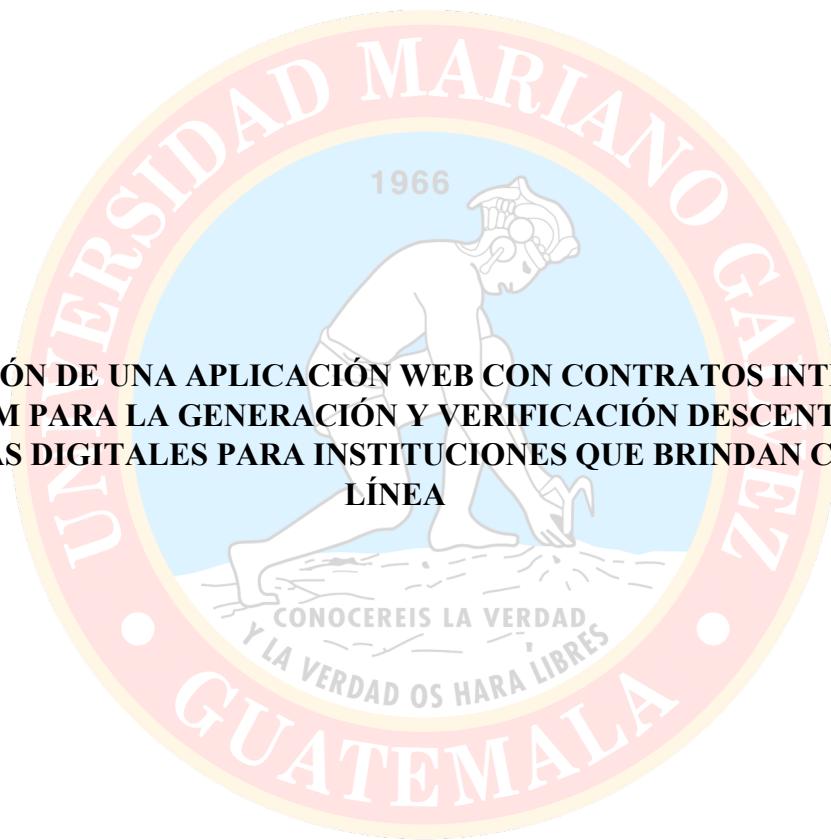


UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

INTEGRACIÓN DE UNA APLICACIÓN WEB CON CONTRATOS INTELIGENTES
EN ETHEREUM PARA LA GENERACIÓN Y VERIFICACIÓN DESCENTRALIZADA
DE DIPLOMAS DIGITALES PARA INSTITUCIONES QUE BRINDAN CURSOS EN
LÍNEA

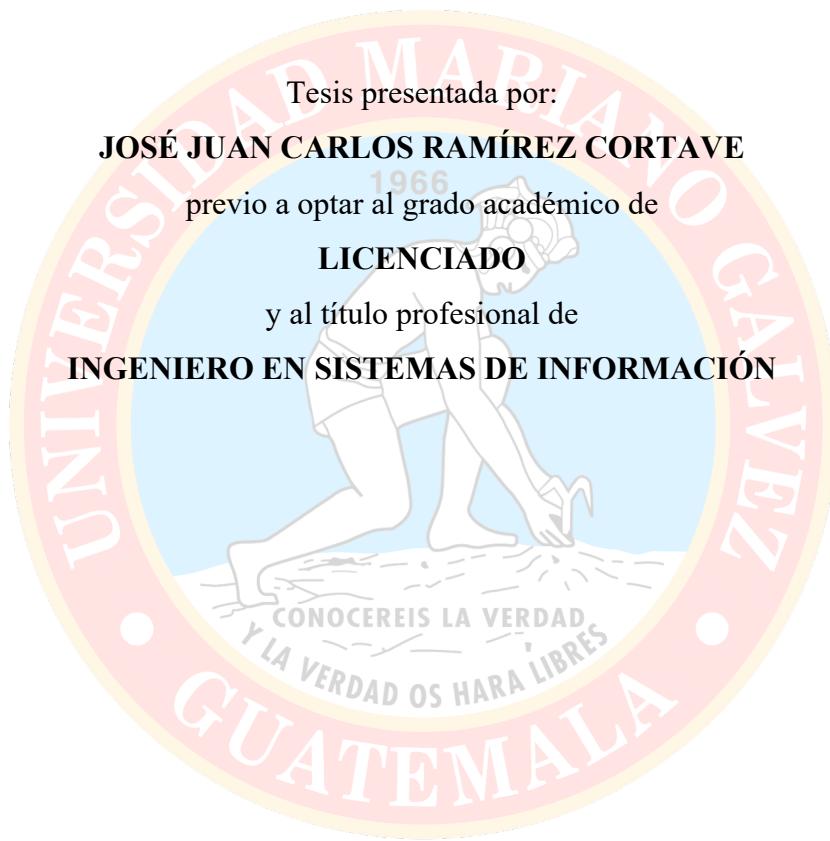


José Juan Carlos Ramírez Cortave

Guatemala, noviembre de 2024

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**INTEGRACIÓN DE UNA APLICACIÓN WEB CON CONTRATOS INTELIGENTES
EN ETHEREUM PARA LA GENERACIÓN Y VERIFICACIÓN DESCENTRALIZADA
DE DIPLOMAS DIGITALES PARA INSTITUCIONES QUE BRINDAN CURSOS EN
LÍNEA**



Guatemala, noviembre de 2024

**AUTORIDADES DE LA FACULTAD
Y TRIBUNAL QUE PRACTICÓ EL EXAMEN DEL TRABAJO DE GRADUACIÓN**

DECANO DE LA FACULTAD: Ing. Jorge Alberto Arias Tobar
SECRETARIO DE LA FACULTAD: Ing. Hugo Adalberto Hernández Santizo

TRIBUNAL EXAMINADOR

PRESIDENTE: Ing. Tribunal examinador
SECRETARIO: Ing. Secretario
VOCAL: Ing. Vocal

AUTORIZACIÓN PARA LA IMPRESIÓN DEL TRABAJO DE GRADUACIÓN

REGLAMENTO DE TRABAJO DE GRADUACIÓN

Artículo 8º. RESPONSABILIDAD

Solamente el autor es responsable de los conceptos expresados en el trabajo de tesis. Su aprobación en manera alguna implica responsabilidad para la Universidad.

X

Índice

Introducción	1
1. Capítulo I – Anteproyecto de investigación.....	2
1.1. Antecedentes	2
1.2. Justificación.....	5
1.3. Planteamiento del Problema.....	6
1.4. Objetivos	8
1.4.1. Objetivo General	8
1.4.2. Objetivos Específicos.....	8
1.5. Viabilidad.....	9
1.5.1. Viabilidad de mercado	9
1.5.2. Viabilidad Técnica /Tecnológica	10
1.5.3. Viabilidad Soporte.....	10
1.5.4. Viabilidad administrativa	11
1.5.5. Viabilidad Económica	12
1.6. Alcance.....	13
1.6.1. Geográfico.....	13
1.6.2. Tecnológico o Técnica	14
1.6.3. Persona/Empresa	14
1.6.4. Temporal	14
1.6.5. Temático.....	15
1.7. Pregunta de Investigación	15
1.7.1. Pregunta general	15
1.7.2. Preguntas específicas.....	16
1.8. Hipótesis.....	16
1.9. Variables	16
1.9.1. Variables Independientes	16
1.9.2. Variables Dependientes.....	17
1.10. Indicadores	17
1.11. Supuestos.....	18
1.12. Métodos de investigación o metodología.....	19

1.12.1. Generalidades	19
1.12.2. Población y muestra	20
1.12.3. Instrumentos de la investigación	20
1.12.4. Marco de trabajo SCRUM.....	21
1.13. Otras consideraciones.....	22
1.13.1. Diagramas UML.....	23
1.14. Planificación de capítulos	23
1.15. Estimación de recursos.....	24
1.15.1. Recursos humanos.....	24
1.15.2. Recursos materiales.....	25
1.15.3. Recursos financieros	25
1.15.4. Recursos intangibles.....	25
2. Capítulo II – Marco teórico.....	26
2.1. Blockchain.....	26
2.1.1. Historia	26
2.1.2. Funcionamiento	26
2.1.3. Tipos de Blockchain.....	28
2.2. Ethereum	29
2.2.1. Máquina virtual de Ethereum.....	30
2.2.2. Gas.....	31
2.3. Contratos inteligentes.....	31
2.3.1. Solidity	32
2.3.2. Herramientas	32
2.4. Aplicaciones descentralizadas.....	32
2.4.1. Aplicaciones Web	34
2.4.2. Web3.js.....	34
2.5. Certificaciones.....	34
2.5.1. Certificados digitales.....	35
2.5.2. Certificados educativos digitales.....	35
2.6. Leyes y Normativa sobre la Emisión de Títulos y Diplomas.....	36
2.6.1. Instituciones Encargadas	36

2.6.2.	Emisión de Títulos y Diplomas	36
2.6.3.	Castigos por Falsificación de Títulos y Diplomas	36
2.6.4.	Ley Integral de Protección de Datos Personales	37
3.	Capítulo 3 – Análisis y diseño	38
3.1.	Diseño	38
3.1.1.	Frontend	38
3.1.2.	Backend.....	39
3.1.3.	Smart contract	40
3.2.	Arquitectura.....	40
3.3.	Requerimientos	41
3.3.1.	Requerimientos funcionales	41
3.3.2.	Requerimientos no funcionales	42
3.4.	Diagramas	43
3.4.1.	Caso de Uso.....	43
3.4.2.	Despliegue.....	43
3.4.3.	Diagrama de componentes	44
3.4.4.	Entidad Relación	46
3.4.5.	Diagrama de Secuencia	46
3.4.6.	Diagramas de actividad	48
3.5.	Prototipos	50
3.6.	SCRUM.....	53
3.6.1.	Backlog.....	53
3.6.2.	Sprint Backlog.....	54
3.6.3.	Incrementos	55
4.	Capítulo 4 – Desarrollo de software	57
4.1.	Patrones de diseño	57
4.1.1.	Modelo-Vista-Controlador	57
4.1.2.	Modelo-Vista-VistaModelo (MVVM).....	57
4.2.	Herramientas de desarrollo	58
4.2.1.	WebStorm.....	58
4.2.2.	Docker	59

4.2.3.	Postman	60
4.2.4.	Metamask	61
4.2.5.	Alchemy	62
4.3.	Módulos desarrollados	63
4.3.1.	Backend.....	63
4.3.2.	Frontend	77
4.4.	Pruebas de ejecución	84
4.4.1.	Prueba de despliegue de contrato inteligente	84
4.5.	Ambientes de la aplicación	86
4.5.1.	Ambiente de desarrollo	87
4.5.2.	Ambiente de integración	88
4.5.3.	Ambiente de producción	90
5.	Capítulo 5 – Pruebas de validación.....	91
5.1.	Pruebas técnicas	91
6.	Capítulo 6 – Pruebas de certificación	96
6.1.	Hallazgos.....	96
7.	Capítulo 7 – Implementación.....	101
7.1.1.	Frontend	102
7.1.2.	Backend.....	105
8.	Capítulo 8 – Mantenimiento	111
8.1.	Actualización de librerías.....	111
8.2.	Deposito de Ethereum	111
8.3.	Mejoras solicitadas.....	111
8.3.1.	Monitoreo del rendimiento de la aplicación.....	112
8.3.2.	Creación de servicio especializado para generación y validación de certificados	
	112	
8.3.3.	Otras blockchain.....	113
9.	Conclusión	114
10.	Recomendaciones	116
11.	Anexos	117
	Anexo A - Cuadro de ideas	117

Anexo B – Mapa mental	119
Anexo C – Formatos de las entrevistas	120
Anexo D – Formato de cuestionario	122
Anexo E – Formato de Encuesta.....	123
Anexo F - Especificación de Requerimientos de Software (SRS).....	124
12. Glosario.....	125
13. Referencia bibliográfica.....	128

Índice de figuras

Figura 1 <i>Ubicación de la empresa Cloud Solutions Network</i>	13
Figura 2 <i>Cronograma de actividades</i>	24
Figura 3 <i>Simplificación del funcionamiento de Blockchain</i>	28
Figura 4 <i>Arquitectura de aplicaciones distribuidas</i>	33
Figura 5 <i>Arquitectura descentralizada híbrida</i>	41
Figura 6 <i>Caso de uso para la generación y validación de certificados</i>	43
Figura 7 <i>Diagrama de Despliegue</i>	44
Figura 8 <i>Diagrama de componentes</i>	45
Figura 9 <i>Diagrama entidad relación</i>	46
Figura 10 <i>Diagrama de secuencia para la autenticación JWT</i>	47
Figura 11 <i>Diagrama de secuencia para la creación de certificado</i>	47
Figura 12 <i>Diagrama de secuencia para la verificación</i>	48
Figura 13 <i>Diagrama de actividad para la creación de usuario</i>	49
Figura 14 <i>Diagrama de actividad en la generación de certificado</i>	49
Figura 15 <i>Diagrama de actividad para la validación</i>	50
Figura 16 <i>Pantalla de inicio sesión</i>	50
Figura 17 <i>Pantalla de listado de usuarios</i>	51
Figura 18 <i>Pantalla de curso y asignación de estudiantes</i>	51
Figura 19 <i>Pantalla de generación de certificado</i>	52
Figura 20 <i>Pantalla de carga de archivo</i>	52
Figura 21 <i>Backlog en Jira</i>	54
Figura 22 <i>Sprint en Jira</i>	55
Figura 23 <i>Código fuente desde Github Desktop</i>	56
Figura 24 <i>Pantalla de WebStorm</i>	58
Figura 25 <i>Archivo docker compose del proyecto</i>	59
Figura 26 <i>Ejecución del docker compose</i>	60
Figura 27 <i>Ventana del Docker Desktop</i>	60
Figura 28 <i>Ejemplo de petición en Postman</i>	61

Figura 29 Cuenta de ejemplo en Metamask	62
Figura 30 Pantalla principal del sitio web de Alchemy	63
Figura 31 Código fuente en Solidity del contrato inteligente	64
Figura 32 Función en Solidity que despliega el contrato inteligente en Ethereum	65
Figura 33 Entidad de usuarios	66
Figura 34 Estrategia de autenticación	67
Figura 35 Prueba de autenticación en Postman	67
Figura 36 Controlador de cursos	68
Figura 37 Consulta al recurso de cursos	69
Figura 38 Servicio de estudiante	70
Figura 39 Creación de estudiante desde Postman	71
Figura 40 Módulo en NestJS	72
Figura 41 Consulta de asignación en Postman	73
Figura 42 Servicio de certificados.....	74
Figura 43 Creación de certificado desde Postman	75
Figura 44 Data Transfer Object para registros de validación	76
Figura 45 Consulta el listado de registros de validación	76
Figura 46 Pantalla de Login	77
Figura 47 Componente del inicio de sesión	78
Figura 48 Listado de usuarios	79
Figura 49 Página para validar un certificado	79
Figura 50 Certificado inválido	80
Figura 51 Ejemplo de certificado	81
Figura 52 Página principal de la transacción	82
Figura 53 Sección para más detalles	82
Figura 54 Información adicional del certificado dentro de la Blockchain	83
Figura 55 Mensaje de certificado exitoso	83
Figura 56 Despliegue de smart contract desde la terminal	84
Figura 57 Smart contract desde Etherscan	85
Figura 58 Transacción en Etherscan	85
Figura 59 Detalle del despliegue en Alchemy	86

Figura 60 <i>Error en variables de entorno .env</i>	96
Figura 61 <i>Carta de certificación</i>	100
Figura 62 <i>Archivos para el despliegue del frontend</i>	102
Figura 63 <i>Bucket en S3 para el Frontend</i>	103
Figura 64 <i>Bucket policy en S3</i>	103
Figura 65 <i>Configuración de CORS</i>	104
Figura 66 <i>Frontend desde el navegador</i>	105
Figura 67 <i>Creación de base de datos en AWS</i>	106
Figura 68 <i>Instancia de la base de datos en RDS</i>	106
Figura 69 <i>Instancia EC2</i>	107
Figura 70 <i>Ejecución de instancia</i>	108
Figura 71 <i>Cambio de usuario</i>	109
Figura 72 <i>Instalacion de NVM y NodeJS</i>	109
Figura 73 <i>Configuración de clave SSH</i>	110
Figura 74 <i>Mapa mental de la idea principal</i>	119
Figura 75 <i>Entrevista a participantes</i>	120
Figura 76 <i>Entrevista a empleados de Cloud Solutions Network</i>	121
Figura 77 <i>Formato de cuestionario a participantes</i>	122
Figura 78 <i>Formato de la encuesta de la empresa</i>	123

Índice de Tablas

Tabla 1 <i>Viabilidad económica. Costo inicial</i>	12
Tabla 2 <i>Viabilidad económica. Costo de mantenimiento.....</i>	13
Tabla 3 <i>Organización de los roles SCRUM dentro de la empresa</i>	21
Tabla 4 <i>Organización de los eventos SCRUM dentro de la empresa</i>	22
Tabla 5 <i>Requerimientos funcionales</i>	42
Tabla 6 <i>Requerimientos no funcionales</i>	42
Tabla 7 <i>Herramientas utilizadas en el ambiente de desarrollo</i>	87
Tabla 8 <i>Herramientas utilizadas en el ambiente de integración</i>	89
Tabla 9 <i>Resultado de pruebas técnicas realizadas.</i>	91
Tabla 10 <i>Cuadro de ideas de tema de tesis.....</i>	117

Introducción

En la actualidad, la educación en línea ha experimentado un crecimiento significativo, impulsado tanto por la pandemia de Covid-19 como por los avances tecnológicos que han facilitado el acceso a cursos en línea. Sin embargo, la autenticidad de los certificados digitales emitidos por estas plataformas se ha convertido en un desafío debido a la facilidad con la que pueden ser falsificados. Esta problemática afecta tanto a las instituciones educativas como a los estudiantes, quienes dependen de la validez de sus diplomas para progresar en sus carreras profesionales.

La tecnología Blockchain, y en particular los contratos inteligentes en la plataforma Ethereum, presentan una solución viable para la generación y verificación de certificados digitales de manera descentralizada y segura. Esta investigación tiene como objetivo principal implementar una aplicación web que utilice estos contratos inteligentes para generar y verificar diplomas digitales, proporcionando una herramienta confiable y eficiente tanto para los estudiantes como para las instituciones educativas.

Esta tesis busca abordar la problemática de la falsificación de certificados digitales mediante la integración de tecnología Blockchain, asegurando la integridad y autenticidad de los diplomas emitidos. Se espera que esta implementación no solo incremente la confianza en los certificados digitales, sino que también mejore los procesos administrativos de las instituciones educativas, facilitando una validación rápida y segura.

1. Capítulo I – Anteproyecto de investigación

1.1. Antecedentes

En los últimos años, el aprendizaje en línea ha experimentado un notable incremento, la pandemia del Covid-19 y los avances tecnológicos han acelerado la adopción de esta nueva alternativa de estudio. Debido a sus características y facilidad de acceso, presentan una ventaja significativa y son más las personas que pueden acceder a ellos.

Los diplomas son un comprobante de la participación de una persona, los cursos en línea normalmente brindan un certificado digital al finalizar, pero de la misma manera que los diplomas físicos, se ven en peligro de su falsificación. El aumento en el número de empresas que ofrecen estos cursos, junto con la disponibilidad de herramientas que facilitan la manipulación de documentos digitales, plantea un desafío para las empresas al momento de validar la autenticidad de estos documentos. *Blockchain* es una tecnología que por sus características puede ser de gran utilidad para la generación de certificados en diferentes campos.

Wegelid (2019) en su trabajo de tesis “*Storing digital certificates using blockchain*” detalla los beneficios de utilizar *Blockchain* para solucionar el problema de los certificados físicos utilizados por los granjeros para demostrar la validez de sus productos con etiqueta KRAV, y para ello propone el desarrollo de una aplicación que permita generar certificados dentro de la *Blockchain*. Siendo la descentralización y la seguridad de la tecnología los aspectos más importantes según el autor.

Indica, que con la tecnología Hyperledger Sawtooth, que es software de código abierto, se pudo crear una aplicación que permite la generación de certificados para la validación de los productos dentro de la *Blockchain*, expone que, aunque el proyecto funcionó, es necesario realizar

más estudios y realizar un análisis más profundo si se desea aplicar en soluciones de producción, debido a que la investigación se llevó a cabo con un grupo pequeño de participantes.

Capece et al. (2020) en su artículo “*Blockchain Technology: Redefining Trust for Digital Certificates*” explican como *Blockchain* inicialmente fue utilizado para las criptomonedas, pero una vez ya consolidado en ese sector, empezaron a surgir iniciativas para utilizar dicha tecnología en otros campos, como es el caso de Blockcerts, un proyecto de código abierto realizado por la MIT y *Learning Machine*.

Los autores explican que uno de los factores más importantes para la adopción de una tecnología es la confianza que los usuarios tienen sobre ella. En el 2018 empezó un programa piloto para utilizar esta tecnología para la generación de certificados académicos utilizando Blockcerts y realizaron encuestas a distintos estudiantes de la Universidad de Roma “*Tor Vergata*”, se comprobó que la mayoría de los estudiantes estaban interesados en utilizar *Blockchain* para la certificación de su diploma educativo. Indican que el programa se llevó a cabo realizando pequeños avances, debido al poco conocimiento que existe de la tecnología y las herramientas que actualmente están disponibles. Concluyen que, *Blockchain* no es una solución para todos los problemas actuales que existen en la generación de certificados, pero es una gran alternativa, a pesar de ser una tecnología reciente donde no existe un estándar para la generación de certificados digitales.

Parra Candia (2020) en su investigación de tesis “Análisis de las capacidades y limitaciones de la tecnología de Blockchain como herramienta para la gestión de certificados institucionales” determinó que la tecnología *Blockchain* puede ser una alternativa viable para generar certificados emitidos por Instituciones, en el documento se expone el caso de uso en la Universidad del Bío-

Bío, donde fue posible utilizar un prototipo para generar certificados utilizando el proyecto Blockcerts. Considera que esta tecnología no implementa por sí sola el control de los certificados, ya que si se desea revocar un certificado, esto debe ser implementado fuera de la plataforma. A pesar de ello, cree que la tecnología *Blockchain* es una de las mejores alternativas para la generación de certificados digitales.

De la Rosa Rojas, et al. (2021) en su trabajo de investigación experimental para la maestría “Plataforma basada en *Blockchain* de emisión y validación de certificados digitales” encontraron que ha habido un incremento en el uso de certificados falsos debido al avance de tecnologías de edición. Proponen la tecnología *Blockchain* para reducir el uso de certificados falsos y así, tanto estudiantes como empleadores tienen la certeza de los certificados obtenidos. Indican, que esta tecnología permitirá a las organizaciones emitir certificados digitales inalterables, válidos y a perpetuidad. Finalmente, concluyen que la plataforma de generación de certificados digitales reduce el tiempo de entrega de los certificados, además de aumentar la seguridad y la confiabilidad.

Vipie et al. (2023) explicaron en su trabajo de investigación “*Blockchain-Based Educational Certificates: A Proposal*” cómo la Unión Europea aspira a implementar Blockchain en los servicios públicos, incluido el sector educativo, donde los certificados y diplomas digitales podrían ofrecer beneficios significativos a varios actores. Las instituciones educativas mejoran la eficiencia y seguridad de la información estudiantil; los estudiantes obtienen un mayor control sobre sus certificados y diplomas, facilitando la gestión y compartición de su información; y las empresas interesadas validan la autenticidad de los certificados digitales de manera más fiable y segura. Indican que existen proyectos piloto para impulsar el uso de la tecnología en diversas áreas. Aunque hoy en día no existe una aplicación exitosa, se espera que, en los próximos años, a medida

que la tecnología avance, nuevas aplicaciones puedan surgir.

1.2. Justificación

Los certificados educativos digitales permiten a las personas demostrar su participación en cursos o diplomados en línea. Sin embargo, actualmente, la mayoría de las entidades que brindan estos cursos no cuentan con un proceso eficaz para la generación y validación de estos documentos. Esto dificulta tanto a los estudiantes, quienes usualmente deben esperar varios días o semanas para obtener su certificado, como a las instituciones que desean validar la autenticidad de dichos documentos.

Últimamente, el aprendizaje en línea ha incrementado su popularidad, se prevé que para el año 2029, el número de usuarios que toman cursos en línea aumentará a más de 1,000 millones (Statista, 2023). Esto se refleja en el crecimiento de empresas que ofrecen estos cursos, lo que implica que la generación y validación de certificados se convertirá en un problema mayor si no se definen soluciones que beneficien a todos los involucrados.

Las características inherentes a la tecnología Blockchain, junto con las herramientas desarrolladas en torno a ella en los años recientes, representan una alternativa prometedora para abordar los desafíos que enfrentan los usuarios en la gestión y validación de la autenticidad de los certificados. Además, los contratos inteligentes introducidos por Ethereum facilitan la integración de aplicaciones web con información almacenada en la red Blockchain.

La implementación de la certificación digital de documentos educativos es imprescindible para agilizar el proceso, desde la emisión hasta la gestión por parte de los estudiantes y la validación por parte de las entidades y personas interesadas. Esto no solo reducirá los tiempos y costos asociados, sino que también brindará mayor confianza y seguridad en la integridad de los

certificados. Los principales beneficiarios de este estudio serán tanto los estudiantes, que podrán gestionar sus certificados de manera más eficiente, como las entidades verificadoras, que respaldarán un proceso de validación más ágil y confiable.

1.3. Planteamiento del Problema

Los certificados y diplomas son documentos que permiten demostrar la participación en una carrera, curso o entrenamiento al momento de solicitar un trabajo. Pero debido a los avances de la tecnología, cada vez son más fáciles de falsificar estos documentos, siendo el proceso de validación un problema. Este es un asunto que afecta a la mayor parte del mundo, solo en los inicios del 2024 se han dado a conocer distintos casos en diferentes países. Nganga (2024), menciona que, en Kenia, debido a un defecto en el sistema encargado de la verificación de certificados, alrededor de 2000 personas obtuvieron un trabajo en el Estado. La mayor parte de los documentos falsificados son diplomas universitarios. India Times (2023), refiere que en India se han encontrado al menos dos casos este año de personas que falsificaron títulos universitarios para que otras pudieran obtener visas de estudiante.

En Latinoamérica la situación no es diferente. Ruiz (2023), expresa que el año pasado en Perú encontraron personas que habían falsificado el título de Magisterio y habían obtenido puestos de trabajo en escuelas del Estado como profesores. El Ministerio de Educación indicó que se debió a fallas del sistema y necesitaron más de dos meses para resolver el problema.

En México, el Instituto Tlaxcalteca para la Educación de los Adultos advierte no comprar certificados escolares, ya que adquirir dichos documentos es un delito. Indican que existen personas que ofrecen certificados escolares para los niveles de primaria, secundaria y preparatoria. López (2023), expone que, en San Luis de la Paz del mismo país, la administración municipal ha detectado certificados escolares falsos. Debido al crecimiento industrial, cada vez son más las

empresas que solicitan tener algún grado educativo y varias personas han optado por adquirir estos certificados escolares.

Guatemala cuenta en su código penal un conjunto de artículos que castigan la falsificación de documentos. Código Penal (1984) establece los delitos y sus castigos en los artículos del 321 al 327, a pesar de ello también se ve afectada por la falsificación de documentos. Gramajo (2019), menciona que, se dio a conocer de una red de falsificadores que trabajaba cerca de la Torre de Tribunales, los títulos universitarios eran falsificados en alrededor de cinco días a un costo de cinco mil quetzales.

De León (2021), señala que un ex decano fue despedido por la falsificación de documentos al certificar a dos abogados que se postularon para magistrados del Tribunal Supremo Electoral, los títulos les ayudaron a obtener más puntos y así adquirir los puestos de trabajo.

A nivel institucional, la empresa *Cloud Solutions Network* se encarga de brindar cursos en línea a distintas empresas; al finalizar, son entregados como muestra de participación. Actualmente, no existe un sistema eficiente para gestionar los certificados; el poco control que se tiene se realiza en hojas de Excel. Al momento en que una persona o empresa desea validar la autenticidad de un diploma, el proceso es lento y en al menos dos ocasiones la información no fue encontrada, aunque el estudiante haya participado.

La falsificación de diplomas es otro problema que puede ocurrir, debido a la plantilla que se utiliza para la generación de los documentos, además de no tener ningún sistema de validación como un código QR.

Los estudiantes y personas que participan en estos cursos se ven afectados si la empresa cierra operaciones o pierde la información, debido a que está a su vez fungir como ente validador

de la autenticidad de los diplomas generados. Al momento de ya no existir, los diplomas pierden su validez. A causa de ello, se vuelve necesario generar un mecanismo de validación independiente de la empresa, que permita tener el control de las certificaciones a los estudiantes y personas participantes.

A causa del aumento de cursos en línea, cada vez son más las personas que utilizan esta alternativa para adquirir conocimientos. La falta de validación genera un problema tanto para los participantes, que no pueden verificar la validez de su certificado de manera sencilla y eficaz, así también para las empresas, que se ven expuestas a personas que no cumplen los requisitos necesarios para optar a una plaza de trabajo.

Por las características que brinda la tecnología *Blockchain* lleva al cuestionamiento si es viable para disminuir la falsificación de certificados educativos digitales, facilitando el control, la generación y verificación de los documentos en una manera eficiente y efectiva.

1.4. Objetivos

1.4.1. Objetivo General

Implementar la tecnología Blockchain de Ethereum en el proceso de generación y validación de certificados educativos digitales en la empresa Cloud Solutions Network, S.A.

1.4.2. Objetivos Específicos

- Desarrollar una aplicación web descentralizada que reduzca el tiempo de espera en la generación de los certificados digitales.
- Almacenar la información de los certificados educativos digitales de manera segura dentro de la Blockchain de Ethereum.
- Definir una funcionalidad dentro de la aplicación web que permita obtener el porcentaje de

intentos de falsificación.

- Incrementar el nivel de satisfacción de los estudiantes y de la empresa en el proceso de generación y validación.

1.5. Viabilidad

1.5.1. *Viabilidad de mercado*

La viabilidad del proyecto es alta, la empresa Cloud Solutions Network cuenta con una máquina virtual EC2 en *Amazon Web Services* con la versión de *Ubuntu Server 20.04*, además de una base de datos PostgreSQL en Amazon RDS. Se cuenta con archivos de *Excel* y documentos que son utilizados para la generación de certificados, esto facilita la toma de requerimientos y la definición de los contratos a utilizar.

La implementación de la certificación digital de documentos educativos mediante la empresa Cloud Solutions Network se ve respaldada por su experiencia y capacidad para crear plataformas eficientes, que es crucial para asegurar el éxito del proyecto. Su participación garantiza la implementación de un sistema que no solo cumpla con los estándares de seguridad y eficiencia requeridos, sino que también se adapte a las necesidades específicas de las instituciones educativas y las empresas interesadas.

Actualmente, existen distintas plataformas que facilitan la publicación y administración de los contratos inteligentes, pero una de las más utilizadas es *Alchemy*, la cual facilita el despliegue de estos a la red principal de *Ethereum*, además de proporcionar herramientas que permiten la integración con aplicaciones web descentralizadas.

Las instituciones educativas asociadas a Cloud Solutions Network experimentarán mejoras significativas en sus operaciones, al contar con una plataforma que les permite ofrecer certificaciones digitales de manera más eficiente y confiable. Esto es crucial en un contexto donde

la demanda de aprendizaje en línea está en aumento constante.

1.5.2. Viabilidad Técnica /Tecnológica

La implementación de la certificación digital de documentos educativos mediante la tecnología Blockchain cuenta con un respaldo sólido de conocimientos y herramientas tecnológicas. Existen nuevos avances y herramientas que permiten al desarrollo de aplicaciones basadas en Blockchain que garantizan la efectividad y seguridad del proyecto. La tecnología *Blockchain* presenta distintos beneficios que pueden llegar a ser importantes para la generación y validación de los diplomas educativos digitales. Debido a ser una red distribuida, la información se almacena en distintos nodos que forman parte de la red, esto permite tener disponibilidad de la información en todo momento. También permite la integridad de la información, ya que, al momento de crear un bloque, este es validado por cada nodo de la red, en caso la información sea falsa, el nodo que proporciono el bloque es removido de la red. Esto es de gran importancia para los usuarios involucrados, debido a que pueden asegurarse de que la información almacenada no fue modificada.

Los contratos inteligentes permiten el almacenamiento de código dentro de la red, en estos se puede programar distintas funcionalidades que permiten a aplicaciones externas aprovechar los beneficios de la *Blockchain* y así extender la funcionalidad a otros ámbitos. Debido a esto, la tecnología *Blockchain* y los contratos inteligentes brindan una alternativa para el almacenamiento y validación de documentos digitales.

1.5.3. Viabilidad Soporte

Actualmente, el desarrollo de contratos inteligentes en *Ethereum* involucra diferentes herramientas entre las que se puede mencionar:

- *Solidity* es el lenguaje de programación más utilizado para la creación de contratos inteligentes. Es un lenguaje que está en constante actualización, tiene una documentación detallada en su sitio web y un foro para la comunidad.
- Ethereum proporciona la librería Web3, la cual se utiliza en las aplicaciones web para la integración con los contratos inteligentes. Tiene un gran soporte por parte de la comunidad y por parte de *Ethereum*.
- *Alchemy* es una empresa que permite trabajar con aplicaciones web3, las cuales utilizan Blockchain. Cuentan con soporte para los usuarios. También proporcionan cursos y documentación para los desarrolladores que desean utilizar su plataforma.

