

Carrera: Ingeniería en Sistemas de Información

Asignatura: Desarrollo con tecnologías Blockchain

Planificación a partir del Ciclo Lectivo 2026

1. Datos administrativos de la asignatura

Nivel en la carrera	5	Duración	Cuatrimestral
Plan	2023		
Bloque curricular:	Asignatura Electiva		
Carga horaria presencial semanal (hs. cátedra):	6	Carga Horaria total (hs. reloj):	72
Carga horaria no presencial semanal (hs. reloj) (si correspondiese)		% horas no presenciales (hs. reloj) (si correspondiese)	

2. Presentación, Fundamentación

La asignatura se creó con la intención de que el estudiante complemente su formación con distintas habilidades técnicas y blandas necesarias sobre todo en las primeras experiencias laborales.

- Relación de la asignatura con el perfil de egreso: Brinda aprendizajes relativos al análisis, modelado, desarrollo e implementación de aplicaciones descentralizadas con tecnologías Blockchain.
- Relación de la asignatura con los alcances del título: Brinda las herramientas necesarias para modelar y construir desarrollos con tecnologías Blockchain desde la mirada de un profesional de sistemas. Aporta a las competencias específicas de la Carrera de especificación, proyecto y desarrollo de software, y proyecto y dirección en lo referido a seguridad informática.

3. Relación de la asignatura con las competencias de egreso de la carrera

En la tabla siguiente se establece la relación de la asignatura con las competencias de egreso: Específicas, Genéricas Tecnológicas y Genéricas Sociales, Políticas y Actitudinales de la carrera. Se incluyen las competencias de egreso a las que tributa, aportes reales y significativos de la asignatura, y en qué nivel (no aporta, bajo, medio, alto).

Competencias	Nivel
--------------	-------

Competencias genéricas tecnológicas (CG):	
CG.1. Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.	Alto
CG.2. Concepción, diseño y desarrollo de proyectos de Ingeniería en Sistemas de Información/Informática	Alto
CG.3. Gestión, planificación, ejecución y control de proyectos de Ingeniería en Sistemas de Información/Informática.	No aporta
CG.4. Utilización de técnicas y herramientas de aplicación de Ingeniería en Sistemas de Información/Informática.	Alto
CG.5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	No aporta
Competencias genéricas sociales, políticas y actitudinales (CG)	
CG.6. Fundamentos para el desempeño en equipos de trabajo.	Bajo
CG.7. Fundamentos para una comunicación efectiva.	Bajo
CG.8. Fundamentos para una actuación profesional ética y responsable.	No aporta
CG.9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.	No aporta
CG.10. Aprender en forma continua y autónoma.	Alto
CG.11. Fundamentos para el desarrollo de una actitud profesional emprendedora	No aporta
Competencias Específicas de la carrera	
CE1.1. Especificar, proyectar y desarrollar sistemas de información para concebir soluciones tecnológicas que permitan resolver situaciones en las organizaciones mediante el empleo de metodologías de sistemas y tecnologías asociadas a los sistemas de información.	No aporta
CE1.2. Especificar, proyectar y desarrollar sistemas de comunicación de datos, evaluando posibles soluciones tecnológicas disponibles para dar soporte a los sistemas de información en lo referido al procesamiento y comunicación de datos.	No aporta
CE1.3. Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable.	Alto
CE2.1. Proyectar y dirigir lo referido a seguridad informática para seleccionar y aplicar técnicas, herramientas, métodos y normas, garantizando la seguridad y privacidad de la información procesada y generada por los sistemas de información.	Alto

CE.3.1. Establecer métricas y normas de calidad de software para medir, evaluar, controlar y monitorear el rendimiento, impulsando mejoras de acuerdo a técnicas y normas vigentes definidas por los organismos de estandarización.	No aporta
CE.4.1. Certificar el funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software para asegurar la generación de los resultados deseados en función de restricciones de tiempo y recursos establecidos.	Bajo
CE.5.1. Dirigir y controlar la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de alcanzar los objetivos fijados por la organización.	Medio
CE.6.1. Asesorar y capacitar a organizaciones, empresas, organismos públicos o privados en la adquisición, instalación y uso, en lo que respecta a sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de un uso correcto de los sistemas intervinientes.	No aporta
CE.7.1. Realizar pericias, tasaciones y arbitrajes relacionados con su actividad profesional, respetando marcos normativos y jurídicos con el objeto de asesorar a las partes o a los tribunales de Justicia.	No aporta

4. Contenidos Mínimos

No corresponde por ser asignatura electiva

5. Objetivos establecidos en el DC

No corresponde por ser asignatura electiva

6. Resultados de aprendizaje

Los siguientes resultados de aprendizaje se promueven en el desarrollo de la asignatura

Identificador de RA	Redacción
RA1	Definir el modelado de Contratos Inteligentes para el desarrollo de una aplicación descentralizada con tecnologías Blockchain en función de los estándares de calidad de desarrollo de software.
RA2	Construir aplicaciones descentralizadas utilizando Hyperledger Fabric, Solidity, Docker y Node.JS con el fin de poner en práctica la programación back-end con entornos modernos en función de los estándares de calidad de desarrollo de software.

