



UNIVERSITAT DE  
BARCELONA

**Treball de Fi de Grau**

**GRAU D'ENGINYERIA INFORMÀTICA**

**Facultat de Matemàtiques i Informàtica  
Universitat de Barcelona**

---

**Desenvolupament d'una Aplicació Web de  
Venda de Roba**

---

**Marc Colominas**

Director: Jordi José Bazán  
Realitzat a: Departament de  
Matemàtiques i Informàtica  
Barcelona, 10 de Juny de 2024

# Resum

En l'era digital actual, la presència en línia és fonamental per a qualsevol empresa que desitgi entrar en un mercat altament competitiu. OkonoFit, una marca emergent de roba esportiva, s'enfronta al desafiament de passar de vendre els seus productes a través de mitjans tradicionals i xarxes socials a establir una plataforma de comerç electrònic. Aquest treball de fi de grau té com a objectiu desenvolupar una web completa per a OkonoFit, permetent-li expandir el seu abast de venda i millorar la seva presència en el mercat digital.

El projecte es divideix en diverses fases que inclouen la planificació, l'anàlisi, el disseny, la implementació i la publicació d'una solució de comerç electrònic.

S'ha utilitzat la metodologia Agile per gestionar de manera eficient el desenvolupament del projecte, adaptant-se a les necessitats específiques i als reptes tècnics que s'han anat presentant.

Per a la implementació del frontend, s'ha optat per utilitzar ReactJS juntament amb Next.js, Tailwind CSS i Flowbite, proporcionant una interfície d'usuari moderna i intuïtiva. Pel que fa al backend, s'ha utilitzat Flask per a la creació d'una API robusta que gestiona les operacions CRUD, connectant-se a la base de dades de Firebase per a l'emmagatzematge i recuperació de dades.

El procés de desenvolupament ha inclòs la identificació i especificació detallada de casos d'ús, el disseny dels endpoints necessaris per satisfer els requisits del sistema i la implementació d'un sistema de maneig d'errors eficaç per garantir una experiència d'usuari consistent i sense interrupcions.

Aquest treball no pretén proporcionar a OkonoFit una eina completament funcional per augmentar les seves vendes en línia, sinó crear la primera part d'una plataforma en línia per acostar-se a publicar la web en un futur pròxim. A través d'aquest projecte, es busca demostrar la importància del comerç electrònic en l'actualitat i oferir una solució tecnològica que respongui a les necessitats reals d'una empresa emergent en creixement.

# Summary

In today's digital era, having an online presence is essential for any business aiming to introduce himself to a competitive market. OkonoFit, an emerging sportswear brand, faces the challenge of transitioning from selling its products through traditional means to establishing an e-commerce platform. This final degree project aims to develop a complete website for OkonoFit, allowing it to expand its sales reach and enhance its digital market presence.

The project is divided into several phases including planning, analysis, design, implementation, and deployment of an e-commerce solution. The Agile methodology has been adopted to efficiently manage the project's development, adapting to the specific needs and technical challenges encountered along the way.

For the frontend implementation, ReactJS has been chosen along with Next.js, Tailwind CSS, and Flowbite, providing a modern and intuitive. For the backend, Flask has been used to create the API that handles CRUD operations, connecting to the Firebase database for data storage and retrieval.

The development process included identifying and detailing use cases, designing the necessary endpoints to meet the system's requirements, and implementing an effective error handling system to ensure a consistent and uninterrupted user experience. The solution has been deployed using hosting platforms such as Vercel for the frontend and Heroku for the backend, ensuring continuous integration and facilitating future maintenance.

This project not only aims to provide OkonoFit with a functional tool to increase its online sales but also serves as a foundation for future improvements and expansions. Through this project, the importance of e-commerce in today's world is demonstrated, and a technological solution that responds to the real needs of a growing emerging company is offered.

# Índex

<b>Introducció</b>	<b>5</b>
Motivació	5
Necessitat i utilitat	6
Estructura de la memòria	6
<b>Objectius</b>	<b>8</b>
<b>Comerç electrònic</b>	<b>10</b>
<b>Planificació</b>	<b>12</b>
Punt de partida	12
Metodologia	13
<b>Anàlisi i disseny</b>	<b>16</b>
Entrevista amb OkonoFit	16
Requisits de l'Aplicació	17
Anàlisi tecnològic	18
Elaboració casos d'ús	23
Disseny de la API	24
Disseny del frontend	25
<b>Implementació</b>	<b>27</b>
Backend	27
Frontend	33
Desplegament	37
<b>Proves realitzades</b>	<b>38</b>
Resultats	40
<b>A futur</b>	<b>42</b>
<b>Conclusions</b>	<b>44</b>
<b>Bibliografia</b>	<b>45</b>
<b>Annexos</b>	<b>50</b>
Entrevista OkonoFit	50
Casos d'ús	52
Endpoints	60
Proves	71

# Introducció

Aquest és el treball de fi de grau pel Grau en Enginyeria Informàtica a la Universitat de Barcelona, en el que aplico una part dels coneixements adquirits durant la carrera per aprendre com començar a treballar per fer la creació d'un e-commerce per a una empresa de roba anomenada OkonoFit.

OkonoFit és una empresa emergent que ha creat una marca de roba esportiva, és una marca molt recent amb ganes de créixer i fer-se un nom en un sector ja establert.

En aquest primer apartat del treball veurem quina és la motivació que m'ha portat a fer aquest projecte, quina utilitat tindrà i com s'ha estructurat la memòria del projecte.



Imatge. OkonoFit logo

## Motivació

Durant la de carrera he tocat diversos llenguatges i he conegut diferents formes de veure la programació, però mai vaig tenir clar cap a quin sector volia conduir la meva carrera professional un cop acabada l'acadèmica.

Això es va resoldre fa relativament poc, quan buscant pràctiques d'empresa on poder tenir una primera experiència laboral i a on poder ampliar els meus coneixements, vaig trobar una oferta com a frontend developer. Aquesta oferta consistia en una formació inicial amb React i posteriorment la incorporació com a becari en un equip professional.

Durant aquests mesos de pràctiques se'm van aclarir les idees, i vaig trobar un camí pel qual continuar. És per aquesta raó que m'agradaria continuar formant-me com a programador frontend al mateix temps que anar agafant coneixements de backend de cara a poder acabar convertint-me en un desenvolupador full-stack.

Donat aquest context, he tingut bastant clar cap a on enfocar el meu treball de fi de carrera. Vull crear una aplicació web des de zero, on tingui el repte de crear tant la part frontal de la web com el seu corresponent backend.

Un cop arribat a aquest punt, només necessitava trobar un tema sobre el qual crear la web. Aquí vaig pensar en OkonoFit, una marca de roba propera que acaba de començar i que avui en dia només té vendes per contacte directe.

## Necessitat i utilitat

Com hem vist en el punt anterior, OkonoFit és una empresa recentment creada que busca créixer, ja que actualment només té capacitat de vendre els seus productes a través del boca a boca o través d'Instagram. Aquesta limitació en les seves vies de venda ha portat l'empresa a plantejar-se com a objectiu ampliar el seu rang de venda.

Per aconseguir-ho, contempen tres opcions principals:

1. Obrir una botiga física: Aquesta opció és inviable per a un projecte com el seu, ja que no disposen dels recursos econòmics necessaris per dur-la a terme en l'actualitat.
2. Buscar un proveïdor per vendre els seus productes a altres botigues: Tot i que aquesta opció podria ser viable, implica una sèrie de desafiaments logístics i dependència de tercers.
3. Obrir un portal de venda online: Aquesta és l'opció que es veu més factible i adequada per a les necessitats actuals de l'empresa.

El punt 3 ens condueix a nosaltres. Davant la necessitat de créixer i ampliar els seus mètodes de venda, obrir una botiga online és l'opció més accessible i amb major potencial de rendibilitat per OkonoFit en aquest moment. Aquest projecte no només permetrà a OkonoFit augmentar la seva visibilitat i arribar a un públic més ampli, sinó que també els proporcionarà una plataforma sòlida i professional per gestionar les seves vendes i interaccions amb els clients de manera més eficient.

## Estructura de la memòria

La memòria està estructurada per proporcionar una visió clara i detallada dels passos seguits durant el desenvolupament del projecte de creació d'una plataforma de comerç electrònic per una marca de roba. A continuació es descriu l'estructura de la memòria, destacant els apartats principals i el contingut que s'hi inclou a cada un:

1. **Introducció:** En aquest apartat es presenta una visió general del projecte, incloent-hi la motivació, la necessitat i utilitat del projecte, així com una breu explicació sobre l'estructura de la memòria.
2. **Objectius:** Es defineixen els objectius generals i específics del projecte, tant des d'un punt de vista tècnic com personal.

3. **Comerç electrònic:** Es du a terme un anàlisi de l'estat actual del comerç electrònic, incloent-hi els diferents tipus, avantatges i desavantatges, així com el context específic de la marca OkonoFit.
4. **Planificació:** Es descriu el procés de planificació del projecte, fent una recapitulació a l'estat de partida, juntament amb metodologia utilitzada i la planificació seguida.
5. **Anàlisi i disseny:** Aquesta secció inclou l'entrevista amb OkonoFit per recollir els requisits de l'aplicació, l'anàlisi tecnològic, l'elaboració dels casos d'ús, el disseny de la API i del frontend.
6. **Implementació:** Es detalla el procés de desenvolupament tant del backend com del frontend de l'aplicació, l'arquitectura del sistema, i la integració entre el backend i el frontend. Finalment, es descriu el desplegament del sistema en plataformes de hosting.
7. **Proves i resultats:** Es presenta la metodologia de prova utilitzada juntament amb els resultats obtinguts.
8. **A futur:** Es descriuen les implementacions necessàries per la publicació de la web al públic i altres possibles millores alternatives
9. **Conclusions:** En aquest apartat es reflexiona sobre els èxits i els reptes del projecte, es fa una valoració general del treball fet i es proporcionen les conclusions finals sobre el desenvolupament de la plataforma de comerç electrònic per a OkonoFit.

# Objectius

Fins ara hem vist les raons per les quals he decidit afrontar aquest projecte. Ara toca centrar-nos en quins són els objectius establerts a l'inici d'aquest.

L'objectiu principal d'aquest treball és desenvolupar una plataforma web de comerç electrònic que faciliti les vendes de productes per a una tenda de roba. Aquesta tenda ha de permetre als usuaris realitzar tot el procés de compra, des de veure un producte que els agradi, crear una sessió i finalment simular la compra. Els usuaris podran accedir al catàleg de productes i a la informació detallada de cadascun d'ells. Si ho desitgen, podran efectuar la compra un cop estiguin registrats i hagin iniciat sessió.

A més, hi haurà una segona web exclusiva per administradors on es podrà modificar productes, i modificar dades d'altres usuaris com assignar permisos d'administrador.

A continuació, es presenta un llistat dels principals objectius del treball:

## Objectius Generals

1. Desenvolupar una plataforma web de comerç electrònic que permeti als usuaris realitzar tot el procés de compra de manera eficient i segura.
2. Desenvolupar una web d'administració per gestionar productes i usuaris.

## Objectius Específics

1. Desenvolupar un backend de qualitat:
  - Implementar tota la funcionalitat necessària per nodrir la pàgina web.
  - Assegurar la seguretat i eficiència del servidor.
2. Desenvolupar un frontend de qualitat:
  - Crear un disseny d'interfícies atractiu i funcional.
  - Facilitar una navegació fàcil i intuïtiva per als usuaris.
  - Garantir una correcta gestió d'errors i una experiència d'usuari satisfactòria.
3. Dissenyar una base de dades robusta:
  - Guardar totes les dades del projecte de manera segura.
  - Assegurar la integritat i consistència de les dades.
4. Integrar correctament el backend amb el frontend:
  - Garantir una comunicació fluida i eficient entre el client i el servidor.
  - Assegurar que totes les funcionalitats es desenvolupin de manera coordinada i sense errors.

## Objectius Tècnics

1. Utilitzar les millors pràctiques de React:
  - Desenvolupar una pàgina web utilitzant les bones pràctiques de React.
  - Assegurar l'eficiència i sostenibilitat del codi.
2. Utilitzar les millors pràctiques de Flask:
  - Desenvolupar el backend seguint les directrius i pràctiques recomanades per Flask.
  - Garantir l'escalabilitat de l'aplicació.

## Objectius Personals

1. Millorar les habilitats en React:
  - Continuar formant-me i coneixent a fons el framework React.
  - Aprendre a desenvolupar una pàgina web de qualitat des de zero.
2. Aprendre a utilitzar Flask:
  - Desenvolupar competències en el framework Flask.
  - Comprendre com desenvolupar un backend robust i segur.

Amb els objectius clars, aquest projecte es proposa no només proporcionar una solució tecnològica efectiva per a OkonoFit, sinó també fomentar el meu creixement personal en el camp del desenvolupament web.

# Comerç electrònic

El comerç electrònic es basa en la compra i venda de productes o serveis a través d'internet, així com la transferència de diners i dades per efectuar aquestes transaccions. Principalment, es tracta de la venda de productes físics en línia, però també inclou altres tipus de transacció comercial que es realitzen a través d'internet.

A continuació, veurem diferents tipus de comerç electrònic, identificarem el nostre cas específic, i analitzarem els avantatges i desavantatges del comerç electrònic.

## Tipus de Comerç Electrònic

Existeixen diferents tipus de comerç electrònic que s'expliquen segons la relació entre consumidors i empreses:

1. B2B (Business to Business): Empreses o negocis que venen els seus productes o serveis a altres companyies.
2. B2C (Business to Customer): És el perfil més comú, referint-se a una empresa que ven als consumidors individuals.
3. C2B (Customer to Business): Un consumidor que ven els seus propis productes o serveis a una empresa.
4. C2C (Customer to Customer): Un consumidor ven a un altre consumidor. Les empreses creen portals que connecten consumidors.
5. B2G (Business to Government): Les empreses venen a governs o entitats governamentals.
6. C2G (Customer to Government): Els consumidors venen a governs o entitats governamentals.
7. G2B (Government to Business): Els governs o entitats governamentals venen a empreses.
8. G2C (Government to Customer): Els governs o entitats governamentals venen a consumidors.

En el nostre cas, la plataforma que volem desenvolupar en aquest treball és un exemple de B2C, al tractar-se d'una empresa que ven els seus productes a consumidors finals.

