Herois: Una aplicació Android desenvolupada amb Jetpack Compose

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

Nom Consultor: Eduard Martin Lineros

Maig 2022

FITXA DEL TREBALL FINAL

Títol del treball:	*Herois: Una aplicació Android desenvolupada amb Jetpack Compose, MVVM i Clean Architecture*
Nom de l'autor:	*Miquel Moragues Mas*
Nom del consultor:	*Eduard Martin Lineros*
Data de lliurament (mm/aaaa):	*05/2022*
Titulació:	*Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils*

Resum del Treball (màxim 250 paraules):

Aquest treball descriu l'anàlisi, disseny i implementació de l'aplicació *Herois*.

Herois és una aplicació per a dispositius Android amb una versió 5.0 o superior.

Aquesta aplicació s'ha implementat seguint les bones pràctiques i guies d'arquitectura recomanades per Google com son el patró de disseny MVVM, l'arquitectura Clean Architecture i la Injecció de Dependències.

Com a llenguatge de programació s'ha fet servir Kotlin i per a la creació de les interfases d'usuaris (UI) s'ha emprat Jetpack Compose que és el nou kit d'eines que ha creat Google per compilar interfases d'usuari natives fent ús del llenguatge Kotlin.

Tant Kotlin com Jetpack Compose estan cridats a convertir-se en les eines bàsiques per a la creació d'aplicacions Android natives, ja que Google està apostant clarament per ells relegant a Java i a les interfícies basades en fitxers XML a un segon pla.

Pel que fa a l'aplicació desenvolupada, *Herois* ofereix als usuaris una doble funció:

Per un costat, és una aplicació de consulta on es té accés a informació de qualsevol superheroi (poders, biografia del personatge, editorial que el va crear, etc.) i que permet crear-nos una llista de herois favorits.

Per l'altre, també permet jugar contra el mòbil a un joc de cartes fent servir la puntuació dels poders dels superherois (força, velocitat, etc.) com a valors de les cartes.

Herois s'ha dissenyat per poder treballar sense necessitat de connexió a Internet, amb opció de mode clar i fosc i amb traducció a l'anglès, català i castellà.

Abstract (in English, 250 words or less):

This paper describes the analysis, design, and implementation of *Herois.*

Herois is an app for Android devices with version 5.0 or higher.

This application has been implemented following the best practices and architectural guidelines recommended by Google, such as the MVVM design pattern, Clean Architecture, and Dependency Injection.

Kotlin has been used as the programming language and Jetpack Compose has been used to create the user interfaces (UI) which is the new toolkit created by Google to compile native user interfaces using Kotlin.

Both, Kotlin and Jetpack Compose, are set to become the basic tools for creating native Android applications, as Google is clearly betting on them by relegating Java and XML-based interfaces to the background.

In terms of the application developed, *Herois* offers users a dual function:

On the one hand, it is a query application where you have access to information about any superhero (powers, biography of the character, publisher who created it, etc.) and that allows us to create a list of favorite heroes.

On the other hand, it also allows you to play a card game against your mobile using the score of superhero powers (strength, speed, etc.) as card values.

Herois is designed to be able to work without being connected to the Internet, both light and dark modes and translations into English, Catalan and Spanish.

Paraules clau (entre 4 i 8):

Jetpack Compose, MVVM, Clean Architecture, Hilt, Corroutine, Flow, Kotlin

Índex

Llista de Taules

Llista d'Il·lustracions

1 Introducció

1.1 Context i justificació del Treball

Al Juliol de 2021, uns pocs mesos abans de la realització d'aquest TFM, Google va llançar la primera versió estable de Jetpack Compose [1].

Jetpack Compose és el nou kit d'eines que ha creat Google per compilar interfases d'usuari (UI) natives fent ús del llenguatge Kotlin.

El gran canvi que introdueix Jetpack Compose es que està basat en el paradigma de la programació declarativa (com ja passa en altres llenguatges com ara Flutter o SwiftUI) i no amb el paradigma de la programació imperativa com passava fins ara amb els XML clàssics d'Android.

La gran diferencia entre aquests dos paradigmes és que, quan apliquem el paradigma imperatiu, som els responsables d'escriure cada instrucció que el sistema ha d'executar per aconseguir que la pantalla quedi a l'estat desitjat, mentre que al paradigma declaratiu el programador ja no té la responsabilitat de definir com s'han de fer les coses sinó de definir què s'ha de fer, establint una sèrie de condicions que s'han de complir i és el sistema el que ha de fer el necessari per a que aquestes condicions es compleixin sempre.

Dins aquest context, aquest projecte pretén desenvolupar una aplicació que permeti aprofundir en el coneixement de Jetpack Compose així com de la resta de bones pràctiques i guies d'arquitectura recomanades per Google i que durant la realització del Màster o bé no s'han tractat o bé s'han tractat superficialment, com son el patró de disseny MVVM, l'arquitectura Clean Architecture i la Injecció de Dependències.

L'aplicació que es desenvoluparà serà una aplicació per a mòbils Android que tindrà una doble funció, per una part permetrà a l'usuari tenir accés a la informació de qualsevol superheroi (poders, biografia del personatge, editorial que el va crear, etc.) i per l'altre també podrà jugar contra el mòbil a un joc de cartes fent servir la puntuació dels poders dels superherois (força, velocitat, etc.) com a valors de les cartes.

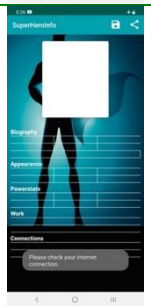
El desenvolupament d'aquesta aplicació ens permetrà tant aprofitar tots els avantatges que ens ofereix Jetpack Compose, aplicar les bones practiques que Google recomana pel desenvolupament d'aplicació així com aplicar i tots els coneixements adquirits durant el màster (connexió a APIs, bases de dades locals, etc.)

1.1.1 Benchmarking

Després de fer un a cerca exhaustiva només s'ha trobat una aplicació amb característiques similars a la que es desenvoluparà en aquest TFM i que es diu ***"Superhero Information"***. [2]

Una vegada analitzada aquesta aplicació s'han detectat una sèrie de mancances importants que la nostra aplicació pretén solucionar així com un punt fort que la nostra aplicació ha de replicar.

A la següent taula es mostren els punts febles detectats.

Punts febles *Superhero Information*		
Interfase d'usuari poc atractiva	La pàgina principal només mostra un llistat de noms sense cap tipus de disseny. La pantalla de detall és difícil de llegir	 
No permet buscar ni filtrar	La pàgina principal no permet fer cerques ni filtrar els superherois, pel que trobar el que busques es complicat	
No té llista de favorits	No permet tenir un llistat de superherois favorits. Només permet guardar o compartir el detall del superheroi com una imatge a la llibreria d'imatges del mòbil	
No funciona si no hi ha connexió a Internet	Si no hi ha connexió a Internet l'aplicació no funciona correctament	
No està "gamificat"	Només funciona com a base de dades on consultar informació, no ofereix cap altre al·licient a l'usuari	

Taula 1. Punts febles Superhero Information

Una vegada identificats els punts febles, a la següent taula es mostra el punt fort d'aquesta aplicació i que la nostra aplicació també haurà de tenir.

Punts forts *Superhero Information*		
Gran quantitat de superherois inventariats	L'aplicació té fins a 731 superherois al seu catàleg. La nostra aplicació també haurà de tenir com a mínim, el mateix nombre de superherois	

Taula 2. Punts forts Superhero Information

1.2 Objectius del Treball

L'objectiu principal d'aquest treball és la realització d'una aplicació per a mòbils Android que permeti:

- Obtenir informació sobre qualsevol superheroi, independentment de quina editorial l'hagi creada

- Jugar contra el mòbil fent servir la puntuació de superherois elegits a l'atzar. L'usuari haurà d'elegir un dels poders del seu superheroi i guanya si el seu valor és superior al del superheroi que té el mòbil

1.2.1 Requeriments funcionals

Els requeriments funcionals que considerem bàsics són:

- **Llistat de tots els superherois:** L'usuari ha de tenir accés a la informació de tots els superherois de qualsevol editorial.

- **Cerca de superherois per nom:** L'aplicació ha de permetre que l'usuari busqui un superheroi en concret pel seu nom.

- **Crear una llista de favorits:** S'han de poder marcar els superherois com a favorits i l'usuari ha de poder accedir a ells de forma ràpida.

- **Descobrir nous superherois:** L'aplicació ha de tenir l'opció de mostrar al usuari superherois de forma aleatòria

- **Mostrar detalls del superherois:** La informació a mostra de cada superheroi ha de contenir:

    - Imatge del superheroi
    - Poders puntuats de 0 a 100
    - Dades físiques
    - Dades biogràfiques

    - Ocupació
    - Relacions amb altres superherois

- **L'aplicació ha de poder funcionar sense accés a Internet:** L'aplicació ha de poder funcionar offline. Es pot assumir que en cas de no tenir accés a Internet no es carreguin les imatges dels superherois però s'ha de poder mostrar la resta de informació i s'ha de poder jugar sense problemes.

1.2.2 Requeriments no funcionals

Es defineixen el següents requeriments no funcionals:

- **Llenguatge de desenvolupament:** L'aplicació s'ha de desenvolupar amb el llenguatge Kotlin.

- **Entorn UI:** La interfície de l'usuari ha de ser intuïtiva, senzilla i atractiva per l'usuari. Aquest entorn s'ha de desenvolupar fent servir Jetpack Compose.

- **Patró de disseny:** Es farà servir el patró de disseny MVVM (Model-View-Viewmodel) que ens ajudarà a separar la capa de presentació de la lògica.

- **Arquitectura de l'aplicació:** Es dissenyarà l'aplicació fent servir Clean Architecture per tal de desacoplar el màxim possible les diferents unitats del codi, permetent que el codi sigui més fàcil d'entendre, modificar i testear.

- **Injecció de Dependències:** Es farà servir la llibreria Hilt com a mètode per implementar la Injecció de Dependències al nostre projecte.

1.3 Enfocament i mètode seguit

Com s'ha indicat en apartats anteriors, només s'ha trobat una aplicació en el mercat amb característiques similars a la que estem proposant en aquest TFM. Si bé no podem dir que la nostra aplicació és un producte nou, a la pràctica i al introduir-hi moltes millores, tant a nivell d'interfície d'usuari com de funcionalitats, l'usuari final el percebrà com un producte totalment diferent i millorat respecte a la competència.

Com que una de les motivacions principals per la realització d'aquest TFC és la d'explorar les capacitats que ens ofereix Jetpack Compose, es decideix que la plataforma de destí de l'aplicació sigui Android i per tal de que la pugin fer servir el major nombre possible d'usuaris, s'escollirà l'API 21, Android 5.0 (Lollipop), que el suporta un 98% dels dispositius actuals.

Per la realització d'aquest treball farem servir la metodologia de desenvolupament Waterfall. [3] [4]

Considerem que aquesta metodologia és la més adequada per aquest projecte, ja que no hi ha molta incertesa amb el que es desitja realitzar, els requeriments estan molt acotats i no canviaran durant el cicle de vida del desenvolupament i les fites de lliurament estan totalment marcades pels lliuraments de les diferents PACs i el lliurament final.

1.4 Planificació del Treball

1.4.1 Recursos per realitzar el projecte

Per a la realització d'aquest projecte disposem dels següents recursos.

Hardware:

- **MacBook Air (M1, 2020)** amb SO macOS Moterey (12.2.1) on es farà el desenvolupament i documentació del projecte
- **Samsung Galaxy A51** amb Android 11 – API 23 on es podrà provar l'aplicació en un terminal real

Software:

- **Android Studio Bumblebee:** IDE pel desenvolupament de l'aplicació i proves en terminal virtual
- **Axurre RP 10:** Per l'elaboració de prototips de baixa i alta fidelitat
- **Microsoft Project:** Per a la realització de la planificació i diagrames de Gantt
- **Microsoft Word:** Per a la realització de la memòria del TFM
- **Microsoft Power Point:** Per a la realització de la presentació final
- **Microsoft Visio:** Per crear diagrames UML i arbre de navegació

1.4.2 Tasques a realitzar

A continuació es detallen les tasques a realitzar durant el TFM a cada una de les fases

Fase 1 - Pla de Treball

1. Definició del TFM
   1.1. Benchmarking
   1.2. Contextos i Justificació del TFM
2. Definició d'objectius
   2.1. Requeriments funcionals

2.2. Requeriments no funcionals
3. Enfocament
4. Planificació del TFM
   4.1. Determinar Recursos
   4.2. Definir tasques
   4.3. Identificar Riscos
   4.4. Planificació
5. Revisió document
6. Lliurament PAC1

Fase 2 - Disseny

1. Disseny centrat a l'usuari (DCU)
   1.1. Definició d'Usuaris i context d'ús
   1.2. Disseny Conceptual
      1.2.1. Definir Usuaris focals
      1.2.2. Definir Escenaris
      1.2.3. Crear fitxes Persona/Escenari
      1.2.4. Crea Arbre de Navegació
2. Prototipat
   2.1. Esbossos a mà alçada
   2.2. Prototips de baixa fidelitat
   2.3. Prototips d'alta fidelitat
   2.4. Avaluació dels prototips
3. Disseny de l'arquitectura de l'aplicació
   3.1. Definició casos d'ús
   3.2. Disseny del patró de disseny
   3.3. Disseny Arquitectura de l'aplicació
   3.4. Disseny de la BBDD
4. Lliurament PAC 2

Fase 3 – Implementació

1. Desenvolupament
   1.1. Configuracions comunes
      1.1.1. Configuració Dependències
      1.1.2. Classes auxiliars
      1.1.3. Definició de constants
      1.1.4. Navegació entre screens
      1.1.5. UIEvents
      1.1.6. Injecció de Dependències
   1.2. Capa Domain
      1.2.1. Model de dades del Domini
      1.2.2. Interfície del Repositori
      1.2.3. Casos d'ús
   1.3. Capa Data
      1.3.1. Connexió remota (Server)
         1.3.1.1. Model de dades servidor remot
         1.3.1.2. Configuració Api Retrofit
      1.3.2. Connexió local (BBDD)

        1.3.2.1. Model de dades BBDD Local (Entity)
        1.3.2.2. Configuració Dao
        1.3.2.3. Configuració Converters
        1.3.2.4. Configuració Database Room
        1.3.2.5. Implementació del Repositori
    1.4. Capa UI
        1.4.1. Splash Screen
        1.4.2. HeroList Screen
            1.4.2.1. Components gràfics de la pantalla
            1.4.2.2. HeroListEvents
            1.4.2.3. ViewModel
        1.4.3. HeroDetail Screen
            1.4.3.1. Components gràfics de la pantalla
            1.4.3.2. HeroDetailEvents
            1.4.3.3. ViewModel
        1.4.4. FindHero Screen
            1.4.4.1. Components gràfics de la pantalla
            1.4.4.2. FindHeroEvents
            1.4.4.3. ViewModel
        1.4.5. Game Screen
            1.4.5.1. Components gràfics de la pantalla
            1.4.5.2. FindHeroEvents
            1.4.5.3. ViewModel
        1.4.6. FavoritesHeroes Screen
            1.4.6.1. Components gràfics de la pantalla
            1.4.6.2. FavoriteHeroesEvents
            1.4.6.3. ViewModel
2. Proves
    2.1. Proves unitàries
    2.2. Proves amb usuaris
    2.3. Millores al codi
3. Redacció apartat Implementació
4. Lliurament PAC 3

Fase 4 – Lliurament final

1. Redacció de la memòria
2. Vídeo de presentació
3. Lliurament PAC 4

1.4.3 Riscos detectats

A continuació es detallen els riscos identificats que poden afectar a la planificació del projecte així com la seva contingència.

