

“Año de la lucha contra la corrupción e impunidad”, nomenclatura que será utilizada como membrete en documentos oficiales del Estado”

PROYECTO

Trabajo Final

CURSO: Programación Concurrente y Distribuida (CC65)

Profesor: Canaval, Luis

Ciclo: 2019-2

Alumno:

Alosilla Lajo Juan Diego

Galarza Rosales, Pablo Jhoverson

Rojas Aliaga, Jordy Jhonatan

Monterrico, octubre 2019

**Índice**

[1. Objetivo del Estudiante 3](#_Toc23271647)

[2. Capitulo 1: Presentación 3](#_Toc23271648)

[3. Capitulo 2: Marco 3](#_Toc23271649)

[3.1. Marco Teórico 3](#_Toc23271650)

[3.2. Contexto del proyecto 3](#_Toc23271651)

[4. Capitulo 3: Gestión 3](#_Toc23271652)

[4.1. Evidencia de trabajo multidisciplinario 3](#_Toc23271653)

[4.2. Estimaciones de esfuerzo 3](#_Toc23271654)

[4.3. Plan de comunicación y colaboración usando nueva tecnología 3](#_Toc23271655)

[5. Capitulo 4: Implementación de solución 3](#_Toc23271656)

[6. Conclusiones 3](#_Toc23271657)

[7. Recomendaciones 3](#_Toc23271658)

[8. Glosario 3](#_Toc23271659)

[9. Bibliografía 3](#_Toc23271660)

# Objetivo del Estudiante

# Capítulo 1: Presentación

# Capítulo 2: Marco

* 1. Marco Teórico
  2. Contexto del proyecto

# Capítulo 3: Gestión

* 1. Evidencia de trabajo multidisciplinario
  2. Estimaciones de esfuerzo
  3. Plan de comunicación y colaboración usando nueva tecnología

# Capítulo 4: Implementación de solución

# Conclusiones

# Recomendaciones

# Glosario

# Bibliografía

**Sistema de transacciones.**

**Análisis:**

Un registro público que permita ser monitoreado por cada una de las partes involucradas, e incluso en el caso de fondos del Estado peruano pudiese ser accesible para todos los ciudadanos de la nación, permitiría que cada una de las transacciones realizadas sea reconocida y se pueda verificar su procedencia, su fin y destino; reduciendo así las posibilidades de actos de corrupción en gran magnitud.

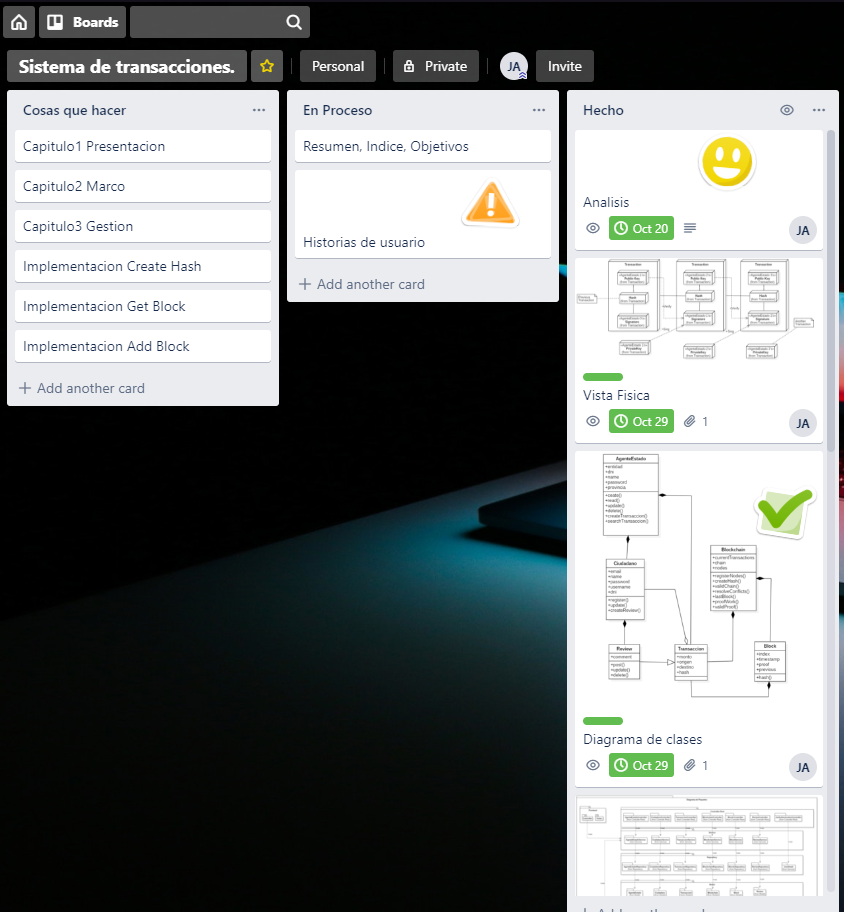
Blockchain puede darle un duro golpe a la corrupción, ya que las características de su tecnología representan una nueva forma más eficaz de preservar la transparencia en cualquier transacción.



