



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

El Musau El Bouhali, Ahmed; Benítez Fernández, Yolanda, dir. Shop Smartly App: Aplicació per comparació de preus de supermercats amb Web Scraping. 2024. (Grau en Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/298907>

under the terms of the  license

Shop Smartly App: Aplicació per comparació de preus de supermercats amb Web Scraping

Ahmed El Musau El Bouhali

28 de juny de 2024

Resum– Aquest informe detalla el procés de desenvolupament i implementació del meu Treball de Final de Grau (TFG), centrat en la creació d'un servidor i una aplicació mòbil per a la gestió de compres en diferents supermercats. S'ha utilitzat una arquitectura hexagonal per al servidor i el model de vista-vista-model (MVVM) per a l'aplicació, juntament amb tecnologies com Kotlin, Jetpack Compose i Selenium per al web scraping. Després d'una anàlisi exhaustiva dels requisits, s'han superat els reptes i imprevistos amb replanificacions adequades, resultant en una implementació completa i funcional. Aquest projecte no només ha millorat les meves competències tècniques sinó que també ha reforçat la meva capacitat per gestionar projectes complexos amb èxit.

Paraules clau– Desenvolupament de Servidor, Aplicació Mòbil, Gestió de Compra d'Aliments, Arquitectura Hexagonal, MVVM (Model-Vista-ViewModel), Bones Pràctiques de Programació, Patrons de Disseny, Kotlin, Jetpack Compose, Selenium, Web Scraping, Anàlisi de Requisits, Replanificació

Abstract– This report details the development and implementation process of my Bachelor's Thesis, focused on creating a server and mobile application for grocery shopping management across different supermarkets. The server employs a hexagonal architecture, while the mobile app follows the Model-View-ViewModel (MVVM) pattern, using technologies such as Kotlin, Jetpack Compose, and Selenium for web scraping. Through thorough requirements analysis and effective replanning to address challenges and unforeseen issues, the project culminated in a fully functional implementation. This endeavor has not only enhanced my technical skills but also reinforced my ability to successfully manage complex projects.

Keywords– Development of Server, Mobile Application, Food Purchase Management, Hexagonal Architecture, MVVM (Model-View-ViewModel), Best Programming Practices, Design Patterns, Kotlin, Jetpack Compose, Selenium, Web Scraping, Requirements Analysis, Replanning

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EN aquest Treball de Fi de Grau (TFG), s'ha desenvolupat una aplicació Android anomenada ShopSmartly. Aquesta aplicació sorgeix com a resposta a l'actual situació d'inflació, on els preus dels productes de

primera necessitat han augmentat significativament mentre els salaris s'han mantingut estables. Aquesta situació ha creat una necessitat creixent entre les famílies per trobar opcions de compra més econòmiques i accessibles, amb l'objectiu de gestionar millor el seu pressupost mensual.

ShopSmartly està dissenyada per ajudar els usuaris a comparar preus de productes en diversos supermercats, permetent-los trobar les millors ofertes de manera ràpida i eficient. L'aplicació permet als usuaris introduir la seva llista de la compra o seleccionar productes específics, i proporciona informació en temps real sobre els millors preus disponibles. Això permet als usuaris prendre decisions informades i maximitzar el seu poder adquisitiu, fent que les

- E-mail de contacte: 1601230@uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Yolanda Benítez Fernández (Departament de Ciències de la Computació)
- Curs 2023/24

seves compres siguin més intel·ligents i econòmiques.

Per obtenir les dades dels preus dels productes, ShopSmartly utilitza una tècnica anomenada web scraping [2] que recull informació de diverses fonts en línia. Aquesta informació és processada i emmagatzemada en un servidor desenvolupat específicament per a aquest projecte. El servidor s'encarrega de gestionar les dades, actualitzar els preus i proporcionar les dades necessàries a l'aplicació en temps real.

Aquest informe detallarà l'estat de l'art de l'aplicació, els objectius del TFG, la metodologia aplicada, la planificació seguida, el disseny de l'aplicació i els requisits, el desenvolupament realitzat, els resultats obtinguts, els tests efectuats i les conclusions finals. També s'inclouran annexos amb informació rellevant del projecte per il·lustrar millor el treball realitzat.

2 ESTAT DE L'ART

Abans de començar el desenvolupament de l'aplicació "ShopSmartly", es va dur a terme un estudi de l'estat de l'art per investigar l'existència d'altres aplicacions amb un propòsit similar al d'aquest TFG. Durant aquesta investigació, es va trobar un article [8] que abordava diverses aplicacions que ofereixen funcionalitats relacionades amb l'optimització del preu de la compra, com "Soysuper", "Onprice", "OCU Market", etc. Totes aquestes aplicacions comparteixen un objectiu semblant: facilitar als usuaris la comparació de preus en diferents establiments i optimitzar les seves despeses.

Les funcionalitats comunes trobades en aquestes aplicacions inclouen la possibilitat de buscar productes específics o llistes de la compra i la capacitat d'escanejar codis de barres per obtenir informació de preus en temps real. Tot i això, "ShopSmartly" va buscar diferenciar-se principalment per la seva interfície moderna i altament intuïtiva. L'objectiu era crear una experiència d'usuari atractiva i fàcil d'utilitzar, en contrast amb les interfícies més antiquades de les altres aplicacions investigades.

A més, amb "ShopSmartly" es va voler ampliar les funcionalitats que ofereixen les aplicacions similars més enllà de la simple comparació de preus, tenint en compte l'aportació que aquestes noves funcionalitats podrien tenir per a la societat. Per exemple, es va estudiar la possibilitat d'afegir la cerca de productes segons la seva qualitat, origen, marca, entre altres aspectes rellevants. No obstant això, la inclusió d'aquestes funcionalitats addicionals es va decidir en funció de diversos factors, com la seva complexitat tècnica i la seva utilitat per als usuaris.

3 OBJECTIUS

Els objectius del TFG es van classificar en dos tipus segons la seva importància per assegurar un desenvolupament eficient de l'aplicació "ShopSmartly". Aquesta classificació va permetre prioritzar les tasques en cas de no tenir temps suficient per completar-los tots. Tot i això, s'han aconseguit tots els objectius plantejats:

3.1 Objectius principals

- **Optimitzar la recerca de productes:** Permetre als usuaris buscar productes específics i visualitzar les opcions disponibles en diferents supermercats per trobar la millor relació qualitat-preu.
- **Facilitar la comparació de preus:** Proporcionar una manera fàcil i intuïtiva de comparar preus en diversos supermercats, oferint una visió clara de les opcions més econòmiques.
- **Millorar l'experiència d'usuari:** Simplificar el procés de cerca d'ofertes amb una interfície amigable i eficient, eliminant la necessitat de cerques manuals.
- **Fomentar la presa de decisions informades:** Capacitar els usuaris per prendre decisions informades oferint informació detallada sobre els productes i les ofertes disponibles, així com els preus en diferents supermercats.

