Incorporació d'un Explorador de blocs en un simulador de la Lightning Network

Robert Barrero Colao

Resum–
El protocol Lightning Network (LN) ha estat una de les solucions més adoptades per resoldre el problema d'escalabilitat de Bitcoin. Actualment, existeixen diversos simuladors de la LN que permeten l'estudi i anàlisi d'aquesta tecnologia, però cap opció incorpora un explorador de blocs. Aquest article aborda el desenvolupament d'un treball de fi de grau que té la finalitat d'integrar-ne un al simulador Polar, per tal d'ampliar les utilitats que ofereix. Així mateix, es presenten els aspectes fonamentals de cadascuna de les tecnologies involucrades en aquest projecte.
Paraules clau– Bitcoin, Blockchain, Docker, Explorador de blocs, Lightning Network, Peer-to-peer, Regtest

Abstract– The Lightning Network (LN) protocol has been one of the most widely adopted solutions to solve Bitcoin's scalability problem. Currently, there are several LN simulators that allow the study and analysis of this technology, but none of them incorporates a block explorer. This article deals with the development of a final degree project that aims to integrate one into the Polar simulator, in order to extend the utilities it offers. It also presents the fundamental aspects of each of the technologies involved in this project.
Keywords– Bitcoin, Blockchain, Block explorer, Docker, Lightning Network, Peer-to-peer, Regtest

1 Introducció

Des de moments propers als seus orígens, Bitcoin [1] ha suposat una revolució per als mercats i comerç digitals i ha esdevingut en una de les referències dels protocols de criptomonedes. Tanmateix, disposa de certs dèficits vinculats a la seva escalabilitat i capacitat per gestionar petites transaccions. Una de les solucions a aquests problemes amb més acollida en l'actualitat és el protocol Lightning Network (LN) [2], un esquema basat en canals bidireccionals entre nodes *peer-to-peer* i situat per sobre del protocol Bitcoin, que permet conduir petites transaccions externes a la cadena de blocs. Per tal de facilitar la comprensió d'aquestes tecnologies, hi ha simuladors de xarxes LN, com Polar [3], que permeten el seu estudi amb una interfície interactiva i senzilla.

En aquest context, l'objectiu d'aquest treball és incorporar un explorador de blocs [4] al simulador Polar per tal d'ampliar les seves capacitats per analitzar i reproduir escenaris amb aquests protocols.

Aquest article comença per una breu explicació de la motivació darrere del projecte, els seus objectius i la metodologia emprada per complir-los. Seguidament, es realitza una presentació de les tecnologies principals amb les quals es tracta, així com el context actual dels exploradors de blocs, les seves característiques i la selecció de l'explorador a integrar amb el simulador. Tot seguit, es fa una breu descripció de la part de desenvolupament del projecte, i d'un conjunt de proves que verifiquen que el resultat del treball dut a terme coincideix amb els objectius planificats.

• E-mail de contacte: robert.barrero@autonoma.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Cristina Pérez Solà (dEIC)
• Curs 2021/22

1.1 Motivació

Dins el context dels simuladors com Polar, no hi ha evidències de cap que contingui la funcionalitat d'un explorador de blocs. Tenint en compte les capacitats d'anàlisis de xarxes que ofereix un explorador, poder integrar-ne un dins Polar i així combinar dues eines de gran utilitat analítica, ha estat la principal motivació.

Altrament, i en el que és personal, dur a terme aquest projecte dona la possibilitat de poder treballar amb una bona varietat de tecnologies recents i alhora, permet aprofundir en l'estudi i coneixement dels protocols Bitcoin i LN.

1.2 Objectius

Els objectius del projecte es divideixen en principals i secundaris. Els primers són implicits del projecte, li donen el sentit i el defineixen, i els segons sorgeixen com a conseqüència dels primers.

Dintre dels objectius principals es troben els següents:

1. Ampliar les funcionalitats interactives que ofereix el programa Polar, mitjancant la incorporacio d'un explorador de blocs.
2. Aprofundir en el coneixement sobre la matèria de les tecnologies Blockchain, concretament sobre els protocols Bitcoin i LN i donar continuitat a l'assignatura *Tecnologia Blockchain i Criptomonedes*.
3. Contribuir a un projecte de codi obert.

Quant als objtectius secundaris, aquests són:

1. Obtenir com a resultat del projecte, documentació tècnica i de qualitat que doni suport al treball desenvolupat.
2. Ampliar el coneixement de Docker [5].
3. Comprendre l'arquitectura i el funcionament dels exploradors de blocs, i com interaccionen els serveis que els componen.
4. Comprendre com funciona a baix nivell el simulador Polar.
5. Posar a prova els hàbits de treball, el compromís i l'organització personal.

1.3 Metodologia

La metodologia principal que ha definit el fil de desenvolupament del projecte ha estat la metodologia cascada. Es tracta d'un projecte de petita escala, amb fases i activitats definides on la majoria ha requerit la finalització de l'activitat precedent per tal de continuar. A raó d'això, aquesta metodologia és la que millor s'ha adaptat a la pràctica.

Per a la gestió de l'organització a nivell individual del treball, s'ha empleat la metodologia Kanban [6] a través de l'aplicació Trello, fent servir el taulell de gestió d'activitats que proporciona per tal de fer el seguiment de l'estat en que es troben les tasques que conformen cada activitat planificada.

D'altra banda, i paral·lelament al desenvolupament del treball, s'ha anat realitzant documentació del procés a mesura que s'anaven realitzant avenços, per tal d'anar elaborant progressivament una base per al contingut d'aquest mateix article, que a més ha servit de suport per al desenvolupament del projecte.

Quant al seguiment en conjunt amb la tutora, s'han anat realitzant sessions de seguiment i de plantejament de dubtes amb una frequència d'entre una i dues setmanes.

1.4 Aspectes ètics i de tractament de dades

En relació amb les dades que es produeixen mitjançant el programari amb el qual es tracta en aquest projecte, aquestes són d'àmbit privat, no són productives ni generen cap benefici i no es comparteixen. El simulador amb el qual es treballa, funciona mitjançant l'ús de xarxes Regtest [7]. L'ús d'aquestes xarxes no comporta cap conseqüència en la xarxa original, tal com s'explica en l'apartat 2.3 d'aquest article.

2 Tecnologies involucrades

A continuació, es presenten les tecnologies fonamentals que es tracten al llarg del projecte.

2.1 Protocol Bitcoin

Bitcoin, sovint definit com una criptomoneda, consisteix en un conjunt de tecnologies que funcionen conjuntament per tal d'oferir un ecosistema d'intercanvi de divises digitals. Aquest ecosistema és, conceptualment, similar a una gran base de dades distribuïda que té la peculiaritat d'admetre noves dades, i no pas modificar-les o eliminar-les. Per tant, es tracta d'un sistema complex que és resistent a modificacions i consistent amb el pas del temps.

