

TXOSTENA

1- Erronka | 2PAAG3

2023

AURKIBIDEA

[ERRONKAREN TXOSTENA 2](#_Toc150928588)

[**EMPRESA KUDEAKETA SISTEMA** 2](#_Toc150928589)

[Odooren instalazioa eta konfigurazioa: 2](#_Toc150928590)

[**MULTIMEDIA PROGRAMAZIOA ETA GAILU MUGIKORRAK** 5](#_Toc150928591)

[Aplikazioaren diseinua pentsatu: 5](#_Toc150928592)

[Aplikazioaren login pantalaila: 5](#_Toc150928593)

[Aplikazioaren menu: 5](#_Toc150928594)

[Aplikazioak dituen leiho desberdinak: 6](#_Toc150928595)

[Datu baseko konexioa: 8](#_Toc150928596)

[Datu base lokaleko sarrerak: 9](#_Toc150928597)

[**INTERFAZE GARAPENA** 11](#_Toc150928598)

[Aplikazioaren hasiera: 11](#_Toc150928599)

[Datu baseko konexioak: 11](#_Toc150928600)

[Datuen gestioa: 12](#_Toc150928601)

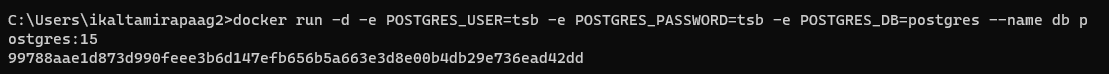
[Aplikazioak dituen aukerak: 13](#_Toc150928602)

# ERRONKAREN TXOSTENA

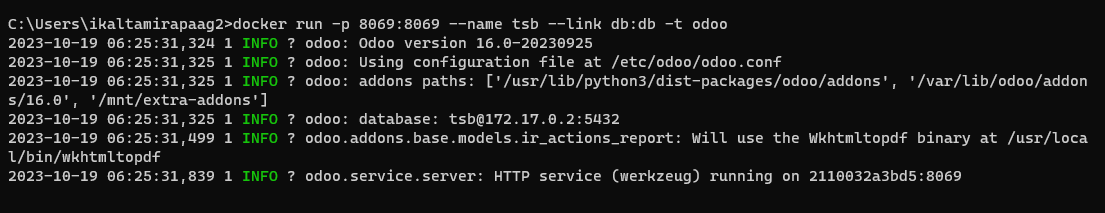
## **EMPRESA KUDEAKETA SISTEMA**

### Odooren instalazioa eta konfigurazioa:

* Odoo-ren instalazioarekin hasteko Docker Desktop instalatu dugu ekipoan. Ondoren, Docker Hub erabilita behar ditugun komanduak berreskuratu eta exekutatu ditugu.
* Lehenengo, datu basea listo jarriko dugu, horretarako hurrengo komandoa exekutatuko dugu “CMD” baten, gure kasuan izena eta pasahitza aldatu dugu.



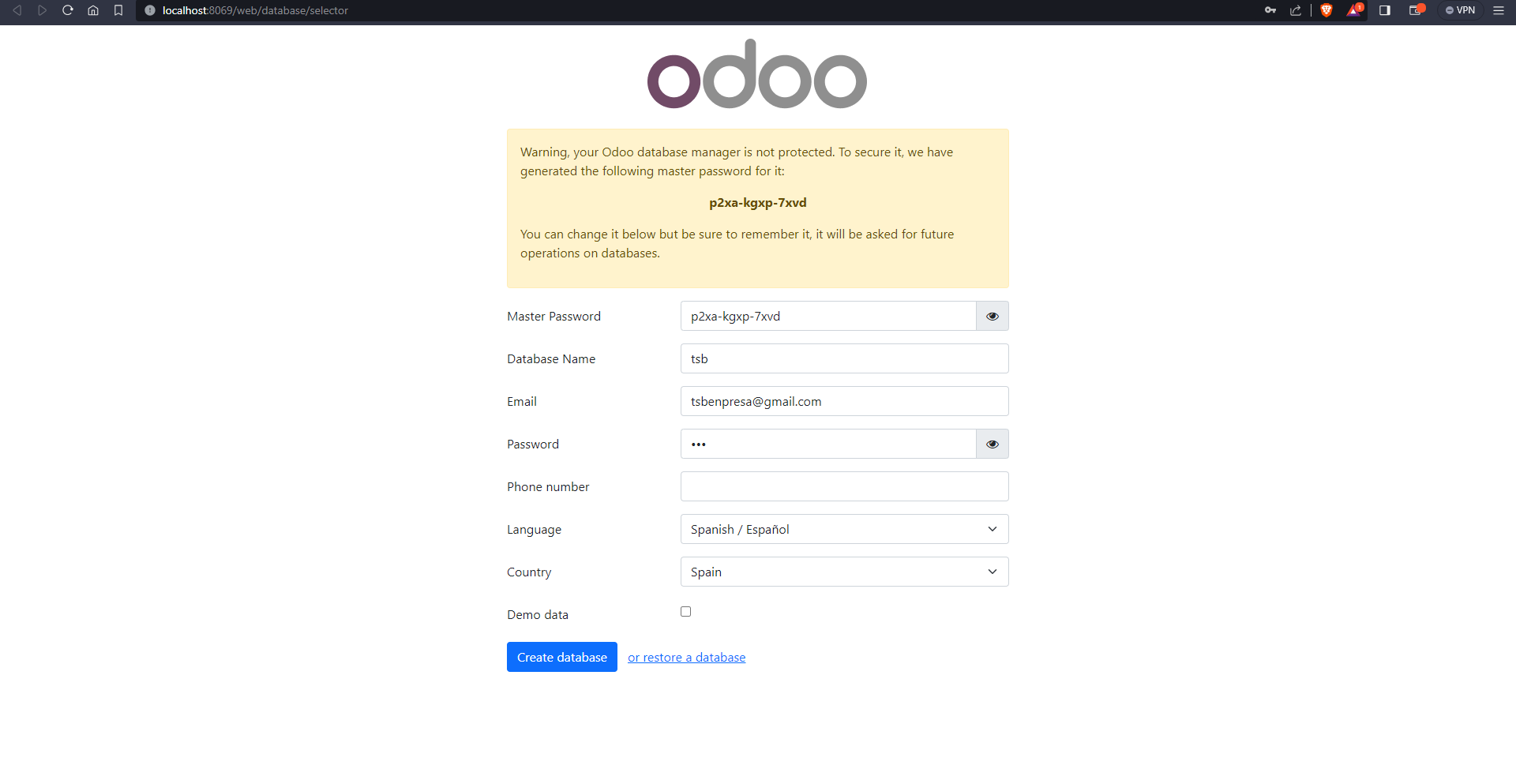
* Ondoren, Odoo erp-a jarriko dugu martxa, hurrengo komandoarekin.



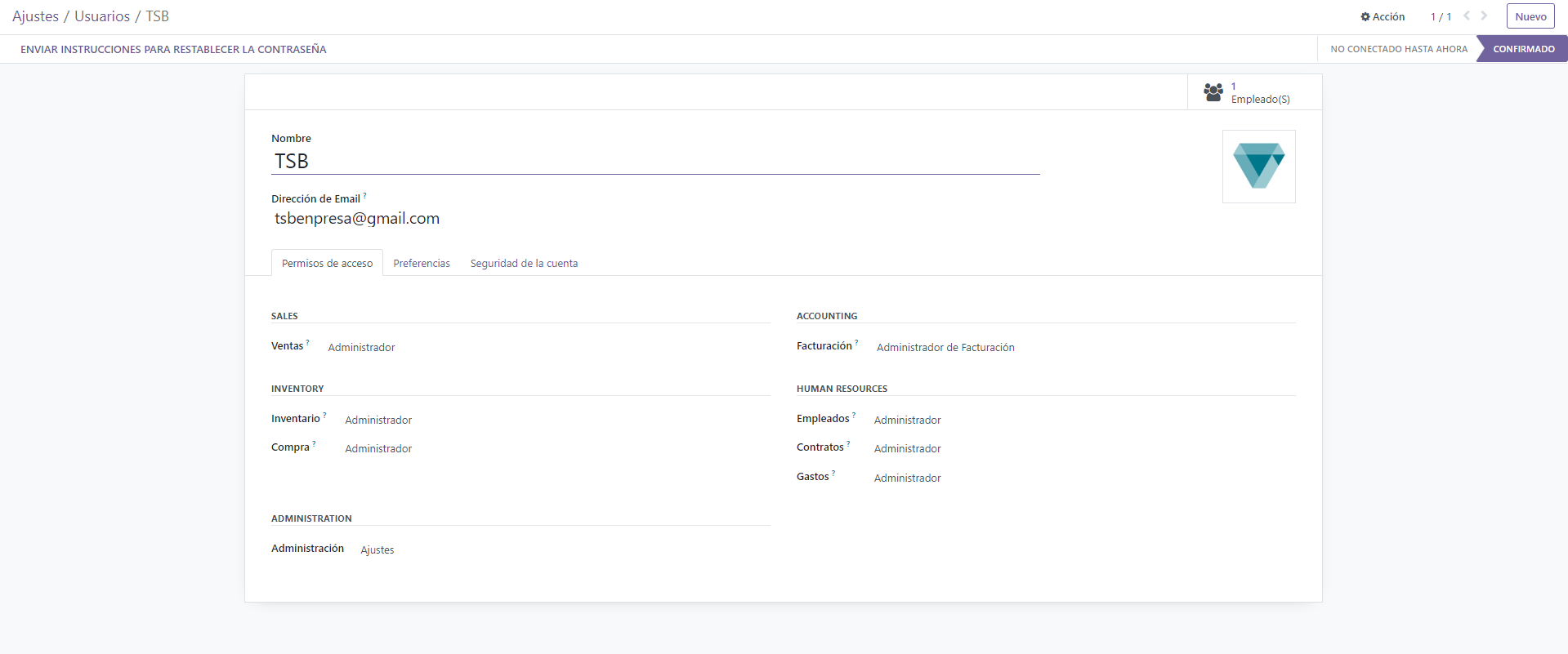
* Instalazioarekin bukatzeko, Docker-a horrela geratuko da.



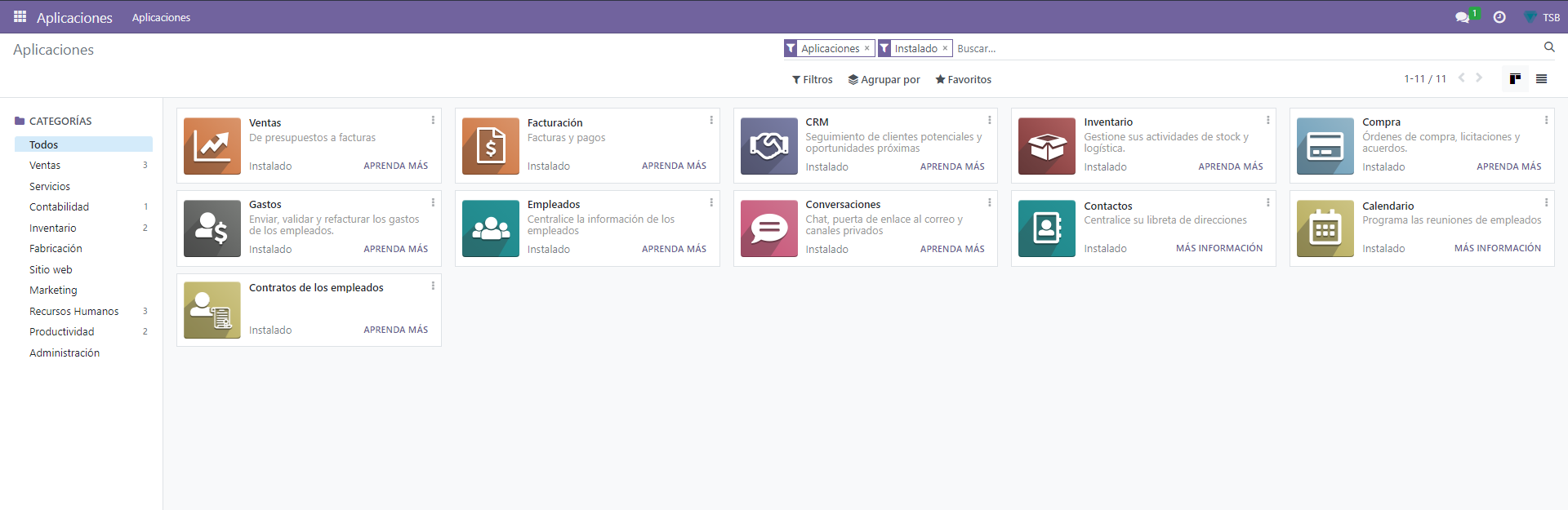
* Instalazioa atzean utzita, orain Odoo-ra sartuko gara, beraren lehenengo konfigurazioa egiteko. Gure kasuan, gure datuak bete eta demo database aukeratu gabe utzi dugu (guk geure datu basea erabiliko dugu).



