

# TALDEKO PROPOSAMENA



EGILEAK

---

Iker Hernández Navas

Achraf Allach Chahboun

# AURKIBIDEA

## 1. SARRERA

*1.1 Dokumentuaren helburu nagusia*

*1.2 Testuingurua*

*1.3 Dokumentuaren egitura*

## 2. PROIEKTUAREN HELBURUAK

*2.1 Helburu Orokorra*

*2.2 Helburu Espezifikoak*

## 3. PROIEKTUAREN ESPARRUA

*3.1 Funtzionalitate nagusiak*

*3.2 Mugak eta lehenengo bertsioaren ezaugarriak*

## 4. BEHAR TEKNIKOAK

*4.1 Erabiltzaileen funtzionalitateak*

## 5. SISTEMA INFORMATIKOAREN GARAPENA

*5.1 Programazioa: Java Bidezko Garapena*

*5.2 Datu-Basea: SQL Workbench bidezko kudeaketa*

*5.3 Garapenerako Kudeaketa: GitHub bidezko kontrola*

*5.4 Zerbitzariak eta Docker Azpiegitura*

*5.5 Web Plataforma eta Erabiltzaileen Interakzioa*

*5.6 Administrazioaren Aplikazioa*

*5.7 Justifikazioa eta Egitura Teknikoaren Arrazoiak*

## 6. PLANIFIKAZIOA

*6.1 Lehen fasea: Planifikazioa*

*6.2 Bigarren fasea: Garapena eta Exekuzioa*

*6.3 Hirugarren fasea: Abiaraztea eta Azken Ukituak*

# 1. SARRERA

## 1.1 Dokumentuaren helburu nagusia

Dokumentu honek AlaiktoMUGI enpresaren beharrei erantzuten dien **taxi esleipen sistema informatiko baten diseinu eta garapena** proposatzen du. Sistemaren helburua **bezeroek eta gidariek dituzten beharrak asetzea, esleipen prozesua automatizatzea eta zerbitzuaren koordinazioa hobetzea** da, zerbitzua **eraginkorragoa eta eskalagarria** izan dadin.

## 1.2 Testuingurua

AlaiktoMUGIk Gipuzkoako taxi-merkatuan **lidergoa bereganatu du**, eta bere zerbitzuak **Araba, Bizkaia eta Nafarroara** zabaltzeko asmoa du. Hazkunde honekin batera, taxi zerbitzuaren **kudeaketa eta koordinazioa optimizatzeko beharra** identifikatu da. Sistema informatiko berriak gidariek eta bezeroek **bidaiak modu eraginkorrean eskatu eta kudeatzeko aukera izango dute**, zerbitzuaren **kalitatea hobetzeko**.

## 1.3 Dokumentuaren egitura

Dokumentu hau **hainbat ataletan egituratua** dago, behar teknikoak, garapen metodologia eta sistemaren funtzionamenduaren analisisa aurkezteko:

- **Proiektuaren helburuak:** Sistemaren garapenaren arrazoiak eta helburu espezifikoak.
- **Proiektuaren esparrua:** Funtzionalitate nagusiak eta lehenengo bertsioaren mugak.
- **Behar teknikoak:** Erabiltzaileen rolak eta funtzionalitateak.
- **Garapen metodologia:** Softwarearen diseinu eta garapen prozesua.
- **Teknologia proposatua:** Erabiliko diren plataformak eta tresnak.
- **Segurtasuna eta mantentzea:** Datuen babesa eta sistemaren etengabeko hobekuntza.
- **Ondorioak eta hurrengo urratsak:** Proiektuaren aurreikuspenak eta ekintza planak.

## 2.PROIEKTUAREN HELBURUAK

### 2.1 Helburu Orokorra

Proiektu honen helburua da taxi-zerbitzuaren kudeaketa digitalizatzea, bidai eskaeren eta esleipenen prozesua automatizatuz. Sistema informatiko berri bat garatuko da, bezeroen eta gidarien arteko interakzioa hobetzeko eta enpresaren eraginkortasuna handitzeko.

### 2.2 Helburu Espezifikoak

- **Erabiltzaileen interakzioa erraztea:** Plataforma bat garatuko da, bezeroek taxia eskatzeko eta gidariek eskaerak jasotzeko prozesua sinplifikatzeko.
- **Datuen kudeaketa optimizatzea:** Bidai eta erabiltzaile erregistroak ordenatuta eta zentralizatuta gordeko dira, enpresak kudeaketa azkarrago eta fidagarriagoa izan dezan.
- **Plataforma anitzetatik eskuragarria den sistema bat diseinatzea:** Web aplikazioa eta mugikorretarako aplikazioa garatuko dira, edozein gailutik erabilgarri izateko.

## 3.PROIEKTUAREN ESPARRUA

### 3.1 Funtzionalitate nagusiak

Sistemak funtzio hauek izango ditu:

- **Erabiltzaileen eta gidarien erregistroa:** Plataforma barruan kontu bat sortu eta datuak kudeatzea.
- **Bidai eskaeren kudeaketa:** Bezeroek taxiak eskatzeko aukera izango dute, eta gidariak esleipen prozesua kudeatuko dute denbora errealean.
- **Bidai historikoa:** Erabiltzaile bakoitzak bere bidai historikoa ikusi ahal izango du, zerbitzuaren datuak gordez.
- **Kudeatzailearen aplikazio eskusiboa:** Administrazioentzako aplikazio bat egongo da, gidariak, bidaiak eta estatistikak kudeatzeko.

### 3.2 Mugak eta lehenengo bertsioaren ezaugarriak

- Lehenengo bertsioan ez da ordainketa integratua egongo, erabilgarritasunaren funtzio nagusiak lehenetsiz.
- Gidariak manualki hautatuko dituzte bidaiak, optimizazio algoritmo bat ezarri aurretik.
- Bezeroek eskaerak aplikazio bidez soilik egin ahal izango dituzte, web bidezko aukera etorkizunean garatzeko.

