

# TSB EMPRESA

---

ERRONKA 1 – Lander Chamorro eta Ander Ibañez

# AURKIBIDEA

- Gure enpresa
- Erronkaren lehenengo pausuak
- Egunerokoa
- ERP Programa eta Datu Basea
- Komertzial APP (kudeapp)
- KudeApp Konexioa
- KudeApp Diseinua
- Enpresaren APP (OdooControl)
- OdooControl Konexioa
- OdooControl Diseinua
- Figma Disenua
- Enpresa ideia
- Forma juridikoa

# GURE ENPRESA

- Egia esan da, enpresa pentsatzea erreza egin zitzaigun. Gure enpresa sortuta genuen aurreko urteko Erronkarengatik, orduan, logoa, izena... Pentzatuta genituen.
- Izena dakizuten bezala “TSB” da, TSB Enpresa, eta gure eslogana “Innovation for a better future”.



#00778B




#5BA4B6



#89BCCC


# ERRONKAREN LEHENENGO PAUSUAK

- Erronkan zer egin behar den jakin eta gero, parametroak zehazten hasi ginen. Azkenean parametroak guk zer egin behar dugun jakiteko balio digu. Parametro barruan hainbat sekzio definitu ditugu, Android aplikazioa, enpresaren atala, mahaigaineko aplikazioa...





**TSB RETURNS | Parametroak**  
 TS Bren lanerako parametroak Errorea 1


MULTIMEDIA



**Ander Ibañez Miguez** 1me

**APP Disenua**

- Menu txiki bat pentsatu (Web Orriaren menua)
- Disenua ulergarri eta erraz erabiltzekoa
- Slide menu batekin egitea, opzio guztiekin



 0
  0


 Añadir comentario


**Ander Ibañez Miguez** 1me


**APP Leihoak**

- Sesioa hasteko leihoa
- Odooren gestioa egiteko
- Orokorrean, modulu guztiekin leihoak
- Postgres-ekin konexioa eta gero sqllite-ra pasa datu guztiak
- SQLite datu basea sortu
- Grafika leihoak egin

 0
  0



 Añadir comentario


ERP



**Ander Ibañez Miguez** 1me

**Odoo-ren instalazioa**

- Docker erabiliz datu basea eta odoo zerbitzaria instalatu


 0
  0


 Añadir comentario



**Ander Ibañez Miguez** 1me

**Odoo-ren konfigurazioa**

- Moduluak instalatu
- Erabiltzaileak, taldeak,... sortu
- Moduluen gestioak



 0
  0


 Añadir comentario


**Ander Ibañez Miguez** 1me


**Datu basea instalatu eta konfiguratuta**

- PostgreSQL datu basearen instalazioa egin
- Konfigurazio egokia egin
- Taulak, Erabiltzaileak,... sortu
- Konexioa ezarri

 0
  0



 Añadir comentario


ZERBITZU ETA PROZESUAU


**Ander Ibañez Miguez** 1me


**Nun eta nola erabili zerbitzuak**

- Haria erabili Postgreko konexioa egiteko android-en
- App bat egin python-en non Dockerrea ireki eta kontainerrak piztu egiten duen.

 0
  0



 Añadir comentario


INTERFAZE GARAPENA



**Ander Ibañez Miguez** 1me

**Interfazearen diseinua**

- Menua
- Disenua
- Zein leiho behar ditugun



 0
  0


 Añadir comentario


**Ander Ibañez Miguez** 1me


**Leihoak**

- Hasiera leihoa
- Grafikoa gestionatzeko leihoa
- Taulak erakusteko leihoak
- Mysql-rekin konexioa egin

 0
  0



 Añadir comentario


DATU ATZIPENA



**Ander Ibañez Miguez** 1me

**Datu basea**

- Beharrezko libreriak eta konektoreak lortu
- Konexioa ondo ezarri
- SQL sententziak egin
- Datuen gestioa egin
- Taula egokiak sortu behar dugun informazioarekin