3.2 Objectius secundaris

- **Personalitzar les preferències de compra:** Permetre als usuaris seleccionar marques i supermercats preferits, millorant així la seva experiència de compra.
- **Gestionar cistelles de compra:** Donar als usuaris la possibilitat de crear i gestionar cistelles de compra amb una llista de productes desitjats, mostrant el cost total de la cistella i permetent guardar-la per a consultes futures.
- **Reportar incidències:** Oferir als usuaris l'opció de posar incidències en cas que detectin anomalies en la informació proporcionada per l'aplicació.
- **Gestió d'incidències per administradors:** Crear un usuari especial d'administrador que pugui veure totes les incidències, gestionar-les i indicar l'evolució de les mateixes a través de l'aplicació.

4 METODOLOGIA

Per al desenvolupament de l'aplicació Android "ShopSmartly", s'ha optat per la metodologia en cascada amb iteracions. Aquesta metodologia combina els principis de la cascada tradicional amb la flexibilitat addicional de les iteracions, permetent una adaptació contínua als canvis durant el procés de desenvolupament.

La metodologia en cascada es compon de les etapes seqüencials d'anàlisi, disseny, implementació, testeig i manteniment. En aquest projecte, s'han incorporat iteracions després de la fase de testeig, permetent una revisió i ajust continu de l'aplicació en funció dels nous requeriments i descobriments.

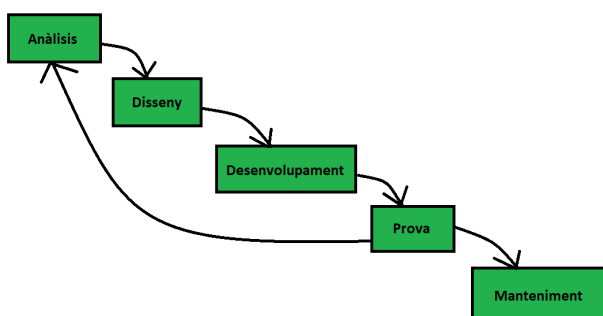


Fig. 1: Metodologia en cascada amb iteracions

Aquesta metodologia es va triar per diverses raons:

- **Familiaritat i adaptabilitat:** La cascada amb iteracions és àmpliament coneguda i permet una adaptació flexible a canvis en els requeriments del projecte.
- **Claredat i estructura:** Proporciona una guia clara i ordenada per al desenvolupament, assegurant que cada pas es completi de manera metòdica i evitant errors o omissions.
- **Adequació al projecte:** En un projecte ben definit com "ShopSmartly", aquesta metodologia s'adapta millor que metodologies àgils, les quals són més adequades per a projectes grans i amb requeriments menys definits.

Durant el desenvolupament, es van completar diverses iteracions del procés. Cada iteració va incloure l'anàlisi, el disseny, la implementació i el testeig, permetent ajustar i millorar l'aplicació segons les necessitats detectades. Aquest enfocament iteratiu va garantir una millora contínua i una adaptació flexible als nous descobriments, assegurant l'alineació constant amb els objectius del projecte.

5 PLANIFICACIÓ

La planificació del projecte "ShopSmartly" es va estructurar en diverses iteracions, cadascuna enfocada a aspectes clau del desenvolupament. Aquesta metodologia iterativa va permetre ajustar el procés de treball segons les necessitats i els problemes trobats durant el projecte. A continuació, es detallen les iteracions principals:

- **Primera iteració (15 de febrer - 31 de març):** En aquesta primera iteració es va realitzar l'anàlisi de requisits, el disseny de diagrames de classes, el model relacional de la base de dades i els casos d'ús. També es va implementar el servidor i es va testejar amb proves unitàries i d'integració.
- **Segona iteració (1 d'abril - 16 de maig):** En aquesta iteració es van revisar i ajustar els diagrames i els requisits. A més, es va desenvolupar l'aplicació i es va testejar amb proves unitàries i d'integració.
- **Tercera iteració (17 de maig - 16 de juny):** En la tercera iteració es van revisar els diagrames i l'anàlisi per verificar que tot estigués implementat correctament i per analitzar si hi havia noves funcionalitats a afegir.

També es va realitzar un testeig global del sistema, es van corregir els errors detectats i es van fer canvis en la interfície de l'aplicació per millorar l'experiència d'usuari a partir dels resultats del test d'usuari.

- **Quarta iteració (17 de juny - 30 de juny):** En aquest període es va fer manteniment del servidor i de l'aplicació a mesura que es detectaven detalls millorables. També es va preparar l'informe final, es va dissenyar el cartell del TFG i es va preparar la presentació.
- **Cinquena iteració (1 de juliol - 5 de juliol):** Es va practicar la presentació per assegurar una exposició efectiva i ben polida.

Aquest enfocament iteratiu va permetre una adaptació contínua a les necessitats del projecte i va assegurar que tots els objectius es complissin de manera eficient.

6 FASE D'ANÀLISIS

La fase d'anàlisi del projecte ShopSmartly ha implicat una sèrie d'etapes clau per definir els requisits del sistema i el seu disseny. En cada iteració, s'ha realitzat un aprofundit anàlisi per garantir que tots els aspectes del projecte estiguin ben definits i entesos abans de passar a la fase de desenvolupament.

Durant la primera iteració, es va dur a terme una anàlisi exhaustiva dels requisits del sistema, tant funcionals com no funcionals. Els requisits funcionals defineixen les funcionalitats que l'aplicació ha de proporcionar als usuaris, com ara la identificació d'usuaris, la gestió de perfils, la creació de llistes de compra, la cerca de productes, i la gestió d'incidències, entre altres (vegeu 6.0.1). Els requisits no funcionals, per altra banda, defineixen les característiques de qualitat que el sistema ha de complir, com ara la usabilitat, el rendiment, la seguretat i la compatibilitat amb diverses plataformes i tecnologies (vegeu 6.0.2). A continuació estan els requisits funcionals i no funcionals del projecte:

6.0.1 Requisits Funcionals

ID	DESCRIPCIÓ
RF - 1	Identificació d'usuari: Per poder utilitzar ShopSmartly, l'usuari s'ha d'identificar amb el seu compte. En cas de no tenir-ne un, s'ha de poder crear un compte nou.
RF - 2	Tipus d'Usuaris: L'aplicació ha de permetre l'existència de dos tipus d'usuaris: usuari administrador i usuaris normals. Les funcionalitats accessibles depenen del tipus d'usuari; per defecte, tot compte creat és considerat un compte d'usuari normal.
RF - 3	Consultar informació l'usuari: ShopSmartly ha de disposar d'una secció de perfil de l'usuari pels dos tipus d'usuaris perquè puguin consultar la seva informació i modificar-la. La informació que ha de tenir la pantalla de perfil d'usuari ha de ser: nom d'usuari, correu electrònic i idioma preferit.