**Organización de Tareas**

Para la organización de tareas se utilizó la herramienta trello.

<https://trello.com/b/epJ8IqoR>



**Hito 2**

1. **Elabora**

Artefactos de diseño para la solución.

* 1. **Historias de Usuario**

**Módulo de Transacciones**

|  |  |
| --- | --- |
| **Historia de Usuario #1** | **Código** HU01 |
| **Titulo** Listar transacciones | |
| **Prioridad** Media | |
| **Como** ciudadano **quiero** ver la lista de las transacciones realizadas en la aplicación **para** poder ver su información relevante | |
| **Condiciones** **de** **aceptación**   1. Estar en la aplicación de visitas de la empresa. | |

|  |  |
| --- | --- |
| **Historia de Usuario #2** | **Código** HU02 |
| **Titulo** Buscar por transacción | |
| **Prioridad** Media | |
| **Como** ciudadano **quiero** ver una determinada transacción realizada en la aplicación **para** poder ver en detalle dicha transacción. | |
| **Condiciones** **de** **aceptación**   1. Estar en la aplicación de visitas de la empresa. | |

|  |  |
| --- | --- |
| **Historia de Usuario #3** | **Código** HU03 |
| **Titulo** Registrar una transacción. | |
| **Prioridad** Alta | |
| **Como** administrador **quiero** registrar una determinada transacción de un presupuesto **para** tener conocimiento de la existencia de este. | |
| **Condiciones** **de** **aceptación**   1. Estar en la aplicación administrativa de la empresa. 2. Contar con un usuario administrativo. | |

|  |  |
| --- | --- |
| **Historia de Usuario #4** | **Código** HU04 |
| **Titulo** Actualizar una transacción. | |
| **Prioridad** Alta | |
| **Como** administrador **quiero** actualizar una determinada transacción de un presupuesto **para** corregir la falla cometida en una transacción realizada. | |
| **Condiciones** **de** **aceptación**   1. Estar en la aplicación administrativa de la empresa. 2. Contar con un usuario administrativo. | |

|  |  |
| --- | --- |
| **Historia de Usuario #5** | **Código** HU05 |
| **Titulo** Borrar una transacción. | |
| **Prioridad** Alta | |
| **Como** administrador **quiero** borrar una determinada transacción de un presupuesto **para** corregir la falla cometida en una transacción realizada. | |
| **Condiciones** **de** **aceptación**   1. Estar en la aplicación administrativa de la empresa. 2. Contar con un usuario administrativo. | |

* 1. **Requerimientos**

**Módulo de transacciones**

Modulo encargado de todos los servicios informativos de las transacciones con respecto a dos tipos de usuario, ciudadano y estado. Estos servicios permitirán listar todas las transacciones o asignaciones de presupuestos hechas por el estado, asimismo podrán ver a detalle cada una de estas transacciones. Este módulo se comunica con el módulo de comentarios.

**Módulo de seguridad**

Modulo esencial para cualquier sistema, encargado de proteger información que pueda ser sensible al uso mal intencionado; como también, delimitar los servicios y funciones para los dos tipos de usuarios según el rol que estos manejen. Este módulo no se relacionará directamente con otros módulos.

