Blockstack

# Whitepaper Técnico de Blockstack v 2.0

Mayo 2019

Por Blockstack PBC

Algunas partes de este whitepaper (libro blanco) han sido publicadas previamente en las siguientes conferencias y revistas científicas:

- M. Ali, J. Nelson, R. Shea and M. J. Freedman, “Blockstack: A Global Naming and Storage System Secured by Blockchains”, 2016 USENIX Annual Technical Conference, Denver, CO, June 2016.
- J.Nelson, M. Ali, R. Shea and M.J. Freedman,“Extending Existing Blockchains with Virtualchain”, Workshop on Distributed Cryptocurrencies and Consensus Ledgers, Chicago, IL, July 2016.
- M. Ali, J. Nelson, R. Shea and M. J. Freedman, “Bootstrapping Trust in Distributed Systems with Blockchains”, USENIX ;login: Issue: Vol. 41, No. 3, Pages 52-58, Fall 2016.

La versión 2.0 del whitepaper describe los grandes cambios que ha habido desde el whitepaper 1.0 publicado en 2017. Se puede encontrar la primera versión del whitepaper en:

- M. Ali, R. Shea, J. Nelson and M. J. Freedman, “Blockstack: A New Internet for De- centralized Applications”, Whitepaper Version 1.1, Oct 2017.

Algunos de los sistemas y conceptos descritos en este whitepaper también han sido debatidos en las siguientes disertaciones doctorales de los respectivos autores de este whitepaper:

- M. Ali, Trust-to-Trust Design of a New Internet, PhD dissertation, Princeton University, June 2017.
- J. Nelson, Wide-area Software-defined Storage, PhD dissertation, Princeton University, June 2018.

**DISCLAIMER (Advertencia Legal):** Los tokens de Blockstack, “Tokens Stacks” o “Stacks”, son un crypto activo que está siendo desarrollado por Blockstack Token LLC, empresa de responsabilidad limitada de Delaware, cuyo sitio web se puede encontrar en www.stackstokens.com.

Este whitepaper no constituye una oferta o venta de Stacks tokens o cualquier otro mecanismo para la compra de Stacks tokens. Cualquier oferta o venta de tokens Stacks o de cualquier instrumento relacionado, solo ocurrirá y se basará en aquellos documentos definitivos relacionados con la oferta de tokens Stacks.

Esta comunicación puede ser considerada como material “experimental” bajo la Regulación A bajo la Ley de Valores del 1993 (Securities Act of 1993). No tenemos ninguna obligación de completar una oferta bajo lo establecido en la Regulación A. Podemos tomar la decisión de realizar una oferta de venta para algunas personas en concreto, y no para todas aquellas personas que muestren interés en invertir, además de que, dicha oferta, no tiene por qué estar sometida a lo establecido en la Regulación A. Solo podremos realizar una oferta de venta, una vez la Comisión Nacional del Mercado de Valores (Securitites and Exchange Comission, SEC, en inglés) haya “calificado” la declaración de la oferta que hemos presentado a la SEC. La declaración de la oferta presentada ofrece más información de la que exponemos ahora, y, aun así, esta puede variar en aspectos importantes. Debería leer el documento presentado a la SEC antes de realizar cualquier tipo de inversión.

No se está solicitando dinero u otro tipo de consideraciones, y en caso de enviarse por parte de un tercero, este, no sería aceptado. Ninguna oferta de venta para comprar estos valores (securities), puede ser aceptada, y, ninguna parte del precio de compra puede ser percibido hasta que el documento de la oferta enviado a la SEC no haya sido calificado por dicha entidad. Tal oferta de venta puede ser rechazada o revocada, sin ninguna obligación o compromiso de ningún tipo, en cualquier momento antes de la notificación de aceptación obtenida el día después de la calificación.

Cualquier indicación de interés no representa ningún tipo de obligación o compromiso de cualquier tipo.

Cualquier persona interesada en la oferta de venta de los tokens Stacks debería revisar nuestras advertencias legales, y la documentación pública enviada y relacionada con la oferta de venta, pudiéndose encontrar una copia de esta, en la página web www.sec.gov.

Blockstack no está registrada, autorizada, o supervisada como Broker de Bolsa o como un asesor de inversiones bajo la regulación de la Comisión Nacional de Valores (Securities and Exchange Comission, SEC, en inglés), o el regulador financiero de EEUU (Financial Industry Regulatory Authority, FINRA, en inglés), ni cualquier regulador financiero, además de que no dispone la licencia para proporcionar servicios o asesoramiento financiero.

# La Red Computacional Descentralizada de Blockstack.

Muneeb Ali Jude Nelson Aaron Blankstein
Ryan Shea Michael J. Freedman[1]

*https://blockstack.org*
Versión 2.0 del Whitepaper
30 de Mayo del 2019

**Resumen**

En este documento, presentamos la red computacional descentralizada Blockstack. Blockstack proporciona una alternativa full-stack en comparación con la computación tradicional en la nube para poder desarrollar aplicaciones seguras y privadas.Una de las diferencias clave entre las aplicaciones de internet tradicionales y las aplicaciones descentralizadas que habilita blockchain, reside en que la mayoría de la lógica de negocio, y el procesamiento de los datos, en lugar de alojarse en un servidor central contratado por los desarrolladores de la aplicación, este, se aloja en la el cliente.

En muchos de los sentidos, la transición hacia la computación descentralizada es similar a la transición de servidores centrales a los ordenadores de escritorio en los años 1980.

Blockstack sigue un principio de diseño end-to-end (proceso completo) para mantener el núcleo de la red lo más simple posible, al mismo tiempo que se empuja por llegar a los límites complejos, por ejemplo, los dispositivos de usuario y almacenamiento controlados por el usuario. En el núcleo de la de nuestra red, está la Stacks blockchain, que ha sido diseñada para (a) escalar aplicaciones descentralizadas, y para, (b) incentivar a los desarrolladores para que desarrollen aplicaciones de alta calidad en la red. La cadena de bloques Stacks, (Stacks blockchain) utiliza una “Tunable Proof of Work (Prueba de Trabajo Afinable)” de sistema de elecciones para arrancar (bootstrap) de manera segura la blockchain. Nuestro nuevo lenguaje de programación de smart contracts (contratos inteligentes), *Clarity*, permite una expresabilidad muy amplia onchain mientras optimiza la seguridad y predictibilidad de los smart contracts (contratos inteligentes), además de admitir un análisis estático para todas las transacciones.

Uno de los componentes clave de nuestra arquitectura se encuentra en la escalabilidad y eficiencia, del sistema de almacenamiento llamado Gaia, dado que el sistema, permite controlar los contenedores privados de datos de los usuarios. Los usuarios conectan sus contenedores privados de datos al software del cliente de Blockstack, de tal manera que las aplicaciones escriben directamente en ellos. Un ID universal, así como un sistema de autentificación, llamado Blockstack Auth, eliminan la necesidad de registrarse de manera individual para cada aplicación, eliminando así la necesidad de usar las identificaciones basadas en contraseñas, que son conocidas por ser mucho menos seguras que una autentificación criptográfica.

Nuestras herramientas de desarrollo y SDKs hacen que desarrollar una aplicación en Blockstack no sea más complicado que desarrollar una aplicación tradicional de internet, además de que los desarrolladores no tienen que preocuparse de utilizar servidores o bases de datos. Blockstack está ya desplegado en producción, con un total aproximado de 100 aplicaciones independientes que han sido desarrolladas

---

[1]Profesor de ciencias informáticas en la Universidad de Princenton y Advisor Técnico de Blockstack PBC

desde principios del 2019. La arquitectura de Blockstack ha evolucionado después de varios años gracias a la experiencia obtenido al tener la red en producción, y todos los comentarios recibidos por los desarrolladores de aplicaciones. Este documento es una revisión importante de nuestro whitepaper del 2017.

# 1. Introducción

El internet diseñado hace más de 40 años, ha crecido desde poco más que un proyecto de investigación, a algo que se relaciona con absolutamente casi todas las interacciones digitales en el mundo. Aunque el núcleo, los protocolos de bajo de nivel de internet, se hayan mantenido consistentes desde el 1990, la capa de aplicaciones de internet, y la infraestructura de servidores ha evolucionado profundamente para poder soportar el crecimiento masivo que ha tenido las aplicaciones de internet.

El modelo principal para desarrollar aplicaciones en internet es el modelo del cliente/servidor [1], que fue popularizado en los años 90s. Este modelo ha sido una bendición a corto plazo, pero ha tenido unas consecuencias negativas en el largo plazo. Permitió que la web despegara, pero ha terminado causando que los servicios web se conviertan muy dependientes de los servidores remotos. La computación en la nube es una evolución del modelo básico de cliente/servidor. A día de hoy, los proveedores de servicios en la nube almacenan datos privados del usuario, ejecutan la lógica de la aplicación, manejan las credenciales de acceso, etcétera.

En la última década, hemos empezado a experimentar las consecuencias negativas de la computación en la nube, lo que ha puesto en duda todo el modelo de desarrollo de software mientras se confié en el modelo de cliente/servidor. Las filtraciones masivas de datos [2], pérdida de privacidad de los usuarios [3], falta de portabilidad en los datos, y la desconfianza amplia en los gigantes tecnológicos [4] provienen del diseño centralizado del modelo cliente/servidor. Dado la creciente importancia de la informática en la sociedad humana, no podemos permitir que los modelos anticuados sean los que definan la manera en la que vivimos.

