Universidad del País Vasco
Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Github link (Hemen klikatu eta lana gustatu bazaizue izar bat eman mesedez :D)

WhatWeb aplikazioaren dokumentazioa

Egileak:
Alvaro Hernandez eta Asier Rosa

2020ko Abendua

Aurkibidea

1 Proiektuaren helburuen dokumentua

1.1 Sarrera

Proiektu hau 2020-2021 kurtsoko Informazio Sistemen Antolakuntzaren Diseinua izeneko irakasgaian dago kokatua eta bere proposamena WhatWeb izeneko aplikazioa erabiliz webguneen informazioaren eskaneoa eta informazio horren txertaketa datu base batean egitea da. Proiektu hau 2020ko irailaren 28an hasi genuen eta bere epe muga abenduak 14 da.
Proiektu hau aurrera eramateko ikasturte honetan jasotako ezagutzak aplikatu dira (interfaze grafikoak, aplikazioekin loturak, etab.).
Gure programa ordenagailuetan erabiltzeko sortu da, hiru sistema operatibo nagusientzat: Linux, Windows eta MacOS.

```
arrosa@asipc:~$ whatweb https://ikasten.io
https://ikasten.io [200 OK] Apache[2.4.41], Country[UKRAINE][UA], Email[admin@domain.admi,guest@where.eve
r], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[95.216.157.127], JQuery, MetaGenerator[Wo
rdPress 5.5.3], Open-Graph-Protocol[website], Script[text/javascript], Title[Ikasten.IO &#8211; Learning,
 Aprendiendo], UncommonHeaders[link], WordPress[5.5.3]
arrosa@asipc:~$
```

Figure 1: whatweb aplikazioa terminaletik exekutatzerakoan sortzen den emaitzaren adibidea

1.2 Proiektuaren deskribapena

- Inplementatuko ditugun funtzionalitate nagusiak:

  Lehenik eta behin erabiltzaileak WhatWeb izeneko aplikazioa era erosoago batean erabili dezan interfaze grafiko bat sortuko dugu, bezeroak nahi duen webgunearen informazioa lortzeko.
  Gure programaren kasuan, bilduko den informazioa webguneak erabiltzen duen CMSa, honen bertsioa eta baita webguneak zein web zerbitzari erabiltzen duen izango da. Gainera, datu base bat erabiliko dugu, eskuratzen den informazio guztia biltegiratzeko. Honi esker erabiltzaileak bere azken bilaketen buruzko informazioa automatikoki eta modu antolatuan gordeta eduki ahal izango du, eta honen bidez, webgune hauei buruz behar duen informazioa taulen bidez era erraz eta dotore batean ikusi.

  CMS menuan filtro bat gehitu dugu, webguneak izenaren arabera eta baita haien CMSen arabera bilatu ahal izateko.

- Inplementatuko **EZ** ditugun funtzionalitate nagusiak:

  Datu basea SQLite motakoa izango da. Ez dugu NoSQL datubase sistema inplementatu (MongoDB), sistema honek zerbitzari bat izatea

suposatzen duelako, eta SQLite erabiliz, fitxategi bakar batean beharrezko informazio guztia gorde dezakegu.

1.3 Proiektuaren arkitektura

- Gure WhatWebFX programaren arkitektura lokala izango da. Izan ere biltzen diren datu guztiak gure ordenagailuan aurkitzen den datubase batean gordeko dira. Hala ere, zerbitzari baten menpe funtzionatuko du, whatweb aplikazioak zerbitzari batekin egiten baitu l eta ezinbestekoa izango delako aplikazioaren erabilerarako.

- Gure aplikazioaren funtzionamendua modu erraz batean azaldu ahal izateko eskema bat egin dugu:

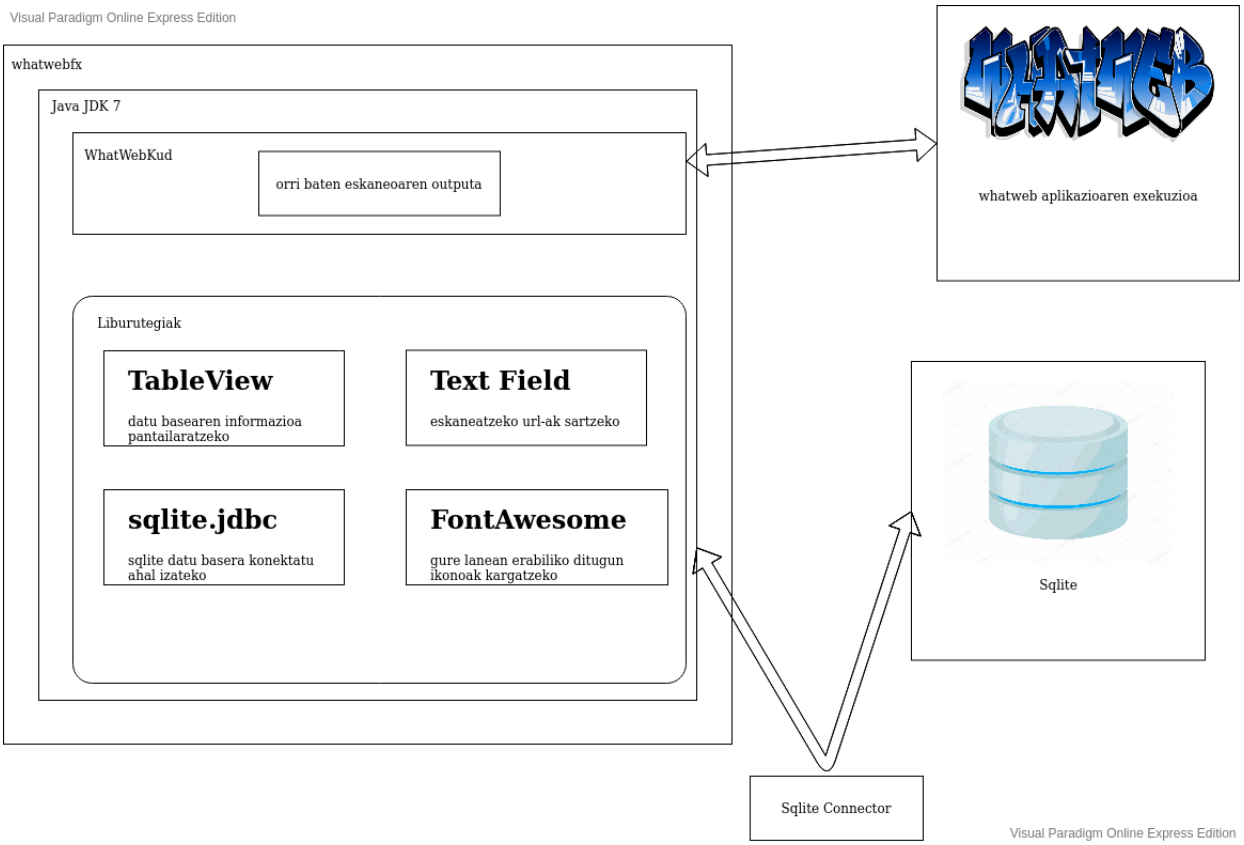

Figure 2: whatweb aplikazioaren funtzionamendua era erraz batean azaltzen duen eskema

Eskeman ikusten den moduan, gure aplikazioak hiru bloke nagusi izango ditu. Alde batetik Java blokea dugu. Bloke honetan aplikazioa programatu dugu eta honi esker funtzionatuko dute beste elementu guztiek. Esan daiteke hau dela aplikazioaren motorea.

Honekin batera whatweb aplikazioaren zerbitzaria dugu, zeinetik web orri guztien eskaneoak egingo ditugu **(link-a)**. Azkenik datu basea daukagu, SQLite motakoa, zerbitzaririk behar ez duena, eta datu guztiak fitxategi bakar batean gordetzeko ahalmena duena.

Lehen aipatu dugun moduan, gure programaren funtzio nagusia webgune ezberdinen informazioa lortzea da. Horregatitik ezinbestekoa da whatweb aplikazioa. Behin eskaneoa eginda, eta heltzen den informaziotik interesatzen zaiguna identifikatu ostean, SQLite fitxategi batean gordeko da. Horretarako sqlite.jdbc driver-a erabili da.