## 4. BEHAR TEKNIKOAK

### 4.1 Erabiltzaileen funtzionalitateak

Sistema informatikoak hiru erabiltzaile mota izango ditu: Bezeroak, Gidariak eta Kudeatzailea. Haien behar teknikoak honela definitzen dira:

#### Bezeroak

- Zerbitzuan erregistratzeko aukera izango dute.
- Bidaiak unean bertan eskatu edo programatu ahal izango dituzte.
- Bidaiaren egoera denbora errealean ikusi ahal izango dute.
- Zerbitzuaren erabileraren historikoa kontsultatzeko aukera izango dute.

#### Gidariak

- Sistemara sarbidea izango dute, behar duten informazioa uneoro eskuragarri izateko.
- Bere bidaiak soilik ikusi ahal izango dituzte, esleipen pertsonalizatua izanik.
- Esleitutako bidaiak onartu edo baztertu ahal izango dituzte.
- Bidaiaren hasiera eta amaiera erregistratu ahal izango dute, zerbitzuaren jarraipena egiteko.

#### Kudeatzailea

- Bere aplikazio propioa izango du, beste inork sartu ezingo duen ingurune eksklusiboa izanik.
- Gidarien kudeaketa egingo du: gehitu, kendu, editatu eta haien jardueraren jarraipena egin.
- Erabiltzaile, gidari eta bidaien historiala ikusi, filtratu eta ordenatu ahal izango du.
- Sistemaren segurtasun eta datuen babesa bermatuko ditu.

## 5. SISTEMA INFORMATIKOAREN GARAPENA

Sistema informatikoa **modulu espezifikoetan banatu da**, prozesua eraginkorragoa bihurtzeko eta **garapenaren antolamendua hobetzeko**. Modulu bakoitzak bere funtzionalitateak izango ditu, eta haien **erabilera teknikoa** justifikatuta egongo da.

**Atalak honela egituratu dira:**

- **5.1 Programazioa:** Java lengoaiarekin aplikazioaren garapena.
- **5.2 Datu-basea:** Workbench bidezko datu gordailua eta administrazioa.
- **5.3 Garapenerako Kudeaketa:** GitHub bidezko kontrola eta talde-lana.
- **5.4 Zerbitzari eta Azpiegitura:** Docker bidezko zerbitzari-egitura.
- **5.5 Web Plataforma:** PHP eta HTML bidezko web interfazeak.
- **5.6 Administratzailearen Aplikazioa:** Taxi zerbitzuaren kudeaketa-sistema.

## 5.1 Programazioa: Java Bidezko Garapena

Sistemaren garapenerako **Java** lengoaia hautatu da. Java aukera egokia da honako arrazoiak direla eta:

- **Multiplataforma** da, edozein sistema eragiletan exekutatu daiteke.
- **Errendimendua altua** da aplikazio konplexuetan.
- **Eskalagarria da**, egitura handitzeko aukera eskaintzen du.

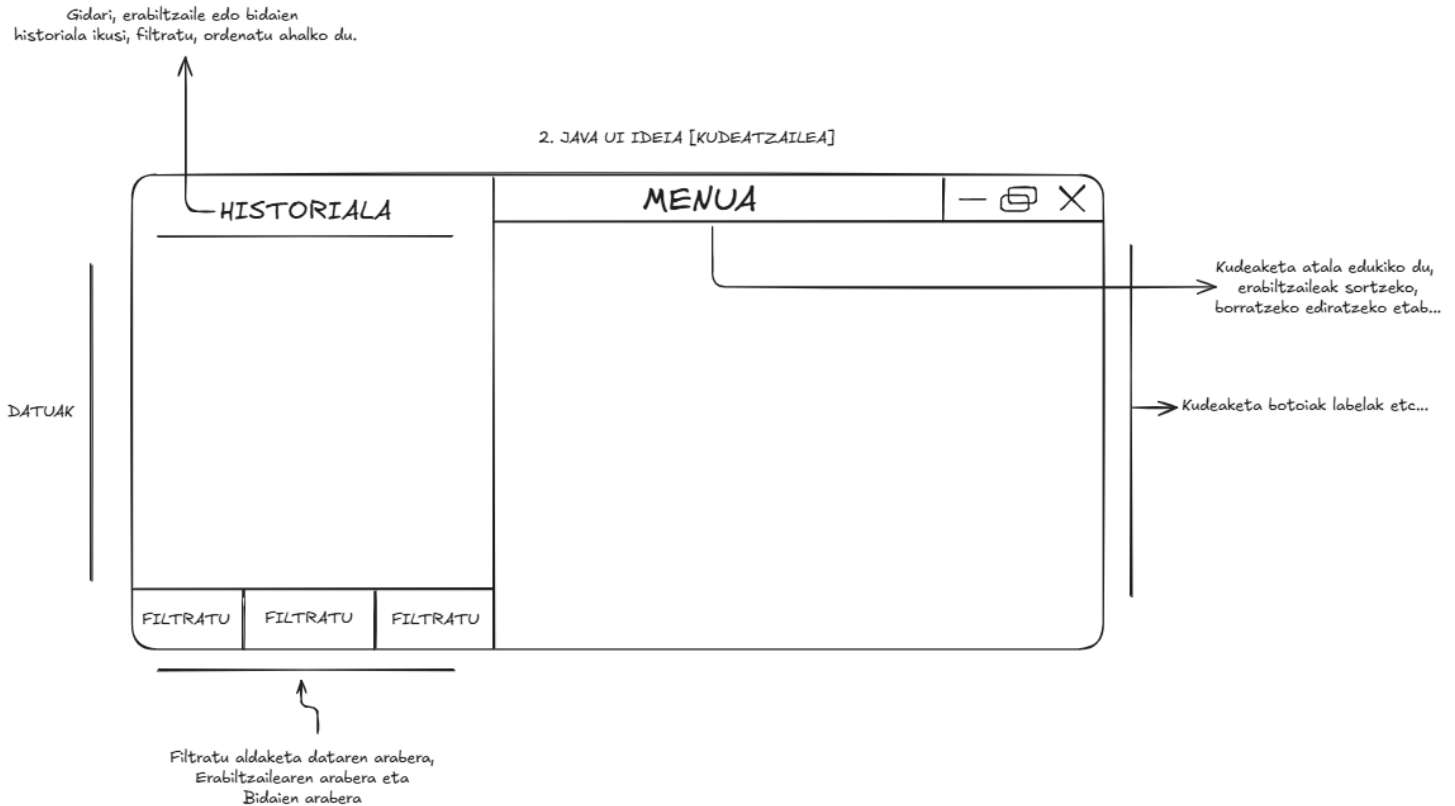
**IntelliJ IDEA** garapen ingurunea hautatu da, programazio prozesua errazteko. IntelliJ erabiltzea erabaki dugu **GUI diseinua modu errazagoan garatzeko**, izan ere:

- Interfaze grafikoa natively sortzeko aukera eskaintzen du, plugin osagarririk gabe.
- Errendimendu altua eta IDE integratua eskaintzen ditu.
- Plugin biltegi handia deneriko plugin bilduma dago.

**Proiektu honetan bi aplikazio garatuko dira:**

1. **Administratzailearen aplikazioa:** Taxi zerbitzuaren kudeaketa egiteko.
2. **Web aplikazioa:** Bezeroek eta gidariek elkarreragiteko.





## Administratzailearen Interfaze Grafikoa (Java)

Java bidez garatutako mahaigaineko aplikazioa izango da, kudeatzaileentzat eksklusiboa. Interfazeak atal hauek izango ditu:

- **Goiko menua:** Nabigaziorako eta kudeaketa funtzionalitateetarako botoiak (erabiltzaile berriak gehitu, ezabatu, editatu...).
- **Ezkerreko panela (HISTORIALA):** Gidari, erabiltzaile eta bidai historiala ikusteko.
- **Beheko botoiak (FILTRATU):** Datuak filtratzeko aukerak: data, erabiltzailea edo bidai motaren arabera.
- **Eskuineko panela:** Kudeaketa botoiak eta aukerak (sortu, aldatu, ezabatu...).

Interfaze sinplea, erabilerraza eta erantzun azkarrekoa izango da, datuen kudeaketa eraginkorra ahalbidetzeko.

## 5.2 Datu-Basea: SQL Workbench bidezko kudeaketa

Sistema informatikoaren funtzionamendu egokia bermatzeko, datu-basea erabiliko dugu. MySQL Workbench bidez informazioa egituratuta eta seguru gordeko da, erabiltzaileen eta administratzaileen beharretara egokituz.

### Datu-basearen helburua eta erabilera

Datu-baseak sistemaren funtsezko osagai guztiak gordetzeko erabiliko da, aplikazioaren eta webgunearen egonkortasuna mantenduz. Honako informazioa bilduko da:

- **Erabiltzaileen kudeaketa:** bezeroen, gidarien eta administratzaileen kontu eta harremanetarako datuak.
- **Bidaia eta erreserbak:** bezeroek egindako eskaerak, haien egoera eta esleipen-prozesua.
- **Gidari eta langileen historikoa:** gidariak egin dituzten zerbitzuak eta erregistratutako bidaia.
- **Administrazioaren kontrola:** bidai esleipenen jarraipena eta langileen jardura kontrolatzeko sistemak.
- **Segurtasun mekanismoak:** baimenen kudeaketa, autentifikazio sistema eta datuen osotasuna bermatzeko mekanismoak.

### Datu-baseko taulak eta haien erabilera

Datu-baseak informazioa egituratzeko hainbat taula izango ditu, bakoitza bere funtzio espezifikoarekin:

- **Erabiltzaileak:** Zerbitzua erabiltzen duten bezero eta gidarien kontuak gordeko dira. Bertan jasoko dira haien identifikazio-datuak, harremanetarako informazioa eta kontu egoera, erregistroak administratzaileek kontrolatzeko aukera izateko.
- **Administratzaileak:** Administratzaileei sisteman sartzeko baimena emateko gordeko da informazioa. Haien datuak, sarbide-eskubideak eta kontu-egoera erregistratuko dira, administrazio-lanak errazteko eta langileen identifikazioa egokia bermatzeko.

- **Historikoa:** Gidariek egin dituzten zerbitzuak erregistratzeko erabiliko da. Historiko honetan gordeko dira gidari bakoitzak egindako bidaiak, haien data eta egoera, bezeroek zerbitzuak kontsultatu ahal izateko, eta administratzaileek langileen jarduera eta errendimendua kontrolatzeko.
- **Bidaiak:** Bezeroek eskatutako bidaiak gordeko dira. Erregistro bakoitzak jatorria, helmuga, data eta esleitutako gidaria jasoko du. Horrek sistemaren antolaketa erraztuko du eta eskaeren egoera kontrolatzeko aukera emango du.
- **Erreserbak:** Administratzaileek bidaiak kudeatzeko erabiliko da. Bertan gordeko dira bezeroen eskaerak, administratzaileek aztertzeko eta bidaiak gidariei esleitzeko. Gainera, erreserben egoera mantenduko da, sistemaren funtzionamendua argi eta kontrolatuta izateko.

## Teknologia eta integrazioa

Datu-basea beste sistemekin konektatuko da, informazioa eguneratua eta erabilgarri mantentzeko:

- Web aplikazioarekin konektatuko da, bezeroek bidaiak eskatzeko eta kontuak kudeatzeko.
- Administratzailearen aplikazioarekin lotuko da, zerbitzua esleitzeko eta kontrolatzeko.

## Zergatik MySQL Workbench?

MySQL Workbench aukeratu dugu taulen egitura eta erlazioak argi definitzeko, baita segurtasuna eta datuen osotasuna bermatzeko ere. Honako abantailak eskaintzen ditu:

- Datuen egituraketa argia eta bisuala, informazioa ordenatuta mantentzeko.
- Kotsulten optimizazioa, datu-atzipena azkartzeko.
- Erabiltzaileen autentifikazioa eta segurtasuna, baimenak kontrolatzeko eta datuak babesteko.
- Sistemaren egonkortasuna eta eskalagarritasuna, erabiltzaile kopurua handitu ahala errendimendua mantentzeko.

