
This is the **published version** of the bachelor thesis:

Grau Lirio, Xavier. *Disseny i desenvolupament d'una aplicació mòbil multiplataforma per a la gestió intel·ligent i gamificada d'hàbits saludables*.
Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026
(Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334451>

under the terms of the  license.

LevelUpXGL: disseny i desenvolupament d'una aplicació mòbil multiplataforma per a la gestió intel·ligent i gamificada d'hàbits saludables

Xavier Grau Lirio - NIU 1670093

Resum— Aquest article presenta el disseny, desenvolupament i validació de LevelUpXGL, una aplicació mòbil multiplataforma orientada a la gestió d'hàbits saludables. El projecte integra entrenament, nutrició, perfil d'usuari, component social i gamificació dins d'un sistema client-servidor desplegat al núvol. La solució combina un frontend Flutter amb un backend Spring Boot, persistència PostgreSQL i autenticació mitjançant JWT. El treball s'ha desenvolupat amb una metodologia iterativa, incorporant millores funcionals i visuals a mesura que es validaven els mòduls. Els resultats mostren una versió funcional estable, amb proves superades i temps de resposta dels endpoints principals per sota del límit definit de 300 ms.

Paraules clau— Aplicació mòbil, Flutter, Spring Boot, PostgreSQL, gamificació, nutrició, entrenament, hàbits saludables, API REST, Google Cloud.

Abstract— This paper presents the design, development and validation of LevelUpXGL, a cross-platform mobile application focused on healthy habit management. The project integrates workout tracking, nutrition logging, user profile configuration, social interaction and gamification in a cloud-based client-server system. The solution combines a Flutter frontend, a Spring Boot backend, PostgreSQL persistence and JWT authentication. The work follows an iterative methodology, adding functional and visual improvements as each module is validated. The final result is a stable functional version, with successful backend tests and response times for the main endpoints below the required 300 ms threshold.

Index Terms— Mobile application, Flutter, Spring Boot, PostgreSQL, gamification, nutrition, workout tracking, healthy habits, REST API, Google Cloud.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

L'ús d'aplicacions mòbils per registrar activitat física i alimentació és habitual, però moltes solucions continuen actuant com a eines de registre aïllades. L'usuari ha d'introduir dades repetitives, consultar pantalles separades i interpretar per compte propi si les seves accions diàries contribueixen realment als seus objectius. Aquesta fricció redueix la constància i dificulta que el seguiment es converteixi en un hàbit sostingut.

El problema abordat en aquest TFG és la manca d'una experiència integrada que combini entrenament, nutrició i motivació en un mateix flux. LevelUpXGL neix amb la voluntat de convertir el seguiment saludable en una experiència més clara i lúdica: l'usuari registra sessions d'entrenament i aliments, consulta el progrés del dia, rep experiència en funció de les accions completades i visualitza recompenses com nivells, medalles, raxes i evolucions de personatge.

La proposta s'ha materialitzat com una aplicació mòbil multiplataforma amb arquitectura client-servidor. El frontend s'ha implementat amb Flutter, mentre que el backend exposa una API REST desenvolupada amb Spring Boot i desplegada a Google Cloud Run. Les dades es persisteixen en PostgreSQL, amb separació per usuari, validacions de negoci i control d'accés a través de JWT. L'objectiu no ha estat construir només una interfície atractiva, sinó un

- E-mail de contacte: 1670093@uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Ronny Romero Navarrete (Computació)
- Curs 2025/26

sistema complet, coherent i verificable que permeti demostrar criteris propis d'enginyeria del software: modularitat, integració, seguretat, persistència, rendiment i mantenibilitat.

La resta de l'article s'organitza de la manera següent. Primer es defineixen els objectius i el nivell d'assoliment. A continuació es presenta l'estat de l'art i les eines relacionades. Seguidament es descriu la metodologia i el disseny tècnic. Finalment es mostren els resultats, la discussió, les conclusions, l'ús de la IA generativa i la bibliografia.

2 OBJECTIUS

L'objectiu principal del treball és dissenyar i implementar una aplicació mòbil multiplataforma que permeti gestionar hàbits saludables de manera integrada, incorporant un sistema de gamificació desacoblat que reforci la motivació de l'usuari i faciliti la continuïtat del seguiment.

Aquest objectiu principal s'ha desglossat en objectius específics, definits de manera que cada un pugui ser verificat mitjançant funcionalitats o criteris tècnics concrets:

- O1. Definir una arquitectura client-servidor modular, amb separació entre frontend, backend i persistència.
- O2. Implementar registre, inici de sessió i renovació de sessió amb tokens JWT.
- O3. Desenvolupar el mòdul d'entrenament: rutines, exercicis, sèries i registre de sessions.
- O4. Implementar el mòdul de nutrició: aliments, registres diaris, càlcul de calories i macronutrients, objectius automàtics i objectius editables.

- O5. Dissenyar un motor de gamificació basat en esdeveniments, amb XP, nivells, medalles, raxes i evolució visual del personatge.

- O6. Afegir perfil d'usuari, dades físiques, selecció de personatge i una capa social basada en seguiment d'altres usuaris.

- O7. Validar el sistema mitjançant proves funcionals, proves backend i mètriques de rendiment dels endpoints habituals.

- O8. Documentar el procés de disseny, desenvolupament, problemes trobats i conclusions finals.

El criteri general d'assoliment és obtenir una versió estable en què el flux principal sigui funcional: registre, configuració de perfil, entrenament, nutrició, progrés i recompenses. Els objectius opcionals plantejats inicialment, com la lectura de codis de barres o recomanacions més avançades, s'han considerat línies futures quan no eren imprescindibles per completar el nucli del sistema.

2.1 Requisits i criteris de qualitat

A més dels objectius funcionals, el projecte ha estat guiat per requisits no funcionals. Aquests requisits són especialment importants en un TFG de la meua menció "Enginyeria del Software", perquè permeten valorar la solució més enllà de la quantitat de pantalles implementades.

TAULA 1. Requisits no funcionals i estratègia aplicada.

Requisit	Criteri	Aplicació al projecte
Rendiment	Endpoints habituals <300 ms	Mesures sobre rutines, perfil i progrés
Seguretat	Sessió protegida i dades per usuari	JWT, refresh token i filtres per usuari autenticat
Mantenibilitat	Codi modular i extensible	Separació per features i per mòduls backend
Fiabilitat	Persistència i validacions	PostgreSQL, constraints, validacions i errors coherents
Usabilitat	Flux clar i feedback	Pantalles amb càrrega, errors, confirmacions i estil coherent
Privacitat	Evitar accés creuat	Relacions i consultes restringides per usuari

Aquest conjunt de requisits ha condicionat decisions concretes. Per exemple, el rendiment ha motivat índexs per usuari i data; la seguretat ha portat a repositoris que filtren per usuari; i la usabilitat ha justificat una iteració específica de redisseny visual abans de tancar la validació tècnica.

3 ESTAT DE L'ART

L'estat de l'art s'ha analitzat des de dues perspectives: productes existents relacionats amb el problema i tecnologies utilitzades per construir la solució. Aquesta revisió ha permès delimitar el valor diferencial de LevelUpXGL i justificar les decisions tècniques.

3.1 Productes relacionats

MyFitnessPal és una de les aplicacions de referència en seguiment nutricional. Ofereix registre d'aliments, càlcul de calories i macronutrients, i una base de dades extensa [8].