RA3	Incorporar prácticas de desarrollo seguro en la construcción de aplicaciones descentralizadas para adoptarlas como estrategia de disminución de riesgos según los estándares de seguridad de la información del proceso de desarrollo de software.
RA4	Verificar el funcionamiento de las unidades más pequeñas del desarrollo de aplicaciones descentralizadas y redes Blockchain para garantizar su correcto funcionamiento según los estándares de calidad de desarrollo de software.

7. Relación de los RA y las competencias

En la tabla siguiente se indica con X la tributación de cada Resultado de Aprendizaje con las competencias de egreso: específicas, genéricas tecnológicas, sociales, políticas y actitudinales de la carrera.

RA	CE1.1	CE1.2	CE1.3	CE2.1	CE3.1	CE4.1	CE5.1	CE6.1	CE7.1	CG1	CG2	CG3	CG4	CG5	CG6	CG7	CG8	CG9	CG10	CG11
RA1	-	-	X	X	-	-	X	-	-	X	X	-	X	-	X	X	-	-	X	-
RA2	-	-	X	X	-	-	-	-	-	X	X	-	X	-	-	-	-	-	X	-
RA3	-	-	X	X	-	-	X	-	-	X	X	-	X	-	-	-	-	-	X	-
RA4	-	-	X	X	-	X	-	-	-	X	X	-	X	-	-	-	-	-	X	-

8. Asignaturas correlativas previas

Para cursar y rendir debe tener cursadas:

- Asignatura/s:
Redes de Datos
Backend de Aplicaciones

Para cursar y rendir debe tener aprobada:

- Asignatura/s:
Paradigmas de Programación
Desarrollo de Software

9. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Asignatura/s:
No corresponde

10. Programa analítico

Este programa analítico contempla los contenidos mínimos, previstos en el DC vigente, y aquellos que se consideran necesarios para desarrollar los resultados de aprendizaje propuestos.

Unidad N°: 1

Título: BLOCKCHAIN, CRIPTOMONEDAS Y CONTRATOS INTELIGENTES

Contenidos: Características de Blockchain. Criptomonedas. Bitcoin. Altcoins y tokens. Blockchain y la computación cuántica. Trazabilidad de alimentos con Blockchain. Blockchain centrado en las personas. Función hash. Cómo funciona Bitcoin. Protocolo de consenso prueba de trabajo. Estructura de un bloque Bitcoin. Ethereum. Protocolo de consenso prueba de participación. Naturaleza jurídica de criptomonedas. Definiciones de nociones básicas de Blockchain. Guía práctica para la identificación trazabilidad e incautación de criptoactivos. Marco regulatorio de Blockchain, cripto y tokenización en Argentina y El Salvador. Estafas piramidales. ATH, halving y ETF de Bitcoin. ETF, ETP, ETC y ETN, características y diferencias. Comparativo entre Bitcoin y Ethereum. Hardware para minar criptomonedas. Granjas de minería en Argentina. DApps, aplicaciones distribuidas. Ejemplo de DApp de trazabilidad. DLT y Blockchain. IOTA. Actividad Criptomonedas y protocolos de consenso. Corba, Stablecoins, Memecoins, Ripple. El problema de los generales bizantinos. Criptomonedas y protocolos de consenso. Protocolos PBFT, DPoS, PoET, RAFT. Tipos de billeteras virtuales. Generación y

uso de una billetera virtual. Creación de una billetera virtual “caliente” (en línea). Exchanges. Seguimiento de precios de criptomonedas. Análisis comparativos. Bots de compraventa de criptomonedas.

Carga horaria por Unidad: 9 horas.

Unidad N°: 2

Título: CÓMO FUNCIONAN LOS CONTRATOS INTELIGENTES

Contenidos: Concepto y características de contratos inteligentes. Casos de uso. DAOs, Organizaciones Autónomas Descentralizadas. LNet, Hyperledger y redes Ethereum. Actividad Casos de uso de Blockchain.

Carga horaria por Unidad: 3 horas.

Unidad N°: 3

Título: MODELADO DE CONTRATOS INTELIGENTES

Contenidos: Etapas del diseño de una aplicación. Definir el proceso. Definir correctamente los problemas. Costo de corrección de errores. Diseño Dirigido por Modelos. Qué un proceso de negocios. Seleccionar y priorizar los proceso. Niveles de definición de procesos. Patrones. Patrones de negocio. Ejemplo de aplicación práctica. Innovación. Triángulo de Sábato. Pensamiento de Diseño. Metodología y ejercicio práctico. Analizar la situación actual. Entrevista. Ejercicio de cuestionario. Tipos de entrevistados. Modelar. DTE Diagrama de Transición de Estados. Símbolos, herramientas de creación en línea y ejercicio práctico. Eventos EPCIS de estándares GS1. Ejemplos. Actividad Patrones de negocio. Implementar. Gestión del cambio. Modelar un contrato inteligente. Definir el caso de uso. Definir cambios de estado. Seleccionar la red Blockchain y lenguaje de desarrollo. Modelar activos, participantes, contratos y transacciones. Ejemplos prácticos. El plan maestro: de diseños ancestrales a fortunas digitales. Modelado ágil utilizando patrones de negocio. Patrones de negocio. Selección y catalogado de Patrones. Herramienta PatCat 3.0. Integración de Inteligencia Artificial y modelado de contratos inteligentes. Taxonomía de modelo de negocios en Blockchain. Uso de patrones de eventos para el modelado. Modelado dirigido por casos de uso. Ejemplo práctico. Modelo integrado del contrato inteligente. Actividades Modelado de contrato inteligente y Taxonomía de modelo de negocios en Blockchain.