Risc	Probabilitat	Impacte	Contingència
Endarreriment durant la fase d'implementació per la poca experiència amb Jetpack Compose	Mitja	Alt	Dedicar 1 hora més al dia de les inicialment estimades a la realització del projecte durant la fase inicial de la fase d'implementació per agafar més experiència amb aquesta tecnologia
Subestimar les tasques per afegir la pantalla de Joc	Alta	Alt	Dedicar 1 hora més al dia de les inicialment estimades per la realització d'aquesta pantalla. Si tot i així no es factible la correcta implementació d'aquesta pantalla, l'aplicació es lliuraria sense ella

Taula 3. Riscos del TFM

1.4.4 Planificació temporal

La planificació temporal del projecte ve marcada per les dates de lliurament de les diferents PACs.

La dedicació diària per a la realització del projecte son 3h diàries (de dilluns a diumenge). A la següent taula es mostren les dates d'inici i fi de les PACs i les hores de dedicació prevista per a cada una d'elles.

Lliurament	Data Inici	Data Fi	Hores
PAC1 – Pla de Treball	16/1/22	9/3/22	66 h.
PAC2 – Disseny	10/3/22	30/3/22	63 h.
PAC3 – Implementació	31/3/22	11/5/22	126 h.
PAC4 – Lliurament Final	12/5/22	30/5/22	38 h.
		Total	**312 h.**

Taula 4. Dates d'inici i fi de les PACs i hores dedicades

A continuació es mostra la planificació global de les fases del TFM.

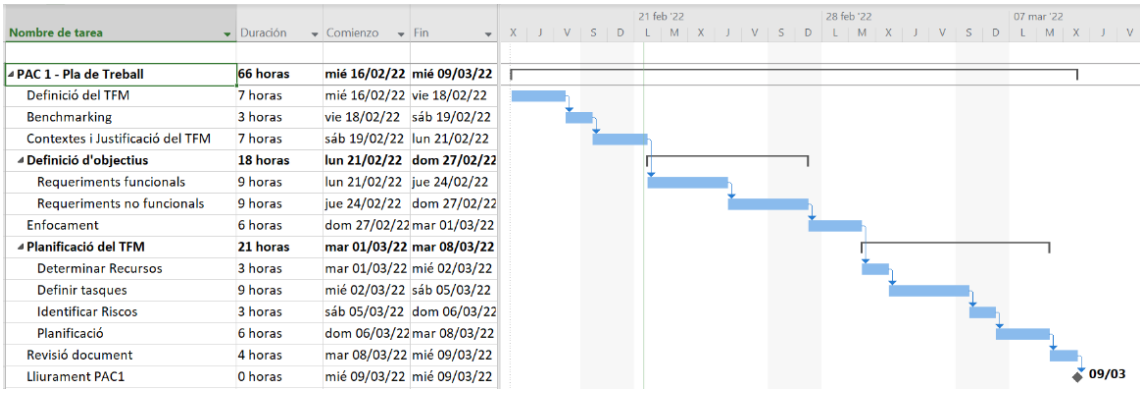
Nombre de tarea	Duración	Comienzo	Fin
PAC 1 - Pla de Treball	66 horas	mié 16/02/22	mié 09/03/22
PAC 2 - Disseny	63 horas	jue 10/03/22	mié 30/03/22
PAC 3 - Implementació	126 horas	jue 31/03/22	mié 11/05/22
PAC 4 - Lliurament Final	57 horas	jue 12/05/22	lun 30/05/22

Il·lustració 1. Planificació global

1.4.4.1 Fase Pla de Treball

Durant aquesta fase es definirà i planificarà el projecte. També es determinaran els requeriments de l'aplicació a desenvolupar comparant-los amb altres aplicacions ja en producció i que tinguin funcionalitats similars a les que volem fer nosaltres.

A continuació es mostra el detall de la planificació d'aquesta fase.



Nombre de tarea	Duración	Comienzo	Fin
PAC 1 - Pla de Treball	**66 horas**	**mié 16/02/22**	**mié 09/03/22**
Definició del TFM	7 horas	mié 16/02/22	vie 18/02/22
Benchmarking	3 horas	vie 18/02/22	sáb 19/02/22
Contextes i Justificació del TFM	7 horas	sáb 19/02/22	lun 21/02/22
Definició d'objectius	**18 horas**	**lun 21/02/22**	**dom 27/02/22**
Requeriments funcionals	9 horas	lun 21/02/22	jue 24/02/22
Requeriments no funcionals	9 horas	jue 24/02/22	dom 27/02/22
Enfocament	6 horas	dom 27/02/22	mar 01/03/22
Planificació del TFM	**21 horas**	**mar 01/03/22**	**mar 08/03/22**
Determinar Recursos	3 horas	mar 01/03/22	mié 02/03/22
Definir tasques	9 horas	mié 02/03/22	sáb 05/03/22
Identificar Riscos	3 horas	sáb 05/03/22	dom 06/03/22
Planificació	6 horas	dom 06/03/22	mar 08/03/22
Revisió document	4 horas	mar 08/03/22	mié 09/03/22
Lliurament PAC1	0 horas	mié 09/03/22	mié 09/03/22

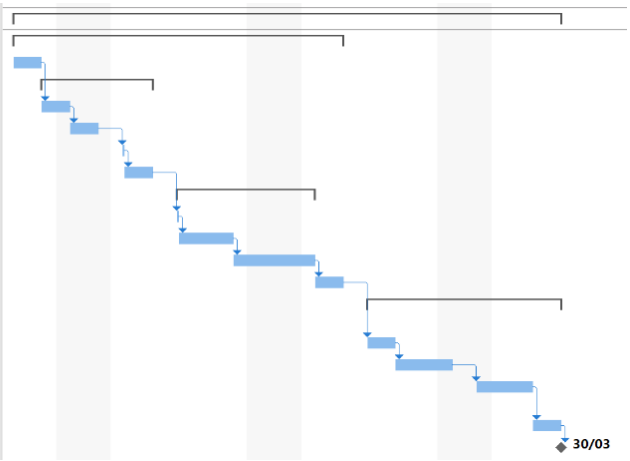
Il·lustració 2. Planificació Pla de Treball

1.4.4.2 Fase de Disseny

Durant la fase de disseny s'analitzaran als usuaris que faran servir la nostra aplicació i poder així definir les funcionalitats que seran necessàries que la nostra aplicació implementi. També es dissenyaran i avaluaran els prototips tant de baixa com d'alta fidelitat i es dissenyarà l'arquitectura i la BBDD.

A continuació es mostra el detall de la planificació d'aquesta fase.

PAC 2 - Disseny	**63 horas**	**jue 10/03/22**	**mié 30/03/22**
Disseny centrat a l'usuari (DCU)	**39 horas**	**jue 10/03/22**	**mar 22/03/22**
Definició d'Usuaris i context d'ús	4 horas	jue 10/03/22	vie 11/03/22
Disseny Conceptual	**14 horas**	**vie 11/03/22**	**mar 15/03/22**
Definir Usuaris focals	4 horas	vie 11/03/22	sáb 12/03/22
Definir Escenaris	4 horas	sáb 12/03/22	dom 13/03/22
Crear fitxes Persona/Escenari	2 horas	lun 14/03/22	lun 14/03/22
Crea Arbre de Navegació	4 horas	lun 14/03/22	mar 15/03/22
Prototipat	**17 horas**	**mié 16/03/22**	**lun 21/03/22**
Esbossos a mà alçada	2 horas	mié 16/03/22	mié 16/03/22
Prototips de baixa fidelitat	6 horas	mié 16/03/22	vie 18/03/22
Prototips d'alta fidelitat	9 horas	vie 18/03/22	lun 21/03/22
Avaluació dels prototips	4 horas	lun 21/03/22	mar 22/03/22
Disseny de l'arquitectura de l'aplicació	**24 horas**	**mié 23/03/22**	**mié 30/03/22**
Definició casos d'ús	4 horas	mié 23/03/22	jue 24/03/22
Disseny del patró de disseny	8 horas	jue 24/03/22	sáb 26/03/22
Disseny Arquitectura de l'aplicació	8 horas	dom 27/03/22	mar 29/03/22
Disseny de la BBDD	4 horas	mar 29/03/22	mié 30/03/22
Lliurament PAC 2	0 horas	mié 30/03/22	mié 30/03/22



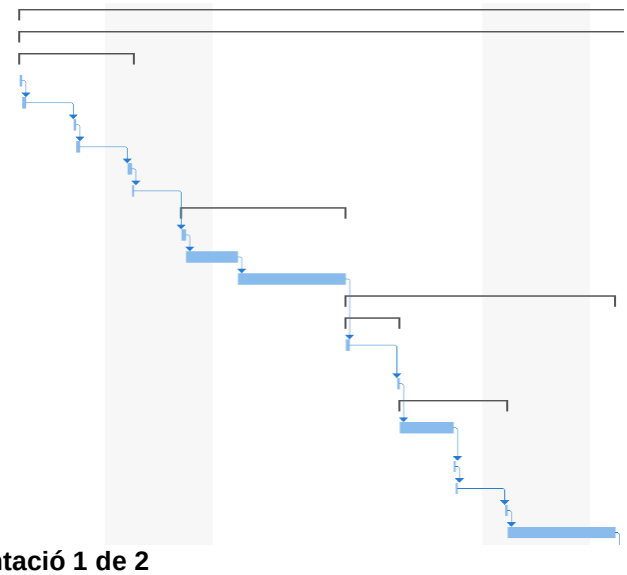
Il·lustració 3. Planificació fase de Disseny

1.4.4.3 Fase d'Implementació

Aquesta és la fase amb més càrrega de treball. En aquesta fase es desenvoluparà l'aplicació seguint el disseny definit a la fase anterior. També es faran les proves de l'aplicació així com les millores al codi en cas que alguna de les proves no sigui satisfactòria.

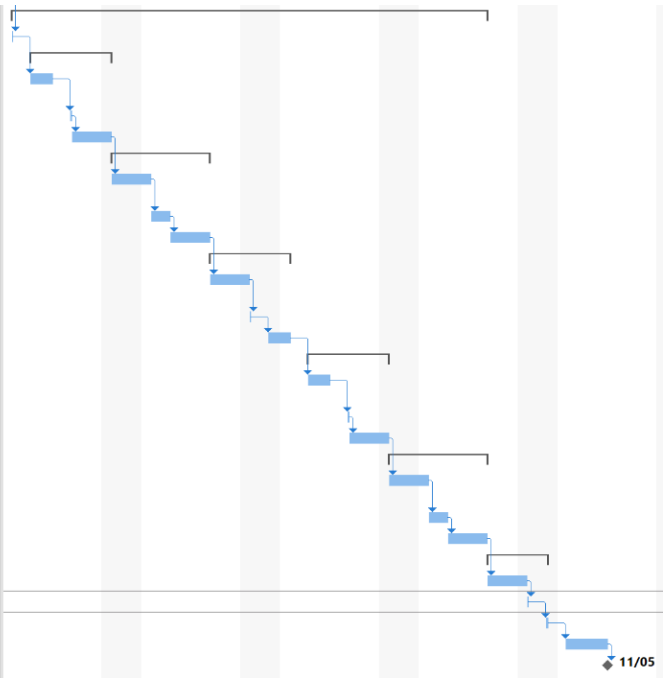
A continuació es mostra el detall de la planificació d'aquesta fase. Per la seva longitud es mostra separat en dues parts

PAC 3 - Implementació	**126 horas**	**jue 31/03/22**	**mié 11/05/22**
Desenvolupament	**106 horas**	**jue 31/03/22**	**jue 05/05/22**
Configuracions comunes	**9 horas**	**jue 31/03/22**	**sáb 02/04/22**
Configuració Dependencies	1 hora	jue 31/03/22	jue 31/03/22
Classes auxiliars	2 horas	jue 31/03/22	jue 31/03/22
Definició de constants	1 hora	vie 01/04/22	vie 01/04/22
Navegació entre screens	2 horas	vie 01/04/22	vie 01/04/22
UIEvents	2 horas	sáb 02/04/22	sáb 02/04/22
Injecció de Dependencies	1 hora	sáb 02/04/22	sáb 02/04/22
Capa Domain	**10 horas**	**dom 03/04/22**	**mié 06/04/22**
Model de dades del Domini	2 horas	dom 03/04/22	dom 03/04/22
Interficie del Repositori	2 horas	dom 03/04/22	lun 04/04/22
Casos d'ús	6 horas	lun 04/04/22	mié 06/04/22
Capa Data	**15 horas**	**mié 06/04/22**	**lun 11/04/22**
Connexió remota (Server)	**3 horas**	**mié 06/04/22**	**jue 07/04/22**
Model de dades servidor remot	2 horas	mié 06/04/22	mié 06/04/22
Configuració Api Retrofit	1 hora	jue 07/04/22	jue 07/04/22
Connexió local (BBDD)	**6 horas**	**jue 07/04/22**	**sáb 09/04/22**
Model de dades BBDD Local (Entity)	3 horas	jue 07/04/22	vie 08/04/22
Configuració Dao	1 hora	vie 08/04/22	vie 08/04/22
Configuració Converters	1 hora	vie 08/04/22	vie 08/04/22
Configuració Database Room	1 hora	sáb 09/04/22	sáb 09/04/22
Implementació del Repositori	6 horas	sáb 09/04/22	lun 11/04/22



