

EUSKAL HERRIKO UNIBERTSITATEA



INFORMAZIO SISTEMEN ANALISIA ETA DISEINUA

WhatWebJM

Egileak:

Jon Ander Asua, Ena Verhorst eta Ander San Juan

<https://github.com/UPV-EHU-Bilbao/WhatwebJM>

2020

Gaien Aurkibidea

Irudien aurkibidea	1
1 Proiektuaren helburuen dokumentua	2
1.1 Sarrera	2
1.2 Deskribapena	2
1.2.1 Implementatutako funtzionalitateak	3
1.2.2 Ez Implementatutako funtzionalitateak	3
1.3 Plangintza	3
2 Analisia	5
3 Diseinua	6
3.1 Sekuentzia Diagrama	6
4 Implementazioa	7
4.1 Egin Beharrekoak	7
4.2 Ekarpen Berriak	8
4.3 Etorkizunerako planak eta etengabeko hobekuntza	9
5 Bibliografia	10

Irudien aurkibidea

1	Komando lerrotik WhatWeb-en itxura	2
2	SQLite	4
3	SQL	4
4	SceneBuilder	4
5	Domeinuaren eredua	5
6	Sekuentzia diagrama	6

1 Proiektuaren helburuen dokumentua

1.1 Sarrera

Ikasturtearen hasieran *Informazio Sistemen Analisia eta Diseinua* ikasgaiko proiektuko enuntziatua(1) jaso genuen, WhatWeb(2) aplikazioa erabiltzeko interfaze grafiko bat sortzea.

```
C:\Users\Ander>wsl whatweb ikasten.io
http://ikasten.io [301 Moved Permanently] Apache[2.4.41], Country[UKRAINE][UA], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[95.216.157.127], RedirectLocation[https://ikasten.io/], Title[301 Moved Permanently]
https://ikasten.io/ [200 OK] Apache[2.4.41], Country[UKRAINE][UA], Email[admin@domain.admi.guest@where.ever], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[95.216.157.127], JQuery, MetaGenerator[WordPress 5.5.3], Open-Graph-Protocol[website], Script[text/javascript], Title[Ikasten.IO &#8211; Learning, Aprendiend o], UncommonHeaders[link], WordPress[5.5.3]
```

Irudia 1: Komando lerrotik WhatWeb-en itxura

Aplikazioak hurrengo puntuak bete behar ditu.

- WhatWeb-en bidez analizatutako webgunearen informazioa bistaratzeko aukera.
 - Webguneak erabiltzen duen *CMS*(3) eta honen bertsioa.
 - Webguneak erabiltzen duen zerbitzari mota eta honen bertsioa
 - Webgunea eta webgunearen *screenshot*-a irekitzeko aukera
- Testu-eremu baten bidez webgune bat analizatzeko aukera.
- Hainbat sistema eragileetan erabil daitekeen aplikazioa sortzea.
- Sortutako aplikazioa software libre bezala banatzea eta honen kodea GitHub-en eskuragarri egotea.

1.2 Deskribapena

Aplikazio honekin lortu nahi duguna, web orrialdeen informazioa da. Hau da, web orrialde bakoitzak erabiltzen dituen aplikazio eta baliabideak, adibidez;

- Zerbitzari mota (internetara konektatzeko erabiltzen duen programa)
- CMS (Content Management System) mota (web orrialdea sortu eta kudeatzeko tresna) eta bertsioa
- Ip helbidea

1.2.1 Implementatutako funtzionalitateak

Aplikazio honetan inplementatu ditugun funtzionalitate nagusiak hurrengoak dira:

- Erabiltzaileak sartutako URL-aren informazioa bilatzea eta datu basean txertatzea:
Hau egiteko, URL-a hartu eta WhatWeb aplikazioa erabiliz, terminalean exekutatu eta informazioa lortu. Behin informazio hori lortuta Datu Basean gorde dugu, WhatWeb-ek eskaintzen duen komando baten bidez (whatweb -log-sql=FILE) fitxategi bat lortzen dugu, datu basean web-gune horren datuak sartzeko INSERT guztiekin.
- Informazioa pantailaratzea:
Jadanik Datu Basean gordeta dauden datuak SceneBuilder-en sortutako taulan (TableView) txertatu ditugu.
- Eskumako botoiarekin klik egitean Menu bat ireki:
 - Web Guneak ezabatu:
Taulan ikusgai dauden orrialdeak Datu Basetik ezabatzeko aukera dago.
 - Web Gunearen ScreenShot-a lortu
Orrialdean sartzean ikusten den lehenengo leihoaren argazki bat ikusiko da.
 - Web Gunera sartu
 - Orrialdea eguneratzeko botoia:
Web-gune baten informazioa eguneratu nahi badugu, ezabatu eta berriro sartzearen ordeztu botoi hau sakatu eta eguneratuko du.

