

## Gradu Amaierako Lana

Informatika Ingeniaritzako Gradua

Software Ingeniaritza

---

### **Earler: Webguneen irisgarritasuna aztertze arakatzaile-luzapena**

---

*Julen Larrañaga Aguirrezabala*

#### **Zuzendariak**

Juan Miguel López Gil

2023.eko ekainaren 18



# Laburpena

Web irisgarritasunaren helburua, webean kokatzen diren edukiak edonork erabili edo kontsumitu ahal izateko modukoak direla bermatzea da. Web irisgarritasunean oinarritutako estandarrek orrialdeen irisgarritasunaren betetze maila neurtu ahal izateko irizpideak definitzen dituzte, baina neurketa hori egitea konplexua izan daitekeenez, garatzaileei lagundu ahal izateko mota ezberdinetako tresnak sortu dira. Tresnak webgune bidezkoak, luzapenak, liburutegiak, etab. izan daitezke, baina orokorrean funtzionalitate berak eskaintzen dituzte.

Earler luzapenaren bitartez, tresna ezberdinen emaitzak irisgarritasun txostenetan bateratu, eta txostenak erabiltzaileen artean partekatu eta berrerabili ahal izateko funtzionalitateak eskaintzen dira. Gainera, aipatutako tresnen funtzionalitate orokorrak ere inplementatzeaz aparte, txostenak luzapenean bertan, edo nahiago bada WCAG EM Report Tool tresnara igoz, editatzeko aukera ere ematen du. Hori guztia Google Chrome arakatzailerako luzapen bat sortuz eta aparteko zerbitzari batez baliatuz lortu da, eta memoria honen bitartez egindako lan guztia dokumentatuta utziko da<sup>1</sup>.

---

<sup>1</sup>Earler tresnarentzako instalazio gida eranskinen "Instalazio gida" 9.3 atalean dago



# Gaien aurkibidea

<b>Gaien aurkibidea</b>	<b>iii</b>
<b>Irudien aurkibidea</b>	<b>vii</b>
<b>Taulen aurkibidea</b>	<b>x</b>
<b>1 Sarrera</b>	<b>1</b>
1.1 Testuingurua . . . . .	1
1.2 Helburu orokorrak . . . . .	2
<b>2 Erlazionatutako lanak</b>	<b>3</b>
2.1 Arakatzaille-luzapenak . . . . .	3
2.2 Webguneak . . . . .	6
2.3 Liburutegiak, toolkitak, etab. . . . .	10
2.4 AC-Check . . . . .	11
2.5 Tresna gehiago . . . . .	12
2.5.1 Webguneak . . . . .	12
2.5.2 Luzapenak . . . . .	12
2.5.3 Bestelakoak . . . . .	12
<b>3 Plangintza</b>	<b>13</b>
3.1 Irismena . . . . .	13
3.1.1 Helburu zehatzak . . . . .	13
3.1.2 Eskakizunak . . . . .	13
3.1.3 Baztertutakoen deskribapena . . . . .	15
3.1.4 Emangarriak . . . . .	16
3.1.5 Lan paketeak . . . . .	16
3.2 Atazak egingo diren denboraldiak eta emangarriak bukatzeko datak . . . . .	17
3.2.1 Egin beharreko atazen deskribapena . . . . .	17
3.2.2 Atazen garapen denboraldiak . . . . .	20
3.3 Ataza bakoitzari emango zaion denboraren estimazioak . . . . .	21
3.4 Proiektuko informazio eta komunikazio sistemak . . . . .	23
3.4.1 Informazio sistema . . . . .	23
3.4.2 Komunikazio sistema . . . . .	23
3.5 Arriskuak . . . . .	23
<b>4 Trebakuntza</b>	<b>25</b>
4.1 Zerbitzari berria . . . . .	25

4.1.1	Luzapenerako argazkiak . . . . .	25
4.1.2	CORS arazoak . . . . .	26
4.1.3	Zerbitzariko eskaerak nola jaso eta kudeatu aurkitu . . . . .	26
4.2	React, JSON-LD eta Puppeteer ikasten . . . . .	26
<b>5</b>	<b>Diseinua</b>	<b>27</b>
5.1	Erabilpen kasuen diagrama . . . . .	28
5.1.1	Aktoreak . . . . .	28
5.1.2	Erabilpen kasuak . . . . .	28
5.2	Datu-basearen diseinua . . . . .	30
5.3	Erabilpen kasuen sekuentzia diagramak . . . . .	31
5.3.1	Login . . . . .	31
5.3.2	Register . . . . .	31
5.3.3	Logout . . . . .	32
5.3.4	Evaluate selected scope . . . . .	33
5.3.5	Remove current report . . . . .	35
5.3.6	Download current report . . . . .	35
5.3.7	Upload report file . . . . .	36
5.3.8	Store current report . . . . .	36
5.3.9	Load stored report . . . . .	37
5.3.10	Edit found cases . . . . .	37
5.4	Luzapenaren diseinu grafikoa . . . . .	39
5.4.1	Luzapenaren oinarritzko egoerak . . . . .	39
5.4.2	Erregistratzeko eta kontuan sartzeko egoerak . . . . .	40
5.4.3	Datu-basean gordetako ebaluaketa txostenak kargatzeko egoera . . . . .	41
5.4.4	Emaitza taulak . . . . .	42
5.5	Irisgarritasun txostenen formatua . . . . .	43
<b>6</b>	<b>Inplementazioa</b>	<b>51</b>
6.1	Luzapenaren implementazioa . . . . .	51
6.1.1	React "components" edo osagaiak . . . . .	51
6.1.2	Osagaien estiloak . . . . .	57
6.1.3	Funtzio erabilgarriak . . . . .	58
6.1.4	Interfaze grafikoaren diseinu finala . . . . .	60
6.1.5	Funtzionalitate gehigarriak . . . . .	66
6.2	Zerbitzariaren implementazioa . . . . .	68
6.2.1	Zerbitzariari egindako eskaeren kudeaketa . . . . .	68
6.2.2	ScrapingController . . . . .	69
6.2.3	UserController . . . . .	70
6.2.4	ReportController . . . . .	71
6.3	Aurkitutako arazoak . . . . .	72
6.3.1	Estiloen arazoak . . . . .	72
6.3.2	LocalStorage arazoak . . . . .	72
6.3.3	Scraping egitearen arazoak . . . . .	72
6.3.4	HTML elementuak eskuratzeko erreferentziekin arazoak . . . . .	73
6.3.5	Ebaluaketa automatikoen arazoak . . . . .	73
<b>7</b>	<b>Probak</b>	<b>75</b>

7.1	Ebaluaketa automatikoen probak . . . . .	75
7.1.1	Denbora probak . . . . .	75
7.1.2	Emaitzen test kasuak . . . . .	78
7.2	Txostenen emaitzak aldatzeko probak . . . . .	82
7.2.1	Irizpide bateko aurkitutako kasuen ezabaketa . . . . .	82
7.2.2	Irizpide bateko kasu baterako ebaluatzaileen ezabaketa . . . . .	82
7.2.3	Irizpide bateko kasu baterako edizioa . . . . .	84
7.3	Txostenak datu-basean gorde eta bertatik kargatzeko probak . . . . .	86
7.3.1	Txostenak gordetzea . . . . .	86
7.3.2	Txostenak kargatzea . . . . .	88
<b>8</b>	<b>Jarraipen eta kontrola</b>	<b>91</b>
8.1	Atazen dedikazio denboren desbiderapenak . . . . .	91
8.2	Garapen denboraldien desbiderapenak . . . . .	94
<b>9</b>	<b>Ondorioak</b>	<b>95</b>
9.1	Ondorioak . . . . .	95
9.2	Ikasitakoa eta esperientzia pertsonala . . . . .	96
9.3	Etorkizunerako hobekuntza posibleak . . . . .	96
	<b>Eranskinak</b>	<b>99</b>
	<b>Erabilitako tresnak</b>	<b>101</b>
	AccessMonitor . . . . .	101
	AChecker . . . . .	101
	A11y . . . . .	101
	Body-parser . . . . .	101
	ChatGPT . . . . .	101
	Chrome arakatzailleko luzapenentzako baliaibideak . . . . .	101
	Cors . . . . .	102
	Diagrams . . . . .	102
	Express . . . . .	102
	GitHub . . . . .	102
	GlooMaps . . . . .	102
	JSDoc . . . . .	102
	JavaScript . . . . .	102
	Lighthouse . . . . .	102
	Mauve++ . . . . .	103
	Node.js . . . . .	103
	Nodemon . . . . .	103
	Online Gantt . . . . .	103
	Overleaf . . . . .	103
	Pa11y . . . . .	103
	Puppeteer . . . . .	103
	React . . . . .	103
	React-spinners . . . . .	103
	Sass . . . . .	104
	Sqlite3 . . . . .	104

XPath . . . . .	104
<b>Instalazio gida</b>	<b>105</b>
Node.js instalatzen . . . . .	105
Earler deskargatu eta martxan jartzen . . . . .	106
<b>Bibliografia</b>	<b>109</b>



# Irudien aurkibidea

2.1	AInspector luzapena ehu.eus webgunean . . . . .	3
2.2	AInspector zerrendako elementua gorritz nabarmenduta . . . . .	4
2.3	WAVE luzapena ehu.eus webgunean . . . . .	4
2.4	WAVE irisgarritasun irizpidea ikusteko "Reference" atala . . . . .	5
2.5	AccessMonitor ebaluaketa egin aurretiko orrialdea . . . . .	6
2.6	Mauve++ orrialdearen ebaluaketa . . . . .	7
2.7	Mauve++ adierazitako hainbat orrialde aldi berean . . . . .	8
2.8	Mauve++ sakonera bateraino ebaluatu . . . . .	8
2.9	Mauve++ orrialdea grafikoki ikusteko eremua . . . . .	9
2.10	Lighthouse tresnaren hasierako interfazea . . . . .	10
2.11	Lighthouse irisgarritasun ebaluaketa emaitzak . . . . .	11
3.1	Proiektuaren WBS/LDE diagrama . . . . .	17
3.2	Hasierako Gantt diagrama . . . . .	20
4.1	manifest.json konfigurazio fitxategian luzapeneko argazkiak adierazteko . . . . .	25
5.1	Erabilpen kasuen diagrama . . . . .	28
5.2	Erabiltzaileak gordetzeko taula . . . . .	30
5.3	Txostenak gordetzeko taula . . . . .	30
5.4	Login sekuentzia diagrama . . . . .	31
5.5	Register sekuentzia diagrama . . . . .	32
5.6	Logout sekuentzia diagrama . . . . .	32
5.7	Evaluate selected scope sekuentzia diagrama . . . . .	34
5.8	Remove current report sekuentzia diagrama . . . . .	35
5.9	Download current report sekuentzia diagrama . . . . .	35
5.10	Upload report file sekuentzia diagrama . . . . .	36
5.11	Store current report sekuentzia diagrama . . . . .	36
5.12	Load stored report sekuentzia diagrama . . . . .	37
5.13	Edit found cases sekuentzia diagrama . . . . .	38
5.14	WCAG EM Report Tool tresnako botoien diseinua . . . . .	39
5.15	Luzapena oinarritzeko egoera . . . . .	40
5.16	Kontuan sartzeko eta erregistratzeko egoerak . . . . .	41
5.17	Datu-baseko txostenak kargatzeko egoera . . . . .	41
5.18	Aurreko urteko taulen diseinua . . . . .	42
5.19	WCAG EM Report Tool 1.pausua . . . . .	46
5.20	WCAG EM Report Tool 2.pausuko azpiatala . . . . .	46
5.21	WCAG EM Report Tool 5.pausua . . . . .	47

5.22	WCAG EM Report Tool 4.pausua irizpide bateko emaitzak	49
6.1	Luzapeneko React osagaiak	52
6.2	Extension osagaiak itzultako zati nagusia	52
6.3	UserAuthentication osagaiak itzultako zati nagusia	53
6.4	Erabiltzaileak sartutako web orrialdeak zerrendatzeko	53
6.5	Defektuzko ebaluatzaileen aukeraketarako checkbox elementuen egoera	54
6.6	Checkbox elementuak zerrendatzeko eta erakusteko kodea	54
6.7	Erabiltzaile guztientzako aukerak	54
6.8	Kontuan sartutako erabiltzaileentzako aukerak	55
6.9	Datu-basean gordetako txostenak erakusteko taularen egitura	55
6.10	Erabiltzaileak kontuan izan nahi duen conformanceLevel aukeratzeko kodea	56
6.11	Kargatutako txostenaren emaitzak erakusteko kodea	56
6.12	Koloreak berrerabiltzen	57
6.13	Sass webguneko argazkia nesting kontzeptua ulertzeko	57
6.14	Proiektuko estilo fitxategi guztiak	58
6.15	Zerbitzarira eskaerak egiteko funtzioa	58
6.16	Orrialdeko elementuak eskuratzeko funtzioa	59
6.17	Datu-basean gordetako txostena lortzeko funtzioa	59
6.18	Ebaluatzaile aukeren atala aldatua	60
6.19	Evaluation scope atalaren diseinu finala	60
6.20	Select evaluators atalaren diseinu finala	61
6.21	Login eta Register formularioen diseinu finala	61
6.22	Datu-baseko txostenak kudeatzeko diseinua	62
6.23	Aukeratutako conformance level eta laburpen taulako tab-aren arabeko emaitzak	63
6.24	Webgunea ebaluatu da baina orrialdea ez	63
6.25	Ebaluaketa emaitzen adibidea	64
6.26	Irizpideen emaitzak aldatzen	64
6.27	Erreferentziei dagokien kode zatiak urdinez koloreztatuta	65
6.28	Aukeratutako erreferentzia gorri koloreztatuta	65
6.29	Luzapena alde batean ezkutatzeko funtzionalitatea	66
6.30	Nabarmendutako elementuan kurtsorea jartzean azaltzen den popup elementua	67
6.31	Luzapeneko konfigurazio orria	67
6.32	Zerbitzariko eskaeren kudeaketarako index.js fitxategia	68
6.33	ScrapingController ebaluaketa automatikoa	69
6.34	Ebaluatzaileen ebaluaketa txostenak bateratzeko funtzioa	70
6.35	Erabiltzailea datu-basean erregistratzeko kodea	70
6.36	Erabiltzailea datu-basean dagoen egiaztatze kodea	71
6.37	Datu-basetik txostena ezabatzeko funtzioa	71
7.1	ACT test kasuen JSON fitxategiaren egitura	79
7.2	1.1.1 irizpideko kasuak kentzen	82
7.3	1.3.1 irizpideetatik ebaluatzaileak kenduko dizkiogun failed kasua	83
7.4	1.3.1 irizpideko failed kasuko ebaluatzaileak kentzen	84
7.5	1.3.1 irizpideko cantTell kasuan edizio moduan sartu eta aldaketak egin	85
7.7	Txostena igotzean ikusten den mezua	86
7.8	ehu.eus webguneko igotako 3 txostenak	87

7.9	ataria.eus webgunerako igotako txostena . . . . .	87
7.10	Txostenen taulak . . . . .	88
7.11	ataria.eus webgunerako txosten bakarra ezabatzean . . . . .	89
7.12	azken txostena igo ondoren guraso bera duten bi txosten ditugu . . . . .	89
8.1	Amaierako Gantt diagrama . . . . .	94
1	Node.js ingurunearen webguneko deskargen orrialdea . . . . .	105
2	Node.js instalatzeko dependentzien aukera . . . . .	106
3	Zerbitzaria martxan jartzean ikusten dena . . . . .	107
4	Luzapena inportatzeko lehen pausua . . . . .	107
5	Luzapena inportatzeko bigarren pasua . . . . .	107

# Taulen aurkibidea

3.1	Estimatutako ataza bakoitzeko dedikazioa . . . . .	22
5.1	Emaitzentzako kolore paleta . . . . .	42
7.1	ehu.eus webgunerako ebaluaketa automatikoen denbora probak . . . . .	76
7.2	ataria.eus webgunerako ebaluaketa automatikoen denbora probak . . . . .	76
7.3	The Majestic Million zerrendatik ebaluatuko ditugun hoberen klasifikatutako 8 webguneak . . . . .	77
7.4	google.com webgunerako ebaluaketa automatikoen denbora probak . . . . .	78
7.5	facebook.com webgunerako ebaluaketa automatikoen denbora probak . . . . .	78
7.6	Ebaluatzaile bakoitzak testak pasatzean jasotako emaitzak . . . . .	80
7.7	Gutxieneko passed kopurua izan dituzten test kopuruak . . . . .	80
7.8	Ebaluatzaile guztien arteko konbinazio hoberenak . . . . .	81
7.9	AChecker kenduta, gainontzeko ebaluatzaileen arteko konbinazio hoberenak . . . . .	81
8.1	Estimatutako eta errealitatean dedikatutako ataza ahoitzeko denborak orduetan . . . . .	93

# Sarrera

Dokumentu hau Julen Larrañaga Aguirrezabalaren Informatikako Ingeniaritzako Gradu Amaierako Lanaren memoria da. Proiektua Juan Miguel López Gil tutorearen gidaritzapean garatu da, Euskal Herriko Unibertsitateko (UPV/EHU) Donostiako Informatika Fakultatean.

Memoria honen bitartez Earler [1] izeneko proiektua aurrera eramateko egindako lana azalduko da, aurreko urtean Mikel Iturriak bere Gradu Amaierako Lanaren garatu zuen AC-Check [2] luzapenaren jarraipena izanik.

Kapitulu honen bitartez, proiektuaren testuingurua zehaztu eta helburu orokorrak zeintzuk diren azalduko dira. Helburu orokor horiek betetzeko definitu beharreko helburu zehatzak, aurrerago plangintza kapituluko irisgarritasunari dagokion atalean zehaztuko dira, ikus 3.1.1 helburu zehatzen atala.

## 1.1 Testuingurua

Web irisgarritasunaren helburua, webean kokatzen diren tresnak edonork erabili ahal izateko modukoak direla bermatzea da, hau da, desgaitasuna duen pertsona batek gai izan behar du beste edonork erabil ditzakeen tresna berberak erabili, edota kontsumitu ditzakeen edukia kontsumitu ahal izateko ere.

Horretarako, webguneetan berregokitzapenak egin eta baliabide espezializatu edota berezituak eskaini behar dira. Web irisgarritasunean oinarritutako estandarren bitartez, guneen baliabide horien inplementatze eta egokitze mailak neurtu daitezke. Hori kontuan izanda, webgune baten irisgarritasuna neurtzen dela esatean, webgune horren domeinuan kokatzen diren orrialdeen betetze maila neurtzen dela adierazten da.

Aipatutako estandar horiek, World Wide Web Consortium edo W3C-k zehaztutako irisgarritasun estandarrak [3] dira, irisgarritasuna bermatzeko moduaren arabera sailkatuak. Adibidez, jarraibideak definitzen dituztenetakoak nagusiki hiru dira: Authoring Tool Accessibility Guidelines edo ATAG, User Agent Accessibility Guidelines edo UAAG eta Web Content Accessibility Guidelines edo WCAG. Beste estandar baten adibide bat, Accessible Rich Internet Applications edo WAI-ARIA izango litzateke, kasu honetan irisgarritasuna bermatzeko zehaztutako teknikoak ezartzen dituenak.

Ikus daitekeenez, hainbat estandar daude, eta horietako bakoitzak barrutik konplexutasun handia du. Hori dela eta, irisgarritasuna bermatzeko prozesuan estandarrek definitutako arauak banan-banan irakurri eta betetzen diren ikusteak lan asko eskatzen du, eta ondorioz, garatzaileei lana erraztu eta laguntza emateko, estandarretan oinarritzen diren mota ezberdinetako tresnak sortu dira: webguneak, arakatzailentzako estentsio edo luzapenak, liburutegiak, etab. [4, 5]

Tresna horiek antzekotasun handia izaten dute, eta orokorrean oinarritzko baliabideak eskaintzen dituzte. Hau da, erabiltzaileak ebaluatu nahi duen webgune edo orrialdearen esteka emanda, irisgarritasuna automatikoki aztertze, irizpideen betetze mailak adierazteko, dagozkien kode zatiak nabarmentzeko, eta egindako azterketen datuak jasotzen dituzten fitxategi edo txostenak jaitsi ahal izateko aukerak izaten dituzte.

Horretaz aparte, nahiz eta tresna bakoitzak bere ezaugarri eta berezitasunak izan ditzakeen, aipatutakoez gain ez dute oso interesgarria edo lagungarria izan daitekeen baliabide gehigarri eskaintzen.

### **1.2 Helburu orokorrak**

Hori guztiagatik, proiektu honen helburu nagusia, webguneen irisgarritasunak aztertze prozesuan erabiltzaileei ahalik eta laguntza handiena ematen dien produktu bat garatzea da. Horretarako, dagoeneko sortutako tresnen alderdi onak bildu eta erabiltzaileentzako erabilgarriak izan daitezkeen baliabide berriak eskaini beharko ditu.

Gainera, lan honen beste helburu orokor bat nire informatika ingeniartzako hezkuntza prozesuan beste pausu bat ematea da, graduan zehar ikasitakoa praktikan jarritz, eta orokorrean garatzaile batek lan egiteko izan beharreko gaitasunak landuz.

## Erlazionatutako lanak

Sarreran aipatu bezala, web orrialdeen irisgarritasuna aztertzeko tresna mordera dago dagoeneko. Hori dela eta, proiektuarekin hasi aurretik, web irisgarritasun analisirako tresna ezberdinen azterketa bat egitea komenigarria da. Modu horretara, dagozkien ezaugarri interesgarriak kontuan hartu eta izan ditzaketen gabeziak garatuko dugun tresnaren bitartez hobetzen saia gaitezke.

### 2.1 Arakatzaille-luzapenak

Luzapenekin hasiz, normalean unean kokatutako web orrialdea automatikoki ebaluatu eta emaitzak taula edo zerrenda moduko batean erakusten dute. Modu horretara, zerrendako elementuetan sakatzean, lortutako emaitzak orrialdean bertan koloreztatuz edota kutxetan sartuz nabarmentzeko joera izaten dute. Adibidez, Firefox arakatzailleko AInspector [6] tresnaren kasuan, aipatutako izaera hori nabarmena da, ikus 2.1 eta 2.2 irudiak.

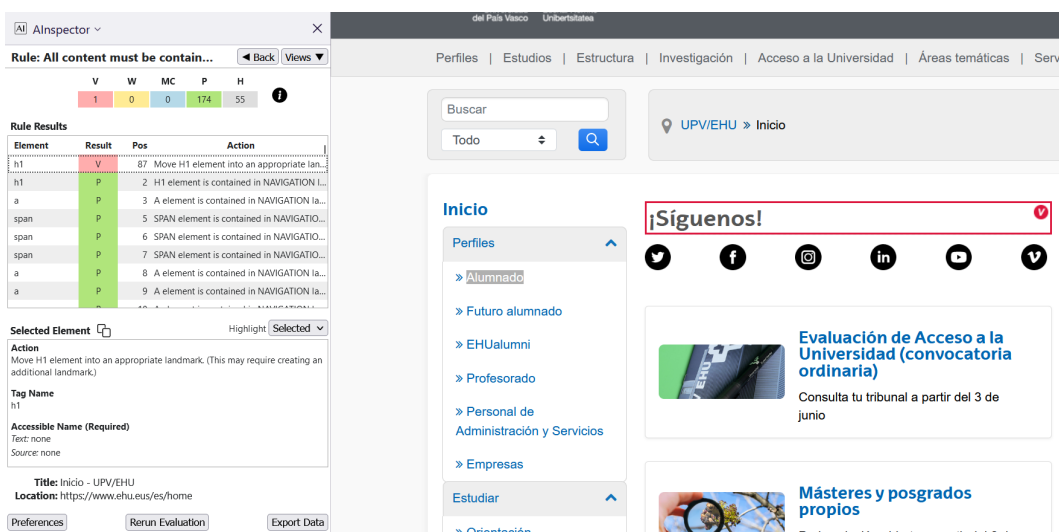
The screenshot shows the AInspector tool interface. At the top, there's a browser address bar with the URL 'https://www.ehu.es/es/home'. Below it, the AInspector summary shows 'Summary: All Rules' with a 'Back' button and 'Views' dropdown. A summary bar displays counts for Violations (6), Warnings (3), Minor Catches (39), and Passes (31). Below this is a table of WCAG Guidelines with columns for Violations (V), Warnings (W), Minor Catches (MC), and Passes (P).

Guideline	V	W	MC	P
1.1 Text Alternatives	0	0	4	3
1.2 Time-based Media	0	0	0	0
1.3 Adaptable	5	1	5	7
1.4 Distinguishable	0	0	4	0
2.1 Keyboard Accessible	0	0	1	1
2.2 Enough Time	0	0	2	0
2.3 Seizures	0	0	1	0
2.4 Navigable	0	2	8	9
3.1 Readable	0	0	1	1
3.2 Predictable	0	0	6	0
3.3 Input Assistance	0	0	5	4
4.1 Compatible	1	0	2	6
All Rules	6	3	39	31

At the bottom of the interface, there are buttons for 'Preferences', 'Rerun Evaluation', and 'Export Data'. The page title is 'Inicio - UPV/EHU' and the location is 'https://www.ehu.es/es/home'. On the right side, there's a search bar and a list of profiles including 'Alumnado', 'Futuro alumnado', 'EHUalumni', 'Profesorado', and 'Personal de Administración y Servicios'.

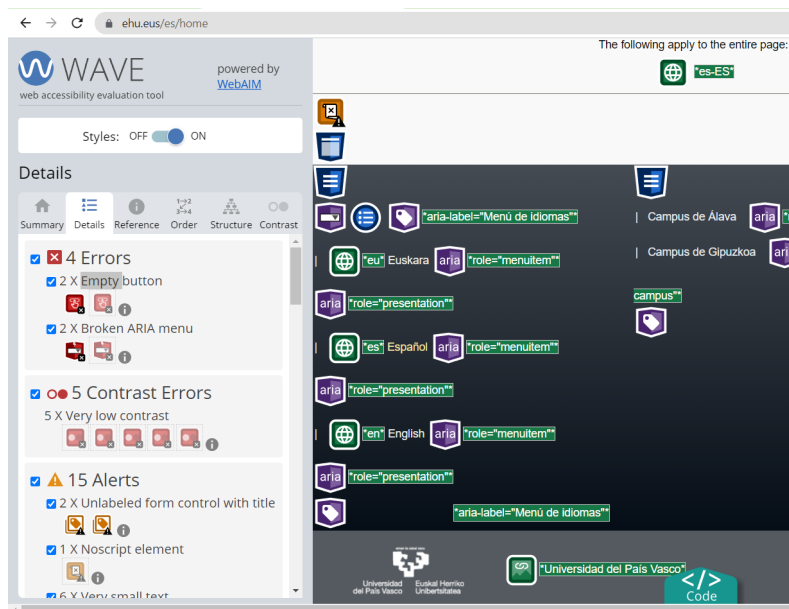
2.1 Irudia: AInspector luzapena ehu.es webgunean

## 2. ERLAZIONATUTAKO LANAK



2.2 Irudia: AlInspector zerrendako elementua gorritz nabarmenduta

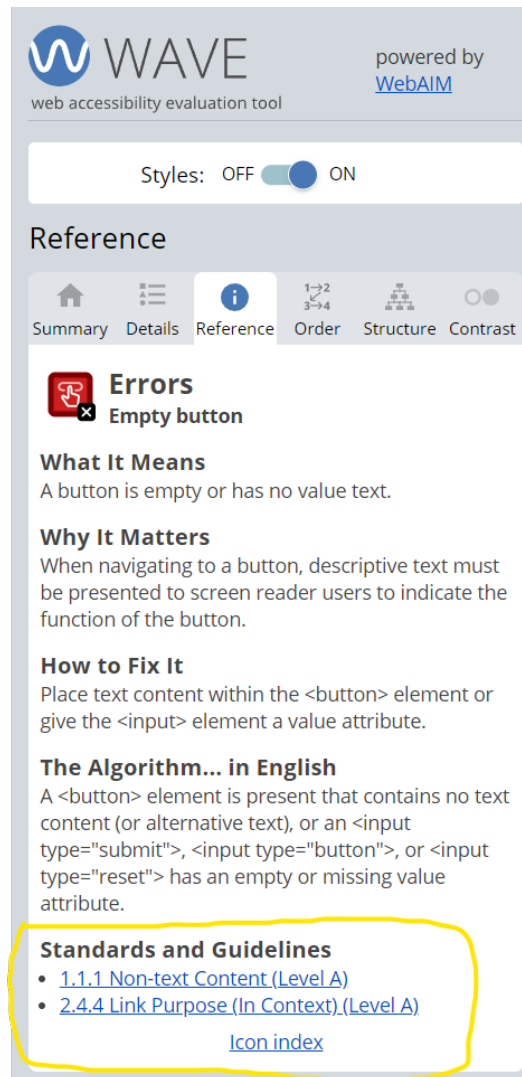
Beste adibide bat, Google Chrome arakatzailerako WAVE Evaluation Tool [7] luzapena izango litzateke. Kasu honetan, elementu guztiak zuzenean nabarmenduta etortzen dira, eta ondoren zerrendan aukeraketa egin daiteke, ikus 2.3 irudia. Erabiltzen dituen kuxkak eta irudiak ezagutzen badituzu ikusterekin bakarrik asko lagundu dezakete, baina begi-bistakoa denez, orrialdean egiten dituen aldaketak gehiegizkoak suerta daitezke, erabiltzailea nahasteko arriskuarekin.



2.3 Irudia: WAVE luzapena ehu.es webgunean

Gainera, AlInspector tresnaren aurkako jokabidea du aurkitutako arazoa zein irisgarritasun irizpideri dagokion jakiteko, hau da, 2.3 argazkian azaltzen den zerrendako informazio botoietan klikatu behar da, ondoren dagokion "Reference" atalean dagokion irizpidea ikusi ahal izateko, ikus 2.4 irudia.





#### 2.4 Irudia: WAVE irisgarritasun irizpidea ikusteko "Reference" atala

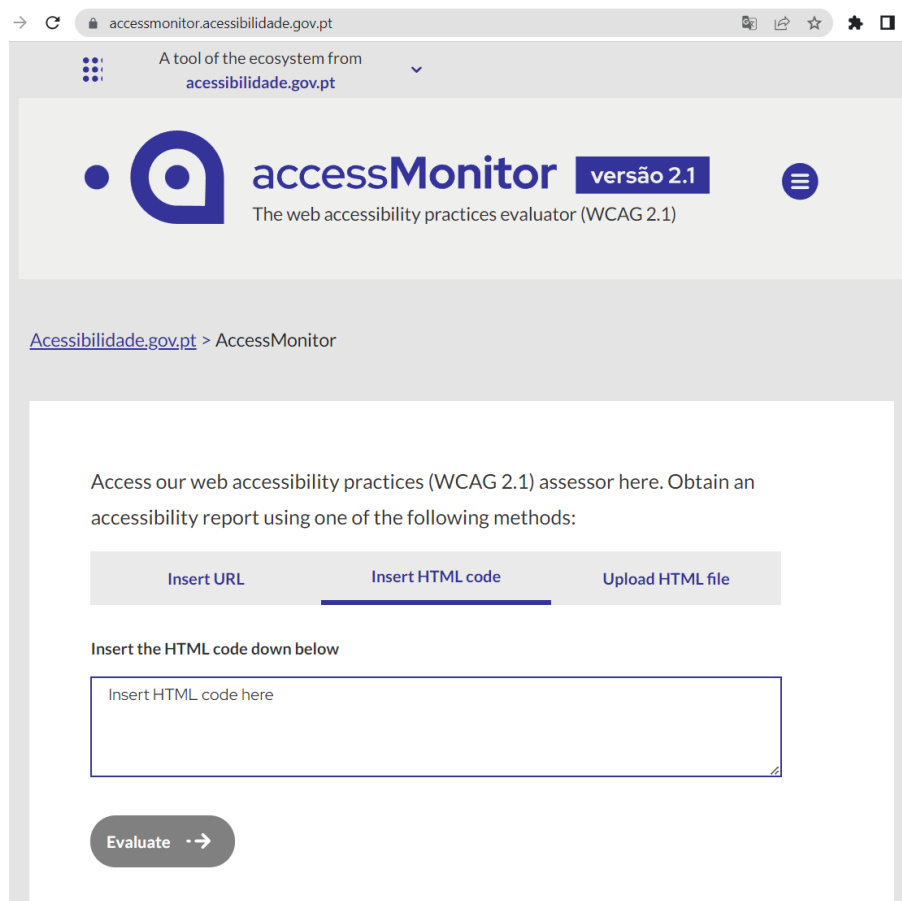
Aurreko 2.1 irudian ikusten zen bezala, AInspector-ek zuzenean WCAG irizpide multzoetan zerrendatzen ditu. Horri esker, baliteke garatzailea den erabiltzaile batentzako datuak ikuskatzeko modu erosoagoa izatea.

Horretaz aparte, WAVE tresnak baliabide gehigarri batzuk ere ditu. Adibidez, koloreen erabilera egokiagoa egin ahal izateko "Contrast eremu bat du, orrialdeen HTML-a nola egituratuta dagoen ikusteko beste eremu bat, etab. baina gure kasuan, irizpide konketuak konpontzera bideratutako horrelako funtzionalitate espezifikoetan ez gara sartuko.

## 2.2 Webguneak

Webgune motako tresnen aldetik, aipatutako luzapenen oso antzekoak diren oinarriko baliabideak eskaintzen dituzte. Horien alde garrantzitsua, ebaluatu nahi den orrialdean kokatzen ez direla da, hau da, hasierako orrialdean erabiltzaileak aztertu nahi duen orrialderako esteka eskatzen dute, eta denbora bat ebaluatzen pasa ondoren emaitzak erakusten dituzte.

Erabiltzaileak emaitzak lortzeko itxaron egin behar duenez, luzapenekiko desabantaila bat suposa dezake, baina aldi berean, ebaluazioa egin ahal izateko ez da beharrezkoa esteka bat pasatzea. Adibidez, aurreko urtean Mikelek erabili zituen AccessMonitor [8] eta AChecker [9] webguneetan, ebaluatu nahi den orrialdearen kodea zuzenean fitxategi batean edo testuz emateko aukera dute, ikus 2.5 irudia. Beste adibide bat Mauve++ [10] ebaluatzailearena izango litzateke aipatutako beste bi kasu horien aukera berak eskaintzen dituen, ikus 2.6 irudia.



