Polkdot: Interacción y operabilidad entre blockchains

Danilo Antonio Tovar Arias
Ingeniería de Sistemas
Universidad de Antioquia
Medellín, Antioquia
danilo.tovar@udea.edu.co

***Resumen*—En este artículo se realiza una exploración teórica sobre una tecnología emergente en mundo *blockchain* llamada *Polkadot*, enfocada en la interoperabilidad y escalabilidad entre diferentes tipos de *blockchain* existentes y nuevas, a través de la implementación de un sistema de cadena fragmentada y del desacoplamiento de mecanismos de creación y consolidación de bloques; así mismo se realiza una revisión práctica mediante la creación de un contrato inteligente en una red de prueba relacionada.**

***Palabras clave*—blockchain, polkadot, tecnología**

I. Introducción

En las últimas décadas, se ha experimentado un progreso acelerado en las tecnologías de la información. Desde la creación de dispositivos de comunicación global, como la televisión satelital, hasta sistemas integrados de información y comunicación, como los teléfonos inteligentes, la sociedad ha cambiado sus hábitos y métodos de gobernanza sobre la información y cómo ésta se distribuye alrededor del mundo.

Asimismo, en los últimos años han surgido una serie de tecnologías innovadoras que prometen revolucionar la forma en que vivimos una vez más, especialmente en los métodos de control de activos y la gestión de trazabilidad de acciones en internet. Una de estas tecnologías que ha avanzado lo suficiente como para generar interés en diferentes sectores globales es la *blockchain*, generalmente definida como una base de datos descentralizada. Sin embargo, representa un cambio significativo en la forma en que están construidos los sistemas actuales en la web, debido a sus diversas derivaciones y casos de uso para problemas relacionados con la seguridad, persistencia y privacidad de la información.

Aunque la *blockchain* es una tecnología con muchas oportunidades todavía siguen existiendo problemas que la limitan, por lo que se explorará, describirá y probará una *blockchain* emergente que propone solucionar problemas de interoperabilidad y comunicación efectiva entre *blockchains* ya existentes, llamada *Polkadot*. Y para entender mejor la relevancia de la *blockchain* y la necesidad de un sistema de interacción entre diferentes sistemas basados en esta tecnología, primero es fundamental comprender algunos conceptos básicos sobre ella.

II. Blockchains

La *blockchain* nace a partir de la motivación por crear un sistema de intercambio o transferencia de confianza entre usuarios anónimos sin la necesidad de un intermediario, y consiste en una base de datos con las transacciones que realizan los miembros de una comunidad descentralizada sobre sus activos, organizados en una lista de bloques encadenados secuencialmente [1]. En la primera *blockchain Bitcoin*, esto se logra a partir del uso de ciertos elementos:

- Transacciones, es decir, acciones de intercambio o modificación sobre los activos y su posesión, como realizar un pago o alterar el precio de un producto en un mercado.
- *Merkle Tree*, correspondiente a un proceso de combinación de pares de valores y su encriptación mediante un método como SHA-256.
- *Merkle root*, el valor final obtenido del *Merkle Tree*
- *Hash*, entendido como el resultado o acción de encriptar mencionada anteriormente.
- Prueba de trabajo, mejor conocido como *Proof of work*, corresponde a un proceso de consolidación del bloque sobre la cadena existente a través del uso de esfuerzo computacional para cumplir una condición.
- *Nounce*, número que cumple con la condición necesaria de la prueba de trabajo.
- Número único de bloque, también conocido como *block hash*, corresponde a la identificación de cada bloque en la cadena, y es generado a partir realizar un *hash* sobre la combinación del número del bloque anterior, el *Merkle root* del bloque actual y el *nounce* con el que se resolvió la prueba de trabajo.
- Nodos, servidores o computadores donde se realiza la interacción con la *blockchain*.

La primera *blockchain* lanzada en 2009, *Bitcoin*, utiliza los componentes anteriores además de un conjunto de reglas de validación de transacciones, protocolos de comunicación entre los diferentes nodos de la red, protocolos de consenso sobre el estado de la cadena y un sistema de recompensas basadas en participación, para cumplir con el objetivo de mantener un intercambio de activos seguro, anónimo, confiable y prácticamente irreversibles entre diferentes usuarios.