Aquestes característiques les obté a raó de les tecnologies en les quals es fonamenta: primerament, parteix de la tecnologia de cadena de blocs, i adquireix la seva estructura de dades com a base per a l'emmagatzematge. D'altra banda, es constitueix d'un sistema *peer-to-peer* distribuït i descentralitzat, on cada node participant es comunica amb la resta fent servir un protocol de consens particular, que garanteix un acord global sobre l'estat de la cadena i que possibilita que la informació present a la mateixa sigui pública i verificable. L'altre pilar

fonamental de Bitcoin és la criptografia, principalment l'ús de funcions hash i signatures digitals basades en clau pública, que permeten garantir la integritat, coherència i seguretat de l'estat de la cadena a més de facilitar la seva comprovació, entre d'altres.

Bitcoin emmagatzema els pagaments que es van generant amb el temps i cadascun d'aquests, s'efectua mitjançant unes operacions anomenades transaccions, que es verifiquen per mitjà del protocol de consens i seguidament, queden enregistrades en la cadena de blocs.

Aquesta tecnologia, que tant ha donat de que parlar, també presenta una sèrie de limitacions relacionades amb la seva escalabilitat. A causa de la naturalesa del protocol de consens, la concatenació de blocs requereix un temps mínim (aproximat als deu minuts), mentre que, d'altra banda, la mida dels blocs està limitada al mega byte. A raó d'això, Bitcoin presenta un *throughput* limitat que no permet operar amb el mateix volum de dades per segon que ho fa un sistema bancari tradicional, com és el cas de Visa. També, és poc rendible tractar amb transaccions petites degut a la quantitat de comissions que continuen requerint.

És en aquest punt, on té sentit parlar de la segona gran tecnologia relacionada amb l'àmbit d'aquest projecte: el protocol LN, la solució als problemes d'escalabilitat de Bitcoin que està tenint més adopció en l'actualitat.

2.2 Protocol Lightning Network

La LN consisteix en una xarxa *peer-to-peer* que actua per sobre i conjuntament amb la xarxa Bitcoin i que està constituïda per nodes connectats entre si mitjançant canals bidireccionals. Aquesta xarxa permet efectuar micropagaments de manera quasi instantània a través d'aquests canals amb l'objectiu de repercutir de manera mínima en l'escalabilitat de Bitcoin, tan sols requerint la publicació d'una transacció per obrir un canal i una segona per tancar-lo. Cada canal es crea entre dos participants de la xarxa Bitcoin i la transacció d'obertura és de tipus multi signatura 2 de 2. Aquesta transacció determina el balanç en bitcoins que tindrà el canal, i és a partir d'aquest balanç que es pot realitzar diferents micropagaments.

Una de les aspiracions de la LN, al marge d'esdevenir en una solució sòlida al problema d'escalabilitat que pateix el protocol Bitcoin, és permetre-li suportar totes les operacions econòmiques que es fan en temps real globalment. Es tracta d'una tecnologia amb perspectiva de futur, no obstant això, per mantenir-la, cal el suport d'usuaris i de desenvolupadors que participin en el seu ús i l'ajudin a créixer. És aquí on són necessàries certes eines que ajudin a comprendre el funcionament complex d'aquest protocol. Una d'elles és l'aplicació Polar, de la qual se'n parla a continuació.

2.3 Aplicació Polar

Polar és un simulador de xarxes LN, de codi obert i escrit en Typescript i React, que destaca per la seva facilitat d'ús i per permetre crear escenaris funcionals que implementen els protocols Bitcoin i LN, d'una manera realment senzilla en comparació a les altres alternatives. Es tracta del programa protagonista d'aquest treball, on es desenvolupa la integració de l'explorador de blocs.

Fa ús de la tecnologia Docker per tal de desplegar els nodes que composen cada escenari. Els nodes són contenidors que executen el servei corresponent (Bitcoin o Lightning) i el programa fa servir Docker-compose [8] per gestionar-los en un entorn aïllat.

Adicionalment, fa servir el mode Regtest de Bitcoin, el qual permet originar xarxes Bitcoin d'ús local i sense restriccions quant a la generació de blocs. Les xarxes Regtest funcionen amb el mateix protocol, però utilitzen el seu propi actiu imitant a l'original. Són essencials per tal de poder desenvolupar aplicacions que fan servir aquestes tecnologies i posar a prova diferents escenaris i possibles millores del protocol, ja que simulen el funcionament de la xarxa original, però en un entorn local on es té control dels nodes que hi participen i els actius i dades que es generen no són compatibles amb les xarxes operatives, ni tenen cap valor. Aquest fet contribueix a que cada escenari Polar sigui ideal per reproduïr diferents situacions dels protocols Bitcoin i LN sense tenir cap impacte real.

És una aplicació potent que inclou una llarga llista de virtuts entre les quals es destaca la representació gràfica i completament interactiva de les xarxes, que mostra la informació rellevant de forma visual, com els balanços de cada canal o qui inicia cada comunicació. També, permet crear nous nodes Bitcoin i Lightning, de diferents versions i implementacions, crear o tancar canals, minar blocs manualment i fins i tot, accedir a la interfície CLI de qualsevol node per consultar més detalls que no estan disponibles a simple vista.

A la figura 1 es mostra un petit esquema per tal de donar una idea aproximada de l'arquitectura de Polar.

3 Exploradors de blocs

En aquesta secció, es tracta l'estat de l'art dels exploradors de blocs, la seva trajectòria i altres aspectes fonamentals com les tecnologies en les quals es basen o la seva utilitat. També, s'elabora la selecció de l'explorador en qüestió per ser integrat amb Polar.

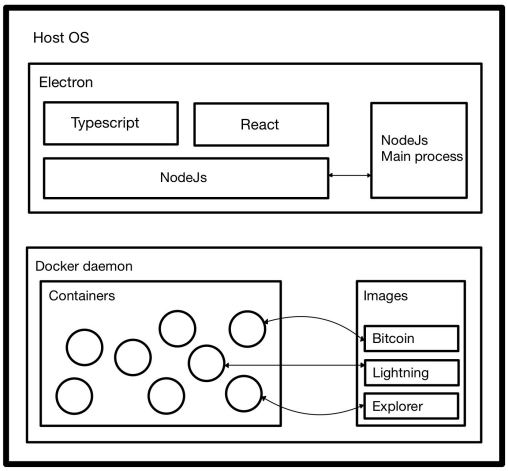

Fig. 1: Arquitectura de Polar

3.1 Definició

Els exploradors de blocs consisteixen en navegadors específics de Blockchain, freqüentment elaborats en forma de servei web, que permeten la recerca i visualització d'informació fonamental i de gran utilitat relacionada amb les transaccions i els blocs que hi tenen lloc, monitorar en temps real l'estat de la cadena i realitzar el seguiment del seu historial recent, entre d'altres. De forma alternativa, es poden veure com motors de cerca, ja que permeten la consulta i anàlisi de qualsevol transacció, adreça, o bloc a través del seu identificador o alçada. Un altre dels aspectes importants dels exploradors de blocs, és el fet que presenten la informació de manera llegible i gràfica, facilitant la seva comprensió i esdevenint en una eina no només de consulta sinó també formativa. És, per tant, que els exploradors de blocs juguen un paper rellevant en la transparència envers els clients que ofereix la tecnologia Blockchain i la seva experiència d'ús.