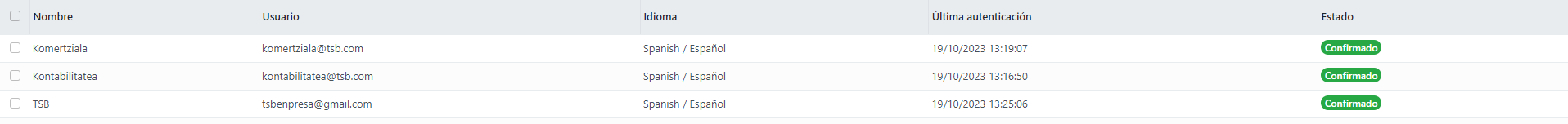
* Behin, barruan egonda, erabiltzailearen izena aldatu dugu, gure enpresaren izena mantentzeko administradore giza.



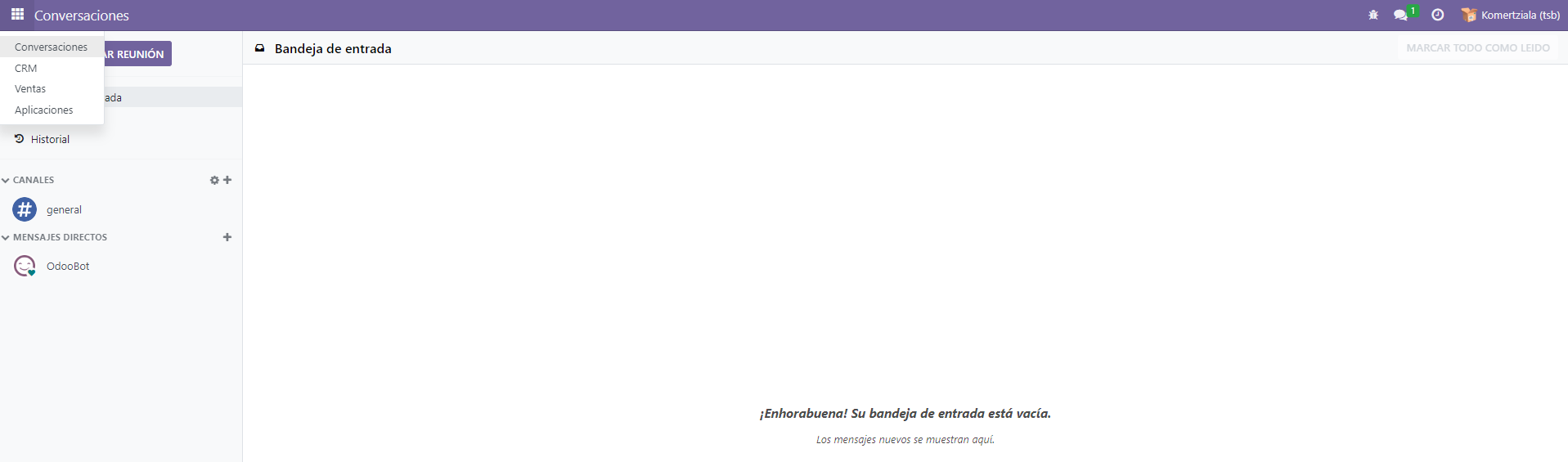
* Ondoren, beharrezko moduluen instalazioa egin dugu, hemen ditugun moduluen kaptura.



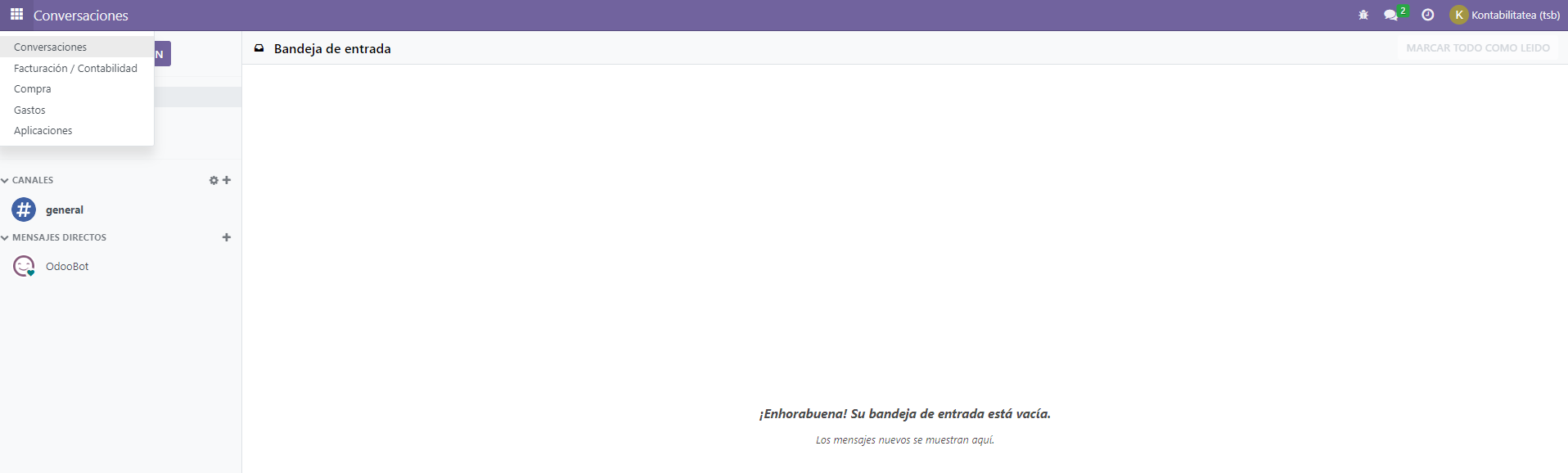
* Bukatzeko, erabiltzaileak sortu eta beraien pantailak/baimenak konfiguratuko ditugu.
* Horretarako, “komertzial” eta “kontabilitatea” erabiltzaileak sortu ditugu. Beraien pantailak konfiguratzeko, modulo batekin eta baimenekin egin dugu.



* Hemen nola geratu den “KOMERTZIALA” erabiltzailearen pantaila



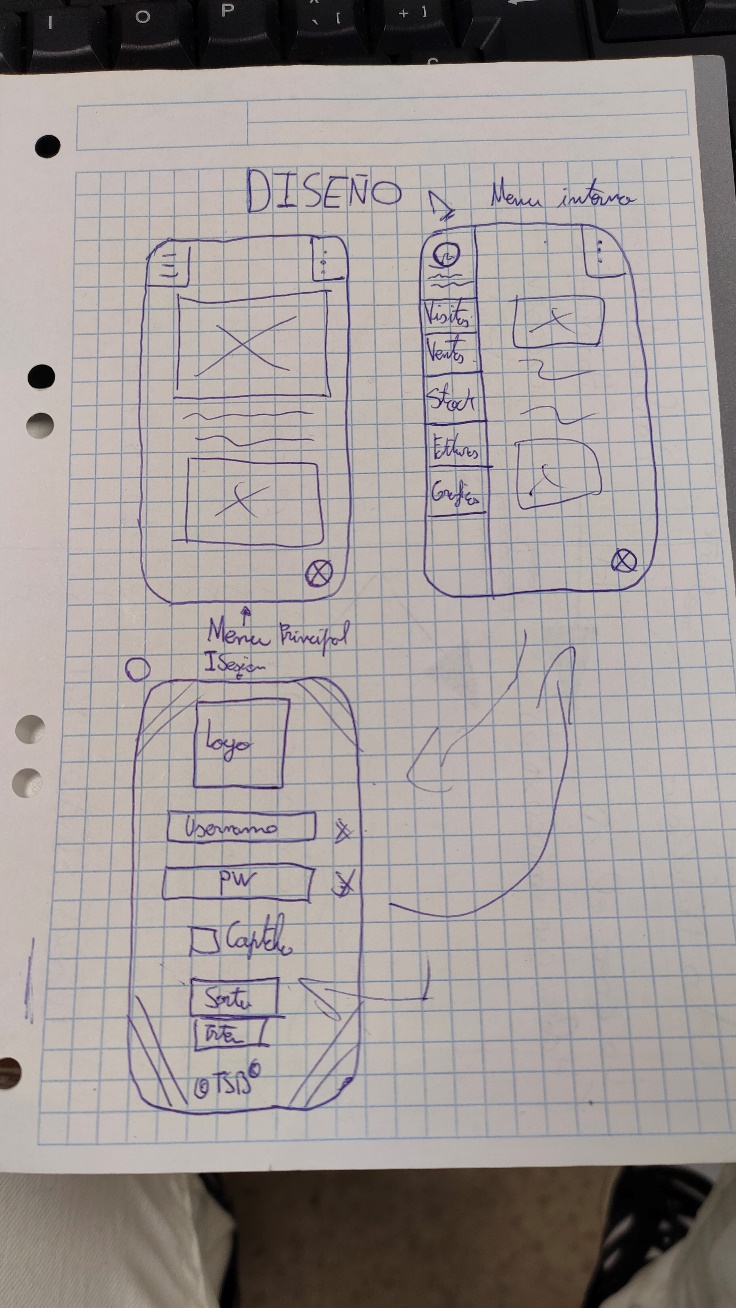
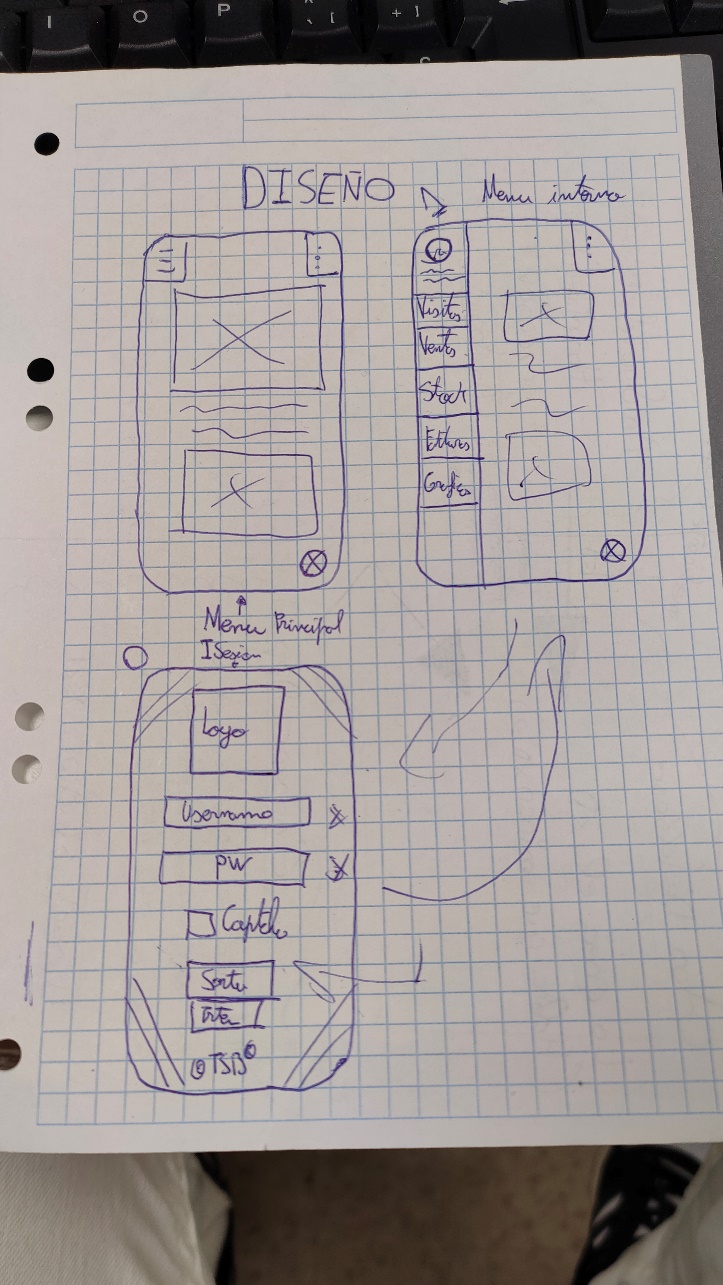
* Amaitzeko, “KONTABILITATEA” erabiltzailearen pantaila.



