

Gestió d’una falla

Projecte final de cicle formatiu



[Date]

[Company name]

[Company address]

|  |  |
| --- | --- |
| **Nom i cognoms** | Joel Fuster Bosch |
| **NIF/NIE** | 020942567D |
| **Curs i CF** | 2n DAM |

|  |  |
| --- | --- |
| **Títol del projecte** | Gestió d’una falla |
| **Nom del tutor individual** | Juan Bautista Talens |
| **Nom del tutor del grup** | Jose Alfredo Murcia |
| **Resum** | *Gestió d’una falla que permet als usuaris fer compres amb les seues polseres de la falla amb tecnologia NFC (Near Field Communication)*. |
| **Abstract** | *Management of a Falla-making committee that provides the users doing shopping with their bracelets of the Falla-making committee with NFC technology (Near Field Communication).* |
| **Mòduls implicats** | * SGE (Sistemes de gestió empresarial) * PMDM (Programació multimèdia i dispositius mòbils) |
| **Data de presentació** | 2 de juny de 2025 |

Contents

[1.Introducció del projecte 3](#_Toc199238583)

[1.1Descripció del projecte 3](#_Toc199238584)

[1.2 Objectius del projecte: 4](#_Toc199238585)

[1.3 Tecnologies usades per al projecte 4](#_Toc199238586)

[1.3.1 Flutter 5](#_Toc199238587)

[1.3.2 Dart 5](#_Toc199238588)

[1.3.3 VSCode 6](#_Toc199238589)

[1.3.4 Github 7](#_Toc199238590)

[1.3.5 Figma 7](#_Toc199238591)

[1.3.6 NodeJS 8](#_Toc199238592)

[1.3.7 PostgreSQL 8](#_Toc199238593)

[1.3.8 Odoo 9](#_Toc199238594)

[2.Tecnologies i ferramentes 9](#_Toc199238595)

[2.1 Gestor d’estats: Provider 9](#_Toc199238596)

[2.2 Arquitectura CLEAN 10](#_Toc199238597)

[2.2 Tecnologia NFC 12](#_Toc199238598)

[2.3 Codi QR 13](#_Toc199238599)

[9.Dificultats i millores futures 15](#_Toc199238600)

[9.1 Dificultats 15](#_Toc199238601)

[9.2 Millores futures 15](#_Toc199238602)

[9.3 Que he aprés 16](#_Toc199238603)

[10.Annexos 16](#_Toc199238604)

[10.1 Github 16](#_Toc199238605)

[10.2 Bibliografia 16](#_Toc199238606)

# 1.Introducció del projecte

## 1.1Descripció del projecte

Aquest projecte té com a objectiu unir als fallers en grups anomenats families, els quals es poden unir altres fallers per a una gestió més comoda de la realitat a l’hora de poder unir-se a esdeveniments, comprar menjar/begudes en la parada de la falla...,etc. tot en una sola aplicació per a que puguen dur a terme els esdeveniments de la falla de forma més comoda.

## 1.2 Objectius del projecte:

Gestionar tots els esdeveniments, productes, tickets...,etc gràcies amb les polseres que funcionen amb la tecnologia NFC(Near-Field-Comunication) i alternativament amb ús de QR(Quick-Response-Code) per als dispositius que no conten amb la tecnologia NFC en una sola aplicació.

## 1.3 Tecnologies usades per al projecte

Primerament vaig a fer un resum de les específicacions dels meus dispositius que han sigut involucrats en aquest projecte:

El projecte ha sigut desenvolupat en el meu ordinador personal, un portàtil amb les següents específicacions:

-Processador: Intel Core I5-11400H @ 2.70GHz, 2668MHz, 6 processadors principals, 12 proccessadors lògics

-Sistema operatiu: Microsoft Windows 11 Pro

-Memòria RAM:16GB

I ha sigut provat en un dispositiu mòbil Android de les següents específicacions:

-Versió de Android:14

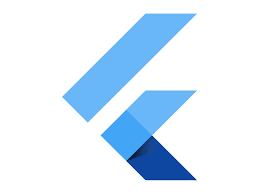
-Processador: MediaTek Dimensity 700 8 nuclis

-Espai d’enmagatzenament: 128GB

-Memòria RAM: 4GB

Ara esmentare les eines que he usat per a que aquest projecte haja sigut possible.

### 1.3.1 Flutter



Framework multiplataforma desenvolupat per Google, tenia com a proposit l’ús de software de la comanyia, però amb el temps s’ha transformat en una plataforma de codi obert per al desenvolupament multiplataforma.