Java blokean interfaze grafikorako erabilitako liburutegi eta aplikazioak aipatu behar dira. Alde batetik SceneBuilder, non aplikazioko interfaze grafikorako FXML fitxategiak sortu diren (TableView, TextField, Botton), eta bestetik FontAwesome, interfazean ikonoak sartzeko erabili duguna, ukitu pertsonala eskeintzeko.

Programak funtzionatu ahal izateko sortuko diren klaseak hiru bloke ezberdinetan banatu daitezke. Alde batetik, datu basean aldaketak egingo dituzten klaseak. Hauek programa eta datu basea konektatuko dituzte, eta aldaketak kudeatzeaz arduratuko dira. Bestetik, interfaze grafikoen kudeatzaileak egongo dira eta erabiltzailearen eta programaren arteko harremana kudeatuko dute. Gainera 'model' izeneko motako klaseak ditugu, eta hauek informazioa programan zehar garraiatzen lagunduko digute. Bukatzeko Main eta Config klaseak ditugu. Config klasea sistemaren informazioa gordeko du, hala nola, datu basearen kokapena eta datu basean gorde nahi dugun informazioaren denbora mugatuko (/tmp/) fitxategiak.

Hau guztia kontuan izanda MVC ("Model View Controller) patroia erabili dela esan dezakegu. Antolaketa honi esker arazoak arinago aurkitu eta konpondu daitezke eta nola ez, etorkizuneko mantentze lanak erraztu.

Aurretik azaldutakoa horrela laburtu daiteke:

- Aurkezpena: Goian dagoen azala da, erabiltzaileak ikusiko duena, Interfaze Grafikoa.
- Aplikazio maila: Bertan negozio logika eta interfazea lotzeaz arduratuko diren klaseak egongo dira sartuta. Beraz, kontrolatzaileak eta XML artxiboak sartzen dira hemen.

– Negozio logikoa: Javaz egindako programa da, aplikazio atzean dagoena. Ezinbestekoa da programaren exekuzioak funtziona dezan.
– Datuak: Programak datu hauekin egindo du lan. Proiektu honetan erabiliko diren datuak SQLite motako fitxategi bakar batean gordeko dira.

1.4 Plangintza

- Hasiera:

  Lehenik eta behin, proiektua hasteko orduan pausu garrantzitsu batzuk jarraitu beharko dira:

  - Lehenik proiektuak zehar erabiliko ditugun tresnak prestatuko ditugu:
    - Hardware:
      Lanean ibiliko diren taldekide guztiek ordenagailu bat beharko dute, eta internet konexioa whatweb aplikazioarekin frograk egin ahal izateko.
    - Software:
      - IntelliJ:
        Javan programatzeko erabiliko dugun ingurunea izango da. Gradle-i esker beharrezko dependentzia guztiak kudeatuko ditugu era eroso eta eraginkorrean. Oso tresna ahaltsua da, eta beste aplikazioekin loturak sortzeko aukera onena.
      - Scene Builder:
        FXML-ak sortzeko erabiliko dugu. Tresna honi esker gure interfaze grafikoa sortzea lortuko dugu. CSS-ak erabili ahal izango ditugu, taulak sortu, botoiak, textua sartzeko label-ak, combobox-ak...
      - DB Browser for SQLite:
        Tresna hau oso erabilgarria izango da gure datu basea aztertzeko. Modu grafikoan ikusi ahalko ditugu taulak, haien egitura... Aldaketak egitea erraza da eta SQL aginduak frogatzeko aukera izango dugu, emaitzak aztertu eta erroreak izanez gero, azkar aurkitzeko.
  - Ondoren beste pausu garrantzitsu bat whatweb aplikazioa instalatzea izango da. Izan ere, gure programa aplikazio honetan oinarrituko da. Horretarako, erabilitako sistema operatiboaren arabera hainbat metodo egongo dira. Erabiltzailea laguntzeko jarraitu beharreko pausuak Github-eko errepositorioko Readme fitxategian azalduko zaizkio.
  - Lanean hasi

- Domeinu Eredua: Lehenik eta behin gure aplikazioaren datu basea ondo kudeatzeko bere domeinuaren eredua egingo dugu. Oso garrantzitsua da SQLitek nola funtzionatzen duen ulertzea, ondoren hau erabiliko baitugu aplikazioak erabiliko dituen datu guztiak maneiatzeko.