Sistema hau datuen kudeaketa optimizatzeko eta informazioa seguru mantentzeko diseinatuta dago. MySQL Workbench erabiliz, datu-basearen egonkortasuna, segurtasuna eta erabilgarritasuna bermatuko dira.

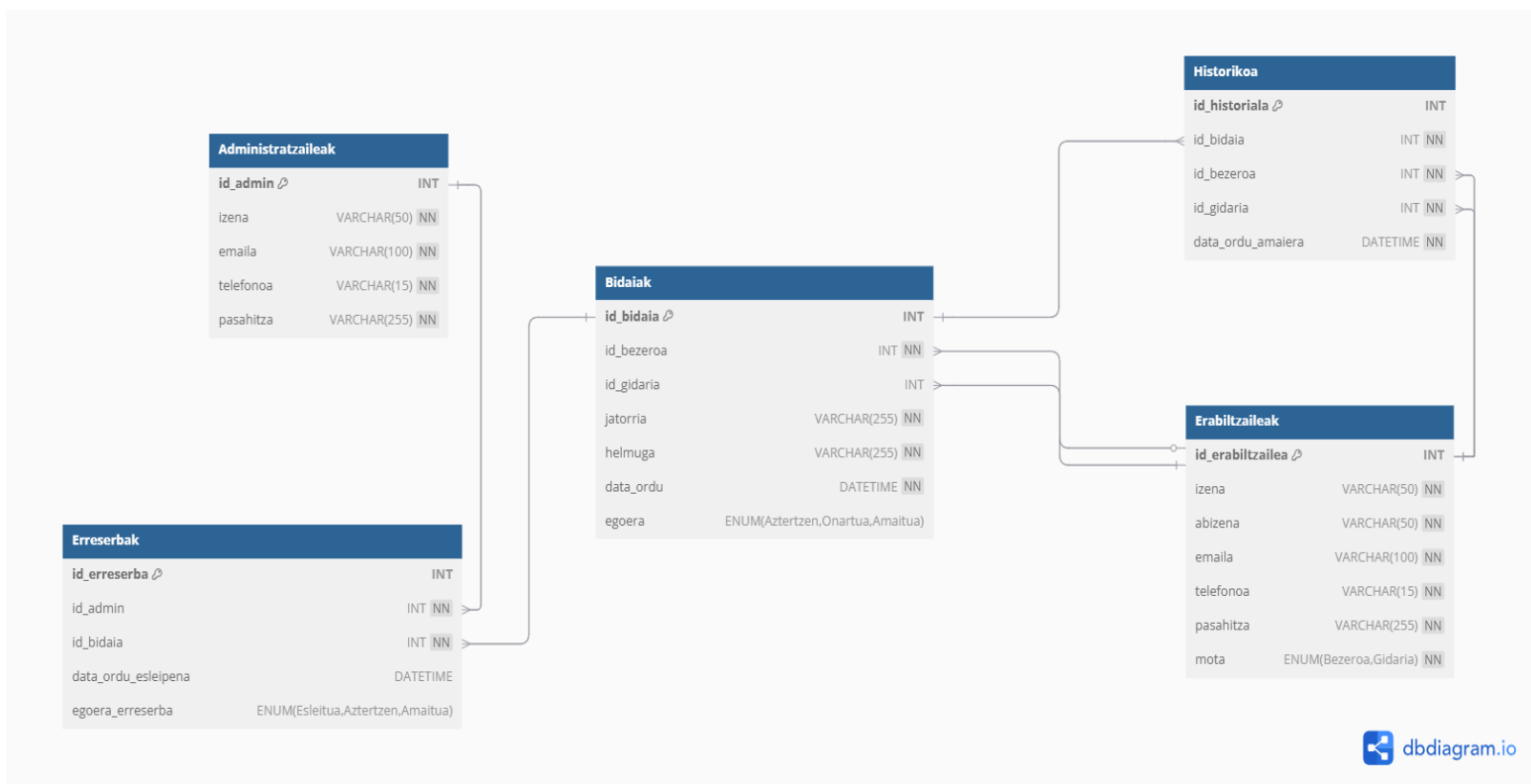
## Datu-Basearen Eskema Kontzeptuala

Sistema eraginkorra garatzeko, datu-basearen eskema kontzeptuala egingo da. Honek datu-baseko taulen arteko harremanak eta erabiltzaile bakoitzak zer informazio kudeatzen duen argi zehaztuko ditu.

## Zergatik egiten dugu eskema kontzeptuala?

- **Datuen egitura bisualizatzea**, sistemaren diseinu teknikoa hobetzeko.
- **Erabiltzaileen eta entitateen arteko erlazioak definitzea**, datuen antolaketa argitzeko.
- **Eragiketak optimizatzeko**, kontsultak eta transakzioak eraginkorrak izateko.

## Eskema Kontzeptualaren Irudia



## Datu-Hiztegi Taula

Sistema informatikoaren garapenean, datu-hiztegi taula erabiliko dugu, taula bakoitzak duen informazioa eta atributuak zehazteko.

### Zergatik behar dugu datu-hiztegi taula?

- **Datu-baseko elementu bakoitzaren definizio zehatza** edukitzeko.
- **Datu motak eta erlazioak argi ezartzeko**, garapen prozesua egokia izan dadin.
- **Garapenerako dokumentazioa indartzeko**, datuen antolaketa ezaguna izan dadin talde osoarentzat.

