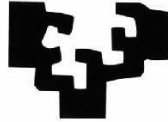


eman ta zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea

Informatika fakultatea

BERTSOTARAKO ARBEL DIGITALA

Manex Agirrezabal Zabaleta

Zuzendariak:

Iñaki Alegria Loinaz
Bertol Arrieta Kortajarema

Eragilea:

Aitzol Astigarraga Pagoaga

ESKERRAK

Lerro hauen bitartez, eskerrak eman nahi nizkieke proiektu honetan modu zuzenean edo ahal izan duten bezala lagundu didaten pertsona guztiei (gogoratzen ditudanak).

Lehenik eta behin, familia eta lagunak eskertu nahiko nituzke, haien babesik gabe ez bainuke ezer egingo.

Ondoren, Oier Lakuntza eta Ekaitz Astiz eskertu nahiko nituzke, haiek izan baitziren bertsolaritzara jauzi egiten animatu ninduten bi pertsonak eta bereziki Oierri esker baitakit orain bertsolaritzaz dakidana.

Ez nuke albo batera utzi nahi Josu Fernandez, zeinarekin ikasi bainuen AJAX teknika erabiltzen eta harekin lortu baikeuen silaba kontatzailea web aplikazioari txertatzea.

Aipatzekoa da ere Aitzol Astigarragak hartutako lan eta konpromezua, edozein gauza behar izan dudanean lagundu baitit denarekin.

Jarraitzeko, Xenpelar Dokumentazio zentroa eskertu nahiko nuke eta bereziki bertako arduraduna, Nere Erkiaga, nire eskari guztiak ahalik eta azkarren erantzun baititu (eta oraindik erantzuten jarraitzen du, eta nik eskatzen jarraitzen dut).

Bertsolari-ableslaria garatzeko, azpimarratu nahiko nuke *Aholab* ikerkuntza taldeak izan duen inplikazioa, eta bereziki Daniel Erro eta Iñaki Saiz aipatu nahi ditut, haiek jasan behar izan baininduten Bilbon.

Errima-bilatzailea garatzeko, proiektu honetako zuzendari den Iñaki Alegria eta Mans Hulden eskertu gabe ezin ditut utzi, foma tresnaren nondik norakoak ulertzeko giltzarri izan baitira.

Bukatze aldera, IXA talde osoari eskerrak eman nahi nizkieke, behar izan ditudan tresna guztiak eskaini dizkidalako, behar izan dudanean momentuan.

Benetan bukatzeko Ibaetako *unibertsoskola* eta *Zumarraga-Urretxuko* bertso-eskolak eskertu nahiko nituzke, probalari bikainak izan baitira.

Manex Agirrezabal Zabaleta

Gaien Aurkibidea

1	Motibazioa	5
2	Proiektuaren helburuak	6
2.1	Helburuak	6
2.2	Azpiatazen zerrenda	7
2.3	Lanaren deskonposaketa egitura	8
2.4	Denbora-estimazioa	9
2.5	Lan egiteko metodologia	10
2.6	Proiektuaren arriskuak	11
2.6.1	Arriskuen zerrenda	11
2.6.2	Arriskuen soluzioak	11
3	Sistemaren analisia eta diseinua	12
3.1	Analisia	12
3.2	Diseinua	16
3.2.1	Klase-diagrama	16
3.2.2	Datu-basearen diseinua	18
3.2.3	Interfazearen diseinua	18
3.2.4	Erabilpen-kasu diagrama	21
3.3	Segurtasunaren bermatzea	21
3.3.1	SQL injection	22
4	Erabilitako teknologia eta azpiegitura	28
4.1	Programazioa eta datu-baseen kudeaketa	28
4.1.1	Lengoaia nagusiak: PHP eta Javascript	28
4.1.2	AJAX teknologia	29
4.1.3	XML	29
4.1.4	MYSQL eta phpMyAdmin	29
4.1.5	Flash	29
4.2	Hizkuntza-teknologia	29
4.2.1	FOMA	30

4.2.2	Xuxen	30
4.2.3	Euskal WordNet	30
4.2.4	Festival	31
4.2.5	Aholab	31
4.3	Baliabide linguistikoak	32
4.3.1	Bertsolaritzaren datu-basea	32
4.3.2	Egunkariako corpusa	32
5	Garapena	33
5.1	Silaba-kontatzailea	33
5.2	Errimak	35
5.2.1	Errima-bilatzailea	35
5.2.2	Errima-egiaztatzailea	37
5.3	Bertso-zuzentzailea	38
5.4	Sinonimo-bilatzailea	38
5.5	Neurria zehazteko zerbitzua	39
5.6	Atalen arteko komunikazioa	40
5.7	Bertso-abeteslaria	41
5.7.1	Festival	41
5.7.2	Aholab	42
5.8	Integrazioa interfazean	45
5.9	Probak	53
6	Ondorioak eta etorkizuneko lana	54
6.1	Ondorioak	54
6.2	Etorkizuneko lana	59
6.2.1	Epe laburrean egitekoak	59
6.2.2	Epe ertainean egitekoak	59
6.2.3	Epe luzean egitekoak	60
A	Bilera-aktak	63
B	Atal ezberdinen sekuentzia-diagramak	74
B.1	Bertsoa gorde	74
B.2	Doinuak lortu	75
B.3	Neurria bidali	75
B.4	Neurriak lortu	76

Irudien Zerrenda

2.1	LDE diagrama	8
2.2	Estimatutako orduen banaketaren grafikoa	10
3.1	Klase-diagrama	24
3.2	Datu-basearen diseinua. Entitate-erlazio diagrama	25
3.3	Arbel digitalaren hasierako diseinua	26
3.4	Arbel digitalaren diseinu berria	26
3.5	Erabilpen-kasu diagrama	27
5.1	Errima-bilatzailearen algoritmoa	36
5.2	BertsoaZuzendu() funtzioaren sekuentzia-diagrama	38
5.3	Arbela zortziko txiki bat jasotzeko prest	49
5.4	Silaba-kontatzailearen marka posibleak: motz, ondo eta luze	50
5.5	Bertsoaren zuzenketa	51
5.6	Errima-bilatzailearen erantzuna	52
6.1	Sortu den arbelaren erabilpen-kasu diagrama	57
B.1	Bertsoa gorde	74
B.2	Doinuak lortu	75
B.3	Neurria bidali	75
B.4	Neurriak lortu	76

Kapitulua 1

Motibazioa

Gipuzkoako bertsolari-txapelketa, *gpuntua* delakoa, jokutzen den urte honetan iritsi da Euskal Herrira bertsotarako arbel digitala.

Proiektu honen helburu nagusia robot bertsolaria, bertsobOT delakoa, sortzea da, baina horra iristeko lan mardula dago aurretik. Beraz, oraingoz bertsotan trebatzeko laguntzaile bat garatu da, bertsotarako arbel digitala, hain zuzen ere.

Askok izan dugu bertso bat idazteko gogo askotan, baina errima eta abilezia falta medio, albo batera utzi izan dugu. Zenbat alditan eskatu izan diogu bertsolari ezagun bati bertso bat idazteko norbaiten urtebetetze egunerako? Maiz utzi dugu bertso bat idatzi gabe osatu ahal izan ez dugulako. Honi irtenbidea emango dio bertsotarako arbel digitalak, 24 orduz eskuragarri izango dugun laguntzaileak.

Hemen aurkezten den lana bertsotan ikasi nahi duen edonorentzat laguntza bat izateko sortu da. Hasiera-hasieratik aipatu beharra dago, honek ez duela inondik inora ere bertso-eskolen lana ordezkatzeko intentziorik. Baina aukera bat izango da handik kanpo, bertsoak idazten laguntzeko, hau da, bertso-eskolen osagarri gisa ulertu behar da.

Amaitzeko, egindako lanak bere fruituak eman dituela esan beharra dago, eta ordenagailua bertso ezberdinak kantatzeko gai dela. Lan honetan azaltzen da nola lortu den hori eta zein lan dagoen etorkizunean egiteko.

Kapitulua 2

Proiektuaren helburuak

2.1 Helburuak

Atal honetan proiektuaren helburuak eta garatu beharreko lan-paketeak finkatuko dira. Lan-pakete bakoitzari eskaini beharreko denbora ere zehaztuko da. Bukatzeko, proiektuaren kostuaren eta arriskuaren analisia egingo da eta proiektuaren bizi-zikloa definituko da.

Hemen aurkezten den proiektuaren helburua, hizkuntza-teknologiak bertsolaritzaren eremuan aplikatzea da. Informatika eta hizkuntza biltzen dituen diziplinari deitzen diogu hizkuntza-teknologia, non informatika erabiltzen den hizkuntza aztertu eta prozesatzeko.

Gaur egun, bertsolaritza mundu digitalean nahiko sartua dagoela esan daiteke. Interneten bertso pila dago, eta baita hainbat webgune interesgarri ere: Bertsolaritzaren datu-basea¹ (Xenpelar dokumentazio zentroa), bertsolarIXA² (Arrieta eta Odriozola)[8] eta bertso-plaza digitala³ (bertso idatziak, audioa eta bideoa). Bertsolaritza jorratzeko blogak ere badira, Bertsokatea⁴ dugu horren lekuko.

Informazio hori, denborarekin, geroz eta handiagoa denez, aplikazio-teknologiko berriak behar-beharrezkoak dira. Baina berrikuntza nolakoa izango den jakiteko, orain arte zein aplikazio garatu diren jakitea komenigarria litzateke. Hauek dira topatu eta aztertu ditugunak:

¹<http://bdb.bertsozale.com>

²<http://ixa2.si.ehu.es/bertsolarixa>

³<http://www.bertsoa.com>

⁴<http://bertsokatea.wordpress.com>

- BertsolarIXA: Interneten atzigarri dagoen errima-bilatzaile automatikoa[8], Bertol Arrietak garatua eta Oihan Odriozolak hobetua.
- Silaba-kontatzailea: Aitzol Astigarragak perl lengoaiaz garatutako silaba-kontatzailea.
- Xenpelar dokumentazio zentroko datu-basea: Bertan bertso-estrofen egiturak, doinu eta bertso ugari kontsulta daitezke, bdb.bertsozale.com da haren helbidea. Datu-basea berriki hobetu eta aurkeztu dute.

Proiektu honetan oraintxe aipatu diren teknologia hauek erreminta bakarrean bateratzea planteatzen da, ahal den heinean haiek hobetuz edota berriak sortuz. Azken helburua bertsoak automatikoki asmatuko dituen robota sortzea litzateke, baina *Deep Blue* bera egun batean sortu ez zen bezala, honek ere pauso batzuk izango ditu aurretik. Garatuko den sistemak bertsolari edo bertsozaleari, bertsoak sortzeko laguntza automatiko bat emango dio. Gehienbat, bertso-eskoletara zuzenduta dago lan hau, baina edozein bertsozalerentzat baliagarri izan nahi du.

2.2 Azpiatazen zerrenda

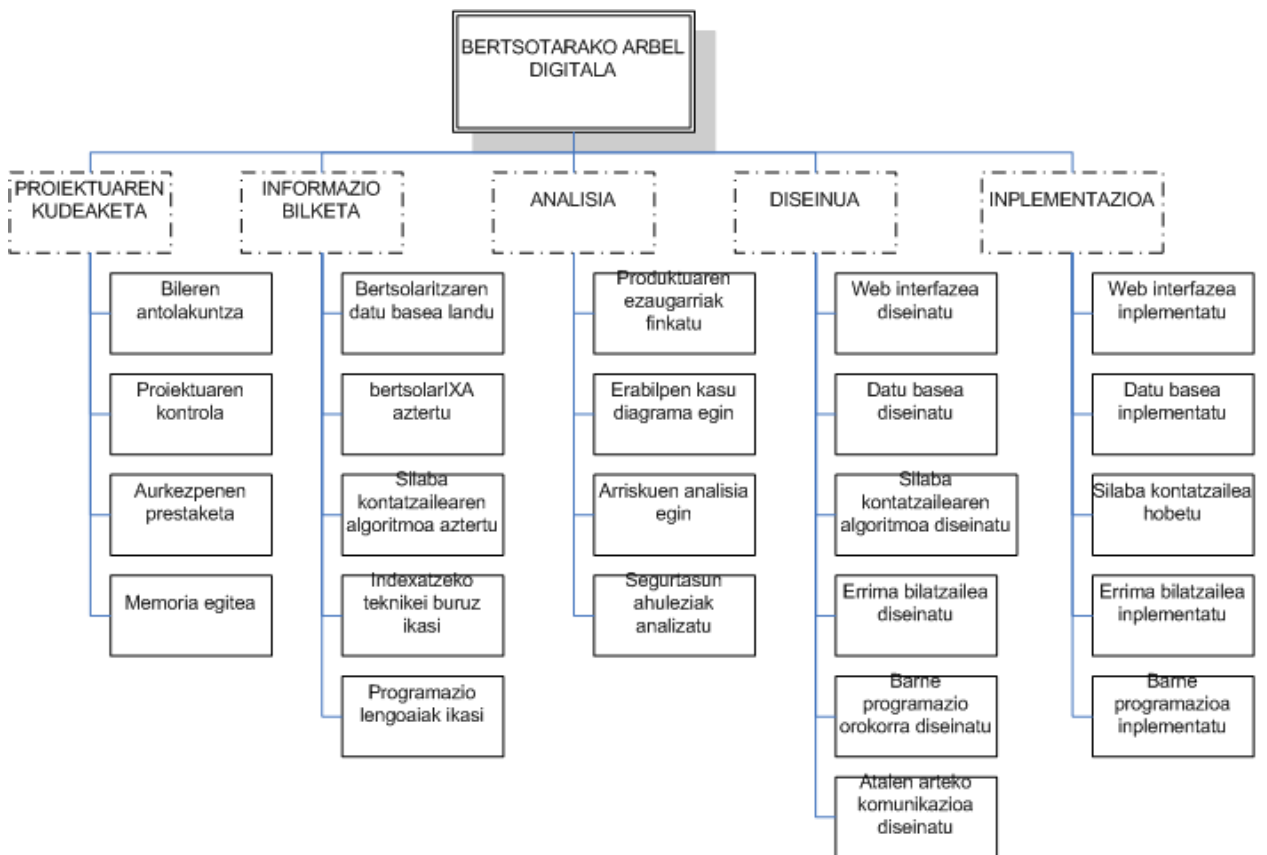
Proiektu honek bost ataza nagusi izango ditu. Ataza hauek proiektuaren bizi-zikloa irudikatzen dute. Eta horiek beste azpiataza batzuetan zabaltzen dira:

- Proiektuaren kudeaketa: Bileren antolakuntza, aurkezpenak, proiektuaren kontrola eta memoria egitea leudeke honen barruan.
- Informazio bilketa: Ataza honetan, proiektuarekin erlazioa duen informazio bilaketa oro sartuko da, baina bilketa hauek multzokatuta jarritz gero, honako azpiatazak aurki ditzakegu: bertsozale elkarteko doinuen eta bertsoen datu-basea lantzea, bertsolarIXA analizatzea, silaba-kontatzailearen algoritmoa aztertzea eta indexatzeko teknikak eta programazio-lengoaia batzuk ikastea.
- Analisia: Kasu honetan sistemaren analisiaren baitan egindako lanak izango dira kontuan, honako hauek izanik azpiatazak: proiektuaren ezaugarriak finkatzea, erabilpen-kasuen diagrama eta arriskuen analisia egitea eta segurtasun ahuleziak analizatzea.
- Diseinua: Hemen web-interfazearen eta datu-basearen diseinua egingo da. Honetaz aparte, silaba-kontatzailearen algoritmoa ere eraikiko da, dagoen scriptetik abiatuta, errima-bilatzailearen diseinua eraiki eta beste dena web-programazioaren diseinuaren barruan sartuko da. Ezin da ahaztu ezta ere, AJAX teknika erabiltzean, atal ezberdinen arteko komunikazioa ere diseinatu egin beharko dela.

- Implementazioa: Implementazioaren azpiatalak diseinukoaren berdina dira, hau da, bertan diseinatutako elementu guztiak implementatuko dira.

2.3 Lanaren deskonposaketa egitura

Hemen behean ageri den irudi honetan, aurreko atalean azaldutako ataza eta azpiatazak grafikoki irudikatzen dira, bost atal nagusitan banatuta.



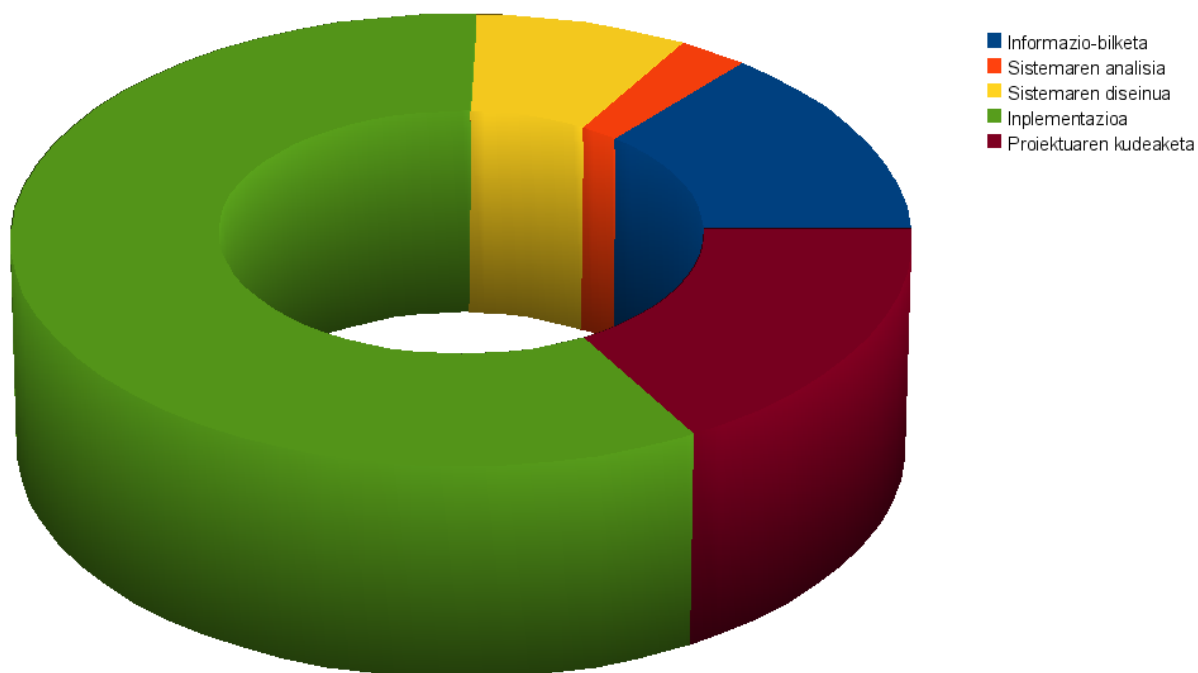
Irudia 2.1: LDE diagrama

2.4 Denbora-estimazioa

Ondorengo taulan lan-pakete bakoitzari eskaini beharreko denbora azaltzen da, baina lan-pakete asko zeudenez, batzuk multzokatu egin dira modu argiagoan ikusteko.

Atazak	Orduak
informazio bilketa	47
Bertsolaritzaren datu-basea landu	10
BertsolarIXA analizatu	15
Silaba-kontatzailearen algoritmoa aztertu	4
Indexatzeko teknikak ikasi	8
Programazio-lengoiak ikasi	10
Analisia	9
Sistemaren analisia egin	9
Diseinua	27
Diseinu orokorra egin	9
Barne programazio orokorra diseinatu	15
Atalen arteko komunikazioa diseinatu	3
Inplementazioa	200
Web-interfazea implementatu	10
Datu-basea implementatu	5
Silaba-kontatzailea hobetu	15
Errima-bilatzailea hobetu / eraiki	20
Barne programazioa garatu	150
Proiektuaren kudeaketa	57
Bileren antolakuntza	8
Proiektuaren kontrola	4
Aurkezpenen prestaketa	10
Memoria egitea	35
BATURA	340

Hemen, ordea, aurreko taulan agertu diren eremuak bost ataza nagusitan bilduta daude, lanaren deskonposaketa egiturako ataza nagusietan, hain zuzen ere. Modu grafikoan azaltzen da zeini eskainiko zaion denbora gehiago.