1.5.4. Viabilidad administrativa

La implementación de la certificación digital de documentos educativos a través de Cloud Solutions Network se apoya en la capacidad técnica de esta empresa en el desarrollo de soluciones tecnológicas y su experiencia en el proceso de generación de certificados digitales. A pesar de que el proceso actual se realiza utilizando herramientas como Excel y Google Drive, el conocimiento adquirido facilita la implementación de la presente investigación.

La aplicación necesita de dos tipos de usuarios, el administrador y el usuario final. El usuario administrador es toda aquella persona que puede realizar operaciones dentro de la aplicación, este usuario podrá administrar usuarios, los estudiantes, los cursos, generar certificados y asignar estudiantes a cursos, este usuario será una persona de la empresa. El usuario final es aquel que podrá ingresar a la aplicación web solamente para verificar la autenticidad de un certificado digital sin la necesidad de tener un usuario registrado en la aplicación. Para ello existirá una URL a la cual cualquier persona podrá acceder para verificar la autenticidad de un certificado.

1.5.5. Viabilidad Económica

Luego de verificar distintas empresas que proporcionen servicios para la publicación de contratos inteligentes. Se optó por la utilización de *Alchemy* para el trabajo relacionado con la *Blockchain*.

Para la aplicación web se considera a Amazon AWS. Esto involucra el servidor web, la instancia de base de datos y la transferencia de datos. A continuación, en la tabla 1 se detallan los gastos generales para desarrollar la aplicación de generación y validación de certificados educativos digitales, estos se refieren solamente al costo inicial, la tabla 2 detalla los costos de mantenimiento por año.

Tabla 1

Viabilidad económica. Costo inicial

Concepto de Costo Inicial	Monto
Salario del programador durante la duración del proyecto	Q 36,000.00
Salario del encargado de infraestructura	Q 12,000.00
Análisis del sistema	Q 9,500.00
Análisis	Q 5,000.00
Pruebas	Q 2,500.00
Capacitación	Q 2,000.00
Instancia de Amazon EC2	Q 1000.00
Instancia de Amazon RDS	Q2,000.00
Costos por transferencia de datos	Q 250.00
Servicio de Alchemy	Q2000.00
Sub Total	Q57,500.00
Total	Q 62,750.00

Nota: Estimación de gastos en quetzales durante el desarrollo del proyecto.

Tabla 2

Viabilidad económica. Costo de mantenimiento

Concepto de costo	Monto (Q)	Observaciones
Instancia de Amazon EC2	Q 2,000.00	Por año
Instancia de Amazon RDS	Q 4,000.00	Por año
Costos de transferencia de datos	Q 500.00	Por año
Servicio de Alchemy	Q 4,000.00	Por año
Total	Q 10,500.00	

Nota: Estimación de gastos en quetzales por año de mantenimiento.

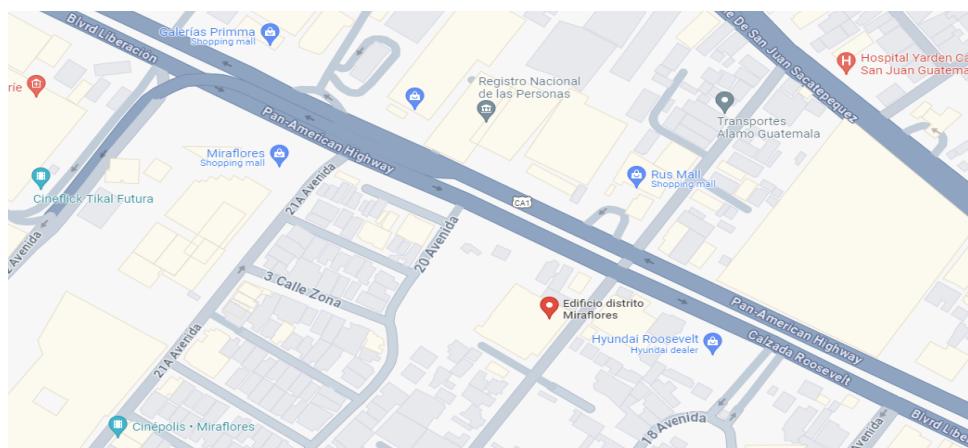
1.6. Alcance

1.6.1. Geográfico

El proyecto se desarrollará exclusivamente en las instalaciones de *Cloud Solutions Network*, situadas en la 19 Avenida 2-78, zona 11, Edificio Distrito Miraflores, oficina 501, en la ciudad de Guatemala, como se muestra en la Figura 1.

Figura 1

Ubicación de la empresa Cloud Solutions Network



Nota. La figura representa la ubicación utilizando Google Maps.

1.6.2. *Tecnológico o Técnica*

El proyecto se basa en una infraestructura tecnológica específica para su desarrollo. Se utilizarán servicios y herramientas disponibles en la plataforma de *Amazon Web Services* (AWS) y tecnologías asociadas al desarrollo de aplicaciones web y contratos inteligentes en Blockchain.

AWS es una solución que brinda herramientas que facilitan el desarrollo y despliegue de aplicaciones web. Actualmente, es la empresa *Cloud* más importante del mundo. Se implementará un servidor virtual utilizando EC2 en como plataforma base, configurado con *Ubuntu Server* 20.04, el cual es uno de los sistemas operativos más utilizados alrededor del mundo, además de ser de código abierto, lo cual facilita el soporte y el acceso de información.

La aplicación web utilizará Javascript como lenguaje de programación y Node.js como entorno de ejecución de este en lado del servidor debido a que muchas de las herramientas para el desarrollo Blockchain están hechas en esta plataforma, además de la documentación y la gran comunidad, brindando una facilidad para el desarrollo e integración con tecnologías web3.

1.6.3. *Persona/Empresa*

Brindar a la empresa *Code Solutions Network* una aplicación web que permita la administración de los cursos, estudiantes y la generación de certificados digitales, facilitando el proceso actual.

Además, proporcionar una aplicación web para todos los estudiantes y personas interesadas en validar la autenticidad de un diploma digital generado por *Code Solutions Network*.

1.6.4. *Temporal*

El desarrollo del proyecto se llevará a cabo en un período total de 10 meses. Durante los primeros 4 meses del año 2024 se llevará a cabo la fase de requerimientos. Durante este período,

se recopilarán y analizarán los requisitos del proyecto, incluyendo las necesidades específicas de *Cloud Solutions Network* y las expectativas del cliente. Se llevarán a cabo reuniones con los actores involucrados para definir claramente los objetivos y alcances del proyecto.

Después de la fase de requerimientos, se iniciará el desarrollo de la aplicación, que se extenderá a lo largo de 4 meses. Durante este tiempo, se llevará a cabo el diseño, la programación y la implementación de la solución tecnológica propuesta, asegurando su alineación con los requisitos identificados en la fase anterior.

Los dos meses siguientes estarán dedicados a la implementación, pruebas y capacitación del personal de *Cloud Solutions Network* en el uso de la nueva aplicación. Durante este período, se realizarán pruebas exhaustivas para garantizar el funcionamiento adecuado y la integridad de la aplicación. Se proporcionará capacitación adecuada al equipo para maximizar su eficiencia y productividad en el uso de la nueva herramienta.

1.6.5. Temático

El proyecto se enfoca en la aplicación de tecnologías *Blockchain* para la gestión y validación de certificados educativos digitales. Esta iniciativa abarcará el análisis, diseño y desarrollo de un sistema descentralizado basado en esta tecnología para administrar, emitir y verificar la autenticidad de certificados educativos en formato digital.

1.7. Pregunta de Investigación

1.7.1. Pregunta general

¿Es posible implementar la tecnología Blockchain de Ethereum en el proceso de generación y validación de certificados educativos digitales para mejorar la seguridad y eficiencia en la empresa Cloud Solutions Network, ubicada en el departamento de Guatemala y dedicada a

la enseñanza de cursos en línea?

1.7.2. Preguntas específicas

- ¿Cómo desarrollar una aplicación web descentralizada que reduzca el tiempo de espera en la generación de los certificados digitales?
- ¿Se puede almacenar la información de los certificados educativos digitales de manera segura dentro de la *Blockchain* de Ethereum?
- ¿Es posible definir una funcionalidad dentro de la aplicación web que permita obtener el porcentaje de intentos de falsificación?
- ¿Se puede incrementar el nivel de satisfacción de los estudiantes y los empleados de la empresa al implementar esta aplicación?
- ¿Se puede adaptar esta solución a cualquier entidad que emite certificados digitales?

1.8. Hipótesis

La tecnología Blockchain de Ethereum se implementará entre un 50% y 80% en el proceso de generación y validación de certificados educativos digitales en la empresa Cloud Solutions Network.

1.9. Variables

1.9.1. Variables Independientes

- Proceso de generación de certificados: Procedimiento implementado dentro de la aplicación web para la generación de certificados educativos digitales, incluyendo la creación del documento PDF y su envío por correo electrónico.
- Método de almacenamiento de información en la *Blockchain*: Procedimiento utilizado para almacenar la información necesaria en la *Blockchain* sin exponer información sensible de los participantes.

- Retroalimentación del usuario: Mecanismo que permite a los usuarios indicar si un certificado es válido o no, recopilando estos datos para crear un historial que sirva de referencia a la empresa.

1.9.2. Variables Dependientes

- Tiempo de entrega de los certificados educativos: El tiempo transcurrido desde la finalización del curso hasta la entrega del certificado por correo electrónico, medido en días.
- Nivel de privacidad y seguridad de la información almacenada en la *Blockchain*: Grado de protección de la información de los usuarios, garantizando que solo se almacenen identificadores para mantener la privacidad.
- Costo de generación de certificados: El costo asociado a la generación de cada certificado a partir de la implementación de la aplicación.
- Porcentaje de falsificaciones detectadas: El porcentaje de certificados falsos identificados por los usuarios mediante la validación a través de la aplicación web.
- Nivel de satisfacción de los estudiantes: Medido mediante encuestas o retroalimentación directa de los estudiantes sobre su experiencia con los certificados digitales.
- Nivel de satisfacción de la empresa: Evaluado a través de métricas internas y retroalimentación sobre la eficiencia y efectividad del nuevo sistema de generación y validación de certificados.

1.10. Indicadores

- El tiempo de generación de un certificado, actualmente el proceso para la entrega de un certificado es de una a dos semanas desde que termina el curso.

- La reducción del tiempo empleado por la persona que genera los certificados, debido a que esta persona no está solamente a cargo de la generación de certificados, utiliza parte de su tiempo en este proceso. Al finalizar un curso puede llevarse de 4 a 8 horas para generar y enviar todos los certificados.
- Nivel de satisfacción de los estudiantes, determinar la aceptación de los estudiantes de la nueva plataforma y proceso para generación de los certificados.
- Reducción en el tiempo de validación de un certificado, actualmente el tiempo de espera es alrededor de una semana, debido a que la persona a cargo tiene otras tareas bajo su responsabilidad.

1.11. Supuestos

- Según Fredick Wegelid, la tecnología Blockchain permite almacenar los certificados digitales para los productos que necesitan la etiqueta KRAV de manera transparente, segura y descentralizada. Asimismo, señala que la tecnología cuenta con capacidades de escalabilidad que permite ser utilizada en entornos reales. (Wegelid, 2019).
- De acuerdo con el artículo “*Blockchain Technology: Redefining Trust for Digital Certificates*”, la tecnología Blockchain no es la solución a todos los problemas de credenciales existentes. Es una tecnología nueva y compleja, por lo que su implementación no es inmediata y requiere más tiempo para utilizarse en proyectos más prometedores. A pesar de ello, ofrece la posibilidad de mejorar los sistemas de credenciales actuales. (Capece, et al. 2020)
- Pivaral y Cercado (2020), indican que el desarrollo de una aplicación web con integración de *smart contracts* permite almacenar la información de manera íntegra y fiable, pero

recomiendan utilizar información encriptada para la comunicación de la aplicación web y la red *Blockchain* y así facilitar el acceso a la información.

- Según De la Rosa et al. (2021), por medio de *Blockchain*, las organizaciones podrían emitir certificados digitales inalterables, válidos y a perpetuidad. A diferencia de los sistemas actuales, los certificados digitales basados en la tecnología *Blockchain* aumentarán su propuesta de valor de manera importante, eliminando de manera progresiva los certificados en papel, procesos de emisión, verificación, distribución engorrosa y altamente operativa.

1.12. Métodos de investigación o metodología

1.12.1. Generalidades

El enfoque de la investigación es cuantitativo, ya que examina la medición precisa y objetiva de las variables relacionadas con la generación y validación de certificados educativos digitales en una empresa guatemalteca mediante el uso de contratos inteligentes en la plataforma *Blockchain*. Esta metodología permite recolectar los datos numéricos que serán analizados para confirmar la hipótesis definida.

El alcance de la investigación será del tipo correlacional, debido a que permite identificar las relaciones existentes entre la implementación de contratos inteligentes y la eficacia en la reducción de tiempos de validación y entrega de certificados, así como en la disminución en las falsificaciones.

El diseño de esta investigación es pre experimental, dado que se centra en manipular una variable independiente, en el presente trabajo, la implementación de contratos inteligentes en la generación de certificados digitales y su efecto sobre la reducción en el tiempo de validación y entrega de certificados educativos digitales, así como la reducción en las falsificaciones.

1.12.2. Población y muestra

La población objetivo de esta investigación está constituida por empresas guatemaltecas que ofrecen cursos en línea. Estas incluyen diversas plataformas web educativas y entidades académicas, como por ejemplo el Intecap, Planeta Virtual 502 y la Universidad Rafael Landívar, las cuales proporcionan formación tanto a individuos como a otras empresas.

Para este estudio, se seleccionará una muestra no probabilística por conveniencia, compuesta por la empresa Cloud Solutions Network, ubicada en el departamento de Guatemala, dedicada a la enseñanza de cursos en línea. El criterio de selección se basa en la disposición de la empresa a participar en el estudio y en su capacidad para cumplir con los criterios de inclusión definidos. Este tipo de muestreo se justifica por la necesidad de trabajar con una entidad que pueda proporcionar un acceso inmediato y controlado a los procesos y datos necesarios para realizar un estudio experimental efectivo.

1.12.3. Instrumentos de la investigación

Los instrumentos de la investigación permiten obtener la información necesaria para conocer más acerca del problema, el proceso actual dentro de la empresa y el resultado de la implementación de la aplicación. Teniendo en cuenta la información necesaria para el proyecto, los instrumentos de medición serán:

- Entrevista
- Encuesta
- Análisis de documentos
- Requerimientos funcionales y Requerimientos no funcionales

Estos documentos se encuentran disponibles en los Anexos C, D y E.

1.12.4. Marco de trabajo SCRUM

SCRUM es un marco ágil utilizado para la gestión y desarrollo de proyectos, especialmente en el área del desarrollo de software. Implementa ciclos breves y consistentes, conocidos como *sprints*, que permiten ajustar y refinar el trabajo continuamente basándose en la retroalimentación recibida.

Debido a su naturaleza ágil, esta metodología es la adecuada para el proyecto debido al uso de Blockchain, que, dada su novedad y su escaso uso en la generación de certificados digitales, exige un análisis constante. Además, promueve una mejora sustancial en la comunicación y colaboración con los usuarios finales, al ser iterativo e incremental, SCRUM optimiza la eficiencia del desarrollo y minimiza los riesgos, garantizando el cumplimiento de los plazos y presupuestos establecidos.

La metodología consiste en tres roles, en la Tabla 3 se detalla la organización de estos dentro de la empresa para el proyecto.

Tabla 3

Organización de los roles SCRUM dentro de la empresa

Rol	Cargo	Descripción
Scrum Master	Full Stack Developer	Para facilitar el cumplimiento de la metodología y cada uno de sus eventos.
Product Owner	Gerente IT	Utilizará los conocimientos de las herramientas actuales y definirá los módulos requeridos para cumplir con el objetivo del proyecto.
Developer	Full Stack Developer	Cada uno de los involucrados en la implementación de la aplicación. Desde el desarrollo de la aplicación web, administración y creación de los servidores, como también la ejecución de pruebas.
	Cloud & DevOps Engineer	
	Gerente IT	

Backend Developer

Nota: Los encargados de cada rol pueden estar sujetos a cambios.

La metodología cuenta con cinco eventos, en la Tabla 4 se detalla la organización de estos dentro de la empresa para el proyecto

Tabla 4

Organización de los eventos SCRUM dentro de la empresa

Evento	Ubicación	Planificación	Duración
Sprint Planning	Google Meet	Último jueves de cada sprint de 20:00 a 22:00	2 horas
Daily Scrum	Google Meet	Lunes, miércoles y viernes de 21:30 a 21:45	15 minutos
Sprint Review	Google Meet	Último miércoles de cada sprint de 20:30 a 21:15	1 hora
Sprint Retrospective	Google Meet	Último miércoles de cada sprint de 21:15 a 21:30	15 minutos
Sprint	Slack / Google Meet		4 sprints de 4 semanas

Nota: Los horarios pueden estar sujetos a cambios.

1.13. Otras consideraciones

Además, se integrarán contratos inteligentes utilizando *Solidity* en la red de *Blockchain* de *Ethereum* debido a que hoy en día, es la mayor red para la generación de contratos inteligentes, además de integrar la mayor parte de tecnologías desarrolladas que facilitan el desarrollo y despliegue de información. *Alchemy* será la plataforma utilizada para la creación y gestión de estos contratos inteligentes, garantizando su funcionamiento y seguridad dentro del ecosistema blockchain.

Estas acciones se han tomado con el objetivo de asegurar la compatibilidad, eficiencia y capacidad para cumplir con los objetivos del proyecto.

1.13.1. Diagramas UML

El lenguaje de modelado unificado (UML) es un lenguaje estandarizado para el modelado que consiste de un conjunto de diagramas desarrollados para ayudar a especificar, visualizar, construir y documentar los componentes de sistemas de software.

Debido a la facilidad de uso y a su expresividad, estos diagramas serán utilizados para definir los componentes del sistema y el flujo completo de la aplicación. Actualmente existen 14 diagramas UML, divididos en dos categorías, diagramas estructurales y comportamiento, pero en el proyecto se utilizarán solamente cinco, los diagramas de secuencia, de actividades, de caso de uso, de componente y de despliegue, estos debido a las características que proporcionan y se desea definir al momento del análisis del proyecto.

1.14. Planificación de capítulos

La planificación de los capítulos definida en la *Figura 2* permite identificar las distintas fases de las cuales se compone el proyecto de investigación. Para la planificación se tomó en cuenta el tiempo estipulado por la Universidad Mariano Gálvez.

Figura 2

Cronograma de actividades



1.15. Estimación de recursos

La estimación de recursos es un aspecto importante en la planificación del proyecto, ya que proporciona una previsión de los recursos necesarios para alcanzar los objetivos. A continuación, se detallan los recursos para el presente proyecto.

1.15.1. Recursos humanos

1.15.1.1. Desarrollador Blockchain. Persona encargada de la programación de los contratos inteligentes en Ethereum y su despliegue.

1.15.1.2. Desarrollador Full Stack. Persona encargada del desarrollo de la aplicación web3 y su integración con los contratos inteligentes.

1.15.1.3. Ingeniero Cloud. Persona encargada de la creación y administración de los recursos dentro de la Cloud de AWS.

1.15.1.4. Gestor de proyecto. Encargado de definir y detallar los requerimientos del proyecto.

1.15.2. Recursos materiales

1.15.2.1. Hardware. Servidor de pruebas en AWS y computadoras portátiles para el desarrollo de la aplicación.

1.15.2.2. Software. Entornos de desarrollo Remix IDE y Visual Code.

1.15.3. Recursos financieros

1.15.3.1. Salarios del personal. Salarios de los desarrolladores y personas involucradas.

1.15.3.2. Servicios Cloud. Costos de la instancia de EC2 y la base de datos en Amazon RDS.

1.15.3.3. Servicios de Blockchain. Costo de Ether para la publicación de los contratos inteligentes y generación de certificados.

1.15.4. Recursos intangibles

1.15.4.1. Conocimiento y experiencia. Experiencia del equipo del desarrollo en las diferentes tecnologías para llevar a cabo el proyecto.

1.15.4.2. Reputación de la empresa. Uso de la reputación de la empresa para poner en práctica el proyecto y atraer a más usuarios.

2. Capítulo II – Marco teórico

2.1. Blockchain

2.1.1. Historia

Haber y Stornetta (1991) publicaron el documento *How to Time-Stamp a Digital Document*, en el cual describieron el problema de validar si un documento digital ha sido manipulado y propusieron dos soluciones utilizando criptografía. Dichas soluciones emplean funciones *hash* para generar un identificador único basado en la información de la fecha de creación y el contenido del documento.

Satoshi Nakamoto (2008) publicó el *whitepaper Bitcoin: A Peer-to-peer Electronic Cash System*, donde se describe la creación de una moneda digital basada en funciones *hash*. En este sistema, distintos nodos distribuidos se encargan de validar las transacciones y generar los bloques que serán añadidos a la *Blockchain*, dando origen Bitcoin, la primera aplicación de esta tecnología.

2.1.2. Funcionamiento

Frizzo-Barker et al. (2019) describen *Blockchain* como un libro mayor digital descentralizado que facilita la transferencia de persona a persona de diferentes valores digitales, desde una moneda digital hasta productos tangibles o títulos de propiedad, sin la necesidad de intermediarios como bancos, contadores o abogados. Cada nodo cuenta con una copia completa de la *Blockchain*, lo que facilita la verificación y la transparencia de todas las transacciones realizadas. Esta estructura permite establecer un mecanismo confiable sin la necesidad de intermediarios debido a su infraestructura descentralizada (Rumamurthy, 2020).

Esta tecnología permite almacenar transacciones dentro de una red distribuida utilizando bloques que son encriptados, validados y verificados por todos los nodos participantes. Todo inicia

con una transacción, la unidad más pequeña de almacenamiento de información dentro de la cadena de bloques. Lantz y Cawrey (2020) la definen como la representación del movimiento de valor desde una dirección hacia otra. Cada transacción debe estar firmada por una clave privada, la cual es posteriormente confirmada mediante la clave pública antes de ser añadida a un bloque.

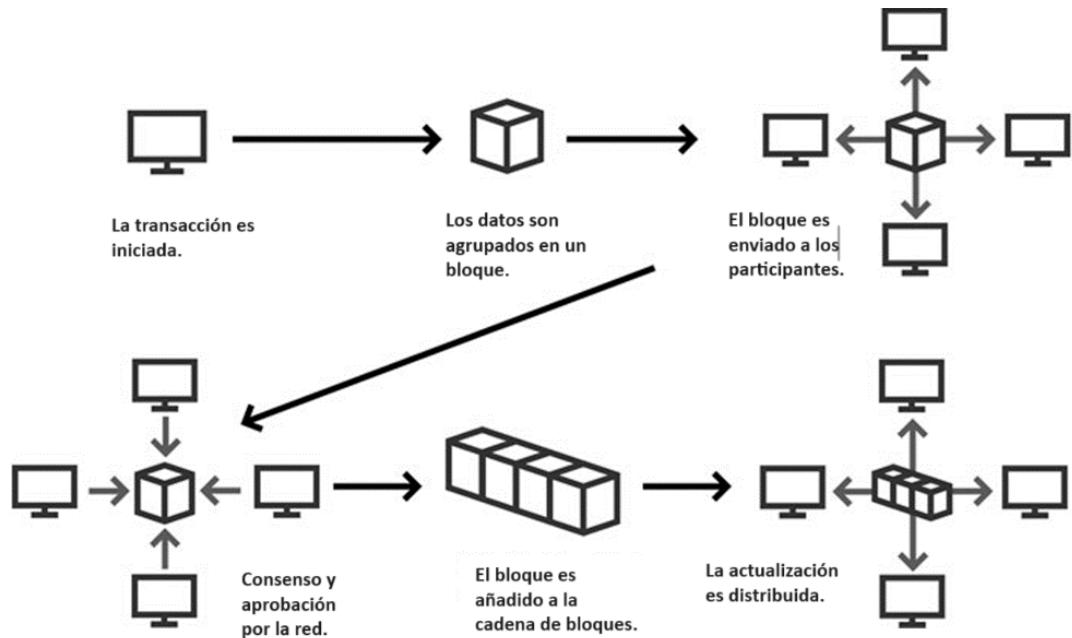
Un bloque contiene el listado de las transacciones que fueron almacenadas dentro de la *Blockchain* en un tiempo determinado. Cada cadena de bloques define las dimensiones, intervalos de tiempo y eventos que determinan cuándo se crea el bloque (Mohan, 2023). También conocidos como bloques hash, al ser creados almacenan un *snapshot* con la información hasta ese momento dentro de la cadena de bloques. Este *hash* es utilizado por los diferentes nodos de la red para verificar si la información ha sido manipulada, ya que si la información es distinta, el *hash* generado es diferente.

Los nodos son todas las computadoras que forman parte de la red *Blockchain*, dependiendo del tipo de nodo, la funcionalidad de estos puede variar dentro de la red. Los nodos completos (*full nodes*) son aquellos que tienen una copia completa de la cadena bloques, entre sus funcionalidades se destacan, la validación de bloques, enrutamiento, minería y servicios de billeteras. Los nodos ligeros son dispositivos que tienen un subconjunto de la información; estos solamente pueden verificar transacciones utilizando los nodos *archive full nodes* (Antonopoulos y Harding, 2023).

El funcionamiento simplificado de *Blockchain*, se puede ver reflejado en la figura 3.

Figura 3

Simplificación del funcionamiento de Blockchain



Nota. Adaptado de A P2P Optimistic Fair-Exchange (OFE) Scheme for Personal Health Records Using Blockchain Technology, - Scientific Figure on ResearchGate.

(https://www.researchgate.net/figure/How-Blockchain-works-22_fig1_342407089)

2.1.3. Tipos de Blockchain

Blockchain es una tecnología de registro descentralizado que permite la transferencia y almacenamiento seguro de información digital. Su estructura consiste en bloques de datos enlazados de manera secuencial, formando una cadena. Cada bloque contiene un conjunto de transacciones verificadas, un sello de tiempo y un hash del bloque anterior, lo que garantiza la inmutabilidad y la seguridad de la información registrada.

El funcionamiento de la *Blockchain* se basa en una red distribuida de nodos que colaboran en la validación y registro de las transacciones. Cuando una transacción es iniciada, se agrupa con

otras transacciones pendientes en un bloque. Este bloque es transmitido a la red de nodos, donde cada nodo verifica la validez de las transacciones contenidas. Una vez que un bloque es validado, se añade a la cadena de bloques existente, y su contenido se convierte en inmutable, es decir, no puede ser alterado sin modificar todos los bloques subsecuentes, lo que requeriría el consenso de la mayoría de los nodos de la red.

- **Blockchain 1.0.** Swan (2015) describe Blockchain 1.0 como la criptomoneda, fue la primera funcionalidad de esta tecnología a través de Bitcoin. Este tipo de cadena de bloques está diseñada únicamente para la transferencia de dinero digital y no tiene ningún soporte para instrucciones lógicas. Bitcoin, por su parte, solo soporta un script mínimo para la definición de condicionales en la creación de transferencias, según Ramamurthy (2020).
- **Blockchain 2.0.** Este tipo de Blockchain permite el control y la transferencia de criptomonedas, pero también agrega el uso de lógica a través de los contratos inteligentes (Ramamurthy, 2020). Esta nueva versión amplía las posibilidades de aplicación de esta tecnología. Además de las criptomonedas, también puede usarse en otros ámbitos como bienes raíces, documentos legales y contratos de propiedad. Ethereum, creado en 2015, fue la primera Blockchain 2.0.

2.2. Ethereum

Antonopoulos y Harding (2018) indican que, al igual que Bitcoin, Ethereum también es una máquina de estados distribuida; sin embargo, en lugar de solo llevar el control de las transacciones de la criptomoneda, también permite almacenar información de propósito general dentro de las transacciones a través de contratos inteligentes. El sitio de Ethereum (s.f.) se define como: “Red de computadores alrededor del mundo que comparten un conjunto de reglas llamadas el protocolo Ethereum. La red Ethereum actúa como los cimientos para comunidades, aplicaciones, organizaciones y recursos digitales que cualquiera puede construir y utilizar.”

Ethereum fue desarrollado como una plataforma que facilita contratos persona a persona y aplicaciones utilizando su moneda, *ether*. Es también un *framework Blockchain* completo de Turing, ya que brinda la base para los lenguajes de programación con los que se pueden escribir los contratos inteligentes capaces de resolver cualquier problema computacional.

2.2.1. Máquina virtual de Ethereum

Ethereum es controlado por la Máquina Virtual de Ethereum (EVM, por sus siglas en inglés). Es una máquina virtual basada en consenso que decodifica los contratos inteligentes compilados en *bytecode* y los ejecuta dentro de la red de nodos Ethereum. También utiliza algoritmos para prevenir ataques de denegación de servicios, que son comunes en los mercados de criptomonedas (Mohanty, 2018). Es el programa que está en todos los nodos de la red y permite el funcionamiento de esta, independientemente del sistema operativo.

Wu et al. (2023) indican que inicialmente Ethereum utilizaba el mecanismo de consenso, prueba de trabajo (Proof of work) en su versión 1.0. A partir de la versión 2.0, publicada en 2022, utiliza el mecanismo prueba de participación (Proof of stake). La prueba de trabajo consiste en que los nodos de la red compiten entre ellos para resolver un problema matemático. El primero en resolverlo recibe una recompensa, una cantidad de criptomonedas y es el encargado de actualizar la *Blockchain* con el nuevo bloque de transacciones. Debido al alto consumo de recursos, actualmente, la nueva versión de Ethereum utiliza la prueba de participación, que consiste en que los participantes utilicen su criptomoneda como prueba. Cuanto mayor sea la cantidad, mayor es la probabilidad de ser elegido como encargado de añadir el nuevo bloque. Este nuevo consenso reduce el consumo de recursos, facilitando la generación de bloques. Además de la recompensa por añadir un nuevo bloque, el nodo recibe una comisión por cada transacción dentro del bloque, ya que al momento de realizar cada transacción se debe pagar una cantidad de gas dependiendo de

la cantidad de operaciones que se realizarán dentro de la máquina virtual y la cantidad de información que se desea almacenar.

2.2.2. *Gas*

Es el combustible utilizado para la ejecución de los contratos inteligentes dentro de la EVM, este puede ser adquirido comprando *ether*. Para su obtención es necesario intercambiar monedas en el mercado de criptomonedas o darse de alta como minero (Wu et al., 2023). El costo del gas varía dependiendo de la demanda de las transacciones, cuantos más usuarios estén dispuestos a pagar más por una transacción, mayor será el precio del gas. Por lo tanto, no hay un precio exacto y dependerá del precio del *ether* y la demanda de la red.

2.3. Contratos inteligentes

Según Antonopoulos (2018), los contratos inteligentes son programas informáticos inmutables que se ejecutan de manera determinística dentro de la máquina virtual de Ethereum como parte del protocolo de red. A pesar del nombre, no son contratos inteligentes ni legales. Estos programas tienen características específicas que los separan de los programas de software tradicionales debido a su naturaleza inmutable; una vez el código ha sido publicado, este no puede ser modificado y se debe crear una nueva instancia cada vez que se realice un cambio. Asimismo, al ser determinístico, el resultado de la ejecución del programa es el mismo, independientemente de qué nodo lo ejecute.

Los contratos inteligentes son escritos en un lenguaje de alto nivel llamado *Solidity*, que posteriormente es compilado a *bytecode* y ensamblado con *opcodes* encriptados y para finalmente ser desplegados dentro de la EVM. Para este proceso se puede crear un nodo o utilizar plataformas *SaaS* como Alchemy. Estas plataformas proporcionan servicios *REST* para comunicarse con el nodo de Ethereum. Alchemy e Infura son las plataformas más populares hoy en día para la

contratación de nodos. Además de proveer interfaces para acceder a las funcionalidades de Ethereum, permiten la administración y despliegue de contratos inteligentes de manera sencilla.

2.3.1. *Solidity*

Es un lenguaje de programación desarrollado para trabajar con los contratos inteligentes. El sitio oficial de Solidity lo define como: “Lenguaje orientado a objetos de alto nivel para la implementación de contratos inteligentes. Es un lenguaje diseñado para trabajar dentro de la máquina virtual de Ethereum y está influenciado por C++, Python y Javascript”. A pesar de ser un lenguaje de programación reciente, es el más popular para el desarrollo de contratos inteligentes. Existen herramientas creadas específicamente para facilitar su desarrollo, entre las cuales, destacan: Remix IDE, Hardhat, Truffle y Foundry.