 0
  0


 Añadir comentario


**Ander Ibañez Miguez** 1me


**Datuekin ibili**

- Salbuespenak egin
- Datuen kontrola
- Datuen salikapena

 0
  0



 Añadir comentario


ENPRESA


**Lander Chamorro Jimenez** 27m

**Planifikazioa eta egiterakoak**

- Enpresa ideia egin
- Forma Juridikoa
- Canvas taula
- SDG
- Merkata ikerketa

 0
  0

 Añadir comentario

# EGUNEROKOA

- Egin behar dugun lana beti eguneratuta emateko, eguneroko bat erabili dugu. Hasieratik, aste bakoitzean zer lortu behar den zehaztu genuen.

ASTEA	ATAZA	ENTREGA	NOLA EGINGO DA			AURREIKUSITAKO DENBORA	ARDURADUNA	ATAZAREN EGOERA	IRAKASLEAREN ONESPENA
			TALDEAN	BINAKA	BANAKA				
16/10/2023 - 20/10/2023	TALDEA: Taldea garatu (izena, lema, logoa...)		TSB				ANDER	EGINDA	
16/10/2023 - 20/10/2023	TALDEA: Talde kontratua ezarri eta sinatu (txantiloiarekin egin)		TSB				ANDER	EGINDA	
16/10/2023 - 20/10/2023	TALDEA: Parametroak zehaztu		TSB				TSB	EGINDA	
16/10/2023 - 20/10/2023	TALDEA: Proposamenak egin eta aurkeztu				TSB		TSB	EGINDA	
16/10/2023 - 20/10/2023	TALDEA: Erronkak definizioak sortu eta aurkeztu				TSB		TSB	EGINDA	
16/10/2023 - 20/10/2023	TALDEA: Talde proposamenak juntatu		TSB				TSB	EGINDA	
16/10/2023 - 20/10/2023	APP: Diseinua egin eta ideia printzipalak azaldu.		TSB				TSB	EGINDA	
16/10/2023 - 20/10/2023	APP: Logoa prestatu, itxura txukundu...				ANDER		ANDER	EGINDA	
16/10/2023 - 20/10/2023	APP: Zein datu base erabiliko dugun aukeratu (SQLite)		TSB				LANDER	EGINDA	
16/10/2023 - 20/10/2023	APP: Menuaren itxura egin				ANDER		ANDER	EGINDA	
16/10/2023 - 20/10/2023	ERP: Odoon nun edukiko dugun eta nola aukeratu		TSB				LANDER	EGINDA	
16/10/2023 - 20/10/2023	ERP: Datu basearen instalazioa (PostgreSQL)		TSB				LANDER	EGINDA	
16/10/2023 - 20/10/2023	ERP: Odoon instalatu eta konfiguratu (moduluak)		TSB				TSB	EGINDA	
16/10/2023 - 20/10/2023	ERP: Erabiltzaileak sortu eta beraien pantailak egin (KOMERTZIALA ETA KONTABILITATEA)				ANDER		ANDER	EGINDA	
16/10/2023 - 20/10/2023	ENPRESA: Enpresaren ideia pentsatu eta idatzi		TSB				TSB	EGINDA	
16/10/2023 - 20/10/2023	ENPRESA: Forma juridikoa				LANDER		LANDER	EGINDA	
16/10/2023 - 20/10/2023	ENPRESA: ODS				LANDER		LANDER	EGINDA	
16/10/2023 - 20/10/2023	ENPRESA: Canvas taula				LANDER		LANDER	EGINDA	
16/10/2023 - 20/10/2023	ENPRESA: Merkatu ikerketa				LANDER		LANDER	EGINDA	
30/10/2023 - 03/11/2023	ERP: Datu baseko beharrezko taulak aukeratu		TSB				ANDER	EGINDA	
30/10/2023 - 03/11/2023	ERP: Portuak aldatu		TSB				TSB	EGINDA	
30/10/2023 - 03/11/2023	INTERFAZ: Logina bukatu				ANDER		ANDER	EGINDA	
30/10/2023 - 03/11/2023	INTERFAZ: Diseinuarekin hasi				ANDER		ANDER	EGINDA	
30/10/2023 - 03/11/2023	INTERFAZ: Pantailak planifikatu				ANDER		ANDER	EGINDA	
30/10/2023 - 03/11/2023	APP: Konexio klasea sortu (PostgreSQL) eta lortu				ANDER		ANDER	EGINDA	
30/10/2023 - 03/11/2023	APP: SQLite-n sarrerak egiteko klasea				ANDER		ANDER	EGINDA	
30/10/2023 - 03/11/2023	APP: SQLite-ko taulak sortzeko klasea				ANDER		ANDER	EGINDA	
06/11/2023 - 10/11/2023	INTERFAZ: PostgreSQL-ra konektatzeko konexio klasea sortu				ANDER		ANDER	EGINDA	