Con la creación de *Bitcoin*, comienza la revolución de la tecnología *blockchain* que se conoce en la actualidad como Web3, pues ”se inició una primera ola de descentralización en campos tan importantes como son los pagos, las transferencias internacionales y las remesas” [2]. A medida que ha

ido avanzando esta tecnología también se han desarrollado nuevos usos y mejoras sobre la idea inicial propuesta por *Bitcoin*, por ejemplo, el intercambio de los activos de una red por un valor equivalente en otra red, la creación de redes *blockchain* privadas para uso empresarial, la capacidad de crear contratos inteligentes que realizan tareas específicas, o aplicaciones sobre la red donde las interacciones con la misma sean almacenadas sobre una *blockchain* conocidas como *dApps* (*decentralized Applications*, en inglés), entre muchos otros.

Sin embargo, a pesar de los avances realizados en el desarrollo tecnológico con *blockchain*, existen problemas o limitaciones intrínsecas a esta tecnología expresados en [3] como:

- **Comunicación:** Entre diferentes redes es complicado hacer transferencia de información sobre transacciones realizadas sin intermediarios, debido a que la estructura de la información en cada *blockchain* es diferente y está generalmente ligada con el otros elementos dentro de la misma cadena.
- **Escalabilidad:** No se ha observado la aplicabilidad de la tecnología a gran escala ya que el promedio de transacciones por segundo (tps) de las *blockchains* es bajo dado los sistemas de validación, consolidación y consenso de los bloques, por ejemplo, *Bitcoin* posee una velocidad de aproximadamente 7 tps y *Ethereum* posee aprox. 25 tps, mientras que sistemas centralizados como Visa posee una velocidad de aprox. 1736 tps. Así mismo, debido a la necesidad de procesamiento de una gran cantidad de transacciones se producen comisiones y recargos excesivos sobre las transacciones que se realizan.
- **Seguridad:** Debido a que la seguridad de los bloques está asociada con la frecuencia y velocidad de creación de nuevos bloques, muchas redes nuevas son hackeadas por no tener una comunidad activa antes de su creación.
- **Personalización:** La creación de funcionalidades específicas para satisfacer las necesidades de las aplicaciones construidas en *blockchains* es complicada debido a que las cadenas poseen funcionalidades limitadas a su enfoque, lo que genera la posibilidad de necesitar empezar de cero con la creación de una *blockchain* con las capacidades requeridas.
- **Gobernanza:** El futuro de las redes *blockchain* están generalmente limitado por la falta de un sistema de gobernanza sobre la cadena, produciendo que las decisiones sean tomadas por un grupo central de usuarios o en ad hoc, ó incluso que no exista un toma de decisiones.
- **Actualización:** Las mejoras que se realizan sobre una red *blockchain* pueden causar divisiones en la comunidad o la cadena debido a que se generan bloques diferentes entre los usuarios que hacen uso de la versión anterior y la versión actualizada de la cadena, por lo que las aplicaciones deben de actualizarse con antelación para acomodarse a los cambios y evitar problemas cuando se actualiza la *blockchain*.

Este conjunto de dificultades son afrontadas por diferentes *blockchains* a través de cambios a la estructura o a los protocolos con los que funcionan; pero existe una alternativa que permitiría a los desarrolladores concentrarse en las aplicaciones que quieren realizar mientras que internamente estos problemas son resueltos de manera automática, correspondiente a *Polkadot*.

III. Polkadot

A. Visión

Polkadot corresponde a una *blockchain* de capa-0, es decir, una infraestructura diseñada para soportar y facilitar procesos de interacción y operación entre diferentes *blockchain* capa-1 como Bitcoin o Ethereum. Inicialmente en [4] Wood expresa que la estrategia que adopta Polkadot para crear una plataforma que permita la escalabilidad de sistemas de computo descentralizados consiste en desacoplar la arquitectura de consenso y el mecanismo de transición de estados, y el protocolo a utilizar para cumplir con este propósito mantiene el mismo nombre. Así mismo, Wood expresa que la propuesta inicial es experimental y funciona como un punto de partida para el desarrollo de mejoras y cambios a partir de las opiniones, criticas e ideas que surjan desde la comunidad al ser un proyecto abierto, es decir, *Polkadot* esta diseñado para ser un banco de pruebas de desarrollo, implementación e interacción de *blockchains* totalmente extensibles y escalables.

B. Polkadot protocol y soluciones

Siguiendo con lo expuesto en [4] *Polkadot* es una multicadena heterogénea escalable, es decir, a diferencia de implementaciones anteriores de *blockchain* que se han centrado en proporcionar una única cadena con diversos grados de generalidad para la realización de aplicaciones, *Polkadot* en sí está diseñado para no proporcionar ninguna funcionalidad inherente a la aplicación. En su defecto, *Polkadot* proporciona la "cadena de retransmisión" (*Relay chain*, en inglés) sobre la cual se pueden alojar una gran cantidad de estructuras de datos dinámicas validables y globalmente coherentes. A estas estructuras de datos las llamamos cadenas "paralelizadas" o *parachains*, aunque no existe una necesidad específica de que sean de naturaleza *blockchain*.