2.3.2. *Herramientas*

- **Foundry.** Es un conjunto de herramientas que facilita el desarrollo de contratos inteligentes. La página oficial describe las características de la herramienta como: “Herramienta que administra las dependencias, compila el proyecto, ejecuta pruebas, despliega y permite interactuar con las distintas funcionalidades desde la línea de comandos y scripts de Solidity.”
- **Ganache.** Es una blockchain personal para el desarrollo rápido de aplicaciones distribuidas en Ethereum y Filecoin. Permite el desarrollo, prueba y despliegue de las DApps en un ambiente seguro y determinístico.

2.4. Aplicaciones descentralizadas

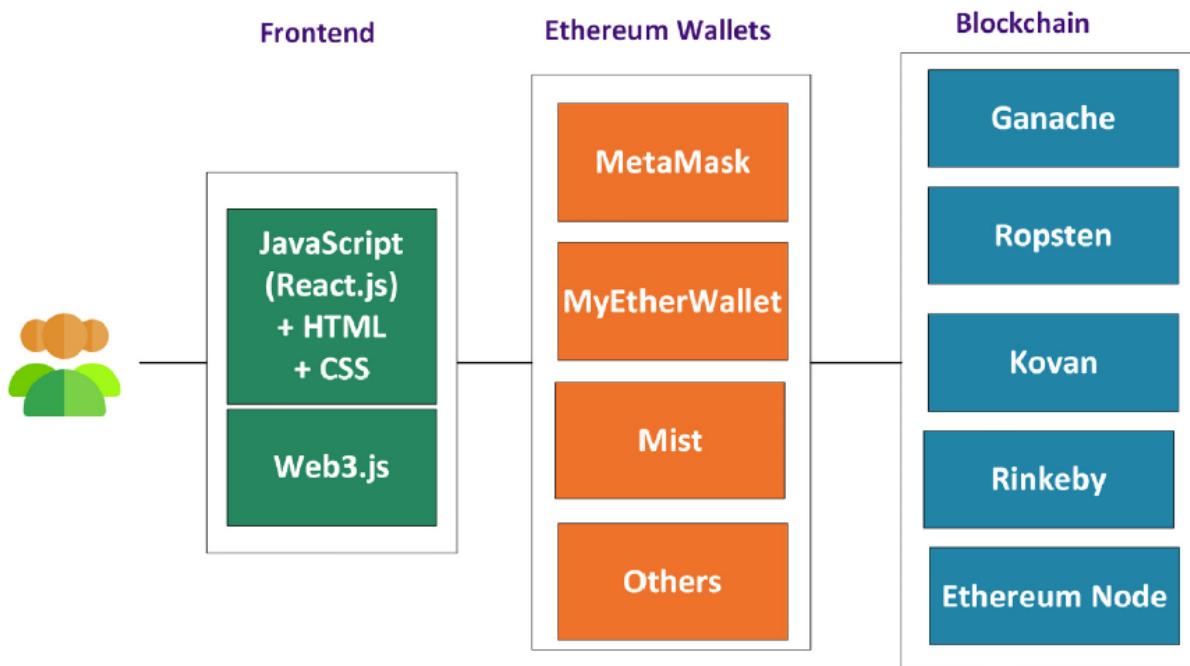
Antonopoulos (2019) indica que las aplicaciones descentralizadas, también conocidas como DApps, son aplicaciones que están completa o mayormente descentralizadas. Esto involucra todos los aspectos de las mismas: el frontend, el backend, el almacenamiento y la comunicación. Debido a su naturaleza descentralizada, presentan beneficios como la resiliencia, transparencia y

resistencia a la censura. Actualmente, existen muy pocas aplicaciones que sean totalmente descentralizadas. La mayor parte de las aplicaciones que se conectan a una red *Blockchain* tienen componentes que aún utilizan redes centralizadas. Sin embargo, se espera que a medida que la tecnología avance, cada vez haya más aplicaciones totalmente descentralizadas.

Una aplicación DApp consiste típicamente en una aplicación web que interactúa con el backend, el cual está implementado utilizando contratos inteligentes. La página web realiza llamadas al backend usando la API web3.js (Wu et al., 2023). En la figura 4 se puede observar una arquitectura simplificada de aplicaciones descentralizadas.

Figura 4

Arquitectura de aplicaciones distribuidas



Nota. Adaptado de *Learn Ethereum* (p. 671), por Wu et al., 2023

2.4.1. Aplicaciones Web

Según Wolf et al. (2017) definen las aplicaciones web como sistemas que se ejecutan desde un servidor web y son accesibles a los usuarios a través de un navegador web. Este tipo de aplicaciones permite a los usuarios interactuar mediante interfaces que realizan peticiones a sitios web.

2.4.2. Web3.js

Según el sitio oficial, Web3.js es una colección de librerías robustas y flexibles en Javascript y Typescript que permiten a los desarrolladores interactuar con nodos locales o remotos de Ethereum mediante conexiones HTTP, IPC o Websocket. Esta colección de herramientas permite crear aplicaciones eficientes y robustas dentro del ecosistema de *Blockchain*.

2.5. Certificaciones

Bertrand (2000) define la certificación como el proceso a través del cual se aseguran las competencias y habilidades de un individuo en relación con una norma formalizada. Se trata, por ello, de la certificación de unas cualificaciones individuales, de un nivel de conocimientos, de unas habilidades y probablemente de unas capacidades de aprendizaje. Para el autor, la certificación se asocia generalmente a un proceso de reconocimiento de las competencias obtenidas a través de un sistema de aprendizaje formalizado, mientras que la validación se refiere al reconocimiento de los logros menos normalizados y más diversos de los adultos. La certificación propiamente dicha debería responder a diversos objetivos que afectan en distinto grado a las personas, pero también a las empresas y a la sociedad.

2.5.1. Certificados digitales

La certificación digital tiene sus inicios a mediados de la década de 1970, cuando Whitfield Diffie y Martin Hellman (1976) publicaron el artículo titulado *New Directions in Cryptography*. Este artículo introdujo el concepto de criptografía asimétrica y propuso la utilización de un archivo público que sería consultado por las entidades pertinentes para verificar las claves públicas del sistema. Para prevenir que un atacante pudiera interceptar y modificar dicho archivo, todas las comunicaciones debían ser llevadas a cabo mediante firmas digitales. No obstante, esta propuesta inicial presentaba diversos inconvenientes tanto en términos de rendimiento como de seguridad.

La necesidad de asociar de manera confiable las claves públicas de los usuarios a su identidad llevó a Loren Kohnfelder (1978) a proponer el concepto de certificado digital de clave pública. Este certificado es un documento firmado digitalmente que contiene tanto la clave pública como la identidad del poseedor de la clave privada correspondiente, eliminando la dependencia de la base de datos pública centralizada propuesta anteriormente. Esta estructura de datos formada por la clave pública y la identidad del poseedor de la clave privada correspondiente debía ser emitida y firmada por una entidad de confianza para los usuarios de la comunicación. De esta manera, este documento firmado digitalmente, junto con un mecanismo de comprobación de autenticidad e integridad, permitía su almacenamiento en cualquier contexto no confiable de forma replicada, asegurando su disponibilidad y no alteración.

2.5.2. Certificados educativos digitales

Los certificados digitales permiten la validación oficial de habilidades y competencias adquiridas a través de cursos y programas educativos. Estos certificados son ampliamente reconocidos en los ámbitos académico y profesional, facilitando oportunidades de empleo y el progreso educativo. Según un estudio de la Universidad de Puerto Rico, estos certificados permiten

a los educadores demostrar su capacidad en la enseñanza en línea y la creación de materiales educativos digitales (Universidad de Puerto Rico, 2024).

2.6. Leyes y Normativa sobre la Emisión de Títulos y Diplomas

2.6.1. Instituciones Encargadas

En Guatemala, el sistema educativo nacional se rige por el Decreto Legislativo No. 12-91, conocido como la Ley de Educación Nacional. Según esta ley, el Ministerio de Educación, a través de sus Direcciones Regionales, es la entidad encargada de extender los diplomas y títulos que acrediten la validez de los estudios realizados en los niveles y modalidades de su competencia (Ley de Educación Nacional, 1991, art. 77).

2.6.2. Emisión de Títulos y Diplomas

La validez de los estudios realizados en los centros educativos del sistema se acredita mediante certificados extendidos por cada establecimiento y avalados por la autoridad correspondiente del Ministerio de Educación. Estos certificados aseguran que se han cumplido con los planes y programas de estudios autorizados (Ley de Educación Nacional, 1991, art. 75).

2.6.3. Castigos por Falsificación de Títulos y Diplomas

El Código Penal de Guatemala establece sanciones específicas para la falsificación de documentos públicos, entre los cuales se incluyen los títulos y diplomas educativos. La falsificación de documentos públicos es castigada con prisión y multas. En particular, el artículo 323 del Código Penal estipula que la falsificación de documentos públicos, entre ellos los títulos académicos, conlleva penas de prisión de dos a seis años y multas económicas (Código Penal de Guatemala, art. 323).

Estas disposiciones legales aseguran que tanto la emisión como la validez de los títulos y diplomas se realicen de manera adecuada y legítima, protegiendo así la integridad del sistema educativo y penalizando cualquier intento de falsificación.

2.6.4. Ley Integral de Protección de Datos Personales

- **Instituciones Encargadas y Ámbito de Aplicación.** La Ley Integral de Protección de Datos Personales en Poder de Terceros tiene como objetivo principal garantizar la protección de los datos personales en poder de terceros, asegurando su tratamiento legítimo, adecuado, proporcional, seguro, controlado e informado. El Consejo Nacional de Protección de Datos Personales y el Instituto Guatemalteco de Protección de Datos Personales son las autoridades responsables de la supervisión y aplicación de esta ley.
- **Consecuencias para Empresas e Instituciones.** Las empresas e instituciones que almacenen información personal deben cumplir con estrictas normativas de protección de datos. La ley impone la obligación de implementar medidas técnicas y organizativas para garantizar la seguridad de los datos personales. Las principales consecuencias para el incumplimiento de esta ley incluyen sanciones administrativas, daño a la reputación, acciones legales y la intervención de autoridades.
- **Impacto en la Certificación de Documentos Digitales.** La Ley Integral de Protección de Datos Personales también afecta la certificación de documentos digitales. Las instituciones que emiten y certifican documentos digitales deben asegurar que los datos personales contenidos en estos documentos estén protegidos adecuadamente. Esto incluye la implementación de cifrado de datos, autenticación segura y auditorías regulares. El incumplimiento de estas medidas puede resultar en sanciones y comprometer la validez y reconocimiento de los documentos digitales emitidos.

3. Capítulo 3 – Análisis y diseño

El estudio de diversas tecnologías y arquitecturas se realizó en función de las necesidades y requisitos del problema presentado, con el objetivo de satisfacer las demandas de la empresa Cloud Solutions Network. Se optó por Angular para la interfaz frontend, dado que la compañía ya tiene experiencia con este framework y, además, es una solución robusta que facilita la creación de sistemas complejos. Para el desarrollo del API en el backend, se eligieron Node.js como entorno de ejecución Javascript y el framework Nest.js, debido a su similitud con Angular y Java, lo que agiliza el desarrollo gracias al conocimiento existente de estas herramientas. Asimismo, la integración con la red Blockchain se simplificó mediante el uso de la librería Web3.js, desarrollada por el equipo de Ethereum para Javascript. Finalmente, se decidió utilizar Solidity para la programación de contratos inteligentes, dado que es el lenguaje más utilizado y con mayor documentación en la actualidad, siendo una opción confiable para esta tecnología emergente.

3.1. Diseño

3.1.1. Frontend

La aplicación Frontend es la interfaz donde el usuario podrá interactuar con todo el sistema. En ésta se podrá acceder a todas las funcionalidad de la aplicación a través de una ventana de inicio de sesión, podrá administrar usuarios, estudiantes, cursos, asignar estudiantes y generar certificados digitales. Para ello se va a utilizar Angular como framework, ya que facilita el desarrollo de aplicaciones web y brinda un conjunto de herramientas necesarias, como la posibilidad de realizar peticiones HTTP, las cuales serán utilizadas para comunicarse con la aplicación Backend.

Debido a Angular, la aplicación se va a desarrollar con Typescript, un superset de Javascript que aumenta las características del lenguaje de programación, como lo son interfaces,

decoradores y el tipado estatico. Este framework además define un conjunto de elementos para el desarrollo de aplicaciones, los componentes son el elemento principal, estos estan compuestos por un archivo Typescript que define la lógica del componente, un archivo HTML que define la estructura y un archivo CSS para definir los estilos.

Además de los componentes, se van a utilizar los servicios, los cuales son los elementos que definen los datos de la aplicación, en este caso van a realizar las conexiones HTTP a los recursos de la aplicación Backend. Los módulos se encargan de agrupar los componentes que pertenecen a una sección y facilita la interacción y la estructura entre entre ellos. Finalmente se van a utilizar los Guards y las rutas para definir los accesos a las distintas partes de la aplicación, con ello se van a definir las rutas que van hacer pública y las privadas, que necesitan que el usuario este autenticado para poder acceder.

3.1.2. *Backend*

Esta aplicación es la encargada de la interacción de la base datos y el smart contract con la aplicación Frontend. La aplicación estará desarrollada con Nestjs que es un framework Nodejs para el desarrollo de aplicaciones web, utilizará Typescript. Nestjs está basado en la arquitectura de Angular para facilitar el diseño de las aplicaciones, además utiliza el patrón de diseño MVC que permite organizar el código de manera eficiente.

La aplicación hace uso de las librerías Passport y JWT para definir la autenticación de los usuarios para ello utiliza el estándar Passport que facilita la creación de estrategias de autenticación y JWT que es la especificación que va a hacer utilizada en el proyecto debido a su popularidad, comunidad y seguridad. También se utiliza la librería RxJS que permite la programación reactiva y facilita el uso de llamadas asíncronas las cuales son utilizadas al momento de realizar peticiones HTTP a los servicios de los contratos inteligentes. La estructura de este proyecto estará definida

por módulos que encapsularán tareas específicas y que contendrán, controladores que definen los recursos que los usuarios podrán acceder, los servicios, que definen la lógica de los componentes, entidades, las cuales modelan las tablas de la base de datos y objetos de transferencia de datos, las cuales son clases que indican los datos que se van a compartir en las interacciones de los elementos.

Para la conexión con la base de datos se va a utilizar TypeORM, el cuál es una librería que permite la integración con la base de datos y proporciona las clases necesarias para interactuar con la misma, facilitando el proceso de creación de consulta y procesamiento de los datos. Esta librería será la encargada de conectarse con la base de datos en PostgreSQL, para ello también se utilizará la librería pg, que es requerida por TypeORM.

3.1.3. *Smart contract*

El contrato inteligente se creará por utilizando Solidity, que es un lenguaje de programación específico para esta tarea y se utilizarán librerías de Javascript para poder realizar el despliegue en la red Blockchain de Ethereum y posteriormente para su interacción. Las librerías que serán utilizadas son Hardhat y Ether. Hardhat define un conjunto de herramientas que facilitan la compilación de los archivos escritos en Solidity, el despliegue utilizando nodos de Blockchain y la interacción con librerías específicas que permiten la interacción con los recursos de los contratos inteligentes.

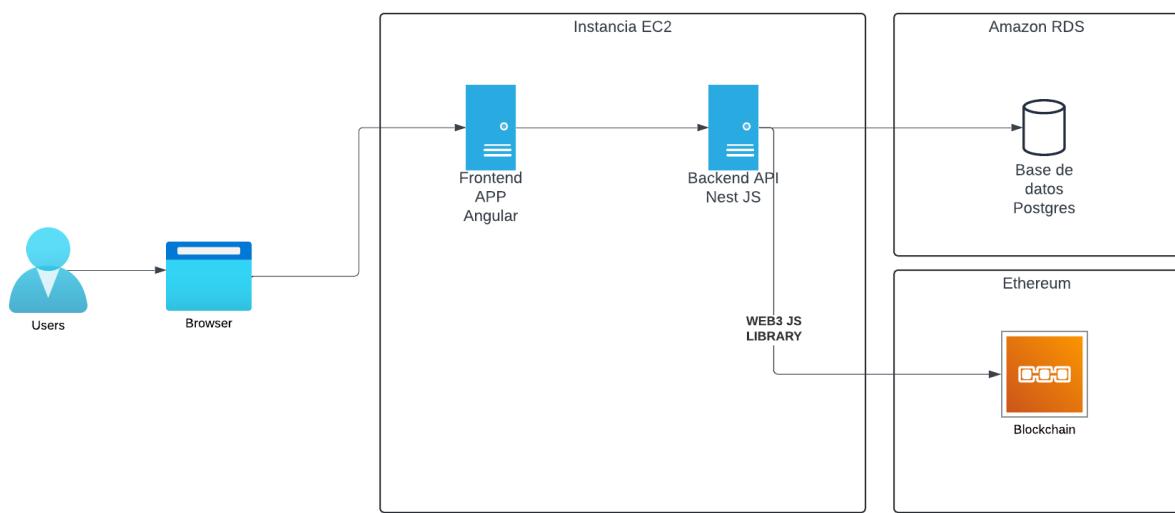
3.2. Arquitectura

Se optó por una arquitectura híbrida de aplicación descentralizada, como se muestra en la figura 5, debido a la falta de herramientas necesarias para desarrollar aplicaciones completamente descentralizadas. Sin embargo, esta arquitectura ofrece varios beneficios que permiten satisfacer las necesidades de la empresa.

En esta configuración, tanto la aplicación frontend como el backend se alojarán en un servidor EC2, mientras que la base de datos será gestionada en Amazon RDS. La información relacionada con los certificados se almacenará en la red blockchain de Ethereum, lo que garantiza un acceso descentralizado a los datos.

Figura 5

Arquitectura descentralizada híbrida



3.3. Requerimientos

3.3.1. Requerimientos funcionales

Los requerimientos funcionales especifican las características que la aplicación debe cumplir. Después de realizar reuniones, encuestas y cuestionarios, se identificaron los siguientes requerimientos funcionales esenciales para el desarrollo de la aplicación. Algunos de ellos se detallan en la Tabla 5 y el resto en el Anexo F, donde se describen las funcionalidades necesarias para este proyecto.

Tabla 5*Requerimientos funcionales*

Requerimiento	Descripción del requerimiento
RF-01	La aplicación debe permitir la autenticación de usuarios mediante JWT (JSON Web Token).
RF-02	La aplicación debe generar un token JWT válido al iniciar sesión con credenciales correctas.
RF-03	La aplicación debe permitir el acceso a las funcionalidades solo a los usuarios autenticados con un token válido.
RF-04	La aplicación debe devolver un mensaje de error cuando las credenciales de acceso sean incorrectas.

3.3.2. Requerimientos no funcionales

Los requerimientos no funcionales establecen las características de rendimiento, seguridad y calidad que la aplicación debe cumplir. En la Tabla 6 se describen algunos de los requerimientos no funcionales, fundamentales para el correcto desarrollo y operación del proyecto, el resto se encuentran en el Anexo F.

Tabla 6*Requerimientos no funcionales*

Requerimiento	Descripción del requerimiento
RNF-01	La autenticación mediante JWT debe ser segura y seguir los estándares actuales de seguridad.
RNF-02	El tiempo de respuesta para la validación de certificados no debe exceder los 30 segundos en condiciones normales de operación.

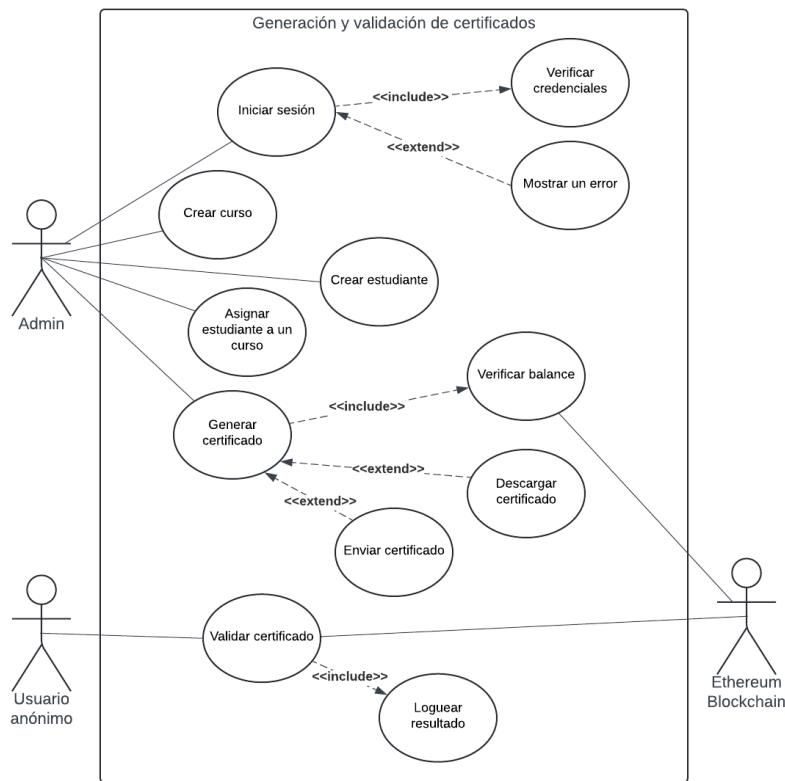
3.4. Diagramas

3.4.1. Caso de Uso

El diagrama de casos de uso, mostrado en la figura 6, ilustra la interacción entre los diferentes actores y sistemas involucrados en el funcionamiento general de la aplicación.

Figura 6

Caso de uso para la generación y validación de certificados



3.4.2. Despliegue

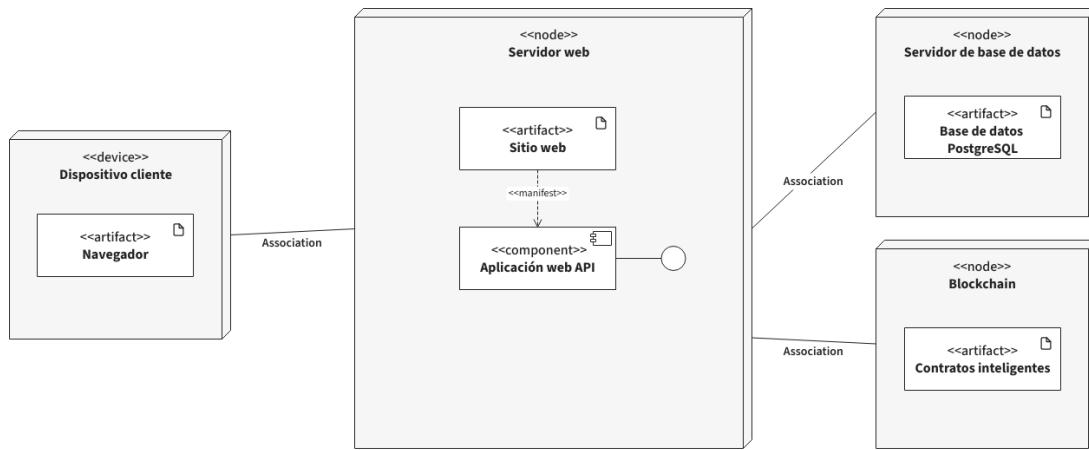
El diagrama de despliegue es una representación visual que muestra la disposición física de los componentes de software y hardware en el sistema, detallando cómo se distribuyen e interconectan para garantizar el correcto funcionamiento de la aplicación. En la figura 7 se observa la disposición de los dispositivos que utilizarán los usuarios, incluidos un navegador web, el

servidor web que alojará tanto la aplicación *frontend* como *backend*, el servidor de base de datos y el nodo *blockchain* empleado para realizar llamadas a los contratos inteligentes.

Figura 7

Diagrama de Despliegue

UML Deployment Diagram



3.4.3. Diagrama de componentes

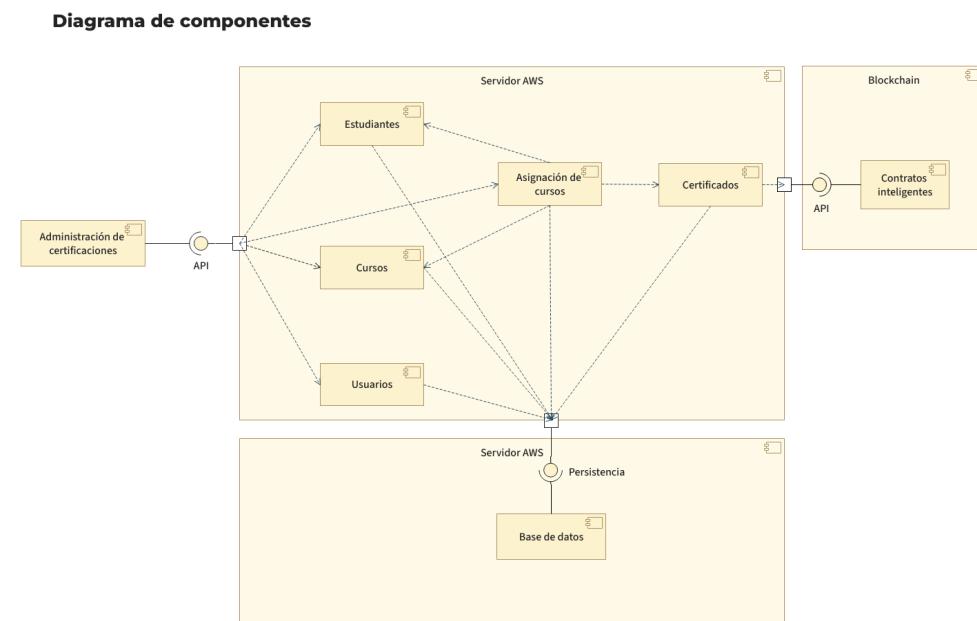
El diagrama de componentes es una representación visual que muestra cómo los distintos componentes de software interactúan dentro de un sistema. Este tipo de diagrama permite visualizar las relaciones y dependencias entre los componentes, así como las interfaces que conectan las diferentes partes del sistema, proporcionando una comprensión clara de la estructura y organización del software.

En la figura 8 se presenta un diagrama de componentes que representa la arquitectura de la aplicación para la administración de certificaciones. Los elementos principales del diagrama son los siguientes:

- **Servidor AWS:** Aloja los componentes de la aplicación, tales como Estudiantes, Cursos, Usuarios, Asignación de cursos y Certificados. Estos componentes interactúan entre sí para gestionar la información de los estudiantes, los cursos impartidos y la emisión de certificados.
- **Base de datos:** Ubicada en un servidor AWS separado, este componente es responsable de la persistencia de los datos, almacenando la información necesaria para el funcionamiento de la aplicación.
- **Blockchain:** Se utiliza para almacenar los certificados en una red descentralizada. Los contratos inteligentes se implementan aquí, garantizando la integridad y seguridad de los certificados emitidos.

Figura 8

Diagrama de componentes

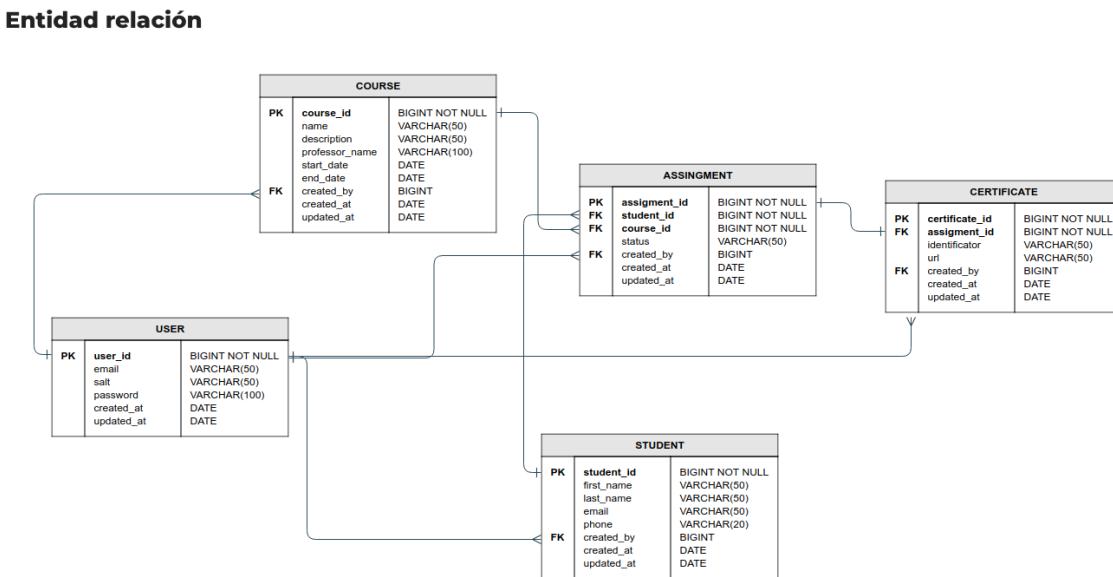


3.4.4. Entidad Relación

El diagrama entidad-relación permite visualizar la estructura de la base de datos. En este proyecto, se utilizarán las entidades *user*, *course*, *student*, *assignment* y *certificate*, las cuales permitirán almacenar la información solicitada por la empresa. La relación entre las distintas entidades se muestra en la figura 9.

Figura 9

Diagrama entidad relación



3.4.5. Diagrama de Secuencia

El diagrama de secuencias se utiliza en el modelado de sistemas para representar la interacción temporal entre objetos o componentes. A continuación, se presentan los diagramas de secuencias que describen las interacciones más importantes de la aplicación.

Figura 10

Diagrama de secuencia para la autenticación JWT

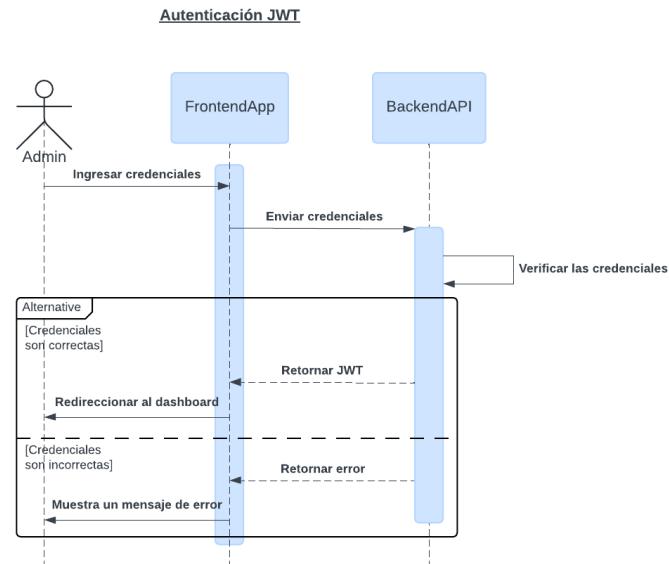


Figura 11

Diagrama de secuencia para la creación de certificado

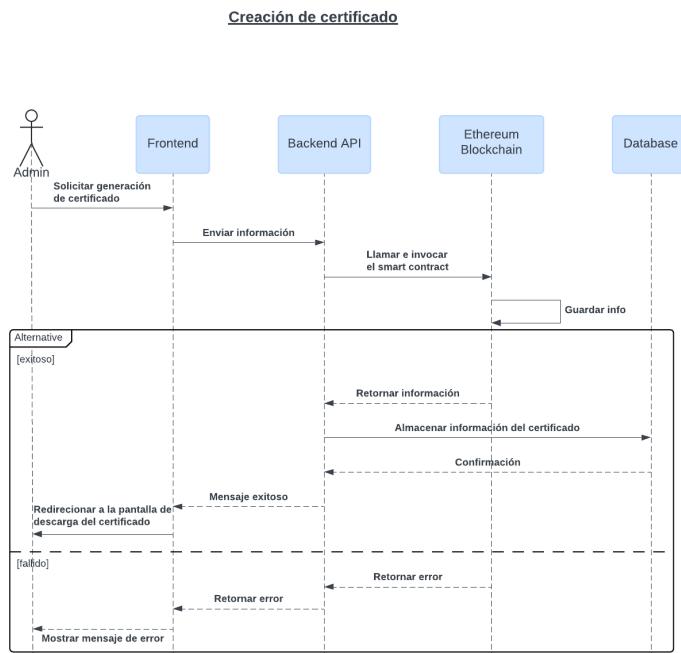
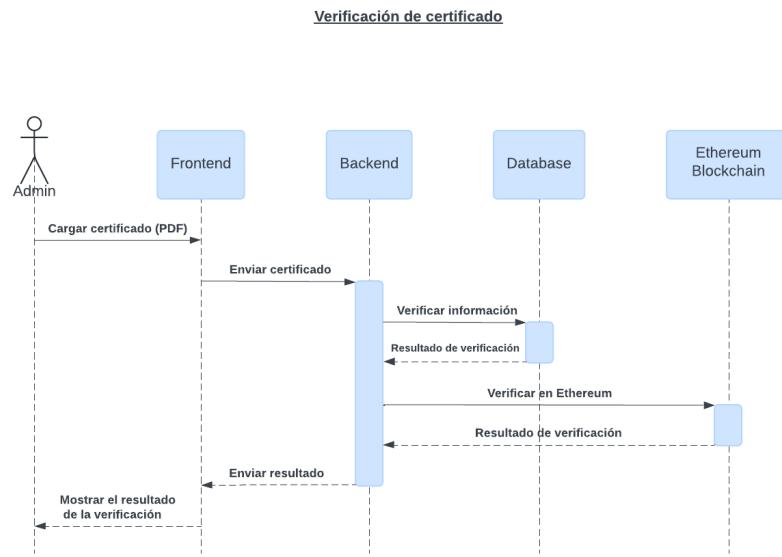


Figura 12

Diagrama de secuencia para la verificación



3.4.6. Diagramas de actividad

El diagrama de actividad es una representación visual que muestra el flujo de trabajo o las actividades dentro de un sistema o proceso. A continuación, se muestra los diagramas de actividad que describen los flujos más importantes de la aplicación.