Il·lustració 4. Planificació fase Implementació 1 de 2

Capa UI	**72 horas**	**lun 11/04/22**	**jue 05/05/22**
Splash Screen	2 horas	lun 11/04/22	lun 11/04/22
HeroList Screen	**14 horas**	**mar 12/04/22**	**sáb 16/04/22**
Components gràfics de la pantalla	6 horas	mar 12/04/22	mié 13/04/22
HeroListEvents	2 horas	jue 14/04/22	jue 14/04/22
ViewModel	6 horas	jue 14/04/22	sáb 16/04/22
HeroDetail Screen	**14 horas**	**sáb 16/04/22**	**jue 21/04/22**
Components gràfics de la pantalla	6 horas	sáb 16/04/22	lun 18/04/22
HeroDetailEvents	2 horas	lun 18/04/22	mar 19/04/22
ViewModel	6 horas	mar 19/04/22	jue 21/04/22
FindHero Screen	**14 horas**	**jue 21/04/22**	**lun 25/04/22**
Components gràfics de la pantalla	6 horas	jue 21/04/22	sáb 23/04/22
FindHeroEvents	2 horas	sáb 23/04/22	sáb 23/04/22
ViewModel	6 horas	dom 24/04/22	lun 25/04/22
Game Screen	**14 horas**	**mar 26/04/22**	**sáb 30/04/22**
Components gràfics de la pantalla	6 horas	mar 26/04/22	mié 27/04/22
FindHeroEvents	2 horas	jue 28/04/22	jue 28/04/22
ViewModel	6 horas	jue 28/04/22	sáb 30/04/22
FavoritesHeroes Screen	**14 horas**	**sáb 30/04/22**	**jue 05/05/22**
Components gràfics de la pantalla	6 horas	sáb 30/04/22	lun 02/05/22
FavoriteHeroesEvents	2 horas	lun 02/05/22	mar 03/05/22
ViewModel	6 horas	mar 03/05/22	jue 05/05/22
Proves	**11 horas**	**jue 05/05/22**	**dom 08/05/22**
Proves unitaries	6 horas	jue 05/05/22	sáb 07/05/22
Proves amb usuaris	2 horas	sáb 07/05/22	sáb 07/05/22
Millores al codi	3 horas	dom 08/05/22	dom 08/05/22
Redacció apartat Implementació	9 horas	lun 09/05/22	mié 11/05/22
Lliurament PAC 3	0 horas	mié 11/05/22	mié 11/05/22



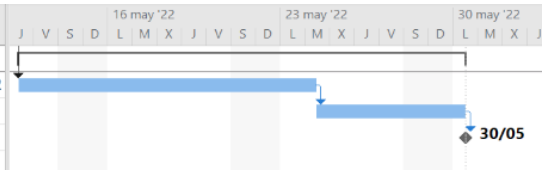
Il·lustració 5. Planificació fase d'Implementació 2 de 2

1.4.4.4 Fase Lliurament Final

En aquesta fase es realitzarà la memòria final i el vídeo de presentació.

A continuació es mostra el detall de la planificació d'aquesta fase.

Nombre de tarea	Duración	Comienzo	Fin
PAC 4 - Lliurament Final	**57 horas**	**jue 12/05/22**	**lun 30/05/22**
Redacció de la memòria	37 horas	jue 12/05/22	mar 24/05/22
Video de presentació	20 horas	mar 24/05/22	lun 30/05/22
Lliurament PAC 4	0 horas	lun 30/05/22	lun 30/05/22



Il·lustració 6. Planificació fase Lliurament Final

1.5 Breu sumari de productes obtinguts

Els productes obtinguts en aquest treball són:

- Codi font i APK de l'aplicació nativa per a mòbils Android 5.0 (API 21) o superior
- Manual d'usuari
- Document de la memòria del projecte
- Vídeo de presentació del projecte

1.6 Breu descripció dels altres capítols de la memòria

Als següents capítols d'aquesta memòria es descriuran cada una de les fases del procés de desenvolupament de l'aplicació. A continuació es fa una breu descripció del mateixos:

- Capítol 2 - Disseny de l'aplicació: En aquest capítol es desenvolupa el disseny de l'aplicació.
  En el primer apartat es farà servir una metodologia de Disseny Centrat a l'Usuari (DCU) per determinar les necessitats dels nostres usuaris i poder així definir correctament les funcionalitats a desenvolupar, les pantalles que tindrà l'aplicació i la navegació entre elles.
  En el segon apartat, es detallarà el disseny tècnic de l'aplicació, determinant els casos d'ús que tindrà l'aplicació, les diferents capes de la seva arquitectura, la relació entre les capes a la vegada que detallaran les classes que es faran servir i es definirà l'estructura de la base de dades.

- Capítol 3 – Implementació: En aquest capítol es detalla la fase de codificació del projecte i es recullen les decisions més importants que s'han pres durant aquesta fase. També es detallen les llibreries utilitzades i les proves realitzades.

- Capítol 4 – Conclusions: Exposem les conclusions del projecte, objectius aconseguits i lliçons apreses durant el procés. També es comenten les futures millores que pot tenir l'aplicació.

- Capítol 5 – Glossari: Es mostra el glossari amb els termes més importants d'aquest document.

- Capítol 6 – Bibliografia: Es mostra la bibliografia utilitzada per la redacció d'aquest document.

- Capítol 7 – Annexos: Conté el manual d'usuari de l'aplicació.

2 Disseny de l'aplicació

En aquest capítol es desenvolupa el disseny de l'aplicació a implementar.

En el primer apartat es fa l'anàlisi dels nostres usuaris focals amb l'objectiu de conèixer les seves característiques i necessitats.
Coneixent els nostre usuaris podrem definir les funcionalitats que ha de tenir l'aplicació i tenir una idea clara de com organitzar la informació en les diferents pantalles.
Com a resultat d'aquest anàlisi obtindrem un primer prototip avaluable de com serà l'aplicació.

Una vegada es té clar què s'ha de fer, en el segon apartat fem el disseny tècnic, entrant en detall a l'arquitectura del sistema, de la base de dades i de les classes a implementar.

2.1 Disseny centrat a l'usuari (DCU)

Per tal de conèixer amb el màxim detall possible quins seran els usuaris de la nostra aplicació així com identificar els seus objectius, motivacions i necessitats s'ha seguit un procés de disseny centrat en el usuari (DCU) [4].

Mantenir a l'usuari en el centre del nostre disseny ens permetrà tant cobrir totes les seves necessitats com reduir el temps i costos del desenvolupament, ja que ens podrem centrar en aquelles funcionalitats que realment aporten valor als usuaris.

2.1.1 Definició d'usuaris i context d'ús

Seguint aquest procés DCU, s'han identificat com són els nostres usuaris focals, en quins contextos faran servir l'aplicació i quines necessitats volen cobrir amb ella.

S'han identificat 2 perfils d'usuaris que faran servir la nostra aplicació:

- Pares de nens de Primària amb interès pels còmics i/o col·leccionar cromos de superherois: Les necessitats principals d'aquests usuaris és la de compartir amb els seus fills els seus interessos relacionats amb els còmics i els superherois. Per a ells aquesta aplicació és una eina de consulta que li ajudarà a introduir al seus fills al món dels còmics i dels superherois

- Homes joves amants dels còmics: Aquests usuaris faran servir l'aplicació per cobrir la seva necessitat de conèixer tots els detalls possibles dels diferents superherois. També la faran servir per passar una bona estona posant a prova els seus coneixements sobre els diferents superherois jugant contra el mòbil

A continuació es mostra la fitxa de persona i escenari de cada un d'aquests perfils d'usuari:

Font: pixabay.com

Nom: Pere
Edat: 43
Nivell d'Estudis: Llicenciatura
Professió: Enginyer

Descripció de la persona:

El Pere es pare d'un nen de 8 anys. Des de petit li han agradat els còmics i tot el món dels superherois. Aquesta afició li ha transmès al seu fill que, des de que sap llegir, devora còmics i comparteix amb el seu pare les histories que llegeix. Compartir amb el seu fill aquest hobby ha reforçat molt els vincles entre pare i fill i fa que el Pere recordi moments molt especials de la seva infantesa.

Descripció d'un escenari:

Diumenge al matí al mercat de Sant Antoni de Barcelona. El Pere i el seu fill han anat a intercanviar cromos i a buscar algun còmic de l'època en que el Pere era petit.
En una de les paradetes troben còmics amb superherois quasi oblidats i que al Pere li encantaven quan era petit. Al seu fill no li acaba d'agradar perquè no els coneix i només vol llegir sobre el seu superheroi favorit.
El Pere vol que el seu fill conegui altres superherois, no només els que ara estan de moda i que tots els nens coneixen, així que busca a la nostra aplicació aquests superherois antics i, amb la informació que obté, li pot explicar la relació que hi ha entre els antics superherois i el superheroi que tant li agrada al seu fill, quins poders tenen i quins són bons i quins dolents.
Al explicar-li tot això i fullejar els còmics, el seu fill li agafen ganes de llegir aquests còmics antics.
El Pere també veu altres còmics amb superherois que ja ni recordava, els busca a l'aplicació i els guarda a favorits per tenir-los apuntats per a la propera vegada que vingui a comprar còmics.

Font: www.istockphoto.com

Nom: Èric
Edat: 22
Professió: Estudiant universitari

Descripció de la persona

L'Èric és un estudiant universitari de 22 anys. Li agrada tot tipus de tecnologia i gadgets tecnològics. Des de sempre, una de les seves passions són les pel·lícules i còmics de superherois i li encanta conèixer nous superherois. Va cada dia a la universitat en transport públic i passa aquesta estona de viatge amb el seu mòbil jugant o navegant per Internet.

Descripció d'un escenari:

Dilluns al matí, l'Eric entra al metro a les 7:30 del matí per anar a la universitat. Avui troba un seient lliure i pot seure. Per passar l'estona agafa el seu mòbil per jugar una estona. Avui entra a la nostra aplicació i es posa a jugar.
Coneix les característiques i poders de molts superherois i passa un bona estona intentant guanyar a l'aplicació posant a prova els seus coneixements en la matèria.
Després d'unes partides, com que encara li queden uns minuts per arribar al seu destí, fa que la nostra aplicació li ensenyi alguns superherois a l'atzar, dels 5 que li mostra l'aplicació, en coneixia 3, però els altres 2 han sigut tota una descoberta per a ell i els guarda a la seva llista de favorits.

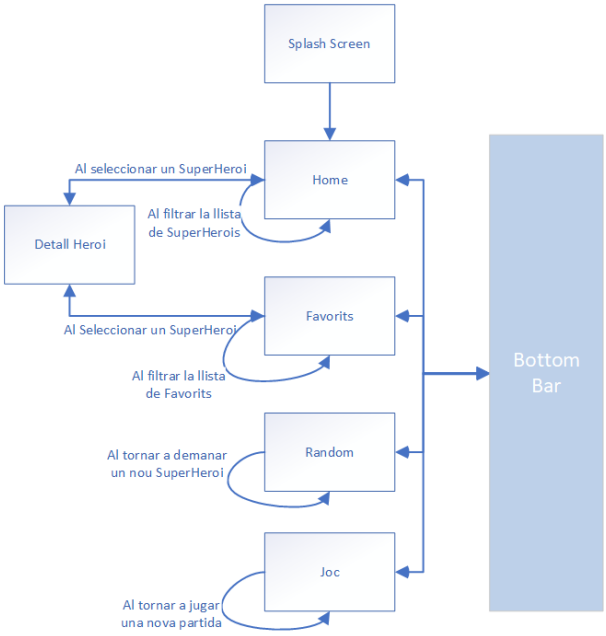
2.1.2 Definició de les funcionalitats

Una vegada coneixem els nostres usuaris i les seves necessitats, podem definir les funcionalitats que ha de tenir la nostra aplicació:

- **Cerca de superherois per nom:** La nostra aplicació ha de poder permetre la cerca de superherois pel seu nom
- **Guardar a favorits:** L'aplicació ha de permetre tenir una llistat de favorits
- **Descobrir nous superherois:** L'aplicació ha de tenir l'opció de mostrar nous superherois als usuaris
- **Joc utilitzant característiques del superherois:** L'aplicació ha de tenir un joc que faci servir les dades disponibles dels superherois

2.1.3 Arbre de navegació

Una vegada definides les funcionalitats que ha de tenir la nostra aplicació, podem crear l'arbre de navegació que volem que tingui.



Il·lustració 7: Arbre de navegació de l'aplicació

Com es pot observar a la il·lustració anterior, l'aplicació començarà amb un pantalla de "Splash" i seguidament arribarem a la pantalla d'inici (Home) on es mostraran el llistat de tots els superherois disponibles a l'aplicació.

Per navegar entre pantalles hi haurà una barra inferior que permetrà navegar d'una pantalla una altre.

A la pantalla "Favorits" es mostraran els superherois que l'usuari ha anant seleccionant com a favorits.

Des de la pantalla "Home" i "Favorits" es pot anar a la pantalla de detall d'un Superheroi en concret. En aquestes pantalles també es pot filtrar el seu llistat de superherois per mostrar només aquells que ens interessi.

A la pantalla "Random" es mostrarà un superheroi de forma aleatòria i des de la mateixa pantalla es podrà tornar a demanar un nou superheroi aleatori.

A la pantalla "Joc" es podrà jugar contra l'aplicació i, una vegada finalitzada la partida, es tindrà l'opció de tornar a jugar una nova partida.

2.1.4 Prototipat

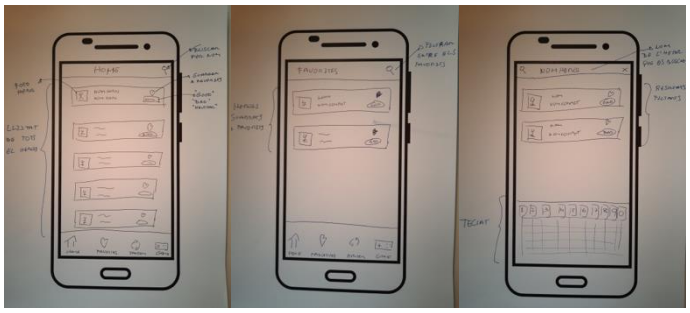
Una vegada definit l'arbre de navegació, es comença a fer el prototipatge de l'aplicació. Aquest prototipatge s'ha realitzat ens 3 etapes:

- Esbossos a mà alçada
- Prototips de baixa fidelitat
- Prototips d'alta fidelitat

2.1.4.1 Esbossos a mà alçada

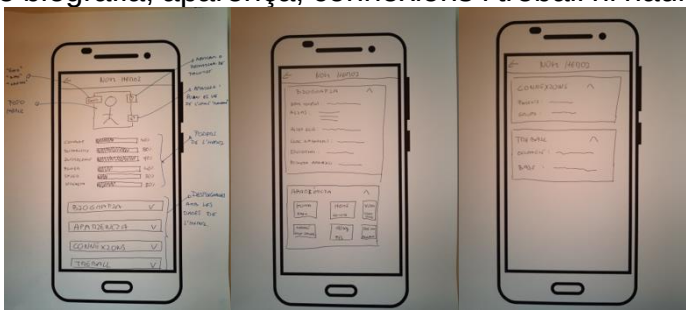
Per començar a donar forma a l'aplicació, s'han fet una sèrie d'esbossos en paper on s'ha apuntat les idees generals de les diferents pantalles de l'aplicació.