**Módulo de asignación de presupuestos**

Modulo encargado de manejar todos los servicios administrativos, los que permitirán realizar transacciones o asignación de presupuesto a distintas jurisdicciones dentro del estado o entidades públicos. Este módulo no se relacionará directamente con otros módulos.

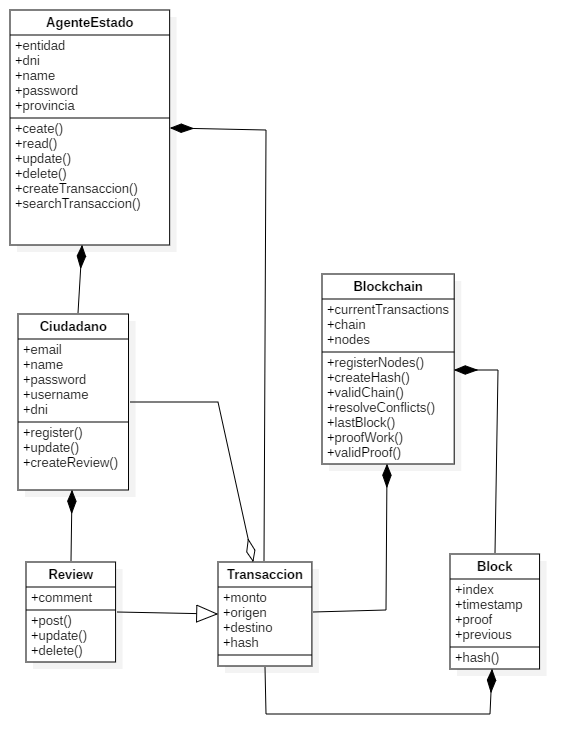
**Módulo de comentarios**

Modulo responsable tanto de publicar como listar comentarios en cada detalle de transacciones, por parte de los usuarios. Este módulo está relacionado con el módulo de transacciones.

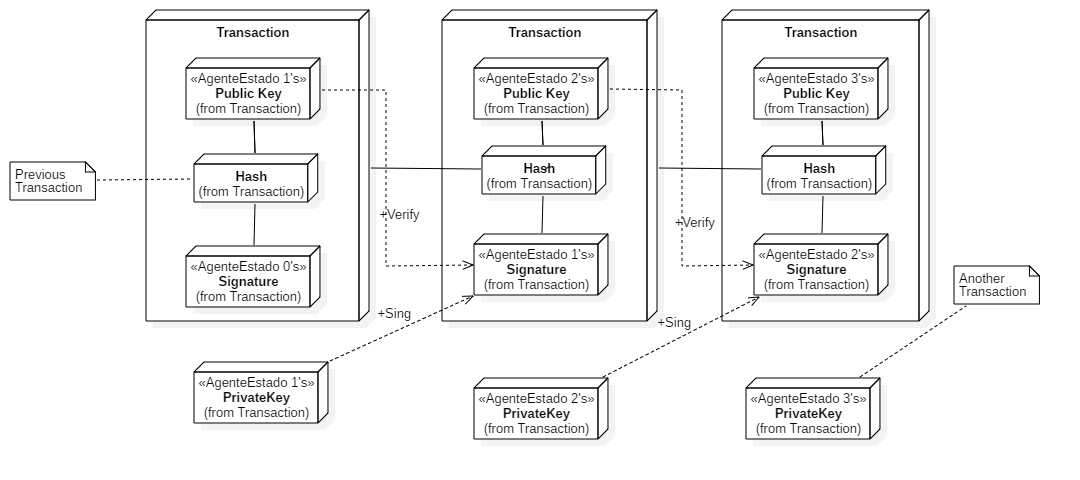
**Requerimientos Funcionales y No Funcionales**