- Sekuentzia diagrama: Hasieran programaren funtzionamenduaren ideia izateko metodo nagusien nondik norakoak pentsatu eta diagrama baten adierazten dira.

- Interfaze grafikoekin lan egiten hasi: Ondoren gure aplikazioa izango duen interfaze grafikoa garatu egingo dugu, horretarako scenebuilder, css orriak eta FontAwesome-ko ikonoak erabiltzea oso komenigarria da.

- Gure Interfaze grafikoa gure datu basearekin erlazionatu: Gure interfaze grafikoan TableView-ak erabiliko ditugu datu basearen balioak pantailaratzeko, beraz honekin lan egingo dugu, datu basean query-ak eginez nahi ditugun balioak lortzeko.

- scan-ak egitea: scan-ak egiteko whatweb komandoa Javan nola exekutatzen den jakin behar izango dugu. Badakigu whatweb-ek nola funtzionatzen duen terminaleak, baina hori Javatik egingo dugu orain. Whatwebek eskaintzen dizkigun komando ezberdinak ezagutuko ditugu eta haiekin lan egin.

- CMS menuko aukeran filtroa egin: Behin aurreko funtzionalitate guztiak ondo funtzionatzen dutela, filtroa inplementatzen da. Hau egitea nahiko zaila denez ia azken pausuan jarri dugu. Izan ere, funtzio honek ez du baliorik beste guztiak kale egiten badu.

- Instalatzailea: instalatzailea sortu eta sistema operatibo guztietan funtzionatzen duela konprobatu.