Estos componentes corresponden a la base fundamental sobre la que funciona *Polkadot*, y permiten dar solución a los problemas de *blockchain* expuestos anteriormente cuando se combinan con algunos aspectos más técnicos que se explicarán en la siguiente sección, permitiendo según [3] las siguientes posibilidades:

- Conectar diferentes redes: A través de *blockchains* integradas al sistema de *Polkadot* por medio del método *sharding* correspondiente a una separación de datos únicos entre diferentes nodos; adicionalmente el uso de *bridges* permiten la comunicación con redes tradiconales como *Bitcoin*.
- Habilita la posibilidad de creación de redes personalizadas: *Blockchains* en *Polkadot* son construidas para

cumplir con propósitos específicos, donde se puede escoger las áreas en las que se desempeñará y modificarlas para cumplir con las necesidades de las aplicaciones.
- Maneja tráfico pesado a escala: La capacidad teórica máxima de transacciones por segundo de la red es mayor a 166.000 tps, con estudios más recientes presentando a posibilidad de superar las 500.000 tps, mientras que Visa posee una capacidad máxima de 65.000 tps.
- Revoluciona la governanza en línea a través de un sistema abierto y dirigido por la comunidad: A través de un sistema de propuestas y votación similar a un sistema democrático donde la comunidad que hace uso de *Polkadot* puede participar.
- Seguridad lider en la industria: Permite a las *blockchain* integrarse al sistema de seguridad compartida de *Polkadot*, sin necesidad de desviarse del propósito para el que son creadas.
- Actualización propia de la red: *Polkadot* puede actualizarse sin dividir la comunidad o las *blockchains*, debido a su capacidad de realizar el proceso de manera interna sin la necesidad de acciones manuales por parte de sus usuarios.

C. Arquitectura

La arquitectura de *Polkadot* está formada por diferentes componentes especializados para cumplir con el objetivo de su creación, a partir de la información en [4]–[6] estos son:

1) ***Relay chain:*** Como se ha mencionado anteriormente, corresponde a la base principal de *Polkadot* y funciona como un centro de operaciones para las *parachains*, donde se coordina y asegura el correcto funcionamiento de la red. Así mismo, cabe destacar que la *Relay chain* posee un número de espacios o *slots* de procesamiento disponibles para *parachains*, similar a como funcionan los núcleos en el procesador de una computadora, de manera que las actividades que realizan las *parachains* hacen uso de esta potencia de computo para lo que necesiten, y solo requieren generar una constancia validable sobre los cambios generados en la cadena interna; estos cambios se almacenan en bloques llamados *parablocks* pero estos no son almacenados por la *Relay chain*, en cambio se utilizan las cabeceras de los bloques definidas como *para-headers*.
2) ***Parachain*:** Definidas anteriormente como estructuras de datos validables y coherentes, son creadas a partir de la necesidades especificas de la aplicación que soportan para cumplir con diferentes implementaciones o funciones; y gracias a su naturaleza pueden paralelizar el procesamiento de transacciones y alcanzar gran escalabilidad. De igual manera, poseen la capacidad de heredar la seguridad económica de la *Relay chain*, debido a la concentración de incentivos en la misma a través de los validadores, que se encargan de asegurar los cambios de estado de las *parachains* conectadas.
3) **DOT:** Token nativo de la red *Polkadot* similar a BTC para *Bitcoin* o Ether en *Ethereum*; utilizado para cumplir con diferentes funcionalidades y decisiones en la red.
4) **Actores:** *Polkadot* hace uso de diferentes nodos que poseen responsabilidades especificas y cumplen con los siguientes roles:
   - ***Validadores (Validators):*** Encargados de incluir, verificar y autenticar nuevos bloques en la *Relay chain* a través de dos etapas de validación llamadas *Inclusion Pipeline* y *Approval proccess*. Estos nodos requieren comprometer DOT, validar constancias de *parachains* y definir el consenso con otros validadores; así mismo requieren de un alto grado de disponibilidad y estabilidad. Sin embargo, no todos los validadores son utilizados en el proceso de consenso, pues deben tener una cantidad mínima de depósitos de DOT para ser parte del sistema.
   - ***Coladores (Collators):*** Son nodos completos de la *Relay chain* y una *parachain*, y se encargan de recolectar transacciones de la *parachain* y producir constancia de transiciones de estado para los validadores de la *Relay chain*. También pueden enviar y recibir mensajes de otras *parachains* por del uso de XCM *Passing*.
   - ***Nominadores (Nominators):*** Apoyan un conjunto de validadores comprometiendo DOT sobre los resultados entregados, tal que reciben recompensas si las acciones que realizan son correctas ó pierden DOT si los validadores nominados actúan de manera incorrecta.
   - ***Pescadores (Fisherman):*** Monitorean las transacciones incluidas en los bloques de los coladores para detectar aquellas invalidas o incorrectas, sin embargo son nodos deprecados en la versión actual de *Polkadot* y su funcionalidad ha sido delegada a los validadores.
5) **NPoS:** Abreviación de *Nominated Proof of Stake*, consiste en el proceso de ”apostar” DOT en las acciones de los validadores en conjunto con los nominadores para fortalecer la seguridad de las red, así mismo promueve a la creación de nuevos validadores debido a las recompensas por honestidad de los mismos y permite a los validadores ser seleccionados para realizar el proceso de consenso. Cabe mencionar que ”apostar” tokens se conoce como *bonding*, y estos tokens se les llama *staked* o *bonded* tokens; y similarmente la acción de soportar a un validador o grupo de validadores se define como *backing* o *nominating* validadores.
6) **XCM:** *Cross Consensus Messaging* corresponde al formato y lenguaje de comunicación común y genérico utilizado por los coladores para comunicarse entre diferentes sistemas de consenso, y define XCM *Passing* (XCMP) como protocolo de envío de mensajes en la red entre *parachains* y estan relacionados igual que REST y RESTful. En particular, funciona a través de un sistema