# ERP PROGRAMA ETA DATU BASEA

- Gure kasuan Odoo ERP-a erabili dugu eta datu base PostgreSQL. Biak Docker Hub baten barruan konfiguratu ditugu, Odook 8069 eta PostgreSQL-ak 8070 portuekin.
- Odoo barruan hiru erabiltzaile sortu ditugu: TSB (Administradoreak), KOMERTZIALA eta KONTABILITATEA.
- Erabiltzaile bakoitzari bere pantaila pertsonalizatu diogu. Hau ondo lortzeko, “Hide Menu” plugin bat erabili dugu.

POSTGRESQL



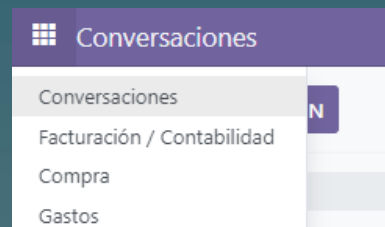
DBEAVER



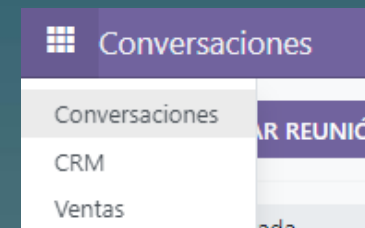
## ERABILTZAILEAK

<input type="checkbox"/> Nombre	Usuario	Idioma
<input type="checkbox"/> Komertziala	komertziala@tsb.com	Spanish / Español
<input type="checkbox"/> Kontabilitatea	kontabilitatea@tsb.com	Spanish / Español
<input type="checkbox"/> TSB	tsbenpresa@gmail.com	Spanish / Español

## KONTABILITATEA



## KOMERTZIALA



# KOMERTZIALEN APP (KUDEAPP)

- Honen helburua, komertzialek datuak erraz bistaratu ahal izateko aplikazio bat egitea da, kontutan hartuta ez dutela beti konexiorik edukiko.
- Aurrekoa lortzeko, bi datu base erabiltzen ditugu, PostgreSQL (printzipala, Odooren datuak lortzeko) eta SQLite (datu base lokala). Hauek erabili ahal izateko, libreriak eta mugikorraren sareko konexioa erabiltzeko baimenak.

## SAREKO KONEXIOA

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

## LIBRERIAK

```
implementation ("org.postgresql:postgresql:42.2.5")  
implementation ("androidx.sqlite:sqlite:2.2.0")
```