RF – 4	Modificació de contrasenya i eliminació del compte d'usuari: L'aplicació ha de permetre als usuaris modificar la seva contrasenya i eliminar el seu compte des del mateix perfil d'usuari.
RF – 5	Selecció i modificació de l'Idioma de l'aplicació: ShopSmartly ha de ser multilingüe, permetent a qualsevol usuari canviar l'idioma entre català, castellà i anglès des de la pantalla d'inici de sessió com també des de la pantalla de perfil d'usuari. Per defecte, l'idioma de l'aplicació serà l'anglès.
RF – 6	Buscar un producte en diferents supermercats: El sistema ha de permetre als usuaris normals cercar un producte específic per consultar les diferents opcions que ofereixen els supermercats per aquell producte i a quin preu.
RF – 7	Crear cistelles de compres: L'aplicació ha de permetre als usuaris normals crear una llista o cistella de compra i mostrar-los en quin supermercat poden adquirir cada producte i a quin preu perquè puguin escollir l'opció més econòmica que desitgin.
RF – 8	Modificar la llista de compra: El sistema ha de permetre als usuaris normals modificar la llista de compra prèviament definida.
RF – 9	Filtrar per marques de productes: S'ha de proporcionar la funcionalitat per filtrar els resultats de la cerca segons les marques de productes d'interès de l'usuari normal, permetent només considerar les opcions de les marques desitjades.
RF – 10	Filtrar per supermercats: L'aplicació ha de permetre als usuaris normals filtrar els resultats de la cerca segons els supermercats desitjats, mostrant només les opcions disponibles en els establiments seleccionats.
RF – 11	Buscar productes manualment: Els usuaris normals han de poder buscar o afegir a la llista de compres un producte manualment, ja sigui cercant-lo per nom o introduint el seu codi de barres.
RF – 12	Escanejar codi de barres: Els usuaris normals han de poder buscar o afegir a la llista de compres un producte escanejant el seu codi de barres amb la càmera del dispositiu.
RF – 13	Navegació per categories de productes: L'aplicació ha de facilitar la navegació dels usuaris normals per les diferents categories de productes, permetent-los trobar opcions específiques navegant per les diferents seccions.
RF – 14	Reportar incidències en l'aplicació: ShopSmartly ha de permetre als usuaris normals poder reportar incidències relacionades amb l'aplicació (productes inexistents o error en la informació d'algun producte).
RF – 15	Gestió d'Incidències: Els administradors han de tenir la capacitat de rebre i gestionar les incidències enviades pels usuaris normals. Quan un usuari normal reporti una incidència, aquesta s'enviarà automàticament a l'administrador per al seu tractament i resolució.

RF – 16	Gestió de l'estat de les incidències: Les incidències creades pels usuaris normals han d'estar, per defecte, en estat "pendent d'atenció". Els administradors han de tenir la capacitat de canviar l'estat d'una incidència a "finalitzada" o "en progrés". Els administradors també han de poder afegir comentaris relacionats amb les incidències, els quals seran visibles només per ells mateixos. Això permetrà una millor comunicació interna i seguiment de les incidències.
---------	---

6.0.2 Requisits no Funcionals

ID	DESCRIPCIÓ
RNF – 1	L'aplicació ha de ser intuïtiva: L'aplicació ha de tenir una interfície d'usuari clara i intuïtiva per a una fàcil navegació i utilització, garantint una experiència d'usuari satisfactòria.
RNF – 2	L'aplicació es desenvoluparà per a dispositius Android: L'aplicació ha de ser desenvolupada per a dispositius Android, garantint la compatibilitat amb aquest sistema operatiu.
RNF – 3	Utilitzar WebScraping per obtenir informació dels productes: S'ha d'implementar un programa en Python per obtenir informació mitjançant WebScraping i actualitzar de manera regular la base de dades de productes.
RNF – 4	Utilitzar el MongoDB Atlas com a base de dades central: S'ha d'utilitzar el MongoDB Atlas com a base de dades central per emmagatzemar tota la informació rellevant, inclosos perfils d'usuari i detalls dels productes.
RNF – 5	Utilitzar el Realm com a base de dades local de l'aplicació: L'aplicació ha de fer servir Realm com a base de dades local per garantir la disponibilitat d'informació en cas de connectivitat limitada, sempre sincronitzant-se amb MongoDB Atlas per obtenir les dades més recents.
RNF – 6	Encriptar les contrasenyes fent servir el hash SHA-512: Les contrasenyes dels usuaris s'han d'emmagatzemar en el MongoDB Atlas encriptades amb la funció hash SHA-512 per a garantir la seguretat de les dades personals dels usuaris.
RNF – 7	La interfície de l'aplicació ha d'estar disponible en mode clar i fosc: L'aplicació ha de disposar de modes clar i fosc per a la interfície, oferint una experiència d'usuari personalitzada i adaptable a les preferències visuals de l'usuari.

RNF – 8	Utilitzar en el desenvolupament de l'aplicació el patró de disseny MVVM i l'arquitectura hexagonal: S'ha d'aplicar el patró de disseny Model-View-ViewModel (MVVM) i l'arquitectura hexagonal en el desenvolupament de l'aplicació per a una millor organització del codi, facilitant el manteniment i l'escalabilitat del sistema.
RNF – 9	Internacionalització de l'aplicació: L'aplicació ha d'estar implementada amb la capacitat de ser internacionalitzada des del principi, poden anar afegint diferents idiomes segons la zona geogràfica dels usuaris més comuns.

A partir d'aquests requeriments, es van elaborar el diagrama de casos d'ús [3] de l'aplicació, el diagrama de classes del servidor [4] i el disseny de la base de dades [5]. Aquestes eines van permetre una comprensió clara i detallada de les funcionalitats del sistema i la seva estructura interna.

