Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS3E1. Blockchain y Sistemas descentralizados (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS3E1. Blockchain y Sistemas descentralizados
2.2 Ciclo	:	CS300 Ciclo.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	▪ CS3P1. Computación Paralela y Distribuída. (8vo Sem) ▪ CS292. Ingeniería de Software II. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Blockchain y los sistemas descentralizados han pasado de ser un tema de nicho asociado a las criptomonedas a convertirse en un paradigma de cómputo de propósito general, usado en finanzas, cadenas de suministro, identidad digital e infraestructura pública. Este curso electivo brinda a los estudiantes una comprensión rigurosa, a nivel de sistemas, de cómo funciona realmente un blockchain: las estructuras de datos y primitivas criptográficas (funciones hash, firmas digitales, árboles de Merkle) que hacen que un registro distribuido sea a prueba de manipulaciones, las redes peer-to-peer y los protocolos de consenso (PoW, PoS, familia BFT) que permiten que nodos mutuamente desconfiados acuerden una única historia sin una autoridad central, y el entorno de ejecución (la EVM) que convierte a un blockchain en una plataforma de cómputo de propósito general para contratos inteligentes.

Sobre estos fundamentos, los estudiantes desarrollan, prueban y auditan contratos inteligentes usando Solidity y herramientas estándar de la industria, y diseñan la arquitectura de aplicaciones descentralizadas que combinan un backend en cadena con un frontend Web3 e identidad basada en billeteras (wallets). Dado que los contratos inteligentes gestionan valor económico real y no pueden parchearse fácilmente después de su despliegue, el curso pone especial énfasis en las clases de vulnerabilidad propias de este dominio – reentrancia, desbordamiento de enteros, fallas de control de acceso y manipulación de oráculos – y en la disciplina de auditoría necesaria para detectarlas antes del despliegue.

El curso cierra situando estas habilidades técnicas en un contexto más amplio: las técnicas de escalabilidad e interoperabilidad (Capa 2, puentes) y las primitivas económicas (tokens, creadores de mercado automatizados, protocolos de préstamo) que sustentan las finanzas descentralizadas, y las preguntas de gobernanza, regulación y ética – votación en cadena, fallas de gobernanza de las DAO, y el tratamiento legal en evolución de los criptoactivos – que todo ingeniero que trabaje en este campo debe estar preparado para analizar.

5. OBJETIVOS

- Explicar los fundamentos de estructura de datos, criptografía y redes peer-to-peer que hacen de un blockchain un registro descentralizado a prueba de manipulaciones.
- Analizar y comparar mecanismos de consenso, incluyendo Prueba de Trabajo (PoW), Prueba de Participación (PoS) y protocolos de la familia BFT, en términos de seguridad, desempeño y supuestos de confianza.
- Diseñar, programar y probar contratos inteligentes en una plataforma compatible con la EVM usando Solidity y herramientas modernas de desarrollo y pruebas.
- Diseñar la arquitectura de aplicaciones descentralizadas que integren un frontend Web3, una billetera (wallet) y un backend de contrato inteligente en cadena.

- Identificar vulnerabilidades de seguridad comunes en contratos inteligentes y aplicar técnicas y herramientas de auditoría para mitigarlas.

- Evaluar los enfoques de escalabilidad e interoperabilidad de blockchain, las primitivas económicas detrás de las finanzas descentralizadas (DeFi), y las implicancias de gobernanza y regulación de los sistemas descentralizados.

6. RESULTADOS DEL ESTUDIANTE

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Assessment)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usage)

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Assessment)

