



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA
ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN
CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE
REGISTROS**

Marvin Emmanuel Pivaral Orellana

Anicka Michelle Cercado Figueroa

Asesorado por el Ing. Marlon Antonio Pérez Turk

Guatemala, noviembre de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA
ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN
CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE
REGISTROS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MARVIN EMMANUEL PIVARAL ORELLANA

ANICKA MICHELLE CERCADO FIGUEROA

ASESORADO POR EL ING. MARLON ANTONIO PÉREZ TURK

AL CONFERÍRSELE EL TÍTULO DE

INGENIEROS EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés De La Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. Miguel Ángel Cancinos Rendón
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE REGISTROS

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería (su Escuela), con fecha julio de 2019.



Anicka Michelle Cercado Figueroa



Marvin Emmanuel Pivaral Orellana

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL. 24439500 EXT. 1534

Guatemala, 22/mayo/2020

Ingeniero Carlos Azurdia
Coordinador Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería, USAC
Presente

Estimado Ing. Azurdia:

Por este medio informo que he revisado y aprobado el Trabajo de Tesis titulado: "BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE REGISTROS", de los estudiantes Marvin Emmanuel Pivaral Orellana (Carnet número 201213587), quien se identifica con DPI número 2488 94609 0101 y Anicka Michelle Cercado Figueroa (Carnet número 201212859), quien se identifica con DPI número 2215 58446 0101.

Con base en la evaluación realizada hago constar que he evaluado la calidad, validez, pertinencia y coherencia de los resultados obtenidos en el trabajo presentado, por lo cual el trabajo evaluado cuenta con mi aprobación.

Agradeciendo su atención y deseándole éxitos en sus actividades profesionales me suscribo.

Atentamente,

"Id y Enseñad a Todos"

MARLON ANTONIO PEREZ TURK
INGENIERO EN CIENCIAS Y SISTEMAS
COLEGIADO No. 4492

MA. Ing. Marlon Antonio Pérez Türk
Asesor
Colegiado No. 4492



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 29 de mayo del 2020

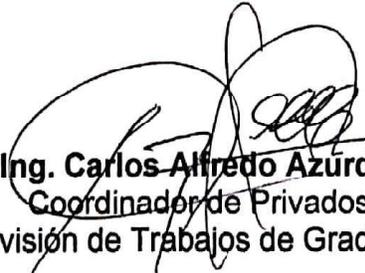
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **MARVIN EMMANUEL PIVARAL ORELLANA** con carné 201213587 y CUI 2488 94609 0101 y **ANICKA MICHELLE CERCADO FIGUEROA** con carné 201212859 y CUI 2215 58446 0101, titulado: "BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE REGISTROS", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azúrdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



SISTEMAS
Y
CIENCIAS
EN
INGENIERÍA
DE
ESCUELA

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE REGISTROS”**, realizado por los estudiantes, MARVIN EMMANUEL PIVARAL ORELLANA y ANICKA MICHELLE CERCADO FIGUEROA aprueba el presente trabajo y solicita la autorización del mismo.*



Digitally signed by Carlos
Gustavo Alonzo
DN: 2.5.4.13=Profesional
Titulado, c=GT, l=Guatemala /
Escuela de Ingeniería en Ciencias y Sistemas
Via 5 3-65
zona 4 Ed. El Angel 5to nivel of
52, 2.5.4.20=22347420, ou=NA,
o=NA, title=Ingeniero en
Ciencias y Sistemas Colegiado.
6358, serialNumber=2278 03167
0101, 2.5.4.45=29020980,
2.5.4.27=06/03/79,

Msc. Carlos Gustavo Alonzo
Director

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 06 de noviembre 2020

DTG. 359.2020.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **BLOCKCHAIN DE ETHEREUM COMO BASE DE DATOS ACADÉMICA PARA ESTUDIANTES DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, UTILIZANDO UNA APP PARA GESTIÓN Y CONSULTA DE REGISTROS**, presentado por los estudiantes universitarios: **Marvin Emmanuel Pivaral Orellana y Anicka Michelle Cercado Figueroa**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Inga. Anabela Cordova Estrada
Decana

Guatemala, noviembre de 2020

AACE/asga

ACTO QUE DEDICO A:

- Dios** Por su amor y gracia hacia mí, también por abrirme las puertas y poner a las personas correctas a lo largo de mi carrera.
- Mis padres** Marvin Pivaral y Berta Orellana, por su amor y apoyo incondicional en todo momento.
- Mis hermanos** Beryit, Idony, Jonathan, Gabriela Pivaral, por estar siempre conmigo en el transcurso de esta carrera.
- Mis amigos** Por estar en los momentos más difíciles de la carrera y motivarme a no rendirme y seguir adelante.

Marvin Emmanuel Pivaral Orellana

ACTO QUE DEDICO A:

Dios	Por su infinito amor y misericordia hacia mí, también por brindarme la fortaleza y paciencia a lo largo de mi carrera.
Mis padres	Byron de León y Anicka Figueroa, por su amor y apoyo incondicional en todo momento.
Mi hijo	Byron Cercado por alegrar cada uno de mis días y ser ese motor que me inspira a ser mejor.
Mis hermanas	Raissa y Debbie de León, por su valioso apoyo y comprensión a lo largo de mi carrera.
Mis familiares	Por estar presentes en cada etapa a lo largo de mi vida, y brindarme su ayuda cuando la he necesitado.

Anicka Michelle Cercado Figueroa

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser la casa de estudios que me brindó el conocimiento necesario para mi formación profesional.
Mi familia	Por brindarme todos los recursos necesarios para cumplir este sueño.
Amigos	Por hacer de los momentos divertidos y entretenidos durante el transcurso de la carrera.
Marlon Turk	Por su apoyo y conocimientos brindados para el trabajo de graduación.

Marvin Emmanuel Pivaral Orellana

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser parte de mi formación como profesional en mi carrera.
Mi familia	Por motivarme y brindarme lo necesario para culminar esta carrea.
Amigos y compañeros de estudio	Por los buenos momentos y ayuda mutua en cada reto.
Marlon Turk	Por el valioso aporte en el trabajo de graduación.

Anicka Michelle Cercado Figueroa

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XV
OBJETIVOS	XVII
INTRODUCCIÓN	XIX
1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA	1
1.1. Blockchain	1
1.1.1. Definición	1
1.1.2. Características	2
1.1.2.1. Descentralizado	2
1.1.2.2. Anónimo	4
1.1.2.3. Inmutable	4
1.2. Ethereum	5
1.2.1. Moneda	6
1.2.2. Bloque Ethereum	7
1.3. Smart Contracts	8
1.3.1. Solidity	9
1.4. DApp	10
1.4.1. Base de datos	10
1.4.2. Características	11
1.4.2.1. Descentralización	11
1.4.2.2. Código abierto (<i>Open Source</i>)	11
1.4.2.3. Blockchain	11

2.	IDENTIFICACIÓN DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN ..	13
2.1.	Antecedentes	13
2.2.	Descripción del problema	14
2.3.	Mercado objetivo	15
2.4.	Benchmark de la aplicación.....	15
2.4.1.	Portal Ingeniería USAC	15
2.4.1.1.	Proceso para generar certificación de cursos aprobados	16
2.4.1.1.1.	Paso 1	16
2.4.1.1.2.	Paso 2	17
2.4.1.1.3.	Paso 3	17
2.4.1.1.4.	Paso 4	18
2.4.1.2.	Proceso para visualizar el listado de cursos aprobados	20
2.4.1.2.1.	Paso 1	20
2.4.1.2.2.	Paso 2	21
2.4.2.	GradeChain	21
2.4.2.1.	Página de inicio	22
2.4.2.2.	Certificación de cursos aprobados	22
2.4.2.3.	Certificación de cierre de pensum.....	23
2.4.3.	Métricas de comparación.....	24
2.4.3.1.	Tiempo para gestionar una certificación de cursos aprobados.....	28
2.4.3.2.	Estudiantes inscritos	29
2.4.4.	Análisis de resultados.....	30
3.	DISEÑO DE LA APLICACIÓN GRADECHAIN BAJO LA NECESIDAD IDENTIFICADA.....	33
3.1.	Prototipos	33

3.1.1.	Diagrama general de la solución	33
3.1.2.	Página principal	35
3.1.3.	Estudiante	36
3.1.4.	Certificación de cierre de pensum	37
3.1.5.	Login	38
3.1.6.	Registrar estudiante	39
3.1.7.	Registrar curso.....	40
3.1.8.	Registrar nota	41
3.2.	Diseño intuitivo y usabilidad.....	42
3.2.1.	Redireccionamiento	42
3.2.2.	Menú de página sin autenticación	43
3.2.3.	Menú de página con autenticación.....	43
3.2.4.	Diseño responsivo	44
3.3.	Arquitectura del sistema	47
4.	DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN	49
4.1.	Frontend	49
4.1.1.	Página principal	49
4.1.2.	Página de inicio de sesión	51
4.1.3.	Página de consulta de estudiante	52
4.1.4.	Página de registro de estudiante.....	53
4.1.5.	Certificado cierre de pensum	55
4.1.6.	Página de registro de curso	56
4.1.7.	Página de registro de nota	58
4.2.	Requisitos.....	59
4.2.1.	Herramientas	59
4.2.1.1.	GIT.....	59
4.2.1.2.	GitHub.....	61

4.2.2.	Hardware.....	63
4.2.2.1.	AWS	63
	4.2.2.1.1. EC2	63
4.2.3.	Software	63
4.2.3.1.	Backend.....	64
	4.2.3.1.1. Node.js	64
	4.2.3.1.2. Web3	65
	4.2.3.1.3. Infura	65
4.2.3.2.	Frontend	66
	4.2.3.2.1. AdonisJs	66
4.2.4.	Metodología SCRUM.....	67
4.2.4.1.	Planificación	67
4.2.4.2.	Cronograma.....	68
4.2.4.3.	Product Backlog.....	69
4.2.4.4.	Sprint 1	71
	4.2.4.4.1. Sprint Backlog.....	71
4.2.4.5.	Sprint 2	74
	4.2.4.5.1. Sprint Backlog.....	74
4.2.5.	Tutorial de desarrollo y referencias.....	76
4.2.5.1.	Vista general del proyecto	76
	4.2.5.1.1. Carpeta APP	78
	4.2.5.1.2. Carpeta Contracts.....	79
	4.2.5.1.3. Carpeta HTTP.....	80
	4.2.5.1.4. Carpeta Public	81
	4.2.5.1.5. Carpeta View	82
4.2.6.	Consideraciones de implementación	83

CONCLUSIONES85
RECOMENDACIONES87
BIBLIOGRAFÍA.....89

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Gráfico de servidores	3
2.	Diagrama de redes.....	3
3.	Logo de Ethereum.....	6
4.	Bloque de Ethereum	8
5.	Arquitectura DApp.....	10
6.	Portal de la Facultad de Ingeniería.....	16
7.	Inicios de sesión Facultad de Ingeniería	17
8.	Boleta de pago.....	18
9.	Certificado de cursos aprobados.....	19
10.	Inicio de sesión Facultad de Ingeniería	20
11.	Listado de cursos aprobados	21
12.	Prototipo login.....	22
13.	Prototipo cursos aprobados	23
14.	Prototipo de certificación de cierre de pensum.....	24
15.	Pregunta 1	25
16.	Pregunta 2	27
17.	Diagrama de la solución.....	35
18.	Prototipo página principal.....	36
19.	Prototipo página de estudiante.....	37
20.	Prototipo de certificación cierre de pensum en PDF.....	38
21.	Prototipo página login.....	39
22.	Prototipo página registrar estudiante.....	40
23.	Prototipo página registrar curso	41

24.	Prototipo página registrar nota.....	42
25.	Menú de página sin autenticación.....	43
26.	Cabecera de página con autenticación.....	44
27.	Diseño responsivo 1.....	45
28.	Diseño responsivo 2.....	46
29.	Diseño responsivo 3.....	46
30.	Diseño responsivo 4.....	47
31.	Arquitectura del sistema: modelo, vista, controlador.....	48
32.	Página principal de la aplicación.....	50
33.	Página inicio de sesión de la aplicación.....	51
34.	Página de consulta de estudiante de la aplicación.....	52
35.	Página de registro de estudiante de la aplicación.....	54
36.	Documento PDF certificado de cierre de pensum de la aplicación.....	55
37.	Página de registro de curso de la aplicación.....	57
38.	Página de registro de nota 1 de la aplicación.....	58
39.	Esquema de ramas.....	60
40.	Comandos de Git.....	61
41.	Perfil de GitHub.....	62
42.	Detalle de cambios en GitHub.....	62
43.	Logo Node.js.....	64
44.	Logo Infura.....	65
45.	Logo AdonisJs.....	67
46.	Cronograma de actividades.....	69
47.	Vista general del proyecto.....	78
48.	Carpeta APP.....	79
49.	Carpeta Contracts.....	80
50.	Carpeta HTTP.....	81
51.	Carpeta Public.....	82
52.	Carpeta View.....	83

TABLAS

I.	Frecuencia de certificaciones solicitadas.....	26
II.	Promedio de certificaciones solicitadas.....	26
III.	Copias solicitadas	27
IV.	Promedio copias por persona.....	28
V.	Actividades por minuto	28
VI.	Estudiantes inscritos por año	29
VII.	Promedio de estudiantes inscritos por año.....	29
VIII.	Comparativa.....	31
IX.	Planificación.....	68
X.	Product Backlog GradeChain.....	69
XI.	Sprint 1 Backlog	71
XII.	Sprint 2 Backlog	74

LISTA DE SÍMBOLOS

Símbolo	Significado
Eth	Ether
I18n	I – eighteen letters -N

GLOSARIO

<i>API</i>	Interfaz de programación de aplicaciones.
<i>Blockchain</i>	Lista creciente de registros, llamados bloques, que están vinculados mediante criptografía.
<i>Criptomoneda</i>	Medio digital de intercambio que utiliza criptografía fuerte para asegurar las transacciones.
<i>DAPP</i>	Aplicación que ejecuta su código <i>Backend</i> en una red descentralizada punto a punto.
<i>Ether</i>	Moneda Ethereum.
<i>Ethereum</i>	Plataforma de cómputo distribuido de código abierto, pública, basada en <i>Blockchain</i> y un sistema operativo con funcionalidad de contrato inteligente.
<i>Framework</i>	Patrón en el desarrollo de una aplicación.
<i>P2P</i>	Red de punto a punto, es una arquitectura de aplicación distribuida que divide tareas o cargas de trabajo entre pares.

Smart Contract

Es un protocolo de computadora destinado a facilitar, verificar o hacer cumplir digitalmente la negociación o ejecución de un contrato.

RESUMEN

En el primer capítulo se describen las tecnologías utilizadas para construir la Dapp GradeChain, estas tecnologías son: Blockchain y sus características, seguido de Ethereum, que es una capa de tecnología construida sobre Blockchain, en ella se almacenará la data de GradeChain a través de bloques de Ethereum y *Smart Contracts*, los cuales fueron codificados en el lenguaje Solidity; finalmente, se representan las DApps, éstas representan las aplicaciones que interactúan con el usuario final para la administración de los bloques de Ethereum.

El segundo capítulo presenta y establece la razón por la cual el proceso GradeChain simplificará y mejorará la actual gestión para los usuarios finales, esto se establece a través de los antecedentes, descripción del problema, concretar el mercado objetivo para finalmente, presentar un benchmarking de la aplicación.

En el tercer capítulo se muestran los prototipos diseñados para GradeChain, señalando y nombrando uno a uno sus objetos, adicional a esto, se describe la funcionalidad de cada objeto y página, también la arquitectura propuesta; así mismo, los atributos de calidad que lo hacen intuitivo y fácil de utilizar para los usuarios finales.

El cuarto capítulo presenta y describe las herramientas utilizadas, software y hardware para desarrollar GradeChain; así mismo la implementación de la metodología SCRUM en el proyecto, también se realiza un tutorial de usuario y desarrollo con sus consideraciones.

OBJETIVOS

General

Diseñar una aplicación descentralizada para gestionar el control de los estudiantes y el listado de cursos aprobados de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.