Carga horaria por Unidad: 9 horas.

Unidad N°: 4

Título: INTRODUCCIÓN A HYPERLEDGER FABRIC

Contenidos: Internet, protocolos de comunicación. Arquitecturas cliente/servidor. Redes Blockchain. Las matemáticas de Blockchain. Internet vs. Blockchain. Confianza. Qué es un contrato inteligente. Cómo logramos confianza. Aplicación y software tradicional vs. descentralizada. Introducción a tipos de redes Blockchain y protocolos de consenso. LF Decentralized Trust. Laboratorio Chaincode. Estructura de datos e implementaciones Blockchain. Cronología de ideas clave de Bitcoin. Listas enlazadas. Funciones Hash. Cadena de hashes. Inmutabilidad. Merkle Tree. DLT. World state. Redes peer2peer. Nodos de red y protocolos de consenso. Flujo transaccional del consenso. Laboratorio FabCar. Desarrollo con Hyperledger Fabric. Componentes. Fabric peer. Aplicación cliente. Gateways de aplicativos. Cómo conectarse a múltiples redes. Fabric-CA. Seteo de red. Ciclo de vida de un Chaincode. Políticas de endoso. Sintaxis y ejemplos. Servicio de membresía. Organizaciones. Red permissionada. MSP Proveedor de servicio de membresía. Identidades de usuarios y nodos. MSP organizacional. Registro e inscripción de usuarios. Firma de transacciones. Cómo funcionan las "wallets". Ejemplo de una aplicación cliente. Full Stack Asset Transfer. Configuración. Laboratorio Full Stack Asset Transfer - Desarrollo Chaincodes.

Carga horaria por Unidad: 21 horas.

Unidad N°: 5

Título: HERRAMIENTAS DE DESARROLLO PARA HYPERLEDGER FABRIC

Contenidos: Introducción a Docker y Kubernetes. Docker. VM Máquinas virtuales. VM versus contenedores Docker. Ventajas de contenedores. Componentes y línea de comandos. Dockerfile, imagen y contenedor. Imágenes y capas. Cómo funciona. Almacenamiento y versiones de Docker. Docker Compose. Kubernetes, orquestando contenedores. Glosario y cluster Kubernetes. Laboratorio Docker. Hyperledger Fabric: Local Setup, Tips para Windows, Conceptos de Smart Contract - Conceptos. Creación de un smart contract paso a paso. Hyperledger Fabric Gateway: App Cliente, Conceptos. Creación de una App Cliente paso a paso. Laboratorio Chaincode sistema de seguros.

Carga horaria por Unidad: 6 horas.

Unidad N°: 6

Título: DESARROLLO DE UNA DAPP CON HYPERLEDGER FABRIC

Contenidos: Despliegue en la nube. Introducción a las plataformas de nube. Ventajas, beneficios, principales proveedores y tipos de servicios. Despliegue de la API: opciones en AWS. IaC Cloudformation e Infraestructura como código. Laboratorio Docker + Compose + IaC. Laboratorio Registro de Certificados Académicos Parte 1 (Hyperledger Fabric + Node/TS). Laboratorio Registro de Certificados Académicos con Hyperledger Fabric Parte 2 (TypeScript +

API rest). Laboratorio Ejercicios con Solidity – Hardhat. Roadmap de Hyperledger Fabric. Ejemplos. Ejercicio de DApp para gestión de pagarés. Testing de smart contracts. Automatización: bots, triggers, integraciones. Despliegue de la aplicación en la nube. Carga horaria por Unidad: 21 horas.

Unidad N°: 7

Título: INTRODUCCIÓN AL DESARROLLO CON SOLIDITY

Contenidos: Qué es Ethereum y para qué se utiliza. EVM Máquina Virtual Ethereum. Desarrollo de contratos inteligentes en Ethereum. Introducción a Solidity. DApps y su arquitectura. Tokens y estándares ERC en Ethereum. Herramientas y recursos para desarrolladores. Desafíos y consideraciones en Ethereum. Hyperledger Fabric vs. Ethereum. Casos de uso Ethereum. Flujo y herramientas del entorno Ethereum. Ejercicio práctico “Hello world” con Remix. Laboratorio Solidity - Desarrollando con Remix.

Carga horaria por Unidad: 6 horas.