3.2 Història i precedents

L'inici d'aquest tipus de programari és proper als orígens de la tecnologia Blockchain i van sorgir com a necessitat per tenir disponible una eina que facilités la consulta d'informació sobre les transaccions, sovint relacionada amb l'estat d'aquestes, el nombre de confirmacions, o detectar possibles atacs de doble despesa [9].

Aquesta informació era de difícil accés a raó de la naturalesa no llegible que té la indexació i emmagatzematge d'aquesta, i l'única manera directa que hi havia de consultar-la era mitjançant una interfície de línia d'ordres (o CLI), i un node complet. Per consegüent, es tractava d'una solució poc útil per al públic general.

Llavors, en novembre de l'any 2010 es va presentar BlockExplorer.com [10], conegut com un dels primers exploradors de blocs públics i creat per Theymos, d'ús exclusiu per a l'exploració de la Blockchain de Bitcoin. Es tractava d'un explorador que ja permetia comprovar la informació de blocs, transaccions i adreces de manera llegible així com altra informació en temps real, fent servir una interfície força senzilla que podria recordar a la interfície oferta per l'explorador Abe [11]. Seguidament, el juny de 2011 es va anunciar aquest últim, esdevenint en un dels primers exploradors de codi lliure i amb una interfície basada en la qual utilitzava BlockExplorer.com. Encara que la identitat del primer explorador de blocs és incerta, aquests dos exponents van ser dels primers que van aparèixer i esdevenir en una certa referència i punt de partida per a la resta d'exploradors que avui dia es coneixen.

3.3 Informació proporcionada per un explorador de blocs

Com s'ha pogut veure, els exploradors de blocs juguen un paper clau tant en la comprensió i utilització de la tecnologia Blockchain. A continuació, es descriu la informació que atorga un explorador de blocs habitual que, en aquest document, s'organitza en tres grups per tal de classificar-la: dades estrictament presents a la cadena, metadades i dades complementàries.

El primer de tots, està format per aquelles dades que estan explícitament presents a la cadena. La informació dels blocs que pertany a aquest conjunt de dades inclou: la seva mida, les transaccions que incorporen i la quantitat total, el *timestamp*, el *nonce* [12], la seva versió, l'arrel de Merkle [13], i l'apuntador hash [14] del bloc previ, entre d'altres. En relació a les transaccions, els exploradors habitualment mostren: la seva versió, el temps de bloqueig [15], i les entrades i sortides d'aquestes. Seguidament i dins de les sortides, es troba el valor associat a l'import de cada una i, tant a dins de les entrades com de les sortides, hi ha contingut el codi dels scripts [16] (*ScriptSig* per al desbloqueig i *ScriptPubKey* per al bloqueig, respectivament).

El segon grup, el de les metadades, són aquelles dades que s'obtenen directament a través del conjunt anterior, però, aquestes no es troben explícitament a la cadena. Les metadades que els exploradors acostumen a mostrar en referència a les transaccions són: els identificadors, les comissions, el seu tipus, la mida (en Bytes) i la mida virtual (en vBytes) de les transaccions. I en referència als blocs, aquestes són: el hash que els identifica, l'alçada, la dificultat en diferents formats, el *block reward* (valor generat en crear-se), entre d'altres. Dins aquesta categoria també s'inclouen les comissions per byte pagades per a cada transacció i total dels blocs, el pes [17] de tots dos, el *hashrate* [18] i el total de diners que estan en circulació a la xarxa.

Quant a la informació complementària, consisteix en aquella que depèn de l'estat d'un node particular i serveix per donar suport a la resta d'informació. Aquí, es trobarien les dades de la *memory pool* [19], incloent-hi el seu volum, quantitat de transaccions que inclou o el seu ús de memòria. També, les adreces IP dels nodes que comparteixen un bloc nou, qualsevol mena de registres d'esdeveniments, o la identificació de les associacions de mineria [20] (*mining pools*) i quins blocs han minat.

3.4 Arquitectura habitual d'un explorador de blocs

Habitualment, els exploradors de blocs obtenen la informació a través d'un node complet connectat a la cadena de blocs, i mitjançant el conjunt de tecnologies que facin servir, l'emmagatzemen en una base de dades i la disposen als usuaris per mitjà de la interfície web (veure figura 2).

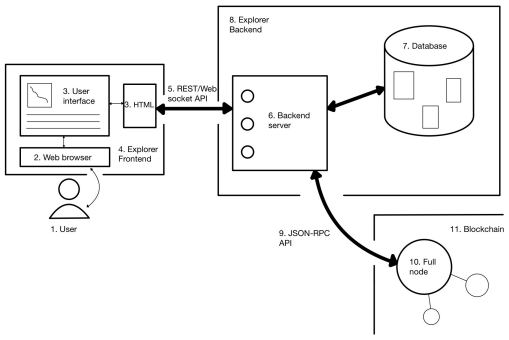

Fig. 2: Arquitectura habitual d'un explorador de blocs.

Com es pot apreciar a la figura 2, el servidor backend de l'explorador (element 6) és l'encarregat de coordinar les comunicacions entre el node complet (element 10), la base de dades (element 7) i el frontend de l'explorador (element 4). Sovint, aquesta informació l'aconsegueixen mitjançant crides RPC [21] a través d'una API, dirigides al node que està connectat a la blockchain, però també existeixen maneres alternatives d'obtenir aquesta informació, com per exemple mitjançant la lectura directa dels fitxers d'informació que contenen les dades originals. Aquestes dades estan originalment en format binari, i no són aptes per a la seva interpretació humana.

Per aquest motiu, l'explorador emmagatzema en una base de dades la informació que va assolint mitjançant el node connectat, de manera organitzada i en un format estructurat, que permet realitzar cerques posteriorment. Tanmateix, no tots els exploradors fan servir una base de dades, però la seva utilització influeix en el seu rendiment.

