



HEROIS

Una aplicació Android desenvolupada amb Jetpack Compose

Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils

Miquel Moragues Mas

Agenda

1

Introducció

2

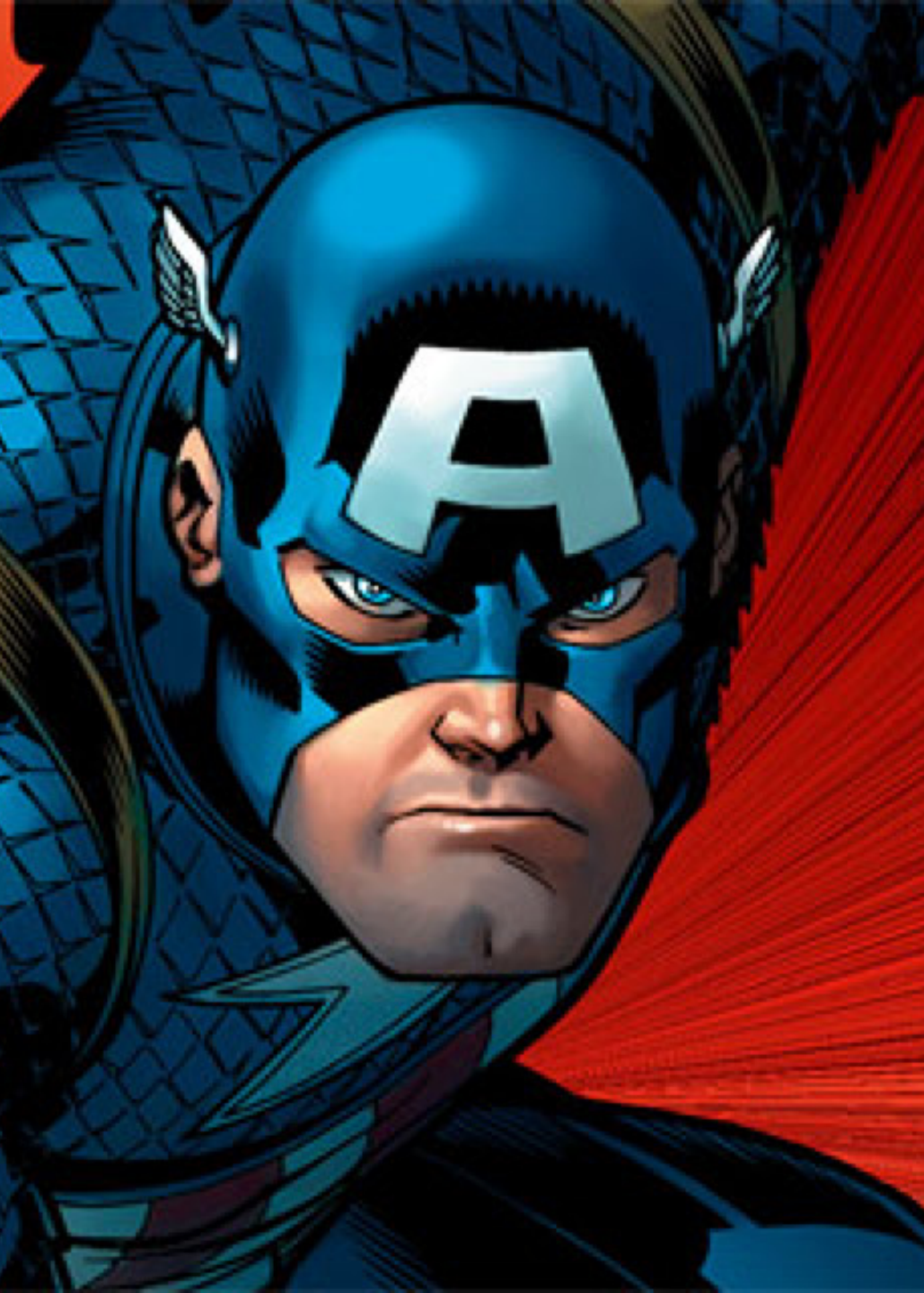
Disseny

3

Implementació

4

Demostració



Introducció

1

Introducció

Amb el desenvolupament d'una aplicació de consulta i entreteniment basat en dades de superherois, s'han fet servir les darreres eines de desenvolupament Android i s'han implementat les millors pràctiques i arquitectures recomanades per Google