## Avantatges del Comerç Electrònic

El comerç electrònic ha crescut exponencialment els últims anys, entre altres coses gràcies als nombrosos avantatges que ofereix en comparació amb el comerç tradicional:

1. Més exposició: Com que el negoci no està limitat a un punt físic, sinó que és accessible des de qualsevol navegador web, els usuaris poden accedir-hi des de qualsevol lloc.
2. Disponibilitat 24/7: No cal tenir treballadors presencialment per mantenir la botiga en funcionament, així que és accessible les 24 hores del dia.
3. Costos reduïts: No és necessari disposar d'un local físic, estalviant en despeses fixes com lloguer, llum, aigua, etc. A més, els costos de comercialització, atenció al client i gestió d'inventari es redueixen gràcies a la digitalització.
4. Millor coneixement dels clients: Quan un client es registra per comprar en línia, l'empresa obté més informació sobre els seus hàbits i preferències, la qual cosa permet millorar les estratègies de màrqueting.
5. Facilitat per mostrar més productes: En una botiga en línia no hi ha limitacions d'espai, permetent mostrar un catàleg més ampli i detallat dels productes.

## Desavantatges del Comerç Electrònic

Tot i els avantatges mencionats, el comerç electrònic també presenta certs desavantatges respecte al comerç tradicional:

1. Dependència d'Internet: Una connexió a internet i el seu correcte funcionament són indispensables per utilitzar la plataforma. Qualsevol error a la xarxa pot fer-la inaccessible.
2. Falta de confiança: Encara que el comerç electrònic està molt present en la societat, per a moltes persones encara és una novetat, la qual cosa pot generar desconfiança a l'hora d'introduir dades bancàries.
3. Productes intangibles: Els clients no poden provar els productes abans de comprar-los en línia, de manera que només poden saber si el producte compleix les seves expectatives quan el reben.

Aquesta reflexió ajuda a entendre millor el context i les implicacions del comerç electrònic, així com els desafiaments i oportunitats que presenta per a una empresa emergent com OkonoFit.

# Planificació

A continuació, darem un repàs de quina ha estat la planificació i com s'ha afrontat el projecte. Però per fer-ho, primer cal posar en context d'on partim.

## Punt de partida

Aquest és el segon intent de desenvolupar el meu treball de fi de grau centrat en el desenvolupament d'una tenda de roba online. Per aquesta raó, considero oportú fer una recapitulació de per què no va ser satisfactori el primer intent.

Des d'un inici, proposava implementar de zero una web per a una marca de roba, i tot i l'interès a dur-lo a terme, l'avaluació rebuda indicava una realitat: no havien complert els objectius. L'execució del projecte va estar marcada per una significativa falta de planificació i organització, acompanyada d'una conducta de treball lluny de la requerida per un projecte com aquest.

Amb això no busco justificar el treball fet, sinó entendre les seves causes i buscar un camí per millorar-lo.

## Anàlisi de Problemes

### **Planificació**

Tot i que inicialment havia plantejat treballar amb una metodologia Agile, durant el desenvolupament del treball no vaig ser capaç d'aplicar-la i la vaig abandonar per acabar treballant dia a dia en el que creia convenient. En aquell moment vaig pensar que era la millor opció per continuar, però el resultat obtingut em va deixar clar que estava equivocat. Per això considero necessari tornar a optar per la metodologia plantejada inicialment aplicant les millores necessàries.

En aquest segon intent de desenvolupar el treball aplicaré la metodologia Agile, utilitzant sprints per organitzar-me i establint petits objectius a complir cada dues setmanes.

### **Conducta de Treball**

Un altre dels problemes va ser la conducta de treball i la mala organització. No vaig saber enfocar-me en el projecte i aprofitar les hores dedicades. La pèrdua de temps en variar constantment els objectius a curt termini va ser un factor clau en la mala conducta. A més, d'adonar-me que és necessari tenir més dedicació d'hores.

De cara a continuar el projecte, m'he compromès amb mi mateix a seguir la planificació inicial, complint amb les dades marcades dins del possible i dedicar el temps necessari per complir amb els objectius.

## Lliçons Apreses

1. No deixar de seguir la planificació: La planificació és crucial per mantenir un rumb clar i evitar treballar de manera desorganitzada.
2. L'organització és innegociable: Per no acabar fent tot a última hora, és imprescindible seguir els objectius i dates marcats..
3. Definició de requisits abans de començar: Cal tenir clar què s'ha de fer abans de començar a treballar per evitar desviacions.
4. Dedicació: Incrementar les hores dedicades i aprofitar-les de manera eficient és fonamental per complir amb els objectius establerts.

## Conclusions

En definitiva, el fracàs del projecte en la primera entrega es deu a la mala planificació, organització i dedicació. Considero que si faig un pas endavant en aquests aspectes, tinc els coneixements necessaris per fer un bon projecte i aconseguir els objectius establerts.

En aquesta segona oportunitat, em comprometo a aplicar rigorosament la planificació, mantenir una organització estricta, definir clarament els requisits abans de començar i dedicar el temps necessari per assegurar l'èxit del projecte. Amb aquestes millores, confio a poder desenvolupar una tenda de roba online que compleixi amb les expectatives i necessitats del projecte.

## Metodologia

Com he comentat, no vaig ser capaç d'aplicar la metodologia Agile durant el primer intent del projecte, però també he extret la conclusió que el problema no residia en la metodologia escollida, sinó en la meva organització i en el temps i qualitat dedicats al projecte.

Per això, he decidit tornar a aplicar de nou aquesta metodologia basada en Agile, personalitzada segons les meves necessitats i el fet de ser l'únic membre de l'equip en aquest projecte.

## Agile

Agile és un enfocament popular en el desenvolupament de programari que destaca per la seva flexibilitat, adaptabilitat i l'èmfasi a entregar valor durant tot el projecte.

He dividit el projecte en Sprints, que són períodes de temps concrets on s'executen tasques específiques. En el meu cas, cada Sprint dura dues setmanes. Aquesta estructura m'ajudarà a treballar de forma més organitzada i sistemàtica, facilitant el seguiment dels avenços.

Per tal de complementar els Sprints, juntament amb el tutor hem intentat anar fent reunions cada dues setmanes, on poder comunicar l'estat del projecte en cada punt i mostrar resultats obtinguts en cas necessari.

## Sprints i Tasques

### **Sprint 0: 1 a 10 de Març**

- **Tasques:**
  - Repàs de l'estat actual del projecte
  - Planificació
  - Formació en Flask (curs a Udemy)

### **Sprint 1: 11 a 25 de Març**

- **Tasques:**
  - Punt de partida
  - Entrevista amb Okono
  - Establiment de requisits
  - Redactar casos d'ús
  - Inicialització base de dades

### **Sprint 2: 26 de Març a 8 d'Abril**

- **Tasques:**
  - Definició d'endpoints
  - Implementació endpoints
  - Planificació del frontend

### **Sprint 3: 9 a 22 d'Abril**

- **Tasques:**
  - Disseny del frontend
  - Implementació frontend de clients

### **Sprint 4: 23 d'Abril a 6 de Maig**

- **Tasques:**
  - Implementació frontend de clients
  - Disseny del frontend de gestos
  - Gestió d'errors

### **Sprint 5: 7 a 20 de Maig**

- **Tasques:**
  - Finalització frontend de clients
  - Implementació frontend de gestors

### **Sprint 6: 21 de Maig a 10 de Juny**

- **Tasques:**
  - Redactar memoria del projecte
  - Arreglar defectes detectats
  - Proves finals

Aquest enfocament ha estat clau per mantenir el projecte en el bon camí, assegurant que cada component es desenvolupés correctament i que el resultat final complís amb les expectatives establertes. L'organització basada en Sprints m'ha ajudat a anar complint objectius a poc a poc de manera més eficient i estructurada.

# Analisi i disseny

Una bona anàlisi i disseny és crucial per comprendre les necessitats del projecte, definir les especificacions tècniques i establir una base sòlida per al desenvolupament.

En aquesta secció, el primer que veurem és una entrevista amb els responsables de la marca OkonoFit per obtenir una visió clara dels seus objectius i requisits, i així poder acabar d'establir requisits generals de l'aplicació.

## Entrevista amb OkonoFit

L'entrevista amb els responsables de la marca ens ha proporcionat informació valuosa sobre els objectius d'aquesta, la seva identitat, el públic objectiu i les funcionalitats indispensables per al lloc web. A continuació es detallen les principals conclusions de l'entrevista.

En un annex al final del projecte es pot observar l'entrevista completa.

### Objectius de la Marca

- Principals objectius: Fer-se un nom en el sector, inspirar un estil de vida actiu i saludable i oferir productes d'alta qualitat que combinin comoditat, funcionalitat i estil.
- Identitat i valors: La seva identitat es basa a fomentar un estil de vida concret, no només en vestir una marca de roba. Amb la Intenció de relacionar OkonoFit amb uns valors.
- Públic objectiu: Homes, sobretot entre 17 i 35 anys, que valoren la salut, el benestar i l'activitat física.
- Filosofia de la marca: La marca va néixer de la creença que tot és possible amb determinació i esforç. Els agradaria que aquesta filosofia impregni el lloc web a través d'històries inspiradores, consells d'entrenament i una comunitat en línia on els usuaris puguin compartir els seus èxits.

### Funcionalitats Essencials

- Tenda online: Simple i efectiva, de fàcil ús i que inclogui una secció d'atenció al client. Ha de permetre filtrar els productes segons les seves característiques.
- Contingut compartible: Incloure un apartat de contingut per poder compartir-lo amb els clients.

- Estructura i navegació: Preferim una estructura clara i simple, amb filtres específics per ajudar al client en la seva cerca. La navegació ha de ser intuïtiva, amb un menú principal fàcil d'accedir des de qualsevol pàgina.
- Actualització i manteniment: Ens agradaria tenir un sistema de gestió per actualitzar el contingut de la web.

## **Estratègia de Màrqueting**

- Atraure trànsit: Una combinació de SEO, contingut de valor a través de les nostres xarxes socials i publicitat pagada.

## **Conclusions de l'Entrevista**

Les conclusions extretes són que OkonoFit busca destacar-se en el mercat oferint més que simples productes esportius. La seva identitat es basa a fomentar un estil de vida actiu i saludable, i la marca vol ser percebuda com un símbol d'aquest estil de vida. El seu públic principal són homes entre 17 i 35 anys que valoren la salut, el benestar i l'activitat física.

La web ha de ser simple però efectiva. Una botiga en línia amb una navegació intuïtiva i una estructura clara és essencial per guiar el client en la seva compra. Els filtres de producte permeten una experiència de cerca efectiva, una secció d'atenció al client i de contingut ajuden a establir una relació més profunda amb els clients.

## **Requisits de l'Aplicació**

Basat en les conclusions de l'entrevista, s'han definit els següents requisits per a l'aplicació web:

### **Requisits Generals**

- Funcional: Ha de permetre realitzar els passos necessaris per efectuar una compra.
- Fàcil navegació i ús: El disseny del lloc web ha de ser clar i intuïtiu, permetent als usuaris trobar fàcilment la informació o els productes que busquen. Això inclou una estructura de navegació senzilla i velocitat de càrrega òptima.
- Seguretat: Implementar protocols de seguretat estàndard, com HTTPS, per protegir la informació dels usuaris.

### **Funcionalitats i Estructura**

- Botiga en línia: Integració d'un sistema de gestió de comerç electrònic per a la visualització de productes, carret de la compra i processament de comandes.

- Gestió d'inventari: Integració d'un sistema de gestió d'inventari.
- Registre d'usuaris: Permetre als usuaris crear comptes per gestionar les seves dades personals i historial de comandes.
- Cerca i filtrat de productes: Funcionalitats avançades de cerca i filtrat per permetre als usuaris trobar productes fàcilment segons les seves preferències.
- Contacte i suport: Formularis de contacte i informació de contacte per resoldre dubtes o problemes dels usuaris.
- Blog (Opcional): Secció per a articles que proporcionin valor afegit al públic objectiu, inclosos consells de salut, nutrició, entrenament i històries motivacionals.

### **Contingut Visual i Disseny**

- Galeria de productes: Imatges d'alta qualitat dels productes des de diferents angles, amb la possibilitat d'incloure vídeos de demostració d'ús.
- Disseny de marca: Integració dels elements de disseny de la marca, com ara logotip, paleta de colors i tipografies, per a una experiència de marca coherent.

### **Anàlisi tecnològic**

Al llarg d'aquest apartat es realitza un estudi de les diferents tecnologies disponibles per desenvolupar un projecte d'aquestes característiques i veurem una explicació detallada de les tecnologies finalment escollides per dur-lo a terme.

### **Mètodes per crear una web**

En l'actualitat, existeixen diferents mètodes per crear una tenda online:

- Prestashop: Plataforma de comerç electrònic de codi obert, ideal per aquells amb coneixements tècnics en desenvolupament web.
- Shopify: Solució popular per la seva facilitat d'ús i ràpida configuració, ideal per a principiants.
- Wordpress: Potent solució per crear tendes online, amb una gran varietat de temes i plugins.
- Magento: Plataforma de codi obert robusta i escalable, adequada per a tendes grans.
- BigCommerce: Plataforma tot en un amb plantilles i eines de màrqueting integrades.
- Programació personalitzada: Crear una web des de zero, oferint el màxim nivell de personalització.

En el nostre cas, hem optat per la programació personalitzada per la llibertat total que ens ofereix a l'hora de dissenyar i construir cada aspecte de la tenda, així com per la seva capacitat d'escalabilitat futura i la propietat total del codi.

## Possibles Tecnologies

### Frontend

El frontend és la part visible de l'aplicació amb la qual interactuen directament els usuaris. Inclou el disseny, gràfics i la presentació de la informació. A continuació, es presenten algunes tecnologies populars que podrien utilitzar-se per desenvolupar el frontend de la web:

- Vue: Framework de Javascript per crear interfícies d'usuari. Com a part bona te que inclou una corba d'aprenentatge fàcil, documentació clara i rendiment eficient. Un inconvenient que te una comunitat petita, cosa que ens podria perjudicar en cas de necessitar buscar dubte/problemes específics i la seva solució.
- Angular: Framework de codi obert desenvolupat per Google. Inclou robustesa i suport per Typescript. Com a inconvenient, té una corba d'aprenentatge gran.
- React: Biblioteca de Javascript de codi obert per crear interfícies d'usuari. Permet crear components reutilitzables, té una gran comunitat i moltes llibreries. De la mateixa manera que Angular, té una ran corba de dificultat d'aprenentatge.

### Backend

El backend gestiona la lògica de l'aplicació, el processament de dades i les operacions de fons. Algunes tecnologies populars per la creació de backends són:

- Express.js (Node.js): Framework minimalista i ràpid per a Node.js. Avantatges: utilització de Javascript tant al front com al back, gran quantitat de mòduls. Desavantatges: complexitat en el maneig d'errors.
- Django (Python): Framework alt nivell que segueix el patró MVC. Avantatges: seguretat robusta i altament utilitzat. Desavantatges: poca flexibilitat.
- Spring Boot (Java): Framework basat en microserveis. Avantatges: popular en aplicacions empresarials. Desavantatges: corba d'aprenentatge gran.
- Flask (Python): Microframework lleuger i minimalista. Avantatges: flexibilitat, ideal per projectes petits o mitjans. Desavantatges: requisits de configuració manual.