Tanmateix, el seu focus principal és la nutrició, i la gamificació no és el centre de l'experiència. LevelUpXGL adopta la idea del resum nutricional diari, però la connecta directament amb el progrés de l'usuari.

Strava és un referent en registre d'activitat física i component social. Permet registrar activitats i compartir-les amb altres usuaris [9]. Tot i això, està especialment orientada a activitats cardiovasculars i GPS, mentre que LevelUpXGL s'ha centrat en rutines de gimnàs, sèries, repeticions i càrregues.

Fitbod destaca en planificació d'entrenaments de força adaptats a objectius, nivell i equipament [10]. Aquesta aproximació és propera a una possible evolució futura del projecte, però el TFG s'ha enfocat primer en construir un sistema propi de registre, persistència i gamificació. Habilita, per la seva banda, és un exemple clar de gamificació aplicada a tasques quotidianes [11], però no resol de forma específica el seguiment nutricional i d'entrenament.

La conclusió d'aquesta revisió és que hi ha eines potents en àmbits parcials, però l'oportunitat del projecte és combinar registre saludable, component visual i motor de progrés en una sola experiència. Això no implica competir amb productes comercials consolidats, sinó demostrar una solució integrada i tècnicament coherent dins de l'abast del TFG.

3.2 Tecnologies utilitzades

Flutter s'ha escollit per desenvolupar una aplicació multi-plataforma amb una única base de codi i una interfície personalitzada [1]. Riverpod s'ha utilitzat per gestionar estat i dependències al frontend [2], i GoRouter per definir una navegació clara entre pantalles [3].

Al backend, Spring Boot proporciona una base adequada per construir APIs REST modulars [4], mentre que Spring Security permet incorporar autenticació i autorització. PostgreSQL s'ha emprat com a sistema de persistència relacional [5], adequat per modelar usuaris, rutines, aliments, sessions, esdeveniments i recompenses amb integritat referencial. Google Cloud Run i Cloud SQL han permès desplegar la solució en un entorn cloud real [6], [7]. Finalment, Flyway ha facilitat l'evolució del model de dades mitjançant migracions versionades [12].

4 METODOLOGIA I DISSENY DE LA SOLUCIÓ

La metodologia seguida ha estat iterativa i incremental. El projecte es va planificar en vuit iteracions, cadascuna amb un entregable funcional. Aquesta estratègia ha estat adequada perquè permetia validar aviat les parts crítiques - autenticació, persistència, integració frontend-backend i desplegament - abans d'afegir mòduls amb més dependències.

El flux habitual de treball ha estat: definició del model de dades, implementació de lògica backend, exposició d'endpoints REST, integració al frontend, desenvolupament de pantalles, proves funcionals i correcció d'errors. Aquest ordre ha evitat que la interfície es desenvolupés sobre contractes d'API inestables i ha facilitat detectar inconsistències de models o permisos.

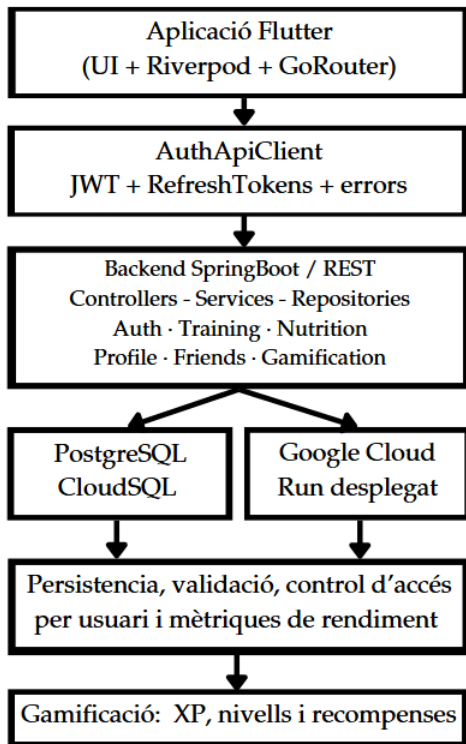


Figura 1. Arquitectura lògica simplificada de LevelUpXGL.

La planificació inicial es va seguir en termes generals. Les primeres iteracions van consolidar l'arquitectura, l'autenticació, l'entrenament, el perfil i el component social. Posteriorment es van implementar nutrició i gamificació. Un ajust rellevant va ser invertir l'ordre entre la iteració de validació tècnica i la iteració de millores visuals: abans d'optimitzar i validar de forma intensiva, es va considerar més útil establir la navegació, reduir inconsistències visuals i millorar el feedback de les pantalles.

Durant el desenvolupament també es va valorar migrar el backend cap a una arquitectura hexagonal. La idea era millorar la separació entre domini, ports i adaptadors. No obstant això, l'intent va generar problemes d'integració amb repositoris, serveis i configuració ja implementada. Dins del temps disponible, es va prioritzar mantenir una arquitectura per capes modular, que ja era funcional i permetia complir els objectius del projecte. Aquesta decisió va ser coherent per no introduir una refactorització estructural d'alt risc si no aporta un benefici verificable abans de l'entrega.

4.1 Arquitectura del backend

El backend s'ha organitzat en mòduls funcionals. Cada mòdul agrupa controladors, DTOs, serveis, repositoris i entitats relacionades amb una responsabilitat concreta. Aquesta organització no és una arquitectura hexagonal estricta, però sí una arquitectura per capes amb separació suficient per mantenir el codi comprensible i ampliable.

Els controladors defineixen el contracte REST i reben peticions validades amb anotacions de Jakarta Validation. Els serveis concentren la lògica de negoci: creació de rutines, verificació de propietat, càlcul de resums, sincronització nutricional o recalcul de gamificació. Els repositoris

encapsulen l'accés a dades mitjançant Spring Data JPA. Finalment, les entitats JPA representen el model persistent i defineixen relacions com usuari-rutines, rutina-exercicis, sessió-sèries, usuari-aliments, usuari-medalles i usuari-esdeveniments.

Aquesta estructura ha resultat adequada per al projecte perquè cada iteració podia afegir un mòdul nou sense reescriure els anteriors. Per exemple, la gamificació es va integrar posteriorment amb nutrició i entrenament, però mantenint el seu propi servei i els seus propis repositoris. Això facilita provar cada part i redueix l'acoblament entre pantalles i regles de negoci.

Un altre aspecte important és el tractament d'errors. El backend inclou un manejador global que transforma errors de validació i excepcions HTTP en respostes JSON consistents. Això ha estat útil al frontend, perquè permet mostrar missatges més clars i no dependre de formats diferents segons l'endpoint. També s'ha utilitzat OpenAPI per tenir una referència del contracte REST durant el desenvolupament.

4.2 Model de dades i persistència

El model de dades s'ha dissenyat de forma incremental. Primer es van crear usuaris, rols i relació usuari-rol. Després es van afegir rutines, exercicis i sèries. Més endavant es van incorporar perfil físic, seguiments entre usuaris, aliments guardats, registres nutricionals, sessions d'entrenament, progrés de gamificació, medalles, esdeveniments i objectius nutricionals.

La base de dades utilitza claus foranes amb eliminació en cascada quan té sentit, com en rutines i exercicis d'un usuari. També s'han definit restriccions per evitar dades incoherents: noms de rutina duplicats per usuari, ordres repetits dins d'una rutina, número de sèrie duplicat dins d'un exercici o relacions socials duplicades. Aquestes restriccions són importants perquè no es confia únicament en la validació del frontend.

TAULA 2. Principals entitats de persistència.