En la segona iteració, es va revisar i millorar l'anàlisi realitzat a la primera iteració. Es van fer ajustos als diagrames i als requisits funcionals basant-se en les observacions i les necessitats detectades durant el desenvolupament. A més, es va dissenyar el prototip de l'aplicació per a l'usuari normal/comprador [6] i per a l'usuari administrador [7], identificant les pantalles i les interaccions previstes per als usuaris.

Gràcies a aquestes anàlisis, s'ha obtingut una llista completa de requisits funcionals i no funcionals, així com els diagrames corresponents i el prototip de l'aplicació. Aquesta fase ha estat fonamental per establir les bases del projecte i garantir una implementació exitosa del sistema.

7 FASE DE DESENVOLUPAMENT

La fase de desenvolupament del projecte ShopSmartly es distingeix per incloure una part tant de front-end com de back-end, essent la part de desenvolupament dividida entre la creació del servidor, encarregat de realitzar el scraping de dades, i el desenvolupament de l'aplicació. Aquesta divisió ha permès abordar de manera eficient les diferents necessitats tècniques i funcionals del projecte.

7.1 Desenvolupament del servidor

En primer lloc, m'agradaria deixar clar la utilitat o funcionalitat del servidor. El servidor en aquest context no actua com un element que rep peticions externes, sinó com un programa que, en el moment de l'execució, inicia el scraping de productes des de les pàgines web dels supermercats. Utilitzant la tecnologia Selenium, simula la interacció humana amb les pàgines per evitar la detecció de les seves activitats com a bots per part dels llocs web. Durant el procés, recupera la informació dels productes, la qual emmagatzema de manera provisional. Posteriorment, aquestes dades es mapegen i es guarden en una base de dades remota, específicament en MongoDB Atlas.

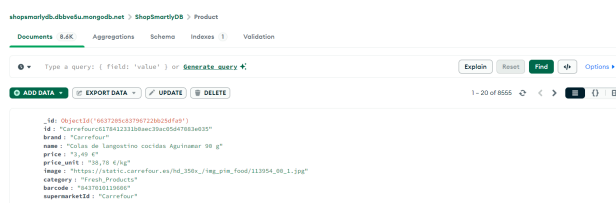


Fig. 2: En la imatge es pot veure com la base de dades de ShopSmartly en Mongo DB Atlas està inicialitzada amb un total de 8555 productes obtinguts durant el scraping fet pel servidor.

Més enllà de la implementació bàsica del scraping i l'emmagatzematge de dades, el desenvolupament del servidor inclou l'aplicació de diversos patrons de disseny per millorar la seva mantenibilitat i flexibilitat.

7.1.1 Patró Repositori

El Patró Repositori s'ha utilitzat per separar la lògica d'accés a les dades de la lògica de negoci. S'han creat interfícies per a cada repositori, que defineixen les operacions disponibles per als diferents tipus de dades (productes, supermercats, etc.). Aquestes interfícies tenen implementacions específiques per MongoDB, facilitant la seva reutilització i manteniment. Per exemple, el Repositori de Productes proporciona mètodes per interactuar amb les dades dels productes, i té una implementació específica per MongoDB Atlas:

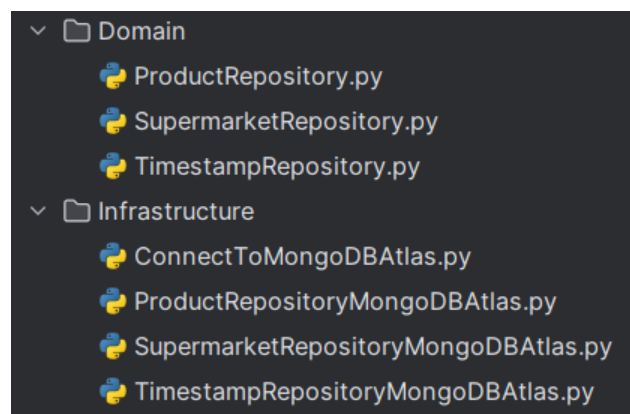


Fig. 3: En la carpeta domini es veu les interfícies dels repositoris que s'utilitzen en el servidor i en la carpeta infraestructura es pot veure les seves implementacions.

7.1.2 Patró Mapper

El Patró Mapper s'ha utilitzat per convertir les dades entre diferents formats. En aquest projecte, s'ha definit una interfície de Mapper que converteix les dades en format JSON al format d'objectes de domini, com ara Producte i Supermercat. Aquesta interfície té implementacions específiques per al *scraping*, permetent una major flexibilitat i independència en el procés de mapeig de dades en cas que en un futur s'opti per utilitzar una altra eina o tècnica, ja que la resta del codi no depèn exclusivament del scraping.

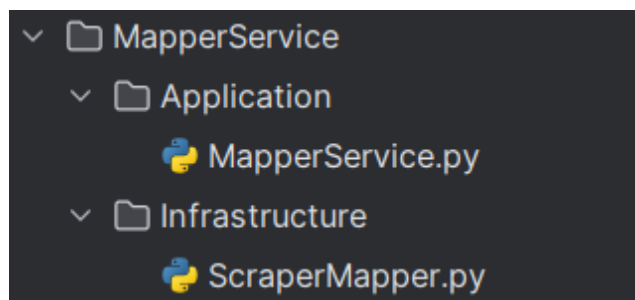


Fig. 4: En la carpeta aplicació es veu la interfície dels mapper que s'utilitza en el servidor i en la carpeta infraestructura es pot veure la seva implementació pel cas de dades obtingudes amb web scraping.

7.1.3 Arquitectura Hexagonal

L'altre patró aplicat és un patró d'arquitectura, el patró d'Arquitectura Hexagonal, també conegut com patró de Ports i Adaptadors. Aquest patró és una estructura de disseny de programari que busca organitzar el codi de manera modular i independent de la tecnologia utilitzada. Es divideix en diverses capes: la capa de Domini, on es troba la lògica de negoci o regles de negoci de l'aplicació; la capa d'Aplicació, on es gestiona la lògica relacionada amb casos d'ús específics i serveis de l'aplicació; i la capa d'Infraestructura, que conté tota la lògica amb dependències externes, com ara accés a bases de dades, serveis externs o altres sistemes.

Un exemple concret de com s'aplica aquesta arquitectura al desenvolupament és la manera de com s'organitza la lògica de la base de dades. En la capa de Domini, s'han definit interfícies per als repositoris que gestionen l'accés a les dades dels productes, supermercats i timestamps. A la capa d'Aplicació, es troba la implementació dels casos d'ús com la recuperació de dades i el servei de base de dades, que executa l'acció de scraping i guarda les dades a la base de dades. Finalment, en la capa d'Infraestructura, es troben les implementacions concretes dels repositoris que s'encarreguen de connectar-se a MongoDB Atlas per gestionar la persistència de les dades. Aquesta estructura permet una clara separació de responsabilitats i facilita el manteniment i la modificació del codi al llarg del temps.