Irudia 2.2: Estimatutako orduen banaketaren grafikoa

2.5 Lan egiteko metodologia

Lan egiteko metodologia arrunta izango du proiektu honek. Hasiera batean proiektuaren helburuak finkatu dira, lortu nahi zena, hain zuzen ere.

Behin helburuak finkatu ondoren, proiektuaren gauzatzeari ekingo zaio. Gauzatze honen barruan sartzen dira arbelaren atal ezberdinen diseinua eta hauen programazioa. Baina, gauzatze hori egin bitartean, nolabaiteko ebaluazio bat ere egingo zaio, inguruko jendearen laguntzaz, hala nola, bertso-eskola ezberdinetan probak eginez. Honekin esan nahi dena da, ez dela proiektu guztia goitik behera sortu eta gero probatuko, baizik eta garatu bitartean probatuko da, gertuko eremu batean bakarrik bada ere.

2.6 Proiektuaren arriskuak

2.6.1 Arriskuen zerrenda

Proiektu orotan bezala, lana egin bitartean hainbat arrisku daude. Hauek ezagutzeak haien soluzioak ematea ahalbidetzen du, ahal den heinean ekidinez eta ezin denean haien kalteak murriztuaz.

Hauek dira proiektu honek dituen arrisku nagusienak:

- Kurtsoan zehar garatuko denez, beste irakasgaiei espero baino denbora gehiago eskaini behar izatea, eta, beraz, estimatu baina denbora gutxiago proiektuari.
- Proiektuaren inplementazioaren datuen galera.
- IXA taldeko tresna ezberdinak beharko ditugun momentuan eskura ez izatea.
- Xenpelar dokumentazio zentroko datuak eskura ez izatea.

2.6.2 Arriskuen soluzioak

Agertu diren arazoei irtenbidea eman behar zaie, eta hemen azaltzen da arazo bakoitzari zein soluzio planteatu zaion, hurrenez hurren:

- Proiektuaren entrega-data atzeratu egingo da.
- IXA taldeko subversion zerbitzarira proiektuaren datuak igotzeko baimena dugu, beraz, programatzen den bakoitzean, `svn commit` agindua exekutatu eta sinkronizatua egongo da egindako lana. Subversion erabilia, datuak gordezteaz aparte, bertsio ezberdinak kudea daitezke, eta elkarlanerako aukera ere ematen du, programazioa adar ezberdinetan banatuz.
- Tresnaren bat erabilgarri egongo ez balitz, haren funtzionamendua simulatzen duen zerbait prestatu beharko litzateke, baina denbora asko horretan erabili gabe.
- Datuen bat eskura ez baldin badago, eskuz sartuko dira probarako datu batzuk.

Kapitulua 3

Sistemaren analisia eta diseinua

3.1 Analisia

Sistemaren analisia egingo da atal honetan. Sistemaren analisia egiterakoan, proiektuaren helburuak mahai gainean jartzen dira eta zehaztasun osoz zer egin behar den erabakitzen da, dauden baliabideen arabera.

Proiektu honen helburua jakina da, bertsoak sortzeko egun dagoen hizkuntza-teknologia erabilia, haiek bateratzea eta modu errazean erabiltzeko interfaze bat sortzea. Interfaze edo arbel digital honek erabiltzaileari laguntza eskainiko dio bertso bat idazteko.

Helburu horiek lortzeko, pentsatu beharra dago zeintzuk diren bertso baten ezau-garriak, eta horrekin definituko dugu zer beharko duen garatuko den bertsotarako arbel digitalak.

Bertsoa, modu labur batean adierazita, **lerro kopuru** zehatz bat eta lerro bakoitzean **silaba kopuru** zehatz bat dituen berbaldia da, **abestu** egiten dena, lerro batzuen bukaerek **errimatzen** dutelarik. Hona hemen Xabier Amurizaren koplak bat, bertsoaren definiziorako lagungarri dena:

Neurriz eta errimaz
kantatzea hitza
horra hor ze kirol mota
den bertsolaritza

Bertso hau, lau txiki bat da. Horregatik, lau lerro ditu eta bere lehen eta hirugarren lerroek 7 silaba izango dituzte, bigarrenak eta laugarrenak, aldiz, 6 silaba. Honetaz aparte, bigarren eta laugarren lerroek errimatu egiten dute: *hitza* eta

bertsolaritza dira errimatzen duten hitzak.

Ezaugarri hauek izanda, pertsona bati bertso bat idazten laguntzeko behar diren tresnetan pentsatzeko ordua da.

Lehenik eta behin bertsoaren **neurria** bera daukagu, beraz, erabiltzaileari zenbat lerro idatzi behar dituen adierazi behar zaio, eta lerro bakoitzean idatzi beharreko silaba kopurua ere adierazi beharko zaio gainera. Azkenik, errimatu beharreko lerroak ere nolabait markatu egingo dira, ikasle-bertsolariak zein lerrotan errimatu behar duen jakin dezan.

Aurrera segiz, ongi letorke **errima-bilatzaile** bat, zeinak hitz bat sartuta, harekin errimatzen duten hitzak itzuliko dituen. Horrekin batera, gerta daiteke esaldi batean hitz bat sartu nahi baina silaba gehiegi edo gutxiegi izatea, horretarako **sinonimo-bilatzaile** bat oso ondo letorkiguke antzeko esanguradun hitzak bilatu ahal izateko.

Berezitasun hauek guztiak jarrita, bertsoak nola idatzi behar ditugun esango digu sistemak, baina, behin bertsoa idatzita, arlo teknikoan (semantika kontutan izan gabe) ondo dagoen edo ez jakitea interesgarria da. Horretarako, **bertso-zuzentzaile** bat jarri beharko litzateke. Honi esker erabiltzaileak jakingo du ea errima txukunak jarri dituen eta silaba kopuru egokia duen lerro bakoitzean.

Horrez gain, bertso bat idazterako orduan, asko laguntzen du doinu batean pentsatzeak. Orduan, behin neurria aukeratuta, neurri horren arauak errespetatzen dituzten **doinuak erabiltzaileari erakustea** oso lagungarria da, haren doinua entzuteko eta partitura ikusteko aukerarekin. Gainera, sareko bideo baterako esteka bat ere erakuts daiteke (suposatzen da bideo hori doinu hartan abestutako bertso batena izango dela), audioa eta partituraren osagarri gisa.

Gainera, mesedegarria da bertsoa idatzi aurretik edo bitartean, **ideia** bat izatea. Ideia horrek ez du zertan silaba kopuru zehatz bat izan behar, ezta errimarik ere. Horretarako ongi etorriko litzateke ideiak idazteko eremutxo bat, bertsoa idatzi bitartean ikusi ahal izango litzatekeena. Hortik ideiak hartu eta neurrian sartzen saiatuko litzateke erabiltzailea.

Bertsoa idatzi ondoren, idazteak duen meritua kontuan izanda, erabiltzaileari bertsoa **gordetzeko** aukera eman behar zaio. Bertso hauek gordetzeko datu-base bat beharko da. Normala den bezala, bertsolariak norberaren izenean gorde nahi izango du bertsoa, horretarako **izen-emate sistema** ere garatu beharko da, hau

da, erabiltzailea eta pasahitza egiaztatu ostean gorde ahal izango ditu bertsoak.

Hauetaz aparte, ona litzateke bertsoak **sortetan antolatzeko aukera** izatea, gai bati erantzunez sortutakoak, esaterako. Beraz, gordetzerakoan sortetan gordeko dira.

Gordeta dago bertsoa, baina ez al litzateke interesgarria izango guk idatzi dugunaren **antzeko bertsoak datu-basean** aurkitzea? Hori ere egin lezake arbelak, gure eta datu-baseko kantuen oinak edo erlazio lexiko-semantikoak analizatuz. Baina horretarako bertsoak indexatu egin beharko dira, horrela bilaketa-motoreek egiten duten gisan bilaketak modu azkarrean egiteko.

Bukatzeko, ondo egongo zen bertsoari doinurik ezarri ez badiogu, aurreko kasuan bezala erlazio lexiko-semantikoak aztertuz, **doinu bat gomendatzea** sistemak.

Jadanik bertsoaren definizioan agertzen ziren lau ideia nagusietatik hiru aipatu dira (lerro kopurua, silaba kopurua eta errima), baina ezin ahaztu bertsolaritza ahozkotasunean oinarritzen dela, beraz, polita izango litzateke sistemak bertsozaleak sortutako bertso hori **abestea**.

Laburbilduz, honakoak izango dira, arbelak izango dituen ezaugarriak:

- Neurria aukeratzeko eremua
- Neurriari dagozkion doinuak
- Bertsoa idazteko eremua
- Ideia idazteko lekua
- Errima-bilatzailea
- Sinonimo-bilatzailea
- Bertso-zuzentzailea
- Bertsoa gordetzeko aukera
- kautotze-sistema
- Sortetan antolatzeko sistema
- Antzeko bertsoak bilatzeko zerbitzua
- Doinu-gomendatzailea

- Bertso-abeteslaria

3.2 Diseinua

Hemen analisisian azaldutako kontzeptuak nola gauzatuko diren plazaratzen da.

3.2.1 Klase-diagrama

Bertsotarako arbel digitala diseinatzerako orduan, analisisian agertu diren atal guztiak antolatzeko beharra zegoen. Diseinua egiteko objektuei zuzendutako metodologia erabili da.

Sistema klaseka antolatzerakoan, analisisian agertu diren elementuak barneratu dira. 3.1 irudian ageri den bezala, hainbat elementu daude sistema osoan. Klaseka banatu dira elementu garrantzitsuenak. Hauek dira bertan dauden klaseak eta haien atributu zein funtzioak:

- Bertsoa: Bertsoa bera adierazteko klasea. Honen atributuak nahiko intuitiboak dira; bertso aldagaian bertsoa bera testu formatuan dago. *Ident* bertsoaren identifikadorea da, bakarra. *Idea* aldagaia bertsoaren ideia idazteko erabiltzen da. *Neurria* eta *doinua* bertsoaren neurriaren eta doinuaren identifikadoreak besterik ez dira. *Egile* eremuan bertsoaren egilearen identifikadorea gordetzen da, hau datu-basean dagoen erabiltzaile bat izango delarik. Gero, *sorta* aldagaia dago, bertsoak multzokatzeko kontzeptua, hemen sortaren identifikadorea soilik gordeko da. Azkenik, bertsoa publikoa den edo ez adieraziko du *publikoa* aldagaiak.
- Neurria: Klase honen barruan neurriari dagozkion ezaugarri batzuk aurki daitezke, baina baita neurriaren inguruko metodo erabilgarri batzuk ere. Klasearen atributuak neurri baten oinarrizko elementuak dira, hala nola, bertso-estrofaren *izena*, *lerro kopurua*, *identifikadorea*, *silaba-bektorea* eta *errima-bektorea*. Azken bi hauek lerro bakoitzean zenbat silaba idatzi behar diren eta errimatu behar den edo ez adierazteko balio dute, hurrenez hurren. Datu-basean, bi bektore hauen elementuak marratxo (-) batekin banatuta egongo dira, datu-baseetan ezin baitira bektoreak gorde. Hauetaz aparte, badaude beste metodo estatiko batzuk, adibidez, *neurriakLortu*, orrialde nagusian erakutsiko diren neurriak lortzeko sortutako metodoa.
- Doinua: Klase hau atributu minimoekin sortu da, izan ere, aurrerago Xenpelar dokumentazio zentroko datu gehiago gehituko zaizkio. Bere atributuak honako hauek dira: doinuaren *izena*, *identifikadorea*, *doinu mota* (kontakizunezkoa,

irrizkoa, erlijiozkoa, ...), doinuaren *neurria* eta azkenik honen *partitura* (irudia), *audioa* (midi edo mp3) eta *bideoa* (youtube) dagoen biltokiaren helbidea. Hemen ere badago metodo estatiko garrantzitsu bat, *neurriarekinDoinuakLortu*, hain zuzen ere. Funtzio hori da, neurri bat aukeratutakoan hari dagozkion doinuak arbelean erakusten dituena.

- **Sorta:** Honen atributuak bertso sorta batek izan ditzakeenak dira. Sortaren *gakoa*, *pasahitza* (hutsa izan daiteke), *mota* (oraingoz, gaia motako sortak bakarrik daude), sortaren *gaia* eta sortaren *neurria*. Hemen dauden hiru metodo estatikoak datu-basea atzitu edo aldatzeko dira: *motaGaiaLortu*, gako eta pasahitza emanda sortaren mota eta gaia lortzeko, *nireSortakLortu*, erabiltzaile bat emanda bere sortak lortzeko, eta *sortaSortu*, datu-basean sorta bat sortzeko.
- **Silaba:** Klase honetan silaba kontatzailearen inguruko metodoak multzokatu dira, modu estatikoan. Bi metodo nagusi daude: *silabakKontatu*, esaldi bat emanda silaba kopurua itzultzen duena eta *azken2Silabak*, esaldi bat emanda azken bi silabak itzultzen dituena errima-bilatzaileerako.
- **Errima:** Hemen ere errimarekin erlazionatutako metodoak dira nagusi. Metodo nagusia, *errimakEgiaztatu* izenekoa da. Honek bi parametro jasotzen ditu, errimatu beharreko lerroen sekuentzia eta errimen egitura adieraziko duen bektore bat. Honek itzuliko duen emaitza, bektore bat izango da, lerro bakoitzean errimaren zuzenketarekin.
- **Erabiltzaile:** Klase hau oraingoz oso sinplea da. Erabiltzaileen kautotze-sistema egiteko erabiltzen da. Bertan metodo bakarra dago, *erabDago* izenekoa, zeinak true itzultzen duen baldin eta soilik baldin erabiltzailea existitzen bada. Aurrerago agian, eremu gehiago gehituko zaizkio.
- **Zerrendak:** Klase hau hitz zerrendak kudeatzeko erabiltzen da. Hitz bektoreek maiztasunarentzako eremu bat dute, eta haren arabera ordenatzeko *mergesort* algoritmoa inplementatu da klase honetan, *ordenatu* eta *merge* metodoekin. Baina beste bi metodo erabilgarri daude, 0 maiztasuneko hitzak eta hitz hutsak kentzeko, *zeroMaiztasunekoakKendu* eta *hitzHutsekoakKendu*, hurrenez hurren.
- **DB:** Hau datu-basea adierazteko erabiltzen den klasea da. PHPko *mysqli* klasearen metodo eta atributuak heredatzen ditu. Baina badu metodo bat gehiago, *kontsultaEgin*, alegia. Metodo honi parametro gisa kontsulta bat emanda, kontsulta burutu eta haren emaitza itzultzen du. Berez ez du ezer berezirik egiten, baina gauzak argiago ikusteko balio du.

- MaiztDB: Klase honek egiteko bakarra dauka. Hitz-bektore bat emanda, haien erabilera maiztasuna adierazten duten zenbakiekin betetzea bektorea. Horretarako, *hitzarenMaiztasunaLortu* metodoa dago, zeinari hitz-bektore bat emanda, bektore berean hitzaren maiztasuna ere idazten duen.

3.2.2 Datu-basearen diseinua

Klase diagrama azalduta, argi dago ezinbestekoa dela datu-base bat izatea eta haren diseinu orokor bat egitea. Datu-baserik izango ez bagenu, neurrien egiturak, adibidez, php lengoaiaz idatzi beharko lirateke eta sistema ez litzateke moldagarria izango bertso-estrofa berriak gehitzeko. Kontuan izan behar da, era berean, 8000 doinu baino gehiago gorde behar direla.

Beraz, ondoriozta daiteke datu-basea proiektuaren oinarri garrantzitsuenetariko bat dela. Bertan bertso-estrofa edo neurri ezberdinen egiturak gordetzen dira, doinuei buruzko datuak, erabiltzaileak eta bertsoak sortetan antolatuak. Sinonimobilatzaileak erabiltzen duen euskal WordNet[11] ere datu-base bat da, IXA taldeak erabilia eta aldatua, baina proiektu honen barruan garatu ez denez, ez dira horren diseinuaren nondik norakoak azalduko.

Datu-basearen diseinua ezer inplementatu baina lehenago egin zen, nahiz eta denboraren poderioz datuak eta eremuak gehitu zaizkion. Gure datu-basea erlazionala da, hau da, hainbat taula sortzen dira gorde beharreko elementu mota bakoitzera, elementu horiek hainbat eremu izango dituztelarik. Azken eremu horiek, hainbat taula elkartzeko gaitasuna dute, horretarako erlazioak aipatu egin behar direlarik. Horrela sortzen dira gure sistemako erlazioak *bertsoa* eta *neurria* tauletan. Bi taula horiek erlazionatuta daude, bertso bakoitzak bere neurria duelako. Erlazio hori sortu ahal izateko, *bertsoa* taulan *neurri* izeneko eremu bat sortzen da, *neurria* taularako erreferentzia bat izango delarik. 3.2 irudian datu-basearen entitate-erlazio diagrama dago.

3.2.3 Interfazearen diseinua

Atal honetan, analisisan ikusitako gauza ezberdinak web-interfazean nola antolatuko ditugun erabaki beharko da. Ezer aipatu baino lehenago, kontuan izan beharrekoa da, hemen aurkezten den hasierako diseinua Aitzol Astigarragarena dela. Proiektu honetan inplementatu egin da.

Hasieran, diseinu pobre xamarra zuen interfaze bat diseinatu zen, azpiegitura oso simpleaz, baina aurrerago hori hobetu egin zen. 3.3 eta 3.4 irudietan ikusten da interfazearen eboluzioa.

Web orria hiru zutabe bertikaletan banatuko da eta Erdiko zutabeari zabalera eta garrantzi handiagoa emango zaio, hor idatziko baita bertsoa. Beraz, has gaitetzen atalez atal behar duguna txertatzen.

Lehenik eta behin, bertsoaren neurria zein den jakin beharko litzateke. Horretarako, ezkerreko zutabean, zerrenda bat jarriko da, erabiltzaileak neurria aukeratzeko. Behin neurria aukeratuta, bertsoa idazteko lekuan neurri horren lerro kopurua agertu beharko da. Baina ez hori bakarrik, lerro bakoitzaren silaba kopurua eta zein lerrok errimatzen duten ere azaldu beharko litzateke. Hori egiteko, *input* motako testu-eremu txiki batzuk gehitzen dira bertsoaren lerro bakoitza adierazteko, eta haren ondoan idatzi beharreko silaba kopurua eta testu bat jarriko da. Idatzitako lerroa eta beharrezko silaba kopuruaren arabera *motz*, *ondo* edo *luze* agertuko da testutxo horretan. Idazteko testu-eremu hauek, gainera, kolore gardena baldin badute, esan nahi du ez dutela errimatu behar. Baina hainbat lerrok kolore berdina baldin badute, haien artean errimatu egin behar dutela esan nahi du.

Kontutan izan, neurri guztien artean, bat berezia egongo dela, neurri zehaztuga-bea, hain zuzen ere. Neurri hau aukeratuz gero, hainbat testu-eremu txiki (lerro batekoak) erakutsi beharrean, hainbat lerrotako testu-eremu handi bat agertuko zaigu, nahi dugun bertsoa nahi dugun neurrian idazteko. Kasu horretan, silaba kopuruak eta errimak ez dira aipatu ere egingo.

Errima-bilatzaile bat interesgarria izango zela aipatu da analisiaren atalean. Beraz, eskuineko zutabean erreminta-kutxa bat txertatuko da, eta erreminta horietako bat errima-bilatzailea izango da. Behin halako kutxa bat jarrita, bertan ere jarriko da, hirugarren zutabean, sinonimo-bilatzailea. Gainera, hitz baten errimak edo sinonimoak bilatzeko, bertsoan aukeratzearekin nahikoa da, behin testua saguarekin aukeratuta, erreminta-kutxan hitza automatikoki idatziko du eta errimak edo sinonimoak bilatu nahi izanez gero, “bilatu” izeneko botoiari sakatzea besterik ez dago.