Unidad N°: 8

Título: DESARROLLO BLOCKCHAIN DIRIGIDO POR MODELOS

Contenidos: NFTs Token No Fungibles. Casos de Beeple, Proof of Humanity, Travelx y Flybondi. Qué significa fungible. Para qué sirve un token. Qué es un NFT y cómo funciona. NFTs en marketing y beneficios. Por qué la gente compra un NFT. Cómo calcular su valor. NFT más allá del arte. NFT e identidad digital descentralizada. UBI Renta Básica Universal. Autenticación de productos y proveedores. Títulos académicos de BFA. Proyecto Diploma de Red Clara. Learning tokens. Ejercicio práctico creación de un NFT en OpenSea. Estándares ERC de Ethereum. Estándar ERC-721. Decentraland. IPFS, almacenamiento descentralizado. Blockchain hoy. Progreso de la tecnología. Estado de la Comunidad Blockchain. Diferenciación de redes Blockchain. Los peligros de las criptomonedas. Redes públicas vs. privadas. Tokens vs. políticas de gobernanza de redes. Hacia una red de redes. Linux Foundation. Proyecto LF Decentralized Trust. Metas y beneficios. Hyperledger, principales tecnologías, lenguajes, herramientas y librerías de desarrollo. Hyperledger Fabric. GBBC Global Blockchain Business Council. Red RIBCi-CYTED. Ejercicio práctico Análisis FODA Fabric vs. Solidity. Actividad Modelado de contratos inteligentes. Tópicos Blockchain. Capas de un contrato inteligente. “Stakeholders” clave de Blockchain. Ejemplo práctico de Pagaré. Analogía protocolo de consenso RAFT. Modelar activos, participantes, contratos, transacciones y redes empresariales. Una jerarquía de contratos. Contrato inteligente vs. Chaincode. Proceso transaccional. Estudio de Casos sobre Blockchain aplicado a la Ciberdefensa y uso dual. Clase magistral internacional de Red RIBCi-CYTED sobre Aplicación de Blockchain a la cadena de

custodia de evidencia digital. No determinismo en Blockchain. Consideraciones de negocio. Requerimientos no funcionales. Web 3.0. Casos de éxito de Web 3.0 y Blockchain. Un marco ontológico de usabilidad para el modelado conceptual de sistemas. Patrones de escenarios. Patrones de usabilidad. Usabilidad web. Ejemplo de aplicación práctica. Definición de Requerimientos de Software a través de escenarios utilizando patrones. Cómo el blockchain y la ciberseguridad están transformando el mundo digital. Actividad Trabajo práctico integrador grupal de desarrollo con Fabric o Solidity. "Demo day" Modelado de contratos inteligentes. "Demo day" Trabajo práctico integrador grupal de desarrollo con Fabric o Solidity. Carga horaria por Unidad: 21 horas.

Carga horaria por tipo de formación práctica de toda la asignatura

Tipo de formación práctica	Horas reloj
Formación experimental	18
Análisis y resolución de problemas de ingeniería y estudios de casos	21
Formulación, análisis y desarrollo de proyectos.	9

Bibliografía Obligatoria:

1. Tapscott, D. y Tapscott, A. (2016). "La Revolución Blockchain". Ed. Deusto. <https://s3-sa-east-1.amazonaws.com/asdisal/wp-content/uploads/2019/06/07201007/2016-La-Revolucion%CC%81n-Blockchain-Don-Tapscott.pdf>
2. Querro, S.E. (2020). "Smart Contracts. Qué son, para qué sirven y para qué no servirán". IJ Editores. https://www.academia.edu/45442700/SMART_CONTRACTS_QU%C3%89_SON_PARA_QU%C3%89_SIRVEN_Y_PARA_QU%C3%89_NO_SERVIR%C3%81N
3. Nakamoto, S. (2008). "Bitcoin: A Peer-To-Peer Electronic Cash System". <https://bitcoin.org/bitcoin.pdf>
4. Notas de la cátedra.
5. Januszewski, M. (2022). "Criptosía 20.22 ". Ed. Deloscuatrovientos.
6. Negrino, T.; Smith, D. (2000). "Guía de aprendizaje: Javascript". Ed. Prentice Hall.
7. Documentación oficial de Node.JS. <https://nodejs.org/en/>
8. Documentación oficial de Docker. <https://www.docker.com/>
9. Documentación oficial de Solidity. <https://www.soliditylang.org/>
10. Documentación oficial de desarrollo Ethereum. <https://ethereum.org/es/developers/docs/>

11. Documentación oficial de Hyperledger Fabric. <https://hyperledger-fabric.readthedocs.io/en/release-2.5/>

Bibliografía optativa y otros materiales a utilizar en la asignatura:

1. Medina, O.C., Marciszack, M.M., Groppo, M.A. (2021). "Una aproximación a la incorporación de Patrones de Negocio en el modelado de aplicaciones de Contratos Inteligentes", publicado en actas de Jornadas de Ciencia y Tecnología 2021 de U.T.N. – F.R.S.F., art. 871, Editorial edUTecNe, U.T.N.
https://www.researchgate.net/publication/356645561_Una_aproximacion_a_la_incorporacion_de_Patrones_de_Negocio_en_el_modelado_de_aplicaciones_de_Contratos_Inteligentes
2. Angini, I., Ruiz Santillán, R. (2020). "Tecnología Blockchain aplicada al comercio de la energía eléctrica". Proyecto final Ing. en Sistemas de Información U.T.N. - F.R.C.
3. Marciszack, M.M., Moreno, J.C., Sánchez, C.E., Medina, O.C., Delgado, A.F., Castro, C.S. (2018). "Patrones en la construcción del Modelo Conceptual para sistemas de información", Editorial edUTecNe, U.T.N.
4. Medina, O.C., Meloni, B.E., Strub, A.M., Marciszack, M.M. (2025). "Incorporación de Machine Learning en la selección de patrones de eventos para el modelado de contratos inteligentes". Revista Ciencia y Tecnología N° 25. Editorial Universidad de Palermo.
5. Bradley, M., Gorman, D., Lucas, M., Golby-Kirk, M. (2019). "Getting Started with Enterprise Blockchain. A Guide to Design and Development". Editorial O'Reilly.
6. Solis-Osorio, C.A., Pérez-Cortés, E., Cervantes-Maceda, H. (2019). "Hacia una metodología para el diseño de contratos inteligentes", ReCIBE, Año 8 No. 1, México, Ed. UAM.
<https://www.redalyc.org/journal/5122/512259512009/html/>
7. Endeavor (2018). "Endeavor: Insight blockchain. Blockchain, ¿la promesa de una revolución?", Ed. Endeavor México. N° 3. <https://www.endeavorcampus.com/endeavor-insight-blockchain-la-promesa-de-una-revolucion/>
8. Weking, J., Mandalenakis, M., Hein, A., Hermes, S., Böhm, M., Krcmar, H. (2019). "The impact of blockchain technology on business models – a taxonomy and archetypal patterns", Electronic Markets (2020) 30, págs. 285–305, Ed. Springer.
https://www.researchgate.net/publication/338118210_The_impact_of_blockchain_technology_on_business_models_-_a_taxonomy_and_archetypal_patterns
9. Besancon, L., Ferreira Da Silva, C., Ghodous, P., Gelas, J. (2017). "A Blockchain Ontology for DApps Development", Ed. IEEE. https://repositorio.iscte-iul.pt/bitstream/10071/25567/1/article_88890.pdf
10. Bartoletti, M. and Pompianu, L. (2017). "An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns", en actas de International Conference on Financial Cryptography and Data Security 2017, págs. 494-509, Ed. Springer.