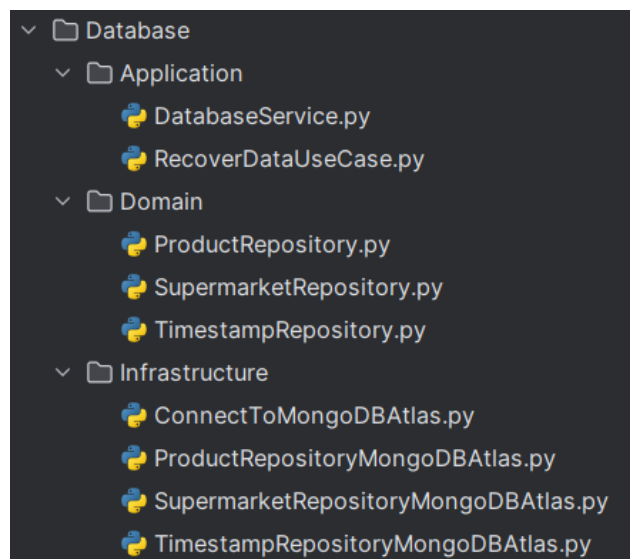


Fig. 5: En la imatge es pot veure com està estructurat la lògica relacionada amb la base de dades segons l'arquitectura hexagonal.

API de MongoDB Atlas: A més del servidor de *scraping*, s'ha implementat també una API remota amb diversos Endpoints per MongoDB Atlas per permetre que l'aplicació es comuniqui amb la base de dades mencionada. Aquesta API ofereix diversos Endpoints, accessibles a través del protocol HTTPS, que permeten que l'aplicació faci peticions específiques a la base de dades, com ara obtenir, afegir, eliminar o actualitzar dades. Aquesta capa intermèdia és crucial per raons de seguretat, ja que Android requereix que les aplicacions es comuniquin amb bases de dades remotes a través d'APIs en lloc de fer-ho directament.

Endpoint Route	HTTP Method	Linked Function	Status	Last Modified	Actions
/deleteCart	GET	removeCart	Enabled	04/03/2024 20:53:12 UTC	[icon]
/deleteUserAccount	GET	removeUser	Enabled	04/21/2024 21:02:33 UTC	[icon]
/getIncidences	GET	fetchIncidences	Enabled	04/24/2024 19:58:20 UTC	[icon]
/getProducts	GET	fetchProducts	Enabled	04/23/2024 21:00:20 UTC	[icon]
/getProductCarts	GET	fetchProductCarts	Enabled	04/24/2024 19:59:14 UTC	[icon]
/getSupermarkets	GET	fetchSupermarkets	Enabled	04/24/2024 19:48:38 UTC	[icon]
/getTimestamp	GET	fetchTimestamp	Enabled	04/23/2024 21:00:15 UTC	[icon]
/searchEmail	GET	searchEmail	Enabled	04/10/2024 18:50:07 UTC	[icon]
/changeUserPassword	POST	changeUserPassword	Enabled	04/16/2024 19:54:14 UTC	[icon]

Fig. 6: En la imatge es pot veure l'API amb els diferents Endpoints que ofereix.

7.2 Desenvolupament de l'aplicació

L'aplicació Android de ShopSmartly ha estat dissenyada per proporcionar als usuaris una plataforma on puguin visualitzar i comparar productes disponibles en diferents supermercats. Aquesta aplicació permet buscar productes per nom, categoria, marca o supermercat, i afegir-los a una cistella de la compra per veure el preu total. També es poden guardar cistelles per seguir l'evolució dels preus al llarg del temps. A més de les funcionalitats bàsiques, s'inclouen opcions com el registre i inici de sessió, la modificació del perfil d'usuari, i la diferenciació entre usuaris administradors i usuaris normals, amb funcionalitats específiques per

a cada tipus d'usuari.

Inicialment, es va considerar l'ús de XML i Java per al desenvolupament d'aplicacions Android, donat el coneixement previ en aquestes tecnologies. No obstant això, després d'investigar sobre les últimes tendències, es va optar per utilitzar Jetpack Compose i Kotlin per aprofitar els avantatges i funcionalitats més recents. Aquest canvi de tecnologia, tot i que va requerir una corba d'aprenentatge significativa, ha proporcionat beneficis a llarg termini.

Pel que fa a l'organització del projecte, s'ha seguit una estructura coherent amb les millors pràctiques de desenvolupament de programari. S'han utilitzat patrons de disseny com el Patró de Repositori i el MVVM (Model-View-ViewModel), amb algunes incursions en l'arquitectura hexagonal per assegurar una organització clara i modular del codi. Aquesta estructura ha resultat en un codi més escalable i mantenible.

L'aplicació diferencia entre dos modes d'ús: mode usuari normal i mode administrador. A continuació, es detallen les funcionalitats implementades per a cada mode:

Funcionalitats

- **Mode Usuari Normal:** Aquest mode està dissenyat per ajudar els usuaris a seleccionar la cistella de la compra més econòmica. Els usuaris poden visualitzar tots els supermercats disponibles i els seus productes, així com consultar detalls de productes i visualitzar productes similars. També poden reportar incidències si detecten que un producte està incorrecte o que un producte desitjat no existeix. La cerca de productes es pot fer aplicant filtres com el més econòmic, per marca, per supermercat, etc. A més, els usuaris poden crear i gestionar cistelles de compra, modificar-les un cop creades, i ajustar les configuracions de la seva aplicació i dades personals des de la secció de perfil.
- **Mode Administrador:** Aquest mode està destinat a persones autoritzades que gestionen l'aplicació. Els administradors poden veure totes les incidències reportades pels usuaris en temps real, fer-ne seguiment, assignar-les i posar observacions explicatives sobre l'estat de la incidència. El perfil d'usuari en aquest mode és el mateix que en el mode normal, amb l'addició de la pantalla de gestió d'incidències.

Patrons de Disseny Aplicats: S'han aplicat diversos patrons de disseny per millorar la qualitat i mantenibilitat del codi com ara el patró de repositori. També s'han aplicat principis de la Programació Orientada a Objectes (OOP) i de la Programació Funcional, aprofitant les característiques de Kotlin.

Internacionalització: L'aplicació ha estat internacionalitzada per donar suport a diversos idiomes (català, castellà i anglès), ampliant així la seva accessibilitat i usabilitat per a diferents grups d'usuaris.