La próxima evolución de la computación en la nube se aprovechará de dispositivos cliente más potentes, computación edge, y conectividad global para reducir la dependencia sobre estas plataformas centralizadas. La evolución hacia una computación descentralizada ya ha comenzado, y creemos que es el cambio tecnológico más importante en la industria de la informática desde la llegada de los ordenadores de escritorio después de los servidores centrales. La computación descentralizada puede cambiar cómo se desarrolla y utiliza el software. Está otorga a los desarrolladores una nueva serie de herramientas para trabajar y cambiar la relación que los consumidores tienen con el software: este, existe para proteger a los usuarios y optimizar los beneficios del usuario antes que cualquier otra cosa.

Blockstack es un esfuerzo open-source para diseñar, desarrollar y hacer crecer a una red computacional descentralizada que ofrece una alternativa full stack en comparación con la tradicional computación en la nube. Blockstack está reimaginando la capa de aplicaciones del internet tradicional y proporciona a los usuarios una manera de poseer y tener el control de sus datos directamente [5]. Blockstack utiliza la capa de transporte del internet existente y los protocolos subyacentes de comunicación, eliminando puntos de centralización en la capa de aplicaciones. Seguimos el principio de diseño end-to-end [6, 7] para mantener el núcleo de la red lo más sencillo posible, mientras se empuja por la complejidad de los clientes. Para conseguir escalar las aplicaciones, minimizamos los cambios globales de estado y proporcionamos un sistema fiable de almacenamiento descentralizado, que ofrece un rendimiento similar a los servicios de almacenamiento en la nube. Además, nuestro enfoque full-stack ofrece opciones predeterminadas para todos los componentes stack

de desarrollo necesarios para las aplicaciones descentralizadas. Blockstack es modular y los desarrolladores pueden fácilmente customizar e integrar tecnologías alternativas.

Este documento es una revisión importante de nuestro whitepaper del 2017 e incorpora la evolución de nuestro diseño con lo aprendido gracias a el despliegue de versiones en producción, así como los comentarios recibidos por parte de los desarrolladores de aplicaciones. Algunas partes de nuestras publicaciones revisadas por pares (peer reviewed) del 2016 [8, 9, 10] también se encuentran obsoletas, y animamos a los lectores a dirigirse a este documento para tener la foto del último diseño de Blockstack. Este documento introduce el nuevo diseño de la Stacks blockchain, que está diseñada para escalar aplicaciones descentralizadas y proporcionar incentivos a los desarrolladores para que desarrollen aplicaciones de alta calidad (sección 2). Presentamos nuestro nuevo lenguaje de programación de smart contracts, *Clarity*, optimizado para la seguridad y predictibilidad de los smart contracts (sección 3). Definimos el diseño del sistema descentralizado de almacenamiento Gaia (Sección 4), el protocolo de autentificación (sección 5), las herramientas de desarrollo (sección 6) y subrayamos alguna de las maneras en las que actualmente se está usando por los desarrolladores de aplicaciones en Blockstack (sección 7).

## 1.1 Visión General de la computación descentralizada

Los sistemas descentralizados son un tipo particular de sistemas distribuidos dónde ninguna entidad tiene el control de la infraestructura subyacente, y dónde los nodos tienen incentivos económicos para participar en la red. El reciente interés por las redes descentralizadas comenzó con la publicación del whitepaper de Bitcoin [11]. Las blockchains y las criptomonedas tienen un papel fundamental en los sistemas descentralizados contemporáneos. Recomendamos a los lectores ver la referencia [12] para obtener información más detallada sobre las blockchains y las criptomonedas.

A día de hoy, hay muchos distintos tipos de sistemas descentralizados en producción. El objetivo principal de Bitcoin, la primera, y actualmente la red blockchain más voluminosa, es rastrear y solucionar la propiedad de la divisa digital Bitcoin. El objetivo de Ethereum [13] es un propósito más general: construir un “ordenador mundial” que permita ejecutar smart contracts y aplicaciones descentralizadas. Filecoin [14] es un proyecto que quiere construir una red descentralizada para el almacenamiento de archivos con el espacio de almacenamiento disponible en los nodos de la red. Por el contrario, Blockstack intenta realizar un enfoque full-stack para la computación descentralizada, centrándose en habilitar aplicaciones seguras y privadas dónde la capa de la blockchain maneja una lógica y estado mínimo.

## 1.2 Objetivos del Diseño

El diseño de blockstack está optimizado para tener las siguientes propiedades:

1. **Facilidad de Uso**. Las aplicaciones descentralizadas deberán ser tan fáciles de usar para los usuarios finales como las aplicaciones actuales de internet. Además, las aplicaciones descentralizadas deberán ser tan fáciles de desarrollar como lo es desarrollar a día de hoy aplicaciones en la nube.
2. **Escalabilidad**. Las aplicaciones descentralizadas deberán soportar y escalar para tener tantos usuarios como tiene internet a día de hoy, por ejemplo, pasar de cientos de millones de usuarios, a billones de usuarios. Para ello, la red (incluyendo la blockchain) debe escalar a un ritmo paralelo al crecimiento de usuarios y las aplicaciones que se ejecutan.
3. **Control del usuario**. Las aplicaciones que utilicen computación descentralizada deberán poner por defecto en control a los usuarios. En lugar

de confiar en los servidores centralizados que son operados por las aplicaciones, los usuarios deberían ser capaces de poder proporcionar sus recursos de poder computacional y almacenamiento.

Teniendo estos objetivos de diseño en mente, Blockstack ha tomado una serie de elecciones de diseño que lo diferencian de los enfoques contemporáneos de la computación descentralizada con la filosofía de diseño de las blockchains “pesadas” y “ordenador mundial” [13, 15, 16, 17].

**Lógica y estados mínimos en la capa blockchain:** Para conseguir tener *escalabilidad*, Blockstack minimiza la lógica de la aplicación y los datos en nuestra capa “ligera” de blockchain. Utilizar las operaciones en blockchain para lo lógica de la aplicación y almacenamiento es inherentemente más lento que los enfoques “off-chain”; la necesidad para sincronizar y validar los estados en una amplia selección de diferentes redes y dispositivos impone significantes límites sobre la velocidad del rendimiento de dichas operaciones. El factor limitante es el subyacente ancho de banda de la conectividad global y la disponibilidad de memoria/almacenamiento en los nodos comunes de la red, por ejemplo, las limitaciones físicas (vs cualquier limitación del protocolo).

**Cambio de estado localizado vs cambio de estado global**: La red Blockstack utiliza un enfoque full-stack para asegurarse que las aplicaciones desarrolladas en blockstack son *escalables*: las interacciones con las aplicaciones resultan en cambios de los estados locales vs los estados globales, que cambian cada vez que sea posible. Por eso, nuestro sistema de almacenamiento (Gaia, ver sección 4) y el protocolo de autentificación (ver sección 5) son componentes fundamentales para nuestra red. Permiten a las aplicaciones interactuar con los contenedores privados de datos del usuario y autentificarse sin necesidad de realizar ninguna transacción en la blockchain. La Stacks blockchain solo se utiliza para coordinar la transición de los cambios de estado globales de forma consistente (como puede ser, el registro global de un nombre de usuario único) y de una manera descentralizada.

**Almacenamiento en la nube con necesidad de confiar en un tercero vs almacenamiento de pares (P2P):** Las aplicaciones desarrolladas en Blockstack guardan los datos de los usuarios (utilizando los contenedores privados de datos de los usuarios), además, no necesitan almacenar ningún tipo de dato del usuario o credenciales de acceso en la parte del servidor. Este enfoque no solo ofrece a los usuarios el control de sus datos, sino que también reduce aspectos complejos para los desarrolladores: los desarrolladores ya no necesitan usar servidores, ni bases de datos, por lo que no tienen que pagar las facturas de mantener la infraestructura en la nube en nombre de sus usuarios. Además, evitamos los problemas de confianza y rendimiento inherentes con las aplicaciones de almacenamiento de par a par (P2P) [18] y así, poder reutilizar los proveedores existentes de almacenamiento en la nube utilizando un sistema de archivos descentralizado – la capa de blockchain solo guarda punteros que apuntan a los contenedores privados de datos de los usuarios.

**Full-stack SDK para desarrolladores**: Blockstack hace un enfoque “full-stack” y proporciona opciones por defecto para todas las capas necesarias para el desarrollo de una aplicación descentralizada. Los SDKs de desarrollo abstraen la complejidad de trabajar con la blockchain y otro tipo de tecnologías; los desarrolladores pueden desarrollar sus aplicaciones con facilidad, gracias a la interfaz del SDK (sección 6). Varias de las capas de la pila de desarrollo son modulares y pueden ser utilizadas con otro tipo de tecnologías en caso de que fuera necesario.

Además de estas diferencias sobre los enfoques computacionales descentralizados contemporáneos, nuestro lenguaje de smart contracts tiene unas decisiones de diseño únicas que buscan optimizar la seguridad y predictibilidad de los smart contracts (ver sección 3 para más detalle).

## 1.3 Un nuevo modelo para las Aplicaciones

Blockstack proporciona a los desarrolladores un nuevo modelo para desarrollar aplicaciones, asegurándose que, gracias a este, las aplicaciones sean descentralizadas y pongan al usuario en control por defecto.