Específicos

1. Almacenar la información de los estudiantes y los cursos aprobados sobre la cadena de bloques de Ethereum por medio de un *Smart Contract*.
2. Crear una aplicación de usuario que interactúe con el *Smart Contract* para permitir, de manera pública, que cualquiera pueda consultar los datos de los estudiantes y el listado de cursos aprobados.
3. Innovar la forma en que se almacena la información, garantizando acceso público y seguro.

INTRODUCCIÓN

La Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, cuenta actualmente con su propia infraestructura para administrar el almacenamiento de la información de los estudiantes que se inscriben en ella y los cursos que estos han aprobado.

Como constancia de la información de los estudiantes, se extienden certificados impresos, los cuales son membretados y firmados por los responsables de la facultad, los certificados se obtienen siguiendo una serie de pasos establecidos.

Los certificados extendidos por la facultad resultan muy útiles, ya que por lo general cuando una persona solicita un nuevo trabajo o cualquier tipo de trámite académico, se necesita que la persona cumpla con ciertos requisitos, y esto se valida por medio de un certificado de cursos aprobados o un certificado de cierre de pensum.

Como anteriormente se ha mencionado, actualmente la única forma de autenticar la información es por medio de certificaciones físicas las cuales resultan ser no seguras y fáciles de falsificar, restando credibilidad a estos documentos.

Es importante resaltar que son necesarios recursos como el tiempo del estudiante y las hojas para imprimir certificaciones, estos recursos se hacen significativos cuando se evalúa la cantidad total de estudiantes respecto a la cantidad de certificaciones adquiridas.

1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA

En este capítulo se describen las tecnologías que se estudiaron para utilizarlas en GradeChain como software; entre estas se listan: Blockchain y sus características, seguido de Ethereum esta es una tecnología desarrollada sobre la tecnología Blockchain, en esta se almacenará la data de GradeChain, bloque Ethereum, *Smart Contracts*, estos fueron codificados en el lenguaje Solidity y por último las DApps, que son las aplicaciones que interactúan con el usuario final para la administración de los bloques de Ethereum.

1.1. Blockchain

Es una tecnología desarrollada con el fin de mantener un sistema descentralizado en el cual los participantes puedan mantener su anonimato realizando transacciones comúnmente de valor.

1.1.1. Definición

Blockchain también conocido como cadena de bloques es una estructura de datos creada por una persona o grupos de personas denominada Satoshi Nakamoto, se desconoce al creador o creadores originales, la cadena de bloques es considerada como un libro mayor de contabilidad que no puede ser modificado, pero que permite establecer reglas sobre esta estructura para especificar qué transacciones pueden suceder o no.

1.1.2. Características

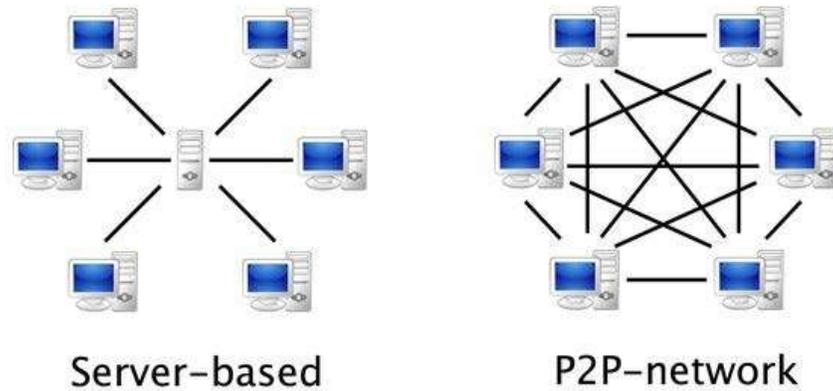
Entre las características de Blockchain se pueden agrupar en 3 específicamente que serían:

- Descentralizado
- Anónimo
- Inmutable

1.1.2.1. Descentralizado

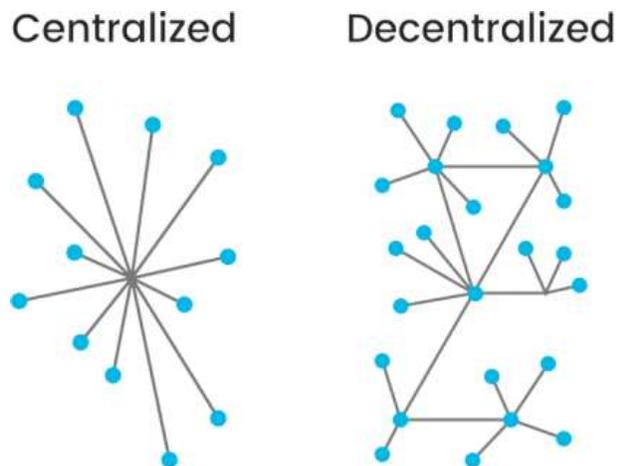
La cadena de bloques es considerada descentralizada porque no existe una persona o ente que administre la transaccionalidad y almacene la información, no hay dueño del Blockchain, que es una base de datos compartida donde cada persona que forma parte de la comunidad tiene una copia independiente en su computador que es llamado nodo, y la misma comunidad se encarga de validar y transmitir las transacciones, por esto se considera que Blockchain es una red distribuida donde los nodos se comunican por medio de una red P2P. En las siguientes imágenes se puede observar el parecido de una red P2P con una red descentralizada, y como una red Cliente - Servidor (Server Based) es parecida a una red centralizada.

Figura 1. **Gráfico de servidores**



Fuente: Quora. *What is a peer to peer connection?* www.quora.com/What-is-a-peer-to-peer-connection. Consulta: octubre de 2019.

Figura 2. **Diagrama de redes**



Fuente: FORESTLYN. *Five Ways Blockchain Technology is Transforming the Marketing, Digital Media & Advertising Industries.* www.forestlyn.com/blog/2019/5/4/five-ways-blockchain-technology-is-transforming-the-marketing-digital-media-and-advertising-industries. Consulta: octubre de 2019.

1.1.2.2. Anónimo

Blockchain se considera anónimo ya que para participar dentro de la red es necesario tener una cuenta, que en la mayoría de Blockchain existentes es conocida como Wallet, la Wallet, como su nombre lo indica, es una cartera donde se tiene asociado un balance y con ella se gestionan las transacciones dentro de la red, esta Wallet tiene asociado una dirección hash que dentro de la red es única e irreplicable, por lo que no es necesario tener información personal registrada en ella, gracias a ello dentro de la red cada Wallet es una dirección hash y no se puede saber con exactitud quién es el propietario de dicha dirección. Para la creación de una Wallet, en la mayoría de Blockchain, basta con ingresar una contraseña que se encriptará junto a ella, y esta genera una llave pública y privada utilizando sus aplicaciones clientes para acceder a ellas, donde la llave pública es la dirección hash con la cual son vistos en la comunidad, y la llave privada es la que se utiliza para autorizar transacciones con una Wallet.

1.1.2.3. Inmutable

Blockchain es considerado inmutable, es decir, casi imposible de hackear, esto se logra gracias a que su estructura de datos, donde cada bloque que es agregado a la red, es encriptado, creando así un puntero hash entre cada uno de los bloques con el valor encriptado de los bloques siguientes y anteriores, esto quiere decir que para poder modificarlo se tendría que modificar todos los bloques anteriores a un bloque, la mayoría de Blockchain utilizan un encriptado de 256 bits, además de encriptar cada uno de los bloques, también se utiliza el algoritmo Merkle Tree que se usa para encriptar cada una de las transacciones contenidas en los bloques, siendo así más difícil de hackear, como está descentralizada, se estima que para poder hackear toda la red, sería necesaria modificar más del 50 % de los nodos y como cada uno de esos nodos es una

computadora diferente con diferentes medios de seguridad lo hace difícil de hackear.

1.2. Ethereum

Es una plataforma desarrollada sobre la tecnología de blockchain, con una capa extra conocida como “*Smart Contracts*” o “Contratos Inteligentes”, lo que permite a cualquier persona poder desarrollar aplicaciones descentralizadas basadas en estos contratos. Ethereum cuenta con su propia criptomoneda, conocida como “Ether” o “Eth” y se puede transferir entre las Wallet de esta red. Esta criptomoneda es utilizada como recompensa para los nodos encargados de minar y realizar los cálculos para registrar transacciones.

Ethereum fue desarrollado por Vitalik Buterin, por medio de un financiamiento colectivo, el sistema se publicó el 30 de julio del 2016.

Ethereum, al estar basado en la tecnología Blockchain, es considerado un sistema descentralizado, y éste utiliza una máquina virtual en su sistema conocido como “Ethereum Virtual Machine” o “EVM”, esta máquina virtual es la encargada de ejecutar el código a un nivel intermedio, también conocida como bytecode, de los contratos existentes en su sistema.

Los programas que se ejecutan en la red, conocidos como contratos inteligentes, son escritos en su propio lenguaje de alto nivel conocido como Solidity.

Figura 3. **Logo de Ethereum**



Fuente: BUTERIN, Vitalik. *Ethereum*. www.coinhouse.com/learn/ethereum.

Consulta: octubre de 2019.

1.2.1. Moneda

La criptomoneda del sistema de Ethereum es conocida como “Ether” o “Eth”, y es la que se usa para impulsar aplicaciones distribuidas. La criptomoneda es utilizada por personas dentro de la red para realizar pagos o como recompensa a las máquinas que se encargan de ejecutar las operaciones dentro de la red. El Ether asegura que la red se mantenga activa por medio de recompensar a las personas que prestan sus máquinas para ejecutar las operaciones, estas máquinas son conocidas dentro de la red de Ethereum como mineros.

1.2.2. Bloque Ethereum

Como se mencionó, Blockchain es una estructura de datos y como Ethereum utiliza esta tecnología cada uno de sus bloques contiene la siguiente información:

- Peso o número de bloque
- Tiempo en que fue creado
- Número de transacciones internas
- Hash resultante de su encriptación
- Hash del bloque padre o bloque anterior
- Hash resultante del Merkle Tree
- Dirección de quien lo mino
- Dificultad
- Dificultad total
- Tamaño en bytes
- Gas usado
- Límite de gas
- Recompensa por bloque en Ether
- Extra data

Figura 4. Bloque de Ethereum

The screenshot shows the Etherscan website interface. At the top, there is a search bar with the text "Search by Address / Txhash / Block / Token / Ens" and a "GO" button. Below the search bar are navigation links: HOME, BLOCKCHAIN, TOKENS, RESOURCES, and MORE. The main content area displays "Block #6360049" and a breadcrumb trail: Home / Blocks / Block Information. There are two tabs: "Overview" (selected) and "Comments". Below the tabs is a "Block Information" section with a list of details:

Height:	6360049
TimeStamp:	22 secs ago (Sep-19-2018 10:55:01 AM +UTC)
Transactions:	53 transactions and 9 contract internal transactions in this block
Hash:	0x24c15cc8feacb5df25c5eec85689ff87af0b1cf0c3c4a5cafd5f1a626c1a149
Parent Hash:	0x23d762c35d7ad64aa507aa82fa1f35b9f5712cb5e55df55afd3294174cd0500a
Sha3Uncles:	0x1dcc4de8dec75d7aab85b567b6cc41ad312451b948a7413f0a142fd40d49347
Mined By:	0x829bd824b016326a401d083b33d092293333a830 (F2Pool_2) in 7 secs
Difficulty:	3,176,280,266,416,612
Total Difficulty:	6,719,154,199,451,356,717,750
Size:	11834 bytes
Gas Used:	8,002,010 (99.93%)
Gas Limit:	8,007,768
Nonce:	0xd6347eb8091ec41c
Block Reward:	3.08173696411917823 Ether (3 + 0.08173696411917823)
Uncles Reward:	0
Extra Data:	七彩神仙鱼 (Hex:0xe4b883e5bda9e7a59ee4bb99e9b1bc)

Fuente: BUTERIN, Vitalik. *Etherscan*. www.ropsten.etherscan.io/block/6360049.

Consulta: octubre de 2019.

1.3. Smart Contracts

Un contrato inteligente es el encargado ejecutar un conjunto de instrucciones dentro de la red Ethereum, donde pueden participar una o más partes, validando que se cumpla un conjunto de condiciones para la ejecución de una o varias instrucciones.

Cuando las condiciones se cumplen, el contrato ejecutará el conjunto de instrucciones, lo que se denomina cláusula.

Un *Smart Contract* puede ser creado y ejecutado por personas, pero también lo pueden utilizar máquinas o programas que interactúan con él. El *Smart Contract* no depende de autoridades, por naturaleza es visible para todos y una vez éste se crea dentro de la red, su lógica no puede ser modificada.

1.3.1. Solidity

Es un lenguaje de alto nivel desarrollado para la elaboración de contratos inteligentes, tiene similitudes con otros lenguajes como C++, Python y JavaScript, este lenguaje está diseñado para funcionar dentro de la máquina virtual de Ethereum, conocida como Ethereum Virtual Machine (EVM).

EL lenguaje permite usar herencia, bibliotecas y estructuras de datos definidas por el usuario.

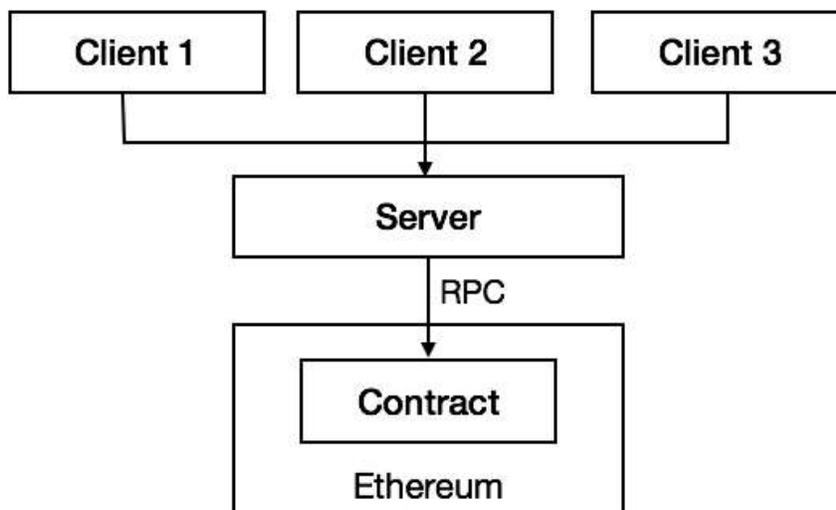
Tipos de datos:

- Booleanos
- Enteros
- Dirección
- Arreglos
- Cadenas
- Bytes

1.4. DApp

DApp conocida como “*decentralized application*”, es una aplicación descentralizada, ya que esta está construida o utiliza Blockchain dentro de su estructura para su funcionamiento, normalmente es considerada una aplicación de frontend que consume los servicios o apis de una red descentralizada. Esta puede ser elaborada para dispositivos móviles, computadoras, o cualquier otra plataforma.

Figura 5. **Arquitectura DApp**



Fuente: LIN, Zih-Ci. *gRPC in DApp Architecture*. www.medium.com/getamis/gRPC-in-dapp-architecture-8c34125356c7. Consulta: octubre de 2019.

1.4.1. Base de datos

Normalmente la información que utilizan las aplicaciones es almacenada en disco duros o unidades de estados sólidos entre otros, estos pueden ser físicos o virtuales, por medio de la nube. La información de las aplicaciones

descentralizadas se almacena en la cadena de bloques, utilizando así un sistema inmutable y descentralizado, donde la información no puede ser modificada tan fácilmente sin que se cumplan con las condiciones establecidas dentro de la red o un contrato inteligente, además al utilizar la red de cadena de bloques cuenta con un carácter distribuido ya que existen diversas copias de la red.

1.4.2. Características

A continuación, se presentan las principales características de una DApp:

1.4.2.1. Descentralización

Todo funciona de manera autónoma, sin que exista un ente que la controle, dejándolo en manos de toda la comunidad.

1.4.2.2. Código abierto (*Open Source*)

El código fuente de la aplicación descentralizada está abierta a posibles modificaciones y mejoras por los usuarios.

1.4.2.3. Blockchain

Los datos deben ser almacenados utilizando la cadena de bloques.