Interfície d'Usuari: S'ha dissenyat una interfície d'usuari moderna i intuïtiva utilitzant Jetpack Compose, que facilita la creació d'interfícies complexes amb menys codi i una millor gestió de l'estat. L'aplicació ofereix una experiència

fluida tant en mode clar com en mode fosc, millorant l'experiència d'usuari en diferents condicions de llum.

En resum, la fase de desenvolupament ha implicat un treball exhaustiu tant en la part de servidor com en la part d'aplicació. S'han aplicat múltiples tecnologies i patrons de disseny per assegurar la qualitat, mantenibilitat i escalabilitat del projecte. El resultat és una aplicació completa i funcional que satisfà els objectius plantejats al començament del projecte.

8 FASE DE TEST

Durant la fase de test, s'han emprat diverses estratègies per assegurar la qualitat i la fiabilitat de l'aplicació desenvolupada. A continuació, es detallen els tipus de tests realitzats i com han contribuït a garantir un producte final òptim.

8.1 Test de Codi

Els tests de codi són crucials per assegurar el bon funcionament de l'aplicació en nivells micro i macro. S'han realitzat dos tipus principals de tests: els tests unitaris i els tests d'integració. Els tests unitaris s'han enfocat a verificar el comportament de les funcionalitats de manera individual, assegurant-se que cadascuna actua de manera correcta segons les expectatives. Per contra, els tests d'integració han examinat el flux d'execució global de l'aplicació, garantint que totes les parts es comuniquen i funcionen de manera sincronitzada i eficient. Aquesta metodologia de tests ha permès detectar i corregir errors en el codi, assegurant una experiència d'usuari lliure d'incidències durant l'execució.

8.2 Test Plan

El test plan (vegeu [A.0.1](#)) s'ha centrat en validar que l'aplicació respon correctament a les diferents accions de l'usuari i escenaris possibles. A través d'un conjunt de casos d'ús definits en un document, s'han provat les diverses funcionalitats i s'ha comparat la resposta obtinguda amb l'esperada. Aquest enfocament ha permès verificar que l'aplicació mostra la informació i les funcionalitats adequades en cada situació, garantint una experiència coherent i satisfactòria per als usuaris.

8.3 Test d'usuari

El test d'usuari ha estat essencial per avaluar l'experiència d'usuari i la usabilitat de l'aplicació. Mitjançant la realització de tasques específiques assignades a diferents participants, s'ha avaluat la intuïtivitat de l'aplicació i la facilitat d'ús. Aquesta metodologia ha permès identificar àrees de millora en la interfície de l'aplicació, com ara canvis en la disposició dels botons o la inclusió de nous elements per millorar la navegació i l'accessibilitat. A més, el feedback dels usuaris ha ajudat a adaptar l'aplicació per reduir la corba d'aprenentatge inicial i millorar la satisfacció general de l'experiència d'usuari.

9 RESULTATS

Després de completar el treball de fi de grau, que ha abastat aproximadament 3 mesos i ha requerit un total de 300 hores,

s'ha assolit un conjunt de resultats significatius. En primer lloc, s'ha desenvolupat un servidor dedicat que realitza el scraping de dades de productes de diversos supermercats, incloent Carrefour, Dia i Alcampo. Aquestes dades s'han mapejat i emmagatzemat en MongoDB Atlas, formant una base de dades completa i actualitzada. A més, s'ha implementat una API de MongoDB Atlas en el núvol que ofereix totes les funcionalitats necessàries per a l'aplicació, incloent la gestió de productes, supermercats, cistelles de compra i incidències.

Per complementar la part del servidor, s'ha desenvolupat una aplicació Android completa que ofereix una experiència d'usuari rica i versàtil. Aquesta aplicació ofereix dos modes d'ús: mode usuari normal i mode administrador. El mode usuari permet als usuaris consultar els preus de diversos productes, crear i gestionar cistelles de compra per trobar les opcions més econòmiques, i seguir l'evolució dels preus de les seves compres al llarg del temps. D'altra banda, el mode administrador proporciona funcionalitats addicionals, com ara la gestió d'incidències reportades pels usuaris.

A continuació, es presenten captures de pantalla de les diverses pantalles desenvolupades, que il·lustren les funcionalitats i l'aspecte de l'aplicació implementada.

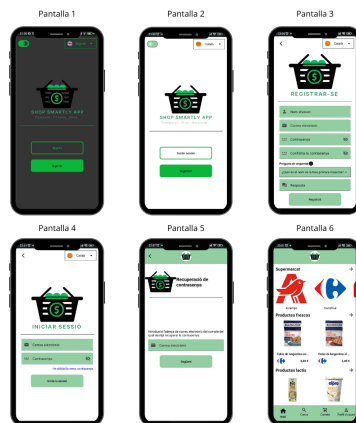


Fig. 7: Pantalles de l'aplicació Shop Smartly App (1)

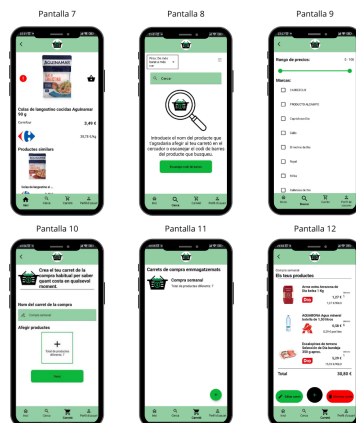


Fig. 8: Pantalles de l'aplicació Shop Smartly App (2)



Fig. 9: Pantalles de l'aplicació Shop Smartly App (3)