**2.5 Irudia:** AccessMonitor ebaluaketa egin aurretiko orrialdea

The screenshot shows the 'Single page evaluation' interface of the Mauve++ tool. At the top, there is a navigation bar with links for 'Home', 'About', 'Info', 'Bug report', and 'Accessibility guidelines'. A 'Start New Evaluation' button and a 'Log in' link are also present. Below the navigation bar, the main heading reads 'Single page evaluation'. The interface is divided into three sections: 'Your URL', 'Your File', and 'Your Code'. The 'Your URL' section contains a text input field with the placeholder 'Enter the URL you would like to check' and a 'VALIDATE' button. The 'Your File' section is currently empty. The 'Your Code' section is also empty. Below these sections, there is a section titled 'YOUR CONTENT WILL BE VALIDATED AGAINST:'. This section includes a 'Level of conformance' dropdown menu set to 'AA', with radio buttons for 'WCAG 2.1' (selected) and 'WCAG 2.0'. There are also radio buttons for 'Static Web Page Validation (faster)' (selected) and 'Server Side Rendering Validation (slower but more precise)'. Below this, there is a 'SELECT A DEVICE' section with a 'User Agent' dropdown menu set to 'DESKTOP - Windows - Chrome'.

### 2.6 Irudia: Mauve++ orrialdearen ebaluaketa

Horretaz aparte, Mauve++ tresnak oso interesgarria den berezitasun bat du: webgune bateko orrialde bakarra ebaluatu beharrean, webguneko hainbat orrialde edota sakonera zehatz batetaraino ebaluatzeko aukerak ditu, ikus [2.7](#) eta [2.8](#) irudiak.

Agian sakonera implementatzearena gehiegizkoa izango zen, baina erabiltzaileak hainbat orrialde ebaluatzeko zehaztuz automatikoki denak batera ebaluatzeko aukera eskaintzea aproposa izango litzateke gure tresnarako.

# New Project

## Project title

Enter the project title

## Select project type

Single page  Simplified  In depth

### STANDARD

3.2. (a) The home, login, sitemap, contact, help and legal information pages

Homepage

https://

Login

https://

Sitemap

https://

Help page

https://

Add Standard URL

## 2.7 Irudia: Mauve++ adierazitako hainbat orrialde aldi berean

# New Project

## Project title

Enter the project title

## Select project type

Single page  Simplified  In depth

Enter the URL you would like to check

### PARAMETERS:

Max number of pages

1

Max depth

1

### FREQUENCY:

Repeat:

Just once

On:

Thursday

### SELECT A DEVICE

User Agent

DESKTOP - Windows - Chrome

### YOUR CONTENT WILL BE VALIDATED AGAINST:

Level of conformance

WCAG 2.1  WCAG 2.0

AA

CREATE PROJECT

## 2.8 Irudia: Mauve++ sakonera bateraino ebaluatu

Kontuan izateko beste gauza bat, mota honetako tresna batzuek aztertutako orrialdeak eta irisgarritasun irizpideei dagozkien kode zatiak erakusten ez dituztela da. Adibidez, Mauve++ tresnak "Live preview" izeneko eremu baten bitartez erakusten du, ikus 2.9 irudia, baina AChecker tresnak ez du horrelakorik, segur aski zaharkitu xamarra dagoelako.

## 2.2. Webguneak

### 2.9 Irudia: Mauve++ orrialdea grafikoki ikusteko eremua

Gainera, Mauve++ tresnaren beste alderdi interesgarri bat, ebaluazio txostena jaisteko garaian JSON-LD formatua eskaintzen duela da, JSON normala beharrean. Horri esker, irisgarritasun txostenek egitura estandarragoa dute WCAG irizpideak betetzeko garaian, hori ere kontuan izateko gauza bat izanik.

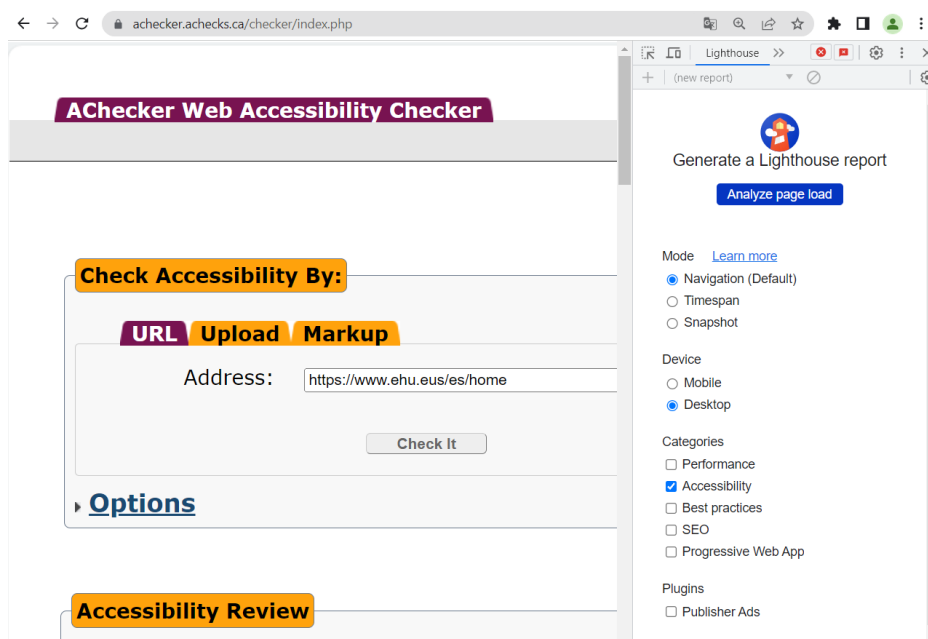
### 2.3 Liburutegiak, toolkitak, etab.

Aurrekoez aparte, tota11y [11], Sa11y [12] eta pa11y [13] bezalako "toolkit" edo liburutegi moduko tresnak ere badaude, aztertu beharreko HTML testuan bertan JavaScript fitxategien erreferentzien bitartez edo zuzenean liburutegiek eskaintzen dituzten funtzioei dei eginez erabil daitezkeenak.

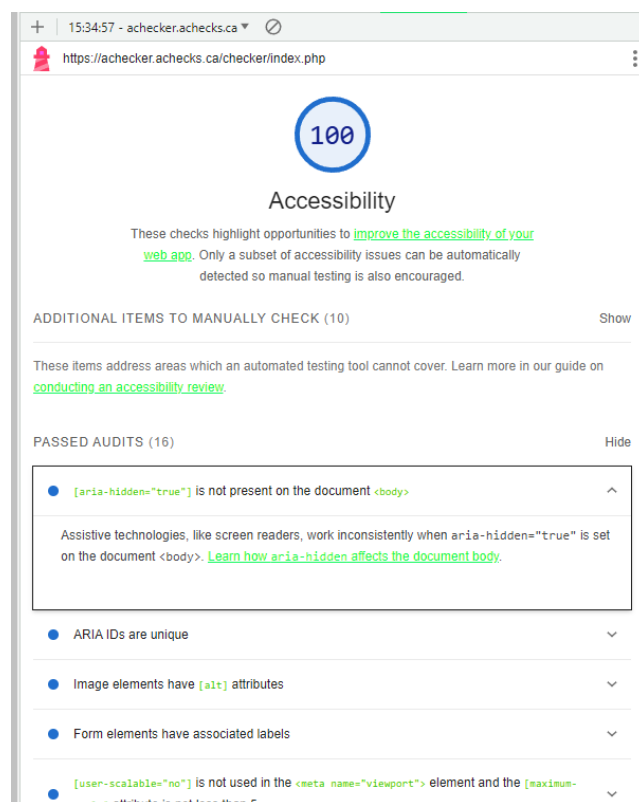
Adibidez, Tota11y tresnaren kasuan, HTMLan script bat ezarri, botoi bat agertzen da orrialdearen bazter batean, non bertan sakatzen badugu irismenari buruzko informazioa eskaintzen duen. Aldiz, Pa11y-ri dagokionez, liburutegiak dakarren funtzio bati deitu eta emaitzak JSON formatuan itzultzen ditu garatzaileak nahi duen bezala erabili ahal izateko. Hori kontuan izanda Pa11y motako liburutegiak gure tresnan erabiltzea erreza izango litzateke.

Gainera, web nabigatzaileek ere beraien barruan irisgarritasuna aztertzeko tresnak implementatuta dituzte. Firefox-ek eskaintzen duena ez da oso erabilgarria, batez ere aurretik ikusitako tresnekin alderatuta, baina Google Chromek "DevTools" delako garatzaile tresnetan implementatua duen Lighthouse [14] liburutegia oso erabilgarria da. Kasu honetan nabigatzaileak barnean dakar implementatuta, baina aparteko Node.js-ko liburutegi moduan [15] ere eskuragarri dago.

Lighthouse zuzenean Chrometik erabiltzeko, lehenik ebaluatu nahi dugun orrialdean kokatu, "Categories" atalean "Accessibility" aukeratuta dagoela ziurtatu, ikus 2.10 irudia, eta analisia egiteko botoiari eman ondoren ebaluazioa automatikoki gauzatzen du. Bukatzean, emaitzak zerrenda moduko batean erakusten ditu, aurreko tresnetan ikusi dugun antzera, arazo bakoitza konpontzeko dokumentaziorako estekak eskainiz, ikus 2.11 irudia.



2.10 Irudia: Lighthouse tresnaren hasierako interfazea



2.11 Irudia: Lighthouse irisgarritasun ebaluaketa emaitzak

## 2.4 AC-Check

Aurten garatuko dugun Earler tresna aurreko urteko AC-Check luzapenaren jarraipen bat izango denez, gutxienez aurrekariaren ezaugarri garrantzitsuenak eta izan ditzakeen mugak aipatu beharra daude.

Bere bereizgarri garrantzitsuenetarikoa, irisgarritasuna aztertzeko ebaluatzaile ezberdinen emaitzak bateratzea eta txosten bakarrean eskaintzea izango litzateke. Ebaluatzaile ezberdinen emaitzak bateratzeko funtzionalitatea ondo dago, batez ere ebaluatzaile bakoitzaren gabeziak besteek oreka ditzaketelako, baina bakarrik AChecker eta AccessMonitor aukerak eskaintzen dituzenez errealitatean ez da propietate hori guztiz aprobetxatzen. Gainera, bi aukera horiek webgune motakoak direnez, ebaluazio prozesuak liburutegiak erabili izango baziren baina nahiko motelagoak dira.

Horretaz aparte, ebaluazio txostenak jaitsi eta ondoren WCAG EM Report Tool [16] tresnara igo eta eskuz editatu ahal izateko aukera ematen du, baina aldaketa horiek norberaren fitxategian egiten dira, gainontzeko erabiltzaileekin modu errazean partekatzen baliabiderik eskaini gabe. Partekatze hori egitea interesgarria izango litzateke, adibidez, erabiltzaile batek egindako lana gainontzekoek berrerabiltzeko aukera izateko.

## 2.5 Tresna gehiago

Aipatutakoez aparte, tresna gehiago ere begiratu ditugu:

### 2.5.1 Webguneak

- [WAVE](#)
- [ACCESSIBILITY Checker](#)
- [AccessiBe](#)
- [Taw](#)

### 2.5.2 Luzapenak

- [Siteimprove](#) (Google Chrome)
- [LERA](#) (Google Chrome)
- [WAVE Accessibility Extension](#) (Firefox)
- [Tota11y](#) (Firefox)

### 2.5.3 Bestelakoak

- [Axe-core](#)
- [Tenon.io API](#)



## Plangintza

Kapitulu honetan, proiektua nola antolatu eta gauzatuko den planifikatuko dugu. Horretarako proiektuaren irismena, atazak egingo diren denboraldiak eta emangarriak bukatzeko datak, atazei emango zaizkien denboren estimazioak, proiektuko informazio eta komunikazio sistemak eta arriskuak zehaztuko dira.

### 3.1 Irismena

#### 3.1.1 Helburu zehatzak

Honakoak dira proiektua garatzerakoan helburu orokorrak bete ahal izateko kontuan izan beharreko helburu zehatzak:

- Estandarretan oinarrituz, eta ebaluatzaile ezberdinak erabiliz, webguneen irisgarritasun azterketak automatikoki egiteko tresna bat garatzea.
- Azterketekin lortutako emaitzak modu errazean ikuskatzeko eta editatzeko baliabide ezberdinak eskaintzea.
- Emaitzak ebaluazio txostenetan bilduz, horiek kudeatzeko eta erabiltzaile ezberdinen artean partekatzeko sistema garatzea.
- Garapenerako, lan munduan erabiliak diren eta ezagutzen ez ditudan tresna berriak erabiltzea, eta aurretik garatutako kodea ulertzeko eta refaktORIZATZEKO gaitasuna lortzea.

#### 3.1.2 Eskakizunak

Proiektu honek arrakasta izan dezan, sortutako emangarriek tutoreak aipatutako eskakizunak bete beharko dituzte. Hasteko, proiektu hau aurreko urtean garatutako AC-Check tresnaren jarraipena denez, hainbat ezaugarri mantendu beharko ditu:

- Earler tresna Google Chrome arakatzailerako luzapena izaten segi beharko du.

### 3. PLANGINTZA

---

- Aurreko urteko funtzionalitate guztiak mantendu beharko dira.
- W3C-ren irisgarritasun estandarretatik WCAG 2.1 estandarra izango da erabiliko dena.
- Luzapenez aparte, horren funtzionamendu egokirako zerbitzari bat izaten jarraituko da.
- Ebaluazio txostenek, WCAG EM Report Tool tresnan igotzeko eta bertan editatzeko modukoak izan beharko dute.

Mantendu beharrekoak ezaugarriez aparte honako beste ezaugarriak ere bete beharko direla zehaztu da:

- Luzapenean bertan txostenen aldaketak egin ahal izateko aukerak ematea, eta aldaketa horiek ebaluazio berriak egitean babestea editatutako datuak galdu ez daitezen. Noski, WCAG EM Report Tool tresnara igota editatzeko aukera gehiago egongo dira, baina luzapenean bertan modu sinple batean bada ere, aldaketa txikiak egiteko aukera izatea lagungarri izan daiteke.
- Erabiltzaileek erregistratzeko eta kontuan sartzeko aukera izateko. Kontuan sartutako erabiltzaileek funtzionalitate gehigarriak izango dituzte, eta ondorioz kontuak sortzeko eta horietara sartzeko oinarritzko sistema bat garatzea beharrezkoa da.
- Kontuan sartutako erabiltzaileek datu base orokor batean ebaluazio txostenak igo eta jaisteko aukerak ematea, txostenen domeinuaren araberrako bertsio kontrol sinple bat eskainiz. Funtzionalitate honi esker, erabiltzaileek beraien artean egindako lana partekatzeke eta berrerabiltzeko aukera izango dute, besteak beste.
- Frontend zatitari dagokion atala React framework-aren bitartez berriro egitea. Aurreko urtean JavaScript, HTML eta CSS erabiltzen ziren, baina gaur egun normalean framework-ak erabiltzen dira, dagoeneko sor daitezkeen behar eta arazo ezberdinentzako soluzioak emateko baliabideak eskaintzen dituztenak.
- React Node.js ingurunekoa dela kontuan izanda, ingurune bereko tresnak erabiliz zerbitzaria ere aldatzea. Tresna guztiak ingurune berekoak izanik, kasu honetan node.js, teknologia berean oinarrituko dira sistemako atal ezberdinen arteko komunikazioa erraztuz. Gainera, instalazio kontuutarako ere lagungarria izan daiteke.
- Aurreko justifikazio berarekin scraping egiteko Puppeteer tresna erabiliko da.
- Datu basearen aldetik konplexutasunik behar ez dugunez, Node.js-ko SQLite3 erabiltzearekin nahikoa izango da. Hasiera batean pare bat taula izatea espero dugu eta horretarako datu-baseak fitxategi lokal batean gordetzeko aukera ematen duen SQLite bezalako oinarritzko datu base kudeaketa sistema bat erabiltzea aproposa da.
- Luzapenak kalitate gehigarriko elementuak izatea, adibidez, konfigurazio orri baten bitartez erabiltzaileak luzapena pertsonalizatzeko aukera sinpleak eskainiz.

Bukatzeko, antzeko tresnei buruz egindako azterketan oinarrituz interesgarriak izan daitezkeen hainbat ezaugarri ikusi ditugunez, alderdi horretatik tresnak honako baldintzak bete beharko dituela erabaki dugu:

- Webgune bateko orrialde bakarria beharrean, hainbat orrialde aldi berean aztertzekeo gai izatea.
- Ebaluatzaile automatikoen artean, webgunez aparte liburutegiak ere inplementatzea, suposatzen duten azkartasun eta mantenu erraza dela eta.
- Ebaluaketa txostenen egitura Mauve++ tresnan ikusitako JSON-LD formatuaren bitartez ahalik eta gehien estandarizatzea.
- Ebaluaketa automatikoak egitean, 6 ebaluatzaile ezberdin erabiltzeko aukera izatea, 3 webgune eta beste 3 liburutegi. Aukeratu ditugun webgune motako ebaluatzaileak:
  - AChecker: agian zaharkitu xamarra geratzen ari da baina aurreko urtean erabiltzen zenez Earler-en ere mantenduko dugu. Scraping egiteko erabilitako tresna aldatzen badugu ere, aurreko urtean egindakoa moldatzea zerotik hasita baina errezagoa izan beharko luke.
  - AccessMonitor: Eguneratutako tresna da, nahiko azkarra, eta kasu honetan ere aurreko urteko AC-Check tresnan erabili zen. Beraz, AChecker aukeratu izanaren arrazoi antzekoengatik mantentzea erabaki dugu.
  - Mauve++: Ebaluaketak azkar eta WCAG 2.1 estandarraren gainean egiten ditu, eta scraping bidez beharrezko datuak eskuratzeko egitura erraza du. Eskuratu beharreko datu guztiak "Live preview" izeneko atalean zerrendatzen ditu, aurkitutako arazoen kode zatietarako erreferentziekin batera. Gainera, gure txostenen formatua zehazten dugunean, posible da esportatzeko eskaintzen dituen JSON-LD txostenak behar ditugun datu guztiak barne izatea, eta beraz scraping prozesua fitxategi hori jaitsi eta datuak bertatik eskuratzeko oinarritzea.

Aukeraturako luzapen motako irisgarritasun ebaluatzaileak:

- Pa11y: Hainbat Pa11y motako tresna daude baina gure kasuan oinarritzeko liburutegia erabiliko dugu. Beste liburutegi batzuekin konparatuz, azpitik Puppeteer erabiltzen duelako aukeratu dugu.
- Lighthouse: Google Chrome arakatzailan bertan integratuta dagoenez, interesgarria iruditu zaigu dagokion liburutegi bertsioa erabiltzea, lortutako emaitzak arakatzailan inplementatutakoarekin konparatuz ondo inplementatu dugun ikusteko. Gainera, Chrome arakatzailaren irekitako instantzia bat behar duenez, eta Puppeteer hasieratzean Chrome arakatzailako instantzia bat sortzen duenez, instantzia bera berrerrabil dezakegu.
- A11y: Kasu honetan AInspector luzapenak azpitik erabiltzen duen liburutegia duenez, A11y inplementatzea eta AInspector tresnaren emaitzekin alderatuz genezakeela iruditu zaigu.

### 3.1.3 Baztertutakoaren deskribapena

Alde batetik, erabiltzaileen interfazeak diseinatzeko garaian, prototipoak egiteaz aparte bezero edo azken erabiltzaileak izango direnek proba ditzaten egiten da, horiek eskaintzen duten feedbacka jasoz eta interfazeetan egokitzapenak eginez. Kasu honetan, diseinuan prototipo bat egingo dugu, baina proiektua aurrera doala, egokitzapen guztiak tutorea eta

### 3. PLANGINTZA

---

bion arteko irizpideak kontuan izanik egingo ditugu, ez dugu beste pertsonarik kontuan izango.

Bestetik, diseinu kontuetan hainbesteko denbora ez pasatzeko, erabilpen kasuak eta sekuentzia diagramak egiterakoan luzapenaren oinarriko funtzionalitate garrantzitsuenak bakarrik izango ditugu kontuan, hau da, aipatutako kalitate gehigarriko funtzionalitateak ez ditugu aurretik diseinatuko.

Gainera, badakigu erabiltzaileen datuak eskuartean izatean, horiek zifratuz eta teknika ezberdinak erabiliz babestu behar direla, baina gure kasuan garatuko dugun produktua ez denez bezero batentzako edo salmentan jartzeko izango, horiek alde batera utziko ditugu. Behar bada, hurrengo lan baterako aipatutakoak inplementatzea ondo egongo litzateke.

#### 3.1.4 Emangarriak

Alde batetik, EHUri eta proiektuko zuzendariari dagozkien emangarriak:

- **Memoria:** Honako proiektuaren memoria dokumentua.
- **Aurkezpena:** Proiektuaren defentsarako aurkezpena.

Bestalde, sortutako produktuaren erabiltzaile izango denarentzako:

- **Luzapena:** Earler Google Chromerako luzapena.
- **Zerbitzaria:** Luzapenaren funtzionamendurako zerbitzaria.
- **Gida:** Garatutakoa instalatzeko eta martxan jartzeko gida.

Gainera, beste garatzaile edo ikasle batek egindako tresnaren kodea aztertu nahiko balu:

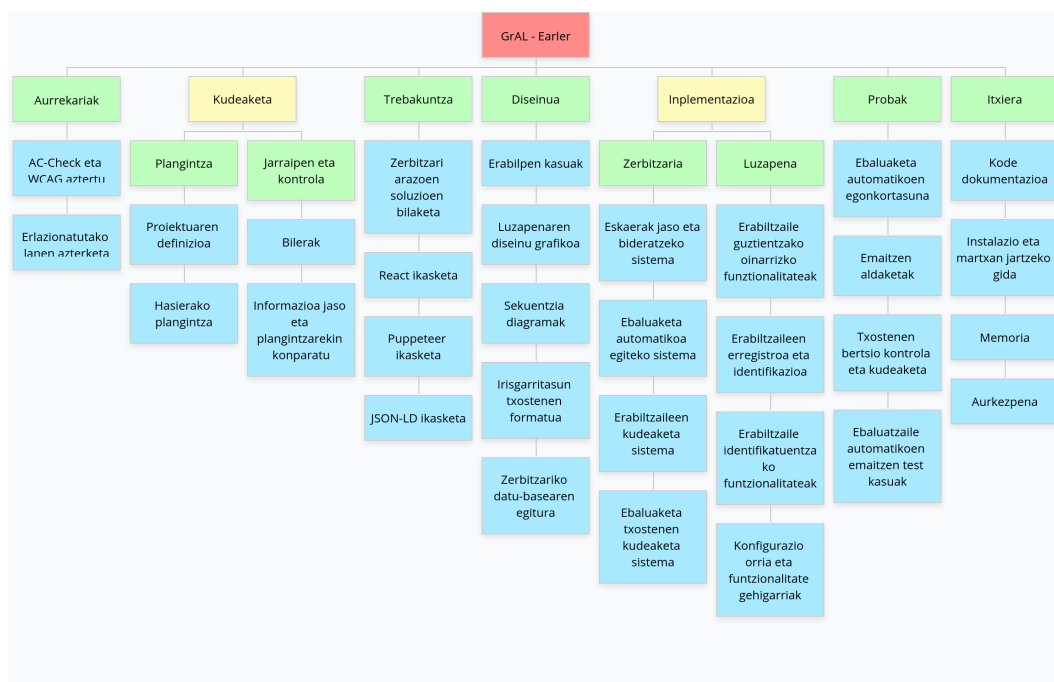
- **Kodearen dokumentazioa:** Garatutako tresnaren kodearen automatikoki sortutako dokumentazioa.

#### 3.1.5 Lan paketeak

Lan-paketeen helburua, proiektuan zehar egin beharreko lanak multzo txikiago eta maneia-garrietan banatzea da. LDE diagraman kolore berdez ikus daitekeen bezala, ikus [3.1](#) irudia, proiektu hau 9 lan-paketeetan banandu dugu, horietatik 4 binakako adarretan multzokatuak. Hona hemen, lan-pakete horien azalpenak:

1. **A - Aurrekariak:** Proiektuarekin hasi aurretik testuinguruan kokatu ahal izateko egin beharreko ikerketa eta proba lanak.
2. **PL - Plangintza:** Proiektuaren garapen egokirako hasieran ondo zehaztu beharreko plangintzaren garapena.
3. **JK - Jarraipen eta kontrola:** Lan-pakete honetan, proiektua ibilbide egokian doala ziurtatzeko lanarekin zerikusia duten atazak biltzen dira, hau da, batez ere garapena plangintzarekin bat etortzen ari den ikusteko balioko du.

### 3.2. Atazak egingo diren denboraldiak eta emangarriak bukatzeko datak



3.1 Irudia: Proiektuaren WBS/LDE diagrama

4. **T - Trebakuntza:** Proiektua garatu ahal izateko eskakizunetan aipatutako tresnen ikasketa, eta arazoaren soluzio posibleen bilaketa.
5. **D - Diseinua:** Pakete honetan, tresnaren egitura orokorra, egitura horretako elementuen arteko komunikazioa, luzapenaren diseinu grafiko, etab. nolakoak diren diseinatu eta zehaztuko dira.
6. **Z - Zerbitzaria:** Zerbitzariak luzapenaren baliabide guztiak bermatu ahal izateko eskaini beharreko funtzionalitateak garatzeko atazak izango ditu barne.
7. **L - Luzapena:** Erabilpen kasu guztiak kontuan dituen luzapenaren inplementazioa egin ahal izateko gauzatuko diren atazak biltzen ditu.
8. **PR - Probak:** Funtzionalitate konplexueneren funtzionamendu egokia eta ahalik eta egonkorrena bermatzeko zereginarekin zerikusia duten atazak.
9. **I - Itxiera:** Proiektua amaitzean dagoenean garatutako tresnaren erabilera modu egokian dokumentatzeko atazak biltzen ditu.

## 3.2 Atazak egingo diren denboraldiak eta emangarriak bukatzeko datak

### 3.2.1 Egin beharreko atazen deskribapena

LDE diagraman kolore urdinez ikus daitezkeen bezala, ikus 3.1 irudia, honakoak dira aurretik definitutako lan pakete bakoitzeko ataza guztiak eta dagozkien deskribapen laburrak:

#### 1. A - Aurrekariak

### 3. PLANGINTZA

---

- **A.ACW - AC-Check eta WCAG aztertu:** Aurreko urteko AC-Check tresna instalatzea, probatzea, kodearen nondik norakoak aztertzea, eta erabiltzen duen WCAG terminologia ulertzea.
- **A.EL - Erlazionatutako lanen azterketa:** Dagoeneko 2. kapituluan azaldutako erlazionatutako lanen azterketa egitea eta idaztea.

#### 2. PL - Plangintza

- **PL.PD - Proiektuaren definizioa:** Proiektuaren testuingurua, helburuak eta eskakizunak definitzea.
- **PL.HP - Hasierako plangintza:** Emangarriak, lan-paketeak, atazak, dedikazioak, mugarriak, diagramak, etab. definitzea.

#### 3. JK - Jarraipen eta kontrola

- **JK.B - Bilerak:** Proiektuaren bizi-zikloan zehar, tutorearekin 2 asteko tartetan bilerak egingo dira, besteak beste, ataza bakoitzerako definitutako denborak eta helburuak jarraitzen ari diren ikusteko, eta horren arabera erabakiak hartzeko.
- **JK.JK - Informazioa jaso eta plangintzarekin konparatu:** Proiektuaren garapenari buruzko informazioa jasotzea eta plangintzarekin alderatzea desbiderapenak dauden ikusteko, eta definitutako atazak betetzen ari direla ziurtatzeko.

#### 4. T - Trebakuntza

- **T.SB - Zerbitzari arazoaren soluzioaren bilaketa:** Zerbitzari berria izateak sortuko dituen arazoak nola konpondu bilatzea.
- **T.R - React ikasketa:** React framework-aren erabileraren ikasketa.
- **T.P - Puppeteer ikasketa:** Webgune motako ebaluatzaileen emaitzak jasotzeko Puppeteer scraper tresnaren erabileraren ikasketa.
- **T.LD - JSON-LD ikasketa:** Lortutako ebaluaketa emaitzak jasotzeko erabiliko den JSON-LD formatuaren berezitasunen ikasketa.

#### 5. D - Diseinua

- **D.EK - Erabilpen kasuak:** Erabiltzaileak eskuragarri izango dituen oinarritzko funtzionalitate guztiak adierazteko erabilpen kasuen diagrama egitea.
- **D.DG - Luzapenaren diseinu grafikoa:** Luzapenak grafikoki izango duen itxura zehaztea.
- **D.SD - Sekuentzia diagramak:** Erabiltzaile bakoitzak oinarritzko funtzionalitate bat erabiltzean, tresnak azpitik kodean egingo duen ibilbidea adierazteko sekuentzia diagramak diseinatzea.
- **D.TF - Irisgarritasun txostenen formatua:** Adierazitako eskakizunak betetzeko eta luzapenaren funtzionalitateak eskaini ahal izateko, irisgarritasun emaitzen txostenek izango duten egitura espezifiko zehaztea.
- **D.DB - Zerbitzariko datu-basea:** Erabiltzaile erregistratuen datuak eta horiek gordetzen dituzten txostenen bertsio kontrola egin ahal izateko sqlite3 datu baseko taulak nolakoak izango diren definitzea.

## 6. Z - Zerbitzaria

- **Z.JB - Eskaerak jaso eta bideratzeko sistema:** Zerbitzariak, eskaerak jaso, horiek kudeatu, eta emaitzak itzultzeko sistema garatzea.
- **Z.EA - Ebaluaketa automatikoa egiteko sistema:** Zerbitzariak aurretik aipatutako ebaluatzaileen ebaluaketak egin eta emaitzak jasotzeko inplementazioa.
- **Z.EK - Erabiltzaileen kudeaketa sistema:** Erabiltzaileen erregistratzeko eta kontuan sartzeko eskaerak jasotzean, dagokion datu baseko taulan kudeaketak egiteko sistema sortzea.
- **Z.TK - Ebaluaketa txostenen kudeaketa sistema:** Erabiltzaileek pasatako txostenak edota eskatutakoak berreskuratzeko eskaerak kudeatzeko sistema garatzea.

## 7. L - Luzapena

- **L.OF - Erabiltzaile guztienezako oinarrizko funtzionalitateak:** Ataza honen, ebaluaketa automatikoa egiteko, definitutako 6 ebaluatzailearen artean aukeratzeko, webguneko zein orrialde ebaluatu adierazteko, emaitzak taulan zerrendatu eta klikatuz orrialdean ikusteko, txostenak deskargatzeko, eta txostenak fitxategi baten bidez igotzeko aukerak biltzen ditu.
- **L.EI - Erabiltzaileen erregistroa eta identifikazioa:** Erabiltzaileek beraien datuak emanda erregistratzeko eta dagozkien kontuetan sartzeko aukera izateko sistema garatzea.
- **L.IF - Erabiltzaile identifikatuentzako funtzionalitate gehigarriak:** Kontuan sartutako erabiltzaileek funtzionalitate gehigarriak izango dituzte. Txostena zerbitzariko datu base zentralizatu batean gorde ahal izango dute, datu-basean gordetako txosten guztiak ikusteko eta bat aukeratuz luzapenean kargatu ahal izango dute, eta txostenak beraien izenean editatu ahal izango dituzte.
- **L.KG - Konfigurazio orria eta funtzionalitate gehigarriak:** Aipatu bezala, luzapenak konfigurazio orri bat izango du, eta bertan garapenean zehar interesgarriak eta gehiegizko konplexutasuna ez duten funtzionalitate gehigarriak eskainiko ditu. Horretaz aparte baliteke luzapenerako interesgarriak diren funtzionalitate gehigarriak ere garatzea.

## 8. PR - Probak

- **PR.EA - Ebaluaketa automatikoen egonkortasuna:** Ebaluaketa automatikoen emaitzak egonkorak diren eta ebaluaketa prozesuan erroreak gertatzen diren egiaztatzea.
- **PR.AE - Emaitzen aldaketak:** Emaitzen aldaketak egitean, txostenen ondo gordetzen direla, eta aldatutako emaitzak ebaluaketa berriak egin edo txosten berriak kargatzean berridazten ez direla egiaztatzea.
- **PR.BK - Txostenen bertsio kontrola eta kudeaketa:** Txostenak ondo gorde eta kargatzen direla egiaztatzea.
- **PR.TK - Ebaluaketa automatikoekin lortutako emaitzen test kasuak:** Ebaluatzaile automatikoek W3C-ren ACT test kasuak zein neurritan betetzen dituzten egiaztatu eta ondorioak ateratzea.

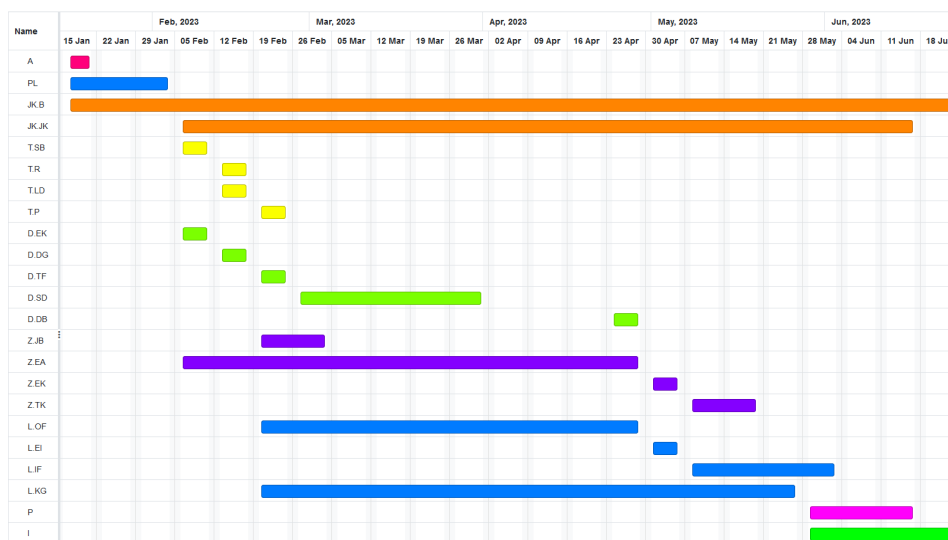
9. I - Itxiera

- **I.KD - Kodearen dokumentazioa:** Garatutako kodea ChatGPT bidez dokumentatzea eta bukaeran dokumentazio guztia biltzen duen HTML orrialde bat automatikoki sortzea.
- **I.G - Instalazio eta martxan jartzeko gida:** Tresna edonork erabili ahal izateko gida labur bat idaztea.
- **I.M - Memoria:** Honako dokumentu hau bukatzea.
- **I.A - Aurkezpena:** Proiektuaren defentsa egiteko aurkezpena.