2. IDENTIFICACIÓN DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Este capítulo tiene como objetivo establecer la razón por la que el proceso GradeChain simplificará y mejorará la actual gestión para los usuarios finales, esto se establece a través de los antecedentes, descripción el problema, concretar el mercado objetivo y por último realizando un benchmarking de la aplicación.

2.1. Antecedentes

Una cadena de bloques o Blockchain es una base de datos, cuando se requiere crear o modificar información, se crea un nuevo bloque y este se replica en distintos ordenadores con información encriptada. La descripción anterior cumple con las 3 características principales del Blockchain: (1) descentralizado, (2) anónimo, (3) inmutable.

La cadena de bloques más popular es el Bitcoin, un bloque de Bitcoin contiene información sobre balance de saldos, es decir, un conjunto de transacciones. La cadena de bloques de Ethereum cuenta con contratos inteligentes como valor agregado, un contrato inteligente contiene código criptográfico y se ejecuta para distintos fines como acuerdos entre usuarios, identidad, entre otros.

La identidad basada en Blockchain protege la información de un individuo por medio de encriptación y codificación, esta información puede ser verificada en cualquier parte del mundo, por ejemplo: pasaportes digitales, certificados de

nacimiento, matrimonio, defunción, identificación personal. Actualmente, existen compañías que trabajan en identidad utilizando el blockchain de Ethereum, tal es el caso de uport (<https://www.uport.me/>).

Descrito lo anterior, en la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, la información académica del estudiante es almacenada en bases de datos propias de la facultad, se propone facilitar la consulta y gestionar información a interesados por medio de una DApp, utilizando la cadena de bloques de Ethereum.

2.2. Descripción del problema

En la Facultad de Ingeniería, Universidad de San Carlos de Guatemala, la data académica y profesional es almacenada en un sistema manejador de bases de datos (DBMS), con servidores *On-Premise*, y la data de los cursos de cada estudiante se restringe a cierta cantidad de usuarios, por ejemplo los catedráticos gestionan notas y los estudiantes visualizan las misma a través de un sistema con autenticación previa, por lo tanto, no existe una certificación digital pública con la que el estudiante se pueda respaldar.

La aplicación GradeChain permitirá que la información académica de los estudiantes sea de dominio público utilizando la cadena de bloques de Ethereum, por medio de los *Smart Contracts*, los cuales se programaron para ser inmutables, es decir, que los registros de notas de los estudiantes no pueden ser editados de ninguna manera, solo creados y visualizados dentro de la misma red por medio de una aplicación descentralizada, por las características anteriores los datos persisten a través del tiempo, ya que hay distintas réplicas de la data en la nube.

Por lo tanto, utilizar una base de datos basada en una cadena de bloques soluciona la pérdida de información a través del tiempo, así como provee certificaciones públicas de la información académica de los estudiantes.

2.3. Mercado objetivo

La Facultad de Ingeniería de la Universidad de San Carlos de Guatemala que es la responsable de extender las certificaciones académicas a: estudiantes, catedráticos, personas a cargo de procesos académicos o reclutamiento y otras universidades.

2.4. Benchmark de la aplicación

Benchmarking es un proceso en el cual se comparan estrategias que la competencia implementa, en base a esto, se definen y aplican mejoras y se aplican para una organización. A partir de lo anterior se procede a realizar el benchmarking del portal ingeniería USAC (<https://portal.ingenieria.usac.edu.gt/>) y la aplicación GradeChain.

2.4.1. Portal Ingeniería USAC

Portal web utilizado por los estudiantes para poder generar certificaciones de cursos aprobados.

2.4.1.1. Proceso para generar certificación de cursos aprobados

Actualmente en el portal Ingeniería USAC para generar una certificación de cursos aprobados es necesario seguir una serie de pasos, que se detallan a continuación:

2.4.1.1.1. Paso 1

Ingresar a la página de inicio del portal Ingeniería USAC <https://portal.ingenieria.usac.edu.gt/>, seguidamente buscar la opción de iniciar sesión.

Figura 6. Portal de la Facultad de Ingeniería



Fuente: Facultad de ingeniería, Universidad San Carlos de Guatemala. *Portal de Ingeniería*.

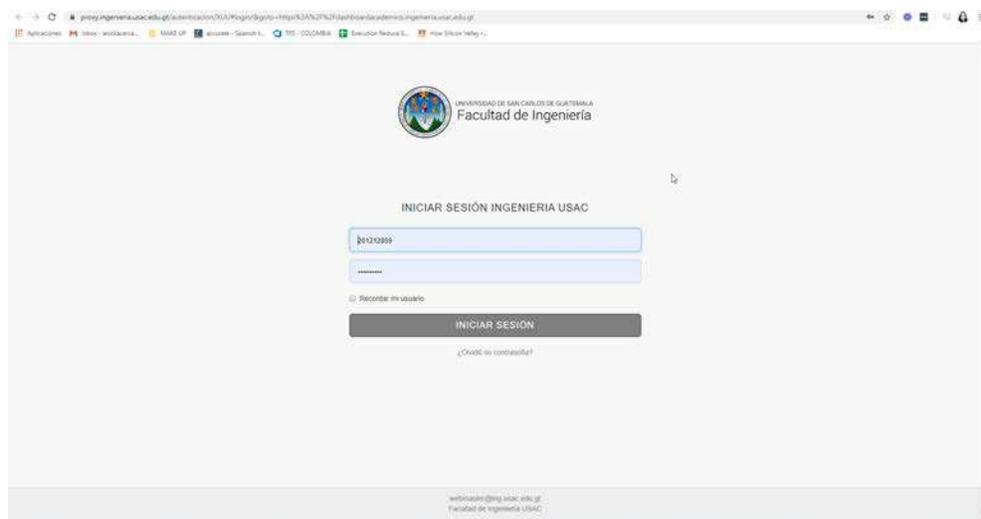
www.portal.ingenieria.usac.edu.gt/.

Consulta: octubre de 2019.

2.4.1.1.2. Paso 2

Iniciar sesión en Ingeniería USAC, colocando las credenciales de estudiante.

Figura 7. Inicio de sesión Facultad de Ingeniería

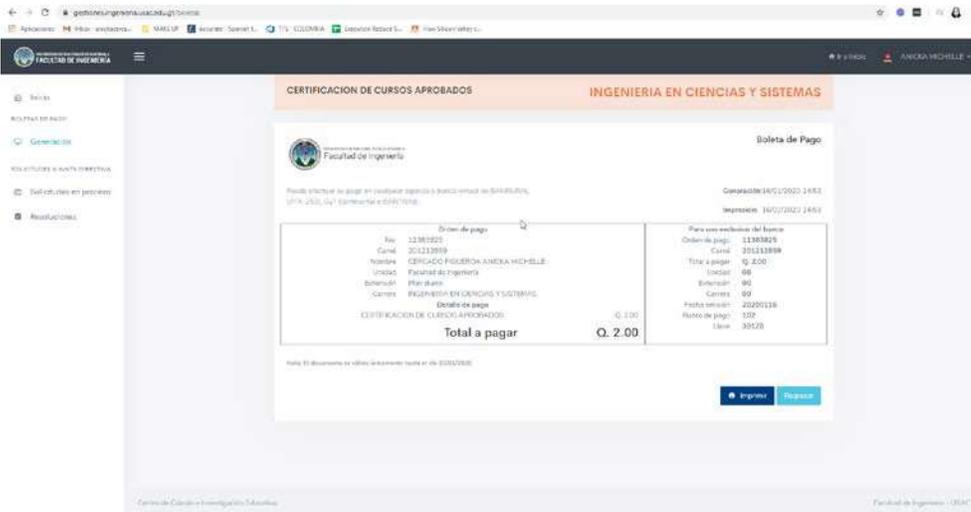


Fuente: Ingeniería USAC. *Inicio de sesión Facultad de Ingeniería*. www.proxy.ingenieria.usac.edu.gt/autenticacion/XUI/#login/&goto=https%3A%2F%2Fdashboardacademico.ingenieria.usac.edu.gt.
Consulta: octubre de 2019.

2.4.1.1.3. Paso 3

Ir al menú y seleccionar boletas de pago seguidamente identificar la opción para certificación de cursos aprobados y generar boleta de pago.

Figura 8. **Boleta de pago**



Fuente: Ingeniería USAC. Boleta de pago. www.gestiones.ingenieria.usac.edu.gt/boletas.
Consulta: octubre de 2019.

2.4.1.1.4. Paso 4

Realizar el pago de la boleta ya sea presencial o virtual, después dirigirse a la Facultad de Ingeniería USAC y obtener la certificación física.

Figura 9. Certificado de cursos aprobados

CERTIFICACION No. **358819**
SERIE "G-1"
HOJA No.1 de 2

FACULTAD DE INGENIERIA
EL INFRASCrito SECRETARIO DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.

CERTIFICA: QUE TUVO A LA VISTA EL EXPEDIENTE DEL ESTUDIANTE UNIVERSITARIO, **CERCADO FIGUEROA, ANICKA MICHELLE** CARNET: 201212859 EN EL CUAL CONVA SUO CURSO Y APROBO LAS SIGUIENTES ASIGNATURAS DE LA CARRERA DE:

INGENIERIA EN CIENCIAS Y SISTEMAS

CODIGO CURSO	NOMBRE DEL CURSO	CALIF.	FECHA DE APROBACION
0005 3	TECNICAS DE ESTUDIO E INVESTIGACION	078	05-2012
0006 2	IDIOMA TECNICO 1	065	05-2012
0017 4	AREA SOCIAL HUMANISTICA 1	077	05-2012
0039 1	DEPORTES 1	091	05-2012
0069 3	AREA TECNICA COMPLEMENTARIA 1	084	05-2012
0101 7	AREA MATEMATICA BASICA 1	078	05-2012
0019 4	AREA SOCIAL HUMANISTICA 2	081	06-2012
0003 1	ORIENTACION Y LIDERAZGO	067	11-2012
0008 2	IDIOMA TECNICO 2	061	11-2012
0103 7	AREA MATEMATICA BASICA 2	071	11-2012
0348 3	DINAMICA GENERAL 1	066	11-2012
0009 2	IDIOMA TECNICO 3	062	05-2013
0107 10	AREA MATEMATICA INTERMEDIA 1	068	05-2013
0147 5	FISICA BASICA	064	05-2013
0795 2	LOGICA DE SISTEMAS	076	05-2013
0960 5	MATEMATICA PARA COMPUTACION 1	095	05-2013
2005 0	PRACTICAS INICIALES	085	05-2013
0010 2	LOGICA	086	11-2013
0048 1	CONGRESOS ESTUDIANTILES 1	APR	11-2013
0112 5	AREA MATEMATICA INTERMEDIA 2	069	11-2013
0732 0	ESTADISTICA 1	067	11-2013
0770 4	INTROD A LA PROGRAMACION Y COMP 1	070	11-2013
0114 5	AREA MATEMATICA INTERMEDIA 3	072	01-2014
0150 6	FISICA 1	065	01-2014
0116 5	MATEMATICA APLICADA 3	079	05-2014
0118 6	MATEMATICA APLICADA 1	083	05-2014
0736 4	ANALISIS PROBABILISTICO	063	05-2014
0771 5	INTROD A LA PROGRAMACION Y COMP 2	072	05-2014
0796 3	LENGUAJES FORMALES Y DE PROGRAMACION	078	05-2014
0962 5	MATEMATICA PARA COMPUTACION 2	083	05-2014
0152 6	FISICA 2	063	07-2014
0014 4	ECONOMIA	066	08-2014
0601 5	INVESTIGACION DE OPERACIONES I	073	11-2014
0650 3	CONTABILIDAD 1	073	11-2014
0777 4	ORG DE LENGUAJES Y COMPILADORES 1	079	11-2014
0964 3	ORGANIZACION COMPUTACIONAL	069	11-2014
0772 5	ESTRUCTURAS DE DATOS	065	01-2015
0603 5	INVESTIGACION DE OPERACIONES II	066	05-2015

SEGUN: C.E.U. ACTA 80 2006, PUNTO SEGUNDO, DE FECHA 8 DE FEBRERO DEL 2006, Y JUNTA DIRECTIVA FAC. DE ING. ACTA 10-088, PUNTO UNICO, DE FECHA 14 DE JUNIO DE 2006 Y ACTA 30-088, PUNTO 1.2, DE FECHA 20 DE AGOSTO DE 2006.

CREDITOS APROBADOS **162** DE UN TOTAL DE: **250**

LA CALIFICACION ES DE CERO (0) A SESENTA (60) PUNTOS; REPROBADO, DE SESENTA Y UNO (61) A CIN (100) PUNTOS; APROBADO, ESTA CERTIFICACION NO CONVALIDA NINGUNA DE LAS MATERIAS QUE EL ALUMNO HUBIERE CURSADO Y APROBADO SIN AJUSTARSE A LOS REQUISITOS ESTABLECIDOS EN EL PLAN DE ESTUDIOS DE ESTA FACULTAD.

Y A SOLICITUD DEL INTERESADO SE EXTENDE LA PRESENTE CERTIFICACION, EN LA CIUDAD DE GUATEMALA, AL DIA

QUINTACUATRO DE DOS MIL DIECISEIS HONORARIOS Q. 1.00 POR HOJA

SECRETARIO ACADÉMICO

SECRETARIO DE INGENIERIA

Fuente: Faculta de Ingeniería, certificado de cursos aprobados.

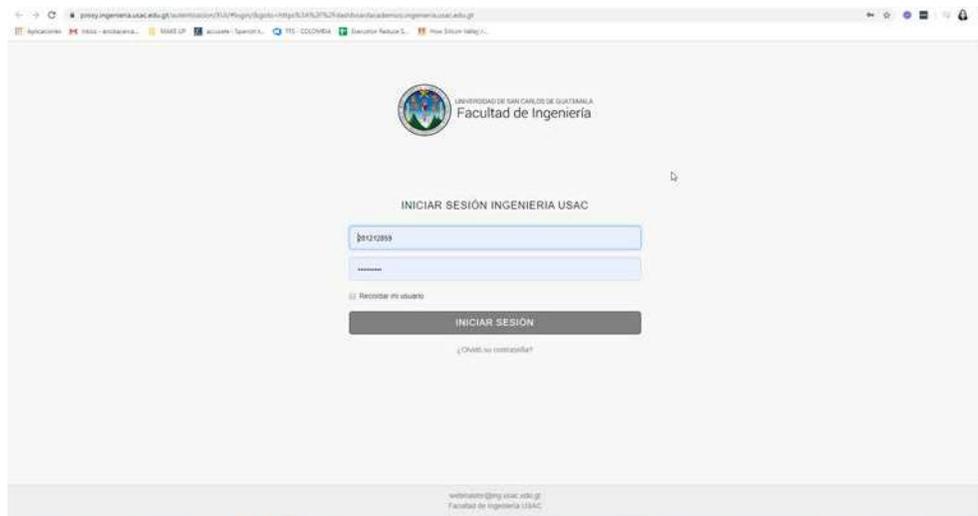
2.4.1.2. Proceso para visualizar el listado de cursos aprobados

Actualmente en el portal ingeniería USAC para visualizar el listado de cursos aprobados es necesario seguir una serie de pasos los cuales se detallan a continuación:

2.4.1.2.1. Paso 1

Ingresar a la página de inicio del portal ingeniería USAC <https://portal.ingenieria.usac.edu.gt/>, seguidamente buscar la opción de iniciar sesión.