Figura 13

Diagrama de actividad para la creación de usuario

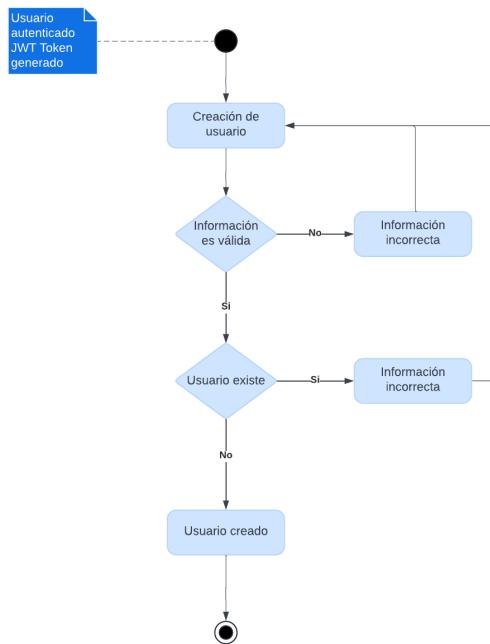


Figura 14

Diagrama de actividad en la generación de certificado

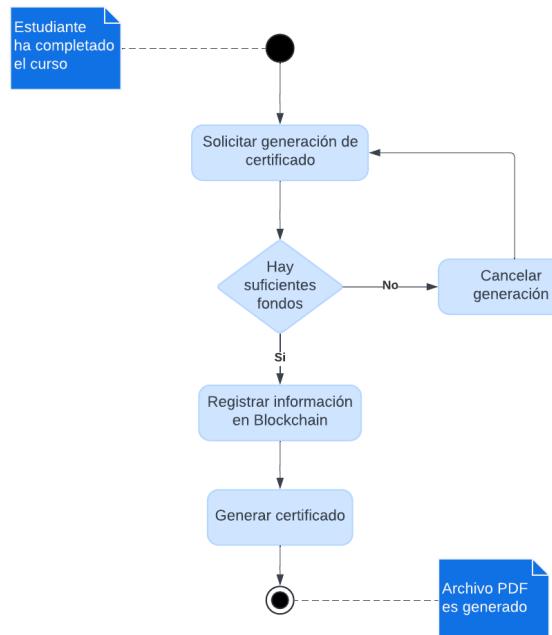
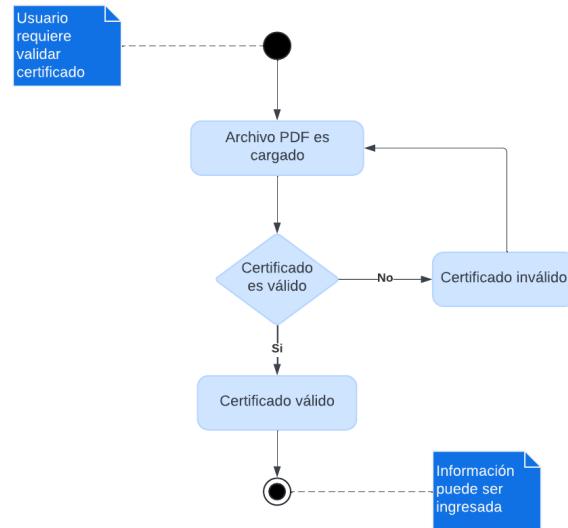


Figura 15

Diagrama de actividad para la validación



3.5. Prototipos

Debido a la funcionalidad y requerimientos solicitados, la aplicación a desarrollar será de tipo web. A continuación, se presentan algunos prototipos de la aplicación.

Figura 16

Pantalla de inicio sesión

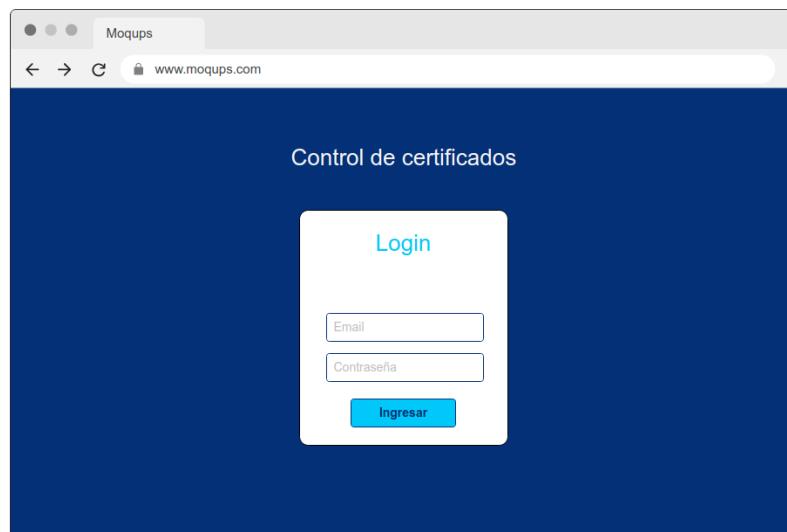


Figura 17

Pantalla de listado de usuarios

A screenshot of a web browser window titled 'Moquups'. The address bar shows 'www.moquups.com'. On the left, there is a dark sidebar with a user profile picture of 'John Doe' and a gear icon. The sidebar menu includes 'Configuración' (with 'Usuarios' selected), 'Administración' (with 'Cursos', 'Estudiantes', and 'Certificaciones' listed), and a right-pointing arrow icon. The main content area has a light gray background and displays the text 'Usuarios' and 'Listado de usuarios'. A blue button labeled 'Nuevo usuario' is located in the top right. Below it is a table with the following data:

Nombre	Apellido	Correo	Rol	Acciones
John	Doe	johndoe@mail.com	Admin	
Alex	Doe	alexdoe@mail.com	Admin	

Figura 18

Pantalla de curso y asignación de estudiantes

The screenshot shows a web interface for managing a course. On the left, a sidebar for 'John Doe' includes sections for Configuration, Users, Administration, Courses, Students, and Certifications. The main content area displays course details: 'Curso: Introducción de Java', Professor: Vicente Suc, Description: 'Curso ideal para principiantes en el cual se iniciaran con las bases del lenguaje Java.', Start Date: 01/01/2024, End Date: 31/12/2024. Below this, a table lists assigned students: Karen Doe and Sofia Doe, each with edit and delete icons. A blue button labeled 'Asignar estudiante' is visible.

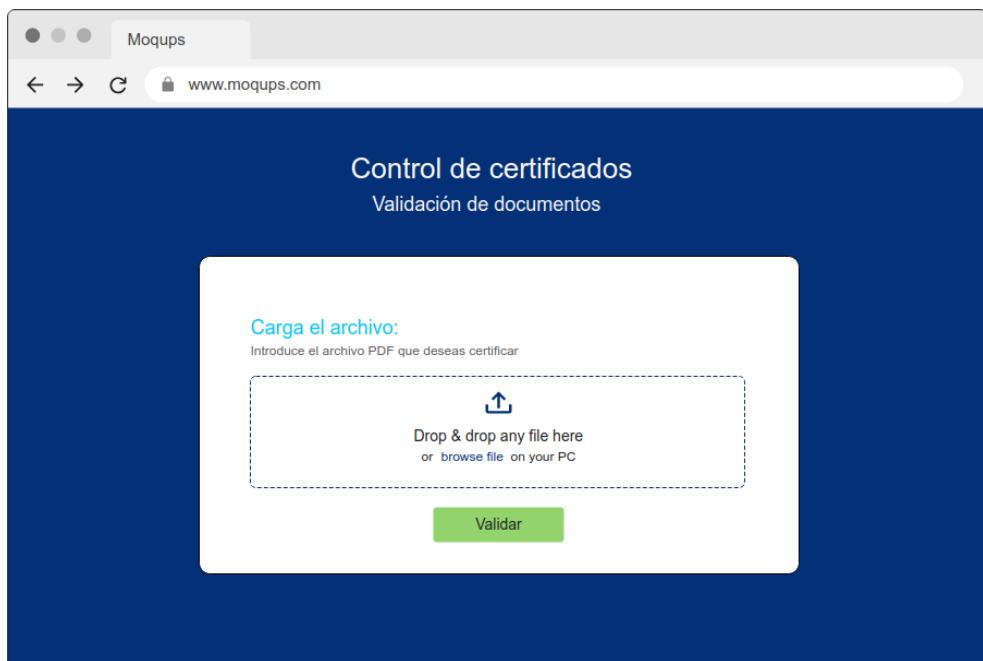
Figura 19

Pantalla de generación de certificado

This screenshot shows the same application interface as Figure 19, but with a modal dialog box in the center asking '¿Estás seguro en generar la certificación?' (Are you sure you want to generate the certificate?). The dialog has 'Cancelar' (Cancel) and 'Certificar' (Generate) buttons. The rest of the page, including the course details and student list, remains visible in the background.

Figura 20

Pantalla de carga de archivo



3.6. SCRUM

3.6.1. Backlog

El backlog define todas las tareas que se van a realizar en el proyecto, debido a que se está utilizando el software Jira para implementar SCRUM, este permite el uso de epics, los cuales agrupan un conjunto de tareas para definir una característica de la aplicación. A continuación, se puede observar en la figura 21 la ventana principal del Backlog.

Figura 21

Backlog en Jira

The screenshot shows the Jira Backlog interface for the project "graduation-project-umg-jc". The left sidebar displays navigation options like "Issues without epic" and a list of epics: "Gestión de Usuarios", "Gestión de Cursos", "Gestión de Estudiantes", "Certificación", and "Validación". The main area shows a backlog of 11 issues under the "Backlog" section. Each issue is listed with its ID, title, and status. The issues are categorized by epic:

Epic	Issue ID	Title	Status	Assignee
GESTIÓN DE USUARIOS	GPUJ-7	Creación de usuarios	TO DO	JC
GESTIÓN DE USUARIOS	GPUJ-8	Edición de usuarios	TO DO	JC
GESTIÓN DE USUARIOS	GPUJ-9	Visualización de usuarios	TO DO	JC
GESTIÓN DE USUARIOS	GPUJ-10	Eliminación de usuarios	TO DO	JC
GESTIÓN DE CURSOS	GPUJ-11	Creación de cursos	TO DO	JC
GESTIÓN DE CURSOS	GPUJ-12	Edición de cursos	TO DO	JC
GESTIÓN DE CURSOS	GPUJ-13	Visualización de cursos	TO DO	JC
GESTIÓN DE CURSOS	GPUJ-14	Eliminación de cursos	TO DO	JC
GESTIÓN DE ESTUDIANTES	GPUJ-15	Creación de estudiantes	TO DO	JC

3.6.2. Sprint Backlog

El sprint Backlog define las tareas que se realizan dentro de un sprint, el proyecto va a tener 4 sprints. A continuación, en la figura 22 se muestra el sprint backlog del presente sprint.

Figura 22

Sprint en Jira

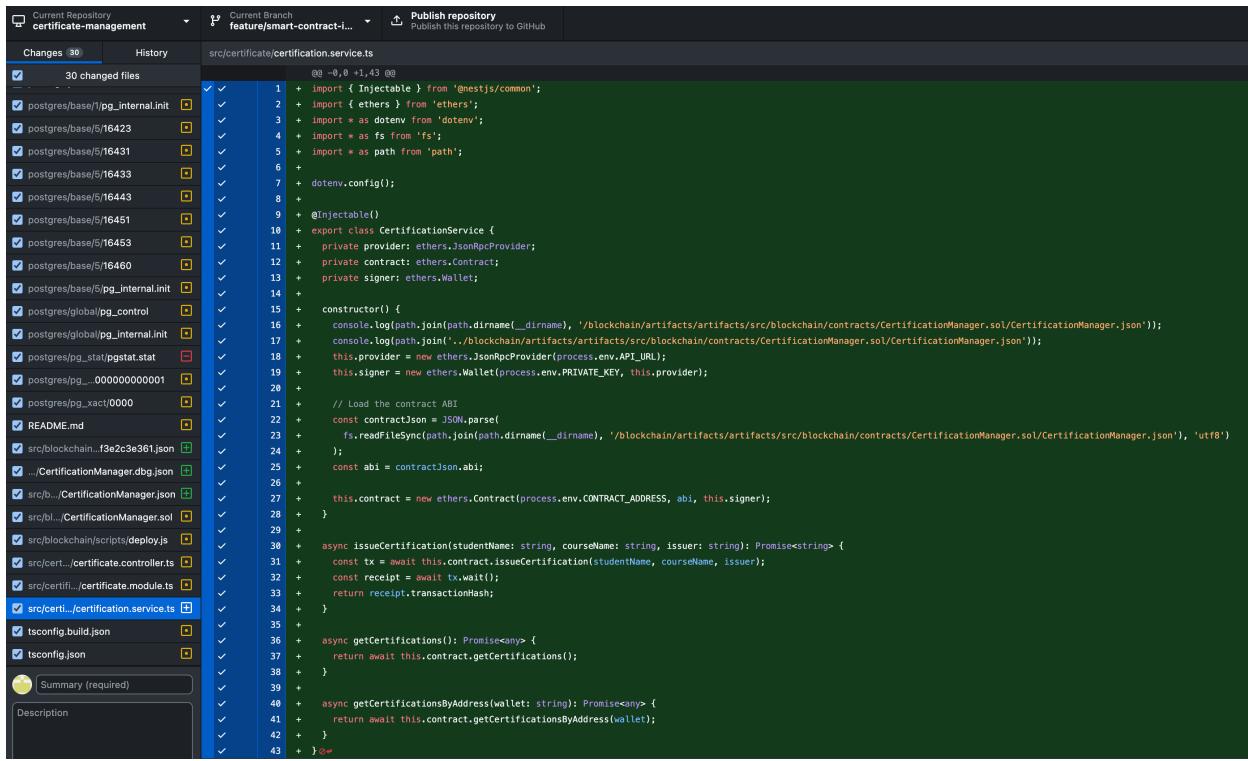
The screenshot shows a Jira project interface for 'certification-manager'. On the left, a sidebar lists project navigation options like Planning, Backlog, and Board. The 'Board' option is selected and highlighted in blue. The main area is titled 'CM Sprint 3' and displays a Kanban-style board with three columns: 'TO DO 2', 'IN PROGRESS 2', and 'DONE 3'. Each column has a green checkmark icon at the top right. The 'TO DO' column contains two tasks: 'Copiar el archivo ABI al Backend' and 'Crear servicio para integrarse con smart contract', both associated with the 'SMART CONTRACT' epic. The 'IN PROGRESS' column contains two tasks: 'Desplegar el smart contract' and 'Crear cuenta en Metamask', both associated with the 'SMART CONTRACT' epic. The 'DONE' column contains three tasks: 'Definir los atributos del smart contract', 'Crear smart contract', and 'Crear proyecto en Alchemy', all associated with the 'SMART CONTRACT' epic. Each task card includes a checkbox indicating its status and a user icon.

3.6.3. Incrementos

Los incrementos es el producto final de un sprint, para el manejo de los incrementos se utiliza Git, esta nos permite llevar control de los cambios que se realizan durante el proyecto. A continuación en la figura 23 se muestra la ventana de Github Desktop que permite visualizar los cambios existentes en el proyecto.

Figura 23

Código fuente desde Github Desktop



The screenshot shows a GitHub Desktop interface with a dark theme. On the left, a sidebar lists 30 changed files, including PostgreSQL and Ethereum-related files like `pg_internal.init`, `pgstat.stat`, and `certification.service.ts`. The main pane displays a code diff for `src/certificate/certification.service.ts`. The code uses `nestjs-common`, `ethers`, `dotenv`, `fs`, and `path` modules. It defines a `CertificationService` class with methods for issuing certifications and getting them by address. The code includes comments explaining the loading of the contract ABI and the use of `JSON.parse` and `fs.readFileSync`.

```
00 -0.0 +1.43 00
1 + import { Injectable } from '@nestjs/common';
2 + import { ethers } from 'ethers';
3 + import * as dotenv from 'dotenv';
4 + import * as fs from 'fs';
5 + import * as path from 'path';
6 +
7 + dotenv.config();
8 +
9 + @Injectable()
10 + export class CertificationService {
11 +   private provider: ethers.JsonRpcProvider;
12 +   private contract: ethers.Contract;
13 +   private signer: ethers.Wallet;
14 +
15 +   constructor() {
16 +     console.log(path.join(path.dirname(__dirname), '/blockchain/artifacts/artifacts/src/blockchain/contracts/CertificationManager.sol/CertificationManager.json'));
17 +     console.log(path.join('../blockchain/artifacts/artifacts/src/blockchain/contracts/CertificationManager.sol/CertificationManager.json'));
18 +     this.provider = new ethers.JsonRpcProvider(process.env.API_URL);
19 +     this.signer = new ethers.Wallet(process.env.PRIVATE_KEY, this.provider);
20 +
21 +     // Load the contract ABI
22 +     const contractJson = JSON.parse(
23 +       fs.readFileSync(path.join(path.dirname(__dirname), '/blockchain/artifacts/artifacts/src/blockchain/contracts/CertificationManager.sol/CertificationManager.json'), 'utf8')
24 +     );
25 +     const abi = contractJson.abi;
26 +
27 +     this.contract = new ethers.Contract(process.env.CONTRACT_ADDRESS, abi, this.signer);
28 +   }
29 +
30 +   async issueCertification(studentName: string, courseName: string, issuer: string): Promise<string> {
31 +     const tx = await this.contract.issueCertification(studentName, courseName, issuer);
32 +     const receipt = await tx.wait();
33 +     return receipt.transactionHash;
34 +   }
35 +
36 +   async getCertifications(): Promise<any> {
37 +     return await this.contract.getCertifications();
38 +   }
39 +
40 +   async getCertificationsByAddress(wallet: string): Promise<any> {
41 +     return await this.contract.getCertificationsByAddress(wallet);
42 +   }
43 + }
```

4. Capítulo 4 – Desarrollo de software

4.1. Patrones de diseño

4.1.1. *Modelo-Vista-Controlador*

El modelo-vista-controlador es un patrón de diseño que divide las distintas partes de una aplicación en una capa de interfaz de usuario, una de datos y otra de lógica del negocio. Esto permite una separación de responsabilidades, facilitando el desarrollo del software. Estas tres capas son el modelo, el cual se encarga de los datos que se van a utilizar dentro la aplicación, la vista, que se encarga de mostrar la información al usuario final, cabe mencionar que en el caso del backend, es un REST API y finalmente el controlador, el cual se encarga de toda la lógica del negocio, utiliza los datos, los procesa y finalmente los muestra en una vista.

Debido al uso de NestJS, el cuál utiliza como base ExpressJS, un Framework minimalista que se basa en el patrón MVC y permite la creación de aplicaciones web de manera sencilla del lado del backend. El proyecto sigue las prácticas definidas por estos frameworks y se implementa este patrón de diseño para facilitar el desarrollo.

4.1.2. *Modelo-Vista-ViewModel (MVVM)*

El patrón de arquitectura MVVM es utilizado en la aplicación Frontend, la cual utiliza el Framework Angular para la implementación. Este patrón facilita la separación de las capas en la creación de interfaces de usuario. De igual manera que el patrón MVC, esté tiene tres componentes, la vista es la encargada de definir la estructura para la presentación de la interfaz, el modelo es el encargado de la información y el vista modelo es el encargado de controlar la interacción que el usuario realiza sobre la interfaz a través de métodos y sobre la conexión de los datos con la vista, ya que se encarga de actualizar el estado de la vista cada vez que los datos son actualizados.

Aunque los patrones MVC y MVVM comparten similitudes, estos difieren en que el MVVM está enfocado para aplicaciones que tiene una mayor interacción del usuario con interfaces, como es el caso de aplicaciones web y móviles, mientras el patrón MVC está más enfocado para aplicaciones web.

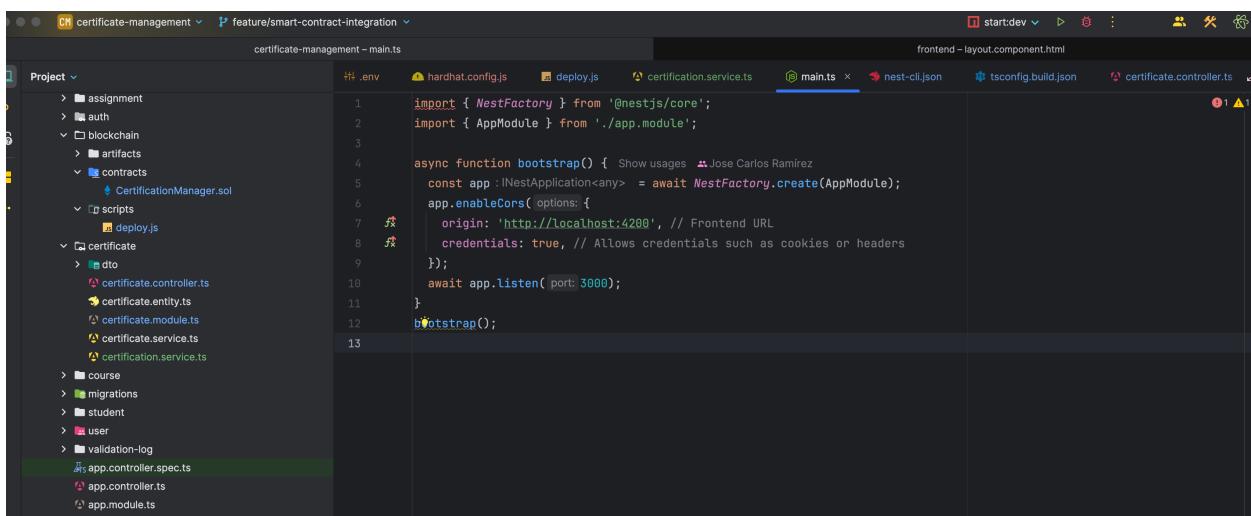
4.2. Herramientas de desarrollo

4.2.1. *WebStorm*

Entorno de desarrollo integrado desarrollado por JetBrains que facilita el desarrollo y ejecución de aplicaciones web, en la figura 24 se puede observar la pantalla de WebStorm. Este IDE está enfocado especialmente para aplicaciones web que trabajan con Javascript y NodeJS. Facilita la ejecución de las distintas aplicaciones web, tanto Frontend como Backend, permite el autocompletado de código y permite la integración con distintos plugins para ampliar su capacidad.

Figura 24

Pantalla de WebStorm



The screenshot shows the WebStorm IDE interface. At the top, there's a toolbar with icons for start:dev, run, stop, and others. Below the toolbar, the title bar shows 'certificate-management' and 'feature/smart-contract-integration'. The main area has two tabs: 'certificate-management - main.ts' (active) and 'frontend - layout.component.html'. On the left, the 'Project' sidebar shows a file tree with various folders like assignment, auth, blockchain, artifacts, contracts, scripts, certificate, dto, migrations, student, user, validation-log, and some smart-contract-related files. In the center, the code editor displays the 'main.ts' file:

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';

async function bootstrap() {
  Show usages ↗ Jose Carlos Ramírez
  const app :NestApplication<any> = await NestFactory.create(AppModule);
  app.enableCors( options: {
    origin: 'http://localhost:4200', // Frontend URL
    credentials: true, // Allows credentials such as cookies or headers
  });
  await app.listen( port: 3000 );
}
bootstrap();
```

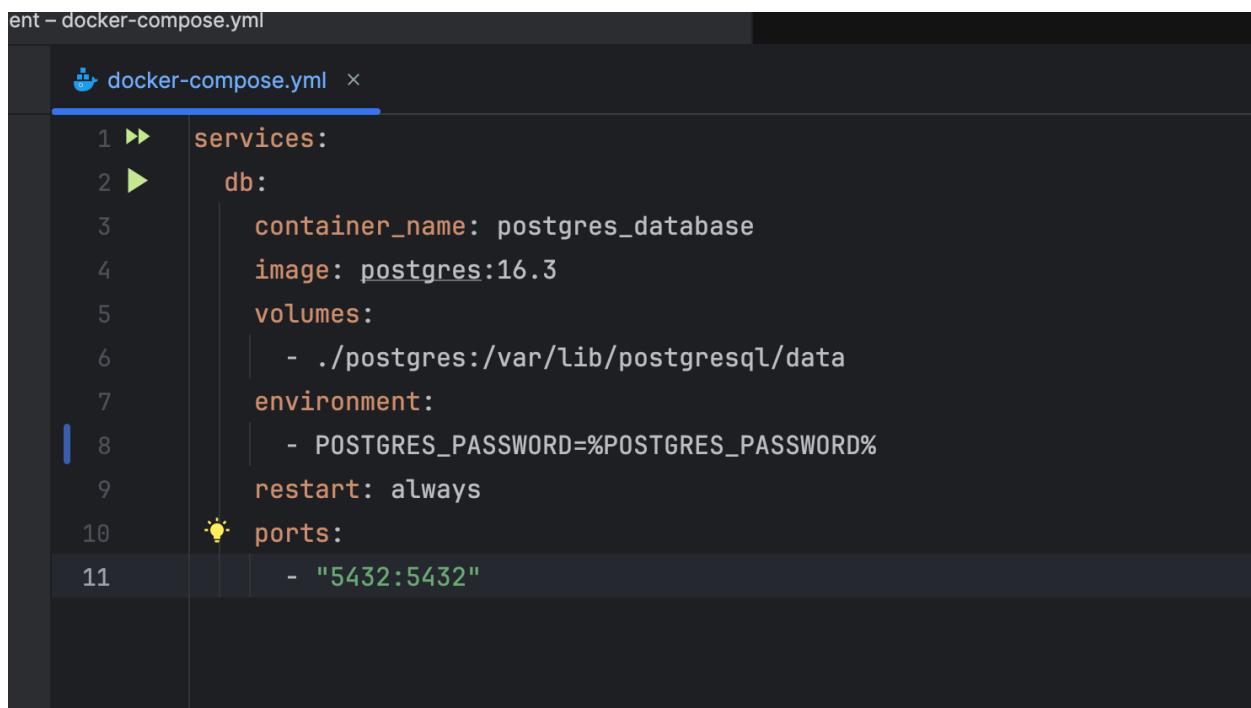
4.2.2. Docker

Es un software que permite crear, probar, desplegar y compartir contenedores. Los contenedores son componentes ejecutables y estandarizados que permiten la combinación del código fuente con partes de sistemas operativos, librerías y dependencias necesarias para ejecutar el código. Esto facilita la distribución y ejecución de aplicaciones o software sin la necesidad de crear máquinas virtuales.

En el proyecto se utiliza Docker para la base de datos PostgreSQL, ya que la empresa utiliza está mismo software para su base de datos de producción y de esa manera facilitar la integración al momento del despliegue. Para ejecutar la base de datos dentro de Docker se creó un archivo docker-compose.yml dentro de la aplicación Backend como se muestra en la figura 25.

Figura 25

Archivo docker compose del proyecto



```
ent - docker-compose.yml
  docker-compose.yml x
  1 ► services:
  2 ►   db:
  3     container_name: postgres_database
  4     image: postgres:16.3
  5     volumes:
  6       - ./postgres:/var/lib/postgresql/data
  7     environment:
  8       - POSTGRES_PASSWORD=%POSTGRES_PASSWORD%
  9     restart: always
 10    ports:
 11      - "5432:5432"
```

Posteriormente se ejecuta el comando `docker compose up -d` como se muestra en la figura 26, el cual indica al servicio de Docker levantar los servicios indicados en el archivo de manera detach, es decir, sin mostrar los logs en la terminal.

Figura 26

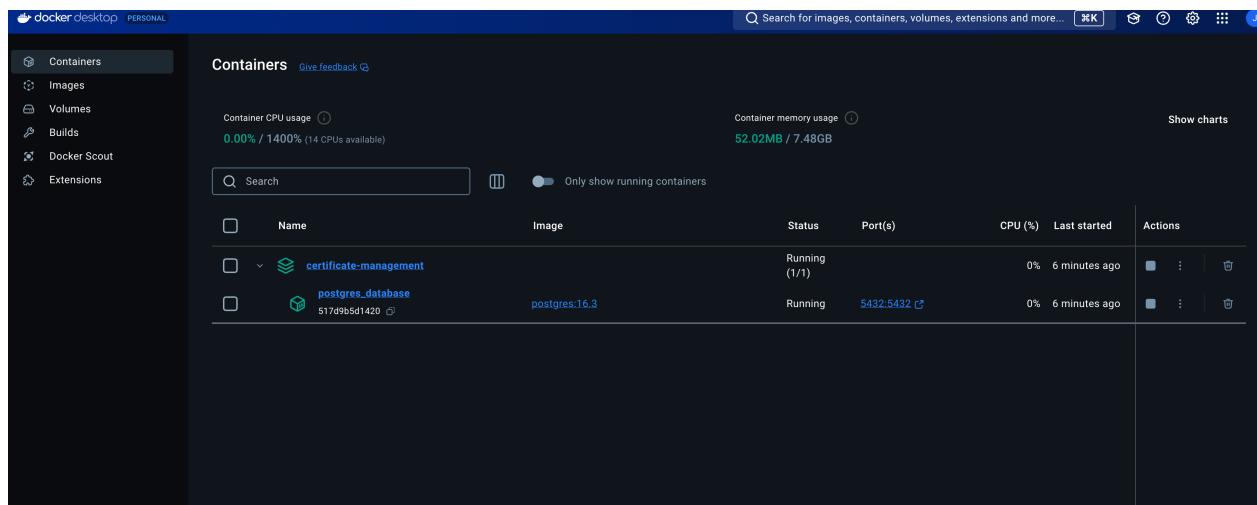
Ejecución del docker compose