### Datu-Hiztegi Taularen Irudia

ENTITATEA	ATRIBUTUA	DATU-MOTA	MURRIZPENA	NULL-BALOREA	ADIBIDEA
ERABILTZAILEAK	id_erabiltzailea	INT	--	EZ	101
	izena	VARCHAR(50)	--	EZ	Patxi
	abizena	VARCHAR(50)	--	EZ	Arrigomezgorta
	emaila	VARCHAR(100)	@ eduki behar du	EZ	parrigomezgorta@gmail.com
	telefonoa	KRT(9)	Krt. num.	EZ	688 447 893
	pasahitza	VARCHAR(255)	--	EZ	Pa\$\$word
ADMINISTRATZAILEAK	mota	ENUM('Bezero', 'Langilea')	--	EZ	Langilea
	id_admin	INT	--	EZ	25
	izena	VARCHAR(50)	--	EZ	Joxean
	emaila	VARCHAR(100)	@ eduki behar du	EZ	joxean25@gmail.com
	telefonoa	KRT(9)	Krt. num.	EZ	688 447 893
	pasahitza	VARCHAR(255)	--	EZ	Pa\$\$word
BIDAIAK	id_bidaia	INT	--	EZ	25068
	id_bezeroa (FK)	INT	--	EZ	218
	id_gidaria (FK)	INT	--	EZ	24
	jatorria	VARCHAR(250)	--	EZ	Kale Nagusia 12, Tolosa
	helmuga	VARCHAR(250)	--	BAI	Okendo Kalea 8, Donosti
	data_ordu	DATETIME	uu/hh/ee oo:mm	EZ	25/12/2024 12:00
HISTORIKOA	egoera	ENUM('Azterzen', 'Onartua', 'Amailua')	--	EZ	Onartua
	id_historiala	INT	--	EZ	30254
	id_bidaia (FK)	INT	--	EZ	125
	id_bezeroa (FK)	INT	--	EZ	26
	id_gidaria (FK)	INT	--	EZ	12
	data_ordu_amaiera	DATETIME	uu/hh/ee oo:mm	EZ	21/05/07 5:17
ERRESERBAK	id_erreserba	INT	--	EZ	369
	id_admin (FK)	INT	--	EZ	5
	id_bidaia (FK)	INT	--	EZ	361269
	data_ordu_esleipena	DATETIME	uu/hh/ee oo:mm	EZ	25/01/07 12:00
	egoera_erreserba	ENUM('Esleitua', 'Azterzen', 'Amailua')	--	EZ	Amailua

## 5.3 Garapenerako Kudeaketa: GitHub bidezko kontrola

Garapen lan guztiak GitHub plataforman kudeatuko dira, kodearen kontrola eta talde-lana errazteko. GitHub-en erabilerak garapen prozesua egituratu eta aldaketak kontrolatzeko mekanismo fidagarria eskaintzen du.

### GitHub aukeratzeak hainbat abantaila eskaintzen ditu:

- Bertsio-kontrol zorrotza, aldaketak dokumentatzeko.
- Kodearen segurtasuna, baimen-egiturak kudeatzeko.
- Backup sistema, garapenaren historia gordetzeko.
- Taldekideen arteko lankidetzak, kodea partekatzeko eta aldaketak kontrolatzeko.

GitHub bidez, **datu-basea, aplikazioak eta web garapena** integratuko dira, **garapenaren egitura koherentea bermatzeko**.

## 5.4 Zerbitzariak eta Docker Azpiegitura

Zerbitzariak Docker kontenedoreen bidez kudeatuko dira, azpiegitura fidagarria eta eskalagarria bermatzeko. Docker-ek aukera ematen du zerbitzariak independenteki exekutatzeke, baliabideak hobeto optimizatzeke eta segurtasuna handitzeke.

Sistema hiru kontenedoretan banatuko da, bakoitza bere funtzio espezifikoarekin:

### 1. Datu-base zerbitzaria (SQL Workbench)

- Datuak gordetzeke eta kudeatzeko erabiliko da.
- SQL Workbench-en bidez, transakzioak exekutatu eta erabiltzaileen informazioa kudeatuko da.
- Erabiltzaileek eta administratzaileek datu-basearekin komunikatu ahal izango dute.

### 2. Web zerbitzaria (PHP eta HTML)

- Web aplikazioa exekutatzeke erabiliko da, bezeroek eta gidariek zerbitzuarekin elkarreragiteke.
- Web zerbitzaria HTML, CSS, JavaScript eta PHP teknologiekin garatuko da.
- Bezeroek bidaiak eskatu, kontua kudeatu eta gidariekin interakzioa izango dute.

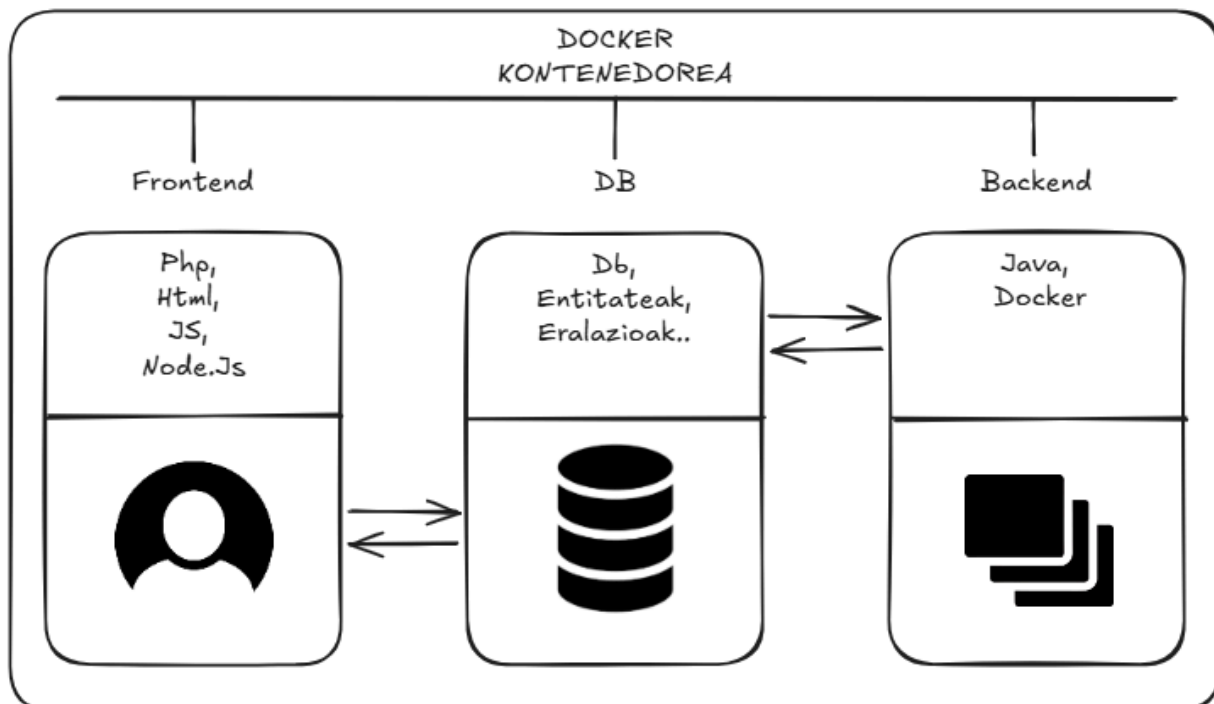
### 3. Java zerbitzaria (Administratzailearen aplikazioa)