Figura 10. Inicio de sesión Facultad de Ingeniería



Fuente: Ingeniería USAC. *Inicio de sesión Facultad de Ingeniería.*

www.proxy.ingenieria.usac.edu.gt/autenticacion/

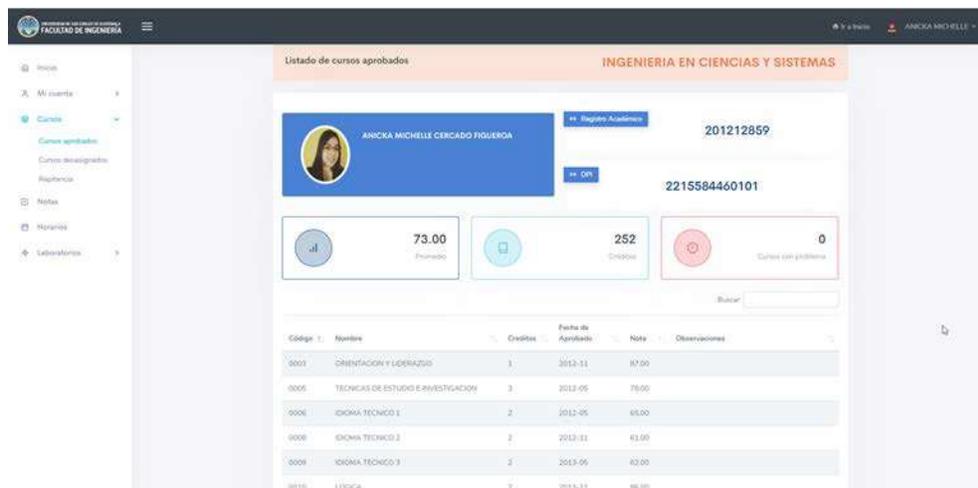
<https://portal.ingenieria.usac.edu.gt/XUI/#login/&goto=https%3A%2F%2Fdashboardacademico.ingenieria.usac.edu.gt>

Consulta: octubre de 2019.

2.4.1.2.2. Paso 2

Seleccionar cursos, luego cursos aprobados y visualizar nombre completo, registro académico, DPI, promedio, créditos, cursos con problema y el listado de cursos aprobados.

Figura 11. Listado de cursos aprobados



Código	Nombre	Créditos	Fecha de Aprobación	Nota	Observaciones
0003	ORIENTACION Y LIDERAZGO	1	2013-11	87.00	
0005	TECNICAS DE ESTUDIO E INVESTIGACION	3	2013-05	78.00	
0006	IDIOMA TECNICO 1	2	2013-05	80.00	
0008	IDIOMA TECNICO 2	2	2013-11	81.00	
0009	IDIOMA TECNICO 3	2	2013-05	82.00	
0010	LÓGICA	2	2013-11	86.00	

Fuente: Ingeniería USAC. *Listado de cursos.*

aprobados.www.infoestudiantes.ingenieria.usac.edu.gt/cursosAprobados.

Consulta: octubre de 2019.

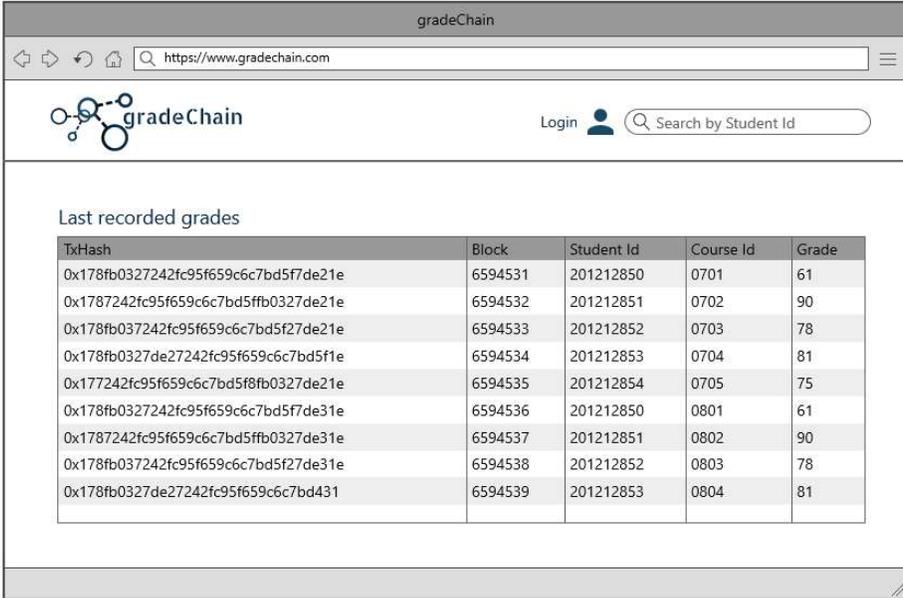
2.4.2. GradeChain

Herramienta implementada como solución para fortalecer la validez de los certificados de cursos aprobados por los estudiantes.

2.4.2.1. Página de inicio

En la pantalla de inicio de GradeChain, se encuentra una lista con la data de las últimas notas registradas, cada fila de la lista redirige hacia la transacción, bloque y registro del estudiante por medio del carnet. Esta página no necesita autenticación.

Figura 12. Prototipo login



The screenshot shows a web browser window with the URL <https://www.gradechain.com>. The page features the GradeChain logo, a 'Login' button with a user icon, and a search bar labeled 'Search by Student Id'. Below the navigation bar, there is a section titled 'Last recorded grades' containing a table with the following data:

TxHash	Block	Student Id	Course Id	Grade
0x178fb0327242fc95f659c6c7bd5f7de21e	6594531	201212850	0701	61
0x1787242fc95f659c6c7bd5ffb0327de21e	6594532	201212851	0702	90
0x178fb037242fc95f659c6c7bd5f27de21e	6594533	201212852	0703	78
0x178fb0327de27242fc95f659c6c7bd5f1e	6594534	201212853	0704	81
0x177242fc95f659c6c7bd5f8fb0327de21e	6594535	201212854	0705	75
0x178fb0327242fc95f659c6c7bd5f7de31e	6594536	201212850	0801	61
0x1787242fc95f659c6c7bd5ffb0327de31e	6594537	201212851	0802	90
0x178fb037242fc95f659c6c7bd5f27de31e	6594538	201212852	0803	78
0x178fb0327de27242fc95f659c6c7bd431	6594539	201212853	0804	81

Fuente: elaboración propia, empleando Balsamiq 2018.

2.4.2.2. Certificación de cursos aprobados

La certificación de cursos aprobados de GradeChain es pública y se accede ingresando en la parte superior derecha el número de carnet, en esta se muestran los nombres, apellidos, carrera y fecha de nacimiento; seguido del listado de cursos aprobados con la siguiente información: transacción, bloque, fecha, código de curso, id del curso, y la nota. Actualmente en el portal ingeniería USAC

únicamente se genera boleta de pago y posterior a esto se obtiene la certificación presencialmente.

Figura 13. **Prototipo cursos aprobados**

The screenshot shows a web browser window with the URL <https://www.gradechain.com/student>. The page header includes the GradeChain logo and a login field with the value '201212859'. The main content area displays a student profile with a placeholder for a photo and the following fields:

- First Name: Anicka Michelle
- Last Name: Cercado Figueroa
- Career: Ingeniería en ciencias y sistemas
- Birthdate: 30/04/1993

Below the profile information is a table with the following data:

TxHash	Block	Date	Course Code	Course Id	Grade
0x178fb0327242fc95f659c6c7bd5f7de	55945	10/07/2017	0281	SISTEMAS OPERATIVOS 1	70
0x1787242fc95f659c6c7bd5ffb0327de	54945	15/07/2017	0775	SISTEMAS DE BASES DE DATOS 2	71
0x178fb037242fc95f659c6c7bd5f27de	53945	13/07/2017	0780	SOFTWARE AVANZADO	72
0x178fb0327de27242fc95f659c6c7bd5	52945	11/07/2017	0972	INTELIGENCIA ARTIFICIAL 1	65
0x177242fc95f659c6c7bd5f8fb0327de	51945	12/07/2017	0285	SISTEMAS OPERATIVOS 2	85

Fuente: elaboración propia, empleando Balsamiq 2018.

2.4.2.3. **Certificación de cierre de pensum**

La certificación de cierre de pensum de GradeChain es pública y se obtiene como un documento PDF, que contiene la siguiente información: carnet, nombres, apellidos, carrera, fecha de nacimiento, listado de cursos aprobados, el último elemento de la lista contiene el cierre de pensum como curso.

Figura 14. Prototipo de certificación de cierre de pensum



Fuente: elaboración propia.

2.4.3. Métricas de comparación

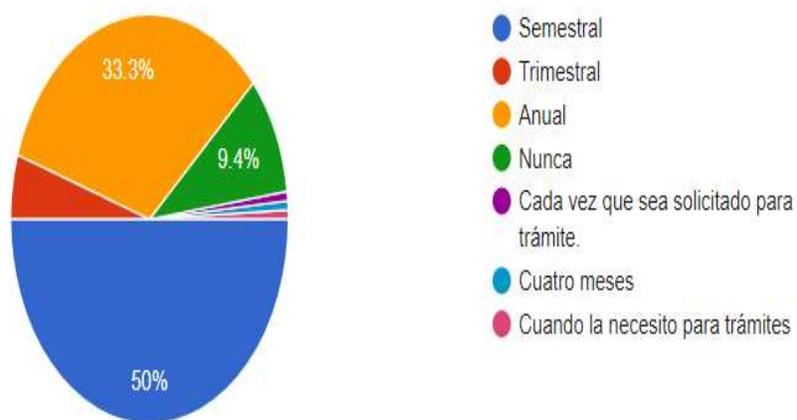
Se realizó una encuesta con dos preguntas a los estudiantes de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala por medio de la herramienta GoogleForms (<https://forms.gle/4PrdP61dtQss6mSg8>), que fue atendida por 138 estudiantes.

El objetivo de la entrevista es medir la cantidad recursos ahorrados por estudiante utilizando GradeChain contra el sistema actual, a continuación, se muestran las preguntas con sus respectivos resultados y análisis.

Figura 15. **Pregunta 1**

¿Con qué frecuencia solicitas certificaciones de cursos aprobados?

138 respuestas



Fuente: elaboración propia.

Tabla I. **Frecuencia de certificaciones solicitadas**

Respuesta	Cantidad anual	Cantidad personas	Cantidad de certificaciones solicitadas en un año
Semestral	2	69	138
Anual	1	46	46
Nunca	0	13	0
Trimestral	4	7	28
Trámites	1	2	2
Cuatro meses	3	1	3
Total		138	217

Fuente: elaboración propia.

Tabla II. **Promedio de certificaciones solicitadas**

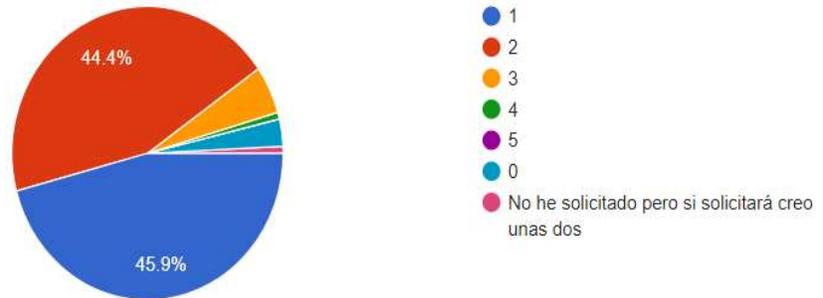
	Fórmula (Cantidad Personas/Número de Copias)	Resultado
Promedio	217/138	1 572463768

Fuente: elaboración propia.

Figura 16. **Pregunta 2**

¿Cuántas copias solicitas?

133 respuestas



Fuente: elaboración propia.

Tabla III. **Copias solicitadas**

Respuesta	Cantidad Personas	Número de Copias
1	61	61
2	60	120
3	7	21
0	9	0
4	1	4
Total	138	206

Fuente: elaboración propia.

Tabla IV. **Promedio copias por persona**

	Formula (Respuesta/Número de Copias)	Resultado
Promedio	226/133	1 492753623

Fuente: elaboración propia.

2.4.3.1. Tiempo para gestionar una certificación de cursos aprobados

El tiempo estimado en el que un estudiante obtiene N certificaciones de cursos aprobados es de 10 minutos, el resultado es basado en la experiencia a través de los años en la Facultad de Ingeniería y no estima tiempo de transporte entre otros factores, se debe resaltar que en un solo trámite se pueden obtener N cantidad de certificaciones de cursos aprobado.

Tabla V. **Actividades por minuto**

Descripción actividad	Tiempo actividad minutos
Ingresar al portal Ingeniería USAC, hacer login y generar boleta (s) de pago.	3
Imprimir boleta (s).	2
Realizar pago (línea o presencial).	3
Entrega de certificación (es) .	2

Fuente: elaboración propia.

2.4.3.2. Estudiantes inscritos

A continuación se detalla la cantidad de estudiantes inscritos en la Facultad Ingeniería USAC durante los últimos 5 años, los datos se obtuvieron de la página <https://datosabiertos.ingenieria.usac.edu.gt/dataset/estudiantes-inscritos>.

Tabla VI. **Estudiantes inscritos por año**

Año	Estudiantes inscritos
2015	14 006
2016	13 674
2017	13 780
2018	13 695
2019	13 807
Total	68 962

Fuente: elaboración propia.

Tabla VII. **Promedio de estudiantes inscritos por año**

	Formula (Estudiantes inscritos/Cantidad de años)	Resultado
Promedio	68962/5	13,792,4

Fuente: elaboración propia.

2.4.4. Análisis de resultados

Según los resultados anteriores un estudiante a lo largo de un año realiza el trámite para tener el certificado de cursos aprobados 2 veces; el tiempo promedio de una carrera en la Facultad Ingeniería USAC es de 5 años, lo que significa que un estudiante realiza 10 veces la gestión obteniendo una única copia del certificado, a continuación, se lista el ahorro de recursos cuando se utiliza la plataforma GradeChain.

- En promedio un estudiante ahorra 20 minutos anuales.
- En promedio un estudiante ahorra 100 minutos a lo largo de su carrera.
- Por cada estudiante se ahorra 4 hojas de papel anuales, asumiendo que el certificado es de dos páginas.
- Por cada estudiante se ahorra 20 hojas de papel a lo largo de su carrera, asumiendo que el certificado es de dos páginas.

En promedio en la Facultad de Ingeniería hay 13,792 estudiantes, por lo tanto:

- El ahorro total en tiempo es de 2, 068,800 minutos.
- El ahorro total en hojas es de 275,840 unidades.

Tabla VIII. **Tabla comparativa**

Aplicativo	Ventajas	Desventajas
Portal de Ingeniería	<ul style="list-style-type: none"> • Intuitivo • Aceptación cultural del aplicativo por parte de los estudiantes y personal de la Facultad Ingeniería. 	<ul style="list-style-type: none"> • El proceso para obtener una certificación de cursos aprobados, requiere de distintos procesos en el portal de ingeniería, banco (en línea o presencial) y recepción del certificado. • El proceso para obtener una certificación de cierre de pensum requiere una serie de pasos en línea y presenciales. • Cada certificado requiere de recursos como papel, sellos y firmas.

Continuación de la tabla VIII.

<p>GradeChain</p>	<ul style="list-style-type: none"> • Intuitivo. • Generación inmediata de certificación de cursos aprobados. • Generación inmediata de certificación de cierre de pensum. • El ahorro promedio en tiempo, para los estudiantes de la Facultad de Ingeniería en un periodo de 5 años, es de 2 068,800 minutos. • El ahorro promedio en hojas, para los estudiantes de la Facultad de Ingeniería en un periodo de 5 años, es de 275,840 unidades. 	<ul style="list-style-type: none"> • No hay evidencias de la aceptación cultural del aplicativo por parte de los interesados. • No hay conocimiento por parte de la mayoría interesados del potencial de la tecnología, hace falta capacitación.
--------------------------	--	--

Fuente: elaboración propia.

3. DISEÑO DE LA APLICACIÓN GRADECHAIN BAJO LA NECESIDAD IDENTIFICADA

En este capítulo se muestran los prototipos diseñados de GradeChain señalando y nombrando uno a uno sus objetos, adicional a esto se describe la funcionalidad de cada objeto y página, también la arquitectura propuesta; así mismo los atributos de calidad que lo hacen intuitivo y fácil de utilizar para los usuarios finales.

3.1. Prototipos

Los prototipos son plantillas utilizadas para generar la idea inicial de cómo se visualizará y estructurará un proyecto.

3.1.1. Diagrama general de la solución

La solución contiene páginas públicas y privadas, cualquier usuario en la red con la URL de GradeChain pueden ingresar a las páginas de acceso público, por el contrario, únicamente personal administrativo puede ingresar al mantenimiento de estudiantes, cursos y notas; las cuales son páginas de acceso privado.