|  |  |
| --- | --- |
| **REQUERIMIENTOS FUNCIONALES** | |
| **REQUERIMIENTOS** | |
| **CÓDIGO** | **DESCRIPCIÓN** |
| RF01 | * La aplicación deberá contar con una función que permita iniciar sesión ya sea como ciudadano o agente del estado. |
| RF02 | * La aplicación debe permitir que el ciudadano pueda agregar un comentario a una transacción. |
| RF03 | * La aplicación debe permitir que el ciudadano pueda ver un listado de las transacciones. |
| RF04 | * La aplicación debe permitir al agente del estado poder realizar una transacción o asignación de presupuesto a otros entes públicos o privados. |
| RF05 | * La aplicación debe permitir al agente del estado poder ver los registros de transacciones o asignaciones realizadas. |
| RF06 | * La aplicación debe permitir al agente del estado poder ver el detalle de una transacción o asignación realizada. |
| RF07 | * La aplicación debe permitir al usuario en general su registro en el aplicativo para su uso. |
| **REQUERIMIENTOS NO FUNCIONALES** | |
| **REQUERIMIENTOS** | |
| **CÓDIGO** | **DESCRIPCIÓN** |
| RNF01 | * La aplicación debe contar con un diseño de interfaz igual en todas sus pantallas; es decir, mismo tipo de letra y color. |
| RNF02 | * La aplicación debe permitir mostrar una o más fotos de referencia sobre el detalle de la transacción para que el ciudadano tenga una idea del propósito de esa transacción. |