de colas de entradas y salidas en las *parachains* que contienen los mensajes enviados (*outbound*) y recibidos (*inbound*) para sus debidas ejecuciones al interior de la *parachain* y verificación por los coladores y validadores.

7) **Protocolos de consenso:** En *Polkadot* existen un protocolo de consenso híbrido con el propósito mencionado en III-A de separar la generación de nuevos bloques y la consolidación de los mismos, formado por:

   - **GRANDPA:** Por las siglas de *GHOST-based Recursive ANcestor Deriving Prefix Agreement*, es el mecanismo de consolidación o *finality* de bloques en la cadena de *Polakdot*, donde todos los nodos validadores realizan múltiples rondas de votación sobre la mejor cadena que conocen, enviando el último bloque de dicha cadena y finalmente el protocolo escoge el bloque en común más "alto" encontrado en la mayoría, correspondiente a 2/3 de los nodos. Esta votación se realiza de manera paralela a la producción de nuevos bloques permitiendo agilizar los procesos de consenso; y de igual manera, al ser un proceso realizado sobre las mejores cadenas posibles, se puede consolidar una gran cantidad de bloques de manera simultánea.
   - **BABE:** Entendido como *Blind Assignment for Blockchain Extension*, es el mecanismo de producción de nuevos bloques utilizado entre los validadores, y está basado en un algoritmo llamado *Ouroboros Praos*. De manera resumida, BABE utiliza un sistema de selección aleatorio separado en ejecuciones secuenciales llamadas etapas (*epochs*), que están dividas en secciones (*slots*) de tiempo de seis segundos, tal que al inicio de cada etapa se definen validadores potenciales para cada *slot*. Por otro lado, los validadores participan en un proceso de lotería (a través de una función aleatoria verificable, VRF) para determinar los *slots* en los que realizar producción de nuevos bloques; debido a que la lotería es un proceso aleatorio pueden existir *slots* con múltiples validadores o cero validadores asignados.
     - Cuando no hay validadores para un *slot*, se escogen de un proceso secundario de selección donde al menos un validador es asignado, estos validadores de respaldo siempre producen bloques, categorizados como bloques secundarios, y son ignorados si para el mismo *slot* existen bloques primarios.
     - Cuando existen múltiples validadores, estos compiten entre si para propagar su bloque a la mayor parte de la red primero, en algunos casos se pueden generar bifurcaciones y ambas se alargan hasta que se toma una decisión en la consolidación de la cadena.

     Las cadenas producidas por los bloques nuevos son agregadas al último bloque consolidado por el protocolo GRANDPA, y las cadenas no consolidadas posterior al consenso de los validadores son eliminadas.

Así, al combinar los dos protocolos anteriores se puede observar en la Fig. 1. como se toman las decisiones sobre las divisiones de la cadena, de manera que la selección siempre se realiza sobre la cadena más larga de bloques primarios sobre el ultimo bloque consolidado.