1. **Bases de Datos sin opacidad:** En el modelo cliente/servidor, las bases de datos son el núcleo de cualquier aplicación dado que el servidor necesita almacenar y consultar gran cantidad de datos de los usuarios. En la computación descentralizada, los desarrolladores no necesitan preocuparse del mantenimiento y la seguridad de las bases de datos, ya que no almacenan ningún dato del usuario. Las bases de datos, si se utilizan, son el equivalente a los “buscadores indexados” del internet antiguo – servicios que indexan datos públicos. Cualquiera puede crear estos indexes utilizando la capa de datos (descentralizados) subyacente.
2. **Sin servidores:** En el modelo cliente/servidor, las aplicaciones escalan al ir añadiendo más servidores y computación para que los usuarios puedan ejecutar sus programas. En la computación descentralizada, las aplicaciones corren en el lado del cliente, y cada usuario nuevo aporta su poder computacional y su capacidad de almacenamiento a la red (en vez de depender de los desarrolladores de aplicaciones). Los desarrolladores solo necesitan suministrar una infraestructura mínima para alojar la aplicación del código, ya que cada usuario aporta almacenamiento y recursos computacionales que necesiten utilizar para su aplicación.
3. **Smart contracts (Contratos Inteligentes):** En el modelo cliente/servidor los cambios globales de estado son coordinados por un cliente central que funciona como la única fuente de verdad en toda la red. En la computación descentralizada, los cambios de estado ocurren a través de smart contracts (contratos inteligentes) ejecutados en una blockchain pública.
4. **Autentificación descentralizada:** En el internet tradicional, los usuarios se autentifican utilizando algún tipo de proceso de autentificación fiable. Si una aplicación mantiene una base de datos de usuarios, su aplicación autentifica los usuarios a través de una contraseña, y algunas veces un segundo factor de autentificación. Si una aplicación se basa en el servicio de autentificación de identidad de un tercero, como puede ser Google o Facebook, utilizará el protocolo OAuth [19] para obtener una aserción del servicio de identidad. Por supuesto, todos estos enfoques eliminan el control de los usuarios en la totalidad del proceso. En la computación descentralizada, la autentificación es realizada a través del cliente del usuario al firmar criptográficamente una declaración que demuestra el control sobre un nombre de usuario particular que ha sido registrado previamente en la blockchain. Cualquier aplicación puede verificar estas pruebas de manera independiente.
5. **Tokens Nativos:** En el internet tradicional, las actividades relacionadas con pagos son comúnmente ejecutadas mediante el uso de servicios de terceros, como, por ejemplo, las tarjetas de crédito. Los tokens digitales son un activo nativo de las plataformas de computación descentralizada como por ejemplo Blockstack y Ethereum. Los usuarios tienen propiedad directa sobre los tokens y pueden utilizarlos directamente para registrar activos digitales y smart contracts, además de poder ejecutarlos. El uso de tokens nativos puede programarse a través de smart contracts que son utilizados para desarrollar servicios de suscripción y automatizar otro tipo de funcionalidades de la aplicación. Este tipo de tokens programables no se encontraban disponibles para los desarrolladores en las aplicaciones tradicionales de internet.

## 1.4 Capas de la computación descentralizada

La red de computación descentralizada Blockstack, lógicamente, existe dentro de la capa de aplicaciones del diseño tradicional de internet. Sin embargo, la red Blockstack de por sí sola, se compone de múltiples sistemas, que en conjunto proporcionan los componentes necesarios para implementar aplicaciones descentralizadas.

1. **Stacks Blockchain:** La base de la red Blockstack es la Stacks blockchain, que permite a los usuarios registrar y controlar activos digitales como nombres de usuario universales, además de poder registrar y ejecutar smart contracts. Los activos digitales como los nombres universales, a su vez, permiten a los usuarios controlar la capacidad de almacenamiento y más - los usuarios pueden vincular sus credenciales de acceso a los contenedores privados de datos a través de sus nombres universales.
2. **Gaia:** El sistema de almacenamiento Gaia es un sistema de almacenamiento controlado por el usuario que permite a las aplicaciones interactuar con los contenedores privados de datos. Los usuarios pueden alojar estos contenedores privados de datos, de manera encriptada en un proveedor de almacenamiento en la nube, disco local o en un almacenamiento remoto. Es importante destacar que el *usuario* tiene el control para poder escoger el proveedor subyacente que desee. En Gaia los datos están encriptados y firmados por la parte del cliente a través del uso de claves criptográficas. Los usuarios, para descubrir nuevos contenedores privados de datos solo tienen que buscar la información dentro de la Stacks Blockchain.
3. **Autentificación de Blockstack:** El protocolo de autentificación de Blockstack es un protocolo para la autentificación en aplicaciones de manera descentralizada. El protocolo permite a los usuarios autentificarse utilizando identidades sobre las que tienen control y proporciona la información sobre qué localización de Gaia deberá usarse para almacenar los datos de la aplicación del usuario.
4. **Librerías Blockstack y SDKs:** En la parte superior de la pila de software se encuentran las librerías para desarrolladores y los SDKs, a través de los cuales, los desarrolladores y los usuarios interactúan con los diversos componentes de la red Blockstack. Por ejemplo, el software del cliente de Blockstack permite a los usuarios registrar y manejar sus propias identidades. Las librerías de desarrollo de Blockstack hacen que desarrollar aplicaciones en Blockstack sea *relativamente igual de fácil* que desarrollar aplicaciones web tradicionales.

## 2. Stacks Blockchain

La capa fundamental de la red Blockstack es la Stacks blockchain. La Stacks blockchain proporciona un consenso global y una capa de coordinación para la red, además de implementar un token nativo dentro de la red Blockstack que se llama, *Stacks*. Los tokens Stacks son consumidos como “gasolina” cuando los usuarios registran activos digitales como pueden ser los nombres universales, licencias de software, punteros a los contenedores de almacenamiento, etc. También, se utilizan para pagar a los mineros por registrar y ejecutar los smart contracts.

En esta sección, presentamos a un alto nivel el diseño de la Stacks blockchain. Para más detalles sobre la implementación y evolución de estos diseños, recomendamos leer el Stacks Improvement Proposals (SIPs, Proposiciones de mejora Stacks, en castellano) para los distintos componentes[2]. Prevemos actualizar este

[2] Disponible en https://github.com/blockstack/blockstack-core/tree/develop/sip

documento a medida que más SIPs sean aceptados a través del proceso de mejoras de Stacks. La Stacks blockchain incorpora este conjunto de decisiones de diseño:

1. Un mecanismo afinable de Proof of Work (Prueba de Trabajo) para realizar la elección del líder.
2. Un algoritmo de minado Proof of Burn (Prueba de Quemado) para reutilizar el hashpower (poder computacional) de las blockchains actuales.
3. Una innovadora red de pares (Atlas) que utiliza gráficos aleatorios para la conectividad de pares y reduce la cantidad de datos necesarios para poder alcanzar un consenso.
4. Un lenguaje de programación de smart contracts (*Clarity*) que no es Turing completo, ni *interpretado*.

**Versiones Blockchain:** El estado actual de la Stacks blockchain es la “Versión 1”, que es una implementación inicial para desplegar funcionalidades básicas. La v1 de la Stacks Blockchain utiliza la red de Bitcoin para implementar su algoritmo de consenso y tiene la capacidad de hacer operaciones con el token Stack como pueden ser transferencias. La v1 de la blockchain de Stacks implementa smart contracts para casos de uso como el Registro de Nombres/Dominios Blockstack [8]. Para más detalles sobre la implementación y el funcionamiento de la versión 1, ver la implementación en Github [20]. El resto de la sección analiza el diseño de la “Versión 2” de la Stacks blockchain. La v2 de la Stacks blockchain implementa una funcionalidad completa tanto de nuestro nuevo algoritmo de consenso como de nuestro lenguaje de smart contracts, siendo una actualización importante sobre la versión 1.

## 2.1 Elección del líder

La blockchain de primera generación de Blockstack operaba de manera lógica sobre la capa 1 (L1) y cada transacción estaba relacionada 1 a 1 con una transacción en la L1, en este caso, Bitcoin. La razón de hacerlo así fue, asegurar que la dificultad de reorganizar la blockchain de Blockstack tiene que ser igual que la de reorganizar la blockchain de Bitcoin – esto fue gracias a una de las lecciones aprendidas sobre los problemas de seguridad de las redes blockchains más pequeñas como Namecoin [8].

La Stacks Blockchain utiliza un mecanismo afinable de pruebas para el proceso de la elección de líder. El mecanismo afinable de pruebas es un sistema de elección de un líder que puede tomar entrada de múltiples mecanismos y adaptar la importancia relativa de cada entrada. Por ejemplo, con las pruebas afinables podemos combinar un algoritmo nativo de Proof of Work, con la funcionalidad añadida de poder reutilizar el hash-power (poder computacional), de otra blockchain más establecida. Nuestro objetivo con las pruebas afinables, es poder empezar de una forma segura una nueva blockchain que permita migrar lentamente hacia un mecanismo nativo de Proof of Work. Las pruebas afinables actuales tienen dos partes (a) un PoW nativo y (b) un Proof of Burn (Prueba de Quemado) de otra criptomoneda.