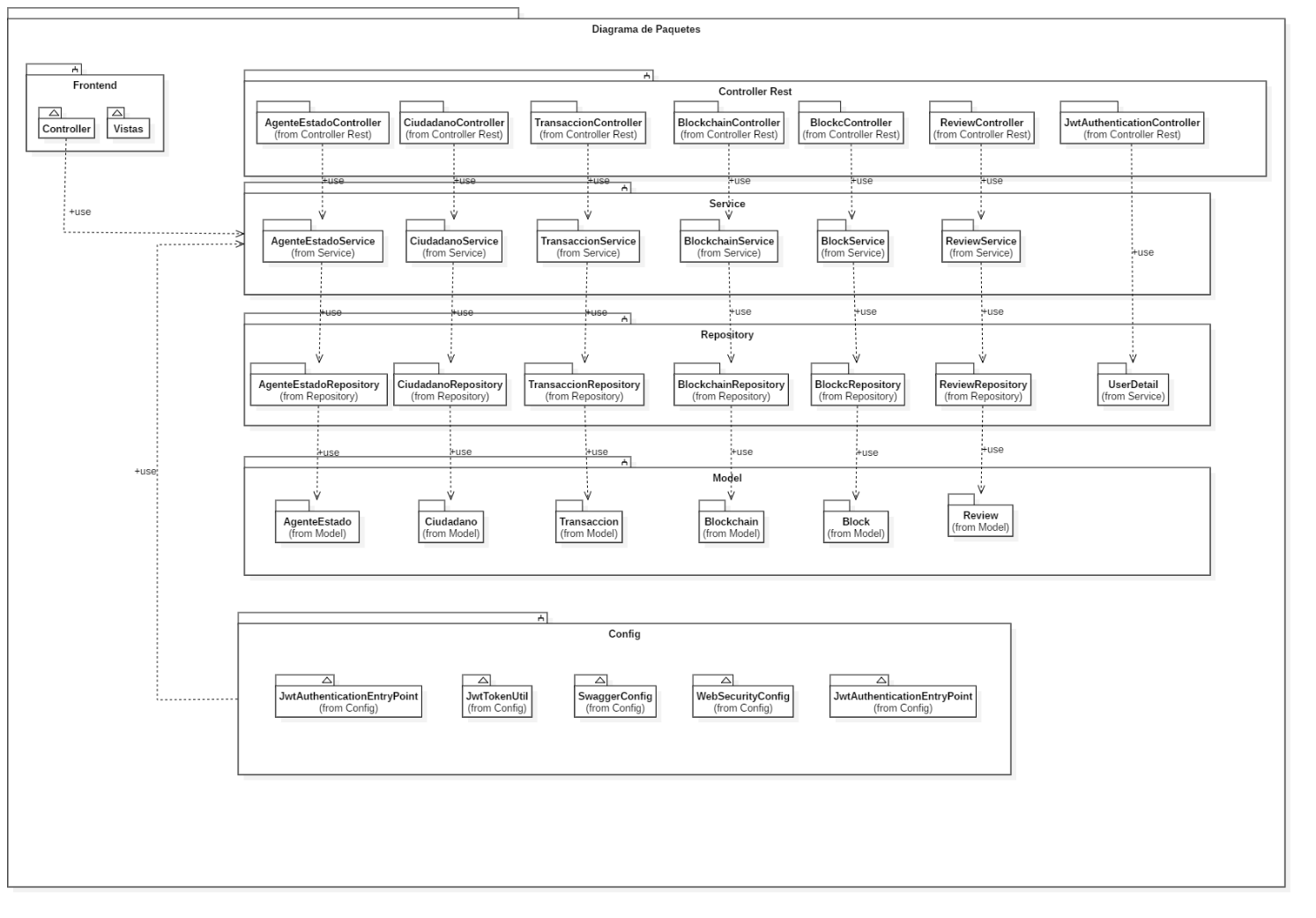
* 1. **Diagrama de clases**



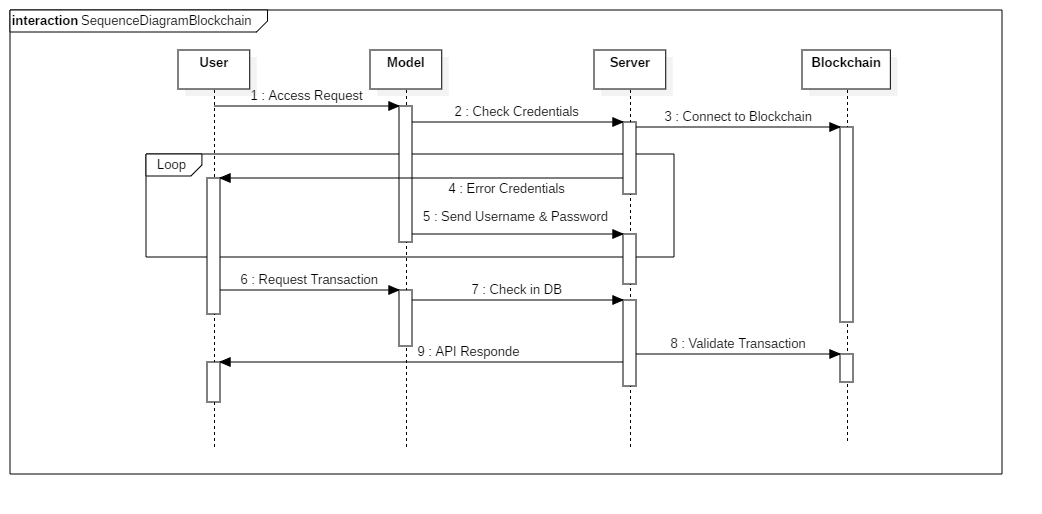
* 1. **Vista Física**



* 1. **Diagrama de paquetes**



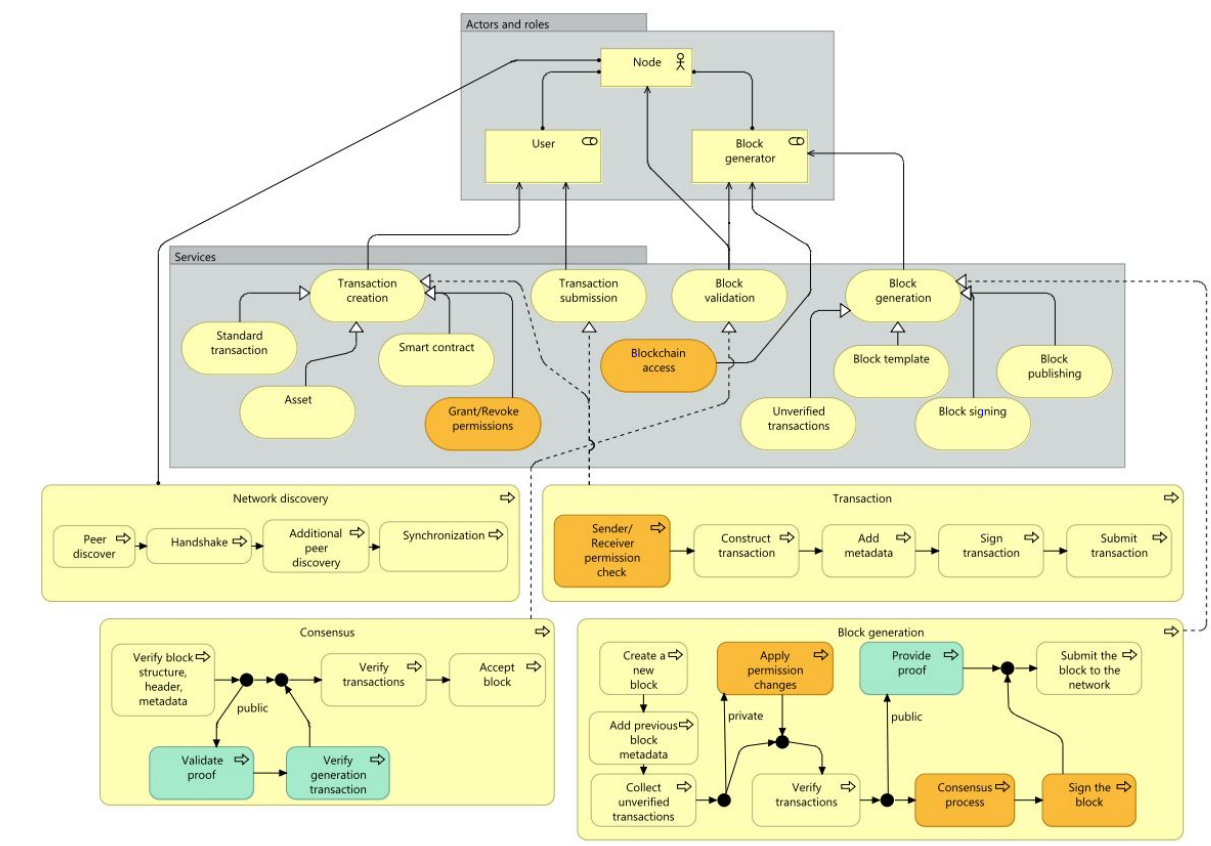
* 1. **Diagrama de secuencias**



* 1. **Casos de Uso**

|  |  |
| --- | --- |
| **DIAGRAMA DE CASOS DE USO** | |
| C:\Users\USUARIO\Documents\GitHub\Blockchain\src\UseCaseDiagramBlockchain.png | |
| **DESCRIPCION DEL DIAGRAMA DE CASOS DE USO** | |
| **ACTORES** | Ciudadano |
| Agente de Estado |
| Blockchain |
| **CASOS DE USO** | Obtener Transacciones |
| Obtener historial de Transacciones |
| Agregar comentarios |
| Obtener Hash |
| Obtener key |
| Generar transacción |
| Validad Transacción |

* 1. **Diagrama de componentes**



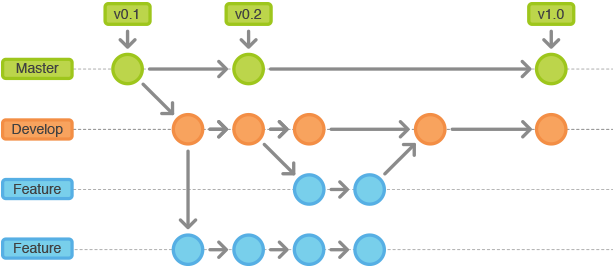
1. **Define**

Para el desarrollo del siguiente proyecto, la aplicación tendrá uso en el ámbito financiero, y permitirá que las transacciones sean lo mas seguras y trasparentes a fin de evitar que se realicen transacciones de origen desconocido.

Se tendrá como datos de entrada, el monto, el origen y el destino, y mientras se realice la transacción, se verificará con los otros nodos para corroborar que la información ingresada no sea de origen desconocido. El resultado esperado es que la validación se ejecute de manera distribuida y el registro de las transacciones se confirme en todos los nodos, de manera que la información este disponible desde cualquier nodo y que sea lo mas seguro posible.