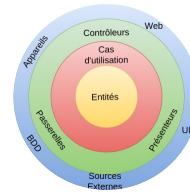
- Al Juliol de 2021 Google va llançar la primera versió estable de Jetpack Compose
- Jetpack Compose és el nou kit d'eines creat Google per compilar UI natives amb Kotlin
- Basat en el paradigma de la programació declarativa (com Flutter o SwiftUI)
- Kotlin i Jetpack Compose estan cridats a convertir-se en les eines bàsiques per a la creació d'aplicacions Android natives



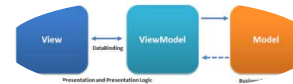
Jetpack Compose



Kotlin (Flow, Coroutines)



Clean Architecture



Patró MVVM



Injecció de Dependències

Disseny

2

Disseny Centrat a l'Usuari (DCU)

Mantenir a l'usuari al centre del nostre disseny ens ha permès identificar els seus objectius, motivacions i necessitats

Dos perfils d'usuaris:

- Pares de nens de primària amb interès pels còmics i/o col·leccionar cromos de superherois
- Homes joves amants dels còmics



Cerca de Superherois per nom



Poder tenir una llista de favorits



L'aplicació ha de mostrar nous superherois a l'usuari

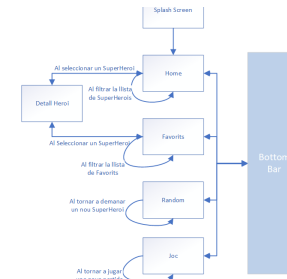


L'aplicació ha de tenir opció de jugar

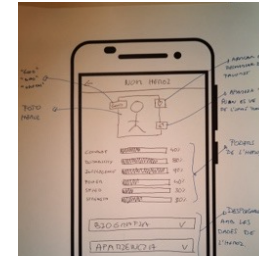
Arbre de navegació i Prototipatge

Amb la informació obtinguda de l'anàlisi dels nostres usuaris, es pot decidir com seran les pantalles de l'aplicació i com interactuaran entre elles

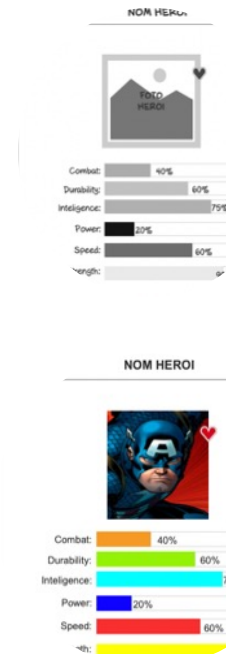
- Arbre de navegació: Com es navega dins la nostra aplicació
- Esbossos a mà alçada: Primera aproximació de com seran les pantalles
- Prototips de baixa fidelitat: Ens permet comprovar com es veuria les diferents pantalles en un dispositiu real
- Prototips d'alta fidelitat: Permet tenir interacció entre les pantalles i avaluar el nostre disseny



