la asignatura “Introducción a la tecnología blockchain y smart contracts” ofertada a los grados de la Facultad de Informática debido a la disparidad de perfiles de los estudiantes y al tratarse de una asignatura más básica. En esta asignatura se han preparado dos sesiones extra de laboratorio para trabajar con la herramienta. En la asignatura “Análisis de sistemas concurrentes y distribuidos” impartida en el Máster de Métodos Formales se han utilizado las herramientas EthIR, GASTAP, GASOL y SAFEVM. En este caso se han utilizado de apoyo en las clases teóricas para explicar los fundamentos formales que forman la base de estas herramientas, además de utilizarlas para la realización de ejercicios evaluables durante el curso.

- Extensión y mejora de las herramientas seleccionadas. Debido a la rápida evolución del compilador de Solidity, se han tenido que actualizar todas las herramientas para adaptarlas a las nuevas novedades introducidas y permitir analizar contratos inteligentes compilados con la última versión del compilador, al ser utilizado por todas ellas. Así mismo se han tenido que realizar mejoras en la escalabilidad de GASTAP y GASOL.

- Instalación y uso. En relación con la instalación, se han adaptado las herramientas para facilitar su funcionamiento en distintos sistemas operativos. Así, se han adaptado las herramientas para poder ejecutarlas tanto en sistemas operativos basados en MacOS como en Windows. Por otro lado, con el fin de facilitar el uso de las herramientas evitando un proceso complejo de instalación, se ha desarrollado un interfaz web que permite la ejecución tanto de GASTAP como de SAFEVM desde un navegador web. Además, se han desarrollado interfaces de usuario más sofisticadas para su uso por desarrolladores expertos: por una parte, un plugin de Eclipse utilizando como base un plugin de Solidity de código abierto para poder ejecutar la herramienta GASTAP; por último, se han incorporado también las herramientas en un plugin de Visual Studio Code. Para ello se ha desarrollado un servicio REST que permita realizar peticiones a un servidor donde estará instalado el backend de la herramienta. En este caso las herramientas han de estar instaladas localmente en el equipo y de momento solo está operativa en sistemas operativos basados en Linux.

- Calidad de software y buenas prácticas de programación. Como parte del temario de las asignaturas involucradas en este proyecto se ha llevado a cabo una introducción a los alumnos en los conceptos de calidad de software y buenas prácticas de programación en el ámbito de los contratos inteligentes y Ethereum y utilización de

herramientas automáticas dentro del proceso general de desarrollo de contratos inteligentes. Así mismo se ha incidido en la importancia de tener en cuenta el consumo de gas y las vulnerabilidades a la hora de desarrollar contratos inteligentes. El efecto de este subobjetivo se ha reflejado en las evaluaciones obtenidas por los estudiantes, mejorando la del curso anterior.

- Evaluación. El impacto de las herramientas en la formación de los estudiantes se ha evaluado mediante ejercicios prácticos evaluables realizados durante el curso. Además, en el caso de la asignatura “Introducción a la tecnología blockchain y smart contracts” los estudiantes han tenido que desarrollar una práctica final como parte de la evaluación de la asignatura mientras que en el caso de la asignatura “Análisis de sistemas concurrentes y distribuidos” han tenido que estudiar y presentar un artículo de investigación sobre las temáticas cubiertas por las herramientas incluidas en el proyecto y relacionarlas con las mismas. Por otro lado, se ha comparado la evaluación de este curso con la obtenida por los estudiantes en el curso académico 2022-2023, observando mejoras al introducir las herramientas de análisis y verificación en el proceso de aprendizaje.

El proyecto también ha permitido identificar puntos de mejora como es la interacción con las interfaces de usuario para desarrolladores expertos. El plugin de Eclipse no ha sido utilizado y no es una de las opciones preferidas. En cambio, se puede mejorar la interacción con el plugin de Visual Studio Code para que la información de los análisis aparezca también en el código fuente del contrato y no solo por la consola del entorno de programación.