3.2.2 Atazen garapen denboraldiak

Atazak garatzeko esleitu diren denboraldiak ondorengo 3.2 irudiko Gantt diagraman ikus daitezke. Honakoak dira diagrama egiterakoan kontuan izandako puntu nagusiak:

- Erabiltzaileen erregistroa eta dagozkien funtzionalitateak beranduago egin behar izango direla suposatzen denez, dagozkien atazak, datu-basearen diseinua barne, proiektuaren bukaeratik hurbil kokatu dira.
- Diseinurik edo tresna berezirik behar ez denez liburutegi motako ebaluatzaileen lehen emaitzak lortzen hasi ahal izateko, ebaluatzaile automatikoak proiektuaren hasieratik implementatzen hastea erabaki dugu, pixkanaka egiten joateko eta proiektuaren garapenean aukera gehiago izateko.
- Ekaina aldera lanean hastea pentsatzen dudanez, ordurako implementazioa bukatuta izateko moduan egituratu ditut atazak. Modu horretara, desbiderapen garrantzitsurik gertatzen ez bada, memoria idaztea eta probak egitea besterik ez zaizkit geratuko.
- Garrantzitsua da egindako lana ekainak 25 egunerako ADDI plataformara igota egon beharko duela kontuan izatea. Beraz, proiektuak ordurako bukatuta egon beharko duela kontuan izan dut.



3.2 Irudia: Hasierako Gantt diagrama



### 3.3 Ataza bakoitzari emango zaion denboraren estimazioak

Ataza bakoitzari dedikatuko zaion orduen estimazioa ondorengo 3.1 irudiko taulan ikus daiteke. Kontuan izandako gauza nagusiak:

- Zerbitzariko ebaluatzaile automatikoen inplementazioari dagokionez, seguruenik webgune motakoek liburutegiek baina dedikazio gehiago dute. Scraping egiteak arazo asko eman ditzake, eta A11y liburutegiaren kasuan ere, barrutik nola inplementatuta dagoen ere ikusi beharko dut.

Gainera, AccessMonitor eta AChecker tresnen scraper programa eginak daude baina segur aski horretarako tresna berri bat erabiltzen ari naizenez, aldaketa asko egin beharko ditut. Beraz, nahiz eta liburutegien kasuan seguruenik denbora askoz gutxiago beharko dudana, batz bestea ebaluatzaile bakoitzeko 10 ordu inguru estimatzea pentsatu dut.

Diseinatutako JSON-LD txostenak sortzeko programa egiteak, eta ebaluatzaile ezberdinak eta aukeratutako webguneko orrialdeak paraleloan ebaluatu daitezten egiteak seguruenik kopuru nabarmena beharko du. Beraz, nahiz eta seguruenik desbiderape-nak egongo diren, dena 75 orduko estimazioa esleitu diot atazari.

- Luzapeneko oinarrizko funtzionalitateei dagokionez, txostenetako datuak taulan ikusi ahal izateko mapaketa egitea, zerrendako elementuetan klikatzean web orrialdean nabarmentzea, kokatutako orrialdearen arabera emaitzak erakustea, etab. denbora nahiko handia behar izatea espero dudanez, 45 orduko dedikazioa ezarri diot atazari. Egia esan ez dakit oso egokia den, ez dakidalako zein puntutaraino gai izango naizen aurreko urteko kodea React bidez berreraabiltzeko.
- Identifikatutako erabiltzaileentzako funtzionalitateak inplementatzeak kostu nahiko handia izatea ere espero dut, batez ere datuak taulatik eraldatzeko egituraketa egoki bat eta datu horiek maneiatzeak konplexutasun handia izango duelako.

Orokorrean ez nago guztiz ziur inplementazioan beharko dudana dedikazioaz, batez ere erabili ez ditudan hainbat tresna erabili beharko ditudalako. Beraz, seguruenik desbidera-pen garrantzitsuak egongo dira inplementazioari dagokion dedikazioaren estimazioan eta errealitatean egingo denaren artean.

### 3. PLANGINTZA

---

ATAZA	ESTIMATUTAKO ORDUAK
A - Aurrekariak	13
ACW	7
EL	6
PL - Plangintza	15
PD	3
HP	12
JK - Jarraipen eta kontrola	10
B	8
JK	2
T - Trebakuntza	16
SB	8
R	4
P	2
LD	2
D - Diseinua	20
EK	2
DG	2
SD	10
TF	4
DB	2
Z - Zerbitzaria	80
JB	2
EA	75
EK	1
TK	2
L - Luzapena	75
OF	45
EI	2
IF	20
KG	8
PR - Probak	28
EA	4
AE	4
BK	4
TK	16
I - Itxiera	34
KD	6
G	3
M	20
A	5
<b>Denera:</b>	<b>291</b>

**3.1 Taula:** Estimaturako ataza bakoitzeko dedikazioa

## 3.4 Proiektuko informazio eta komunikazio sistemak

### 3.4.1 Informazio sistema

Alde batetik, proiektuan zehar sortutako kodea eta egindako aldaketa guztiak GitHub plataformara igo eta bertan kudeatuko dira.

Bestetik, Google Drive zerbitzuko karpeta bat izango dut, bileren datak eta iraupenak apuntatzeko, atazei dedikatutako denborak ezartzeko, memoriaren idazketa aurreratzen joateko dokumentuak idazteko, bileretan aipatutako estekak eta erreferentziak gordetzeko, etab.

Bukatzeko, memoriaren idazketa Overleaf tresnaren bidez egingo dut, egindako lana periodikoki jaitsiz backup moduan izateko.

### 3.4.2 Komunikazio sistema

Tutorearekin bilera periodikoak gauzatuko dira 2 astez behin ostegun goizetako 11:30tan. Edozer gauza dela eta bilera horietako bat gauzatzeko aukerarik egongo ez balitz, posta elektronikoko bitartez beste egun eta ordu bat finkatuko litzateke. Besterik ez bada adierazten, bilerak Webex bidez egingo dira, egindakoa ordenagailuko pantaila partekatuz erakutsiz. Edozein zalantza argitzeko bilera horiek erabiliko dira, eta nahikoa ez balitz posta elektronikoz idazteko aukera ere egongo litzateke.

## 3.5 Arriskuak

- **A1:** Arrisku garrantzitsu bat, lanerako erabiltzen den gailua matxuratu edo galtzea-gatik, garatutako kodea, jasotako informazioa edo memoria bera galtzeko arriskua izango litzateke. Arrisku honen eragina oso handia izan daiteke, baina aldi berean saihesteko oso erraza da. Nire kasuan, kode guztia GitHub plataforman dagoenez, arriskua gertatuko balitz, plataformara igotako azkenengo bertsiotik aurrera egindako aldaketak galduko lirateke bakarrik, eta beraz aldaketak denbora tarte txikietan eginez galdutako aurrerapenak ez lirateke handiak izango. Gainera, memoria eta gainontzeko informazioak online plataformetan ditudanez, ez dute erabiltzen dudan gailuaren menpekotasunik eta beraz ez daude arrisku honen menpean.
- **A2:** Scraping egiteak arrisku handi bat suposatzen du, datuak lortzeko erabiltzen den orrialdean aldaketak eginez gero, hau da, HTML orriaren egitura edo elementuen class eta id atributuak aldatzen badira, scraping egiteko egindako programak funtzionatzen uzteko probabilitatea handia dago. Arrisku hau ekiditeko ezin daiteke ezer asko egin, datuak eskuratzeko erabiltzen den moduaren arabera egonkortasun gehiago lor daiteke, baina ezin daiteke inolako bermerik izan. Agian, scraping egiteko kodea txukun eta ondo dokumentatua izateak lagun dezake arriskua gertatzen denean ahalik eta azkarren konpontzeko gai izateko.
- **A3:** Beste arrisku bat gainontzeko irakasgaiekin suposatu baina denbora gehiago behar izatea da. Hau da, proiektuarekin batera irakasgai gehiago ere kurtsatzen ari nahiz eta horiek aurrera eramateko pentsatu baina denbora gehiago beharko banu, proiektuari dedikatzeke izango nukeen denbora murriztuko litzateke. Dena dela, nire kasuan 2 irakasgai bakarrik dira, eta zailtasun aldetik errazak kontsidera daitezke,

### 3. PLANGINTZA

---

hori dela eta, arrisku hau gertatzeko aukerak txikiak dira, eta gertatzera joango balitz ere, kurtsuan zehar irakasgaietan dudan errendimendua ikustea edukiko nuke horren arabera proiektuaren plangintza aldatu eta egoerara egokitu.

- **A4:** Gainera, beste arrisku bat teknologia ezberdin asko eta niretzako berriak direnak erabili beharko ditudala izan daiteke. Arrisku horrek bi alderdi ditu: Alde batetik, niretzako tresna berriak izateak, estimatutako dedikazio denborak ondo finkatzea zaila egiten du, eta ondorioz desbiderapen garrantzitsuak egon daitezke. Bestetik, hainbat teknologia direnez beraien artean gatazkak sortzea, edota zuzenean baten batek erroreak sortzea gerta daiteke.

Prebentzio moduan, proiektuarekin hasi aurretik trebakuntza atalean erabiliko ditudan tresnak nola erabiltzen diren ikasiz, egindako denbora estimazioak neurri batean zuzentzeko aukera izango dut. Horretaz aparte, erabiliko ditudan tresnak ingurune berekoak izango direnez eta dokumentazio aldetik orokorrean aberatsak direnez, ez dut uste beraien arteko gatazkak egongo direnik, eta erroreak sortzen direnean ez litzateke dokumentazioaren bitartez arazoak konpontzeko zailtasun handiegirik egon beharrik.

# Trebakuntza

Kapitulu honetan, formakuntza lan-paketeko atazen gauzatzeari buruz hitz egingo da, batez ere, zerbitzari arazoaren soluzioen bilaketan sakonduz.

## 4.1 Zerbitzari berria

Luzapenaren funtzionalitateak eskaini ahal izateko derrigorrezkoa da zerbitzari bat izatea. Horretarako, aurreko urtean Flask [17] frameworka erabili zen, baina aurtengoan, React-ek Node.js [18] erabiltzen duela aprobetxatuz, zerbitzariak ingurune horretako tresnak erabiliz lan egin beharko du. Ondorengo azpiataletan zerbitzaria nola antolatuko den eta zein tresna erabiliko dituen jakiteko egindako azterketa azalduko da.

### 4.1.1 Luzapenerako argazkiak

Zerbitzariaren funtzionalitate bat luzapenarentzako argazkiak lortzea zen. Arazo horri soluzio emateko, Google Chrome arakatzailleak eskaintzen duen `web_accessible_resources` [19] aukera aurkitu dut. Tresna hori luzapenek erabiltzeko pentsatuta dago, eta ondorioz inplementatzea nahiko erraza da.

Egin behar den bakarra, luzapeneko `manifest.json` konfigurazio fitxategian **`web_accessible_resources`** aukera txertatzea, eta barnean erabili beharreko argazkiak edota fitxategiak adieraztea da, ikus 4.1 irudia. Horri esker, irudien path edo helbideak lortu ahal izateko, `chrome.runtime.getURL("irudia");` funtzioa erabiltzearekin nahikoa da.

```
"web_accessible_resources": [
  {
    "resources": ["/images/icon128.png", "/images/contractedArrow.png", "/images/contractedArrow.png"],
    "matches": ["<all_urls>"]
  }
]
```

4.1 Irudia: `manifest.json` konfigurazio fitxategian luzapeneko argazkiak adierazteko

### 4.1.2 CORS arazoak

AC-Checken zerbitzarian Flask-CORS [20] bitartez CORS [21] arazoak ekiditen ziren, eta aurtan hasiera batean soluzio azkar moduan proxy bat erabiltzea pentsatu dugu. CORS bitartez, zerbitzarira egindako eskaeren jatorrizko IP eta burukoaren arabera, eskaera horiek blokeatu edo ez erabakitzen da. Proxy bat erabiliz IP-ak pertsona errealean simulatzea lor genezake, baina oraindik buruko egokiak jartzeak arazoak ekar ditzake.

Gainera, gida pare bat gainetik begiratu ondoren [22, 23], HTTPS protokolorako proxy zerbitzaria egitea konplexua izan daitekeela ikusi dut. Hori dela eta, dagoeneko sortutako liburutegi bat bilatzeko prozesuan, gure beharretara egokitzen den Node.js-ko cors-anywhere [24] liburutegiaren aipamena [25] aurkitu dut.

Tresna hori, Node.js bidez exekuta daiteke, defektuz localhost:8080 helbidetik pasatzen diren eskaera guztien tartekari edo proxy zerbitzari baten funtzioa eginez. Gainera, eskaeren burukoak aldatzen ditu CORS sistemak blokeatu ez ditzan, eta erabiltzeko egin beharreko gauza bakarra, eskaera jasoko duen helbidea `http://localhost:8080/` ondoren jartzea da. Adibidez: `http://localhost:8080/https://mauve.isti.cnr.it/singleValidation.jsp`.

### 4.1.3 Zerbitzariko eskaerak nola jaso eta kudeatu aurkitu

Aurrekoaz aparte, zerbitzarira eskaerak nola egin, eta zerbitzariak eskaera horiek nola jaso eta kudeatu ditzan ere jakitea beharrezkoa da. Eskaerak jaso eta kudeatzeko, Node.js-ko http liburutegiaren gida bat [26] aurkitu dut, eta JavaScript bidez zerbitzarira eskaerak egiteko, tutoreak esan bezala, fetch tresna erabil dezakedala ikusi dut. Fetch erabiltzen ikasteko gida honetaz baliatu naiz: [27].

## 4.2 React, JSON-LD eta Puppeteer ikasten

Hasteko, aurreko urtean erabiltzailearen interfaze grafikoaren zati guztia JavaScript, HTML eta CSS erabiliz egin zen, baina gaur egun normalean tresna ahaltuagoak erabiltzen dira. Hori dela eta, JavaScripten oinarritutako React framework-a [28] erabiltzea erabaki dugu. React ikasteko, dagokion dokumentazioa begiratzeaz aparte, bi gida jarraitu ditut: lehena erabilera orokorra ikasteko [29], eta bigarren React bidez luzapena nola egin ikasteko [30].

Bestalde, JSON formatua aurretik ezaguna eta erabilia nuen, baina JSON-LD buruz ez nekien ezer. Hori dela eta, formatuari buruzko ideia orokor bat egiteko, dagokien webguneko dokumentazio atalera jo dut [31], eta "Introductory Material" atalean, dagozkion nondik norakoak azaltzen dituzten hainbat bideo topatu eta ikusi ditut.

Bukatzeko, ez da nahikoa web ebaluatzaileen HTML kodea zuzenean fetch bidez atzitzea, webgune motako ebaluatzaile guztiek URL bat idaztea, botoi bat klikatzea, datuak kargatu bitartean itxarotea, etab. behar dutelako. Hori dela eta, scraping teknikaz baliatzea beharrezkoa da eta aurreko urtean, Flask zerbitzaritik scraping egiteko Selenium [32] eta BeautifulSoup Soup [33] tresnak erabili ziren.

Aurtan, Node.js ingurunea aprobeztatuz, Node.js-ko Puppeteer [34] scraperra erabiltzea erabaki dugu, eta ondorioz tresna berriarekin scraping nola egiten den ikasten hasteko, dokumentazioari begirada bat eman eta tutorial bat segi dut [35].

## Diseinua

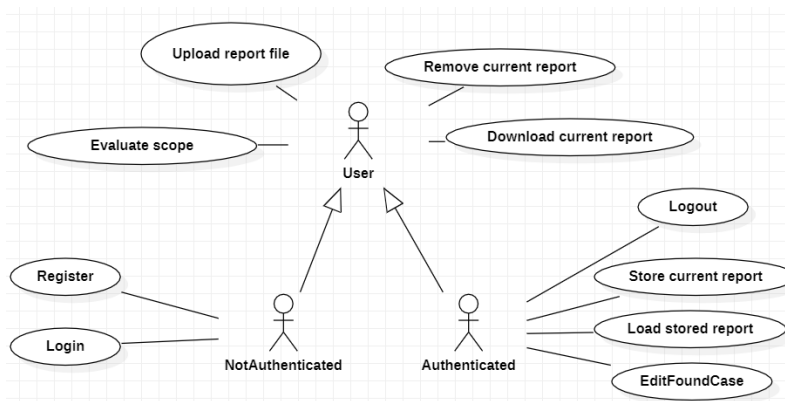
Software produktu edo tresna bat garatzerako garaian, kodea idazten hasi aurretik hainbat gauza diseinatu behar dira: produktuak eskaini beharreko funtzionalitateak, izango duen interfaze grafikoa, datu-basearen egitura, etab. Hori dela eta, kapitulu honetan ondorengo 4 atalak izango ditugu:

- **Erabilpen kasuen diagrama 5.1:** Earler tresnak izango dituen oinarrizko erabilpen kasuak eta horiek erabiliko dituzten aktore ezberdinak biltzen dituen erabilpen kasuen diagramaren definizioa eta azalpenak izango ditu.
- **Datu-basearen diseinua 5.2:** Funtzionalitate batzuetarako sqlite3-ko datu-base bat beharko dugu, eta atal honen bitartez datu-base horretako taulek izango duten egitura zehaztuko dugu.
- **Erabilpen kasuen sekuentzia diagramak 5.3:** Aipatutako erabilpen kasuak nola inplementatuko diren eta kodeko zati ezberdinen artean egongo diren elkarrekintzak adierazteko sekuentzia diagramen diseinuak erakutsi eta azalduko ditugu.
- **Luzapenaren diseinu grafikoa 5.4:** Atal honetan erabiltzaileak interfazea nola ikusiko duen zehaztuko dugu, hau da, luzapenak grafikoki izango duen egitura, funtzionalitate mota ezberdinak nola erakutsiko diren, osagai ezberdinek izango dituzten koloreak, tamainak, etab.
- **Irisgarritasun txostenen formatua 5.5:** Irisgarritasun azterketekin sortutako datuak guri interesatzen zaigun moduan gorde behar direnez, atal honetan horretarako sortuko ditugun txostenak nolako egitura izango duten definituko dugu.

Kontuan izan erabilpen kasuen aldetik, oinarrizko funtzionalitateak bakarrik izan ditugula kontuan, bestela garatzea pentsatzen ditugun kasu guztientzako sekuentzia diagramak egin beharko bagenitu denbora gehiegi behar izango genuke.

## 5.1 Erabilpen kasuen diagrama

Erabilpen kasuen diagramaren bitartez, software produktu batek dituen erabiltzaile mota ezberdinak (aktoreak) eta horiek erabili ditzaketen funtzionalitateak (erabilpen kasuak) grafikoki modu errazean adierazten dira. Honakoa da Earler proiektuko erabilpen kasuen diagrama:



5.1 Irudia: Erabilpen kasuen diagrama

### 5.1.1 Aktoreak

- **User** izeneko aktoreak erabiltzaile guztiak erabili ahal izango dituzten erabilpen kasuak izango ditu, eta **NotAuthenticated** eta **Authenticated** aktoreek **User**-en erabilpen kasuak heredatuko dituzte.
- **NotAuthenticated** aktoreak kontuan sartu gabeko edozein erabiltzaile errepresentatzen du, hau da, ez du zertan datu basean erregistratuta egon beharrik.
- **Authenticated** aktoreak kontuan sartutako erabiltzaileak errepresentatzen ditu, hau da, datu basean erregistratuta egonik dagozkien kontuetan sartuta dauden erabiltzaileak.

### 5.1.2 Erabilpen kasuak

- **Login:** Dagoeneko erregistratuta dagoen erabiltzailea kontuan sartzeko.
- **Register:** Erabiltzaile berri bat erregistratzeko.
- **Logout:** Erabiltzailea kontutik ateratzeko.
- **Evaluate scope:** Erabiltzaileak hautatutako ebaluatzaile automatikoekin eta adierazitako orrialdeak kontuan izanik, ebaluaketa automatikoa egin, emaitzak lortu eta gordetzeko prozesua.
- **Remove current report:** Unean luzapenean kargatuta dagoen txostena ezabatzeko aukera.
- **Download current report:** Unean kargatuta dagoen txostena deskargatzeko aukera.



- **Upload report file:** Fitxategi bateko ebaluaketa txostena kargatzeko aukera.
- **Store current report:** Unean kargatuta dagoen ebaluaketa txostena zerbitzariko datu basean gordetzeko aukera.
- **Load stored report:** Zerbitzarian unean kokatutako gunea ebaluatzen duten txostenak ikusi eta aukeratutako kargatu ahal izateko funtzionalitatea.
- **Edit found cases:** Emaitzen taulako irisgarritasun irizpide zehatz bateko aurkitutako emaitzak aldatzeko funtzionalitatea.

## 5.2 Datu-basearen diseinua

Normalean datu-base bat diseinatzeko garaian, grafiko mota ezberdinak egiten dira taulak izango dituzten datuak eta gainontzeko taulekiko erlazioak ondo zehazteko, optimizazioa egiteko, etab. Gure kasuan, sortuko dugun datu-basea oso txikia eta konplexutasun gutxikoa denez, ez dugu horrelakorik egingo.

Bi taula besterik ez ditugu izango, bata erabiltzaileak eta bestea ebaluaketa txostenak gordetzeko.

Erabiltzaileen aldetik ez dut ezer berezirik behar izango, beraz, 5.2 irudian ikus daitekeen bezala sortuko dut, email atributuaren balioen arabera erabiltzaileak identifikatuz.

```
CREATE TABLE IF NOT EXISTS users (  
  email TEXT PRIMARY KEY,  
  username TEXT NOT NULL,  
  password TEXT NOT NULL  
)
```

5.2 Irudia: Erabiltzaileak gordetzeko taula

Txostenak gorde eta kudeatzeko, 5.3 irudiko reports izeneko taula sortu dut. Kasu honetan erabiltzaileen taularekin baina azalpen gutxi batzuk gehiago eman beharko ditut:

Alde batetik, txosten bakoitzari identifikatzaile bat automatikoki esleituko zaio oinarriko gako moduan, ondoren parentId gako arrotzarekin erreferentzia egiteko. Erreferentzia horrek, txosten hori sortzeko zein txostenetatik abiatu den adierazten du. Ez badu inongo gurasorik parentId NULL izango da. Gurasorik izan badu, txosten berriaren bertsioa gurasoarena + 1 izatea da ideia.

Bestalde, domain atributuak txostenak zein domeinuko orrialdeak ebaluatu dituen gordeko du, modu horretara, luzapena kokatutako domeinua ebaluatu duten txostenak modu errazean lortuko ditut.

Azkenik, uploadedBy atributuaren balioak txostena igo duen erabiltzailearen username balioa adieraziko du.

```
CREATE TABLE IF NOT EXISTS reports (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  domain TEXT NOT NULL,  
  version INTEGER NOT NULL,  
  uploadedBy TEXT NOT NULL,  
  report TEXT NOT NULL,  
  parentId INTEGER,  
  FOREIGN KEY (parentId) REFERENCES reports(id)  
);
```

5.3 Irudia: Txostenak gordetzeko taula

## 5.3 Erabilpen kasuen sekuentzia diagramak

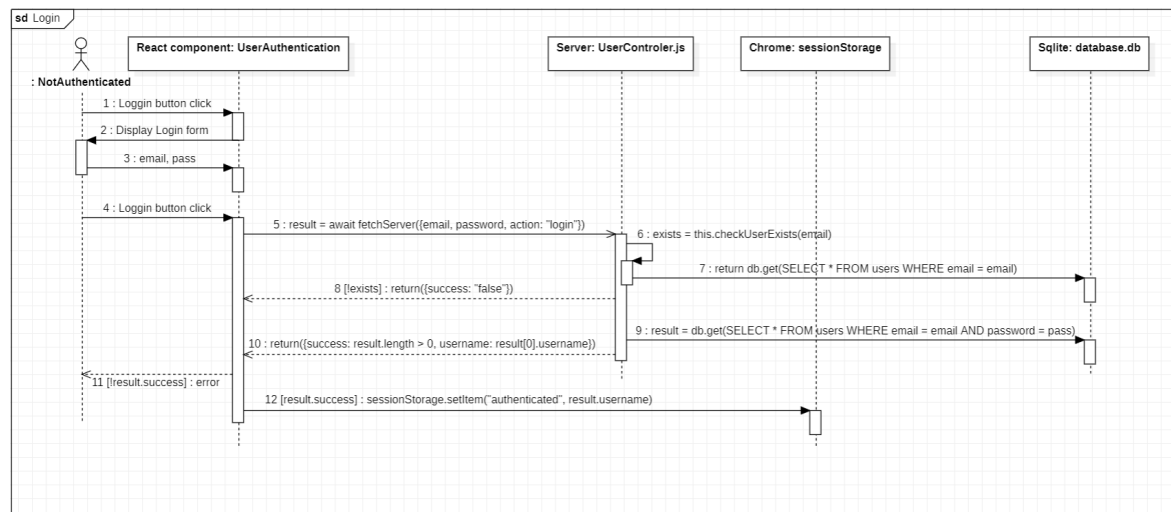
Atal honetan aurretik aipatutako erabilpen kasuen sekuentzia diagramak erakutsi eta gainetik azalduko ditut, aktoreek, luzapenak eta zerbitzariak erabilpen kasu bakoitzean egiten duten elkarrekintzaren ideia orokor bat egin ahal izateko. Kontuan izan error jartzen dudanean errealitatean window.alert bidez erabiltzaileari feedback-a ematea adierazten dudala.

### 5.3.1 Login

Hasieran, erabiltzailea kontura sartzeko login formulariora joaten da eta datuak sartu ondoren kontura sartzeko botoiari ematen dio. Luzapenak login motako eskaera asinkronoa egiten dio zerbitzariari eta itxaroten geratzen da.

Ondoren, zerbitzariak erabiltzailea existitzen dela ikusten bada aurrera segitzen du eta erabiltzailearen pasahitza egokia den jakiteko kontsulta egiten du, bide batez erabiltzailearen username balioa eskatuz eta emaitza luzapenera itzuliz.

Azkenik, emaitza egokia bada luzapenak erabiltzailearen izena gordetzen du kontuan sartuta dagoela adieraziz, eta bestela errorea ematen du.



5.4 Irudia: Login sekuentzia diagrama

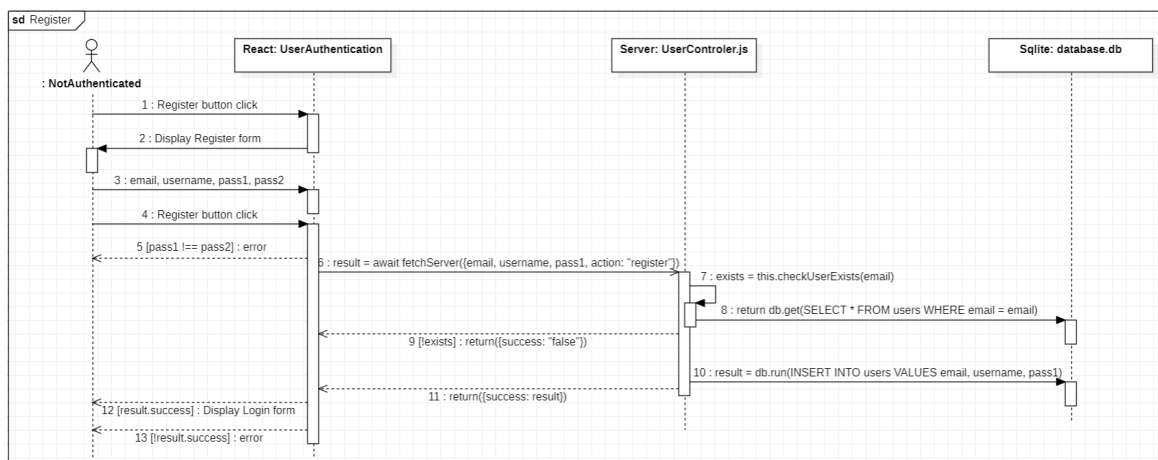
### 5.3.2 Register

Lehenik, erabiltzailea erregistratzeko register formulariora joaten da eta datuak sartu ondoren kontura sartzeko botoiari ematen dio. Pasahitzak berdinak badira luzapenak register motako eskaera asinkronoa egiten dio zerbitzariari eta itxaroten geratzen da.

Ondoren, zerbitzariak adierazitako email bera duen erabiltzailerik ez dela existitzen egiaztatzen du, eta horrela bada erabiltzaile berria users taulan txertatu eta emaitza itzultzen du.

Azkenik, emaitza egokia bada luzapenak login formularioa kargatzen du eta bestela errorea ematen du.

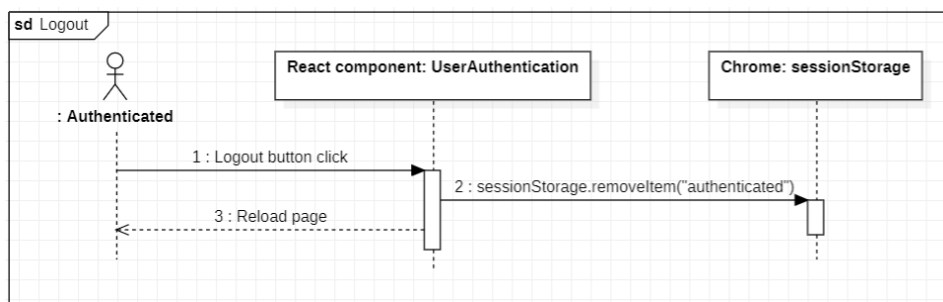
## 5. DISEINUA



5.5 Irudia: Register sekuentzia diagrama

### 5.3.3 Logout

Kontuan sartutako erabiltzaileak kontutik ateratzeko logout botoiari ematen dio, eta luzapenak erabiltzailearen gordetako username balioa ezabatu eta kokatutako orrialdea birkargatzen du.



5.6 Irudia: Logout sekuentzia diagrama

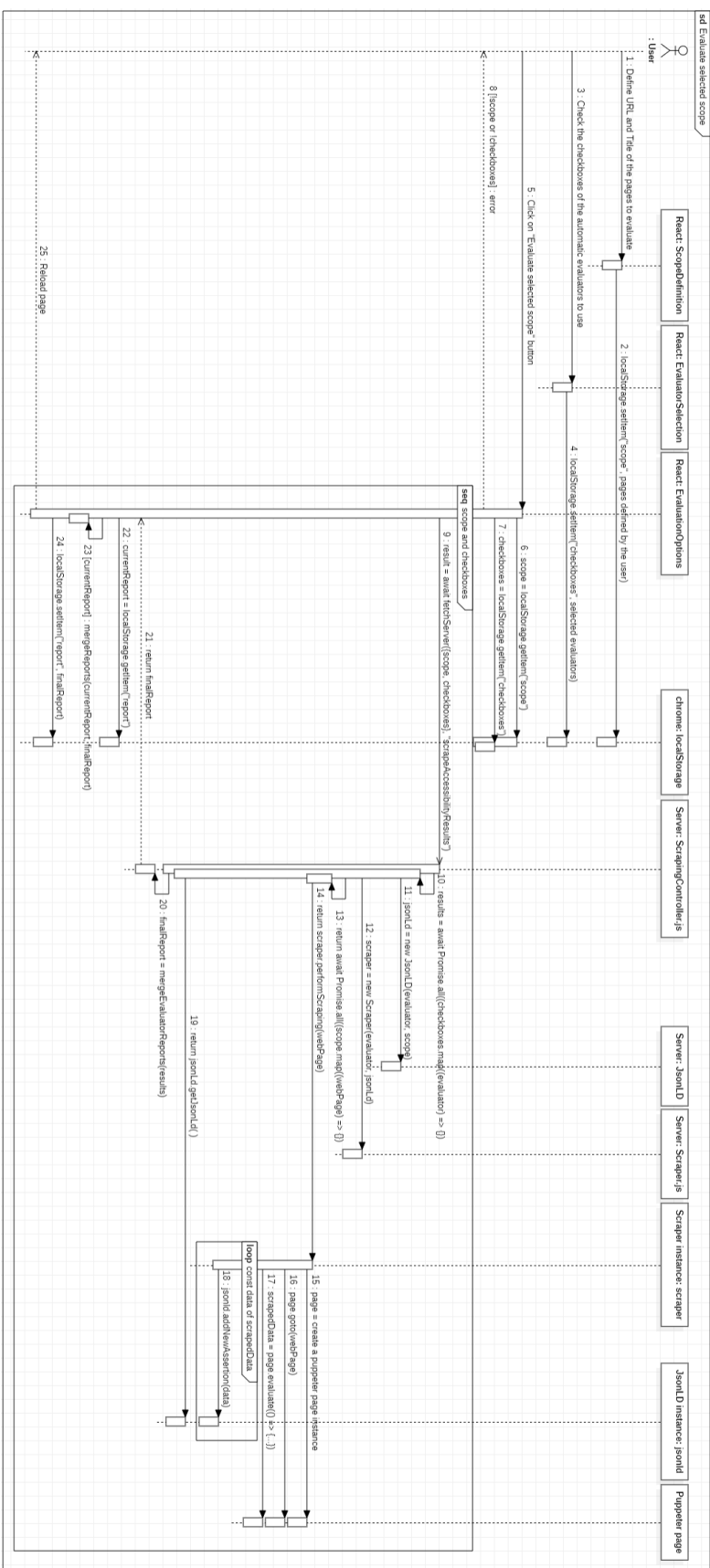
#### 5.3.4 Evaluate selected scope

Hasteko, erabiltzaileak ebaluatu nahi dituen web orrialdeen estekak finkatu eta erabili nahi dituen ebaluatzaile automatikoak aukeratu ondoren, ebaluaketa automatikoa martxan jartzeko botoiari ematen dio eta luzapenak zerbitzarira ebaluaketa automatikoa egiteko eskaera asinkronoa egiten du.

Ondoren, zerbitzariak lehenengo Promise.all bat erabiltzen du aukeratutako ebaluatzaileak paraleloan erabili ahal izateko eta bakoitzarentzako jsonLd txostenak sortzeko klasearen eta scraping egiteko klasearen instantziak sortzen ditu. Instantziak sortuta, ebaluatzaile bakoitzeko, ebaluatu beharreko orrialde bakoitza paraleloan ebaluatzeko bigarren Promise.all bat erabiltzen du, bakoitzarentzako scraping prozesua hasi aurretik.

Scraping prozesuan jasotako datuak jsonld instantzian kargatu eta jsonld.getJsonLd funtzioaren bitartez instantziak barnean gordetzen duen txostena itzultzen du. Ebaluatzaile guztiek amaitzen dutenean, txosten guztiak mergeEvaluatorReports bidez bateratzen dira eta lortutako txostena luzapenera bidaltzen da.