https://www.researchgate.net/publication/315454656_An_Empirical_Analysis_of_Smart_Contracts_Platforms_Applications_and_Design_Patterns.

11. Microsoft Corporation (2008). "Microsoft Official Course 6461A Visual Studio 2008: Windows Communication Foundation Developer". Ed. Microsoft.

12. Januszewski, M. (2022). "Las aventuras de Bitman: el primer criptohéroe de la historia universal". Ed. Autores de Argentina.

11. Metodología de enseñanza

- Todas las clases se realizarán en los laboratorios de informática de la Facultad usando software ya disponible, Visual Studio Code, Docker, NPM, Node.JS y SDK de lenguajes de desarrollo Hyperledger Fabric y Solidity (software libre y de código abierto). Se destaca el uso de herramientas de la nube para intercambio de información con los alumnos relativa a trabajos prácticos (Google Drive - git) a fin de familiarizarse con su uso.
- Los docentes llevarán su propio control de asistencia a fin de lograr un seguimiento más personalizado del alumnado.
- En https://github.com/utn-frc-desabc-2026/material_clase se publicarán algunas presentaciones, cuestionarios, notas de cátedra.
- Para la enseñanza de los lenguajes de desarrollo se ejemplifica y analiza con modelos presentados por los profesores; para luego asignar a los estudiantes resolución de problemas prácticos y ejercicios de programación. Para evaluar los conocimientos teóricos y prácticos en forma individual se utilizarán cuestionarios Moodle y exámenes.
- Los trabajos prácticos y las tareas semanales integran los conocimientos de las distintas unidades, y parte del tiempo de clase se usa para analizar dudas y elementos relevantes referidos a esos trabajos. Se evaluarán en presentaciones orales los resultados de los trabajos prácticos grupales.

12. Recomendaciones para el estudio

- Todas las clases se realizarán en los laboratorios de informática de la Facultad usando software ya disponible, Visual Studio Code, Docker, NPM, Node.JS y SDK de lenguajes de desarrollo Hyperledger Fabric y Solidity (software libre y de código abierto). Se destaca el uso de herramientas de la nube para intercambio de información con los alumnos relativa a trabajos prácticos (Google Drive - git) a fin de familiarizarse con su uso.
- Se sugiere a los estudiantes que instalen en sus equipos personales el software de desarrollo para que puedan realizar más horas de práctica luego de finalizar las clases en los laboratorios, y previo al inicio de las clases, preparar dudas y consultas que les surjan en la mencionada práctica.
- Los docentes llevarán su propio control de asistencia a fin de lograr un seguimiento más personalizado del alumnado.

- o En https://github.com/utn-frc-desabc-2026/material_clase se publicarán algunas presentaciones, cuestionarios, notas de cátedra. Durante cada clase se trabaja un tema central, y el mismo se ejemplifica y analiza con modelos presentados por los profesores.
- o Los trabajos prácticos y las tareas semanales integran los conocimientos de las distintas unidades, y parte del tiempo de clase se usa para analizar dudas y elementos relevantes referidos a esos trabajos.

13. Metodología de evaluación

El modelo de enseñanza basado en competencias implica la aplicación de metodologías e instrumentos de evaluación que permiten conocer, a docentes y estudiantes, el nivel de desarrollo de las competencias que aborda la asignatura.

La evaluación ocurre:

A través de 4 evaluaciones parciales y 1 trabajo integrador grupal se miden aspectos como la interpretación de consignas, resolución de problemas, usando lógica de programación, manejo de componentes de aplicaciones descentralizadas y actualización de datos en una red Blockchain.

A través de la observación de las / los estudiantes en clase.

A través de la exposición de cada grupo de su trabajo grupal clase a clase.

A través del seguimiento del trabajo grupal, de las entregas pactadas y de los entregables.

Serán considerados los siguientes aspectos al evaluar:

- Puntualidad en evaluaciones y tiempos de entregas.
- Dominio técnico del tema evaluado.
- Capacidad de integrar conceptos y aplicaciones.
- Dominio de terminología técnica.
- Expresión clara y técnicamente correcta.