Llavors, una vegada es té la informació emmagatzemada i organitzada, l'explorador podrà establir les cerques eficients, que donin resposta al que l'usuari demani mitjançant la interfície web (element 3). Aquestes peticions s'envien al servidor backend mitjançant una API (habitualment de tipus REST o WebSockets, representada per l'element 5), que respon a la interfície d'usuari (element 3) i aquesta, juntament amb l'API, envien la informació en format HTML al navegador de l'usuari.

3.5 Estat de l'art dels exploradors i les seves característiques

L'objectiu principal d'aquest TFG és el d'integrar un explorador de blocs amb un simulador de la LN de Bitcoin. A continuació, s'expliquen les diferents alternatives d'exploradors que s'han tingut en compte per a la tria a integrar. Seguidament, es realitza una comparació d'aquests i finalment es mostra la tria que s'ha realitzat mitjançant uns criteris de selecció, també exposats. Cal indicar que tots ells són exploradors de codi obert i orientats a Bitcoin.

3.5.1 Abe

Abe [11] és un explorador de blocs desenvolupat per John Tobey i anunciat el 26 de juny de 2011. Està desenvolupat amb Python i el seu funcionament és diferent de la resta d'opcions de la comparativa: per tal d'obtenir la informació de la blockchain el que fa és llegir-la directament dels fitxers de dades originals del client de Bitcoin, per després transformar-la i carregar-la a una base de dades SQL. És per això, que té un inici molt més lent que la majoria d'exploradors, això no obstant, mitjançant l'ús de la base de dades juntament amb el fet de no estar basat en crides RPC per al seu funcionament, fan que funcioni relativament de pressa una vegada ja s'ha carregat tota la informació de la cadena. Un problema que té és que si es canvia la còpia de dades, s'ha de tornar a carregar de nou tota la informació.

Una altra qüestió també és que es tracta d'un explorador relativament antic en comparació amb la resta i no és compatible amb les transaccions del tipus *SegWit* [22] i a més requereix funcionar amb les versions 2.6 o 2.7 de Python, les quals ja no estan suportades des de gener de 2020. Per tant, per a la seva instal·lació és recomanable utilitzar un entorn virtual de Python amb les dependències d'Abe instal·lades (també es podria fer servir una imatge de Docker). Per solucionar la incompatibilitat amb les transaccions de tipus *SegWit*, és recomanable descarregar una implementació d'Abe que sí que suporta aquest tipus de transaccions, com és el cas d'aquesta [23]. A causa de la seva antiguitat, disposa d'una interfície força austera, basada en una gran taula d'informació completa amb enllaços URL que porten cap a

la informació i permeten navegar a través dels blocs i transaccions.

3.5.2 BTC RPC Explorer

BTC RPC Explorer [24] és un explorador que recopila la informació d'una manera més habitual a com ho fan la majoria d'exploradors: mitjançant la realització de peticions a través de l'API JSON-RPC [25] cap a un node connectat a la xarxa. És important que el node sigui complet i que la indexació del node arribi fins al primer bloc, per tal de permetre a l'explorador aconseguir tota la informació possible. També funciona amb nodes podats, no obstant extraurà menys informació (blocs sense llistat de transaccions, transaccions sense detalls dels inputs, etc.). Una característica especial d'aquest explorador és que no disposa d'una base de dades com sí ho fan la major part dels exploradors, fent que sigui una alternativa més senzilla de configurar amb la penalització de rendiment que això pot comportar a l'hora de processar grans volums de dades repetidament. D'altra banda, permet observar les dades en format original, historial d'adreces i disposa d'una interfície rica en informació.

3.5.3 Esplora Block Explorer

Esplora Block Explorer [26], igual que altres exploradors de la comparativa, funciona mitjançant la comunicació amb un node connectat mitjançant peticions JSON-RPC. Això no obstant, com a backend, utilitza una API HTTP personalitzada [27] que va per sobre de la interfície de crides RPC que fa servir Bitcoin de forma estàndard. Aquesta API afegeix noms més curts i abreviats a les crides i també l'obtenció d'informació sobre les transaccions més extensa (transaccions en procés de despesa, outputs previs, etc.) Aquesta API també ofereix certes millores de rendiment, com per exemple l'emmagatzematge dels enllaços entre identificadors i les seves transaccions (en format original) corresponents, per tal d'evitar realitzar peticions constantment al servidor bitcoin i, en el seu lloc, realitzar peticions periòdiques només per actualitzar l'estat de la mempool i els nous blocs que es van formant. En aquest sentit, pot recordar al funcionament d'Abe, però dut a terme d'una forma més subtil.

Pel que fa a la informació que dona aquest explorador, es destaca que disposa d'interfície adaptativa, i una vista detallada que permet observar el codi dels scripts en format hexadecimal o bé en assemblador, les dades que pertanyen al *Witness*, i els detalls dels outputs, entre d'altres. Després, encara dona informació bàsica com els detalls dels blocs, adreces i transaccions així com de les seves entrades i sortides. També, cal destacar que ofereix una interfície neta i polida, i amb menys informació que altres exploradors de la comparativa, com és el cas de BTC RPC Explorer o Mempool [28].

3.5.4 Mempool

Mempool destaca per una interfície que mostra gran quantitat de detalls alhora que ho fa d'una manera agradable a la vista. Aquests detalls són d'utilitat, i comprenen des de dades de l'estat general de la blockchain (dificultat global, comissions, transaccions sense confirmar, volum de transaccions que s'estan processant, etc.) fins a detalls de les transaccions valuosos com el seu tipus, si admet *replace-by-Fee* [29] o si és múltiple signatura [30], entre d'altres. També inclou la possibilitat d'afegir quadres gràfics de monitoratge, motor de cerca (també habitual en altres exploradors) i l'observació de dades comunes que ofereixen els altres candidats de la comparativa.

Quant al funcionament d'aquest explorador, està construït en JavaScript (requereix funcionar amb Node.js), i obté la informació de la mateixa manera que Esplora. De fet, fa servir la mateixa API com a backend i, per tant, funciona mitjançant crides RPC que es van fent esporàdicament cada cop que l'usuari vol consultar una nova dada. També fa servir una base de dades de tipus SQL i disposa de dos funcionaments: configuració només per tal de funcionar amb un node bitcoin, o bé funcionament mitjançant un servidor que implementa el wallet Electrum [31] per sobre de bitcoin. Es pot desplegar fàcilment gràcies a la implementació que inclou amb Docker Compose.