TSB\_KUDEAPP LOGO



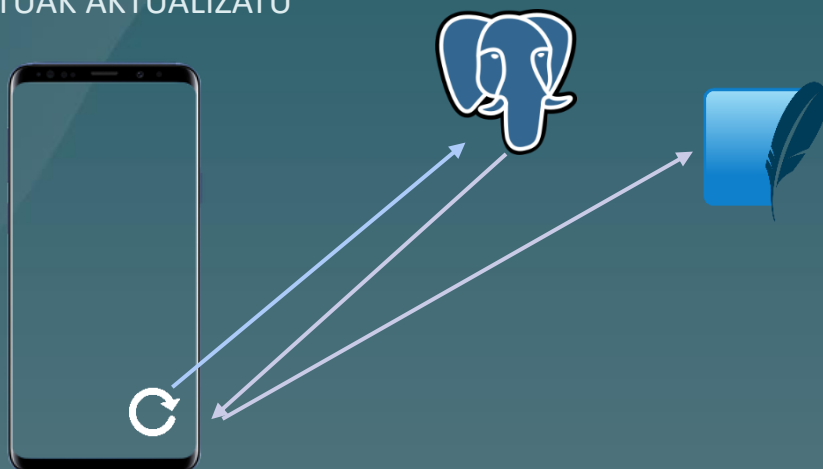
SQLite



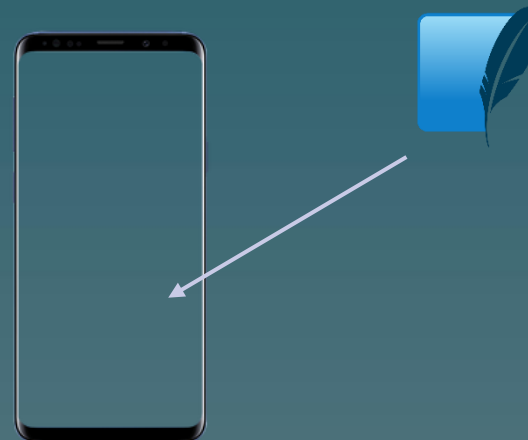
# KUDEAPP KONEXIOA

- Aplikazioak bi datu base ditu, horrela segurtasuna handitzen dugu eta baita, aplikazioak konexiorik gabe funtzionatu ahal izateko. Hau funtzionatzeko, datu base lokalaren datuak berritzeko botoia dugu, horrekin PostgreSQL datu basera konexioa egin eta datuak hartzen ditu, gero SQLite ditugun tauletan sartzeko datuak.
- Konexioa asíncrono moduan egiten dugu, horrela, erabiltzailearen erabilgarritasuna hobetu eta konexioa ez egitearen aurkako erroreak kontrolatzen ditugu.

DATUAK AKTUALIZATU



DATUAK BISTARATU





# KUDEAPP DISEINUA



- Aplikazioaren diseinua bastante simplea dugu, azkenean erabilgarria, ulergarria eta txukua izan behar du.
- Diseinua bistaratu:



# ENPRESAREN APP (OdooControl)

- OdooControl, enpresa barneko aplikazio bat da, berarekin enpresako datu garrantzitsu batzuk ikusteko aukera ematen digu.
- Honek ere, bi datu basearekin egiten du lana, PostgreSQL (printzipala) eta MySQL (lokala). Horretarako konektoreak instalatu behar izan ditugu.
- Bi klase printzipal ditu, datubaseko konexioak kudeatzeko: PostgreSQLConnection eta MySQLConnection (hauekin konexioa ireki, itxi edo lortzeko erabiltzen ditugu).

## KONEKTOREAK

	<b>MySQL.Data</b> por Oracle MySQL.Data.MySqlClient .Net Core Class Library	8.2.0
	<b>Npgsql</b> por Shay Rojansky, Nikita Kazmin, Brar Piening, Yoh Deadfall, Austin Drenski, Emil Lenngren, Francisco Figueiredo Jr., Kenji Uno Npgsql is the open source .NET data provider for PostgreSQL.	7.0.6