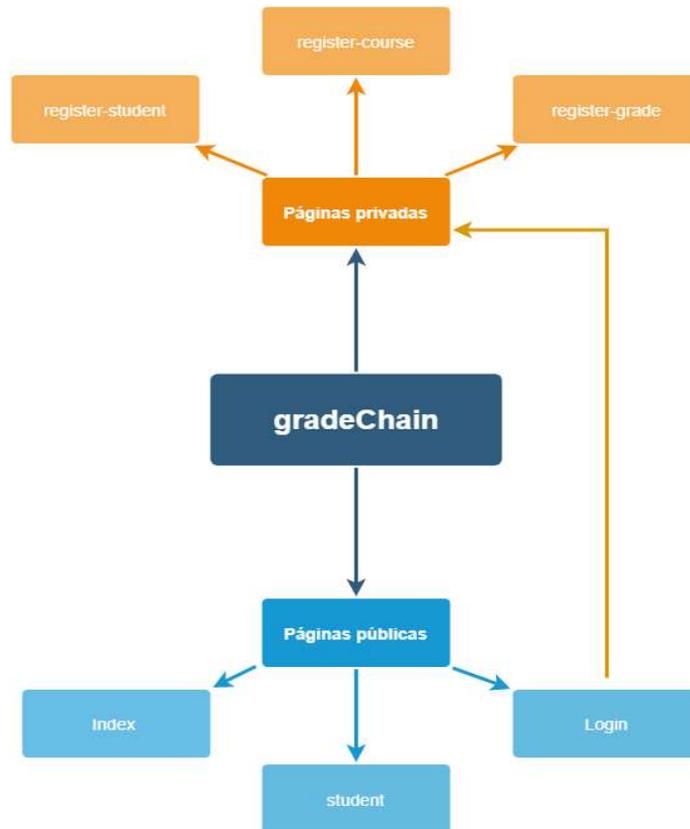
Las páginas públicas se dividen en:

- Página principal: página de inicio donde se listan los últimos registros ingresados.
- Estudiante: contiene información pública de los estudiantes.
- Login: inicio de sesión para personal administrativo.

Las páginas privadas se dividen en:

- Registrar estudiante: página de adición de estudiantes.
- Registrar curso: página de adición de curso.
- Registrar nota: página de adición de notas.

Figura 17. Diagrama de la solución

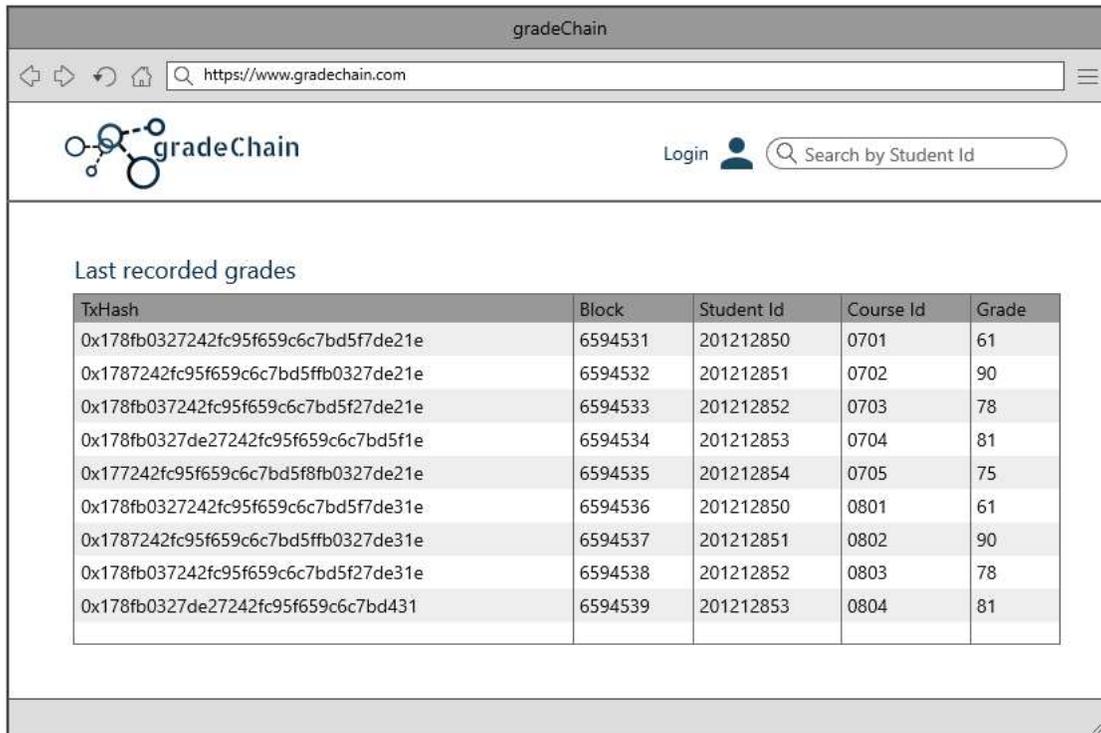


Fuente: elaboración propia, empleando draw.io 2020.

3.1.2. Página principal

Esta vista contiene los objetos: Esta vista contiene logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados), por carnet y listado de los últimos registros de notas.

Figura 18. Prototipo página principal



Fuente: elaboración propia, empleando Balsamiq 2018.

3.1.3. Estudiante

Esta vista contiene los objetos: logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados), por carnet, nombre, apellido, carrera, fecha de nacimiento y el listado de los cursos aprobados.

Figura 19. Prototipo página de estudiante

The screenshot shows a web browser window with the URL <https://www.gradechain.com/student>. The page header includes the gradeChain logo and a login field with the value 201212859. The main content area displays a student profile with a placeholder for a profile picture and the following information:

- First Name: Anicka Michelle
- Last Name: Cercado Figueroa
- Career: Ingeniería en ciencias y sistemas
- Birthdate: 30/04/1993

Below the profile information is a table listing completed courses:

TxHash	Block	Date	Course Code	Course Id	Grade
0x178fb0327242fc95f659c6c7bd5f7de	55945	10/07/2017	0281	SISTEMAS OPERATIVOS 1	70
0x1787242fc95f659c6c7bd5ffb0327de	54945	15/07/2017	0775	SISTEMAS DE BASES DE DATOS 2	71
0x178fb037242fc95f659c6c7bd5f27de	53945	13/07/2017	0780	SOFTWARE AVANZADO	72
0x178fb0327de27242fc95f659c6c7bd5	52945	11/07/2017	0972	INTELIGENCIA ARTIFICIAL 1	65
0x177242fc95f659c6c7bd5f8fb0327de	51945	12/07/2017	0285	SISTEMAS OPERATIVOS 2	85

Fuente: elaboración propia, empleando Balsamiq 2018.

3.1.4. Certificación de cierre de pensum

Se visualiza como documento PDF, y contiene la siguiente información: carnet, nombres, apellidos, carrera, fecha de nacimiento, listado de cursos aprobados, el último elemento de la lista contiene el cierre de pensum como curso.

Figura 20. Prototipo de certificación cierre de pensum en PDF

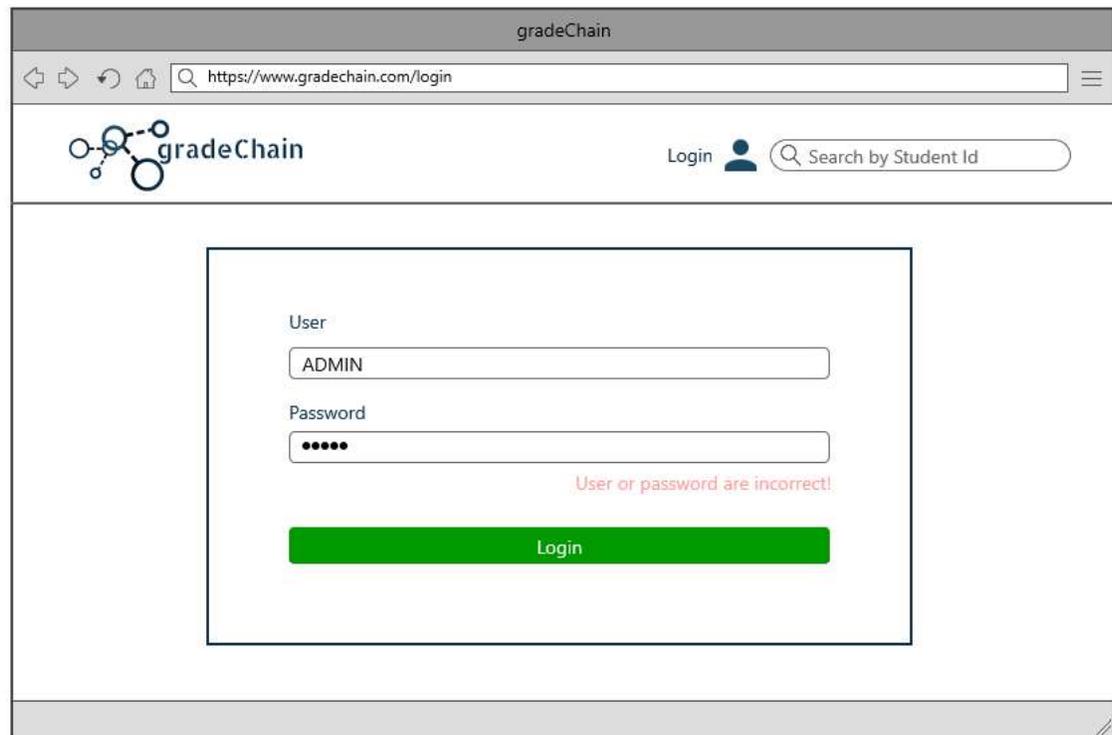


Fuente: elaboración propia.

3.1.5. Login

Esta vista contiene los objetos: logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados) por carnet, usuario, contraseña y botón de login.

Figura 21. Prototipo página login



Fuente: elaboración propia, empleando Balsamiq 2018.

3.1.6. Registrar estudiante

Esta vista contiene los objetos: logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, carnet, nombre, apellido, carrera, fecha de nacimiento, imagen y botón submit.

Figura 22. **Prototipo página registrar estudiante**

gradeChain

Logout Search by Student Id

[Register Student](#) [Register Course](#) [Register Grade](#)

Student Id	<input type="text" value="201212859"/>
First Name	<input type="text" value="Anicka Michelle"/>
Last Name	<input type="text" value="Cercado Figueroa"/>
Career	<input type="text" value="Ingeniería en ciencias y sistemas"/>
Birthdate	<input type="text" value="30 / 04 / 2013"/>
Image	<input type="text" value="img178953.jpg"/> <input type="button" value="OK"/>

Fuente: elaboración propia, empleando Balsamiq 2018.

3.1.7. Registrar curso

Esta vista contiene los objetos: logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, id del curso, nombre del curso y botón submit.

Figura 23. **Prototipo página registrar curso**

The image shows a web browser window with the following elements:

- Browser title: gradeChain
- Address bar: https://www.gradechain.com/register-course
- Header: gradeChain logo, Logout button, Search by Student Id input field, and links for Register Student, Register Course, and Register Grade.
- Main content area: Course Id input field (0281), Course Name input field (SISTEMAS OPERATIVOS 1), and a green Submit button.

Fuente: elaboración propia, empleando Balsamiq 2018.

3.1.8. Registrar nota

Esta vista contiene los objetos: logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, objetos para hacer la búsqueda de estudiante, objetos para hacer la búsqueda de curso, nota y submit, esta vista solo puede ser accedida por el administrador.

Figura 24. Prototipo página registrar nota

The screenshot shows a web browser window with the URL <https://www.gradechain.com/register-grade>. The page header includes the gradeChain logo, a Logout button, and a search bar labeled "Search by Student Id". Below the header, there are three navigation links: "Register Student", "Register Course", and "Register Grade". The main form contains the following fields and buttons:

- Student Id:** Input field with value "201212859" and a red "Search" button.
- Student Information:** A table with four rows:

First Name	Anicka Michelle
Last Name	Cercado Figueroa
Career	Ingeniería en ciencias y sistemas
Birthdate	30/04/1993
- Course Id:** Input field with value "201212859" and a red "Search" button.
- Course Name:** Input field with value "SISTEMAS OPERATIVOS 1".
- Grade:** Input field with value "70" and a green "Submit" button.

Fuente: elaboración propia, empleando Balsamiq 2018.

3.2. Diseño intuitivo y usabilidad

El diseño que se utilizó busca que se fácil de usar para un usuario final y que su estructura defina el comportamiento o acciones que la pagina tendrá.

3.2.1. Redireccionamiento

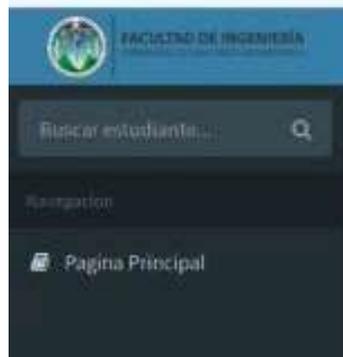
El diseño de la aplicación cuenta con distintos objetos en el menú de página, estos objetos son fácil de ubicar por sus etiquetas intuitivas que hacen

redireccionamiento de forma simple a las distintas funcionalidades de GradeChain.

3.2.2. Menú de página sin autenticación

El menú de página contiene el logo de la aplicación GradeChain, este redirecciona a la página de inicio (index), esta cabecera se encuentra presente en todas las páginas, también contiene link con imagen para ir al login, y un buscador de estudiante, por número de carnet, el cual redirige a la página estudiante que contiene la certificación de cursos aprobados.

Figura 25. Menú de página sin autenticación



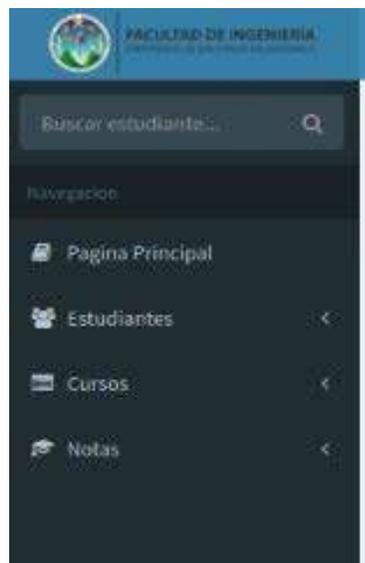
Fuente: elaboración propia, empleando Adonis JS 4.1.

3.2.3. Menú de página con autenticación

El menú de página contiene el logo de la aplicación GradeChain, este redirecciona a la página de inicio (index), esta cabecera se encuentra presente

en todas las páginas, también contiene link con imagen para cerrar sesión, y un buscador de estudiante, por número de carnet, que redirige a la página estudiante que contiene la certificación de cursos aprobados, en la parte inferior de este se encuentran links hacia el registro de estudiante, curso y nota.

Figura 26. **Cabecera de página con autenticación**



Fuente: elaboración propia, empleando Adonis JS 4.1.

3.2.4. **Diseño responsivo**

Las páginas de la aplicación GradeChain contienen un diseño responsivo, es decir, el contenido se ajusta a distintas dimensiones de pantalla, esto permite hacer uso de la aplicación en cualquier dispositivo; por ejemplo, smartphones, tablets, computadoras.

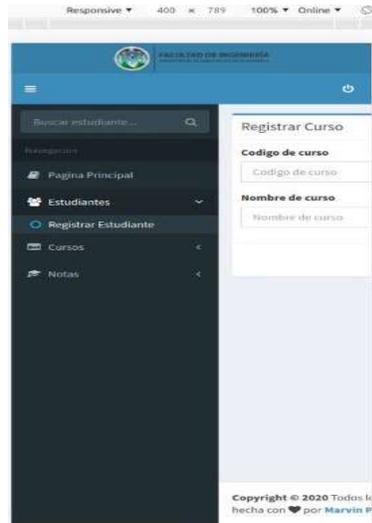
Figura 27. **Diseño responsivo 1**



The image shows a browser window displaying a responsive web form. The browser's address bar indicates 'Responsive' mode with a width of 400 x 789 pixels and a zoom level of 100%. The page title is 'Online'. The form is titled 'Registrar Curso' and is part of a website for 'FACULTAD DE INGENIERIA'. It features two input fields: 'Codigo de curso' and 'Nombre de curso'. A blue 'Registrar' button is positioned at the bottom right of the form. The footer contains the text: 'Copyright © 2020 Todos los derechos reservados. | Pagina hecha con ❤️ por Marvin Pivaral & Anicka Cercado.'