3.5.5 Iquidus

Iquidus [32] és un explorador també escrit en JavaScript (Node.js), i molt configurable, que permet una alta personalització de la seva interfície. Destaca per fer servir una base de dades no relacional (MongoDB), a diferència de la resta de candidats que o bé no en fan servir o són de tipus SQL. En aquest cas, requereix una configuració més aviat lenta a causa de la quantitat de configuracions de les quals disposa i principalment a què per defecte no ofereix funcionament mitjançant Docker com si ho fan Mempool o Esplora. Quant a les dades que ofereix per defecte són: un motor de cerca ràpida, les dades habituals relacionades amb els detalls dels blocks, adreces i transaccions i dades de l'estat general de la blockchain en qüestió, com el throughput o la dificultat global.

3.6 Resum comparatiu i selecció de l'explorador a integrar

Quant a la tria de l'explorador a incorporar al programa Polar, s'han tingut en compte certs

criteris: l'experiència d'ús de cadascuna de les interfícies, el nivell de detall d'informació que donen, com es troba disponible aquesta i si disposen de compatibilitat amb SegWit. Aquest últim requisit és força important degut a que una de les motivacions en fer servir l'explorador en un simulador de la LN, és que pugui interpretar aquelles transaccions que obren els canals. Finalment, els criteris més rellevant són la dificultat que pot presentar la seva integració, juntament amb la seva compatibilitat amb Polar.

Un cop s'han observat les característiques que identifiquen a cada explorador i a mesura que s'han anat posant a prova, s'ha elaborat una taula comparativa (taula 1) per tal de resumir el que pot oferir cada explorador pretendent. Es pot observar que tots els candidats ofereixen els detalls bàsics sobre la xarxa en qüestió, i comparteixen gairebé la major part de funcionalitats que poden oferir, excloent Abe, probablement per ser el més antic.

Polar utilitza Docker-Compose per gestionar els nodes dels escenaris (un arxiu YAML per escenari), per aquest motiu, exploradors com Mempool que també el fan servir presenten més dificultats per integrar-se amb Polar. És la simplicitat de BTC RPC Explorer que facilita en gran manera la integració amb el programa (i el motiu principal de la seva tria), ja que està dissenyat per funcionar dins un sol contenidor. També, aquesta senzillesa permet crear una imatge d'una sola capa, mantenint la mateixa estructura que tenen la resta d'imatges que fa servir Polar. A més a més, es tracta d'un explorador que té una interfície fàcil d'emprar i és visualment agradable, encara que no és tan vistosa i treballada com la de Mempool.

4 Desenvolupament de la integració de l'explorador

A continuació, s'explica breument com és el procés de desenvolupament que cal dur a terme per a la integració de l'explorador en el programa.

Aquesta integració consisteix a afegir una nova implementació de tipus de node, que incorpori tant el servei Bitcoin (perquè pugui ser un node funcionalment independent), com el servei de l'explorador (BTC RPC Explorer).

Primer de tot i abans d'iniciar les modificacions del codi font, cal crear una carpeta on s'emmagatzemen els fitxers que utilitza Docker per instanciar els nodes (*Dockerfile* i *Entrypoint.sh*), i s'ubica en el directori */docker*. Quant a aquests fitxers, *Dockerfile* segueix els mateixos passos que el *Dockerfile* que empra el programa per instanciar nodes Bitcoin, però s'afegeix el codi necessari perquè es descarregui l'explorador, el compili i el configuri perquè es connecti al mateix servei Bitcoin que corre dins el contenidor. D'aquesta manera no depèn d'altres nodes Bitcoin per poder observar la xarxa. Després, en el fitxer *Entrypoint.sh*, s'afegeix el codi per iniciar el servei de l'explorador. Un cop afegit el nou directori amb els fitxers corresponents, s'actualitza el fitxer JSON *nodes.json*, ubicat en el mateix directori que el pas anterior (*/docker*) i s'afegeix la nova implementació, les seves versions, i s'incrementa el comptador de versions totals anomenat *version*.

Polar disposa de diverses interfícies que defineixen les diferents implementacions dels seus nodes ubicades al fitxer *shared.ts*. Per tal d'implementar un nou node, s'ha de començar per afegir la seva interfície. En aquest cas, però, cal fer un pas previ: és necessari modificar la interfície Bitcoin per poder tenir llibertat més endavant de crear noves implementacions amb altres varietats de ports (com és el cas del port 3002 de l'explorador). Això significa haver de crear una primera interfície Bitcoin que cal substituir en la resta d'arxius. Després, s'afegeix la segona interfície *BtcRpcExplorerNode*, que correspon a la nova implementació.

A partir d'aquí i un cop definides les propietats de la interfície, cal definir quines són les constants que defineixen la seva interacció amb Docker, i en definitiva, el comportament del node. És dins el fitxer */src/utils/constants.ts*, on s'indica quina és la comanda de Docker amb la qual s'inicia el servei Bitcoin o quin és el nom de la imatge Docker, entre d'altres.

En aquest punt, cal implementar els mètodes encarregats d'instanciar el node explorador. Primer, cal editar el fitxer */src/utils/networks.ts* i afegir el mètode de creació del node, encarregat d'omplir les propietats definides a la interfície prèviament. Dins aquest fitxer també cal actualitzar el mètode *getOpenPorts*, encarregat de cercar els ports disponibles al sistema, per tal que també cerqui el port web de l'explorador. També cal modificar els fitxers ubicats a */lib/docker* per implementar la interacció entre docker-compose i el nou node explorador.

Un cop definits els mètodes que permeten instanciar el node, tan sols queda fer la crida dels mètodes. Dins */src/lib/store/models*, cal editar el fitxer *network.ts*, on es controla la lògica del programa, per tal d'afegir instàncies del node explorador als escenaris polar. També, es modifica el fitxer *designer.ts* per afegir la funcionalitat d'iniciar el node, arrossegant-lo des del menú principal. Un cop arribat a aquest punt, ja es pot instanciar un node explorador completament funcional, tan sols falta editar els fitxers ubicats a */src/components/designer* per tal d'editar el menú associat als nodes exploradors i afegir la descripció del port del servei web de BTC RPC Explorer.