- Administratzaileak bidaiak kudeatzeko eta gidarien egoera kontrolatzeko erabiliko da.
- Java eta Docker erabiliz garatuko da, sistemaren funtzionamendua malgutasunez mantentzeke.
- Zerbitzari honek administratzailearen interfaze grafikoa exekutatuko du, erabilgarritasuna hobetuz.

Docker-ek hainbat abantaila eskaintzen ditu:

- **Independentzia:** Kontenedore bakoitza isolatuta dago eta ez du beste sistemaren osagaietan eraginik izango.
- **Eraginkortasuna:** Kontenedoreak azkar exekutatzen dira eta baliabideak optimizatuta mantentzen dituzte.
- **Eskalagarritasuna:** Sistemaren beharren arabera kontenedore gehiago gehitu edo kendu daitezke.
- **Mantentze-lanak errazteko** → Kontenedore bakoitza modu modularrean kudeatzen da, eguneraketak bakarka exekutatzeko aukera emanez.

Docker azpiegitura erabiltzeak sistemaren konplexutasuna murrizten du, osagai bakoitza bereizita exekutatzea ahalbidetuz. Horrek zerbitzuaren egonkortasuna hobetzen du eta sistema etorkizunean handitzeko aukera ematen du.





## 5.5 Web Plataforma eta Erabiltzaileen Interakzioa

Web plataformak bezeroek eta gidariek taxi zerbitzua erabiltzeko interfaze bisuala izango du.

### Bezeroak (Web bidez):

- Erregistratu eta saioa hasi.
- Taxi bat eskatu (jatorria, helmuga, ordutegia).
- Historiala ikusi.
- Zerbitzuaren egoera ikusi (onartua, aztertzen, amaituta...).
- Txatean komunikatu gidariarekin.

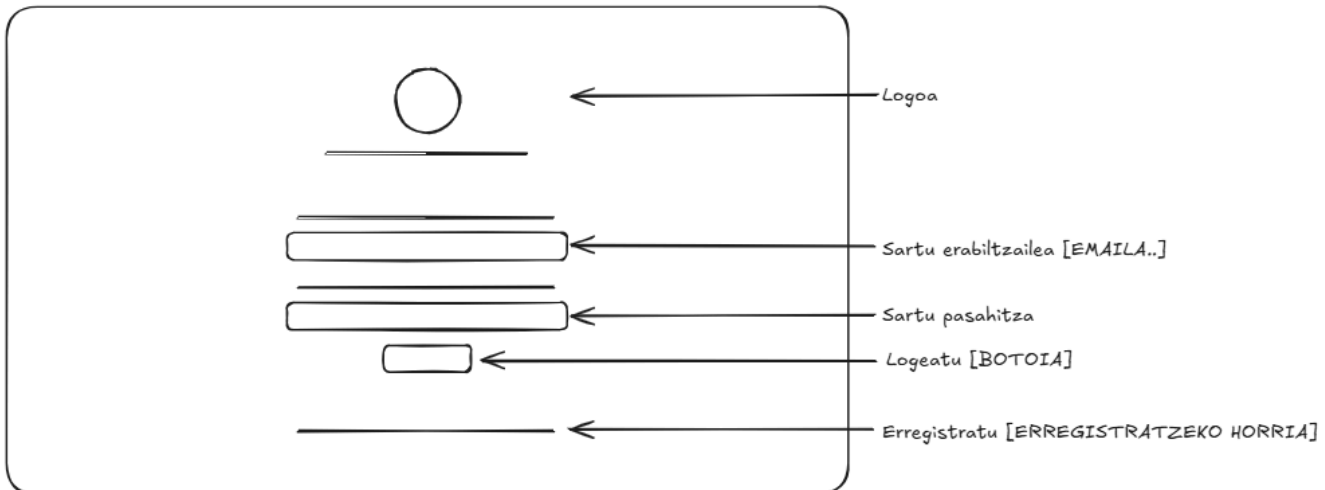
### Gidariak/Langileak (Web bidez):

- Saioa hasi administratzaileak sortutako kontuarekin (ezin du erregistratu).
- Esleitutako bidaiak ikusi.
- Bidaiaren egoera aldatu (martxan, amaituta...).
- Bezeroarekin komunikatu txat bidez.