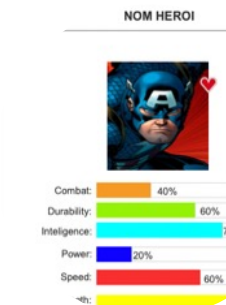
Arbre de navegació



Esbossos a mà alçada



Prototips de baixa fidelitat



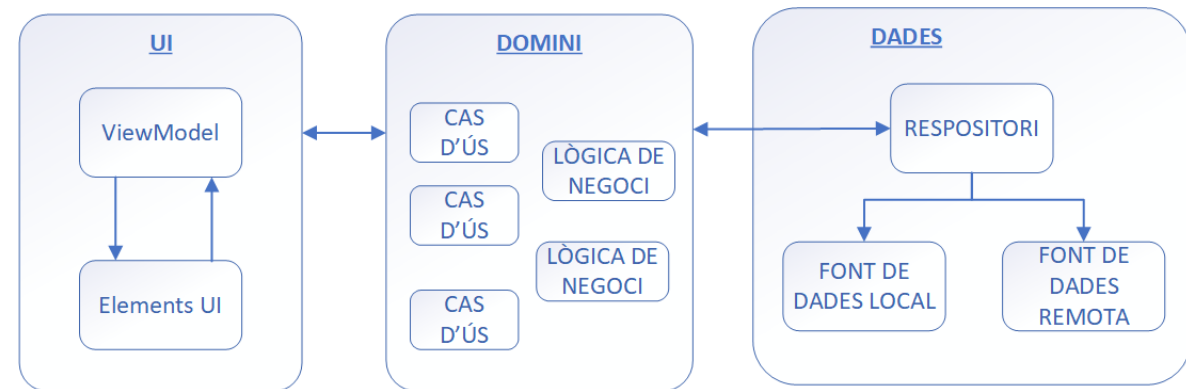
Prototips d'alta fidelitat

Arquitectura de l'Aplicació

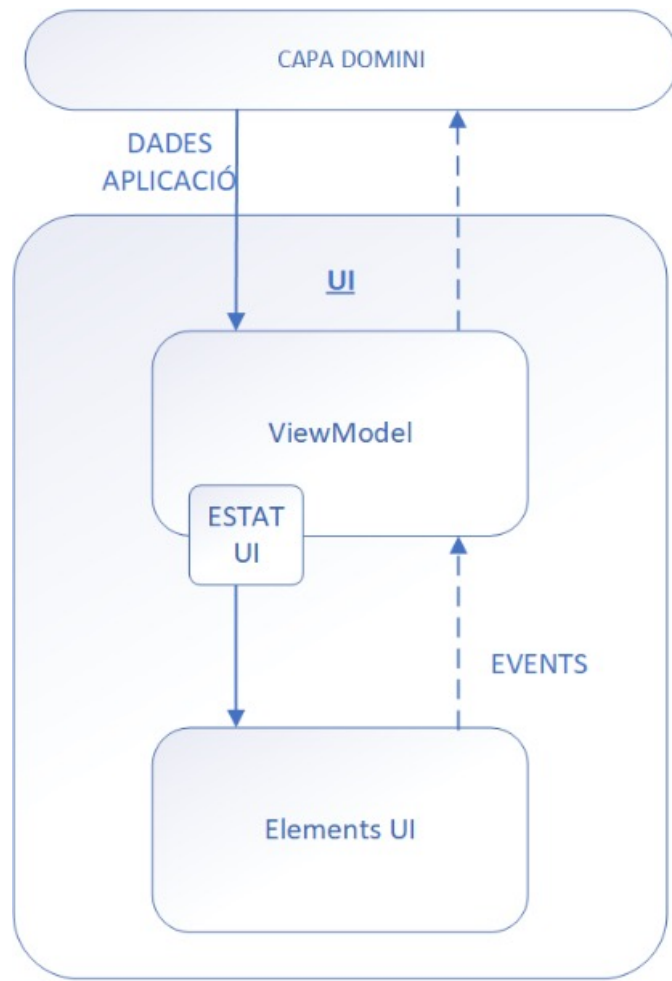
Les architectures permeten desacoblar diferents unitats del codi

- ✓ Codi més fàcil d'entendre
- ✓ Facilita el seu manteniment i modificacions
- ✓ Facilita les proves del codi

- Arquitectura basada en la Clean Architecture de Uncle Bob, adaptada a les necessitats específiques de les aplicacions Android
 - Capa UI
 - Capa Domini
 - Capa Dades



Capa UI



- ❑ Punt principal d'interacció amb l'usuari
- ❑ Es fa servir el Patró MVVM (Model-View-ViewModel)
- ❑ El ViewModel és el contenidor d'estat de la UI
- ❑ S'implementa el Patró de Flux Unidireccional de dades
- ❑ Implementació de la pantalles amb Jetpack Compose

Jetpack Compose

Jetpack Compose ens permet construir les pantalles de forma declarativa amb Kotlin i no de forma imperativa com passava amb els XML clàssics