Permet el desenvolupament en les paltaformes de:

-Android

-iOS

-Windows

-Linux

-macOS

-Web

En aquest projecte s’ha desenvolupat únicament per a plataformes Android, degut a que no dispose d’un sistema iOS per a fer proves i volia centrar-me en el ecosistema de Android que és el sistema operatiu mòbil més usat en Espanya un 78,8% en 2023 segons CNMC(Comisión Nacional de los Mercados y la Competencia), teniu un link en la bibliografia del article de CNMC.

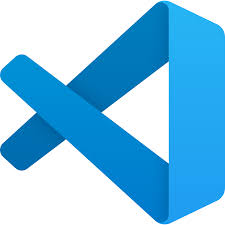
### 1.3.2 Dart



Dart es un llenguatge de programació de codi obert desenvolupat per Google, va ser revelat en la conferènci *goto* el 10 d’octubre de 2011. L’objectiu de Dart és oferir una alternativa més moderna a JavaScript per al desenvolupament web, es pot veure reflexat en les declaracions de Lars Bak, ingenier de software en Google que defineix a Dart com un *“llenguatge estructurat però flexible per a programació web”*.

He utilitzat Dart degut a que el framework de Flutter usa el llenguatge de programació de Dart m’he vist en l’obligació de programar en ell.

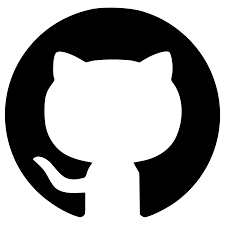
### 1.3.3 VSCode



Visual Studio Code llançat el 18 de novembre de 2015 és un editor de text pla desenvolupat per Microsoft de codi obert basat en Electron, un framework que s’utilitza per a implementar Chronium i NodeJS com a aplicacions d’escriptori, que s’executa en el motor de disseny Blink.

He usat Visual Studio Code com el meu IDE perquè és el IDE que més conec i més gaste actualment, però tens que tindre Android Studio instal·lat per les dependencies de Android que proveeix aquest últim junt amb les extensions de Flutter, i aquest últim té per defecte les llibreries de Git, així que m’ha ajudat molt per al control de versions Github.

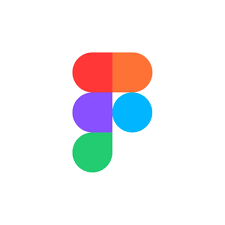
### 1.3.4 Github



Github, publicat en 2008 i desenvolupat amb Ruby on Rails és una forja(paltaforma de desenvolupament col·laboratiu) per a allotjar projectes utilitzant el sistema de control de versions de Git. S’usa principalment per a la creació de codi font de programes d’ordinador.

He fet servir GitHub com a repositori públic, perquè qualsevol persona puga veure el codi font del projecte, descarregar-lo o clonar-lo. Està tot organitzat per carpetes i podeu trobar tant el frontend com la resta del projecte. No he treballat amb branques ni commits complexos, però l’he utilitzat com a lloc de referència i accés al codi.

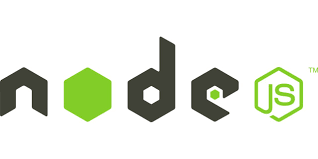
### 1.3.5 Figma



Figma, desenvolupat per Dylan Field i Evan Wallace començaren a treballar en Figma en 2012 i el van llançar el 27 de septembre de 2016 és un editor de gràfics vectorial i una ferramenta de generació de prototips, principalment basada en la web, amb caracteristiques off-line adicionals habilitades per aplicacions d’escriptori en macOS i Windows.

He utilitzat Figma únicament per a fer el prototip d’alta fidelitat del frontend. Em va servir per planificar el disseny visual abans de començar a programar l’aplicació, i assegurar-me que l’estructura i els estils quedaven clars.

### 1.3.6 NodeJS



\*Historia node\*

He usat nodeJS com a backend i com a API, en Flutter faig peticions en el servidor de Node, que aquesta és connectara a la base de dades i torna la petició amb les dades que és mostraran en pantalla, però originalment anava a usar Odoo, després ho explicaré.

### 1.3.7 PostgreSQL



\*Historia Postgre\*

He usat PostgreSQL com a base de dades que està connectada al servidor amb NodeJS, aquesta guarda tota la informació sobre usuaris, families, productes...,etc i el servidor de Node arreplegara aquesta informació i la passara a l’aplicació en Flutter.