Àrea	Taules principals	Finalitat
Identitat	users, roles, user_roles	Autenticació i permisos bàsics
Entrenament	routines, routine_exercises, exercise_sets	Planificació de rutines
Històric	training_sessions, training_session_exercises, training_session_sets	Sessions executades
Nutrició	saved_foods, food_logs, user_nutrition_goals	Registre i objectius diaris
Social	user_follows	Seguiment entre usuaris
Gamificació	gamification_events, user_gamification_progress, user_medals	XP, nivells i recompenses

També s'han creat índexs sobre camps consultats habitualment, com usuari-data en registres nutricionals i sessions d'entrenament. Aquesta decisió està alineada amb el requisit de rendiment, ja que les pantalles diàries consulten dades per usuari i data de forma repetida.

4.3 Arquitectura del frontend

El frontend s'ha estructurat per features, seguint una separació entre models, fonts de dades, repositoris, provedors i pantalles. Aquesta organització ha estat útil perquè cada mòdul - autenticació, entrenament, nutrició, perfil o gamificació - manté les seves pròpies classes i no concentra tota la lògica en una única capa visual.

Els models transformen el JSON rebut de l'API en objectes Dart. Els datasources encapsulen les crides HTTP. Els repositoris ofereixen una interfície més propera al domini de l'aplicació, i els provedors de Riverpod exposen dependències i estat a les pantalles. En el cas de la gamificació, per exemple, la pantalla de progrés consumeix un provider que obté el progrés actual i es pot invalidar quan nutrició o entrenament modifiquen l'XP.

La navegació s'ha resolt amb rutes diferenciades per login, registre, home, perfil, entrenament, nutrició, progrés, calendari i amics. El camp `profileCompleted` és especialment important perquè decideix si després de l'autenticació l'usuari ha d'anar al perfil inicial o a la pantalla principal. Aquesta decisió evita fluxos incomplets i millora la coherència de l'experiència. El frontend també ha requerit treball específic en gestió d'estats. S'han incorporat indicadors de càrrega, missatges d'error, botons de reintent, `SnackBar`s i confirmacions. Aquestes decisions no són secundàries: sense elles, una API correcta pot donar una experiència pobre si l'usuari no entén què està passant o si una operació tarda uns segons.

4.4 Desplegament i configuració

El backend s'ha desplegat a Google Cloud Run, una plataforma adequada per executar serveis sense gestionar servidors directament. La base de dades PostgreSQL s'ha allotjat a Cloud SQL. Les variables sensibles, com credencials de base de dades i secret JWT, s'han gestionat com a variables d'entorn o secrets, evitant codificar-les directament al repositori.

Aquest desplegament ha permès provar l'aplicació des d'un dispositiu mòbil real, no només en local. Aquesta diferència és rellevant perquè apareixen problemes que no sempre són visibles durant el desenvolupament local: latència de xarxa, configuració de CORS o rutes, credencials, connexió a base de dades i temps de resposta del servei desplegat.

La validació sobre un entorn cloud també aporta valor a la memòria del TFG, ja que demostra que el projecte no és només una maqueta local sinó un sistema distribuït amb frontend mòbil, API remota i persistència externa.

4.5 Planificació real i desviacions

La planificació original definia una seqüència d'iteracions amb entregables acumulatius. Aquesta estructura s'ha mantingut, però el projecte ha requerit ajustos raonables. Les iteracions inicials van ser més tècniques, mentre que les finals van concentrar integració, estabilització i presentació visual.

TAULA 3. Evolució de les iteracions del projecte.

It	Focus	Resultat	Comentari
1	Arquitectura i auth	Completada	Base client-servidor i JWT
2	Entrenament	Completada	Rutines i sessions
3	Perfil i social	Completada	Perfil i seguiment
4	Nutrició	Completada	Macros, calories i aliments freqüents
5	Gamificació	Completada	XP, nivells, raxes i medalles
8	UX i estabilitat	Avançada	Es va executar abans de la validació final
6	Validació tècnica	Completada	Tests i mètriques <300 ms
7	Extres opcionals	Parcial - Altres opcionals fets	Barcode i recomanacions queden futures

El canvi més important va ser avançar la iteració d'UX. Aquesta decisió va tenir impacte positiu perquè va permetre validar posteriorment un producte més estable visualment. Si s'hagués fet la validació tècnica abans del redisseny, part dels resultats podrien haver quedat desactualitzats per canvis posteriors en navegació i fluxos. La metodologia iterativa també ha ajudat a gestionar incertesa. En un projecte amb backend, frontend, base de dades, cloud i disseny visual, és habitual que apareguin dependències no previstes. Treballar per entregables ha permès tancar funcionalitats, detectar errors i documentar decisions abans de passar al següent bloc.

5 DESENVOLUPAMENT IMPLEMENTAT

5.1 Autenticació, perfil i seguretat

El sistema d'autenticació permet registrar usuaris, iniciar sessió i renovar el token d'accés amb un refresh token. El backend genera tokens JWT i associa cada petició protegida a l'usuari autenticat. Al frontend, els tokens es guarden amb Flutter Secure Storage i les peticions passen per un client HTTP centralitzat que afegeix la capçalera `Authorization` i reintenta la petició quan és necessari renovar la sessió.

La incorporació de `profileCompleted` ha permès resoldre un problema habitual en aplicacions amb perfil inicial: després del registre, l'usuari no ha d'entrar directament al flux principal si encara no ha indicat dades físiques bàsiques. Aquest camp condueix la navegació cap a la configuració inicial, on es defineixen altura, pes actual, pes objectiu, edat, objectiu i personatge.

La seguretat s'ha reforçat també a nivell de dades. Les consultes sensibles es fan filtrant per l'usuari autenticat, evitant accessos creuats. A més, la base de dades incorpora claus foranes, restriccions d'unicitat i comprovacions de valors, com pesos no negatius o sèries positives. El backend centralitza els errors de validació amb un manejador global, facilitant respostes consistents al frontend.

5.2 Mòdul d'entrenament

El mòdul d'entrenament permet crear i editar rutines, afegir exercicis ordenats i definir sèries amb repeticions i pes. El backend ofereix endpoints per a rutines, exercicis i sessions, i el frontend inclou pantalles per llistar rutines, editar-les i registrar sessions d'entrenament. Un punt important ha estat separar la rutina planificada de la sessió real executada. Quan l'usuari guarda una sessió, es persisteix una còpia de la informació rellevant: nom de rutina, exercicis, ordre, sèries, repeticions i pesos. Això evita que canvis futurs a la rutina modifiquin l'històric d'entrenaments. També es calcula un resum de sessió amb volum total i diferències respecte als valors previstos, oferint feedback immediat a l'usuari. El desenvolupament d'aquest mòdul va revelar errors de sincronització entre el frontend i el backend, especialment quan es modificaven exercicis i després s'obria el registre de sessió. La solució va consistir a construir el draft de sessió a partir de la informació actual de la rutina i conservar l'estat fins que l'usuari decideix guardar o descartar la sessió.

5.3 Mòdul de nutrició

El mòdul nutricional permet registrar aliments amb calories, proteïnes, hidrats de carboni i greixos. També permet guardar aliments freqüents i reutilitzar-los, reduint la fricció d'introducció de dades. La pantalla principal mostra un resum del dia amb progrés de calories i macros, i permet editar objectius.

Inicialment els objectius nutricionals es generaven automàticament a partir del perfil. Durant el desenvolupament es va detectar que aquest enfocament era massa rígid. Per això es va afegir la possibilitat de mantenir objectius automàtics o modificar-los manualment. Aquesta decisió millora la usabilitat perquè diferents usuaris poden tenir necessitats específiques tot i compartir dades físiques similars.