Inicialmente, la Proof of Burn (Prueba de Quemado) para la parte del minado tiene más importancia. Con la prueba de quemado, los mineros queman criptomonedas para indicar el interés que tienen a la hora de participar en el proceso de minado. Para poder ser elegido líder, un candidato deberá quemar la criptomoneda subyacente (por ejemplo, Bitcoin) y comprometerse a realizar una serie de transacciones en el bloque de posibles líderes. Este compromiso también sirve para la selección de bifurcación del líder: el hash del consenso del bloque debe incluir la cabecera del hash del bloque anterior. En el caso de varias bifurcaciones legítimas en competición, los líderes que eligen “minar” en las blockchains perdedoras no obtiene la recompensa de bloques, las tasas de las transacciones o recuperar su criptomoneda quemada.

El mecanismo de la Proof of Burn (Prueba de Quemado) se utiliza en la Stacks blockchain y permite:

**Alto rendimiento de validación**. El número de transacciones Stacks procesadas es desacoplado velocidad de procesamiento de transacciones de la cadena subyacente para “quemar tokens” (por ejemplo, Bitcoin). Al utilizar las elecciones, la Proof of Burn (Prueba de Quemado) permite realizar *bloques enteros* de transacciones para confirmarse, por cada nuevo bloque en la cadena de quemado subyacente.

**Inclusión de bloques de baja latencia:** al permitir una elección de líder única, nuestro algoritmo de consenso Proof of Burn (Prueba de Quemado) permite al líder actual incluir *inmediatamente*, nuevas transacciones del memepool en su bloque de stacks. Este modelo de *streaming de bloques* permite a los usuarios saber si el bloque incluye una transacción en cuestión de segundos.

**Set de líderes abiertos:** La elección de la Proof of Burn (prueba de quemado) permite a *cualquiera* ser un líder. Este mecanismo asegura que la Stacks blockchain sea una blockchain pública (al contrario de las blockchain privadas que se basan en un conjunto fijo de líderes, o los sistemas con prueba de participación delegada ,Delegated Proof of Stake, que se comportan y funcionan como las blockchains privadas). Además, al realizar una elección de un único líder, nuestro algoritmo de consenso garantiza que los potenciales líderes no necesitan coordinarse entre ellos.

**Participación sin hardware de minado:** El trabajo requerido para participar como líder solo requiere “quemar” una criptomoneda, al contrario del tradicional esquema de minado con Proof of Work (Prueba de Trabajo). Debido a esto, no se necesita hardware de minado para participar como líder. Cualquiera puede adquirir la criptomoneda y quemarla para participar en el minado, aunque solo puedan permitirse una cantidad mínima.

**Cooperativas de minado justas:** La Stacks blockchain soporta de manera nativa las cooperativas de minado justas. Cualquiera que participe en la red puede quemar la criptomoneda para mostrar el apoyo en la elección de un líder concreto. Los usuarios que cometen tales “quemas para apoyar a otros usuarios (líderes)” comparten en la misma proporción que el líder los beneficios que se deriven del bloque de Stacks.

**Capacidad de computación por error:** El diseño asegura que en el evento de que la blockchain de quemado se convierta en inusable, o inadecuada para el minado de la Stacks blockchain, esta puede utilizar otra blockchain diferente para el “quemado”.

Puede encontrar más detalles sobre nuestro componente de Proof of Burn (Prueba de Quemado) en la referencia [21]. Es posible que el componente de Proof of Work (Prueba de Trabajo) no sea necesario una vez ya haya suficiente poder computacional en la Stacks blockchain.

## 2.2 Pruebas Afinables

La Stacks blockchain incluye un componente nativo de Proof of Work (Prueba de Trabajo) en el algoritmo de consenso además de la parte de Proof of Burn (Prueba de Quemado). Esta combinación permite compartir la responsabilidad de la seguridad de la cadena con el sistema de elección de Proof of Burn descrito en el SIP-001. Esta combinación de un Proof of Work nativo con un Proof of Burn se llama Proof of Work Afinable, y es la implementación actual de las pruebas afinables de nuestro sistema que permite una introducción responsable de un minado PoW nativo,

dónde la Proof of Burn garantiza la estabilidad de la cadena, aún y cuando haya poco interés en el Proof of Work. La parte afinable abre un camino hacia la posibilidad de una migración en caso de que se deteriore la blockchain de quemado subyacente. Las pruebas afinables nos permiten investigar sobre otros tipos de mecanismos de Prueba de Trabajo (Proof of Work) o Prueba de Participación (Proof of Stake) para poder ir introduciendolos lentamente de una manera afinable.

El componente actual de Proof of Work en la elección de un líder funciona al permitir a los candidatos a líder poder incluir de manera opcional, un nonce del Proof of Work en su transacción de quemado. La cantidad de trabajo necesaria para producir un nonce (por ejemplo, alguna función para obtener el número de ceros iniciales en el hash resultante) serán contada en la “cantidad quemada” del candidato. Inicialmente, habrá un límite del 5% en la Proof of Work nativa (en relación con la cantidad de quemado proporcionada). El componente nativo del Proof of Work todavía se encuentra en fase de diseño y desarrollo. En la medida que se obtengan nuevos detalles del desarrollo, esta sección (como la SIP correspondiente) será actualizada.

## 2.3 Red de Pares (P2P) Atlas

La red de pares (P2P) Atlas es una red de pares de contenido direccionable que implementa un protocolo gossip (“cotilleo”) dónde cada par mantiene un registro de los pares existentes en la red, y cada par intenta almacenar una copia completa de todos los datos de la red. La capacidad de la red está limitada por la Stacks blockchain: cualquier registro nuevo en el set de datos debe estar asociado con una transacción en la Stacks blockchain. La red de pares Atlas funciona como un subsistema de la Stacks blockchain. Está diseñada para ser una red de pares desestructurada para evitar problemas con los nodos a la hora de conectarse o desconectarse de la red [18, 22]. Además, dado que todos los nodos mantienen una copia de todos los datos, y hay un indexador de datos disponible en la Stacks blockchain, los nuevos nodos de Atlas pueden sincronizar rápidamente los datos que necesitan almacenar, ya que de antemano saben los datos de los otros pares que deben de estar almacenando (generalmente, esta funcionalidad no está disponible para los nodos en las redes de par a par, P2P).

La red Atlas funcionan como un subsistema de la Stacks blockchain a modo de “extensión del almacenamiento”. Nuestro enfoque en el diseño es confiar *lo menos posible* en la interacción directa con la propia Stacks blockchain y almacenar en esta el menor número de datos posibles. Para muchas de las aplicaciones dentro de Blockstack, como el smart contract con el Sistema de Registro de Nombre de Blockstack (BNS), es esencial poder tener un mecanismo para almacenar datos de manera *inmutable* y con una *marca de tiempo (Timestamped)*. En BNS utilizamos esto para poder asociar los nombres de usuario con la información de enrutamiento que se utiliza para descubrir el nombre del usuario y los datos de aplicación. En muchas blockchains, se está almacenando este tipo de datos, *directamente* en la blockchain. Sin embargo, hemos escogido almacenar solo los hashes en la blockchain (dónde el espacio para almacenar es caro) y hemos implementado una red de pares separada para intercambiar los datos que se corresponden a esos hashes.

## 2.4 Uso de los tokens Stacks

Los tokens nativos Stacks están implementados dentro de la Stacks blockchain y permiten realizar varias operaciones fundamentales dentro de la red:

1. **Gasolina para registrar activos digitales.** Los tokens Stacks son utilizados para registrar diferentes tipos de activos digitales como nombres de usuarios, nombres de dominios, licencias de software, podcasts y mucho más.

2. **Gasolina para registrar y ejecutar smart contracts**. Para ejecutar smart contracts es necesario tener “gasolina” para financiar el coste de verificar la exactitud del Smart contract y el coste de su ejecución. Los tokens Stacks también pueden ser utilizados para cubrir los costes de almacenamiento del smart contract en la Stacks blockchain.
3. **Tasas de las transacciones**. Los tokens Stacks son utilizados para pagar las tasas derivadas de incluir una transacción dentro de la Stacks Blockchain.
4. **Anchored app chains (Aplicaciones de la cadena ancladas**). Para que las apps se conviertan populares en Blockstack, nuestra blockchain tiene una “rampa de escalabilidad” dónde una app puede inicializar su blockchain encima de la Stacks blockchain. Dicha “aplicación cadena” quema los tokens Stacks para su minado y progresión.

La lista descrita arriba no es exhaustiva – a medida que la red Blockstack madura, esperamos que los participantes de la red descubran e inventen diferentes usos para los tokens Stacks. Actualmente nos encontramos investigando sobre mecanismos para “App Staking” dónde los tenedores del token pueden potencialmente participar en nuestro programa de incentivos para desarrolladores denominado “App Mining (Minado de Apps)” [23].

## 3. Lenguaje de Programación de Smart Contracts, Clarity

La Stacks blockchain soporta el lanzamiento y ejecución de smart contracts para programar el control de activos digitales. Este nuevo lenguaje de programación de smart contracts, llamado *Clarity*, optimiza la seguridad y predictibilidad, que informa de algunos de los objetivos de diseño clave que hace que se diferencie de los sistemas de programación de smart contracts previos.