```
apple ~ de/university/certificate-management > ⌘ P feature/smarter-integration +1 !18 ?2 docker compose up -d
[+] Running 2/2
  ✓ Network certificate-management_default  Created
  ✓ Container postgres_database              Started
```

Una vez ejecutado el comando, se puede visualizar dentro de la aplicación de Docker como se puede observar en la figura 27 los contenedores que se están ejecutando.

Figura 27

Ventana del Docker Desktop



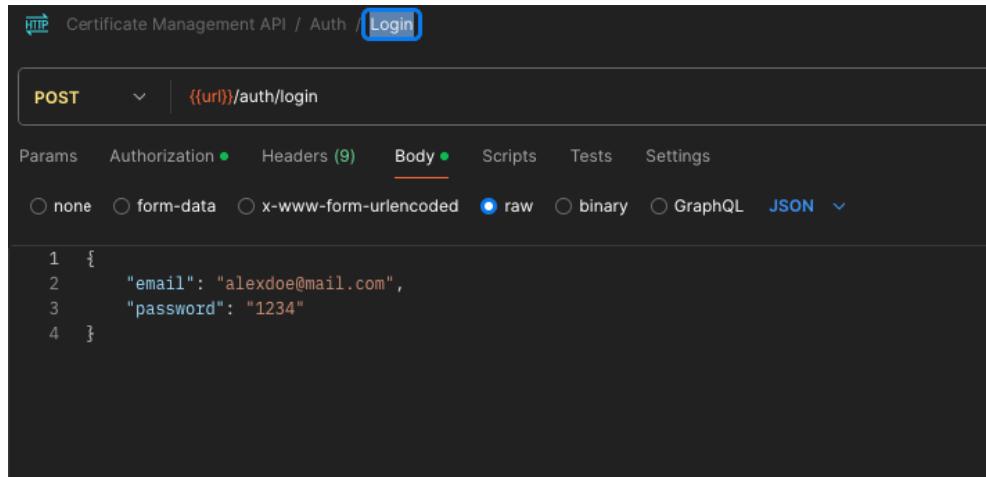
4.2.3. Postman

Plataforma que permite la construcción y uso de APIs. Este software se utiliza para las pruebas de los servicios REST en el Backend. Como se puede visualizar en la figura 28, el software

facilita las consultas y la visualización tanto de las peticiones como las respuestas, facilitando la construcción de los servicios REST.

Figura 28

Ejemplo de petición en Postman

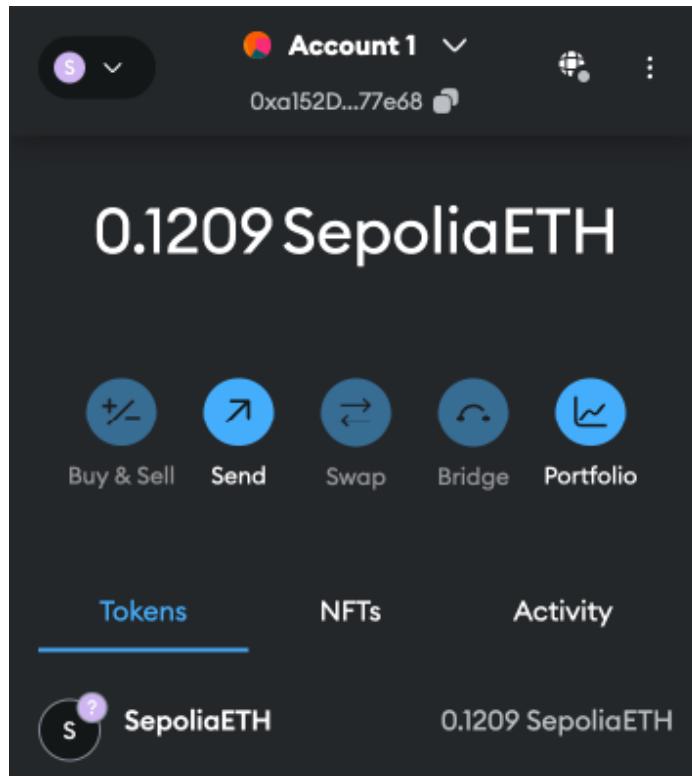


4.2.4. Metamask

Es una billetera de criptomonedas utilizada para interactuar con la blockchain de Ethereum. Debido a la manera en que funciona las redes blockchain, es necesario la creación de una cuenta para poder interactuar con la red y así poder desplegar los contratos inteligentes. Metamask facilita la creación de cuentas y se utilizó para la creación de una cuenta cómo se puede ver en la figura 29, que posteriormente fue utilizada para poder desplegar los contratos inteligentes.

Figura 29

Cuenta de ejemplo en Metamask



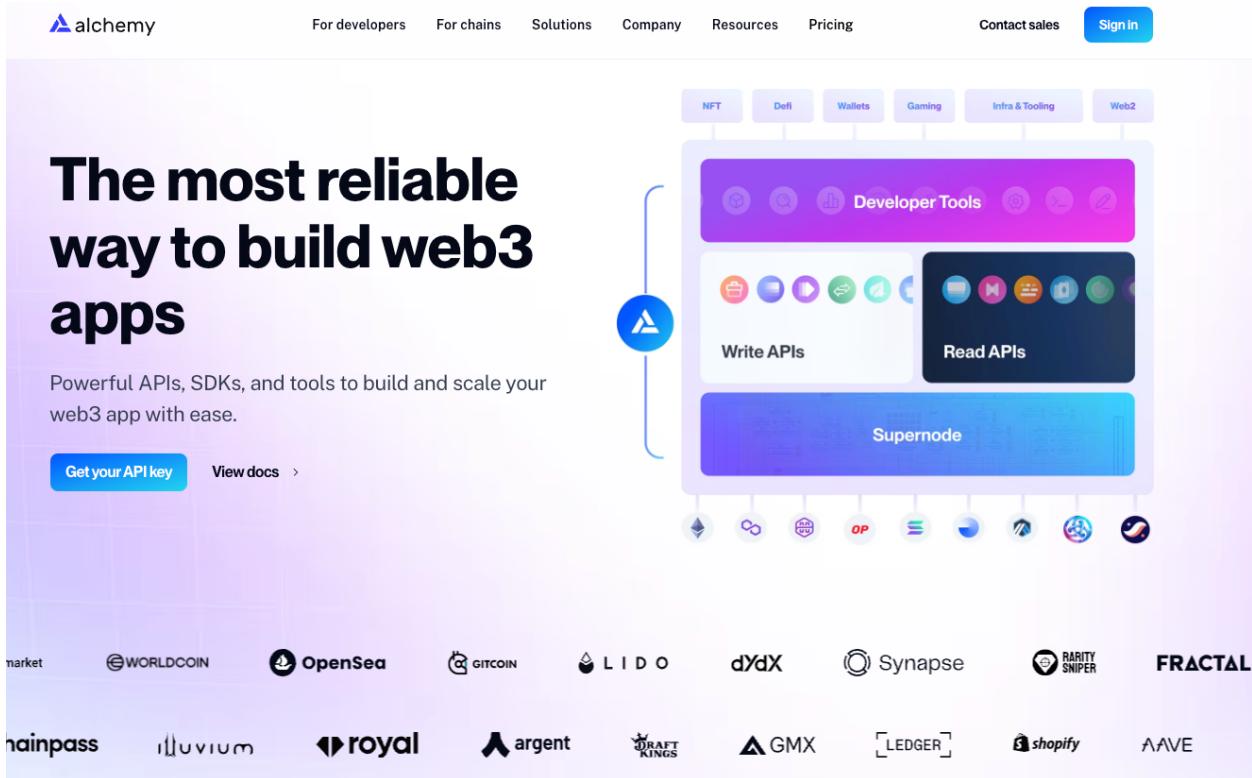
4.2.5. Alchemy

Plataforma que facilita el desarrollo de aplicaciones web3. Alchemy permite la integración con diferentes blockchain a través de sus nodos, brinda servicios REST que facilitan la interacción con los contratos inteligentes desplegados.

Esta plataforma es utilizada para el despliegue de los contratos inteligentes, debido a que es necesario un nodo de Ethereum para poder comunicarse con la red, Alchemy proporciona un nodo que es utilizado para la creación del smart contract. En la figura 30 se puede observar la página principal de la plataforma.

Figura 30

Pantalla principal del sitio web de Alchemy



4.3. Módulos desarrollados

4.3.1. Backend

4.3.1.1. Creación de contrato inteligente

El contrato inteligente se desarrolló en la aplicación Backend, este fue desarrollado utilizando el lenguaje de programación Solidity. El contrato inteligente de la figura 31 cuenta con una estructura que es utilizada para la creación de un mapa de certificaciones, el cual almacenará la información de todas las certificaciones generadas por este Smart contract. Se tiene el método issueCertification el cual permite generar una certificación para el estudiante, getCertification para obtener una certificación y getCertificationsByAddress para ver las certificaciones generadas.

Figura 31

Código fuente en Solidity del contrato inteligente

```
event CertificationIssued(address indexed student, string course, string issuer, uint256 date);

// Issue a new certification for a student
function issueCertification(
    string memory _studentName,
    string memory _courseName,
    string memory _issuer
) public {
    Certification memory newCert = Certification({
        studentName: _studentName,
        courseName: _courseName,
        issuer: _issuer,
        date: block.timestamp,
        studentAddress: msg.sender
    });

    certifications[msg.sender].push(newCert);
    emit CertificationIssued(msg.sender, _courseName, _issuer, block.timestamp);
}

// Retrieve all certifications for a student
function getCertifications() public view returns (Certification[] memory) {
    return certifications[msg.sender];
}

// Retrieve certifications for a specific student address (only by student)
function getCertificationsByAddress(address _student) public view returns (Certification[] memory) {
    require(_student == msg.sender, "Access restricted to certification owner");
    return certifications[_student];
}
```

4.3.1.2. Despliegue de contrato inteligente

El despliegue del contrato inteligente a la Blockchain de Ethereum se realizó utilizando la herramienta Hardhat y la librería Etherjs. Estás utilizan las claves privadas generadas en Metamask y en Alchemy que son necesarias al momento del despliegue. Cabe mencionar que la cuenta utilizada para desplegar el Smart contract necesita tener la suficiente cantidad de Ether para poder pagar el costo de la transacción.

Figura 32

Función en Solidity que despliega el contrato inteligente en Ethereum

```
4  async function main () :Promise<void> { Show usages new *
5    const CertificationManager = await ethers.getContractFactory('CertificationManager');
6    console.log("Deploying contract...");
7    const certificationManagerInstance = await CertificationManager.deploy();
8    console.log("Contract was deployed to address: " + certificationManagerInstance.target);
9    process.env['CONTRACT_ADDRESS'] = certificationManagerInstance.target;
10   }
11 }
12
13 main().then(() :never => process.exit( code: 0)) new *
14 .catch(error => {
15   console.log(error);
16   process.exit( code: 1);
17 });
```

4.3.1.3. Módulo de usuarios

Este módulo es el encargado la administración de los usuarios a través de una REST API.

Este permite la creación, actualización, eliminación y lectura de los usuarios registrados en la base de datos. En la siguiente figura 33 se puede visualizar la entidad utilizada para la creación de los usuarios.

Figura 33

Entidad de usuarios

```
@Entity() Show usages ✎ Jose Carlos Ramírez
export class User {
    @PrimaryGeneratedColumn( strategy: 'uuid' )
    id: string;

    @Column( options: { unique: true } )
    email: string;

    @Column( options: { unique: true } )
    username: string;

    @Column()
    salt: string;

    @Column()
    password: string;

    @Column( options: { type: 'timestamp', default: () => 'CURRENT_TIMESTAMP' } )
    created_at: Date;

    @Column( options: { type: 'timestamp', default: () => 'CURRENT_TIMESTAMP' } )
    updated_at: Date;

    @ManyToOne( typeFunctionOrTarget: () => User, inverseSide: (user :User ) => user.id )
    created_by: User;

    @ManyToOne( typeFunctionOrTarget: () => User, inverseSide: (user :User ) => user.id )
    updated_by: User;
}
```

4.3.1.4. Módulo de autenticación

Modulo encargado de la autenticación de los usuarios, la generación del JWT tokens y la posterior validación del token para cada recurso donde es obligatorio. Para la creación de este módulo se utilizó la librería Passport que facilita la autenticación utilizando distintos estándares, en el proyecto se hace uso JWT (Json Web Token) para poder autenticar y validar un usuario. En la figura 34 se puede observar la implementación de la estrategia de autenticación.

Figura 34

Estrategia de autenticación

```
5  @Injectable() Show usages ↗ Jose Carlos Ramírez
6  export class JwtStrategy extends PassportStrategy(Strategy) {
7      constructor() { no usages ↗ Jose Carlos Ramírez
8          console.log('JwtStrategy');
9          super({
10             jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
11             ignoreExpiration: false,
12             secretOrKey: process.env.JWT_SECRET || 'secretKey',
13         });
14     }
15
16     async validate(payload: any) { no usages ↗ Jose Carlos Ramírez
17         console.log('validate');
18         console.log(payload);
19         return { userId: payload.sub, email: payload.email };
20     }
21 }
```

La siguiente figura 35 muestra la autenticación desde la aplicación Postman

Figura 35

Prueba de autenticación en Postman

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** {{url}}/auth/login
- Body:** Raw JSON (selected)
- Body Content:**

```
1 {
2     "email": "alexdoe@mail.com",
3     "password": "1234"
4 }
```
- Response Status:** 201 Created
- Response Headers:** 206 ms, 877 B
- Response Body (Pretty JSON):**

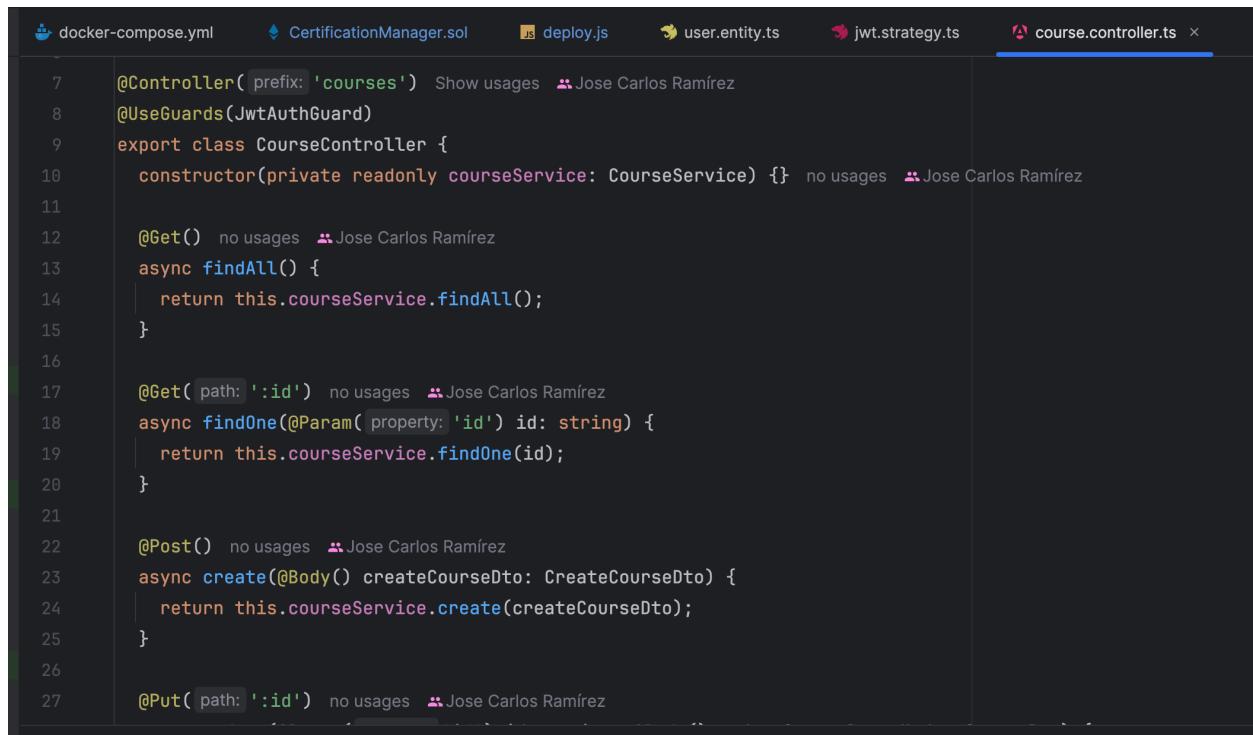
```
1 {
2     "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmtZSToImFsZXhkb2UiLCJzdWlI0iI4OGJkZTRjMiiLYTi2LTrhNjQtOGM3OS1hZDlkOTZkZTQ2YjQjLCJpYXQiOjE3MjY3MzEzNzIsImV4cCI6MTcyNjczNDk3Mn0.1d3-Htd6Whsm2lpMs-SZq-pEM6-R3Xbz_8hPgPi1iwi",
3     "user": {
4         "id": 1,
5         "name": "Alex Doe",
6         "email": "alexdoe@mail.com",
7         "password": "$2b$10$uPm6R3Xbz_8hPgPi1iwi",
8         "role": "user"
9     }
10 }
```

4.3.1.5. Módulo de cursos

Módulo encargado del control de los cursos a través de una REST API. Este permite la creación, actualización, eliminación y lectura de los cursos registrados en la base de datos. En la figura 36 se puede observar el controlador para el módulo de cursos.

Figura 36

Controlador de cursos

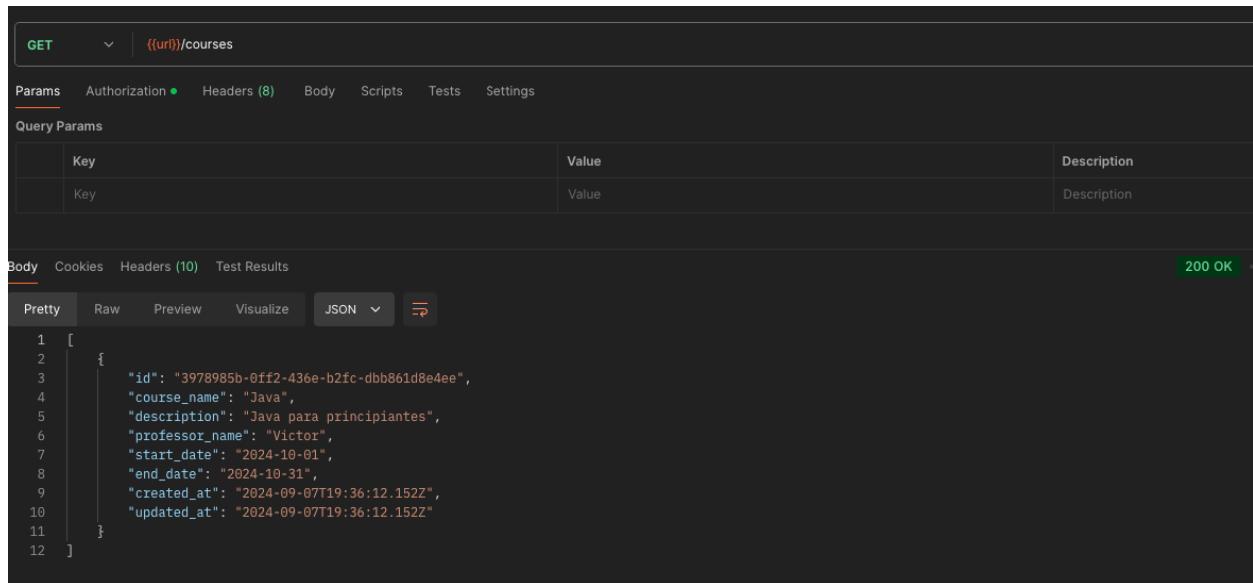


```
  docker-compose.yml  CertificationManager.sol  deploy.js  user.entity.ts  jwt.strategy.ts  course.controller.ts
1  @Controller('prefix: "courses"') Show usages  Jose Carlos Ramírez
2  @UseGuards(JwtAuthGuard)
3  export class CourseController {
4      constructor(private readonly courseService: CourseService) {}  no usages  Jose Carlos Ramírez
5
6      @Get()  no usages  Jose Carlos Ramírez
7      async findAll() {
8          return this.courseService.findAll();
9      }
10
11      @Get('path: ":id"')  no usages  Jose Carlos Ramírez
12      async findOne(@Param('property: "id"') id: string) {
13          return this.courseService.findOne(id);
14      }
15
16      @Post()  no usages  Jose Carlos Ramírez
17      async create(@Body() createCourseDto: CreateCourseDto) {
18          return this.courseService.create(createCourseDto);
19      }
20
21      @Put('path: ":id"')  no usages  Jose Carlos Ramírez
22  
```

La figura 37 muestra el listado de cursos que retorna una consulta a la REST API.

Figura 37

Consulta al recurso de cursos



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** {{url}}/courses
- Params:** (highlighted in red)
- Headers:** (8)
- Body:** (highlighted in red)
- Tests:**
- Settings:**

Query Params:

Key	Value	Description
Key	Value	Description

Body: (highlighted in red)

200 OK

Pretty Raw Preview Visualize JSON

```
1 [  
2 {  
3   "id": "3978985b-0ff2-436e-b2fc-dbb861d8e4ee",  
4   "course_name": "Java",  
5   "description": "Java para principiantes",  
6   "professor_name": "Victor",  
7   "start_date": "2024-10-01",  
8   "end_date": "2024-10-31",  
9   "created_at": "2024-09-07T19:36:12.152Z",  
10  "updated_at": "2024-09-07T19:36:12.152Z"  
11 }]  
12 ]
```

4.3.1.6. Módulo de estudiantes

Este módulo es el encargado de la administración de los estudiantes. Este permite la crear, editar, eliminar y leer estudiantes dentro de la aplicación. De igual manera que los anteriores módulos, este está compuesto por entidad, controlador, servicio, y un módulo. A continuación, en la figura 38 se muestra el servicio del presente módulo.

Figura 38

Servicio de estudiante

```
@Injectable() Show usages ↗ Jose Carlos Ramírez
export class StudentService {
  constructor( no usages ↗ Jose Carlos Ramírez
    @InjectRepository(Student)
    private readonly studentRepository: Repository<Student>,
  ) {}

  async findAll(): Promise<Student[]> { Show usages ↗ Jose Carlos Ramírez
    return this.studentRepository.find();
  }

  async findOne(id: string): Promise<Student> { Show usages ↗ Jose Carlos Ramírez
    return this.studentRepository.findOne( options: { where: { id } } );
  }

  async create(createStudentDto: CreateStudentDto): Promise<Student> { Show usages ↗ Jose Carlos Ramírez
    const newStudent :Student = this.studentRepository.create(createStudentDto);
    return this.studentRepository.save(newStudent);
  }

  async update(id: string, updateStudentDto: UpdateStudentDto): Promise<void> { Show usages ↗ Jose Carlos Ramírez
    await this.studentRepository.update(id, updateStudentDto);
  }

  async remove(id: string): Promise<void> { Show usages ↗ Jose Carlos Ramírez
    await this.studentRepository.delete(id);
  }
}
```

La figura 39 muestra la creación de un estudiante desde la REST API.

Figura 39

Creación de estudiante desde Postman

The screenshot shows the Postman application interface. At the top, there is a 'POST' method dropdown and a URL field containing '{{url}}/students'. Below the URL, there are tabs for 'Params', 'Authorization', 'Headers (11)', 'Body', 'Scripts', 'Tests', and 'Settings'. The 'Body' tab is currently active, indicated by a green dot. Under the 'Body' tab, there are several input options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (which is selected), 'binary', 'GraphQL', and 'JSON'. The 'JSON' option has a dropdown arrow next to it. Below these options, there is a code editor area with line numbers (1 to 6) and JSON data:

```
1 {  
2   "first_name": "John",  
3   "last_name": "Doe",  
4   "email": "johndoe@mail.com",  
5   "phone": "55550000"  
6 }
```

Below the code editor, there are tabs for 'Body', 'Cookies', 'Headers (10)', and 'Test Results'. The 'Body' tab is selected, indicated by a green dot. Under the 'Body' tab, there are three output formats: 'Pretty' (selected), 'Raw', and 'Visualize'. To the right of these, there is a 'JSON' dropdown with a dropdown arrow. Below these, there is another code editor area with line numbers (1 to 9) and JSON data, including additional fields like 'id', 'created_at', and 'updated_at':

```
1 {  
2   "first_name": "John",  
3   "last_name": "Doe",  
4   "email": "johndoe@mail.com",  
5   "phone": "55550000",  
6   "id": "97e4f36e-a91c-4a20-9678-f6ae99e78db5",  
7   "created_at": "2024-09-19T13:48:04.306Z",  
8   "updated_at": "2024-09-19T13:48:04.306Z"  
9 }
```

4.3.1.7. Módulo de asignación de estudiantes

Este módulo se encarga de asignar un estudiante a un curso existente. El módulo también permite desasignar un estudiante para un curso siempre que esté no tenga una certificación. A continuación, se muestra en la figura 40 el módulo de asignación de estudiantes.

Figura 40

Módulo en NestJS

```
1 import { Module } from '@nestjs/common';
2 import { TypeOrmModule } from '@nestjs/typeorm';
3 import { Assignment } from './assignment.entity';
4 import { AssignmentService } from './assignment.service';
5 import { AssignmentController } from './assignment.controller';
6 import { Student } from '../student/student.entity';
7 import { Course } from '../course/course.entity';
8
9 @Module({
10   metadata: {
11     imports: [TypeOrmModule.forFeature([Assignment, Student, Course])],
12     providers: [AssignmentService],
13     controllers: [AssignmentController],
14     exports: [AssignmentService],
15   }
16 })
17 export class AssignmentModule {}
```

La figura 41 muestra la lectura de una asignación de un estudiante a un curso desde la REST API.

Figura 41

Consulta de asignación en Postman

The screenshot shows the Postman application interface. At the top, there is a header bar with a 'GET' button and a URL placeholder {{url}}/assignments/5e758cbd-716a-43c6-acce-57204a0c9430. Below the header, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Scripts', 'Tests', and 'Settings'. The 'Authorization' tab is selected, showing 'Auth Type' set to 'Bearer Token'. A note on the right says 'Heads up! These parameters hold variables.' In the main body area, there are tabs for 'Body', 'Cookies', 'Headers (10)', and 'Test Results'. The 'Body' tab is selected, showing a JSON response with line numbers from 1 to 25. The JSON data represents an assignment object with student and course details.

```
1 {  
2   "id": "5e758cbd-716a-43c6-acce-57204a0c9430",  
3   "status": "ACTIVE",  
4   "created_at": "2024-09-07T19:40:37.696Z",  
5   "updated_at": "2024-09-07T19:40:37.696Z",  
6   "student": {  
7     "id": "2e79af1c-e0e3-486a-845f-e1a435ba2b15",  
8     "first_name": "Jose",  
9     "last_name": "Ramirez",  
10    "email": "jose@mail.com",  
11    "phone": "55555555",  
12    "created_at": "2024-09-07T19:38:08.223Z",  
13    "updated_at": "2024-09-07T19:38:08.223Z"  
14  },  
15  "course": {  
16    "id": "3978985b-0ff2-436e-b2fc-dbb861d8e4ee",  
17    "course_name": "Java",  
18    "description": "Java para principiantes",  
19    "professor_name": "Victor",  
20    "start_date": "2024-10-01",  
21    "end_date": "2024-10-31",  
22    "created_at": "2024-09-07T19:36:12.152Z",  
23    "updated_at": "2024-09-07T19:36:12.152Z"  
24  }  
25 }
```

4.3.1.8. Módulo de certificados

Este módulo almacena los certificados generados, una asignación puede tener solamente un certificado, es decir, un estudiante asignado a un curso puede tener un certificado. El certificado almacena el identificador de la transacción en la Blockchain y la url de la transacción. En la figura 42 se puede observar la creación de un certificado.

Figura 42

Servicio de certificados

```
3  @Injectable() Show usages new *
4  export class CertificationService {
5    private provider: ethers.JsonRpcProvider;
6    private contract: ethers.Contract;
7    private signer: ethers.Wallet;
8
9    constructor() { no usages new *
10      console.log(path.join(path.dirname(__dirname), '/blockchain/artifacts/artifacts/src/blockchain/contracts/CertificationManager.sol/CertificationManager.json'));
11      console.log(path.join('../blockchain/artifacts/artifacts/src/blockchain/contracts/CertificationManager.sol/CertificationManager.json'));
12      this.provider = new ethers.JsonRpcProvider(process.env.API_URL);
13      this.signer = new ethers.Wallet(process.env.PRIVATE_KEY, this.provider);
14
15      // Load the contract ABI
16      const contractJson = JSON.parse(
17        fs.readFileSync(path.join(path.dirname(__dirname), '/blockchain/artifacts/artifacts/src/blockchain/contracts/CertificationManager.sol/CertificationManager.json'))
18      );
19      const abi = contractJson.abi;
20
21      this.contract = new ethers.Contract(process.env.CONTRACT_ADDRESS, abi, this.signer);
22    }
23
24    async issueCertification(studentName: string, courseName: string, issuer: string): Promise<string> { no usages new *
25      const tx = await this.contract.issueCertification(studentName, courseName, issuer);
26      const receipt = await tx.wait();
27      return receipt.transactionHash;
28    }
29  }
```

La figura 43 se muestra la creación de un certificado desde la API REST.

Figura 43

Creación de certificado desde Postman

The screenshot shows the Postman application interface. At the top, there is a 'POST' method dropdown and a URL field containing '{{url}}/certificates'. Below the method, there are tabs for 'Params', 'Authorization', 'Headers (11)', 'Body', 'Scripts', 'Tests', and 'Settings'. The 'Body' tab is currently active, indicated by a blue underline. Under the 'Body' tab, there are several radio button options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (which is selected), 'binary', 'GraphQL', and 'JSON'. The 'JSON' option is also underlined in blue. The main area shows the JSON payload for creating a certificate:

```
1 {
2     "status": "ACTIVE",
3     "certificate": {
4         "id": "5e758cbd-716a-43c6-acce-57204a0c9430"
5     },
6     "identifier": "0xeeca006ac252233ec84333683dd774b517196a701cf333fe8d14780327e2dd7d7",
7     "url": "https://sepolia.etherscan.io/tx/0xeeca006ac252233ec84333683dd774b517196a701cf333fe8d14780327e2dd7d7"
8 }
```

Below the JSON input, there are tabs for 'Body', 'Cookies', 'Headers (10)', and 'Test Results'. The 'Body' tab is selected. Underneath these tabs, there are three buttons: 'Pretty', 'Raw', and 'Preview'. The 'Pretty' button is highlighted with a dark grey background. To the right of these buttons is a 'JSON' dropdown menu with a downward arrow icon. Further to the right is a copy icon (a clipboard with a double-headed arrow). The 'Raw' tab is also visible. The JSON output is identical to the input:

```
1 {
2     "identifier": "0xeeca006ac252233ec84333683dd774b517196a701cf333fe8d14780327e2dd7d7",
3     "url": "https://sepolia.etherscan.io/tx/0xeeca006ac252233ec84333683dd774b517196a701cf333fe8d14780327e2dd7d7",
4     "assignment": {
5         "id": "5e758cbd-716a-43c6-acce-57204a0c9430",
6         "status": "ACTIVE",
7         "created_at": "2024-09-07T19:40:37.696Z",
8         "updated_at": "2024-09-07T19:40:37.696Z"
9     },
10    "id": "b1b531da-dadd-4c83-99f8-76f1ea9bc553",
11    "created_at": "2024-09-19T16:47:57.814Z",
12    "updated_at": "2024-09-19T16:47:57.814Z"
13 }
```

4.3.1.9. Módulo de log de validación

Módulo encargado de almacenar un registro de las validaciones realizadas, en este se puede obtener información de la cantidad de certificados validados y el feedback de los usuarios para indicar si un certificado es válido o no y llevar un registro que posteriormente puede ser utilizado por la empresa para futuras mejores o implementaciones. A continuación, se muestran en la figura 44 los DTO (Data Transfer Object) utilizados para este módulo

Figura 44

Data Transfer Object para registros de validación

The image shows two side-by-side code editors. The left editor contains the code for `CreateValidationLogDto.ts`, which defines a class with properties `certificate_id` (string), `status` (string), and annotations `@IsUUID()`, `@IsNotEmpty()`, and `@IsString()`. The right editor contains the code for `UpdateValidationLogDto.ts`, which defines a class with a single optional property `status?` (string) and annotations `@IsString()` and `@IsOptional()`.

```
create-validation-log.dto.ts
1 import { IsNotEmpty, IsString, IsUUID } from 'class-validator';
2
3 export class CreateValidationLogDto {
4     @IsUUID()
5     @IsNotEmpty()
6     certificate_id: string;
7
8     @IsString()
9     @IsNotEmpty()
10    status: string;
11 }
```

```
update-validation-log.dto.ts
1 import { IsOptional, IsString } from 'class-validator';
2
3 export class UpdateValidationLogDto {
4     @IsString()
5     @IsOptional()
6     status?: string;
7 }
```

La figura 45 muestra el listado de logs al hacer una consulta al API REST.

Figura 45

Consulta el listado de registros de validación

The image shows a screenshot of the Postman application interface. It's a GET request to `{{url}}/validation-logs`. The Authorization tab is set to "Bearer Token". The Body tab is selected and displays a JSON response with 38 items, each representing a validation log entry. The response is pretty-printed and includes fields like `id`, `status`, `created_at`, and a nested `certificate` object with its own properties such as `id`, `identifier`, `url`, and timestamps.