Una millora d'experiència incorporada és el botó d'oli. En comptes d'obligar l'usuari a introduir manualment una quantitat petita però freqüent, una pulsació mantinguda calcula els grams segons una configuració de grams per segon i afegeix l'aliment amb les calories i greixos corresponents. Aquest tipus de microinteracció mostra com el projecte ha intentat reduir fricció en casos d'ús reals.

5.4 Perfil social i visualització d'usuaris

El component social s'ha implementat com un sistema de seguiment unidireccional. L'usuari pot cercar altres usuaris, afegir-los com a coneguts i consultar una visualització limitada del seu perfil. Aquesta decisió simplifica el model respecte a una amistat mútua completa, però és suficient per demostrar relacions entre usuaris, restriccions de privacitat i pantalles de perfil conegut.

El model de dades impedeix seguir-se a un mateix i evita duplicats mitjançant restriccions d'unicitat. La informació visible d'altres perfils es limita a dades bàsiques i al personatge seleccionat, sense permetre modificació de dades alienes.

5.5 Motor de gamificació

La gamificació és el mòdul que diferencia LevelUpXGL d'un simple tracker. El sistema atorga experiència per accions saludables: completar objectius de macronutrients, assolir calories, registrar una sessió d'entrenament i mantenir una raxa. Els nivells van de l'1 al 100, les medalles apareixen a partir del nivell 5 i cada deu nivells el personatge evoluciona visualment. La decisió més important ha estat modelar la gamificació a partir d'esdeveniments. En lloc de dependre només d'increments acumulatius, cada acció rellevant genera un registre amb tipus, data, XP i descripció. El progrés es recalcula a partir d'aquest històric. Això resol inconsistències detectades durant les proves, com diferències entre XP total i nivell real, o recompenses que continuaven visibles després d'eliminar aliments o canviar objectius nutricionals.

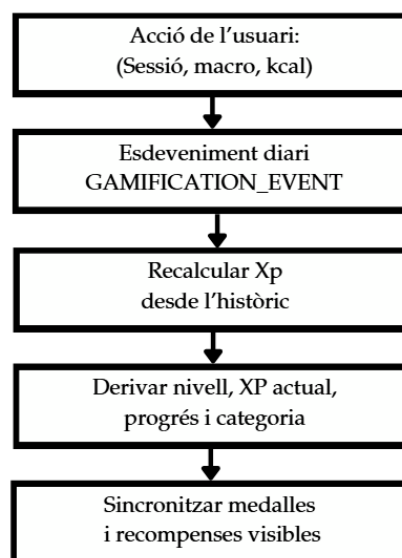


Figura 2. Flux de recalculació del progrés a partir d'esdeveniments.

Aquesta aproximació fa que l'estat derivat - nivell, XP actual, percentatge, categoria, medalles i recompenses - sigui reconstruïble. En termes de qualitat del software, redueix el risc d'inconsistències i facilita corregir esdeveniments passats sense haver de modificar manualment tots els camps acumulats.

5.5.1 Regles de progressió i recompenses

Les regles de progressió s'han definit amb criteris simples per ser comprensibles per l'usuari i fàcils de validar. El sistema premia accions concretes i limita alguns premis a una vegada per dia per evitar increments artificials. Aquesta simplicitat és intencionada: una gamificació massa opaca pot confondre l'usuari i dificultar la depuració del sistema.

TAULA 4. Regles principals de XP.

Acció	XP	Limitació	Efecte
Completar proteïna	1	Diària	Progrés nutricional
Completar carbs.	1	Diària	Progrés nutricional
Completar greixos	1	Diària	Progrés nutricional
Arribar a kcal	5	Diària	Objectiu energètic
Guardar entrenament	2	Màx. 1/dia	Activitat física
Raxa diària	Variable	Segons dia	Motivació continuada
Nivell 5, 15, 25...	-	Per nivell	Medalla visible
Nivell 10, 20, 30...	-	Per nivell	Evolució de personatge

La corba d'experiència és creixent. Els primers nivells requereixen poca XP per donar sensació d'avanç inicial, mentre que els nivells alts requereixen més esforç. Aquesta decisió busca equilibrar motivació immediata i progressió a llarg termini. El nivell màxim s'ha fixat en 100, que és un límit suficient per representar una evolució extensa sense complicar visualment el sistema.

Les medalles s'han situat cada deu nivells començant pel nivell 5, perquè així l'usuari rep una recompensa intermèdia abans de cada evolució de categoria. Les evolucions de personatge cada deu nivells reforcen la sensació de canvi visual. Aquesta combinació genera dos ritmes de recompensa: un de curt-mitjà termini amb medalles i un de més destacat amb canvis de personatge.

Un avantatge del disseny basat en respostes de backend és que el frontend no necessita conèixer totes les regles internes. La pantalla només mostra el nivell, XP, percentatge, imatge, medalles i recompenses que retorna l'API. Això facilita modificar regles futures sense reprogramar totes les pantalles.

5.6 Experiència d'usuari i interfície

La interfície adopta una estètica de fantasia amb colors foscos, daurats, fons il·lustrats i personatges evolutius. Aquesta decisió no és només estètica: reforça la metàfora de progrés i converteix accions repetitives en passos d'una aventura. Durant la fase final es van redissenyar diverses pantalles per millorar jerarquia visual, consistència i feedback.

Les pantalles principals inclouen login, registre, home, entrenament, nutrició, progrés, calendari, perfil, amics i perfil conegut. S'han afegit estats de càrrega, missatges d'error, refresh manual i confirmacions en accions destructives, com eliminar exercicis o aliments.



Figura 3. Flux bàsic: accés, pantalla principal i registre de sessió i nutrició

5.7 Històric, calendari i coherència temporal

A més de les pantalles principals, el projecte incorpora una visió d'històric i calendari. Aquesta funcionalitat és rellevant perquè els hàbits saludables no s'avaluen només en una acció puntual, sinó en la continuïtat. El calendari permet consultar dies concrets i diferenciar activitats de dieta i gimnàs.

Des del punt de vista tècnic, aquesta funcionalitat depèn d'un model de dades amb dates normalitzades. Els registres d'aliments utilitzen `log_date`, les sessions d'entrenament utilitzen `session_date` i els esdeveniments de gamificació utilitzen `event_date`. Aquesta coherència facilita consultar un dia i reconstruir què va passar en entrenament, nutrició i gamificació.

La gestió temporal també és important per limitar XP. Per exemple, la sessió d'entrenament només ha de premiar una vegada per dia i els objectius de macro o calories han de sincronitzar-se amb el dia corresponent. Sense aquesta separació temporal, l'usuari podria rebre experiència duplicada o perdre recompenses de forma incorrecta.

5.8 Integració entre mòduls

Una de les parts més complexes del projecte ha estat aconseguir que els mòduls treballassin com un sistema únic. Entrenament i nutrició són funcionalitats diferents, però totes dues poden generar progrés. Perfil afecta objectius nutricionals, nutrició afecta XP, XP afecta nivell i nivell afecta recompenses visuals. Aquesta cadena d'efectes ha requerit definir bé quines dades són fonts de veritat i quines són derivades.

En el projecte, els registres reals de l'usuari - aliments, sessions i esdeveniments - actuen com a dades base. Els resums diaris, el progrés de barres, les medalles i les recompenses són dades calculades o sincronitzades. Aquesta distinció ha estat clau per resoldre errors, perquè quan una dada derivada no coincidía, s'ha pogut tornar a calcular a partir de l'històric.