La pantalla inicial mostrarà tot el llistat d'herois, mentre que la pantalla de favorits només els que s'hagin marcat com a favorits. A les dues pantalles es tindrà l'opció de filtrar per mostrar només aquells herois que tinguin el nom que busquem.



Il·lustració 8: Esbós pantalla inicial i llista de favorits

Al seleccionar un heroi, ens apareixerà una fitxa amb la seva amb la foto i poders. Per les dades de biografia, aparença, connexions i treball hi haurà desplegables.



Il·lustració 9: Esbós fitxa dels herois

En la pantalla del joc es mostraran dues cartes donades la volta. L'usuari haurà d'elegir una d'elles per començar el joc i l'aplicació jugarà amb l'altre carta.
Una vegada elegida la carta, al usuari se li mostrarà la foto i el nom de l'heroi i haurà d'apostar per algun del seus poders.
Si el poder elegit per l'usuari és superior al de l'altre carta, l'usuari guanya, si és inferior, perd.

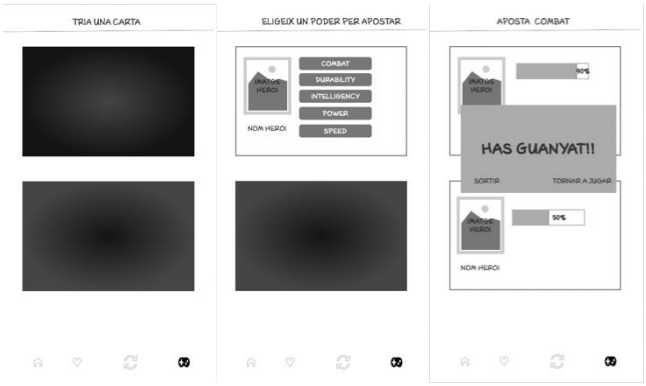
Il·lustració 10: Esbós pantalles del joc

2.1.4.2 Prototips de baixa fidelitat

En aquesta etapa del prototipatge, s'han passats els esbossos anteriors a un prototip de baixa fidelitat (wireframe) que ens ha permès comprovar com es veurien les diferents pantalles en un dispositiu real.

Il·lustració 11: Wireframe pantalla inici i llista de favorits

Il·lustració 12: Wireframe fitxa dels l'herois i heroi aleatori



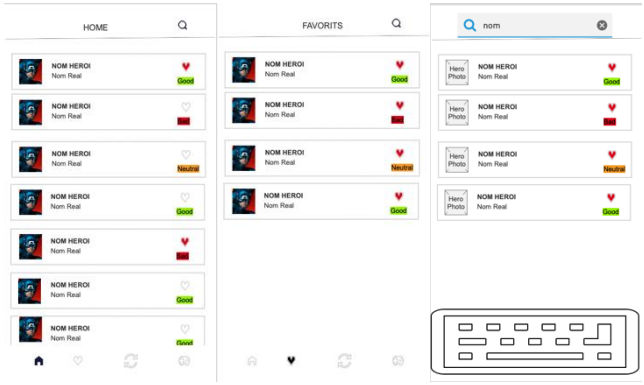
Il·lustració 13: Wireframe pantalles de joc

2.1.4.3 Prototips d'alta fidelitat

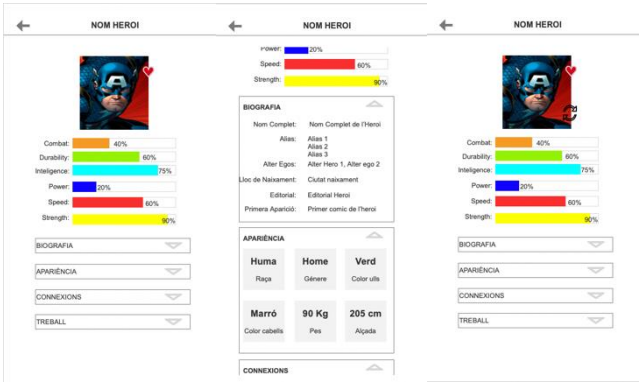
En aquesta darrera etapa del prototipatge, es realitza un prototip d'alta fidelitat (Mockup) que, al ser interactiu, ens ha permès testejar les nostres idees més enllà del paper.

Aquest prototip es pot veure a la següent URL: https://ybf8cl.axshare.com

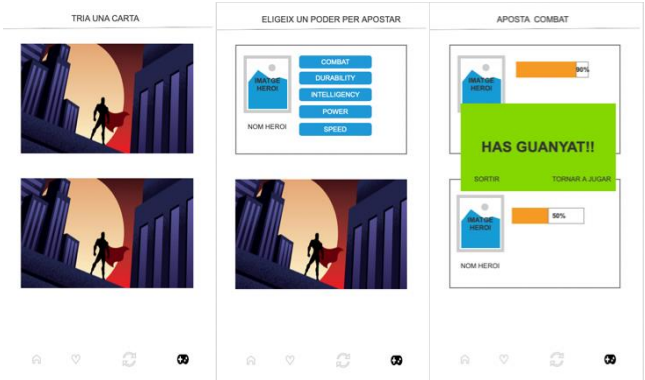
Adjuntem algunes captures de les pantalles d'aquest prototip d'alta fidelitat.



Il·lustració 14: Mockup pantalla inici i llista de favorits



Il·lustració 15: Mockup fitxa de l'heroi i heroi aleatori



Il·lustració 16: Mockup pantalla de joc

2.1.4.4 Avaluació dels prototips

Per tal d'avaluar els prototips de les diferents pantalles, es fa una avaluació amb diferents usuaris on se li demana una sèrie de tasques a realitzar:

- Baixa per la llista de superherois i marcar com a favorit el primer superheroi que sigui neutral
- Buscar un superheroi en concret i marcar-lo com a favorit
- Anar a la llista de favorits i desmarcar un superheroi
- Entrar a la fitxa d'un superheroi:
  - Trobar si pertany a algun grup de superherois
  - Trobar quin color d'ulls té el superheroi
  - Trobar quin és el nom real del superheroi
- Fer que l'aplicació et mostri un superheroi a l'atzar:
  - Marcar-lo com a favorit
  - Tornar a fer que l'aplicació et mostri un nou superheroi a l'atzar
  - Torna a la llista de tots els superherois
- Jugar dues partides seguides.
  - A la primera partida, elegir la carta que hi ha a la part superior
  - A la segona partida, elegir la carta que hi ha a la part inferior
  - Una vegada acabada la segona partida, tornar a la pantalla d'inici

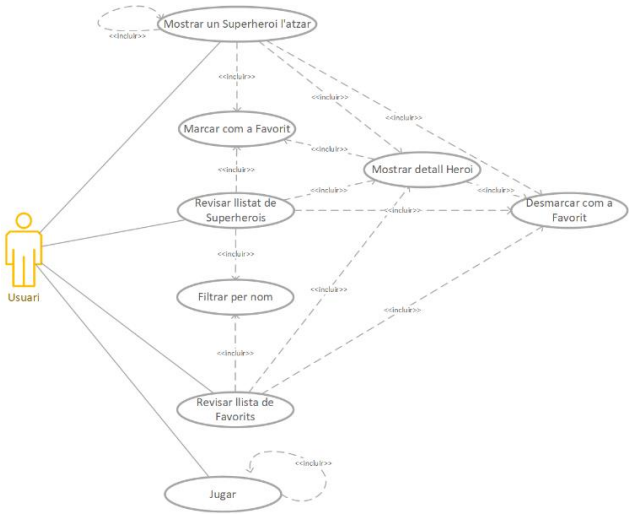
En tots els casos, els resultats de les proves varen ser satisfactòries i tots els usuaris enquestats varen saber realitzar totes les tasques sense problemes.

2.2 Disseny tècnic

En aquest segon punt es realitzarà el disseny tècnic de l'aplicació.

2.2.1 Definició dels casos d'ús

A continuació es mostra un esquema dels casos d'ús de l'aplicació i es fa una explicació de cada un d'ells:



Il·lustració 17: Casos d'ús de l'aplicació

Cas d'ús 1: Revisar llistat de Superherois	
Descripció	L'usuari pot recórrer el llistat de tots els superherois disponibles a l'aplicació
Actors	Usuari
Prioritat	Alta
Precondicions	L'usuari ha iniciat l'aplicació o ha navegat a la pàgina "Home"
Fluxe	Alternativa 1: • L'usuari ha iniciat l'aplicació i entra a la pàgina d'inici Alternativa 2: • L'usuari accedeix a la pàgina d'inici des de la barra de navegació inferior
Postcondicions	Es mostra el llistat de tots els superherois de l'aplicació
Notes	S'han de mostrar tots els superherois, tant si el mòbil està connectat a Internet com si no hi està. Es pot assumir que en aquest segon cas no es mostrin les imatges

Cas d'ús 2: Revisar llista de favorits	
Descripció	L'usuari pot recorre el llistat dels superherois que ha guardat com a favorits
Actors	Usuari
Prioritat	Alta
Precondicions	L'usuari ha navegat a la pàgina de "Favorits" a través de la barra de navegació inferior
Fluxe	L'usuari accedeix a la pàgina de "Favorits" des de la barra de navegació inferior
Postcondicions	Es mostra el llistat dels superherois que l'usuari té guardats com a favorits. Si no té cap superheroi guardat com a favorit, s'ha de mostrar un missatge informant d'aquest fet
Notes	S'han de mostrar tots els superherois marcats com a favorits, tant si el mòbil està connectat a Internet com si no hi està. Es pot assumir que en aquest segon cas no es mostrin les imatges

Cas d'ús 3: Filtrar per nom	
Descripció	L'usuari ha de poder filtrar les llistes dels superherois
Actors	Usuari
Prioritat	Mitja
Precondicions	L'usuari està a la pantalla "Home" o "Favorits"
Fluxe	Alternativa 1: • L'usuari ha iniciat l'aplicació i entra a la pàgina d'inici • Va a la barra superior d'aquesta pagina i posa el nom del superheroi que vol buscar Alternativa 2: • L'usuari va a la pàgina de "Favorits"

	• Va a la barra superior d'aquesta pagina i posa el nom del superheroi que vol buscar
Postcondicions	A mesura que l'usuari va escrivint, la llista de superherois de la pàgina es va filtrant per mostrar aquells on el seu nom conté el que l'usuari va escrivint.
Notes	Si no hi ha cap superheroi que coincideixi amb la cerca s'ha de mostrar un missatge informant d'aquest fet

Cas d'ús 4: Mostrar detall superheroi	
Descripció	Es mostra el detall d'un superheroi en concret
Actors	Usuari
Prioritat	Alta
Precondicions	L'usuari ha seleccionat un superheroi de la llista de superherois o de la llista de favorits
Fluxe	Alternativa 1: • L'usuari està a la pantalla "Home" i selecciona un dels superherois per veure els seus detalls Alternativa 2: • L'usuari està a la pantalla "Favorits" i selecciona un dels superherois per veure els seus detalls
Postcondicions	Es mostra el detall del superheroi seleccionat
Notes	S'han de mostrar els detalls de tots superherois, tant si el mòbil està connectat a Internet com si no hi està. Es pot assumir que en aquest segon cas no es mostrin les imatges

Cas d'ús 5: Marcar un superheroi com a favorit	
Descripció	Es marca un superheroi com a favorit
Actors	Usuari
Prioritat	Alta
Precondicions	L'usuari marca un superheroi com a favorit
Fluxe	Alternativa 1: • L'usuari està a la pantalla "Home" i marca un superheroi de la llista com a favorit Alternativa 2: • L'usuari està a la pantalla "Detall Heroi" i el marca com a favorit Alternativa 3: • L'usuari està a la pantalla de "Random" i marca com a favorit el superheroi que li ha sortit
Postcondicions	El superheroi té la marca que indica que és un dels favorits i surt a la llista de Favorits
Notes	Els superherois es marcaran com a favorits amb un cor de color vermell. Si no és favorit, el cor tindrà fons blanc

Cas d'ús 6: Desmarcar un superheroi com a favorit	
Descripció	Es desmarca un superheroi com a favorit
Actors	Usuari

Prioritat	Alta
Precondicions	L'usuari desmarca un superheroi com a favorit
Fluxe	Alternativa 1: • L'usuari està a la pantalla "Home" i desmarca un superheroi de la llista com a favorit Alternativa 2: • L'usuari està a la pantalla "Detall Heroi" i el desmarca com a favorit Alternativa 3: • L'usuari està a la pantalla de "Random" i desmarca com a favorit el superheroi que li ha sortit Alternativa 4: • L'usuari està a la pantalla "Favorits" i desmarca un dels superherois
Postcondicions	El superheroi ja no té la marca que indica que és un dels favorits i desapareix de la llista de Favorits
Notes	Els superherois es marcaran com a favorits amb un cor de color vermell. Si no és favorit, el cor tindrà fons blanc

Cas d'ús 7: Mostrar un superheroi a l'atzar	
Descripció	L'aplicació mostra un superheroi a l'atzar a l'usuari
Actors	Usuari
Prioritat	Alta
Precondicions	L'usuari ha navegat a la pàgina "Radom"
Fluxe	Alternativa 1: • L'usuari navega fins a la pàgina "Random" des de la barra de navegació inferior Alternativa 2: • L'usuari està a la pantalla "Random" i apreta el botó "Random" que surt al superheroi que es mostra de forma aleatòria
Postcondicions	Es mostra el detall d'un superheroi de forma aleatòria
Notes	Tant si el mòbil està connectat a Internet com si no hi està, l'usuari ha de poder veure un superheroi a l'atzar. En cas que el mòbil no estigui connectat a Internet es pot assumir que no es mostrin les imatges

Cas d'ús 8: Jugar	
Descripció	L'aplicació inicia el joc de cartes amb els superherois
Actors	Usuari
Prioritat	Mitja
Precondicions	L'usuari ha navegat a la pàgina "Game"
Fluxe	Alternativa 1: • L'usuari navega fins a la pàgina "Game" des de la barra de navegació inferior Alternativa 2: • L'usuari ha finalitzar un joc i demana tornar a jugar una nova partida
Postcondicions	S'inicia una nova partida del joc de cartes

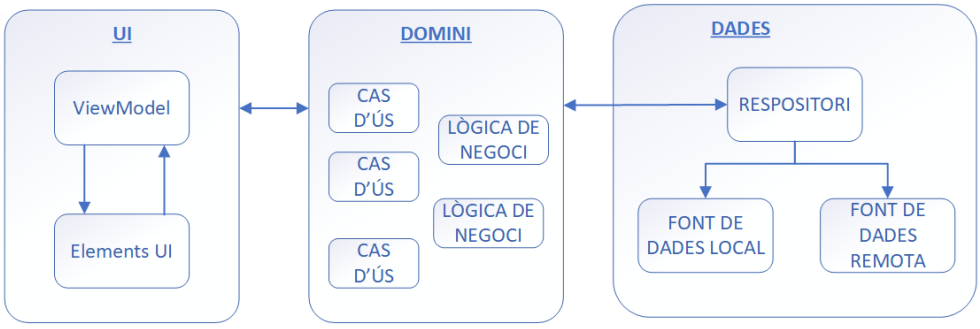
Notes	El joc te les següents regles: 1. Es mostren el revers de dues cartes elegides a l'atzar per l'aplicació 2. L'usuari ha de triar-ne una 3. La carta elegida es dona la volta i mostra a l'usuari el nom i la imatge del superheroi amb el que juga 4. L'usuari ha d'apostar per algun dels poders d'aquest superheroi sense saber el seu valor 5. Una vegada elegit el poder, es dona la volta l'altre carta i es comparen els valors del poder dels dos superherois 6. Guanya el que tingui un valor superior

2.2.2 Arquitectura de l'aplicació

Les arquitectures permeten desacoblar diferents unitats del codi de forma organitzada, d'aquesta manera el codi es fa més fàcil d'entendre, modificar i provar.