### Base de Dades

Les bases de dades permeten l'emmagatzematge i recuperació de dades. Algunes opcions populars poden ser:

- MongoDB: Base de dades NoSQL orientada a documents. Avantatges: maneig de gran volum de dades no estructurades. Desavantatges: complexitat en transaccions multidocument.
- MySQL: Sistema de gestió de bases de dades relacional. Avantatges: àmpliament utilitzat, compatible amb molts llenguatges. Desavantatges: problemes d'escalabilitat en aplicacions grans.
- PostgreSQL: Sistema de gestió de bases de dades relacional. Avantatges: suport per tipus de dades avançats. Desavantatges: administració complexa.
- Firebase: Plataforma al núvol amb base de dades NoSQL. Avantatges: sincronització de dades en temps real. Desavantatges: flexibilitat limitada i dependència de Google.

## Tecnologies escollides

Després d'avaluar diferents opcions, s'han escollit les següents tecnologies per al desenvolupament del projecte:

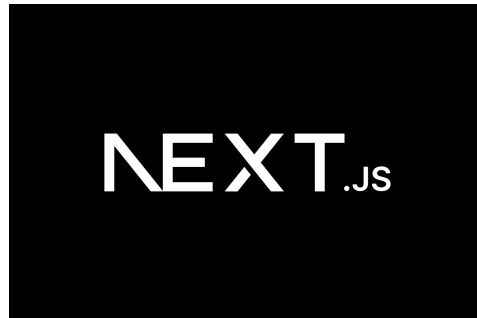
### Frontend

- ReactJS:
  - Descripció: Biblioteca de Javascript de codi obert per crear aplicacions web renderitzades al costat del client.
  - Avantatges: Components reutilitzables, gran comunitat de suport, diversitat de llibreries.
  - Característiques clau: Utilització de DOM virtual, components amb props per transmetre informació.
  - Exemples d'ús: Facebook, Instagram, WhatsApp, Netflix.



- NextJS:
  - Descripció: Marc de treball per a aplicacions React amb renderització al servidor o generació de llocs web estàtics.

- Avantatges: Optimització del rendiment, millora del SEO, suport per CSS-in-JS.
- Característiques clau: Pre-renderització de pàgines, routing basat en el sistema de fitxers.



- Tailwind CSS:
  - Descripció: Framework de CSS amb classes predefinides per a dissenyar la interfície d'usuari.
  - Avantatges: Desenvolupament ràpid i eficient, personalització a través d'un fitxer de configuració.
  - Característiques clau: Classes d'utilitat per a cada propietat de CSS, integració senzilla.
- Flowbite:
  - Descripció: Biblioteca de components basada en Tailwind CSS.
  - Avantatges: Components reutilitzables i fàcils de personalitzar.
  - Característiques clau: Components preconstruïts per a botons, formularis, targetes.

## Backend

- Flask:
  - Descripció: Microframework de Python centrat en el rendiment i disseny minimalista.
  - Avantatges: Flexibilitat, facilitat d'aprenentatge, ideal per projectes petits o mitjans.
  - Característiques clau: Model-Vista-Controlador, servidor de desenvolupament integrat, suport per proves unitàries.



- Postman:
  - Descripció: Aplicació per realitzar proves API.
  - Avantatges: Interfície gràfica d'usuari per testejar HTTP requests.
  - Característiques clau: Validació de respostes, suport per col·leccions d'APIs.

## Base de Dades

- Firebase:
  - Descripció: Plataforma al núvol per al desenvolupament d'aplicacions web i mòbils.
  - Avantatges: Autenticació d'usuaris, sincronització de dades en temps real.
  - Característiques clau: Firebase Auth per a autenticació, Firestore Database per a emmagatzematge NoSQL.



Aquestes tecnologies han estat seleccionades segons les meves preferències i necessitats personals juntament amb la seva capacitat de satisfer les necessitats del projecte, oferint flexibilitat, rendiment i facilitat de manteniment. Així poder garantir la construcció d'una plataforma robusta i escalable, alineada amb els objectius d'OkonoFit.

## Elaboració casos d'ús

Un cop tenim clares les necessitats i requisits a cobrir, és necessari elaborar un llistat de casos d'ús per comprendre com interactuaran els usuaris amb el sistema. Aquesta etapa és crucial, ja que ens ajuda a definir un disseny de l'aplicació que permetrà aplicar tots els requisits necessaris de manera efectiva.

Els casos d'ús descriuen les accions que els usuaris poden realitzar en el sistema i les respostes esperades d'aquest. Cada cas d'ús identifica un objectiu que l'usuari vol aconseguir i descriu els passos necessaris per assolir aquest objectiu, així com les possibles variacions i excepcions.

### Exemple de cas d'ús

#### C-01 Registre

- **Nom:** Registre
- **Actor:** Visitant
- **Objectiu:** Registrar-se a l'aplicació
- **Precondicions:** -
- **Postcondicions:** L'usuari té un compte creat
- **Flux:**
  - L'usuari introdueix el nom, correu electrònic i contrasenya al formulari de registre.
  - Es redirigeix a l'usuari a la pàgina de login.
- **Casos Alternatiu:**
  - L'usuari no introdueix les dades correctament al formulari.
    - L'usuari introdueix un correu electrònic ja registrat.
    - L'usuari no introdueix un format correcte de correu electrònic.
    - L'usuari no introdueix un format correcte de contrasenya.

Com es pot observar en l'exemple anterior, per cada cas d'ús hi trobem:

- **Nom:** Nom amb el qual identifiquem el cas d'ús.
- **Actor:** Indica el tipus d'usuari que utilitza la web.
- **Objectiu:** Finalitat que busca realitzar l'actor
- **Precondicions:** Condicions que s'han d'haver complert anteriorment per realitzar l'operació.
- **Postcondicions:** Indica l'estat en què es queda el sistema un cop finalitza l'operació.
- **Flux:** Descriu les fases que formen el cas d'ús
- **Cas d'ús alternatiu:** Descriu situacions que es poden donar en cas de no realitzar correctament els passos indicats.

S'ha afegit un annex al final del projecte amb el llistat complet de casos d'ús. Això ens ajudarà a revisar que tots els requisits quedin ben coberts i que el sistema es dissenya tenint en compte totes les possibles interaccions dels usuaris.

## Disseny de la API

Un cop tenim les necessitats i requisits recollits en els casos d'ús, cal dissenyar una API que ens ajudi a complir aquests requisits. L'objectiu és identificar i dissenyar l'estructura del backend per desenvolupar tots els endpoints necessaris per posar en marxa la web.

### Identificació dels Endpoints

Per començar, s'ha analitzat cada cas d'ús per determinar què ha de cobrir el backend. A partir d'aquesta anàlisi, s'ha creat un llistat de tots els endpoints necessaris per aconseguir un funcionament correcte de l'aplicació.

#### **Exemple d'Endpoint Identificat**

A continuació, es presenta un exemple detallat d'un dels endpoints identificats:

#### **Iniciar sessió /POST login**

##### **Entrada:**

- Correu
- Contrasenya

##### **Sortida:**

- Token

##### **Errors:**

- Usuari inexistent
- Contrasenya incorrecta

**Descripció:** Aquest endpoint permet als usuaris iniciar sessió a l'aplicació. Requereix que l'usuari proporcioni el seu correu electrònic i contrasenya. Si les credencials són correctes, el sistema genera i retorna un token d'usuari que serà utilitzat per a l'autenticació en futures peticions. En cas d'error, es retorna un missatge indicant si l'usuari no existeix o si la contrasenya és incorrecta.

### Paràmetres d'Entrada i Sortida

Per cada endpoint, s'ha establert quins han de ser els seus paràmetres d'entrada i sortida, així com els possibles errors que poden sorgir durant la crida. Això assegura que cada petició a l'API és ben definida i manejada de manera adequada.

De la mateixa manera que s'ha fet amb els casos d'ús, s'ha afegit un annex al projecte amb el llistat complet dels endpoints dissenyats. Aquest annex inclou una descripció detallada de cada un, els seus paràmetres d'entrada i sortida, i els possibles errors.

## Disseny del frontend

El disseny del frontend és crucial per crear una experiència d'usuari atractiva i intuïtiva. A continuació, es detalla el disseny inicial del frontend de la tenda online, incloent-hi l'estructura de les pàgines, els components principals, i els principis d'usabilitat que s'han tingut en compte.

El frontend s'ha dissenyat per ser modular i escalable, utilitzant ReactJS com a framework principal. Això permet crear components reutilitzables i gestionar l'estat de l'aplicació de manera eficient.

### Estructura del Frontend del Client

1. Login Page: Formulari d'inici de sessió.
2. Register Page: Formulari de registre per a nous usuaris.
3. Home Page: Vista principal que presenta productes destacats, informació i novetats.
4. Catàleg: Mostra el llistat de productes amb opcions de filtratge.
5. Producte: Mostra diverses imatges del producte, la seva informació i l'opció d'afegir-lo al carret de compra.
6. Pàgina de Compra: Formulari per introduir les dades finals de compra.
7. Perfil: Visualitza les dades de l'usuari amb l'opció de modificar-les.
8. Comandes: Visualitza el llistat de comandes d'un usuari.
9. Atenció al Client: Formulari per obrir un tiquet de suport.

### Estructura del Frontend del Gestor

1. Login Page: Formulari d'inici de sessió.
2. Home Page: Mostra l'opció de modificar dades d'un usuari o d'un producte.

## Principis d'Usabilitat

Per garantir una experiència d'usuari òptima, s'han seguit diversos principis d'usabilitat en el disseny del frontend:

- Simplicitat: La interfície ha de ser clara i senzilla, amb una navegació intuïtiva.
- Consistència: Mantenir un estil visual consistent a través de totes les pàgines i components.
- Feedback: Proporcionar feedback immediat a les accions de l'usuari, com ara confirmacions al efectuar una compra o errors en el formulari de pagament.
- Rendiment: Optimitzar la càrrega de les pàgines per oferir una experiència ràpida i fluida.

El disseny del frontend ha estat planificat per garantir que l'experiència d'usuari sigui intuïtiva, atractiva i efectiva. Mitjançant l'ús de tecnologies com ReactJS, Tailwind CSS i Flowbite, es busca crear una interfície d'usuari que sigui fàcil de mantenir i escalar. Aquest disseny inicial estableix una base sòlida per al desenvolupament del projecte i assegura que totes les necessitats i requisits identificats siguin complerts de manera òptima.

# Implementació

## Backend

El backend del nostre projecte s'ha desenvolupat utilitzant Flask, un microframework de Python conegut per la seva senzillesa i flexibilitat. La configuració inicial del backend inclou la creació d'una API amb diferents classes que gestionen les operacions necessàries per a cada recurs de l'aplicació. Aquestes classes tenen mètodes per a les operacions CRUD (Create, Read, Update, Delete) que es corresponen amb els mètodes HTTP POST, GET, PUT i DELETE. A més, el backend està connectat a Firebase, que s'encarrega de la persistència de les dades.

## Estructura i funcionament del Backend

El backend segueix una estructura modular que facilita la mantenibilitat i l'escalabilitat. Aquesta estructura es basa en diversos fitxers i carpetes que separen les diferents responsabilitats del codi.

### Fitxers i Carpetes Principals

1. app.py
  - Aquest és el fitxer principal de l'aplicació Flask. S'encarrega de configurar l'aplicació, inicialitzar extensions com Flask-RESTful i CORS, i registrar els blueprints de l'API.
  - Gestiona la configuració inicial, com la configuració de CORS per permetre sol·licituds des de les diferents webs i la configuració dels endpoints de l'API per gestionar les operacions CRUD per a diversos recursos com usuaris, productes, comandes, etc.
2. db.py
  - Aquest fitxer s'encarrega de la configuració i inicialització de la connexió amb Firebase. Proporciona una interfície per interactuar amb Firestore, la base de dades NoSQL de Firebase.
  - Inclou la configuració de les credencials de Firebase i inicialitza la seva connexió, permetent als models de dades interactuar amb la base de dades per emmagatzemar i recuperar informació.
3. models/
  - Aquesta carpeta conté els models de dades de l'aplicació. Cada model representa una entitat (per exemple, Usuari, Producte) i defineix els seus atributs i mètodes. Els models s'encarreguen de la lògica de negoci i de les operacions amb la base de dades.
  - Exemple: El model User defineix les propietats i mètodes associats als usuaris, com la creació d'un nou usuari i la verificació de tokens.
4. resources/

- Aquesta carpeta conté les rutes de l'API i les operacions CRUD associades a cada recurs. Utilitza Flask-RESTful per simplificar la creació d'API RESTful. Cada fitxer en aquesta carpeta representa un recurs específic i defineix les rutes i els mètodes HTTP corresponents.
- Exemple: El recurs Users defineix les rutes per a la gestió d'usuaris, incloent-hi operacions per crear, obtenir, actualitzar i eliminar usuaris.

Aquesta estructura modular permet un desenvolupament i manteniment més fàcil del codi, ja que cada part del backend està encapsulada i pot ser gestionada independentment. A més, facilita l'escalabilitat del projecte, pel fet que es poden afegir nous models i recursos sense afectar significativament l'estructura existent.

### **Exemple de Flux de Treball: Creació d'un Usuari**

1. Creació d'un Usuari:
  - El client envia una petició POST a l'endpoint /user amb les dades de l'usuari.
  - Flask rep la petició i la classe User del fitxer resources/users.py processa les dades.
  - El mètode add\_user del model User envia les dades a Firebase per el seu emmagatzematge.
  - Firebase retorna una resposta amb l'ID de l'usuari creat.
  - Flask envia una resposta al client amb l'ID de l'usuari i un missatge de confirmació.

```

user = User(
    nombre=data["Nombre"],
    apellidos=data["Apellidos"],
    correo=data["Correo"],
    contraseña=data["Contraseña"],
    rol=data.get("Rol", ""),
    pais=data.get("Pais", ""),
    calle=data.get("Calle", ""),
    piso=data.get("Piso", ""),
    codigo_postal=data.get("CodigoPostal", ""),
    ciudad=data.get("Ciudad", ""),
)

try:
    user_record = auth.get_user_by_email(data["Correo"])
    if user_record:
        return {
            "message": "Ya existe un usuario con ese correo electrónico"
        }, 400
except auth.UserNotFoundError:
    pass

try:
    user.add_user()
    return {
        "message": "El usuario {} ha sido añadido".format(data["Nombre"])
    }, 200
except Exception as e:
    return {"message": "Error al crear el usuario: " + str(e)}, 500

```

## 2. Obtenció d'un Usuari:

- El client envia una petició GET a l'endpoint /user.
- Flask rep la petició i la classe User del fitxer resources/users.py gestiona la sol·licitud.
- El mètode get\_user busca les dades de l'usuari a Firebase.
- Firebase retorna les dades de l'usuari.
- Flask envia una resposta al client amb les dades de l'usuari.

```

def get(self):
    token_header = request.headers.get("Authorization", None)
    if not token_header:
        return {"message": "No se proporcionó token"}, 401

    token = token_header.split(" ")[1]

    try:
        decoded_token = auth.verify_id_token(token)
    except exceptions.FirebaseError as e:
        return {"message": "Token inválido o expirado"}, 402

    correo = decoded_token.get("email")
    if not correo:
        return {"message": "No se pudo obtener el correo del token"}, 400
    us = db.collection("users").document(correo).get().to_dict()
    if not us:
        return {"message": "Usuario no encontrado"}, 404

    return us, 200

```

La implementació del backend amb Flask i Firebase proporciona una base robusta i flexible per a l'aplicació. Aquesta configuració em permet gestionar eficientment les operacions CRUD necessàries i mantenir la integritat i seguretat de les dades. Amb aquest backend ben estructurat, l'aplicació està preparada per oferir una experiència d'usuari consistent i fiable.