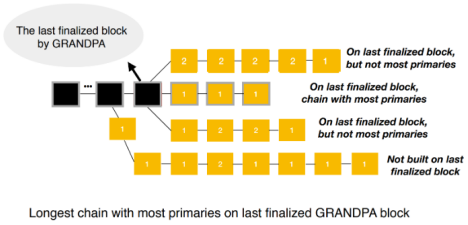

Fig. 1. Decisión en divisiones internas de la cadena

8) ***Bridges*:** Son el método que permite la comunicación entre la *Relay chain* y redes *blockchain* externas como Bitcoin o Ethreum. Estos sistemas de comunicación entre *blockchains* son bastante inseguros, al estar relacionados con un gran número de incidentes de hackeo debido a la dependencia con intermediarios centralizados, por ello *Polkadot* utiliza diferentes componentes en cadena y fuera de la cadena para fortalecer la seguridad y la eficiencia en la comunicación de *bridges*, como serían:

   - **BEEFY:** *Bridge Efficiency Enabling Finality Yielder* corresponde a un protocolo secundario de GRANDPA que permite la comunicación eficiente entre la *Relay chain* y las redes externas a través de la comprobación de firmas en bloques consolidados en la *Relay chain*, es decir, mediante clientes ligeros se realiza una revisión sobre el estado o bloque necesario y que contenga las firmas necesarias como constancia para las redes externas, evitando así la necesidad de realizar un análisis sobre las justificaciones realizadas por GRANDPA o sobre las bifurcaciones de la cadena.
   - ***Bridge Pallets*:** Herramienta utilizada para comunicar redes de *blockchain* basadas en *Substrate* (*framework* base de *Polkadot*), a través de una instancia de *Substrate* ejecutada a nivel del sistema *Polkadot* como *parachain* permitiendo recibir mensajes de la cadena externa, similarmente para recibir mensajes en la cadena externa desde *Polkadot* se requiere una instancia de *Substrate* que reciba estos mensajes.
   - ***Smart contracts*:** En casos donde la *blockchain* externa no esté basada en *Substrate* pero tiene capacidades de ejecución de contratos inteligentes, se utilizan clientes ligeros de la cadena en *Polkadot* para obtener constancias mediante contratos, y las cadenas externas pueden obtener constancias a

través del protocolo BEEFY, un ejemplo de uso corresponde a *Snowbridge* para la conexión entre *Ethereum* y *Polkadot*.

- ***Higher-order protocols*:** Cuando no hay otras opciones disponibles debido a que la *blockchain* externa no tiene capacidades de ejecución de contratos inteligentes y no está basada en *Substrate*, se utilizan protocolo especializados de alto orden para establecer la comunicación entre cadenas, como caso particular se puede observar de XCLAIM para la conexión entre *Polkadot* y *Bitcoin*.
- **Componentes fuera de la cadena:** Existen componentes externos que están conectados a los nodos correspondientes de cada cadena y se encargan de trasmitir los mensajes sobre los cambios y los mensajes de confirmación entre origen y destino.

Una parte de los elementos enumerados anteriormente se pueden ver representados en la Fig. 2. tal que un conjunto de nodos coladores reciben y procesan transacciones externas para ser enviadas a los validadores, luego los validadores se encargan de realizar las verificaciones necesarias para ser agregadas a nuevos bloques de la *Relay chain*, y los cambios son transmitidos a las cadenas conectadas correspondientes y los cambios necesarios son ejecutados sobre las cuentas dentro de las mismas. Así mismo se puede observar una *parachain* que funciona como puente de comunicaciones con la red de *Ethereum*.

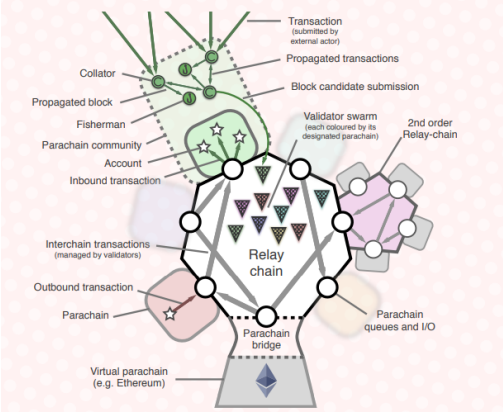

Fig. 2. Esquema resumido del sistema *Polkadot*

D. Información adicional

La descripción realizada sobre la arquitectura de *Polkadot* es generalizada y no abarca explicaciones sobre las funcionalidades especificas de casos de uso, detalles particulares de la estructura y diferentes componentes adicionales creados alrededor de la misma. Sin embargo, existe información relevante a compartir para una mejor comprensión sobre la magnitud del sistema de *Polkadot*:

- ***Kusama*:** Es la *canary network* de *Polkdot*, es decir, es una red *blockchain* activa y establecida sobre la cuál se despliegan actualizaciones y realizan cambios de manera constante para probar su comportamiento en un ambiente de producción; esta red posee su propio token llamado KSM, así como diferentes cambios con respecto sobre las limitaciones y los tiempos de ejecución en comparación con *Polkadot*, y además posee múltiples funcionalidades asociadas a la red *Polkadot*, como un *bridge* entre las dos redes.
- ***Parachain Slot Auctions*:** Como se había mencionado en la sección III-C *Polkadot* contiene una serie de espacios dedicados para conectar *parachains*, sin embargo existe una cantidad máxima que aumenta periódicamente para *parachains* de uso constante, por lo que se realiza una subasta sobre el espacio de la *parachain*, esta subasta no es una subasta cualquiera pues tiene un conjunto de periodos de ofertas y condiciones de decisión diferentes; de acuerdo a [7] se preparan ofertas durante un periodo de tiempo determinado (*opening period*) y la decisión sobre quien gana se realiza de forma aleatoria sobre un tiempo dado después del tiempo de ofertas (*ending period*), de modo que cada bloque tiene un conjunto de ofertas y un ganador de la subasta en cada bloque, y durante el periodo de decisión se selecciona uno de estos bloques tal que la probabilidad de ganar de una persona corresponde al número de bloques en los que gana dicha persona entre la cantidad de bloques totales durante el proceso de decisión.
- ***Agile Coretime*:** Inicialmente *Polkadot* asignaba un *"core"* o núcleo de procesamiento por *parachain*, sin embargo esto produce problemas de rendimiento al producir bloques vacíos y limitando la posibilidad de uso de la red para grupos medianos o pequeños que tienen que competir con aquellos que son propietarios del espacio. Para solucionar estos problemas, se hace uso de *Agile Coretime* que, según [6], es una tecnología en desarrollo que permite aprovechar en mayor medida la capacidad de computo disponible en la red como se observa en la Fig. 3., a través del uso preciso de tiempos de computo por diferentes *parachains*, aplicaciones y sistemas adicionales con la premisa de utilizar los núcleos como recurso para todos, y permitiendo la compra-venta de tiempos mínimos de uso sobre los núcleos de la red incluso para la producción de un único bloque mensual.

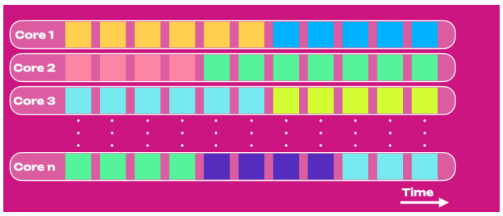

Fig. 3. Esquema de funcionamiento para *Polkadot's Agile Coretime*

- ***Polkadot-JS*:** Corresponde a un conjunto de herramientas, utilidades y librerias utilizadas para la interacción con *Polkadot* desde *JavaScript*, enfocada principalmente para desarrolladores.

- ***OpenGov*:** Definido en [8] como un mecanismo sofisticado de gobernanza, corresponde a un conjunto de métodos y funcionalidades de votación y decisión sobre los cambios y propuestas que se establecen alrededor y sobre la red *Polkadot*, donde la mayor parte de los participantes de la red tienen control sobre el futuro de la misma.
- ***Testnets*:** Existen un par de redes de prueba que se utilizan para experimentar funcionalidades y desarrollar prototipos para el aprendizaje del funcionamiento de la red o como versión preliminar de despliegues en la red principal, correspondientes a *Westend* (*testnet* oficial de *Polkadot*) y Paseo (*testnet* de la comunidad y mantenida por la misma).

IV. Implementación de un contrato inteligente en una Testnet de Polkadot

Para comprender mejor los aspectos técnicos de la red *Polkadot* se realizará una implementación sencilla sobre una *parachain* de *Paseo* llamada *POP* siguiendo una serie de pasos tomados de diferentes recursos [**?**], [9]–[13] como guía de desarrollo:

1) **Instalación de WSL:** Como se muestra en la Fig. 4, se requiere instalar una instancia de Ubuntu para la ejecución del proceso de instalación de dependencias y la creación del contrato inteligente en Windows, abriendo una ventana de Powershell y utilizando:

wsl - -install

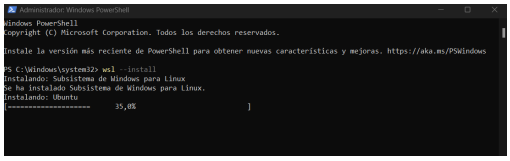

Fig. 4. Instalación de WSL

2) **Configuración inicial:** Realizar los pasos correspondientes en [9], [10] para definir el ambiente de desarrollo, a través de la instalación de múltiples dependencias como git, openssl, clang, curl, rust, pop, entre otras; se puede ver el progreso en las Fig. 5, 6 y 7.