Com es pot apreciar a la figura 3, el node implementat resultant incorpora tant el servei Bitcoin com el servei de l'explorador. Mitjançant el

TAULA 1: Resum comparatiu dels exploradors

	Abe	Esplora	Iquidus	Mempool	BTC RPC Explorer
Environment	Python	Node.js	Node.js	Node.js	Node.js
DB type	SQL	SQL	NoSQL	SQL	-
Web server	Built-in	Nginx	Built-in	Nginx	Nodejs
Tx main details	✓	✓	✓	✓	✓
Tx Input details	✓	✓	✓	✓	✓
Tx Output details	✓	✓	✓	✓	✓
Segwit support	Mod required	✓	✓	✓	✓
Raw Tx info	✓	✓	✓	✓	✓
Raw script code	✗	✓	✓	✓	✓
Confirmations	✗	✓	✓	✓	✓
Time lock indication	✗	✗	✓	✓	✗
Type of Tx tag	✗	✗	✗	✓	✗
Block details	✓	✓	✓	✓	✓
Unconfirmed Tx	✗	✗	✗	✓	✗
Miner Id	✗	✗	✗	✓	✓
Hash rate	✗	✗	✓	✗	✓
Global Difficulty	✗	✗	✓	✓	✓
Mempool size	✗	✗	✗	✓	✓
Dashboard	✗	✗	✗	✓	✗
Quick search	✓	✓	✓	✓	✓

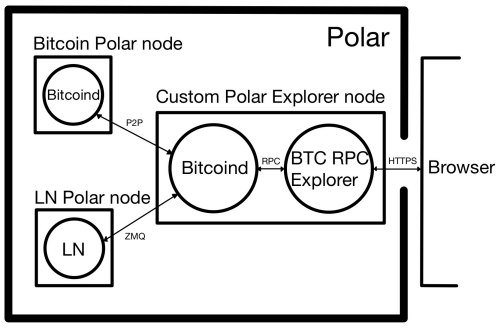

Fig. 3: Arquitectura del node explorador.

protocol Bitcoin, es comunica amb la resta de nodes Bitcoin de l'escenari, i és capaç d'establir connexions a la LN. D'altra banda, l'explorador disposa del servei web al qual es pot accedir des del navegador. Finalment, l'explorador realitza les consultes directament al propi servei Bitcoin que s'executa dins el mateix contenidor.

5 Conjunt de proves realitzades

El conjunt de proves realitzades busca verificar que l'explorador és funcional i estén les capacitats del programa sense intervenir en cap altra de les seves funcionalitats. Les proves es divideixen en dos tipus: proves d'integració amb Polar i les proves funcionals de l'explorador.

5.1 Proves d'integració amb Polar

Les proves d'integració (PI), consisteixen en proves per verificar que la implementació del node està ben integrada amb el programa original i que compleix amb les funcionalitats mínimes. Les proves són les següents:

1. PI.1. Una instància del node funciona independentment d'altres nodes Bitcoin per iniciar una Regtest. Això s'ha de complir ja que ha de ser un dels avantatges derivats d'incloure el servei Bitcoin juntament amb l'explorador dins la implementació.
2. PI.2. El node permet obrir una interfície CLI i realitzar crides RPC, de la mateixa manera que es pot fer amb la resta de nodes.
3. PI.3. El node implementat es pot connectar a les implementacions de nodes Lightning que inclou el programa.
4. PI.4. Una instància del node implementat es connecta correctament a altres nodes Bitcoin.

5.2 Proves funcionals de l'explorador

Les proves funcionals (PF) pretenen verificar que el servei de l'explorador que inclou la implementació respon correctament a les interaccions que tenen lloc a l'escenari. Aquestes són:

1. PF.1. Quan s'inicia un escenari, es mostra el primer bloc (anomenat bloc Génesis).
2. PF.2. L'explorador mostra una transacció que es fa d'un node a un altre.
3. PF.3. L'explorador s'actualitza i mostra l'alçada de bloc correcta quan es mina un bloc de forma automàtica.

4. PF.4. L'explorador s'actualitza després de minar manualment 1000 blocs.

Cadascuna d'aquestes proves s'ha realitzat amb resultats satisfactoris; tanmateix, es mostren els detalls a l'Apèndix A.1., on es pot veure un resum de cadascuna, imatges que les verifiquen, i unes observacions per concloure.

A continuació, s'exposen les figures 4 i 5, que mostren un canal recentment obert amb el seu identificador i la transacció de creació del canal, respectivament. Es pot observar com l'inici i el final de l'identificador del canal coincideix amb el de la transacció, però amb l'endianitat canviada. Es tracta d'una petita mostra de com l'explorador permet observar les transaccions d'obertura de canals.

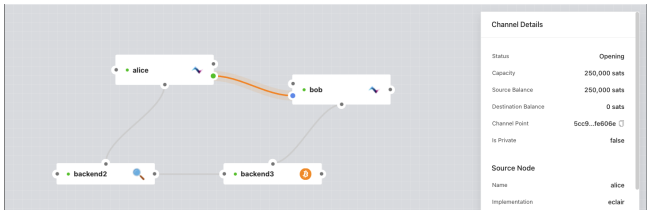

Fig. 4: Polar mostra un canal recenment obert.

Fig. 5: Detalls de la transacció d'obertura del canal, observats en l'explorador.

6 Conclusions

El resultat a destacar de la realització d'aquest projecte és la integració amb èxit de l'explorador BTC RPC Explorer al programa Polar.

Com es pot observar en la secció de proves i en l'annex, l'explorador és funcional i no interfereix en el funcionament del programa. Tan sols li dota d'una eina més que pot ser d'utilitat a l'hora de tenir un instrument que faciliti l'anàlisi de les Regtest dels escenaris amb els quals es treballa, i així, estenent les capacitats interactives que ofereix Polar. Amb això, s'assoleixen els objectius principals definits, incloent-hi l'aportació a un projecte de codi obert.

Altrament, el plantejament inicial de la implementació, consistia en un nou node que tan sols ofereix el servei de l'explorador. Això no obstant, aquesta idea ha sigut modificada en favor de la implementació d'un node Bitcoin adaptat que incorpora un explorador concret, per tal de fer que aquest sigui independent a l'hora de realitzar les consultes a la blockchain.

Quant a possibles línies de continuació del treball, durant el desenvolupament d'aquest han sorgit dues possibles idees. Una primera línia de continuació podria ser afegir una nova interfície de nodes exploradors dins Polar, amb la intenció que es vagin realitzant diferents implementacions, cadascuna amb un explorador diferent. Aquesta proposta afegiria al simulador una varietat d'aplicacions d'exploradors, cadascun amb interfícies diferents i maneres de funcionar peculiars, i amb els que es podria jugar juntament amb altres implementacions de Bitcoin i Lightning dins els escenaris de simulació. Això mateix, porta cap a una possible segona línia de continuació: analitzar de manera detallada i en diferents casos d'ús, el rendiment d'un conjunt d'exploradors de blocs disponibles. La varietat de particularitats del comportament de cadascun i com aquestes determinen el seu rendiment, fan que sigui una segona proposta interessant que estendria la comparativa realitzada en aquest projecte.

Agraïments

Agrair a la meva família i en especial, a la Joana, pel seu suport, i a la doctora Cristina Pérez Solà, tutora d'aquest treball, per la seva constant ajuda i dedicació.