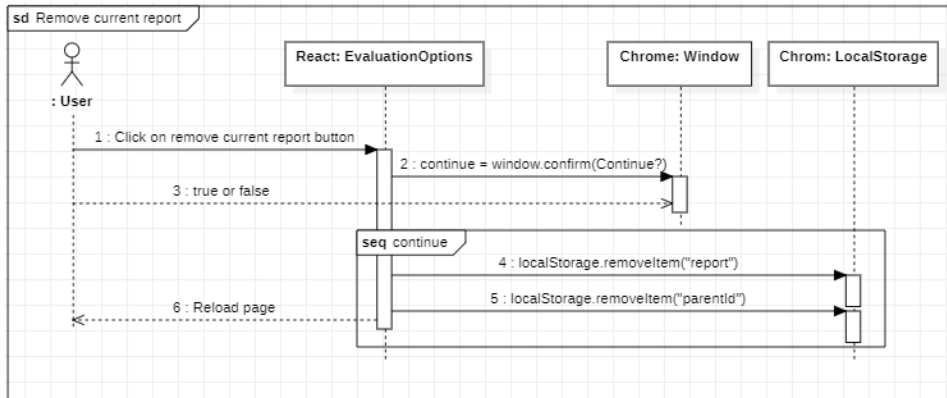
Azkenik, luzapenak txosten berria jasotzean aurretik txostenik dagoen egiaztatu eta horrela bada txosten berria editatzen du, aurreko txostenean editatutako emaitzak ez galtzeko. Txostena jaso eta orrialdea birkargatzen du.



5.7 Irudia: Evaluate selected scope sekuentzia diagrama

### 5.3.5 Remove current report

Lehenik, erabiltzaileak unean kargatutako txostena ezabatzeko botoia sakatu eta ziur dagoela adierazten du. Ondoren, luzapenak alde batetik kargatutako txostena ezabatu eta bestetik txosten horren gurasoaren gordetako Id balioa ezabatzen du. Bukatzeko, luzapenak orria birkargatzen du.



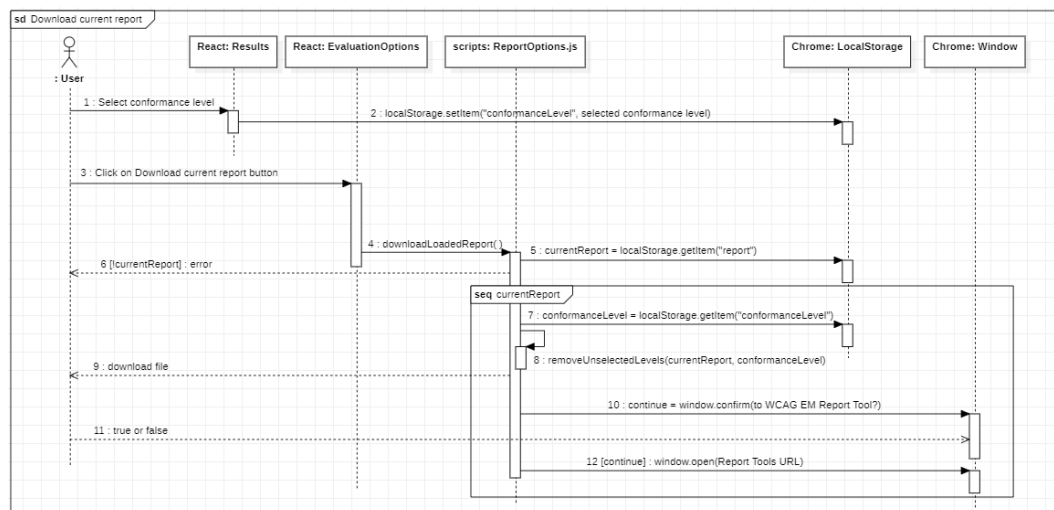
5.8 Irudia: Remove current report sekuentzia diagrama

### 5.3.6 Download current report

Hasteko, erabiltzaileak Download current report botoia sakatu aurretik A, AA eta AAA conformanceLevel balioen artean bat aukeratuko du edo bestela defektuzko AA balioa erabiliko du.

Ondoren, luzapenak gordetako txostena eskuratzen du, eta aukeratutako conformance-Level arabera, maila horretakoak ez diren irizpideak txostenetik kentzen ditu.

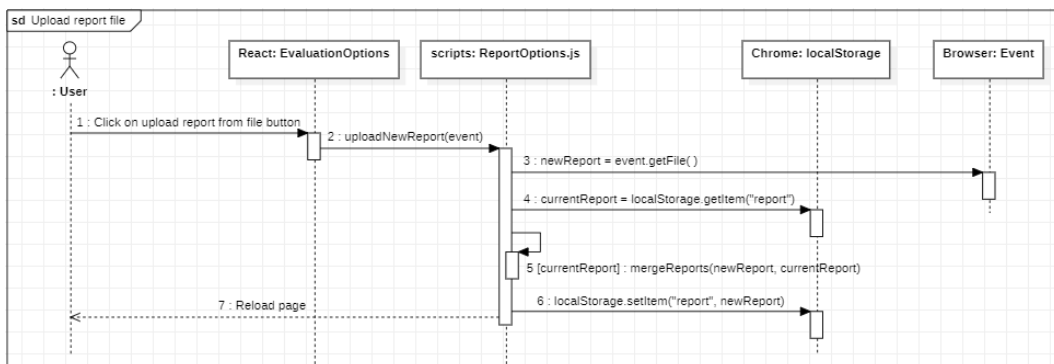
Azkenik, txostena fitxategi moduan jaitsi eta luzapenak erabiltzaileari WCAG EM Rerport Tool tresnara joan nahi duen galdetzen dio. Erabiltzaileak baiezkoa esaten badu tab berri batean aipatutako gunea irekitzen du.



5.9 Irudia: Download current report sekuentzia diagrama

### 5.3.7 Upload report file

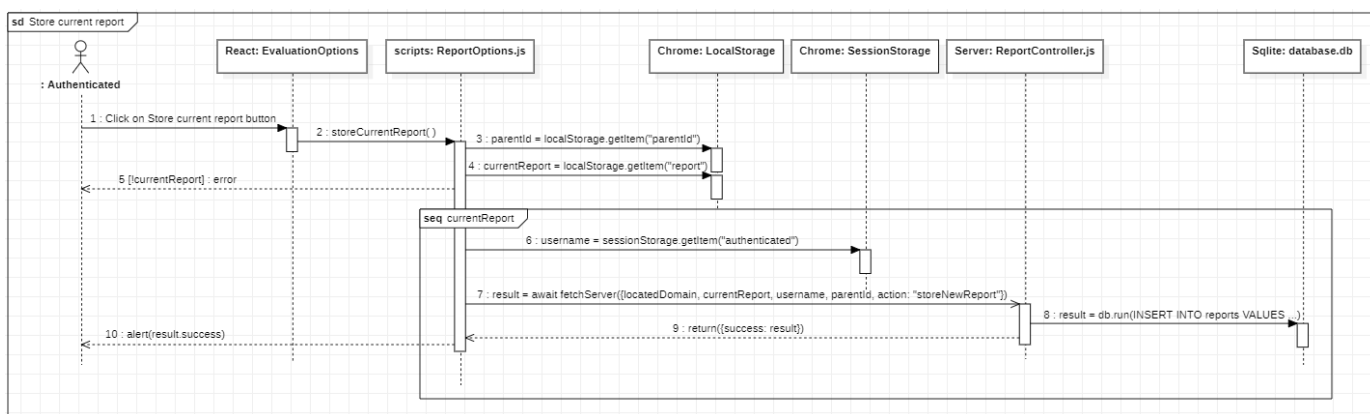
Lehenik, erabiltzaileak txostena fitxategi batetik igotzeko botoia sakatu eta fitxategia auke-ratzen du. Ondoren, luzapenak igoera gertaeratik txosten berria eskuratzen du eta aurretik txostenik kargatuta dagoen egiaztatzen du. Aurretik txostenik badago, txosten berrian alda-ketak egiten ditu aurreko txostenen editatutako emaitzak ez galtzeko. Azkenik, luzapenak txosten berria gorde eta orrialdea birkargatzen du.



5.10 Irudia: Upload report file sekuentzia diagrama

### 5.3.8 Store current report

Hasteko, erabiltzaileak unean kargatutako txostena datu basean gordetzeko botoia sakatzen du. Luzapenak txostena eta dagokion guraso txostenaren id balioa lortzen du eta ondoren erabiltzailearen kontuaren username balioa lortu eta zerbitzarira txostena gordetzeko eskaera egiten du. Ondoren, zerbitzariak txostena reports datu basean txertatu eta eragiketa ondo egin den itzultzen du. Azkenik, luzapenak eragiketa ondo egin den jakinarazten dio erabiltzaileari.



5.11 Irudia: Store current report sekuentzia diagrama

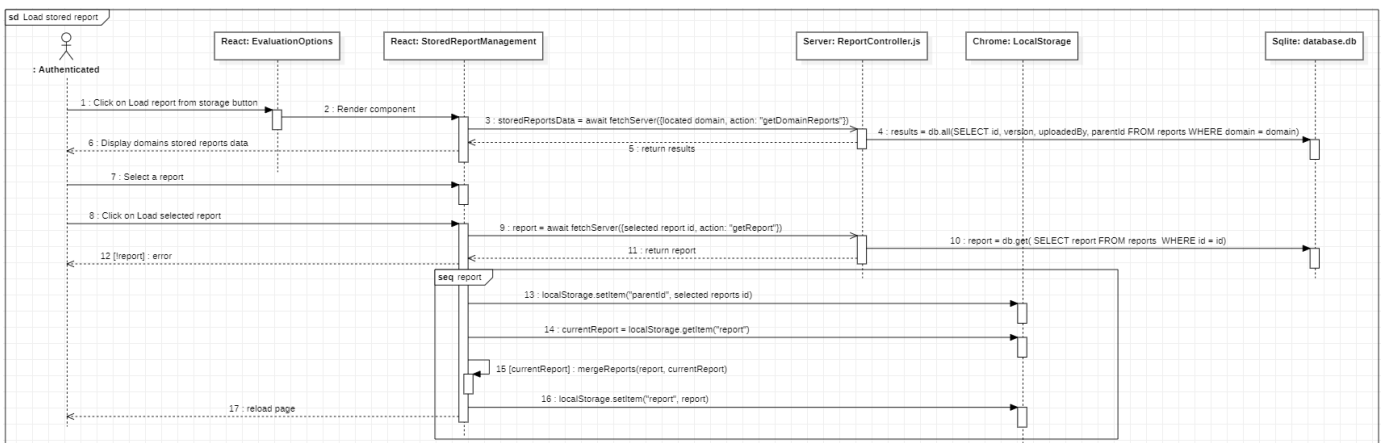


### 5.3.9 Load stored report

Hasteko, erabiltzaileak datu-basetik txostena kargatzeko botoian sakatzen du eta luzapenak zerbitzarira uneko domeinuko txostenak lortzeko eskaera egiten du. Zerbitzariak reports taulatik domeinu bereko txostenen datu guztiak eskuratu eta itzultzen ditu.

Ondoren, luzapenak jasotako txostenen datuak taula batean erakusten ditu. Erabiltzaileak taulako txostenetako bat aukeratu eta load botoian sakatzen du. Luzapenak aukeraturako txostenaren id-a erabiliz zerbitzarira txosten hori lortzeko eskaera egiten du.

Azkenik, zerbitzariak reports taulatik id bera duen txostena eskuratu eta itzultzen du. Luzapenak txosten berriaren id balioa parentId bezala ezartzen du, eta aurretik txostenik kargatuta dagoen egiaztatzen du. Aurretik txostenik bazegoen txosten berriaren aurrekoaren editatutako emaitzak gordetzen ditu. Luzapenak txosten berria gorde eta orrialde birkargatzen du.



5.12 Irudia: Load stored report sekuentzia diagrama

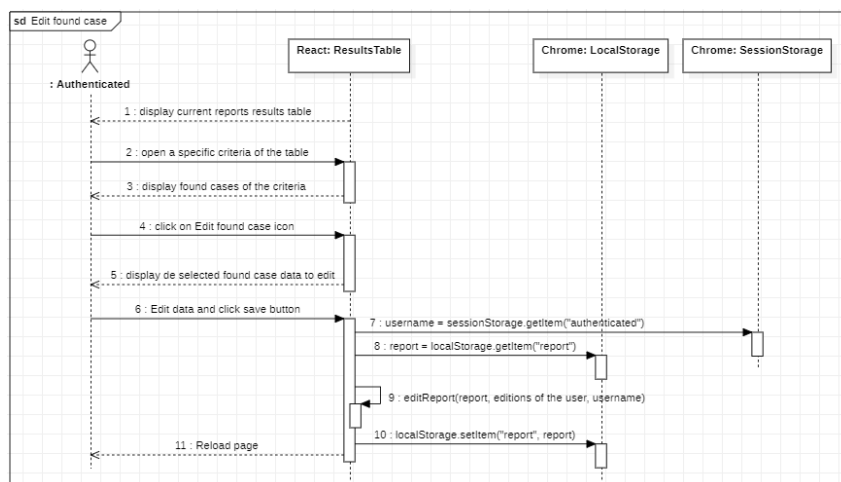
### 5.3.10 Edit found cases

Hasteko, erabiltzaileak emaitzen taulako irizpide bat hautatzen du. Luzapenak irizpideari dagozkien aurkitutako emaitzak erakusten ditu. Erabiltzaileak emaitzetako batean editatzeko ikonoa sakatzen du.

Luzapenak erabiltzaileak aukeraturako emaitzaren datuak input motako HTML elementuetan jartzen ditu, erabiltzaileak aldaketak egiten ditu eta gordetzeko botoian sakatzen du eta luzapenak aldaketak egin dituen erabiltzailearen username balioa eta kargatutako txostena lortzen ditu.

Luzapenak txostenean aldaketak egiten ditu erabiltzailearen izenean eta txostenaren bertsio berria gordetzen du aurrekoa berridatziz eta kokatutako orrialdea birkargatuz.

## 5. DISEINUA



5.13 Irudia: Edit found cases sekuentzia diagrama

## 5.4 Luzapenaren diseinu grafikoa

Hasteko, kontuan izan beharreko lehen gauza, luzapenak sidenav motakoa izaten segiko duela da, hau da, kokatutako orrialdeko bazter batean kokatzen direnetakoa. Kasu honetan, aurreko urteko diseinua segiz ezkerrean kokatuko da.

Bestalde, erabiltzailearentzako intuitiboa izateko, 5.14 irudian ikusten diren WCAG EM Report Tool tresnaren diseinua segiz, botoi guztiak beraien izaeraren arabera bi taldetan sailkatuko dira: Lehen taldekoak (argazkian ezkerrekoa) funtzionalitate nagusi modukoak izango dira, eta bigarrenekoak ordezkioak.



5.14 Irudia: WCAG EM Report Tool tresnako botoien diseinua

Azkenik, aurten AC-Check beharrean Earler izena erabiltzea erabaki dugu, EARL irisgarritasun ebaluaketarako hiztegi tekniko zehazten duen lengoia bat dela kontuan izanda. Hori dela eta, aurreko urteko argazkiko AC-CHECK testua EARLER testuagatik aldatu dut, baina hasiera batean argazkian azaltzen den irudia mantentzea erabaki dut. Ondorengo ataletako irudietan ikusten da aipatutako irudia.

### 5.4.1 Luzapenaren oinarrizko egoerak

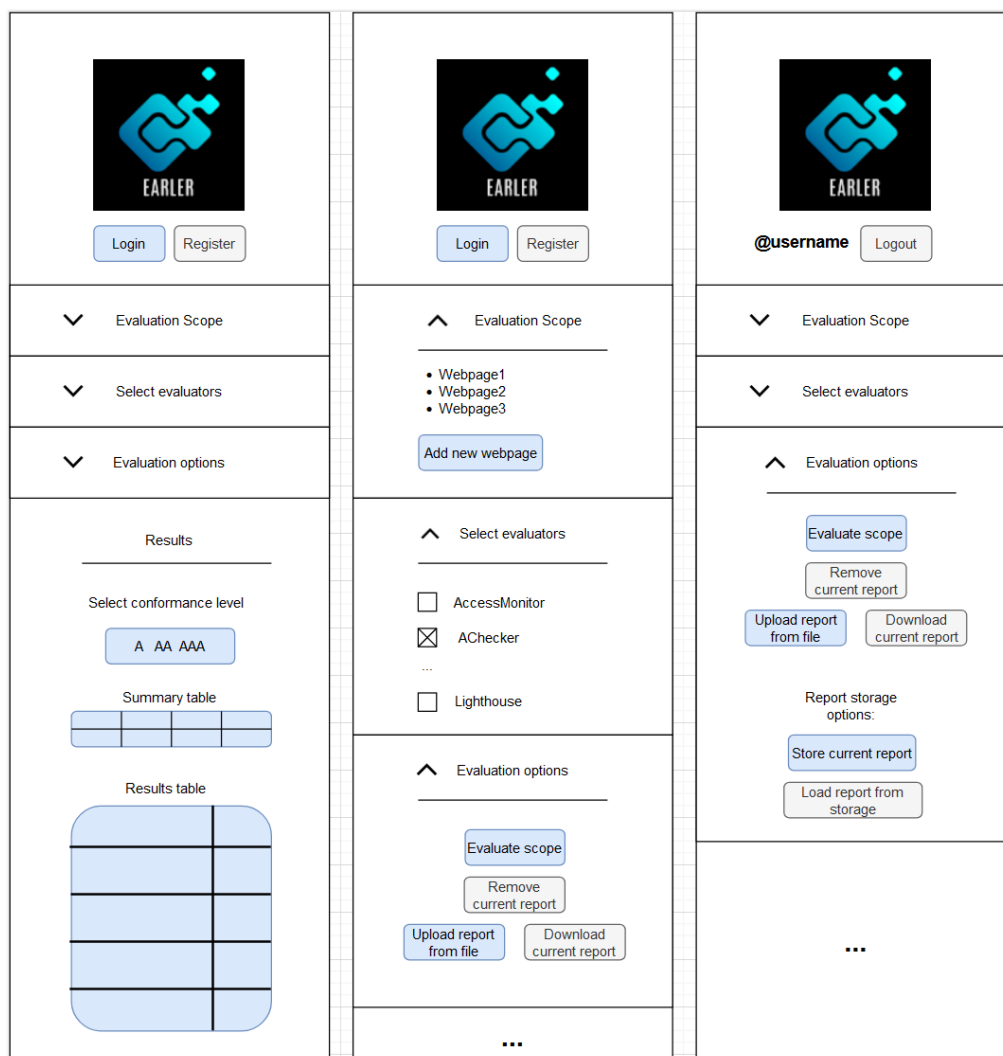
5.15 irudian luzapenaren 3 egoera azaltzen dira. Ezkerreko eta erdiko egoerek erabiltzailea kontuan sartu gabe dagoeneko egoera adierazten dute, lehena dropdown edo atal desplegableak itxita daudela, eta bigarrena aukera guztiak irekita daudela. Eskuineko diseinuak erabiltzailea kontuan sartuta dagoeneko egoera adierazten du.

Orokorrean luzapenak 5 sekzio izango dituela ikus dezakegu:

- **Luzapenaren burukoa:** Tresnaren logoa eta erabiltzaileen kontuetarako aukerak izango ditu.
- **Evaluation scope desplegablea:** Erdiko diseinuak adierazten duen bezala, erabiltzaileak ebaluatu nahi izango dituen webguneak adieraziko ditu zerrenda baten bitartez. Botoia sakatuz zerrendako elementu berri bat gehituko litzateke, dagozkion datu berriak sartzeko aukera emanez.
- **Select evaluators desplegablea:** Berriro ere erdiko irudian begiratu, ebaluatzaile automatikoen checkbox zerrenda egongo da. Dagoeneko erabaki dugun bezala, 6 ebaluatzaile izango dira, eta ebaluatzaileak erabili nahi dituenak aukeratuko ditu.
- **Evaluation options desplegablea:** Kasu honetan erdiko eta eskuineko irudiak konparatu, alde batetik erabilpen kasuetan definitutako User aktorearen 4 erabilpen kasu orokorrak ditugu, eta bestetik, kontuan sartutako erabiltzaileentzako beste bi aukera egongo dira, store current report eta load report from storage.
- **Results atala:** Ezkerreko irudian adierazten den bezala, atal hau denbora gutzian irekita egongo da eta emaitzak kargatuta egonez gero, 3 aukera eskainiko ditu. Lehenik select conformance level, zein mailatarainoko irisgarritasun irizpideen arabera

## 5. DISEINUA


emaitzak ikusi nahi ditugun ikusteko. Bigarrena, summary table, orrialdeko emaitzen kopuruen laburpena erakutsiko duena. Eta azkenik results table, irizpideen arabera sailkatutako emaitza guztiak erakutsiko dituen taula.



5.15 Irudia: Luzapena oinarrizko egoera

### 5.4.2 Erregistratzeko eta kontuan sartzeko egoerak

5.16 irudian adierazten den bezala, egoera bakoitzean formulario bat besterik ez dugu izango, atzera joateko cancel botoia eta kasu bakoitzeko erregistratzeko edo kontuan sartzeko botoiekin. Kontuan izan irudiak moztuta daudela eta luzapenak denbora gutzian irekitako nabigatzailearen altuera osoa beteko duela. Bi egoera hauetan, formularioen azpian ez da ezer gehiago egongo.



EARLER

---


Register

Email

Username

Password

Repeat password



EARLER

---

Login


Email

Password

5.16 Irudia: Kontuan sartzeko eta erregistratzeko egoerak

### 5.4.3 Datu-basean gordetako ebaluaketa txostenak kargatzeko egoera

5.17 irudiko egoeran, taula bat erakutsiko da luzapena unean kokatutako domeinuko ebaluaketa txosten guztien datuekin. Honen ideia, erabiltzaileak kargatu nahiko lukeen txostenaren lerroa sakatzuz aukeratu, eta ondoren Load botoian sakatzuz kargatzeko aukera izango luke. Aukeraketa egitean lerroaren fondoa aldatuko litzateke.



EARLER

---

Stored reports:

Id	UploadedBy	ParentId
1	Julen	
2	a	1
4	Mikel	2
7	Julen	2

5.17 Irudia: Datu-baseko txostenak kargatzeko egoera

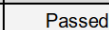
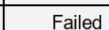
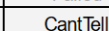
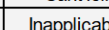
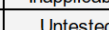
#### 5.4.4 Emaidza taulak

Emaidza tauletarako egitura aurreko urtean oinarritua izango da, eta dagokien irizpide bakoitzerako lortutako emaitzak errepresentatzeko erabiliko den kolore paleta ere, aurreko urtean AC-Check tresnarako erabili zen bera izango da, ikus [5.18](#) eta [5.1](#) irudiak.

Report content:

P	F	CT	NP	NC			
2	5	13	0	30			
Standard			P	F	CT	NP	NC
1	Perceivable		0	3	3	0	14
2	Operable		0	1	6	0	10
3	Understandable		1	1	4	0	4
4	Robust		1	0	0	0	2

5.18 Irudia: Aurreko urteko taulen diseinua

Emaidza mota	EARL terminoa	Colore kodea	Kolorea
Bete da	Passed	#C8FA8C	
Hutsegitea	Failed	#FA8C8C	
Abisua	CantTell	#F5FA8C	
Ezin da aplikatu	Inapplicable	#FFFFFF	
Ez da egiaztatu	Untested	#8CFafa	

5.1 Taula: Emaidzetzako kolore paleta

## 5.5 Irisgarritasun txostenen formatua

Hasteko, JSON-LD formatua JSON formatuaren estentsio moduko bat dela esan dezakegu, datuen kontestuak eta esanahiak zehazteko anotazio edo aurrizki modukoak zehazten dituelako. Gure kasuan aurrizki horiek oso interesgarriak dira, txostenean estandarretan oinarritutako emaitzak erabil ditzakegulako. Gainera, WCAG EM Report Tool tresnak JSON-LD formatuko aurrizkiak erabiltzen ditu tresnako atal bakoitzeko datuak txostenetik eskuratzeko. Hori dela eta, JSON-LD formatua erabiltzera behartuta gaude.

Mauve++ ebaluatzaileak esportatzeko eskaintzen duen JSON-LD egitura txantilo mo-  
duan erabiltzea pentsatu genuen, baina WCAG EM Report Tool tresnara igotzen saiatzean  
datu asko galtzen direla ikusi dut. Hori ikusita, zuzenean Report Tool-aren GitHub errepo-  
sitorioan begiratu eta beste adibide batzuk aurkitu ditut <sup>1</sup>.

Adibide horiek ez dute gorde behar ditudan datu guztientzako eremuak, baina bai daturik  
galdu gabe karga daitezkeela Report Toolean. Beraz, Mauve++ eredutik eta aurkitutako  
adibideetatik interesatzen zaizkidan zatiak hartu eta egokitzapenak egin ditut noizbehinka  
probak eginez datu guztiak irakurtzen segitzen dutela ziurtatzeko.

Ondoren, lortu dudana egitura zatika aztertuko dugu. Hasteko, @context atala:

```
{
  "@context": {
    "@vocab": "http://www.w3.org/TR/WCAG-EM/#",
    "wcag2": "http://www.w3.org/TR/WCAG21/#",
    "earl": "http://www.w3.org/ns/earl#",
    "dct": "http://purl.org/dc/terms/",
    "wai": "http://www.w3.org/WAI/",
    "sch": "http://schema.org/",
    "xmlns": "http://xmlns.com/foaf/0.1/",
    "ptr": "http://www.w3.org/2009/pointers#",
    "evaluationScope": {
      "@id": "step1",
      "@type": "EvaluationScope"
    },
    "siteScope": "step1a",
    "conformanceTarget": {
      "@id": "step1b",
      "@type": "@id"
    },
    "accessibilitySupportBaseline": "step1c",
    "additionalEvalRequirement": "step1d",
    "structuredSample": {
      "@id": "step3a",
      "@type": "Sample"
    },
    "auditSample": "step4",
    "siteName": "sch:name",
    "website": "wcag2:dfn-set-of-web-pages",
    "webpage": "wcag2:dfn-web-page-s",
    "Assertion": "earl:Assertion",
    "test": {
```

<sup>1</sup><https://github.com/w3c/wcag-em-report-tool/tree/master/docs/examples>

```

    "@id": "earl:test",
    "@type": "@id"
  },
  "assertedBy": {
    "@id": "earl:assertedBy",
    "@type": "@id"
  },
  "subject": {
    "@id": "earl:subject",
    "@type": "@id"
  },
  "result": "earl:result",
  "mode": {
    "@id": "earl:mode",
    "@type": "@id"
  },
  "outcome": {
    "@id": "earl:outcome",
    "@type": "@id"
  },
  "locationPointersGroup": "ptr:PointersGroup",
  "elementPointersGroup": "ptr:PointersGroup",
  "title": "dct:title",
  "description": "dct:description",
  "hasPart": "dct:hasPart",
  "creator": {
    "@id": "dct:creator",
    "@type": "earl:Assertor"
  },
  "id": "@id",
  "type": "@type"
},

```

"@context" atalean definitutako objektuak erreferentzia moduan jokatzen dute, hau da, adibidez "earl": "http://www.w3.org/ns/earl" jarrita, ondoren "earl:termino bat" bezala erabil dezaket, kasu honetan EARL lengoaiak definitzen duen termino zehatz bati erreferentzia egiteko. Modu horretara datuak partekatzerako garaian formatu eta datu mota "estandarrik" erabil daitezke.

Ondoren, ebaluaketari buruzko datuak definitzeko zatia aztertuko dugu:

```

"type": "Evaluation",
"@language": "en",
"title": "Accessibility Evaluation Report for ###---webgunea---### website",
"commissioner": "https://github.com/larraagirrejulen/GrAL/tree/main/ac_check%20(react)",
"dct:date": "###---data---###",
"assertors": [
  {
    "id": "._:AccessMonitor",
    "type": "earl:Assertor",
    "xmlns:name": "AccessMonitor",
    "description": "https://accessmonitor.acessibilidade.gov.pt"
  },

```



```

    ###---ebaluatzaile gehiago---###
  ],
  "creator": {
    "id": " _:assertors",
    "xmlns:name": " ###---ebaluatzaile izenak & bidez konkatenatuak---### "
  },
  "evaluationScope": {
    "website": {
      "id": " _:website",
      "type": [
        "earl:TestSubject",
        "sch:WebSite"
      ],
      "siteName": "###---webgunearen esteka---###",
      "siteScope": "###---ebaluatuko diren orrialdeen estekak ', ' bidez
        separatuak---###"
    },
    "conformanceTarget": "wai:WCAG2AAA-Conformance",
    "accessibilitySupportBaseline": "Google Chrome latest version",
    "additionalEvalRequirement": "The report will include location pointers to
      the cases found for each result"
  },
  "structuredSample": {
    "webpage": [
      {
        "id": "https://www.ehu.eus/es/home",
        "type": [
          "earl:TestSubject",
          "sch:WebPage"
        ],
        "description": "https://www.ehu.eus/es/home",
        "source": " _:website",
        "title": "Inicio - UPV/EHU",
        "tested": true
      },
      ###---orrialde gehiago---###
    ]
  },
  "auditSample": [],
  "dct:summary": "Some errors where found..."
}

```

"title", "comisioner", "dct:date", "creator" barruko "xmlns:name", eta bukaeran azaltzen den "dct:summary" balioek, 5.21 irudian ikusten diren eremuak betetzen dituzte, "evaluationScope" objektuko balioek 5.19 irudiak ikusten direnak, eta "structureSample-ko balioek 5.20 irudiak ikusten den pausuko Structured Sample Web Pages atala.

## Step 1: Define the Evaluation Scope

Define the overall parameters and scope of the evaluation. Ideally, this is done with the person who commissioned the evaluation (who may or may not be the website owner), to ensure common expectations about the scope of the evaluation.

Step 1: Define the Evaluation Scope

Website name Show info +

Scope of the website Show info +

WCAG Version Show info +

Conformance target Show info +

Accessibility support baseline Show info +

Additional evaluation requirements Show info +

### 5.19 Irudia: WCAG EM Report Tool 1.pausua

## Structured Sample Web Pages Show info +

<b>Sample page Inicio - UPV/EHU</b> <span>✕</span>	
Short name:	Address (URL) or description:
<input type="text" value="Inicio - UPV/EHU"/>	<input type="text" value="https://www.ehu.eus/es/home"/>
<b>Sample page alumni</b> <span>✕</span>	
Short name:	Address (URL) or description:
<input type="text" value="alumni"/>	<input type="text" value="https://www.ehu.eus/es/web/ehualumni"/>
<b>Sample page profesorado</b> <span>✕</span>	
Short name:	Address (URL) or description:
<input type="text" value="profesorado"/>	<input type="text" value="https://www.ehu.eus/es/irakasleak"/>

### 5.20 Irudia: WCAG EM Report Tool 2.pausuko azpiatala

## Step 5: Report the Evaluation Findings

Provide additional information about the evaluation that you want included in the report. The evaluation findings that you entered in the previous step are included below under 'Detailed audit results' for review and to help you write the executive summary.

Step 5: Report the Evaluation Findings

Report title Show info +  
 Accessibility Evaluation Report for https://www.ehu.eus/ website

Evaluation commissioner Show info +  
 https://github.com/larraagirrejulen/GrAL/tree/main/ac\_check%20(react)

Evaluator Show info +  
 AccessMonitor & AChecker & MAUVE & A11Y & Pa11y & Lighthouse

Evaluation date Show info +  
 17/6/2023, 18:01:17

Executive summary Show info +  
 Some errors where found...

### 5.21 Irudia: WCAG EM Report Tool 5.pausua

Aurreko atalean ikusten den "auditSample"atributuko zerrendak irisgarritasun irizpideen emaitzak jasoko ditu. Emaitzak honako formatua izango dute:

```
{
  "type": "Assertion",
  "test": "wcag2:##--- irizpidearen identifikatzailea---##",
  "conformanceLevel": "##--- irizpidearen arabera A, AA edo AAA ---##",
  "subject": "_:website",
  "result": {
    "outcome": "earl:##--- emaitza ---##",
    "description": "##--- deskripzioa ---##"
  },
  "hasPart": []
}
```

Garrantzitsua da "test"eta "outcome"balioak zehazki "earl"eta "wcag2"zehazten dituzte-netakoak izatea, bestela emaitzak ez dira ondo kargatzen. Outcome balioak honakoak izan daitezke: "earl:passed", "earl:failed", "earl:cantTell", "earl:inapplicable"edo "earl:undefined". Irizpideen identifikatzaileak dagokien webgunean aurki daitezke <sup>2</sup>.

Emaitza horietako bakoitzak, "test"eremuan adierazten den irizpiderako webgune osoak jasotako emaitza adierazten du. Baina adibidez ebaluaketa hainbat orrialde ezberdinetan

<sup>2</sup><https://www.w3.org/WAI/WCAG21/quickref/>

rako bada, kokatutako irizpide bera duten orrialde ezberdinetako emaitza guztiak "hasPart" atributuko zerrendan txertatzen dira. Gainera, hasPart zerrendan orrialde bererako hainbat objektu egon daitezke "outcome" balioen eta emaitza bera lortu duten ebaluatzaileen arabera multzokatuak.

"hasPart" eremuko objektuek honako egitura izaten dute:

```
{
  "type": "Assertion",
  "testcase": "wcag2:##--- irizpidearen identifikatzailea---##",
  "assertedBy": [
    {
      "assertor": "##--- ebaluatzailea ---##",
    }
  ],
  "asserted": "##--- ebaluatzaile gehiago ---##",
  "modifiedBy": [],
  "lastModifier": "",
  "subject": "##--- ebaluatutako web orrialdea---##",
  "mode": "earl:automatic",
  "result": {
    "outcome": "earl:##--- emaitza ---##",
    "description": "##--- deskribapena ---##",
    "locationPointersGroup": []
  }
}
```

Egitura horren bitartez, Report Tool-eko 4.pausuko eremuak betetzen dira, ikus [5.22](#) irudia.

"lastModifier" eta "modifiedBy" eremuak aurrerago emaitzetan eskuz aldaketak egin diren eta zeinek egin dituen jakiteko jarri ditut, baina dagokion funtzionalitatea garatzean baliteke aldaketak izatea.

Kontuan izan beharreko beste gauza bat "locationPointersGroup" da. Report Tool tresnak ez ditu kode zatietarako erreferentziak irakurtzen, baina nik luzapeneko taulatik nabarmendu ahal izateko, HTML elementuen path modukoak beharko ditut. Zerrenda horren bitartez, ideia aurkitutako elementu bakoitzeko objektu bat txertatzea da, dagokion beharrezko informazioarekin.