OdooControl



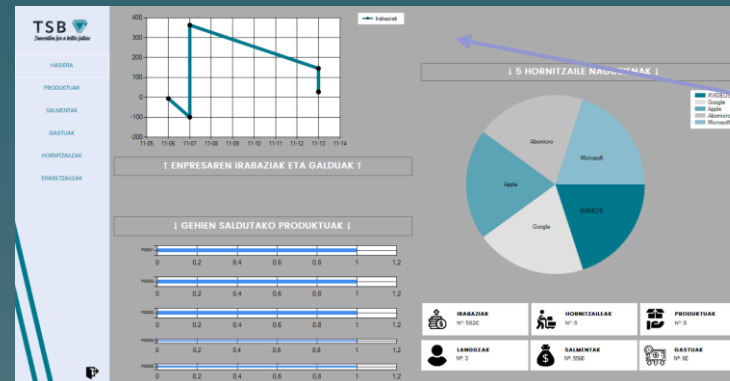
# OdooControl KONEXIOA

- Aplikazioak bi datu base ditu, horrela segurtasuna handitzen dugu eta ez ditugu konexio asko botatzen datu base printzipalera. Aplikazioa irekitzerakoan datua automatikoki aktualizatzen dira, hau da, PostgreSQLko beharrezko datuak hartu eta MySQLko datu basean sartzen ditu, datu bakoitza dagokien tauletan.

## DATUAK EGUNERATU



## DATUAK BISTARATU



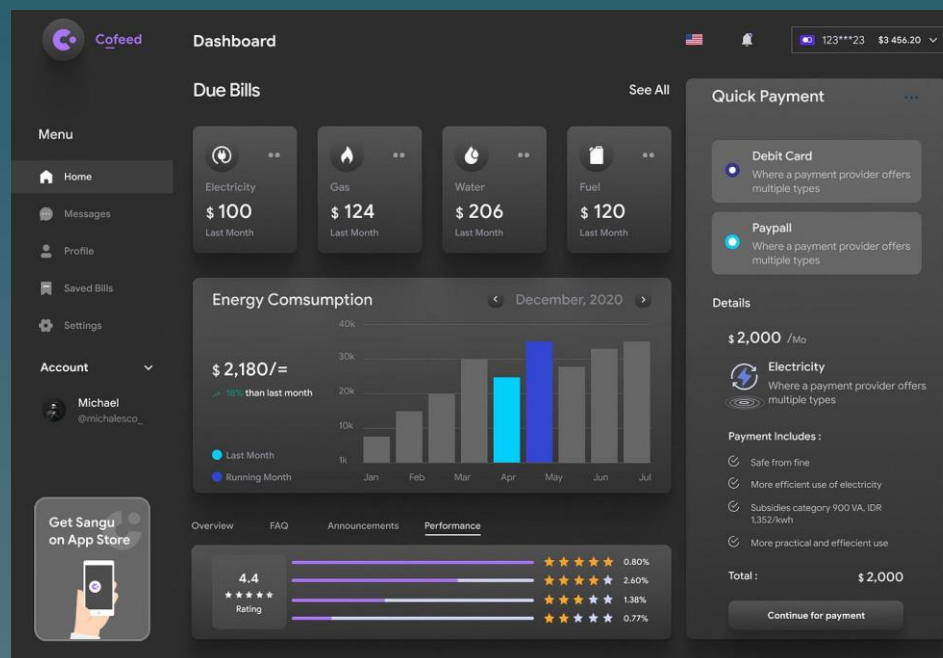
# OdooControl DISEINUA

- Aplikazioaren diseinua bastante simplea dugu, azkenean erabilgarria, ulergarria eta txukuna izatea nahi genuen.
- Diseinua bistaratu:



# FIGMAREN ERABILERA DISEINUENTZAT

- Esan beharra dugu, hasieran figma erabiltzea pentsatu genuela, baina azkenean, geuk eskuz egitea aukeratu genuen, gehiago ulertu eta kontrol gehiago izango genuela Kodigo eta diseinuarekiko.
- Zergaitik, figma dena eginda etortzen da, eta guk bakarrik “backend” programatu beharko genuen, nahi ditugun datuak sartzeko, baino kontrol eta ulermen gutxiago izango genuen.