## Configuració base de dades

La configuració de la base de dades és una part fonamental del projecte, ja que emmagatzema tota la informació necessària per al funcionament de la plataforma de comerç electrònic. Per aquest projecte, he optat per utilitzar Firebase Firestore, una base de dades NoSQL que ofereix una alta escalabilitat i facilitat d'ús.

### Elecció de Firebase Firestore

Firebase Firestore ha estat escollida per diverses raons:

- **Escalabilitat:** Firestore gestiona automàticament la distribució de dades a gran escala i és capaç de gestionar aplicacions de qualsevol mida.
- **Sincronització en temps real:** Proporciona actualitzacions de dades en temps real, cosa que és essencial per a aplicacions que necessiten mostrar dades actualitzades constantment.
- **Integració amb altres serveis de Firebase:** Firestore s'integra perfectament amb altres serveis de Firebase com Firebase Auth, facilitant l'autenticació i la gestió d'usuaris.

- **Model de dades flexible:** Com a base de dades NoSQL, permet emmagatzemar dades en documents que es poden agrupar en col·leccions, cosa que proporciona una gran flexibilitat.

### Com funciona Firebase Firestore

Firestore utilitza una estructura de dades basada en documents i col·leccions:

- **Documents:** Cada document és un registre de dades que emmagatzema un conjunt de parells clau-valor. Els documents són similars a les files d'una base de dades relacional, però tenen un esquema flexible.
- **Col·leccions:** Una col·lecció és un contenidor de documents. Cada col·lecció pot contenir qualsevol nombre de documents, i cada document pot tenir subcol·leccions.

Aquesta estructura jeràrquica permet una gran flexibilitat a l'hora de modelar les dades. Per exemple, es poden tenir col·leccions per a usuaris, productes, comandes, etc., cadascuna amb els seus propis documents que contenen la informació pertinent.

### Estructura de la Base de Dades

En el context d'aquest projecte, la base de dades s'organitza de la següent manera:

- **Col·lecció users:** Emmagatzema la informació dels usuaris.
- **Col·lecció products:** Emmagatzema la informació dels productes.
- **Col·lecció user\_orders:** Emmagatzema la informació de les comandes separades per usuaris.
- **Col·lecció user\_cart:** Emmagatzema els productes guardats a la cistella de compra per usuaris.
- **Col·lecció user\_tickets:** Emmagatzema els tiquets de suport per usuaris.

### Firebase Auth

Firebase Auth s'utilitza per gestionar l'autenticació dels usuaris de l'aplicació. Proporciona diversos mètodes d'autenticació, incloent-hi correu electrònic i contrasenya, autenticació amb Google, Facebook, GitHub, entre d'altres. Firebase Auth facilita la creació, gestió i verificació d'usuaris, integrant-se perfectament amb Firestore per a una gestió completa dels usuaris.

L'ús de Firebase Auth en el projecte inclou:

- **Registre d'usuaris:** Permet als nous usuaris registrar-se a l'aplicació proporcionant un correu electrònic i una contrasenya.
- **Inici de sessió:** Permet als usuaris existents iniciar sessió per accedir a les funcionalitats de l'aplicació.

- **Gestió de sessions:** Manté les sessions d'usuari actives i segures.

## Implementació de la Base de Dades

La implementació de la base de dades s'ha realitzat mitjançant el fitxer db.py, que s'encarrega de la configuració inicial de Firestore. Aquest fitxer inicialitza la connexió a Firestore utilitzant les credencials proporcionades per Firebase i proporciona una interfície per accedir a la base de dades des dels diferents models.

```
from firebase_admin import credentials, initialize_app, firestore
from flask import Flask

key = "AIzaSyBPyshCXNnWnXTlSGzeJcl-cYTpVD-ZNGw"

cred = credentials.Certificate("key.json")
default_app = initialize_app(cred)
db = firestore.client()
```

## Frontend

El frontend de l'aplicació conta de dos webs diferents, la de client i la de gestor. Els dos frontends comparteixen una mateixa estructura. Han estat creats amb Next.js, i estan formats principalment de dues coses: les pàgines i els components.

### Pàgines

Les pàgines són el contenidor principal de cada vista de l'aplicació. Cada pàgina representa una ruta específica de l'aplicació i conté la lògica i les dades necessàries per a aquesta vista. És en les pàgines on es gestionen les peticions a l'API, la gestió de l'estat local i les operacions específiques de cada una.

### Exemple de pàgina

- Login: Formulari d'inici de sessió per als usuaris.



#### Inicia sesión con tu cuenta

Correo electrónico

Contraseña

[¿Has olvidado tu contraseña?](#)

**Iniciar sesión**

[¿No estas registrado? Hazlo aquí](#)

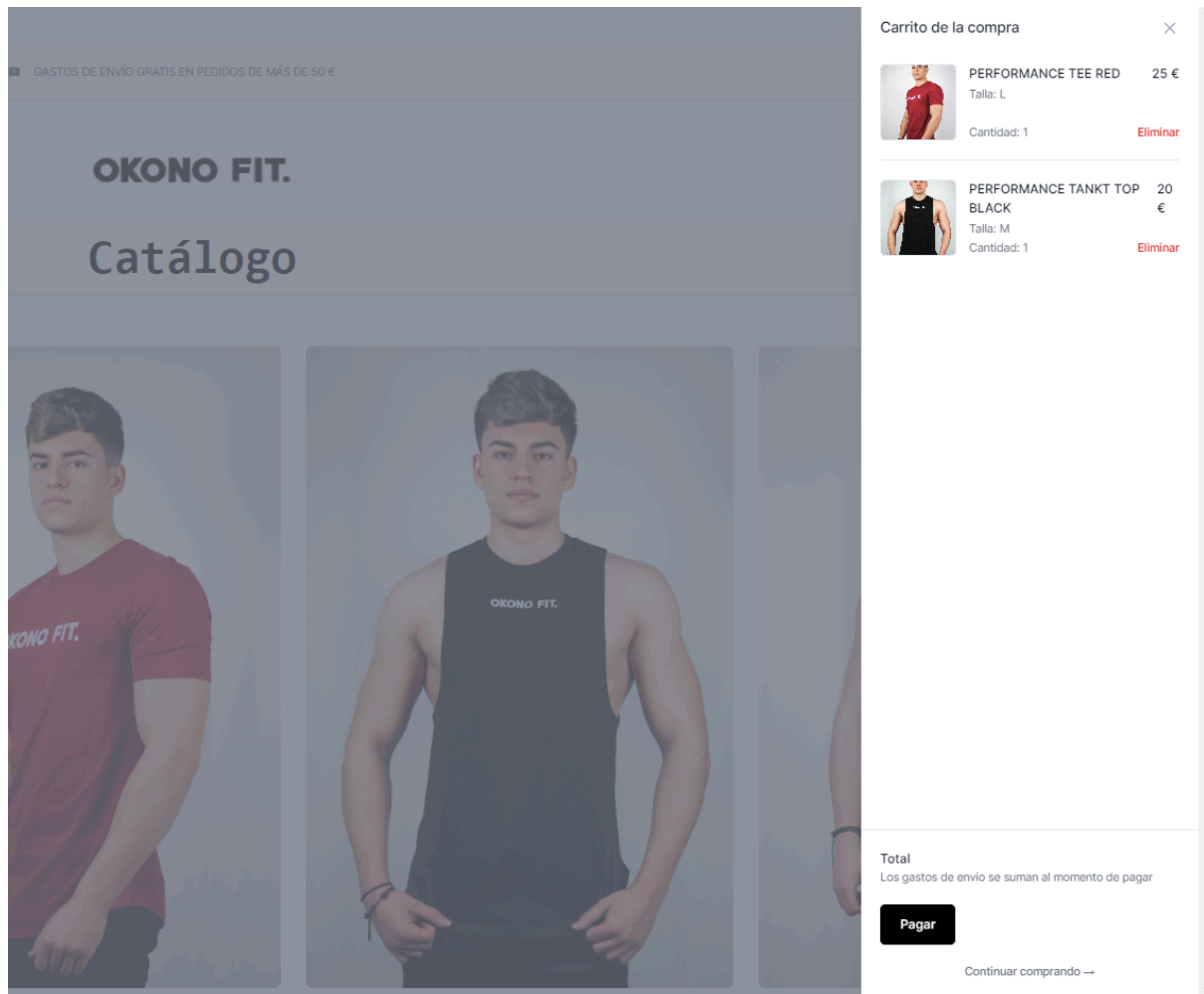
### Componentes

Els components són blocs de construcció reutilitzables que es criden des de les pàgines. Cada component encapsula una part específica de la interfície d'usuari i

pot ser reutilitzat en diverses pàgines de l'aplicació. Això permet mantenir un codi més net i modular, facilitant el manteniment i l'escalabilitat del projecte.

## Exemple de component

- Cistella de la compra: Panel lateral on trobem els productes inclosos en la cistella de compra.



## Conectivitat amb el Backend

Els dos frontends es comuniquen amb el backend mitjançant peticions HTTP als endpoints de l'API. Aquesta comunicació permet obtenir dades, enviar informació i fer operacions CRUD.

## Tipus de Peticions HTTP Utilitzades

1. GET: Utilitzada per recuperar dades del servidor. Aquesta petició es fa per obtenir informació com llistats de productes, detalls d'un producte, informació del perfil d'usuari, etc.
  - Exemple: Obtenir la llista de productes.
2. POST: Utilitzada per enviar dades al servidor per crear un nou recurs. Aquest tipus de petició es fa servir per operacions com el registre d'un nou usuari, la creació d'una nova comanda, etc.
  - Exemple: Crear una nova comanda.
3. PUT: Utilitzada per actualitzar dades existents al servidor. Aquest tipus de petició s'utilitza per operacions com l'actualització de la informació del perfil d'usuari, actualització de productes, etc.
  - Exemple: Actualitzar la informació del perfil d'un usuari
4. DELETE: Utilitzada per eliminar un recurs del servidor. Aquest tipus de petició s'utilitza per operacions com eliminar un compte d'usuari, eliminar un producte, etc.
  - Exemple: Eliminar un producte de la cistella.

## Gestió de l'Autenticació

Per assegurar que només els usuaris autenticats poden accedir a certes funcionalitats, el frontend inclou tokens d'autenticació en les peticions HTTP. Aquest token és rebut del backend durant el procés de login i es guarda en el navegador de l'usuari.

- Enviament del Token: A cada petició que requereix autenticació, el token s'inclou en les capçaleres HTTP.

```
def get(self):
    token_header = request.headers.get("Authorization", None)
    if not token_header:
        return {"message": "No se proporcionó token"}, 401

    token = token_header.split(" ")[1]

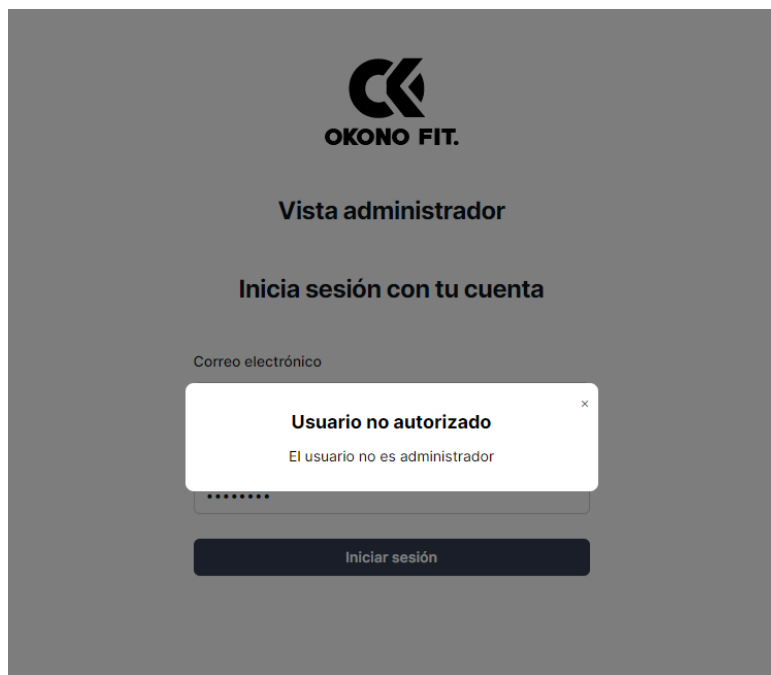
    try:
        decoded_token = auth.verify_id_token(token)
    except exceptions.FirebaseError as e:
        return {"message": "Token inválido o expirado"}, 402
```

## Tractament d'errors

El maneig d'errors és una part crucial per garantir una experiència d'usuari fluida i informativa. Per controlar-los, el backend envia diferents codis d'error segons el tipus d'error que es produeix en cada crida. El frontend, per la seva banda, controla aquests codis d'error i mostra missatges d'error pertinents utilitzant un component de modal d'error.

### Com Funciona el Maneig d'Errors

1. Enviament de Codis d'Error pel Backend:
  - Cada crida al backend pot retornar diferents codis d'error segons la naturalesa del problema. Per exemple, un error d'autenticació podria retornar un codi 401, mentre que un error de validació de dades podria retornar un codi 400.
2. Recepció i Control dels Codis d'Error pel Frontend:
  - El frontend rep la resposta de cada crida HTTP i analitza el codi d'error retornat. Segons el codi d'error, es crida el component ErrorModal amb un ID específic d'error.
3. Component ErrorModal:
  - El component s'encarrega de mostrar el missatge d'error pertinent a l'usuari. Aquest component rep un ID d'error, busca en un fitxer JSON d'errors el títol i text corresponent a aquest ID, i mostra un modal amb el missatge d'error.