## **MULTIMEDIA PROGRAMAZIOA ETA GAILU MUGIKORRAK**

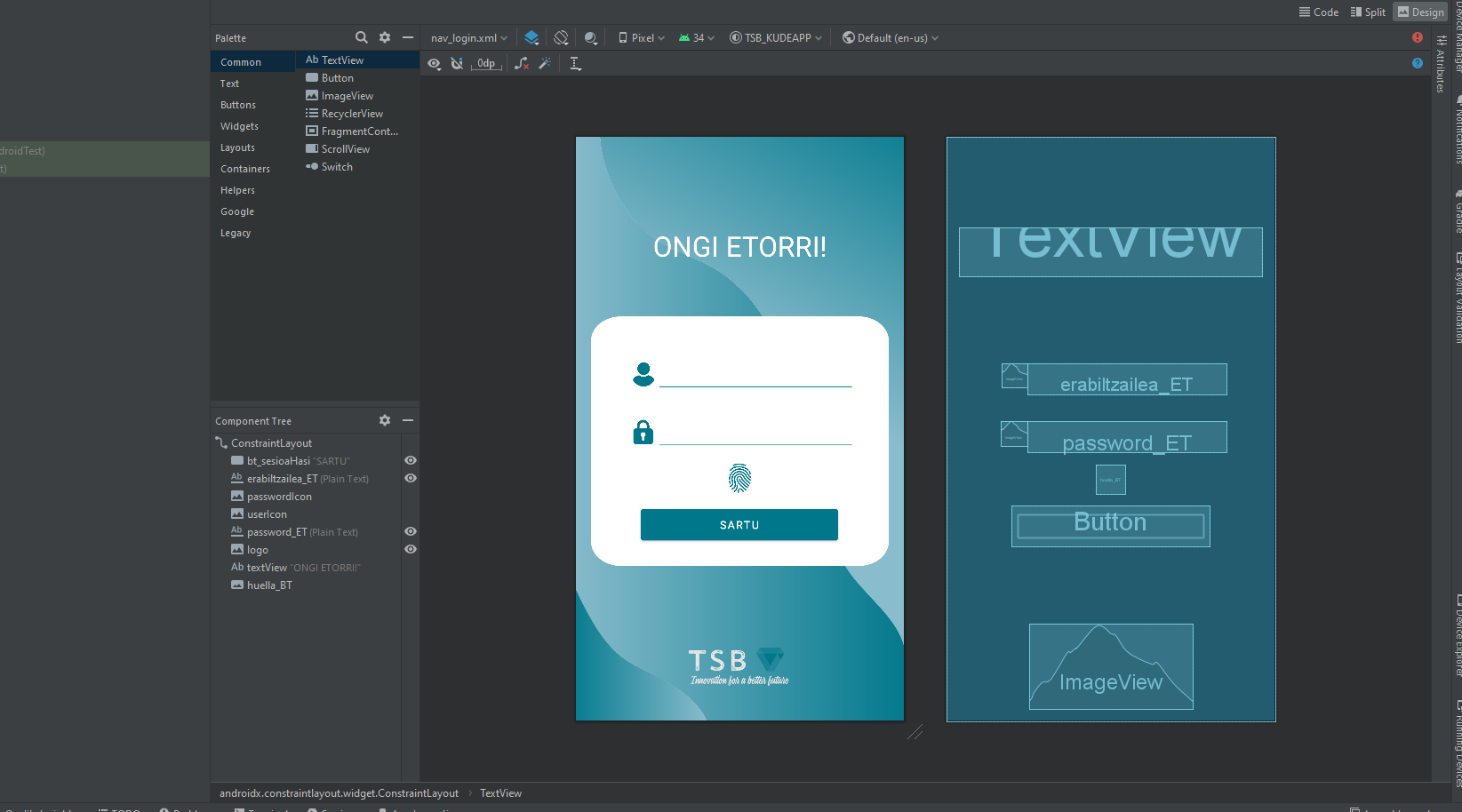
### Aplikazioaren diseinua pentsatu:

* Lehenengo pausua, aplikazioaren diseinua pentsatzea egin dugu, hau da guk pentsatutako diseinua.



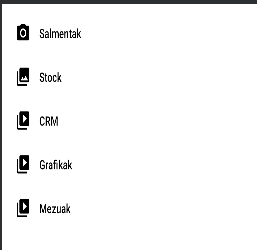
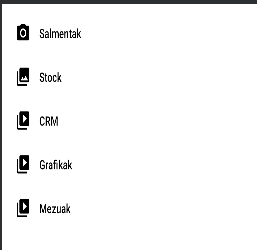
### Aplikazioaren login pantalaila:

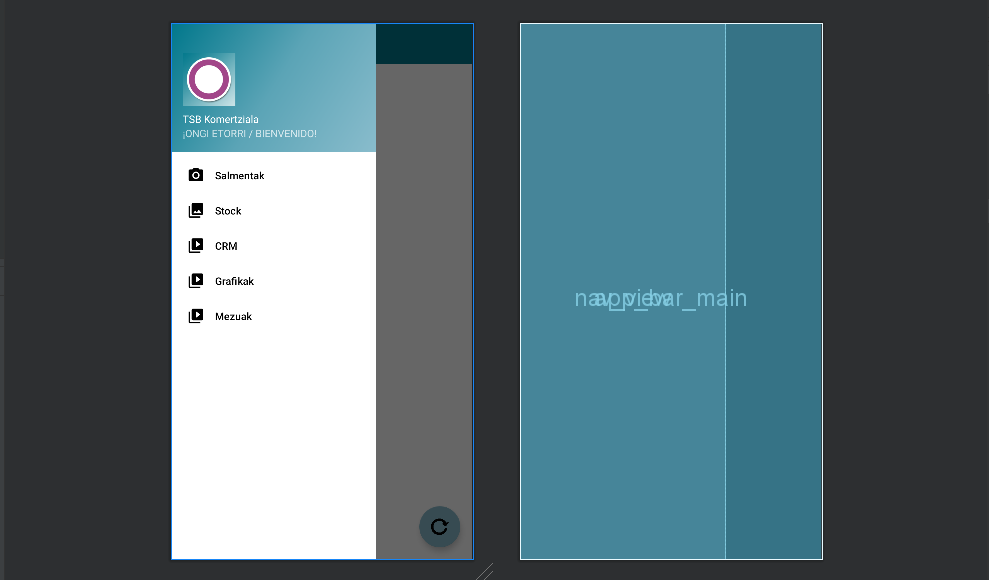
* Lehenengo, pentsatutako diseinua egin dugu. Azkenean gure diseinua horrela geratu da, beraren funtzioekin (hatz marka, user, password…)



### Aplikazioaren menu:

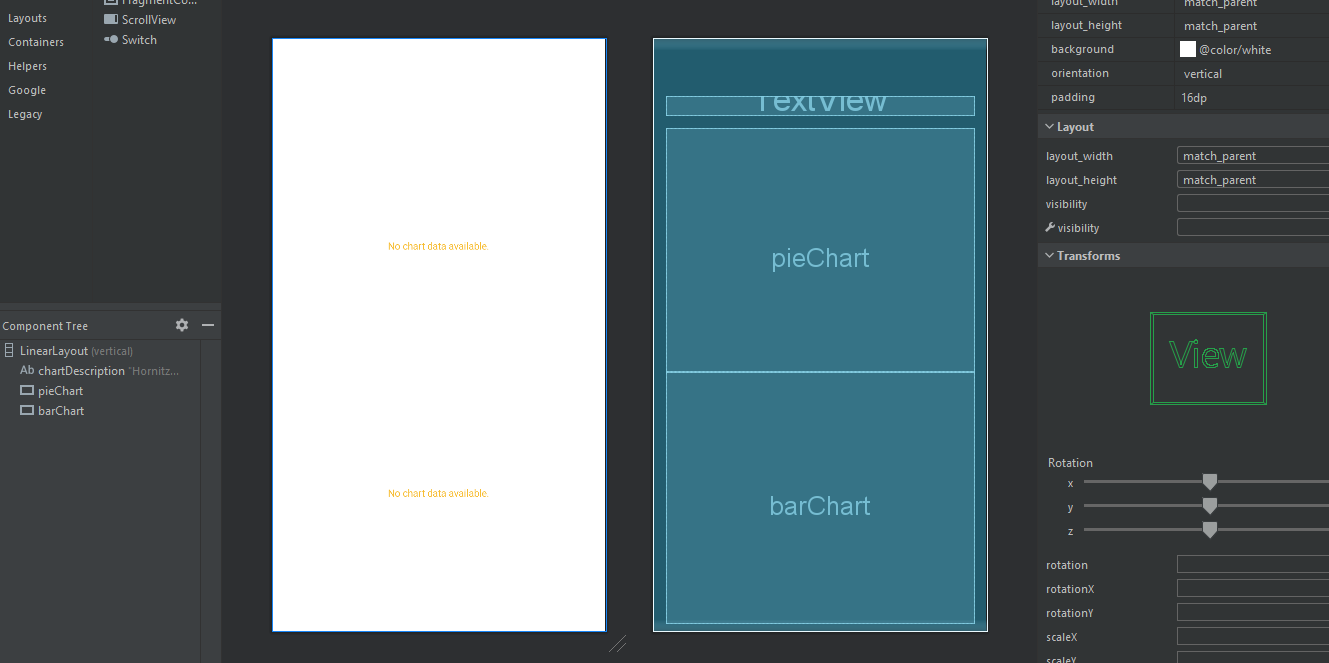
* Menuarren barruan izenak aldatu eta bi aukera berri sartu ditugu. Hau da aplikazioari utzi diogun menua. Salmentak, Stock, CRM, Grafikak eta Hornitzaileak.