3. Metodología empleada en el proyecto

A continuación se identifican las tareas realizadas durante la ejecución del proyecto orientadas a conseguir los objetivos mencionados. Cada tarea ha tenido un coordinador asignado, definido en la propuesta del proyecto y durante el desarrollo de cada tarea se han llevado a cabo tres reuniones de coordinación entre el responsable del proyecto y el de cada tarea: (i) una al inicio de cada tarea donde se desarrolló la hoja de ruta y se identificaron los objetivos de la tarea y acciones concretas para conseguirlos; (ii) una segunda reunión al cumplirse la mitad del tiempo asignado a cada tarea para valorar su estado, identificar posibles problemas y evaluar si se estaba cumpliendo el plan inicial; y (iii) una última reunión al finalizar cada tarea para comprobar que se habían alcanzado los objetivos identificados al inicio de la misma. El desarrollo de cada una de las siguientes tareas será descrito en la Sección 5 de la presente memoria.

Tarea 1. Preparación y evaluación de las herramientas.

Tarea 2. Recopilación de ejercicios.

Tarea 3. Planificación del uso de las herramientas en el programa de las asignaturas.

Tarea 4. Automatización e implementación.

Tarea 5. Aplicación de las herramientas en las asignaturas seleccionadas.

Tarea 6. Evaluación.

Tarea 7. Elaboración del informe final e interpretación de resultados.

El proyecto ha sido realizado con los estudiantes de las asignaturas mencionadas en el primer apartado de esta memoria. El perfil de los estudiantes de la asignatura “Introducción a la Tecnología Blockchain y Smart Contracts” es muy variado, ya que es una asignatura ofertada en todos los grados de la Facultad de Informática, incluyendo el doble grado en Matemáticas e Informática y el grado de Videojuegos. Este año se han matriculado 43 estudiantes de los cuales 12 procedían del programa Erasmus. Aunque las clases teóricas de la asignatura se imparten en español, todo el material de la asignatura está disponible en inglés y los profesores resuelven dudas y atienden a los alumnos en cualquiera de los dos idiomas. Todas las pruebas de evaluación se ofrecen a los alumnos en ambos idiomas. La asignatura “Análisis de Sistemas Concurrentes y Distribuidos” ha contado con 13 estudiantes matriculados, con un perfil más internacional y mucho más teórico, acorde con la naturaleza del máster. En este caso, la asignatura se imparte íntegramente en inglés.

4. **Recursos humanos**

El grupo de trabajo se ha compuesto de tal manera que todos los aspectos del proyecto queden cubiertos por los conocimientos de sus miembros:

Pablo Gordillo: Profesor Ayudante Doctor. Actualmente imparte las asignaturas "Tecnología de la programación II", “Introducción a la tecnología blockchain y smart contracts” y “Análisis de Sistemas Concurrentes y Distribuidos”. Su investigación está centrada en el análisis de contratos inteligentes y código de bytes de Ethereum, siendo coautor de varios artículos de investigación en esta área. Además es uno de los desarrolladores principales de las herramientas incluidas este proyecto.

Jesús Correas: Profesor Contratado Doctor. Imparte las asignaturas "Bases de datos" e "Introducción a la tecnología blockchain y smart contracts", y "Administración de bases de datos" en el Máster de Ingeniería Informática. Es coautor de varias publicaciones asociadas a las herramientas incluidas en este proyecto.

Elvira Albert: Catedrática. Lleva más de 15 años impartiendo clases relacionadas con programación, concurrencia y sistemas distribuidos. En la actualidad imparte y es la responsable de la asignatura "Programación Concurrente". Es coautora de las publicaciones que han desembocado en las herramientas incluidas en este proyecto.

Albert Rubio: Catedrático. Lleva más de 30 años impartiendo clases relacionadas con programación. Actualmente imparte las asignaturas de "Introducción a la tecnología blockchain y smart contracts" y "Análisis de sistemas concurrentes y distribuidos" en el Máster de Métodos Formales. Así mismo es el coordinador del Máster de Métodos Formales, donde se aplicará este proyecto. Su investigación está relacionada con Ethereum y sistemas de blockchain siendo uno de los principales autores de las herramientas incluidas en este proyecto.