Resumidament, la comunicació entre el frontend i el backend mitjançant peticions HTTP és essencial per al funcionament de l'aplicació. Utilitzant mètodes HTTP estàndard, el frontend pot efectuar les operacions necessàries per gestionar dades de manera eficient. La inclusió de tokens d'autenticació a les certes peticions assegura que només els usuaris autoritzats poden accedir a aquestes funcionalitats, mantenint així la seguretat de l'aplicació. Aquest model de comunicació permet una integració fluida i segura entre el client i el servidor, garantint una experiència d'usuari òptima.

## Desplegament

Un cop tenim tant el backend com el frontend de l'aplicació desenvolupats i funcionant, és hora de publicar l'aplicació. Per fer-ho, utilitzarem plataformes de hosting que ens permetin desplegar i mantenir l'aplicació en funcionament de manera eficient i fiable.

### Publicació del Frontend

Per al frontend, hem escollit Vercel com la plataforma de desplegament. Vercel està optimitzada per a aplicacions basades en JavaScript, com les desenvolupades amb React i Next.js. Aquesta plataforma ofereix diverses avantatges que s'alineen perfectament amb les nostres necessitats:

- Desplegaments automàtics: Vercel ofereix la integració directa amb repositoris de GitHub, GitLab i Bitbucket. Això permet que cada vegada que fem un push a la branca principal del nostre repositori, Vercel desplegui automàticament una nova versió de l'aplicació.
- Optimització per a React i Next.js: Vercel està especialment dissenyada per treballar amb aquestes tecnologies, el que facilita el procés de desplegament i assegura un rendiment òptim.
- Oferta gratuïta generosa: Vercel proporciona un pla gratuït molt adequat per a projectes petits i mitjans, oferint serveis de qualitat sense cap cost inicial.

#### **Procés de Desplegament amb Vercel:**

1. Associació del repositori: Hem associat el nostre repositori de GitHub amb Vercel, especificant la carpeta on es troba el frontend.
2. Configuració de desplegament: Hem configurat Vercel per desplegar automàticament cada cop que es fa un push a la branca main del repositori.
3. Desplegament automàtic: Vercel s'encarrega de construir i desplegar la nostra aplicació cada vegada que es fan canvis al repositori.

## URL de la Web:

- Client: <https://okonofit.vercel.app/>
- Admin: <https://okonofit-admin.vercel.app/>

## Publicació del Backend

Per al backend, hem optat per utilitzar Heroku, una plataforma coneguda per la seva facilitat d'ús i integració amb diverses eines de desenvolupament. Heroku ofereix avantatges significatius per a la publicació d'aplicacions web basades en Flask:

- Facilitat d'Ús: Heroku és molt fàcil d'utilitzar, amb una configuració senzilla i una corba d'aprenentatge baixa.
- Desplegaments mitjançant Git: Heroku permet clonar un repositori i desplegar canvis automàticament cada vegada que es fa un push.
- Nivell Gratuït: Heroku ofereix un nivell gratuït que és ideal per a projectes petits i en desenvolupament.

### Procés de Desplegament amb Heroku:

1. Configuració inicial: Hem creat una nova aplicació a Heroku.
2. Associació del repositori: Hem associat el nostre repositori de GitHub amb Heroku per facilitar els desplegaments.
3. Desplegament automàtic: Heroku desplega automàticament una nova versió del backend cada vegada que es fa un push al repositori.

La publicació de l'aplicació utilitzant Vercel per al frontend i Heroku per al backend ens ha permès pujar la web a la xarxa per poder-la utilitzar i provar correctament. Les dues plataformes ens permeten el desplegament automàtic que ens permet actualitzar el contingut tant de la web com de l'api sempre que sigui necessari ràpidament.

## Proves realitzades

Per assegurar el correcte funcionament de l'aplicació, s'han dut a terme diverses proves que comproven tant funcionalitats específiques com el flux general de l'aplicació. Aquestes proves inclouen la verificació de funcionalitats bàsiques, així com proves end to end per validar la integració completa del sistema.

Les proves s'han executat un cop l'aplicació ja ha estat publicada, mitjançant els URL:

- Web client: <https://okonofit.vercel.app/>
- Web gestor: <https://okonofit-admin.vercel.app/>

Per poder provar la web d'administrador es pot accedir amb l'usuari:

- Correu: [admin@gmail.com](mailto:admin@gmail.com)
- Contraseña: 123456Az

## Exemple de Prova bàsica

Prova-01: Accedir amb un usuari registrat

**Descripció:** Aquesta prova verifica que un usuari registrat pot accedir correctament a l'aplicació mitjançant el formulari de login.

**Precondició:** Cap.

**Passos a seguir:**

1. Accedir a la pantalla de login.
2. Omplir el formulari de login amb les credencials de l'usuari registrat.
3. Apretar el botó *Iniciar sesión*.

**Postcondició:** L'usuari és redirigit a la pàgina principal i té accés a la icona d'usuari de la barra de navegació.

**Validació:** Correcte.

**Errors associats:**

- Contrasenya incorrecta: Correcte.
- Usuari inexistent: Correcte.

A més d'aquestes proves unitàries i de funcionalitats específiques, s'han fet proves end to end (E2E) per verificar el correcte funcionament de la web en conjunt. Aquestes proves inclouen tot el flux des del registre de nous usuaris, la navegació pel catàleg de productes, l'addició de productes al carret de la compra, la finalització de comandes i la interacció amb el sistema de suport per part de la web de clients. També s'han dut a terme proves exclusives en la web de gestors.

## Exemple de Prova End to End (E2E)

Prova End-to-End 02: Donar permisos d'administrador

### 1. Iniciar sessió amb usuari no administrador

- Accedir a la pantalla de login de la web d'administrador.
  - Omplir el formulari de login amb un usuari no administrador.
  - Apretar botó *Iniciar sesión*.
  - Visualitzar error Usuario no autorizado
  - Validació: Correcte
- 2. Iniciar sessió amb usuari administrador**
- Omplir el formulari de login.
  - Apretar botó *Iniciar sesión*.
  - Validació: Correcte
- 3. Obtenir dades usuari no administrador**
- Afegir id usuari no administrador
  - Apretar botó *Obtener datos del usuario*
  - Validació: Correcte
- 4. Donar permisos administrador**
- Seleccionar checkbox *Adminsitrador*
  - Apretar botó *Guardar cambios*
  - Validació: Correcte
- 5. Tancar sessió**
- Seleccionar icone d'usuari
  - Apretar botó *Cerrar sesión*.
  - Validació: Correcte
- 6. Iniciar sessió amb nou usuari administrador**
- Omplir el formulari de login.
  - Apretar botó *Iniciar sesión*.
  - Accedir a la web d'administrador
  - Validació: Correcte

## Annex de Proves

Per a complementar la documentació de les proves realitzades, s'ha inclòs un annex amb el llistat complet d'aquestes.

## Resultats

Després de dur a terme les diverses proves sobre la web, es poden extreure les següents conclusions generals:

### Correcte Funcionament de les Funcionalitats Bàsiques

Les proves realitzades han demostrat que les funcionalitats bàsiques de la plataforma funcionen correctament. Això inclou el registre de nous usuaris, l'inici de sessió, el tancament de sessió, la visualització i modificació del perfil d'usuari, així com l'accés al catàleg de productes i la pàgina principal. Aquestes operacions es duen a terme sense errors.

## **Maneig Efectiu dels Errors**

Les proves han confirmat que el sistema respon adequadament a situacions com credencials incorrectes, intents d'accés no autoritzats per part d'usuaris no administradors, i errors en el registre d'usuaris.

## **Funcionalitats Avançades Comprovades**

Les funcionalitats avançades, com l'addició de productes a la cistella de compra, la formalització de comandes, i la gestió de productes per part dels administradors, han estat verificades amb èxit. Les proves han demostrat que aquestes operacions es poden realitzar sense inconvenients.

## **Necessitat de Millores i Futurs Desenvolupaments**

Algunes proves han revelat àrees que requereixen millores futures. Per exemple, la funcionalitat de modificació de comandes no està implementada actualment a la web d'administració, i la gestió de tiquets de suport es realitza fora de la plataforma.

## **Proves End-to-End Completes**

Les proves end-to-end han validat que els fluxos de treball complets, com el registre i la compra, funcionen de manera coherent des del principi fins al final. Això inclou el registre d'un nou usuari, la selecció i compra de productes, i l'obertura de tickets de suport. Aquestes proves són essencials per assegurar que la plataforma ofereix una experiència d'usuari contínua i sense interrupcions.

En resum, els resultats de les proves realitzades mostren que la plataforma és funcional i robusta en les seves operacions bàsiques i avançades. Tot i que s'han identificat àrees per a millores futures, la plataforma actual proporciona una base sòlida per a les operacions de comerç electrònic de la marca, oferint una experiència d'usuari satisfactòria.

# A futur

Tot i que la web desenvolupada per a OkonoFit ja compta amb una base sòlida, encara no està completament preparada per ser llançada al públic general. Per tal de portar la plataforma a un estat òptim per al seu ús comercial, és necessari implementar diverses millores i funcionalitats addicionals. A continuació es detallen les principals coses en les quals es pot treballar per continuar desenvolupant la web i preparar-la per al seu llançament oficial.

## **Implementació de mètodes de pagament real**

Actualment, la web no compta amb un sistema de pagament real integrat. Una de les millores més importants és la integració d'un servei de pagament segur i fiable. Per a aquest propòsit, es proposa utilitzar Stripe, una plataforma de pagaments àmpliament utilitzada i compatible amb Firebase. Stripe permetrà processar pagaments de manera segura i eficient.

Aquesta implementació no estava segur de si incloure-la en aquest projecte, primer per desconeixement de si tindria el temps suficient per fer-ho i segon per inseguretat. Per afegir mètodes de pagament real, on els clients hagin d'incloure les seves dades bancàries vull estar molt segur de com es fa correctament i poder assegurar el servei. Per això, en cas de continuar amb el projecte considero oportú contactar amb algú més expert en el tema.

## **Noves formes de Login**

Una altra funcionalitat a implementar és la possibilitat de login mitjançant serveis externs com Google. Firebase Auth facilita aquesta integració, permetent als usuaris iniciar sessió amb els seus comptes de Google de manera ràpida i segura. Aquesta funcionalitat simplifica el procés de registre i inici de sessió.

## **Vista de preferits**

Per millorar l'experiència de l'usuari, es podria afegir una funcionalitat de preferits. Aquesta funcionalitat permetria als usuaris guardar productes que els interessin per accedir-hi fàcilment més tard. La implementació d'aquesta funcionalitat requereix la creació d'una nova vista de preferits i l'opció de guardar i eliminar productes preferits des del catàleg de productes.

## **Millora de la web d'administrador**

Una altra àrea clau per al desenvolupament futur és la millora de la web d'administrador. Actualment, els tiquets de suport es gestionen a través del correu electrònic. Per oferir una experiència d'usuari més integrada i eficient, es podria implementar un sistema de ticketing dins de la mateixa plataforma. Això permetria

als usuaris crear, revisar i gestionar els seus tiquets de suport directament des de la web. Per als administradors, s'afegirien funcionalitats per gestionar els tiquets des del panell d'administració, facilitant la comunicació i la resolució de problemes.

A més, és essencial implementar la gestió de les comandes des del panell d'administració, una funcionalitat que actualment no està implementada i que és 100% necessària per al correcte funcionament de la botiga online. Aquesta funcionalitat permetria als administradors revisar, actualitzar i gestionar les comandes de manera eficient.

### **Millores en el disseny UX**

Finalment, és recomanable col·laborar amb un expert en UX per millorar el disseny visual de l'aplicació. Un disseny UX ben elaborat pot millorar significativament l'experiència de l'usuari, fent la navegació més intuïtiva i atractiva. Això ajudaria a dissenyar un apartat de contingut a la web, sent una de les opcions que agradava a la gent de OkonoFit i que finalment no s'ha implementat en aquest projecte.

Encara que la plataforma actual cobreix moltes de les necessitats inicials, aquestes millores són essencials per assegurar que la web compleixi amb els estàndards comercials i ofereixi una bona experiència d'usuari. Implementar aquestes funcionalitats i millores prepararà la web per al seu desplegament al públic.

# Conclusions

Aquest projecte ha complert l'objectiu principal de crear una aplicació web des de zero. Tot i que l'aplicació encara no està preparada per ser llançada al públic, s'han assolit tots els objectius específics proposats al començament del projecte, exceptuant alguna funcionalitat en específic.

En el procés de desenvolupament, he aconseguit millorar significativament les meves habilitats en React i he après a utilitzar Flask per crear APIs. Aquestes millores tècniques no només han contribuït, en el que considero, l'èxit del projecte, sinó que també han enriquit les meves competències com a desenvolupador web.

No obstant això, no tot ha estat un camí fàcil. La falta d'organització i dedicació durant el semestre anterior, va endarrerir el progrés del projecte i la finalització d'aquest. No obstant això, aquesta experiència m'ha permès aprendre dels errors i millorar de cara al futur.

El projecte m'ha ajudat a comprendre la importància de planificar adequadament cada fase del desenvolupament, establint objectius clars i realistes i mantenint una rutina de treball constant. Com comentava, aquest aprenentatge serà fonamental per a futurs projectes, tant acadèmics com professionals.

En resum, aquest projecte ha estat una experiència de creixement tant tècnic com personal. Tot i les dificultats inicials, he aconseguit desenvolupar una aplicació web funcional i he adquirit habilitats valuoses que m'ajudaran en la meua carrera com a enginyer informàtic. Amb les millores proposades a futur, en cas que OkonoFit ho consideri, la plataforma podrà evolucionar i convertir-se en una eina efectiva.

# Bibliografia

## **Vue.js - Official Guide**

- Descripció: Guia oficial de Vue.js, que ofereix una completa introducció a aquest popular framework de JavaScript.
- URL: <https://vuejs.org/guide/introduction.html>

## **Tokio School - ¿Qué es Angular?**

- Descripció: Explicació detallada i accessible sobre Angular, un framework avançat per a desenvolupament web.
- URL: <https://www.tokioschool.com/noticias/que-es-angular/>

## **SDI - Tipos de Comercio Electrónico**

- Descripció: Anàlisi dels diferents models i tipus de comerç electrònic i les seves característiques.
- URL: <https://www.sdi.es/actualidad/tipos-de-comercio-electronico/>

## **Amazon - ¿Qué es el E-commerce?**

- Descripció: Un article d'Amazon que ofereix una perspectiva sobre què és el comerç electrònic.
- URL: <https://sell.amazon.com/es/learn/what-is-ecommerce>

## **Clavei - Ventajas de un E-commerce**

- Descripció: Blog que discuteix els avantatges del comerç electrònic davant dels comerços tradicionals.
- URL: <https://www.clavei.es/blog/ventajas-de-un-ecommerce-frente-a-los-comercios-tradicionales/>

## **Gadae - 5 Formas de Crear Tu Tienda Online**

- Descripció: Consells pràctics i estratègies per crear una botiga en línia exitosa.
- URL: <https://www.gadae.com/blog/5-formas-de-crear-tu-tienda-online/>

## **Qindel Group - ¿Conoces React y sus beneficios?**

- Descripció: Article que proporciona una visió general de React i els seus beneficis en el desenvolupament web.