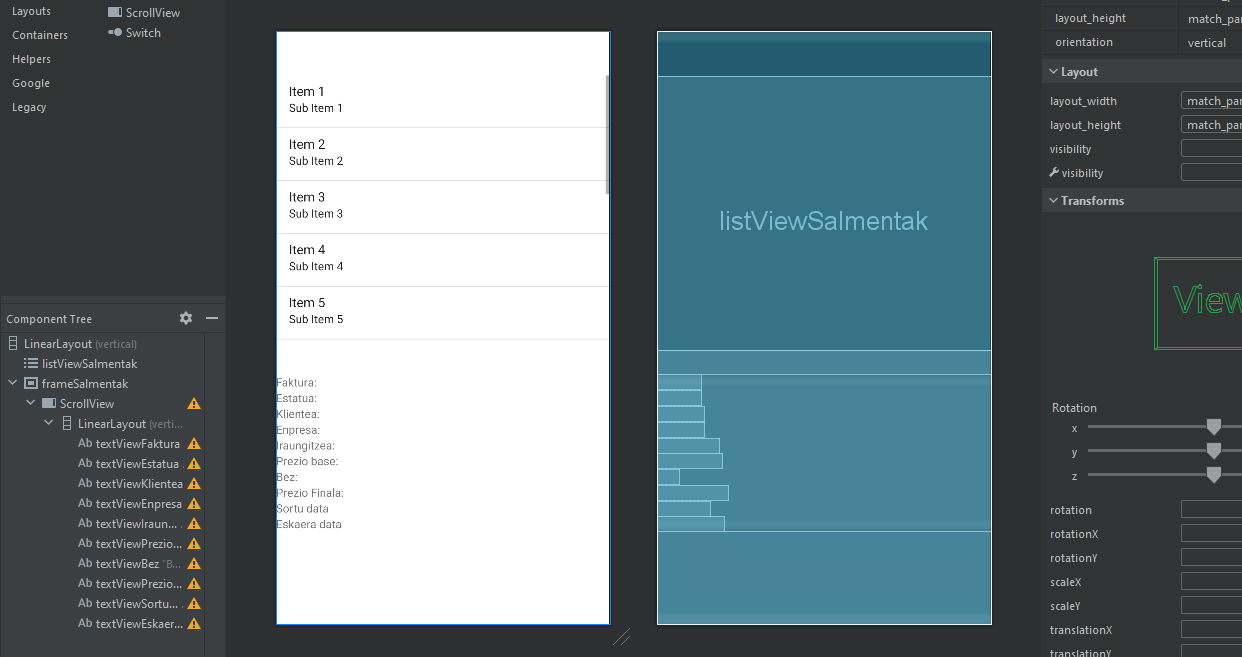


### Aplikazioak dituen leiho desberdinak:

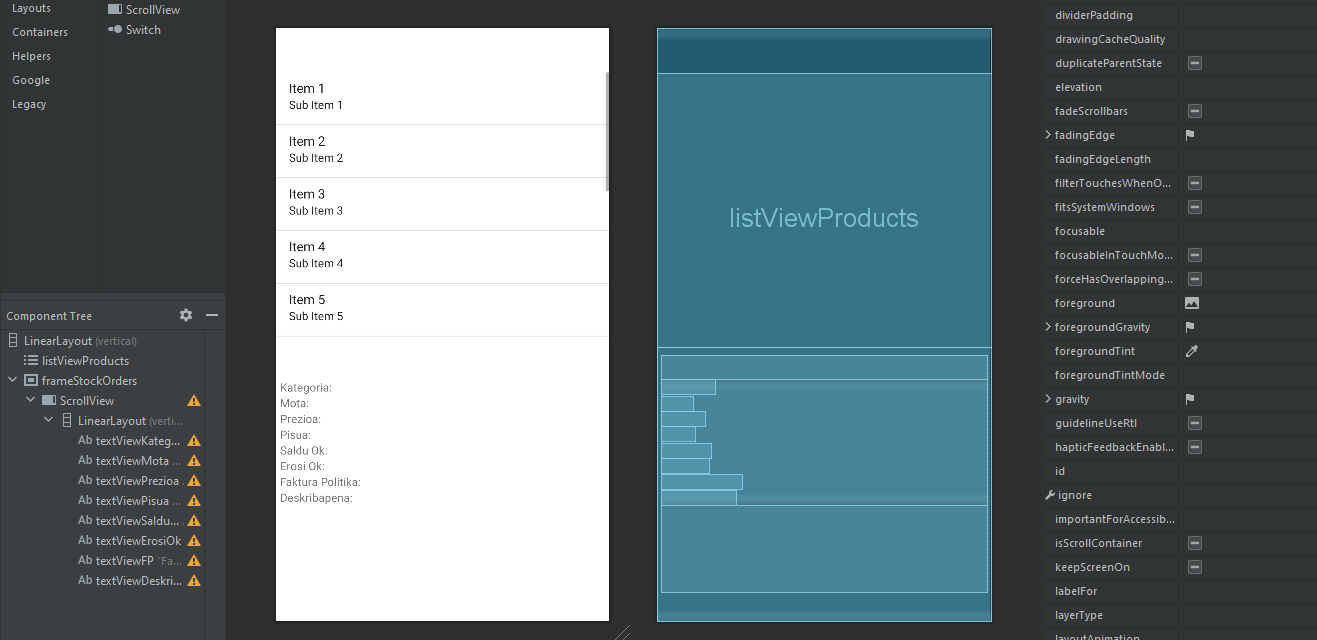
* Gure aplikazioak hainbat leihoa desberdin ditu martxan, adibidez, hainbat grafiko bistaratu ahal izateko leiho bat.
* Grafiko leihoa:



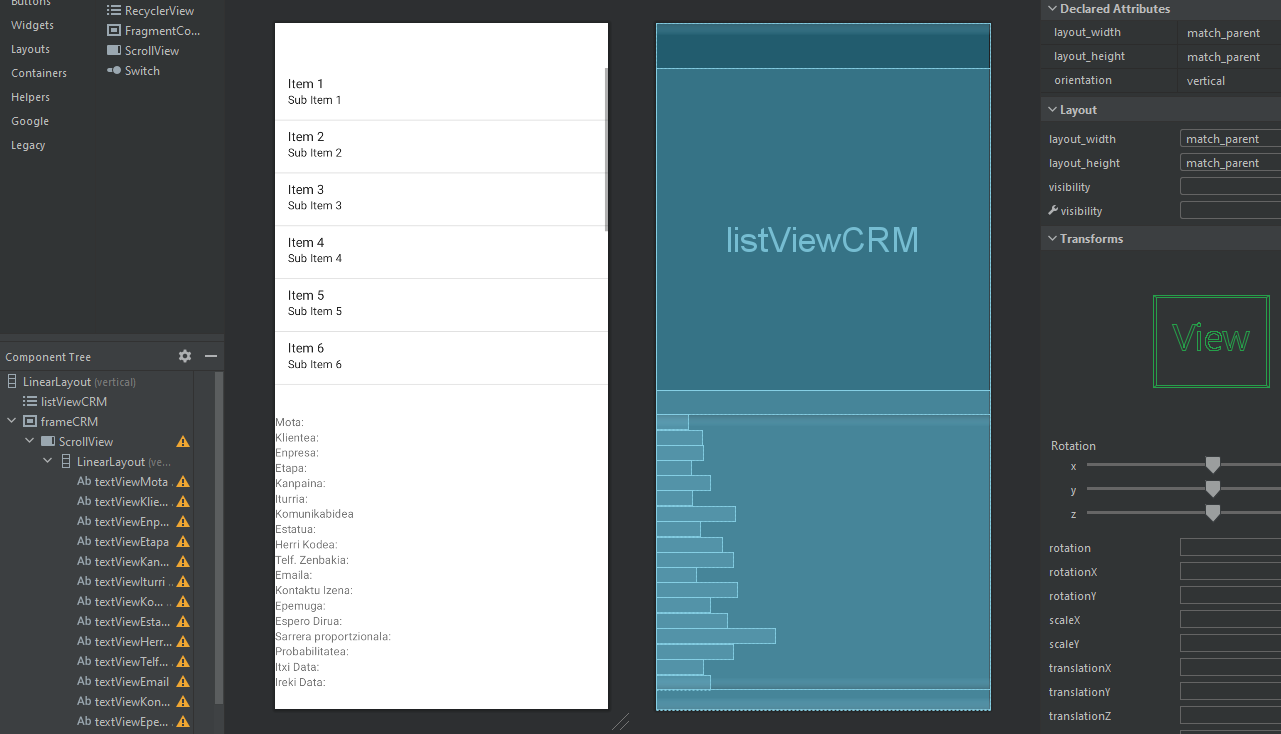
* Hurrengo leihoak, erabiltzaileak/kontableak datuak bistaratu ahal izateko egina dago: Salmentak, Stock, CRM eta Hornitzaileak.
* Salmentak:



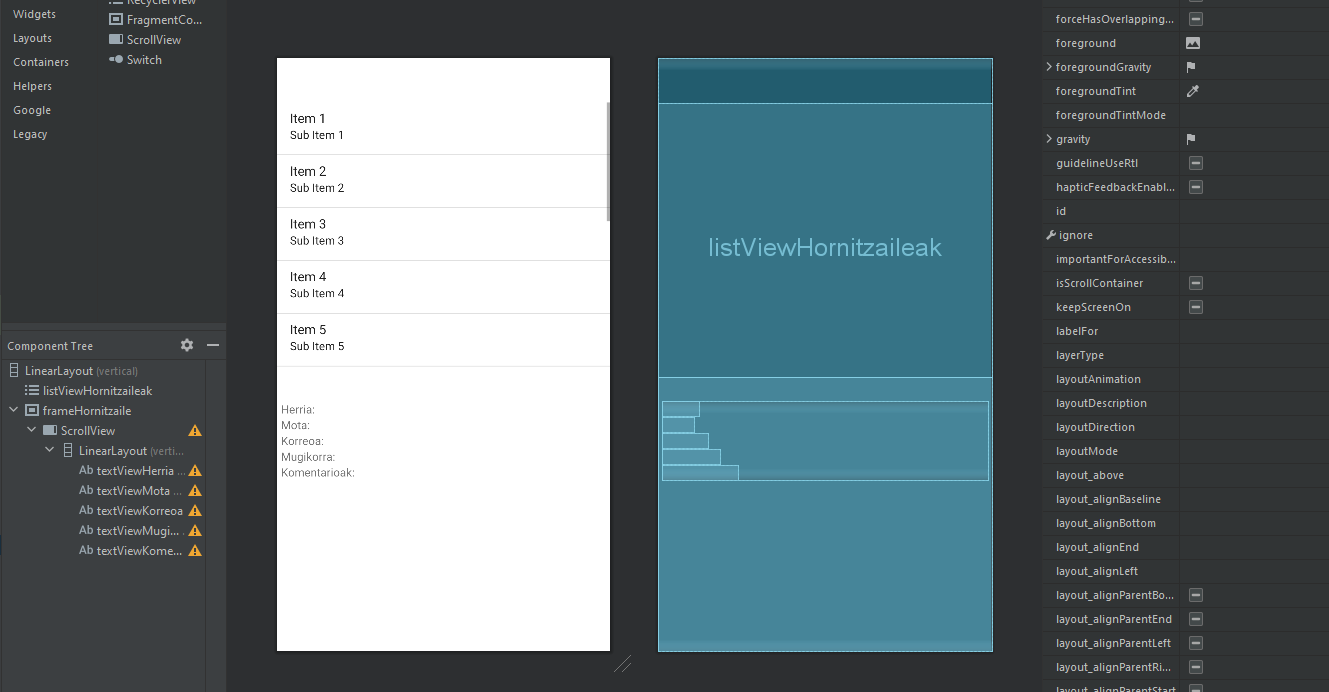
* Stock:



* CRM:

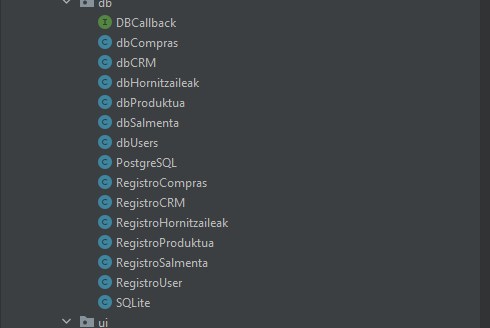


* Hornitzaileak:



### Datu baseko konexioa:

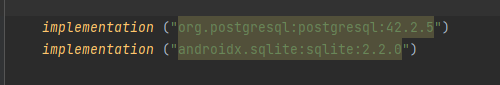
* Datu basera konexioa ondo egiteko, hainbat klase sortu ditugu. Horietako bat “PostgreSQLConnection” da. Horrek egiten du konexioaren hasiera eta bukaera.
* Gero, hainbat klase ditugu, bat lokalean ibiltzeko datu basearen taulak sortzen dituena eta bestea datuak sartzen dituena barruan.