1.2.2 Ez Implementatutako funtzionalitateak

- MongoDB datu basearen erabilera.

1.3 Plangintza

Proiektu hau aurrera eramateko hainbat zatitan banatu dugu lana. Hiru multzo nagusi ezberdindu ahal ditugu; datu basearen kudeaketa, aplikazioaren atal grafikoaren kudeaketa eta Java aplikazioaren kudeaketa.

- Datu Basearen Kudeaketa:
Ataza honetan lortutako informazioa datu basean txertatu, kontsultatu, etab.-etaz arduratu gara. Hasiera batean MySQL datu basea erabili genuen, baina amaieran SQLite erabiltzera animatu ginen.



Irudia 2: SQLite



Irudia 3: SQL

- Atal grafikoaren kudeaketa:
Zati honetarako SceneBuilder(4) aplikazioa erabili dugu JavaFX elementuak kudeatzeko. Aplikazio honekin eta "fontawesome" estilo-orriarekin WhatWeb aplikazioari estilo moderno eta ulergarri bat eman diogu.



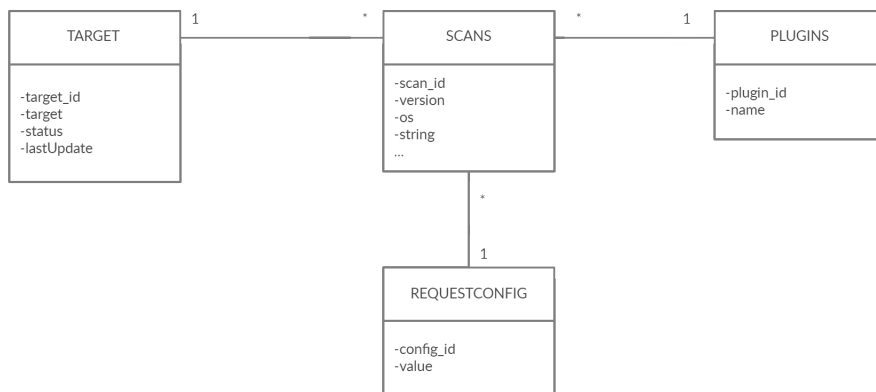
Irudia 4: SceneBuilder

- Java aplikazioaren Kudeaketa:
Aurreko bi atalak elkartzeko eta aplikazioak koherentzia izan dezan, beste aldetik Java proiektuaren pakete/klaseak modu egokian banatzea.

WhatWeb aplikazioa lortzeko, hurrengo pausu orokorrak jarraitzea komenigarria litzateke:

- Java proiektuan pakete ezberdinak sortu
 - Datu Basea kudeatzeko (Datu Basearekin zuzenean konexioa egiteko, eta beste atazekin konektatzeko)
 - UI-ren kudeatzaileak (Orokorrean MainKudeatzaile bat eta leiho bakoitzeko beste kudeatzaile bat)
 - Java kudeatzeko (behar dugun informazioa kudeatu/gordetzeko klaseak)
- Gauzak pixkanaka probatzen joan, hau da, helburu txikiak jarri eta horiek lortutakoan muga handiagoak jarri.

2 Analisia



Irudia 5: Domeinuaren eredua

Gure datu basean lau taula nagusi ditugu:

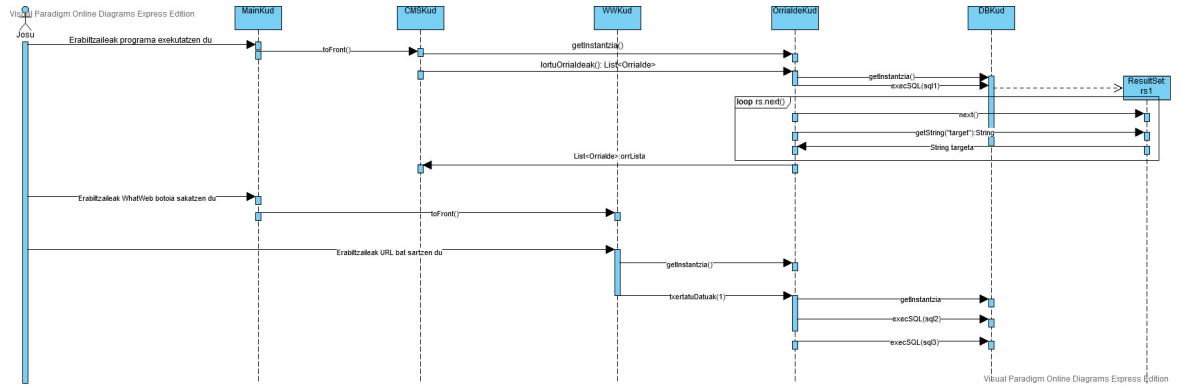
- Plugins: WhatWeb detekta ditzakeen plugin guztiak. WhatWeb aplikazioak proportzionatu digu lista hau.
- Target: Bilatutako webguneen zerrenda. Webgune bat eskaneatu ondoren datu basera txertatzen den lehenengo gauza da.
- RequestConfig: Webgune bat zein WhatWeb bertsioarekin eskaneatu zen gordetzen du.
- Scans: Webgune batean aurkitutako plugin bakoitzaren informazioa. Horretarako, webgunearen id-a, pluginaren id-a eta analisia egiteko momentuan WhatWeb-en bertsioa gordeko du