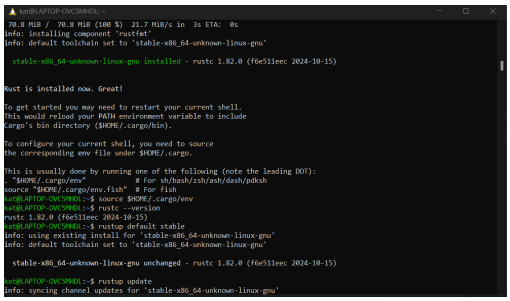

Fig. 5. Progreso de configuración inicial 1

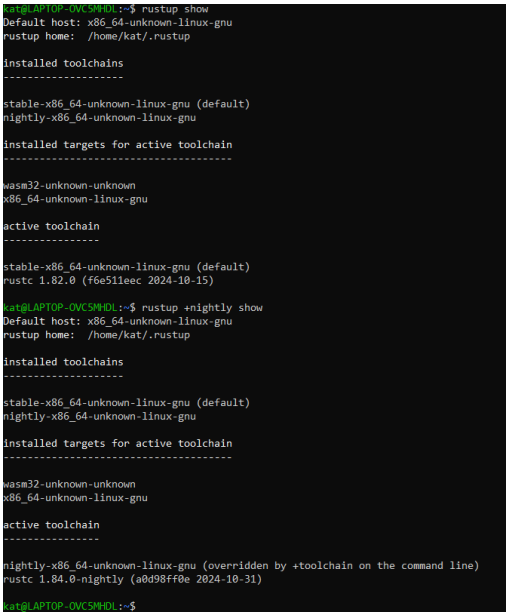

Fig. 6. Progreso de configuración inicial 2

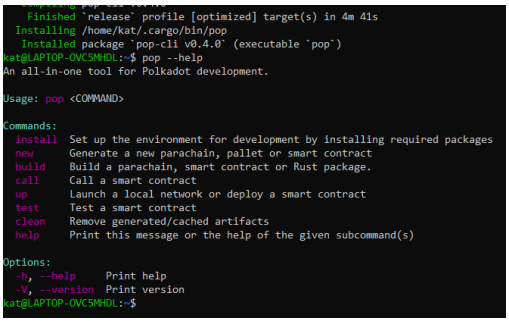

Fig. 7. Progreso de configuración inicial 3

3) **Creación del contrato:** Una vez obtenidas las dependencias, como se explica en [11] se puede crear un contrato vacío utilizando:

pop new contract myContract

Generando una carpeta con el mismo nombre (myContract) y un par de archivos relacionados al contrato.

4) **Definir funcionalidad del contrato:** Utilizando un elemento en la carpeta del contrato llamado *lib.rs* se pueden modificar las acciones que realiza el contrato, por simplicidad se utiliza la funcionalidad base en la creación del contrato correspondiente al almacenamiento de un booleano.
5) **Compilar el contrato:** Para obtener los archivos necesarios para el despliegue se realiza una compilación del contrato utilizando el siguiente comando, indicado en [13] dentro de la carpeta del mismo:

pop build

Al final del proceso de compilación, se obtiene un resultado como en Fig. 8, con la creación de diferentes carpetas y archivos, los más importantes son myContract.contract, correspondiente al ABI del contrato, es

decir, aquel que suministra la estructura general del mismo y permite su utilización en la red, y myContract.wasm, que es utilizado en otras cadenas.

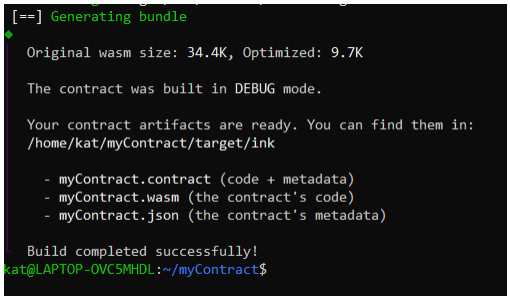

Fig. 8. Compilación del contrato

6) **Despliegue del contrato:** Siguiendo los pasos indicados en [13], se utilizar el siguiente comando para desplegar el contrato a la red de prueba de *POP* utilizando una cuenta con suficientes fondos:

pop up contract –constructor new –args "false" –suri ACCOUNT-SEED –url wss://rpc1.paseo.popnetwork.xyz

Obteniendo un resultado como en Fig. 9. donde se suministra a un nodo remoto con el contrato para su despliegue en la red de pruebas, y, así mismo, se obtiene la dirección del contrato en la red una vez procesado, para poder interactuar con este.