### 1.3.8 Odoo



\*Historia Odoo\*

Originalment el backend del meu projecte abans de que fera el servidor de Node, és paregut el que fa el servidor de Node i la base de dades PostgreSQL, però amb una sola plataforma, va ser substituïda pel meu tutor per a implementar el servidor de Node, el qual està en eta presentació, però per falta de temps no va ser possible implementar les funcions de Node a Odoo.

# 2.Tecnologies i ferramentes

## 2.1 Gestor d’estats: Provider

Provider és un gestro d’estat més avanzat i més sencill que BLOC, un altre gestor de l’estat, ara dire com podeu implementar provider en els vostres projectes:

1.Tindre tant la capa de serveis tant la de repositori ja feta

2.Crear la capa de provider en la carpeta anomenada “lib”

3.Crear el fitxer “funcioProvider.dart” i al crear la classe que implemente “ChangeNotifier”

4.Inicialitzar el provider amb el repository

5.Completar el provider amb les funcions del repositori

6.Inicialitzar al fitxer “main.dart”

7. Fer que retorne “Multiprovider” per a tindre una llista de providers

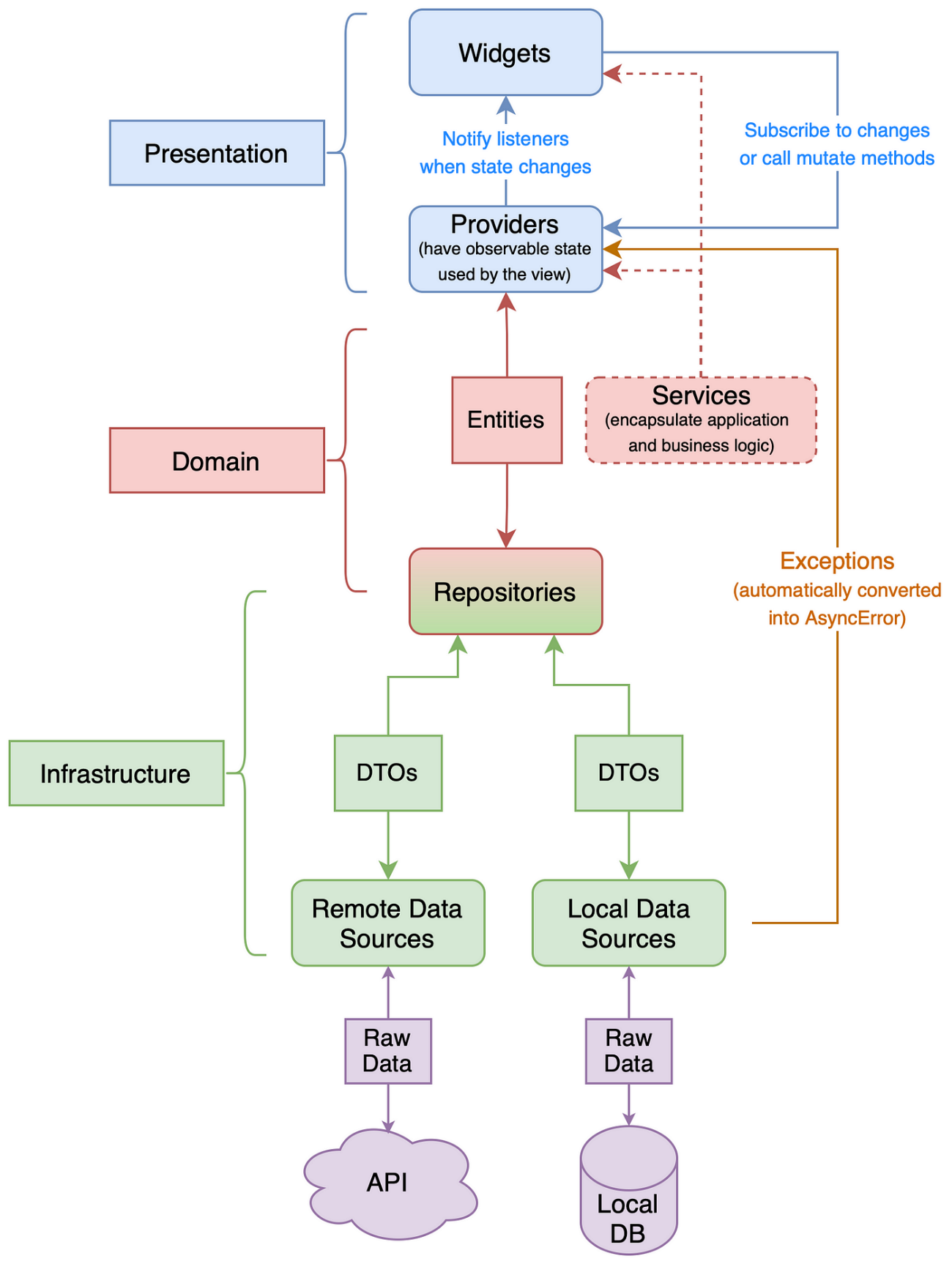
8.En la pantalla que vullgues inicialitza la funció amb “Provider.of<funcioProvider>”