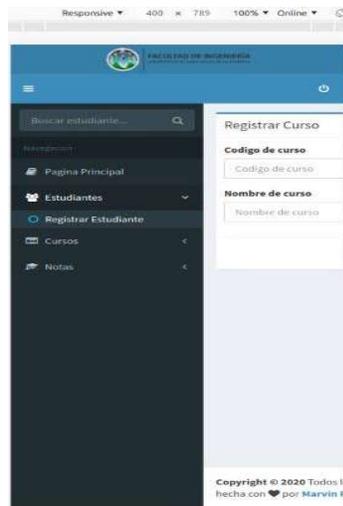
Fuente: elaboración propia, empleando Adonis JS 4.1.

Figura 28. **Diseño responsivo 2**



Fuente: elaboración propia, empleando Adonis JS 4.1.

Figura 29. **Diseño responsivo 3**



Fuente: elaboración propia, empleando Adonis JS 4.1.

Figura 30. **Diseño responsivo 4**

	Numero de Bloque	Carnet Estudiante
a3a24534f9b8d9e93849aa800d9e9	7825839	201213
9882d49aaad8f8b1bf6c2c8b0347	7825819	201213
e9aa75ff0d6cdadff868d23f57fd	7825799	201213
639eaada8dc52eca0b1c7fce3d0	6093999	201213

registros

Copyright © 2020 Todos los derechos reservados. |
Página hecha con por [Marvin Pivaral & Anicka Cercado](#).

Fuente: elaboración propia, empleando Adonis JS 4.1.

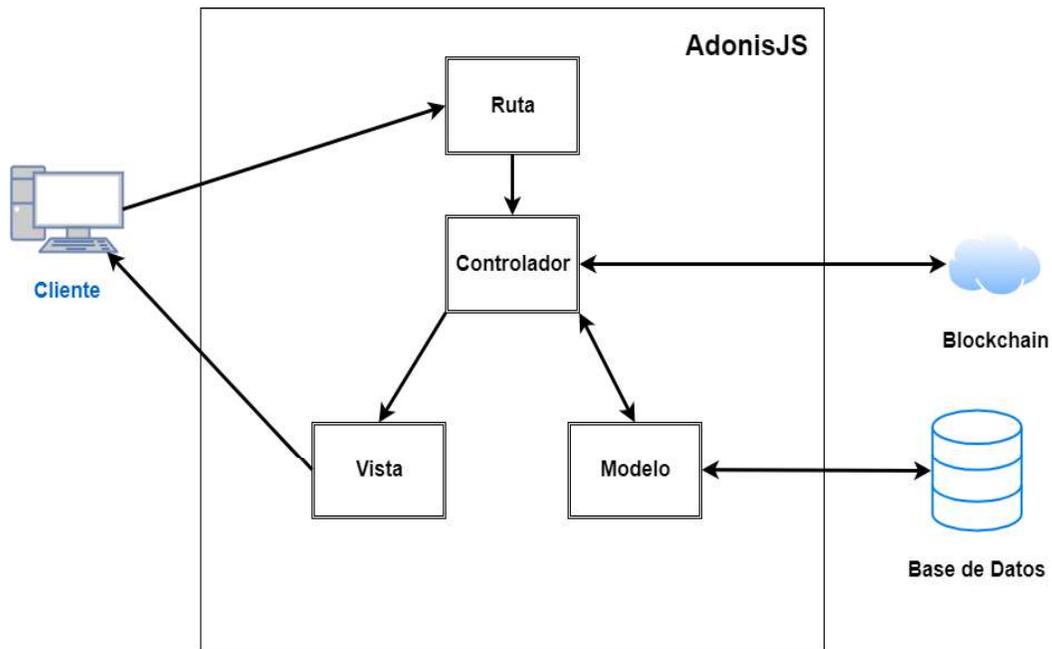
3.3. **Arquitectura del sistema**

Para construir GradeChain se utilizó una arquitectura modelo - vista - controlador, debido a que GradeChain es una aplicación web, es mejor tener separadas la parte visual y la parte lógica de la misma. El cliente accede por medio de una ruta web, el servidor se encarga de redirigir esta al controlador correcto para realizar las instrucciones que sean necesarias, el controlador puede utilizar o no el modelo donde se mapea la base de datos y se accede, modifica, crea o elimina información, una vez el controlador termine de realizar el proceso lógico genera una vista que es devuelta al cliente quien podrá visualizar el resultado.

El controlador también se encarga de la comunicación con la red de Ethereum, es encargado de consultar, y ejecutar los métodos definidos dentro del *Smart Contract* creado para GradeChain.

Esta arquitectura facilita el mantenimiento de la aplicación, ya que la parte lógica y gráfica se encuentran completamente divididas.

Figura 31. **Arquitectura del sistema: modelo, vista, controlador**



Fuente: elaboración propia, empleando draw.io 2020.

4. DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN

Este capítulo tiene como objetivo describir las herramientas utilizadas, software y hardware para desarrollar GradeChain; así mismo la implementación de la metodología SCRUM en el proyecto, también se realiza un tutorial de usuario y desarrollo con sus consideraciones.

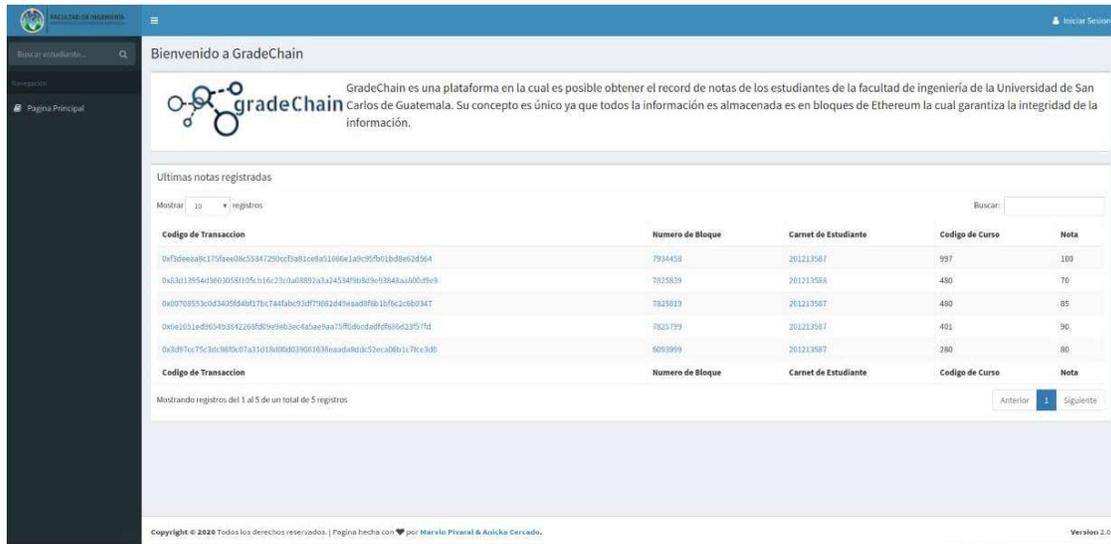
4.1. Frontend

El Frontend es la parte de la aplicación con la cual el usuario final tendrá interacción y esta le mostrará información.

4.1.1. Página principal

Esta página mostrar información referente con las ultimas actualizaciones sobre las notas registradas en el sistema.

Figura 32. **Página principal de la aplicación**



Fuente: elaboración propia, empleando Adonis JS 4.1.

La página principal cuenta con los siguientes campos:

- Botón de iniciar sesión: redirige hacia la página de inicio de sesión.
- Cuadro de búsqueda de estudiante por carnet: se debe ingresar el carnet del estudiante para realizar la búsqueda de los cursos aprobados, si éste existe.
- Tabla con la lista de últimos registros de notas: muestra los últimos registros de notas hechos dentro de la plataforma en la red de Ethereum, la tabla cuenta con los siguientes campos:
 - Código de transacción: código hash de la transacción donde se creó el registro, al hacer clic sobre ella se redirige a la página de Ethereum con la información de la transacción.

- Número de bloque: número de bloque donde se registró la información de la nota del estudiante, al hacer clic sobre ella redirige a la página de Ethereum con la información del bloque.
 - Carnet del estudiante: carnet del estudiante a quien se le registró la nota, al hacer clic, redirige a la página de consulta de estudiante.
 - Código de curso: número del código del curso con que se registró la nota.
 - Nota: nota que se registró al estudiante y curso asociado.
- Logotipo: logotipo de la aplicación, al hacer clic sobre ella redirige hacia la página principal.

4.1.2. Página de inicio de sesión

Esta página solicitará al usuario sus credenciales para iniciar sesión.

Figura 33. **Página inicio de sesión de la aplicación**



GradeChain

Ingresá tus datos para Iniciar Sesión

Usuario

Contraseña

Iniciar Sesión

Fuente: elaboración propia, empleando Adonis JS 4.1.

La página de inicios de sesión cuenta con los siguientes campos:

- Cuadro de nombre de usuario: cuadro de texto donde se ingresa el nombre de usuario con que se quiere iniciar sesión.
- Cuadro de contraseña: cuadro de texto donde se ingresa la clave del usuario con que se quiere iniciar sesión.
- Botón de Iniciar sesión: al hacer clic, si los datos ingresados en los campos anteriores son correctos, se iniciará la sesión.

4.1.3. Página de consulta de estudiante

Esta página por medio del carnet del estudiante mostrara los datos del mismo junto con el listado de calificaciones obtenidas a la fecha.

Figura 34. Página de consulta de estudiante de la aplicación

The screenshot displays the 'Cursos Aprobados' page for a student. The page includes a search bar, a navigation menu, and a student profile section with the following details:

- Carnet del Estudiante:** 201213587
- Nombres del estudiante:** Marvin Emmanuel
- Apellidos del estudiante:** Pizaral Orellana
- Carrera:** Ingeniería en Ciencias y Sistemas
- Nacimiento del estudiante:** 06/31/1993

Below the profile is a table of approved courses with the following data:

Codigo de Transaccion	Codigo de Curso	Nombre de Curso	Nota
6093999	280	Sistemas Operativos	80
7825799	401	Curso Prueba	90
7825819	480	Sistemas Operativos	85

The page also features a 'Fecha: 19/05/2020' indicator and an 'Imprimir' button.

Fuente: elaboración propia, empleando Adonis JS 4.1.

La página de consulta de estudiantes cuenta con los siguientes campos:

- Imagen de identificación: fotografía del estudiante que se tiene registrada en el sistema.
- Nombres: campo que muestra los nombres del estudiante.
- Apellidos: campo que muestra los apellidos del estudiante.
- Carrera: campo que muestra la carrera que el estudiante tiene asignado.
- Fecha de nacimiento: campo que muestra la fecha de nacimiento del estudiante.
- Tabla de cursos aprobados: tabla que muestra el listado de cursos aprobados del estudiante. La tabla cuenta con los siguientes campos:
 - Número de bloque: muestra el número del bloque donde fue registrada la nota del curso.
 - Código del curso: identificador del curso al que fue asignada la nota.
 - Nombre del curso: nombre del curso al que fue asignada la nota.
 - Nota: nota del curso.

4.1.4. Página de registro de estudiante

Esta página solicitará los datos necesarios para poder crear un estudiante en el sistema.

Figura 35. **Página de registro de estudiante de la aplicación**

Registrar Estudiante

Carnet del Estudiante

Carnet del Estudiante

Nombres del estudiante

Nombres del estudiante

Apellidos del estudiante

Apellidos del estudiante

Carrera

Carrera

Nacimiento del estudiante

Nacimiento del estudiante

Foto del estudiante

Seleccionar archivo No se eligió archivo

Registrar

Copyright © 2020 Todos los derechos reservados. | Página hecha con por Marvin Pizaral & Anicka Cercado.

Version 2.0

Fuente: elaboración propia, empleando Adonis JS 4.1.

Página utilizada para registrar la información de un nuevo estudiante, la página cuenta con los siguientes campos:

- Carnet del estudiante: cuadro de texto donde se ingresa el nuevo id del estudiante o carnet, este no debe de existir en el sistema.
- Nombres: nombres del estudiante.
- Apellidos: apellidos del estudiante.
- Carrera: carrera que será asignada al estudiante.
- Fecha de nacimiento: fecha de nacimiento del estudiante.
- Buscador de fotografía: botón para la búsqueda de la fotografía que se utilizara para identificar al estudiante.
- Botón Registrar: botón para guardar los datos en el sistema.

- Cuadro de fotografía: cuadro donde se visualizará la imagen seleccionada para el estudiante.

4.1.5. Certificado cierre de pensum

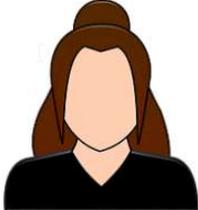
En esta página se observará un documento que contará con una firma digital que dará veracidad al listado de cursos aprobados del estudiante consultado.

Figura 36. Documento PDF certificado de cierre de pensum de la aplicación



FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Constancia de cierre de cursos



Carnet del Estudiante
201213587

Nombres del estudiante
Marvin Emmanuel

Apellidos del estudiante
Pivaral Orellana

Carrera
Ingeniería en Ciencias y Sistemas

Nacimiento del estudiante
08/31/1993

Bloque de Transaccion	Codigo de Curso	Nombre de Curso	Nota
6093999	280	Sistemas Operativos	80
7825799	401	Curso Prueba	90
7825819	480	Sistemas Operativos	85
7934458	997	CIERRE DE PENSUM	100

Esta es una constancia de cierre firmada con blockchain con el codigo de transaccion Firma Digital
(<https://ropsten.etherscan.io/tx/0xf3deaaa8c175faee08c55347290ccf3a81ce8a51066e1a9c95fb01bd8e62d564#eventlog>)

Fuente: elaboración propia, empleando Adonis JS 4.1.

Página utilizada para registrar la información de un nuevo estudiante, la página cuenta con los siguientes campos:

- Imagen de identificación: fotografía del estudiante que se tiene registrada en el sistema.
- Carnet del estudiante: carnet del estudiante.
- Nombres: nombres del estudiante.
- Apellidos: apellidos del estudiante.
- Carrera: carrera que será asignada al estudiante.
- Fecha de nacimiento: fecha de nacimiento del estudiante.
- Tabla de cursos aprobados: tabla que muestra el listado de cursos aprobados del estudiante. La tabla cuenta con los siguientes campos:
 - Número de bloque: muestra el número del bloque donde fue registrada la nota del curso.
 - Código del curso: identificador del curso al que fue asignada la nota.
 - Nombre del curso: nombre del curso al que fue asignada la nota.
 - Nota: nota del curso.

4.1.6. Página de registro de curso

En esta página se solicitarán los datos para registrar un curso en el sistema.

Figura 37. **Página de registro de curso de la aplicación**



Fuente: elaboración propia, empleando Adonis JS 4.1.

Página donde se registran los datos de cursos, la página cuenta con los siguientes campos:

- Código del curso: cuadro de texto donde se ingresa el identificador del curso, este valor no puede repetirse en el sistema.
- Nombre del curso: cuadro de texto donde se ingresa el nombre del curso.
- Botón Registrar: botón con el que se guardan los datos ingresados en los campos anteriores.
- Cerrar sesión: botón para cerrar la sesión actual.
- Estudiante: botón que redirige a la página para registrar un nuevo estudiante.
- Curso: botón que redirige a la página para registrar un nuevo curso.
- Nota: botón que redirige a la página para registrar una nueva nota.

Los objetos cerrar sesión, estudiante, curso y nota aparecen en todas las páginas únicamente cuando se ha iniciado una sesión.

4.1.7. Página de registro de nota

En esta página se solicitarán los datos de los estudiantes y los datos del curso que se registrarán por medio de una calificación en el sistema.

Figura 38. Página de registro de nota de la aplicación

Fuente: elaboración propia, empleando Adonis JS 4.1.

Página donde se registran las notas de los estudiantes existentes en el sistema, la página cuenta con los siguientes campos:

- Carnet del estudiante: cuadro de texto donde se ingresa el id o carnet del estudiante a quien se le registrará la nota.