Goian aipatu da neurria aukeratutakoan bertsoa idazteko lerroak kargatuko direla, baina beste gauza bat ere kargatuko da, oso lagungarri suertatuko dena. Neurria zein izango den erabakitakoan, neurri horrekin bateragarriak diren doinuak azalduko dira. Doinuak neurrien azpian agertuko dira, izan ere, erreminta-kutxan ez lirateke oso txukun geratuko eta ezta bertsoarekin batera ere. Baina doinuak jarrita, interesgarria da doinu hori entzun ahal izatea eta haren partitura ikustea. Horretarako,

hiru esteka egongo dira doinu bakoitzaren ondoan, audioa (botoiari sakatuta piano soinu doinua entzuteko), partitura (Joanito Dorronsororen doinutegiko partitura ikusteko) eta bideoa (doinu hartan kantatutako bertso bat youtuben). Gainera, zerrendaren azpian, beste botoi erabilgarri bat jarriko da, “doinua gomendatu”. Honek bertsoaren hitzak analizatuaz, nolabaiteko doinu egokiena gomendatuko dio erabiltzaileari.

Ez dugu ahaztu behar ezta, lehen agertu den ideiarekin kontzeptua. Honetarako hainbat lerrotako testu-eremu handi bat jarriko da eta ez da inongo silaba eta lerro mugarik ezarriko. Ideia bat idazteko balioko du, bestelako murriztapenik gabe, eta edozein momentutan bertsoa edo ideia ikusteko aukera egon beharko da, horrela bertsolariak ideia beharrezko neurrian sartzeko. Adibide bat jartzearren, ideian horrelako zerbait jar zitekeen; “1901-05-14, goizeko 9ak, Orioko barra aurea, balea agertu zen”. Bertsolariak horko ideia hori neurrian sartzen saiatu eta honako bertso polita lor dezake:

Mila bederatziehun
ta lehenengo urtian
maiatzaren hamalau
garren egunian
Orioko herriko
barraren aurrian
balea agertu zan
beatzik aldian

Behin bertsoa idatzita, ondo dagoen ala ez jakin nahiko dugu, horretarako, bertsoa idazteko lerroen azpian “bertsoa zuzendu” izeneko botoi bat izango da, zeinak sakatuz gero, bertsoaren zuzenketa bat emango digun, aspektu teknikoan soilik, semantika kontuan izan gabe. Hau da, aukeratutako neurriaren arabera, silaba kopurua eta errima egokiak dituen aztertuko du. Baina idatzi dugun bertsoa neurri zehaztugabea idatzi bada, nola zuzendu neurri gabeko bertsoa? Kasu horretan ez da bertsoa zuzentzeko botoia agertuko, beste botoi garrantzitsu bat baizik, “neurria ezarri”.

Neurria ezarri botoia, neurri zehaztugabeko bertsoei neurri bat zehazteko erabiltzen da. Botoi hori bertsoaren azpian agertuko da, “bertsoa zuzendu” botoiaren toki berean. Honek hainbat erabilera izan ditzake, adibidez, bertso klasiko bat aztertu nahi duenean erabiltzaileak eta haren neurria ez dakienean, neurri asignatzaileari bertso hori emanda, neurri posibleak itzuliko dizkigu. Honen azpian funtzionalitate erabilgarri bat izango dugu, “bertsoa gorde” botoia, alegia.

Behin halako bertso polita idatzita eta ongi dagoela egiaztatua, bertsoa gordetzeko botoia sakatuta, uneko bertsoa datu-basean gordeko da. Baina bada kontu bat, bertsoa gorde ahal izateko bertsolari ikaslea kautotuta egon beharko da. Horretarako, goian badago “Saioa“ izeneko botoi bat eta honek izena eta pasahitza sartzeko bi eremu erakutsiko dizkigu. Behin erabiltzailea eta pasahitza sartutakoan, datuak ondo badaude, kautotua geratuko da bertsolaria eta bertsoa gorde ahal izango du “bertsoa gorde“ botoia sakatuta.

Azkenik, aipatu beharra dago bertso-sortaren kontzeptua. Behin bertsoa idatzita erabiltzaileak bere sorta propio batean gorde nahiko balu, sorta aukeratzea besterik ez dauka. Horretarako, neurri zerrenda eta doinuen taularen artean, sorta zenbakia idazteko testu eremu txiki bat egongo da. Hor zenbakia idatzi eta gero ”Sorta aukeratu“ botoia sakatzerakoan, leihotxo batean sortaren pasahitza jarri beharko du, baldin badauka.

3.2.4 Erabilpen-kasu diagrama

Orain arte azaldutako dena bateratuko duen interfazearen egitura azaldu da, orduan, 3.5 irudian arbelak berak eskainiko dituen aukerak irudikatzen dira erabilpen-kasu diagrama baten bidez.

Eskema honetan, arbelaren erabiltzailea da pertsonaia nagusia eta horren inguruan daude prozesu edo ekintza guztiak. Arbelak eskaintzen dituen aukerak grafikoki adieraziak daude. Kontutan izatekoak dira *extends* izeneko lerro batekin elkartuak dauden prozesuak. Horrek zera esan nahi du, goiko ekintza burutzea derrigorrezkoa dela behekoa egin ahal izateko. *Uses* jartzen duen kasuan, ordea, ez da horrela. Adibidez, bertsoa idazteko ez dago neurria zehaztu beharrik, sistemak gomenda diezazuke neurri bat. Baina, bertsoa gordetzeko bai, bertsoa idatzi egin behar da gorde aurretik.

3.3 Segurtasunaren bermatzea

Proiektu honen garapenean, lana sarean erabilgarri jarriko denez, segurtasun-analisi bat egin da. Hemen topa dezakegun segurtasun-zulorik handiena SQL injection delakoarena da.

Segurtasun informatikoa¹, informazio-sistemen babesa bermatzea helburu duen

¹http://es.wikipedia.org/wiki/Seguridad_informática

diziplina da. Babes hori bermatzeko helburuarekin, hainbat protokolo, erremin-ta, metodo eta arau daude. Segurtasuna informatikaren atal bat da, oso zabala, izan ere, bere barnean hartzen ditu, software garapena, datu-baseen kudeaketa eta fitxategi-sistemen azterketa, beste batzuen artean.

3.3.1 SQL injection

SQL injection delakoa, kode-txertaketa bidez egiten den teknika bat da, zeinak datu-baseen maneian dauden segurtasun-zuloak aprobetxatzen dituen. Teknika honek arrisku bat suposatzen du erabiltzailearen eta datu-basearen arteko elkarrekintzan inongo galbaherik ez badago.

Adibidez, suposa dezagun web-orri bat daukagula eta bertan kautotzeko eremu bat, erabiltzaileak bere izena sartu behar duelarik. Bere izena jakin ezean ezingo luke sistemara sartu, baina imagina zerbitzariaren login sistemaren azpian kontsulta hau dagoela:

$$\text{kontsulta} = \text{"SELECT * FROM users WHERE name ="} \\ \text{" + erabIzen + "';"};$$

Galdera egin ostean emaitza bat baldin badago gutxienez, kautotzea onartu egingo da. Kontsulta hori izanda, erabiltzaile izen gisa ' or ' 1 ' = ' 1 izena jarriko bagenu, datu-basean galdetuko zatekeena honakoa izango zen:

$$\text{kontsulta} = \text{"SELECT * FROM users WHERE name = " or ' 1 ' =} \\ \text{' 1 ' ; "}$$

Beraz, ikusten denez, edozein erabiltzailek izango zukeen aukera kautotzeko, erabiltzaile izenik izan gabe.

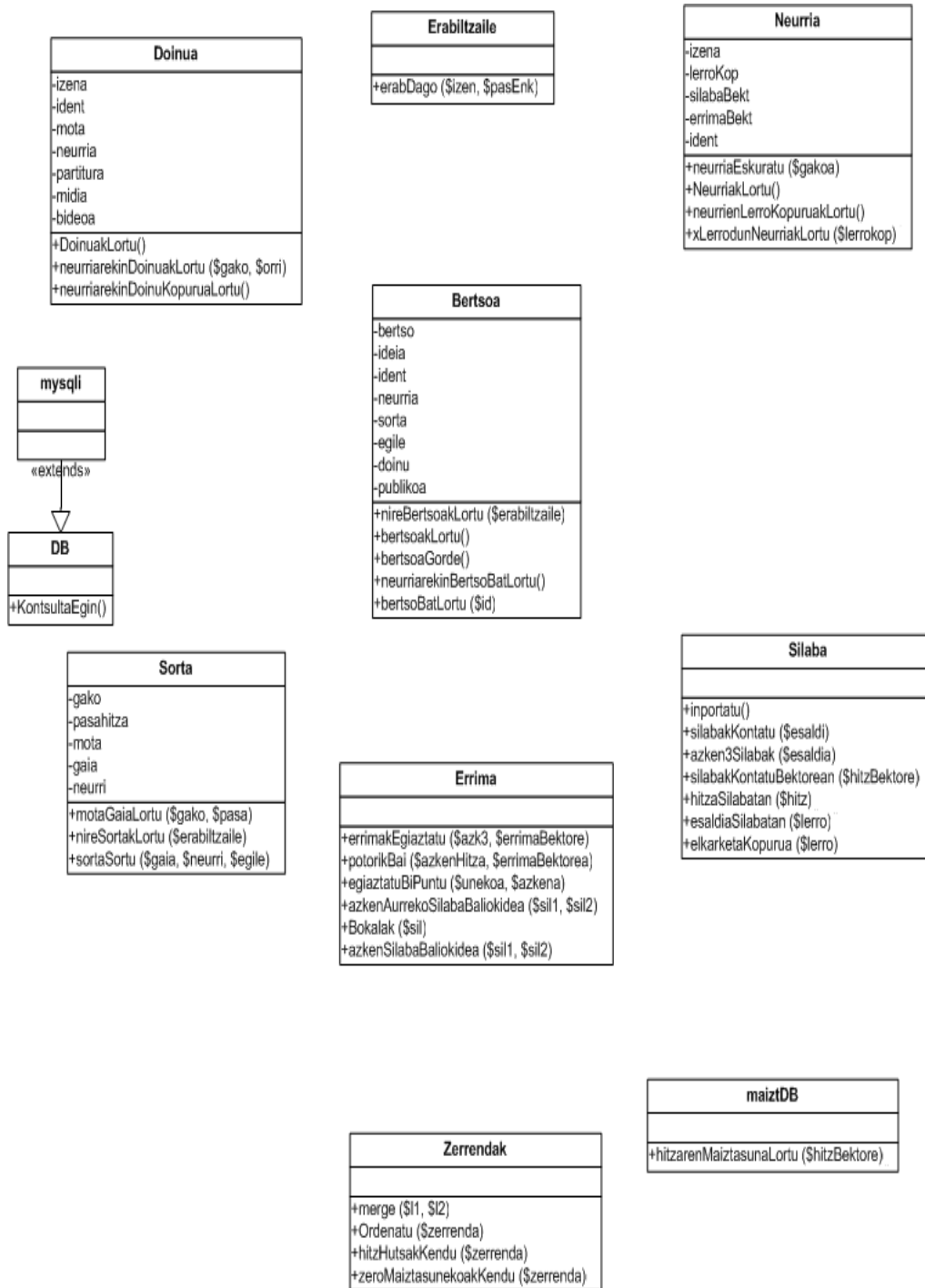
Honi soluzio bat² aurkitu zaio. Lehen aipatu den bezala, kode-txertaketa metodo hau arriskutsua da galbaherik ez bada jartzen. Orduan, modurik sinpleena galbahe bat jartzea da, besterik gabe. Horretarako, programazio-lengoaia ezberdinek aukera ezberdinak eskaintzen dituzte. Gure kasuan, PHPk `htmlspecialchars` izeneko funtzio bat dauka, zeinak karaktere bereziak HTML entitate bihurtzen dituen.

Ondorioz, ' or ' 1 ' = ' 1 erabiltzailearen izena, filtrotik pasa ondoren, ' or ' 1 ' = ' 1 izango litzateke eta kontsulta osoa honakoa:

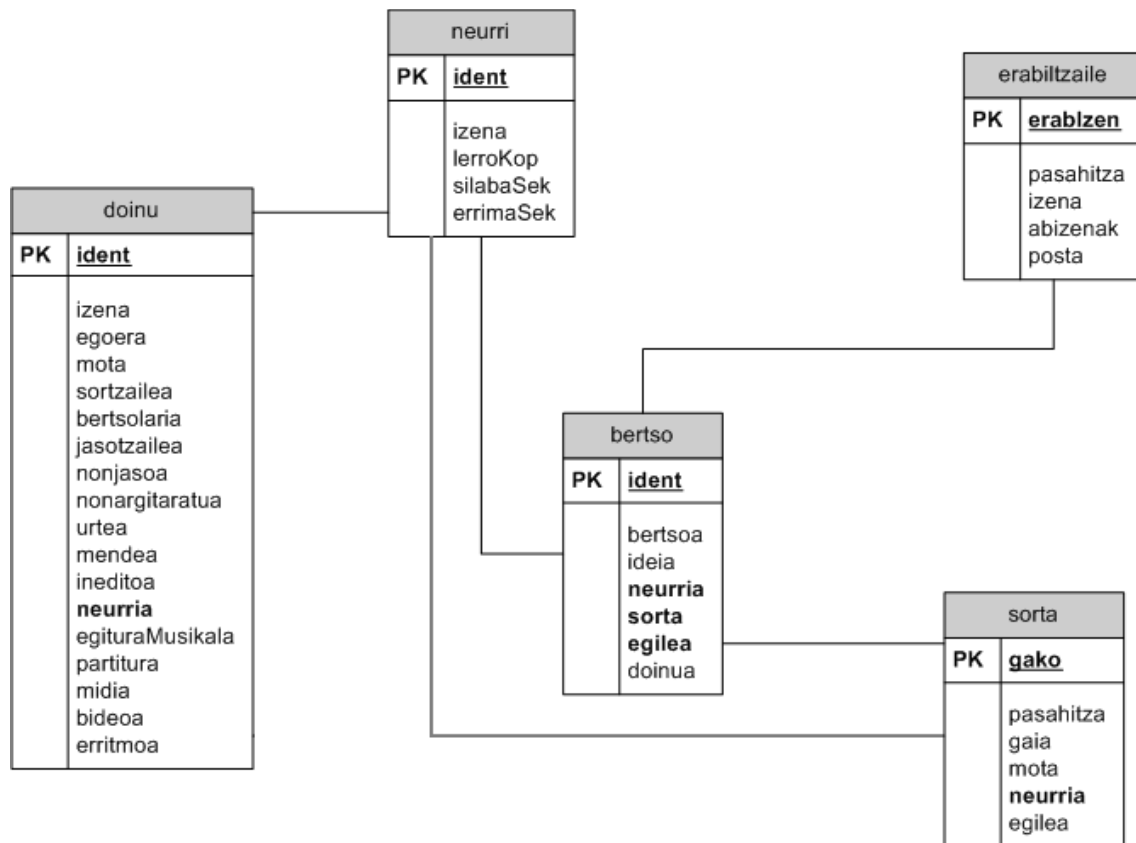
²http://en.wikipedia.org/wiki/SQL_injection

```
kontsulta = "SELECT * FROM users WHERE name =  
' or ' 1 ' = ' 1';"
```

Beraz, argi dago horrela ez dela iruzurrezko kautotzerik onartuko.



Irudia 3.1: Klase-diagrama



Irudia 3.2: Datu-basearen diseinua. Entitate-erlazio diagrama

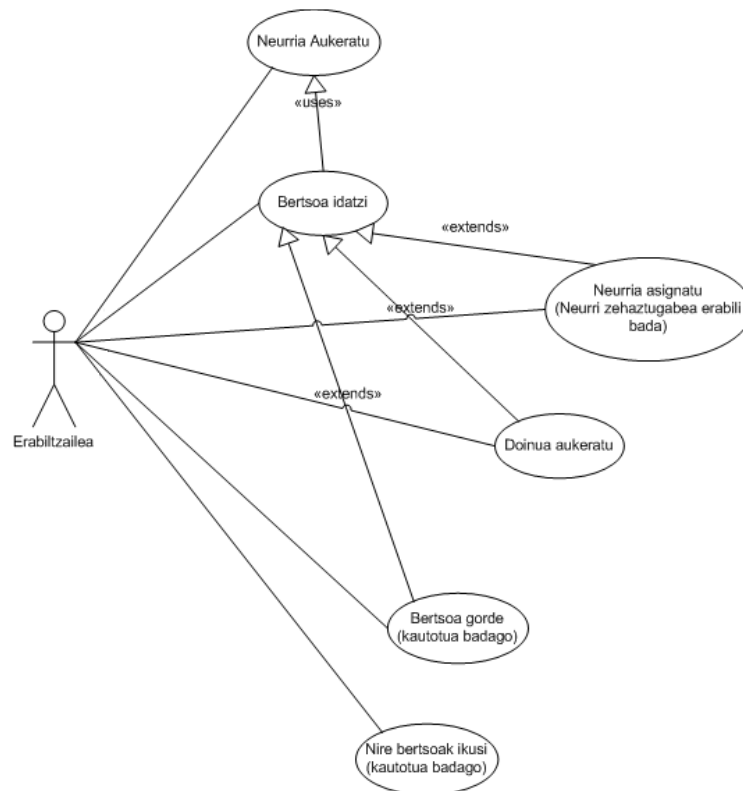
Bertsotarako Arbel Digitala



Irudia 3.3: Arbel digitalaren hasierako diseinua



Irudia 3.4: Arbel digitalaren diseinu berria



Irudia 3.5: Erabilpen-kasu diagrama

Kapitulua 4

Erabilitako teknologia eta azpiegitura

Kasu honetan, garapenerako behar izan diren tresna eta baliabide oro aipatzen da, zertarako erabili den adieraziaz eta hala izan den kasuan, nola garatu den.

Arbel digitala diseinatzerakoan, aho batez erabaki zen azpiegitura gisa web-teknologia erabiltzea, askoz ere erabilgarriagoa delako, aplikazio instalagarri bat baino. Gainera edozein sistema eragilerekin erabili ahal izango litzateke aplikazioa.

Azpiegitura aipatzeko eta azaltzeko garaian, hiru multzotan banatu da kapituluak. Lehenengo, azpiegitura teknologikorako erabilitako teknika eta lengoaiak azaltzen dira. Ondoren, hizkuntza-teknologiako sistemak agertzen dira eta amaitzeko hizkuntza-baliabideak.

4.1 Programazioa eta datu-baseen kudeaketa

Web-teknologian, garatzeko hainbat lengoia eta modu daude. Jarraian azalduko da nola garatu den proiektu hau.

4.1.1 Lengoaia nagusiak: PHP eta Javascript

Argi zegoen arbelak hainbat modulu izango zituela, beraz, objektuei zuzendutako programazioa ahalbidetzen zuen lengoia bat erabiltzeko beharra nabaria zen. Datu-baseekin konexioa ere modu errazean eskaintzen zuen lengoia behar zuen, ondorioz, beharrak mahai gainean jarrita, proiektuaren zerbitzariko atalak **PHP**[3] lengoiairekin garatzea erabaki zen. Hau erabiltzeko beste arrazoi garrantzitsu bat,

Linux eta Windows plataformetan instalatzeko eskaintzen dituen erraztasunak dira. Baina zerbitzariko atalaz aparte, bezeroak ere lan batzuk egin behar ditu eta bezeroako atalak programatzeko **Javascript**[3] erabili da.