Al frontend, la integració s'ha gestionat invalidant providers quan una acció afecta altres pantalles. Per exemple, després d'afegir un aliment es recarrega el resum nutricional i es força la sincronització de gamificació. Després de modificar el perfil, també es pot invalidar el progrés per reflectir canvis de personatge o objectius.

5.9 Seguretat, privacitat i control d'accés

Les dades tractades per l'aplicació són personals: correu electrònic, edat, altura, pes, objectius, entrenaments i alimentació. Per aquest motiu, la privacitat s'ha considerat des del disseny. El sistema no exposa dades d'altres usuaris excepte una visualització limitada de perfil conegut, i no permet modificar informació aliena.

Al backend, moltes operacions utilitzen el patró de buscar entitats per identificador i usuari. Això és més segur que buscar només per identificador, perquè evita que un usuari pugui accedir a una rutina o aliment d'un altre usuari si endevina un id. Les relacions socials també incorporen restriccions per evitar seguir-se a un mateix o duplicar la mateixa relació.

Tot i aquestes mesures, el sistema no s'ha d'interpretar com una implementació completa de compliment normatiu.

Per a una versió comercial caldrien polítiques de privacitat, mecanismes d'exportació i eliminació de dades, xifrat addicional quan fos necessari i revisió de dependències. En l'abast del TFG, s'han aplicat bones pràctiques bàsiques coherents amb un prototip acadèmic.

5.10 Modificació opcional: mode local/offline

Com a extensió opcional, s'ha plantejat una versió local/offline de l'aplicació. Aquesta variant elimina la dependència del backend i permet que l'aplicació funcioni directament al dispositiu, sense necessitat de connexió constant ni servidor desplegat.

Per adaptar-la, s'eliminen les parts que requereixen un entorn multiusuari, com el login i l'apartat social. La lògica que abans es trobava als serveis del backend es trasllada als repositoris del frontend, que passen a gestionar les dades locals i els càlculs principals dels mòduls d'entrenament, nutrició i gamificació.

Aquesta alternativa redueix la complexitat d'infraestructura i facilita conservar una versió funcional encara que el servidor cloud no estigui disponible. Tot i això, també limita funcionalitats com la sincronització entre dispositius, la gestió d'usuaris i les relacions socials, per la qual cosa es considera una extensió complementària i no substitueix l'arquitectura principal client-servidor.

6 AVALUACIÓ, RESULTATS I DISCUSSIÓ

6.1 Resultats funcionals

La versió final permet completar el flux principal plantejat: registre, configuració de perfil, entrenament, nutrició, progrés i recompenses. El backend està desplegat a Google Cloud, la base de dades està allotjada en PostgreSQL i el frontend consumeix l'API mitjançant un client autenticat. El resultat funcional es pot sintetitzar en els blocs següents: autenticació amb JWT i refresh token; perfil físic i selecció de personatge; rutines, exercicis i sessions; registre nutricional diari amb aliments freqüents; objectius nutricionals automàtics i editables; relacions socials; motor de XP, nivells, raxes, medalles i evolucions; i pantalles finals amb estil visual coherent.

TAULA 5. Nivell d'assoliment dels objectius principals.

Objectiu	Assoliment	Evidència
O1 Arquitectura	Assolit	Flutter + Spring Boot + PostgreSQL + Cloud Run
O2 Auth JWT	Assolit	Login, registre, refresh i rutes protegides
O3 Entrenament	Assolit	Rutines, exercicis, sèries i sessions persistents
O4 Nutrició	Assolit	Aliments, macros, calories, objectius i freqüents
O5 Gamificació	Assolit	XP, nivells, medalles, raxes i evolució
O6 Perfil/social	Assolit parcial	Perfil complet i seguiment unidireccional
O7 Validació	Assolit	Tests backend i endpoints <300 ms
Opcional barcode	No assolit	Queda com a línia futura

6.2 Escenaris de validació funcional

La validació funcional s'ha realitzat recurrent els fluxos principals de l'aplicació. L'objectiu era comprovar no només que cada pantalla funcionés aïlladament, sinó que les accions d'un mòdul tinguessin efecte en la resta del sistema quan corresponia.

TAULA 6. Escenaris funcionals validats.

Escenari	Acció validada	Resultat esperat
Alta d'usuari	Registre i login	Token vàlid i redirecció a perfil
Perfil inicial	Completar dades físiques	Accés a home amb objectius calculats
Rutina	Crear exercicis i sèries	Persistència i llistat actualitzat
Sessió	Guardar entrenament	Històric creat i XP diària aplicada
Nutrició	Afegir o eliminar aliment	Macros recalculats i XP sincronitzada
Gamificació	Consultar progrés	Nivell, barra, medalles i personatge coherents
Social	Afegir conegut	Perfil visible sense permisos d'edició

Aquests escenaris han servit per detectar errors reals d'integració. Un exemple va ser la necessitat de recalcular XP quan un aliment es retirava i l'objectiu deixava d'estar complet. Un altre va ser assegurar que la sessió d'entrenament utilitzés la versió actual dels exercicis afegits a la rutina.

6.3 Qualitat del codi i mantenibilitat

La qualitat del codi s'ha treballat amb diverses decisions. Al backend, les validacions es troben en DTOs i serveis, la lògica de negoci no queda dins dels controladors i els repositoris expressen consultes específiques, com trobar una rutina per identificador i usuari. Al frontend, la separació per features evita que totes les pantalles comparteixin un únic fitxer de lògica i facilita localitzar errors.

La mantenibilitat també s'ha tingut en compte en el model de gamificació. El fet que les recompenses depenguin del nivell i que el progrés es derivi d'esdeveniments permet canviar regles sense modificar la UI. Per exemple, es podria ajustar l'XP atorgada per macronutrients o entrenament sense alterar la pantalla de progrés, sempre que l'API mantingui el mateix contracte de resposta.

Un altre element de mantenibilitat és la consistència visual. Al frontend s'han reutilitzat colors, estils, targetes i patrons de capçalera. Això redueix el cost de modificar el disseny i evita que cada pantalla evolucioni de forma independent. La millora visual de la iteració final va ser necessària precisament perquè la suma de funcionalitats inicials havia generat pantalles amb criteris diferents.

6.4 Resultats de rendiment

La validació tècnica s'ha basat en proves backend i en una mesura senzilla dels endpoints habituals. L'objectiu no funcional RNF-02 establia que les operacions comunes del backend havien de respondre en menys de 300 ms en condicions normals. Les mesures realitzades mostren valors molt inferiors a aquest llindar.

TAULA 7. Mesura de rendiment dels endpoints habituals.

Endpoint	Mitjana	Màxim	Errors
/api/routines	38,4 ms	45 ms	0/5
/api/v1/profile	47 ms	71 ms	0/5
/api/v1/gamification/progress	54,6 ms	59 ms	0/5

A més, l'execució de proves del backend va finalitzar amb BUILD SUCCESSFUL, cosa que indica que el projecte podia compilar i superar la bateria de proves definida.

6.5 Problemes trobats i solucions

El primer grup de problemes va aparèixer en la integració frontend-backend. Inicialment, els errors d'autenticació i renovació de token generaven comportaments diferents segons la pantalla. La solució va ser centralitzar les peticions autenticades en un client comú, amb construcció d'headers, refresh automàtic i parseig coherent d'errors.