2 Domeinuaren eredua:

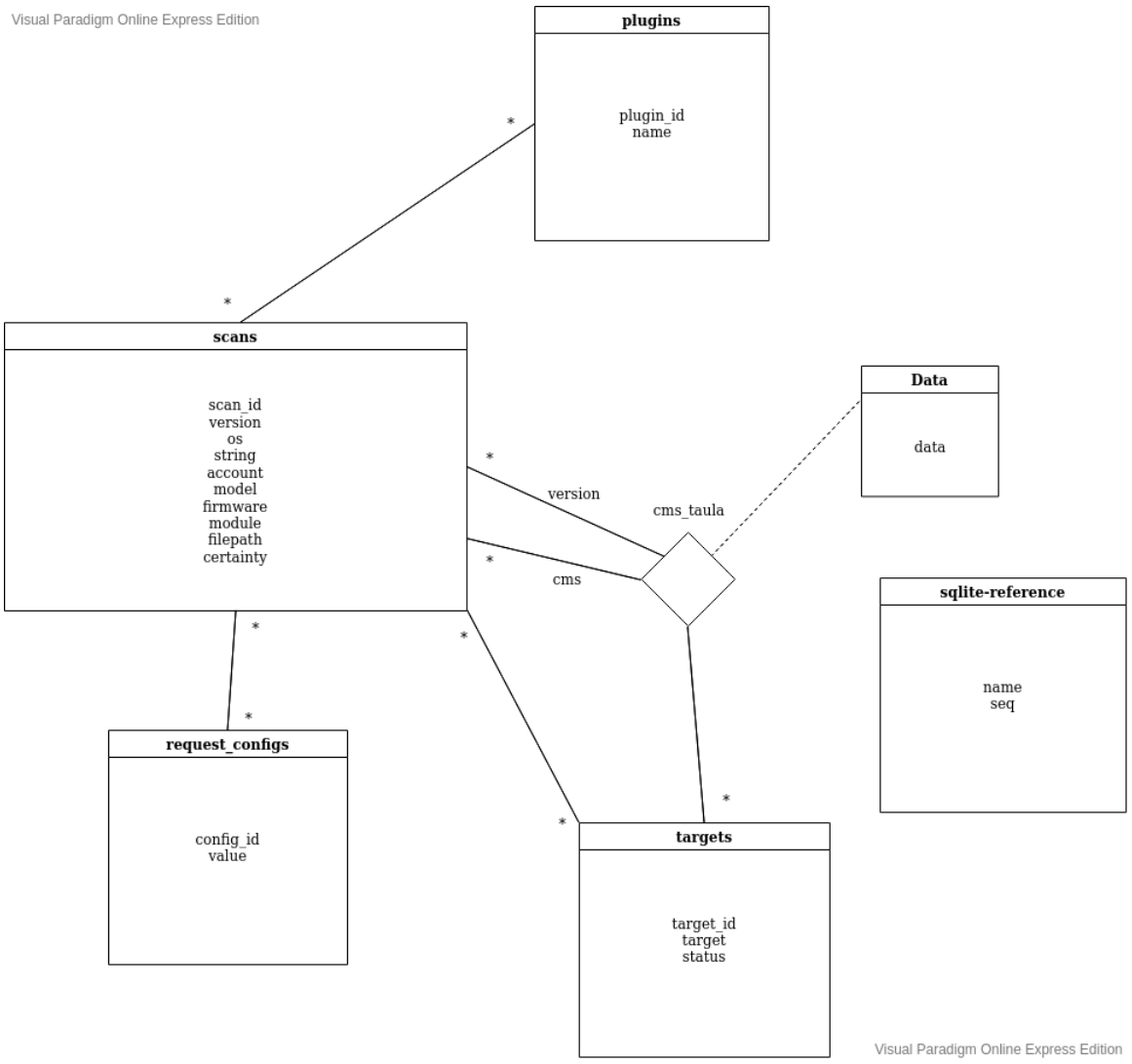

Figure 3: whatwebfx erabiltzen duen datu basearen egitura

- Ikusi dezakegun bezala gure domeinuaren ereduan hainbat elkartze-asozazio aurki ditzazkegu eta honekin batera, erlazio laukoitz bat.
  Datu basea ulertzeko lehenik whatweb-ek egiten duena ulertu behar da. URL bat eskanetzen denean taula ezberdinetan gordetzen dira hainbat datu. 'Scan' bakoitzak bere identifikatzaile propioa izango du, beti desberdina, eta modu honetan erraza izango da eskaneoak bereiztea. Informazio asko jaso egiten du eta horretarako webgune bakoitzean 'plugin' ezberdinak erabiltzen ditu, eta hauek gordeta geratzen dira baita ere. Azkenik, eskaneatzen den bakoitzean 'target' taulan helburua eta egoera gordetzen da.

- Datuen sailkapena hobetzeko "`cms_taula`" deituriko taula laguntzailea erabiltzen dugu. Bertan webgune bakoitzaren CMS-a, bertsioa eta azkenengo eguneraketa gordetzen ditugu, hori izango delako erabiltzaileak begiratuko duena.

- Honekin batera, kontsulta ezberdinak eginez webgune bakoitzak erabiltzen duen web zerbitzariaren eta honen bertsioaren informazioa eskeini ahal dizkiogu erabiltzaileari.

3 Sekuentzia Diagrama:

Sekuentzia diagrama handiegia denez dokumentuan sartzeko, mesedez **link** honetan sartu ikusi ahal izateko.

3.1 GERTAERA FLUXUA

- Lehenik eta behin sekuentzia diagraman whatweb menuan (UI1) egiten diren eskaneoak azaltzen dira. Ikusi daiteken moduan, erabiltzaileak eskaneatu nahi duen URL-a sartu egiten du, eta ondoren "Scan" botoiari ematen dio. Hau egitean WhatWebKud izeneko klasean dagoen sqliteKargatu metodoari deitu egiten zaio eta metodo honek WhatWebDBKud klasean dagoen urlDatuBaseanSartu(String url) metodoari deitu egiten dio. Honela url-a datu basean sartzea da, eta baldintza batzuen arabera lehen aldiz sartu edo datuak eguneratzen dira, orain azaltzen den moduan.

  Honen ondoren, WhatWeb klasean dagoen datuBaseaEguneratu() metodoari deia egiten zaio. Hemen datu baseak jada url hori badu, datu basean target horren datu guztiak eguneratzen dira, eta bestela url hori oraindik datu basean sartuta ez badago, datu basean sartu egiten da.

- Ondoren, server botoian sakatzean gertatzen dena azaltzen da. Hemen, ServerKud klasean dagoen aktualizatuLista() metodoari deitu egiten zaio. Lehenik ServerDBKud klasean dauden targetakLortu() eta serverLortu(targetak) metodoei deitu egiten zaie. Lehenengoak url-en lista bat bueltatzen du, eta besteak url lista horren url bakoitzaren web zerbitzaria (adib: https://ikasten.io —- WordPress 5.5.3).