L'arquitectura de la nostra aplicació estarà basada en la Clean Architecture de Uncle Bob [5] adaptada a les necessitats específiques de les aplicacions Android [6] i [7]

L'aplicació estarà dividida en 3 capes (UI, Domini i Dades) tal i com es mostra a l'esquema següent:



Il·lustració 18: Capes de l'aplicació

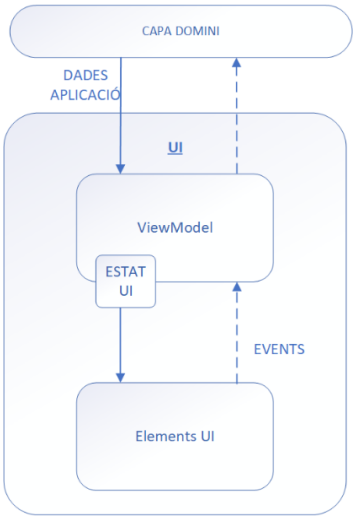
2.2.2.1 Capa de UI (UI Layer)

La funció de la UI és mostrar les dades de l'aplicació en pantalla i servir de punt principal d'interacció amb l'usuari. Quan les dades canvien, ja sigui per una interacció de l'usuari o per una entrada externa, la UI ha d'actualitzar el seu estat i mostrar els canvis que s'han produït.

Les classes que són responsables de la producció de l'estat de la UI i contenen la lògica necessària per realitzar aquesta tasca es denominen contenidors d'estat.

Seguint les recomanacions de Google, farem servir el patró MVVM (Model-View-ViewModel) pel que tindrem una implementació del tipus ViewModel com a contenidor d'estat de la UI.

A la il·lustració següent es pot veure la relació entre els elements de la UI i la seva classe ViewModel.



Il·lustració 19: Diagrama de la relació entre la UI i el seu ViewModel

En aquest esquema es veu com l'estat flueix cap avall i els events flueixen cap amunt. Aquest patró es denomina Flux Unidireccional de Dades. Les implicacions d'aquest patró per a l'arquitectura són les següents:

- El ViewModel conserva i exposa l'estat que consumirà la UI. L'estat de la UI són les dades d'aplicació que transforma el ViewModel.
- La IU notifica al ViewModel els events de l'usuari.
- L'estat actualitzat s'envia al UI per a la seva renderització.
- El passos anteriors es repeteix per a qualsevol event que causi una canvi a l'estat.

Fer servir el patró de Flux Unidireccional de Dades, ens permet:

- Coherència a les dades: Només hi ha una font d'informació per a la UI.
- Capacitat de realitzar proves: Com que la font d'estat està aïllada, es pot provar de forma independent de la UI.
- Capacitat de manteniment: El canvi de l'estat segueix un patró ben definit on els canvis són el resultat dels events de l'usuari i de les fonts de dades de l'aplicació.

Per construir les diferents pantalles, es farà servir Jetpack Compose, que és una nova forma de construir UI natives per a Android.

Jetpack compose ens permet construir les nostres pantalles de forma declarativa en lloc de fer-ho de forma imperativa com passava fins ara quan es feien servir els fitxers XML.

Les principals avantatges de fer servir un paradigma declaratiu com a Jetpack Compose enlloc d'un paradigma imperatiu pel desenvolupament de les UI com són els fitxers XML són:

- Les UI declaratives són més netes, clares de llegir i tenen millor rendiment que les UI imperatives.

- Amb el paradigma imperatiu dels fitxers XML, els desenvolupadors tenen que construir la UI descrivint manualment com reaccionen els elements als canvis i actualitzar els seus estats en conseqüència.
  En canvi, amb el paradigma declaratiu de Jetpack Compose només cal indicar què es vol que la UI mostri a l'usuari enlloc de com s'han de construir els elements. El "com" es deixa en mans del propi framework.

- Manipular les vistes de forma manual com es fa als XML augmenta la probabilitat d'errors.

- Compose permet fer més amb menys codi comparat amb XML.

- Compose permet un disseny més atòmic, permetent la reutilització dels diferents components d'una forma molt senzilla.

- Encaixa perfectament amb el patró de Flux Unidireccional de Dades que volem fer servir, ja que els components admeten estat i emeten events.

Dins la capa de UI es tindran:

- Les diferents pantalles de l'aplicació amb els seus ViewModel corresponents

- Els diferents components que es fan servir a les pantalles

- La definició del tema, colors i tipografies de l'aplicació

- La definició de la navegació entre les diferents pantalles

L'aplicació permetrà que l'usuari eleigeixi entre un tema clar (light theme) i un tema fosc (dark theme).

2.2.2.2 Capa de Domini (Domain Layer)

La capa de Domini és la responsable d'encapsular la lògica empresarial. Aquesta lògia empresarial la poden fer servir un o varis ViewModels de la capa de UI.

Aquesta capa fa servir els repositoris de la capa de dades, és a dir, depenen de les classes del repositori.

Per complir el principi d'inversió de dependències, que ens diu que les classes d'alt nivell no poden dependre d'implementacions de classes de baix nivell sinó d'abstraccions de les mateixes, les classes de la capa de negoci no dependran directament d'una implementació del repositori, sinó de la seva interfície cosa que ens permetrà desacoblar les dues capes i facilitarà les proves unitàries del codi.

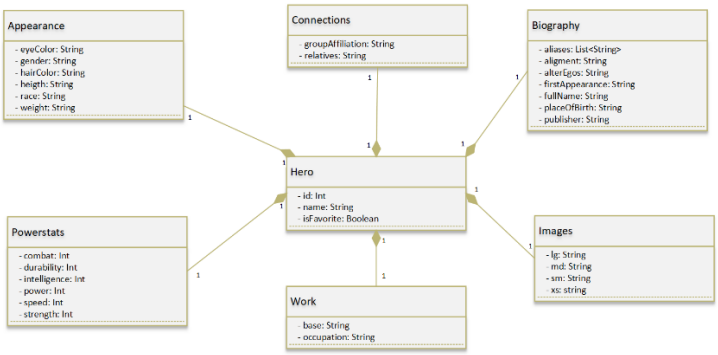
Si bé dins l'arquitectura recomana per Google, la capa de domini és optativa, he decidit implantar-la perquè ofereix els següents beneficis:

- Evita duplicitat de codi.
- Millora la llegibilitat del codi.
- Facilita les proves de l'aplicació
- Evita les classes massa grans, ja que permet dividir les responsabilitats

Dins aquesta capa es tindrà:

- Els diferents casos d'ús de l'aplicació
- El model de dades del domini
- La interfície del repositori que faran servir els diferents casos d'ús

A l'esquema següent es pot veure les classes del model dades del Domini:



Il·lustració 20: Model de dades del Domini

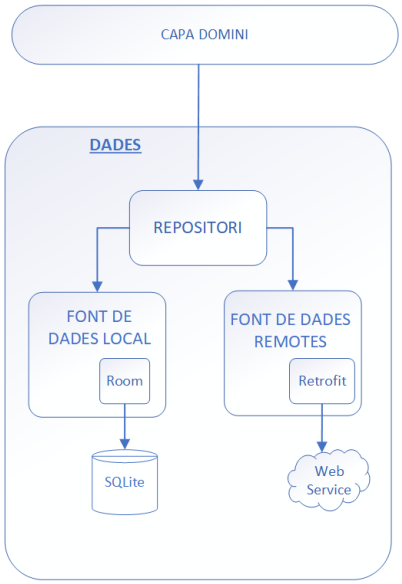
2.2.2.3 Capa de Dades (Data Layer)

La capa de Dades està formada per repositoris que poden contenir de zero a moltes fonts de dades.

En el nostre cas tindrem un repositori que implementarà la interfície definida a la capa de Domini i que tindrà dues fonts de dades:

- Font de dades Remota: Es farà servir Retrofit per obtenir les dades dels superherois a través d'una API pública
- Font de dades local: Es guardaran les dades obtingudes de l'API dins una BBDD local implementada amb Room

Al següent esquema es mostra la capa de Dades i els seus components:



Il·lustració 21: Capa de Dades de l'aplicació

2.2.2.3.1 Repositori de l'aplicació

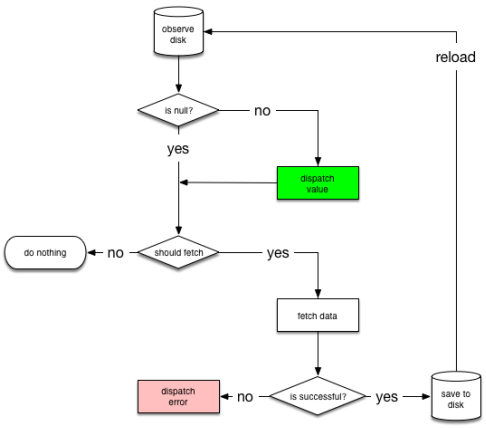
El repositori serà responsable de:

- Exposar les dades a la resta de l'aplicació.
- Centralitzar els canvis de les dades.
- Resoldre conflictes entre les dues fonts de dades (remota i local).
- Abstreure les fonts de dades de la resta de l'aplicació.

- Contenir la lògica de crear, emmagatzemar i canviar les dades de l'aplicació.

Volem que l'aplicació pugui funcionar sense connexió a Internet i que, si hi ha connexió a Internet, cada cop que s'iniciï l'aplicació s'actualitzin les dades del superherois.

Per aconseguir això, el repositori haurà d'implementar el següent esquema de decisió per exposar les dades [8]:



Il·lustració 22: Arbre de decisió del Repositori

D'aquesta manera, ens assegurem que com a mínim mostrem les dades que tenim a la BBDD local, independentment de si tenim o no connexió a Internet.

2.2.2.3.2 Font de dades remota

Com a font de dades remota es farà servir Retrofit per a la comunicació amb l'API que ens proveirà de les dades actualitzades dels superherois.

S'ha valorat dues possibles opcions com a API de l'aplicació:

- SuperHero API (https://www.superheroapi.com)
- Superhero-api (https://akabab.github.io/superhero-api/api/)

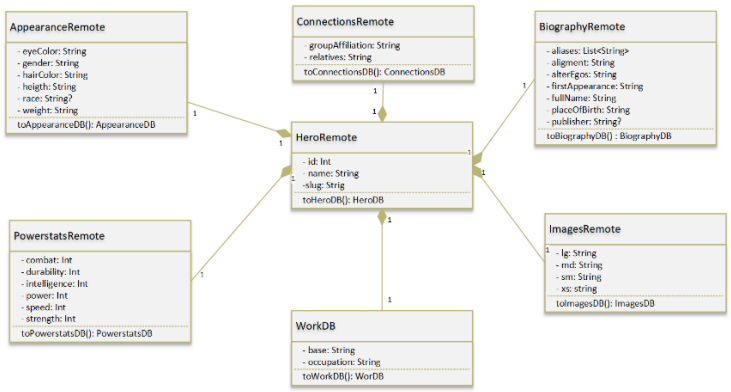
Hem optat per la segona opció perquè, tot i tenir menys superherois disponibles (563 superherois) que la primera opció (731 superherois), hem vist els següents punts a favor:

- Les dades dels superherois estan revisades i completes. Preferim menys entrades però de major qualitat.

- Permet obtenir tots els superherois amb una única crida a l'API, pel que ens permetrà obtenir tots els superherois una vegada i treballar després en local, permetent utilitzar l'aplicació sense necessitat de tenir connexió a Internet.

Seguint les bones pràctiques de Google i desacoblar la capa de dades de la capa del domini, la font de dades remota té les seves pròpies classes de dades o DTO (Data Transfer Object).

Cal notar que aquestes classes tenen unes funcions que permeten mapejar les DTO de la font de dades remota a les DTO de la font de dades local.



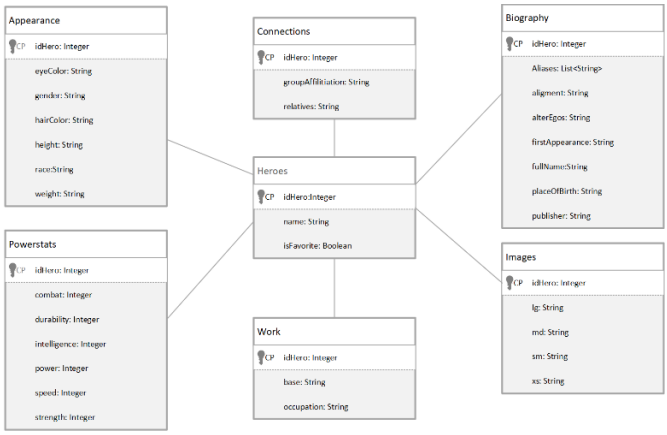
Il·lustració 23: DTO font de dades Remota

2.2.2.3.3 Font de dades local

La font de dades local serà una BBDD SQLite implementada amb Room.

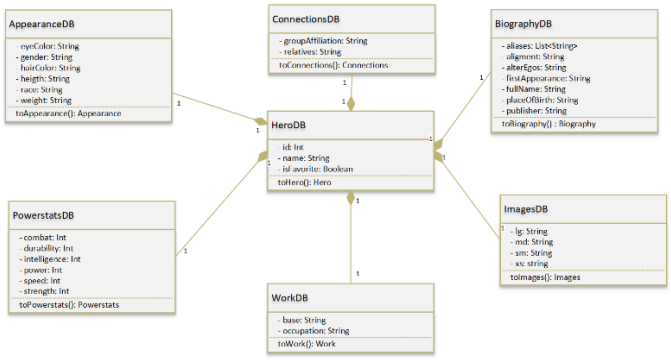
Aquesta font de dades local serà la nostra SSOT (Single Source of Truth). El repositori sempre obtindrà les dades d'aquesta font i s'encarregarà d'actualitzar les seves dades amb les dades obtingues a través de la font de dades remota.