- URL: <https://www.qindiel.com/conoces-react-y-sus-beneficios-si-no-es-asi-te-lo-explicamos/>

### **BOE - Regulación de Comercio Electrónico en España**

- Descripció: Text consolidat del Butlletí Oficial de l'Estat sobre la regulació del comerç electrònic a Espanya.
- URL: <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf>

### **Flowbite-React - UI Components**

- Descripció: Biblioteca de components UI per a React, facilitant el desenvolupament d'interfícies d'usuari.
- URL: <https://www.flowbite-react.com/>

### **Tailwind UI - Componentes de Interfaz de Usuario**

- Descripció: Galeria de components de UI dissenyats amb Tailwind CSS, útils per al desenvolupament de frontends moderns.
- URL: <https://tailwindui.com/components/preview>

### **Next.js - Framework de React**

- Descripció: Documentació oficial de Next.js, un framework de React per a producció.
- URL: <https://nextjs.org/>

### **Flask-RESTful - Documentación**

- Descripció: Guia y documentació oficial de Flask-RESTful, una extensió para la creació de APIs en Flask.
- URL: <https://flask-restful.readthedocs.io/en/latest/>

### **Firestore - Documentación Oficial**

- Descripció: Documentació completa de Firestore, plataforma de desenvolupament d'aplicacions mòbils i web.
- URL: <https://firebase.google.com/docs?hl=es-419>

### **BBVA - Metodología Agile**

- Descripció: Article de BBVA sobre la metodologia Agile i com ha revolucionat les formes de treball.

- URL:  
<https://www.bbva.com/es/innovacion/metodologia-agile-la-revolucion-las-formas-trabajo/>

### **HubSpot - Metodología Agile en Marketing**

- Descripció: Explicació de com s'aplica la metodologia Agile en l'àmbit del màrqueting.
- URL: <https://blog.hubspot.es/marketing/metodologia-agile>

### **Innovadeluxe - Ventajas y Desventajas del E-commerce**

- Descripció: Anàlisi dels avantatges i desavantatges del comerç electrònic.
- URL: <https://www.innovadeluxe.com/ventajas-y-desventajas-del-comercio-electronico/>

### **Bambu Mobile - Tecnologías para Desarrollo Web**

- Descripció: Descripció de les tecnologies més actuals utilitzades en el desenvolupament web.
- URL: <https://www.bambu-mobile.com/tecnologias-mas-actuales-para-desarrollo-web/>

### **Inabaweb - Bases de Datos Populares en Desarrollo Web**

- Descripció: Una revisió de les bases de dades més populars utilitzades en el desenvolupament web.
- URL: <https://www.inabaweb.com/bases-de-datos-mas-populares-en-el-desarrollo-web/>

### **Redigital - Marco Normativo Aplicable a tu Website**

- Descripció: Guia sobre el marc normatiu aplicable a llocs web, incloent-hi aspectes legals i de compliment.
- URL: <https://redigital.economistas.es/marco-normativo-aplicable-a-tu-web-site>

### **PythonES - Solicitudes en Flask: GET y POST**

- Descripció: Explicació pràctica de com manejar sol·licituds GET i POST en Flask.
- URL: <https://pythones.net/solicitudes-en-flask-get-post/>

### **Agile Alliance**

- Descripció: Què és Agile? Agile Alliance.
- URL: <https://www.agilealliance.org/agile101/>

### **ReactJS Official Documentation**

- Descripció: React - Una biblioteca de JavaScript per crear interfícies d'usuari.
- URL: <https://reactjs.org/>

### **Vercel**

- Descripció: Documentació de Next.js.
- URL: <https://nextjs.org/docs>

### **Tailwind Labs**

- Descripció: Documentació de Tailwind CSS.
- URL: <https://tailwindcss.com/docs>

### **Google Firebase**

- Descripció: Documentació de Firebase.
- URL: <https://firebase.google.com/docs>

### **Stripe**

- Descripció: Documentació de Stripe Payments.
- URL: <https://stripe.com/docs/payments>

### **Postman**

- Descripció: Plataforma API de Postman.
- URL: <https://www.postman.com/>

### **Heroku**

- Descripció: Plataforma d'aplicacions al núvol de Heroku.
- URL: <https://www.heroku.com/>

### **Shopify**

- Descripció: Ecommerce per a tothom.
- URL: <https://www.shopify.com/learn>

### **BigCommerce**

- Descripció: Recursos de BigCommerce.
- URL: <https://www.bigcommerce.com/resources/>

## Nielsen Norman Group

- Descripció: Usabilitat 101: Introducció a la usabilitat.
- URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

## Cypress

- Descripció: Documentació de Cypress.
- URL: <https://docs.cypress.io/>

## Selenium

- Descripció: Documentació de Selenium.
- URL: <https://www.selenium.dev/documentation/>

## Posicionamiento Web Systems

- Descripció: Passos per dissenyar una aplicació web.
- URL: [https://www.posicionamientoweb.systems/tecnologia/pasos-disenar-aplicacion-web/#Tipos\\_de\\_desarrollo\\_de\\_aplicaciones\\_web](https://www.posicionamientoweb.systems/tecnologia/pasos-disenar-aplicacion-web/#Tipos_de_desarrollo_de_aplicaciones_web)

## Venngage

- Descripció: Diagrames de casos d'ús.
- URL: <https://es.venngage.com/blog/diagramas-de-casos-de-uso/>

## freeCodeCamp

- Descripció: Routing en Next.js.
- URL: <https://www.freecodecamp.org/news/routing-in-nextjs/>

## Dev.to

- Descripció: Com usar React Context de manera efectiva.
- URL: <https://dev.to/jereef/como-usar-react-context-de-manera-efectiva-1h6l>

## Telerik

- Descripció: Autenticació de Firebase utilitzant token personalitzat.
- URL: <https://www.telerik.com/blogs/firebase-authentication-using-custom-token>

# Annexos

## Entrevista OkonoFit

### Objectius de la Marca

- **Quins són els objectius principals de la marca?**

Fer-nos un nom en el sector, inspirar un estil de vida actiu i saludable, oferir productes de alta qualitat que combinen comoditat, funcionalitat i estil.

- **Com descriuria la identitat i els valors de la marca?**

La nostra identitat es tracta en fomentar un estil de vida, no en vestir una marca de roba. L'intenció es relacionar OkonoFit amb ser una persona que es cuida i activa.

- **Qui és el seu públic objectiu i com creu que la web pot servir millor per a les seves necessitats i expectatives?**

El nostre públic objectiu són homes sobretot els que es troben entre el 17 i 35 anys que valoren la salut, el benestar i l'activitat física. Creiem que la nostra web pot oferir contingut educatiu i motivador, a més de mostrar els nostres productes de forma atractiva i informativa.

- **Hi ha alguna història o filosofia darrere de la marca que us agradaria que es reflectís al lloc web?**

La marca va néixer de la creença que tot és possible amb determinació i esforç. Volem que aquesta filosofia impregni el lloc web a través de històries inspiradores, consells d'entrenament i una comunitat en línia on els usuaris puguin compartir els seus èxits.

### Funcionalitats

- **Quines funcionalitats específiques considereu indispensables per al lloc web?**

Busquem una tenda online simple i efectiva, de fàcil ús i que inclogui una secció d'atenció al client. S'ha de poder filtrar els productes segons les seves característiques.

I ens agradaria incloure un apartat de contingut per poder compartirlo amb el client.

- **Teniu preferències sobre l'estructura i la navegació del lloc web? Per exemple, les seccions que ha de tenir.**

Preferim una estructura clara i simple, amb filtres específics per ajudar al client en la seva busqueda. La navegació ha de ser intuïtiva, amb un menú principal fàcil d'accedir des de qualsevol pàgina.

- **Com planegeu gestionar les actualitzacions del lloc web i el manteniment del contingut?**

Ens agradaria tenir un sistema de gestió per actualitzar el contingut de la web.

## **Màrqueting**

- **Quina és la seva estratègia per atraure trànsit al lloc web?**

Una combinació de SEO, contingut de valor a través de les nostres xarxes i publicitat pagada.

## **Conclusions**

Les conclusions extretes són que OkonoFit busca destacar-se en el mercat oferint més que simples productes esportius. La seva identitat es basa en fomentar un estil de vida actiu i saludable, i la marca vol ser percebuda com un símbol d'aquest estil de vida.

El seu públic principal són homes entre 17 i 35 anys que valoren la salut, el benestar i l'activitat física.

La web ha de ser simple però efectiva. Una botiga en línia amb una navegació intuïtiva i una estructura clara és essencial per guiar el client en la seva compra. Els filtres de producte permeten una experiència de cerca efectiva, i una secció d'atenció al client i de contingut ajuden a establir una relació més profunda amb els clients.

## Casos d'ús

En aquest apartat s'ha realitzat una explicació detallada de tots els casos d'ús que són necessaris per un correcte funcionament de la plataforma.

Per cada cas d'ús hi trobem:

- **Nom:** Nom amb el que identifiquem el cas d'ús.
- **Actor:** Indica el tipus d'usuari que utilitza la web.
- **Objectiu:** Finalitat que busca realitzar l'actor
- **Precondicions:** Condicions que s'han d'haver complert anteriorment per realitzar l'operació.
- **Postcondicions:** Indica l'estat en que es queda el sistema un cop realitzada l'operació.
- **Flux:** Descriu les fases que formen el cas d'ús
- **Cas d'ús alternatiu:** Descriu situacions que es poden donar en cas de no realitzar correctament els passos indicats.

### C-01 Registre

- **Nom:** Registre
- **Actor:** Visitant
- **Objectiu:** Registrar-se a l'aplicació
- **Precondicions:** -
- **Postcondicions:** L'usuari té un compte creat
- **Flux:**
  - L'usuari introdueix el nom, correu electrònic i contrasenya al formulari de registre.
  - Es redirigeix a l'usuari a la pàgina de login.
- **Casos Alternatius:**
  - L'usuari no introdueix les dades correctament al formulari.
    - L'usuari introdueix un correu electrònic ja registrat.
    - L'usuari no introdueix un format correcte de correu electrònic.
    - L'usuari no introdueix un format correcte de contrasenya.

### C-02 Login

- **Nom:** Accés a la plataforma web
- **Actor:** Visitant
- **Objectiu:** Accedir a la web
- **Precondicions:** L'usuari ha d'haver-se registrat
- **Postcondicions:** L'usuari accedeix a la web
- **Flux:**
  - L'usuari introdueix el correu electrònic i contrasenya amb els que s'ha registrat.
  - El sistema comprova que les dades siguin correctes.

- Es redirigeix a l'usuari a la pàgina principal.
- **Casos Alternatius:**
  - L'usuari no introdueix les seves dades correctament al formulari.
    - L'usuari introdueix un correu electrònic no registrat.
    - L'usuari no introdueix la contrasenya correcte.

#### C-03 Login com administrador

- **Nom:** Accés a la plataforma web amb els beneficis d'administrador
- **Actor:** Administrador
- **Objectiu:** Accedir les funcionalitats d'administrador
- **Precondicions:** L'usuari ha de tenir un compte amb permisos d'administrador
- **Postcondicions:** L'usuari accedeix a la web amb les funcionalitats d'administrador visibles
- **Flux:**
  - L'usuari introdueix el seu correu electrònic i contrasenya.
  - El sistema comprova que les dades siguin correctes.
  - Al llistat d'opcions de la barra de navegació apareix un nou camp Administrador.
- **Casos Alternatius:**
  - L'usuari no introdueix les seves dades correctament al formulari.
    - L'usuari introdueix un correu electrònic no registrat.
    - L'usuari no introdueix la contrasenya correcte.

#### C-04 Tancar sessió

- **Nom:** Tancar la sessió
- **Actor:** Client
- **Objectiu:** Tancar la sessió iniciada
- **Precondicions:** L'usuari ha d'estar loguejat
- **Postcondicions:** L'usuari surt de la web
- **Flux:**
  - L'usuari selecciona l'opció de tancar sessió de la barra de navegació.
  - Es redirigeix a l'usuari a la pàgina d'inici de sessió..
- **Casos Alternatius:**
  - L'usuari tanca la web sense haver tancat sessió.

#### C-05 Veure dades d'un usuari

- **Nom:** Veure dades d'un usuari
- **Actor:** Client
- **Objectiu:** Veure les seves dades associades d'un usuari
- **Precondicions:** -
- **Postcondicions:** L'usuari ha pogut veure les seves dades

- **Flux:**
  - L'usuari selecciona l'opció perfil a la barra de navegació.
  - Es redirigeix a l'usuari a la vista de perfil.
  - L'usuari pot observar les seves dades.
- **Casos Alternatius:**
  - -

#### C-06 Actualitzar el compte

- **Nom:** Gestionar i actualitzar les dades del compte
- **Actor:** Client
- **Objectiu:** Modificar els camps del perfil de l'usuari
- **Precondicions:** L'usuari ha d'estar loguejat
- **Postcondicions:** L'usuari actualitza les dades del seu compte
- **Flux:**
  - L'usuari accedeix al seu perfil a través de la barra de navegació.
  - Es redirigeix l'usuari a la pàgina mostrant les seves dades.
  - L'usuari pot omplir o modificar els camps que vulgui.
  - S'actualitza les dades de l'usuari.
- **Casos Alternatius:**
  - L'usuari omple els camps pero no guarda els canvis.

#### C-07 Veure un producte

- **Nom:** Veure un producte
- **Actor:** Visitant
- **Objectiu:** Observar els detalls d'un producte
- **Precondicions:** -
- **Postcondicions:** L'usuari ha pogut observar els detalls d'un producte
- **Flux:**
  - L'usuari selecciona un producte del catàleg.
  - Es redirigeix a l'usuari a la vista del producte.
  - L'usuari pot observar els detalls del producte.
- **Casos Alternatius:**
  - -

#### C-08 Veure el catàleg

- **Nom:** Veure el catàleg
- **Actor:** Visitant
- **Objectiu:** Observar el catàleg sencer de la tenda
- **Precondicions:** -
- **Postcondicions:** L'usuari ha pogut veure tots els productes de la tenda
- **Flux:**

- L'usuari navega a la vista del catàleg.
- Es redirigeix a l'usuari a la vista del catàleg.
- L'usuari pot observar els productes.
- **Casos Alternatius:**
  - -

#### C-09 Gestió usuaris

- **Nom:** Modificar el rol d'un usuari
- **Actor:** Administrador
- **Objectiu:** Gestionar usuaris - modificar el rol
- **Precondicions:** L'usuari administrador ha d'estar loguejat
- **Postcondicions:** El rol d'un usuari s'ha modificat
- **Flux:**
  - L'usuari selecciona l'opció Administrador a la barra de navegació.
  - Es redirigeix a administrador a la vista Administrador.
  - Introdueix l'identificador de l'usuari a modificar.
  - Modifica el rol del usuari
- **Casos Alternatius:**
  - El sistema no ha pogut modificar el usuari, es mostra missatge d'error.