4.1.2 AJAX teknologia

Javascript eta PHP lengoaiak erabiltzean, **AJAX**[4] teknologia erabiltzeko aukera zegoen. AJAX teknologiari esker, HTTPREQUEST izeneko objektuak erabilia, web orrian edozein momentutan zerbitzariko datu bat behar izanez gero, ez dago web orri osoa berriro kargatu beharrik.. Honi esker zerbitzariari egiten zaizkion eskaeren tamaina asko murrizten da.

4.1.3 XML

AJAX erabilia, modulu ezberdinen artean komunikazioa modu txukunean egin beharra zegoen. Horretarako hiru aukera zeuden, JSON notazioa (Javascript Object Notation) erabiltzea, XML lengoia erabiltzea edo bestela emaitzak testu hutsean itzuli eta hauek string gisa tratatzea, hauetatik beharrezko emaitzak ateraz. Testu huts gisa tratatzea hasiera hasieratik baztertutako metodo bat izan zen, komunikaziorako ez baita metodo batere egokia. Beste bien artean, **XML** lengoaiaren aukera gailendu zen, gehienbat, beste lengoia askok ere markatze lengoia honekin bateragarritasun handia dutelako.

4.1.4 MYSQL eta phpMyAdmin

Datu-basearen sorkuntza eta kudeaketarako software libreko **MYSQL**[3] erreminta erabili da. Aukera hori egin da fakultateko irakasgai gehienetan tresna hori erabili delako eta gainera **phpMyAdmin**[6] izeneko interfaze erraz bat erabil genezakeelako. Honekin jarraituz, esan beharra dago erlazioak ere modu errazean egiten direla phpMyAdmin erabilia.

4.1.5 Flash

Flash lengoiaz programatutako aplikazio bat ere erabili da, *audioplay* izeneko. Ez da software irekikoa, baina doakoa bai. Horri esker, doinuen mp3 fitxategiak entzun egin daitezke inongo deskargarik egin behar izan gabe.

4.2 Hizkuntza-teknologia

Hemen proiektu honetan erabili den hizkuntza-teknologia azaltzen da. Aipatuta dagoen bezala, hizkuntza-teknologia hizkuntzalaritza eta informatika biltzen dituen

diziplina da. Atal honetan, hizkuntza teknologia hori nola eta zertarako erabili den azaltzen da.

4.2.1 FOMA

Foma[18] automatak eta transduktoreak sortzeko konpilatzaile bat da. Lengoia naturalaren prozesamenduan asko erabiltzen diren analizatzaile morfologiko eta fonologikoentzako bereziki ona da. Gainera, liburutegi bat ere eskaintzen du, C programazio lengoiaian automatak modu errazean erabili ahal izateko. Etorkizuneko lan batzuk honen APIarekin garatuko dira.

Kasu batzuetan automatekin lan egin behar izan da, kasu horietan, foma software libreko tresna erabiltzea erabaki da. Hori erabiltzeko arrazoi nagusia, software askekoa izateaz aparte, haren programatzailea, Mans Hulden, fakultatean egon dela da. Urte osoan egon da bertan, Donostian, eta honi buruzko ikastaro bat antolatu zuen urtarrilean.

Foma proiektu honetan bi elementuetarako erabili da, errima-bilatzailea garatzeko eta hitzen maiztasunak modu azkarrean eskuratzeko. Bigarren honek konplikazio batzuk izan ditu, izan ere, 463389 hitz dituen fitxategi batean atzipenak egitea ez da lan makala ordenagailuarentzat. Eta sarrera guzti horiek dituen automata bat egitea ere ez da lan erraza, beraz, horretarako foma tresnak eskaintzen duen aukera bat erabili da. "lexc" formatuan[17] idatzita hitz guztiak irakurri eta automata bat sortzea oso azkarra suertatu da.

4.2.2 Xuxen

Xuxen[9] euskarazko zuzentzaile ortografikoa da, IXA taldeak eta Eleka-k garatua. Testu prozesatzaile gehienetan erabil daiteke eta barruan, euskarazko hiztegi handi bat dauka.

Proiektu honetan, fomaz garatutako xuxen hiztegiaren aldaera bat erabili da. Aldaera honi sarrera gisa bukaera bat emanda, askoz jota bi atzizki dituen hitz bat itzultzen du. Hori egiteko arrazoia, zera da, errima aberatsenak atzizki gutxien dituztenak direla suposatu dugulako, eta orduan muga bezala bi atzizki jarri dizkiogu.

4.2.3 Euskal WordNet

Euskal Wordnet[11] euskarazko lehen ezagutza base lexikala da, AEBtako *Princeton*-go unibertsitatean garatutako *Wordnet*-en oinarritutakoa. Honi esker, nahiz eta oraindik datu batzuk gehitzeko egon, euskarazko hitzen sinonimoak, hiponimoak,

antonimoak, etab. lortu daitezke. *Wordnet*-en hitzak sinonimo taldeetan biltzen dira, antzeko esanahidun hitzetan, alegia.

Wordnet-en ezaugarria da ere, hainbat hizkuntzatarako dagoela, eta ohizkoa den bezala, hizkuntza osatuena ingelesa da.

Guk sinonimiarako bakarrik erabili dugu, baina badu arazo bat oraindik. Une honetan "etxea" bilatuz gero sinonimo-bilatzailean, ez lukeela sinonimorik aurkituko, lematizatzaileak ez duelako, izan ere, "etxe" hitza bilatuta sinonimoak aurki ditzake. Beraz, etorkizuneko lan gisa uzten da lematizatzaile bat txertatzea, atzizkidun hitzentzat.

4.2.4 Festival

Festival[14] ahots-sintetizadore bat da, GPL lizentziaduna, hau da, software askekoa. Ahots-sintetizadore hau **CSLU Toolkit**[15] izeneko erreminta-kutxa batek erabiltzen du. Bertan ikusi ahal izan genuen, ahotsari pitch-a (ahotsaren nota) aldatzea ez zela batere zaila. Baina, aurrerago, "singing mode" edo abesteko modua ezagutu genuen. Honen funtzionamendua oso sinplea da; XML dokumentu batean bertsoaren silaba bakoitzarekin batera, horri dagokion musika-nota eta iraupena jarri eta berak egiten du beste lan guztia. Agindu gutxirekin, ordenagailuak bertsoa abestu egiten zuen, baina horrek bazituen alde txar batzuk. Euskaraz abesteko, euskararen fonemak gordeta beharko lituzke, eta hori oraindik ez dagoenez, erdarazko fonemekin abesten zuen eta gauza batzuk nahiko gaizki geratzen ziren. Hori gutxi balitz, silabak kateatu egiten zituen, hau da, modu independentean sintetizatu eta elkartu, beraz, ahotsa artifizial xamar gelditzen zen.

4.2.5 Aholab

Bilboko ingeniariak eskolan ikerkuntza talde bat dago, **Aholab** izenekoa. Beste gauza batzuen artean, euskarazko ahotsaren sintesian lan egiten dute eta hori oso interesgarria da lan honentzat. Beraz, egonaldi txiki bat egin nuen Bilbon, ordenagailua bertsotan jartzeko. Aholab-en sintetizatzailea erabiltzeko alde onak nabariak dira, hala nola, euskarazko ahots-sintesia askoz ere landuagoa dutela, eta silabizazio arrunta egin ordez fonemekin egiten dutela silabizazioa, gainera, silabak ez dituzte zuzenean kateatzen festival-en bezala. Zer esan nahi da horrekin? Adibidez, mainontzi hitza, festival-ek `mai_non_tzi` irakurtzen du, aholab-eko `ahoTTS`[12] sintetizatzaileak, ordea, `ma.ñon.tzi`. Badu, ordea, alde ez oso on bat, oraingoz librea ez dela, beraz, lizentziak pixka bat borrokatu egin beharko da. `AhoTTS` sintetizatzailea erabiltzeaz aparte, `ahoCoder`[13] aplikazioa ere erabiltzen da, `ahoTTS`-ren irteera eraldatuak birkodetzeko.

4.3 Baliabide linguistikoak

4.3.1 Bertsolaritzaren datu-basea

Bertsolaritzaren datu-basea[21] Xenpelar Dokumentazio zentroaren egitasmo bat da. Honek urteetako lana biltzen du, eta eduki ezberdinak dauzka, hala nola, bertso-doinutegia (Joanito Dorronsororen doinutegiaren[20] bertsio digitala), bertso-estrofa edo neurrien zerrenda, bertso zerrenda ikusgarria eta beste hainbat dokumentu. Honen helburu nagusia Xenpelar Dokumentazio zentroaren edukia internet bidez gizarteratzea eta maila guztietako ikerkuntza bultzatzea da.

Lan honetarako berebiziko garrantzia izan du eduki honek, izan ere, probak egin ahal izateko bertso-estrofa, doinu eta bertso ezberdinak eskura izateak asko lagundu digu.

4.3.2 Egunkariako corpusa

Errima-bilatzailearen emaitzak zerrenda ez-ordenatu gisa erakutsi beharrean, ordenatu egin dira. Horretarako IXA taldearen ***Egunkaria*** euskal egunkariko corpus baten hitz zerrenda erabili da. Baina hitz zerrenda horrek berezitasun bat du, hitz bakoitzarekin batera, corpusean izan duen agerpen-kopurua ere idatzita dago, erabilera-maiztasuna, alegia. Kazetaritza corpusa denez, guk behar bezain neutroa dela esan genezake, beraz, hemen gehien agertzen diren hitzak hizketa naturalean ere gehien agertzen direla suposa daiteke. Corpus honetan 463389 hitz daude.

Kapitulua 5

Garapena

Atal honetan, arbelaren barne programazioaren atal batzuk landuko dira. Arbelak hainbat modulu ditu bere barnean eta hemen zati bakoitzaren deskribapen labur bat egingo da.

Arbel digitalaren barruan hainbat elementu egon behar dira, diseinuan aipatu denez. Atal horiek honako hauek dira:

- Silaba-kontatzailea
- Errima-bilatzailea eta errima-egiaztatzailea
- Bertso-zuzentzailea
- Sinonimo-bilatzailea
- Neurria zehazteko zerbitzua

Atal hauek azaldu ostean, guztiaren integrazioa nola burutu den azalduko da eta probak nolakoak izan diren, emaitzak ebaluatuz.

5.1 Silaba-kontatzailea

Esaldi bat eman eta zenbaki natural bat itzultzen duen programa. Programaren hasieran bokal, diptongo, kontsonante eta kontsonante-taldeak definitzen dira. Behin hori dena eginda, hitzetan banatzen da esaldia eta hitz bakoitzarekin `HitzaBanatuSilabatan()` funtzioari deitzen zaio. Honek hitzaren silabak bektore batean itzultzen ditu eta bektore hori itzultzen du. Silaba kopurua jakiteko, PHP-ko bektoreen agindu klasiko bat erabiltzea besterik ez dago, `count()` funtzioa, hain

zuzen ere, zeinak elementu kopurua itzuliko digun. Ondoren, kontatzailearen algoritmo bat dago, lagungarri suertatuko delakoan.

```

EsaldiaSilabatan ()
EsaldiaHitzetanBanatu
HitzakDaudenBitartean
    Hitza = HurrengoHitza
    HitzaAtzekozAurrera = HitzariBueltaEman (Hitza)
    Silabak = HitzaBanatuSilabatan (HitzaAtzekozAurrera)
Itzuli Silabak --> Gero count(Silabak) eginda lortzen da silaba kopurua

```

```

HitzaBanatuSilabatan (HitzaAtzekozAurrera)
HitzaAtzekozAurrera!='' den bitartean
    Azken Letra Bokala eta Hiatoa da?
        BokalaSilabaZerrendanSartu()
        BokalaHitzetikKendu()
    Bokal + (Kontsonante edo Kontsonante taldea) egiturarekin bukatzen da hitza?
        AurkituDenHoriSilabaZerrendanSartu()
        AurkituDenHoriHitzetikKendu();
    (Kontsonante edo Kontsonante taldea) + (Bokal edo diptongo) +
    (Kontsonante edo Kontsonante taldea) egiturarekin bukatzen da hitza?
        AurkituDenHoriSilabaZerrendanSartu()
        AurkituDenHoriHitzetikKendu();
    (Kontsonante edo Kontsonante taldea) + (Bokal edo diptongo)
        AurkituDenHoriSilabaZerrendanSartu();
        AurkituDenHoriHitzetikKendu();
    Bestela --> Azken silaba denean
        AurkituDenHoriSilabaZerrendanSartu();
        AurkituDenHoriHitzetikKendu();
Itzuli SilabaZerrenda

```

Kasu batzuetan, ordea, gerta daiteke esaldi berdinak hainbat silaba-kopuru izatea, izan ere, elkarketak egin daitezke hitz ezberdinen artean. Beraz, silaba-kontatzaileari beste funtzio bat gehituko zaio, `elkarketaKopurua()` izeneko, alegia. Funtzio honek silaba-kontatzaileari parametro gisa pasatako esaldi bera jasotzen du eta honek patroi jakin batzuk bilatzen ditu esaldi honetan, bokal-h-bokal eta bokal-hutsune-bokal patroiak, alegia. Adibide batekin hobeto ulertuko da.

Suposa dezagun bertso baten lerro batean honako esaldia agertzen dela, “ta lehenengo urtian”. Modu arruntean silabak zenbatuta, honek zortzi silaba dituela esango litzateke, ta-le-he-nen-go-ur-ti-an. Baina, ez al dago aukera ta-le-nen-go-ur-ti-an modura abestu eta zazpi silaba izateko? Eta ta-le-nen-gour-ti-an esanda gaizki al dago? Beraz, hemen planteatzen den soluzioa zera da. Aurreko esaldiak berez zortzi silaba dituela esango da, baina sei edo zazpi silaba dituela esatea ere ez dago gaizki (bi elkarketa patroï aurki daitezkeelako). Hori bai, elkarketa dezente egin direnez, erabiltzaile edo ikasle bertsolariari abisu bat emango zaio; izan ere, horrelakoak ekidin egin behar dira. Bi elkarketa edo gehiago egiten badira ematen da abisua.

Erabili den silaba-kontatzailea lehenagotik garatua zegoen beste baten itzulpen eta moldaketa izan da. Hasieran, perl-ez idatzita zegoen (Aitzol Astigarragak egin), baina honen alde txarrak ikusita, hala nola, php script batetik beste lengoia bati deitu behar izatea eta gainera komunikazio-modu ez oso garbi batean (testu lau moduan), itzulpen bat egitea erabaki zen (php-ra). Itzulpena ez zen konplikatu izan gehienbat php eta perl lengoaiak oso antzekoak direlako. Behin itzulpena eginda, programaren abiadura modu nabarian handitu da. Gainera, lehen bertsoak, ez zuen hitzen elkarketaren arazoa konpontzen, hau da, hasieran silaba kopuru baxua itzultzen zuen.

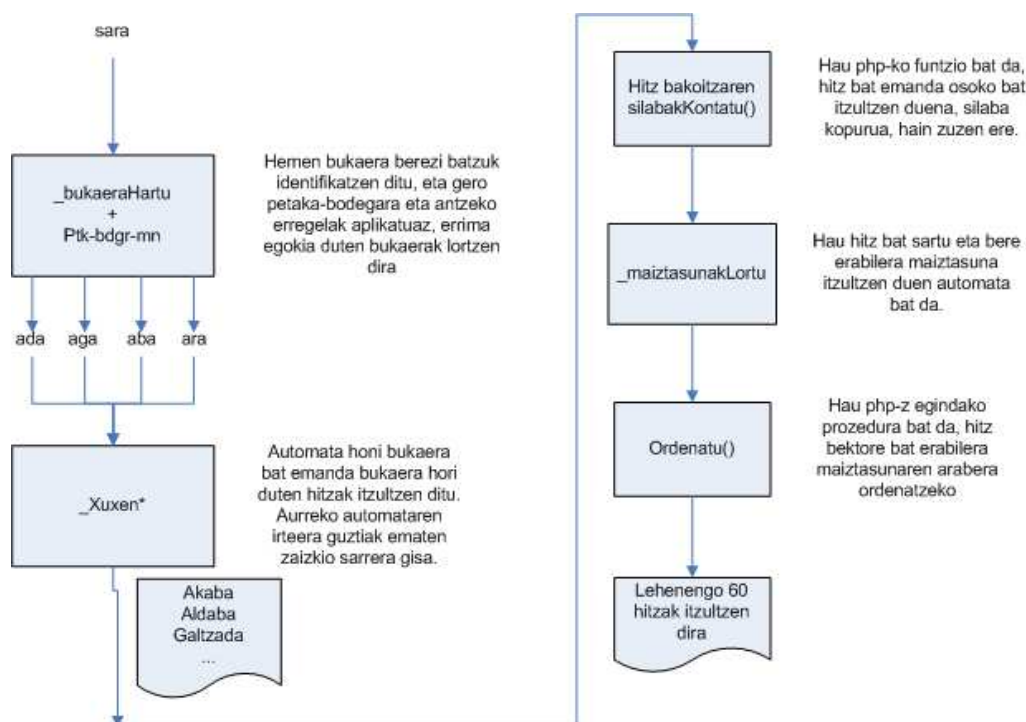
5.2 Errimak

Silaba kontatzaileaz aparte, hitzen errimak bilatzeko gai izan behar zuen arbelak, eta baita bi hitzek errimatzen ote zuten edo ez esateko ere. Jarraian azaltzen da nola garatu diren errimekin lan egiteko sortutako tresnak.

5.2.1 Errima-bilatzailea

Programa honi hitz bat ematen zaio sarrera gisa eta honek hainbat hitz itzultzen ditu. Lehenengo *foma* tresnarekin sarrerako hitzean errima-patroia bilatzen da, adibidez, “hartzera” hitza izango bagenu sarrera gisa hitz hartako azken hiru letrik patroï garrantzitsu bat osatuko zuketean, bokal-(kontsonante_ahostuna edo ‘r’ biguna)-bokal patroia, hain zuzen ere. Horren moduko beste hainbat patroï daude gordeta sisteman. Behin “era” bukaera lortuta, jakin beharrekoa da zein bukaeradun hitzak nahi ditugun, kasu honetan argi dago “era”, “eda”, “ega” eta “eba” bukaera duten hitzak bilatzen ditugula. Lau hitz zati horiek xuxen-en moldaera bati bidaltzen zaizkio eta haiekin bukatzen diren hitz guztiak itzuliko dizkigu honek. Azkenik

emaitzan ageri diren hitz guztiak beren erabilera maiztasunaren arabera ordenatzen dira (Egunkaria-ko Corpora erabilia) eta ondoren, nahi izanez gero, silaba kopuruaren arabera mugatu daitezke. Silaba kopuruaren araberrako muga egiteko, hitz bakoitzaren silaba kopurua kalkulatzen da.



Irudia 5.1: Errima-bilatzailearen algoritmoa

Errima-bilatzailea garatzeko hainbat aukera izan genituen esku artean. Aukeretakoa bat bertsoarIXA[8] erabiltzea zen. Haren alde ona sistema garatu behar ez izatea zen, eta gainera kategorizazio moduko bat egiten zuela, hau da, “pena” hitzaren errimak bilatuz gero ez zituela bakarrik “ena“ bukaera zuten izenak itzultzen. Sorkuntza bat egiteko gai zen eta superlatibo guztiak itzuli beharrean, “adjektibo + en + a” moduko espresio bat itzultzen zuten (handiena, txikiena, ...). Baina bere moteltasuna medio, soluzio hori atzera bota genuen. Horrekin batera, haren funtzionamendua ere ikasi beharko genukeen. Beste aukera bat, foma tresnarekin, xuxen hiztegiaren aldaera bat sortzea zen. Aukera honen alde egin genuen. Horretarako arrazoiak, abiaduraz aparte, foma software askeko tresna izatea zen, eta bertsoarIXA-ren azpian zegoen XEROX, ordea, sistema pribatiboa. Lehen aipatu den bezala, gainera, php lengoaiatz hainbat hobekuntza gehitu zaizkio errima bilatzaile arruntari, hala nola, silaba kopuruaren arabera emaitza murriztea eta hitzek

duten erabilera maiztasunaren arabera emaitzak ordenatzea.