La base de dades tindrà les següents taules i relacions entre elles:



Il·lustració 24: Esquema BBDD local

De la mateixa manera que a la font de dades remota, la font de dades local també tindrà les seves pròpies DTO's amb funcions per mapejar-les a les classes de dades del Model, que són les que el repositori exposarà a la resta de capes de l'aplicació.



Il·lustració 25: DTO font de dades Local

2.2.3 Injecció de Dependències (DI)

La injecció de dependències (DI) és una tècnica molt utilitzada en programació i adequada pel desenvolupament d'Android. [9]

La DI proporciona a les aplicacions els següents avantatges:

- Facilita la reutilització de les classes i el desacoblament de dependències.
- Facilita la refactorització.
- Facilita les proves.

Les classes solen requerir referencies a altres classes. Per exemple, a la nostra aplicació, els ViewModels necessiten una referència a un o més casos d'ús, és a dir, els ViewModels depenen dels casos d'ús per executar-se.

Els ViewModels podrien crear i inicialitzar les seves pròpies instancies dels cassos d'ús que necessitin, però això seria problemàtic perquè el ViewModel i el cas d'ús estarien totalment vinculats, no es podrien fer servir subclasses o implementacions alternatives i faria molt difícil provar de forma aïllada el ViewModels i poder anar modificant el cas d'ús segons els diferents casos de prova.

En lloc de fer que els ViewModels construeixin el seu propi objecte de cas d'ús, les bones pràctiques de Google ens indiquen que les dependències de les classes s'han de passar a través del seu constructor.

D'aquesta manera, desvinculem una classe de l'altre, millorant la reutilització de les classes i facilitant molt les proves.

Per automatitzar el procés de creació i provisió de dependències es farà servir Hilt [10], que és la biblioteca de Jetpack que es recomana per a la injecció de dependències a Android.

3 Implementació

En aquest apartat es descriuen els detalls d'implementació de l'aplicació, llibreries emprades i organització del codi.

3.1 Entorn de desenvolupament

L'entorn de desvolupament utilitzat ha sigut **Android Studio Bumblebee | 2021.1.1 Patch 3**.

L'ordinador utilitzat ha sigut un **MacBook Air (M1, 2020)** amb SO macOS Monterey (12.2.1)

Per les proves en terminal real, s'ha fet servir un **Samsung Galaxy A51** amb Android 11 – API 23

Com a llenguatge de programació s'ha fet servir Kotlin i les vistes de l'aplicació s'han desenvolupament amb Jetpack Compose.

3.2 Llibreries utilitzades

Per poder desenvolupar l'aplicació, s'han utilitzat les següents llibreries externes:

- Injecció de Dependències: Hilt
- Accés a servidor remot: Retrofit
- Base de dades local: Room
- Gestió d'imatges: Coil
- Animacions: Lottie

A continuació es mostra un extracte del fitxer "build.gradle" del mòdul on es detallen les versions utilitzades de cada llibreria:

```
//HILT
implementation 'com.google.dagger:hilt-android:2.41'
kapt 'com.google.dagger:hilt-compiler:2.41'
kapt "androidx.hilt:hilt-compiler:1.0.0"
implementation 'androidx.hilt:hilt-navigation-compose:1.0.0'

// Retrofit
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

// Room
implementation "androidx.room:room-runtime:2.4.2"
kapt "androidx.room:room-compiler:2.4.2"
implementation "androidx.room:room-ktx:2.4.2"

// Coil
implementation "io.coil-kt:coil-compose:1.3.2"

//Lottie
implementation "com.airbnb.android:lottie-compose:4.2.0"
```

3.3 Estructura de l'aplicació

Com s'indicava a l'apart de disseny, per la implementació d'aquesta aplicació es seguirà una arquitectura del tipus Clean Architecture i un patró de disseny MVVM.

Seguint aquest criteri, s'ha organitzat el codi de l'aplicació en els següents directoris:

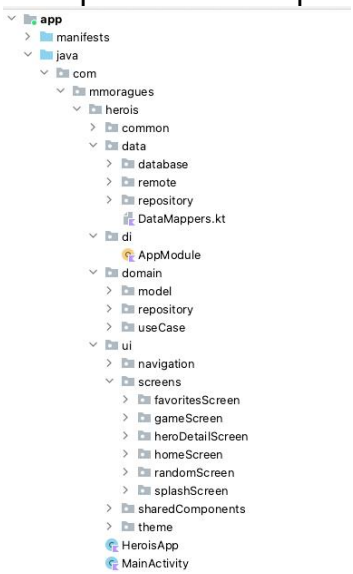
- Common → Directori on hi ha recursos comuns, constants i classes auxiliars de suport

- Data → Capa Data de l'arquitectura. Hi trobem:
  - Interfície i implementació de la "LocalDataSource" i de la "RemoteDataSource"
  - DTO (Data Transfer Objects) de la base de dades local i del servidor remot
  - Implementació del repositori
  - Funcions per mapejar les DTO a classes del domini i a la inversa

- Di → Mòdul Hilt des d'on es fa la injecció de dependències

- Domain → Capa del Domini de l'aplicació. Hi trobem:
  - Classes del model de l'aplicació
  - Interfície del repositori
  - Casos d'ús

- Ui → Capa UI de l'aplicació. Hi trobem
  - El controlador que gestiona la navegació entre les diferents pantalles
  - Les diferents pantalles de l'aplicació i, seguint el patró MVVM, el seu View Model associat
  - Components comuns a les diferents pantalles
  - Tema i colors de l'aplicació. Amb opció clara i fosca

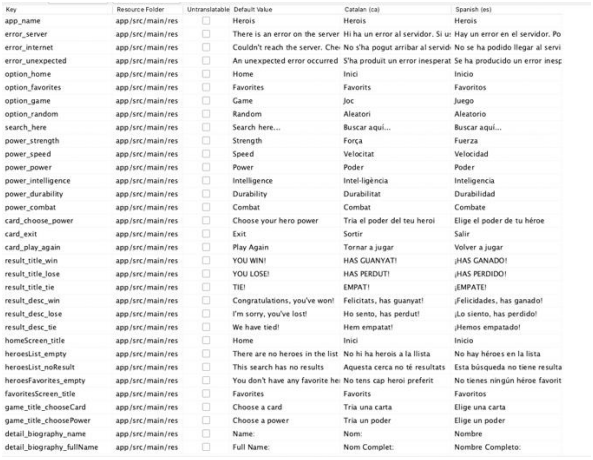


Il·lustració 26: Directoris de l'aplicació

3.4 Internalització de l'aplicació - I18n

L'aplicació s'ha preparat per a permetre la internalització (I18n). L'idioma per defecte és l'anglès i també té traducció al Català i Espanyol.

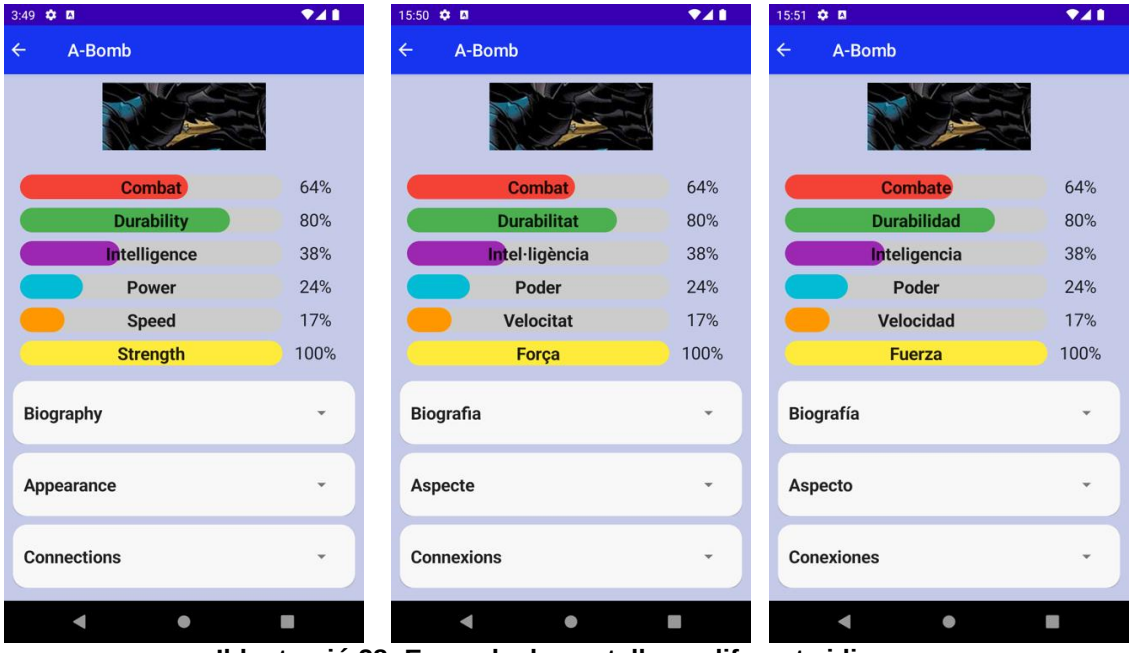
Cal notar que l'API des d'on s'obté la informació dels herois només està en anglès, el que es tradueix són els texts de la pròpia aplicació (botons, missatges, etc.)



Key	Resource Folder	Untranslatable	Default Value	Catalan (ca)	Spanish (es)
app_name	app/src/main/res		Herois	Herois	Herois
error_server	app/src/main/res		There is an error on the server	Hi ha un error al servidor. Si u	Hay un error en el servidor. Po
error_internet	app/src/main/res		Couldn't reach the server. Che	No s'ha pogut arribar al servid	No se ha podido llegar al servi
error_unexpected	app/src/main/res		An unexpected error occurred	S'ha produit un error inesperat	Se ha producido un error inesp
option_home	app/src/main/res		Home	Inici	Inicio
option_favorites	app/src/main/res		Favorites	Favorits	Favoritos
option_game	app/src/main/res		Game	Joc	Juego
option_random	app/src/main/res		Random	Aleatori	Aleatorio
search_here	app/src/main/res		Search here...	Buscar aquí...	Buscar aquí...
power_strength	app/src/main/res		Strength	Força	Fuerza
power_speed	app/src/main/res		Speed	Velocitat	Velocidad
power_power	app/src/main/res		Power	Poder	Poder
power_intelligence	app/src/main/res		Intelligence	Intel·ligència	Inteligencia
power_durability	app/src/main/res		Durability	Durabilitat	Durabilidad
power_combat	app/src/main/res		Combat	Combat	Combate
card_choose_power	app/src/main/res		Choose your hero power	Tria el poder del teu heroi	Elige el poder de tu héroe
card_exit	app/src/main/res		Exit	Sortir	Salir
card_play_again	app/src/main/res		Play Again	Tornar a jugar	Volver a jugar
result_title_win	app/src/main/res		YOU WIN!	HAS GUANYAT!	¡HAS GANADO!
result_title_lose	app/src/main/res		YOU LOSE!	HAS PERDUT!	¡HAS PERDIDO!
result_title_tie	app/src/main/res		TIE!	EMPAT!	¡EMPATE!
result_desc_win	app/src/main/res		Congratulations, you've won!	Felicitats, has guanyat!	¡Felicidades, has ganado!
result_desc_lose	app/src/main/res		I'm sorry, you've lost!	Ho sento, has perdut!	¡Lo siento, has perdido!
result_desc_tie	app/src/main/res		We have tied!	Hem empatat!	¡Hemos empatado!
homeScreen_title	app/src/main/res		Home	Inici	Inicio
heroesList_empty	app/src/main/res		There are no heroes in the list	No hi ha herois a la llista	No hay héroes en la lista
heroesList_noResult	app/src/main/res		This search has no results	Aquesta cerca no té resultats	Esta búsqueda no tiene resulta
heroesFavorites_empty	app/src/main/res		You don't have any favorite he	No tens cap heroi preferit	No tienes ningún héroe favorit
favoritesScreen_title	app/src/main/res		Favorites	Favorits	Favoritos
game_title_chooseCard	app/src/main/res		Choose a card	Tria una carta	Elige una carta
game_title_choosePower	app/src/main/res		Choose a power	Tria un poder	Elige un poder
detail_biography_name	app/src/main/res		Name:	Nom:	Nombre
detail_biography_fullName	app/src/main/res		Full Name:	Nom Complet:	Nombre Completo:

Il·lustració 27: Fitxer Strings per a la Internalització

A continuació es mostra un exemple de la mateixa pantalla en els tres idiomes.

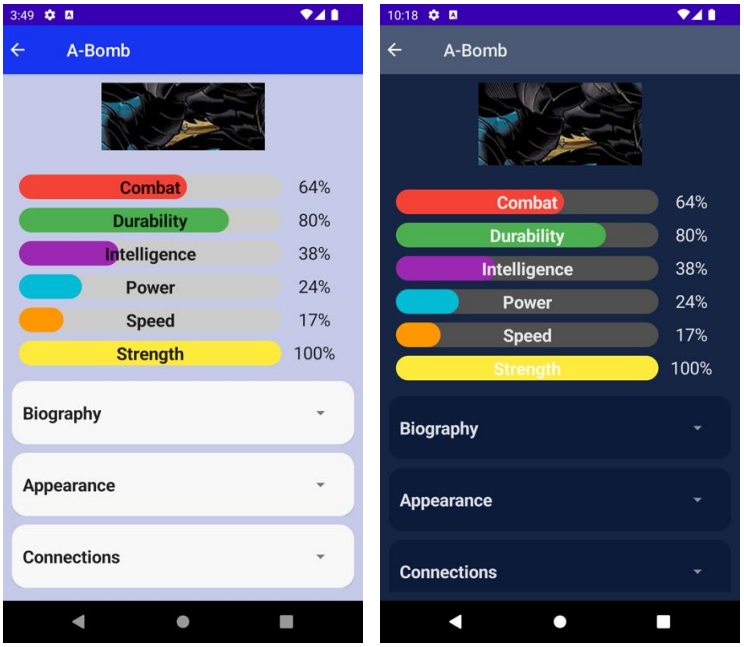


Il·lustració 28: Exemple de pantalla en diferents idiomes

3.5 Modes Clar i Fosc

L'aplicació permet a l'usuari elegir entre el mode clar i el mode fosc, amb patrons de colors diferents per a cada mode.