# PYTHON APP

- Lehenengo, “dockerIreki()” funtzioa dugu, docker hub irekitzeko funtzioa du, honek prozesuekin exekutatzen du.
- Ondoren, Bezero bat sortu dockerrera sartzeko, kontenedore guztiak listatu eta bukle batekin berifikatu egiten du beraien estatua, db kontenedorea pizten du arazoek ez edukitzeko besteekin, ondoren bestiak.
- Bukatzeko, main-ean bi funtzioak deitu eta listo.

```
def dockerIreki():
    # Docker irekitzeko komandoa
    docker_desktop_command = "Start-Process -NoNewWindow -FilePath 'C:\\Program Files\\Docker\\Docker\\Docker Desktop.exe'"

    try:
        # Try-catch bat docker irekitzen danean
        subprocess.run(["powershell", "-Command", docker_desktop_command], shell=True, check=True)
        print("Docker ondo ireki da.")
    except subprocess.CalledProcessError as e:
        print(f"Errore bat gertatu da Docker irekitzerakoan: {e}")

def kontenedoreGuztiakPiztu():
    # Docker kliente bat sortu egiten du
    client = docker.from_env()

    # Kontainer lista bat lortzen du
    all_containers = client.containers.list(all=True)

    if not all_containers:
        print("Ez daude kontenedoreak.")
        return

    # Kontenedore bakoitza pizten du
    for container in all_containers:
        # Bere estatua konprobatzen du
        if container.status == 'exited':
            # Kontenedorea binkulatuta badago ikusten du
            links = container.attrs.get('HostConfig', {}).get('Links', [])

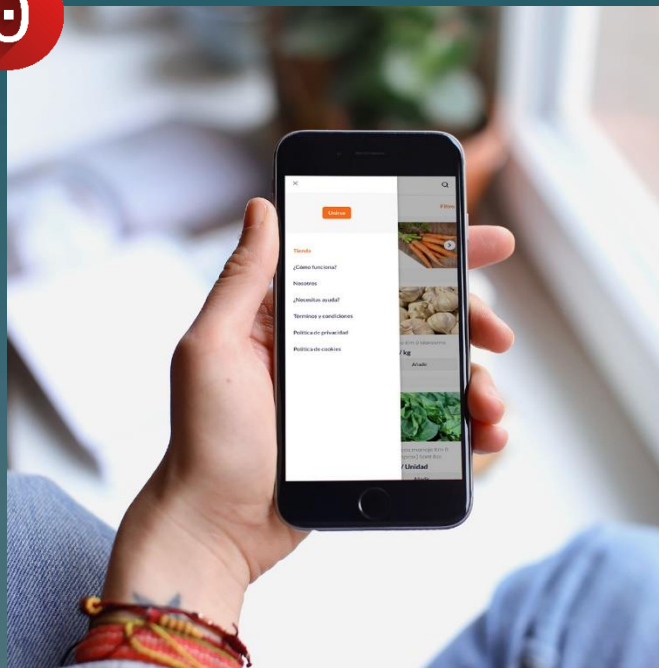
            # Berifikatu egiten du
            if links and any('/db' in link for link in links):
                # /db kontenedorea lehenengo pizten du, erroreak ez edukitzeko
                db_container = client.containers.get('db')
                if db_container.status == 'exited':
                    print(f"/db kontenedorea pizten arazoak ez edukitzeko besteekin.")
                    db_container.start()

            # Kontenedore aktuala pizten du
            print(f"Kontenedorea pizten: {container.name}")
            container.start()
        else:
            print(f"{container.name} kontenedorea piztuta dago.")

# MAIN
if __name__ == "__main__":
    dockerIreki()
    kontenedoreGuztiakPiztu()
```

# ENPRESA IDEIA

- Aplikazioak km0-ko janari guztia agertuko zaizu mapa batean, bertan ikusi dezakezu bere prezioa, nun dagoen, zeinek saltzen duen edota saltzen duen kantitatea, mezulari bat edukiko du saltzailearekin hizketan aritzeko.





# FORMA JURIDIKOA

- Sozietate Mugatua aukeratu dugu: Enpresaren kudeaketa errazteko aukera baliagarriak eta baldintza onak eskaintzen ditu.
- 2 Bazkide eta 7 langile
- Kapital soziala: 100 mila euro
- Erantzukizuna: Mugatua
- Zergak: Sozietate gaineko zergak





# ESPERO DUGU GUSTATU IZANA

---

Galdera edo zalantzarik?