7. TEMAS

<table>
<tr><td colspan="2">Unidad 1: Registros Distribuidos y Arquitecturas Peer-to-Peer (8 horas)</td></tr>
<tr><td colspan="2">Resultados esperados: AG-C09,AG-C12</td></tr>
<tr><td>Temas</td><td>Aprendizaje esperado (Learning Outcomes)</td></tr>
<tr><td>▪ Estructura de un registro distribuido: una cadena de bloques enlazados mediante hashes criptográficos, que proporciona almacenamiento de solo anexado (append-only) y a prueba de manipulaciones de transacciones ordenadas Algoritmos Criptográficos.
▪ Topología de red peer-to-peer y protocolos de propagación por difusión/chismorreo (gossip) para propagar bloques y transacciones sin un servidor central Aplicaciones en Red.
▪ Roles de los nodos: nodos completos, nodos ligeros y mineros/validadores, y las responsabilidades que asume cada uno en el mantenimiento del registro.
▪ Compromisos entre descentralización y centralización: supuestos de confianza, puntos únicos de falla, resistencia a la censura y costos de desempeño Evaluación.</td><td>▪ Explicar cómo la estructura de bloques y el enlace mediante hashes de un registro distribuido proveen evidencia de manipulación (tamper-evidence) [Explicar]
▪ Comparar las arquitecturas peer-to-peer y cliente-servidor en términos de tolerancia a fallos, escalabilidad y confianza [Comparar]
▪ Analizar los compromisos de descentralización de un diseño blockchain dado frente a una alternativa centralizada [Analizar]</td></tr>
<tr><td colspan="2">Lecturas : [Nar+16], [Ant23]</td></tr>
</table>

Unidad 2: Fundamentos Criptográficos para Blockchain (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Funciones hash criptográficas: propiedades (resistencia a preimagen, resistencia a colisiones, efecto avalancha) y su uso para encadenar bloques Algoritmos Criptográficos. ▪ Firmas digitales para la autenticación de transacciones: pares de claves, firma y verificación, y no repudio Algoritmos Criptográficos. ▪ Árboles de Merkle: construcción, raíces de Merkle y pruebas de Merkle para la verificación eficiente de la inclusión de una transacción en un bloque. ▪ Direcciones derivadas de clave pública y consideraciones de gestión de claves Fundamentos de Criptografía.	▪ Explicar qué propiedades de las funciones hash criptográficas hacen que la cadena de bloques sea a prueba de manipulaciones [Explicar] ▪ Implementar un árbol de Merkle y verificar una prueba de inclusión para una transacción dada [Implementar] ▪ Aplicar firmas digitales para autenticar y verificar una transacción de blockchain [Aplicar]
Lecturas : [Nar+16], [Ant23]	

Unidad 3: Memoria y Consistencia (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de memoria, consistencia de datos y tolerancia a fallos: • Modelos de memoria: consistencia secuencial y de liberación/adquisición. • Gestión de memoria; incluyendo la reclamación de datos compartidos; conteo de referencias y alternativas. • Colocación y transferencia masiva de datos; reducción del tráfico de mensajes y mejora de la localidad; superposición de transferencia de datos y computación; impacto del diseño de datos como array de estructuras vs. estructura de arrays. • Emular memoria compartida: memoria compartida distribuida, Acceso Directo a Memoria Remota (RDMA). • Consistencia de almacenes de datos: Atomicidad, linealizabilidad, transaccionalidad, coherencia, orden causal, resolución de conflictos, consistencia eventual, cadenas de bloques. • Fallos, particionamiento y fallos parciales; votación; protocolos como Paxos y Raft. • Compromisos de diseño entre consistencia, disponibilidad, tolerancia a partición (fallos); imposibilidad de cumplir con todos a la vez. • Seguridad y confianza: Fallos bizantinos, prueba de trabajo y alternativas.	▪ Dar un ejemplo de un orden de accesos entre actividades concurrentes (por ejemplo, un programa con una carrera de datos) que no sea secuencialmente consistente [Crear] ▪ Escribir un programa que ilustre el reordenamiento de acceso a memoria o de mensajes [Escribir] ▪ Describir los méritos relativos del control de concurrencia optimista versus conservador bajo diferentes tasas de contención entre actualizaciones [Describir] ▪ Dar un ejemplo de un escenario en el que un intento de actualización optimista podría nunca completarse [Crear] ▪ Modificar un sistema concurrente para usar un almacén de datos más escalable, confiable o disponible [Crear] ▪ Usando una plataforma existente que admita almacenes de datos replicados, escribir un programa que mantenga un mapeo clave-valor incluso cuando uno o más hosts fallen [Escribir]
Lecturas : [Nar+16], [Bas23]	