Oraingoz zerrenda horretako elementuak honela definitu ditut:

```
{
  "id": "_:pointer",
  "type": "ptr:groupPointer",
  "assertedBy": [
    "##--- ebaluatzailea ---##",
    "##--- ebaluatzaile gehiago ---##"
  ],
  "ptr:expression": "##--- path moduko balioa ---##",
  "description": "##--- path horri dagokion HTML elementua ---##",
}
```

**1 Perceivable**

**1.1 Text Alternatives**

**1.1.1: Non-text Content** Level A

[Understanding 1.1.1](#) [How to meet 1.1.1](#)

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. [Show full description](#) +

**Entire sample**

Outcome: **Cannot tell** Observations: [View in report](#)

Webqune emaitzaren deskripzioa

**Show web pages to enter individual results**

**Results for: Inicio - UPV/EHU**

Outcome: **Passed** Observations: [View in report](#)

Inicio - UPV/EHU orrialdeko deskripzioa

**Results for: alumni**

Outcome: **Cannot tell** Observations: [View in report](#)

alumni orrialdeko deskripzioa

5.22 Irudia: WCAG EM Report Tool 4.pausua irizpide bateko emaitzak



# Implementazioa

Kapitulu honetan orain arte definitzen joan garen tresnaren implementazioari buruz hitz egingo dugu, hau da, aurretik pentsatutakoa azkenean nola garatu dugun azalduko dugu<sup>1</sup>. Azalpenak 3 ataletan banatzen dira: lehena luzapenaren implementazioa 6.1, bigarrena zerbitzariaren implementazioa 6.2 eta azkena implementazioa aurrean joan ahala aurkitu ditugun arazo ezberdinak azaltzeko 6.3.

## 6.1 Luzapenaren implementazioa

Atal honetan, luzapena implementatu ahal izateko alderdi ezberdinak azalduko ditugu: React frameworka zer osagaiekin eta nola antolatu dugun 6.1.1, osagai horiei estiloa nola eman diegun 6.1.2 eta zeintzuk izan diren erabili ditugun funtzio garrantzitsuenetariko batzuk 6.1.3.

Gainera, luzapenak azkenean nolako itxura hartu duen, funtzionalitate ezberdinak grafikoki nola erakusten diren 6.1.4 eta garatu diren funtzionalitate gehigarriak zeintzuk diren 6.1.5 ere azalduko ditugu.

### 6.1.1 React "components" edo osagaiak

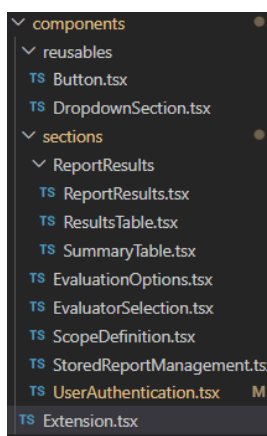
React implementazioari dagokionez, kodea hainbat osagaietan banatu dut horietako bakoitzak duen ardura kontuan izanda. Orokorrean, diseinu grafikoan azaltzen diren atal bakoitzeko osagai bat sortu dut, beraien artean datuak pasatu behar direnean, arakatzailerak eskaintzen dituzten storage aukera ezberdinak erabiliz.

Gainera, behin baina gehiagotan erabiltzen ditudanez, berrerabiltzeko bi osagai ere sortu ditut `Button.tsx` eta `DropdownSection.tsx` fitxategietan.

6.1 irudian ikus daitezke luzapeneko osagai guztiak.

---

<sup>1</sup>Erabilitako tresna eta teknologia guztiak eranskinen ataleko "Erabilitako tresnak" 9.3 atalean azaltzen dira



6.1 Irudia: Luzapeneko React osagaiak

Hasteko, luzapenaren osagai nagusia, hau da, gainontzeko guztiak biltzen dituena, `Extension.tsx` fitxategiko **Extension** osagaia da. Honek luzapenaren egoeraren arabera erabiltzaileari erakutsi beharreko egitura orokorra antolatzen du.

Horretaz aparte, aurrerago ikusiko dugun hiru funtzionalitate implementatzeko erabili dut: luzapena ezkutatzeko, konfigurazio orrira joateko eta konfigurazio orriko aukerara baten arabera luzapenak kokatutako orrialdea alde batera eramateko edo tapatuz gainean jartzeko.

Kodeak itzultzen duen zati garrantzitsuena 6.2 irudian ikus daiteke.

```
<div className="img_container">
  <img
    alt="extension logo"
    src={getImgSrc("icon128")}
    onClick={() => window.open("https://github.com/larraagirrejulien/GrAl", '_blank')}
  />
</div>

{manageStoredReports ? <>
  <StoredReportManagement
    setManageStoredReports={setManageStoredReports}
    authenticationState={authenticationState}
  />
</> : <>
  <UserAuthentication
    authenticationState={authenticationState}
    setAuthenticationState={setAuthenticationState}
  />
  { authenticationState !== "logging" && authenticationState !== "registering" && <>
    <ScopeDefinition />
    <EvaluatorSelection />
    <EvaluationOptions
      authenticationState={authenticationState}
      setLoadingReports={setManageStoredReports}
    />
    <ReportResults />
  </> }}
</>
```

6.2 Irudia: Extension osagaiak itzultitako zati nagusia



Ondoren, **UserAuthentication** osagaia azalduko dugu. Osagai honen eginkizuna, luzapenaren egoeraren arabera erabiltzaileak identifikatzeko dituen aukerak eskaintzea da, hau da, kontuan sartu gabe badago Login eta Register botoiak, kontuan sartzen badago Login formularioa, erregistratzen badago Register formularioa eta azkenik kontuan sartuta badago bere username izena eta Logout botoia.

6.3 irudian ikus daiteke kodearen zati garrantzitsuena. Aipatzeko, argazkian azaltzen diren LoginForm eta RegisterForm fitxategi berean kokatzen diren formularioak kudeatzeko bi osagai dira.

```

<div id="extension_authentication_section">
  {authenticationState === "notLogged" ? <>
    <Button
      classList={"primary"}
      onClickHandler={() => setAuthenticationState("logging")}
      innerText={"Login"}
    />
    <Button
      classList={"secondary spaced"}
      onClickHandler={() => setAuthenticationState("registering")}
      innerText={"Register"}
    />
  </> : authenticationState === "logging" ?
  <LoginForm setAuthState={setAuthenticationState} />
  : authenticationState === "registering" ?
  <RegisterForm setAuthState={setAuthenticationState} />
  : <>
  <label className='userNameLabel'>
    {"@" + authenticationState}
  </label>
  <Button
    classList={"secondary spaced"}
    onClickHandler={onLogoutHandler}
    innerText={"Logout"}
  />
  </> }
</div>

```

6.3 Irudia: UserAuthentication osagaiak itzultitako zati nagusia

**ScopeDefinition** osagaiaren aldetik, erabiltzaileak ebaluatu nahi dituen orrialdeak zerrenda batean txertatzeko aukerak ematen ditu, beti ere, erabiltzaileak sartutako estekak unean kokatutako domeinukoak dituela kontuan izanik. Zerrenda scope izeneko aldagai batean gordetzen da, eta dagozkion elementuak eskusteko ul motako zerrenda ordenatu baten barruan txertatzen dira map baten bitartez, ikus 6.4 irudia.

```

<ul id="extensionScopeInputList">
  {scope.map((webPage:any, index:any) => (
    <li className="scopeInput" key={index}>
      {editItemIndex === index ? (

```

6.4 Irudia: Erabiltzaileak sartutako web orrialdeak zerrendatzeko

**EvaluatorSelection** osagaiari buruz aipatzeko bakarra, 6.5 irudiko defaultCheckboxes aldagaieko ebaluatzaile automatiko bakoitza checkbox moduan erakutsi, eta erabiltzaileak

## 6. INPLEMENTAZIOA

egindako aldaketak gordetzeaz arduratzen dela da. 6.6 irudian zerrendatzeko erabiltzen den kodea ikusten da.

```
const defaultCheckboxes = [
  { checked: false, label: "AccessMonitor - Website", href: "https://accessmonitor.acessibilidade.gov.pt/"},
  { checked: false, label: "AChecker - Website", href: "https://achecker.achecks.ca/checker/index.php"},
  { checked: false, label: "Mauve - Website", href: "https://mauve.isti.cnr.it/singleValidation.jsp"},
  { checked: false, label: "Ally - Library", href: "https://github.com/ainspector/ally-evaluation-library"},
  { checked: false, label: "Pally - Library", href: "https://www.npmjs.com/package/pally"},
  { checked: false, label: "Lighthouse - Library", href: "https://developer.chrome.com/docs/lighthouse/overview/" }
];
```

6.5 Irudia: Defektuzko ebaluatzaileen aukeraketarako checkbox elementuen egoera

```
{checkboxes.map((checkbox:any, index:any) => (
  <div className="checkbox-wrapper">
    <div className="checkbox">
      <input
        type="checkbox"
        checked={checkbox.checked}
        onChange={()=>handleCheckboxChange(index)}
        className={checkbox.checked && "checked" }
      />
      <span onClick={() => { window.open(checkbox.href, '_blank'); }}>
        {checkbox.label}
      </span>
    </div><br/>
    <span>{checkbox.checked ? "Selected" : "Unchecked"}</span>
  </div>
))}
```

6.6 Irudia: Checkbox elementuak zerrendatzeko eta erakusteko kodea

**EvaluationOptions** osagaiari dagokionez, erabiltzaileak kontuan sartuta dagoen kontuan izanda, ebaluaketa aukera ezberdinentzako botoiak erakusten ditu, ikus 6.7 eta 6.8 irudiak.

```
<Button
  classList={"primary br"}
  onClickHandler={()=>{evaluateScope(setAnimateBtn)}}
  innerText={"Evaluate selected scope"}
  isLoading={animateBtn !== "none"}
  animate={animateBtn === "evaluate"}
/><br/>

<div className='dropdownBtn'>
  <div className={"dropdownHead" + (isOpen ? " active" : "")} onClick={()=>setIsOpen(!isOpen)}>
    <label>Current report options</label>
    <img
      src={ isOpen ? getImgSrc("extendedArrow") : getImgSrc("contractedArrow") }
      alt="dropdown_arrow"
    />
  </div>
  {isOpen && (
    <div className='dropdownBody'>
      <label onClick={removeLoadedReport}>Remove report</label>
      <label onClick={downloadLoadedReport}>Export report</label>
      <label id="importReport">
        <input type="file" accept=".json" onChange={(event) => uploadNewReport(event)} />
        Import new report
      </label>
    </div>
  )}
</div>
```

6.7 Irudia: Erabiltzaile guztientzako aukerak

```

{authenticationState !== "notLogged" && ( <
  <div className='loggedOptions'>
    <p>Server storage options:</p>
    <Button
      className={"primary br"}
      onClickHandler={()=>storeNewReport(setAnimateBtn, authenticationState)}
      innerText={"Store current report"}
      isLoading={animateBtn !== "none"}
      animate={animateBtn === "store"}
      disabled={reportIsLoaded !== "true"}
    /><br/>
    <Button
      className={"secondary"}
      onClickHandler={()=>setLoadingReports(true)}
      innerText={"Load report from storage"}
      isLoading={animateBtn !== "none"}
      animate={animateBtn === "load"}
    />
  </div>
</> )}

```

6.8 Irudia: Kontuan sartutako erabiltzaileentzako aukerak

**StoredReportManagement** osagaiak, orrialdeka antolatutako taula batean datu-basean gordetako ebaluaketa txosten guztiak erakusten ditu. Erabiltzaileak horietako bat aukeratu ondoren, kargatzeko edo ezabatzeko aukerak eskaintzen ditu. 6.9 irudian taula nola egituratzen den ikus daiteke.

```

<div className='tableWrapper'>
  <table className='websiteStoredReports'>
    <thead>
      <tr>
        <th style={{width: "50px"}}>Id</th>
        <th style={{width: "60px"}}>Version</th>
        <th style={{width: "80px"}}>Uploaded by</th>
        <th style={{width: "60px"}}>Parent Id</th>
      </tr>
    </thead>
    <tbody>
      {paginatedData[currentPage].map((report: any, index: number) => (
        <tr
          className={` ${selectedIndex === index && 'selected'} `}
          onClick={() => setSelectedIndex(selectedIndex === index ? -1 : index)}
          key={report.id}
        >
          <td>{report.id}</td>
          <td>{report.version}</td>
          <td>{report.uploadedBy}</td>
          <td>{report.parentId}</td>
        </tr>
      ))}
    </tbody>
  </table>
</div>

```

6.9 Irudia: Datu-basean gordetako txostenak erakusteko taularen egitura

Azkenik, emaitzak erakusteko atalaren konplexutasuna handia denez, 3 osagaietan banatu dut:

- **ReportResults:** beste bi osagaiak bateratzeko eta erabiltzaileak conformanceLevel aukeratzeko baliabidea eskaintzen du, ikus 6.10 irudia.
- **SummaryTable:** Kargatuta dagoen txostenaren laburpena duen taula erakusten du. Laburpena webgune osoarena edo unean kokatutako orrialdearena izan daitekeenez,

## 6. INPLEMENTAZIOA

---

tabulazio moduko bat erabiltzen du, ikus 6.11 irudia.

- **ResultsTable:** Kargatutako txostenaren emaitza guztiak erakustez arduratzen da. Elementuetan klikatuz orrialdean nabarmentzeko eta emaitzen edizioak egiteko aukerak barne ditu.

```
<div id="conformanceLevelSelector">
  <p>Select conformance level:</p>
  <div className="level-container">
    [{"A", "AA", "AAA"].map((level:any) => (
      <div
        className={`conformanceLevels ${conformanceLevels.includes(level) ? 'selected' : ''}`}
        onClick={() => handleLevelClick(level)}
      >
        {level}
      </div>
    ))}
  </div>
</div>
```

**6.10 Irudia:** Erabiltzaileak kontuan izan nahi duen conformanceLevel aukeratzeko kodea

```
<div className="tabs">
  <div
    className={activeTab === 'website' ? 'active' : ''}
    onClick={() => setActiveTab('website')}
    style={{width: "90px"}}
  >
    Website
  </div>
  <div
    className={activeTab === 'webpage' ? 'active' : ''}
    onClick={() => setActiveTab('webpage')}
    style={{width: "136px"}}
  >
    {pageSummaries && pageSummaries[window.location.href] && webpage ?
      webpage
      : "Current webpage"}
  </div>
</div>

{activeTab === 'website' && (<
  <table id="summaryTable">
    <tr> <OutcomeHeaders /> </tr>
    <tr> {webSiteOutcomes.map((count:any) => ( <td>{count}</td> ))} </tr>
  </table>
</>)}

{activeTab === 'webpage' && (<
  {pageSummaries && pageSummaries[window.location.href] ? (
    <table id="summaryTable">
      <tr> <OutcomeHeaders /> </tr>
      <tr> {webPageOutcomes.map((count:any) => ( <td>{count}</td> ))} </tr>
    </table>
  ) : (
    <div style={{ textAlign: 'center', padding: '15px 0' }}>
      Current webpage has not been evaluated
    </div>
  )}
</>)}>
```

**6.11 Irudia:** Kargatutako txostenaren emaitzak erakusteko kodea

## 6.1.2 Osagaien estiloak

Hasieran CSS normala erabiltzen hasi nintzen baina kode kopuru handia izanik denborarekin estiloak aplikatzea nahasgarria bilakatzen joan zen. Hori dela eta, estiloak aplikatzeko Sass tresna topatu eta inplementatzea erabaki nuen. Tresna horrekin hainbat gauza egin daitezke, baina nire kasuan batez ere bi gauzetarako erabili dut.

Alde batetik, `_colors.scss` izeneko fitxategi baten bitartez, luzapenean hainbat alditan erabiltzen ditudan koloreak aldagaietan definitu eta ondoren berrerabili ahal izateko definitzen ditut, ikus 6.12(a) eta 6.12(b) irudiak.

```

1
2 $primary-color: #005a6a;
3 $secondary-color: white;
4 $intermediate-color: mix($primary-color, $secondary-color);
5
6 $red-color: red;
7 $gray-borders: #c0c0c0;

```

((a)) Sass coloreen aldagaiak definitzeko fitxategia

```

1
2 @import "colors";
3
4 body.extension-active {
5   position: relative;
6   left: 300px;
7 }
8
9 #hidden_extension_logo{
10  all: initial;
11  position: fixed;
12  top: calc(50vh - 25px);
13  left: -25px;
14  height: 50px;
15  border-radius: 50px;
16  border: 2px solid $intermediate-color;
17  overflow: hidden;

```

((b)) Sass coloreen fitxategia inportatu eta aldagaia erabili

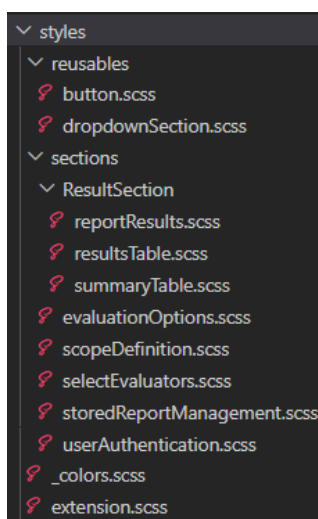
### 6.12 Irudia: Koloreak berrerabiltzen

Bestalde, scss fitxategietan HTML egitura segi daiteke nesting eginez, hau da, adibidez, 6.13 irudian azaltzen diren adibideak baliokideak dira. Modu horretara estilo fitxategien irakurgarritasuna asko handitu daiteke.

SCSS	Sass	CSS
<pre> nav {   ul {     margin: 0;     padding: 0;     list-style: none;   }    li { display: inline-block; }    a {     display: block;     padding: 6px 12px;     text-decoration: none;   } } </pre>	<pre> nav {   ul {     margin: 0;     padding: 0;     list-style: none;   }    li { display: inline-block; }    a {     display: block;     padding: 6px 12px;     text-decoration: none;   } } </pre>	<pre> nav ul {   margin: 0;   padding: 0;   list-style: none; } nav li {   display: inline-block; } nav a {   display: block;   padding: 6px 12px;   text-decoration: none; } </pre>

### 6.13 Irudia: Sass webguneko argazkia nesting kontzeptua ulertzeko

Hasieran CSS fitxategi gutxi batzuk aldatzen hasi nintzen baina azkenean denak SCSS izatera aldatu ditut. 6.14 irudian ikus daitezke geratu zaizkidan fitxategi guztiak. Azkenean, definitutako React-eko osagai bakoitzeko SCSS fitxategi bat dut.



6.14 Irudia: Proiektuko estilo fitxategi guztiak

### 6.1.3 Funtzio erabilgarriak

Luzapenaren funtzionamendurako hainbat funtzio ezberdin sortu behar izan ditut. Horieta-ko batzuk aipatzeagatik:

- Zerbitzarira deiak egiteko **fetchServer** funtzioa, ikus 6.15 irudia. BodyData parametroak zerbitzarian egin beharrekoa egiteko beharrezko parametroak gordetzen ditu, action parametroak zerbitzariko ze zerbitzu erabili nahi den adierazten du, eta timeout parametroak zenbat denbora egongo den itxaroten adierazten du.
- Orrialdeko elementuak eskuratzeko **getElementByPath** funtzioa. Path horiek CSS selector edo XPath motakoak izan daitezke. AChecker ebaluatzailearentzako HTML elementuen barruko testua ere erabili behar izan dut. Ikus 6.16 irudia.
- Erabiltzaileak aukeratutako txostenaren id bitartez, txosten hori datu-basetik eskuratzeko **loadStoredReport** funtzioa, ikus 6.17 irudia.

```
export async function fetchServer(bodyData, action, timeout = 120000) {
  try {
    const controller = new AbortController();
    const timer = setTimeout(() => controller.abort(), timeout);

    const response = await fetch('http://localhost:7070/' + action, {
      body: bodyData,
      method: "POST",
      headers: {"Content-Type": "application/json"},
      signal: controller.signal
    });

    clearTimeout(timer);

    if (!response.ok) throw new Error("HTTP error! Status: " + response.status);

    const fetchData = await response.json();
    return JSON.parse(fetchData);
  } catch (error) {
    throw new Error("Error fetching scraping server => " + error.name === 'AbortError' ? 'fetch timed out!' : error.message)
  }
}
```

6.15 Irudia: Zerbitzarira eskaerak egiteko funtzioa

```
export function getElementByPath(path, innerText) {  
  if (!path) throw new Error("Invalid input: path is null or undefined.");  
  let element = null;  
  
  if (path.startsWith("/")) {  
    element = document.evaluate(path, document, null, XPathResult.FIRST_ORDERED_NODE_TYPE, null).singleNodeValue;  
  } else {  
    const elements = window.document.querySelectorAll(path);  
  
    if (elements.length > 0) {  
      element = elements[0];  
  
      for (let i = 0; i < elements.length; i++) {  
        if (innerText && elements[i].textContent === innerText) {  
          element = elements[i];  
          break;  
        }  
      }  
    }  
  }  
  
  return element;  
}
```

6.16 Irudia: Orrialdeko elementuak eskuratzeko funtzioa

```
export async function loadStoredReport(id){  
  try{  
    const bodyData = JSON.stringify({action: "getReport", id});  
    const storeResults = await fetchServer(bodyData, "reportStoring");  
  
    if(storeResults.success){  
      const reportLoaded = await getFromChromeStorage(window.location.hostname + ".reportIsLoaded", false);  
      if(reportLoaded === "true"){  
        const currentReport = await getFromChromeStorage(window.location.hostname, false);  
        includeEditedFoundCases(storeResults.report, currentReport);  
      }  
      loadReport(storeResults.report);  
    } else {  
      window.alert("Could not get the report, try again later...");  
    }  
  }  
  catch(error){  
    console.log(error);  
  }  
};
```

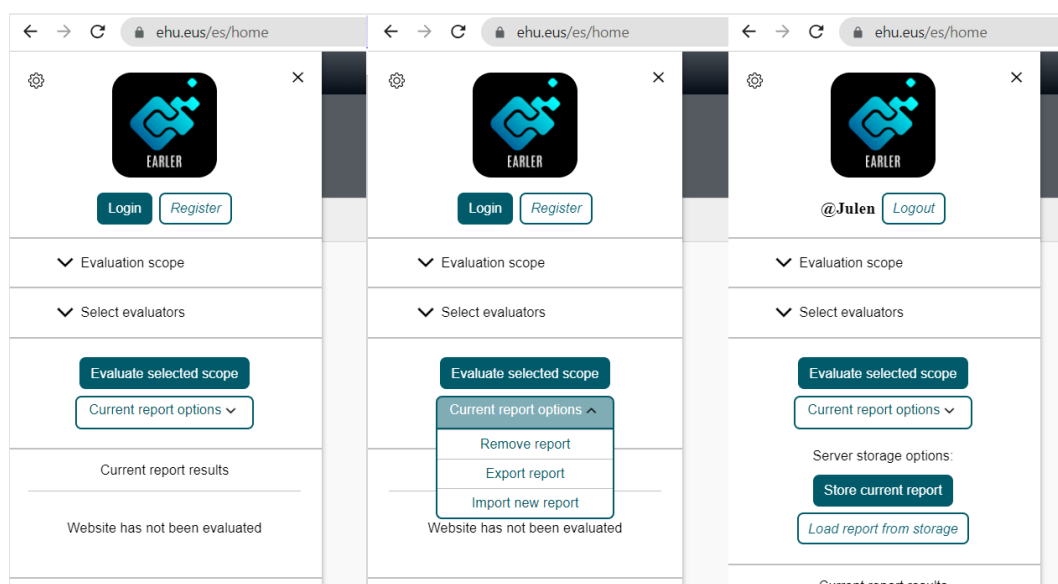
6.17 Irudia: Datu-basean gordetako txostena lortzeko funtzioa

### 6.1.4 Interfaze grafikoaren diseinu finala

Orokorrean diseinuen atalean egindako lana errespetatu da baina pare bat gauza aldatzeko premia ikusi da.

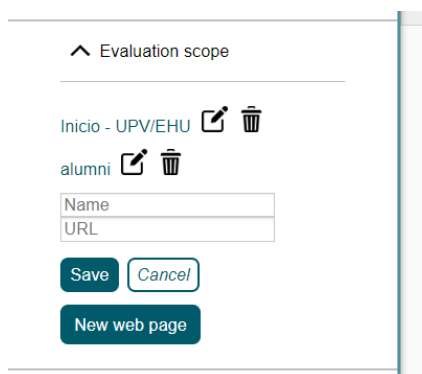
6.18 irudian ikus daitekeen bezala, alde batetik uneko txostena deskargatzeko, ezabatze-ko eta txosten berri bat fitxategi bidez igotzeko aukerak dropdown moduko botoi batean jarri dira. Modu horretara, erabiltzaileak ez ditu hainbeste aukera solte ikusiko.

Bestalde, aurreko aldaketari esker botoiek ez dute hainbesteko tokia okupatzen. Hori dela eta, Evaluation Options izeneko desplegable batean ezkutatuta egon beharrean, aukera guztiak zuzenean erakustea erabaki dugu.



6.18 Irudia: Ebaluatzaile aukeren atala aldatua

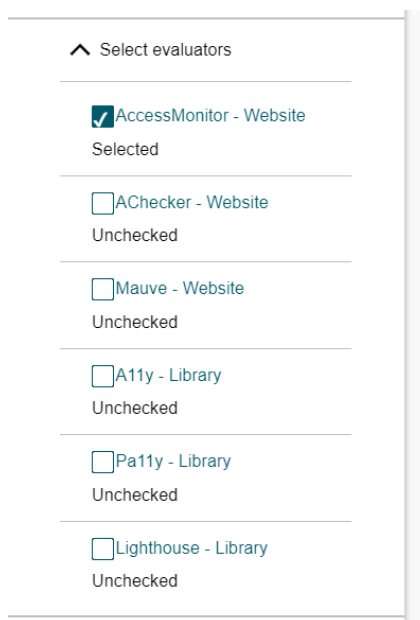
Evaluation scope deplegablearen aldetik, ikus 6.19 irudia, zerrendako elementuentzako editatzeko eta ezabatzeko irudiak jarri ditut, eta elementuetan klik egitean dagokion orrialdea arakatzailerako tab berri batean irekitzen du. Gainera, argazkian elementu berri bat nola txertatzen den ikusten da.



6.19 Irudia: Evaluation scope atalaren diseinu finala

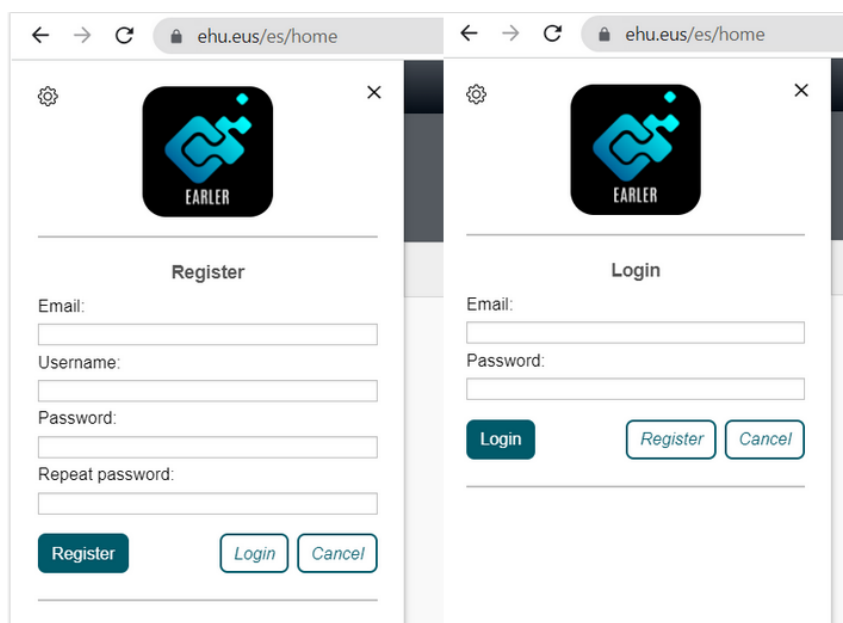


Select evaluators desplegablearen kasuan, ikus 6.20 irudia, ebaluatzaile bakoitzaren izenean klik eginda, dagokion gunera eramaten du.



6.20 Irudia: Select evaluators atalaren diseinu finala

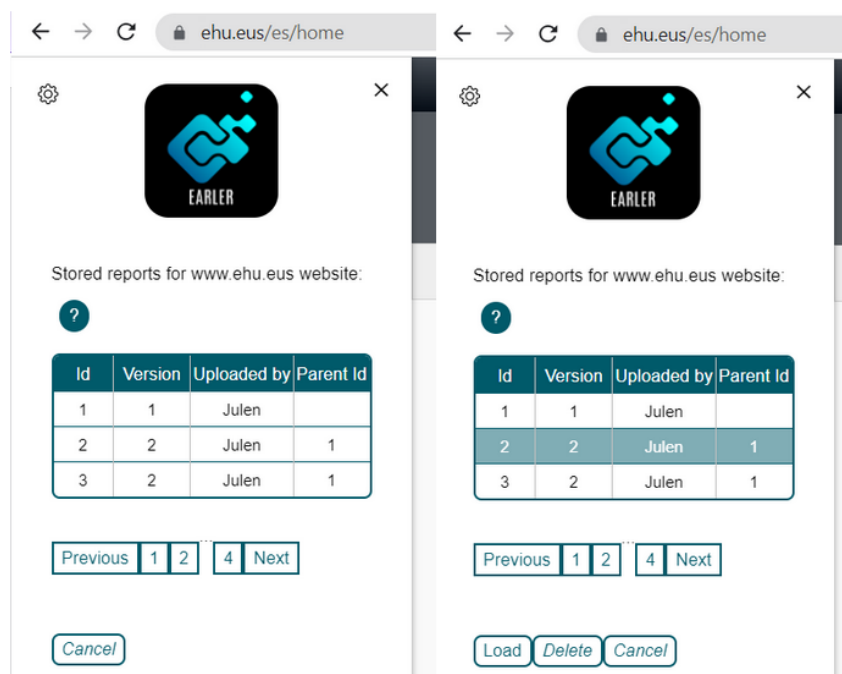
Kontuan sartzeko eta erregistratzeko formularioetan, bakoitzean bestera joateko botoi bat gehitu da, horretarako lehenengo atzera joan behar ez izateko, ikus 6.21 irudia.



6.21 Irudia: Login eta Register formularioen diseinu finala

Datu-basean gordeta dauden txostenak kudeatzeko diseinuan, aukeratutako txostena zurea bada ezabatzeko aukera, txostenen bertsioa ikusteko zutabe berria, taularen orrialde ezberdinak ikusteko botoiak eta taularen funtzionamendua ulertzeko laguntza botoi bat gehitu dira, ikus 6.22 irudia.

Hasieran txosten guztiak orrialde bakarrean jartzea pentsatu genuen, baina modu honetara orrialde bakoitzean "erro" txostenak eta dagozkien semeak jar ditzakegu, hau da, parentId ez duen txosten bakoitzeko orrialde bat egongo da. Ez dugu espero parentId ez duten txosten gehiegi egotea, erabiltzaileek besteek egindako lana berrerabiltzea izango dela joera pentsatzen dugulako. Modu horretara orri gutxi eta lerro asko egongo dira.



6.22 Irudia: Datu-baseko txostenak kudeatzeko diseinua

Emaitzak ikusterakoan, erabiltzaileak conformance level aukeratzea izango du, tauletako emaitzak aldatuz, konparatu 6.23 irudiko ezkerreko eta erdiko emaitzak.

Horretaz aparte, summary table taula txikian azaltzen diren emaitzak, erabiltzaileak aukeratutako tab modukoaren arabera da, hau da, webgunearen edo unean kokatutako orrialdearen arteko emaitzak ikustea edukiko ditu, konparatu 6.23 irudiko erdiko eta eskuineko emaitzak.

Ohartu bigarren kasuan azpiko taulako emaitzak ez direla aldatzen, azpiko taulak orrialdearen emaitzak jasotzen dituelako eta ez webgunearenak.

Kontuan izan, unean kokatutako webgunea ebaluatuta egoteak ez duela uneko orrialdea ebaluatu denik esan nahi, hau da, baliteke webgunea ebaluatu izana baina orrialdea ez, ikus 6.24 irudia.

## 6.1. Luzapenaren implementazioa

Current report results

Select conformance level: A AA AAA

Current Site/Page summary: Website Inicio - UPV/EHU

P	F	CT	NP	NC
3	5	23	4	15

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
1 Perceivable	0	1	7	3	9
2 Operable	1	0	10	1	5
3 Understandable	1	0	9	0	0
4 Robust	1	1	0	0	1

Current report results

Select conformance level: A AA AAA

Current Site/Page summary: Website Inicio - UPV/EHU

P	F	CT	NP	NC
3	7	24	4	40

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
1 Perceivable	0	2	7	3	17
2 Operable	1	0	10	1	17
3 Understandable	1	0	10	0	6
4 Robust	1	1	0	0	1

Current report results

Select conformance level: A AA AAA

Current Site/Page summary: Website Inicio - UPV/EHU

P	F	CT	NP	NC
3	3	27	4	41

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
1 Perceivable	0	2	7	3	17
2 Operable	1	0	10	1	17
3 Understandable	1	0	10	0	6
4 Robust	1	1	0	0	1

6.23 Irudia: Aukeratutako conformance level eta laburpen taulako tab-aren araberako emaitzak

Current report results

Select conformance level: A AA AAA

Current Site/Page summary: Website Current webpage

P	F	CT	NP	NC
3	7	24	4	40

Current report results

Select conformance level: A AA AAA

Current Site/Page summary: Website Current webpage

Current webpage has not been evaluated

6.24 Irudia: Webgunea ebaluatu da baina orrialdea ez

Azkenik, ebaluaketa emaitzen taulari dagokionez, 6.25 irudiko ezkerreko taulan ikus daitekeen bezala, irizpideak emaitzaren arabera koloreztatzen dira. Irudi bereko eskuineko taulan begiratu, irizpidearentzat aurkitutako kasu ezberdinak erakusten dira. Beti kasu okerrenetik onenera ordenatzen dira, eta kasu honetan guztiak ikusten dira. Kasu bakoitzak editatzeko eta ezabatzeko botoi bat ditu.

Normalean, aurkitutako kasu bakoitza irekitzean ebaluatzaileen emaitzak eta kode zatietarako erreferentziak 6.26(a) irudiaren antzera erakutsiko dira, baina kasuko editatzeko botoiari emanda, edukia 6.26(b) irudian bezala aldatuko da. Edizioan aldaketak egin eta gordetzen badira, 6.26(c) irudiko kasuan bezala editatu duenaren izena azaltzen da. Edizio moduan sartu gabe, ebaluatzaileen mezuak ezabatzeko edota blacklist funtzionalitate gehigarrian sartzeko aukera ematen duten bi ikono azaltzen dira ebaluatzaile bakoitzaren alboan.