Samir Genaim: Contratado Doctor. Ha sido profesor responsable de varias asignaturas relacionadas con programación. En la actualidad imparte la asignatura "Tecnología de la Programación II" de la que es responsable en los distintos grados de la facultad. Parte de su investigación en los últimos años está centrada en el desarrollo de un entorno web colaborativo de herramientas de análisis de programas. Así mismo actualmente desarrolla un verificador formal para contratos inteligentes.

Guillermo Román: Profesor Contratado Doctor en UPM. Ha impartido múltiples asignaturas de programación en UPM, como "Algoritmos y Estructuras de Datos", "Concurrencia" o "Programming Project". Es experto en el desarrollo de aplicaciones en Java y aplicaciones web, debido a su experiencia profesional como analista y jefe de proyecto en diversas empresas en el sector TIC. Es coautor de varias publicaciones que han desembocado en el desarrollo de las herramientas incluidas en este proyecto.

Miguel Isabel: Profesor Ayudante Doctor. Ha trabajado como profesor ayudante durante año y medio en la UPM, impartiendo la asignatura "Redes de Computadoras". Actualmente imparte la asignatura "Estructura de Datos" a varios grupos de la Facultad de Informática. Su investigación está centrada en el testing y verificación de programas concurrentes, así como en el desarrollo del compilador de circom, un lenguaje específico de dominio del blockchain y criptografía.

5. Desarrollo de las actividades

Para conseguir los objetivos del proyecto se identificaron las siguientes tareas, sobre las que se ha trabajado tal y como se ha descrito en la Sección 3 de la presente memoria. En cada tarea se han identificado distintas acciones que han permitido alcanzar los objetivos propuestos inicialmente.

Tarea 1. Preparación y evaluación de las herramientas. Durante esta fase se realizó una labor de testing y prueba de las herramientas incluidas en el Anexo de este informe. Como resultado de esta tarea, se identificaron las necesidades de actualización de algunas de las herramientas debido a la rápida evolución del compilador de Solidity. Por último, se asociaron las herramientas con las distintas temáticas cubiertas en las asignaturas de aplicación del proyecto antes mencionadas: “Introducción a la Tecnología Blockchain y Smart Contracts” (TBC) y “Análisis de Sistemas Concurrentes y Distribuidos” (ASCD).

Tarea 2. Recopilación de ejercicios. A través de repositorios públicos de código fuente de contratos como BigQuery (https://cloud.google.com/bigquery) y Etherscan (https://etherscan.io/) se descargaron un conjunto de más de 10.000 contratos inteligentes reales desplegados en la blockchain de Ethereum. Sobre estos se ejecutaron las herramientas propuestas para identificar los más interesantes para ser utilizados en las clases teóricas de las asignaturas. Además se identificaron algunas aplicaciones más complejas implementadas sobre sistemas de blockchain, como son los sistemas de votación de organizaciones descentralizadas (DAO), por ejemplo los ofrecidos por la plataforma Aragon (https://aragon.org/), que sirvieron como base para elaborar el proyecto final de la asignatura TBC.

Tarea 3. Planificación del uso de las herramientas en el programa de las asignaturas. Se han estudiado los programas de ambas asignaturas y se ha adaptado la planificación de los temas de forma que se incluyera el uso y presentación de las herramientas seleccionadas. Como resultado, se han identificado dos grupos de usuarios de estas herramientas: los que no están familiarizados con las técnicas de análisis y verificación de programas, fundamentalmente alumnos de grado (en particular de los grados menos relacionados con los métodos formales y desarrollo de software), así como algunos profesores de otras áreas de conocimiento; y por otra parte alumnos expertos, principalmente de nivel de máster, y profesores de áreas de conocimiento afines a los métodos formales y el desarrollo de software. Para satisfacer ambos tipos de necesidades, en este proyecto se ha optado por desarrollar varias líneas separadas de implementación de las herramientas, como se detalla en la siguiente tarea.