5.2.2 Errima-egiaztatzailea

Errima-egiaztatzaileari bi hitz edo esaldi ematen zaizkio eta honek 0, 1 edo 2 itzultzen du. Errimaren aberastasunaren arabera da ebaluazio hau, nahiz eta ebaluazio metodoak bertsozale elkartearekin berrikusteke dauden. Notarik hoberena 2 izango litzateke. 0 itzultzen du bi sarrerek errimatzen ez badute. 1 itzultzen du baldin eta soilik baldin bi sarrereren azken silabak berdinak edo hoskideak badira, hau da, bokala berdina bada eta kontsonantea berdina edo talde berdinekoa bada. Taldeak, hauek dira: p, t eta k; b, d, g eta r; eta bukatzeko m eta n. Programak 2 erantzungo du baldin eta soilik baldin bi sarrereren azken silabak hoskideak badira eta azken aurreko silabaren bokalak berdinak badira. Hona hemen adibide bat:

$$\begin{array}{l}
 0 \\
 \text{Deba-Sagardo: } ba \neq do \text{ eta } d(e) \neq g(a)r \\
 \text{Deba-esteka: } ba \neq ka \text{ eta } d(e) = t(e) \\
 1 \\
 \text{Deba-Kuba: } ba = ba \text{ eta } d(e) \neq k(u) \\
 \text{Deba-gara: } ba = ra \text{ eta } d(e) \neq g(a) \\
 2 \\
 \text{Deba-hartzera: } ba = ra \text{ eta } d(e) = tz(e) \\
 \text{Deba-Ameba: } ba = ba \text{ eta } d(e) = m(e)
 \end{array}$$

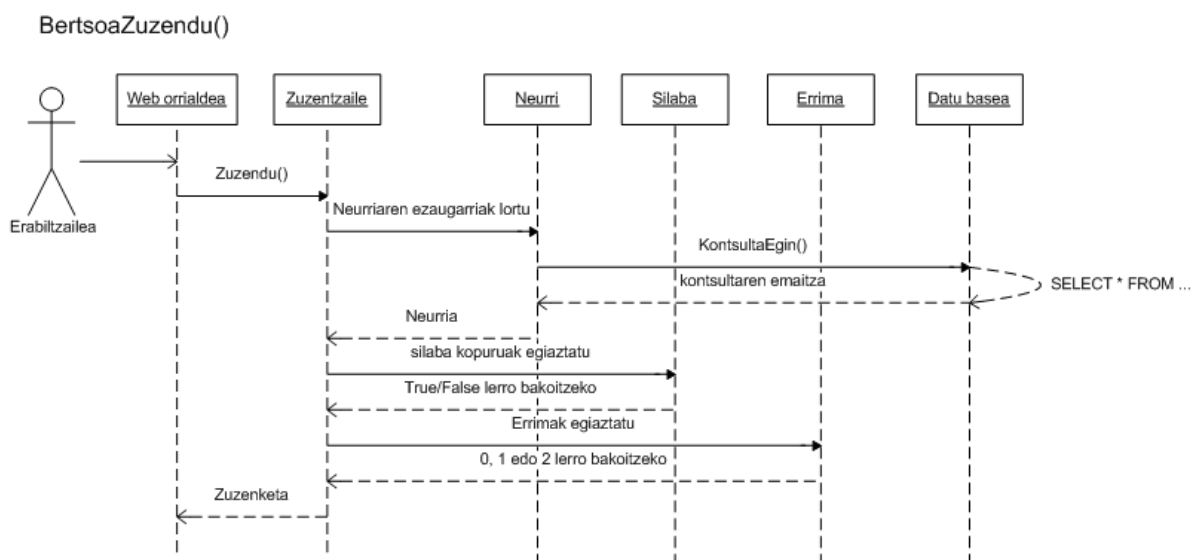
Lehenengo kasuan, adibidez, “DebaÂ· eta Â·sagardoÂ· izenak jartzen dira parezpare errima egiaztatzailean. Ulertzekoa da, azken silabak hoskideak ez direnez, errima txar bezala markatzea. Beste kasu batean, berriz, ”Deba“ izena eta ”gara“ aditza ageri dira. Hemen azken silabak hoskideak dira, “ba” eta “ra”, alegia, baina azken aurreko silabaren bokalak ez dira berdinak, beraz, errima txarra dela ez dugu esango, baina bai pobre xamarra dela. Azken kasuan, adibidez. “Deba” eta “Ameba” izenekin, azken silabak hoskideak eta azken aurreko silabaren bokalak berdinak direnez, errima egokia dela esango da.

Errima-egiaztatzailea egiteko orduan, hiru aukera nagusi izan genituen. Lehen aukera php-z Xabier Amurizaren hiztegi errimatuko [19] erregela guztiak php-z idaztea zen. Sistema oso ona izango litzateke horrela egin izan bagenu, baina lanaren konplikazioa handituz joan zen. Bigarren aukera, foma tresna erabili eta haren API-a erabilita eraikitzea zen, baina izan zezakeen konplikazioagatik, alde batera utzi zen. Hirugarren aukera, askoz ere errazagoa eta sisteman erabiliko zena, php-z azken bi silabak lortu eta konparazio sinple batzuk egitea zen.

Argi ikusten denez, errima-egiaztatzailearen implementazio hau, ez da optimoa, baina bai askoz ere sinpleagoa.

5.3 Bertso-zuzentzailea

Bertsoa idatzi ondoren norbere burua arlo teknikoan ebaluatzeko bertso-zuzentzailea dago. Programa honen funtzionamendua oso sinplea da. Pasatzen zaizkion parametroak bertsoa bera eta bertsoaren neurria dira. Behin datu horiek esku artean izanda, lanean jartzeko ordua da. Lehenik eta behin, neurriaren ezaugarriak lortzen ditu, lerro kopurua, errima sekuentzia eta silaba sekuentzia. Gero, berez, bertso-ko lerroen silaba kopuruak (elkarketa kopuruak kontutan izanda) neurriaren silaba kopuruekin bat datozen begiratzen da. Azkenik, errimak egiztatzeko, errimatu beharreko lerroetako bukaeretan errima egiaztatzailea erabiltzaz. 5.2 irudian funtzionamenduaren eskema bat ikus daiteke.



Irudia 5.2: BertsoaZuzendu() funtzioaren sekuentzia-diagrama

5.4 Sinonimo-bilatzailea

Sinonimo-bilatzaileak *Euskal Wordnet*-en gainean lan egiten du. Kontsulta batekin sarrerako hitzaren adiera ezberdinak, *synset*-ak, lortzen dira eta gero *synset* horiei dagozkien sinonimoak aurkitzen dira. Hasiera batean sinonimo-bilatzaile honi

silaben filtroa jartzea pentsatu zen, baina honen emaitzak hain murrizak izanik (sinonimo-bilatzailearenak), alferrikakoa iruditu zitzaigun. Baita ere aipatu beharra dago, hitzak atzizkiekin jarritz gero, honek ez duela emaitzarik itzuliko, lematizatzailearik ez baitu, hau etorkizuneko lan gisa geldituko delarik.

Wordnet-en hitz baten sinonimoak aurkitzeko bi SQL galdera egitea nahikoa da. Hitzak sinonimo taldetan multzokatuta daudenez, hitz baten sinonimoak bilatzeko, haren taldekideak zeintzuk diren jakitea besterik ez da egin behar. Ondorengo bi SQL galderak dira sinonimo-bilatzailearen oinarria:

```
SELECT offset FROM 'wei_eus-30_variant' WHERE word='$hitza'
```

Goian egindako galderan, hitz bakoitzak duen adiera bakoitzeko *synset* bat itzultzen du. Ondoren, adiera bakoitzari dagokion hitza bilatu behar da, hau da, sinonimoen zerrenda. Hori ondorengo galderan egiten da.

```
SELECT word FROM 'wei_eus-30_variant' WHERE offset='$offseta'
```

5.5 Neurria zehazteko zerbitzua

Zerbitzu honi esker, sistemari bertso bat eman eta hau gai da neurri edo bertso-estrofa bat itzultzeko. Honen funtzionamendua sinplea da.

Lehenengo, bertsoaren lerro kopurua zenbatzen da eta lerro kopuru horrekin bat datozen neurriak kargatzen dira bektore batean. Neurri bakarra aurkituko balu, emaitza gisa neurri hori itzuliko du sistemak. Baina, bat baino gehiago aurkitzen baditu, bilatzen jarraitu beharko du. Ondoren, silaba kopuruak, lerroz lerro banan-banan parekatuko ditu, kontuan izanda silaba kopuru minimoa eta maximoa lerro bakoitzean. Galbahe hau pasatzen duten neurriak itzultzen ditu sistemak, XML bidez. Beraz, web zerbitzu gisa eskeintzeko ez dago inongo arazorik. Ondoren, algoritmo bat agertzen da honen funtzionamendua azaltzen duena:

```
lerroKop = bertsoarenLerroKopuruaLortu ()
neurriZerrenda = XLerrodunNeurriakLortu (lerroKop)
neurriZerrenda-n neurri bakarra dago?
    NeurriaItzuli
Bestela
    bertsokoLerroetanSilabakKontatu ()
    neurriakEtaBertsoarenSilabaKopuruakParekatu ()
```


ondoParekatutakoakXMLzItzuli ()

5.6 Atalen arteko komunikazioa

Orain arte aipatu diren atal ezberdinen artean komunikazio bat ezarri beharra zegoen. Modulu bakoitzak modulu independente gisa jokatzeko du, baina denak bateratu egin dira arbel digitalean. Hori lortzeko php eta javascript barneratzen dituen AJAX teknika erabili da. Arbelaren web orri nagusitik beste atalei deitu egiten zaie modu asinkronoan, orri nagusia berriro kargatu behar izan gabe.

Adibidez, bertsoa idaztean lerro bakoitzean silaba kopuru zehatz bat idatzi beharra dago, eta hori kontrolatzeko AJAX teknika erabiltzen da. Ondorengo lerroetan silaba-kontatzaileari nola deitzen zaion azalduko da.

Lehenik eta behin, bertsoa idazteko lerroetan gertaera berezi bat dagoenean Javascript programa bati deituko zaio. Gertaera hau `onKeyPress` gertaera da, tekla bat sakatzeko guk zehaztutako Javascript kodea exekutatu da, hain zuzen ere.

Beraz, bertsoaren letra bat idatziz gero, zerbitzarian dagoen silaba-kontatzaileari deituko zaio, php-z idatzitako moduluari. Honek beharrezko kalkuluak egin eta emaitza, silaba kopurua, XML dokumentu gisa itzuliko du. XML dokumentu honen egitura hemen behean dago. Bakarrik aipatu, hau *silabaKontatzailea.php* scriptaren emaitza dela, eta honi parametro gisa bi balio eman zaizkiola, esaldia bera (kasu honetan “ta lehenengo urtian”) eta lerro zenbakia, behean ageri dena:

```
<erantzuna>
  <silabaKopuruaMin>6</silabaKopuruaMin>
  <silabaKopuruaMax>8</silabaKopuruaMax>
  <lerroa>2</lerroa>
</erantzuna>
```

Ondoren, Javascript programarekin, XML emaitza horren DOM egitura aztertu eta web-orrialdean agertzen den silaba kopurua eguneratu egingo da datu berriarekin.

Hau azalduka, argi dago AJAX teknikarekin moduluen arteko komunikazioa oso modu garbian ematen dela, batez ere XML erabiltzen delako. Beste moduluek modu berdintsuan egiten dute lan, eta emaitzaren erroa berdina da, `<erantzuna>`, alegia.

5.7 Bertso-abeslaria

Zer izango litzateke bertsolaritza bertso abesturik gabe? Hutsaren hurrengo. Horretarako, eta bertsoBot-aren perspektiba ahaztu gabe, proiektua garatu bitartean ahots sintetizatzaileen mundura salto egin genuen. Diseinuko atalean azaltzen da zergatik erabili den aholab-eko softwarea, baina hasieran festival erabili genuenez, harekin erdietsitakoa azalduko dugu lehenengo azpiatalean. Bigarreanean, berriz, Aholab taldearekin elkarlanean egindako lana.

5.7.1 Festival

Hasiera batean, *CSLU Toolkit* erraminta-kutxa erabiltzen hasi ginen. Honen barruko *Rapid Application Developer* aplikazioak, aukera eskaintzen zuen *festival* ahots sintetizadorearen *pitch*-a (hitz egiterakoan ateratzen den nota) aldatzeko. Beraz, hasiera batean, *RAD* aplikazioarekin, bertso bat simulatzen zuen automata bat sortu genuen. Automataren egoera bakoitza silaba bat zen, eta silaba bakoitzak frekuentzia edo nota bat zuen. Honek bertsoaren antzeko zerbeit sortzen zuela esan daiteke.

Aurrerago, *CSLU Toolkit*-en azpian *Festival* sintetizatzailea zegoela jakinda, aplikazio hau ikertzeari ekin genion. Festival ahots sintetizatzaileak exekuzio modu berezi bat dauka, *singing mode* deritzona. Modu horrekin, diseinuko atalean azaldu den bezala, aplikazioak XML fitxategi bat jaso eta bertsoa abestu egiten zuen. XML fitxategi horrek egitura berezi bat izan beharra dauka. Ondorengo lerroetan ageri da adibide bat:

```
<?xml version="1.0"?>
<!DOCTYPE SINGING PUBLIC "-//SINGING//DTD SINGING mark up//EN"
"Singing.v0.1.dtd"
>
<SINGING BPM="45">
  <PITCH NOTE="G3"><DURATION BEATS="0.15">Mi</DURATION></PITCH>
  <PITCH NOTE="G3"><DURATION BEATS="0.3">la</DURATION></PITCH>
  <PITCH NOTE="E4"><DURATION BEATS="0.15">be</DURATION></PITCH>
  <PITCH NOTE="E4"><DURATION BEATS="0.3">de</DURATION></PITCH>
  <PITCH NOTE="E4"><DURATION BEATS="0.15">ra</DURATION></PITCH>
  <PITCH NOTE="E4"><DURATION BEATS="0.3">tzi</DURATION></PITCH>
  <PITCH NOTE="D4"><DURATION BEATS="0.3">ehun</DURATION></PITCH>

  <PITCH NOTE="E4"><DURATION BEATS="0.15">ta</DURATION></PITCH>
```

```

<PITCH NOTE="F4"><DURATION BEATS="0.3">le</DURATION></PITCH>
<PITCH NOTE="F4"><DURATION BEATS="0.15">nen</DURATION></PITCH>
<PITCH NOTE="F4"><DURATION BEATS="0.3">gour</DURATION></PITCH>
<PITCH NOTE="G4"><DURATION BEATS="0.15">ti</DURATION></PITCH>
<PITCH NOTE="E4"><DURATION BEATS="0.6">an</DURATION></PITCH>
</SINGING>

```

Fitxategi horren egitura oso sinplea da, erroa **SINGING** etiketa da eta horren barruan agertuko da bertsoa. Garrantzitsua da haren atributua den **BPM** zer den azaltzea. Ingelerazko *Beats Per Minute* esan nahi du, erritmoa adierazteko erabiltzen da. Erro horren ume bakoitza bertsoaren silaba bat izango da. **Pitch** elementua izango da ume bakoitza eta hor adieraziko da silabari zein nota dagokion, letra-notazioa erabilia **NOTE** atributuari balio bat emanda. Silabaren iraupena adierazteko, **BEATS** eremuan jarriko da notaren iraupena, **BPM**-rekiko proportzionala. Azken hori **DURATION** elementuaren atributua da. **DURATION** etiketen artean nota bakoitzari dagokion silaba agertuko da.

5.7.2 Aholab

Lehenik eta behin, aholab ikerkuntza taldearen ahoTTS programa erabili genuen. Programa horri testu bat emanda, audio fitxategi bat itzultzen du wav formatuan, testuko edukia ahotsaren bidez emana. Honetaz aparte, beste fitxategi batzuk ere itzultzen ditu.

Proiektu honetan erabili ahal izateko, ordea, programa horri aldaketa bat egin zaio. AhoTTS-k testu bat irakurtzean, silaba bakoitzari iraupen bat esleitzen dio, irakurketa-moduaren arabera (sintetizatzailearen tristura edo alaitasunaren arabera, defini daitekeena). Gure kasuan, ordea, iraupena zehaztuta dago, doinuaren arabera. Guk egindako aldaeran, irakurri beharreko testua (bertsoa) irakurri eta silabaz silaba analizatzen da hori. Silabak, berriz, fonemetan banatzen dira. Beraz, normala den bezala fonema bakoitzari silabari gehitu beharreko denboraren zati bat gehitzen zaio. Beheko algoritmoan ikusten den bezala, une honetan, silaba berdineko fonema guztiak luzatu edo txikitu behar direnean, kopuru berean aldatzen dira.

Adibidez, suposa dezagun, “ba” silabaren iraupena luzatu behar dugula. ba silabak bi fonema ditu, 'b' letrari dagokiona eta 'a' letrari dagokiona. Une honetan, goian aipatu den bezala, 'b'-ren fonemari eta 'a'-ren fonemari denbora berdina gehituko die. Hobe litzateke, ordea, denbora luzatu behar denean, bokalak gehixeago luzatzea eta kontsonanteak gutxiago. Hori, baina, etorkizuneko lan gisa utziko da.

```

for (silaba = lehenSilaba(); silabaDagoenBitartean(); silaba = hurrengoSilaba())
    //bertsoarenIraupena = Berez, doinuaren erritmoak markatutako iraupena
    //silabaIraupena = ahoTTS-k silabari emandako iraupena
    batzekoa = bertsoarenIraupena / silabaIraupena

    for (fonema = LehenFonema();
        fonemaDagoenBitartean();
        fonema=hurrengoFonema())
        iraupena = fonema.iraupenaLortu
        balioBerria = iraupena * batzekoa
        fonema.iraupenaEzarri(balioBerria)
    endFor
endFor

```

Beraz, programaren moldaera berriarekin, bertsoaren erritmoa iada ahotsean nabaritu egingo da.

Komenigarria da azaltzea zein irteera sortzen dituen ahoTTS programak, hurrengo pausoak ulertzeko:

- out.dur: Fitxategi honek hainbat datu ditu bere barruan, fonema bakoitzari buruzko informazioa, silabei buruzko informazioa, haien iraupenak, eta abar.
- out.lf0: Fitxategi hau bitarra da, lerro bakoitzean zenbaki erreal bat dago. Zenbaki hauek frekuentziak dira, out.wav fitxategian dagoen ahotsaren frekuentzia hain zuzen ere, eta segundoko berrehun lagin daude. Fitxategi honen edukiari *pitch* kurba deitzen zaio.
- out.wav: Hau sintetizatzailearen ahotsaren irteera fitxategia da.

Ditugun fitxategi hauek aldatuta, adibidez, out.lf0 fitxategian frekuentziak (*pitch* kurba) aldatuta, gero audio fitxategia birkodetu egin daiteke eta aldaketa hori nabaritu egiten da.

Beraz, argi dago zer egitea falta den; *pitch* kurba dagoen tokian bertsoaren doinuari dagokion frekuentzia sekuentzia jarri eta kitto. Baina frekuentziak ezartzean beste arazo bat dago: kontsonante batzuen frekuentzia. Kontsonante batzuk ahoskatzen direnean (adibidez, K letra), ez du nota berezi bat haren ahoskerak, hari gehitzen zaion bokalak baizik. Hori out.lf0 fitxategian adierazteko zenbaki negatibo oso txiki bat erabiltzen da eta horiek ez dira aldatu behar. Horrekin batera,

bada beste arazo bat, nola jakin zein momentu zehatzetan ahoskatzen den silaba bakoitza? Eta abesten den bitartean egiten diren isiluneak? Horretarako `out.dur` fitxategia erabiltzen da, izan ere, bertan dago fonema (isiluneak barne) eta silaba guztiei eta haien iraupenari buruzko informazioa, beste hainbat daturen artean.

Hori dena egin ondoren, Aholab taldeko Daniel Errok sortutako `ahocoder`[13] programarekin, fitxategi eraldatuak berriro kodetzen dira, eta ordenagailuak bertsoa abestu egiten du.

5.8 Integrazioa interfazeen

Erabiltzailearentzat interfazea egiteak aurreko ataletan azaldu diren elementu guztiak, beren komunikazio egiturekin, leiho batean bateratzea esan nahi du.

Behin interfazearen diseinua erabakita, hasiera batean HTML soilez eraiki zen web orrialdea eredu gisa balio zezan, baina aurrerago, hori dena *php* eta *Javascript* lengoaiekin osatu eta arbelaren atalak txertatu zitzaizkion. 3.3 eta 3.4 irudietan ikusten da eboluzioa.

Beraz, lan guztiaren fruituak honako elementuak dira:

- Lerro kopurua aukeratzeko zerrenda: gure bertsoaren leerro kopurua zehaztu ahal izateko, eta horrela neurri kopurua murrizteko.
- Neurria aukeratzeko zerrenda: hemen Xenpelar dokumentazio zentroko datu-basean dauden neurriak agertuko dira, eta aurreko zerrendan leerro kopuru bat aukeratu bada, harekin bat datozenak.
- Sorta aukeratzeko input-a: hemen sorta zenbakia idatziko dugu gure bertsoa sorta jakin baten barruan idatzi nahi badugu, ezer idazten ez badugu defektuzko sorta batean gordeko da, gorde nahi bada.
- Doinua aukeratzeko zerrenda, neurriaren arabera: kasu honetan, aurretik agertu den neurrien zerrendan aukeraketa bat egin ondoren, harekin bat datozen doinuak erakutsiko dira.
- Laguntza eremua: hemen bertsoaren teoriari buruzko esteka bat, esaera zaharren zerrenda bat eta esteka interesgarri gehiago egongo dira.
- Gaia idazteko eremua: bertso bat gai bati buruzkoa egin nahi bada, eremu honetan idatziko dugu gure gaia. Gai bat sartzerakoan, sorta berri bat sortuko da uneko erabiltzailearen jabetzapean.
- Bertsoa edo ideia ikusteko botoia: bati sakatuz gero, bestea ezkutatu egingo da.
- Bertsoa idazteko eremua: behin neurria aukeratuta, neurri horren arauak arbelean erakutsiko dira, bertsoa idazteko eremuan.

- Silaba-kontatzailea: bertsoa idatzi bitartean, silaba kontatzaileak erabiltzaileak idatzitako lerroaren silabak kontatuko ditu eta motz, ondo edo luze dabilen esango. Gainera, gorriz edo berdez markatuko da zuzenketa erakusten duen tokia, erabiltzaileak silaba kopuru ez egokia sartuko balu. Hasieran, ulertzekoa den bezala, lerro guztietan motz jarriko du, gorriz, ezer idatzita ez egotean, silaba kopurua beharrezkoa baino txikiagoa izango delako.
- Bertso-zuzentzailea: honek bere barnean ditu errima-egiaztatzailea eta silaba-kontatzailea. Honek bertsoaren zuzentasuna egiaztatzen du arlo teknikoan eta emaitza Javascript lengoaiak ahalbidetzen duen alert leiho batean adierazten da.
- Erreminta-kutxa: hau eskubiko zutabearen kokatuta egongo da eta bi elementu izango ditu, errima-bilatzailea eta sinonimo-bilatzailea. Errima-bilatzaileak bi aukera izango ditu, errimak hitzarekin bilatzeko (defektuzkoa) edo bukaera bakarrik idatzita (kasu honetan ez ditu lege fonetikoak aplikatzen). Sinonimo bilatzailearen kasuan, hitz bat idatzi eta haren sinonimoak bilatzen ditu. Momenturen batean erabiltzailea zerbait adierazteko hitzik gabe geratzen bada, hor dauka hitzak bilatzeko aukera.
- Neurria zehazteko sistema: bertso baten neurria zein den ez dakigunean erabilgarri suerta dakiguen sistema.

5.3 irudian arbela bera agertzen da, baina goian azaldutako elementu guztiak non dauden adierazita.

Erabilerari dagokionez, esan beharra dago oso erraz erabiltzen dela. Lehenik eta behin idatzi nahi den bertsoaren neurria aukeratu behar da. Adibidez, zortziko txikia.

Behin neurria aukeratuta 5.3 irudian agertzen den bezala, bertsoa idazteko lerroak agertuko dira, baina baita neurri horren arauak errespetatzen dituzten doinuak ere. Nahi bada, bertsoa idatzi aurretik, bitartean edo ondoren neurri hauek entzun egin daitezke haien izenen ondoan dagoen “Play” botoia sakatuz. Doinuen gainean klik eginez gero, neurria aukeratuta gelditzen da, eta idatzitako bertsoa gordez gero, doinu horrekin gordeko da. Berriro klik egiten bazaio, aukeraketa desegin egingo da. Doinuaren partitura ikusi nahi izanez gero, partitura agertzen den ikonoan klik egitea besterik ez dago. Bi horien tartean, badago beste ikono bat, bideo batena, hain zuzen ere. Honek doinu horretan abestutako bertso baten bideoaren esteka izango du, datu-basean informazio hori baldin badago, behintzat.

Bertsoa idazten goazen heinean, lerroen ondoan dauden testuak eguneratzen joango dira, hau da, silabak gehitu ahala beharrezko silaba kopuruetara iritsitakoan “motz” jartzetik “ondo” jartzera pasako da. Hori bai, lerro berean silaba gehiegi idatziz gero, “ondo” jartzetik “luze” jartzera pasako da. 5.4 irudian ikus daiteke adibide bat.

Hau dena azaldu eta gero, bertsoa idatzi ondoren erabil daitekeen zuzentzailearen funtzionamendua erakutsiko da. Arbelean bertso hau sarturik, esaterako:

Neurria ondo jarri
edo bestela luze
motz dagonean
hola sudurluze
errimak ozta-ozta
segi ordun hola
ez dezu irabaziko
inongo peluxe

Emaitza oso argia da (ikusi 5.5 irudia), bigarren eta hirugarren lerroetan silaba kopurua gaizki dago. Seigarren lerroan, berriz, errima ez da ona. Beraz, bertsolariak hori hobetu beharra dauka. Baina nola aurkitu “peluxe” hitzarekin errimatzen duten hitzak? Errima-bilatzailea horretarako dago! “Peluxe” hitza aukeratuko dugu eta automatikoki errima-bilatzailean eta sinonimo-bilatzailean txertatuko da. Orduan bilatu botoiari sakatu eta emaitza bilatzen dute. “Peluxe” hitzaren kasuan sinonimo-bilatzaileak ez du emaitzarik, baina bai errimategiak. Errimak ikusi ahal izateko errimak erakutsi jartzen duen koadro gorri bat dago. Hor gainetik sagua pasa eta hitz zerrenda ikusgai izango dugu 5.6 irudian ikus daitekeen bezala.

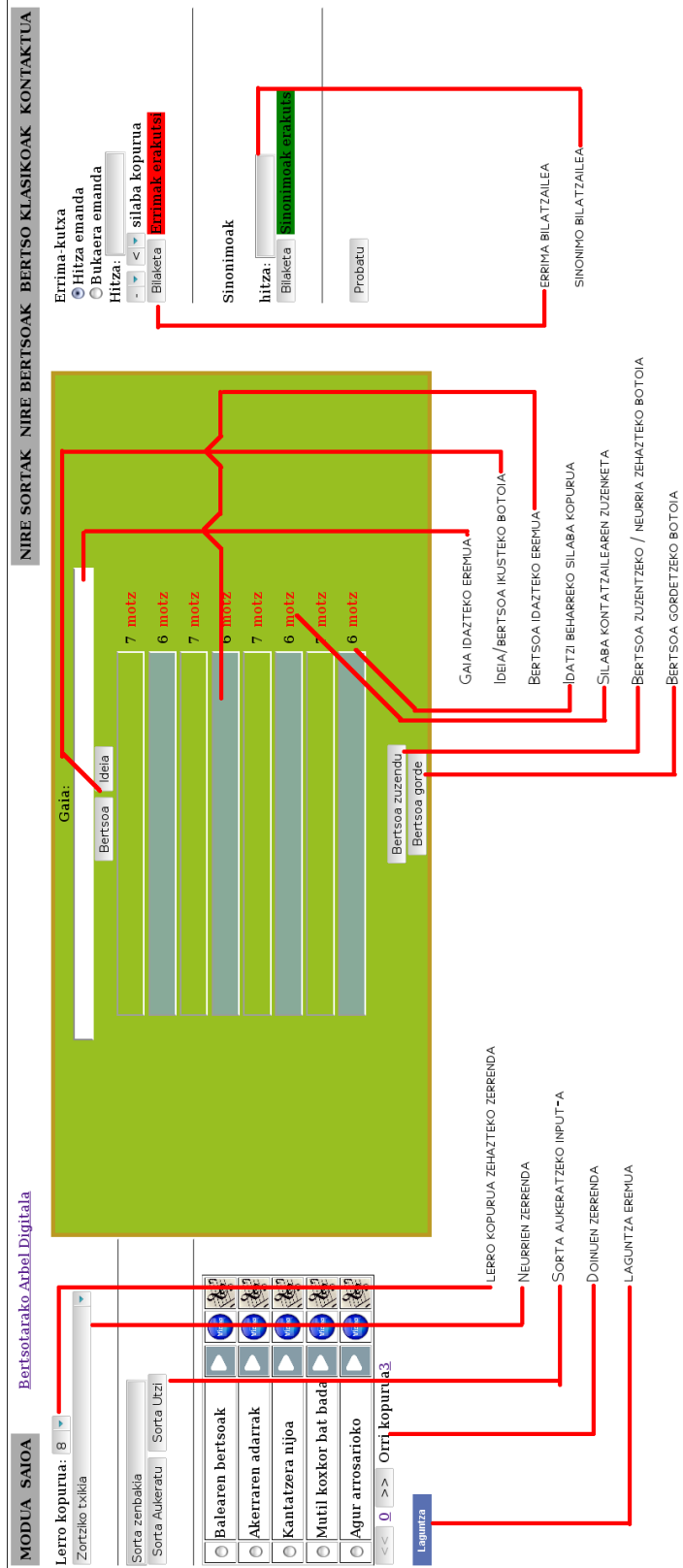
Hitz perfektu bat aurkitu eta bertsoa pixkat moldatu da, ea zein erantzun ematen duen hurrengo zuzenketan. “Segi ordun hola“ jartzen duen lekuan ”segi ordun luze“ jarri da, peluxerekin errimatzen duen hitz bat (errima bilatzailearen erantzuna).

Beraz, zuzentzaileari berriro zuzentzeko eskatu eta errore bat eman du, 2. eta 6. lerroetan dago errorea, izan ere, bietan ”luze“ hitza dago jarrita, eta hau bertsotan onartezina da. Baina, pentsa batzuetan hitz batek adiera bat baino gehiago ditueanean, ez dela pototzat hartzen. Beraz, abisu bat emango zaio erabiltzaileari, ”Ez al duzu pototo egin 2. eta 6. lerroetan? Erabiltzaileak pototo egin duela ziurtatzeko hitzak adiera bakarra duela frogatu beharko litzateke, eta hori etorkizunean garatzeko lan bezala geratuko da.

Behin arazoa konponduta, eta bertsoa ondo dagoenean, bertsoa gordetzeko kautotu egin behar da. Hortarako, goiko barran dagoen "saioa" botoia sakatu behar da eta haren azpian agertzen den izena eta pasahitza eremuetan gure datuak jarri (suposatzen da erregistratua egongo dela erabiltzailea). Behin kautotutakoan, datuak ondo sartu baditugu agurra irakurri ahal izango dugu. Gero, "saioa" botoia berriro sakatuta, login eremua desagertu egingo da eta gure izenean lanean jarraitu dezakegu.

Momentu horretan, "bertsoa gorde" botoiari sakatu eta bertsoa gordeta geratuko da, irakurgarri izango dugularik goiko barran agertzen den "nire bertsoak" atalean. Bertsoa, sorta batean gordeko bagenu, "nire sortak" atalean bertso sorta osoa irakurzeko aukera genuke. Gainera, nire sortak atalean, sortak kudeatu daitezke, hala nola, sortari pasahitza ezarri (etorkizuneko lana) edo sorta ezabatu. Etorkizunean, bertsoa ezabatzeko aukera ere eskainiko da.

Azkenik, aipatu gabe geratu den azken aukera. "modua" jartzen duena, hain zuzen ere. Erabiltzaileak bertsolaritzan esperientzia lortzen duenean, zuzenketarik gabe bertsoa idatzi ahal izateko sistema. Hemen silaba-kontatzailea ez da martxan jartzen, beraz, ez du jartzen ea ondo edo gaizki dagoen lerroa. Baina nahiz eta modu librean aritu, "bertsoa zuzendu" aukera eskura dago.



Irudia 5.3: Arbela zortziko txiki bat jasotzeko prest

MODUA SAIOA Bertsolarako Arbel Digitala NIRE SORTAK NIRE BERTSOAK BERTSO KLASIKOAK KONTAKTUA

Lerro kopurua: 8 Zortziko txikia

Sorta zerbakia
Sorta Aukeratu Sorta Utzi

Balearen bertsoak
Akerraren adarrak
Kantatzera nijoia
Mutil koxkor bat bada
Agur arrosarioko

<< 0 >> Orri kopurua 3

Laguntza

Gaia:
Bertsoa Idela

Neurria ondo jarri 7 ondo
edo bestela luze 6 luze
motz dagoenean 7 motz
hola sudurluze 6 ondo

Bertsoa zuzendu
Bertsoa gorde

Erima-kutxa
Hitza emanda
Bukaera emanda
Hitza:
silaba kopurua
Bilaketa: **erimok erakutsi**

Sinonimoak
hitza:
Bilaketa: **sinonimoak erakutsi**

Probatu

Irudia 5.4: Silaba-kontatzailearen marka posibleak: motz, ondo eta luze

The screenshot shows the 'Bertsotarako Arbel Digitala' web application. The main content area displays a list of bertsos (songs) with their lyrics and syllable counts. A modal dialog box is open in the center, displaying error messages in Basque: 'ERRORREA: 2. lerroan silaba kopurua ez da egokia.', 'ERRORREA: 3. lerroan silaba kopurua ez da egokia.', and 'ERRORREA: 6. lerroan errima ez da egokia.' The interface includes navigation menus at the top, a search bar, and various filters on the left and right sides.

Lyrics	Syllable Count
Neurria ondo jarri	7 ondo
edo bestela luze	6 luze
motz dagonean	7 motz
hola sudurluze	6 ondo
errimak ozta ozta	7 ondo
segi ordun hola	6 ondo
ez dezu irabaziko	7 ondo
inongo peluxe	6 ondo

Irudia 5.5: Bertsorearen zuzenketa

MODUA SAIOA Bertsotarako Arbel Digitala NIRE SORTAK NIRE BERTSOAK BERTSO KLASIKOAK KONTAKTUA

Lerro kopurua: 8
Zortziko bakia

Sorta zenbakia
Sorta Aukeratu Sorta Utzi

Balearen bertsoak
Akerraren adarrak
Kantatzera nijoia
Mutil koxkor bat bada
Agur arrosarioko

Orri kopurua: 3
Logondu

Gaia: Bertsos Idela

Neurria ondo jarri	7 ondo
edo hala luze	6 ondo
motz dagonean berriz	7 ondo
hola sudurluze	6 ondo
errimak ozta ozta	7 ondo
segi ordun luze	6 ondo
ez dezu irabaziko	7 ondo
inongo peluxe	6 ondo

Bertsoa zuzendu
Bertsoa gorde

Errima-kutxa
Hitza emanda
Bukaera emanda
Hitza: peluxe
silaba kopurua

Bilaketa: Errimak erakutsi

luze	berpeltze	lendatze
pegatze	erritaze	antitaze
filatze	tolokaze	begitaze
hautze	peluxe	vauletaze
estikatze	kuntaze	erretaze
hantakaze	behatze	lotaze
subutaze	luzaze	geritaze

FRUOSU

Irudia 5.6: Errima-bilatzailearen erantzuna

5.9 Probak

Proiektua burutu bitartean, helburu dokumentuan azaldu den bezala, proba batzuk egin ziren arbelarekin gertuko ingurune batean. Ibaetako unibertsitateko bertso-eskolan, *unibertsoskolan*, egin ziren proba hauek.

Garapenaren zati handi bat egin zenean, Zumarraga eta Urretxuko bertso-eskolara egin zen jauzi eta bertan egin ziren azkeneko probak.

Bi kasuetan, oso emaitza positiboak lortu dira, eta pozgarria izan da honek izan duen harrera ona. Hobekuntza batzuk gehitu zaizkio arbelari proba hauen ondorioz. Adibidez, kontsonante bat maiuskulaz idazterakoan, kontsonantea irakurri egiten da. Bertsoan L letra jarritz gero, silaba kontatzaileak “l” letra bakarrik kontatu beharrean bi silabako unitatea, “le”, kontsideratuko du.

Hurrengo helburua, ahalik eta azkarren bertsozale elkartearen zerbitzari batean lana instalatu eta bertso-eskola gehiagotan probak egitea izango litzateke. Gainera, iradokizunentzako posta elektronikoko helbide bat ahalbidetu da, inork ezer beharko balu.

Pasadizo gisa, aipatu, Zumarraga eta Urretxuko bertso-eskolan, bazela mutil bat, bertsoak beti zortziko txikian idazten zituena. Arbel digitalaren laguntzaz, hamarreko handira salto egin zuen.

Kapitulua 6

Ondorioak eta etorkizuneko lana

6.1 Ondorioak

Orokorrean, arbel digitala garatu denez, **helburuak bete dira**. Proiektuaren helburu nagusia hizkuntza-teknologiak aztertu eta bertsolaritzan aplikatzea zen. Interfaze lagungarri bat sortu da baliabide hauek erabili ahal izateko. Arbelari honako ezaugarriak jarri zaizkio:

- Errima-bilatzailea
- Errima-egiaztatzailea
- Silaba-kontatzailea
- Bertso-zuzentzailea
- Sinonimo-bilatzailea
- Neurri-esleitzzailea
- Bertsoak gordetzeko aukera

Hasieran, proiektuaren zehaztapena egin zenean, bertsoak indexatzea aurreikusten zen, gero horrekin antzeko bertsoak bilatu ahal izateko sistema bat egiteko. Horretaz aparte, doinu-gomendatzaile bat ere egin nahi zen, baina, denbora falta dela eta, bi lan hauek etorkizuneko lan gisa utzi behar izan dira.

Baina, aipatu beharra dago, hasieran aurreikusten ez ziren lan batzuk ere egin direla, hala nola, bertsolari-abeslaria garatzea¹ eta arbelaren eskuliburua idaztea.

¹Hemendik gutxira web-interfazean erabilgarri egongo da ezaugarri hau

Ondorengo taulan lan-paketeei eskaini beharreko denbora ageri da: bigarren zutabea, estimatutakoa eta hirugarrenean, errealak.

	Estimatutako orduak	Ordu errealak
Atazak		
Informazio bilketa	47	35
Bertsolaritzaren datu-basea landu	10	8
BertsolarIXA analizatu	15	3
Silaba-kontatzailearen algoritmoa aztertu	4	4
Indexatzeko teknikak ikasi	8	0
Programazio-lengoiak ikasi	10	20
Analisia	9	9
Sistemaren analisia egin	9	9
Diseinua	27	54
Diseinu orokorra egin	27	54
Inplementazioa	200	198
Web-interfazea implementatu	10	15
Datu-basea implementatu	5	3
Silaba-kontatzailea hobetu	15	5
Errima bilatzailea hobetu / eraiki	20	20
Barne programazioa garatu	150	120
Bertsolari-ablesaria garatu	0	35
Proiektuaren kudeaketa	57	78
Bileren antolakuntza	8	13
Proiektuaren kontrola	4	5
Eskuliburua idaztea	0	5
Aurkezpenen prestaketa	10	10
Memoriaren idazketa eta zuzenketa	35	45
BATURA	340	374

Hurrengo lerroetan taulako eremu bakoitza aztertuko da, eta eskaini zaion denbora justifikatuko da.