A continuación, se detallan todos los Resultados de Aprendizajes con sus contenidos a desarrollar para alcanzarlos, la mediación pedagógica, metodologías y estrategias de evaluación, tiempo en horas reloj.

Resultados de Aprendizaje	Contenidos según programa	Mediación Pedagógica	Metodología y Estrategias de Evaluación	Tiempos en hora reloj
RA 1	Características de Blockchain. Criptomonedas. Bitcoin. Altcoins y tokens. Blockchain y la computación cuántica. Trazabilidad de alimentos con Blockchain. Blockchain centrado en las personas. Función hash. Cómo funciona Bitcoin. Protocolo de consenso prueba de trabajo. Estructura de un bloque Bitcoin. Ethereum. Protocolo de consenso prueba de participación. Naturaleza jurídica de criptomonedas. Definiciones de nociones básicas de Blockchain. Guía práctica para la identificación	Estrategias: Talleres paso a paso. Resolución de problemas. Exposición dialogada. Aula invertida. Aprendizaje Colaborativo en grupos. Actividades: Investigar de manera individual y grupal los fundamentos del modelado de contratos inteligentes y los estándares de calidad del desarrollo con tecnologías Blockchain. Distinguir los detalles de los casos de uso que abordan los	Instrumentos: Diagrama de transición de estados de un contrato inteligente. Modelo conceptual de un contrato inteligente. Criterios: Escucha activa y respeto a los demás. Puntualidad en evaluaciones y tiempos de entregas. Expresión clara y técnicamente correcta.	• Horas presenciales: ○ Horas de clase: 18 ○ Horas de laboratorio: 9 • Horas extra áulicas: 15

	trazabilidad e incautación de criptoactivos. Marco regulatorio de Blockchain, cripto y tokenización en Argentina y El Salvador. Estafas piramidales. ATH, halving y ETF de Bitcoin. ETF, ETP, ETC y ETN, características y diferencias. Comparativo entre Bitcoin y Ethereum. Hardware para minar criptomonedas. Granjas de minería en Argentina. DApps, aplicaciones distribuidas. Ejemplo de DApp de trazabilidad. DLT y Blockchain. IOTA. Actividad Criptomonedas y protocolos de consenso. Corba, Stablecoins, Memecoins, Ripple. El problema de los generales bizantinos. Criptomonedas y	fundamentos de la tecnología Blockchain identificando y analizando en debates de manera individual y/o en grupo. Construir de manera colaborativa un diagrama de transición de estados y el modelo conceptual de un contrato inteligente.	Vinculación de los conceptos fundamentales de Blockchain y los contratos inteligentes. Identificación de los estándares de calidad de software que aplican al modelado de contratos inteligentes. Dominio técnico del diagrama de transición de estados. Elaboración del modelo conceptual de un contrato inteligente.	
--	---	---	--	--

	protocolos de consenso. Protocolos PBFT, DPoS, PoET, RAFT. Tipos de billeteras virtuales. Generación y uso de una billetera virtual. Creación de una billetera virtual “caliente” (en línea). Exchanges. Seguimiento de precios de criptomonedas. Análisis comparativos. Bots de compraventa de criptomonedas. Concepto y características de contratos inteligentes. Casos de uso. DAOs, Organizaciones Autónomas Descentralizadas. LNet, Hyperledger y redes Ethereum. Actividad Casos de uso de Blockchain. Etapas del diseño de una aplicación. Definir el proceso. Definir correctamente los problemas.			
--	--	--	--	--

	Costo de corrección de errores. Diseño Dirigido por Modelos. Qué un procesode negocios. Seleccionar y priorizar los proceso.Niveles de definición de procesos. Patrones. Patrones de negocio. Ejemplo deaplicación práctica. Innovación. Triángulo de Sábato. Pensamiento de Diseño. Metodología y ejercicio práctico. Analizar la situación actual. Entrevista. Ejercicio de cuestionario. Tipos de entrevistados. Modelar. DTE Diagrama de Transición de Estados. Símbolos, herramientas de creación en línea y ejercicio práctico. Eventos EPCIS de estándares GS1. Ejemplos. Actividad Patrones de negocio.			
--	---	--	--	--

	Implementar. Gestión del cambio. Modelar un contrato inteligente. Definir el caso de uso. Definir cambios de estado. Seleccionar la red Blockchain y lenguaje de desarrollo. Modelar activos, participantes, contratos y transacciones. Ejemplos prácticos. El plan maestro: de diseños ancestrales a fortunas digitales. Modelado ágil utilizando patrones de negocio. Patrones de negocio. Selección y catalogado de Patrones. Herramienta PatCat 3.0. Integración de Inteligencia Artificial y modelado de contratos inteligentes. Taxonomía de modelo de negocios en Blockchain. Uso de patrones de eventos para el			
--	--	--	--	--

	modelado. Modelado dirigido por casos de uso. Ejemplo práctico. Modelo integrado del contrato inteligente. Actividades Modelado de contrato inteligente y Taxonomía de modelo de negocios en Blockchain. Actividad Modelado de contratos inteligentes. Consideraciones de negocio. Requerimientos no funcionales. Web 3.0. Casos de éxito de Web 3.0 y Blockchain. Un marco ontológico de usabilidad para el modelado conceptual de sistemas. Patrones de escenarios. Patrones de usabilidad. Usabilidad web. Ejemplo de aplicación práctica. Definición de Requerimientos			
--	--	--	--	--