Tarea 4. Automatización e implementación. Como consecuencia de la tarea anterior, se ha decidido desarrollar tres líneas de implementación: por una parte se ha desarrollado un servidor web para que las herramientas estén disponibles sin necesidad de ninguna instalación en los ordenadores de los alumnos; por otra parte se ha desarrollado una extensión (plugin) del editor de código Eclipse, al que se ha incorporado una extensión de código Solidity así como toda la interfaz para poder ejecutar los análisis propuestos a través de peticiones a un servidor y; por último, se ha diseñado una extensión (plugin) del entorno de desarrollo Visual Studio Code que utilizan para desarrollar sus contratos inteligentes. En el primer y segundo caso se habilita el uso de herramientas sofisticadas a los alumnos que comienzan a utilizar estas técnicas y a los profesores no familiarizados con ellas. En el segundo caso, los alumnos más avanzados y los profesores pueden obtener toda la potencia de estas herramientas para aplicarlas a contratos más complejos. Así mismo todas las

herramientas son de código abierto y se encuentran disponibles para ser utilizadas a través de línea de comandos.

Tarea 5. Aplicación de las herramientas en las asignaturas seleccionadas. En la asignatura TBC se utilizaron las clases relacionadas con Solidity y la descripción del código de bytes EVM para introducir conceptos como el consumo de gas y análisis estático. Además, se han destinado dos sesiones en la penúltima semana de clases para que los alumnos pudieran aprovechar todos los conocimientos adquiridos durante el curso. En la primera sesión se explicaron en el aula los fundamentos básicos de la herramienta GASTAP para inferir el consumo de gas de un contrato, mientras que la segunda sesión fue fundamentalmente práctica en el laboratorio, donde los alumnos debían utilizar las herramientas propuestas con algunos ejercicios prototípicos extraídos de los recopilados en la Tarea 2. En particular, tuvieron que analizar contratos con diversa complejidad (constante, lineal, constante sobre estructuras de datos lineales) y optimizar el consumo de gas de un contrato sencillo utilizando la información disponible. Para ello utilizaron el interfaz web disponible para poder interactuar con la herramienta y extraer los resultados obtenidos.

En la asignatura ASCD, al ser una asignatura de un máster específico sobre métodos formales y tener un formato de clases diferente (sesiones de 3 horas combinando teoría y práctica) se pudieron aprovechar las herramientas durante las explicaciones teóricas. De esta manera se dedicó una sesión de 3 horas a explicar los fundamentos teóricos de EthIR y GASTAP y una segunda sesión para estudiar la teoría detrás de GASOL. Además se realizaron diversos ejercicios prácticos sobre estas herramientas así como ejercicios evaluables. Por último, todos los alumnos han tenido que estudiar y presentar un artículo de investigación relacionado con la temática cubierta por estas herramientas. Como parte de la presentación han tenido que comparar las herramientas estudiadas durante el curso con las propuestas en los siguientes artículos de investigación:

- MadMax: Surviving Out-of-Gas Conditions in Ethereum Smart Contracts
- Securify: Practical Security Analysis of Smart Contracts
- Asparagus: Automated Synthesis of Parametric Gas Upper-Bounds for Smart Contracts
- Online Detection of Effectively Callback Free Objects with Applications to Smart Contracts

Tareas 6 y 7. Evaluación e interpretación de resultados.

Para evaluar el grado de aceptación de los alumnos sobre el uso de estas herramientas, se han considerado diferentes aspectos:

- Relacionado con la asignatura de TBC:

1. Por una parte, en el ejercicio de la asignatura TBC se les ha preguntado su opinión sobre el uso de herramientas como las presentadas. La gran mayoría de los alumnos que entregaron el ejercicio evaluaron positivamente estas herramientas para diseñar contratos más eficientes. Todos ellos utilizaron la interfaz web y ninguno utilizó el plugin de Eclipse. Al realizar una encuesta sobre su IDE preferido se descubrió que Eclipse no se encuentra entre ellos y no están familiarizados con el entorno. Sin embargo, muchos han comenzado a utilizar en los últimos años Visual Studio Code, razón por la que se decidió desarrollar el plugin para esta plataforma.