- Bertsolaritzaren datu-basea landu: Hasieran 10 ordu eskaini behar zitzaizkion eta 8 eskaini zaizkionez, espero zenaren antzekoa gertatu da. Berez, doinuen

datu-basea ondo aztertzea aurreikusten zen, doinu-gomendatzailea garatzeko, baina hori ez da egin.

- BertsolarIXA analizatu: Hasieran errima-bilatzaile automatiko hau erabiltzea aurreikusten zenez, honen analisirako denbora asko beharko zela aurreikusi genuen, baina azkenean, hau erabli ez dugunez, asko murriztu da honen iraupena.
- Silaba-kontatzailearen algoritmoa azertu: Aurreikuspenak bete dira.
- Indexatzeko teknikei buruz ikasi: Hasieran bertsoak indexatzea eta antzeko bertsoak bilatzea pentsatzen bazen ere, hau garatzeko ez da denborarik izan, beraz, denborarekin ikus daitekeen bezala, garatu gabe gelditu da hau.
- Programazio lengoaiak ikasi: Hasieran esperotakoa baina denbora gehiago behar izan dugu lengoaiak ikasten, uste genuena baino lengoiaia gehiago erabili ditugulako.
- Sistemaren analisia egin: Aurreikuspenak bete dira.
- Diseinu orokorra egin: Programazioa garatu bitartean diseinuaren egiturak aldaketak izan dituzenez, uste baino denbora gehiago behar izan dugu diseinu orokorrerako.
- Web-interfazea inplementatu: Espero baino zerbait gehiago behar izan da honetarako, baina ezberdintasun onargarria izan da.
- Datu-basea inplementatu: Aurreikuspenak baino denbora pixkat gutxiago behar izan da.
- Silaba-kontatzailea hobetu: Espero baino denbora gutxiago behar izan da. Silaba-kontatzailearekin igarotako denbora gehiena, php-ra itzulpena egiten eta zuzenketak egiten pasa da.
- Errima-bilatzailea hobetu / eraiki: Aurreikuspenak bete dira.
- Barne programazioa garatu: Espero baino denbora gutxiago behar izan da, barne programazio orokorrera bideratuta zegoen lana beste lan-pakete batzuen barruan garatu delako.
- Bileren antolakuntza: Estimaturakoa baino denbora gehiago behar izan da bileretarako, hala ere, ez da diferentzia handia.
- Proiektuaren kontrola: Aurreikuspenak bete dira.

- Aurkezpenen prestaketa: Aurreikuspenak bete dira.
- Memoria egitea: Espero baino denbora gehiago behar izan da.
- Bertsola-abelaria garatu: Hasieran hau egitea espero ez zenez, denbora aurreikuspenik ez zen egin.

Taula horretan argi ikusten da, nolabait, indexatzeari zegokion lan-paketea desagertu egin dela, baina beste batzuk sortu direla, adibidez, eskuliburua idaztearena eta bertsolari abeslaria garatzearena. Ordenagailua bertotan jartzearen kontua urruneko kontu bezala ikusten genuen, baina azkenean, tarte bat egin ahal izan dugu proiektu honetan.

6.1 irudian, eraiki den erabilpen-kasu diagrama erreala agertzen da. 3.3 irudian estimatutako erabilpen-kasu diagrama dago.



Irudia 6.1: Sortu den arbelaren erabilpen-kasu diagrama

Proiektuak izan zitezkeen arriskuetatik bakarra bete da. Hau kurtsoan zehar garatu denez, beste irakasgaiei denbora pixkat eskaini diet, beraz, proiektua ekainean aurkeztu beharrean, beranduago aurkeztuko dugu.

6.2 Etorkizuneko lana

Atal honetan etorkizunean egin daitezkeen hobekuntzak azaltzen dira, hiru zatitan banatuta, epe laburrean egitekoak, epe ertainekoak eta epe luzekoak:

6.2.1 Epe laburrean egitekoak

- “Sortak kudeatu“ atala osatu. Sortentzat pasahitza ezartzeko metodoa garatu.
- Bertsoak kudeatzeko atala sortu.
- Diseinu bisuala hobetu, batez ere, sortak eta bertsoak irakurtzeko eremuetan.
- Neurria zehaztu gabeko bertsoetan, neurria esleitu eta neurri zehaztuaren interfazera pasa. Horrela, gainera, bertsoaren errimak zuzentzeko aukera egongo da.

6.2.2 Epe ertainean egitekoak

- Sinonimo-bilatzailean edozein hitz bilatzerakoan hitz horren lemaren sinonimoak bilatu ahal izateko lematizatzailea gehitzea.
- Errima-bilatzailea fintzea. Izan ere, oraindik hitz batzuekin ez da errimarik lortzen. Gainera, kategorizazio bat egitea komeni da, eta gai baten inguruko errimak bilatzeko gaitasuna izatea ere bai. Errimak sailkatu egin behar dira, oinarriko hitzarekin duten antzekotasunaren arabera. Horretaz aparte, emaitzako hitzak, atzizkiak dituzten ala ez kontuan izanda, nolabait markatuko dira. Gainera, errimategiko hitzei klik eginda haien definizioa agertzea ez litzateke batere gaizki etorriko.
- Une honetan PHP lengoaian garatua dagoen errima-egiaztatzailearen erregelak fintzea, bertsozale elkartearekin batera, eta gero FOMA erabiliz garatzea.
- Bertsoak abesteko zerbitzua hobetu. Adibidez, bertsoak neurri zehatz baten egitura zorrozki betetzen ez duenean, nolabaiteko elkarketak egiteko aukera eman liteke. Hobetu ondoren, interesgarria litzateke hau web-zerbitzu gisa eskaini ahal izatea, eta, horrela, arbelaren aukera bat gehiago bezala eskaintzea (beti ere, aholab taldearen meritua aitortuta).
- Bertsoan egindako potoak identifikatzeko sistema, poto posibleen adierak analizatuaz, wordNet erabilia.

6.2.3 Epe luzean egitekoak

- Neurri-esleitzaila hobetzea. Hainbat neurriren arauak betetzen dituztenean, eredu probabilitistikoak erabilia neurri baten aldeko apustua eginez, adibidez. Gainera, interesgarria litzateke automatikoki esleitzea neurria erabiltzaileak eskatu behar izan gabe.
- Antzeko bertsoak bilatzea idatzitako bertsoaren egitura, doinua, erlazio lexiko-semanticokoak, etab. analizatuz. Gainera, bertsoak indexatu egin beharko lirarteke.
- Doinu-esleitzaila garatzea, bertso hitzak eta neurria aztertuz.
- Bertso-sorkuntzaren inguruan ikerketa sakon bat egitea, eta horrela, automatikoki bertsoak sortzeko gai izango den robot bertsolari automatikoa, bertso-BOT delakoa, garatu.
- Orain arte garatu den arbela hizkuntza-anitz bilakatzea, eta hainbat modulu aldatuta, edozein hizkuntzarako plataforma baliagarria izatea.

Bibliografia

- [1] A. Astigarraga, K. Sarasola, K. Gojenola, A. Soroa, “Testu analisirako perl erramintak”, UEU, 2009
- [2] Larry Wall, Tom Christiansen, Jon Orwant, “Programming Perl”, O’Reilly Media, 2000
- [3] Robin Nixon, “Learning PHP, MySQL and JavaScript”, O’Reilly Media, 2009
- [4] Darie C., Brinzarea B., Chereches-Tosa F., Bucica M., “AJAX and PHP: Building Responsive Web Applications”. Packt Publishing Ltd. 2006.
- [5] Javier Eguiluz, “Introducción a AJAX”, Creative Commons, 2008
- [6] phpMyAdmin dokumentazioa, “<http://www.phpmyadmin.net/documentation/>”
- [7] I. Alegria, X. Arregi, B. Arrieta, “Bertsolaritzarako errima-sortzaile informatikoa”, Elhuyar Aldizkaria, 2001
- [8] O. Odriozola, “Euskararen atzetik aurrerako sortzaile morfologikoaren aberaske- ta bertso-ingurune batean”, Karrera bukaerako proiektua, 2001
- [9] E. Agirre, I. Alegria, X. Arregi, X. Artola, A. Diaz de Ilarraza, M. Maritxalar, K. Sarasola, M. Urkia, “Xuxen: A Spelling Checker/Corrector for Basque based in Two-Level Morphology”, IXA, 1992
- [10] B. Arrieta, I. Alegria, X. Arregi, “An assistant tool for Verse-Making in Basque based on Two-Level Morphology”, 2001
- [11] E. Pociello, E. Agirre, I. Aldezabal, “Methodology and construction of the Basque WordNet”, 2010
- [12] Aholab taldea, “AhoTTS, A Text-To-Speech System”, http://aholab.ehu.es/tts/tts_en.html

-
- [13] D. Erro, I. Sainz, I. Saratxaga, E. Navas, I. Hernáez, “MFCC+F0 Extraction and Waveform Reconstruction using HNM: Preliminary Results in an HMM-based Synthesizer”, 2010
- [14] Alan W Black, Paul Taylor and Richard Caley, “The Festival Speech Synthesis System, System documentation”, 1.4 Edition, 2002
- [15] John-Paul Hosom, “The CSLU Toolkit: A Platform for Research and Development of Spoken-Language Systems”, Center for Spoken Language Understanding, Oregon Health & Science University
- [16] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, “Introducción a la teoría de Autómatas, Lenguajes y Computación”, Pearson educación, 2002
- [17] L. Karttunen, “Finite-State lexicon compiler”, Palo Alto Research Center, Xerox Corporation, 1993
- [18] M. Hulden, “Foma: a finite-state compiler and library”, <http://www.aclweb.org/anthology/E09-2008.pdf>, 2009
- [19] X. Amuriza, “Hitzaren kirol nazionala”, Bizkaiko Bertsozale Elkarte
- [20] J. Dorronsoro, “Bertso doinutegia”, Euskal Herriko Bertsozale Elkarte
- [21] J. Dorronsoro, Xenpelar Dokumentazio Zentroa, “<http://bdb.bertsozale.com>”, Xenpelar Dokumentazio Zentroa, 2011
- [22] Bertsozale elkarteko ikerkuntza taldea, “Bertso-estrofak izendatzeko irizpideak”, Euskal Herriko Bertsozale Elkarte, 2009

Eranskina A

Bilera-aktak

1.bilera

Data: 2010eko Maiatzaren 5a
Partaideak: Bertol Arrieta eta Manex Agirrezabal
Lekua: Bertolen bulegoa

- Proiektuaren zehaztasun batzuk aztertzen dira
- Proiektuaren ibilbidea pixka bat zehaztu
- BertsolarIXA konpondu behar da
- Datu iturria datu basea izango da, eta ez fitxategi soilak
- Interfazea erabilgarri eta erakargarria egin behar da
- Bestelako hobekuntzak:
 - Euskal Wordnet
 - Semantika

2.bilera

Data: 2010eko Uztailaren 20a
Partaideak: Iñaki Alegria eta Manex Agirrezabal
Lekua: Iñakiren bulegoa

- Bertsotan ikasteko arbel digital bat egin nahi da.

- Web bidezko interfazea izan beharko du.
- Osagai nagusiak (lehenengo bertsiorako):
 - Silabak, neurria (Silaba kontatzailea (Perl-ez, Aitzolek egina))
 - Errimak (BertsolarIXA)
 - Antzeko bertsoak Neurrian, erriman eta gaian
- Osagai osagarriak (bertsio hobeturako):
- Semantika lantzea

3.bilera

Data: 2010eko Irailaren 15a
Partaideak: Iñaki Alegria eta Manex Agirrezabal
Lekua: Iñakiren bulegoa

- Proiektuaren oinarritzko helburu eta osagaiak finkatzen dira.
 - Silaba kontatzailea.
 - Errima bilatzailea.
 - Doinu hautatzailea.
- Hauetaz aparte proiektuaren osagai osagarriak ere finkatzen dira.
 - Antzeko bertsoen bilatzailea.
 - Bertso irakaslea.
- Bilera dei bat egiten da Bertol eta Aitzolekin, haiek ahal duten orduan, ordua zehazteke oraindik.

4.bilera

Data: 2010eko
Partaideak: Iñaki Alegria, Aitzol Astigarraga eta Manex Agirrezabal
Lekua: Iñakiren bulegoa

- Bertsotarako arbel digitalaren egitura nagusia definitu da
 - Datu Base bat izango dugu, zeinean honako hau gordeko den:
 - * Doinuak
 - * Neurriak
 - * Antzeko bertsoak
 - Doinuei buruz informazio hau gordeko da:
 - * Izena
 - * neurria (Neurriaren izena, identifikadorea)
 - * musika
 - * Partitura
 - Neurriei buruz informazio hau gordeko da:
 - * Lerro kopurua
 - * Izena
 - * errima (abab)
- Modulu ezberdinen sarrera eta irteerak ere definitu dira:
 - errima bilatzailea: sarrera hitza, irteera hitz zerrenda
 - silaba kontatzailea: sarrera esaldia, irteera zenbaki osoa
 - antzeko bertso bilatzailea: Oin bat eta neurri bat (eta doinu bat) emanda, bertso bat itzultzen du
 - Doinu hautatzailea: Bertso bat, eta neurri bat emanda, bertsoaren tematikarekin bat datorren doinua itzuliko duen sistema
- Silaba kontatzaileari bokal elkarketen aukera gehitu behar zaio.

5.bilera

Data: 2010eko Urriaren 21a
Partaideak: Iñaki Alegria, Aitzol Astigarraga eta Manex Agirrezabal
Lekua: Iñakiren bulegoa

- Bertsotarako arbel digitalaren egitura zehaztu da, modu formalean.
 - Interfazeak ezkerrean zutabe bat izango du, zeinean aukeran dauden neurri ezberdinak agertuko diren.

- Neurrietako bat aukeratzerakoan, erdiko zutabearen egitura marraztuko zaio erabiltzaileari, nahiz eta Neurria botoiean klik egin arte ez den hau interfazean txertatuko. Honetaz aparte, neurria aukeratzerakoan, doinu zerrenda bat erakutsi behar da, eta ondoan audioa entzuteko botoitxo bat izango du hauetako bakoitzak.
 - Erabiltzaileari ere aukera emango zaio neurria definitu gabe uzteko, neurri definitugabe bat zehaztuz, kasu horretarako.
 - Behin neurria aukeratutakoan, egitura osoa txertatu behar da interfazean, neurriaren lerro kopuru, silaba kopuru eta errima-lerroekin.
 - Bertsoarentzako textarearen gainean beste testu eremu bat definituko da gaia jartzeko, gero antzeko bertsoak bilatzeko lagungarria izan daitekeelakoan.
 - Eskuin zutabearen errima bilatzailearen interfazetxo bat txertatuko da, zeinarekin errima-gako bat emanda (errimatu behar den lerroko azken bi silabak, beste kolore batez markatuak egongo direnak) harekin errimatzen duten hitzak taula batean lortuko diren. Honi aukera ezberdinak jarriko zaizkio checkbox-ak erabilia, hala nola, lege fonetikoak erabiltzea, morfologia eta testu eremu batekin silaba kopuru minimoa ezartzea, adibidez.
 - Azkenik, errima bilatzaile honen azpian sinonimo bilatzaile bat egongo da.
- Horraino egitura kontuak, baina gehitu daitezkeen beste funtzionalitate batzuk ere aipatu dira, hala nola saio kudeaketa bat, bertso eskoletako (edo kaleko) jendea kautotzeko sistema bat, honek ikasleari bertsoa gordetzeko aukera emanik. Beraz, bi botoi gehitu beharko liriateke interfazera, "nire bertsoak" delakoa, norberak idatzitako bertsoak ikusteko botoi bat, eta gorde botoi bat, bertso bat gordetzeko. Azken bi aukera hauek ikusi ahal izateko kautotu egin behar delarik.
 - Ezkerreko zutabearen laguntza modura hiru esteka haueke ere jarriko dira:
 - Antzeko gaia duten bertsoak, laguntza modura.
 - Esaera zaharrak.
 - Bertsoaren teoria, aitzolek lortuko du dokumenturen bat, oinarritzko gida bat.
 - -Bertso baten errima bektoreari eta silaba bektoreari hash funtzio bat aplikatuko zaio, horrekin neurria esleitzerakoan (behar den kasuan) zerbitzariari lan karga kenduko diogulakoan.

6.bilera

Data: 2011ko Otsailaren 7a

Partaideak: Iñaki Alegria, Aitzol Astigarraga eta Manex Agirrezabal

Lekua: Iñakiren bulegoa

- Egindako lanak ikusi eta ebaluatu dira.
- Silaba kontatzailearen anbiguotasunak ekiditeko sistema onargarria da.
- Datu basearen diseinuan aldaketa pare bat proposatu dira:
 - Sorta bat doinu batekin erlazionatu beharrean, bertsoa neurriarekin elkartuko da
 - Sorta ez da doinu batekin erlazionatuko, ez da mugatzen sorta bateko doinua, neurria ezta (koplarekin galdetu eta koplarekin erantzun)
 - Bi erabiltzaile/egile mota ezberdindu behar dira, bertsolari klasikoa eta ikaslea.
 - Neurriekin ere kategorizazio bat egingo da, zortzikoak, hamarrekoak, koplak.
- Errima egiaztatzailearen kontu batzuk hitz egin dira, baina hurrengo asterako bilera bat egitea erabaki da honetaz bakarrik hitz egiteko (Iñaki eta Manex)

7.bilera

Data: 2011ko Martxoaren 31

Partaideak: Iñaki Alegria eta Manex Agirrezabal

Lekua: Iñakiren bulegoa

- Egindako lanak ikusi eta ebaluatu dira.
- Errima bilatzailean hitzak ordenatzeko sistema bezala, maiztasunen fitxategi bat erabiltzea erabaki dugu. Fitxategi hori perl-ez pixkat bat aldatu, eta php-z atzigarria izango den bektore bat sortuko da, non bektorearen indizea hitza bera den, eta edukia maiztasuna.
- Errima bilatzailean silaba kopuruaren arabera ere hitzak bilatzeko aukera eman

- Sinonimoak euskal Wordnet-en bidez bilatuko dira, APIa lortu behar.
- Festival-eko singing-modea erakutsi diot Iñakiri. Baina arazo bat dauka horrek, gutxi ulertzen zaiola. Beraz, Bilbora joan beharreko egun batean aholab-en lan egiten duten ikerlariekin bilera bat egiten saiatuko gara.
- Bertso sortak pdf-n gordetzeko aukera planteatu, hasieran planteatzen dena, egindako liburutegi bat erabiltzea da, baina Iñakik XSLT-FO erabiltzea gomendatzen dit.
- Iñakik midi-player bat gehitzeko eskatzen dit, doinuen audioak entzuteko.
- Aparte, lerro saltoak txikitzeke ere eskatzen dit.
- Bertso guztiak ikusteko aukera ere kentzeko eskatu dit Iñakik, eta bertso osoak ez agertzeko (googlen antzeko zerbaiteko)
- Goiko barra txikitu egin behar da, garrantzia arbelak izan dezan.
- Probatu sakatzerakoan eta bertsoa datu basetik kargatzerakoan, bertsoa zuzendu, hau da, dena motz-motz-... jarri beharrean, behar den bezala jartzeko

8.bilera

Data: 2011ko Maiatzaren 5a

Partaideak: Aitzol Astigarraga eta Manex Agirrezabal

Lekua: Fakultateko kafetegia

- Egindako lana aurkeztu zaio Aitzoli, eta ebaluazio positibo bat egin zaio orokorrean, baina hobetzeko aspektu batzuk markatu dira
- Arbeleko errorearen artean, erabilitako errima zerrenda ere erakutsi, errima gaizki baldin badago
- Lerroren bat osatu gabe baldin badago, ez dezala potu markatu sistemak. Eta potu eginez gero, beste errore arruntekin batera jarri.
- Letra tamaina handiagoa jarri
- Proposamen e-mail bat jarri
- Gorka Azkaraterekin (Zumarraga-Urretxuko bertso eskolako irakaslea) ordu bat finkatu tresna erakusteko

- Hitzen maiztasunen datuak dauzkan php fixtategia banatu
- Ateratako proposamenak: maiztasun fitxategiak lortzeko, hainbat corpus erabiltzea, arkitektura, botanika, artea, bertsoan erabilitako errimak, ...
- Uztailan bilera, bertsozale elkartearekin

10.bilera

(Nere Erkiagaren akta)

Data: 2011ko Uztailaren 3a

Partaideak: Manex Agirrezabal, Aitzol Astigarraga, Mikel Lersundi (IXA taldea, EHU), Karlos Aizpurua (Bertsozale elkarte, ikerkuntza saila), Nere Erkiaga (Bertsozale elkarte, dokumentazioa eta datu-baseak)

Ezin izan dute etorri: Iñaki Alegria eta Ixiar Eizagirre (Bertsozale elkarte, transmisio saila) Lekua: Subijana Etxea, Villabona

Helburua: Manex Agirrezabalek egindako Arbela Digitala egitasmoa aurkeztu eta Bertsozale Elkartearekin elkarlanerako aukerak aztertzea.