Fig. 9. Despliegue del contrato en la red de pruebas de *POP*

7) **Iteracción:** Como se mencionó anteriormente, se puede interactuar con el contrato a través de la plataforma de *Polkadot-JS*, específicamente la interfaz web, agregando un contrato al sistema local de la red, como se observa en Fig. 10., y posteriormente interactuando con sus funciones, como se muestra en Fig. 11:

Fig. 10. Proceso de asociar un contrato al sistema local

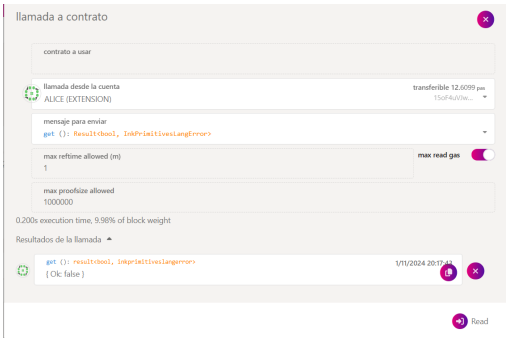

Fig. 11. Proceso de interactuar con el contrato

V. Conclusiones

Como conclusión, la red *Polkadot* posee muchas capacidades y utilidades, así como potencial de desarrollo, para las aplicaciones que se creen sobre y alrededor de la misma, ya que gracias a su interoperabilidad y escalabilidad abre las puertas hacia una nueva generación de *blockchains* que se comunican unas con otras y permiten la interacción entre diferentes plataformas con diferentes entornos sociales. Sin embargo, todavía se encuentra en un estado experimental donde es difícil encontrar información actualizada para la utilización adecuada de las funciones que otorga la red.

References

[1] F. J. Moreno-Arboleda, Rodríguez-Camacho, Johan S, y D. Giraldo-Muñoz, "Comparación de Dos Plataformas de Blockchain: Bitcoin y Hyperledger Fabric", Ingeniería y competitividad, vol. 24, no. 1, p. -, 2022, doi: https://doi.org/10.25100/iyc.24i1.11027.

[2] A. Preukschat and C. Kuchkovsky, Blockchain : la revolución industrial de Internet. Barcelona Booket, 2019.

[3] Polkadot, "What is Polkadot? A Polkadot for Beginners Guide and Intro to Blockchain", YouTube, May 21, 2020, https://www.youtube.com/watch?v=kw8eu2VadFA (accessed Oct. 18, 2024).

[4] G. Wood, "Polkadot: Vision for a Heterogeneous Multi-Chain Framework", Polkadot, 2016, https://polkadot.com/papers/Polkadot-whitepaper.pdf (accessed Oct. 18, 2024).

[5] H. Abbas, M. Caprolu, and R. Di Pietro, "Analysis of Polkadot: Architecture, Internals, and Contradictions", IEEE Xplore, Aug. 01, 2022. https://ieeexplore.ieee.org/document/9881859 (accessed Oct. 24, 2024)

[6] "Architecture", Polkadot, Oct. 3, 2024, https://wiki.polkadot.network/docs/learn-architecture (accessed Oct. 18, 2024).

[7] "Parachain Slot Auction", Polkadot, Oct. 25, 2024, https://wiki.polkadot.network/docs/learn/learn-auction (accessed Oct. 29, 2024).

[8] "Introduction to Polkadot OpenGov", Polkadot, Sep. 19, 2024, https://wiki.polkadot.network/docs/learn-polkadot-opengov (accessed Oct. 29, 2024).

[9] "Windows development environment", Substrate.io, 2019, https://docs.substrate.io/install/windows/ (accessed Nov. 01, 2024).

[10] "Linux", Onpop.io, Jul. 29, 2024, https://learn.onpop.io/cli/installing-pop-cli/linux (accessed Nov. 01, 2024).

[11] "Create a new contract", Onpop.io, Sep. 03, 2024, https://learn.onpop.io/contracts/guides/create-a-new-contract (accessed Nov. 01, 2024).

[12] "Build your contract", Onpop.io, Sep. 03, 2024, https://learn.onpop.io/contracts/guides/build-your-contract (accessed Nov. 01, 2024).

[13] "Deploy on Pop Testnet", Onpop.io, Sep. 03, 2024, https://learn.onpop.io/contracts/guides/deploy-on-pop-testnet (accessed Nov. 01, 2024).