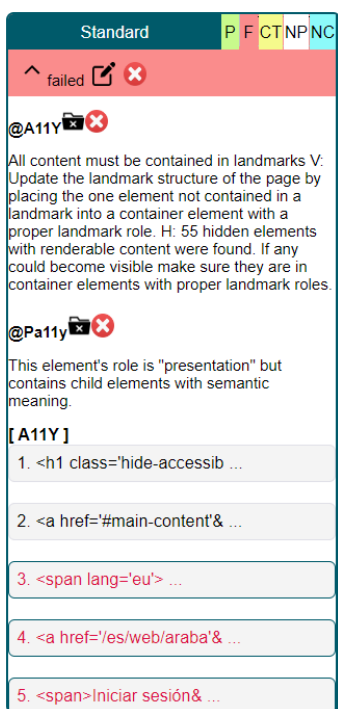
## 6. INPLEMENTAZIOA

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	<b>0</b>	<b>2</b>	<b>7</b>	<b>3</b>	<b>17</b>
1.1 Text Alternatives	0	0	1	0	0
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.3.1: Info and Relationships	earl:failed				
1.3.2: Meaningful Sequence	earl:cantTell				
1.3.3: Sensory Characteristics	earl:cantTell				
1.3.4: Orientation	earl:untested				
1.3.5: Identify Input Purpose	earl:untested				
1.3.6: Identify Purpose	earl:untested				
1.4 Distinguishable	0	1	4	0	8
<b>2 Operable</b>	<b>1</b>	<b>0</b>	<b>10</b>	<b>1</b>	<b>17</b>
<b>3 Understandable</b>	<b>1</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>6</b>
<b>4 Robust</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>

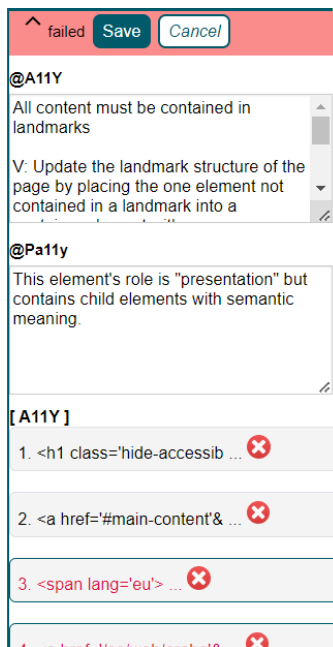
  

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	<b>0</b>	<b>2</b>	<b>7</b>	<b>3</b>	<b>17</b>
1.1 Text Alternatives	0	0	1	0	0
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.3.1: Info and Relationships	earl:failed				
1.3.2: Meaningful Sequence	failed				
1.3.3: Sensory Characteristics	cantTell				
1.3.4: Orientation	passed				
1.3.5: Identify Input Purpose	inapplicable				
1.3.6: Identify Purpose	inapplicable				
1.4 Distinguishable	0	1	4	0	8
1.3.2: Meaningful Sequence	earl:cantTell				
1.3.3: Sensory Characteristics	earl:cantTell				
1.3.4: Orientation	earl:untested				
1.3.5: Identify Input Purpose	earl:untested				
1.3.6: Identify Purpose	earl:untested				

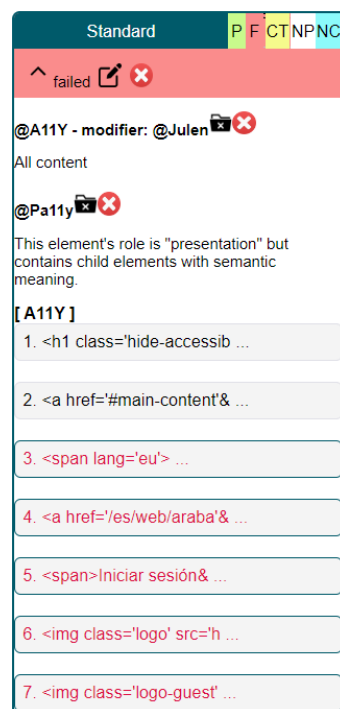
6.25 Irudia: Ebaluaketa emaitzen adibidea



((a)) Irizpidearen aurkitutako kasuaren emaitzak



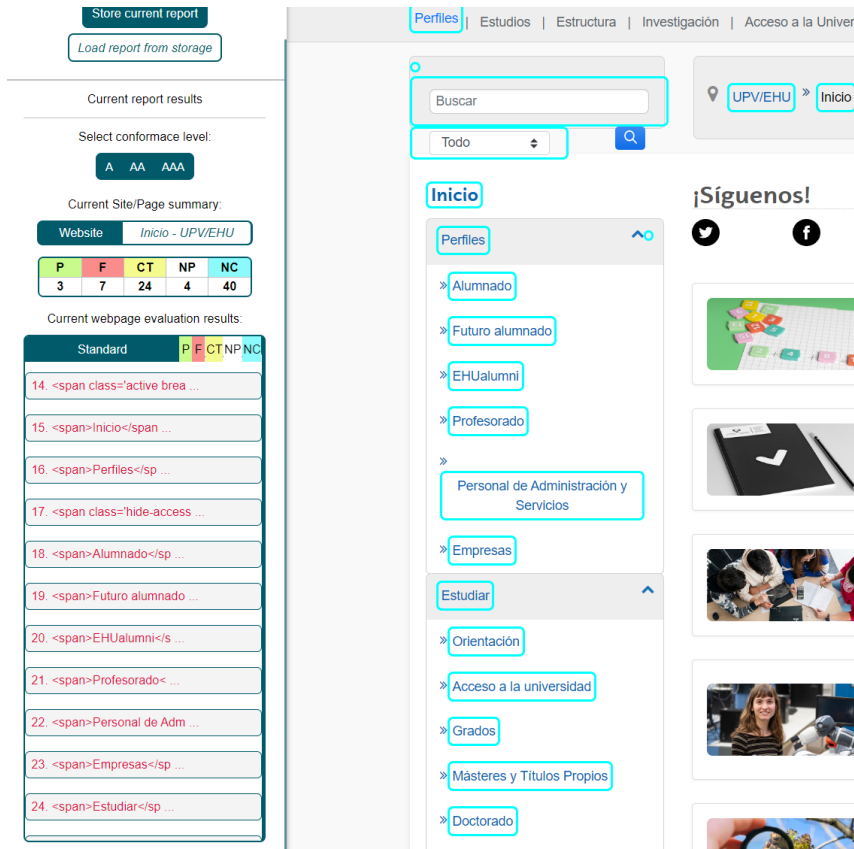
((b)) Irizpidearen aurkitutako kasuaren emaitzen edizioa



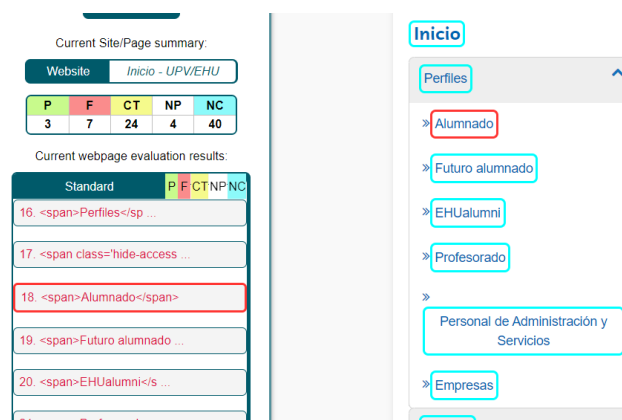
((c)) Irizpidearen aurkitutako kasuaren emaitzak editatuta

6.26 Irudia: Irizpideen emaitzak aldatzen

Gainera, aurkitutako kasu horietako bat irekitzean, barnean dauden erreferentziei dagozkien orrialdeko kode zatiak urdinez nabarmentzen dira. Ondoren gorritz koloreztatuta dauden erreferentzia batean sakatuz, orrialdeko dagokion elementua ere gorritz koloreztatzen da, ikus 6.27 eta 6.28 irudiak.



6.27 Irudia: Erreferentziei dagokien kode zatiak urdinez koloreztatuta

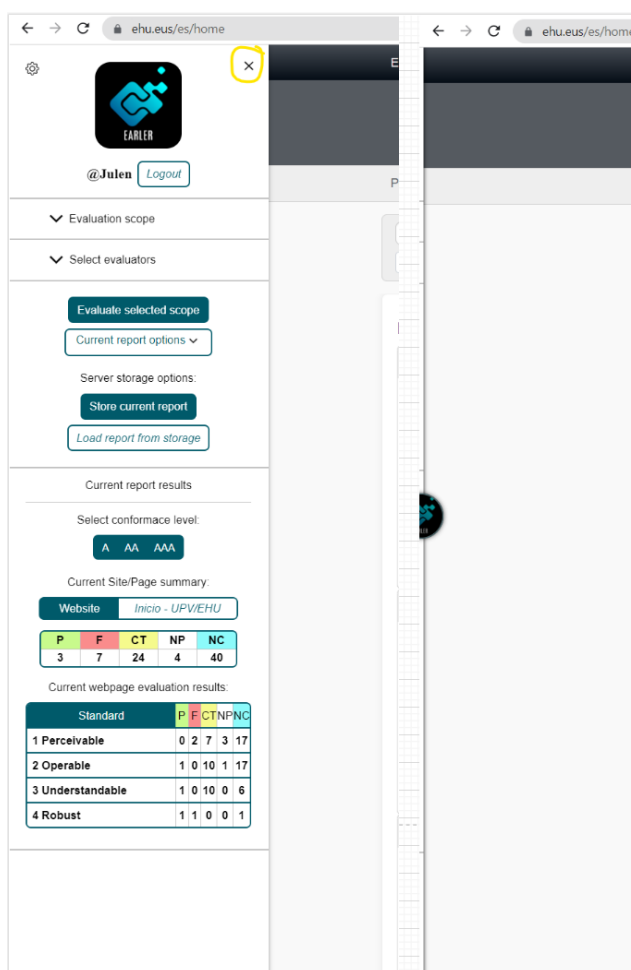


6.28 Irudia: Aukeratutako erreferentzia gorritz koloreztatuta

### 6.1.5 Funtzionalitate gehigarriak

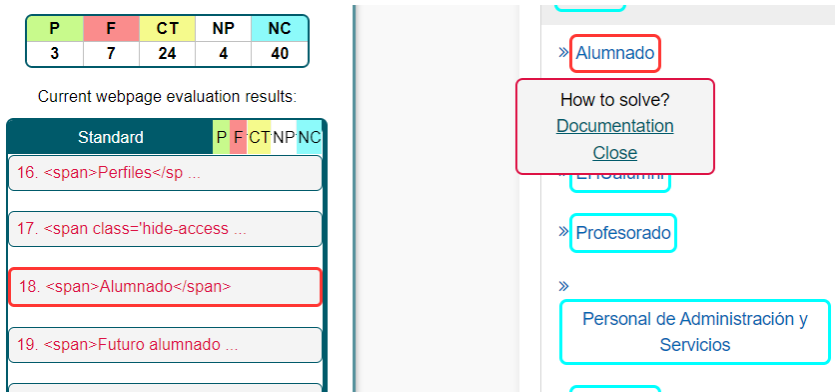
Inplementatu ditudan funtzionalitate gehigarriak honakoak izan dira:

- Luzapena ezkutatu eta erakusteko aukera, ikus 6.29 irudia.
- Emaitzetako kode zatietarako erreferentzia bat aukeratuta izatean, kurtsorea geruratuz gero popup moduko bat irekitzea irizpidearen araberako dokumentaziora eramateko aukerarekin, ikus 6.30 irudia.
- Konfigurazio orriko aukerak, ikus 6.31 irudia:
  - ShiftWebPage bidez luzapenak orrialdea tapatu edo alde batera mugituko duen zehaztu daiteke.
  - MaintainExtendedElements bidez, emaitzen taulako atal bat irekita izanda beste bat irekitzean aurretik irekitakoa itxi edo mantendu egin nahi den zehazten du.
  - Blacklist bidez, erabiltzaileak ebaluatzaile ezberdinen emaitza zehatzak zerrenda batean sartu ditzake aurrerantzeko ebaluaketa berrietan emaitza horiek ez erakusteko.

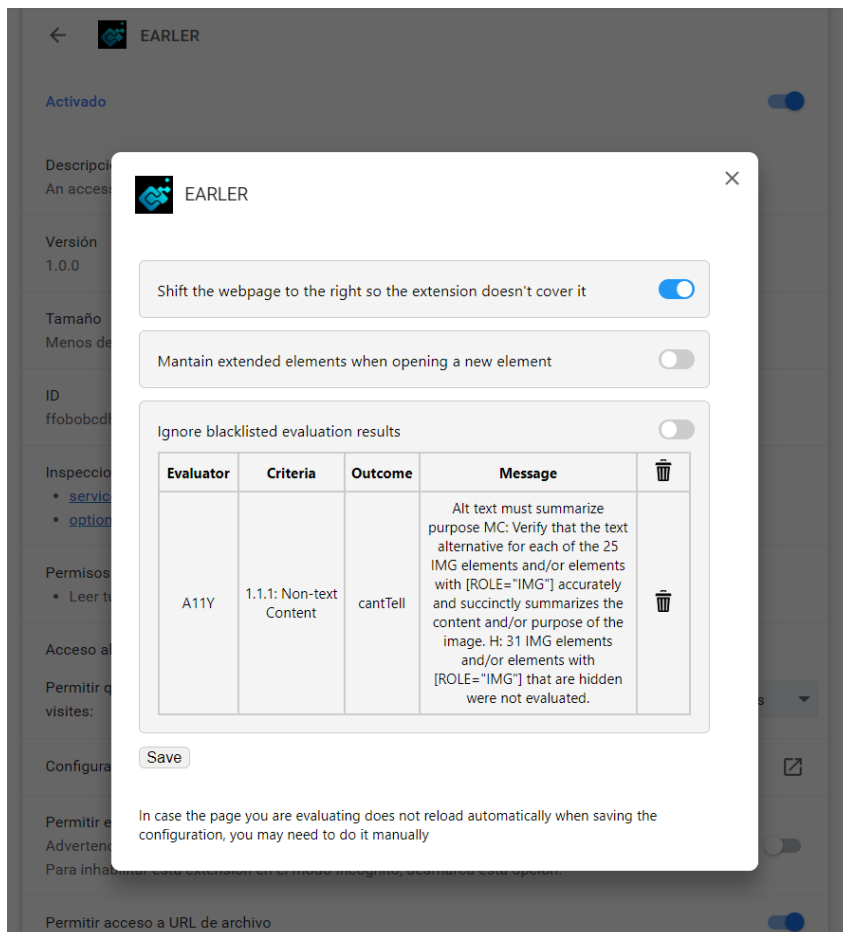


6.29 Irudia: Luzapena alde batean ezkutatzeko funtzionalitatea

## 6.1. Luzapenaren implementazioa



6.30 Irdia: Nabarmendutako elementuan kurtsorea jartzean azaltzen den popup elementua



6.31 Irdia: Luzapeneko konfigurazio orria

## 6.2 Zerbitzariaren implementazioa

Atal honetan zerbitzaria nola implementatu dugun, bete behar dituen eginkizun ezberdinetarako zein kontrolatzaile sortu ditugun eta horietako bakoitzari buruzko azalpen bat emango dugu.

### 6.2.1 Zerbitzariari egindako eskaeren kudeaketa

Esan bezala zerbitzaria http eta cors arazoak ekiditeko proxy liburutegien bidez inplementatzea pentsatu genuen. Hori horrela izan da implementazioaren denbora baterako, baina azkenean soluzio hobea bat bilatu dut. Express eta cors liburutegiak erabili ditut eskaerak jaso eta kudeatzeko, eta txostenak tamaina handia zutenez body-parser liburutegia ere erabili beharra izan dut, 6.32 irudian ikus daiteke eskaerak kudeatzeaz arduratzen den zerbitzariko index.js fitxategia.

```

11 const app = express();
12 app.use(cors()); // Enable CORS for all routes
13 app.use(bodyParser.json({ limit: '10mb' })); // Parse JSON request bodies with a higher limit
14
15
16 function routeHandlerWrapper(controllerMethod) {
17   return async (request, response) => {
18     try {
19       const body = await controllerMethod(request.body);
20       response.json(body);
21       console.log("\n### ----- Request SUCCESS ----- ###");
22     } catch (error) {
23       response.sendStatus(500);
24       console.log("\n ERROR: \n" + error + "\n\n### ----- Request FAILURE ----- ###");
25     }
26   };
27 }
28
29 const userController = new UserController();
30 const reportController = new ReportController();
31
32
33 app.post('/scrapeAccessibilityResults', routeHandlerWrapper(scraperController));
34 app.post('/userAuthentication', routeHandlerWrapper(userController.handleUserRequest.bind(userController)));
35 app.post('/reportStoring', routeHandlerWrapper(reportController.handleReportRequest.bind(reportController)));
36
37
38 // Start the server listening on the specified port.
39 const PORT = 7070;
40 app.listen(PORT, () => {
41   console.log("listening on: 0.0.0.0:" + PORT);
42 });

```

6.32 Irudia: Zerbitzariko eskaeren kudeaketarako index.js fitxategia

Irudian ikusten den bezala, eskaerak 3 kontrolatzailetara bideratzen dira eskaerako URL balioaren arabera:

- ScrapeController: ebaluaketa automatikoak egiteko eskaerez arduratzeko.
- UserController: erabiltzaileen kontuan sartzeko eta erregistratzeko.
- ReportController: ebaluaketa txostenak gorde eta kudeatzeko.



## 6.2.2 ScrapingController

Ebaluaketa automatikoei dagokionez, aipatzeko puntu batzuk:

- Hasieran ez nintzen konturatu hainbat web orrialde ebaluatzeko aukera izateak, liburutegiak erabiltzeko orrialde horietara joan beharko nuela suposatzen zuela. Ondorioz, liburutegiakin ere Puppeteer erabili beharra izan dut orrialde bakoitzera joan eta emaitzak lortu ahal izateko.
- Pa11y liburutegiaren kasuan, bera arduratzen da azpitik Puppeteer erabiltzeaz, eta ondorioz ez dut berak erabili dezan orrialderik pasa behar.
- Lighthouse liburutegiak arakatzailerik baten instantzia bat besterik ez du behar, beraz Puppeteer bidez sortutako arakatzailerik pasatzeaz aparte ez dut orrialderik sortu behar.

Ebaluatzaile bakoitzarekin sortzen diren txostenak bakarrean bateratzeko, 6.33 irudian ikusten den moduan, 6.34 funtzioa erabili dut.

```

const results = await Promise.all(activeEvaluators.map(async (evaluator) => {
  console.log("\nInitiating " + evaluator.toUpperCase() + " scraping process ...");

  const jsonLd = new JsonLd(evaluator, request.scope);
  const scraper = new Scraper(evaluator, jsonLd);

  switch (evaluator) {
    case "lh":
      await withBrowser(async (browser) => {
        for(const webPage of request.scope){
          await scraper.performScraping(webPage, null, browser);
        }
      });
      break;

    case "pa":
      await Promise.all(request.scope.map(async (webPage) => {
        await scraper.performScraping(webPage);
      }));
      break;

    default:
      await withBrowser(async (browser) => {
        await Promise.all(request.scope.map(async (webPage) => {
          await withPage(browser)(async (page) => {
            await scraper.performScraping(webPage, page);
          });
        }));
      });
      break;
  }

  console.log(`\n${evaluator.toUpperCase()} scraping successfully finished !!!`);

  return jsonLd.getJsonLd();
}));

console.timeEnd('Execution Time');

for(let i = 1; i<results.length; i++){
  mergeJsonLds(results[0], results[i]);
}

```

6.33 Irudia: ScrapingController ebaluaketa automatikoa

## 6. INPLEMENTAZIOA

```
function mergeJsonLds(jsonLd1, jsonLd2){
  try{
    jsonLd1.assertors.push(...jsonLd2.assertors);
    jsonLd1.creator["xmlns:name"] += " & " + jsonLd2.creator["xmlns:name"];

    for (let i = 0; i < jsonLd1.auditSample.length; i++) {
      const newOutcome = jsonLd2.auditSample[i].result.outcome;

      if(newOutcome === "earl:untested") continue;

      const currentOutcome = jsonLd1.auditSample[i].result.outcome;

      if(currentOutcome === "earl:untested"){
        jsonLd1.auditSample[i] = jsonLd2.auditSample[i];
      } else {
        if((currentOutcome === "earl:inapplicable" && newOutcome !== "earl:inapplicable")
          || (currentOutcome === "earl:passed" && (newOutcome === "earl:cantTell" || newOutcome === "earl:failed"))
          || (currentOutcome === "earl:cantTell" && newOutcome === "earl:failed")){
          jsonLd1.auditSample[i].result = jsonLd2.auditSample[i].result;
        }

        mergeFoundCases(jsonLd1.auditSample[i].hasPart, jsonLd2.auditSample[i].hasPart);
      }
    }
  } catch(error) {
    throw new Error("Error merging JsonLd reports => " + error.message);
  }
}
```

6.34 Irudia: Ebaluatzaileen ebaluaketa txostenak bateratzeko funtzioa

### 6.2.3 UserController

Kasu honetan ez dago ezer berezirik erakusteko. 2 funtzio nagusi daude, bata erabiltzaileak erregistratzeko eta bestea logeatzeko, ikus 6.35 eta 6.36 irudiak.

```
// Function to register a new user
async #registerUser(email, username, password) {
  return this.#checkUserExists(email)
    .then((userExists) => {
      if (userExists) return {"success": false};

      return new Promise((resolve, reject) => {
        this.#db.serialize(() => {
          this.#db.run(`INSERT INTO users (email, username, password) VALUES (?, ?, ?)`, [email, username, password], (err) => {
            if (err){
              console.error(err.message);
              reject(err);
            } else {
              resolve({"success": true});
            }
          });
        });
      });
    });
}
```

6.35 Irudia: Erabiltzailea datu-basean erregistratzeko kodea

```

// Function to login a user
async #loginUser(email, password) {

  return this.#checkUserExists(email)
  .then((userExists) => {

    if (!userExists) return {"success": false, "error": "User does not exist"};

    return new Promise((resolve, reject) => {
      this.#db.get("SELECT * FROM users WHERE email = ? AND password = ?", [email, password], (err, row) => {
        if (err) {
          console.error(err.message);
          reject(err);
        } else {
          resolve(row ? {"success": true, "username": row.username} : {"success": false, "error": "Wrong password"});
        }
      });
    });
  });
}

```

6.36 Irudia: Erabiltzailea datu-basean dagoen egiaztatze kodea

### 6.2.4 ReportController

Txostenen kudeaketari dagokionez, ez da konplexua baina sekuentzia diagrametan azaltzen ez zenez, txostenak ezabatzeko funtzioa azalduko dut, ikus 6.37. Funtzio horren bitartez, lehenik ezabatuko den txostenaren parentId balioa lortu eta aldagai batean gordetzen da, ondoren txostena ezabatu eta azkenik ezabatutako txostenaren seme guztien parentId-ak aldatzen dira ezabatutako txostenaren gurasoarenarekin.

```

async #removeReport(id) {

  const parentId = await new Promise((resolve, reject) => {

    this.#db.all(
      "SELECT parentId
      FROM reports
      WHERE id = ?",
      [id], function(err, rows) {
        if (err) {
          reject(err);
        } else {
          resolve(rows[0].parentId);
        }
      });
  });

  return new Promise((resolve, reject) => {

    this.#db.run(
      "DELETE FROM reports
      WHERE id = ?",
      [id], function(err) {
        if (err) {
          reject(err);
        }
      });

    this.#db.run(
      "UPDATE reports
      SET parentId = ?
      WHERE parentId = ?",
      [parentId, id], function(err) {
        if (err) {
          reject(err);
        } else {
          resolve({"success": true});
        }
      });
  });
}

```

6.37 Irudia: Datu-basetik txostena ezabatzeko funtzioa

### 6.3 Aurkitutako arazoak

Atal honetan inplementazio prozesua aurrera joan ahala aurkitzen joan garen arazoak azalduko ditugu. Mota ezberdinetakoak izan dira: orrialdeei estiloa emateko, nabigatzailan datuak gordetzeko, scraping egiteko, ebaluaketa ematen irizpideetan aurkitutako kodeetarako erreferentziekin HTML elementuak eskuratzeko, eta ebaluaketa automatikoak egiteko.

#### 6.3.1 Estiloen arazoak

Proiektuaren garapena aurrera joan ahala, kokatutako webgunearen arabera luzapenaren itxura aldatzen dela kontura naiz. Aldaketa horiek webgune ezberdinek atzetik estilo aldaketak egiten dituztelako, eta segur aski horretarako id edota class bezalako HTML elementuen atributuak erabili beharrean zuzenean HTML elementuen tag propioak erabiltzen dituztelako gertatzen dira.

Estiloa ahalik eta zehatzena eginez gutxi gorabehera itxura mantentzea lortu dut, baina oraindik nabaria izan daiteke webgune batzuen kasuan.

Seguruenik, horretarako soluzio hobereana luzapenerako estiloak isolatzeko edo antzeko zerbaiteko tresna bat bilatzea eta erabiltzea izango zen, baina gure kasuan alde batera uztea eta aurrera segitzea erabaki dugu.

#### 6.3.2 LocalStorage arazoak

Aurkitu dudan beste arazo bat, webgune batzuek beraien orrialdeak birkargatzean localStorage osoaren reset bat egiten dutela da, hau da, bertan gordetako balio guztiak ezabatzen dituzte, ondoren defektuzkoak berriro ezartzeko.

Horrek, egindako inplementazioan arazo garrantzitsu bat suposatu du, localStorage hainbat gauza gordetzeko erabiltzen nuelako: ebaluatzeo egindako orrialdeen aukeraketa, aukeratutako ebaluatzaileak, unean kargatutako txostenaren gurasoaren id balioa txostenaren bertsio berria gordetzean erabiltzeko eta kokatutako webgunearan txosten bat kargatuta dagoen ala ez jakiteko.

Horien artean, gutxienezko oinarrizko funtzionalitateak mantentzeko, txostenik kargatuta dagoen jakiteko eta txosten gurasoen id-ak gordetzeko zuzenketa bat egin dut.

Zuzenketa hori egiteko, chrome.storage.local erabili dut aipatutako balio hori luzapenean bertan gordetzeko, baina horren arazoa, localStorage baliabidearekin ez bezala, gordetako balioak webgune guztientzako berdina direla da. Nire kasuan webgune bakoitzarentzako balio ezberdinak gorde behar ditudanez, balioak gordetzeko "reportIsLoaded" gako erabili beharrean, webgunearan domeinua + ".reportIsLoaded" moduan gorde behar izan ditut.

Gainontzeko balioentzako soluzio bera erabil daiteke, baina nire kasuan denboraren ondorioz zuzenketa horiek alde batera uztea erabaki dut.

#### 6.3.3 Scraping egitearen arazoak

Arriskueta aipatu bezala, scraping egitean zure esku ez dauden aldaketen menpe zaude zure zerbitzuak funtzionatzen segitzeko, eta AccessMonitor tresnaren kasuan arazo hori

gertatu da.

Lehenago ebaluatzeko orrialdearen esteka URL-an bertan adieraz zitekeen parametro moduan, baina orain derrigorrezkoa da interfazearekin interakzioa egitea.

Gainera, `<a href="..." />` HTML elementuen linkak segitzean ere lehen nahikoa zen horien href atributuko esteka lortu eta zuzenean joatea. Orain elementuan sakatzea derrigorrezkoa da, estekak eramaten duen orrialdeko informazioa kargatzeko.

Kasu honetan, sortutako arazoa ez da zuzentzeko oso zaila izan eta azkar konpondu ahal izan dut.

### 6.3.4 HTML elementuak eskuratzeko erreferentziekin arazoak

Orrialdeko elementuak nabarmentzerako garaian, path batzuek ezkutatutako elementuetara eramaten dute. Emaitzen taulan azaltzen diren kasu horiek kolore gabe utzi ditugu gorriz beharrez, horiek klikatzean ez delako ezer gertatzen. Berez, automatikoki garatzaile tresnetan irekitzea eta ezkutatutako elementua bertan erakustea bururatu zitzaigun, baina ez dut hori egiteko modurik aurkitu.

### 6.3.5 Ebaluaketa automatikoen arazoak

Orokorrean ebaluatzaileek ez dituzte erroreak ematen, baina noizbehinka ebaluatzaile baten batek orrialdea kargatzeko garaian blokeatuta geratzen dela dirudi eta ondorioz timeout errorea eragiten dute.

Ez dakit zergatik izan daitekeen, balieke nire interneteko gauza izatea, puppeteer instantzia gehiegi aldi berean izatea, edo beste edozer gauza. Ez dakigunez arrazoia, eta noizbehinka bakarrik gertatzen denez, konpondu gabe uztea erabaki dugu.



# Probak

Kapitulu honetan, Earler tresnaren funtzionamendu egokia egiaztatzeko egindako probak aztertuko ditugu. Zehazki honako funtzionalitateak izango ditugu kontuan: ebaluaketa automatikoak egitea [7.1](#), txostenen emaitzak aldatzea [7.2](#) eta txostenak datu-basean gordetzea eta bertatik kargatzea [7.3](#).

## 7.1 Ebaluaketa automatikoen probak

Ondorengo azpiataletako ebaluaketa automatikoen probak bi ezaugarritan oinarrituko dira, alde batetik ebaluaketen azkartasunean, eta bestetik emaitzen zuzentasunean. Azkartasuna ebaluatzaileek orrialde ezberdinetan irisgarritasun ebaluazioak egiteko behar dituzten denborak neurtuz aztertuko da. Zuzentasunarentzako, hau da, ebaluatzaileetatik jasotzen ditugun emaitzak egokiak diren jakiteko, ACT test kasuak erabiliko ditugu.

### 7.1.1 Denbora probak

Atal honetan ebaluatzaile bakoitzak duen errendimendua ikusiko dugu, ebaluaketak egiteko behar dituzten denborak aztertuz (segundutan).

Probak 4 motakoak izango dira eta ehu.eus eta ataria.eus webguneetan gauzatuko dira. Mota horiek honakoak dira:

- Ebaluatzaileak banaka orrialde bakarra aztertuz.
- Ebaluatzaileak banaka 3 orrialde aldi berean aztertuz.
- Ebaluatzaile guztiak batera 3 orrialde aldi berean aztertuz.
- Ebaluatzaile guztiak (AChecker kenduta) 3 orrialde aldi berean aztertuz.

Proba horien emaitzak [7.1](#) eta [7.2](#) irudietako tauletan ikus daitezke. Kontuan izan, gainontzeko ebaluatzaileekin ez bezala, Lighthouse liburutegiaren kasuan orrialdeak seriean ebaluatzen direla.

## 7. PROBAK

Lehendabizi, oso agerikoa egiten da AChecker ebaluatzaileak gainontzeko ebaluatzaileek baina denbora askoz gehiago behar duela eta ondorioz orokorrean ez da oso erabilgarria suertatzen. Izatekotan, orrialde bakarra ebaluatzeko kasuetan erabiltzea legoke, baina orokorrean gehiegizko denborak behar ditu erabiltzailearentzako interesgarria izan dadin.

ehu.eus					
Ebaluatzaileak	Orrialdeak	Denbora 1	Denbora 2	Denbora 3	Batez beste (s)
AccessMonitor	es/home	24,594	19,742	20,674	21,67
AChecker		69,52	68,238	78,466	72,07466667
Mauve		19,575	18,955	16,577	18,369
A11y		2,968	2,871	2,863	2,900666667
Pa11y		3,643	3,827	3,402	3,624
Lighthouse		8,623	8,797	9,472	8,964
AccessMonitor	es/home, es/web/ikasleak, es/web/iip	22,577	22,034	23,577	22,72933333
AChecker		Timeout > 180	167,095	Timeout > 180	~175 >
Mauve		23,508	19,29	17,749	20,18233333
A11y		3,274	3,101	4,149	3,508
Pa11y		5,402	8,678	5,743	6,607666667
Lighthouse		26,186	25,081	25,523	25,59666667
Denak batera	es/home, es/web/ikasleak, es/web/iip	178,49	AChecker Timeout > 180	179,691	~180 >
Denak AChecker ezik	es/home, es/web/ikasleak, es/web/iip	34,268	37,777	34,977	35,674

7.1 Taula: ehu.eus webgunerako ebaluaketa automatikoen denbora probak

ataria.eus					
Ebaluatzaileak	Orrialdeak	Denbora 1	Denbora 2	Denbora 3	Batez beste (s)
AccessMonitor	/	21,359	20,387	30,009	23,91833333
AChecker		66,18	85,19	72,743	74,70433333
Mauve		17,936	18,071	19,013	18,34
A11y		4,688	4,64	4,667	4,665
Pa11y		6,502	6,298	5,934	6,244666667
Lighthouse		14,071	13,798	13,655	13,84133333
AccessMonitor	/, agenda/ kideak/abantailak/	26,538	26,078	40,692	31,10266667
AChecker		Timeout > 180	Timeout > 180	177,383	~180 >
Mauve		28,238	17,847	17,402	21,16233333
A11y		5,442	5,419	5,414	5,425
Pa11y		32,562	8,434	8,567	16,521
Lighthouse		32,007	36,624	37,241	35,29066667
Denak batera	/, agenda/ kideak/abantailak/	179,742	170,057	Timeout > 180	~175 >
Denak AChecker ezik	/, agenda/ kideak/abantailak/	48,615	52,668	46,125	49,136

7.2 Taula: ataria.eus webgunerako ebaluaketa automatikoen denbora probak

Bestalde, orrialde bakarra ebaluatzean, Lighthouse liburutegia aukera ona izan daiteke denbora aldetik nahiko azkarra delako, baina 3 orrialdetik aurrera nahiko motela



bilakatzen da, egindako ebaluatzaileen aukeraketaren arabera "bottleneck" efektua sortzeko arriskuarekin.

Gainontzeko ebaluatzaileei dagokionez, orrialdeen ebaluaketen paralelizazioak positiboki eragiten diela ikus daiteke, eta horien arteko edozein konbinaketa eginda, orrialde kopuru gehiegizkoa ezartzen ez den bitartean, emaitzak erabiltzailearentzako denbora onargarrietan lortuko dira.

Gainera, espero zitekeen bezala, a11y eta pa11y ebaluatzaileak liburutegiak direla kontuan izanik eta gainontzeko ebaluatzaileekin konparatuz, denbora aldetik oso azkarrak dira.

Ondoren, AChecker-en arazoa ikusi dugunez alde batera utziko dugu eta gainontzeko ebaluatzaileekin, proba mota berdinak segiz, webgune gehiagoren azterketa egingo dugu. WebAIM webgunean<sup>1</sup> urtero 1,000,000 webguneentzako irisgarritasun azterketak egiten dituzte, eta webgune horien aukeraketarako 3 zerrendaz baliatzen dira. Beraz, guk ere zerrenda horietako batetik aukeratzea erabaki dugu. Zehazki, The Majestic Million<sup>2</sup> zerrendako lehen 2 webguneak ebaluatuko ditugu, ikus 7.3 irudia. Berez, webgune gehiagorekin egitea pentsatzen genuen baina denbora falta gatik 2 egitera mugatu naiz.