- Azkenik, cms botoian sakatzean gertatzen dena azaltzen da. Kasu honetan, CMSKud klaseko urlSartu() metodoari dei bat egiten zaio. Barnean CMSDBKud klaseko cmsListaLortu() metodoari deitu egiten zaio, eta honek eskaneatutako url guztien (cms-taula laguntzailetik) target-a, bertsioa, cms-a eta LastUpdated-a bueltatzen ditu (adib: https://ikasten.io —- 5.5.3 —- WordPress 5.5.3 —- 2020-12-08).

3.2 SQL aginduak

- Query 1 = SELECT target FROM targets t, scans s WHERE scan-id=(SELECT MAX(scan-id) FROM scans) AND s.target-id=t.target-id
- Query 2 = SELECT DISTINCT target FROM cms-taula
- Query 3= UPDATE cms-taula SET lastUpdated = data, cms = 'cms', version = 'version' WHERE target LIKE ''
- Query 4= UPDATE cms-taula SET lastUpdated = data, cms = 'unkown', version = '0' WHERE target LIKE ''
- Query 5= INSERT INTO cms-taula (target, version, cms, lastUpdated) VALUES('target', 'version', 'cms', 'data')
- Query 6= INSERT INTO cms-taula (target, version, cms, lastUpdated) VALUES('target', '0', 'unknown', 'data')
- Query 7= SELECT DISTINCT target FROM targets, scans ORDER BY scan-id
- Query 8= SELECT DISTINCT target, string, MAX(s.scan-id) AS id FROM targets t, scans s WHERE t.target-id=s.target-id AND s.plugin-id=268 AND t.target LIKE ''
- Query 9= SELECT target, version, cms, lastUpdated FROM cms-taula ORDER BY lastUpdated DESC;
- lortuCMSAtributuak = SELECT DISTINCT target, s.version, c.string, MAX(s.scan-id) as id FROM scans s INNER JOIN targets t ON s.target-id=t.target-id INNER JOIN scans c ON t.target-id=c.target-id WHERE (c.plugin-id=192) AND (s.plugin-id=1152 OR s.plugin-id=132 OR s.plugin-id=337 OR s.plugin-id=72 OR s.plugin-id=241 OR s.plugin-id=283 OR s.plugin-id=315 OR s.plugin-id=1129 OR s.plugin-id=1419 OR s.plugin-id=131 OR s.plugin-id=88 OR s.plugin-id=824 OR s.plugin-id=822) AND t.target = 'target' ORDER BY s.scan-id

  *data = new SimpleDataFormat("yyyy-MM-dd").format(new Date())

4 Inplementazioa:

Gure aplikazioaren helburua erabiltzaileari whatweb aplikazioaren bidezko webguneen eskaneoak egitea eta emaitzen datuen kudeaketa erraza eskeintzea da. Horretarako aplikazioan, aukera ezberdinak 3 leihotan ikusi ditzazkegu:

- WhatWeb:
  Hau leiho nagusia izango da. Hemen erabiltzaileak nahi duen url-aren eskaneoa egin ahalko du era erraz eta intuitibo batean. Gainera terminaletik agertuko litzateken testua ikusi daiteke. Eskaneo bat egiten den bitartean beste funtzio batzuk egin daitezke, aplikazioa ez baita blokeatzen, harien erabilera dela eta.
  Whatweb funtzionalitatea erabiltzeko erabiltzaileak baldintza batzuk bete behar ditu, batez ere erabiltzen duen sistema eragilearen arabera. Eskaneoak egiteko whatweb aplikazioa instalatuta egon behar du. Horregaitik bereizketak egin behar dira kodean, adibidez Windows SE kasuan komandoe 'wsl' gehituz.
  Datu basearen sorketarekin arazo batzuk agertu dira. Whatweb-en bertsioaren arabera aplikazioak berak '–log-sql-create' komandoaren bitartez eskeintzen duen datu baseak plugin identifikatzaile ezberdinak ematen ditu. Hori dela eta bertsio bateko datu basea oinarritzat hartu dugu eta bezeroari datu base hori ematen diogu, SQL aginduekin arazoak sortu ez daitezen. Datu base hori GitHub-en dago eskuragarri, eta kodeak automatikoki instalatzen dio erabiltzaileari.