```
[{"id": "a2ae2bed-608f-46e1-9706-63eaccfb7b37", "status": "VALID", "created_at": "2024-09-19T16:57:45.776Z", "certificate": {"id": "3339b1aa-8ba2-4f0c-b776-f43a99ca2cae", "identifier": "GU4A328943284", "url": "http://www.test.com", "created_at": "2024-09-07T19:43:11.576Z", "updated_at": "2024-09-07T19:43:11.576Z"}, {"id": "3b91bcdf-f74b-4125-b04a-8b92af7f58ee", "status": "VALID", "created_at": "2024-09-19T16:57:50.512Z", "certificate": {"id": "3339b1aa-8ba2-4f0c-b776-f43a99ca2cae", "identifier": "GU4A328943284", "url": "http://www.test.com", "created_at": "2024-09-07T19:43:11.576Z", "updated_at": "2024-09-07T19:43:11.576Z"}, {"id": "8a0dc2cb-34c7-42e1-92bd-0d4f242babbb", "status": "INVALID", "created_at": "2024-09-19T16:57:58.020Z", "certificate": {"id": "3339b1aa-8ba2-4f0c-b776-f43a99ca2cae", "identifier": "GU4A328943284", "url": "http://www.test.com", "created_at": "2024-09-07T19:43:11.576Z", "updated_at": "2024-09-07T19:43:11.576Z"}]
```

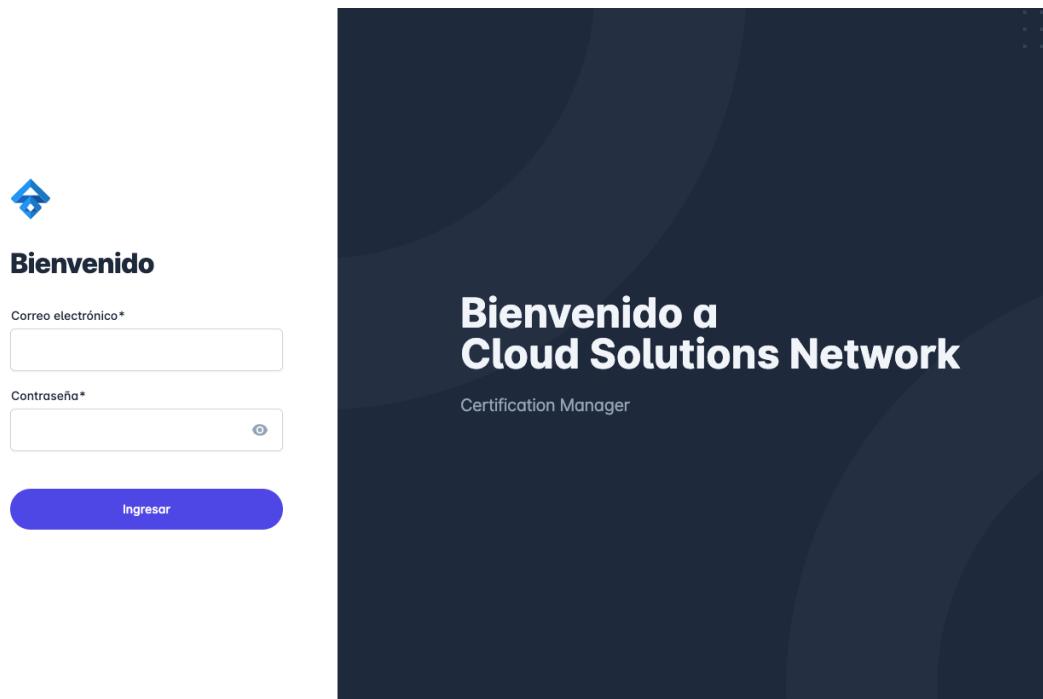
4.3.2. Frontend

4.3.2.1. Módulo de autenticación

Este se encarga de la vista que utiliza el usuario para acceder a la aplicación y la lógica utilizada en Angular para conectarse con la REST API en NestJS y validar la información retornada. A continuación, en la figura 46 se muestra la pantalla de login.

Figura 46

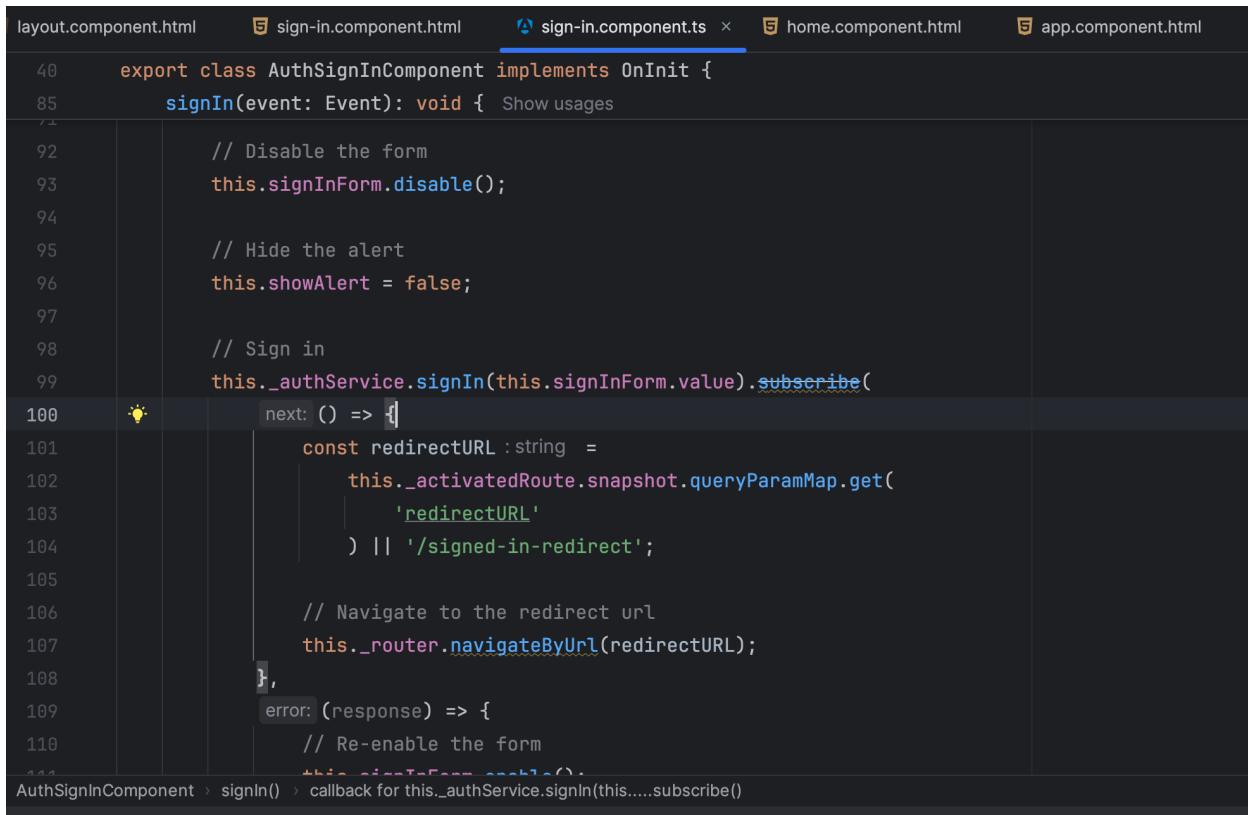
Pantalla de Login



La figura 47 muestra parte del código utilizado en el componente de autenticación.

Figura 47

Componente del inicio de sesión



```
layout.component.html      sign-in.component.html      sign-in.component.ts      home.component.html      app.component.html
40  export class AuthSignInComponent implements OnInit {
85    signIn(event: Event): void { Show usages
71
92      // Disable the form
93      this.signInForm.disable();
94
95      // Hide the alert
96      this.showAlert = false;
97
98      // Sign in
99      this._authService.signIn(this.signInForm.value).subscribe(
100        next: () => {
101          const redirectURL :string =
102            this._activatedRoute.snapshot.queryParamMap.get(
103              'redirectURL'
104            ) || '/signed-in-redirect';
105
106          // Navigate to the redirect url
107          this._router.navigateByUrl(redirectURL);
108        },
109        error: (response) => {
110          // Re-enable the form
111          this.signInForm.enable();
112        }
113      );
}
AuthSignInComponent > signIn() > callback for this._authService.signIn(this....subscribe()
```

4.3.2.2. Módulo de usuarios

Módulo encargado de la interfaz para la interacción con los servicios REST, permite la administración de los usuarios, en la figura 48 se puede observar la pantalla de listado de usuarios.

Figura 48

Listado de usuarios

The screenshot shows a user interface titled "Listado de usuarios". On the left, there is a sidebar with the user's name "alexdoe" and email "alexdoe@mail.com". The sidebar also contains links for "Usuarios", "Estudiantes", "Cursos", and "Certificados", along with the FUSE logo. The main content area is titled "Listado de usuarios" and contains a table with the following data:

No.	Usuario	Correo	Opciones
1	alexdoe	alexdoe@mail.com	
2	admin	alexdoe2@mail.com	

Below the table, there is a pagination control with "Items per page: 5" and "0 of 0 | < > >>|". A blue button at the top right says "+ Nuevo estudiante".

4.3.2.3. Modulo de validación de certificado

Módulo encargado en validar que un certificado es correcto o no, en esté módulo se utiliza el archivo PDF del certificado que fue generado anteriormente y se carga sobre el campo de archivo que se muestra en la pantalla que se puede visualizar en la figura 49.

Figura 49

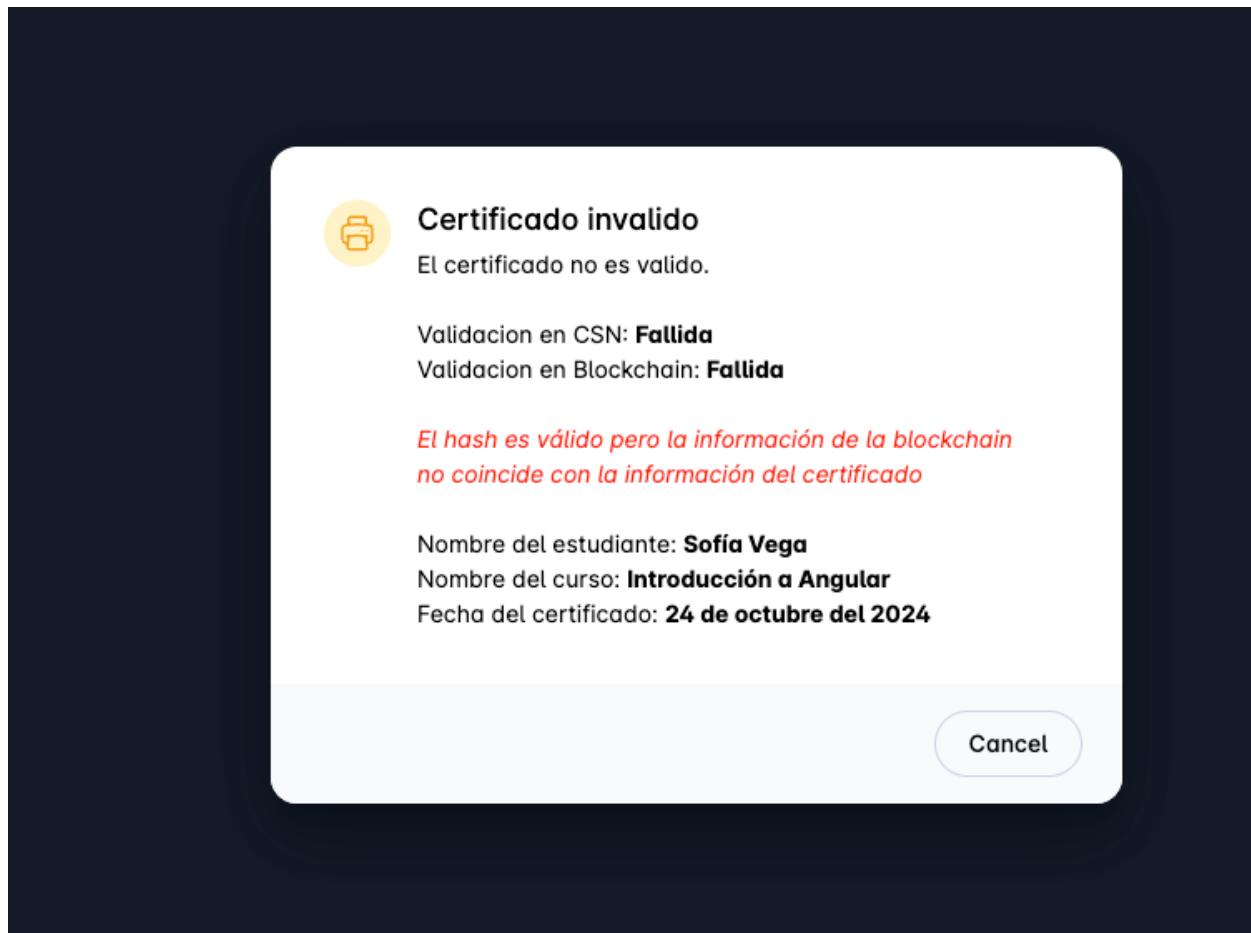
Página para validar un certificado

The screenshot shows a web page titled "CSN Certificate Validator" with a dark background. The title is centered at the top. Below it, the text "Sube tu certificado" (Upload your certificate) is displayed. There is a file input field with the placeholder "Selecciona un archivo:" (Select a file:) and a "Choose File" button. Below the file input is a blue button labeled "Cargar certificado" (Load certificate). The overall design is minimalist and modern.

Una vez cargado el certificado realiza la verificación, se está verificando el hash del código QR. En la figura 50 se muestra un certificado que tiene un hash válido pero sus valores no coinciden con los almacenados en la Blockchain.

Figura 50

Certificado inválido



En la figura 51 se muestra la información del PDF.

Figura 51

Ejemplo de certificado



Para comprobar que información fue almacenada dentro de la Blockchain se puede verificar el sitio de etherscan, el cual permite verificar todos los bloques y transacciones realizadas. Al momento de utilizar un lector QR nos lleva a la página en etherscan de la transacción realizada para este certificado, la página se muestra en la figura 52.

Figura 52

Página principal de la transacción

The screenshot shows the Etherscan interface for a Sepolia Testnet transaction. At the top, there's a search bar and navigation links for Home, Blockchain, Tokens, NFTs, and More. The main section is titled "Transaction Details" with tabs for Overview (selected), Logs (1), and State. A note says "[This is a Sepolia Testnet transaction only]". The transaction details are listed as follows:

- Transaction Hash: 0x2d19d9505b5bad2fafc6749c441d0f234d090752b25b2d35bc当地6245ec1a3a7c
- Status: Success
- Block: 6940805 | 144042 Block Confirmations
- Timestamp: 21 days ago (Oct-25-2024 05:58:00 AM UTC)
- Transaction Action: Call | 0xa152Df40...792377e68 on 0xd6D29b5E...418dBC5b1
- From: 0xa152Df401d2Be1620E9469cE2044A2D792377e68
- To: 0xd6D29b5E05Cb83B837d70F9f9AEf3C0418dBC5b1
- Value: 0 ETH
- Transaction Fee: 0.000030498654088151 ETH
- Gas Price: 0.157851541 Gwei (0.000000000157851541 ETH)

At the bottom, it says "Gas Limit & Usage by Txn: 195,912 | 193,211 (98.62%)".

Para ver más detalles de la trasacción se debe hacer click sobre la opción *Click to show more* como se muestra en la figura 53.

Figura 53

Sección para más detalles

This screenshot shows the expanded details section of the transaction from Figure 52. It includes the same transaction details as before, plus a "More Details" button at the bottom left which is highlighted in red. To its right is the text "+ Click to show more".

Esto muestra más detalle de la transacción donde se puede visualizar la información almacenada como se muestra en la figura 54.

Figura 54

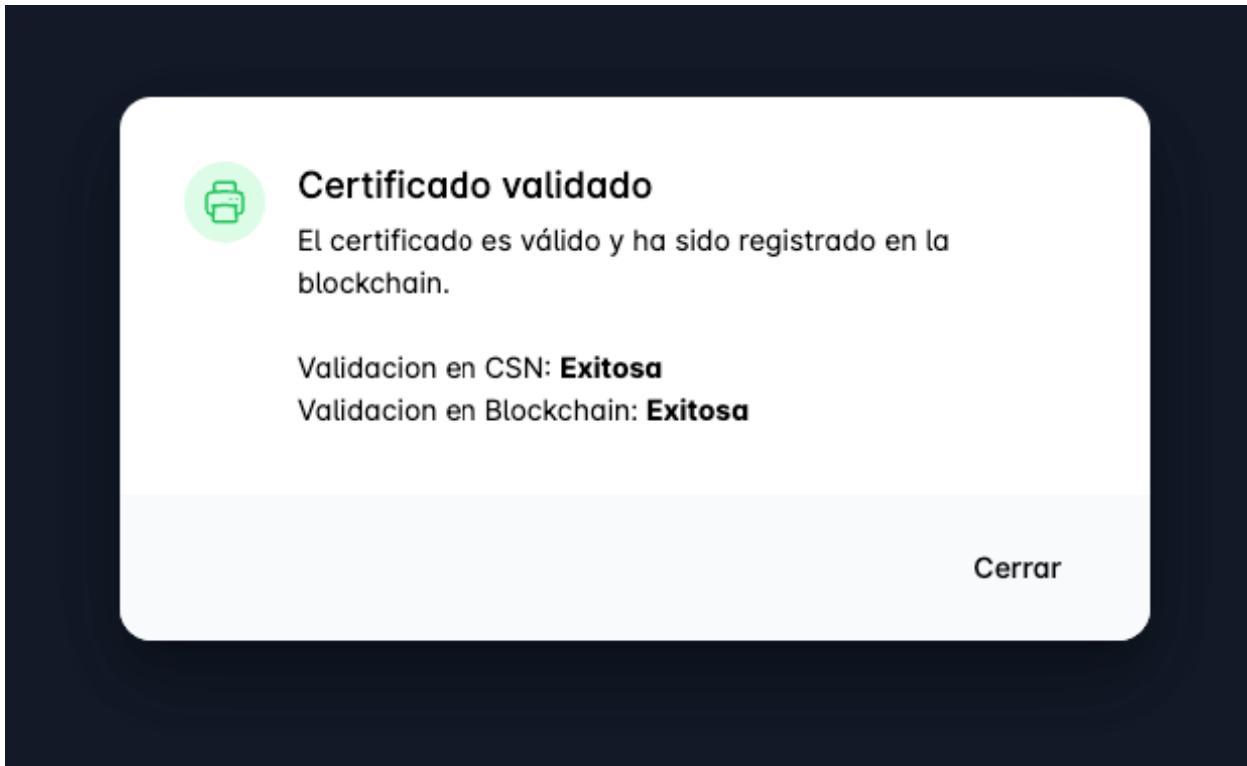
Información adicional del certificado dentro de la Blockchain

The screenshot shows a detailed view of a blockchain transaction. At the top, it displays gas limit and usage: 195,912 | 193,211 (98.62%). Below that are gas fees: Base: 0.074690136 Gwei | Max: 0.237976697 Gwei | Max Priority: 0.083161405 Gwei. It also shows burnt and savings fees: Burnt: 0.000014430955866696 ETH (\$0.00) and Txn Savings: 0.000015481061515916 ETH (\$0.00). Under 'Other Attributes', it says Txn Type: 2 (EIP-1559),Nonce: 28, and Position In Block: 93. The 'Input Data' section contains a large hex string: ^" + @ €g 3d\$f880943a-4abc-4a46-8901-d290d6b813d5 SofÃa Vega IntroducciÃ³n a AngularB0x5b138e4376bb170dd0d26070f80f6930751c0a9b95c575f1070f4d4fad865029. A 'View Input As' button is below this. At the bottom, there are 'More Details:' and a 'Click to show less' link.

Si el certificado es válido, muestra un mensaje indicando la validación exitosa como se muestra en la figura 55.

Figura 55

Mensaje de certificado exitoso



4.4. Pruebas de ejecución

Las pruebas de ejecución permitieron observar las distintas partes del proyecto ejecutándose correctamente. En éstas se pudo observar el contrato inteligente desplegado dentro de la red de Ethereum, el Frontend ejecutándose en el navegador, el Backend permitiendo el acceso a los distintos recursos REST y la base de datos permitiendo conexiones desde el Backend.

4.4.1. Prueba de despliegue de contrato inteligente

El contrato inteligente se desplegó en la red Ethereum utilizando un nodo proporcionado por Alchemy y una cuenta creada por Metamask. Después de ejecutar la archivo deploy.js el cual tiene el código para comunicarse al nodo de Ethereum. La figura 49 muestra el resultado de la ejecución del archivo de despliegue.

Figura 56

Despliegue de smart contract desde la terminal