Arbela Digitala egitasmoa, Manex Agirrezabalen EHUko Informatika Fakultateko ikasketa bukaerako lana da, eta hizkuntzen teknologia bertsolaritzan aplikatuaz, bertsoan ikasteko web programa sortzea du helburu.

Ezaugarriak:

- Silaba kontatzailea: Aitzol Astigarragak garatutako aplikazioa erabili da, Perl hizkuntzarekin garatua zegoen eta PHP hizkuntzara bihurtu da.
- Errima bilatzailea: Aukera ezberdinak aztertuta, FOMA tresna hobetsi da hiru arrazoi nagusiengatik: Lehena; abiadura handiagoa du BertsolarIXAk baino. Gainera, errimak bilatzeko bukaera bakarrik idatzi beharrik ez dago. Eta azkenik, silaba kopuru zehatz bat duten hitzak aurkitzeko gaitasuna du. Hobekuntza posiblea: bi hitzen bidez sortutako errimak bilatzea (Adibidez: ikustEN DU, KENDU).

- Errima egiaztatzailea: Aukera ezberdinak aztertuta (FOMA, Amurizaren errimategia,...), PHP bidez garatutako erregela zerrenda simple bat erabili da errimak egiaztatzeko (azken silaba eta aurreko bokalak errimatzea). Silaba kopuruarekin malgutasunez jokatzeko badaki, “h” ahoskatzen denerako, edo sinalefen kasuan eta abar. Hobekuntza posiblea: Bertsozale Elkartearekin batera erregelak fintzea
- Hiztegia
 - Corputa: XUXEN hiztegia, Berria egunkarian agertutako maiztasunaren arabera eskaintzen dira. (bertsolaritzaren corputa erabiltzeak, homogeneizazio gehiago sortu dezake)
 - Sinonimoak: Euskal Wordnet (IXA) bidez
- (Aurreko puntuaren inguruan) IXA taldeak zerrenda gehiago ditu, erabilgarriak izan daitezkeenak. Ahoskeraren arabeko corputa sortu eta aplikatu daiteke (Adibidez: [astua], [gaztia]...)
- Doinuak Bertso-doinutegia. Momentuz, duplikatu baten bidez, baina egokiena doinutegia bera zuzenean erabiltzea izango da, abiadura arazorik sortzen ez badu behintzat (zerbitzari berean izateak lagunduko luke)
- Bertso kantatzailea: Doinua, erritmoa eta bertsoa emanez gero, bertsoa automatikoki kantatzen du.
 - Aholab-ekin (<http://aholab.es>) harremanetan, ahoskera hobetzeko, gaztelararen ahoskeraren araberakoa baita orain duena. (Iñaki Sainz, Eva Navas, Igor Odriozola). Donostian beste talde bat ere ari da alor honetan: Nora Barroso, Ezeiza,....
 - Doinuen errekonozimendu automatikoa garatzeko egitasmoa garatu daiteke bide honetatik.
- Neurri asigantzailea: Bertsoaren testua eman eta dagokion neurria (estrofaren izena) identifikatzen du (banan-banan parekatzen ditu). Parekatze guztiak erakusteko moduan hobetu daiteke. Bertsozale Elkarteak estrofen izenen zerrenda osatu eta finkatu dezake, neurri posible guztiak aurreikusteko moduan.
- Erabiltzaileen kontrola: Erabiltzaile bezala izena eman eta bakoitzak sortutako bertsoak eta bertsoaldiak gordetzeko aukera dago. Bertsoaldiak beste erabiltzaileek ikusteko moduan jarri daitezke.
- Moduak:

- Irakasle moduan: bertsoa idatzi ahala, mezuak erakusten ditu, ohar bezala.
- Librean: ez du mezurik erakusten.
- Laguntzagunea: Interneten dauden hainbat hizkuntza-tresnen loturak eskaintzen ditu: atsotitzak,...

Funtzionamendua:
(Oraindik ez dago interneten ikusteko moduan)

Pantailaren erdian bertsoa idazteko arbela dago. Arbelaren ezker eta eskuman neurria, doinua, hitzak eta sinonimoak aukeratzeko tresnak daude. Arbelaren goiko aldean, bertsoak gordetzeko eta abarretarako aukerak daude.

- 1.Neurria (estrofa) aukeratu eta hari dagozkion doinuak agertuko dira azpian.
- 2.Doinua aukeratu eta arbelean bertsoa idazteko “lerroak” agertuko dira.
- 3.Bukaerako hitza jarri eta Aitzolek garatutako programaziotik hitzari dagozkion bukaerak jaso eta XUXENen hiztegitik errimatutako hitzak eskainiko dira, Berria egunkarian gehien erabili direnen arabera.
- 4.Silabak mugatu daitezke
- 5.Behar den hitzaren sinonimoak ere eskainiko ditu.
- 6.Bertsoa idatzi ahala, lerro bakoitzak dituen silaba kopurua zenbatuko ditu.
- 7.Bertsoa bukatzean, errimen konprobazioa egingo du.

Proposamena:

Bertsozale Elkartek Manex Agirrezabalek garatutako Arbela Digitala bere zerbitzarian eskaintzea, eta etorkizunean tresnak beharko dituen hobekuntza eta bestelako garapenez arduratzea.

Horretarako, EHU, IXA taldea eta Bertsozale Elkartearen arteko hitzarmena sinatuko da, Manex Agirrezabalen egiletza, erabilitako tresnen lizentziak (printzipioz arazorik ez, libreak baitira), eta etorkizunean Arbela Digitala egitasmoak izan ditzaken garapen aukerak finkatu ahal izateko.

- Arbela Digitala Bertsozale Elkarteko zerbitzarian kokatzeko aukerak aztertuko dira: BDB datu-basearen interfazean kokatu eta beste datu-baseekin (doinutegia) lotzeko aukerak aztertuko ditu Nere Erkiagak Iametzta enpresako informatikariarekin batera.

- 1.Arbela Digitala publiko egin aurretik behar dituen oinarrizko zuzenketak egiteko baliabideak bilatuko ditu Bertsozale Elkarteak, Manex Agirrezabalen hemendik aurrerako lan-orduak eta diseinu-egokitzapenak ordaindu ahal izateko.
- 1.Arbela Digitalaren oinarrizko zuzenketak egiteko taldea osatuko da (Manex Agirrezabal, Aitzol Astigarraga, Karlos Aizpurua, Ixiar Eizagirre, Nere Erkiaga...), ondoren bertso-eskoletako kide batzuei frogak egiteko eskaini eta atzeman ez diren oinarrizko zuzenketak atzeman eta konpontzeko.
- 1.Arbela Digitalaren aurkezpen publikoa udazkenean egiteko ahalegina egingo da. (Bertsozale Elkarteak aurkezpenerako komunikazio laguntza eskaini dezake, behar izanez gero).
- 1.Etorkizunean, edukizko hobekuntzak identifikatu eta garapenak bideratzeko taldea osatuko da: Manex Agirrezabal, Ixiar Eizagirre (edo transmisio saileko kidea), Karlos Aizpurua (edo ikerkuntza saileko kidea), ...

11.bilera

Data: 2011ko Irailaren 12a

Partaideak: Nere Erkiaga (Xenpelar dokumentazio zentroa), Iker Alkoz (iametza, informatikaria) eta Manex

Lekua: Zumarragako Zelai Arizti kultur etxea

- Egindako lana aurkeztu zaio Ikerri, dauden baliabideak nahikoa izango diren jakireko
- Neurrien taula bat sortu behar dela erabakitzen da, eta gainera gehien erabiltzen diren neurriak zeintzuk diren azalduko dena. Izenak ere humanoagoak izango dira, hala nola, Bederatzikoa Handiaren moldekoa 6 puntuz esan beharrean, Iparragirrearen neurria
- Oraingoz bideo eremua kentzeko eskatu didate, sor ditzakeen gaizkiulertuegatik
- Iametzak badauka errepositorioa, beraz, nik errepositorio hartara igoko nituzke bertsiio berriak
- Bertsozale elkartearen orrira arbela moldatzeko beharko diren orduak zenbatu behar dira

- Neurrien taula prest dagoenean abisatuko zait eta orduan arbela bertsozale elkartearen beharretara egokitzen hasi

12.bilera

Data: 2010eko

Partaideak: Iñaki Alegria eta Manex Agirrezabal

Lekua: Iñakiren bulegoa

- Egindako memoria aurkeztu zaio Iñakiri
- Arbela bera ere erakutsi zaio
- Iñakik proiektuarentzako izen bat proposatu du, "bertsoBot: lehen urratsak"
- Bilbon aholab-en garatutako bertso abeslaria ere aurkeztu zaio
- Astelehenean Nere eta Iker Alkozekin izandako bilerako aspektu batzuk aztertu dira
- Uzaileko bilerako gauza batzuk ere gogoratu dira
- Lizentziekin arazoak ez daudela egiaztatu dut
- Hobekuntza posibleak proposatu dira, hala nola, sinonimo bilatzailean lematizatzailea, errima egiaztatzailea fomarekin, errimak atzizkien arabera markatzea
- Kantatzearen bidezko doinu bilaketa masterrean zehar garatuko da, beta bertso bat, ikasketa automatikoa irakasgaiaren lan praktikoko gisa

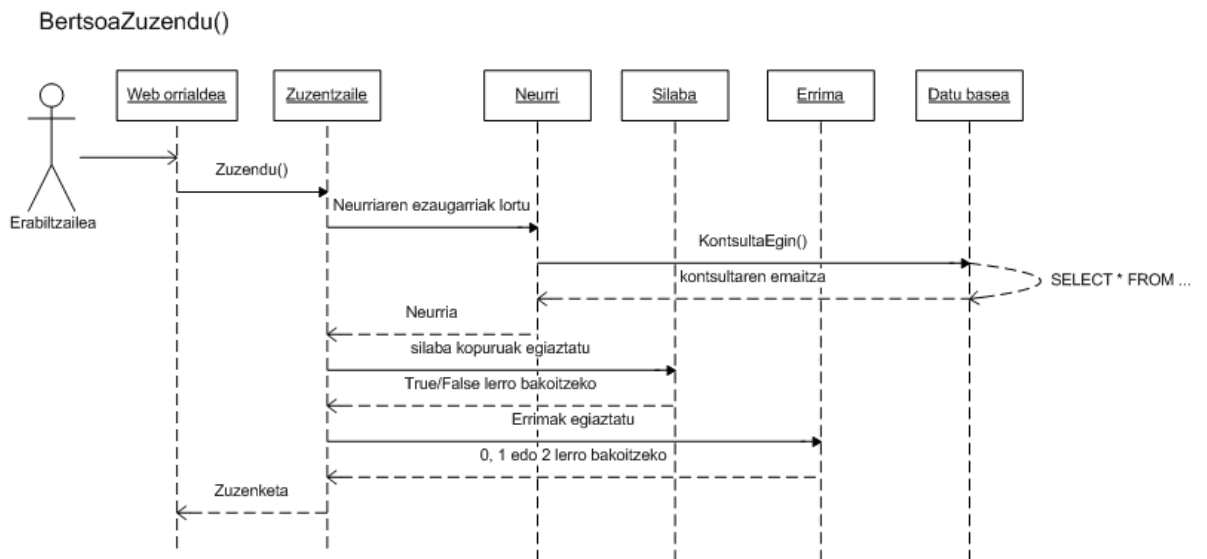
Eranskina B

Atal ezberdinen sekuentzia-diagramak

Hemen garapenean zehar lagungarri suertatu izan zaizkigun lau algoritmo daude, bakoitzaren egitekoaren azalpen labur eta zehatz batekin batera.

B.1 Bertsoa gorde

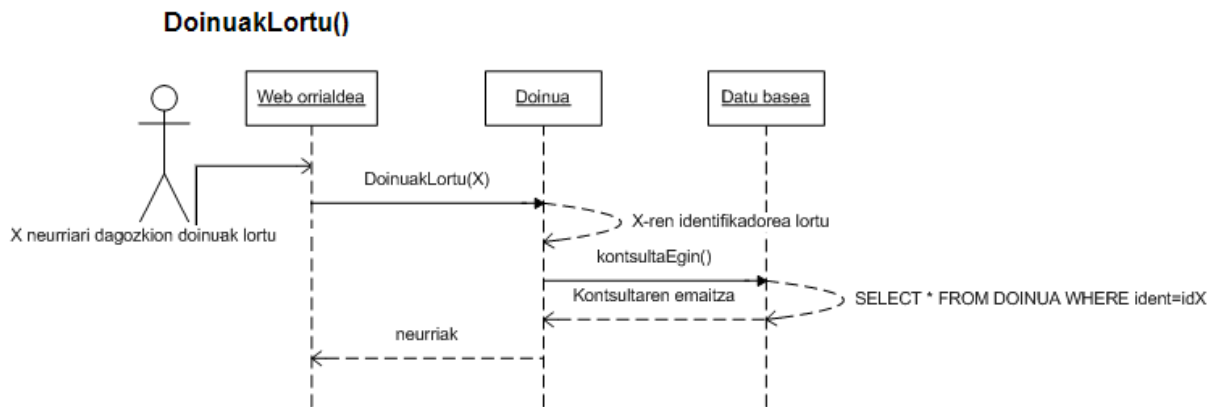
Funtzio honek, bertso bat eta informazio osagarria jaso, eta datu-basean txertatzen du hau.



Irudia B.1: Bertsoa gorde

B.2 Doinuak lortu

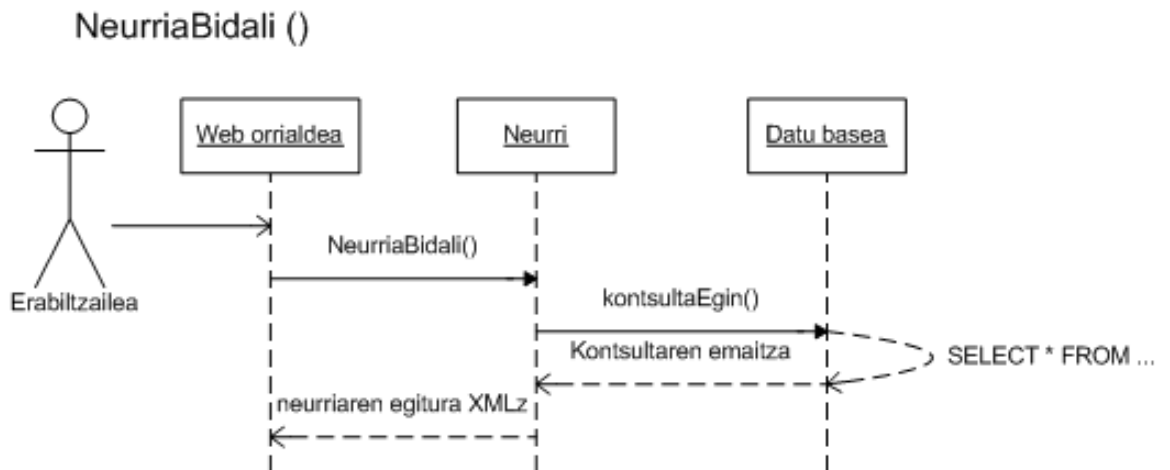
Honek parametro gisa neurri bat jasotzen du eta datu-basean begiratu ondoren, hari dagozkion doinuak itzultzen ditu.



Irudia B.2: Doinuak lortu

B.3 Neurria bidali

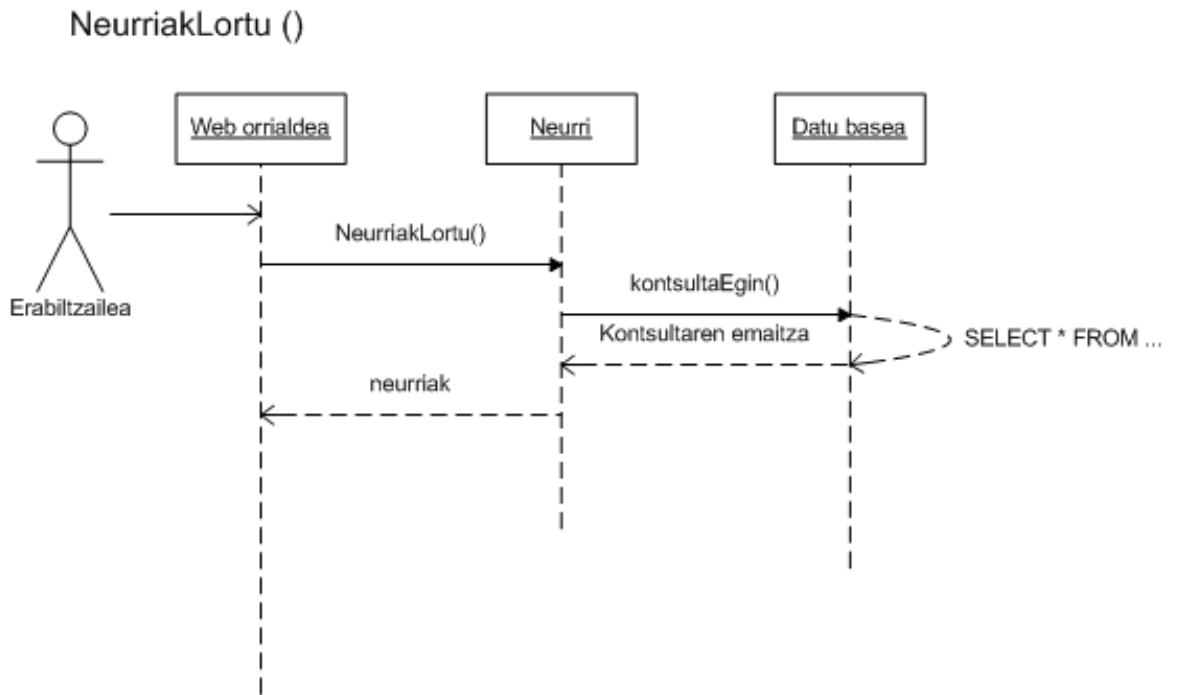
Izena horren esanguratsua izan ez arren, funtzio honen egitekoa neurri baten egitura eskuratzea da.



Irudia B.3: Neurria bidali

B.4 Neurriak lortu

Amaitzeko, honen lana neurri guztiak eskuratzea da. Modu berdinsuan funtzionatzen du XLerroduNeurriakLortu funtzioak.



Irudia B.4: Neurriak lortu

bertsoBot Robot artista honek
sortu du kriston eztanda
eta bete da Euskal Herriko
bertsozaleon demanda
memoria idazten
orduk burua nahasten
ta baita zenbat parranda
lana izan arren esan behar da
dena plazer bat izan da

Doinua: Iparragirre abila dela
Neurria: Bederatzikoa, handiaren moldekoa, sei puntuz