1. **Pantalla 1:** Pantalla d'accés a l'aplicació en mode fosc, on es mostra els botons d'iniciar sessió i registrar-se per a l'aplicació.
2. **Pantalla 2:** Pantalla d'accés a l'aplicació en mode clar, on es mostra els botons d'iniciar sessió i registrar-se per a l'aplicació.
3. **Pantalla 3:** Pantalla de registre, on els usuaris poden introduir les seves dades per crear un nou compte a l'aplicació.
4. **Pantalla 4:** Pantalla d'inici de sessió, on els usuaris poden iniciar sessió amb les seves credencials existents.
5. **Pantalla 5:** Pantalla d'oblidar contrasenya, on en cas de pèrdua de contrasenya, aquesta pantalla permet als usuaris restablir-la mitjançant un correu electrònic de restabliment.
6. **Pantalla 6:** Pantalla home de l'usuari normal, on es mostren els supermercats i els productes classificats per categories per facilitar la navegació als usuaris normals.
7. **Pantalla 7:** Pantalla de detall del producte, en aquesta pantalla es mostra la informació detallada d'un producte específic seleccionat per l'usuari normal.
8. **Pantalla 8:** Pantalla del buscador on els usuaris poden cercar productes introduint el nom o escanejant el codi de barres amb la càmera del dispositiu.
9. **Pantalla 9:** Pantalla de filtratge de resultats, en aquesta pantalla els usuaris poden filtrar els resultats de la cerca de productes segons preu, marca i supermercat.
10. **Pantalla 10:** Pantalla per crear la cistella, en aquesta pantalla els usuaris poden afegir productes a la seva cistella de compra.
11. **Pantalla 11:** Pantalla de cistelles de compra, on els usuaris poden visualitzar les cistelles de la compra que han creat i gestionar-ne el contingut.
12. **Pantalla 12:** Pantalla de detall d'una cistella on es mostra la informació detallada d'una cistella de compra específica, inclosos els productes continguts.

13. **Pantalla 13:** Pantalla d'incidències, on els administradors poden reportar problemes o errors relacionats amb l'aplicació per a una millor resolució i seguiment per part dels administradors.
14. **Pantalla 14:** Pantalla de perfil d'usuari, on els usuaris poden veure i modificar la seva informació personal, com ara el nom, la contrasenya i la configuració del compte.

10 LÍNIES OBERTES

Després d'aconseguir completar amb èxit tots els objectius inicials establerts per al projecte, hi ha diverses àrees que es poden explorar i desenvolupar en un futur proper. A continuació, s'enumeren les línies obertes i les tasques que es preveuen realitzar:

10.1 Desplegament del servidor en el núvol

Actualment, el servidor està implementat i funciona correctament en un entorn local. No obstant això, a causa de les limitacions del portàtil que s'ha estat fent servir durant aquest TFG i la manca de resposta per part de les autoritats universitàries per a allotjar-ho, no ha estat possible desplegar-lo en un entorn de núvol. El pròxim pas és migrar el servidor a una infraestructura de núvol, que permeti l'execució contínua i l'actualització de les dades en temps real, sense les restriccions actuals de disponibilitat física.

10.2 Ampliació del scraping per incloure més supermercats

Actualment, el servidor recull dades de productes només de Carrefour, Alcampo i Dia, utilitzant tècniques de scraping per obtenir aquesta informació disponible públicament. Amb l'objectiu d'enriquir i diversificar la base de dades de productes, s'explorà la possibilitat d'integrar informació de més supermercats. A més, es considerarà l'ús d'APIs de supermercats que ofereixin dades de manera estructurada, així com continuar utilitzant el scraping quan sigui necessari. Aquesta integració múltiple no només ampliarà les opcions per als usuaris finals, sinó que també mantindrà la coherència i eficiència del servidor gràcies a les bones pràctiques de programació implementades.

10.3 Millora del disseny de l'aplicació

Tot i que l'experiència d'usuari actual és excel·lent, gràcies a la realització de test d'usuari per obtenir opinions i millores funcionals, reconec que el disseny visual pot ser refinat per millorar encara més l'experiència global de l'usuari. L'aplicació actual té un disseny minimalista funcional, però es buscarà millorar la seva estètica i presentació per crear una interfície més atractiva i única. Aquesta millora no només aspira a un disseny visual agradable, sinó que també s'orientarà a augmentar la usabilitat i l'accessibilitat, assegurant que el disseny compleixi amb els estàndards més alts de qualitat i satisfacció de l'usuari.

10.4 Publicació de l'aplicació a Play Store

Una vegada s'hagin completat les millores en el disseny i s'hagi realitzat el desplegament del servidor en el núvol, es preveu publicar l'aplicació a Play Store. Això permetrà que un públic més ampli tingui accés a la plataforma i pugui beneficiar-se de les seves funcionalitats per optimitzar les seves compres en diversos supermercats.

11 CONCLUSIONS

En aquest informe final, hem repassat exhaustivament el procés complet de desenvolupament del meu Treball de Final de Grau (TFG). S'ha destacat la evolució des dels requisits inicials fins a la realització final, abordant les adaptacions necessàries en la planificació per superar els imprevistos que van sorgir. Hem analitzat en profunditat el desenvolupament tant del servidor com de l'aplicació mòbil, destacant els reptes trobats i les solucions implementades.

El servidor s'ha completat amb èxit, superant tots els obstacles i desafiaments que han sorgit al llarg del procés de desenvolupament. Quant a l'aplicació mòbil, tots els components i funcionalitats han estat implementats amb èxit, incloent la llista de productes per supermercats i categories, el buscador de productes, la cistella de compra i la pantalla d'incidències per a l'administrador. Aquesta fase final ha consolidat el meu coneixement i experiència en tecnologies com Kotlin i Jetpack Compose per al desenvolupament d'aplicacions modernes, així com en tècniques avançades com el web scraping amb Selenium per al servidor.

Aquest TFG ha estat una oportunitat invaluable per aplicar i ampliar els meus coneixements acadèmics i professionals. Ha requerit l'aplicació d'arquitectures avançades com l'hexàgonal i models de vista-vista-model (MVVM), àmpliament utilitzats en el desenvolupament de software actual. A nivell personal, gestionar aquest projecte ha suposat un creixement significatiu en la capacitat de resolució de problemes i adaptació a situacions inesperades, demostrant una dedicació constant per assolir els objectius establerts.

En resum, aquest TFG representa una fita important en el meu desenvolupament acadèmic i professional. Gràcies a l'èxit en l'execució i finalització d'aquest projecte, estic segur que he adquirit les competències i habilitats necessàries per afrontar reptes més grans en el futur professional, així com una base sòlida per continuar aprenent i millorant les meves capacitats tècniques.

12 AGRAÏMENTS

Agreixo profundament als meus pares, germans i família per donar-me suport emocional incondicional, no només durant el desenvolupament d'aquest TFG, sinó també al llarg de tota la meua carrera acadèmica. Sense el seu suport, no hauria arribat fins aquí.

A la meua tutora de TFG, Yolanda Benítez Fernández, per estar sempre disponible per ajudar-me i donar-me suport en tot moment.

A tots els professors d'enginyeria informàtica que m'han fet classe, ja que, tot i no em coneguin personalment, m'han ajudat molt amb el seu esforç constant per explicar les coses de la millor manera possible.

Finalment, a tots els meus estimats companys d'enginyeria amb qui he compartit aquest camí i que sempre han estat presents per ajudar-nos mútuament a avançar en els nostres estudis.