Unidad 4: Plataformas de Ejecución Blockchain: El Modelo EVM (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitectura de la EVM: máquina virtual basada en pila, opcodes y el modelo de ejecución de bytecode Aspectos Comunes: Plataformas, APIs y Restricciones. ▪ Modelo de gas: contabilidad del costo de ejecución, límites de gas y su papel en acotar el cómputo y prevenir la denegación de servicio. ▪ Modelo de cuentas y estado: cuentas de propiedad externa frente a cuentas de contrato, y almacenamiento persistente de contratos. ▪ Ciclo de vida de un contrato inteligente: compilación a bytecode, despliegue, invocación y recibos de transacción.	▪ Explicar cómo el modelo de gas acota la ejecución de un contrato inteligente y mitiga el riesgo de denegación de servicio [Explicar] ▪ Describir cómo se ejecuta el bytecode de la EVM sobre la pila y el almacenamiento persistente del contrato [Describir] ▪ Comparar las cuentas de propiedad externa y las cuentas de contrato en términos de capacidades y estado [Comparar]
Lecturas : [AW18]	

Unidad 5: Desarrollo y Pruebas de Contratos Inteligentes (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ El lenguaje de programación Solidity: estructura de contratos, variables de estado, funciones, modificadores y eventos Aspectos Comunes: Lenguajes de Plataforma y Patrones Web. ▪ Frameworks y herramientas de desarrollo de contratos inteligentes (por ejemplo, Hardhat, Foundry, Truffle) para compilar, desplegar y automatizar contratos Entorno de Desarrollo e IDE. ▪ Redes de prueba locales y públicas para simular la ejecución de contratos antes del despliegue en la red principal (mainnet). ▪ Pruebas unitarias y de integración de contratos inteligentes, incluyendo cobertura de pruebas y simulación (mocking) de dependencias externas Prácticas Avanzadas de Pruebas.	▪ Programar un contrato inteligente en Solidity que implemente una regla de negocio especificada [Programar] ▪ Escribir pruebas unitarias y de integración para un contrato inteligente usando un framework de desarrollo [Escribir] ▪ Determinar si la suite de pruebas de un contrato inteligente cubre adecuadamente sus transiciones de estado [Determinar]
Lecturas : [AW18], [SKH19]	

Unidad 6: Arquitectura de Aplicaciones Descentralizadas (DApp) e Integración Web3 (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitectura de una DApp: backend de contrato inteligente en cadena frente a frontend y almacenamiento fuera de cadena Frontend, Frameworks y Estándares Web. ▪ Las billeteras (wallets) como agentes de identidad y firma: custodia de claves, firma de transacciones y protocolos de conexión. ▪ Bibliotecas cliente Web3 (por ejemplo, ethers.js, web3.js) para leer el estado de la blockchain y enviar transacciones firmadas Frontend, Frameworks y Estándares Web. ▪ Oráculos y fuentes de datos fuera de cadena para suministrar datos externos a los contratos en cadena.	▪ Diseñar la arquitectura en cadena/fuera de cadena de una aplicación descentralizada para un caso de uso dado [Diseñar] ▪ Integrar un frontend web con la billetera de un usuario mediante una biblioteca cliente Web3 para firmar y enviar transacciones [Integrar] ▪ Explicar el papel de los oráculos en el suministro de datos confiables fuera de cadena a los contratos inteligentes [Explicar]
Lecturas : [AW18], [SKH19]	