2. En segundo lugar, se han estudiado los resultados de la evaluación detallada de los proyectos finales de la asignatura en los cursos 2023-2024 y se han comparado con los resultados del curso 2022-2023. En ambos cursos el grupo de alumnos era de tamaño similar (43 alumnos en el curso actual; 42 alumnos en el curso anterior). De las entregas de proyecto realizadas, el número de entregas con problemas de consumo de gas se ha reducido del curso anterior (11) al curso actual (9). De hecho, en el curso actual los criterios de evaluación han sido más estrictos en este aspecto que en el curso anterior.

- Relacionado con la asignatura de ASCD:

1. Los alumnos del Máster en Métodos Formales tienes una formación más teórica y están mucho más familiarizados con las temáticas que se imparten en la asignatura. Además, todos ellos están acostumbrados a trabajar con línea de comandos y distintos entornos de programación, aunque su preferido es Visual Studio Code. Gracias a su formación la implantación y utilización de las herramientas ha sido más rápido y sencillo que en el grupo anterior. Así mismo los alumnos han sugerido la posibilidad de integrar los análisis utilizados en Remix, el editor online más utilizado para programar contratos inteligentes en Ethereum. Como consecuencia se propondrá un trabajo de fin de grado/trabajo de fin de máster que integre los mismos en el editor.
2. Las temáticas cubiertas en la asignatura, apoyadas por el uso de las herramientas propuestas, han suscitado un mayor interés en los alumnos, incrementando su participación en clase. Además, este interés también se ha visto reflejado en la selección de los artículos de investigación que han tenido que presentar en clase. Mientras que el año pasado solo se ofertaron dos artículos relacionados con smart contracts, este año se han ofertado el doble y todos ellos han sido seleccionados para su presentación, repitiendo uno de los mismos.
3. Los estudiantes han realizado tres sesiones prácticas apoyadas por las herramientas de análisis incluidas en el proyecto. Los ejercicios propuestos han sido los mismos en el curso 2022-2023 y en el presente curso. La diferencia entre ambos cursos ha sido la utilización de EthIR, GASTAP y GASOL como apoyo durante este año. Se ha comprobado que, pese a necesitar más tiempo para familiarizarse con su uso y entender los resultados generados por las herramientas los resultados este año han sido más positivos, obteniendo una nota media de 8,3.

El curso 2021-2022 no se ha podido comparar con los resultados de este curso porque los criterios de evaluación fueron significativamente diferentes en ambas asignaturas.

6. **Anexos**

6.1 Herramientas utilizadas durante el proyecto

Los profesores integrantes de esta solicitud han desarrollado su actividad investigadora durante los últimos años en análisis y verificación de contratos inteligentes en Ethereum, desarrollando las siguientes herramientas utilizadas durante la ejecución del proyecto:

- EthIR: Decompilador de código de bytes de contratos inteligentes. Permite generar una representación intermedia a partir de la cual desarrollar nuevos análisis o aplicar otros ya existentes. Además, permite la generación de forma visual del grafo de control de flujo de los contratos analizados.

- GASTAP: Analizador de consumo de gas de los contratos. Implementa un análisis de recursos basado en el modelo de coste de Ethereum. De esta forma, permite inferir una cota superior del gas máximo que consumirá cada una de las funciones públicas del contrato. Este dato resulta fundamental, ya que el usuario, antes de ejecutar una función, tiene que proporcionar una cantidad máxima de gas que esta puede consumir. En caso de que el gas proporcionado sea inferior al consumido por la función, la ejecución se abortará y el usuario perderá todo el dinero enviado a la función.

- GASOL: Optimizador de contratos inteligentes que trabaja de forma ortogonal tanto a nivel de código fuente (Solidity) como a nivel de código de bytes, permitiendo reducir el consumo de gas (y como consecuencia el coste económico del contrato).

- Verificadores formales: Permiten asegurar formalmente que la ejecución del contrato será la deseada, demostrando que no se alcanzará ningún punto del programa que implique el aborto de la ejecución. Así mismo pueden demostrar la equivalencia semántica entre el código original y el código optimizado. Esto permite asegurar que el código optimizado generará el mismo resultado que el original, pero consumiendo menos gas. En este ámbito destaca la herramienta SAFEVM.