	de Software a través de escenarios utilizando patrones.			
RA2 Elija un elemento.	Internet, protocolos de comunicación. Arquitecturas cliente/servidor. Redes Blockchain. Las matemáticas de Blockchain. Internet vs. Blockchain. Confianza. Qué es un contrato inteligente. Cómo logramos confianza. Aplicación y software tradicional vs. descentralizada. Introducción a tipos de redes Blockchain y protocolos de consenso. LF Decentralized Trust. Laboratorio Chaincode. Estructura de datos e implementaciones Blockchain. Cronología de ideas clave de Bitcoin. Listas enlazadas. Funciones Hash. Cadena de hashes. Inmutabilidad. Merkle Tree. DLT. World sate. Redes peer2peer. Nodos de red y protocolos de consenso. Flujo transaccional del consenso. Laboratorio FabCar. Desarrollo con Hyperledger Fabric. Componentes. Fabric peer. Aplicación cliente. Gateways de aplicativos. Cómo conectarse a múltiples redes. Fabric-CA. Seteo de red.	Estrategias: Talleres paso a paso. Resolución de problemas. Exposición dialogada. Aula invertida. Aprendizaje Colaborativo en grupos. Actividades: Investigar de manera individual y grupal los fundamentos de la programación back-end y los estándares de calidad del desarrollo con tecnologías Blockchain. Configurar ambientes de desarrollo Hyperledger Fabric y Solidity para programación back-end.	Instrumentos: Ambientes de desarrollo Hyperledger Fabric y Solidity. Ejercicios de programación utilizando Hyperledger Fabric, Solidity, Docker y Node.JS. Ejercicio de DApp para gestión de pagarés. Criterios: Escucha activa y respeto a los demás. Puntualidad en evaluaciones y tiempos de entregas.	• Horas presenciales: ○ Horas de clase: 9 ○ Horas de laboratorio: 24 • Horas extra áulicas: 20

	Políticas de endoso. Sintaxis y ejemplos. Servicio de membresía. Organizaciones. Red permitida. MSP Proveedor de servicio de membresía. Identidades de usuarios y nodos. MSP organizacional. Registro e inscripción de usuarios. Firma de transacciones. Cómo funcionan las “wallets”. Ejemplo de una aplicación cliente. Full Stack Asset Transfer. Configuración. Laboratorio Full Stack Asset Transfer - Desarrollo Chaincodes. Introducción a Docker y Kubernetes. Docker. VM Máquinas virtuales. VM versus contenedores Docker. Ventajas de contenedores. Componentes y línea de comandos. Dockerfile, imagen y contenedor. Imágenes y capas. Cómo funciona. Almacenamiento y versiones de Docker. Docker Compose. Kubernetes, orquestando contenedores. Glosario y cluster Kubernetes. Laboratorio Docker. Hyperledger Fabric: Local Setup, Tips para Windows, Conceptos de Smart Contract -	Construir de manera colaborativa ejercicios de programación utilizando Hyperledger Fabric, Solidity, Docker y Node.JS.	Dominio técnico de configuración de ambientes de desarrollo Hyperledger Fabric y Solidity para programación back-end. Construcción de ejercicios de programación back-end.	
--	---	---	--	--

	Conceptos. Creación de un smart contract paso a paso. Hyperledger Fabric Gateway: App Cliente, Conceptos. Creación de una App Cliente paso a paso. Laboratorio Chaincode sistema de seguros. Qué es Ethereum y para qué se utiliza. EVM Máquina Virtual Ethereum. Desarrollo de contratos inteligentes en Ethereum. Introducción a Solidity. DApps y su arquitectura. Tokens y estándares ERC en Ethereum. Herramientas y recursos para desarrolladores. Desafíos y consideraciones en Ethereum. Hyperledger Fabric vs. Ethereum. Casos de uso Ethereum. Flujo y herramientas del entorno Ethereum. Ejercicio práctico "Hello world" con Remix. Laboratorio Solidity - Desarrollando con Remix. NFTs Token No Fungibles. Casos de Beeple, Proof of Humanity, Travelx y Flybondi. Qué significa fungible. Para qué sirve un token. Qué es un NFT y cómo funciona. NFTs en marketing y beneficios. Por qué la gente compra un NFT. Cómo calcular su valor. NFT			
--	---	--	--	--

	más allá del arte. NFT e identidad digital descentralizada. UBI Renta Básica Universal. Autenticación de productos y proveedores. Títulos académicos de BFA. Proyecto Diploma de Red Clara. Learning tokens. Ejercicio práctico creación de un NFT en OpenSea. Estándares ERC de Ethereum. Estándar ERC-721. Decentraland. IPFS, almacenamiento descentralizado. Tópicos Blockchain. Capas de un contrato inteligente. "Stakeholders" clave de Blockchain. Ejemplo práctico de Pagaré. Analogía protocolo de consenso RAFT. Modelar activos, participantes, contratos, transacciones y redes empresariales. Una jerarquía de contratos. Contrato inteligente vs. Chaincode. Proceso transaccional. Estudio de Casos sobre Blockchain aplicado a la Ciberdefensa y uso dual. Clase magistral internacional de Red RIBCi-CYTED sobre Aplicación de Blockchain a la cadena de			
--	---	--	--	--