El segon problema important va ser la sincronització de la gamificació. El sistema acumulatiu inicial provocava incoherències quan l'usuari eliminava aliments, modificava objectius o reduïa el nivell durant proves. La solució basada en esdeveniments com a font de veritat va permetre recalculer el progrés i sincronitzar medalles d'acord amb el nivell real.

També es van observar problemes de càrrega a mesura que augmentaven els mòduls. Les pantalles amb diverses dades - per exemple nutrició i progrés - podien requerir recarregar informació de backend i invalidar providers. Els temps actuals són acceptables, però la discussió final identifica com a millora futura l'ús de memòria cau, paginació d'històrics i optimització de consultes.

Finalment, la interfície va requerir una iteració específica. Les primeres versions eren funcionals, però algunes pantalles tenien massa informació o una jerarquia visual poc clara. El redisseny va millorar consistència, navegació i feedback, i va fer que el producte final fos més presentable per a una defensa de TFG.

6.5.1 Casos límit revisats

La fase de correcció va incloure diversos casos límit. El primer era eliminar un aliment que havia fet que un objectiu nutricional estigués complert. En aquest cas, el sistema havia de retirar l'esdeveniment corresponent, eliminar també el bonus de raxa si depenia d'aquell dia i recalculer XP i nivell. Sense aquesta revisió, l'usuari podia mantenir XP no merescuda.

Un segon cas límit era modificar objectius nutricionals després d'haver registrat aliments. Si l'usuari augmenta l'objectiu de calories, pot deixar de tenir-lo complert; si el redueix, pot passar a complir-lo. Això obliga a sincronitzar gamificació després de canviar objectius, no només després d'afegir aliments.

Un tercer cas límit era l'entrenament diari. El sistema havia de premiar l'activitat física, però evitant que múltiples guardats de prova generessin XP indefinida el mateix dia. Per això s'ha utilitzat una comprovació d'esdeveniment existent per usuari, tipus i data.

També es van revisar errors de navegació i càrrega. Quan una pantalla depèn de dades remotes, cal decidir què passa si la petició falla, si el token caduca o si l'usuari torna enrere

abans de rebre la resposta. Les solucions adoptades han estat comprovar que el widget continuï muntat abans de fer setState, mostrar errors llegibles i oferir botons de reintent.

6.5.2 Verificació de dades persistides

La validació no s'ha limitat a veure pantalles. També s'han consultat dades persistides en la base de dades per comprovar que els registres creats per l'aplicació quedaven emmagatzemats correctament. Les captures de l'apèndix mostren consultes sobre taules i dades reals, cosa que reforça l'evidència que el backend persisteix rutines, exercicis, sèries i esdeveniments.

Aquesta comprovació és important perquè una aplicació pot semblar correcta si només manté estat local temporal, però el requisit del TFG era tenir persistència real. En aquest projecte, reiniciar l'aplicació o consultar des d'un altre moment conserva les dades perquè el backend i PostgreSQL actuen com a font persistent.

6.6 Discussió dels resultats

Els resultats obtinguts són satisfactoris perquè el sistema resol el problema principal dins de l'abast proposat: uneix entrenament, nutrició i gamificació en una sola aplicació, amb persistència real, backend desplegat i validació tècnica. La part més rellevant no és cada funcionalitat per separat, sinó la integració entre mòduls: una sessió o un aliment poden modificar objectius, raxes i progrés visual.

6.7 Avaluació dels requisits no funcionals

L'avaluació no funcional mostra que el projecte compleix raonablement els criteris definits. El rendiment és positiu en els endpoints mesurats; la seguretat bàsica s'ha cobert amb tokens, control d'accés i repositoris filtrats; la fiabilitat es recolza en persistència i restriccions de base de dades; i la usabilitat ha millorat amb la iteració visual final.

La mantenibilitat és acceptable per a l'escala del projecte. La divisió per mòduls permet localitzar canvis i afegir noves funcions. Per exemple, incorporar lectura de codis de barres podria afegir un datasource al mòdul de nutrició i un endpoint de consulta sense modificar l'entrenament. Igualment, afegir noves recompenses podria fer-se ampliant la resposta de gamificació i les regles del servei.

La principal àrea de millora és l'observabilitat. Actualment el projecte disposa de mètriques bàsiques del servei cloud i comprovacions manuals, però una aplicació en producció hauria d'incloure logs estructurats, traça d'errors, alertes i dashboards més específics. Aquesta extensió ajudaria a detectar problemes de càrrega, errors d'API i ús real de cada pantalla.

6.8 Impacte sobre l'objectiu de motivació

El projecte no ha realitzat un estudi amb usuaris suficient per afirmar estadísticament que millora l'adherència. Tanmateix, sí que ha implementat mecanismes dissenyats per atacar la falta de motivació: feedback immediat, barra d'experiència visible, recompenses periòdiques, personatges evolutius i raxes. Aquests elements converteixen activitats repetitives en objectius petits i visualment reconeixibles.

La decisió de connectar nutrició i entrenament amb el mateix sistema de XP és rellevant. L'usuari no veu cada

acció com una dada aïllada, sinó com una contribució al progrés global. Aquesta integració és el principal valor conceptual de LevelUpXGL respecte a una aplicació que només llista calories o sessions.

En una fase futura, aquest impacte podria validar-se amb proves d'usabilitat: temps necessari per registrar un aliment, nombre de passos per guardar una sessió, comprensió del sistema de recompenses, percepció de motivació i preferència entre pantalles abans i després del redisseny.

6.9 Comparació final amb l'estat de l'art

Comparat amb MyFitnessPal, LevelUpXGL és menys complet en base de dades d'aliments i automatització del registre, però integra la nutrició amb un progrés de personatge que dona una resposta més lúdica. Comparat amb Strava, el projecte és menys potent en comunitat i GPS, però s'adapta millor a rutines de força i sèries de gimnàs. Comparat amb Fitbod, no genera entrenaments intel·ligents avançats, però ofereix una arquitectura pròpia i un model integrat amb nutrició. Comparat amb Habitica, aplica gamificació a un domini concret de salut, no a tasques genèriques.

En general s'ha buscat una alternativa més simple on pugués gestionar els teus hàbits de forma totalment autònoma i lliure. Està enfocat llavors, en un públic experimentat que vol gestionar aspectes del dia a dia de manera més còmoda al extreure indicacions i ajudes innecessaries per aquest tipus d'usuari.

6.10 Aprenentatges del projecte

Un dels aprenentatges principals és que la complexitat no apareix només en implementar pantalles, sinó en mantenir coherència entre dades. La gamificació, per exemple, semblava inicialment una suma d'XP, però en realitat requeria gestionar eliminacions, canvis d'objectius, límits diaris i recompenses derivades. Aquesta experiència ha reforçat la importància de definir fonts de veritat i dades calculades. Un altre aprenentatge és que l'experiència d'usuari ha de considerar-se durant tot el projecte. En les primeres iteracions, les pantalles funcionaven, però no sempre eren intuïtives. Quan el nombre de mòduls va créixer, es va fer evident que la navegació, el disseny visual i el feedback eren part essencial de la qualitat del software, no només una capa estètica final.

Finalment, el desplegament cloud ha permès entendre problemes propis d'un sistema real: variables d'entorn, secrets, connexió amb base de dades, latències i comprovació de logs o mètriques. Aquest procés ha aportat una visió més completa que desenvolupar únicament en local.