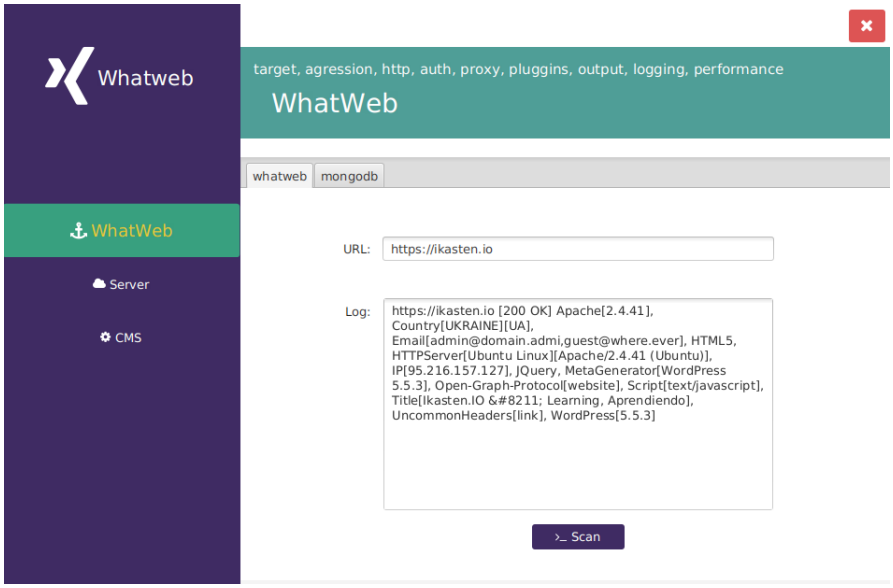

Figure 4: WhatWeb izeneko menuan URL bat eskaneatzean pantailaratzen dena

- Server:
  Leiho hau erabiltzailearen bilaketen historiko gisa funtzionatuko du, modu honetan erabiltzaileak era erraz batean bere azken bilaketak ikusi ditzazke. Gainera eskaneatutako webguneek erabiltzen duten web zerbitzaria ikusi dezake. Hau lortzeko SQL agindu bakar batekin egin daiteke. Gero bueltatzen den lista TableView-an sartu eta bezeroari informazioa era dotore baten aurkezten zaio.