I amb això ja pots usar la teua provider en qualsevol lloc de l’aplicació.

## 2.2 Arquitectura CLEAN

ARQUITECTURA: CLEAN (UI -> Repository -> DataSource)

L’arquitectura CLEAN es pot resumir en la següent imatge:



Explicació:

UI: Interficie de l’aplicació (el que és mostrara en la app).

Repository: Capa intermitja que connecta el DataSource i la intreficie(UI).

DataSource: Capa que fa peticions a APIs o a altres llocs per a arreplegar la informació i després mostrar-la en la interficie(UI).

L’arquitectura CLEAN és compatible amb els gestors d’estat com BLOC, Provider, que carregarà la lògica de negoci de la aplicació.

## 2.2 Tecnologia NFC

NFC o per les seues sigles (Near Field Communication) funciona a través d’ones electromagnètiques que conseguiran comunicar un dispositiu a un altre fins a una distancia de 10 a 15 cm a una velocitat de 424Kb/s, ara esmentare alguns exemples.

Exemples de NFC:

-Tarjetes tant de crèdit com d’accés

-Polseres amb la tecnología abans esmentada (aquesta serà en la que treballarem)



-Alguns rellotges intel·ligents



-Etiquetes



-Algunes figuretes col·leccionables (per exemple els Amiibo de Nintendo per als sistemes de Nintendo 3DS, Nintendo Switch i possiblement també en Nintendo Switch 2)



Els dispositius que conten amb la tecnologia NFC tenen 2 formes per a comunicar-se:

-Pasiva: Sols un dels dispositus està actiu i l’altre intercambiara les dades.

-Activa: Els 2 dispositius generen un camp electromagnètic i intercambien dades.

## 2.3 Codi QR

També conegut com (Quick-Respnse-Code), va ser inventat en 1994 pel ingenier japones **Masahiro Hara**, va ser desenvolupat per la empresa **Denso Corporation** la qual és una subsidiaria de Toyota, el codi QR és una millora al codi de barres tradicional, degut a que és un mètode per a enmagatzemar informació mitjançant constractes de color de forma bidimensional(2D), tenint com a objectiu una letura ràpida del contingut del codi, per això el seu nom de **Q**uick-**R**esponse-Code, és pot usar de diverses fromes, com per exemple:

-Productes en general



-Cartes de restaurant



-Correus electrònics



-Anuncis



-Accés ràpid a documents gubernamentals



-Alguns videojocs(per a conseguir algunes reconpenses o et du a la pàgina oficial del videojoc)



# 9.Dificultats i millores futures

## 9.1 Dificultats

-Adaptar la pantalla del mòbil per a que no es sobreisca en horizontal.

## 9.2 Millores futures

-Passar del servidor de Node a Odoo mitjançant jsonrpc.

-Compatibilitat en iOS.

## 9.3 Que he aprés

-Que és la tecnologia nfc i per a que serveix.

# 10.Annexos

## 10.1 Github

El repositori del projecte en Github: <https://github.com/JoelFusterBosch/Projecte_Final_DAM_Gestio-_d_una_falla>

## 10.2 Bibliografia

1.Introducció

Sistemes mòbils més usats CNMC(Comisión Nacional de los Mercados y la Competencia): [https://www.cnmc.es/prensa/panel-hogares-usos-internet-20231103#:~:text=Un%2078%2C8%20%%20de%20los%20espa%C3%B1oles%20con,Internet%20y%20los%20servicios%20OTT%20en%20Espa%C3%B1a.](https://www.cnmc.es/prensa/panel-hogares-usos-internet-20231103%23:~:text=Un%2078%2C8%20%25%20de%20los%20espa%C3%B1oles%20con,Internet%20y%20los%20servicios%20OTT%20en%20Espa%C3%B1a.)

10.Annexos

<https://github.com/JoelFusterBosch/Projecte_Final_DAM_Gestio-_d_una_falla>