Paradigma Imperatiu:

- Es centre en el “com” aconseguir la solució desitjada
- Cal descriure de forma explícita tots els passos i en quin ordre
- Cal descriure manualment com reaccionen els elements als canvis i actualitzar el seus estats en conseqüència
- Aquesta manipulació manual augmenta la probabilitat d'errors

Paradigma Declaratiu:

- Es descriu el resultat final que es busca, el “què”
- S'ha d'indicar què es vol que la UI mostri al usuari enlloc de com s'han de construir els elements
- El “com” es deixa al framework



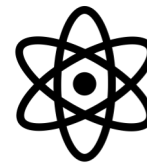
Codi de les UI més net i clar de llegir



Millor rendiment que les UI Imperatives



Permet fer més amb menys codi comparat amb XML

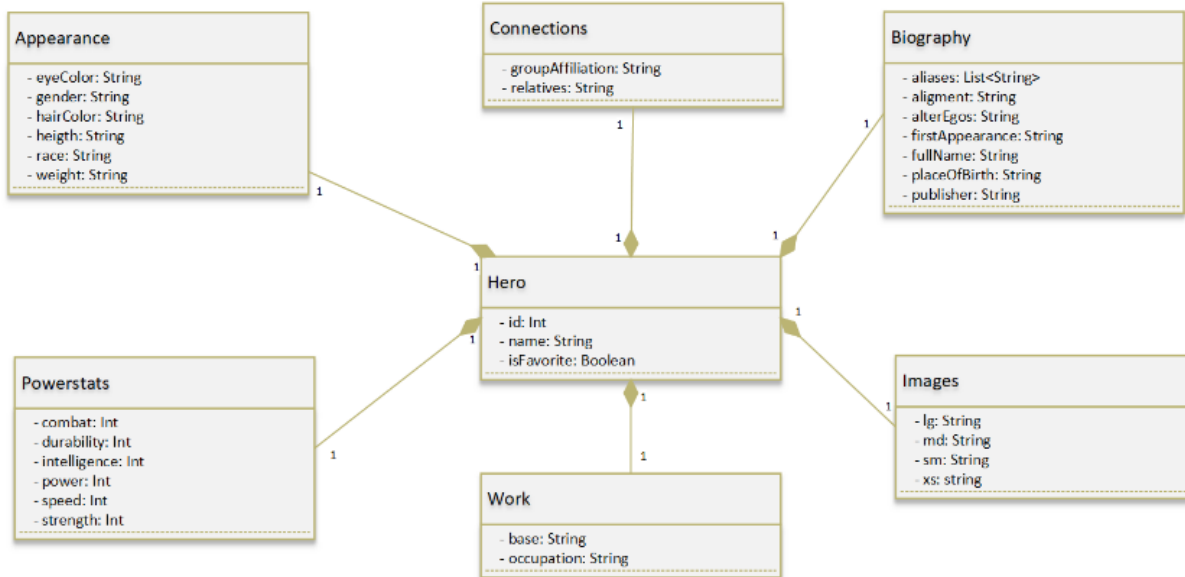


Disseny més atòmic i facilita la reutilització dels diferents components



Encaixa amb el patró de Flux Unidireccional de Dades ja que els components admeten estat i emeten events