Referències

[1] Andreas M. Antonopoulos. *Mastering Bitcoin*. 2018. url: https://github.com/bitcoinbook/bitcoinbook. (accessed: 01.04.2022).

[2] Andreas M. Antonopoulos. *Mastering the Lightning Network*. 2021. url: https://github.com/lnbook/lnbook. (accessed: 01.04.2022).

[3] Jamal James. *Polar: One-click Bitcoin Lightning Networks for local app development testing*. url: https://github.com/jamaljsr/polar. (accessed: 27.03.2022).

[4] Harry Kalodner et al. *BlockSci: Design and applications of a blockchain analysis platform*. USENIX Association, ag. de 2020, pàg. 2721-2738. isbn: 978-1-939133-17-5. url: https://www.usenix.org/conference/usenixsecurity20/presentation/kalodner.

[5] Jeffrey Nickoloff i Stephen Kuenzli. *Docker in action*. Simon i Schuster, 2019.

[6] Muhammad Ovais Ahmad, Jouni Markkula i Markku Oivo. *Kanban in software development: A systematic literature review*. 2013.

[7] Bitcoin. *Testing applications*. url: https://developer.bitcoin.org/examples/testing.html. (accessed: 13.05.2022).

[8] *Overview of Docker Compose*. url: https://docs.docker.com/compose/. (accessed: 10.05.2022).

[9] Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016. Cap. 2, pàg. 58.

[10] *Blockexplorer.com*. url: https://en.bitcoin.it/wiki/BlockExplorer.com. (accessed: 03.04.2022).

[11] Bitcoin-Abe. *Bitcoin-abe/Bitcoin-Abe: Abe: Block browser for Bitcoin and similar currencies*. url: https://github.com/bitcoin-abe/bitcoin-abe. (accessed: 04.05.2022).

[12] Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016. Cap. 5, pàg. 131-132.

[13] Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016. Cap. 1, pàg. 32-36.

[14] Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016. Cap. 1, pàg. 32-36.

[15] Timelock. *Timelock - Bitcoin Wiki*. url: https://en.bitcoin.it/wiki/Timelock. (accessed: 04.05.2022).

[16] Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016. Cap. 3, pàg. 79-88.

[17] BitcoinWiki. *Block weight*. url: https://en.bitcoinwiki.org/wiki/Block_weight. (accessed: 06.05.2022).

[18] BitcoinWiki. *Hashrate*. url: https://en.bitcoinwiki.org/wiki/Hashrate. (accessed: 06.05.2022).

[19] Bybit Learn. *Bitcoin Mempool: What happens to the unconfirmed transactions?* url: https://learn.bybit.com/blockchain/what-is-bitcoin-mempool. (accessed: 06.05.2022).

[20] Andreas M. Antonopoulos. *Mastering Bitcoin*. 2018. Cap. 8. url: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc#mining. (accessed: 01.04.2022).

[21] Andrew D Birrell i Bruce Jay Nelson. "Implementing remote procedure calls". A: *ACM Transactions on Computer Systems (TOCS)* 2.1 (1984).

[22] Bitcoin. *Bips/BIP-0141.mediawiki at master · bitcoin/bips*. url: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki. (accessed: 07.05.2022).

[23] Mario Schlipf. *MARIOSCHLIPF/Bitcoin-Abe: Abe: Block browser for Bitcoin and similar currencies*. url: https://github.com/marioschlipf/bitcoin-abe. (accessed: 06.05.2022).

[24] Dan Janosik. *Janoside/BTC-RPC-Explorer: Database-free, self-hosted Bitcoin explorer, via RPC to Bitcoin Core*. url: https://github.com/janoside/btc-rpc-explorer. (accessed: 01.05.2022).

[25] BitcoinWiki. *API reference (JSON-RPC)*. url: https://en.bitcoin.it/wiki/API_reference_(JSON-RPC). (accessed: 07.05.2022).

[26] Blockstream. *Blockstream/esplora: Explorer for bitcoin and liquid*. url: https://github.com/Blockstream/esplora. (accessed: 06.05.2022).

[27] Blockstream. *Esplora/api.md at master · Blockstream/Esplora*. url: https://github.com/Blockstream/esplora/blob/master/API.md. (accessed: 06.05.2022).

[28] Mempool. *Mempool/Mempool: An open-source Explorer developed for the Bitcoin community, focusing on the emerging transaction fee market to help our transition into a multi-layer ecosystem.* url: https://github.com/mempool/mempool. (accessed: 06.05.2022).

[29] BitcoinWiki. *Replace by fee*. url: https://en.bitcoin.it/wiki/Replace_by_fee. (accessed: 07.05.2022).

[30] BitcoinWiki. *Multi-signature - Bitcoin Wiki*. url: https://en.bitcoin.it/wiki/Multi-signature. (accessed: 07.05.2022).

[31] Electrum. *Welcome to the Electrum Documentation!* url: https://electrum.readthedocs.io/en/latest/. (accessed: 06.05.2022).

[32] Iquidus. *Iquidus/explorer: An open source block explorer*. url: https://github.com/iquidus/explorer. (accessed: 06.05.2022).

Apèndix

A.1 Detall de les proves realitzades

En aquesta secció, es mostra un compendi de les proves realitzades en forma de taula (veure taula 2), juntament amb un conjunt de captures que mostren els seus resultats i verifiquen que es compleixen de forma satisfactòria. En últim lloc, també es presenten unes observacions sobre la interfície de l'explorador, relatives a on es troben les transaccions.

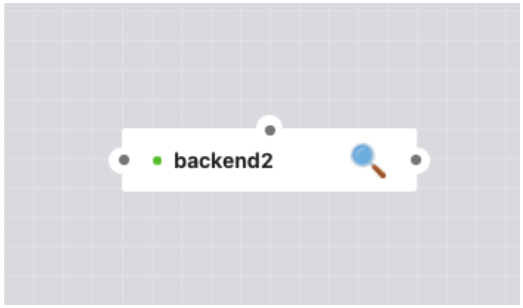

Fig. 6: PI.1. La implementació és independent.

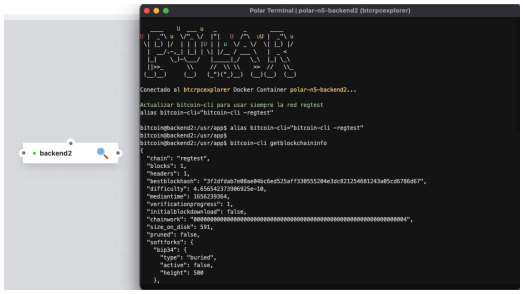

Fig. 7: PI.2. CLI de la implementació en ús.