The Majestic Million

The million domains we find with the most referring subnets

Free search and download of the top million websites, from majestic.com.

Want more Big Data?  
Commission a bespoke Majestic Analytics report  
[Check it out](#)

Search: example1.com, example2.com [SEARCH](#)

Type: Majestic Million

Filter: Select TLD

Export: CSV (~80MB)

License: Licensed under a [Creative Commons Attribution 3.0 Unported License](#)

Position +/-	Domain	TLD	TLD Rank	Referring Subnets	Referring IPs
1	google.com	.com	1	495,673 484 ↑	2,403,393 1,532 ↑
2	facebook.com	.com	2	494,748 410 ↑	2,556,034 1,458 ↑

**7.3 Taula:** The Majestic Million zerrendatik ebaluatuko ditugun hoberen klasifikatutako 8 webguneak

Ondorengo, 7.4 eta 7.5 tauletan, proba ezberdinetarako jasotako probak gordetzen dira. Google-ko hasiera orrialdea ebaluatzean AccessMonitorren emaitzak jasotzeko eginda dudan scrapperra itxaroten geratzen da timeout eman arte. Arraroa da beste webguneekin funtzionatzen duelako baina baliteke nire errorea izatea. Facebook webguneko ebaluaketa emaitzen artean, Pa11y-ren kasuan liburutegiak berak ez du ebaluaketa amaitzen eta timeout errorea ematen du, beraz ezin dut horri buruz ezer gehiago esan.

<sup>1</sup><https://webaim.org/>

<sup>2</sup><https://majestic.com/reports/majestic-million>

## 7. PROBAK

Google.com					
Ebalutzaileak	Orrialdeak	Denbora 1	Denbora 2	Denbora 3	Batez beste (s)
AccessMonitor	/	errorea	errorea	errorea	0
Mauve		5,81	5,335	6,634	5,926333333
A11y		2,437	2,15	2,231	2,272666667
Pa11y		2,327	1,612	1,594	1,844333333
Lighthouse		9,997	9,995	9,897	9,963
AccessMonitor	/, /imghp, /preferences	errorea	errorea	errorea	0
Mauve		9,054	8,551	8,686	8,763666667
A11y		3,957	3,88	3,913	3,916666667
Pa11y		1,962	1,921	1,921	1,934666667
Lighthouse		30,334	30,588	31,645	30,855666667
Denak batera	/, /imghp, /preferences	30,316	30,132	32,052	30,833333333

7.4 Taula: google.com webgunerako ebaluaketa automatikoen denbora probak

facebook.com					
Ebalutzaileak	Orrialdeak	Denbora 1	Denbora 2	Denbora 3	Batez beste (s)
AccessMonitor	/	10,832	10,472	10,397	10,567
Mauve		5,007	4,567	4,284	4,619333333
A11y		1,519	1,541	1,483	1,514333333
Pa11y		errorea	errorea	errorea	0
Lighthouse		11,624	11,124	11,176	11,308
AccessMonitor	/, /pages/create/ /lite/	15,252	15,004	14,841	15,032333333
Mauve		6,656	7,182	7,563	7,133666667
A11y		1,818	2,057	2,029	1,968
Pa11y		errorea	errorea	errorea	0
Lighthouse		31,555	30,412	32,67	31,545666667
Denak batera	/, /pages/create/ /lite/	33,124	32,47	32,861	32,818333333

7.5 Taula: facebook.com webgunerako ebaluaketa automatikoen denbora probak

Erroreak alde batera utzita, denborak aurreko kasuetan lortutako oso antzekoak dira. Nabarmen ikusten den ezberdintasuna Mauve++ ebalutzaileak azken bi webguneak ebaluatzen errendimendu hobea izan duela da, baina horretaz aparte emaitzak esperotakoak dira eta ez dago ezer berezirik kontatzeko.

### 7.1.2 Emaitzen test kasuak

Ebalutzaileen emaitzen zuzentasuna neurtzeko, W3C zehaztutako ACT test kasuak erabiltzeko ditugu. Test kasu horiek 7.1 egitura duen JSON fitxategi batean eskuragarri daude<sup>3</sup>. Egitura horretan test kasuak "testcases" atributuko zerrendan kokatzen dira, eta "count" atributuaren balioan ikus daitekeen bezala dena 1132 test daude.

<sup>3</sup><https://www.w3.org/WAI/content-assets/wcag-act-rules/testcases.json>

```

{
  "name": "ACT Task Force test cases",
  "website": "https://www.w3.org/WAI/standards-guidelines/act/rules/",
  "license": "https://act-rules.github.io/pages/license/",
  "count": 1132,
  "testcases": [
    {
      "ruleId": "97a4e1",
      "ruleName": "Button has non-empty accessible name",
      "ruleAccessibilityRequirements": {
        "wcag20:4.1.2": {
          "forConformance": true,
          "failed": "not satisfied",
          "passed": "further testing needed",
          "inapplicable": "further testing needed"
        }
      },
      "expected": "passed",
      "testcaseId": "a4cc71b0434f71f4ea0069c409f73e0207dfb403",
      "testcaseTitle": "Passed Example 1",
      "relativePath": "testcases/97a4e1/a4cc71b0434f71f4ea0069c409f73e0207dfb403.html",
      "url": "https://www.w3.org/WAI/content-assets/wcag-act-rules/testcases/97a4e1/a4cc71b0434f71f4ea0069c409f73e0207dfb403.html",
      "rulePage": "https://www.w3.org/WAI/standards-guidelines/act/rules/97a4e1/proposed/"
    }, ... ]
  }
}

```

### 7.1 Irudia: ACT test kasuen JSON fitxategiaren egitura

Test kasu horien artean, "ruleAccessibilityRequirements" atributuan WCAG estandarreko irizpideak dituztenak interesatzen zaizkigu, eta ondorioz, 1132 horietatik errealitatean 996 izango ditugu kontuan.

Test bakoitzerako, ebaluatzaileekin "url" eremuan adierazten den estekan ebaluatuz lortutako irizpidearen emaitza "expected" balioaren berdina bada (passed, failed edo inapplicable), edo "expected": "inapplicable" izanik ebaluatzaileak irizpide horrentzako "passed" edo "failed" EZ den emaitza bat itzultzen badu, testa ontzat hartuko da, eta bestela okerra izango da.

Gainera, kontuan izan behar da ebaluatzaileek ez dituztela irizpide guztientzako emaitzak ematen, eta ondorioz kasu askotan ez dela konparatzeko emaitzarik egongo. Kasu horietan testak "untested" izan direla esango dugu, eta beraz, testak 3 emaitza mota izan ditzakete: passed, failed, eta untested.

Testak egiteko zerbitzarian testController.js izeneko kontroladore berri bat sortu dut, ebaluatzaileen scraping egiteko funtzio sinplifikatuak erabiltzen dituena. Sinplifikatze hori, ebaluatzaileen emaitzetan irizpide bakoitzeko emaitzak bakarrik behar ditugulako da, hau da, aurkitutako kasuetarako erreferentziak ez ditugu behar, eta ondorioz testak azkarrago egiteko aukera dugu. Horretaz gain, ebaluatzaileak paraleloan exekutatzeko Promise.all batez ere baliatzen da.

## 7. PROBAK

### 7.1.2.1 Test kasuen emaitzak

Hasteko, 7.6 taulan ebaluatzaile bakoitzak banaka izan dituen passed, failed, eta untested testen emaitza kopuruak agertzen dira. Portzentaiek, untested izan ez diren testen artetik asmatutakoen proportzioa adierazten dute. Kontuan izan, pa = Pa11y, mv = Mauve++, ac = AChecker, am = AccessMonitor eta lh = Lighthouse ebaluatzaileak direla.

Ebaluatzailea	passed tests	failed tests	passed %	untested tests
a11y	202	493	29,0647482	301
pa	60	81	42,55319149	855
mv	151	175	46,3190184	670
ac	71	249	22,1875	676
am	197	289	40,53497942	510
lh	215	327	39,66789668	454

7.6 Taula: Ebaluatzaile bakoitzak testak pasatzean jasotako emaitzak

Asmatutakoen portzentaia bakarrik kontuan izanik, hoberenak pa, mv, am, eta lh ebaluatzaileak izan dira, baina testak egin gabe utzi dituzten kopuruei erreparatuz, 4 horien artetik am eta lh aukera hobekak izango lirakeke, irizpide gehiagoren emaitzak ematen dituztelako eta aldi berean asmatze portzentaia mantentzen dute. Dena dela, orokorrean asmatze portzentaia nahiko baxuak dira eta ez da gomendagarria ebaluatzaileak banaka erabiltzea.

Ondoren, 7.7 taulan, zenbat testek izan duten gutxienez passed lortu duen ebaluatzaile bat, zenbatek bi, eta horrela 6 arte azaltzen da.

Test bakoitza kontatzeko gutxienezko passed kopurua	Gutxienezko passed kopurua betetzen duten test kopurua
1	531
2	254
3	88
4	21
5	2
6	0

7.7 Taula: Gutxienezko passed kopurua izan dituzten test kopuruak

Lehen lerroarekin honakoa jakin dezakegu, gehienez jota eta ebaluatzaile bakoitzerako emaitza egokia aukeratuz, ebaluatzaile guztiekin 531 test pasako genituzke. Horrek ez du esan nahi gainontzekoak failed izan direnik, horietako asko untested izango dira.

Gainera, test berean passed emaitza lortzen dituzten 2 ebaluatzaile egotea test guztietatik %25-ean baina gutxiagotan gertatzen da, eta 3 ebaluatzaile egotea %10-ean baina gutxiagotan. Hori dela eta, ebaluatzaileen emaitzak nahiko sakabanatuak direla ondoriozta dezakegu eta orokorrean beraien artean konpentsatuko dira.

Dena dela, interesgarria izan daiteke ebaluatzaile konbinaketa hoberenak zeintzuk diren ikustek. 7.8 taulan 6 konbinazio hoberenak ikus daitezke, eta 7.9 taulan, AChecker tresnak duen errendimendua kontuan izanda aukera posibleetatik baztertuz, gainerako ebaluatzaileen arteko 6 konbinaketa hoberenak azaltzen dira.

Konbinaketak	Pasatako test kopurua
guztiak	531
a11y, pa, mv, am, lh	530
a11y, mv, ac, am, lh	511
a11y, mv, am, lh	507
a11y, pa, mv, ac, am	498
a11y, pa, mv, am	490

7.8 Taula: Ebaluatzaile guztien arteko konbinazio hoberenak

Konbinaketak	Pasatako test kopurua
a11y, pa, mv, am, lh	530
a11y, mv, am, lh	507
a11y, pa, mv, am	490
pa, mv, am, lh	477
a11y, mv, am	463
mv, am, lh	452

7.9 Taula: AChecker kenduta, gainontzeko ebaluatzaileen arteko konbinazio hoberenak

2 taula horietatik ondorio ezberdinak atera daitezke, baina nire ustez garrantzitsuenak 2 dira:

Alde batetik, orrialde bat edo bi ebaluatu behar baditugu, aukera hoberena ebaluatzaile guztiak AChecker kenduta erabiltzea izango litzateke, emaitzak azkar lortuko lirateke eta emaitzen asmatzea maximizatuko litzateke.

Bestetik, 3 orrialde edo gehiago ebaluatu beharko balira, Lighthouse liburutegiak azkartasun aldetik atzean gera daiteke eta hobe izango litzateke ez erabiltzea. Hori kontuan izanda, aukera hoberena a11y, pa, mv, eta am aukera guztiak erabiltzea izango litzateke.

## 7.2 Txostenen emaitzak aldatzeko probak

Emaitzen aldaketari dagokionez, ezabaketa eta edizio aukerak ondo lan egiten dutela egiaztatzeko probak egingo ditugu eta horretarako 7.2(a) irudian ikusten den txostena erabiliko dugu.

### 7.2.1 Irizpide bateko aurkitutako kasuen ezabaketa

Proba honetarako, 7.2(a) irudian ikusi zitekeen 1.1.1 irizpidearen bi kasuak kenduko ditugu. Konturatu, summary taulako (taula txikiko) emaitzak ere aldatzen direla.

Lehenengo, cantTell emaitzeko kasua ezabatzen badugu, geratzen diren kasuen artean emaitza okerreana passed denez, irizpidearen emaitza cantTell izatetik passed izatera aldatzen da, ikus 7.2(b) irudia.

Ondoren, geratzen den passed kasua kenduta, irizpideak kasu gehiagorik ez duenez untested izatera pasatzen da, ikus 7.2(c) irudia.

Current report results

Select conformace level:

A AA AAA

Current Site/Page summary:

Website Inicio - UPV/EHU

P	F	CT	NP	NC
3	3	27	4	41

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
1 Perceivable	0	2	7	3	17
1.1 Text Alternatives	0	0	1	0	0
^ 1.1.1: Non-text Content		earl:cantTell			
v cantTell		✖			
v passed		✔			
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.4 Distinguishable	0	1	4	0	8
2 Operable	1	0	10	1	17
3 Understandable	1	0	10	0	6
4 Robust	1	1	0	0	1

Current report results

Select conformace level:

A AA AAA

Current Site/Page summary:

Website Inicio - UPV/EHU

P	F	CT	NP	NC
4	3	26	4	41

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
1 Perceivable	1	2	6	3	17
1.1 Text Alternatives	1	0	0	0	0
^ 1.1.1: Non-text Content		earl:passed			
v passed		✔			
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.4 Distinguishable	0	1	4	0	8
2 Operable	1	0	10	1	17
3 Understandable	1	0	10	0	6
4 Robust	1	1	0	0	1

Current report results

Select conformace level:

A AA AAA

Current Site/Page summary:

Website Inicio - UPV/EHU

P	F	CT	NP	NC
3	3	26	4	42

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
1 Perceivable	0	2	6	3	18
1.1 Text Alternatives	0	0	0	0	1
^ 1.1.1: Non-text Content		earl:untested			
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.4 Distinguishable	0	1	4	0	8
2 Operable	1	0	10	1	17
3 Understandable	1	0	10	0	6
4 Robust	1	1	0	0	1

((a)) Emaitzen aldaketa probatzeko erabiliko dugun txostena

((b)) 1.1.1 irizpideko cantTell kasua kendu ondoren

((c)) 1.1.1 irizpideko passed kasua kendu ondoren

7.2 Irudia: 1.1.1 irizpideko kasuak kentzen

### 7.2.2 Irizpide bateko kasu baterako ebaluatzaileen ezabaketa

Proba honetan 1.3.1 irizpideko failed kasuko ebaluatzaileen ezabaketa probatuko dugu, ikus 7.3. Kontuan izan ebaluatzaile bat kentzeak dagokion deskripzioa eta kode zatietarako erreferentziak kentzea adierazten duela.

Lehenik, A11y-ren emaitzak ezabatzen baditugu (@a11y ebaluatzailearen eskuineko x ikonoan klik eginez), oraindik failed kasuko P11y ebaluatzaileak aurkitutako emaitzak izango ditugu, eta ondorioz irizpide failed izaten segiko du, ikus 7.4(a) irudia.

Ondoren, P11y-ren emaitzak ezabatuz, ez denez failed kasu horren emaitzarik aurkitu duen ebaluatzailearik geratzen, failed kasua automatikoki ezabatzen da eta irizpidearen emaitza geratzen diren kasuetatik okerreza izatera pasatzen da, hau da, cantTell izatera, ikus 7.4(b) irudia.

Current Site/Page summary:

Website Inicio - UPV/EHU

P	F	CT	NP	NC
3	3	26	4	42

Current webpage evaluation results:

Standard P F CT NP NC

1.3.1: Info and Relationships earl.failed

failed

@A11y

All content must be contained in landmarks V: Update the landmark structure of the page by placing the one element not contained in a landmark into a container element with a proper landmark role. H: 55 hidden elements with renderable content were found. If any could become visible make sure they are in container elements with proper landmark roles.

@Pa11y

This element's role is "presentation" but contains child elements with semantic meaning.

[A11Y]

- <h1 class='hide-accessib ...
- <a href='#main-content& ...
- <span lang='eu'> ...

Standard P F CT NP NC

- <span>Grados</span> ...
- <span>Másteres y Títu ...
- <span>Doctorado</s ...
- <span>eCampus: campus ...

[Pa11y]

- <li role='presentation'& ...
- <li class='sign-in btn b ...

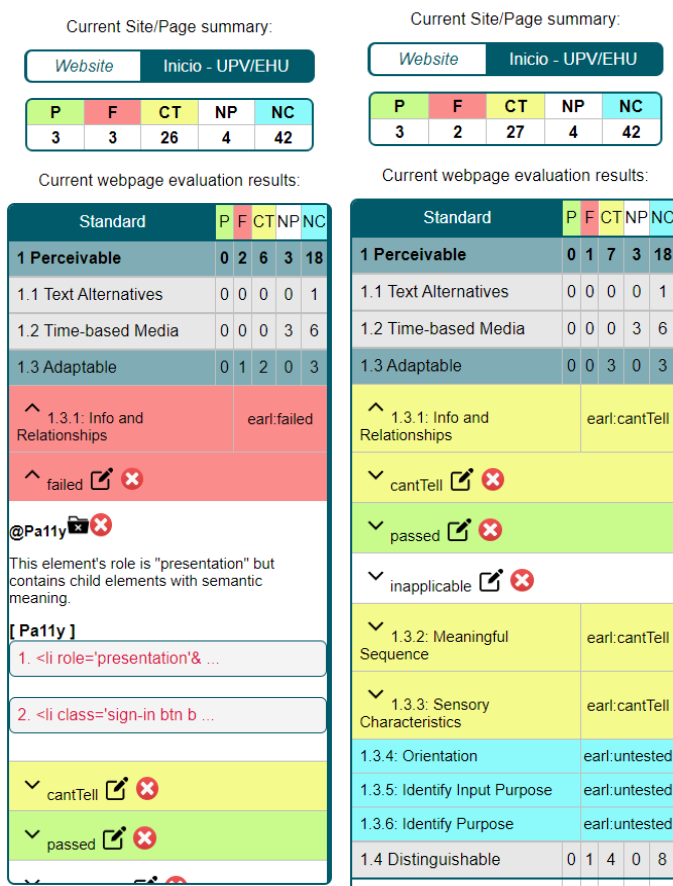
cantTell

passed

inapplicable

1.3.2: Meaningful Sequence earl.cantTell

7.3 Irudia: 1.3.1 irizpideetatik ebaluatzaileak kenduko dizkiogun failed kasua



((a)) A11y ebaluatzailea kentzean ((b)) Pa11y ebaluatzailea kentzean

7.4 Irudia: 1.3.1 irizpideko failed kasuko ebaluatzaileak kentzen

7.2.3 Irizpide bateko kasu baterako edizioa

Orain, 1.3.1 irizpideko cantTell kasuko edukia editatuko dugu, kasuaren eskuinean azaltzen den lehen ikonoa sakatuz, edizio moduan sartu ondoren a11y-ren deskripzioa aldatuz eta lehen erreferentzia ezabatuz, ikus 7.5.

Aldaketak egin ondoren Cancel botoia sakatzen bada egindako aldaketak deusezten dira, eta bestela, Save botoia sakatzen bada aldaketak erabiltzailearen izenean gordetzen dira. Guk egindako aldaketak a11y ebaluatzailearen gainean izan direnez, edizioa egin duen erabiltzailearen izena a11y-ren alboan bakarrik azalduko da bi ebaluatzaileetarako beharrean, ikus ?? irudia.

Edizioak egin ostean, adibidez beste ebaluatzaile ezberdin batzuekin ebaluaketa berri bat egiten badugu, editatutako emaitzak mantendu behar dute, hau da, gure kasuan @a11y-ren datuak mantenduko dira, baina @pa11y-ren datuak ez dituzenez inorek aldatu horiek berridatzi egingo dira, ikus 7.6(b) irudia.



## 7.2. Txostenen emaitzak aldatzeko probak

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	0	1	7	3	18
1.1 Text Alternatives	0	0	0	0	1
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	0	3	0	3

1.3.1: Info and Relationships

earl:cantTell

cantTell ✖

@A11Y ✖

REGION landmark must have accessible name

MC: Determine if any of the 19 elements with ARIA role of REGION should be landmarks, and if so, add an accessible name to the those elements.

@Pa11y ✖

If this selection list contains groups of related options, they should be grouped with optgroup.

[A11Y]

- <section class='layout-p ...
- <section class='portlet' ...

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	0	1	7	3	18

1.3.1: Info and Relationships

cantTell Save Cancel

@A11Y

REGION landmark must have accessible name

MC: Determine if any of the 19 elements with ARIA role of REGION should be landmarks, and if so, add an accessible name to the those elements.

@Pa11y

If this selection list contains groups of related options, they should be grouped with optgroup.

[A11Y]

- <section class='layout-p ...
- <section class='portlet' ...

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	0	1	7	3	18

1.3.1: Info and Relationships

cantTell Save Cancel

@A11Y

REGION landmark must have accessible name

@Pa11y

If this selection list contains groups of related options, they should be grouped with optgroup.

[A11Y]

- <section class='portlet' ...
- <section class='portlet' ...
- <section class='upv-ehu- ...

(a) Editatuko dugun cantTell kasua      (b) Edizio moduan sartzean      (c) Kasuko A11y-ren datuak aldatuz

### 7.5 Irudia: 1.3.1 irizpidetako cantTell kasuan edizio moduan sartu eta aldaketak egin

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	0	1	7	3	18
1.1 Text Alternatives	0	0	0	0	1
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	0	3	0	3

1.3.1: Info and Relationships

earl:cantTell

cantTell ✖

@A11Y - modifier: @Julen ✖

REGION landmark must have accessible name

@Pa11y ✖

If this selection list contains groups of related options, they should be grouped with optgroup.

[A11Y]

- <section class='portlet' ...
- <section class='portlet' ...
- <section class='upv-ehu- ...

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	0	1	7	3	18

1.3 Adaptable

1.3.1: Info and Relationships

earl:failed

failed ✖

cantTell ✖

@AccessMonitor ✖

The web pages must be marked with a hierarchical structure of headings and subheadings. Each page should have, at least, one level 1 heading. The level 2 should mark the sections and the level 3 the subsections of these. This same logic must be applied until the subsections of level 6. Organize a page using headings

@MAUVE ✖

Using ARIA landmarks to identify regions of a page

@A11Y - modifier: @Julen ✖

REGION landmark must have accessible name

[A11Y]

- <section class='portlet' ...

(a) Egindako aldaketak erabil-      (b) Editatutako datuak mantentzailearen izenean gordetzean      tzea

## 7.3 Txostenak datu-basean gorde eta bertatik kargatzeko probak

Atal honetan txostenak zerbitzarian gordetzeko eta bertatik eskuratzeko funtzionalitateak probatuko ditugu. Datu-baseko reports taula hutsarekin hasiko gara, beraz agertuko diren txosten bakarrak ondoren igotakoak izango dira.

### 7.3.1 Txostenak gordetzea

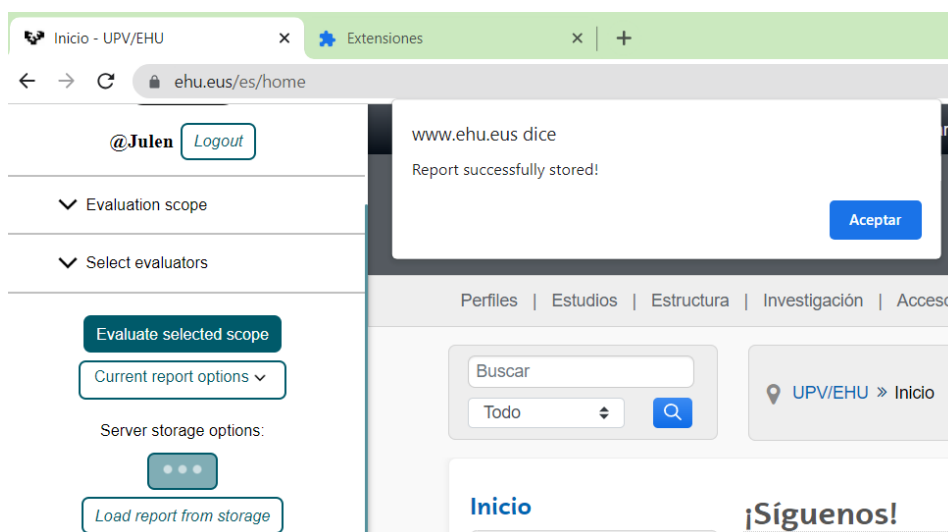
Lehenik, ehu.eus eta ataria.eus guneentzako txosten batzuk igoko ditugu: ehu.eus webgunerako 3 txosten editatu, eta ataria.eus gunerako txosten bakarra.

Igoko dugun lehen txostena <https://www.ehu.eus/es/home> orrialdea a11y eta pa11y liburutegiekin ebaluatuz sortutako izango da, ikus [7.8\(a\)](#) eta [7.7](#). Ondoren, txosten bereko 1.1.1 irizpideko CantTell kasua ezabatu eta 1.1.1 irizpidearen emaitza passed izatera pasata, ikus [7.8\(b\)](#) irudia, txosten aldatua datu-basera igoko dugu. Bigarren txostena, lehenengotik abiatu denez, bere parentId balioa lehenengoaren id balioa izango da.

Igoko dugun 3. txostena orrialde berean AccessMonitor eta Mauve++ ebaluatzaileak erabiliz sortutako izango da, ikus [7.8\(c\)](#) irudia, hau da, igotzean parentId balioa null izango da, txosten berria sortu dugulako beste batetik abiatu beharrean.

Kontuan izan lehen bi txostenak @Julen erabiltzaileak igotakoak direla, baina hirugarren hau @Mikel erabiltzaileak igo duela.

Azkenik, ataria.eus gunerako igoko dugun txostena, <https://ataria.eus/> orrialderan gainean Lighthouse liburutegia erabiliz sortutakoa izango da, ikus [7.9](#) irudia.



7.7 Irudia: Txostena igotzean ikusten den mezua

### 7.3. Txostenak datu-basean gorde eta bertatik kargatzeko probak

Current report results

Select conformance level: **A AA AAA**

Current Site/Page summary: Website *Inicio - UPV/EHU*

P	F	CT	NP	NC
3	3	27	4	41

Current webpage evaluation results:

Standard	P	F	CT	NP	PNC
<b>1 Perceivable</b>	0	2	7	3	17
1.1 Text Alternatives	0	0	1	0	0
^ 1.1.1: Non-text Content	earl:cantTell				
^ cantTell	✖				
^ passed	✖				
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.4 Distinguishable	0	1	4	0	8
<b>2 Operable</b>	1	0	10	1	17
<b>3 Understandable</b>	1	0	10	0	6
<b>4 Robust</b>	1	1	0	0	1

((a)) Igotako lehen txostena

Current report results

Select conformance level: **A AA AAA**

Current Site/Page summary: Website *Inicio - UPV/EHU*

P	F	CT	NP	NC
4	3	26	4	41

Current webpage evaluation results:

Standard	P	F	CT	NP	PNC
<b>1 Perceivable</b>	1	2	6	3	17
1.1 Text Alternatives	1	0	0	0	0
^ 1.1.1: Non-text Content	earl:passed				
^ passed	✖				
1.2 Time-based Media	0	0	0	3	6
1.3 Adaptable	0	1	2	0	3
1.4 Distinguishable	0	1	4	0	8
<b>2 Operable</b>	1	0	10	1	17
<b>3 Understandable</b>	1	0	10	0	6
<b>4 Robust</b>	1	1	0	0	1

((b)) Igotako bigarren txostena

Current report results

Select conformance level: **A AA AAA**

Current Site/Page summary: Website *Inicio - UPV/EHU*

P	F	CT	NP	NC
4	14	9	0	51

Current webpage evaluation results:

Standard	P	F	CT	NP	PNC
<b>1 Perceivable</b>	1	7	4	0	17
1.1 Text Alternatives	0	1	0	0	0
^ 1.1.1: Non-text Content	earl:failed				
^ failed	✖				
^ cantTell	✖				
^ passed	✖				
1.2 Time-based Media	0	0	0	0	9
1.3 Adaptable	1	2	1	0	2
1.4 Distinguishable	0	4	3	0	6
<b>2 Operable</b>	1	4	2	0	22
<b>3 Understandable</b>	2	2	1	0	12
<b>4 Robust</b>	0	1	2	0	0

((c)) Igotako hirugarren txostena

### 7.8 Irudia: ehu.eus webguneko igotako 3 txostenak

Current report results

Select conformance level: **A AA AAA**

Current Site/Page summary: Website *Tolosaldea interneten - Tolosaldeko ataria*

P	F	CT	NP	NC
4	4	0	5	65

Current webpage evaluation results:

Standard	P	F	CT	NP	NC
<b>1 Perceivable</b>	2	2	0	1	24
<b>2 Operable</b>	1	0	0	2	26
<b>3 Understandable</b>	1	0	0	2	14
<b>4 Robust</b>	0	2	0	0	1

7.9 Irudia: ataria.eus webgunerako igotako txostena

## 7. PROBAK

The image shows three browser windows displaying EARLER reports. Each window has a settings gear icon and a close 'x' icon. The first window shows reports for 'www.ehu.eus website' with a table containing two rows. The second window shows reports for 'www.ehu.eus website' with a table containing one row. The third window shows reports for 'ataria.eus website' with a table containing one row. Each window also has a 'Cancel' button and a pagination control.

Id	Version	Uploaded by	Parent Id
1	1	Julen	
2	2	Julen	1

Id	Version	Uploaded by	Parent Id
3	1	Mikel	

Id	Version	Uploaded by	Parent
4	1	Mikel	

((a)) ehu.eus taulako lehen orrialdea ((b)) ehu.eus taulako bigarren orrialdea ((c)) ataria.eus taulako orrialde bakarra

### 7.10 Irdia: Txostenen taulak

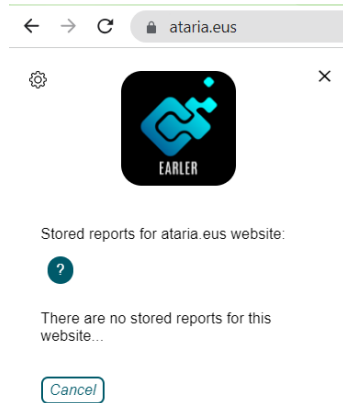
#### 7.3.2 Txostenak kargatzea

Txosten batzuk kargatu ondoren, kokatutako domeinuko igotako txostenak ikusi eta kargatzeko funtzionalitateak eskaintzen dituen luzapeneko zatira joango gara. 7.10(a) eta 7.10(b) irudian ikus daiteke ehu.eus webgunearen kasuan bi orrialde daudela, lehenak, igo ditugun lehen bi txostenak barne ditu, eta taulako bigarren orrialdeak igo dugun 3. txostena du.

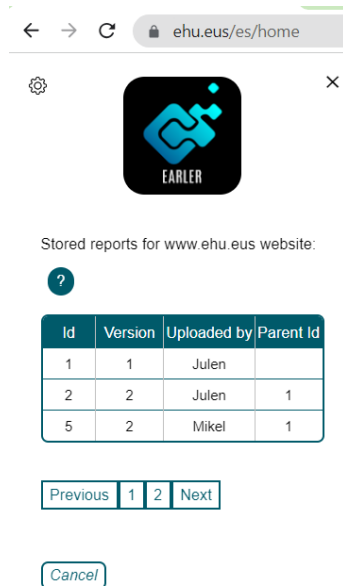
Edozein kasutan ez da ataria.eus gunera igo dugun txostena azaltzen, taulako txostenak uneko webgunekoak direlako, hau da, ataria.eus webgunean berdina egiten badugu, taulak orrialde bakarra izango du, eta bertan igo dugun azken txostena bakarrik aurkituko dugu, ikus 7.10(c). Beraz, ataria.eus guneko txosten bakarra ezabatzen badugu, ez dira txosten gehiago egongo eta 7.11 irudian ikusten dena azalduko zaigu.

Azkenik, ehu.eus gunera igotako lehen txostena aukeratu eta kargatu ondoren, 1.1.1 irizpideko cantTell eta passed bi kasuak kendu eta berriro igoko dugu. Txosten berri hau lehen txostena kargatu ondoren igo denez, bere parentId lehenengoaren id balioa izango da, eta ehu.eus txostenen taulara joaten bagara, guraso bereko bi txosten ditugula ikusiko dugu, ikus 7.12 irudia.

### 7.3. Txostenak datu-basean gorde eta bertatik kargatzeko probak



#### 7.11 Irudia: ataria.eus webgunerako txosten bakarra ezabatzean



#### 7.12 Irudia: azken txostena igo ondoren guraso bera duten bi txosten ditugu



# Jarraipen eta kontrola

Kapitulu honen bitartez, proiektuaren garapenean zehar jasotako informazioaz baliatuz, proiektuko plangintzan zehaztutako eta errealitatean egindako lanaren arteko ezberdintasunak eta desbiderapenak ikusi eta azalduko ditugu.

## 8.1 Atazen dedikazio denboren desbiderapenak

Ondorengo 8.1 taulan hasieran estimatutako eta errealitatean dedikatu diren ordu kopuruen arteko ezberdintasunak ikus daitezke.

Hasteko, luzapenaren implementazioaren lan paketea izan da ordu kopuruen aldetik desbiderapen handiena izan duena. Hori gertatzeko arrazoiak oinarritzko funtzionalitateen implementazioak eta funtzionalitate gehigarrien implementazioak estimatutako denbora baina gehiago eman duela da.

Oinarritzko funtzionalitateen aldetik, desbiderapena izateko arrazoiak honakoak izan dira:

- Aurkitutako emaitzen HTML elementuak orrialdeetan nabarmendu ahal izateko erreferentziak beharrezkoak dira, erreferentzia horiek CSS selector motakoak edo XPath motakoak izan dezakete, eta nahiz eta orokorrean ebaluatzaileek eskaintzen dituzten, AChecker-ek ez du horrelakorik ematen eta ondorioz beste soluzio batzuk bilatzen eta implementatzen denbora bat pasa behar izan dugu.
- Luzapenean aplikatu beharreko estilo kopuru handiaren eraginez, estiloak hobeto antolatzeke Sass teknologia erabiltzera pasa behar izanak ere denbora gehigarria ere behar izan du.
- Horretaz aparte, nahiz eta React erabili, aurreko urteko proiektuko kodea neurri batean berrerabiltzea edukiko nukeela espero nuen, baina errealitatean kasu gutxi batzuetan egitura mantentzeaz aparte dena aldatu behar izan dut. Gainera, nahiz eta hasieran trebakuntza atalean React tresnari buruz ikasteko tutorialak egitea lagundu duen, oraindik ere nahiko galduta nengoen eta denbora bat behar izan dut frameworkaren alderdi ezberdinak ondo barneratzeko. Arrazoi horiek ere desbiderapenean lagundu dute.