Capa de Domini



❑ Encapsula la lògica empresarial que fan servir un o més ViewModels

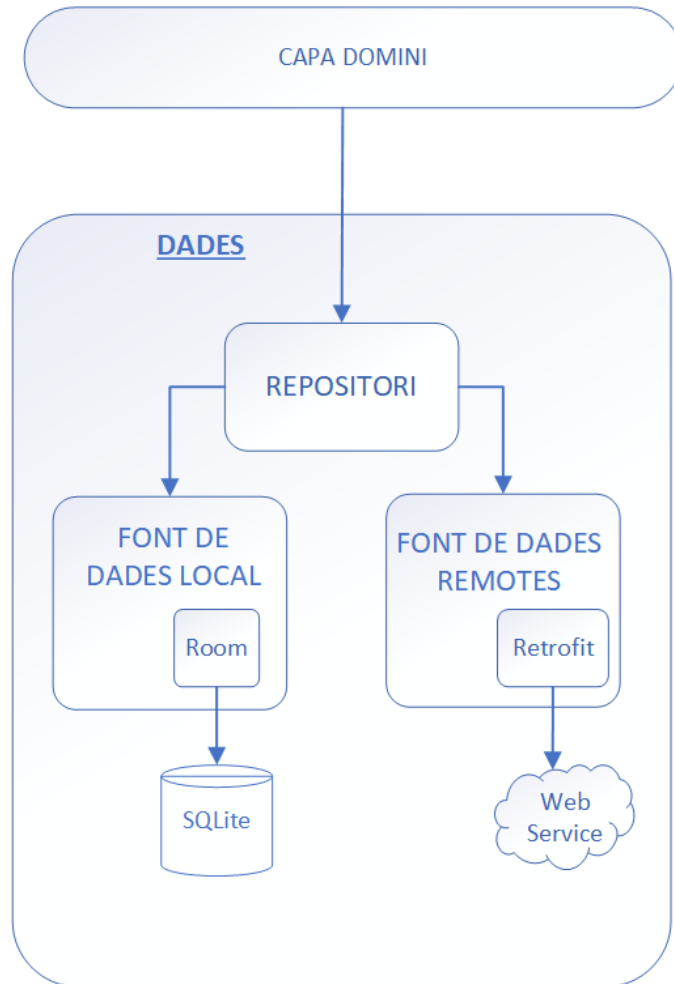
❑ Conté:

- Casos d'ús de l'aplicació
- Model de dades del domini
- Interfície dels repositoris que fan servir els casos d'ús

❑ A l'arquitectura recomanada per Google és una capa optativa, però s'ha implantat perquè:

- Evita duplicitat de codi
- Millora la llegibilitat del codi
- Facilita les proves
- Evita classes massa gran dividint les responsabilitats

Capa de Dades

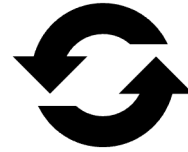


- ❑ Conté la implementació de la interfície del repositori definida a la Capa del Domini
- ❑ El repositori té dues fonts de dades:
 - Font de dades remota (obté les dades d'una API)
 - Font de dades Local (SSOT)
- ❑ El repositori és el responsable de:
 - Exposar les dades a la resta de l'aplicació
 - Centralitzar els canvis de les dades
 - Resoldre conflictes entre les dues fonts de dades (local i remota)
 - Abstreure les fonts de dades de la resta de l'aplicació
 - Contenir la lògica de crear, emmagatzemar i canviar les dades de l'aplicació

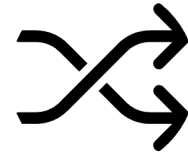
Injecció de Dependències (DI)

La DI és una tècnica molt utilitzada en programació i adequada pel desenvolupament d'Android

- Les classes requereixen de referències a altres classes pel que es creen dependències entre elles
- Per poder desvincular les classes entre elles, les dependències s'han de passar a través del constructor
- Per automatitzar el procés de creació i provisió de dependències fem servir Hilt