7 CONCLUSIONS

El TFG ha assolit l'objectiu principal de construir una aplicació mòbil multiplataforma per a la gestió gamificada d'hàbits saludables. LevelUpXGL disposa d'una arquitectura client-servidor funcional, backend desplegat, persistència PostgreSQL, autenticació JWT i mòduls integrats d'entrenament, nutrició, perfil, social i gamificació.

Des del punt de vista d'enginyeria del software, el projecte ha permès aplicar decisions rellevants: modularització per

funcionalitats, separació entre frontend i backend, ús de migracions de base de dades, validació de dades, control d'accés per usuari, maneig centralitzat d'errors i validació de rendiment. A més, els problemes trobats han reforçat la qualitat del sistema, especialment en la gestió del progrés de gamificació mitjançant esdeveniments.

Els objectius no assolits totalment són sobretot extensions opcionals. La lectura de codis de barres no s'ha implementat i les recomanacions adaptatives han quedat limitades respecte a la idea inicial. Tot i que, mencionar que s'han afegit altres funcionalitats com el mode offline o la millora en la nutrició del botó de l'oli.

Com a treball futur es proposa: integrar una base de dades externa d'aliments i lectura de codis de barres; desenvolupar recomanacions basades en històric d'entrenaments i nutrició; afegir notificacions i recordatoris; ampliar proves automatitzades i monitoratge; i preparar la publicació en botigues mòbils. Amb aquestes millores, LevelUpXGL podria evolucionar des d'un prototip acadèmic sòlid cap a un producte més complet.

7.1 Viabilitat tècnica de continuació

La viabilitat tècnica de continuar LevelUpXGL és alta perquè la base actual ja separa responsabilitats. El frontend pot evolucionar afegint pantalles o millorant proveïdors sense canviar el backend complet. El backend pot incorporar nous endpoints o serveis mantenint l'estructura per mòduls. La base de dades, en estar versionada amb migracions, permet afegir taules noves sense perdre traçabilitat dels canvis.

Una possible evolució seria convertir les regles de gamificació en configuració persistent. Actualment les regles principals estan definides a la lògica del servei. En una versió més avançada, es podrien guardar en taules de configuració per permetre ajustar XP, recompenses o categories sense recompilar el backend. Això faria el sistema més parametrizable i més proper al disseny inicial de motor desacoblat.

També seria viable incorporar una capa de caché o càrrega incremental. Les pantalles actuals funcionen correctament, però algunes recarreguen dades després de cada acció. Una millora seria diferenciar dades que canvien poc, com aliments freqüents o informació de perfil, de dades que canvien cada dia, com resum nutricional o esdeveniments. Aquesta separació reduiria latència percebuda i consum de xarxa.

7.2 Consideracions ètiques i de privacitat

Les aplicacions de salut i hàbits personals requereixen prudència. LevelUpXGL registra informació de pes, objectius i alimentació, dades que poden ser sensibles per a alguns usuaris. Per això, qualsevol evolució del projecte hauria d'evitar missatges que fomentin conductes poc saludables o comparacions excessives amb altres persones. El component social s'ha mantingut limitat precisament per no exposar més informació de la necessària.

La gamificació també s'ha de tractar amb equilibri. Donar recompenses pot augmentar motivació, però també pot generar pressió si el sistema penalitza massa els errors. En el projecte, la raxa i la XP busquen reforç positiu, no càstig.

En una versió futura seria recomanable afegir missatges de recuperació, objectius flexibles i explicacions clares perquè l'usuari entengui que el progrés saludable no depèn d'un únic dia perfecte.

Finalment, una publicació real exigiria consentiment informat, política de privacitat, dret d'eliminació de dades, exportació de la informació personal i revisió de seguretat. El TFG no pretén resoldre completament aquests aspectes legals, però sí ha construït una base que ja separa dades per usuari i evita exposicions evidents.

7.3 Valoració final

La valoració final és positiva perquè el projecte ha arribat a una versió demostrable i coherent. El procés ha inclòs decisions tècniques, problemes reals, refactoritzacions descartades, millores visuals i validació amb mètriques. Això reflecteix un procés complet de desenvolupament de software, no només la creació d'un prototip de pantalla.

El resultat més important és haver integrat mòduls que normalment apareixen separats. L'usuari pot entrenar, registrar alimentació, veure el seu perfil, consultar progrés i rebre recompenses dins d'un mateix sistema. Aquesta integració, juntament amb el desplegament cloud i la persistència, dona al TFG una base sòlida per ser defensat com a projecte d'enginyeria.

7.4 Evidències visuals del resultat

La qualitat final també s'ha verificat visualment. Les pantalles finals mostren una línia gràfica coherent entre nutrició, progrés, calendari, perfil i social. Aquesta coherència ajuda a entendre que tots els mòduls formen part d'una mateixa aplicació i no de prototips independents.

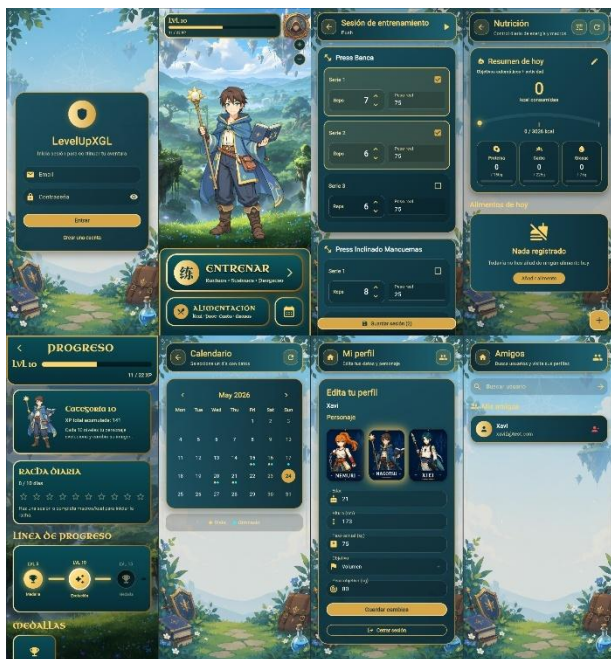


Figura 4. Mostra compacta de les pantalles principals implementades.

A nivell tècnic, les captures de validació mostren el desplegament, la persistència i les mètriques de rendiment. Aquestes evidències complementen la descripció textual dels resultats i donen suport a l'assoliment dels requisits no funcionals.

```
GET https://levelupxgl-backend-523689355803.europe-southwest1.run.app/api/routines
Media: 38.4 ms
Max: 45 ms
Errores: 0/5
Estado: OK

GET https://levelupxgl-backend-523689355803.europe-southwest1.run.app/api/v1/profile
Media: 47 ms
Max: 71 ms

GET https://levelupxgl-backend-523689355803.europe-southwest1.run.app/api/v1/gamification/progress
Media: 54.6 ms
Max: 59 ms
Errores: 0/5
Estado: OK

BUILD SUCCESSFUL in 27s
7 actionable tasks: 7 executed
```

Figura 5. Evidència de mesures de rendiment i proves backend.

8 ÚS DE LA IA GENERATIVA

Durant el desenvolupament s'ha utilitzat IA generativa com a eina de suport puntual, no com a substitut del treball propi. Els usos principals han estat resoldre dubtes tècnics, revisar fragments de codi, proposar estructures alternatives, ajudar a detectar errors, generar recursos gràfics i donar suport a la redacció de documentació.

Les decisions de disseny, la definició de l'arquitectura, la integració dels mòduls i la validació del resultat han estat realitzades per l'autor. El contingut generat o suggerit per IA s'ha revisat, adaptat i provat abans d'incorporar-lo. Per tant, la IA s'ha tractat com una eina comparable a consultar documentació tècnica, fòrums o exemples, mantenint la responsabilitat final com a autor del TFG.