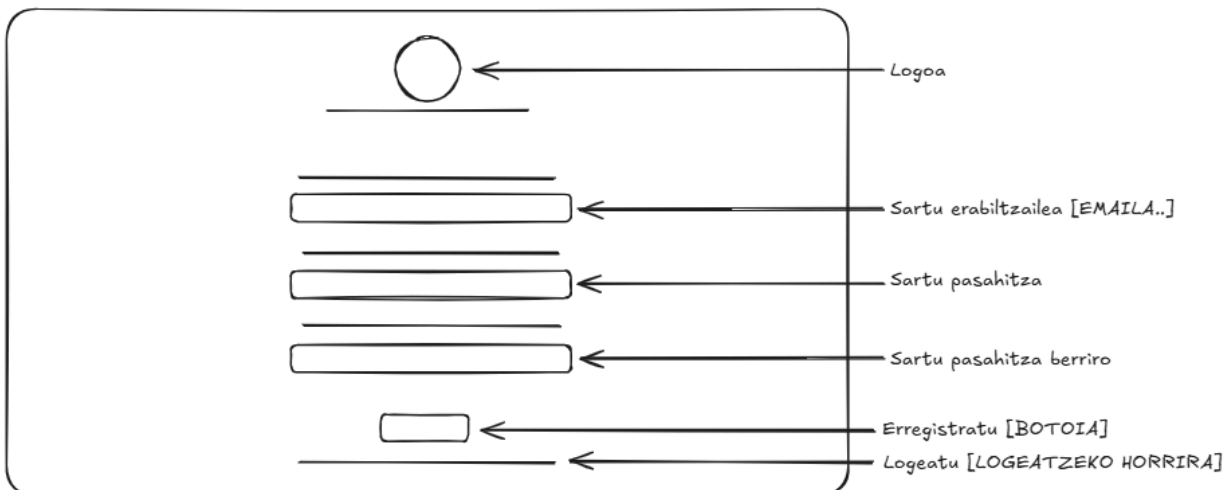
Web plataforma **PHP eta HTML** erabiliz garatuko da, **egitura sinplea eta erabilerraztasuna bermatzeko**.

## Webgune Estruktura Irudia [LOGIN / ERREGISTRATU]

### 1. LOGIN [GENERALA]

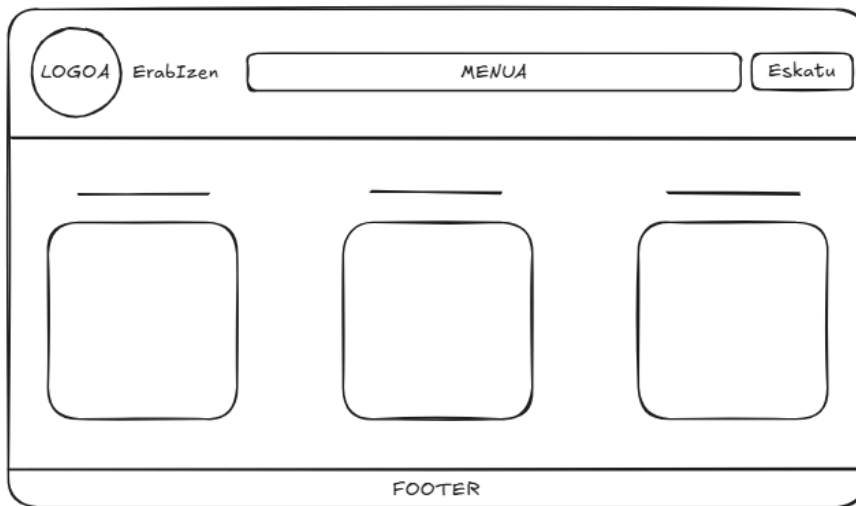


### 1.1 ERREGISTRATU [BEZEROAK]

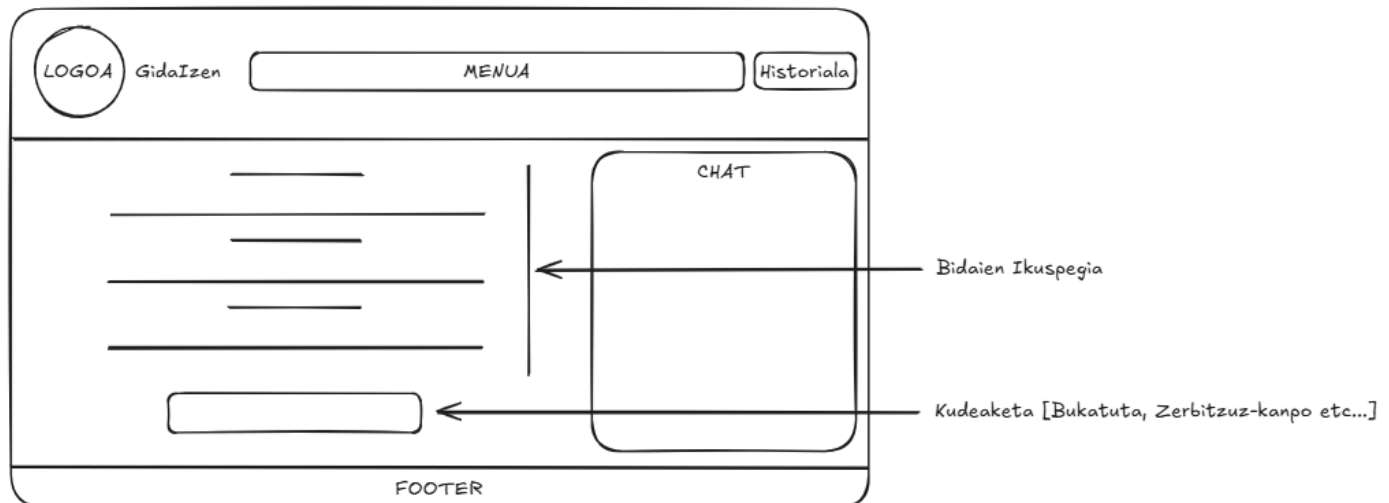


## Webgune Estruktura Irudia [BEZERO / GIDARI] PANELAK

2. WEB IDEIA [BEZEROAK]



2. WEB IDEIA [GIDARIAK]



## 5.6 Administrazioaren Aplikazioa

Administrazioaren aplikazioa Java bidez garatuko da eta honako funtzioak izango ditu:

- **Gidarien kontuak sortu, ezabatu eta editatu.**
- **Gidariaren zerbitzu historikoa ikusi.**
- **Bidai eskaerak ikusi eta esleipenak egin.**
- **Gidariaren egoera erregistratu** (zerbitzua aktibo, zerbitzutik kanpo, okupatuta...).