```
Apple > ~/development/poc/blockchain/blockchain-certificates npx hardhat run scripts/deploy.js --network sepolia
Certificate contract deployed to: 0x43887F3dAA9E5A3e2479b0b109E730B4839875D8
```

Etherscan es una aplicación web que facilita el acceso a la información de la red Ethereum, este permite visualizar transacciones, bloques, cuentas, más información acerca de estos componentes de manera rápida y sencilla. La transacción generada por el despliegue del contrato inteligente como se muestra en la figura, es utilizada dentro de Etherscan para comprobar su despliegue y ver información más detallada del mismo. La figura 50 muestra la información del contrato inteligente.

Figura 57

Smart contract desde Etherscan

The screenshot shows the Etherscan interface for a specific smart contract. At the top, there's a navigation bar with links to Home, Blockchain, Tokens, NFTs, and More. The main content area has three tabs: Overview, More Info, and Multichain Info. The Overview tab shows an ETH BALANCE of 0 ETH. The More Info tab shows the CONTRACT CREATOR as 0xa152Df40...792377e68, created at txn 0xf81004c99be... The Multichain Info section says N/A. Below these are tabs for Transactions, Token Transfers (ERC-20), Contract, and Events. The Transactions tab is active, showing a single transaction: 0xf81004c99be... (Method: 0x60806040, Block: 6720878, Age: 53 secs ago, From: 0xa152Df40...792377e68, To: Contract Creation (0xf81004c99be...), Amount: 0 ETH, Txn Fee: 0.04033117). There's also a note explaining what a smart contract is and a link to the Knowledge Base.

En la figura 51 se puede observar la información de la transacción donde el contrato inteligente fue creado

Figura 58

Transacción en Etherscan

The screenshot shows the Etherscan transaction details page for transaction 0xf81004c99bef44cafe0fd58312050e861b4ed62f618c8fc84a44693136d96b11. The page includes fields for Transaction Hash, Status (Success), Block (6720878, 31 Block Confirmations), Timestamp (6 mins ago, Sep-19-2024 11:15:12 AM UTC), Transaction Action (Call, Method by 0xa152Df40...792377e68), From (0xa152Df401d2Be1620E9469cE2044A2D792377e68), To (Contract Creation [0xf81004c99be...]), Value (0 ETH (\$0.00)), Transaction Fee (0.040331171076805 ETH (\$0.00)), and Gas Price (36.713096091 Gwei (0.000000036713096091 ETH)). A note at the bottom indicates this is a Sepolia Testnet transaction only.

De igual forma se puede observar la información generada dentro de Alchemy al momento de la creación del contrato inteligente. En la figura 52, se puede observar las diferentes consultas realizadas al nodo al momento de crear el smart contract.

Figura 59

Detalle del despliegue en Alchemy

APP	ERROR CODE	HTTP	RESPONSE TIME	SENT	NETWORK
Hello World					
1 eth_getTransactionByHash	Hello World	200	0.063ms	9m ago	Ethereum Sepolia
PARAMS	RAW REQUEST				
0: 0xf81004c99bef44cafe0fd58312050e861b4ed62f618c8fc84a446931...					
RESULT	RAW RESPONSE				
Block Hash:	null				
Block Number:	null				
From:	0xa152df401d2be1620e9469ce2044a2d792377e...				
Gas:	1108436				
Gas Price:	4.51993872e+10				
Hash:	0xf81004c99bef44cafe0fd58312050e861b4ed6...				
Input:	0x608060405234801561001057600080fd5b5061...				
Max Fee Per Gas:	4.51993872e+10				
Max Priority Fee Per Gas:	1000000000				
2 eth_sendRawTransaction	Hello World	200	0.056ms	9m ago	Ethereum Sepolia
PARAMS	RAW REQUEST				
0: 0x02f9136f83aa36a70a843b9aca00850a8617ea688310e9d48000b913...					
RESULT	RAW RESPONSE				
Value: 0xf81004c99bef44cafe0fd58312050e861b4ed62f618c8fc84a446...					
3 eth_chainId	Hello World	200	0.024ms	9m ago	Ethereum Sepolia
PARAMS	RAW REQUEST				
--					
RESULT	RAW RESPONSE				
Value: 11155111					
4 eth_getTransactionCount	Hello World	200	0.038ms	9m ago	Ethereum Sepolia
PARAMS	RAW REQUEST				
0: 0xa152df401d2be1620e9469ce2044a2d792377e68					
1: pending					
RESULT	RAW RESPONSE				
Value: 10					

4.5. Ambientes de la aplicación

Un ambiente de la aplicación consiste en los distintos recursos, herramientas y aplicaciones utilizados al momento de trabajar en una fase concreta de la misma. Debido a la complejidad de la aplicación y a la necesidad de verificar el funcionamiento de las aplicaciones desarrolladas, en el presente proyecto se hizo uso de tres ambientes: ambiente de desarrollo, ambiente de integración y ambiente de producción.

4.5.1. Ambiente de desarrollo

Este fue el entorno que se utilizó para el desarrollo de las aplicaciones. En este se desarrollaron las distintas pruebas de concepto, pruebas técnicas y, en sí, todo el desarrollo antes de pasar al ambiente de integración.

Este entorno se desarrolló solamente en una computadora, la cual contiene todas las herramientas y aplicaciones necesarias para llevar a cabo toda la fase de desarrollo. Las aplicaciones utilizadas durante el desarrollo se definen en la Tabla 7.

Tabla 7

Herramientas utilizadas en el ambiente de desarrollo

Aplicación	Tipo	Descripción
WebStorm	Entorno Integrado de Desarrollo	Brinda distintas herramientas que facilitan el desarrollo de aplicaciones web
Google Chrome	Navegador	Permite acceder a las páginas web desarrolladas
Insomnia	Cliente para Servicios REST	Se utilizó para probar los distintos recursos desarrollados en el backend. Principalmente se utilizó para probar la generación de PDFs.
Postman	Cliente para Servicios REST	Se utilizó para probar los recursos del Backend y diferentes API externas.
Docker	Contenedores	Permitió la creación del contenedor de la base de datos.

MacOS	Sistema Operativo	Sistema operativo de la computadora de desarrollo
Adobe Acrobat Reader	Lector PDF	Utilizado para comprobar que los PDF generados pudieran visualizarse correctamente
Metamask	Extensión en Chrome Billetera Blockchain	Permitió generar la cuenta para poder utilizar los servicios de Alchemy
Alchemy	Plataforma con servicios Blockchain	Se hizo uso del nodo y los servicios que proporciona para crear y utilizar el contrato inteligente.
DataGrip	Entorno de desarrollo integrado para Base de datos	Permitió visualizar los diferentes elementos y registros de la base de datos
iTerm2	Terminal	Herramienta que permitió ejecutar los distintos comandos en la Shell de MacOS

4.5.2. Ambiente de integración

Este ambiente fue utilizado para verificar el funcionamiento de las aplicaciones en Amazon Web Services con el objetivo de definir los pasos y recursos necesarios para el despliegue de la aplicación. Con este ambiente se pudo comprobar el funcionamiento y establecer los cambios que eran necesarios tener en cuenta al momento de desplegar la aplicación en producción. Además, sirvió como referencia y entorno de prueba para los encargados de estas dentro de Cloud Solution Network. Las herramientas utilizadas en este ambiente se detallan en la tabla 8.

Tabla 8*Herramientas utilizadas en el ambiente de integración*

Aplicación	Tipo	Descripción
Amazon Web Services	Servicio en la nube	Plataforma que brinda recursos de infraestructura para la implementación de distintas aplicaciones.
Google Chrome	Navegador	Permite acceder a las páginas web desarrolladas y los servicios web utilizados
Git	Control de versiones	La aplicación Git fue utilizada para llevar el control del código desarrollado y posteriormente se utilizó el servicio Github para almacenar el proyecto, esto debido a la necesidad de clonar el proyecto dentro de la máquina virtual en AWS.
Ubuntu	Sistema Operativo	Sistema operativo del servidor de integración
Adobe Acrobat Reader	Lector PDF	Utilizado para comprobar que los PDF generados pudieran visualizarse correctamente
Metamask	Extensión en Chrome Billetera Blockchain	Permitió generar la cuenta para poder utilizar los servicios de Alchemy
Alchemy	Plataforma con servicios Blockchain	Se hizo uso del nodo y los servicios que proporciona

		para crear y utilizar el contrato inteligente.
DataGrip	Entorno de desarrollo integrado para Base de datos	Permitió visualizar los diferentes elementos y registros de la base de datos
iTerm2	Terminal	Herramienta que permitió ejecutar los distintos comandos en la Shell de MacOS
SSH	Encriptación	Permitió generar las claves necesarias por Github en el servidor de integración.

4.5.3. Ambiente de producción

Es el ambiente utilizado por la empresa Cloud Solution Network para la puesta en marcha de todas las aplicaciones y la base de datos necesarias para el funcionamiento del proyecto. Este ambiente fue definido y creado por los empleados de Cloud Solution Network, siguiendo los lineamientos establecidos en el entorno de integración. Las herramientas utilizadas para este ambiente fueron las mismas del ambiente de integración; estas se detallan en la tabla 8.

5. Capítulo 5 – Pruebas de validación

Las pruebas de validación que se realizaron permiten verificar que los requerimientos funcionales y no funcionales definidos para el proyecto se hayan cumplido exitosamente. Estas pruebas incluyen las diferentes aplicaciones involucradas en el proyecto: el frontend, backend y los contratos inteligentes.

5.1. Pruebas técnicas

Las pruebas técnicas se enfocan en evaluar el correcto funcionamiento de un sistema, verificando que sus módulos y funcionalidades cumplan con los requerimientos funcionales y no funcionales. Estas pruebas aseguran que el software opere de manera eficiente, segura y sin errores.

En el proyecto, se aplicaron para validar la autenticación mediante JWT, garantizando el acceso seguro a las funcionalidades restringidas. Además, se evaluó la correcta gestión de usuarios, cursos y estudiantes, incluyendo la prevención de duplicados y la validación de campos obligatorios. Por último, se probaron los procesos de generación, verificación y revocación de certificados en la Blockchain, asegurando su integridad y adecuado almacenamiento en el sistema. En la tabla 9 se pueden ver el resultado de estas pruebas.

Tabla 9

Resultado de pruebas técnicas realizadas.

No. Prueba	Fecha prueba	Nombre de prueba	Descripción	Resultado	Estado
1	05/10/2024	Validación de autenticación JWT	Verificar que los usuarios pueden autenticarse correctamente utilizando JWT	El sistema genera un token JWT válido al ingresar con credenciales correctas.	Aprobado

2	05/10/2024	Validación de acceso a funcionalidades restringidas	Asegurarse de que solo los usuarios autenticados pueden acceder a funcionalidades restringidas como la creación de cursos o estudiantes	Solo usuarios con un token JWT válido pueden acceder a funcionalidades restringidas.	Aprobado
3	05/10/2024	Validación de mensajes de error en autenticación	Comprobar que al ingresar credenciales incorrectas se muestre un mensaje de error adecuado	El sistema muestra un mensaje de error claro al ingresar credenciales incorrectas.	Aprobado
4	05/10/2024	Validación de creación de usuarios	Verificar que los administradores pueden crear usuarios con los campos necesarios	Los usuarios son creados con éxito con los campos: email, nombre, apellido, salt, password, created_at, updated_at.	Aprobado
5	05/10/2024	Validación edición y eliminación de usuarios	Verificar que los administradores y puedan editar y eliminar usuarios del sistema	Los administradores y pueden editar y eliminar usuarios correctamente.	Aprobado
6	05/10/2024	Validación de creación de cursos	Probar la creación de cursos por parte de los administradores con los campos requeridos	Los cursos se crean correctamente con los campos: nombre, descripción, profesor, fechas, created_by, updated_by.	Aprobado
7	05/10/2024	Validación edición y eliminación de cursos	Verificar que los administradores y pueden editar y eliminar cursos, y que no se permita la eliminación si el curso tiene estudiantes	La edición y eliminación de cursos funcionan correctamente, y no se puede eliminar un curso con estudiantes asignados.	Aprobado

				asignados		
8	05/10/2024	Validación de generación de certificados	de de	Asegurarse de que los certificados se generen automáticamente cuando las asignaciones de curso se marcan como “finalizadas”	El sistema genera certificados correctamente al finalizar una asignación.	Reprobado
9	05/10/2024	Validación del almacenamiento de certificados en la Blockchain		Probar que la información de los certificados se almacena correctamente en la Blockchain y que el backend recibe el resultado del contrato inteligente	La información se almacena en la Blockchain, y el backend recibe una confirmación de éxito.	Aprobado
10	05/10/2024	Validación de descarga de certificados PDF	de en	Verificar que los certificados generados pueden descargarse en formato PDF con el identificador URL de la Blockchain y el código QR	El certificado se descarga con el identificador de la Blockchain y el código QR de verificación.	Aprobado
11	05/10/2024	Validación de envío por correo electrónico del certificado		Asegurarse de que el sistema permite enviar el certificado por correo electrónico una vez generado	El certificado se envía correctamente por correo al destinatario.	Aprobado
12	05/10/2024	Validación de carga de archivo PDF para verificación pública		Probar que la página pública permite la carga de un archivo PDF para validar su autenticidad	El sistema permite la carga de un PDF y valida su autenticidad.	Aprobado
13	05/10/2024	Validación de manejo de usuarios concurrentes	de	Verificar que la aplicación puede manejar hasta 10 usuarios concurrentes	El sistema maneja hasta 10 usuarios concurrentes sin	Aprobado

14	05/10/2024	Pruebas de seguridad: JWT y encriptación con Bcrypt	concurrentes sin pérdida significativa de rendimiento	degradación en el rendimiento.	
15	05/10/2024	Prueba de rendimiento del tiempo de validación de certificados	Realizar pruebas para verificar que las sesiones de los usuarios y las contraseñas estén seguras, siguiendo las mejores prácticas de seguridad, incluyendo JWT y Bcrypt con salt	Las sesiones y contraseñas son seguras y no vulnerables a ataques de fuerza bruta o de diccionario.	Aprobado
16	05/10/2024	Validación eliminación estudiantes	Medir el tiempo de respuesta del sistema para la validación de certificados, asegurándose de que no supere los 3 segundos	El tiempo de respuesta es menor a 3 segundos.	Aprobado
17	05/10/2024	Validación asignación estudiantes cursos	Verificar que los administradores pueden eliminar estudiantes, y que no se permita la eliminación si el estudiante tiene cursos asignados	La eliminación es exitosa solo para estudiantes sin cursos asignados, y el sistema muestra un error para aquellos que tienen asignaciones activas.	Aprobado
18	06/10/2024	Validación estados asignación cursos	Probar que los administradores a pueden asignar y desasignar estudiantes a los cursos	Los estudiantes se asignan correctamente a los cursos y pueden ser desasignados cuando corresponda.	Aprobado

			de asignación de estudiantes: activo, finalizado, y certificado	según las acciones del administrador.	
19	06/10/2024	Validación de duplicación de cursos	Verificar que el sistema impida la creación de cursos con nombres duplicados	El sistema muestra un error si se intenta crear un curso con el mismo nombre que uno ya existente.	Aprobado
20	06/10/2024	Validación de duplicación de estudiantes	Probar que el sistema no permite la creación de estudiantes con el mismo correo electrónico	El sistema muestra un mensaje de error si se intenta registrar a un estudiante con un correo ya utilizado.	Aprobado
21	06/10/2024	Validación de visualización de lista de cursos	Asegurarse de que los administradores pueden visualizar correctamente la lista de todos los cursos creados	La lista de cursos es visible y muestra todos los cursos con sus respectivos detalles.	Aprobado

6. Capítulo 6 – Pruebas de certificación

Estas pruebas están diseñadas para asegurar que la aplicación cumple con los requisitos funcionales y no funcionales descritos en el documento del proyecto. El encargado del proyecto en Cloud Solution Network ha realizado cada prueba con el propósito de garantizar que el sistema esté listo para su despliegue.

6.1. Hallazgos

Uno de los problemas que encontrados fue la configuración de las variables de entorno, al momento del despliegue, no todas las variables de entorno existian en el archivo .env del ambiente de integración. Esto dio origen a un error al momento de realizar llamadas al contrato inteligente como se muestra en la figura 53, por lo que después de distintas pruebas se logró localizar el error.

Figura 60

Error en variables de entorno .env

Luego de las correcciones realizadas, el encargado de las pruebas en Cloud Solutions Network verificó que los requerimientos solicitados estuvieran presentes en el proyecto. En la tabla 10 se pueden observar las pruebas realizadas.

Tabla 10*Pruebas de certificación*

No. Prueba	Fecha prueba	Tipo de prueba	Resultado	Estado
1	02/10/2024	Validación del acceso inicial	Los usuarios pudieron ingresar correctamente al sistema utilizando credenciales válidas y el mecanismo de autenticación JWT.	Aprobado
2	02/10/2024	Registro usuarios	Los administradores fueron capaces de registrar nuevos usuarios, y la información se almacenó de manera adecuada en la base de datos.	Aprobado
3	02/10/2024	Asignación estudiantes cursos	Se verificó que los estudiantes fueron asignados correctamente a los cursos correspondientes, sin errores en la base de datos.	Aprobado
4	02/10/2024	Generación certificados digitales	Los certificados digitales fueron generados con éxito al finalizar los cursos, cumpliendo con los requerimientos de almacenamiento en la Blockchain.	Aprobado
5	03/10/2024	Verificación pública certificados	Los usuarios externos pudieron verificar la autenticidad de los certificados a través de los enlaces públicos y códigos QR, garantizando su validez en la Blockchain.	Aprobado
6	03/10/2024	Descarga certificados PDF	Los usuarios pudieron descargar sus certificados en formato PDF, incluyendo la información de la transacción en la Blockchain y el código QR.	Aprobado
7	04/10/2024	Revocación certificados	Los administradores revocaron certificados previamente emitidos, y los cambios se reflejaron correctamente en la Blockchain.	Aprobado
8	04/10/2024	Evaluación seguridad	Se probaron los mecanismos de seguridad de la aplicación, incluyendo la prevención de ataques de SQL Injection y Cross-Site Scripting (XSS), asegurando que no se comprometa la integridad del sistema.	Aprobado

9	04/10/2024	Gestión de usuarios	de	Los administradores fueron capaces de crear, editar y eliminar usuarios, garantizando que las contraseñas estuvieran adecuadamente cifradas mediante Bcrypt.	Aprobado
10	04/10/2024	Rendimiento con múltiples usuarios concurrentes	de	Se probó que la aplicación maneja hasta 5 usuarios concurrentes sin problemas de rendimiento o interrupciones del servicio.	Aprobado
11	04/10/2024	Visualización de cursos y estudiantes	de	Los administradores accedieron correctamente a las listas de cursos y estudiantes registrados, con datos precisos y completos.	Aprobado
12	04/10/2024	Edición de información de cursos	de	Los administradores pudieron modificar la información de los cursos y estudiantes, y los cambios se reflejaron correctamente sin afectar otros datos.	Aprobado
13	04/10/2024	Proceso finalización cursos	de	Se validó que el sistema permite finalizar los cursos y habilitar la generación de certificados de manera automática, sin errores.	Aprobado
14	04/10/2024	Cierre de sesión		Los usuarios cerraron sesión sin inconvenientes en todos los dispositivos utilizados durante la prueba.	Aprobado
15	04/10/2024	Validación tiempo respuesta verificación certificados	del de de	Se verificó que la validación de certificados en la Blockchain ocurre en menos de 30 segundos, cumpliendo con los requerimientos de rendimiento.	Aprobado
16	07/10/2024	Verificación de carga de archivos en la página pública	de	Se verificó que la página pública permite la carga de archivos PDF para validar la autenticidad de certificados previos, cumpliendo con los requisitos.	Aprobado
17	07/10/2024	Restricción eliminación cursos	de	Se probó que los cursos con estudiantes asignados no pueden ser eliminados, respetando las restricciones definidas en los requerimientos.	Aprobado
18	07/10/2024	Visualización de certificados históricos	de	Se validó que los usuarios pueden visualizar y descargar certificados emitidos en el pasado sin pérdida de datos.	Aprobado
29	07/10/2024	Validación	de	Se aseguró que las actualizaciones	Aprobado

integridad de en la información de cursos no datos tras afectaron la integridad de los datos modificaciones de los estudiantes ni de los certificados previamente emitidos.

Estas pruebas adicionales completan el ciclo de validación de la aplicación, cubriendo desde la gestión de usuarios y cursos hasta la generación y verificación de certificados en la Blockchain, así como la seguridad y estabilidad del sistema en escenarios reales. A continuación, en la figura 54, se puede observar la carta de certificación de pruebas de funcionalidad que se envió a Cloud Solutions Network.

Figura 61

Carta de certificación



Code Solutions Network
14 Avenida 18-37 Zona 13
Edificio Global Business Center
(502) 2440-4907
devs@mypeopleapps.com

CERTIFICACIÓN DE PRUEBAS DE FUNCIONALIDAD

Guatemala 31 de octubre de 2024

Ingeniero Saúl Orozco
Catedrático Universidad Mariano Gálvez de Guatemala
Sede Boca del Monte
Pte.

Estimado Ingeniero:

Confiando en que sus actividades se desarrolle con éxito, a través de la presente informo que, tras llevar a cabo las diferentes pruebas en la aplicación web para el control de certificaciones del proyecto de tesis titulado: "**INTEGRACIÓN DE UNA APLICACIÓN WEB CON CONTRATOS INTELIGENTES EN ETHEREUM PARA LA GENERACIÓN Y VERIFICACIÓN DESCENTRALIZADA DE DIPLOMAS DIGITALES PARA INSTITUCIONES QUE BRINDAN CURSOS EN LÍNEA**", esta cumple con los requerimientos especificados en el documento SRS.

Por lo tanto, se ha podido verificar que la aplicación satisface los requisitos establecidos en las pruebas realizadas.

En consecuencia, se aprueba y doy mi aceptación en función de la utilidad que representa dicha aplicación.

Atentamente,



Marlon Coti
Engineering Manager
Code Solutions Network
Tel: 4126-4685

A handwritten signature in black ink is placed over a horizontal line. Below the signature, the name "Marlon Coti" is printed in a standard black font, followed by three lines of professional title and contact information.

7. Capítulo 7 – Implementación

Según la Universidad Autónoma del Estado de Hidalgo (2019), es la última fase dentro del desarrollo de un sistema, la cual consiste en el proceso de configurar los equipos y las herramientas necesarias para la instalación de la aplicación, habiendo definido previamente el análisis y diseño.

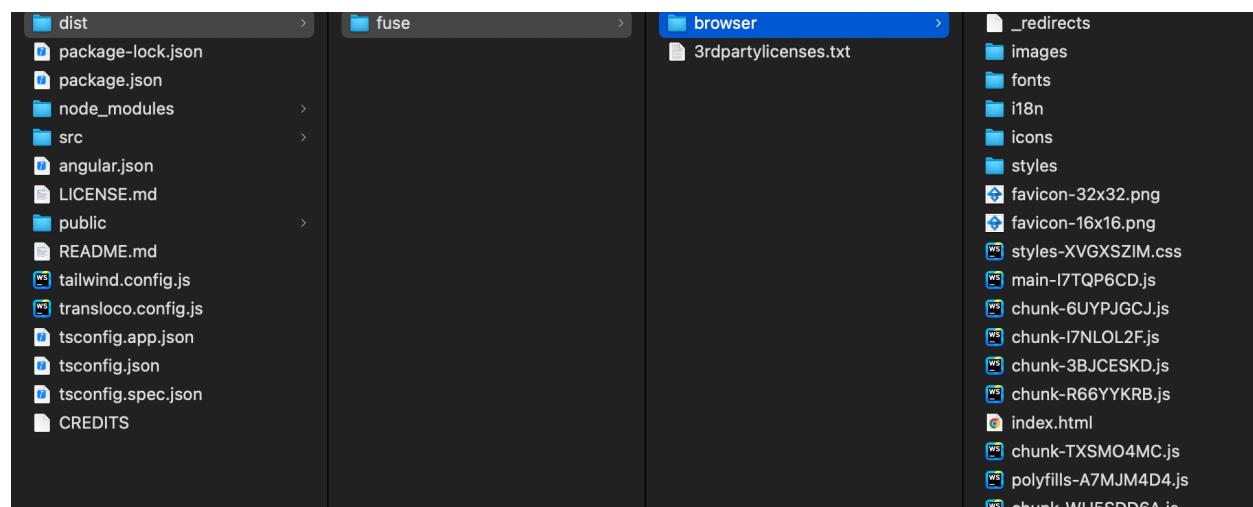
Para la implementación del proyecto se utiliza Amazon Web Services (AWS) para alojar las distintas aplicaciones. Debido a los requerimientos y necesidades del usuario, se definen los pasos necesarios para realizar la instalación de las aplicaciones en los distintos recursos que proporciona AWS.

7.1.1. *Frontend*

Para desplegar la aplicación Frontend, se construye el código necesario para producción. Esto se realiza ejecutando el comando `ng build` en la raíz del directorio, lo cual genera todos los archivos necesarios para la aplicación, como se puede observar en la figura 55.

Figura 62

Archivos para el despliegue del frontend



Debido a la facilidad para el despliegue y la funcionalidad que brinda AWS con el servicio S3, se utiliza un bucket para el despliegue de todos los archivos generados. Como se puede

observar en la figura 56, los archivos generados por el comando build se cargan en un bucket de S3. Posteriormente, este servicio se configura para permitir el acceso público.

Figura 63

Bucket en S3 para el Frontend

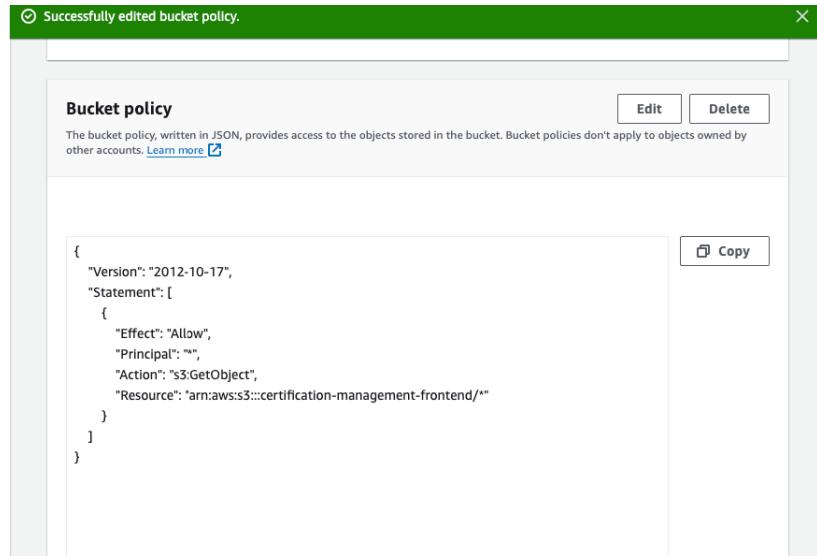
The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various navigation options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. Below that, there are sections for Block Public Access settings, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, and a Feature spotlight. At the bottom of the sidebar, it says 'AWS Marketplace for S3'. The main area shows a breadcrumb path: Amazon S3 > Buckets > certification-management-frontend > browser/. The title bar has a 'Copy S3 URI' button. Below the title, there are tabs for 'Objects' and 'Properties'. A search bar at the top of the object list contains the placeholder 'Find objects by prefix'. The object list table has columns for Name, Type, Last modified, Size, and Storage class. The table shows 41 objects, with the first few being 'redirects', 'chunk-SBICESKD.js', 'chunk-4HNRFIOR.js', 'chunk-SFZUURZ7.js', 'chunk-SZX4ZW7Z.js', 'chunk-GXNHN7I.js', 'chunk-6UYPJGCJ.js', 'chunk-A4P3Z43L.js', and 'chunk-A7ZHCOJM.js'. All objects are of type 'js' and have a standard storage class. The last modified date for all objects is October 10, 2024, at 18:38:09 UTC-06:00, except for the first two which are at 18:38:11 UTC-06:00.

Name	Type	Last modified	Size	Storage class
redirects	-	October 10, 2024, 18:38:09 (UTC-06:00)	21.0 B	Standard
chunk-SBICESKD.js	js	October 10, 2024, 18:38:11 (UTC-06:00)	597.0 B	Standard
chunk-4HNRFIOR.js	js	October 10, 2024, 18:38:05 (UTC-06:00)	2.0 KB	Standard
chunk-SFZUURZ7.js	js	October 10, 2024, 18:38:20 (UTC-06:00)	6.7 KB	Standard
chunk-SZX4ZW7Z.js	js	October 10, 2024, 18:38:00 (UTC-06:00)	6.1 KB	Standard
chunk-GXNHN7I.js	js	October 10, 2024, 18:38:19 (UTC-06:00)	118.7 KB	Standard
chunk-6UYPJGCJ.js	js	October 10, 2024, 18:38:12 (UTC-06:00)	801.0 B	Standard
chunk-A4P3Z43L.js	js	October 10, 2024, 18:38:13 (UTC-06:00)	5.8 KB	Standard
chunk-A7ZHCOJM.js	js	October 10, 2024, 18:38:12 (UTC-06:00)	29.1 KB	Standard
-	-			

Para configurar el punto de acceso y habilitar los permisos en S3, se utiliza una *Bucket policy* que permite definir los permisos de recursos mediante JSON, como se muestra en la figura 57. Este paso es necesario para permitir el acceso.

Figura 64

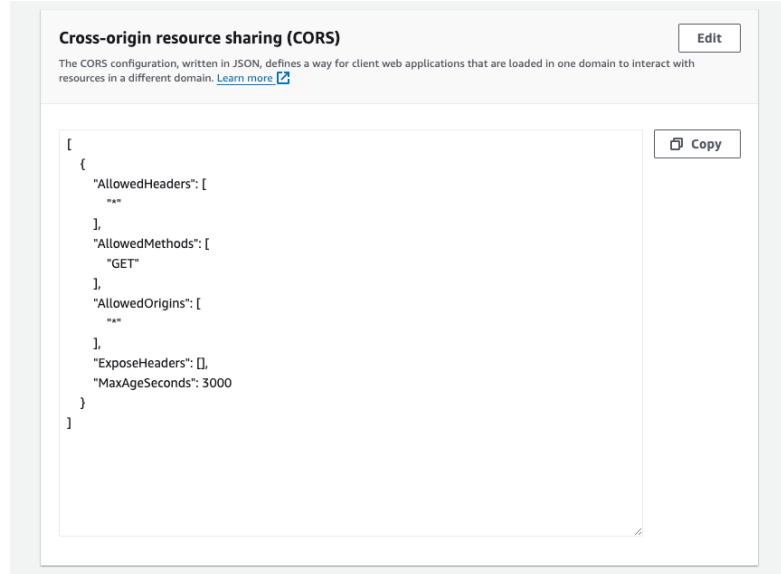
Bucket policy en S3



Finalmente, es necesario habilitar los CORS para los archivos estáticos para que estos puedan ser accesibles por los usuarios. Esto se realiza en la sección Cross-Origin Resource Sharing dentro de S3, como se muestra en la figura 58.

Figura 65

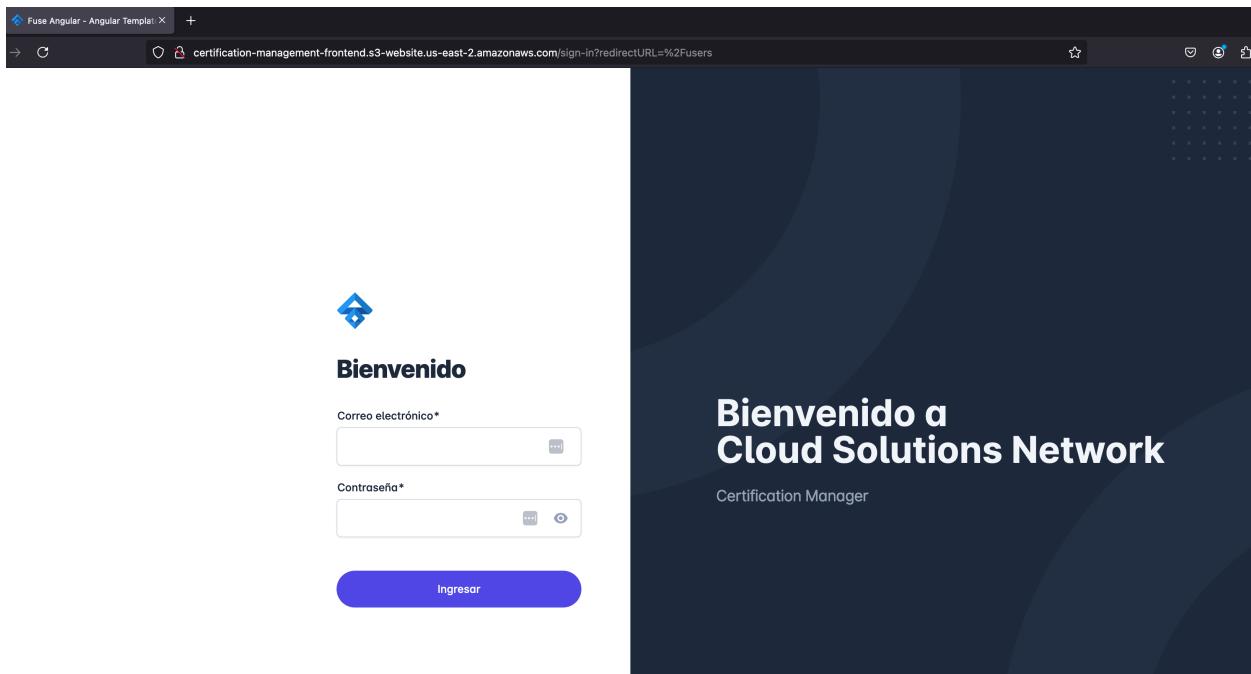
Configuración de CORS



Una vez configurados estos pasos, la aplicación puede ser accesible desde la web. En la figura 59, se puede visualizar la página web desde un navegador.

Figura 66

Frontend desde el navegador



7.1.2. Backend

Para la aplicación backend, que es la encargada de la conexión con la base de datos, la integración con la red Blockchain y la exposición de los servicios para el consumo de la aplicación Frontend, se utilizan distintos servicios de AWS. Inicialmente, se configura la base de datos, utilizando AWS RDS, que es el servicio de bases de datos relacionales. Para el proyecto, se utiliza PostgreSQL, como se visualiza en la figura 60. El tamaño que se utiliza es db.t4g.micro, como se puede observar en la figura 61.

Figura 67

Creación de base de datos en AWS

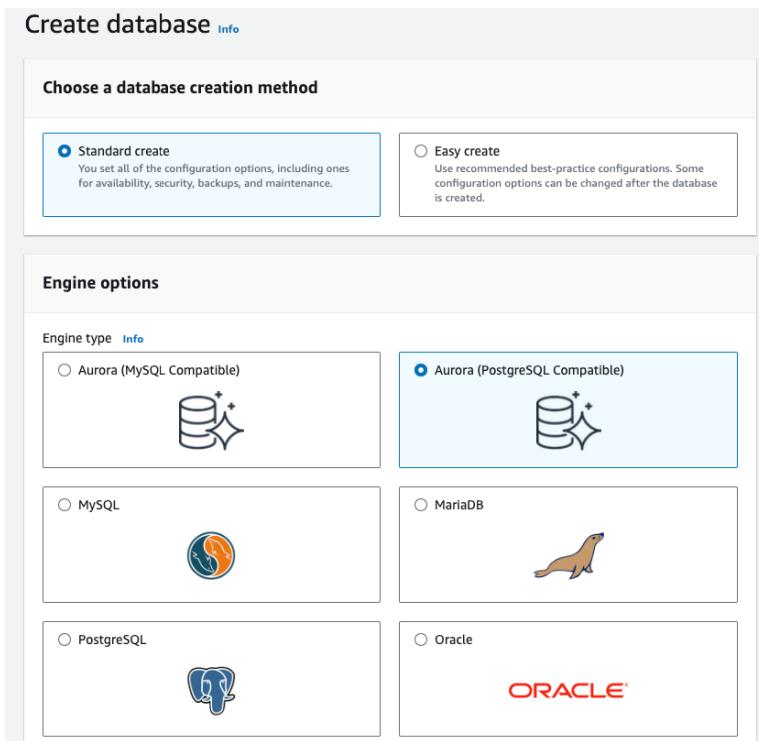
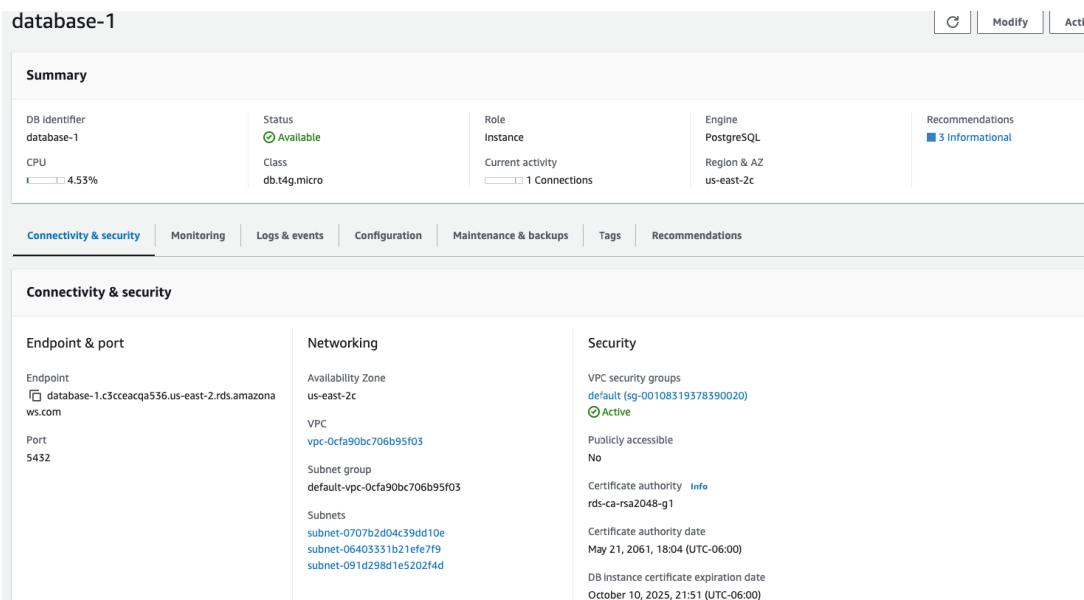


Figura 68

Instancia de la base de datos en RDS



Para la aplicación Backend, se utiliza el servicio EC2, el cual permite la creación de máquinas virtuales. Este servicio se utiliza para montar una máquina en la cual se puedan instalar todas las herramientas necesarias para ejecutar la aplicación. La máquina virtual utilizada es t2.micro con sistema operativo Ubuntu. En la figura 62 se pueden visualizar los detalles de la máquina una vez creada.

Figura 69

Instancia EC2

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. The main content area is titled 'Instance summary for i-0f00c55af311aadba (Backend NestJS)'. It displays the following details:

Detail	Value
Instance ID	i-0f00c55af311aadba (Backend NestJS)
Public IPv4 address	18.116.61.241 open address
Private IPv4 addresses	172.31.23.148
IPv6 address	-
Instance state	Running
Hostname type	IP name: ip-172-31-23-148.us-east-2.compute.internal
Private IP DNS name (IPv4 only)	ip-172-31-23-148.us-east-2.compute.internal
Answer private resource DNS name	IPv4 (A)
Instance type	t2.micro
Elastic IP addresses	-
VPC ID	vpc-0cfa90bc706b95f03
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for r recommendations.
IAM Role	-
Subnet ID	subnet-091d298d1e5202f4d
IMDSv2	Required
Instance ARN	arn:aws:ec2:us-east-2:816069160518:instance/i-0f00c55af311aadba
Auto Scaling Group name	-

Una vez creada la instancia, es necesario configurar todas las herramientas y recursos necesarios para el correcto funcionamiento de la aplicación. Como se puede observar en la figura 63, se ejecuta la terminal de la instancia para proceder con las instalaciones respectivas.

Figura 70

Ejecución de instancia

The screenshot shows the AWS EC2 Dashboard. The left sidebar includes links for EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays 'Resources' for the US East (Ohio) Region, showing counts for various EC2 components. It also features a 'Launch instance' button and a 'Service health' section.

Resource Type	Count
Instances (running)	0
Auto Scaling Groups	0
Capacity Reservations	0
Dedicated Hosts	0
Elastic IPs	0
Instances	0
Key pairs	0
Load balancers	0
Placement groups	0
Security groups	1
Snapshots	0
Volumes	0

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

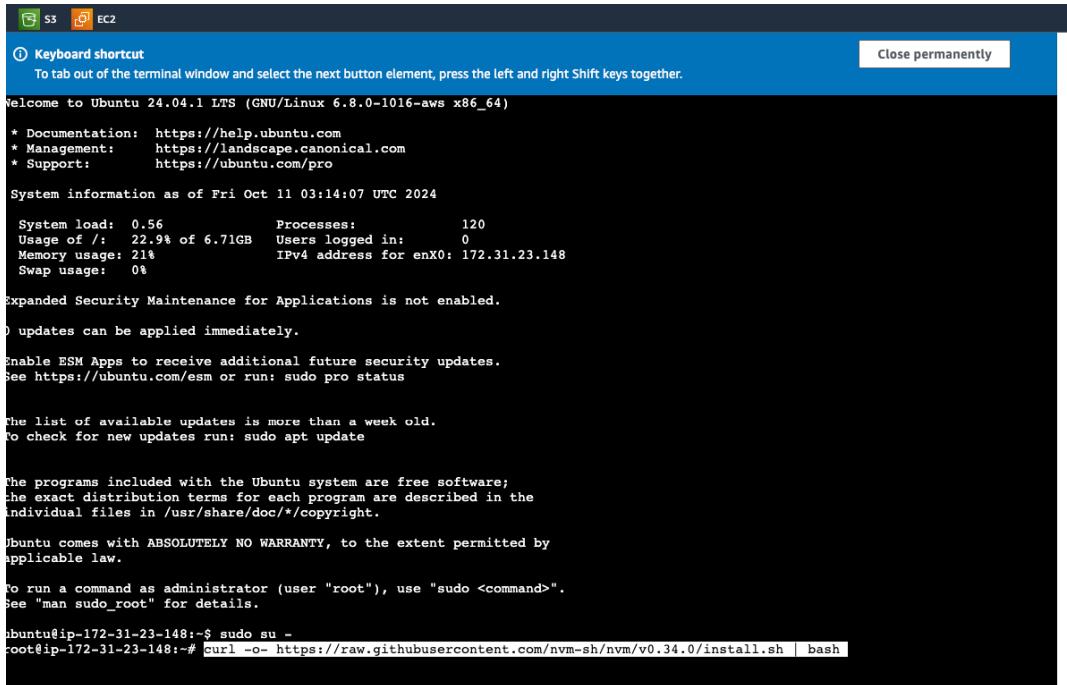
Service health
AWS Health Dashboard

Region
US East (Ohio)

Una vez dentro de la terminal, se utiliza el usuario principal para realizar las instalaciones, como se puede observar en la figura 62. Posteriormente, se procede a la instalación de nvm para manejar las instalaciones de NodeJS y también se instala GIT, como se visualiza en la figura 64.

Figura 71

Cambio de usuario



The screenshot shows a terminal window titled "S3 EC2" with a blue header bar containing "Keyboard shortcut" and "Close permanently". The terminal output is as follows:

```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Oct 11 03:14:07 UTC 2024

 System load: 0.56      Processes:          120
 Usage of /: 22.9% of 6.71GB  Users logged in:    0
 Memory usage: 21%          IPv4 address for enX0: 172.31.23.148
 Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

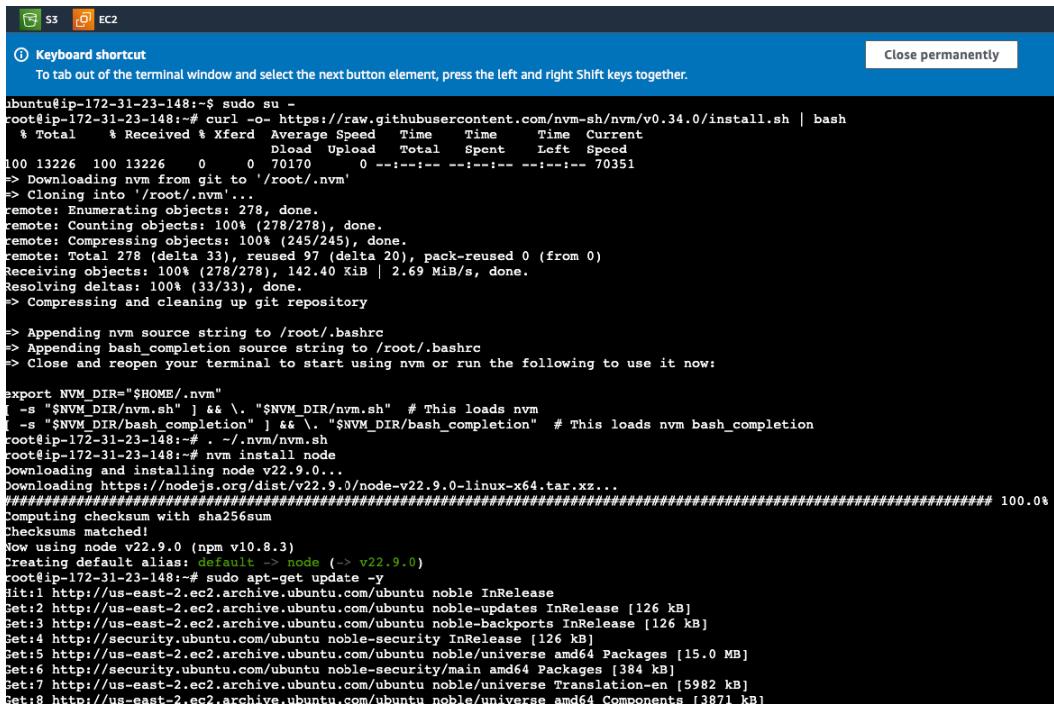
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-23-148:~$ sudo su -
root@ip-172-31-23-148:~# curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
```

Figura 72

Instalacion de NVM y NodeJS



The screenshot shows a terminal window titled "S3 EC2" with a blue header bar containing "Keyboard shortcut" and "Close permanently". The terminal output is as follows:

```
ubuntu@ip-172-31-23-148:~$ sudo su -
root@ip-172-31-23-148:~# curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
  % Total    % Received   Average Speed  Time     Time   Current
          0       0      0KB/s  0:00:00  0:00:00 --:--:-- 70351
->  Downloading nvm from git to '/root/.nvm'
->  Cloning into '/root/.nvm'...
remote: Enumerating objects: 278, done.
remote: Counting objects: 100% (278/278), done.
remote: Compressing objects: 100% (245/245), done.
remote: Total 278 (delta 33), reused 97 (delta 20), pack-reused 0 (from 0)
Receiving objects: 100% (278/278), 142.40 KiB | 2.69 MiB/s, done.
Resolving deltas: 100% (33/33), done.
->  Compressing and cleaning up git repository