AGRAÏMENTS

Agraeixo al tutor, Ronny Romero Navarrete, el seguiment i les indicacions durant el desenvolupament del projecte. També agraeixo el suport de les persones que han provat l'aplicació i han aportat comentaris sobre usabilitat, navegació i claredat de les pantalles.

BIBLIOGRAFIA

- [1] Flutter Documentation: <https://docs.flutter.dev/>
- [2] Riverpod Documentation: <https://riverpod.dev/>
- [3] Flutter packages: https://pub.dev/packages/go_router
- [4] Spring Documentation: <https://docs.spring.io/spring-boot/>
- [5] PostgreSQL GDG: <https://www.postgresql.org/docs/>
- [6] Google Cloud Doc.: <https://cloud.google.com/run/docs>
- [7] Google Cloud SQL: <https://cloud.google.com/sql/docs/postgres>
- [8] MyFitnessPal: <https://www.myfitnesspal.com/>
- [9] Strava: <https://www.strava.com/>
- [10] Fitbod: <https://fitbod.me/>
- [11] Habitica: <https://habitica.com/>
- [12] Flyway Doc.: <https://documentation.red-gate.com/fd>

APÈNDIX

A1. CAPTURES FUNCIONALS DE L'APLICACIÓ

Les captures següents mostren el resultat visual de les pantalles principals. S'inclouen pantalles de nutrició, progrés, calendari, perfil, amics i perfil conegut, per evidenciar que els mòduls principals han estat integrats en una experiència visual coherent.

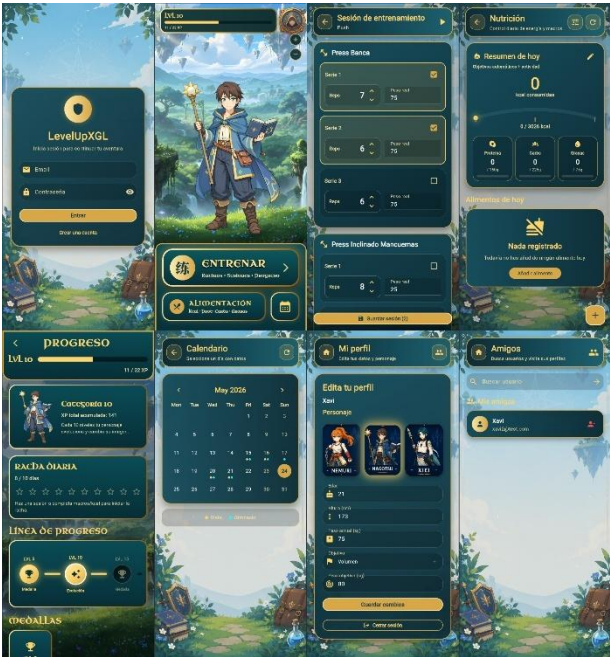


Figura A1. Conjunt de pantalles funcionals de LevelUpXGL.

A2. EVIDÈNCIES DE DESPLEGAMENT I VALIDACIÓ

Aquesta secció recull captures de suport utilitzades per validar el desplegament, la persistència i el rendiment. Inclou consultes sobre la base de dades, mètriques de Cloud Run, mesures de latència i execució de proves backend.

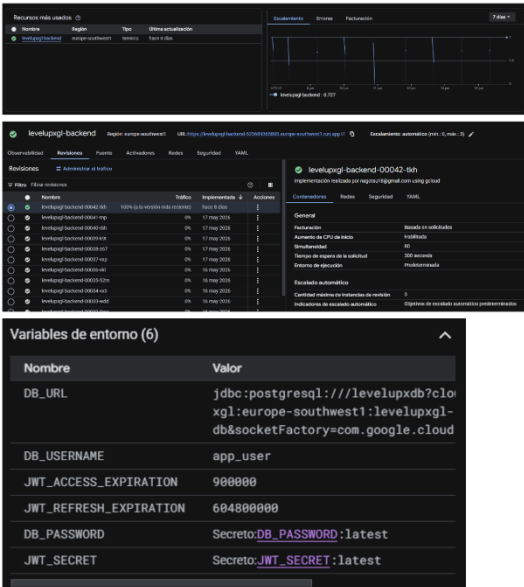


Figura A2. Evidències de desplegament a Google Cloud i configuració del servei.

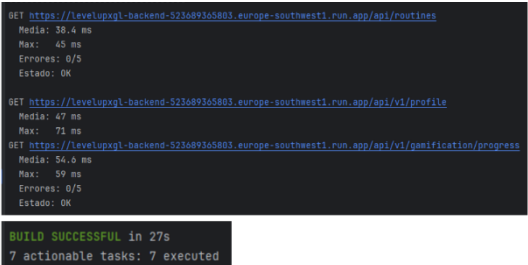


Figura A3. Mesures de rendiment dels endpoints i execució de proves backend.

A3. ARQUITECTURA DEL PROYECTO:

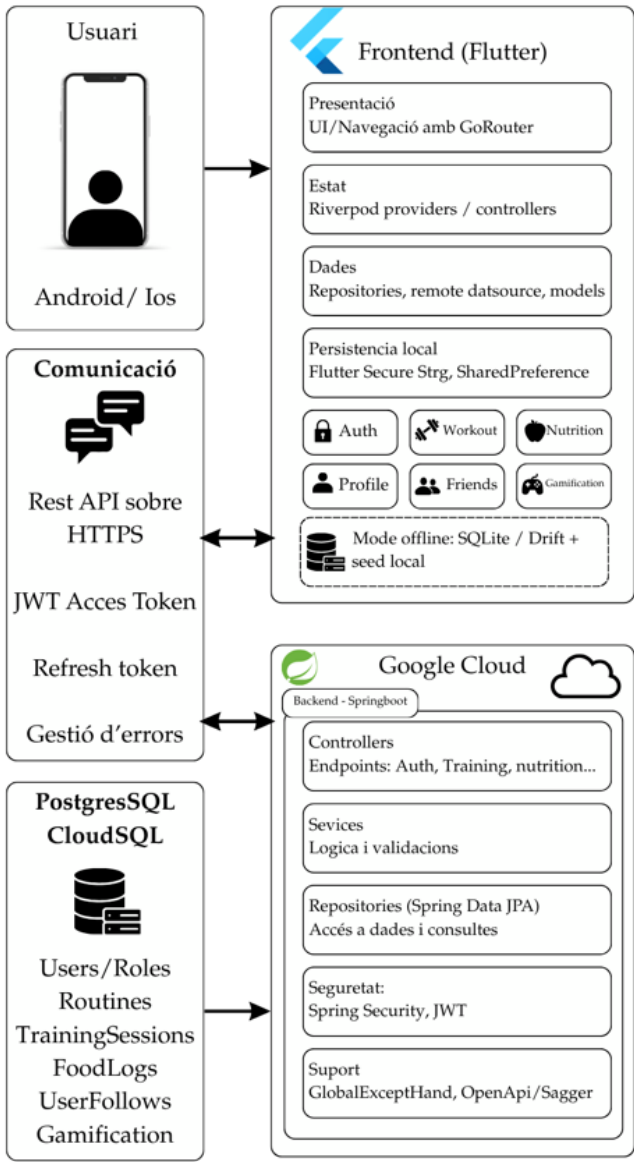


Figura A4. Arquitectura del sistema