A continuació es mostra un exemple d'una pantalla en el dos modes



Il·lustració 29: Mode clar i mode obscur de la mateixa pantalla

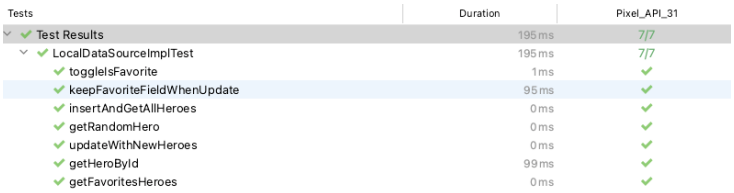
3.6 Proves

En aquest apartat es detallen el conjunt de proves realitzades.

3.6.1 Proves unitàries

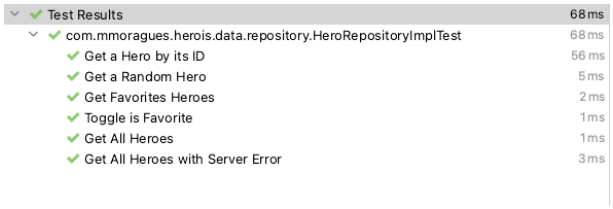
S'han fer proves unitàries de la Base de dades locals, el Repositori i els casos d'ús:

- Base de dades locals: Es prova la base de dades locals amb les següents proves:
  - Marcar a un heroi com a favorit
  - Confirmar que al actualitzar les dades d'un heroi es manté que es un heroi favorit
  - Afegir i obtenir tots els herois
  - Obtenir un heroi aleatori
  - Afegir nous herois a la base de dades
  - Obtenir un heroi pel seu ID
  - Obtenir la llista d'herois favorits



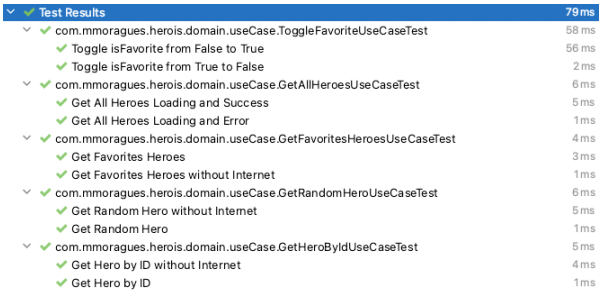
Il·lustració 30: Resultats proves base de dades locals

- Repositori: Es prova la implementació del repositori amb les següents proves:
    - Obtenir un heroi pel seu ID
    - Obtenir un heroi aleatori
    - Obtenir la llista d'herois favorits
    - Afegir un heroi a la llista d'herois
    - Obtenir tots els herois en situació normal
    - Obtenir tots els herois quan no es pot accedir al servidor remot



Il·lustració 31: Resultats proves del repositori

- Casos d'ús: Es proven els diferents casos d'ús amb les següents proves:
    - Afegir i treure un heroi de la llista de favorits
    - Obtenir tots els herois tant quan es té accés al servidor remot i quan no es té
    - Obtenir els herois favorits quan es té accés al servidor remot i quan no es té
    - Obtenir un heroi aleatòria quan es té accés al servidor remot i quan no es té
    - Obtenir un heroi a través seu ID quan es té accés al servidor remot i quan no es té



Il·lustració 32: Resultat proves dels casos d'ús

3.6.2 Proves d'Integració

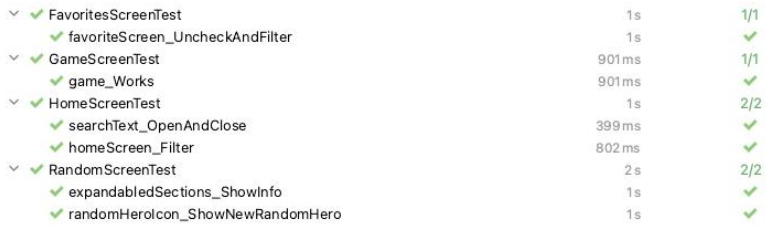
S'han fet proves d'integració de les diferents pantalles amb el seu View Model:

- Pantalla de favorits:
    - Es comprova que surten els missatges correctes quan no hi ha herois a la llista de favorits o quan es filtra per un heroi que no existeix
    - Es comprova que es pot treure un heroi de la llista

- Pantalla de jocs:
    - Es comprova que es pot jugar una partida correctament

- Pantalla d'inici:
    - Es comprova que es pot obrir i tancar correctament el buscador
    - Es comprova que es pot buscar correctament un heroi
    - Es comprova que si es fa una búsqueda d'un heroi que no existeix surt el missatge corresponent

- Pantalla d'heroi aleatori:
    - Es comprova que surt la icona per buscar un nou heroi aleatori i que funciona correctament
    - Es comprova que es poden obrir i tancar totes les seccions amb la informació dels herois



Il·lustració 33: Resultat de les proves d'integració

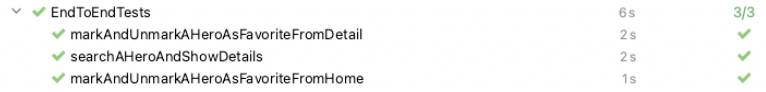
3.6.3 Proves extrem a extrem

Es fan proves d'extrem a extrem, simulant el comportament habitual d'un usuari.

- Busquem un heroi, anem a la seva fitxa de detall i el marquem com a favorit. Es comprova que surt a la llista de favorits. Es torna a la fitxa de detall, el desmarquem i comprovem que ja no surt a la llista de favorits

- Busquem un heroi a la pantalla d'inici, entrem a la fitxa de detall i comprovem que es poden obrir i tancar totes les seccions amb la informació dels herois

- Marquem un heroi com a favorit a la pantalla d'inici. Comprovem que surt a la llista de favorits. Tornem a la pantalla d'inici el desmarquem com a favorit i comprovem que ja no surt a la llista de favorits



3.6.4 Proves en dispositiu real

Es desplega l'aplicació en un dispositiu real i es fan proves d'utilització de l'aplicació i de navegació entres les diferents pantalles de forma satisfactòria.

4 Conclusions

La realització d'aquest treball en ha permès aprofundir en el nou kit d'eines de desenvolupament d'interfícies d'usuari (UI) de Google (Jetpack Compose), que està cridat a ser l'estàndard en la creació d'interfícies d'usuari per Android, deixant enrere el desenvolupament amb fitxers XML tradicionals.

També hem aconseguit implementar de les millors pràctiques recomanades per Google pel desenvolupament d'aplicacions com son el patró de disseny MVVM, l'arquitectura Clean Architecture i la Injecció de Dependències.

Considerem que l'elecció de Kotlin com a llenguatge de programació per a realitzar aquest projecte ha sigut molt encertada, ja que fa un temps que Google va declarar que Android és "Kotlin First", pel que poc a poc Java anirà perdent protagonisme dins l'ecosistema Android.

Seguir un Disseny Centrat a l'Usuari (DCU) ens ha permès centrar-nos en aquelles funcionalitats que aporten valor als usuaris, cosa que ens ha ajudat a concretar molt clarament l'abast del projecte, facilitant així la seva planificació.

Hem aconseguit tots els objectius marcats inicialment, tot i que s'han materialitzat els dos riscos identificats durant la fase de planificació del projecte:

- Endarreriment per la poca experiència amb Jetpack Compose
- Subestimar les tasques per afegir la pantalla de Joc

Per tal d'aconseguir arribar a les fites marcades, s'ha tingut que dedicar més hores de les esperades a la fase d'implementació, passant de les 126 hores estimades inicialment a 168 hores.

Com a punt de millora, caldria destacar que durant la fase de desenvolupament s'ha constatat que ens podríem haver estalviat els "Use Case" que es recomana tenir en una Clean Architecture. En el nostre cas aquesta capa no ens aporta cap benefici ja que únicament reenvia la petició que li fan els View Models cap al Repository, pel que ens podríem haver estalviat aquesta capa i que els View Models fessin les peticions directament al Repository.

Per a següents versions de l'aplicació es podrien desenvolupar funcionalitats d'interacció entre usuaris i millores en el joc, com per exemple:

- Afegir marcadors de les partides guanyades / perdudes
- Permetre jugar contra altres usuaris de l'aplicació i no únicament contra la mateixa aplicació
- Poder compartir la llista de favorits entre usuaris
- Tenir opció d'enviar missatges entre usuaris

5 Glossari

- **Programació Imperativa**: Paradigma de programació clàssic on hi ha una seqüència d'instruccions determinada que l'aplicació ha de seguir. Es centre en el "com" aconseguir la solució desitjada descrivint de forma explicita tots els passos a seguir i en quin ordre s'han de realitzar.

- **Programació Declarativa**: Paradigma de programació que té com a principi fonamental la descripció del resultat final que es busca, el "què", i no els passos que porten a la solució, el "com".

- **UI:** Interface d'usuari (User Interface) és tot allò que l'usuari pot veure i amb el que pot interactuar.

- **Jetpack Compose:** És un nou sistema de creació de UI declaratives per a Android. Està basat en funcions "composables" que permeten definir la UI de forma programàtica, descrivint com s'hauria de veure i proporcionant dependències de dades en lloc de centrar-se en el procés de construcció de la UI (inicialitzar un element, adjuntar-ho al seu pare, etc.).

- **Kotlin:** Llenguatge de programació estàtic de codi obert que admet programació funcional i orientada a objectes. Està dissenyat per tenir seguretat de tipus i contra nuls. Al 2019 Google va declarar aquest llenguatge com a prioritari pel desenvolupament d'aplicacions Android.

- **MVVM (Model-View-ViewModel):** Patró de disseny de la capa de presentació que permet separar el que es representa a la pantalla de la lògica de presentació.

- **Clean Architecture:** Arquitectura que separa les unitats de codi en diferents capes concèntriques, formant anells.
  La regla principal de la Clean Architecture és que el codi dels anells interns no ha de tenir coneixement de les funcions dels anells externs.
  L'anell més extern representa el nivell més baix del software (framework i UI) i que té més probabilitat de canvis i, a mesura que anem avançant, el nivell de software va essent més alt i amb menys probabilitat de canvis fins arribar a les entitats i la lògica de negoci que formen l'anell més intern.

- **Injecció de Dependències:** Patró de disseny en el que es subministren a una classe els objectes que necessita per funcionar enlloc que sigui la pròpia classe la que els creï.

6 Bibliografía

[1] «Android Developers Blog,» [WEB].
https://android-developers.googleblog.com/2021/07/jetpack-compose-announcement.html
[Darrer Accés: 1 Març 2022].

[2] «Google Play,» [WEB].
https://play.google.com/store/apps/details?id=com.zoomapps.superhero.information
[Darrer Accés: 1 Març 2022].

[3] J. Prieto Blàzques y Altres, Tecnologia i desenvolupament a dispositius mòbils, 2ª edició, Barcelona: Oberta UOC Publishing SL, 2017.

[4] J. F. Zampalo, Disseny de productes interactius multidispositiu, 1ª Edició, Barcelona: Oberta UOC Publishing SL, Barcelona, 2017.

[5] «The Clean Code Blog,» [WEB].
https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html
[Darrer Accés: 21 Març 2022].

[6] «Android Architecture Blueprints,» [WEB].
https://github.com/android/architecture-samples/tree/todo-mvp-clean
[Darrer Accés: 22 Març 2022].

[7] «Guide to app architecture,» [WEB].
https://developer.android.com/jetpack/guide
[Darrer Accés: 21 Març 2022].

[8] «NetworkBoundResource,» [WEB].
https://medium.com/@denisvcosta/networkboundresource-moor-offline-capabilities-to-your-flutter-app-e68e62e70cc6
[Darrer Accés: 22 Març 2022].

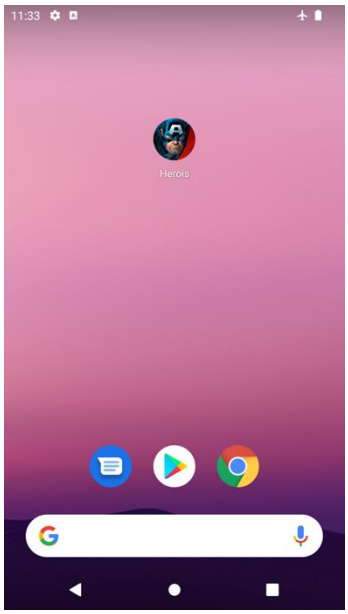
[9] «Dependency injection in Android,» [WEB].
https://developer.android.com/training/dependency-injection
[Darrer Accés: 22 Març 2022].

[10] «Dependency injection with Hilt,» [WEB]
https://developer.android.com/training/dependency-injection/hilt-android
[Darrer Accés: 23 Març 2022].

7 Annexos

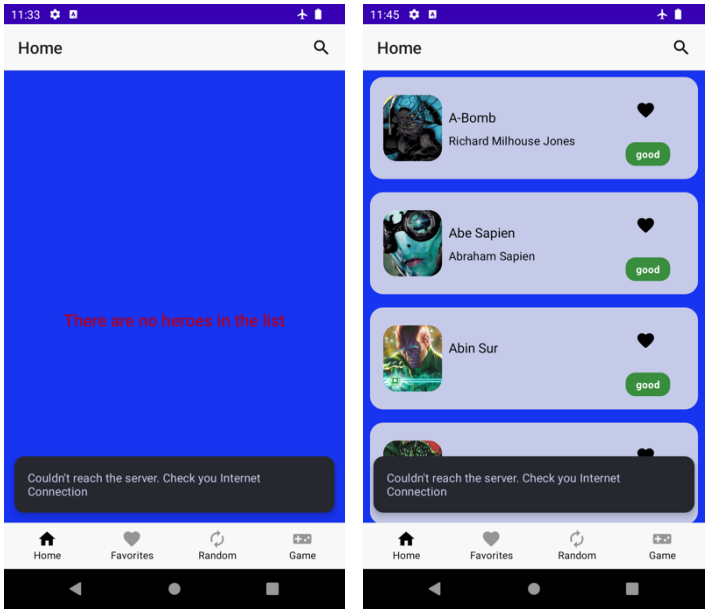
En aquest annex es mostra el manual d'usuari de l'aplicació.

1. Una vegada instal·lat Herois al mòbil, la versió mínima és Android 5.0 (API 21), es pot accedir-hi fent clic a la icona de l'aplicació.