Eskaneo bakoitzaren ostean, momentuko data gordetzen dugu, horrela informazioa zein egunetan eguneratu zen ikus ahal izateko.

3 Diseinua

3.1 Sekuentzia Diagrama

- Gure taldeak WhatWeb leihoan orrialde berri bat sartzen denaren sekuentzia diagrama egin dugu.



Irudia 6: Sekuentzia diagrama

- Eskaerak:
 - sql1: SELECT target FROM targets
 - sql2: SELECT targetId FROM targets WHERE target GLOB "Erabiltzaileak sartutako URL-a"
 - sql3: INSERT INTO scans VALUES(targetId,configId,pluginId,version,os,string,account,model,firmware,module,filepath,certains)
 - 1: Erabiltzaileak sartutako URLaren ondorioz WhatWeb aplikazioak sortutako 'INSERT'ak.

4 Implementazioa

Implementazioa azaltzerakoan bi atal nagusi berezituko ditugu. Alde batetik, derrigorrezkoak diren eta enuntziatuan ageri diren funtzioak azalduko ditugu, eta bestetik, taldeak gehigarri bezala implementatutakoak.

Implementazioaren funtzio gehienak exekutatzeko aplikazioa martxan jarraitzeko Javak eskaintzen duen Task elementuak erabili ditugu. Elementu hauei esker funtzio horiek (adibidez pantaila-argazkia) bigarren mailan exekutatzen dira eta bitartean aplikazioa ez da blokeatzen.(5)

1.2 puntuan aplikazioaren funtzionalitate nagusiak azaldu dira. Orain funtzionalitate horiek nola implementatu diren azalduko dugu.

4.1 Egin Beharrekoak

- Erabiltzaileak sartutako URL-a bilatu eta Datu Basean gordeko dira, honetarako, lehen aipatutako WhatWeb aplikazioak ematen duen aukera erabili dugu (whatweb -log-sql=FILE, non FILE horretan datu basean txertatuko ditugun INSERT guztiak ageri diren). Ondoren, Datu Basean txertatzeko hurrengo eskaera erabili dugu:

```
INSERT INTO scans VALUES (targetId, configId, pluginId, version, os, string, account, model, firmware, module, filepath, certainty)
```

- Datu basean dauden elementuak tauletan argitaratuko dira bi atal bereizten. Lehenengoan CMS eta haren bertsioa eta bigarrenean zein zerbitzari mota erabiltzen duen eta haren IP helbidea erakutsiko zaigu. Funtzionalitate hau bermatzeko eskaera baten bidez orrialdeak sortu ditugu eta *nahi-DugunZutabea.setCellValueFactory(new PropertyValueFactory<Orrialdea>(Orrialdea.klaseko zein parametro nahi dugun))* komandoaren bitartez zutabeei balioak eman dizkiogu. Eskaerak hurrengoak dira:

```
SELECT target FROM targets
```

Ondoren target bakoitzaren informazio gehigarria lortuko dugu:

```
SELECT target.id, lastUpdate FROM targets WHERE target=lehen lortutako target-a
```

Eta informazio gehigarri horrekin, taulan bistaratu behar diren xehetasunak lortuko ditugu:

- Webgunearen CMS-a lortu:
SELECT string FROM scans where string like Ditugun CMS guztiak and targetId = lehen lortutako targetId
 - Webgunearen web zerbitzaria lortu:
SELECT string FROM scans where plugin_id = 295 and targetId = lehen lortutako targetId
 - Webgunearen IP helbidea lortu (eskaneatutako plugin guztiak string zutabea daudenez, hurrengo select-a erabili behar izan dugu, datu basea beste egitura bat izango balu, errazagoa izango litzateke IP helbideak lortzea):
SELECT string FROM scans where targetId=457 and targetId=lehen lortutako targetId
- CMS eta Server leihoetan agertutako helbideak eskuineko botoia sakatzean menu bat aterako da, bertan hiru aukera izango ditugu. Lehenengo aukerak bezala helbide hori datu basetik kentzeko aukera emango digu, bigarrenak helbidearen pantaila-argazki bat aterako digu eta hirugarrenak helbide hori nabigatzailearekin irekitzeko aukera emango digu. Bigarren eta hirugarren atalak "Ekarpen berriak"izeneko sekzioan azalduko da eta lehenengo funtzionalitaterako hurrengo eskaera erabili dugu:
SELECT target_id FROM targets WHERE target=webgunearen URL helbidea
DELETE FROM targets WHERE targetId = helbidearen target-a
DELETE FROM scans WHERE targetId = helbidearen target-a