- Buscar: botón para buscar si el estudiante existe.
- Nombres: campo que muestra los nombres del estudiante buscado.
- Apellidos: campos que muestra los apellidos del estudiante buscado.
- Carrera: carrera del estudiante buscado.
- Fecha de nacimiento: fecha de nacimiento del estudiante buscado.
- Código del curso: cuadro de texto donde se ingresa el id del curso al cual se le asignará la nota.
- Buscar: botón que buscará si el curso existe.
- Nombre del curso: campo que muestra el nombre del curso.
- Nota: cuadro de texto donde se ingresará la nota del curso asignado al estudiante.
- Registrar: botón para guardar los datos ingresados.

4.2. Requisitos

Los requisitos definen las herramientas y tecnologías necesarias para el desarrollo e implementación de la solución, se deben utilizar las herramientas indicadas para poder ejecutar la aplicación.

4.2.1. Herramientas

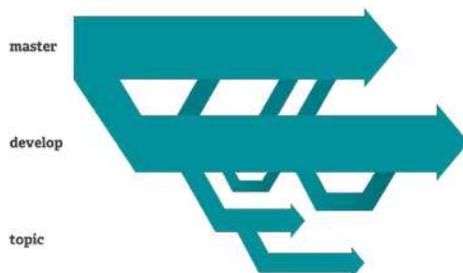
Las herramientas se utilizan para llevar control del código generado y versiones del mismo.

4.2.1.1. GIT

GIT es un sistema de control de versiones de código abierto, su principal valor es el modelo de ramificación que aplica, ya que permite tener múltiples

ramas totalmente independientes entre sí, las operaciones entre estas ramas son fáciles de realizar, entre estas: creación, fusión, eliminación.

Figura 39. **Esquema de ramas**



Fuente: CHACON, Scott & STRAUB, Ben. *Git*. www.git-scm.com/about/branching-and-merging.
Consulta: octubre de 2019.

Las operaciones, en su mayoría, se realizan localmente a través de comandos, motivo por el cual GIT es veloz en comparación con otros sistemas de control de versiones, los cuales son centralizados. Como ejemplo, escribiendo *git help* en consola, se obtiene una lista de los comandos y sus descripciones.

Figura 40. Comandos de Git

```
$ git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate] [--no-pager] [--no-replace-objects] [--noed]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone Clone a repository into a new directory
  init Create an empty Git repository or re-initialize an existing one

work on the current change (see also: git help everyday)
  add Add file contents to the index
  mv Move or rename a file, a directory, or a symlink
  reset Reset current HEAD to the specified state
  rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect Use binary search to find the commit that introduced a bug
  grep Print lines matching a pattern
  log Show commit logs
  show Show various types of objects
  status Show the working tree status

grow, mark and tweak your common history
  branch list, create, or delete branches
  checkout Switch branches or restore working tree files
  commit Record changes to the repository
  diff Show changes between commits, commit and working tree, etc
  merge Join two or more development histories together
  rebase Reapply commits on top of another base tip
  tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch Download objects and refs from another repository
  pull Fetch from and integrate with another repository or a local branch
  push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some concept guides.
See 'git help <command>' or 'git help <concept>' to read about a specific subcommand or concept.
```

Fuente: CHACON, Scott & STRAUB, Ben. *Useful Git commands*.

www.programmingmitra.com/2019/01/useful-git-commands.html.

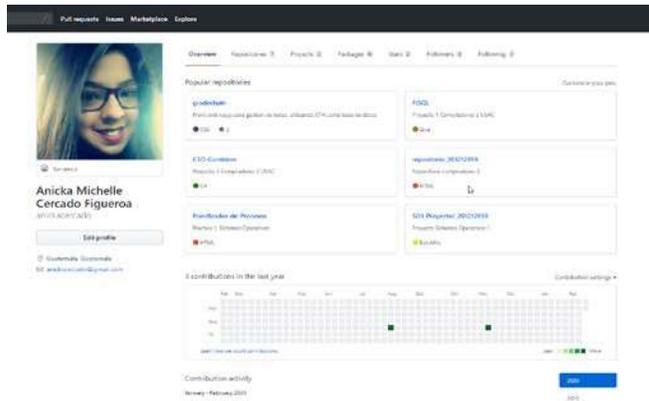
Consulta: octubre de 2019.

4.2.1.2. GitHub

GitHub es una herramienta de control de versiones, en ella los desarrolladores almacenan proyectos y cada proyecto contiene su propia URL por la que se accede al repositorio. Para el caso de GradeChain el repositorio del proyecto se encuentra en la URL <https://github.com/titus1993/gradechain-api>.

GitHub se caracteriza por su apariencia de red social, cada usuario tiene su propio perfil, en el se muestran sus proyectos y colaboraciones, además de esto cuenta con changelogs en los cuales se detallan todos los cambios realizados en el repositorio.

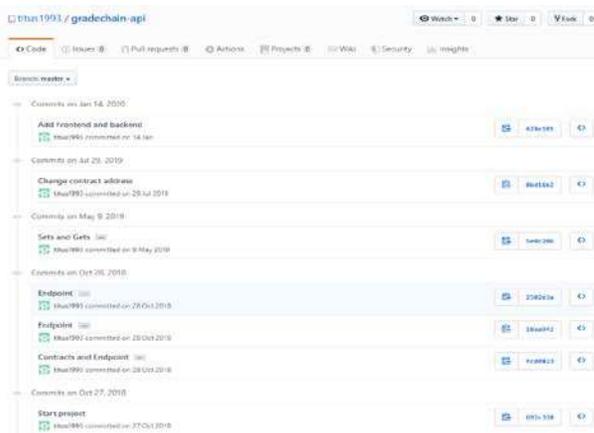
Figura 41. Perfil de GitHub



Fuente: PRESTON, Tom Werner., PRESTON, WANSTRATH, Chris., P. J. HYETT. & CHACÓN, Scott. *Perfil de GitHub Anicka Cercado*. www.github.com/anickacercado.

Consulta: octubre de 2019.

Figura 42. Detalle de cambios en GitHub



Fuente: PRESTON, Tom Werner., PRESTON, WANSTRATH, Chris., P. J. HYETT. & CHACÓN, Scott. *Detalle de cambios rama master*. www.github.com/titus1993/gradechain-api.

Consulta: octubre de 2019.

4.2.2. Hardware

Se define como las partes físicas de un dispositivo (computadora) por ejemplo: monitor, teclado, ratón, procesador, tarjeta de red, módulos de memoria RAM, impresoras, escáneres, entre otros. El hardware es dirigido mediante las instrucciones del software.

4.2.2.1. AWS

Es una plataforma en la nube que ofrece servicios de cómputo bajo demanda a individuos o empresas para crear infraestructura IT

4.2.2.1.1. EC2

EC2 es un servicio que brinda la plataforma de AWS, este servicio permite alquilar computadores virtuales para ejecutar aplicaciones, el usuario puede decidir las características de este equipo virtual como: procesadores, RAM, Almacenamiento, entre otros.

Para fines del proyecto se utilizaron dos procesadores, 1 Gb de RAM y 8 Gb de almacenamiento, utilizando como sistema operativo Ubuntu 18.04.

4.2.3. Software

Es un conjunto de instrucciones que manipulan el hardware, para que un dispositivo pueda funcionar correctamente tal es el caso de las aplicaciones de computadoras.

4.2.3.1. Backend

El Backend es la parte lógica que contiene el código encargado de procesar los datos.

4.2.3.1.1. Node.js

Node.js es un entorno de ejecución de JavaScript que se caracteriza por la construcción de aplicaciones escalables, este entorno está orientado a eventos asíncronos, para el efecto de GradeChain se utiliza para montar el framework AdonisJs.

Figura 43. Logo Node.js



Fuente: OpenJS. *Acerca de Node.js*. [www. nodejs.org/es/about](http://www.nodejs.org/es/about).

Consulta: octubre de 2019.

4.2.3.1.2. Web3

Es un API de Ethereum desarrollada en JavaScript para poder interactuar con un nodo de la red por medio de una conexión HTTP o IPC, a través de ella se pueden mandar comandos para realizar transacciones sobre la red, desde hacer transferencias entre Wallets hasta desplegar o ejecutar aplicaciones.

4.2.3.1.3. Infura

Es una infraestructura que provee servicios para la red de Ethereum y se caracteriza por ser el proveedor más popular de hospedaje de nodos, su principal función es asegurar que una aplicación descentralizada sea escalable para cumplir la demanda de los usuarios. Esta se utiliza por medio de API(s), que soportan JSON-RPC a través de interfaces HTTPS y WebSocket.

Figura 44. Logo Infura



Fuente: LUBIN, Joseph. *El proveedor de infraestructura de nodos de Ethereum, Infura, maneja 10 mil millones de solicitudes por día.* www.steemit.com/spanish/@jungleonion/el-proveedor-de-infraestructura-de-nodos-de-ethereum-infura-maneja-10-mil-millones-de-solicitudes-por-dia.

Consulta: octubre de 2019.

4.2.3.2. Frontend

El Frontend es la parte de la aplicación con la cual el usuario final tendrá interacción y esta le mostrará información.

4.2.3.2.1. AdonisJs

Es un framework MVC, está orientado al desarrollo web, utiliza el entorno de Node.js, templates personalizados, así como una configuración para API Full REST y un Server Side Rendering. AdonisJs es ideal para proyectos Fullstack, para efectos de GradeChain fue utilizado como Frontend, y se denomina DApp, entre sus características se encuentran:

- Una interface de línea de comandos (CLI)
- Autenticaciones
- Sistema modelo vista controlador (MVC)
- Un mapeo-objeto-relacional (ORM)
- Soporte incluido para internacionalización (i18n)
- Soporte de primera clase para pruebas
- Login social
- Sistema de plantillas
- Validaciones
- Servidor de sockets y cliente

Figura 45. **Logo AdonisJs**



Fuente: AdonisJS. *Adonis*. www.adonisjs.com.

Consulta: octubre de 2019.

4.2.4. Metodología SCRUM

Es una metodología de desarrollo ágil, la cual se utiliza para dirigir equipos que trabajan de forma iterativa incremental; el equipo está conformado por los desarrolladores, scrum master y product owner, adicional tienen distintos eventos como la planificación del sprint, scrum diario, revisión del sprint y retrospectiva del sprint; también cuenta con distintos elementos como producto backlog, sprint backlog y el incremento; cada una de las factores mencionados anteriormente aportan valor para que nuevas funcionalidades del producto puedan ser entregadas en pequeños lapsos de tiempo.

4.2.4.1. Planificación

Contiene los elementos para elaborar GradeChain, desde la documentación hasta los sprints de desarrollo en los cuales se libera una nueva versión del producto.

Tabla IX. **Planificación**

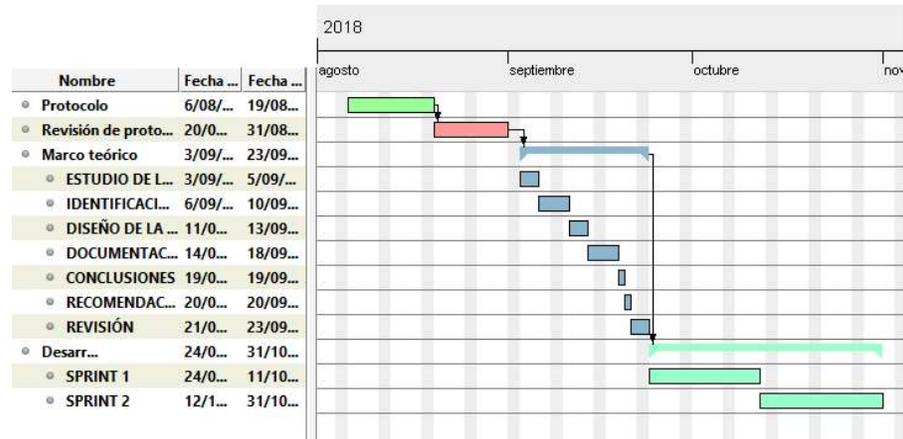
Marco teórico		
Nombre	Fecha Inicio	Fecha Fin
Protocolo	06/08/2019	19/08/2019
Marco teórico	03/09/2019	23/09/2019
Estudio de La tecnología y su impacto en Guatemala.	3/09/2019	5/09/2019
Identificación de problema y solución que la aplicación.	6/09/2019	10/09/2019
Diseño de la aplicación bajo la necesidad identificada.	11/09/2019	13/09/2019
Documentación y tutorial de programación de la aplicación.	14/09/2019	18/09/2019
Conclusiones	19/09/2019	19/09/2019
Recomendaciones	20/09/2019	20/09/2019
Desarrollo de software		
Nombre	Fecha Inicio	Fecha Fin
Sprint 1	24/09/2019	11/10/2019
Sprint 2	12/10/2019	31/10/2019

Fuente: elaboración propia.

4.2.4.2. Cronograma

Contiene el conjunto de elementos los cuales se definieron en la planificación, con un rango de fechas establecidas para ejecutarse.

Figura 46. Cronograma de actividades



Fuente: elaboración propia, empleando GanttProject 2018.

4.2.4.3. Product Backlog

Es la lista de las características que se necesitan para construir GradeChain, cada elemento de la lista se divide en historia de usuario, punto (s) de historia y prioridad de 1 a 3.

Tabla X. Product Backlog GradeChain

Product backlog GradeChain		
Historia de usuario	Punto (s) historia	Prioridad
Creación de contrato para la red de Ethereum que contengan las estructuras de datos estudiante y cursos.	11	1

Continuación de la tabla X.

<p>La aplicación debe poseer una página denominada Index, que debe contener logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados), por carnet y listado de los últimos registros de notas.</p>	<p>4</p>	<p>1</p>
<p>La aplicación debe poseer una página denominada Estudiante y debe contener logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados), por carnet, nombre, apellido, carrera, fecha de nacimiento y el listado de los cursos aprobados.</p>	<p>4</p>	<p>2</p>
<p>La aplicación debe poseer una página denominada Login, y debe contener logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados) por carnet, usuario, contraseña y botón de login.</p>	<p>4</p>	<p>2</p>
<p>La aplicación debe poseer una página denominada Registrar estudiante y debe contener logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, carnet, nombre, apellido, carrera, fecha de nacimiento, imagen y botón submit.</p>	<p>4</p>	<p>3</p>
<p>La aplicación debe poseer una página denominada Registrar curso y debe contener logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados), por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, id del curso, nombre del curso y botón submit.</p>	<p>4</p>	<p>3</p>

Continuación de la tabla X.

La aplicación debe poseer una página denominada Registrar nota y debe contener logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, objetos para hacer la búsqueda de estudiante, objetos para hacer la búsqueda de curso, nota y submit.	4	3
---	---	---

Fuente: elaboración propia.

4.2.4.4. Sprint 1

Es el intervalo de tiempo en el cual se desarrolló el incremento del producto.

4.2.4.4.1. Sprint Backlog

Contiene la lista de características, que se desarrollaran en una iteración, es decir en un sprint, el tiempo de cada sprint es de tres semanas.

Tabla XI. **Sprint 1 Backlog**

Sprint 1		
Historia de usuario	Sprint backlog	Prioridad
Creación de contrato para la red de Ethereum que	Definir la estructura de datos estudiante.	1
	Definir la estructura de datos cursos	1

Continuación de la tabla XI.

contengan las estructuras de datos estudiante y cursos.	Programar las variables dentro del contrato.	1
	Programar función agregar curso.	1
	Programar función agregar estudiante.	1
	Programar función agregar nota.	1
	Programar función obtener estudiante por id.	1
	Programar función obtener curso por id.	1
	Programar función obtener curso y estudiante por posición.	1
	Programar función obtener longitud lista de estudiantes.	1
	Programar función obtener longitud lista de cursos.	1
La aplicación debe poseer una página denominada Index, la cual debe contener logotipo, link hacia el login, buscador de	Realizar prototipo para la página Index .	1
	Realizar funciones de la página Index para la API.	2
	Realizar diseño de página web Index.	2

Continuación de la tabla XI.

estudiante (certificación de cursos aprobados) por carnet y listado de los últimos registros de notas.	Realizar controladores para la página web Index.	3
La aplicación debe poseer una página denominada Estudiante, esta debe contener logotipo, link hacia el login, buscador de estudiante (certificación de cursos aprobados) por carnet, nombre, apellido, carrera, fecha de nacimiento y el listado de los cursos aprobados.	Realizar prototipo para la página Estudiante.	1
	Realizar funciones de la página Estudiante para la API.	2
	Realizar diseño de página web Estudiante.	2
	Realizar controladores para la página web Estudiante.	3
La aplicación debe poseer una página denominada Login, la cual debe contener logotipo, link hacia el login, buscador de	Realizar prototipo para la página Login.	1
	Realizar funciones de la página Login para la API.	2
	Realizar diseño de página web Login.	2

Continuación de la tabla XI.

estudiante (certificación de cursos aprobados) por carnet, usuario, contraseña y botón de login.	Realizar controladores para la página web Login.	3
--	--	---

Fuente: elaboración propia.