#### C-10 Gestió productes

- **Nom:** Modificar dades d'un producte
- **Actor:** Administrador
- **Objectiu:** Modificar les dades d'un producte
- **Precondicions:** L'usuari administrador ha d'estar loguejat
- **Postcondicions:** S'ha modificat les dades del producte
- **Flux:**
  - L'usuari selecciona l'opció Administrador a la barra de navegació.
  - Es redirigeix a administrador a la vista Administrador.
  - L'administrador busca el producte.
  - Modifica el stock del producte.
- **Casos Alternatius:**
  - El sistema no ha pogut modificar el producte, es mostra missatge d'error.

#### C-11 Gestió comandes

- **Nom:** Modificar estat d'una comanda
- **Actor:** Administrador
- **Objectiu:** Modificar l'estat d'una comanda
- **Precondicions:** L'usuari administrador ha d'estar loguejat
- **Postcondicions:** S'ha modificat l'estat d'una comanda

- **Flux:**
  - L'usuari selecciona l'opció Administrador a la barra de navegació.
  - Es redirigeix a administrador a la vista Administrador.
  - L'administrador busca una comanda.
  - Modifica el seu estat.
- **Casos Alternatius:**
  - El sistema no ha pogut modificar la comanda, es mostra missatge d'error.

#### C-12 Veure la cistella de compra

- **Nom:** Veure la cistella de compra
- **Actor:** Client
- **Objectiu:** Veure la cistella de compra
- **Precondicions:** L'usuari ha d'estar loguejat i ha d'haver afegit un producte a la cistella
- **Postcondicions:** S'ha visualitzat la cistella
- **Flux:**
  - L'usuari selecciona la icona de la cistella a la barra de navegació.
  - S'obre el panel lateral amb la cistella.
  - L'usuari pot visualitzar la cistella.
- **Casos Alternatius:**
  - El sistema no ha pogut modificar la comanda, es mostra missatge d'error.

#### C-13 Afegir un producte a la cistella de compra

- **Nom:** Afegir producte a la cistella
- **Actor:** Client
- **Objectiu:** Afegir un producte a la cistella de compra
- **Precondicions:** L'usuari ha d'estar loguejat
- **Postcondicions:** S'ha afegit un nou producte a la cistella
- **Flux:**
  - L'usuari selecciona un producte del catàleg.
  - Es redirigeix a l'usuari a la vista del producte.
  - L'usuari selecciona una talla disponible del producte.
  - L'usuari afegix el producte a la cistella.
- **Casos Alternatius:**
  - El sistema no ha pogut modificar la comanda, es mostra missatge d'error.

#### C-14 Eliminar un producte de la cistella de compra

- **Nom:** Eliminar un producte de la cistella

- **Actor:** Client
- **Objectiu:** Eliminar un producte de la cistella de compra
- **Precondicions:** L'usuari ha d'estar loguejat i tenir productes a la cistella
- **Postcondicions:** S'ha eliminat un producte de la cistella
- **Flux:**
  - L'usuari obre la cistella de la compra.
  - L'usuari elimina el producte que desitja.
- **Casos Alternatius:**
  - El sistema no ha pogut eliminar el producte, es mostra missatge d'error.

#### C-15 Finalitzar una compra

- **Nom:** Finalitzar una compra
- **Actor:** Client
- **Objectiu:** Omplir les dades restants i efectuar la compra
- **Precondicions:** L'usuari ha d'estar loguejat i tenir productes a la cistella
- **Postcondicions:** S'ha efectuat una comanda
- **Flux:**
  - L'usuari obre la cistella de la compra.
  - L'usuari selecciona finalitzar la compra.
  - Es redirigeix a l'usuari a la vista de finalitzar comanda.
  - L'usuari omple el formulari de dades d'entrega.
  - L'usuari fa el pagament.
- **Casos Alternatius:**
  - L'usuari no introdueix direcció d'entrega o no omple tots els camps.
  - El pagament no ha sigut efectiu.

#### C-16 Revisar comandes

- **Nom:** Revisar comandes
- **Actor:** Client
- **Objectiu:** Veure l'historial de comandes i el seus detalls
- **Precondicions:** L'usuari ha d'estar loguejat i haver efectuat una comanda
- **Postcondicions:** L'usuari ha pogut veure el seu historial de comandes
- **Flux:**
  - L'usuari selecciona Comandes a través de la barra de navegació.
  - Es redirigeix a l'usuari a la vista historial de comandes.
  - L'usuari pot observar les diferents comandes efectuades junt amb el seu estat.
- **Casos Alternatius:**
  - L'usuari no ha realitzat mai una comanda, es mostrarà un missatge informatiu.

### C-17 Modificar contrasenya

- **Nom:** Modificar contrasenya
- **Actor:** Client
- **Objectiu:** Modificar la contrasenya per iniciar sessió
- **Precondicions:** L'usuari ha de tenir un usuari creat
- **Postcondicions:** L'usuari pot iniciar sessió amb la nova contrasenya
- **Flux:**
  - L'usuari selecciona l'opció *¿Has olvidado tu contraseña?* a la pantalla de login.
  - El sistema envia un correu a l'usuari amb les indicacions per modificar la contrasenya.
  - L'usuari modifica la contrasenya.
- **Casos Alternatius:**
  - -

### C-18 Filtrar productes

- **Nom:** Filtrar productes
- **Actor:** Visitant
- **Objectiu:** Filtrar els productes del catàleg
- **Precondicions:** -
- **Postcondicions:** L'usuari pot veure els productes que compleixen els seus requisits
- **Flux:**
  - L'usuari navega a la vista del catàleg.
  - Es redirigeix a l'usuari a la vista del catàleg.
  - L'usuari selecciona una opció del panel de filtres.
  - L'usuari pot observar els productes filtrats.
- **Casos Alternatius:**

### C-19 Modificar cistella

- **Nom:** Modificar cistella
- **Actor:** Client
- **Objectiu:** Modificar les unitats d'un producte de la cistella
- **Precondicions:** L'usuari ha de haver iniciat sessió i tenir un producte a la cistella
- **Postcondicions:** S'ha modificat la cistella
- **Flux:**
  - L'usuari obre la cistella.
  - L'usuari modifica les unitats d'un producte a la cistella.

## C-20 Crear tiquet de suport

- **Nom:** Crear tiquet de suport
- **Actor:** Client
- **Objectiu:** Crear un tiquet de suport en cas de trobar-se qualsevol problema
- **Precondicions:** L'usuari ha d'haver realitzat una comanda
- **Postcondicions:** L'usuari ha enviat un tiquet
- **Flux:**
  - L'usuari obre el llistat de pedidos.
  - L'usuari selecciona *Abrir ticket*.
  - L'usuari omple el formulari envia el ticket.

## Endpoints

Registrar usuari -> POST /user

### Entrada:

- Correu
- Contrasenya
- Nom
- Cognom

### Sortida:

- Token

### Errors:

- Usuari ja existent
- Contrasenya no compleix els requisits (controlar des del frontend)

**Descripció:** Aquest endpoint permet als nous usuaris registrar-se a l'aplicació. L'usuari ha de proporcionar el seu correu electrònic, contrasenya, nom i cognom. Si el registre és correcte, es genera i retorna un token que serà utilitzat per a l'autenticació en futures peticions.

Iniciar sessió -> POST /login

### Entrada:

- Correu
- Contrasenya

### Sortida:

- Token

### Errors:

- Usuari inexistent
- Contrasenya incorrecta

**Descripció:** Aquest endpoint permet als usuaris iniciar sessió a l'aplicació. Requereix que l'usuari proporcioni el seu correu electrònic i contrasenya. Si les credencials són correctes, el sistema genera i retorna un token d'usuari que serà utilitzat per a l'autenticació en futures peticions. En cas d'error, es retorna un missatge indicant si l'usuari no existeix o si la contrasenya és incorrecta.

Tancar sessió -> POST /logout

**Entrada:**

- Token

**Sortida:**

- Missatge de confirmació

**Errors:**

- Token no proporcionat
- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet als usuaris tancar sessió a l'aplicació.

Iniciar sessió com a administrador -> POST /admin

**Entrada:**

- Correu
- Contrasenya

**Sortida:**

- Token

**Errors:**

- Usuari inexistent
- Contrasenya incorrecta
- Usuari no administrador

**Descripció:** Aquest endpoint permet als administradors iniciar sessió a l'aplicació. Es requereix que l'usuari proporcioni el seu correu electrònic i contrasenya. Si les credencials són correctes i l'usuari és un administrador, el sistema genera i retorna un token d'usuari que serà utilitzat per a l'autenticació en futures peticions.

Obtenir dades d'usuari -> GET /user

**Entrada:**

- Token

**Sortida:**

- Nom

- Cognom
- País
- Calle
- Piso
- Codi postal
- Ciutat
- Província
- Telèfon
- Rol

**Errors:**

- Token incorrecte/caducat
- Usuari no trobat

**Descripció:** Aquest endpoint permet obtenir les dades d'un usuari registrat. L'usuari ha de proporcionar un token vàlid. Si el token és vàlid, es retornen les dades de l'usuari, incloent el nom, cognom i la seva adreça completa.

Actualitzar dades d'usuari -> PUT /user

**Entrada:**

- Token
- Nom
- Cognom
- País
- Calle
- Piso
- Codi postal
- Ciutat
- Província
- Telèfon
- Rol

**Sortida:**

- Missatge de confirmació

**Errors:**

- Token incorrecte/caducat
- Usuari inexistent

**Descripció:** Aquest endpoint permet als usuaris actualitzar les seves dades personals. L'usuari ha de proporcionar un token vàlid i les dades que vol actualitzar.

Si el token és vàlid i les dades es processen correctament, es retorna un missatge de confirmació.

Eliminar usuari -> DELETE /user

**Entrada:**

- Correu

**Sortida:**

- Missatge de confirmació

**Errors:**

- Usuari inexistent

**Descripció:** Aquest endpoint permet a un usuari o a un administrador eliminar un compte de la base de dades.

Obtenir llistat de productes -> GET /products

**Entrada:**

- Cap

**Sortida:**

- Llistat de productes

**Descripció:** Aquest endpoint retorna un llistat de tots els productes disponibles a la botiga. No es requereix cap entrada específica.

Obtenir producte -> GET /product

**Entrada:**

- IdProducte

**Sortida:**

- Preu
- Talles
- Color
- Nom
- imageSrc
- imageAlt
- Imatges

- Descripció
- href
- Materials
- Categoria
- Id

**Errors:**

- Producte inexistent

**Descripció:** Aquest endpoint retorna la informació detallada d'un producte específic identificat pel seu ID. Si el producte existeix, es retorna tota la informació relacionada amb ell. En cas que no s'especifiqui cap ID, es retornen tots els productes de la base de dades.

Eliminar producte -> DELETE /product

**Entrada:**

- IdProducte

**Sortida:**

- Missatge de confirmació

**Errors:**

- Producte inexistent

**Descripció:** Aquest endpoint elimina un producte de la base de dades.

Afegir producte -> POST /product

**Entrada:**

- Preu
- Talles
- Color
- Nom
- imageSrc
- imageAlt
- Imatges
- Descripció
- href
- Materials
- Categoria
- Id

**Sortida:**

- Missatge de confirmació

**Descripció:** Aquest endpoint permet afegir un producte i la seva informació a la base de dades.

Actualitzar producte -> PUT /product

**Entrada:**

- Preu
- Talles
- Color
- Nom
- imageSrc
- imageAlt
- Imatges
- Descripció
- href
- Materials
- Categoria
- Id

**Sortida:**

- Missatge de confirmació

**Descripció:** Aquest endpoint permet actualitzar la informació d'un producte existent. L'administrador ha de proporcionar els nous detalls del producte.

Obtenir comandes -> GET /orders

**Entrada:**

- Token

**Sortida:**

- Llistat de comandes

**Errors:**

- Token incorrecte/caducat

**Descripció:** Aquest endpoint retorna un llistat de les comandes realitzades per un usuari especificat pel seu correu electrònic, obtingut a través del token. L'usuari ha de proporcionar un token vàlid.

Afegir comanda -> POST /orders

**Entrada:**

- Token
- Items
- Preu total

**Sortida:**

- Missatge de confirmació

**Errors:**

- Token incorrecte/caducat

**Descripció:** Aquest endpoint crea una nova comanda associada a un usuari a la base de dades, li estableix un ID de comanda i guarda la data d'aquesta juntament amb la resta d'informació aportada.

Actualitzar comanda -> PUT /orders

**Entrada:**

- IdComanda
- Token
- Estat

**Sortida:**

- Missatge de confirmació

**Errors:**

- Comanda inexistent
- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet actualitzar l'estat d'una comanda específica. L'usuari ha de proporcionar un token vàlid i el nou estat de la comanda. Si el token és vàlid i la comanda existeix, es retorna l'estat actualitzat.

Restablir contrasenya -> POST /reset

**Entrada:**

- Correu

**Sortida:**

- Missatge de confirmació

**Errors:**

- Usuari inexistent

**Descripció:** Aquest endpoint envia al correu indicat un enllaç per poder restablir la contrasenya de l'usuari.

Obtenir cistella -> GET /shoppingcart

**Entrada:**

- Token

**Sortida:**

- Llistat de productes

**Errors:**

- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet obtenir el contingut de la cistella de la compra d'un usuari especificat pel seu correu electrònic. L'usuari ha de proporcionar un token vàlid per poder extreure el correu associat al token.

Afegir a cistella -> POST /shoppingcart

**Entrada:**

- Items
- Token

**Sortida:**

- Llistat de productes

**Errors:**

- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet afegir productes a la cistella de la compra d'un usuari. L'usuari ha de proporcionar un token vàlid i els detalls dels productes a afegir. Si el token és vàlid i els productes estan disponibles, es retorna el nou contingut de la cistella.