4.2 Ekarpn Berriak

- Helbide horren orrialdearen pantaila-argazki egingo da, horretarako jonander.xyz zerbitzariak 3000. portuan daukan pluginari deituko diogu helbidea parametro bezala bidalita. Zerbitzari honek HTML orri bat bueltatuko digu, gure lana HTML horren barnean dagoen irudia hartzea eta pantailaratzea da.

Zerbitzarian plugin honen funtzionamendua bermatzeko eta *node.js* exekutagarriak bigarren mailan funtzionatzeko pm2 erabili dugu.

Hainbat sarek ez dituztenez usten zerbitzariak portuak erabiltzea, adibidez unibertsitateko Eduroam sareak, nginx zerbitzaria martxan jarri dugu eta *jonander.xyz/screenshooter* bilatzerakoan zerbitzaria berak textitjonander.xyz:3000 helbidean dagoen *screenshooter* funtzioa exekutatu du. (6) Zerbitzari honetan arazoak daudenean eta konexioa ezinezkoa denean gure programak blokeatzen zen, hau ekiditeko hari bat sortu dugu. Hari honek 5 segundo pasa ostean bukatzen da eta zerbitzariarekin konexioa ez bada lortzen konexio hori itxi egiten du. (7)

- Helbidea nabigatzaile batekin irekiko da. Aukera hau aukeratzerakoan, leiho berri bat irekiko da eta erabiltzen diren bi nabigatzaile nagusien artean aukeratuko bezeroak, Chrome (*google-chrome helbidea*) edo Firefox (*firefox helbidea*). Denbora pasa ahala nabigatzaile gehiago sartzen saiatuko gara.
- Gure bezeroek hainbat URL sartu nahi izango dutenez URL horiez osatutako fitxategi bat irakurtzeko aukera eman dugu. Fitxategi horien erroak banan banan exekutatu dira eta whatweb komandoaren bitartez.

4.3 Etorkizunerako planak eta etengabeko hobekuntza

- Hautazkoak ziren beste funtzioen inplementazioa.
- Aplikazioak zure ordenagailuan instalatuta dituzun nabigatzaileak detektatzea eta webgune bat irekitzeko momentuan, soilik instalatuta dauden nabigatzaileak erabiltzeko aukera ematea.
- Webguneen analisi prozesua zerbitzari batean egitea, honela ez genukeen sistema eragile bakoitzerako komando ezberdinak programatu behar:
 1. Erabiltzaileak aplikazioan URL helbidea idaztea.
 2. Botoi bat klikatzea eta helbidea zerbitzarira heltzea.
 3. Zerbitzaria WhatWeb exekutatzeko eta lortutako emaitza datu basean gordetzea.
 4. Programako taula zerbitzariko datu baseko datuekin osatzea.
- Behin datu basea zerbitzarian egonda, webguneen eguneraketa automatizatzea.
- Jonander.xyz zerbitzarian pantaila-argazkiak egiteko dagoen plugin-a hobetu oraingoa demo bat delako.

5 Bibliografia

- [1] J. Pereira, “Whatweb proiektuaren enuntziatua.” [Online]. Available: <https://docs.google.com/document/d/1-fZgXxEMwEvzubbd9xnbA-aU-k9yvnCkGbMGHvkisqs/edit>
- [2] B. C. Andrew Horton, “Whatweb - next generation web scanner.” [Online]. Available: <https://github.com/urbanadventurer/WhatWeb>
- [3] V. Coutinho, “Sistema de gestión de contenidos (cms): ¿por qué implementarlo en tu empresa?” [Online]. Available: <https://rockcontent.com/es/blog/cms/>
- [4] Gluon, “Scene builder.” [Online]. Available: <https://gluonhq.com/products/scene-builder/>
- [5] Benjamin, “Java 8 concurrency tutorial: Threads and executors.” [Online]. Available: <https://winterbe.com/posts/2015/04/07/java8-concurrency-tutorial-thread-executor-examples/>
- [6] J. Luetgebrune, “Guest post: How to set up and deploy a node.js/express application for production.” [Online]. Available: <https://deploybot.com/blog/guest-post-how-to-set-up-and-deploy-nodejs-express-application-for-production>
- [7] kinghomer (Stack Overflow username), “Java url connection timeout.” [Online]. Available: <https://stackoverflow.com/questions/3163693/java-urlconnection-timeout>