REFERÈNCIES

- [1] ChatGPT. *Correccions de faltes d'ortografies i millores gramaticals*. 2024. URL: <https://chat.openai.com>.
- [2] Datademia. *¿Què és el Web Scraping?* 2024. URL: <https://datademia.es/blog/que-es-web-scraping> (cons. 16-02-2024).
- [3] Google Drive. *Diagrama de casos d'ús de l'aplicació ShopSmartly*. 2024. URL: https://drive.google.com/file/d/1wk_fClpBa4t4Vip3oSyUkNQnIKBgTczI/view?usp=sharing (cons. 02-03-2024).
- [4] Google Drive. *Diagrama de classes del servidor de ShopSmartly*. 2024. URL: https://drive.google.com/file/d/1CsSLuBKt4m1Orm-EEVtL30Bb8vO5_i0n/view?usp=sharing (cons. 05-03-2024).
- [5] Google Drive. *Disseny de la base de dades de ShopSmartly*. 2024. URL: https://drive.google.com/file/d/15PGNoUrlgGaTUNxntA0qbRfrN_byTpus/view?usp=sharing (cons. 07-03-2024).
- [6] Google Drive. *Disseny del prototip de l'aplicació ShopSmartly per a usuaris comuns*. 2024. URL: <https://drive.google.com/file/d/1-TG4bctFQ1ycP5Wr-28VWt-lsm3EE3P2/view?usp=sharing> (cons. 01-04-2024).
- [7] Google Drive. *Disseny del prototip de l'aplicació ShopSmartly per a usuaris normals/compradors*. 2024. URL: <https://drive.google.com/file/d/1-TG4bctFQ1ycP5Wr-28VWt-lsm3EE3P2/view?usp=sharing> (cons. 01-04-2024).
- [8] La Vanguardia. *Les millors aplicacions Android per comparar preus de supermercats*. 2022. URL: <https://lavanguardia.com/andro4all/aplicaciones-gratis/las-mejores-aplicaciones-android-para-comparar-precios-de-supermercados> (cons. 18-02-2024).

APÈNDIX

A.0.1 Part del test plan

Cas d'ús	Accions	Resultat esperat
Modificar tema de l'aplicació.	Des de la pantalla principal que es mostra en obrir l'aplicació abans d'iniciar sessió o crear un compte en Shop Smartly, hem de clicar en el switch que està en la part superior esquerra de la pantalla.	Si l'aplicació està en mode clar ha de passar a mode fosc. Si l'aplicació està en mode fosc ha de passar a mode clar.
Modificar tema de l'aplicació.	Des de la pantalla de perfil d'usuari, hem de clicar en el switch que està en la part superior esquerra de la pantalla.	Si l'aplicació està en mode clar ha de passar a mode fosc. Si l'aplicació està en mode fosc ha de passar a mode clar.
Modificar l'idioma de l'aplicació.	Des de la pantalla principal que es mostra en obrir l'aplicació abans d'iniciar sessió o crear un compte en Shop Smartly, hem de clicar en el desplegable que està en la part superior dreta de la pantalla i seleccionar l'idioma que volem provar.	En seleccionar un idioma hem de veure com el lema de Shop Smartly com també les paraules d'iniciar sessió i registrar-se que apareixen en els respectius botons estiguin en l'idioma seleccionat.
Modificar l'idioma de l'aplicació.	Des de la pantalla de perfil d'usuari, hem de clicar en el desplegable que està en la part superior dreta de la pantalla i seleccionar l'idioma que volem provar.	En seleccionar un idioma hem de veure com el text de benvinguda a l'espai personal com la resta de contingut del perfil d'usuari estiguin en l'idioma seleccionat.
Navegar a la pantalla d'enregistrament.	Des de la pantalla principal que es mostra en obrir l'aplicació abans d'iniciar sessió o crear un compte en Shop Smartly, hem de clicar en el botó que diu "Sign Up"(en anglès).	Es mostra la pantalla d'enregistrament on es veu el logo de l'aplicació i els camps que s'han d'emplenar per enregistrar-se.
Enregistrar-se amb camps buits o incorrectes.	Emplenar els camps requerits amb dades incorrectes com ara amb un correu electrònic que no sigui del format "nom@domini.extensió", amb una confirmació de contrasenya incorrecte o deixant alguns camps buits.	En voler clicar en el botó d'enregistrar-se es veu que està deshabilitat i no fa res en clicar en ell.
Enregistrar-se amb un correu electrònic existent.	Emplenar els camps d'enregistrament correctament però fent servir un correu electrònic existent com ara "admin@shopsmartly.com".	En clicar en enregistrar-se es mostra un diàleg informant que el correu ja està enregidrat i que, per tant, no s'ha pogut crear el compte. En tancar el diàleg es mostra la pantalla d'enregistrament amb les dades que ha introduït l'usuari.
Registrar-se a Shop Smartly correctament.	Introduir dades correctes (no deixar camps buits, introduir un correu electrònic en el format correcte i que no estigui enregidrat, etc.).	En clicar en el botó crear compte es mostra un diàleg que s'ha creat correctament el compte i navega a la pantalla d'iniciar sessió.
Navegar a la pantalla d'inici sessió.	Des de la pantalla d'inici hem de clicar en el botó que diu "Sign In"(en anglès).	Es mostra el logo de l'aplicació i de baix es veu el camp de correu electrònic i contrasenya per iniciar sessió a més de l'opció de recuperació de contrasenya.
Iniciar sessió amb dades incorrectes o amb camps buits.	Escriure un correu electrònic que no estigui en el format correcte o deixar-se algun camp (ja sigui el de correu electrònic, contrasenya o ambdós) buit.	En voler clicar en el botó d'iniciar sessió es veu que el botó està deshabilitat i no fa res.
Iniciar sessió amb un correu electrònic no existent.	Escriure un correu electrònic no existent (és a dir, que no estigui enregidrat) i introduir una contrasenya qualsevol.	En clicar en iniciar sessió, l'aplicació informa l'usuari amb un diàleg que el correu electrònic introduït no està enregidrat en l'aplicació. En tancar el diàleg es mostra la pantalla d'iniciar sessió amb les dades que ha introduït prèviament l'usuari.
Iniciar sessió amb una contrasenya incorrecta.	Introduir un correu electrònic registrat en Shop Smartly amb una contrasenya incorrecta.	En clicar en boto d'iniciar sessió es mostra un diàleg informant que la contrasenya introduïda és incorrecta. En tancar el diàleg es veu la pantalla d'iniciar sessió amb les credencials introduïdes prèviament.