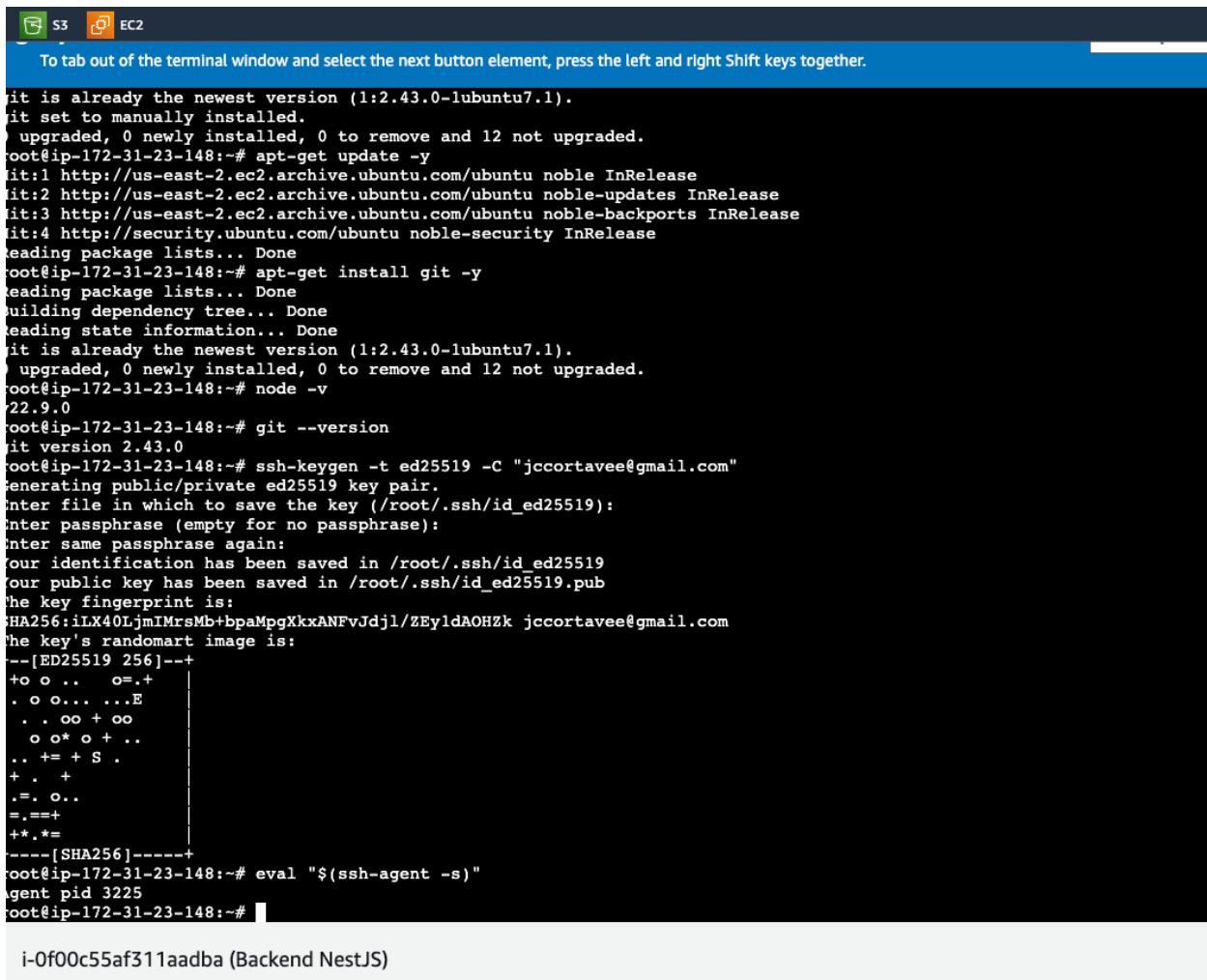
->  Appending nvm source string to /root/.bashrc
->  Appending bash_completion source string to /root/.bashrc
->  Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
root@ip-172-31-23-148:~# . ~/.nvm.sh
root@ip-172-31-23-148:~# nvm install node
Downloading and installing node v22.9.0...
Downloading https://nodejs.org/dist/v22.9.0/node-v22.9.0-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v22.9.0 (npm v10.8.3)
Creating default alias: default -> node (-> v22.9.0)
root@ip-172-31-23-148:~# sudo apt-get update -y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease [126 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [384 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
```

Para clonar el repositorio de Github, es necesaria la creación de una llave SSH para permitir el acceso al código. En la figura 66 se puede visualizar la llave SSH generada.

Figura 73

Configuración de clave SSH



The screenshot shows a terminal window within the AWS CloudWatch interface. The window title is 'EC2'. The terminal content displays the process of generating an SSH key:

```
it is already the newest version (1:2.43.0-1ubuntu7.1).
it set to manually installed.
) upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
root@ip-172-31-23-148:~# apt-get update -y
it:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
it:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
it:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
it:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
root@ip-172-31-23-148:~# apt-get install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
it is already the newest version (1:2.43.0-1ubuntu7.1).
) upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
root@ip-172-31-23-148:~# node -v
v22.9.0
root@ip-172-31-23-148:~# git --version
git version 2.43.0
root@ip-172-31-23-148:~# ssh-keygen -t ed25519 -C "jccortavee@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:iLX40LjmIMrsMb+bpaMpgXkxANFvJdjl/ZEyldAOHzk jccortavee@gmail.com
The key's randomart image is:
-[ED25519 256]-
+o o .. o=,+ |
. o o....E |
.. oo + oo |
o o* o + .. |
.. += + S . |
+ . +
.=. o...
=.=+
+*.*=+
-----[SHA256]-----
root@ip-172-31-23-148:~# eval "$(ssh-agent -s)"
Agent pid 3225
root@ip-172-31-23-148:~#
```

i-0f00c55af311aadba (Backend NestJS)

Finalmente, se procede a la configuración de Alchemy y Metamask, siguiendo un procedimiento similar al descrito en el Capítulo 4, en las figuras 29 y 30. En esta parte, se configura una cuenta en Blockchain por parte del cliente utilizando la herramienta Metamask, y también se crea la conexión entre Alchemy y Metamask para su uso desde el entorno de producción. Este

procedimiento es realizado por un empleado de Cloud Solutions Network para tener control de las claves generadas y garantizar una mayor seguridad para la empresa.

8. Capítulo 8 – Mantenimiento

Como toda aplicación de sistemas es necesario brindar un mantenimiento, esto con el objetivo de minizar errores de seguridad y mejorar el rendimiento de la aplicación. A continuación se describen el proceso de mantenimiento que se llevará a cabo dentro del sistema para continuar con su rendimiento óptimo.

8.1. Actualización de librerías

Debido al constante avance de las herramientas utilizadas, se dará seguimiento mensual a los hallazgos encontrados en las librerías utilizadas y posteriormente se procederá con su instalación. Esto es de gran importancia ya que muchas herramientas son de código abierto y reciben actualizaciones constantes para mejorar el rendimiento y cerrar brechas de seguridad. La actualización de las librerías se empezará a llevar a cabo a partir del próximo mes de enero.

8.2. Depósito de Ethereum

Los contratos inteligentes utilizan Ether para escribir información dentro de Ethereum, por ello es necesario un monitoreo al momento de crear un certificado para evitar fallos en la creación por falta de fondos suficientes. Para futuras actualizaciones de la aplicación se mostrará el balance de la cuenta con el objetivo de informar al usuario antes de la generación de los certificados.

8.3. Mejoras solicitadas

La aplicación se desarrolló siguiendo los requerimientos solicitados por Cloud Solutions Network, pero como toda aplicación se desea seguir mejorando la plataforma y se analizaron formas en mejorar la aplicación actual. Una de ellas es la integración con distintas API de

Blockchain, esto con el objetivo de tener un mejor control del costo de Ethereum, se desea visualizar cuales son los días donde el precio de la criptomonedas es menor y el historial de una semana, esto para poder generar los certificados al menor costo posible, también tener un costo promedio por todos los certificados que se han generado, esto con el objetivo de conocer el precio que es necesario incrementar en los cursos y así tomar en cuenta los gastos que requiere almacenar la información dentro de la red de Ethereum.

8.3.1. Monitoreo del rendimiento de la aplicación

Actualmente la aplicación está desarrollada para un grupo de usuarios que no superan los 100 estudiantes, pero el objetivo es ir creciendo y para ello se debe verificar como el incremento de estudiantes y de validaciones de certificados afecta el rendimiento de la aplicación. Para ello se va a tener un monitoreo constante para visualizar si el hardware utilizado se ve afectado con el mayor uso de la aplicación y así tomar las medidas que considere necesaria la empresa. Debido al uso de AWS para la implementación del proyecto, el proceso de incrementación de recursos no conyeva mayor inconveniente, debido a la facilidad de la plataforma para el crecimiento vertical de servidores.

8.3.2. Creación de servicio especializado para generación y validación de certificados

Después de observar el comportamiento de la aplicación y tomando en cuenta la retroalimentación de personas involucradas en el proceso, se tiene pensado extraer el servicio que genera y valida certificado en la red de Ethereum, esto con el objetivo de utilizar las características de los contratos inteligentes en otros proyectos y si es posible, brindar servicios externos a otras empresas para que ellas puedan utilizar los mismos contratos inteligentes y así obtener beneficios económicos del contrato inteligente actual o futuros contratos que puedan crearse. Para ellos se

tiene pensado la creación de contenedores de Docker, para exponer estos servicios a las distintas plataformas que lo requieran.

8.3.3. Otras blockchain

Para el proyecto se utilizo Ethereum debido a que es una de las redes Blockchain más robustas y con mayor documentación, pero existen otras redes blockchains que también permiten la funcionalidad de blockchain a un costo menor como lo son Cardano, Avalanche y Solana. Esta es una opción viable para reducir costos siempre y cuando el desarrollo sea factible. Con el objetivo de brindar la misma funcionalidad a un menor precio, se desea probar si estas blockchains pueden brindar la funcionalidad actual del proyecto.

9. Conclusión

Una vez implementado el proyecto, como parte de los resultados obtenidos podemos indicar que la plataforma ayuda a reducir los tiempos de espera tanto en la generación como en la validación. El tiempo de generación de certificados educativos dentro de Cloud Solutions Network fue reducido de entre una y dos semanas a un promedio de 2 horas. El tiempo de validación se redujo de una semana a un aproximado de un minuto, esto debido a que el usuario puede validar su certificado desde la página web proporcionada por la empresa o verificar su certificado desde la plataforma de *Etherscan* sin la necesidad de un intermediario.

La implementación de la plataforma para la generación y validación de certificados utilizando Blockchain fue exitosa. Se logró integrar con eficacia los contratos inteligentes en la red Ethereum, permitiendo una gestión automatizada, segura y transparente de los certificados emitidos. Esto responde de manera positiva al objetivo general de la investigación, demostrando que Blockchain es una tecnología efectiva para garantizar la integridad y la trazabilidad de los certificados sin intervención manual.

Ethereum fue identificado como una red adecuada para la implementación de contratos inteligentes debido a su infraestructura robusta, alta adopción y extensa documentación. Esta red cumplió con las expectativas del proyecto en términos de funcionalidad y seguridad. En respuesta al segundo objetivo específico, se concluye que Ethereum es una plataforma viable para la validación de certificados, cumpliendo los estándares de transparencia y seguridad requeridos para la aplicación.

La aplicación logró almacenar los registros de validación de certificados con ello, se pudo obtener un promedio de los intentos de falsificación y así cumplir de manera exitosa uno de los objetivos específicos. El contrato inteligente es capaz de identificar si un hash es inválido y

posteriormente el sistema almacena los resultados que pueden ser utilizados para beneficio de la empresa.

La investigación permitió establecer el nivel de satisfacción de los usuarios de la plataforma, los empleados de la empresa Cloud Solutions Network indican una gran satisfacción con la plataforma ya que redujo el tiempo de trabajo de la persona encargada alrededor de 8 horas, a 2 horas, pudiendo utilizar ese tiempo para otras actividades. Los estudiantes también indicaron su satisfacción y su principal motivo fue el poder validar su certificado sin importar si la empresa deja de existir. Esto demuestra que el uso de la tecnología Blockchain aplicada a los certificados educativos es una alternativa que puede beneficiar a los distintos participantes.

En conclusión, la hipótesis inicial de que la implementación de contratos inteligentes sobre Ethereum es viable para la generación y validación de certificados se confirma en su totalidad. La plataforma permite la creación de certificados dentro de la Blockchain y la verificación de los mismos sin la intervención de un tercero. Ethereum demostró ser una red eficiente y segura para el propósito de la investigación. No obstante, se identificaron alternativas que podrían optimizar los costos operativos, lo que abre nuevas posibilidades para la mejora continua del sistema. El monitoreo constante y el mantenimiento adecuado son fundamentales para asegurar que la plataforma siga siendo eficiente y escalable en el futuro.

10. Recomendaciones

La tecnología Blockchain está en constante evolución desde sus inicios, por eso razón se deben tomar en cuenta las nuevas prácticas y herramientas que surgen conforme más empresas y personas adoptan la tecnología. Es importante que al realizar un proyecto basado en Blockchain se realice una investigación adecuada y un análisis de la documentación, ya que nuevas soluciones pueden surgir, ayudando a mejorar la seguridad y el desarrollo.

También es importante el conocimiento adecuado de las distintas Blockchain que existen. Ethereum fue utilizado dentro de la investigación debido a su popularidad y documentación, debido a que la pionera en los contratos inteligentes, existen una gran cantidad de herramientas y documentación que facilita el desarrollo, pero hoy en día existen más blockchains que permiten esta funcionalidad y pueden ser útiles para alcanzar los objetivos deseados a un menor costo, incrementando los beneficios.

Finalmente es recomendable hacer un análisis de lo que se desea alcanzar, la tecnología Blockchain es una gran alternativa para la generación de certificados digitales y puede ser beneficiosa para otros campos donde es necesario la certificación de documentos, como bienes raíces pero se debe tomar en cuenta los costos, el diseño de la aplicación y los beneficios que se obtendrán para así aprovechar las características que está tecnología brinda y puede llegar a brinda en un futuro.

11. Anexos

Anexo A - Cuadro de ideas

Tabla 10

Cuadro de ideas de tema de tesis

Tema	1. ¿Por qué me gusta el tema?	2. ¿Tengo acceso a la información, a la parte práctica, el tema o personas que manejan la información del tema?	3. ¿Cómo visualizo mi investigación terminada sobre este tema?
Desarrollo de <i>Blockchain</i> es una aplicación basada en web3 para la verificación descentralizada de diplomas digitales en Ethereum	Esta tecnología que se puede aplicar en distintos campos. Esta puede llegar a ser muy beneficiosa y sus características pueden ayudar a crear sistemas descentralizados. Es una tecnología que llama la atención por el alcance que puede llegar a tener y que aún no es utilizada a gran escala.	Tengo acceso a dos empresas que ofrecen cursos tanto a empresas como a personas individuales. Estas empresas no cuentan con una plataforma para la generación de sus diplomas, pero me indican que tengo permiso para trabajar en ello.	Al finalizar un curso, la empresa generará un diploma para el estudiante a través de una aplicación web que se conectará a <i>blockchain</i> . El diploma contendrá un código QR que dirigirá a una página web donde cualquier persona interesada podrá validar la autenticidad del diploma.
Generación de modelo de IA para identificar los clientes de una empresa de eventos según sus preferencias	La inteligencia artificial es una tecnología que tiene muchos alcances y considero que, aplicándola de manera correcta, puede ser de beneficio para las diferentes empresas. En este caso, puede beneficiar a la empresa al generar mayores ingresos y enfocar sus esfuerzos en un grupo	He trabajado para una empresa que tiene diferentes proyectos, y uno de sus clientes maneja información de eventos. He hablado con el encargado del proyecto, y me comentó que tengo permiso para utilizar la información.	El proyecto consistiría en un modelo, el cual sería consultado cada vez que se desee lanzar una campaña de promoción para un evento. Esto retornaría las personas con mayor probabilidad de adquirir boletos para dicho evento.

determinado de personas.

Generación de De igual manera que en el modelo de IA caso anterior, se considera para que la inteligencia artificial determinar el tiempo que un corredor tardará en llegar a la meta

De igual manera que en el caso anterior, se considera para que la inteligencia artificial puede beneficiar a las empresas encargadas de organizar carreras al determinar el tiempo exacto de una carrera, y así reducir el tiempo de bloqueos de las carreteras utilizadas.

La misma empresa está organizando un proyecto para una carrera que se realizará próximamente. Por el momento, no están seguros de si toda la información estará disponible. Sin embargo, me dijo que, si llegan a tenerla, podría utilizar la información.

El proyecto consistiría en un modelo que sería consultado cada vez que se realice una carrera, permitiendo así obtener los tiempos posibles de todos los participantes.

Diseño y Evaluación de Contratos Inteligentes en la Prevención del Uso Indebido de Recetas Médicas Electrónicas

La blockchain permite guardar información que puede ser verificada por distintos nodos, certificando así que la información almacenada no ha sido modificada. Se considera una buena idea aplicar esta tecnología para generar y verificar la veracidad de recetas médicas.

Una farmacéutica está realizando un proyecto para la generación y control de recetas médicas. Como es un proyecto nuevo, me dijeron que podría apoyarlos guardando la información en una blockchain.

El proyecto estaría dividido en dos partes: una aplicación de Ethereum, la cual se encargaría de guardar y leer información de la blockchain, y una interfaz web que permitiría ingresar la información de la receta y consultar, por medio de una API, la aplicación de Ethereum.

ChatGPT para la generación de un modelo de estandarización de direcciones de Guatemala

ChatGPT es una herramienta que facilita el uso de modelos preexistentes y la creación de submodelos que permiten la personalización para tareas específicas. Se considera que el entrenamiento de un modelo que permita estandarizar las direcciones ingresadas por los usuarios beneficiaría a las empresas en la optimización de procesos.

Esa misma empresa tiene diferentes aplicaciones que utilizan direcciones y se han encontrado con el problema de que muchos usuarios las ingresan de manera errónea. Tengo acceso a la información de sus aplicaciones.

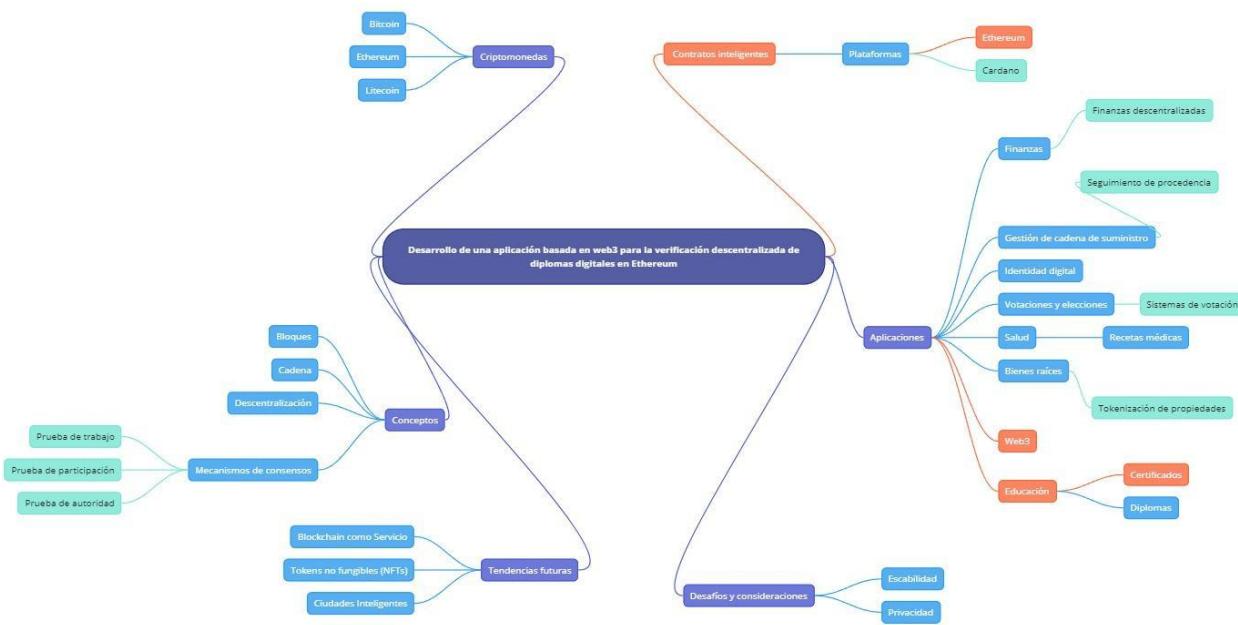
El proyecto consistiría en la creación de un dataset, el cual se utilizaría para la generación de un modelo en ChatGPT, personalizándolo específicamente para la corrección de fechas. Este modelo sería consultado a través de una API que ofrece ChatGPT.

Nota: Descripción de ideas de investigación para tesis.

Anexo B – Mapa mental

Figura 74

Mapa mental de la idea principal



Anexo C – Formatos de las entrevistas

Figura 75

Entrevista a participantes

Entrevista - Cloud Solutions Network **Participantes**

- 1. ¿Ha tenido experiencias previas con certificados digitales? Si es así, ¿cómo fueron?**

- 2. ¿Qué tan confiable considera el proceso actual de emisión de certificados digitales?**

- 3. ¿Está familiarizado con la tecnología Blockchain? ¿Qué opina sobre su posible uso en la validación de certificados?**

- 4. ¿Cree que un sistema basado en Blockchain mejoraría la confianza y seguridad de los certificados digitales?**

Figura 76

Entrevista a empleados de Cloud Solutions Network

Entrevista - Cloud Solutions Network

- 1. ¿Puede describir su experiencia general en la gestión de cursos en linea?**

- 2. ¿Cuáles son los principales retos que enfrenta al gestionar estos cursos?**

- 3. ¿Podría explicar cómo es el proceso de emisión de certificados digitales en su empresa?**

- 4. ¿Qué medidas de seguridad se implementan actualmente para evitar la falsificación de certificados?**

- 5. ¿Qué opinión tiene sobre la posible implementación de Blockchain para la certificación digital?**

- 6. ¿Cree que Blockchain podría resolver algunos de los problemas actuales en el proceso de certificación? ¿Cómo?**

- 7. ¿Qué cambios le gustaría ver en el sistema de gestión y certificación de cursos en los próximos años?**

- 8. ¿Qué impacto cree que tendría una mayor seguridad y eficiencia en la validación de certificados digitales en su trabajo y en la satisfacción de los estudiantes?**

Anexo D – Formato de cuestionario

Figura 77

Formato de cuestionario a participantes

Cuestionario - Cloud Solutions Network **Participantes**

1. **¿Qué tan importante es para usted la autenticidad del certificado digital que recibe al finalizar el curso?**

2. **¿Ha tenido alguna vez problemas para validar un certificado?**

3. **¿Conoce o ha escuchado sobre la tecnología Blockchain para la verificación de certificados?**

4. **¿Qué tan satisfecho está con el tiempo de entrega de su certificado?**

5. **¿Cómo calificaría la seguridad del sistema de certificación actual?**

6. **¿Qué sugerencias tiene para mejorar el proceso de certificación digital?**

Anexo E – Formato de Encuesta

Figura 78

Formato de la encuesta de la empresa

Encuesta - Cloud Solutions Network

- 1. ¿Qué sistemas o herramientas utiliza actualmente para la gestión de cursos y certificación?**

- 2. ¿Qué inconvenientes presentan las herramientas que utiliza actualmente?**

- 3. ¿Cómo describe el proceso actual de emisión de certificados digitales?**

- 4. ¿Ha encontrado problemas con la falsificación de certificados? ¿Con qué frecuencia?**

- 5. ¿Considera viable la implementación de un sistema basado en Blockchain para la certificación de los cursos que gestiona? ¿Por qué?**

- 6. ¿Qué tan satisfecho está con el sistema de certificación actual?**

- 7. ¿Qué mejoras sugeriría para el proceso de certificación y validación de certificados?**

Anexo F - Especificación de Requerimientos de Software (SRS)

En el siguiente enlace se puede encontrar el documento de requerimientos:

<https://dub.sh/OEfHZIM>

12. Glosario

Blockchain: La estructura de datos de la cadena de bloques es una lista ordenada y enlazada hacia atrás de bloques de transacciones. (Antonopoulos y Harding, 2023)

Bloque: Es una estructura de datos contenedora que agrupa transacciones para incluirlas en la blockchain. Está compuesto por un encabezado, que contiene metadatos, seguido de una lista extensa de transacciones que constituyen la mayor parte de su tamaño. El encabezado del bloque tiene un tamaño fijo de 80 bytes, mientras que el total de las transacciones puede alcanzar hasta aproximadamente 4,000,000 bytes. Un bloque completo, con todas las transacciones, puede ser hasta 50,000 veces más grande que el encabezado del bloque.

Bytecode: Es un código objeto de computadora que se compila en código máquina para ser leído por el procesador de la computadora y luego ejecutado por el sistema operativo. (Sheldom, 2022, p. [techtarget.com](https://www.techtarget.com))

Contenedor: Es un entorno de ejecución modificado que aísla procesos y recursos de un sistema operativo, limitando su acceso únicamente a los recursos permitidos explícitamente. Este concepto, que evolucionó a partir de las “jails” de los sistemas operativos UNIX, tiene como objetivo principal proporcionar un entorno aislado para la ejecución segura y eficiente de aplicaciones. (Nickoloff y Kuenzi, 2019)

CORS: El Cross-Origin Resource Sharing (CORS) es un mecanismo basado en encabezados HTTP que permite a un servidor indicar qué orígenes (dominio, esquema o puerto) distintos al suyo propio pueden cargar recursos en un navegador. Además, CORS utiliza un mecanismo mediante el cual los navegadores realizan una solicitud previa, conocida como “preflight”, al servidor que hospeda el recurso de origen cruzado, para verificar que el servidor permitirá la solicitud real. (developer.mozilla.org)

Docker: Es un proyecto de código abierto para construir, distribuir y ejecutar programas. Es un programa de línea de comandos, un proceso en segundo plano y un conjunto de servicios remotos que adoptan un enfoque logístico para resolver problemas comunes de software y simplificar la experiencia de instalar, ejecutar, publicar y eliminar software. Esto se logra utilizando una tecnología del sistema operativo llamada contenedores. (Nickoloff y Kuenzi, 2019)

Etherscan: Etherscan es un explorador de bloques y una plataforma de análisis para Ethereum, una plataforma descentralizada de contratos inteligentes. (etherscan.io)

Framework: Un framework es un conjunto de funcionalidades de software comunes que

proporciona una estructura fundamental para desarrollar una aplicación. Un framework actúa como el soporte esquelético para construir una aplicación. (Spilka, 2019)

GIT: Es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con rapidez y eficiencia. (git-scm.com)

IDE: es una herramienta de software que proporciona un entorno de programación completo para los desarrolladores de software. Este conjunto de herramientas es utilizado para ayudar al desarrollo de software desde un mismo techo. (Platzi, 2019)

JWT: Un JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma de transmitir información de manera segura entre partes como un objeto JSON. Esta información puede ser verificada y confiable porque está firmada digitalmente. Los JWT pueden ser firmados utilizando un secreto (con el algoritmo HMAC) o un par de claves pública/privada usando RSA o ECDSA. (jwt.io)

Metamask: Es la cripto billetera de autocustodia líder. La forma segura y sencilla de acceder a aplicaciones blockchain y web3. (metamask.io)

MVC: Es un patrón de diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Hace hincapié en la separación entre la lógica de negocio del software y su presentación visual. (developer.mozilla.org)

MVVM: Es un patrón de diseño de software que ayuda a separar claramente la lógica de negocio y de presentación de una aplicación de su interfaz de usuario (UI). Mantener una separación limpia entre la lógica de la aplicación y la UI contribuye a resolver numerosos problemas de desarrollo y hace que la aplicación sea más fácil de probar, mantener y evolucionar. (learn.microsoft.com)

Nodo: Es una computadora que participa en una red blockchain. Los nodos mantienen una copia de la blockchain y validan las transacciones y bloques. Los nodos completos validan independientemente cada transacción, aseguran que cumplen con las reglas de consenso y contribuyen a la robustez de la red al permitir transacciones confiables sin depender de terceros. (Antonopoulos y Harding, 2023)

Opcode: El opcode es la parte de una instrucción en lenguaje máquina que especifica qué operación debe realizar la unidad central de procesamiento (CPU). (linfo.org, 2006)

Prueba de trabajo: Es un mecanismo que utiliza un algoritmo hash (como SHA-256) para

resolver problemas computacionales complejos, como encontrar un hash que cumpla con ciertos requisitos. Esto asegura que los bloques añadidos a la blockchain requieran una cantidad considerable de trabajo computacional, dificultando la alteración de los datos. La Prueba de Trabajo también permite que los nodos lleguen a un consenso sobre el estado de la blockchain.

Scrum: Scrum es un proceso empírico, donde las decisiones se basan en la observación, la experiencia y la experimentación. Scrum se sustenta en tres pilares: transparencia, inspección y adaptación. Esto respalda el concepto de trabajar de forma iterativa. Piensa en el empirismo como trabajar a través de pequeños experimentos, aprender de ese trabajo y adaptar tanto lo que estás haciendo como la forma en que lo haces según sea necesario. (scrum.org)

Superset: Es un superconjunto de un lenguaje de programación que agrega funcionalidades no existentes. (Saleem, 2024, p. buttercms.com)

Transacción: Es una instrucción para transferir valor entre usuarios. Se estructura como un libro de contabilidad con entradas (inputs) que gastan fondos y salidas (outputs) que reciben fondos. Las transacciones incluyen firmas digitales para verificar la propiedad de los fondos y asegurar que no han sido gastados previamente. (Antonopoulos y Harding, 2023)

TypeScript: Es un superset de JavaScript, desarrollado y mantenido por Microsoft como un lenguaje de programación de código abierto. TypeScript puede integrarse perfectamente en el código JavaScript existente, añadiendo estructura y detectando errores de seguridad de tipos antes de que afecten el funcionamiento de la aplicación. (Saleem, 2024, p. buttercms.com)

13. Referencia bibliográfica

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal Of Cryptology*, 3(2), 99-111. <https://doi.org/10.1007/bf00196791>

Wright, C. S. (2008). Bitcoin: a Peer-to-Peer electronic cash system. *Social Science Research Network*. <https://doi.org/10.2139/ssrn.3440802>

Frizzo-Barker, J., Chow-White, P. A., Adams, P. R., Mentanko, J., Ha, D., & Green, S. (2020). Blockchain as a disruptive technology for business: A systematic review. *International Journal Of Information Management*, 51, 102029. <https://doi.org/10.1016/j.ijinfomgt.2019.10.014>

Ramamurthy, B. (2020). *Blockchain in Action*. Manning Publications.

Lantz, L., & Cawrey, D. (2020). *Mastering Blockchain*. O'Reilly Media.

Antonopoulos, A. M., & Harding, D. A. (2023). *Mastering Bitcoin*. «O'Reilly Media, Inc.»

Gramajo, J. (2019). La red de falsificadores que funciona a una cuadra de Tribunales.

Soy502. Extraído de: <https://dub.sh/kNt5F94>

Boche, E. (2023). Exdecano investigado por falsificación de títulos compite por una curul en el Congreso. Prensa Libre. Extraído de: <https://dub.sh/CcCPNiU>

Nganga, G. (2024). Flawed verification systems to blame for forged certificates. University World News Africa Edition. Retrieved from: <https://dub.sh/d83LIMS>

TimesLive. (2024). Fake certificates and bogus colleges: South Africa. Retrieved from: <https://dub.sh/wpIHscJ>

Mathrubhumi News. (2024). US Consulate's complaint: Ernakulam woman arrested for issuing fake educational certificates. Retrieved from: <https://dub.sh/xS2IPpY>

Sowek, M. (2024). Over 2,000 Public Servants Were Hired Using Fake Certificates, PSC Reveals. Capital News. Retrieved from <https://dub.sh/8Y4ArRn>

THE TIMES OF INDIA. (2023). AP man forges papers to get US student visa, arrested. Retrieved from: <https://dub.sh/8AMz12N>

Ramos Martinez, P. L. (2023). Mafias tramitan certificados de estudios desde el propio sistema del Minedu, según 'Punto Final'. RPP Noticias. Recuperado de: <https://dub.sh/7CCpAmq>

Ruiz, C. (2023). Fraude educativo: profesores falsificaban títulos para dar clases en colegios nacionales. Infobae._Recuperado de: <https://dub.sh/8Dej1qf>

Swan, M. (2015). Blockchain: Blueprint for a New Economy. «O'Reilly Media, Inc.»

Antonopoulos, A. M., & Wood, G., PhD. (2018). Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly Media.

Mohanty, D. (2018). Ethereum for Architects and Developers: With Case Studies and Code Samples in Solidity. Apress.

Wu, X., Zou, Z., & Song, D. (2023). Learn Ethereum: A practical guide to help developers set up and run decentralized applications with Ethereum 2.0. Packt Publishing Ltd.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions On Information Theory, 22(6), 644-654. <https://doi.org/10.1109/tit.1976.1055638>

Instituto Tlaxcalteca para la Educación de los Adultos. (2023). Exhorta ITEA a no comprar certificados falsos. Recuperado de: <https://dub.sh/BYTe2Lr>

Gramajo, J. (2019). La red de falsificadores que funciona a una cuadra de Tribunales. Soy502. Recuperado de: <https://dub.sh/ZaUhrNo>

Boche, E. (2023). Exdecano investigado por falsificación de títulos compite por una curul

en el Congreso. Prensa Libre. Recuperado de: <https://dub.sh/4MK8xq7>

Redacción Entre Estudiantes. (2021). Los cursos de formación online crecen un 900% durante la pandemia. Entre Estudiantes. Recuperado de <https://dub.sh/HnxT8ZH>

Zorrilla, A. (2023). ¿Por qué estudiar en línea es una tendencia en franco crecimiento? Campus digital idyd. Recuperado de <https://dub.sh/vbeDSU9>

Garduño, M. (2021). Pandemia detona 300% usuarios de cursos en línea. Forbes México. Recuperado de: <https://dub.sh/Qb2rDPR>

Díaz, A. (2023). Localizan títulos y licencias de conducir en blanco para poder ser falsificados en zona 1. TV Azteca Guate. Recuperado de <https://dub.sh/syDWmQ3>

De León, E. (2021). Universidad Da Vinci despide a directivo señalado de certificados falsificados. Soy502. Recuperado de <https://bit.ly/4bgYIsa>

Boche, E. (2023). Exdecano investigado por falsificación de títulos compite por una curul en el Congreso. Prensa Libre. Recuperado de <https://bit.ly/4c9Lm2d>