Funtzionalitate gehigarrien aldetik, estimatutako denboraren bikoitza pasatzetik gertu ibili naiz, eta horren arrazoa batez ere Blacklist funtzionalitatearena izan da. Funtzionalitate hori nahiko konplexua suertatu da, zerrendan elementuak gehitzeko luzapeneko emaitzen taulatik egiten delako, baina elementuak kentzeko konfigurazio orrialdetik egin beharra dago, eta konfigurazio orrialdean aldaketak egitean luzapenari emaitzak blackList-a kontuan izanik eguneratzeko adierazteak nahiko denbora handia kendu zidan. Gainera, funtzionalitatea inplementatzeko txosteneko emaitzak aldatzea beharrezkoa da eta hori egin ahal izateko funtzioak ere konplexutasun nahiko handia dutela kontuan izanda, horretan ere uste baina denbora gehiago pasa nuen.

Ondoren, aipatu beharreko beste gauza bat zerbitzariko irisgarritasun ebaluaketa automatikoari dagokiona da. Alde honetatik espero nuena baina nahiko denbora gutxiago behar izan dut, batez ere Lighthouse, Pa11y eta Mauve++ ebaluatzaileak inplementazioa erraza izan delako. Lighthouse eta Pa11y ebaluatzaileen kasuan funtzio bati deia egin eta jasotako emaitzak dagoeneko txostenean sartzeko bideratuta daude, eta Mauve++ tresnaren kasuan scraping egiteko dena orrialdeko atal beretik egiten denez besteekin alderatuz simplea suertatu da. Horretaz aparte, JSON-LD motako txostenak sortzeko jsonLd klasea sortzeak, AccessMonitor ebaluatzailearen scraping prozesua zuzendu behar izateak (aldatu egin dutelako), ebaluaketak paraleloan egin eta ondoren emaitzak bateratzeak denbora behar izan dute, baina emaitzak eskuratzearen erraztasunagatik konpentsatuta geratu da.

Aurrekoaz aparte, itxieraren lan paketeko memoriaren idazketak pentsatzen nuena baina denbora gehiago behar izan du, nahiz eta dagoeneko beste ataza batzuekin memorian jarri beharrezkoa aurreratu nuen. Nire ustez hori gertatzeko arrazoa, tamaina honetako dokumentu bat aurretik inoiz idatzi ez dudala da. Agian gehienez jota 40 orrialdeko lan baten bat egingo nuen, baina dokumentu honekin konparatuta oso motz geratzen da eta estimazioa egitean nahiko optimista izan nintzen.

Bukatzeko, denbora totalaren aldetik dagoeneko 300 orduetatik askogatik pasata nagoenez, denbora aldetik nahiko justu nabilenez, eta defentsarako aurkezpena errealitatean memoriarekin batera entregatu behar ez denez, aurkezpenaren ataza lana entregatu ondoren egitea erabaki dugu. Hori dela eta, errealitatean egindako dedikazioa 0 ordu bezala adirazi dut.



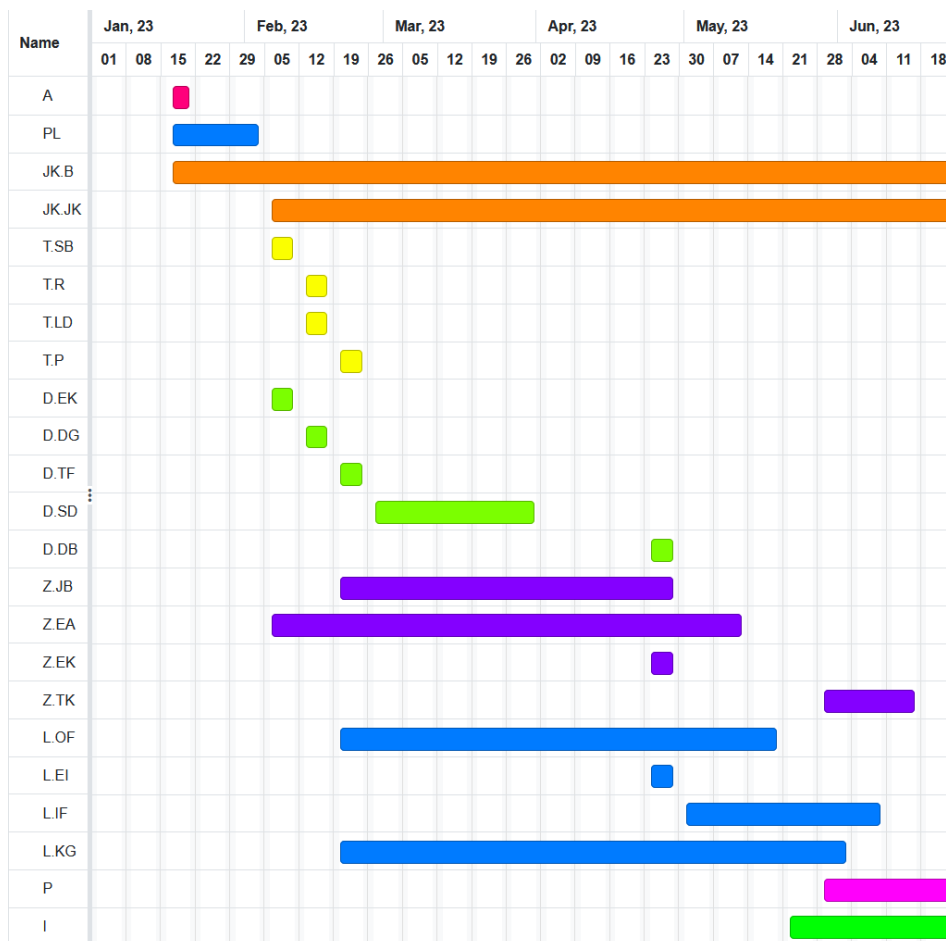
### 8.1. Atazen dedikazio denboren desbiderapenak

ATAZA	ESTIMATUTAKO ORDUAK	ERREALITATEA	DESBIDERAPENA
A - Aurrekariak	13	13	0
ACW	7	7	0
EL	6	6	0
PL - Plangintza	15	16	1
PD	3	3	0
HP	12	13	1
JK - Jarraipen eta kontrola	10	12,5	2,5
B	8	10,5	2,5
JK	2	2	0
T - Trebakuntza	16	15	-1
SB	8	7	-1
R	4	4,5	0,5
P	2	1	-1
LD	2	2,5	0,5
D - Diseinua	20	23,5	3,5
EK	2	1,5	-0,5
DG	2	2,5	0,5
SD	10	13	3
TF	4	5,5	1,5
DB	2	1	-1
Z - Zerbitzaria	80	76	-4
JB	2	3,5	1,5
EA	75	68	-7
EK	1	1	0
TK	2	3,5	1,5
L - Luzapena	75	98,5	23,5
OF	45	61	16
EI	2	1,5	-0,5
IF	20	21	1
KG	8	15	7
PR - Probak	28	24,5	-3,5
EA	4	5	1
AE	4	3,5	-0,5
BK	4	3	-1
TK	16	13	-3
I - Itxiera	34	37	3
KD	6	4	-2
G	3	2	-1
M	20	31	11
A	5	0	-5
<b>Denera:</b>	<b>291</b>	<b>316</b>	<b>25</b>

8.1 Taula: Estimaturako eta errealtatean dedikatutako ataza akointzeko denborak orduetan

## 8.2 Garapen denboraldien desbiderapenak

Denboraldien desbiderapenak ondorengo 8.1 Gantt diagrama eguneratua ikus daitezke (hasierako Gantt diagrama: 3.2).



8.1 Irudia: Amaierako Gantt diagrama

Kontuan izateko gauza batzuk:

- Zerbitzariaren eskaerak jaso eta bideratzeko sistemarentzako bukaera aldera soluzio hobea aurkitu nuen eta horregatik ikusten da horrenbesteko desbiderapena, baina errealitatean orduen aldetik ezberdintasuna txikia izan da.
- Luzapenaren inplementazioa, orokorrean luzatu eta atzeratu egin da aurreko atalean azaldutakoen ondorioz.
- Proben atalak dedikazio aldetik denbora gutxiago behar izan du, baina proiektuaren amaierara arte luzatu behar izan dut. Hori, proiektuaren amaiera aldera lanean hasi naizenez, proiektura dedikatzeke izan dudana denbora murriztagoa izan delako eta gainera memoriak pentsatzen nuena baina denbora gehiago eramaten ari zaidalako gertatu da.

## Ondorioak

Kapitulu honen bitartez, proiektuari amaiera emateko egindako lanarekin lortutako ondorioak zeintzuk diren azalduko da. Gainera, sortutako tresnaren egoera ikusita etorkizunean beste lan baterako egin daitezkeen gehikuntzak edo hobekuntzak ere aipatuko dira.

### 9.1 Ondorioak

Sortutako tresnaren aldetik, ebaluaketa automatikoak egitean noiz behinka sortzen diren errore horiei soluzioa ematea komeniko litzateke, baina hori kenduta sortu dugun luzapena nire ustez egonkorra da, eta erabiltzailearentzako erabilgarria izan daiteke. Erabiltzaileak erabili ahal izateko diseinu aldetik erakargarria dela uste dut, eta eman diogun egiturarekin nahiko intuitiboa eta erabilerraza dela esango nuke. Sortutako kodearen egituraren hobekuntzak egitea egongo ziren, baina orokorrean ondo antolatuta dago eta beste garatzaile batek ulertzeko modukoa da.

Egindako diseinuen aldetik, gure kasuan erabilpen kasuen diagrama eta erabilpen kasuen sekuentzia diagramak erabili ditugu. Normalean diagrama gehiago erabiltzen dira inplementazioarekin hasi aurretik gauzak hobeto zehazteko, baina nire ustez, proiektu honetarako erabilitakoekin nahikoa izan da egin beharreko inplementazioaren ideia bat egiteko, eta horri esker kodea nahiko ondo egituratzeko.

Horretaz aparte, tresnaren proba gehiago eginez ondorio gehiago aterako lirateke, baina egindako proba eta test kasuekin tresnaren funtzionalitate orokorren inplementazioa nola geratu den eta nola lan egiten duten ikusteko nahikoa dela iruditzen zait.

Azkenik, egia da bukaera aldera gauzak zertxobait konplikatu direla, eta azkenean defentsako aurkezpena entregaren ondoren egiteko utzi dela, baina hori kenduta proiektuaren helburuak eta eskakizunak betetzea lortu da. Noski hobekuntzak egitea egongo ziren baina proiektuaren luzapena 300 ordukoa dela gogoratu behar dugu, eta lan hau egitearen alderdietako bat daukagun denborara egokitzeko gaitasuna lantzea dela kontuan izanik, nire ustez lan ona egitea lortu da.

## 9.2 Ikasitakoa eta esperientzia pertsonala

Nire ustez proiektu hau garatzeko erabili diren, metodologiak, teknikak, teknologiak, etab. nire web garatzaile ezagutzak eta trebetasunak osatzen eta maila nahiko ona lortzen lagundu dute, eta aldi berean graduan zehar ikasi ditudan hainbat irakasgai ezberdinetan ikasitako gauzak praktikan jartzea ahalbidetu dit.

## 9.3 Etorkizunerako hobekuntza posibleak

Seguruena hainbat gauza gehiago hobetu edo gehitu daitezke, baina honakoak dira proiektua aurrera joan ahala guk ikusi ditugun hobekuntza posibleak:

- Aurretik aipatu genuen bezala, luzapena kokatuta dagoen orrialdearen arabera, orrialderen estiloak luzapenari eragiten dio eta erabiltzaileak luzapena ikusten duen modua aldarazten du. Luzapeneko elementuen estiloa ahalik eta espezifikoen adieraziz, orokorrean egonkorra izatea lortu dut, baina ez da ondo geratzen tresna batek orrialdetik orrialdera gutxi bada ere itxura aldatzea.

Arazori horri soluzioa emateko, luzapenaren estiloak isolatzeko modu baten bat egon daitekeela ikusi genuen, eta nahiz eta guk alde batera uztea erabaki genuen, etorkizunerako hobekuntza bezala ondo egon daiteke.

- Login eta register formularioetan ez dugu datuen balidaziorik ezta erabiltzailearen datuak babesteko mekanismorik erabili. Ondo egongo litzateke balidazioa luzapenean bertan eta zerbitzarian egitea, eta gutxienez erabiltzaileen pasahitzak zifratuz babestea.
- Datu-basean gordetako txostenak ikuskatzeko txostenak taula batean zerrendatu ditugu, baina hasieran zuhaitz edo grafo moduko baten nodoen bitartez txostenak errepresentatzea interesgarria izango zitekeela pentsatu genuen. Modu horretara, txostenen parent eta child erlazioak eta sortzen diren adarrak grafikoki oso ondo bereiziko lirateke. Saiatu nintzen hori egin ahal izateko moduren bat bilatzen, baina konplexua izango litzatekeela ikusita azkenean taularekin egitea erabaki genuen. Beraz, agian hobekuntza bezala aproposa izango litzateke zuhaitz egiturako grafiko moduko bat implementatzea datu-basean gordetako txostenak eta beraien arteko erlazioak errepresentatzeko.
- Kontuan izateko beste gauza bat, WCAG EM Report Tool tresnak igotako txostenak esportatzean txostenen egitura aldatzen duela da. Hasieran txostenen egitura mantentzen zela suposatzen genuen, eta ondorioz gure luzapeneko txostenak bertan editatu eta luzapenera berriro kargatzea zeudela pentsatzen genuen, baina proiektuaren bukaera aldera ezin daitekeela egiaztatu dugu. Hori ikusita, tresnaren kodea irekia denez eta dagokion GitHub errepositorioan eskuragarri dagoenez, tresna aldatzea bururatu zitzaigun txostenen formatua mantentzeko, eta bide batez luzapenean implementatutako txostenak datu-basera igo, bertatik jaitsi eta erabiltzaileak erregistratu eta logeatzeko funtzionalitateak ere implementatzeko.

Denbora faltaren ondorioz ez dugu egin, baina tresnaren kodeari begirada bat eman, eta adibidez login eta erregistratzeko aukerak nola implementatu daitezkeen ikusi

dut. Egia da kodea React beharrean Svelte framewrok-a erabiliz garatuta dagoela, baina nire ustez aipatutakoak inplementatzea ez litzateke zaila izango. Horregatik, hobekuntza bezala aproposa dela iruditzen zait.

- Gainera, luzapena lokalizatu edo internazionalizatzea egongo liteke, ingelesez ez dakiten erabiltzaileek tresna erabiltzeko gai izateko. Dena dela, erabiltzaileak garatzaileak direla, eta orokorrean garatzaile guztiek ingelesez dakitela kontuan izaten badugu, ez da garrantzitsua izango litzatekeen hobekuntza bat.



# **Eranskinak**





# Erabilitako tresnak

Atal honetan proiektuan zehar erabilitako tresna eta baliabide guztiak zerrendatu eta laburki deskribatuko dira.

## **AccessMonitor**

[8] [Acessibilidade.gov.pt](https://www.accessibilidade.gov.pt) ekosistemako web irisgarritasunaren azterketarako tresna da, WCAG 2.1 estandarrean oinarritua.

## **AChecker**

[9] Web irisgarritasunaren azterketarako tresna da, WCAG 2.0 estandarrean oinarritua.

## **A11y**

[36] AInspector tresnak azpitik erabiltzen duen web irisgarritasunaren azterketarako liburutegia da, ARIA eta WCAG estandarretan oinarritua.

## **Body-parser**

Express-eko middleware bat da, proiektu honetako ebaluazio txostenen tamaina handia dela eta, bezero-zerbitzari komunikazioetako tamaina limitea handitzeko erabili dena.

## **ChatGPT**

OpenAI-k garatutako inteligentzia artifiziala, batez ere kodearen dokumentazioa egiteko erabili dena.

## **Chrome arakatzailleko luzapenentzako baliabideak**

Google Chrome nabigatzailean garatzaileentzat eskuragarri dauden baliabideak eta tresnak dira. Gure kasuan batez ere luzapenentzako irudiak lortzeko eta datuak gordetzeko baliabideak erabili dira.

## Cors

Express-en middleware bat da, web aplikazioetan dagokion baliabideen trukaketa politikak konfiguratu eta kudeatzeko balio duena. Gure kasuan CORS arazoak ekiditeko zerbitzarira edozein URL-tik datozen eskaerak onartzen ditugu.

## Diagrams

[37] Diagrama mota ezberdinak sortzeko eta editatzeko online tresna bat da.

## Express

Node.js-en web framework bat da, web aplikazioak eta API-ak erraz sortzeko baliatzen dena. Gure kasuan zerbitzaria sortu eta honek eskaerak jaso eta kudeatu ditzan erabili dugu.

## GitHub

Software proiektuak jartzeko, kudeatzeko eta partekatzeko plataforma bat da, Git bertsio kontrol sisteman oinarrituta.

## GlooMaps

[38] Grafikoak egiteko doako online zerbitzu bat da, gure kasuan LDE diagramak egiteko erabilia.

## JSDoc

[39] JavaScriptentzako dokumentazio tresna bat da, kodean egindako iruzkinetan oinarrituz erabiltzaileak kodeko modulu bakoitza ulertzeko dokumentu bat sortzeko balio duena.

## JavaScript

Web garapenean askotan erabiltzen den programazio hizkuntza bat da, orrialdeei interaktivitatea eta dinamismoa ematea ahalbidetzen duena.

## Lighthouse

[14] Besteak beste, web-irisgarritasunaren azterketarako liburutegia da, estandarretan oinarrituz webgune baten irisgarritasun arazoak identifikatzeko eta hobetzeko aholkuak eskaintzen dituena. Google Chrome arakatzailak barnean inplementatuta dakar.

## Mauve++

[10] Web-irisgarritasunaren azterketarako webgunea da, WCAG 2.1 estandarrean oinarrituz webgune baten irisgarritasun arazoak identifikatzeko eta hobetzeko aholkuak eskaintzen dituena.

## Node.js

Node.js-en exekuzio-ingurune bat da, web aplikazio eskalagarri eta errendimendu altukoak garatzeko aukera ematen duena. Gure kasuan erabili ditugun tresna guztiak ingurune horretakoak izan dira.

## Nodemon

Zerbitzariko kodean aldaketak egitean zerbitzaria automatikoki berrabiatzen duen tresna da.

## Online Gantt

[40] Gantt-diagramak sortzeko eta kudeatzeko online tresna bat da.

## Overleaf

LaTeX dokumentuak sortzeko eta editatzeko aukera ematen duen online plataforma bat da, honako memoria idazteko erabili dena.

## Pa11y

[13] Irisgarritasunaren azterketarako liburutegi bat da, azpitik Puppeteer erabiltzen duena.

## Puppeteer

[34] Node.js-en liburutegi bat da, Google Chrome arakatzailerako instantziak sortuz orrialdeetara nabigatu eta programazio bitartez erabiltzaileak eskuz egingo lukeena automatikoki egiteko baliabideak eskaintze ditu. Gure kasuan scraping egiteko erabili dugu.

## React

[28] JavaScript erabiltzen den liburutegi bat da, erabiltzaileen interfaze grafiko eta frontend bezala ezagutzen dena garatzeko erabili duguna.

## React-spinners

React-en liburutegi bat da, erabiltzaileak itxaroten duen bitartean animazioak egiteko erabili dena.

## **Sass**

[41] Oinarrizko CSS formatuarekiko funtzionalitate gehigarriak implementatzen ditu.

## **Sqlite3**

[42] SQL datu-base erlazionalak kudeatzeko sistema, kode irekikoa, doakoa eta arina.

## **XPath**

[43] XML edo HTML dokumentuetan elementuak eskuratzeko path delakoak erabiltzen dituen kontsulta hizkuntza bat da.

# Instalazio gida

Instalazio gida honen bitartez garatutako tresna deskargatu eta martxan jartzeko jarraitu beharreko pausuak azalduko dira. Kontuan izan instalazioa Windows 11 sistema eragilerako egin dela, baina gainontzekoetako ere antzekoa izan beharko luke.




## Node.js instalatzen

Lehenik, Node.js inguruneko instalatzailea jaisten hasiko gara. Instalatzailea jaisteko, <https://nodejs.org/es/download> orrialdeko deskargen atalean azaltzen diren aukeretatik dagokizuna aukeratu, ikus 1 irudia.

### Descargas

Versión actual: 18.16.0 (includes npm 9.5.1)

Descargue el código fuente de Node.js o un instalador pre-compilado para su plataforma, y comience a desarrollar hoy.

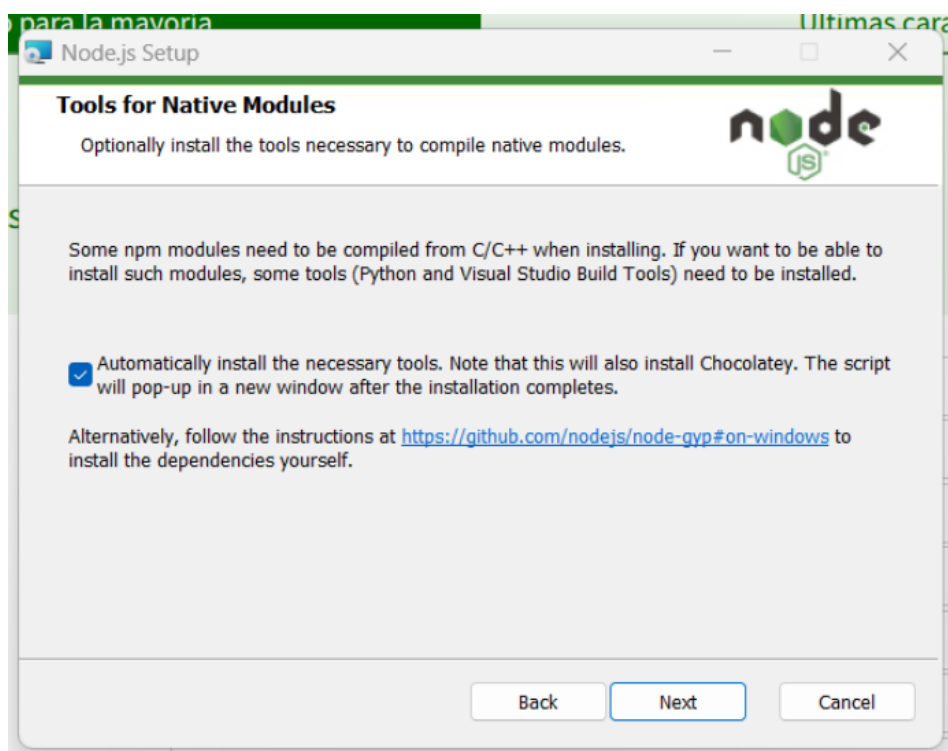
LTS Recomendado para la mayoría	Actual Últimas características	
 Instalador Windows node-v18.16.0-x64.msi	 Instalador macOS node-v18.16.0.pkg	 Código Fuente node-v18.16.0.tar.gz

Instalador Windows (.msi)  
Binario Windows (.zip)  
Instalador macOS (.pkg)  
Binario macOS (.tar.gz)  
Binario Linux (x64)  
Binario Linux (ARM)  
Código Fuente

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v18.16.0.tar.gz	

**1 Irudia:** Node.js ingurunearen webguneko deskargen orrialdea

Instalatzailea abiaratu eta pausuak segi. Adi! dependentziak automatikoki instalatzeko 2 irudian ikusten den aukera sakatu.



**2 Irudia:** Node.js instalatzeko dependentzien aukera

Instalazaile nagusiak bukatu ondoren, terminal bat automatikoki irekiko da dependenziak automatikoki deskargatu eta instalatzeko. Prozesu horretan batzuetan blokeatuta geratu daiteke baina denbora bat pasa ondoren aurrera egiten du.

## Earler deskargatu eta martxan jartzen

Kodea deskargatzeko, proiektuko GitHub repositoriora joan eta bertatik deskargatu<sup>1</sup>. Deskargatutakoa deskonprimatu eta 2 karpeta nagusi aurkituko dituzu:

- earler-extension: luzapenari dagokion kodea, bi azpikarpetaz osatua:
  - react-chrome-app: konpilatu aurretiko luzapenaren kodea.
  - extension: react-chrome-app karpetako kode konpilatua barne duen luzapenaren kodea.
- earler-server: luzapenaren funtzionamendurako zerbitzaria.

Zerbitzaria martxan jartzeko, terminal baten bitartez earler-server karpeta barruan kokatu, eta npm run start komandoa exekutatu. Seguruena lehen aldian "nodemon"ez duela ezagutzen adieraziko du, eta konpontzeko npm install nodemon komandoa erabiliz instalatu beharko duzu. Zerbitzaria martxan jartzean, 3 irudian azaltzen dena edo antzekoa ikusi beharko zenuke.

<sup>1</sup><https://github.com/larraagirrejulien/GrAL-Earler>

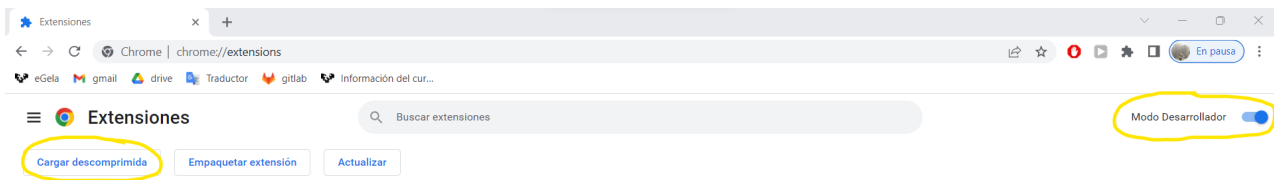
```
PS C:\Users\Julen\Desktop\GrAL-Earler-main\earler-server> npm run start

> puppeteer-scrafer@1.0.0 start
> nodemon index.js

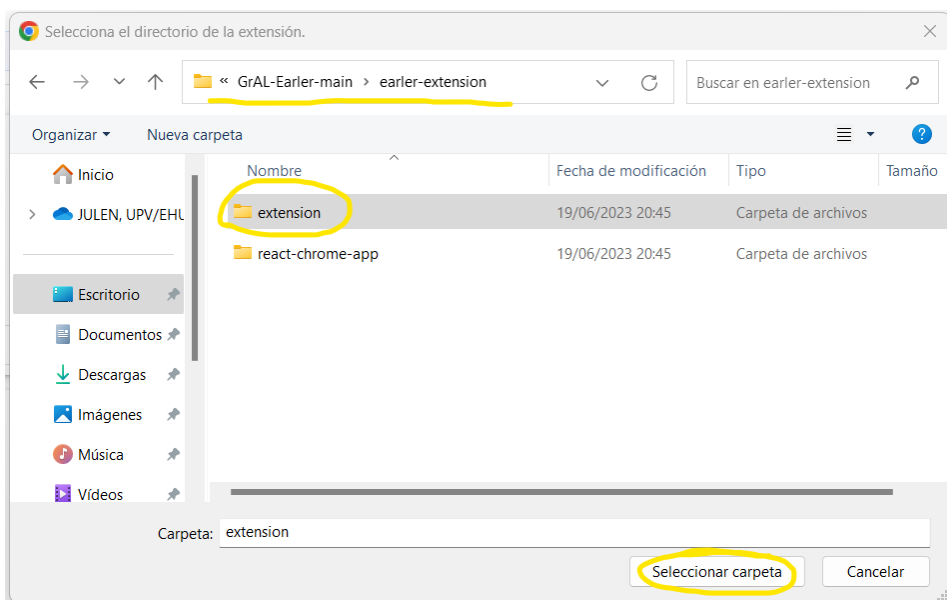
[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
listening on: 0.0.0.0:7070
```

**3 Irudia:** Zerbitzaria martxan jartzean ikusten dena

Zerbitzaria martxan jarrita, Google Chrome arakatzailan luzapenak kudeatzeko orrial-dera joan <sup>2</sup> eta luzapena erabiltzeko 4 eta 5 irudietan horiz adierazitako pausuak segi.



**4 Irudia:** Luzapena inportatzeko lehen pausua



**5 Irudia:** Luzapena inportatzeko bigarren pasua

Aurrekoa amaitzean luzapenak martxan egon beharko luke. Luzapenaren react-chrome-app karpetakoa aldatzen bada, egindakoa konpilatzeko terminal baten bitartez karpetan kokatu eta npm run build komandoa exekutatu. Lehen aldia webpack ezagutzen

<sup>2</sup>[chrome://extensions/](https://chrome://extensions/)

ez duela esaten badu, npm install webpack komandoa erabili beharko duzu. Egindako aldaketen arabera, baliteke orrialdeaz aparte luzapena ere birkargatu behar izatea.



# Bibliografía

- [1] Julen Larrañaga. Ac-check: A chrome extension for web accessibility analysis, aggregation and report generation. <https://github.com/larraagirrejulen/GrAL>, 2023. Ikusi 1 orrialdea.
- [2] Mikel Iturria. Ac-check: A chrome extension for web accessibility analysis, aggregation and report generation. <https://github.com/Itusil/TFG>, 2022. Ikusi 1 orrialdea.
- [3] W3C. Resumen de los estándares de accesibilidad de w3c. <https://www.w3.org/WAI/standards-guidelines/es>. Ikusi 1 orrialdea.
- [4] DigitalA11Y. 43 browser extensions to perform accessibility testing effectively. <https://www.digitala11y.com/accessibility-plug-ins-ie-chrome-firefox-browsers/>. Ikusi 2 orrialdea.
- [5] W3C. Web accessibility evaluation tools list. <https://www.w3.org/WAI/ER/tools/>. Ikusi 2 orrialdea.
- [6] Ainspector. <https://ainspector.disability.illinois.edu/>. Ikusi 3 orrialdea.
- [7] Wave evaluation tool. <https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpxkjmeebjpifedlgcdilcofh>. Ikusi 4 orrialdea.
- [8] Accesmonitor. <https://accessmonitor.acessibilidade.gov.pt/>. Ikusi 6, 101 orrialdeak.
- [9] Achecker. <https://achecker.achecks.ca/checker/index.php>. Ikusi 6, 101 orrialdeak.
- [10] Mauve++. <https://mauve.isti.cnr.it/>. Ikusi 6, 103 orrialdeak.
- [11] Tota11y. [https://khan.github.io/tota11y/?utm\\_source=saashub&utm\\_medium=marketplace&utm\\_campaign=saashub](https://khan.github.io/tota11y/?utm_source=saashub&utm_medium=marketplace&utm_campaign=saashub). Ikusi 10 orrialdea.
- [12] Sa11y. <https://sa11y.netlify.app/>. Ikusi 10 orrialdea.
- [13] Pa11y. <https://pa11y.org/>. Ikusi 10, 103 orrialdeak.
- [14] Lighthouse. <https://developer.chrome.com/docs/lighthouse/overview/>. Ikusi 10, 102 orrialdeak.
- [15] Lighthouse - npm. <https://www.npmjs.com/package/lighthouse>. Ikusi 10 orrialdea.
- [16] W3C. Wcag-em report tool. <https://www.w3.org/WAI/eval/report-tool/>. Ikusi 11 orrialdea.
- [17] Flask. <https://flask.palletsprojects.com/en/2.2.x/>. Ikusi 25 orrialdea.
- [18] Node.js. <https://nodejs.org/es/about>. Ikusi 25 orrialdea.
- [19] Manifest - web accessible resources. [https://developer.chrome.com/docs/extensions/mv3/manifest/web\\_accessible\\_resources/](https://developer.chrome.com/docs/extensions/mv3/manifest/web_accessible_resources/). Ikusi 25 orrialdea.
- [20] Flask-cors. <https://flask-cors.readthedocs.io/en/latest/>. Ikusi 26 orrialdea.
- [21] Cors: Intercambio de recursos de origen cruzado. <https://developer.mozilla.org/es/docs/Web/HTTP/CORS>. Ikusi 26 orrialdea.
- [22] Geeksforgeeks. Creating a proxy webserver in python | set 1. <https://www.geeksforgeeks.org/creating-a-proxy-webserver-in-python-set-1/>. Ikusi 26 orrialdea.

## BIBLIOGRAFIA

---

- [23] Geeksforgeeks. Creating a proxy webserver in python | set 2. <https://www.geeksforgeeks.org/creating-a-proxy-webserver-in-python-set-2/>. Ikusi 26 orrialdea.
- [24] Rob-W. Cors-anywhere. <https://github.com/Rob--W/cors-anywhere/>. Ikusi 26 orrialdea.
- [25] sideshowbarker. Cors-anywhere aipamena. <https://stackoverflow.com/questions/43871637/no-access-control-allow-origin-header-is-present-on-the-requested-resource-when>. Ikusi 26 orrialdea.
- [26] Node.js. Http liburutegia. <https://nodejs.org/es/docs/guides/anatomy-of-an-http-transaction/>. Ikusi 26 orrialdea.
- [27] Dmitri Pavlutin. Fetch guide. <https://dmitripavlutin.com/javascript-fetch-async-await/>. Ikusi 26 orrialdea.
- [28] React. <https://es.react.dev/>. Ikusi 26, 103 orrialdeak.
- [29] React tutorial: Tic-tac-toe. <https://beta.reactjs.org/learn/tutorial-tic-tac-toe>. Ikusi 26 orrialdea.
- [30] Marius Bongarts. How to build a chrome extension using react. <https://web-highlights.com/blog/how-to-build-a-chrome-extension-using-react/>. Ikusi 26 orrialdea.
- [31] Json-ld documentation. <https://json-ld.org/learn.html>. Ikusi 26 orrialdea.
- [32] Selenium. <https://www.selenium.dev/>. Ikusi 26 orrialdea.
- [33] Beautiful soup. <https://beautiful-soup-4.readthedocs.io/en/latest/#>. Ikusi 26 orrialdea.
- [34] Puppeteer scraper. <https://pptr.dev/>. Ikusi 26, 103 orrialdeak.
- [35] Puppeteer guide. <https://www.digitalocean.com/community/tutorials/how-to-scrape-a-website-using-node-js-and-puppeteer>. Ikusi 26 orrialdea.
- [36] A11y accessibility evaluation library. <https://github.com/ainspector/a11y-evaluation-library>. Ikusi 101 orrialdea.
- [37] Diagrams.net. <https://www.drawio.com/>. Ikusi 102 orrialdea.
- [38] Gloomaps. <https://www.gloomaps.com/>. Ikusi 102 orrialdea.
- [39] Jsdoc. <https://jsdoc.app/>. Ikusi 102 orrialdea.
- [40] Online gantt. <https://www.onlinegantt.com/>. Ikusi 103 orrialdea.
- [41] Sass. <https://sass-lang.com/>. Ikusi 104 orrialdea.
- [42] Sqlite. <https://www.sqlite.org/index.html>. Ikusi 104 orrialdea.
- [43] Xpath. <https://developer.mozilla.org/es/docs/Web/XPath>. Ikusi 104 orrialdea.