	custodia de evidencia digital. No determinismo en Blockchain.			
RA3 Elija un elemento.	Despliegue en la nube. Introducción a las plataformas de nube. Ventajas, beneficios, principales proveedores y tipos de servicios. Despliegue de la API: opciones en AWS. IaC Cloudformation e Infraestructura como código. Laboratorio Docker + Compose + IaC. Laboratorio Registro de Certificados Académicos Parte 1 (Hyperledger Fabric + Node/TS). Laboratorio Registro de Certificados Académicos con Hyperledger Fabric Parte 2 (TypeScript + API rest). Laboratorio Ejercicios con Solidity – Hardhat. Roadmap de Hyperledger Fabric. Ejemplos. Ejercicio de DApp para gestión de pagarés. Blockchain hoy. Progreso de la tecnología. Estado de la Comunidad Blockchain. Diferenciación de redes Blockchain. Los peligros de las criptomonedas. Redes públicas vs. privadas. Tokens vs. políticas de gobernanza de redes. Hacia una red de redes.	Estrategias: Lección Magistral Participativa Talleres paso a paso. Resolución de ejercicios. Actividades: Configurar ambientes de desarrollo que incorporen recomendaciones de desarrollo seguro en Blockchain. Programar ejercicios de contratos inteligentes incorporando buenas prácticas de desarrollo seguro.	Instrumentos: Ambiente de desarrollo Hyperledger Fabric y Solidity. Ejercicio de contrato inteligente e integración con una aplicación front-end. Ejercicio de DApp para registro de automotores. Criterios: Escucha activa y respeto a los demás. Puntualidad en evaluaciones y tiempos de entregas.	• Horas presenciales: ○ Horas de teoría: 3 ○ Horas de laboratorio: 3 • Horas extra áulicas: 3

	Linux Foundation. Proyecto LF Decentralized Trust. Metas y beneficios. Hyperledger, principales tecnologías, lenguajes, herramientas y librerías de desarrollo. Hyperledger Fabric. GBBC Global Blockchain Business Council. Red RIBCi-CYTED. Ejercicio práctico Análisis FODA Fabric vs. Solidity. Cómo el blockchain y la ciberseguridad están transformando el mundo digital. Actividad Trabajo práctico integrador grupal de desarrollo con Fabric o Solidity. “Demo day” Modelado de contratos inteligentes. “Demo day” Trabajo práctico integrador grupal de desarrollo con Fabric o Solidity.		Dominio técnico de configuración de ambientes de desarrollo Blockchain seguros. Construcción de ejercicios de contratos inteligentes seguros.	
RA4 Elija un elemento.	Testing de smart contracts. Automatización: bots, triggers, integraciones. Despliegue de la aplicación en la nube. Requerimientos no funcionales. Ciclo de vida de una chaincode.	Estrategias: Lección Magistral Participativa Talleres paso a paso. Resolución de ejercicios. Actividades: Configurar ambiente de prueba de contratos inteligentes. Realizar pruebas y actividades de aseguramiento de calidad de	Instrumentos: Ambiente de prueba de contratos inteligentes. Ejercicios de prueba de contratos inteligentes. Criterios:	• Horas presenciales: ○ Horas de clase: 3 ○ Horas de laboratorio: 3 • Horas extra áulicas: 3

		programación de contratos inteligentes.	Escucha activa y respeto a los demás. Puntualidad en evaluaciones y tiempos de entregas. Dominio técnico de configuración de ambiente de prueba de contratos inteligentes. Ejecución y análisis de ejercicios de prueba de contratos inteligentes.	
--	--	--	--	--

14. Condiciones de aprobación

Condiciones de regularización:

Parcial o recuperatorio aprobado (se conserva la mejor nota) con nota igual o superior a 4.

Realizar 80% de las actividades propuestas en Moodle.

Trabajo final grupal aprobado con todos requerimientos necesarios y nota igual o superior a 4.

Condiciones de aprobación directa:

Parcial o recuperatorio aprobado (se conserva la mejor nota) con nota igual o superior a 7.

Trabajo final grupal aprobado con todos requerimientos necesarios y nota igual o superior a 7.

Realizar 100% de los cuestionarios múltiple choice propuestos en Moodle con promedio de nota mayor igual a 7.

Promedio final igual o superior a 7.

15. Modalidad de examen

La modalidad de examen es individual y como se detalla a continuación.

- Para los estudiantes que alcanzaron la regularidad: Deben inscribirse al examen final de la asignatura, y rendir completando el desarrollo en PC de laboratorio de un programa que cumpla las consignas de un enunciado particular de un desarrollo Blockchain y completar un cuestionario múltiple choice en Moodle. Ambos trabajos constan de una síntesis de todos los temas abordados en la asignatura.
- Para los estudiantes que alcanzaron la aprobación directa: debe inscribirse al examen final de la asignatura.

16. Recursos necesarios

Las clases se llevan a cabo en laboratorio, donde se requiere:

IDE Visual Studio Code con extensiones de IBM Blockchain Platform (Hyperledger Fabric)

Docker y NPM

Node.JS

Solidity SDK

Gitlab

Guía de trabajo: Publicada en gitlab

Proyector

Se precisa el mismo software para el desarrollo del trabajo grupal.