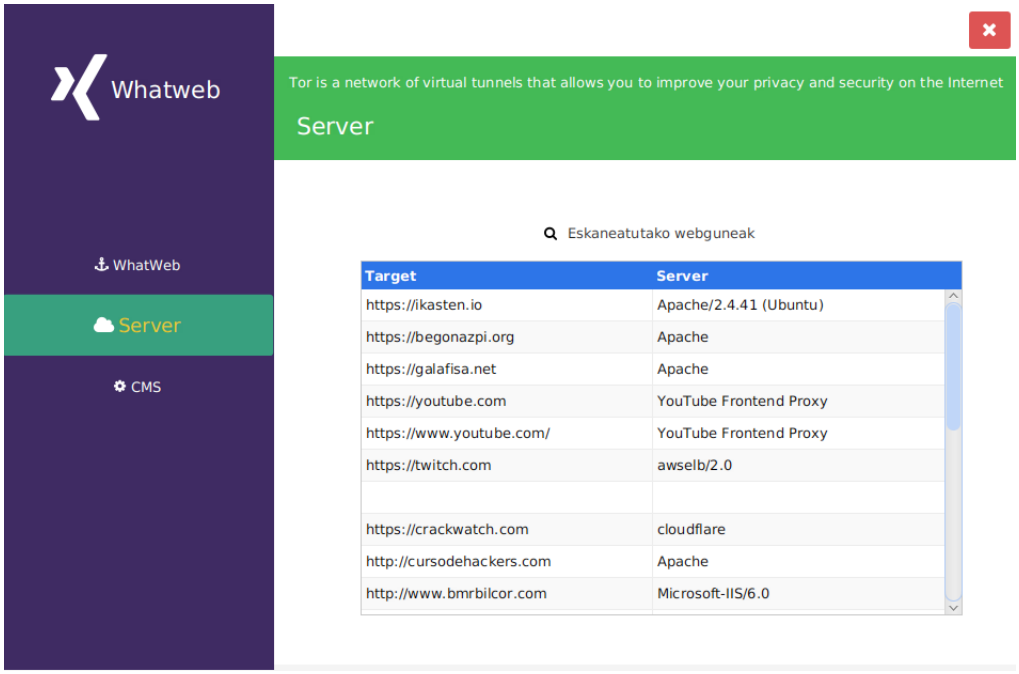

Figure 5: Server menuko aukeran eskaneatutako url-en historikoaren informazioa eta web zerbitzariak

- CMS: Leiho hau erabiltzailearen bilaketa bakoitzaren informazioa era errazean pantailaratzeko egin da, modu honetan erabiltzaileak era erraz batean bilatzen duen url-aren informazio nagusia ikusi dezake. Hau inplementatu ahal izateko datu basean taula laguntzaile bat sortu behar izan da, batez ere interesatzen zaizkigun datuak sartzeko eta eguneratzeko, eta baita azken eguneraketaren dataren balorea sartzeko. Gainera filtro bat eskeintzen da, izenak filtratzeko, eta baita combobox bat, webguneak erabiltzen duen CMS-aren arabera filtratzeko.

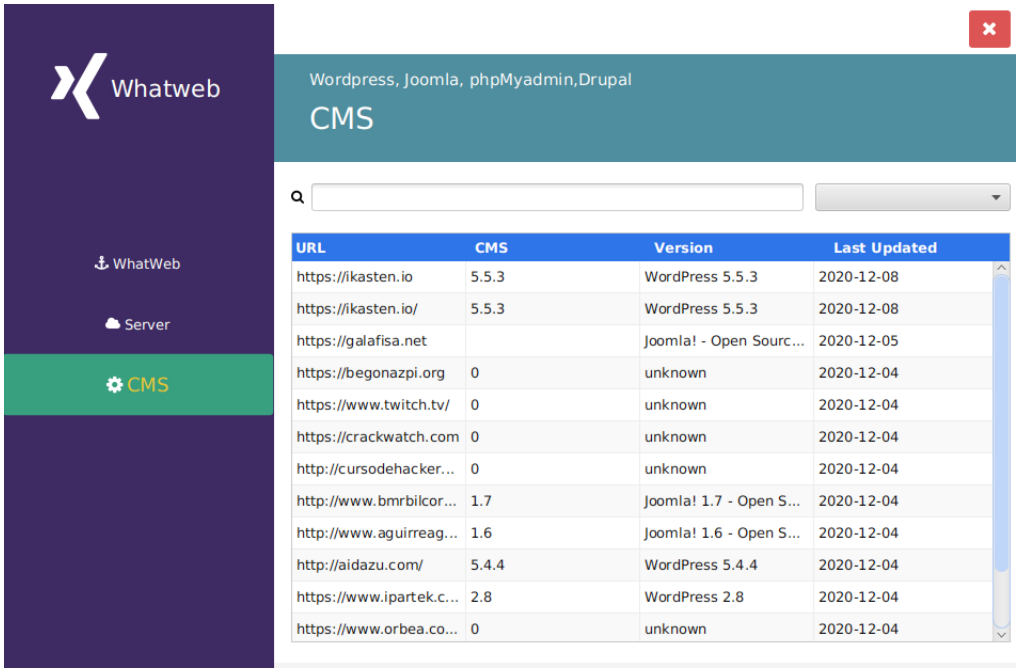

Figure 6: CMS menuko aukeran eskaneatutako url-en informazioa

5 Bibliografia:

[1] **Whatweb Github Repository**

[2] **The Badass JLinkPlugin** jpackage tool introduced in Java 14.

[3] **Java FX Filter table view**

[4] **Threads in Java**

[5] **Descripción de capas lógicas**

[6] **FXML eta Controller-ak**

[7] **Threads, SQLite, StackPaneURLa**

[8] **jlink eta jpackage: instalatzaileak sortzen**

[9] **Github actions**

[10] **Process APIa nola erabili**

[11] **(WSL) Run Linux tools from a Windows command line**

[12] **SceneBuilder + JavaFX + UI Design**