Actualitzar cistella -> PUT /shoppingcart

**Entrada:**

- IdProducte
- Items
- Correu
- Token

**Sortida:**

- Llistat de productes

**Errors:**

- Producte exhaurit
- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet actualitzar el contingut de la cistella de la compra d'un usuari. L'usuari ha de proporcionar un token vàlid i els detalls dels productes a actualitzar. Si el token és vàlid i els productes estan disponibles, es retorna el nou contingut de la cistella.

Eliminar cistella -> DELETE /shoppingcart

**Entrada:**

- Token

**Sortida:**

- Missatge de confirmació

**Errors:**

- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet eliminar la cistella de compra d'un usuari.

Obtenir dades d'usuari com a administrador -> GET /admin/user

**Entrada:**

- Token
- Correu usuari

**Sortida:**

- Nom
- Cognom

- País
- Calle
- Piso
- Codi postal
- Ciutat
- Província
- Telèfon
- Rol

**Errors:**

- Token incorrecte/caducat
- Es requereix rol d'administrador
- Usuari no trobat

**Descripció:** Aquest endpoint permet a un administrador obtenir les dades d'un usuari registrat. L'administrador ha de proporcionar un token vàlid. Si el token és vàlid, es retornen les dades de l'usuari.

Actualitzar dades d'usuari com a administrador -> PUT /admin/user

**Entrada:**

- Token
- Correu usuari
- Nom
- Cognom
- País
- Calle
- 
- Piso
- Codi postal
- Ciutat
- Província
- Telèfon
- Rol

**Sortida:**

- Missatge de confirmació

**Errors:**

- Token incorrecte/caducat
- Usuari inexistent
- Es requereix rol d'administrador

**Descripció:** Aquest endpoint permet als administradors actualitzar les dades personals dels usuaris. L'administrador ha de proporcionar un token vàlid i les dades que vol actualitzar. Si el token és vàlid i les dades es processen correctament, es retorna un missatge de confirmació.

Obrir tiquet de suport -> POST /ticket

**Entrada:**

- Token
- Id comanda
- Títol
- Descripció
- Correu de contacte

**Sortida:**

- Missatge de confirmació

**Errors:**

- Token incorrecte/caducat

**Descripció:** Aquest endpoint permet obrir un tiquet de suport.

Obtenir tiquets de suport -> GET /ticket

**Entrada:**

- Token

**Sortida:**

- Llistat de tiquets

**Errors:**

- Token incorrecte/caducat
- Usuari sense tiquets oberts

**Descripció:** Aquest endpoint permet obtenir tots els tiquets associats a un usuari.

## Proves

### Prova-01

**Descripció:** Accedir amb un usuari registrat

**Precondició:** -

**Passos a seguir:**

1. Accedir a la pantalla de login.
2. Omplir el formulari de login.
3. Apretar botó *Iniciar sesión*.

**Postcondició:** L'usuari és redirigit a la pàgina principal, i té accés a la icona d'usuari de la barra de navegació.

**Validació:** Correcte

**Errors associats:**

- **Contrasenya incorrecte:** Correcte
- **Usuari inexistent:** Correcte

### Prova-02

**Descripció:** Accedir amb un usuari administrador

**Precondició:** -

**Passos a seguir:**

1. Accedir a la pantalla de login de la web d'administradors.
2. Omplir el formulari de login amb les dades d'un usuari administrador.
3. Apretar botó *Iniciar sesión*.

**Postcondició:** L'usuari pot accedir a la web d'administradors.

**Validació:** Correcte

**Errors associats:**

- **Contrasenya incorrecte:** Correcte
- **Usuari inexistent:** Correcte
- **Usuari no administrador:** Correcte

### Prova-03

**Descripció:** Registrar un nou usuari

**Precondició:** -

**Passos a seguir:**

1. Accedir a la pantalla de login.
2. Seleccionar el botó *¿No estas registrado? Hazlo aquí*
3. Omplir el formulari amb un nom, cognoms, correu i contrasenya.
4. Apretar botó *Registrarse*

**Postcondició:** L'usuari ha sigut creat i es redirigeix a la pàgina de login.

**Validació:** Correcte

**Errors associats:**

- **Contrasenya no compleix els requisits:** Correcte
- **Usuari ja existent:** Correcte

#### Prova-04

**Descripció:** Tancar sessió d'un usuari

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

1. Des de qualsevol pàgina de la web, seleccionar la icona d'usuari de la barra de navegació.
2. Seleccionar l'opció *Cerrar sesión*.

**Postcondició:** La sessió de l'usuari s'ha tancat i es redirigeix a la pàgina de login.

**Validació:** Correcte

#### Prova-05

**Descripció:** Accedir a la pàgina principal

**Precondició:** -

**Passos a seguir:**

1. Seleccionar la icona de la part superior esquerra.

**Postcondició:** Es redirigeix a l'usuari a la pàgina principal de la web.

**Validació:** Correcte

#### Prova-06

**Descripció:** Accedir a la pàgina de productes

**Precondició:** -

**Passos a seguir:**

1. Seleccionar *Ver catálogo* a la pàgina principal.

**Postcondició:** Es redirigeix a l'usuari a la pàgina on es mostra tot el catàleg de la web.

**Validació:** Correcte

#### Prova-07

**Descripció:** Accedir a la pàgina de individual d'un producte

**Precondició:** -

**Passos a seguir:**

1. Seleccionar *Ver catálogo* a la pàgina principal.
2. Selecciona un producte del catàleg.

**Postcondició:** Es redirigeix a l'usuari a la pàgina individual d'un producte.

**Validació:**

#### Prova-08

**Descripció:** Afegir un producte a la cistella

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

1. Seleccionar *Ver catálogo* a la pàgina principal.
2. Selecciona un producte del catàleg.
3. Seleccionar una talla del producte.
4. Clicar botó *Añadir a la cesta*.

**Postcondició:** S'ha afegit un nou producte a la cistella de compra.

**Validació:** Correcte

**Errors associats:**

- **Token incorrecte:** Correcte

#### Prova-09

**Descripció:** Accedir a la pàgina de perfil de l'usuari

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

1. Seleccionar l'opció *Perfil* de la barra de navegació.

**Postcondició:** S'ha redirigit a l'usuari a la pàgina de perfil.

**Validació:** Correcte

#### Prova-10

**Descripció:** Actualitzar perfil d'un usuari

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

1. Seleccionar l'opció *Perfil* de la barra de navegació.
2. Omple el formulari
3. Selecciona el botó *Guardar cambios*.

**Postcondició:** S'ha actualitzat el perfil de l'usuari i redirigit a la pàgina principal.

**Validació:** Correcte

**Errors associats:**

- **Token incorrecte:**Correcte

#### Prova-11

**Descripció:** Eliminar un usuari

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

4. Seleccionar l'opció *Perfil* de la barra de navegació.
5. Seleccionar el botó inferior *Eliminar usuari*

**Postcondició:** S'ha eliminar el usuari registrat i s'ha redirigir a la pantalla d'iniciar sessió.

**Validació:** Correcte

**Errors associats:**

- **Token incorrecte:** Correcte

Prova-11

**Descripció:** Veure historial de comandes

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

1. Seleccionar l'opció *Pedidos* de la barra de navegació.

**Postcondició:** S'ha redirigit a l'usuari a la pàgina de comandes, on es mostren, en cas de que existeixin, les comandes realitzades per l'usuari..

**Validació:** Correcte

Prova-12

**Descripció:** Obrir la cistella de la compra

**Precondició:** Tenir una sessió iniciada

**Passos a seguir:**

1. Seleccionar la icona de la cistella de la barra de navegació.

**Postcondició:** S'ha obert el panel que ensenya la cistella.

**Validació:** Correcte

Prova-13

**Descripció:** Eliminar un producte de la cistella

**Precondició:** Tenir una sessió iniciada i un producte a la cistella

**Passos a seguir:**

1. Seleccionar la icona de la cistella de la barra de navegació.
2. Clicar botó *Eliminar* del costat d'un producte de la cistella.

**Postcondició:** S'ha eliminat un producte de la cistella de compra.

**Validació:** Correcte

#### Prova-14

**Descripció:** Formalitzar una comanda

**Precondició:** Tenir una sessió iniciada i mínim un producte a la cistella

**Passos a seguir:**

1. Seleccionar la icona de la cistella de la barra de navegació.
2. Clicar botó *Pagar* del footer de la cistella.
3. Omplir formulari de dades i pagament.
4. Clicar botó *Pagar*.

**Postcondició:** S'ha creat una nova comanda i redirigir a l'usuari a pàgina de Historial de comandes.

**Validació:** Correcte

#### Prova-16

**Descripció:** Editar un producte

**Precondició:** Tenir una sessió iniciada a la pàgina d'administrador

**Passos a seguir:**

1. Seleccionar un producte del desplegable de productes.
2. Seleccionar una nova talla disponible o modificar preu.
3. Apretar botó *Guardar cambios*.

**Postcondició:** S'ha modificat un producte.

**Validació:** Correcte

#### Prova-17

**Descripció:** Editar dades d'un usuari

**Precondició:** Tenir una sessió iniciada a la pàgina d'administrador

**Passos a seguir:**

1. Introduir l'identificador d'un usuari.
2. Seleccionar el checkbox rol administrador.
3. Apretar botó *Guardar cambios*.

**Postcondició:** S'ha modificat un usuari i ara es administrador.

**Validació:** Correcte

#### Prova-18

**Descripció:** Editar una comanda

**Precondició:** Tenir una sessió iniciada a la pàgina d'administrador

**Passos a seguir:**

1. Introduir l'identificador d'una comanda.
2. Introduir el nou estat de la comanda.
3. Apretar botó *Guardar cambios*.

**Postcondició:** S'ha modificat l'estat d'una comanda.

**Validació:** Incorrecte

No hi ha opció de modificar comanda a la web d'administrador, futura millora.

Prova-19

**Descripció:** Filtrar el catàleg de productes

**Precondició:** -

**Passos a seguir:**

1. Seleccionar *Ver catálogo* a la pàgina principal.
2. Seleccionar una talla, a l'apartat de filtres.

**Postcondició:** Es mostren només els productes que tenen la talla disponible.

**Validació:** Correcte

Prova-20

**Descripció:** Obrir un tiquet

**Precondició:** -

**Passos a seguir:**

1. Seleccionar *Ver catálogo* a la pàgina principal.
2. Seleccionar una talla, a l'apartat de filtres.

**Postcondició:** Es mostren només els productes que tenen la talla disponible.

**Validació:** Correcte

Prova-20

**Descripció:** Modificar un tiquet

**Precondició:** -

**Passos a seguir:**

1. Introduir l'identificador d'una comanda.
2. Introduir el nou estat de la comanda.
3. Apretar botó *Guardar cambios*.

**Postcondició:** S'ha modificat l'estat del tiquet

**Validació:** Incorrecte

No hi ha opció de modificar tiquet a la web d'administrador, es gestiona per correu.  
Futura millora.

## Proves End-to-End Completes

Aquestes proves s'han realitzat per garantir el correcte funcionament de la web i les seves funcionalitats principals.

Prova End-to-End 01: Flux de registre i compra

### 1. Registrar usuari

- Accedir a la pantalla de registre.
- Omplir el formulari de registre.
- Apretar botó *Registrarse*.
- Validació: Correcte

### 2. Iniciar sessió

- Accedir a la pantalla de login.
- Omplir el formulari de login.
- Apretar botó *Iniciar sesión*.
- Validació: Correcte

### 3. Seleccionar perfil

- Accedir a la pàgina de perfil.
- Validació: Correcte

### 4. Omplir dades perfil

- Omplir el formulari de dades personals.
- Apretar botó *Guardar cambios*.
- Validació: Correcte

### 5. Seleccionar Ver catàlego

- Accedir a la pàgina de catàleg.
- Validació: Correcte

### 6. Seleccionar un producte

- Seleccionar un producte del catàleg.
- Validació: Correcte

### 7. Afegir producte al carret

- Seleccionar talla del producte
- Apretar botó *Añadir a la cesta*.
- Validació: Correcte

### 8. Obrir carret

- Seleccionar botó *Pagar*.
- Validació: Correcte

### 9. Acabar d'omplir dades

- Omplir el formulari de pagament.
- Apretar botó *Pagar*.
- Validació: Correcte

### 10. Visitar comandes

- Accedir a la pàgina de comandes.
- Validació: Correcte

### 11. Obrir tiquet

- Accedir a la pàgina de suport.
- Omplir el formulari de tiquet .
- Apretar botó *Enviar ticket*.
- Validació: Correcte

## Prova End-to-End 02: Donar permisos d'administrador

### 7. Iniciar sessió amb usuari no administrador

- Accedir a la pantalla de login de la web d'administrador.
- Omplir el formulari de login amb un usuari no administrador.
- Apretar botó *Iniciar sesión*.
- Visualitzar error Usuario no autorizado
- Validació: Correcte

### 8. Iniciar sessió amb usuari administrador

- Omplir el formulari de login.
- Apretar botó *Iniciar sesión*.
- Validació: Correcte

### 9. Obtenir dades usuari no administrador

- Afegir id usuari no administrador
- Apretar botó *Obtener datos del usuario*
- Validació: Correcte

### 10. Donar permisos administrador

- Seleccionar checkbox *Adminsitrador*
- Apretar botó *Guardar cambios*
- Validació: Correcte

### 11. Tancar sessió

- Seleccionar icone d'usuari
- Apretar botó *Cerrar sesión*.
- Validació: Correcte

### 12. Iniciar sessió amb nou usuari administrador

- Omplir el formulari de login.
- Apretar botó *Iniciar sesión*.
- Accedir a la web d'administrador
- Validació: Correcte

## Prova End-to-End 03: Modificar producte

### 1. Iniciar sessió amb usuari client

- Accedir a la pantalla de logi.
- Omplir el formulari de login amb un usuari.
- Apretar botó *Iniciar sesión*.
- Validació: Correcte

### 2. Navegar fins la pàgina d'un producte

- Navegar al catàleg

- Seleccionar un producte
  - Visualitzar les talles disponibles i el preu
  - Validació: Correcte
- 3. Iniciar sessió amb usuari no administrador**
- Accedir a la pantalla de login de la web d'administrador.
  - Omplir el formulari de login amb un usuari administrador.
  - Apretar botó *Iniciar sesión*.
  - Validació: Correcte
- 4. Modificar producte**
- Seleccionar el producte visualitzat a la web de client
  - Modificar talles disponibles.
  - Modificar preu del producte
  - Apretar botó *Guardar producto*.
  - Validació: Correcte
- 5. Iniciar sessió amb usuari client**
- Accedir a la pantalla de logi.
  - Omplir el formulari de login amb un usuari.
  - Apretar botó *Iniciar sesión*.
  - Validació: Correcte
- 6. Navegar fins la pàgina del producte modificat**
- Navegar al catàleg
  - Seleccionar un producte
  - Visualitzar les talles disponibles i el preu actualitza
  - Validació: Correcte