* Konexioaren irikiera egiteko orduan, problemak izan ditugu, baina hariaren erabilerarekin konpondu dugu.

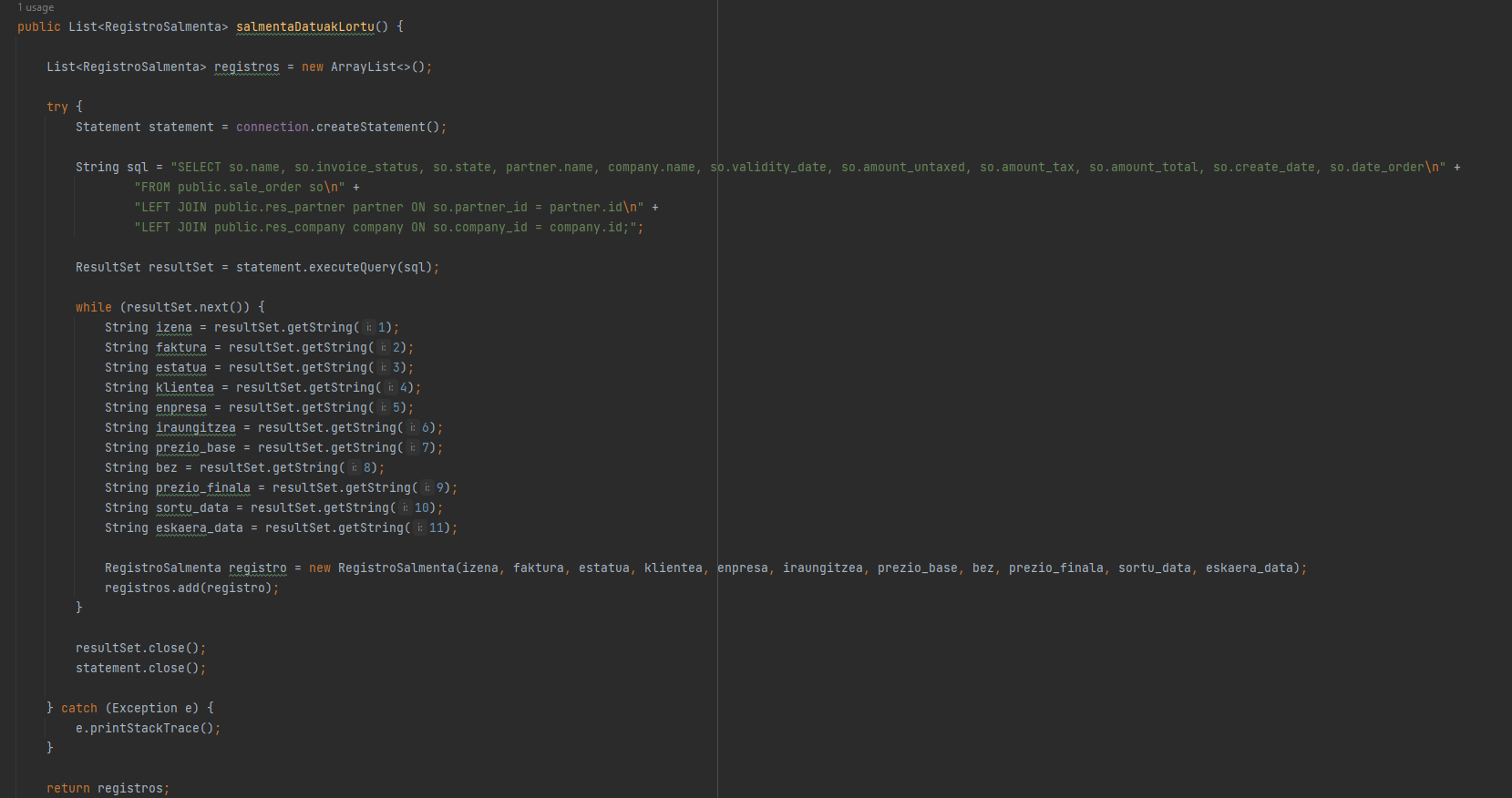


* Aurreko danak egiteko, bi “implement” erabili behar izan ditugu, bat PostgreSQL-rako eta bestea, lokaleko datu basearentzako “SQLite”.

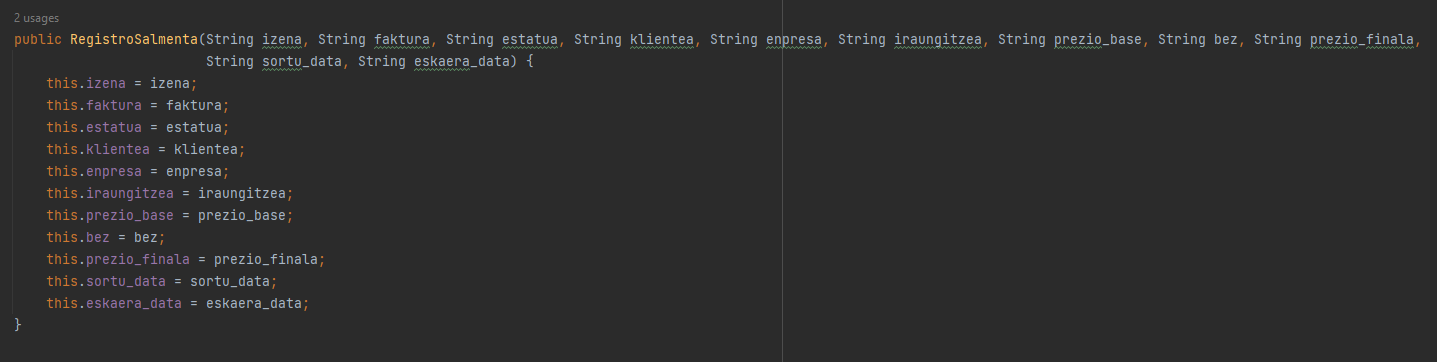


### Datu base lokaleko sarrerak:

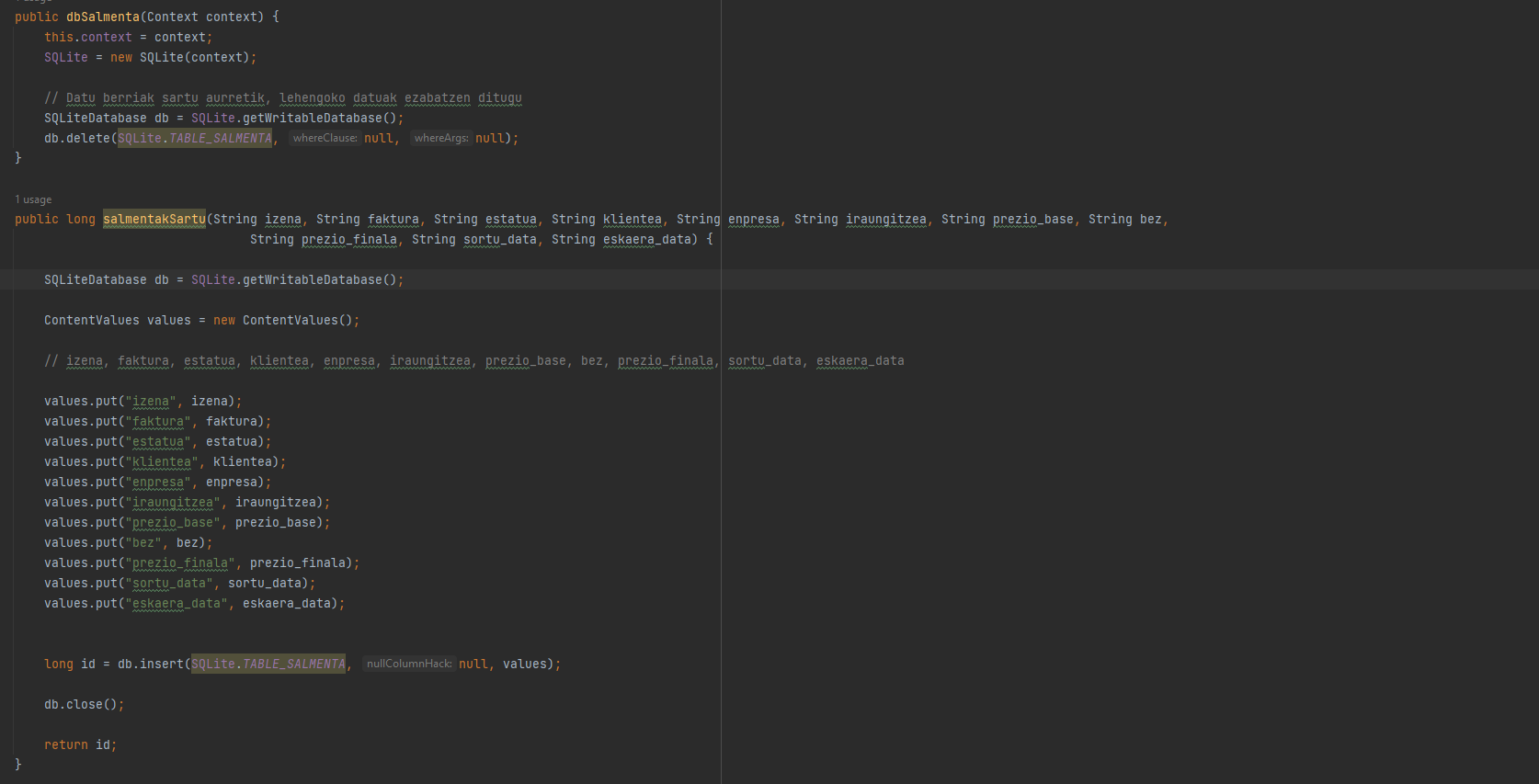
* Datu basera konexioa ondo egiteko, hainbat klase sortu ditugu. Adibidez, “Salmenta” taularen datuekin azalduko dugu.
* Lehenengo pausua datuak hartzea da, gure kasuan PostgreSQL-ko klase barruan hau dugu:



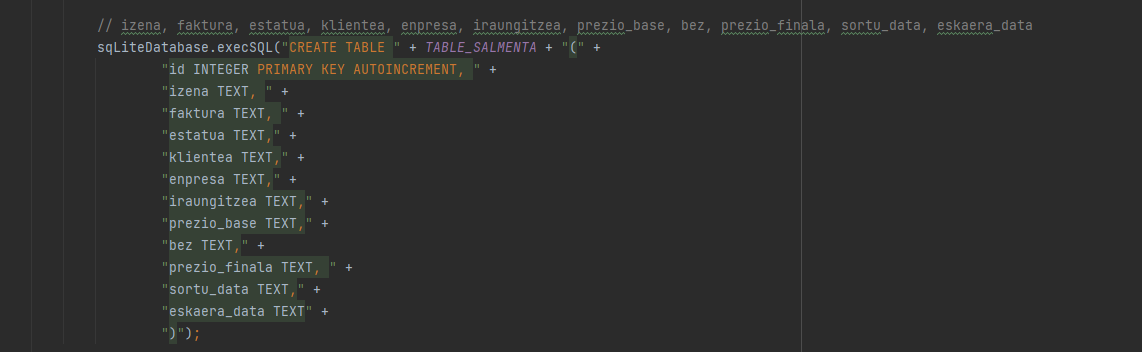
* Ondoren, Salmenta datuak lortuta ditugula, Salmenta objetu lista betetzen dugu Salmenta datuekin.