Administrazioentzako sistema **bereizita egotea beharrezkoa da**, haien funtzioak **beste erabiltzaile moten desberdinak direlako**, eta kudeaketa segurtasunez egin behar delako.

## 5.7 Justifikazioa eta Egitura Teknikoaren Arrazoiak

Proiektu hau moduluetan banatu da, honako arrazoi teknikoak direla eta:

- **Erabiltzaile moten arabera behar ezberdinak daudelako:** bezeroak, gidariak eta administrazioak interfaze ezberdinak behar dituzte.
- **Segurtasuna eta egonkortasuna bermatzeko:** Docker bidez egituratzeak sistemaren segurtasuna areagotzen du.
- **Talde-lana erraztu eta kontrolatzeko:** GitHub bidez garapena koordinatu eta aldaketak kontrolatu daitezke.
- **Mantentze-lanak eta eskalagarritasuna optimizatzeko:** Java, PHP eta SQL tresnak erabiliz, sistema egonkorra eta luzerako eskalagarria izango da.

## 6. PLANIFIKAZIOA

Gure helburua da proiektu hau AlaiktoMUGI enpresaren beharretara egokitzea, eraginkortasuna, eskalagarritasuna eta kalitatea bermatuz. Horretarako, **hiru fase nagusi** definitu ditugu, bakoitza lan espezifikoekin eta helburu argiekin.

### 6.1 Lehen fasea: Planifikazioa

Proiektuaren hasieran, zuekin bildu gara, zuen behar teknikoak eta helburuak zehazteko. Bilera honetan, sistemaren funtzionalitateak identifikatu ditugu, zuen eskakizunak beteko dituen plataforma bat garatzeko.

Helburua izan da taxi zerbitzuaren kudeaketa digitalizatzea, prozesuak automatizatuz eta errazteko, administratzaileentzat, gidarientzat eta bezeroentzat.

Planifikazio fasean:

- **Zuen eskakizunak** aztertu ditugu, behar teknikoak eta erabiltzaileen esperientzia optimizatzeko.
- **Egituraketa teknikoa** definitu dugu, sistemaren arkitektura, datu-basearen diseinua eta interfaze grafikoaren funtzionamendua erabakita.
- **Plataforma teknologikoak** aukeratu ditugu (Java, PHP, SQL Workbench, Docker...), **segurtasuna, erabilgarritasuna eta errendimendua hobetzeko.**
- **Segurtasun-neurriak ezarri ditugu**, datuen babesa eta transakzioen egonkortasuna bermatzeko.

Planifikazio fasea amaitu ondoren, garapen-prozesura igaro gara, diseinatutako egituraren arabera.

## 6.2 Bigarren fasea: Garapena eta Exekuzioa

Fase hau proiektuaren hurrengo urratsa izango da, baina lehenik eta behin, AlaiktoMUGI enpresaren feedback-a jaso behar dugu eta dokumentuaren onarpena lortu behar dugu. Behin proposamena onartuta, garapenean sartuko gara eta sistemaren ezarpen teknikoa hasiko dugu.

### Sistemaren egituraketa

Prozesua **fase ezberdinetan** banatuko dugu, garapena modu koordinatuan gauzatzeko:

1. **Eskemak eta dokumentazioa garatuko ditugu**, sistemaren osagai guztiak argi ulertzeko.
2. **Datu-basearen egitura sortuko dugu**, SQL Workbench erabiliz.
3. **Aplikazioen garapena abiaraziko dugu**:
  - Webgunea PHP eta HTML erabiliz garatuko dugu, bezeroek eta gidariek zerbitzua erabiltzeko.
  - Administratzailearen aplikazioa Java bidez sortuko dugu, erreserbak eta gidarien kontrola kudeatzeko.
  - API bat garatuko dugu, plataforma ezberdinen arteko komunikazioa bermatzeko.
4. **Docker kontenedoreak konfiguratuko ditugu**, zerbitzariak eskalagarria eta independente mantentzeko.

### Segurtasuna eta datuen kudeaketa

Sistema guztiz funtzionala eta **segurua** izan dadin, hainbat babes-neurri ezarriko ditugu:

- **Erabiltzaileen autentifikazioa ezarriko dugu**, pasahitz enkriptazio mekanismoekin.
- **Datu-basean transakzioen kontrolak eta indizeak optimizatuko ditugu**, datuen osotasuna bermatzeko.
- **Administratzailearen sarbidea mugatuko dugu**, segurtasun handiagoa izateko.

Behin garapena amaitzen dugunean, **sistemak bere funtzionalitate guztiak garatu dituela ziurtatuko dugu** eta hurrengo urratsa izango da azken proba eta optimizazio fasea.

## 6.3 Hirugarren fasea: Abiaraztea eta Azken Ukituak

Azken fasean, **sistema martxan jartzeko lan egingo dugu**, egonkortasuna eta kalitate teknikoa bermatzeko.

### Proba fasea

Sistemak behar bezala funtzionatzen duela ziurtatzeko, honako testak burutuko ditugu:

- **Erabiltzaileen interfazea testatuko dugu**, erraztasuna eta erabilgarritasuna baloratzeko.
- **Zerbitzariaren errendimendua neurtuko dugu**, Docker kontenedoreen funtzionamendu egokia bermatzeko.
- **SQL kontsulten eraginkortasuna egiaztatuko dugu**, datuen osotasuna mantentzeko.

### Zerbitzuaren abiaraztea eta sustapena

Bukatutakoan, sistemaren instalazioa eta abiaraztea egingo dugu:

- **Web aplikazioa eta administratzailearen sistema instalatuko dira**, erabiltzaileek zerbitzu berria probatzeko.
- **Dokumentazio teknikoa aurkeztuko dugu**, sistemaren mantentzea eta eguneraketak errazteko.
- **Egonkortasun testak burutuko ditugu**, azken akatsak identifikatzeko eta zuzentzeko.

Behin prozesua amaituta, **sistemaren optimizazio teknikoa zuen eskakizunen arabera egokituko dugu**, azken bertsioan beharrezko doikuntzak eginez.