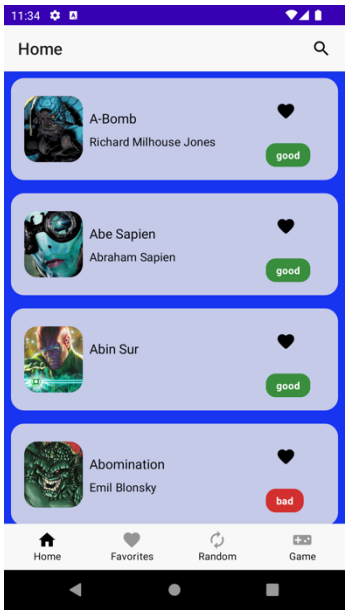
Il·lustració 34: Icona de l'aplicació

2. Una vegada dins l'aplicació, si no s'arriba al servidor per actualitzar els herois sortirà un avís.
   Si l'aplicació ja té els Herois guardats a la Base de dades local l'aplicació funcionarà amb les dades que té guardades.



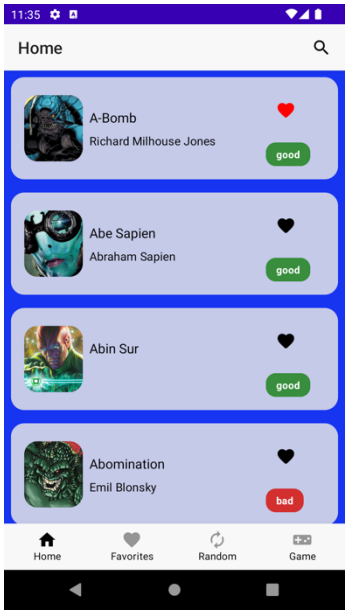
Il·lustració 35: Missatge d'error al no poder accedir al servidor

3. L'aplicació consta de 4 pantalles a les que es pot accedir a través de la barra de navegació inferior.
   Al iniciar l'aplicació, arribem a la pantalla d'Inici, on es mostra el llistat amb tots els herois disponibles a l'aplicació



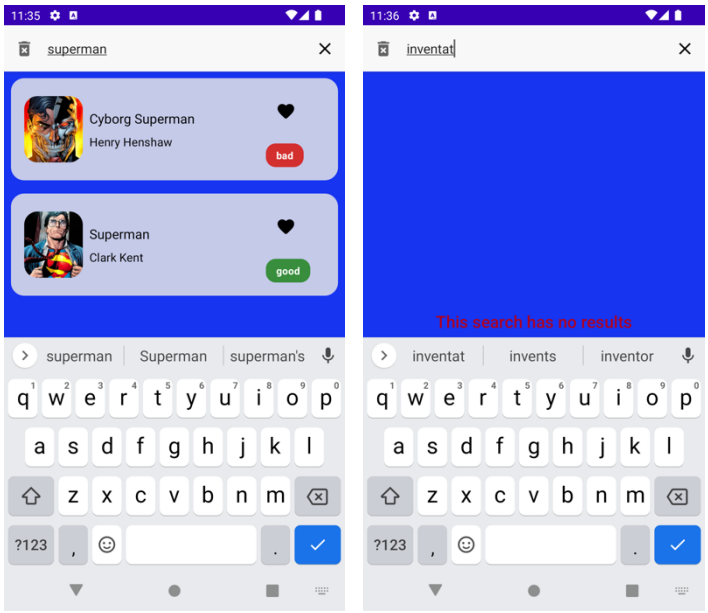
Il·lustració 36: Pantalla d'inici

4. A la pantalla d'inici es pot marcar o desmarcar com a favorit els herois fent clic sobre el cor. Si surt de color vermell l'heroi està marcat com a favorit.



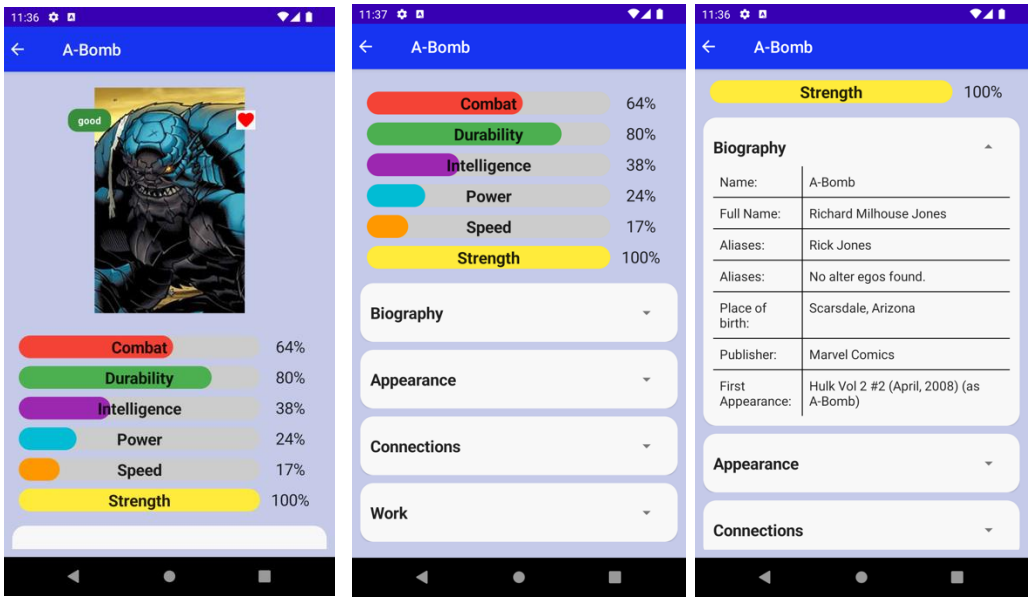
Il·lustració 37: Marcar com a favorit des de la pantalla d'inici

5. A la pantalla d'inici es poden filtrar els herois pel seu nom. Si el nom buscat no existeix s'indica amb un missatge



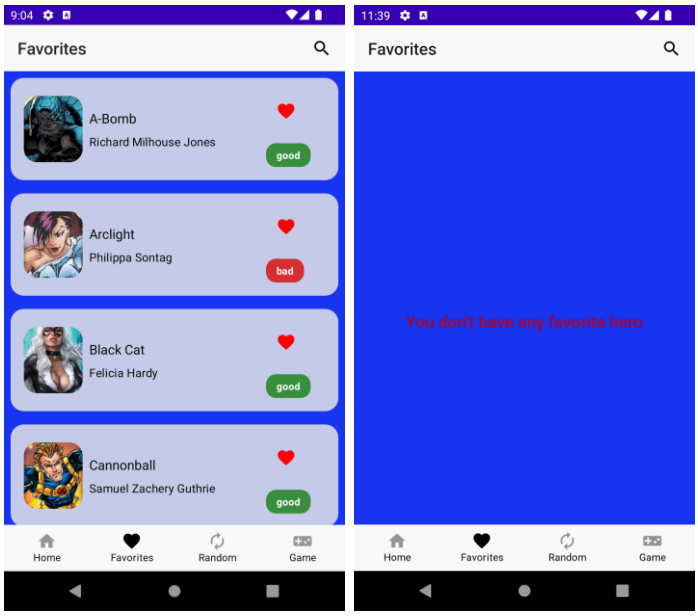
Il·lustració 38: Búsqueda a la pantalla d'inici

6. Si es fa clic sobre la fitxa de qualsevol dels herois, es mostrarà el detall d'aquest heroi.
   En aquesta fitxa hi ha informació de l'heroi
   - Imatge de l'heroi
   - Alineació de l'heroi (bo, dolent o neutral)
   - Botó de favorit per marcar-ho o desmarcar-ho
   - Valoració dels seus poders
   - Desplegables amb informació sobre la seva biografia, aparença física, connexions amb altres personatges i el seu treball



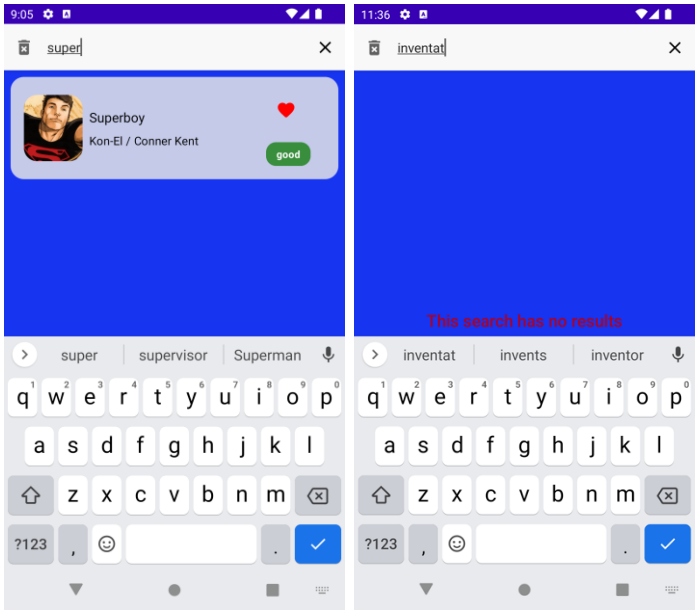
Il·lustració 39: Pantalla de detall de l'heroi

7. Des de la barra inferior de navegació es pot accedir a la pantalla de Favorits on es mostrarà els herois que tenim guardats com a favorits o, si és el cas, un missatge indicant que no tenim cap favorit a la llista.
   Des d'aquesta pantalla es poden desmarcar els herois com a favorits i desapareixen de la llista.
   Si es fa clic sobre la fitxa de qualsevol heroi es mostrarà el seu detall (veure punt 6)



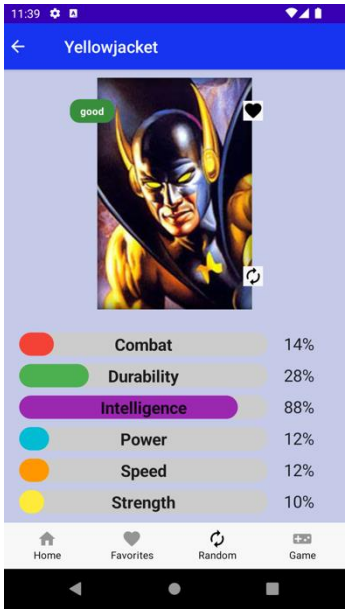
Il·lustració 40: Pantalla Favorits

8. Des de la pantalla Favorits es poden filtrar els herois pel seu nom. Si el nom buscat no existeix s'indica amb un missatge



Il·lustració 41: Búsqueda a la pantalla Favorits

9. Des de la barra inferior de navegació es pot accedir a la pantalla de heroi aleatori. En aquesta pantalla l'aplicació mostra els detalls d'un heroi aleatori.
   Les dades són les mateixes que tenim a la fitxa de detall (veure punt 6) més una icona que ens permet demanar un nou heroi aleatori a l'aplicació.

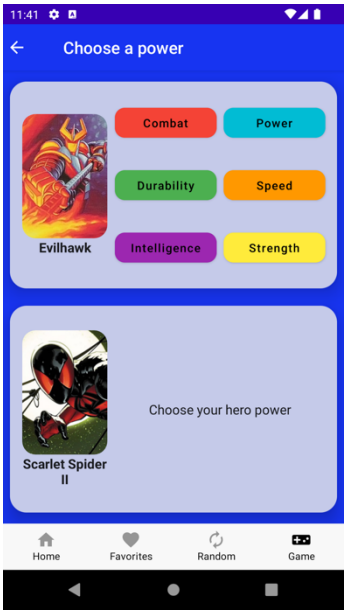


Il·lustració 42: Pantalla d'heroi aleatori

10. Des de la barra inferior de navegació es pot accedir a la pantalla del joc. Al entrar en aquesta pantalla ens trobarem dues cartes i hem d'elegir-na una que serà amb la que jugarem nosaltres.

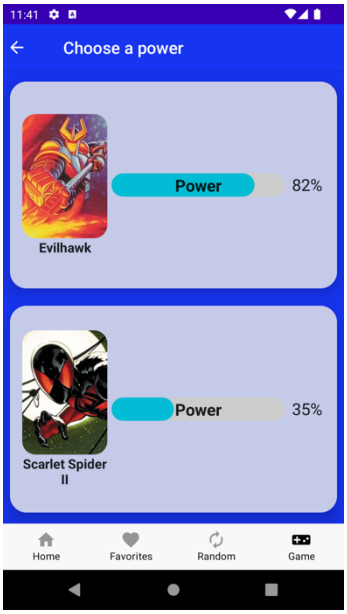
Il·lustració 43: Pantalla d'inici del joc

11. Una vegada elegida la carta, es donaran la volta i ens mostrarà la imatge i el nom dels herois de les dues cartes. A la nostra carta hem d'elegir un superpoder amb el que vulguem apostar contra l'altre heroi.



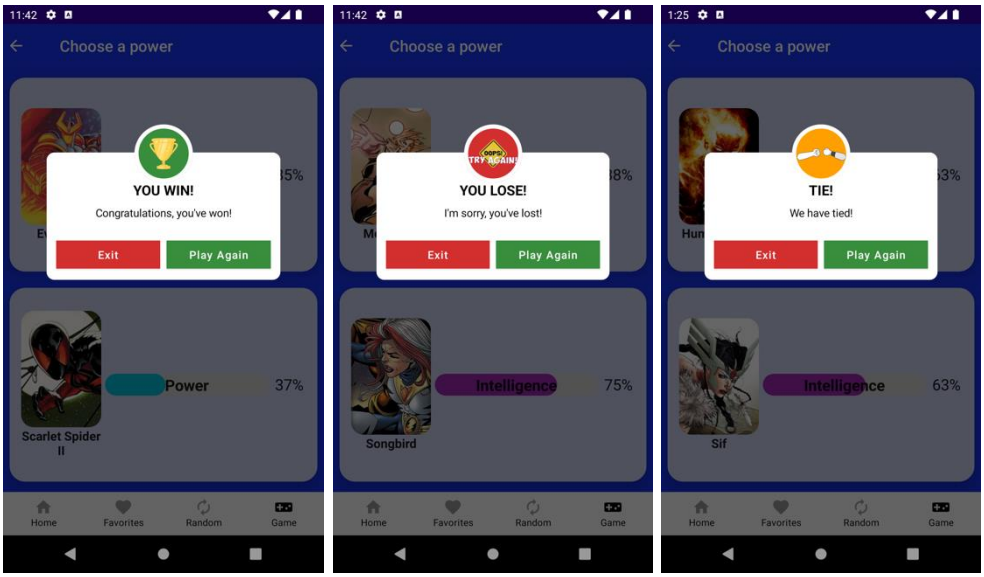
Il·lustració 44: Aposta de superpoder

12. Una vegada elegit el superpoder, hi ha una barra animada que va creixent fins arribar al valor d'aquest superpoder a cada un dels herois.



Il·lustració 45: Resultat de l'aposta

13. Per últim, es mostra el resultat de l'aposta (guanyat, perdut o empatat) i es dona l'opció de tornar a jugar una nova partida o sortir del joc.



Il·lustració 46: Possibles resultats del joc