Facilita la reutilització de les classes



Permet el desacoblament de dependències



Facilita la refactorització

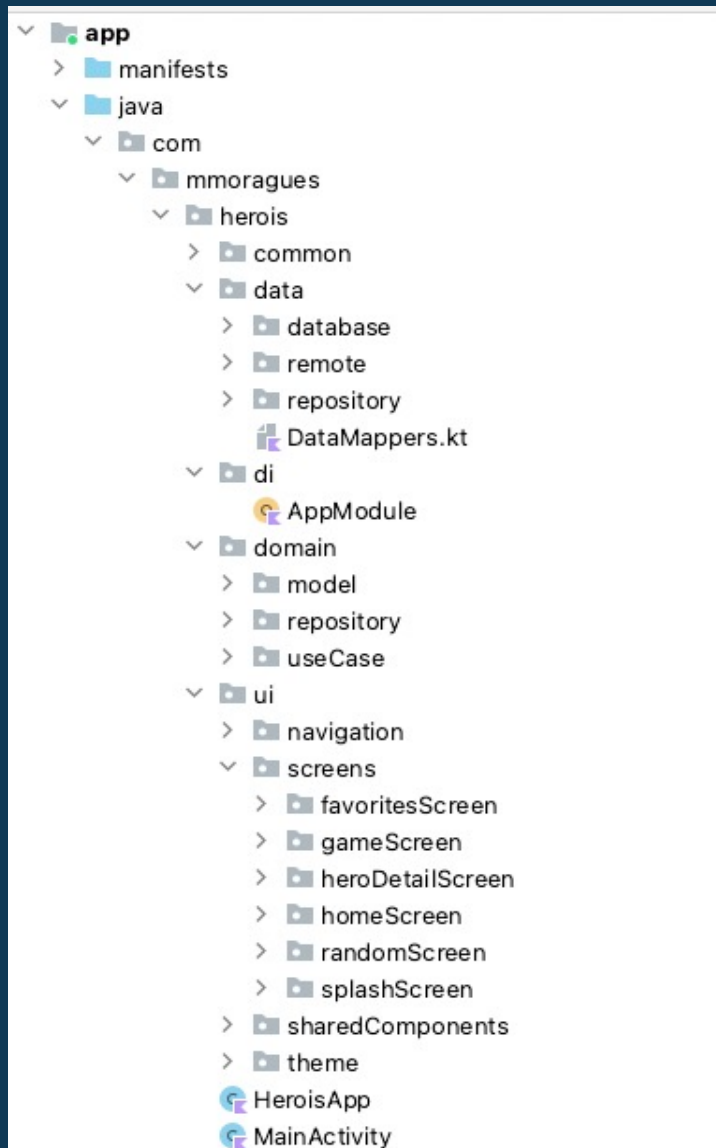


Facilita les proves

Implementació

3

ESTRUCTURA DE L'APLICACIÓ



Common:

- Recursos comuns, constants i classes auxiliars

Data:

- Interfícies i implementació de la "LocalDataSource" i de la "RemoteDatasouce"
- DTO de la BBDD local i del servidor remot i funcions per mapejar-los a la classe del domini
- Implementació del Repositori

Di:

- Mòdul Hilt per la injecció de dependències

Domain:

- Classes del model de l'aplicació
- Interfície del repositori
- Casos d'ús

UI:

- Controlador de la navegació entre diferents pantalles
- Pantalles de l'aplicació i, seguint el patró MVVM els seus View Model
- Components comuns a les diferents pantalles
- Tema i colors de l'aplicació

Llibreries Externes

Es fan servir 5 llibreries externes per gestionar diferents funcionalitats de l'aplicació



Hilt: Injecció de Dependències



Retrofit: Client HTTP accés a servidor remot



ROOM: Base de dades local



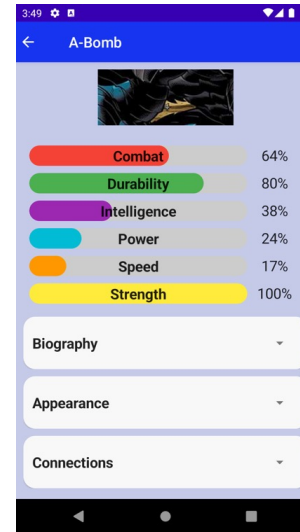
Coil: Gestió d'imatge



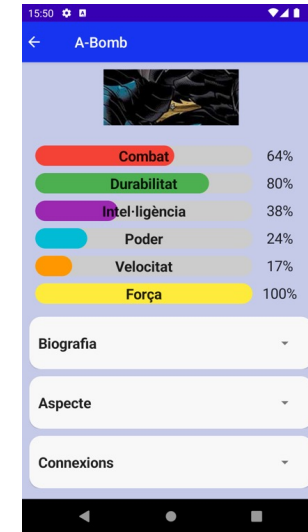
Lottie: Animacions

Internalització

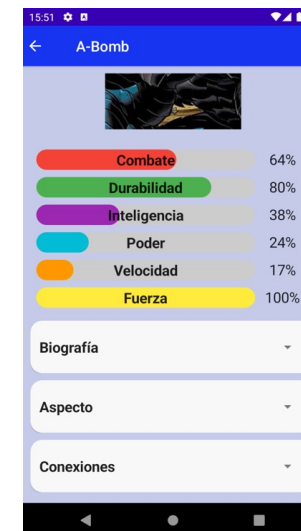
- Aplicació preparada per visualitzar-se en:
- Anglès (idioma per defecte)
- Castellà
- Català