1. **Configura**

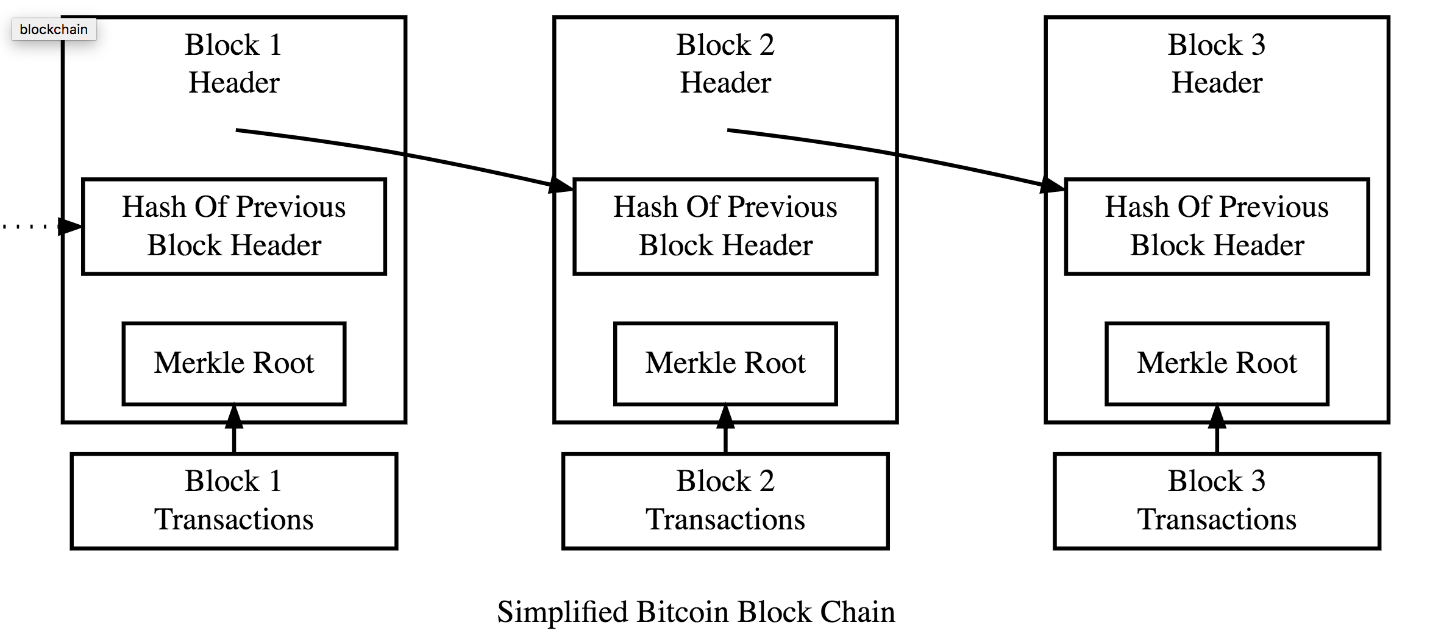
Git, hoy en día es uno de los sistemas de control de versiones más populares, por la flexibilidad de poder trabajar con ramas (branches) y agregar nuevas funcionalidades, sin poner en “peligro” de inconsistencias. Esa por eso que lo aplicaremos al proyecto.



1. **Implementación**

El proyecto fue implementado en el lenguaje Golang, para este proyecto se utilizo 3 estructuras, las cuales fueron, la estructura “Blockchain”, “Block” y la estructura “Transacción”. De manera que primero se instanciaba un “Blockchain”, luego se creaba el primer bloque que tiene como atributos un “Hash” y un “Previous Hash” (aparte de la información de una transacción). Al ser el primer bloque el “Previous Hash” se creaba usando una función para darle el valor de nulo. El “Hash” del primer bloque se obtenía mediante una función que encriptaba los datos del “Index”, el tiempo en el que fue creado y los datos de la transacción.

Al crear el segundo bloque este recibió como “Previous Hash”, el hash del primer bloque y así sucesivamente, donde cada vez que se creaba algún bloque se llamaba a la función de validación para realizar el consenso de información



Por otro lado, utilizamos la biblioteca de Gorilla/mux para poder hacer uso de las peticiones HTTP, de manera que se pueda realizar las transacciones de manera mas ágil.

1. **Conclusiones**