4.2.4.5. Sprint 2

Es el intervalo de tiempo en el cual se desarrolló el incremento del producto.

4.2.4.5.1. Sprint Backlog

Contiene la lista de características, que se desarrollaran en una iteración, es decir en un sprint, el tiempo de cada sprint es de tres semanas.

Tabla XII. **Sprint 2 Backlog**

Sprint 2		
Historia de Usuario	Sprint backlog	Prioridad
La aplicación debe poseer una página	Realizar prototipo para la página Registrar estudiante.	1

Continuación de la tabla XII.

denominada Registrar estudiante, esta debe contener logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados), por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, carnet, nombre, apellido, carrera, fecha de nacimiento, imagen y botón submit.	Realizar funciones de la página Registrar estudiante para la API.	2
	Realizar diseño de página web Registrar estudiante.	2
	Realizar controladores para la página web Registrar estudiante.	3
La aplicación debe poseer una página denominada Registrar curso, la cual debe contener logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, id del curso, nombre del curso y botón submit.	Realizar prototipo para la página Registrar curso.	1
	Realizar funciones de la página Registrar curso para la API.	2
	Realizar diseño de página web Registrar curso.	2
	Realizar controladores para la página web Registrar curso.	3

Continuación de la tabla XII.

<p>La aplicación debe poseer una página denominada Registrar nota, y debe contener logotipo, link hacia el logout, buscador de estudiante (certificación de cursos aprobados) por carnet, link hacia registrar estudiante, link hacia registrar curso, link hacia registrar nota, objetos para hacer la búsqueda de estudiante, objetos para hacer la búsqueda de curso, nota y submit.</p>	Realizar prototipo para la página Registrar nota.	1
	Realizar funciones de la página Registrar nota para la API.	2
	Realizar diseño de página web Registrar nota.	2
	Realizar controladores para la página web Registrar nota.	3

Fuente: elaboración propia.

4.2.5. Tutorial de desarrollo y referencias

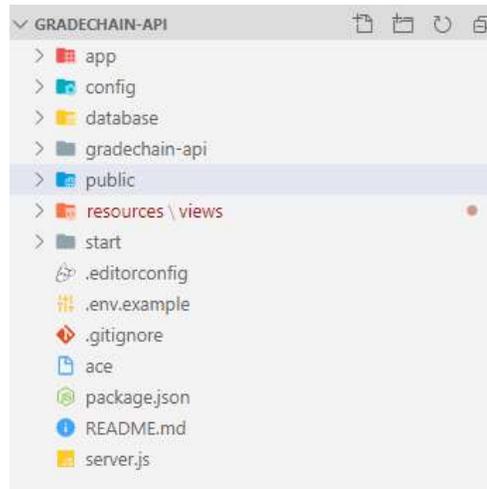
Se presenta la estructura y vista general de cómo se visualizará el código del proyecto.

4.2.5.1. Vista general del proyecto

La estructura del proyecto está compuesta por distintas carpetas, a continuación, se describe el contenido de cada una de ellas.

- Carpeta APP: contiene los contratos inteligentes los cuales fueron codificados en Solidity.
- Carpeta CONFIG: contiene las configuraciones necesarias de AdonisJs para el correcto funcionamiento de GradeChain.
- Carpeta DATABASE: contiene las configuraciones para la base de datos en este caso MySQL.
- Carpeta GRADECHAIN-API: contiene los archivos JavaScript para la realización de la API, la cual se encuentra directamente enlazada con Infura.
- Carpeta PUBLIC: contienen todos los archivos que contribuyen al diseño de cada vista de GradeChain.
- Carpeta VIEWS: contiene el diseño de cada vista de GradeChain.
- Carpeta START: contiene archivos de inicio para AdonisJs.

Figura 47. **Vista general del proyecto**

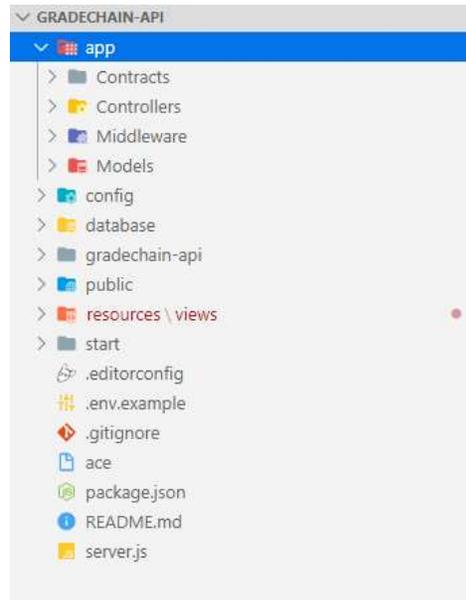


Fuente: elaboración propia, empleando Visual Studio Code 2019.

4.2.5.1.1. Carpeta APP

La carpeta APP se conforma por los contratos desarrollados en Solidity, los controladores, el mediador y los modelos.

Figura 48. **Carpeta APP**



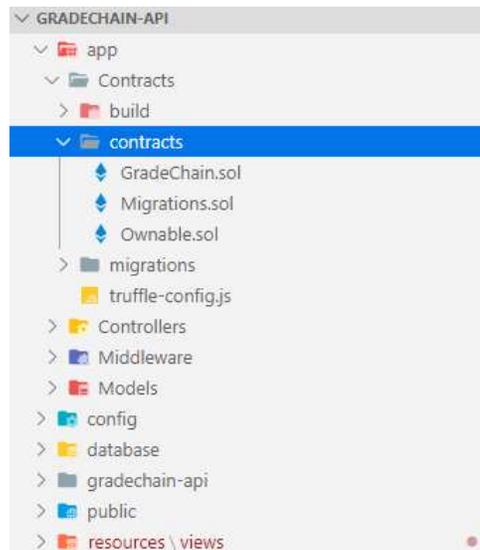
Fuente: elaboración propia, empleando Visual Studio Code 2019.

4.2.5.1.2. **Carpeta Contracts**

En la carpeta CONTRACTS se encuentran los archivos de Solidity (.sol) los cuales contienen la definición y funciones de los *Smart Contracts*, la carpeta está dividida en:

- GradeChain.sol
- Migrations.sol
- Ownable.sol

Figura 49. **Carpeta Contracts**



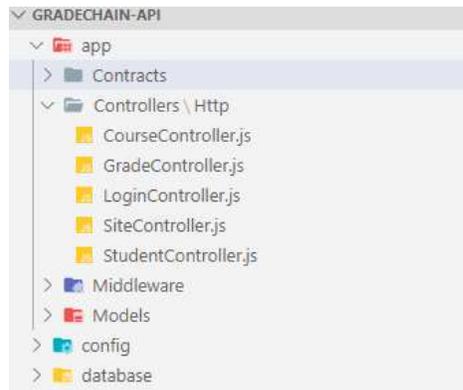
Fuente: elaboración propia, empleando Visual Studio Code 2019.

4.2.5.1.3. **Carpeta HTTP**

En la carpeta HTTP se encuentra la lógica de las páginas, la cual se encuentra dividida en archivos .js, montado sobre el framework AdonisJs, las cuales son estrictamente parte del frontend, la carpeta está dividida en:

- CourseController.js
- GradeController.js
- LoginController.js
- SiteController.js
- StudentController.js

Figura 50. **Carpeta HTTP**



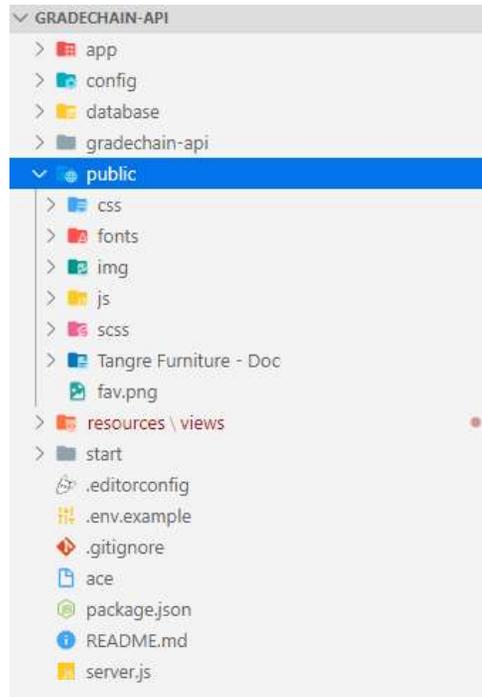
Fuente: elaboración propia, empleando Visual Studio Code 2019.

4.2.5.1.4. **Carpeta Public**

En la carpeta public se encuentra los archivos necesarios para el diseño de las páginas web, la carpeta está dividida en:

- Carpeta css: contiene los archivos CSS para el diseño de HTML.
- Carpeta fonts: contiene las tipografías del template utilizado.
- Carpeta img: contiene las imágenes públicas estáticas utilizadas en cada vista del proyecto web.
- Carpeta js: contiene archivos JavaScript para el manejo de diseños responsivos dentro del template utilizado.
- Carpeta scss: contiene los archivos SCSS para el diseño de HTML.
- Tangre Furniture – Doc: contiene la documentación del template utilizado.
- fav.png: contiene el logo de GradeChain.

Figura 51. **Carpeta Public**



Fuente: elaboración propia, empleando Visual Studio Code 2019.

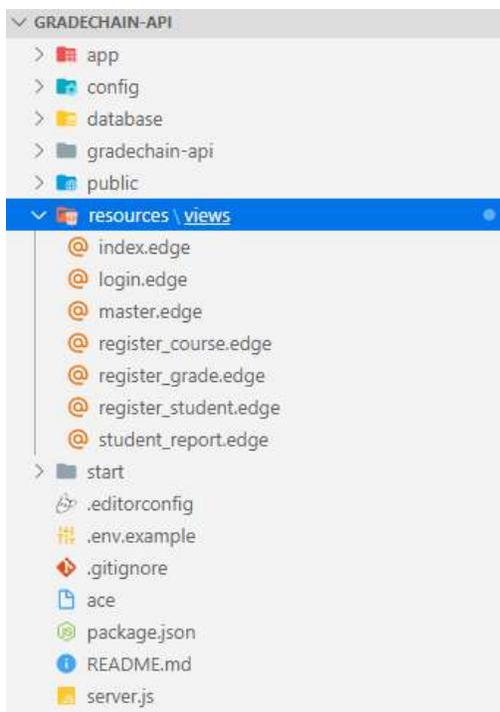
4.2.5.1.5. **Carpeta View**

Contiene cada vista diseñada en la sección de prototipos. Se compone de archivos edge, y se encuentran bajo el framework de AdonisJs, la carpeta está dividida en:

- index.edge
- login.edge
- master.edge
- register_course.edge

- register_grade.edge
- register_student.edge
- student_report.edge

Figura 52. Carpeta View



Fuente: elaboración propia, empleando Visual Studio Code 2019.

4.2.6. Consideraciones de implementación

En cuanto al versionamiento de Solidity, los cambios son muy significativos después de cada liberación, por lo tanto, se debe verificar que la versión de Solidity sea la adecuada para el proyecto, en este proyecto se utilizó la 0.4.25.

La billetera utilizada para los Ethers, en la cual se administran los gastos y consumos por transacciones, debe contener una contraseña única en un archivo aparte y esto no debe ser expuesto, para no estar expuestos a robos cibernéticos.

CONCLUSIONES

1. Se desarrolló la aplicación GradeChain, capaz de almacenar la información de los estudiantes de la Facultad de Ingeniería y sus cursos aprobados sobre una red descentralizada.
2. La aplicación permite interactuar, por medio de un *Smart Contract* con la cadena de bloques de Ethereum.
3. La aplicación GradeChain permite realizar consultas públicas de datos de estudiantes y listados de cursos aprobados, asegurando la integridad y fiabilidad de la información a través del uso de *Smart Contracts*.

RECOMENDACIONES

1. Utilizar datos encriptados entre el *Smart Contract* y la base de datos para tener un atributo adicional para mejorar la relación entre la información local y la almacenada en el Blockchain.
2. Analizar la escalabilidad del *Smart Contract* para poder hacer mejoras o agregar más información en él creando nuevas versiones.
3. Crear diferentes documentos para la consulta de graduaciones.
4. Impartir cursos de *Smart contracts* para que los interesados conozcan sobre la importancia y potencial de la aplicación.

BIBLIOGRAFÍA

1. Acerca de Node.js. *NodeJS*. [en línea]. <www.nodejs.org/es/about/>. [Consulta: octubre de 2019].
2. API de Ethereum e IPFS. *Infura*. [en línea]. <www.infura.io/>. [Consulta: octubre de 2019].
3. Etherscan. *Ropsten*. [en línea]. <www.ropsten.etherscan.io/>. [Consulta: octubre de 2019].
4. Fullstack sin dolor con Adonis.js. *AdonisJS*. [en línea]. <www.platzi.com/tutoriales/1151-nodejs/2555-fullstack-sin-dolor-con-adonisjs-2/>. [Consulta: octubre de 2019].
5. Geth. *Geth*. [en línea]. <www.github.com/ethereum/go-ethereum/wiki/geth>. [Consulta: octubre de 2019].
6. Getting Started. *Infura*. [en línea]. <www.infura.io/docs>. [Consulta: octubre de 2019].
7. GIT comandos básicos. *GIT*. [en línea]. <www.7sabores.com/blog/git-comandos-basicosi>. [Consulta: octubre de 2019].
8. How Do Ethereum Smart Contracts Work? *Smart Contracts*. [en línea]. <www.coindesk.com/learn/ethereum-101/what-is-a-decentralized-application-dapp>. [Consulta: octubre de 2019].

9. How to Develop a Stellar Scrum Product Backlog. *Product Backlog*. [en línea]. <www.lucidchart.com/blog/how-to-develop-a-product-backlog-in-agile>. [Consulta: octubre de 2019].
10. Introducción Adonis Framework. *AdonisJS*. [en línea]. <www.medium.com/@eaca230596/adonisjs-es-un-framework-orientado-al-desarrollo-web-basado-en-node-js-e907803fb8e4>. [Consulta: octubre de 2019].
11. It all boils down to confidence. *AdonisJS*. [en línea]. <www.adonisjs.com/>. [Consulta: octubre de 2019].
12. La Blockchain. *Blockchian*. [en línea]. <www.miethereum.com/blockchain/>. [Consulta: octubre de 2019].
13. Portal de Ingeniería. *Facultad de Ingeniería*. [en línea]. <www.portal.ingenieria.usac.edu.gt/>. [Consulta: octubre de 2019].
14. Solidity. *Solidity*. [en línea]. <www.solidity.readthedocs.io/en/v0.4.24/index.html>. [Consulta: octubre de 2019].
15. Truffle Overview. *Truffle*. [en línea]. <www.trufflesuite.com/docs/truffle/overview>. [Consulta: octubre de 2019].

16. Web3.js - Ethereum JavaScript API. *Web3.js*. [en línea]. <www.web3js.readthedocs.io/en/v1.2.6/>. [Consulta: octubre de 2019].
17. Welcome to the Ethereum Wiki! *Ethereum*. [en línea]. <www.github.com/ethereum/wiki/wiki>. [Consulta: octubre de 2019].
18. What Is GitHub, and What Is It Used For? *GitHub*. [en línea]. <www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>. [Consulta: octubre de 2019].