* Bukatzeko, SQLite datu base barruan sartzen ditugu datuak:



* Hau dana egin aurretik, SQLite barruan “Salmenta” taula sortuta izan behar duzu:

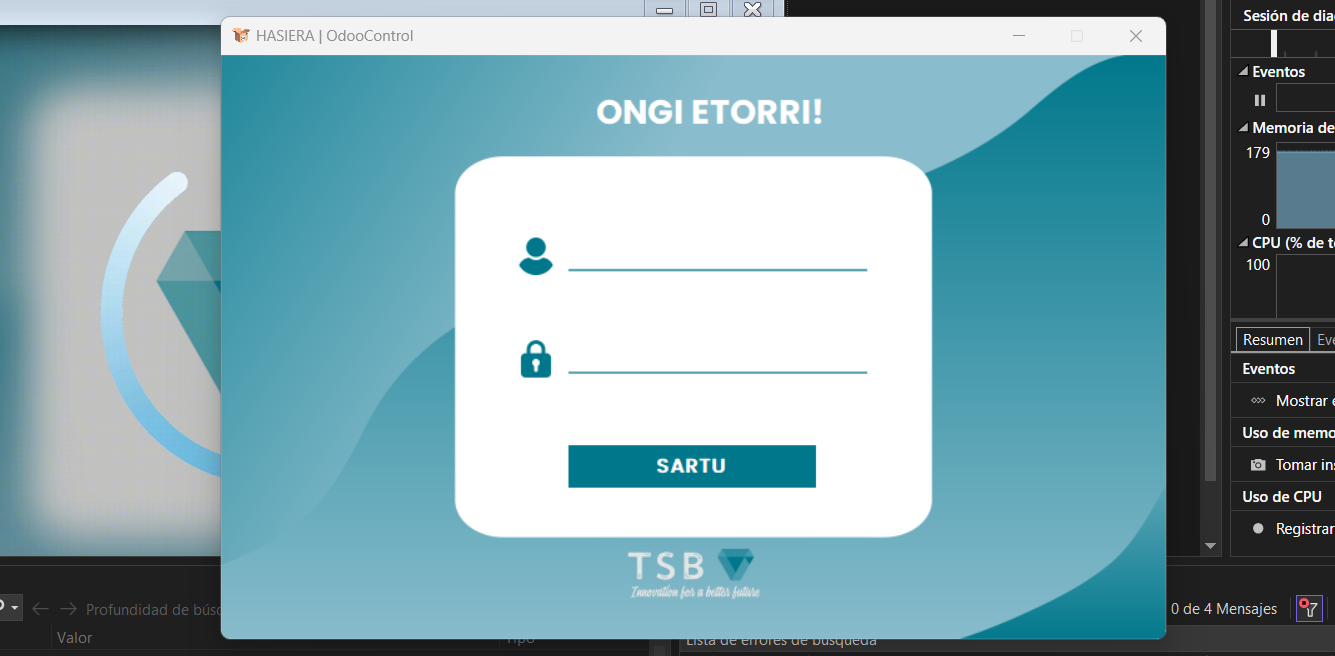


* Esan berra dao, hau dena, aktualizatu botoia sakatu eta PostgreSQLko datu basearekin konexioa duzunean, enpresa barruan egonda.

## **INTERFAZE GARAPENA**

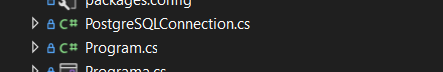
### Aplikazioaren hasiera:

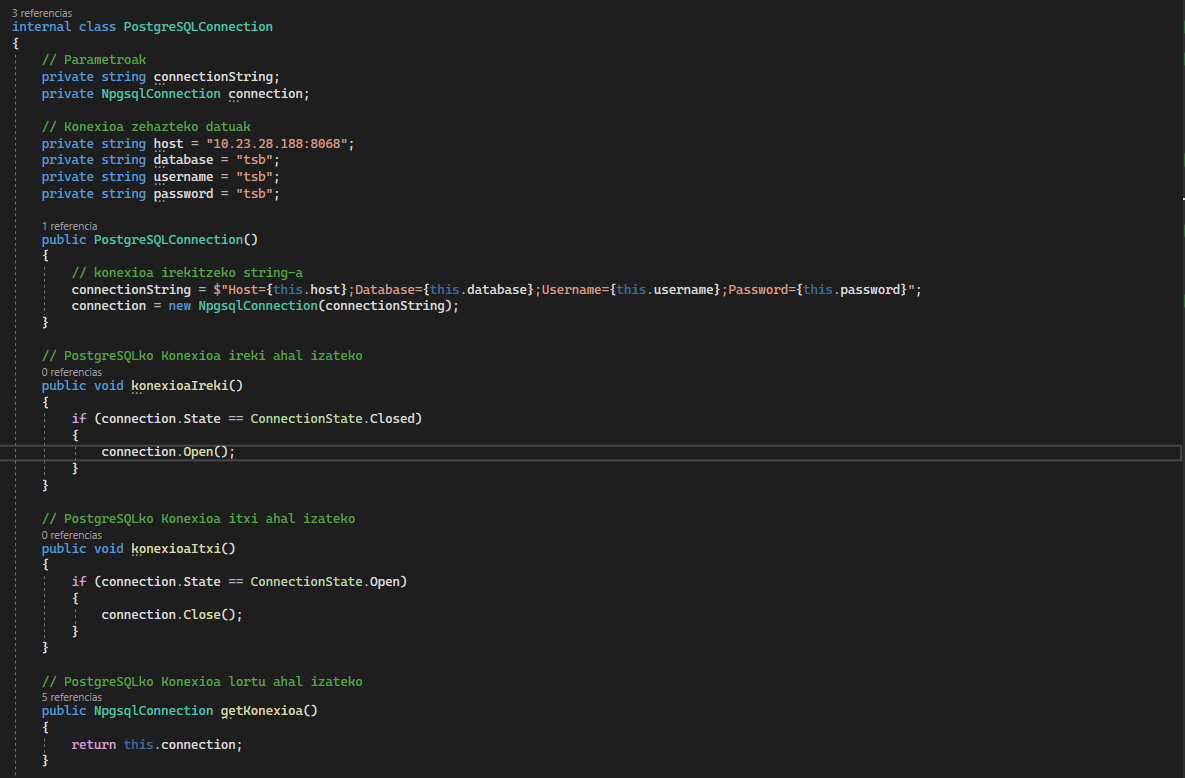
* Lehenengo, aplikazio osoaren diseinua pentsatu dugu.
* Ondoren, gure app-aren hasiera pantaila egin dugu Visual 2022 barruan.



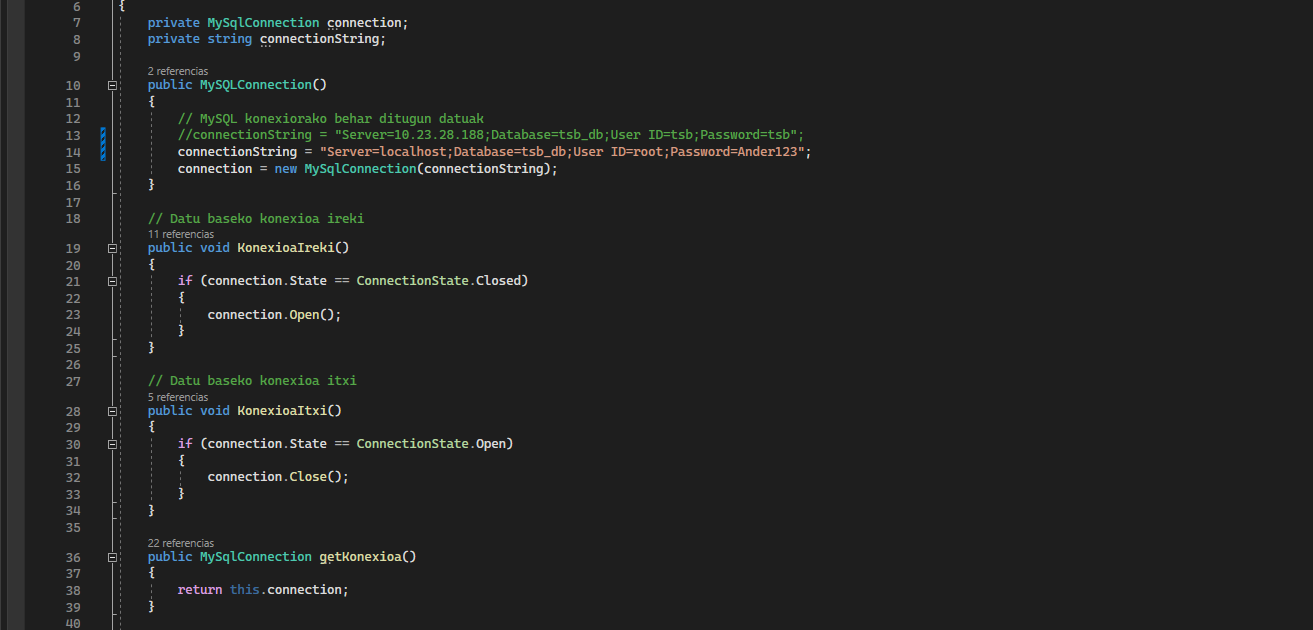
### Datu baseko konexioak:

* Hurrengo pausoa, datu baseko konexioa konfiguratzea izan da, bere klaseekin eta dena. Esan beharra dago, PostgreSQL erabili ahal izateko, Visual-ek errezten digun NuGet bat instalatu behar izan dugu proiektuan.





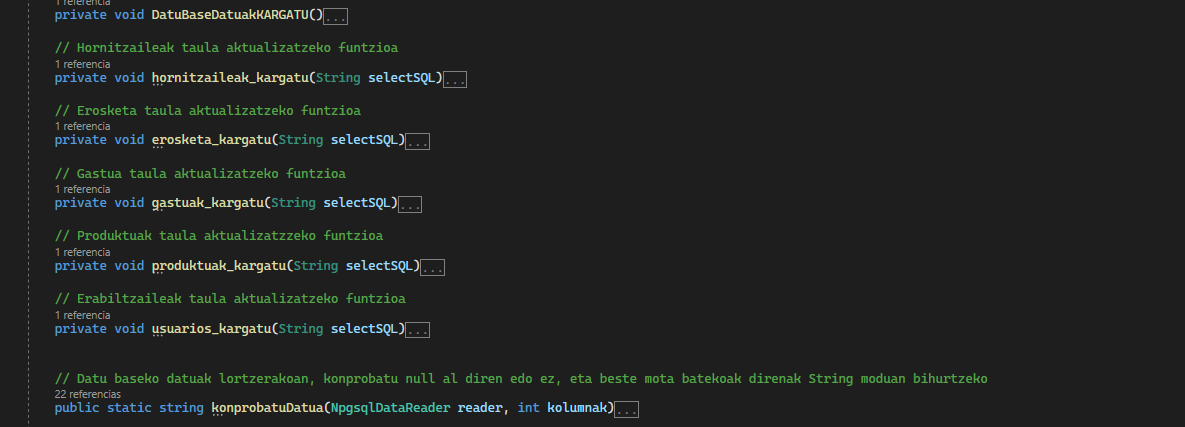
* Gure kasuan, datu base printzipala PostgreSQL esan dezakegu dela, baino aplikazioa bera, MySQL gainean ibiltzen da. Zergaitik, PostgreSQL bakarrik datuak jasotzeko erabiltzen dugu, beste aldetik, segurtasuna aldetik igopena lortzen dugu.
* Hau da gure MySQL datu basera konexioa egin ahal izateko klasea:



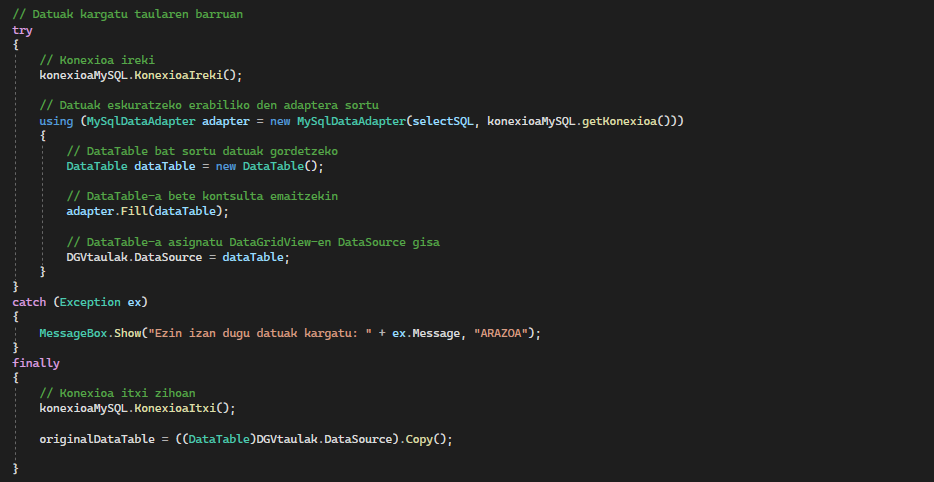
* Hori bai, PostgreSQLrekin pasatzen den moduan, MySQLko libreriak instalatu behar izan ditugu proiektuan, Visualek errezten digun “NuGet”-etatik.