Unidad 7: Seguridad y Auditoría de Contratos Inteligentes (8 horas)	
Resultados esperados: AG-C11,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ataques de reentrancia y patrones de mitigación de verificaciones-efectos-interacciones (checks-effects-interactions) Seguridad de Memoria y Tipos. ▪ Desbordamiento y subdesbordamiento de enteros en la aritmética de contratos, y mitigaciones de aritmética segura (safe-math) Seguridad de Memoria y Tipos. ▪ Fallas de control de acceso en funciones de contratos, como modificadores faltantes o inicialización sin protección Control de Acceso y Aplicaciones. ▪ Ataques de manipulación de oráculos y de precios, y riesgos de front-running/ordenamiento de transacciones. ▪ Metodología de auditoría: revisión manual de código, análisis estático y herramientas de verificación formal Análisis de Amenazas e Ingeniería de Seguridad.	▪ Analizar un contrato inteligente para detectar vulnerabilidades de reentrancia y desbordamiento de enteros [Analizar] ▪ Aplicar patrones de verificaciones-efectos-interacciones y de aritmética segura para remediar una vulnerabilidad identificada [Aplicar] ▪ Evaluar los hallazgos de una herramienta de análisis estático aplicada a un contrato inteligente y priorizar su remediación [Evaluar]
Lecturas : [ABC17], [AW18]	

Unidad 8: Escalabilidad de Blockchain, Interoperabilidad y Primitivas DeFi (6 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Técnicas de escalamiento de Capa 2: rollups, canales de estado (state channels) y sidechains Evaluación. ▪ Interoperabilidad entre cadenas: puentes (bridges) y sus supuestos de confianza. ▪ Tokenomics y primitivas de finanzas descentralizadas (DeFi): tokens fungibles/no fungibles, creadores de mercado automatizados (automated market makers) y protocolos de préstamo Modelos y Estrategias Económicas.	▪ Comparar enfoques de escalamiento de Capa 2 en términos de rendimiento, seguridad y supuestos de confianza [Comparar] ▪ Analizar los supuestos de confianza y los modos de falla de un diseño dado de puente entre cadenas [Analizar] ▪ Describir cómo una primitiva DeFi dada, como un creador de mercado automatizado, usa incentivos de tokens para proveer un servicio financiero descentralizado [Describir]
Lecturas : [Bas23], [Wer+22]	

Unidad 9: Gobernanza de Blockchain, Regulación y Casos de Estudio éticos (4 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gobernanza en cadena (on-chain) y organizaciones autónomas descentralizadas (DAOs): mecanismos de votación y procesos de actualización de protocolo. ▪ Tratamiento regulatorio de los criptoactivos en distintas jurisdicciones y sus implicancias para el cumplimiento normativo Leyes, Políticas y Responsabilidad en Seguridad. ▪ Casos de estudio éticos: fallas de gobernanza notables e intervenciones comunitarias controvertidas Fundamentos y Marcos del Análisis Ético.	▪ Explicar cómo el mecanismo de votación en cadena de una DAO gobierna las actualizaciones de protocolo [Explicar] ▪ Debatir los compromisos entre la innovación sin permiso (permissionless) y el cumplimiento regulatorio en un caso de criptoactivos dado [Debatir] ▪ Evaluar las implicancias éticas y de gobernanza de una falla histórica de gobernanza blockchain [Evaluar]
Lecturas : [FW18], [Bas23]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

[Nar+16] Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. 1st. Princeton University Press, 2016.

[ABC17] Nicola Atzei, Massimo Bartoletti y Tiziana Cimoli. ■A Survey of Attacks on Ethereum Smart Contracts (SoK)■. En: *Principles of Security and Trust (POST)*. Springer, 2017. DOI: 10.1007/978-3-662-54455-6_8.

[AW18] Andreas M. Antonopoulos y Gavin Wood. *Mastering Ethereum: Building Smart Contracts and DApps*. 1st. O'Reilly Media, 2018.

[FW18] Primavera De Filippi y Aaron Wright. *Blockchain and the Law: The Rule of Code*. 1st. Harvard University Press, 2018.

[SKH19] Kevin Solorio, Randall Kanna y David H. Hoover. *Hands-On Smart Contract Development with Solidity and Ethereum*. 1st. O'Reilly Media, 2019.

[Wer+22] Sam M. Werner et al. ■SoK: Decentralized Finance (DeFi)■. En: *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT)*. ACM, 2022. DOI: 10.1145/3558535.3559780.

[Ant23] Andreas M. Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. 3rd. O'Reilly Media, 2023.

[Bas23] Imran Bashir. *Mastering Blockchain: A Deep Dive into Distributed Ledgers, Consensus Protocols, Smart Contracts, DeFi, and More*. 4th. Packt Publishing, 2023.