1. El lenguaje deberá permitir de manera simple y rápida un análisis estático y preciso para el tiempo de ejecución y requerimientos de espacio. Para poder soportarlo, el lenguaje no es Turing completo en cuanto a la ejecución de una sola transacción. Sin embargo, si cogemos el historial de transacciones completo, el lenguaje si es Turing completo.
2. Los smart contracts deberán ser *interpretados* por nuestra VM (Máquina Virtual), sin compilarlos. El código escrito por los desarrolladores deberá ser desplegado directamente sobre la blockchain.

Para conseguir las dos propiedades mencionadas anteriormente, hemos creado una nueva variante LISP, diseñada especialmente para la programación de smart contracts. Para obtener una descripción más detallada del diseño de Clarity, por favor lea SIP-002 [24].

### 3.0.1 Visión General del lenguaje

Clarity es similar a otras variantes-LISP (ejemplo, Scheme), pero con las siguientes diferencias:

1. La recursión está prohibida y no hay función *lambda*.
2. Los loops solo se podrán realizar a través de *map, filter o fold.*
3. Los únicos tipos atómicos son los boleanos, los enteros, buffers de longitud fijada, y los principales.
4. Hay un soporte adicional para las listas de tipos atómicos, sin embargo, la única variable de longitud de las listas, en Clarity está en forma de inputs de las funciones (por ejemplo, no hay soporte para la lista de operaciones como append o join). También soportamos tuplas de nombre y tipo.
5. Las variables solo pueden ser creadas a través de uniones *let* y no hay soporte para mutar funciones de tipo *set*.

6. La definición de constantes y funciones está permitida para poder simplificar el código al utilizar las declaraciones de tipo *define*. Sin embargo, estas, son puramente sintácticas. Si una definición no puede estar en línea, el contrato será rechazado y determinado ilegal. Estas definiciones también son privadas, de tal manera que las funciones definidas solo pueden ser llamadas por otras funciones definidas dentro del propio smart contract.
7. Las funciones especificadas a través de las declaraciones de tipo *define-public* son funciones públicas. Los argumentos de estas funciones deben especificar el tipo de argumento que son.

Los smart contracts son capaces de:

1. Llamar a las funciones públicas de otros smart contracts. Estos smart contracts son identificados a través de un hash y deberán existir en el momento que se publica la ejecución de la llamada a la función del smart contract. Esto, combinado con la ilegalidad de la recursión, hace que sean capaces de prevenir la reentrada de funciones, que es un vector de ataque común dentro de las plataformas de smart contracts actuales.
2. Posesión y control de activos digitales. Los smart contracts son los directores de primer orden, como las claves públicas o las direcciones multifirma.

Cada smart contract tiene su propio espacio para el almacenamiento de datos. Dentro de este espacio, los datos de almacenamiento son almacenados en mapas. Este almacenamiento relaciona un tipo de tuplas con otro tipo de tuplas (casi como un almacenamiento de tipo clave-valor). Al contrario de la estructura de datos en tabla, un mapa solo asocia una clave por cada valor exacto.

Cualquier smart contract puede extraer los datos de los mapas de otro smart contract. Sin embargo, solo los propios smart contracts pueden actualizar directamente los datos sobre sus mapas.

Decidimos utilizar mapas de datos en vez de utilizar otro tipo de estructuras de datos por dos razones:

1. La simplicidad de los mapas de datos permite tanto una implementación simple dentro de una VM (Máquina Virtual), y un razonamiento más sencillo sobre las funciones. Al inspeccionar la definición de una función concreta, está claro que los mapas serán modificados e incluso dentro de estos mapas, habrá una serie de claves afectadas por una invocación concreta.
2. La interfaz de los mapas de datos asegura devolver los tipos de operaciones del mapa en una longitud fija, ya que esto es un requisito para el análisis estático sobre, el tiempo de ejecución de un smart contract, el coste y otras propiedades.

## 3.0.2. Incompletitud de Turing y análisis estático

Crear un lenguaje que no fuera Turing completo era esencialmente una de las consideraciones de diseño. Esto, otorga multitud de beneficios a la hora de programar en el hostil entorno de las blockchains.

1. La incompletitud de Turing permite hacer un análisis estático y determinar el coste de ejecución de una transacción concreta. Esto permite a la red saber, *a priori*, de manera exacta cuánto costará la tasa que se cobra por realizar una transacción. Esto mejora el comportamiento del cliente, porque el coste de emitir una transacción es conocido por el cliente, y puede ser fácilmente transmitida al usuario.

2. La incompletitud de Turing permite hacer un análisis estático para determinar rápidamente propiedades importantes como, por ejemplo, qué contratos pueden invocar una sola transacción. Esto mejora la experiencia de usuario, por que el cliente puede avisar al usuario sobre cualquier posible efecto secundario derivado de la ejecución de una transacción.
3. Tener un análisis estático mejorado y preciso permite a los programadores analizar de manera fiable la existencia de posibles fallos o errores en los smart contracts antes de que se vayan a desplegar.

Fundamentalmente, creemos que es un error considerar los lenguajes de programación de smart contracts como cualquier otro lenguaje de programación. Las propiedades de las blockchains hacen que los detalles en los smart contracts sean muy importantes, y creemos que renunciar a la condición de poder tener una programación más sencilla, a cambio de un incremento en la comprensión humana y de máquina sobre el comportamiento de los smart contracts, es una buena renuncia. Los casos de usos existentes actualmente sobre distintos smart contracts lo confirman.  – la historia de los smart contracts Turing completos es básicamente la historia de los errores en los smart contracts.

En Clarity, al ejecutar el análisis estático antes de desplegar el smart contract, puede proporcionar información como:

1. El coste de emitir la transacción concreta en función del tamaño del input.
2. El número de transacciones que harán falta para modificar una tabla en particular.

En un futuro, se podrá soportar características de análisis más avanzadas, como, por ejemplo, la habilidad de automáticamente comprobar “pruebas” dentro del código de programación de los smart contracts.

## 3.0.3. Lenguajes Interpretados vs Compilados

Una segunda decisión clave del diseño de Clarity fue optar por un lenguaje interpretado, en lugar de un lenguaje compilado (por ejemplo, compilar a WASM). Nuestra decisión de diseño de no utilizar un compilador es fundamentalmente para diferenciarnos de los enfoques contemporáneos. La razón principal sobre esta decisión de diseño es la habilidad para razonar sobre los errores de implementación.

Los errores en la implementación son un hecho de la vida, y aun cuando se utilizan los mejores estándares de programación, son inevitables. Esto aplica a los errores de los smart contracts (blockchains) tanto como aplica a cualquier otro tipo de código. Los errores de los smart contracts son más difíciles de manejar. Varias comunidades blockchain se adhieren a la filosofía de “el código es la ley” y las reglas establecidas en la blockchain son la última fuente de la verdad. Los desarrolladores de smart contracts expresan su intención a través del código fuente, sin embargo, el proceso de compilación es el que traduce sus intenciones en las reglas finales. Esto puede llevar a situaciones en las que las reglas reales difieren de las intenciones del desarrollador por culpa de errores en el compilador. Esto, suele terminar en situaciones desagradables dónde la gente discute sobre si la intención del desarrollador es lo más importante o sobre si las reglas de la blockchain son lo más importante. En la Stacks blockchain evitamos esta situación al eliminar el paso del compilador, y al comprometer directamente la intención del desarrollador con la blockchain de tal manera que la intención de los desarrolladores nunca difiera de las reglas que han programado.

Consideremos el caso de un error en la implementación de nuestro lenguaje de programación de smart contracts (por ejemplo, la VM, Máquina Virtual). Si el lenguaje de programación de smart contracts utiliza un intérprete, es relativamente fácil solucionar el error. Todo el código mundial de los contratos está en la misma blockchain, y se puede arreglar el error corrigiendo en el intérprete, y reiniciando los nodos de blockchain desde el génesis (reaplicando todas las transacciones).

Pero, si el lenguaje de programación de smart contracts se compila, y el error reside en el compilador y no en la VM (Máquina Virtual), entonces, el remedio es mucho menos obvio, por lo que es probable que el error sea mucho más conflictivo. Esto se debe a que, un error en el compilador puede causar que el código generado (que es lo que finalmente se transmite en la blockchain) se comporte de manera diferente en comparación con la intención del código programado por el desarrollador. Bajo la filosofía de “el código es la ley”, latente en las comunidades crypto, este tipo de situaciones suelen ser más complicadas de lidiar. El código desarrollado por el programador es correcto, pero las transacciones que se generan en la propia blockchain son erróneas. No es realista tener que recoger todo el código fuente hecho por cualquier desarrollador para recompilarlo, especialmente cuando no puedes verificar que el código fuente no ha sido modificado. Sospechamos que, en la práctica, en este tipo de situaciones, el código publicado en la blockchain es la última fuente de la verdad en muchos casos. En este caso, los desarrolladores deberían razonar y verificar ese código y no el código fuente. Creemos que utilizar un lenguaje interpretado de alto nivel tiene una suma importancia para poder asegurar la ejecución de un smart contract.

## 4. Gaia: Almacenamiento controlado por el usuario

Blockstack entrega al usuario el control de sus datos al utilizar el sistema de almacenamiento Gaia. Este sistema de almacenamiento que es controlado por el usuario, permite a las aplicaciones interactuar con los contenedores privados de datos. Los usuarios pueden almacenar los contenedores privados de datos en un proveedor en la nube o en otras opciones de almacenamiento, como, por ejemplo, un almacenamiento privado (en local). Es importante destacar que el usuario es quien controla que proveedor utilizar. En Gaia, los datos están encriptados y firmados con unas claves criptográficas en posesión del usuario. Como es lógico, Gaia funciona como un sistema de archivos de área amplia que se puede instalar para almacenar archivos.

Con el sistema de almacenamiento Gaia, los usuarios designan las localizaciones de almacenamiento que recogen los datos. La Stacks blockchain (y el subsistema Atlas) solo almacenan los “punteros” a las localizaciones en Gaia.

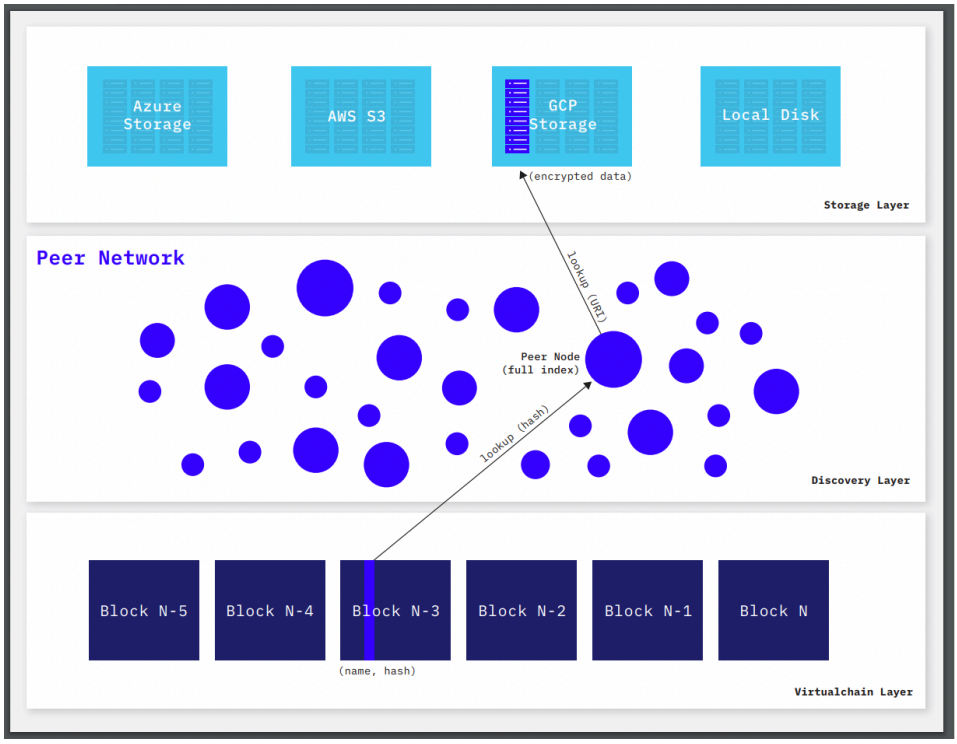


Figura 1: Visión general sobre Gaia y los pasos que hay que realizar en una búsqueda de datos.

Cuando los usuarios inician sesión en las aplicaciones o servicios que utilizan el protocolo 5 de autentificación de Blockstack, pasan la localización a la aplicación; y con esta información, la aplicación ya sabe cómo comunicarse con el sistema Gaia y el contenedor específico de datos privado que ha indicado el usuario.

La filosofía de diseño de Gaia es la de reutilizar los proveedores de almacenamiento en la nube y la infraestructura existente, de tal manera que el usuario final no necesita fiarse de los proveedores de servicios subyacentes. Tratamos a los proveedores de almacenamiento en la nube (como por ejemplo Amazon S3, Google Cloud Storage o incluso un disco local) como “controladores de dispositivos tontos” y almacenamos en ellos datos encriptados y/o firmados. Los proveedores de almacenamiento no tienen ninguna visibilidad sobre los datos del usuario; solo pueden ver conjuntos de datos encriptados. Además, dado que las claves públicas asociadas o hashes de datos, son descubribles a través de la Stacks blockchain, los proveedores de almacenamiento en la nube no tienen la posibilidad de alterar los datos de los usuarios.

Escribir datos en el servidor de Gaia involucra *POST*-ing en la localización adecuada del servidor. Estos *POST*s son validados por el servidor al verificar que cualquier solicitud de escritura tiene un token de *autentificación* firmado. Este token es firmado a través de la clave privada que controla el “sector de almacenamiento” particular sobre el cual se está escribiendo. Con el fin de poder proporcionar compartimentos separados para cada aplicación que pueda utilizar un usuario, el usuario deriva una serie de claves privadas separadas para cada una de las aplicaciones. Cada una de estas claves privadas otorga acceso a un “sector de almacenamiento” específico en el servidor Gaia.

En Gaia, la información de ruta verificada por la blockchain contiene una URL que apunta a un objeto JSON firmado (firmado por la clave que es dueña de ese nombre de usuario concreto). Este objeto JSON firmado contiene una URL que apunta

al contenedor privado de datos del usuario en Gaia. Una vez que la aplicación conoce la localización del contenedor privado de datos del usuario, simplemente solicita un archivo de esa localización a través de una petición normal de HTTP. Para poder buscar un archivo creado por un usuario *diferente*, las aplicaciones pueden realizar peticiones de búsqueda de manera secuencial en la parte del cliente. Mientras que esto impone una penalización de latencia sobre las búsquedas iniciales, gran parte de esta información de enrutamiento se almacenará en el cache de los navegadores locales (o aplicaciones nativas) de tal manera que las próximas búsquedas serán igual de rápidas que las búsquedas tradicionales en internet.

La figura 1 muestra una visión general de Gaia. Búsqueda de datos sobre un nombre, como, por ejemplo, *werner.id*, funciona de la siguiente manera:

1. Búsqueda del *nombre* en la Stacks blockchain para obtener el par (*nombre, y hash*).
2. Búsqueda del *hash (nombre)* en la red de pares Atlas de Blockstack para obtener el nombre del archivo de enrutamiento.
3. Obtener la URL de Gaia de los usuarios en el archivo de enrutamiento y búsqueda de la URL para conectar con el almacenamiento del back-end.
4. Los datos GET/PUT del servicio Gaia designados (desencriptados si hiciera falta y si el lector tiene derechos de acceso) y verificación de la respectiva firma o hash.

Los pasos 1 y 2 son realizados con una simple llamada al *núcleo de Blockstack* en los endpoints */v1/names/<name>.* Estas lecturas y escrituras iterativas son manejadas automáticamente en nuestras librerías de desarrollo.

**Rendimiento:** el objetivo de nuestra arquitectura es ofrecer un rendimiento comparable con las aplicaciones tradicionales de internet desarrolladas sobre los actuales proveedores de almacenamiento en la nube. Introducimos un nivel de seguridad significativo, además de unos beneficios derivados de la tolerancia a fallos (fault-tolerant) al eliminar puntos centrales de fallo o control– merece la pena pagar una pequeña cantidad para obtener un mayor rendimiento de lectura/escritura (read/write) siempre y cuando dicha cantidad no sea, ni significativa, ni perceptible para el usuario medio. Hemos evaluado el rendimiento de la escritura y lectura de Gaia para demostrar que escribe y lee archivos en un ratio competitivo en comparación con los distintos servicios de almacenamiento. Gaia añade un espacio de almacenamiento constante e insignificante a cada archivo debido a la encriptación que utiliza (aproximadamente el archivo es un 5% más grande). Existe una sobrecarga en la CPU debido a la encriptación, pero al ser solo una pequeña diferencia en el tamaño del archivo, el rendimiento de la red para leer/escribir es similar al rendimiento ofrecido por los diferentes servicios de almacenamiento subyacentes.

**Escalabilidad del sistema.** La capa de almacenamiento de nuestra arquitectura no es un cuello de botella para la escalabilidad. Los sistemas de almacenamiento en la nube contemporáneos son altamente escalables [25]. El subsistema de Atlas también es simple de escalar porque no indexa los archivos individuales de usuarios o los trozos de archivos, sino que indexa los punteros para el almacenamiento en el back-end. Los back-ends de almacenamiento lidian con la mayor parte de los datos relacionados con la lectura/escritura (read/write), y la red Atlas solo está involucrada cuando (a) un usuario está cambiando o actualizando sus backends de almacenamiento, o sus mapas de claves públicas, o, (b) cuando se registra un nuevo usuario en el sistema. A la hora de registrar dominios/usuarios, los archivos con los hashes para el enrutamiento deben ser registrados en la Stacks blockchain. Si bien la blockchain pudiera ser un cuello de botella en cuanto a la escalabilidad (en comparación con la red Atlas), se ha programado para que esto sea extremadamente infrecuente para un usuario. Además, el uso de los registros de nombres *offchain* permite registrar a unos cien usuarios en una sola transacción en la blockchain, siendo capaz de soportar el

registro de más de cientos de miles de usuarios al día (comparable al número de usuarios nuevos al día en las plataformas tradicionales de almacenamiento en la nube). Escalar Gaia para miles de millones de usuarios, conlleva, lo más probablemente, encontrarse en la práctica con problemas no conocidos actualmente, además de tener que abordar estos desafíos, también conlleva un área de investigación tanto a futuro como a día de hoy.

## 5. Autentificación

Las cuentas de usuarios son esenciales para las aplicaciones de internet. Blockstack proporciona a los usuarios con un nombre universal que funciona en todas las aplicaciones sin la necesidad de ningún tipo de contraseñas. En lugar de una autentificación basada en contraseñas, los usuarios se autentifican mediante criptografía de claves públicas: un cliente software que corre en local se encarga de las peticiones de inicio de sesión de las respectivas aplicaciones, así como las peticiones de autentificación de las firmas.

Nuestro protocolo de autentificación, Blockstack Auth, también se conecta con el hub (concentrador) de Gaia del usuario y cualquier clave privada específica de la aplicación. Esta información es utilizada por las aplicaciones para almacenar los datos del usuario y verificar que los datos producidos por otros usuarios son auténticos/ reales.

## 5.1 Inicio de Sesión Único

Blockstack Auth utiliza claves criptográficas para la autentificación. El usuario inicia sesión en una aplicación para poder dotar de permiso a la aplicación para generar y almacenar datos firmados, de tal manera que otros usuarios puedan leer y autentificar los datos. Esto, a su vez, prueba a los otros usuarios que el inicio de sesión del usuario es legítimo.

En Blockstack el propósito del inicio de sesión es para proporcionar al cliente de la aplicación la información suficiente para generar y almacenar datos auténticos. Esto significa que la funcionalidad de la autentificación *solo puede "correr" en el ordenador del usuario* en la forma de una aplicación de autentificación. Debido a que todos los nombres están registrados en la Stacks blockchain, cada una de las aplicaciones y autentificadores pueden tener una visión actualizada de (1) todos los nombres que existen y, (2) todas sus claves públicas y hubs en Gaia. Evitando así, por la parte del servidor, la necesidad de un proveedor de identidades.

El cliente de una aplicación necesita poder conectar con otro par en la Stacks blockchain para poder autentificar los datos del usuario. El usuario proporciona a la aplicación la dirección de la red de su par de Stacks preferido a la hora de realizar el inicio de sesión.

Un usuario firma en una aplicación Blockstack al clicar el botón de "inicio de sesión". La aplicación (a través del SDK *blockstack.js*) redirige al usuario a su aplicación de autentificación Blockstack pidiéndole permiso para iniciar la sesión. Al usuario se le presenta la opción de utilizar los diferentes IDs de Blockstack para iniciar sesión, además de una lista con los diferentes permisos que la aplicación necesita por parte el usuario. Tras seleccionar el ID, el autentificador redirige al usuario de vuelta a la aplicación, pasándole a la aplicación estas tres piezas de información:

1. El nombre del usuario (o el hash de su clave pública, si es que aún no tienen un nombre de usuario).
2. Una *aplicación específica* de la clave privada para encriptar y firmar los datos del usuario. Esta clave es generada deterministicamente a partir de clave

privada maestra, el ID que ha utilizado para iniciar la sesión, y el origen de la aplicación HTTP.
3. La URL para el hub de Gaia del usuario y su par de Stacks preferido para utilizar a la hora de buscar otros usuarios y sus datos.

Al hacerlo, el usuario presenta su nombre de usuario, e indica a la aplicación dónde puede encontrar y almacenar sus datos. A partir de ahí, la aplicación puede leer y escribir datos específicos de la aplicación de manera persistente, y acceder a los datos específicos de otros usuarios, - todo esto sin necesidad de proporcionar una solución de almacenamiento o identidad, propia.

El acto de cerrar la sesión simplemente se realiza para borrar el estado local de la aplicación, lo que causa que el navegador web y el cliente olviden la clave privada específica de la aplicación.

## 6. Librerías y SDKs de Blockstack

Blockstack PBC, una sociedad sin ánimo de lucro (Public Benefit Corporation, en inglés), que junto con colaboradores del mundo open-source hemos desarrollado el núcleo de los protocolos y las librerías para Blockstack. Las librerías para desarrolladores hacen que sea más fácil desarrollar aplicaciones en la red Blockstack, y el cliente de Blockstack, permite al usuario interactuar con varios componentes de la red Blockstack, así como con diversas aplicaciones.

## 6.1 Librerías para Desarrolladores

Blockstack está diseñado para que los desarrolladores puedan desarrollar aplicaciones descentralizadas de la manera más fácil posible. La mayor parte de la complejidad a la hora de interactuar con la Stacks blockchain o con las aplicaciones de almacenamiento descentralizado es que, está oculto a los desarrolladores de aplicaciones, por lo que estos, pueden centrarse solo en la lógica de su aplicación. El repositorio open-source de Blockstack contiene librerías para desarrollar sobre distintas plataformas: Un SDK de Javascript WEB (*blockstack.js)* y SKDs para móviles iOS y Android. Todas estas librerías son proporcionadas bajo los términos y condiciones de la licencia MIT, y se encuentran disponibles en https://github.com/blockstack.

Estas librerías proporcionan todas las APIs y código necesario para implementar un protocolo de autentificación, que interactúa directamente con los servidores Gaia, y genera transacciones Stacks. Usar estas librerías permite a los desarrolladores crear aplicaciones descentralizadas que respeten la seguridad y privacidad del usuario de una manera tan simple como si estuvieran desarrollando una aplicación tradicional.

**Radiks**. Para las aplicaciones que deseen compartir datos a través de gráficas sociales complejas, lo que suele ser más útil y eficiente, es, construir *indexes* sobre esos datos. El sistema Radik es una librería de servidor y una de cliente para desarrollar e interactuar con cualquier index. Las librerías Radik permiten a los desarrolladores crear colecciones de datos estructurados de usuarios cruzados dentro de la aplicación, que pueden ser consultados por el valor de los campos. Esto requiere que, por parte del servidor, un componente procese los indexes y las queries, pero crucialmente, que no sea un parte fiable de la computación base de un usuario. Este, solo es capaz de ver los datos de manera cifrada, y algunos metadatos que son necesarios para desarrollar y contestar las queries sobre el index. Puede encontrar más información sobre Radiks en la referencia [26].

## 6.2 Uso del Software

Mientras los desarrolladores de aplicaciones van a usar los SDKs y las librerías para interactuar con la red Blockstack, a los usuarios se les requiere que instalen un software para realizar funciones como el registro de nombres de usuario, designación de los servidores Gaia, y para la autentificación con las aplicaciones. El ecosistema de Blockstack actualmente proporciona dos proyectos open-source que permiten a los usuarios interactuar con la red:

1. **Navegador Blockstack.** Actualmente esta es la implementación del código open-source de referencia para las aplicaciones con autentificación, y permite a los usuarios navegar a través de las aplicaciones disponibles en Blockstack, registrar nombres de usuario, y autentificarse con aplicaciones. Está disponible para instalar en local en el ordenador, o también en formato web.
2. **Blockstack CLI.** Esto es una utilidad de línea de comando que permite a los usuarios avanzados y a los desarrolladores interactuar con el protocolo Blockstack. Además de proporcionar la funcionalidad de autentificación, permite a los usuarios generar transacciones “crudas” e interactuar con una avanzada tarea de gestión de datos con Gaia.

## 6.3 Documentación y Recursos de la Comunidad

La comunidad open-source de blockchain tiene tutoriales, documentación sobre las Api y documentos con el diseño del sistema que se encuentran disponibles en Github y en la web https://docs.blockstack.org

Los usuarios y desarrolladores de Blockstack tienen a su disposición los siguientes recursos “oficiales” de la comunidad para su uso.

- **Github:** Todo el desarrollo de Software tiene lugar a través de Github, https://github.com/blockstack
- **Foro:** Los usuarios avanzados y los desarrolladores suelen contestar a preguntas técnicas, comparten ideas, y se ayudan entre ellos en el foro disponible en https://forum.blockstack.org
- **Slack:** La comunidad de Blockstack utiliza el grupo público de Slack para chatear en tiempo real, disponible en https://blockstack.slack.com

# 7. Aplicaciones y Servicios

Desde principios del 2019, ya hay más de 100 aplicaciones desarrolladas en Blockstack. Los desarrolladores están construyendo diferentes tipos de aplicaciones y hay una lista completa, en crecimiento constante, de aplicaciones sobre Blockstack en *app.co*. Dado que Blockstack es modular, distintas aplicaciones pueden utilizar distintos componentes de manera independiente. Abajo os mostramos una visión general sobre algunos casos de uso.

Aplicaciones actuales de Blockstack en la productividad de una oficina [27, 28, 29, 30] dónde utilizan el autentificador y el almacenamiento Gaia para permitir a sus usuarios crear, editar y compartir documentos. Para poder ayudar a los usuarios a descubrir los documentos de otros usuarios, la app utiliza un indexador de búsqueda de perfiles de Blockstack. Este indexador de búsqueda es descentralizado – por lo que todos los perfiles son visibles y accesibles globalmente, cualquiera puede desplegar y correr un indexador de búsqueda de perfil.

El ecosistema Blockstack también tiene numerosas aplicaciones sociales [31, 32, 33, 34]. Típicamente, estas aplicaciones usan el autentificador de Blockstack en combinación con el despliegue del servidor Radiks para permitir a los usuarios de manera eficiente poder descubrir y extraer los datos de otros usuarios. Al menos en un caso, la aplicación dedica un canal de transmisión (relay channel) para enrutar mensajes encriptados a través de diferentes usuarios [31]. Las aplicaciones de publicidad y almacenamiento en Blockstack [35, 36, 37, 38, 39, 40], no solo utilizan Gaia para almacenar datos del usuario, sino que también lo utilizan para compartirlos con los no-usuarios de Blockstack a través de métodos convencionales como HTTP URLs.

**Incentivos a los Desarrolladores.** La Stacks Blockchain ensancha el concepto de minado, hasta tal punto que los desarrolladores de aplicaciones pueden “minar” tokens Stacks al desarrollar una aplicación de alta calidad que funcione sobre la red. Este mecanismo, llamado, Minado de Aplicaciones, está diseñado para tener un mecanismo que incentive desarrollar aplicaciones de alta calidad en la red. El programa de minado de aplicaciones actualmente está siendo coordinado por Blockstack PBC y varios críticos independientes (freelancers). Los desarrolladores pueden presentar su aplicación para que sea revisada una vez al mes de tal manera que reciben un pago en función del resultado que obtenga la aplicación a través del mecanismo de ranking de aplicaciones. Las aplicaciones son revisadas por una serie de críticos independientes (freelancers), cada uno con su propio criterio sobre lo que se necesita para tener una aplicación de calidad. La aplicación tiene una *puntuación agregada* que determina su orden en el ranking. Los rankings y pagos son publicados mensualmente. Para más detalles sobre el Programa de Minado de Aplicaciones, pueden encontrar más información fuera de este documento deben ir a la web https://app.co/mining

## 8. Conclusión

Blockstack es una red de computación descentralizada que proporciona una solución full-stack para que los desarrolladores pueden desarrollar aplicaciones descentralizadas. Hasta la fecha, más de 100 aplicaciones descentralizadas han sido desarrolladas en la red. Blockstack elimina la necesidad de que los desarrolladores tengan que utilizar servidores y bases de datos: en su lugar, las aplicaciones escriben datos sobre los contenedores privados de datos que controlan los usuarios. Este sistema de almacenamiento descentralizado ofrece un rendimiento similar a las soluciones de almacenamiento en la nube tradicionales y sólo introduce una pequeña sobrecarga debido a la encriptación y desencriptación. Nuestro protocolo de autentificación elimina la necesidad de los inicios de sesión basados en contraseñas que son conocidos por ser menos seguros que la autentificación criptográfica. Los usuarios son capaces de usar un solo perfil través de diferentes servicios y aplicaciones, eliminando la necesidad de estar creando constantemente nuevas cuentas de perfil en los servicios. Nuestras librerías para desarrolladores hacen que desarrollar aplicaciones descentralizadas sea tan sencillo como desarrollar aplicaciones tradicionales de internet.

En este documento, hemos presentado el último diseño de Blockstack. Desde la implementación en producción en el 2016 y 2017, el núcleo del diseño de Blockstack ha evolucionado e incorporado las lecciones aprendidas en la puesta en producción de estos sistemas y los comentarios de los desarrolladores de aplicaciones descentralizadas. Los mayores cambios desde el whitepaper de 2017 incluyen (a) descripciones sobre la Stacks blockchain que utiliza un nuevo mecanismo de pruebas afinables para arrancar (bootstrap) una nueva blockchain, y, (b) descripción sobre nuestro nuevo lenguaje para programar smart contracts que se centra en la seguridad y predictibilidad de los smart contracts. Hemos lanzado Blockstack como proyecto open-source.

## Agradecimientos

A lo largo de los años, mucha gente ha contribuido tanto en el diseño como en la implementación de Blockstack. Queríamos agradecer a Larry Salibra, Ken Liao, Guy Lepage, Patrick Stanley, y John Light por sus tempranas contribuciones e ideas. Hank Stoever, Shreyas Thiagaraj, y Matthew Little por su trabajo sobre las librerías para desarrolladores y SDKs. Jeff Domke, Mark Hendrickson, Thomas Osmonson, Jasper Jansz, y Mary Anthony por el diseño del  roduct y la documentación para desarrolladores; Jesse Wiley, [redacted] y Tim Wells por el desarrollo de la infraestructura; Brittany Laughlin y Diwaker Gupta por su fundamental ayuda con comentarios y feedback. Puede encontrar más contribuciones a Blockstack en: https://github.com/blockstack pero generalmente, con que mucha gente ha contribuido en este proyecto, no es posible agradecer uno a uno sus contribuciones en este documento – estamos enormemente agradecidos por el apoyo de la comunidad open-source de Blockstack.

Gracias a nuestra comunidad por la ayuda en esta traducción del inglés al español.

## Referencias

[1] S. Sulyman,“Client-server model,”IOSR Journal of Computer Engineering, vol. 16,pp. 57–71,012014.

[2] N.Perlroth,“Yahoo says hackers stole data on 500 million users in 2014,”Sept.2016.http://nyti.ms/2oAqn0G.

[3] K. Granville, “Facebook and cambridge analytica: What you need to know as fallout widens,” Mar.2018. https://nyti.ms/2HP4Dr3.

[4] R. Mcnamee, “I mentored mark zuckerberg. i loved facebook. but i can’t stay silent about what’s happening.,” Jan. 2019. http://time.com/5505441/mark-zuckerberg- mentor- facebook- downfall/.

[5] “Blockstackwebsite,”2019.http://blockstack.org.

[6] J.H. Saltzer, D.P. Reed, and D.D.Clark,“End-to-end arguments in system design,”ACMTrans.Com-put. Syst., vol. 2, pp. 277–288, Nov. 1984.

[7] D. D. Clark and M. S. Blumenthal, “The end-to-end argument and application design: The role of trust,” Federal Comm. Law Journal, vol. 63, no. 2, 2011.

[8] M. Ali, J. Nelson, R. Shea, and M. Freedman,“Blockstack:A global naming and storage system secured by blockchains,” in Proc. USENIX Annual Technical Conference (ATC ’16), June 2016.

[9] J. Nelson, M. Ali, R. Shea, and M.J. Freedman,“Extending existing blockchains with virtual chain,”in Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL’16), (Chicago, IL), June 2016.

[10] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, “Bootstrapping trust in distributed systems with blockchains,” USENIX ;login:, vol. 41, no. 3, pp. 52–58, 2016.

[11] Satoshi Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” tech report, 2009. https://bitcoin.org/bitcoin.pdf.

[12] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton, NJ, USA: Princeton University Press, 2016.

[13] V. Buterin,“A next-generation smart contract and decentralized application platform,”tech.rep.,2017. https://github.com/ethereum/wiki/wiki/White- Paper.

[14] “Filecoin: A Cryptocurrency Operated File Network,” tech report, 2014. http://filecoin.io/filecoin.pdf.

[15] https://eos.io.

[16] T. Hanke , M. Movahedi, and D. William, “Dfinity technology overview series consensus system rev.1,” 2018. https://dfinity.org.

[17] “Ethereum 2.0 specifications,” 2019. https://github.com/ethereum/eth2.0-specs.

[18] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, “A survey and comparison of peer- to-peer overlay network schemes,” IEEE Communications Surveys Tutorials, vol. 7, pp. 72–93, Second 2005.

[19] “Oauth.” https://oauth.net.

[20] “Blockstack Core: Stacks blockchain v1,” 2018. https://github.com/blockstack/blockstack- core/tree/v20.0.8.1.

[21] “SIP 001: Burn Election,” 2019. https://github.com/blockstack/blockstack-core/blob/develop/sip/sip- 001- burn- election.md.

[22] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM ’01, pp. 149–160, 2001.

[23] “Appmining.”https://app.co/mining/.

[24] “SIP 002: Smart Contract Language,” 2019. https://github.com/blockstack/blockstack-core/blob/develop/sip/sip- 002- smart- contract- language.md.

[25] “GoogleCloudStorageSLA.”Retrievedfromhttps://cloud.google.com/storage/slainMay 2017.

[26] “Radiks.” https://github.com/blockstack-radiks.

[27] J.E.Hunter,“Graphite docs,”Feb.2019.https://app.graphite-docs.com.

[28] D.Travino,“Noteriot,”Feb.2019.https://note.riot.ai/.

[29] “Forms.id,”Feb.2019.https://forms.id.

[30] “Blockusign,”Feb.2019.https://blockusign.io.

[31] P.BhardwajandA.Carreira,“Stealthy,”Feb.2019.https://www.stealthy.im.

[32] A. Sewrathan, R. Adjei, and F. Madutsa,“Afari,”Feb.2019.https://afari.io.

[33] T.Alves,“Recall,”Feb.2019.https://app.recall.photos/.

[34] T.Alves,“Travelstack,”Feb.2019.https://app.travelstack.club.

[35] J.E.Hunter,“Graphitepublishing,”Feb.2019.https://publishing.graphitedocs.com.

[36] “Decs,”Feb.2019.https://app.decs.xyz.

[37] “Sigle,”Feb.2019.https://app.sigle.io.

[38] “Xorbrowser,”Feb.2019.https://xorbrowser.com.

[39] “Mevaul,”Feb.2019.https://mevaul.com/.

[40] “Xordrive,”Feb.2019.https://xordrive.io.

[41] “Blockstack source code release v 20.0.8,”2019.http://github.com/ blockstack/blockstack- core.