Versió Anglès



Versió Català



Versió Castellà



- Biography
- Appearance
- Connections

Modes clar i obscur

Dues opcions de visualització:

- L'usuari pot visualitzar l'aplicació tant en mode clar com en mode obscur



- Biography
- Appearance
- Connections

Proves

PROVES UNITARIES

Tests	Duration	Pixel_API_31
✓ Test Results	195 ms	7/7
✓ LocalDataSourceImplTest	195 ms	7/7
✓ toggleIsFavorite	1 ms	✓
✓ keepFavoriteFieldWhenUpdate	95 ms	✓
✓ insertAndGetAllHeroes	0 ms	✓
✓ getRandomHero	0 ms	✓
✓ updateWithNewHeroes	0 ms	✓
✓ getHeroById	99 ms	✓
✓ getFavoritesHeroes	0 ms	✓

✓ Test Results	68 ms
✓ com.mmoragues.herois.data.repository.HeroRepositoryImplTest	68 ms
✓ Get a Hero by its ID	56 ms
✓ Get a Random Hero	5 ms
✓ Get Favorites Heroes	2 ms
✓ Toggle is Favorite	1 ms
✓ Get All Heroes	1 ms
✓ Get All Heroes with Server Error	3 ms

✓ Test Results	79 ms
✓ com.mmoragues.herois.domain.useCase.ToggleFavoriteUseCaseTest	58 ms
✓ Toggle isFavorite from False to True	56 ms
✓ Toggle isFavorite from True to False	2 ms
✓ com.mmoragues.herois.domain.useCase.GetAllHeroesUseCaseTest	6 ms
✓ Get All Heroes Loading and Success	5 ms
✓ Get All Heroes Loading and Error	1 ms
✓ com.mmoragues.herois.domain.useCase.GetFavoritesHeroesUseCaseTest	4 ms
✓ Get Favorites Heroes	3 ms
✓ Get Favorites Heroes without Internet	1 ms
✓ com.mmoragues.herois.domain.useCase.GetRandomHeroUseCaseTest	6 ms
✓ Get Random Hero without Internet	5 ms
✓ Get Random Hero	1 ms
✓ com.mmoragues.herois.domain.useCase.GetHeroByIdUseCaseTest	5 ms
✓ Get Hero by ID without Internet	4 ms
✓ Get Hero by ID	1 ms

PROVES DE LA BBDD
LOCAL

PROVES DEL
REPOSITORI

PROVES DELS CASOS
D'ÚS

Proves

PROVES D'INTEGRACIÓ, EXTREM A EXTREM I EN DISPOSITIU REAL

✓ FavoritesScreenTest	1s	1/1
✓ favoriteScreen_UncheckAndFilter	1s	✓
✓ GameScreenTest	901ms	1/1
✓ game_Works	901ms	✓
✓ HomeScreenTest	1s	2/2
✓ searchText_OpenAndClose	399ms	✓
✓ homeScreen_Filter	802ms	✓
✓ RandomScreenTest	2s	2/2
✓ expandableSections_ShowInfo	1s	✓
✓ randomHeroIcon_ShowNewRandomHero	1s	✓

✓ EndToEndTests	6s	3/3
✓ markAndUnmarkAHeroAsFavoriteFromDetail	2s	✓
✓ searchAHeroAndShowDetails	2s	✓
✓ markAndUnmarkAHeroAsFavoriteFromHome	1s	✓



PROVES D'INTEGRACIÓ DE LES PANTALLES AMB ELS SEU VIEW MODELS

PROVES EXTREM A EXTREM

PROVES EN DISPOSTIU REAL

Demostració

4



GRACIES

Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils
Miquel Moragues Mas