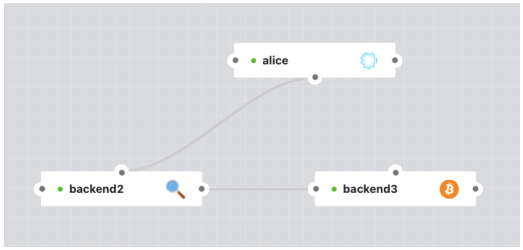

Fig. 8: PI.3. Connexió amb un node C-Lightning.

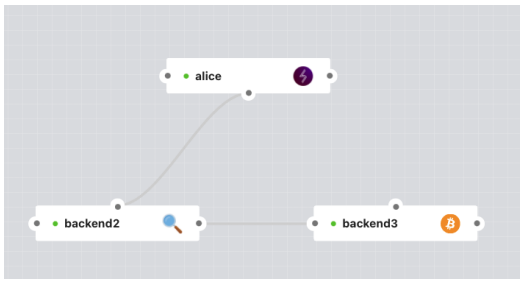

Fig. 9: PI.3. Connexió amb un node LND.

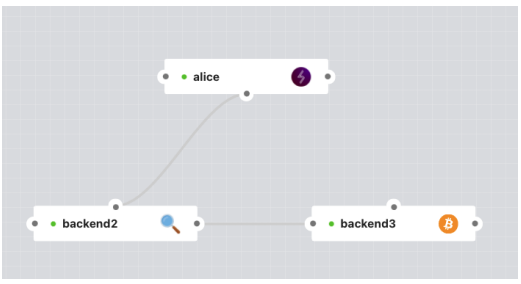

Fig. 10: PI.3. Connexió amb un node Eclair.

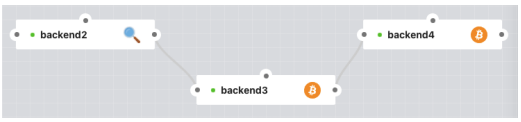

Fig. 11: PI.4. Connexió amb un node Bitcoin.

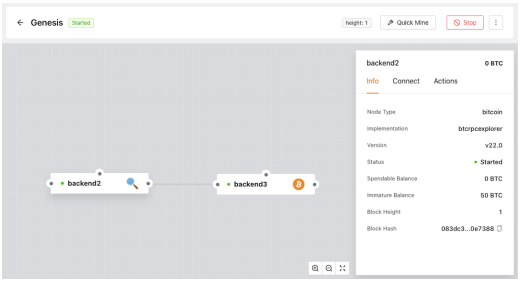

Fig. 12: PE.1. Interfície Polar durant l'inici de l'escenari.

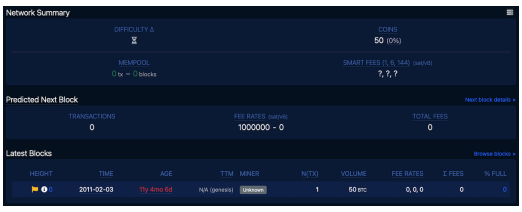

Fig. 13: PE.1. Interfície de l'explorador mostrant el Bloc génesis.

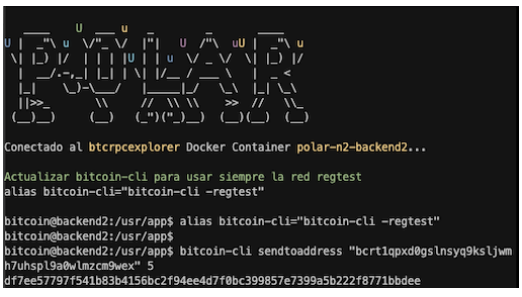

Fig. 14: PE.2. Es genera una transacció desde la línia de comandes.

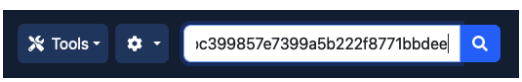

Fig. 15: PE.2. Es cerca la transacció en l'explorador.

TAULA 2: Comprovació de les proves de funcionament del node implementat

Prova	Resultat
PI.1: La instància del node implementat és independent d'altres nodes Bitcoin.	✓
PI.2: El node disposa d'interfície CLI i permet realitzar crides RPC	✓
PI.3: El node implementat permet la connexió amb els diferents nodes Lightning.	✓
PI.4: El node implementat es connecta a altres nodes Bitcoin.	✓
PF.1: Al iniciar un escenari, es mostra el bloc génesis a l'explorador.	✓
PF.2: L'explorador mostra les transaccións de l'escenari	✓
PF.3: L'explorador mostra l'alçada de bloc correcta en el moment d'actualitzar-se.	✓
PF.4: L'explorador presenta la informació correctament després de minar manualment 1000 blocs.	✓

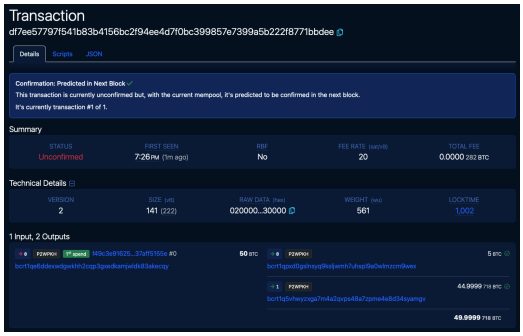

Fig. 16: PE.2. Detalls de la transacció dins la interfície de l'explorador.

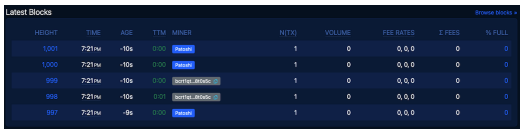

Fig. 17: PE.3. i PE.4. Interfície Polar després d'haver minat manualment 1000 blocs.

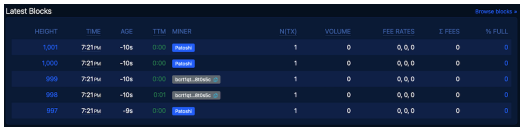

Fig. 18: PE.4. Interfície Polar mostrant els detalls del bloc 1001.

A.1.1 Observacions

En relació a les capacitats que té l'explorador per mostrar les transaccions que tenen lloc a la xarxa de proves, s'ha pogut veure com efectivament es poden localitzar i observar els detalls (veure la figura 16). Tanmateix, per tal d'accedir als detalls de les transaccions, primer cal trobar el bloc que les conté o bé, localitzar-les cercant mitjançant el seu identificador. La pàgina principal de BTC RPC Explorer està orientada a mostrar un compendi dels últims blocs, però de cara a l'estudi d'escenaris que interactuen amb la LN, com és el cas de Polar, resulta més útil disposar d'un llistat de les últimes transaccions generades a la xarxa, per tal de localitzar fàcilment les transaccions involucrades en l'apertura i tancament de canals.