### Datuen gestioa:

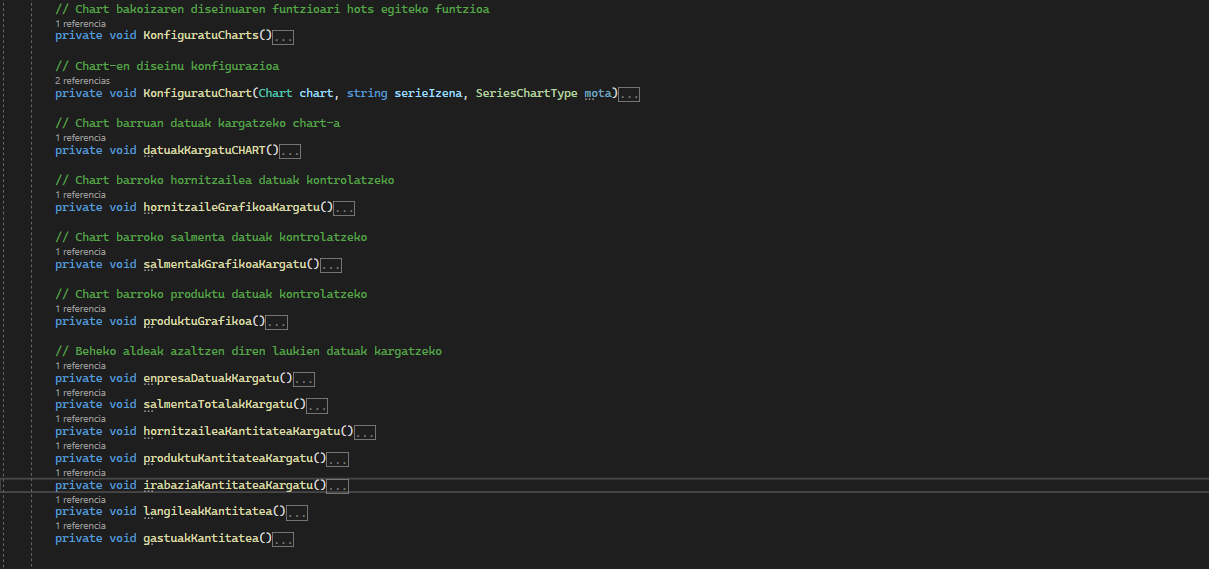
* Orain, behin datu baseko konexioa eginda edukita, beharrezko datuak hartu eta konexioa baldin ba dugu PostgreSQLarekin, datuak eraman MySQL datu basera, gero programan zehar erabili ahal izateko.
* Taula bakoitzaren datuak ondo erabiltzeko, bakoitzari funtzioa bat egin diogu:



* Ikusten den moduan, funtzio bakoitzaren barruan, taula bakoitzaren datuak kontrolatzen ditugu, eta baita, null diren datuek errorerik ez emateko, funtzio bat dugu hori konprobatzeko.
* Taula barruko datuak bistaratzeko orduan, funtzioa hau erabili dugu:

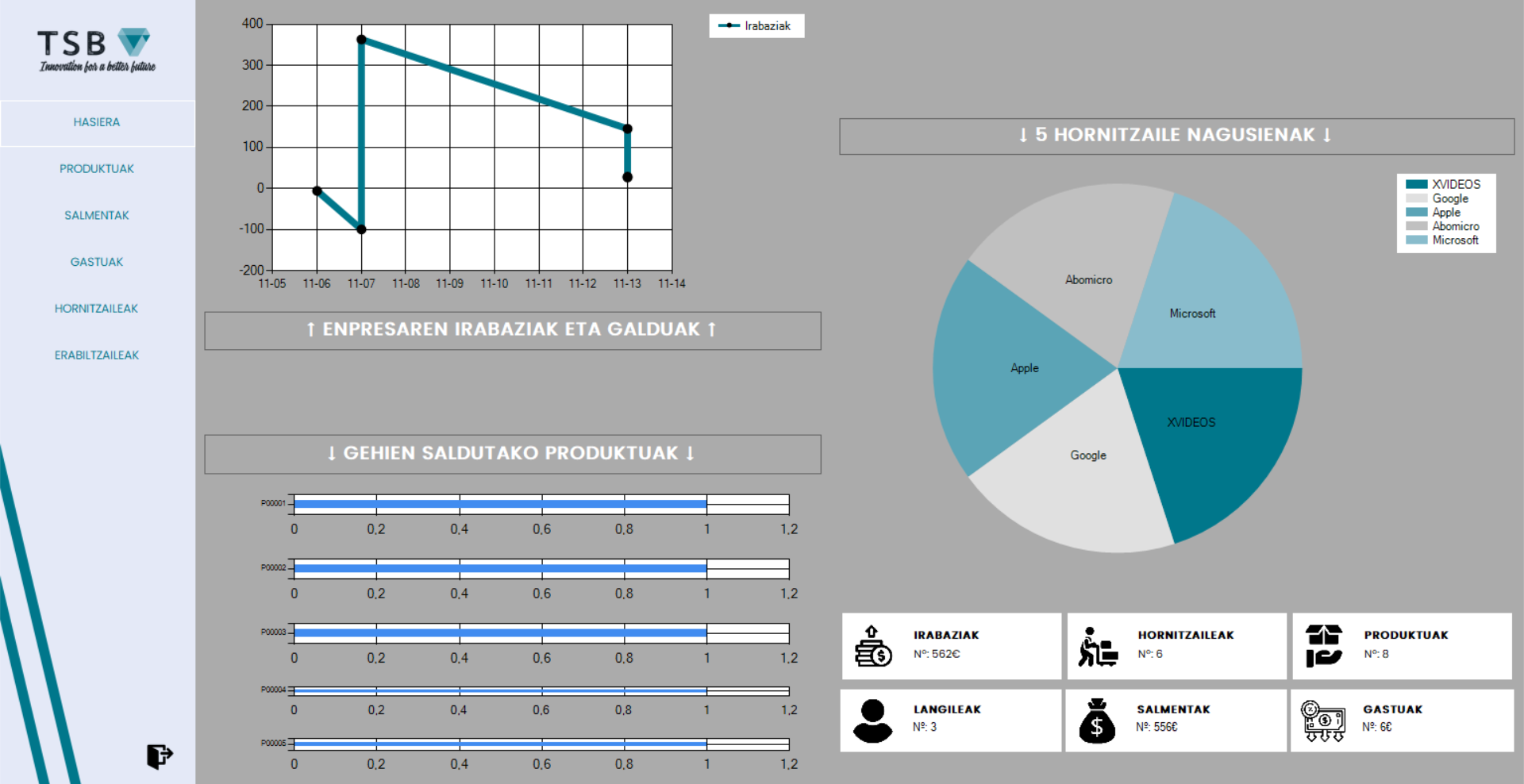


* Funtzio horrekin, datagridview-ko datu guztien kontrola ematen dugu. Behar dugun momentuan, bistaratu nahi ditugun datuak ikusteko.
* Ondoren, grafiko barruko datuak bistaratu ahal izateko, beste era batera egiten dugu, baino azkenean antzekoa da, datuak lortu, MySQL bidez, eta gero era bat edo bestea, grafiko barruan guk nahi bezala jartzea izango zen. Horretarako hainbat funtzio erabiltzen ditugu:

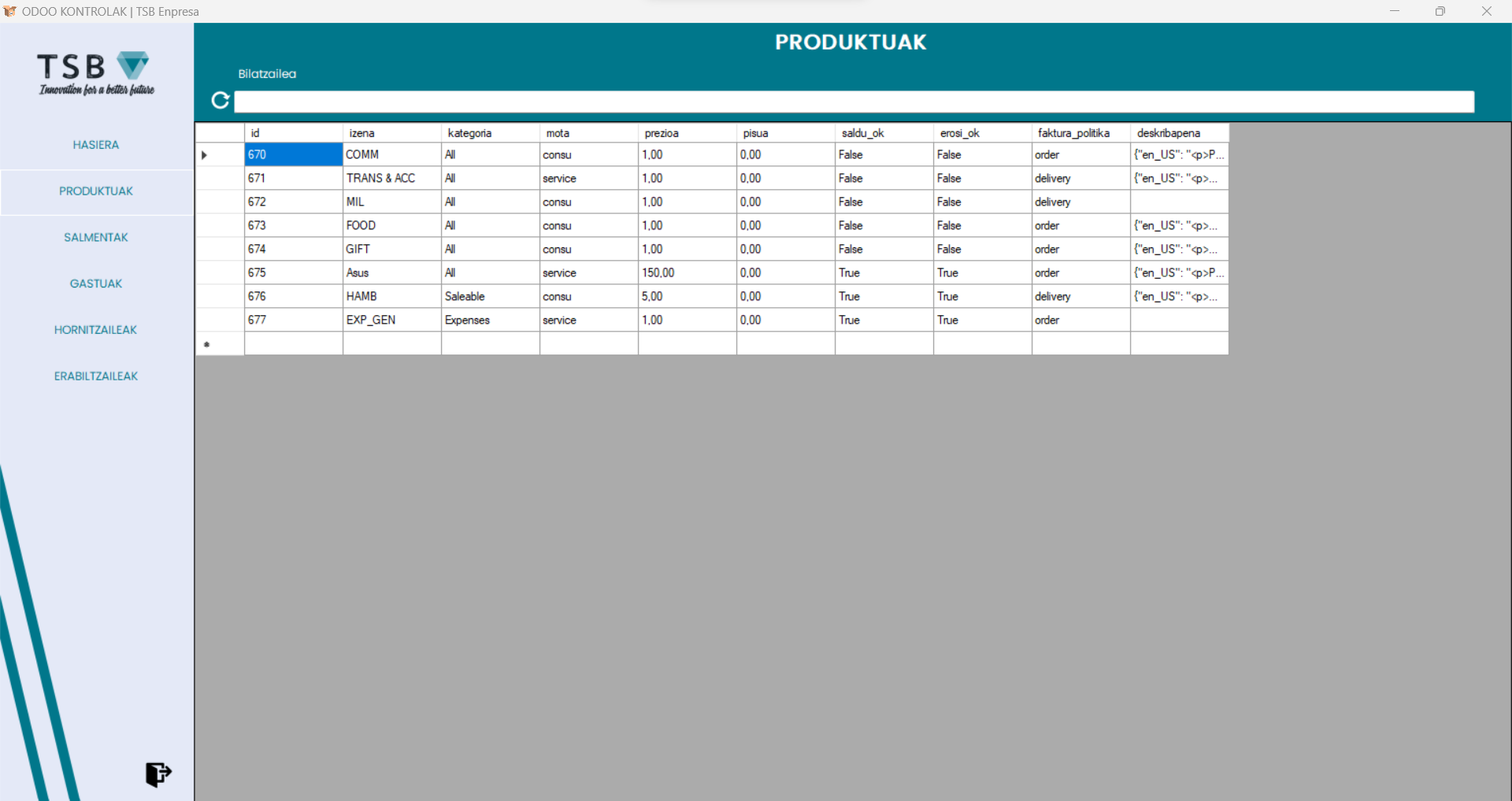


### Aplikazioak dituen aukerak:

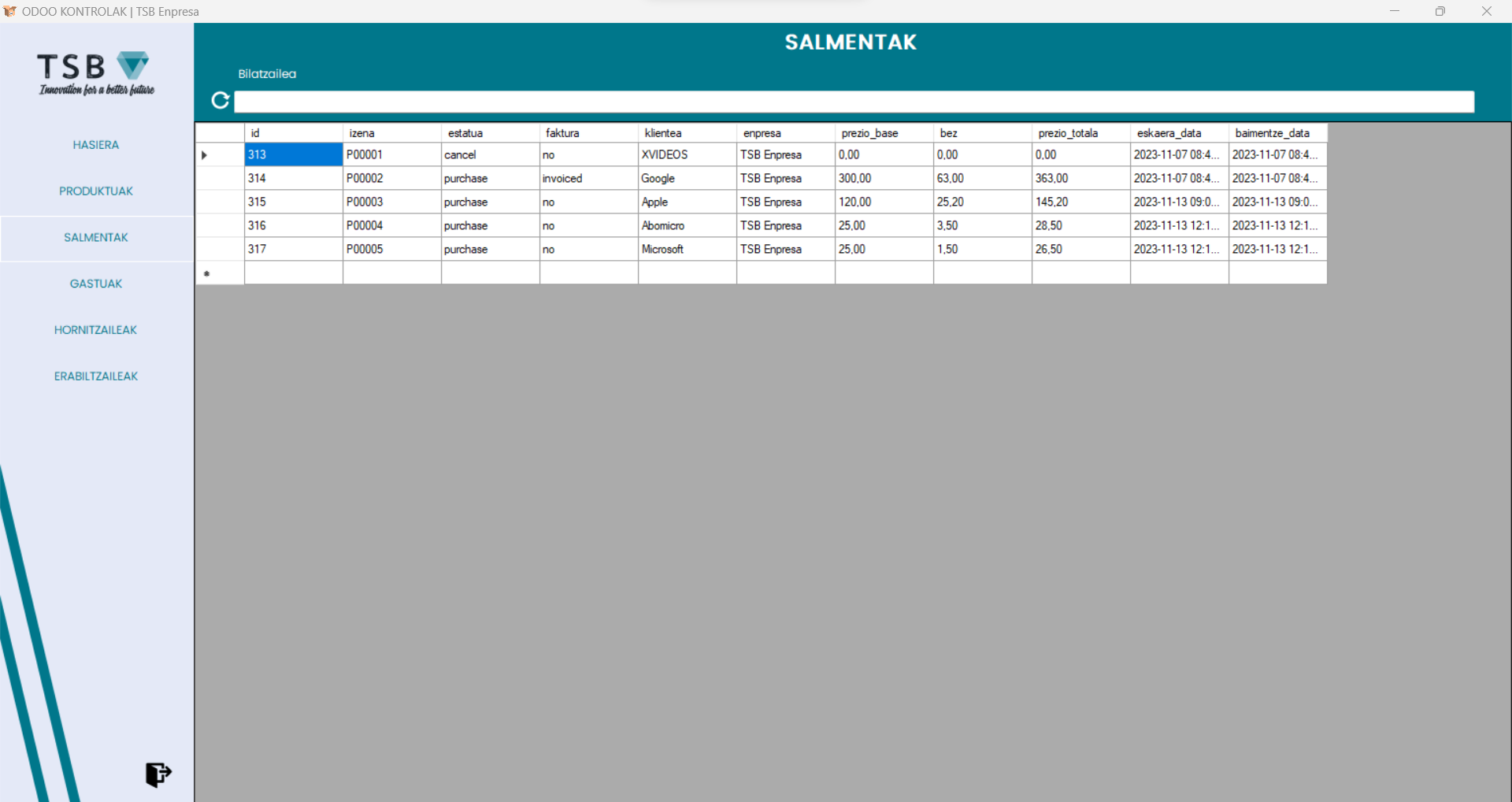
* Gure aplikazioak hainbat leiho ditu, hau da, menuraren barruan ditugu: Hasiera (hemen datu orokorrak bistaratu laike), produktuak (produktu taula bistaratu), Salmentak (Salmentak taula bistaratu), gastuak (gastuak taula bistaratu), Hornitzaileak (hornitzaile taula bistaratu) eta erabiltzaileak (erabiltzaile taula bistaratu).
* Hasiera:



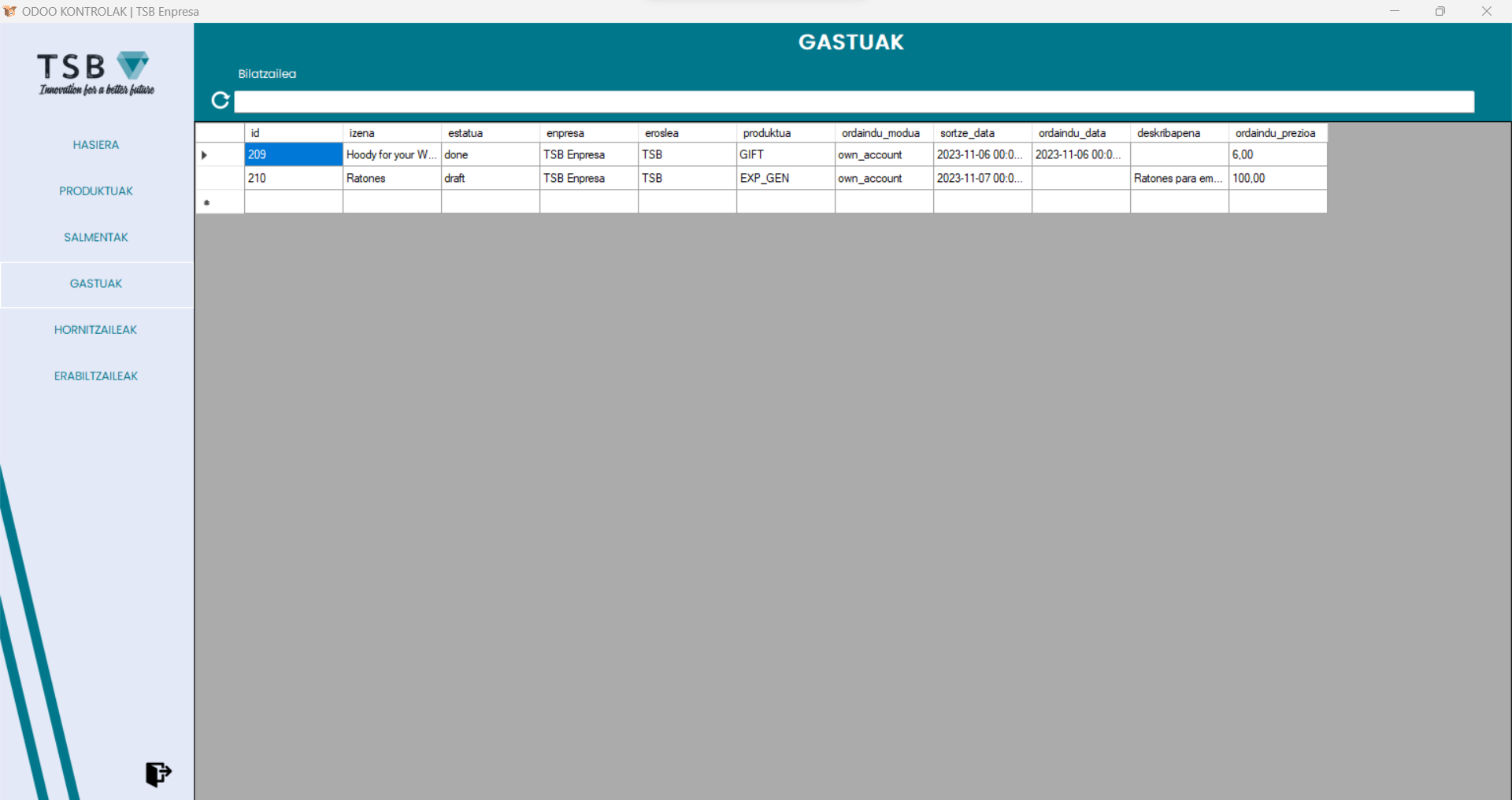
* Produktuak:



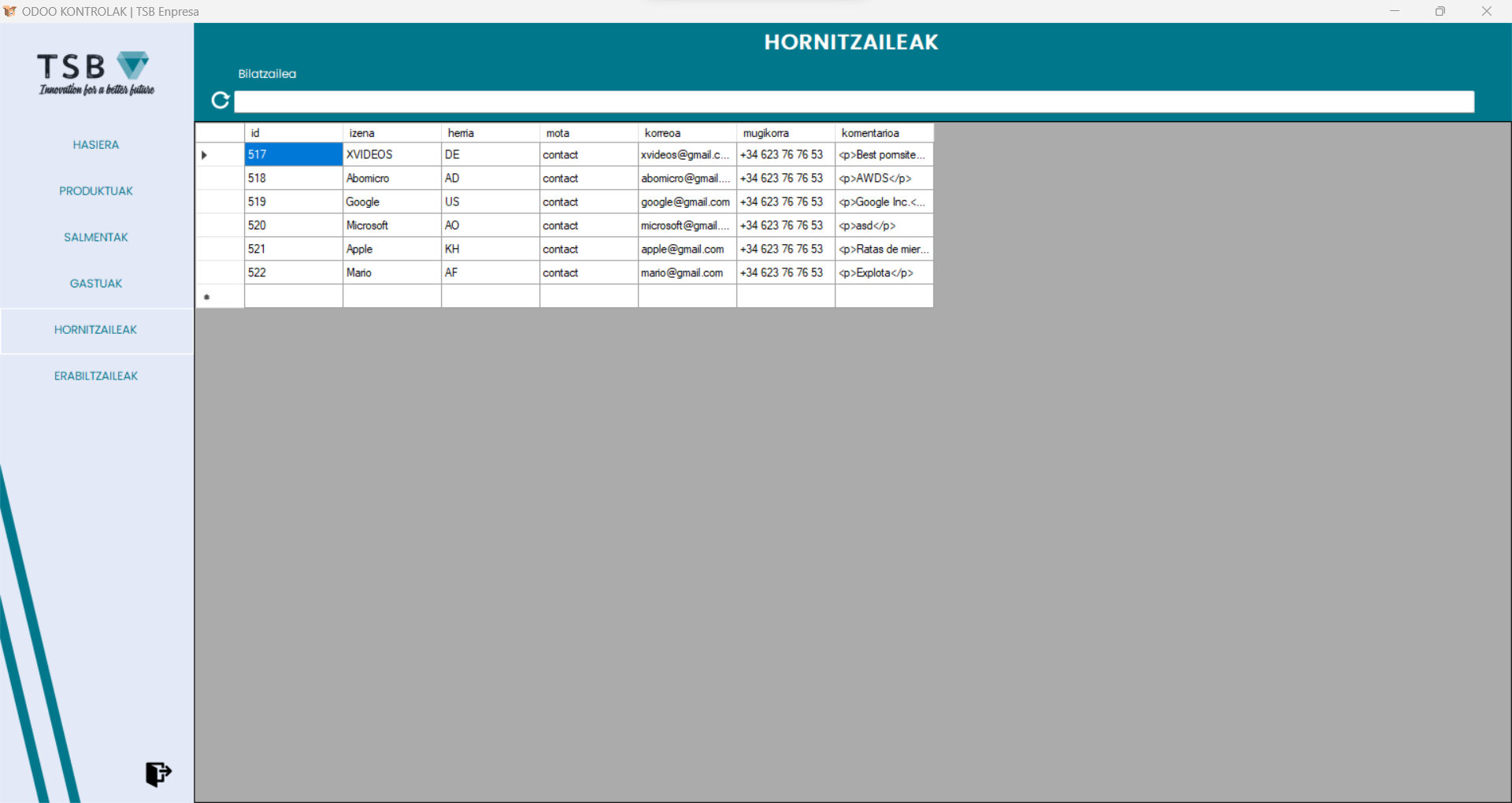
* Salmentak:



* Gastuak:



* Hornitzaileak:



* Erabiltzaileak:

