Plataforma de Ticketing i Reserva de Seients en Temps Real

Pau Esmerats Fité

Resum

El treball consisteix a desenvolupar una aplicació web de reserva de seients de cinema que ofereix una experiència intuïtiva i fàcil d'utilitzar als clients. Està basada en React per a la interfície de la pàgina web i Firebase per a l'emmagatzematge de dades i autenticació d'usuaris. S'utilitza Express.js i Socket.io per a la sincronització bidireccional immediata, evitant la duplicitat de seients. El procés inclou registre i autenticació d'usuaris, selecció de seients en un plànol SVG interactiu, generació de codi QR per validació d'entrades. L'aplicació disposa d'un perfil d'usuari amb l'historial de compres i d'un quadre de control administratiu per a la gestió de les sessions i un lector de QR per a la validació d'entrades. El desenvolupament segueix una metodologia àgil basada en Kan ban, amb GitHub i Trello per al seguiment de versions i tasques. Aquest projecte facilita una gestió eficient dels seients, superant les limitacions de les plataformes existents i oferint una solució escalable i mantinguda en el núvol.

Paraules clau

Cinema, gestió de reserves, temps real, aplicació web, React, websockets, JavaScript, Socket.io, núvol, Firebase, Express.js

Abstract

The work consists of developing a web application for reserving cinema seats that offers an intuitive and easy-to-use experience to customers. It is based on React for the web page interface and Firebase for data storage and user authentication. Express.js and Socket.io are used for immediate bidirectional synchronization, avoiding seat duplication. The process includes user registration and authentication, seat selection on an interactive SVG map, QR code generation for ticket validation. The application has a user profile with purchase history and an administrative dashboard for session management and a QR reader for ticket validation. The development follows an agile methodology based on Kanban, with GitHub and Trello for version and task tracking. This project facilitates efficient seat management, overcoming the limitations of existing platforms and offering a scalable solution maintained in the cloud.

Index Terms

Cinema, reservation management, real-time, web application, React, websockets, JavaScript, Socket.io, cloud, Firebase, Express.js

1 Introducció - Context del treball

En aquesta secció del treball de fi de grau presento de forma general l'arquitectura i les tecnologies emprades per al desenvolupament de l'aplicació web, que més endavant profunditzaré i explicaré amb més detall.

La pàgina web simula la pàgina del *Cinema Catalunya [1]* de Terrassa. És un cinema centenari ubicat al centre de la ciutat i que té dues sales de projecció.

Es realitzarà una aplicació web que permetrà als usuaris autenticar-se, veure la cartellera i poder reservar a temps real les butaques.

Per fer-ho, he optat per utilitzar React per a la interfície d'usuari connectada amb una base de dades i serveis d'autenticació de Firestore, creat i desplegat per Google. L'aplicació de React està desplegada a Vercel.

Per poder fer les reserves en temps real, he desplegat un servidor Express.js a Render, un allotjament que permet tenir un servidor gratuït. Dins d'aquest servidor està creat el websocket server que permetrà fer els intercanvis de dades entre client i servidor.

Finalment, pel tema de correus de confirmació de les entrades, es fa una crida a una API que està a Firebase. Es va crear un compte de Google que és el qui envia a través d'automatització els correus.

Tot el projecte està pujat a internet i s'hi pot accedir a través de l'enllaç https://cinematfg.pauesmerats.com.

- *E-mail de contacte: pau.esmerats@autonoma.cat*
- *Menció realitzada: Enginyeria del Software*
- *Treball tutoritzat per: Sergio Bachiller Rubia*
- *Curs 2024/25*

2 Objectius

2.1 Objectiu principal

L'objectiu principal del treball és crear una aplicació web funcional que permeti als clients reservar seients d'una sala de cinema de manera eficient i en temps real, garantint una bona experiència per al client fluida i intuïtiva.

2.2 Objectius secundaris

- Dissenyar i desenvolupar la interfície web: fer un disseny d'una interfície web intuïtiva i fluida assegurant

que sigui accessible per a diferents dispositius.

- Desenvolupar un back-end: codificar un back-end i base de dades que permeti realitzar les funcionalitats de la interfície.
- Implementar Socket.io [2]: utilitzar el mòdul de JavaScript Socket.io per tal d'implementar una connexió bidireccional entre client i servidor permeten que les reserves de seients sigui en temps real.

3 Estat de l'art

En l'actualitat, els cinemes tenen plataformes digitals com Cinesa [3], que permeten als usuaris poder realitzar reserves de seients i pagaments directament des de la seva pàgina web o aplicacions mòbils. També n'hi ha d'altres, que tan sols ofereixen la venda d'entrades sense seleccionar seients, com la web de Cinema Catalunya de Terrassa.

Tot i que ofereixen una solució bastant completa, no ofereixen que la reserva de seients sigui en temps real, fent que puguin haver-hi duplicitat de reserves de seients. En resposta a aquesta limitació, el meu treball pretén trobar una solució per a aquest problema, fent que l'usuari i el servidor estiguin en sincronització, permeten veure tots els seients reservats i ocupats en temps real, evitant la duplicitat de seients a l'hora de reservar-los.

Aquesta comunicació entre el servidor i l'usuari s'implementarà utilitzant la llibreria Socket.io. Això serà essencial perquè la comunicació sigui immediata i eficaç. Aquesta funcionalitat, que encara no és present en moltes aplicacions comercials, representa un avantatge molt significatiu.

En definitiva, actualment les aplicacions web de cinemes cobreixen les funcionalitats bàsiques com la selecció de seients i autentificació d'usuaris. Malgrat això, la integració d'aquesta llibreria permet superar les limitacions de les actuals plataformes fent una gestió eficient de les dades a tems real i millorant notablement l'experiència de l'usuari.

4 Planificació

Per a la planificació del treball es va dividir el total de la feina a fer en tres fases: **planificació del treball, disseny i desenvolupament.** En la primera fase, l'objectiu d'aquesta era definir els objectius i tecnologies que es farien en el projecte. Un cop definits comença la segona fase, la del disseny, que consistia a prendre els requisits de la plataforma, les funcions que farà i fer una maqueta del disseny de l'aplicació web. Un cop tenint tot això llest, comença la fase tres, el desenvolupament en si de l'aplicació, que és la més llarga del treball.

Vaig desglossar en detall els diferents apartats de cada fase en un diagrama de Gannt, utilitzant el programa Gantt Project [4], posant una estimació del temps de feina i millorant així l'organització del treball.

Nom	Inici	Finalització	Durada
Planificació del treball	24/2/25	3/3/25	6
Definció d'objectius	24/2/25	24/2/25	1
Realitzar diagrama de Gantt	25/2/25	26/2/25	2
Recerca tecnològies	26/2/25	27/2/25	2
Redacció pla de treball	27/2/25	3/3/25	3
Disseny	4/3/25	14/3/25	9
Presa de requisits	4/3/25	4/3/25	1
Product backlog	5/3/25	5/3/25	1
Diagrama de classes	6/3/25	6/3/25	1
Diagrama d'activitats	7/3/25	7/3/25	1
Disseny amb Figma	10/3/25	14/3/25	5
Redacció del document	13/3/25	14/3/25	2
Codificar	17/3/25	28/5/25	53
Programar aplicació	17/3/25	23/5/25	50
Redactar memòria	26/5/25	28/5/25	3
Redactar article final	29/5/25	25/6/25	20
Preparar defensa TFG	26/6/25	7/7/25	8

Figura 1Taula del diagrama de Gannt del projecte.

5 Metodologia

Dins de la fase del desenvolupament, se seguirà una metodologia àgil basada en el Sistema Kanban[5], que permet tenir una gestió visual de les tasques del projecte. Aquesta metodologia garantirà una execució estructurada amb el tauler Kanban on es pot veure l'estat de les tasques en qualsevol moment.

Per garantir una bona organització i seguiment del desenvolupament de l'aplicació web, utilitzaré Trello[6], [7] com a eina per tenir el meu tauler. Aquest estarà estructurat amb diverses columnes que representaran els diferents estats que tenen les tasques:

- **To Do:** seran totes les tasques pendents a realitzar.
- **To Do This Week:** aquest estat del tauler serà on tindré totes les tasques que tinc previstes per a fer durant la setmana, per tal de visualitzar bé el progrés.
- **Doing**: en començar a treballar en una tasca, l'estat passarà de To Do a Doing.
- **Test**: quan s'hagi acabat el desenvolupament d'una tasca, es passarà a l'estat de test, on esperarà allà fins a ser testejada i que compleixi els requisits funcionals.
- **Done**: en acabar de ser testejada la tasca, es traslladarà a l'estat de Done, mostrant que ja s'ha fet la feina a fer.

Per a tenir una bona gestió de versions del codi, he utilitzat Git, conjuntament amb GitHub, per poder anar guardant les diferents versions del codi en commits tan local com al núvol.

6 Tecnologies utilitzades

6.1 React amb Vite

Pel desenvolupament de la interfície d'usuari he escollit React.js [8], una llibreria de JavaScript modular i basada en components, creada i mantinguda per Meta, que facilita la creació d'interfícies reactives i reutilitzables. Per a la configuració del projecte he fet servir Vite, un dev server lleuger

i molt ràpid que optimitza els temps d'arrencada i recàrrega en calent del codi.
A més de React, s'han importat els següents mòduls de npm de JavaScript:

- React-zoom-pan-pinch[9]: per permetre fer el zoom in and out dels mapes SVG de les sales del cinema.
- Socket.io-client: per gestionar la comunicació bidireccional amb el servidor websocket a temps real.
- Axios: per fer peticions HTTP, que s'utilitzarà per cridar a l'API d'enviament de correus electrònics de confirmació.
- Firebase, per a l'autenticació d'usuaris, peticions i emmagatzematge de dades i allotjament d'alguna funció serverless.
- React-icons: per incorporar icones per fer més visual la interfície.
- React-QR-code: per a la generació de codis QR de les entrades i premis.
- @mantine: framework de components UI, estilització i hooks útils.
- @yudiel/React-QR-scanner: component per escaneig de codis QR directament des del navegador web.

6.2 Firebase

Per al projecte s'ha escollit Firebase [10] com a plataforma completa de back-end a causa de la seva forma molt senzilla d'implementació i també que té moltes funcionalitats amb la seva capa gratuïta. Hi ha altres empreses com Supabase que donen uns serveis similars, però tenint prèvia experiència amb Firebase, m'he acabat decantant pel que ja coneixia. He fet ús de tres dels seus serveis:

- **Firestore**: És una base de dades en temps real orientada a documents, on hi ha col·leccions i documents dins de cada col·lecció.
- **Firebase Auth**: S'ha fet servir el mòdul de Firebase Auth per tal de gestionar els registres i accessos d'usuaris amb el correu i contrasenya. És molt senzill d'implementar, ja que gestiona la confidencialitat de les contrasenyes automàticament i el desenvolupador no s'ha de preocupar de la privacitat d'aquestes.
- **Fuctions**: són crides de funcions fetes amb Express.js sense tenir un servidor actiu.

6.3 Express.js

Un servidor amb Express.js on a dins seu hi haurà una instància d'un servidor Socket.io que permetrà fer la gestió de la selecció en temps real. Aquest servidor s'encarregarà de la gestió de la connexió bidireccional entre servidor i client.

Quan arrenca el servidor, s'inicialitza el Socket.io de manera que només accepti peticions del domini establert per a la pàgina web, augmentant així la seguretat.

Llavors, el Socket.io espera a la connexió d'usuari fent socket.on('connection'). Quan un usuari es connecta, envia una petició de 'join' amb l'identificador de la sessió. El que fa això és que el servidor enviarà les peticions diferenciant grups. Per exemple, si hi ha 10 usuaris simultanis a la pàgina, però només dos a la mateixa sessió, el servidor només enviarà peticions d'aquella sessió a aquells dos usuaris, no als 10 usuaris.

L'usuari enviarà fent peticions al servidor amb la funció .emit('seientSeleccionat) quan selecciona un seient i .emit('seientDesseleccionat') quan el desselecciona. El servidor, quan rep una petició d'aquestes, el que fa és emetre 'seleccionat' o 'desseleccionat' a tots els usuaris que són dins del mateix grup que l'usuari ha fet la petició.

Els seients seleccionats es guarden en el servidor per si de sobte algun nou usuari entrés, el servidor pogués enviar-li tots els seients que estan seleccionats. Això sí, els seients seleccionats tenen una caducitat de 10 minuts, per tal de lliurar el seient si l'usuari al final no confirma la reserva i no es quedi el seient seleccionat però no reservat.

El següent diagrama mostra com seria un cas d'ús d'una connexió simultània de dos usuaris a la mateixa sessió d'una pel·lícula. L'usuari es connecta amb el servidor websocket i envia una petició de join amb l'identificador de la sessió.

El servidor li retorna tots els seients que actualment estan seleccionats per a altres usuaris, en aquest cas interpretarem que no hi ha.

Més endavant, l'usuari selecciona el seient 1-1, enviant la petició seientSeleccionat() amb l'identificador del seient (1 sent la fila i 1 la columna del seient). Quan ho rep el servidor, envia a tots els usuaris dins del grup de la sessió, en aquest cas, l'usuari2, la petició des del servidor seleccionat(1-1). L'envia en aquest cas només a l'Usuari2 però l'enviaria per N persones connectades que hi hagués simultànies.

Això també succeeix per als seients 1-2 i 1-3. Quan un usuari rep la notificació seleccionat, el front-end marca de groc el seient en qüestió.

Finalment, l'usuari decideix no agafar el seient 1-3 i el desselecciona, notificant al servidor i ell a la vegada a l'usuari2, actualitzant el mapa de la sala i desmarcant de groc el seient.

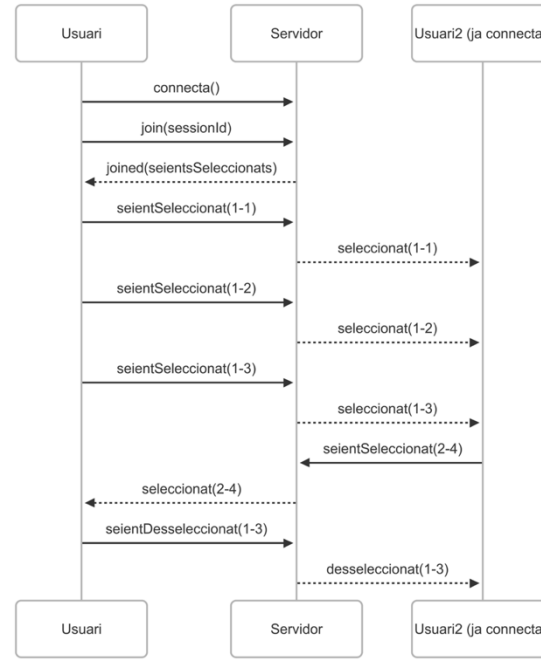


Figura 2 Diagrama de seqüencies dels websockets entre dos usuaris i el servidor.

6.4 Inkscape

Inkscape és un editor de gràfics vectorials de codi obert que permet crear i editar imatges en format SVG de manera precisa i lleugera.

Per aquest projecte, es va utilitzar per dibuixar amb detall els plànols de les dues sales de cinema Catalunya de Terrassa, definint cada seient com un element SVG amb un identificador únic que el codi JavaScript pot llegir i gestionar. Gràcies a l'exportació nativa en SVG, es van obtenir arxius nets i optimitzats, fàcils d'integrar amb React i d'actualitzar en temps real mitjançant Socket.io.

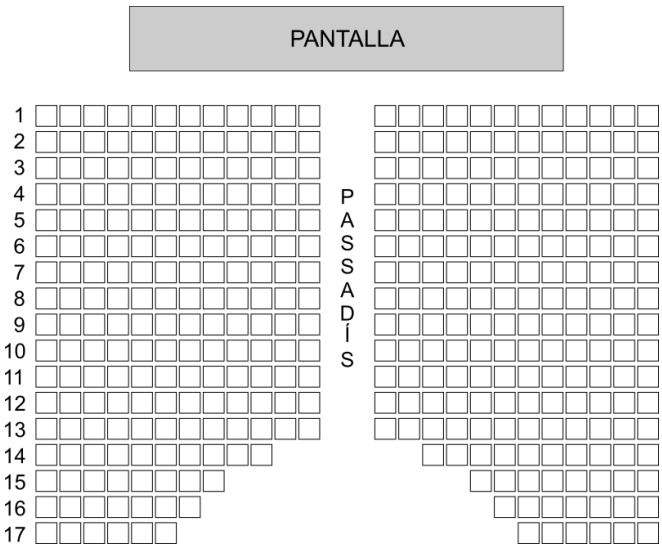


Figura 4 SVG de la sala 1 del Cinema Catalunya.

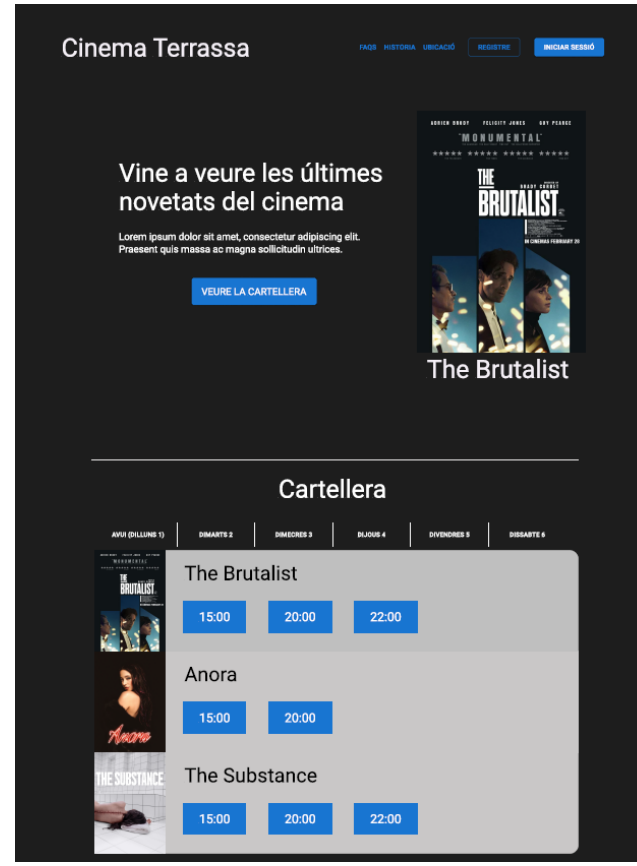


Figura 3 Maqueta de la pàgina principal de l'aplicació web, realitzada amb Figma.

6.5 Figma

Durant la fase de disseny, vaig fer servir el programa de Figma per crear prototips i maquetes de la interfície de l'aplicació, dissenyant de manera visual algunes de les pantalles, definint l'estructura dels components, la navegació i l'estil gràfic abans de començar a escriure codi, amb l'objectiu de tenir una visió clara de l'experiència d'usuari i agilitzar posteriorment el desenvolupament.

Era el meu primer cop fent servir aquest programa i vaig haver de fer un curs introductori anomenat *Learn Figma with Shift Nudge [11]*, d'un programador amb renom que es fa dir @mds [12], per tal de saber fer-lo servir, ja que és un programa que es fa complicat de fer servir si no s'ha utilitzat mai.

Es pot veure en la següent imatge la maqueta inicial que tenia per a la pàgina principal de l'aplicació: una barra de navegació a dalt de la pàgina, seguidament d'una presentació breu de la pàgina i a baix de tot, la cartellera actualitzada de les pròximes sessions.

A l'annex (A2) hi ha dues imatges més de maquetes d'altres pàgines de l'aplicació web.

7 REQUISITS

7.1 Funcionals

- Gestió d'usuaris:
  - **RF-01:** Els usuaris podran registrar-se i iniciar sessió mitjançant el seu correu electrònic i contrasenya.
  - **RF-02:** L'usuari podrà editar la seva informació personal.
  - **RF-03:** Possibilitat de recuperar la contrasenya per correu electrònic.
- Selecció de pel·lícula i sessió:
  - **RF-04:** El visitant podrà veure el llistat de pel·lícules que estan disponibles, l'horari de les sessions de cadascuna i la seva informació detallada.
  - **RF-05:** Possibilitat de poder seleccionar una sessió.
- Reserva de seients:
  - **RF-06:** Un cop el visitant seleccioni una sessió, el sistema mostrarà un mapa interactiu de la sala amb els seients disponibles, ocupats i reservats.
  - **RF-07:** Quan seleccioni un seient, aquest quedarà marcat com a reservat en temps real per tal d'evitar conflictes de duplicitat.

- Confirmació de reserva:
    - **RF-08:** Un usuari, quan ja hagi seleccionat els seients desitjats, el sistema el portarà a una pantalla per a poder confirmar la reserva.
    - **RF-09:** Un visitant, quan hagi seleccionat els seients, li demanarà que es registri com a usuari per tal de fer la reserva i després el redirigirà a la pantalla de confirmació.
    - **RF-10:** Un cop confirmada la reserva, es generarà una pantalla de confirmació on podrà baixar els codis QR de les entrades. També rebrà un correu electrònic amb els codis QR.
- Sistema d'administració:
    - **RF-11:** Els administradors de la pàgina tindran un accés especial per poder afegir pel·lícules i sessions al catàleg.

7.2 No funcionals

- **RNF-01:** El temps de càrrega de les pàgines serà de 5 segons com a màxim per garantir una experiència fluida a l'usuari.
- **RNF-02:** La sincronització de seients reservats ha de ser immediata amb tots els usuaris.
- **RNF-03:** L'aplicació ha de suportar una gran quantitat de consultes simultànies.
- **RNF-04:** Les contrasenyes dels usuaris han d'estar encriptades a la base de dades.
- **RNF-05:** La interfície ha de ser intuïtiva i responsiva per telèfons i ordinadors.
- **RNF-06:** El sistema ha de garantir una bona seguretat en les bases de dades per tal d'evitar fugues de dades dels usuaris.
- **RNF-07:** El sistema ha de tenir disponibilitat el 99,9% del temps.

8 DESENVOLUPAMENT

Durant el desenvolupament de l'aplicació web vaig tenir diverses dificultats que detallaré en els següents subapartats.

8.1 Adaptació per dispositius mòbils

En la fase inicial, bona part de les proves es van fer en entorns d'escriptori, i la interfície basada en CSS modules i grid va comportar problemes de llegibilitat i usabilitat en pantalles menors.

Per assegurar una experiència consistent, es van introduir media queries específiques per a resolucions de 768 px i 480 px, redissenyant certs components i controls perquè s'ajustessin bé sense perdre funcionalitat.

8.2 Desplegament de l'aplicació

La capa front-end es va allotjar amb Vercel [13], que automatitza el procés de build i ofereix HTTPS de forma gratuïta sincronitzant amb el repositori de GitHub. No obstant això, Vercel no admet processos Node.js persistents, indispensables per al servidor de Socket.io.

Després d'analitzar opcions (Heroku, Railway, etc.), es va triar Render [14] per al back-end, ja que la seva capa gratuïta permet mantenir viu un servei web Node.js i executa automàticament npm run build en desplegar-se.

8.3 Mapes SVG de les sales de cinema

Un altre repte en el projecte van ser els mapes de les sales amb SVG [15] per a representar el plànol de seients que em va obligar a buscar una eina capaç d'editar vectors amb precisió. Tot i que Adobe Illustrator és l'estàndard de la indústria, la seva llicència de pagament no encaixava amb els recursos del projecte, així que vaig buscar alternatives [16] fins que finalment vaig instal·lar Inkscape.

Aquest, em va permetre ajustar els camins i les etiquetes dels rectangles que representen cada seient, assegurant-me que el fitxer SVG resultant conservés tots els atributs necessaris per a la interactivitat[17][18][19].

Per tal de gestionar la selecció de butaques en el codi, vaig haver d'assignar a cada element <rect> un atribut ID únic que representés la butaca (per exemple, "1-1", "1-2", etc.).

Aquesta identificació és clau perquè, en fer clic, el handler de React pugui recollir l'ID i transmetre'l al servidor via Socket.io. La creació manual de centenars de rectangles amb el seu ID corresponent va ser feixuga, però va ser necessària per garantir que cada seient fos independent i manipulable des del front-end.

El pas final era convertir aquest arxiu SVG a un fitxer SVG per tal de poder-lo incloure en la pàgina com a un component. La pàgina SVGR Playground [20] va ser de gran ajuda, ja que fa l'exportació automàticament.

8.4 Gestió de dates

Per gestionar la cartellera de la pàgina inicial, les sessions s'havien d'agafar aplicant un filtre a la consulta de la base de dades. Firebase treballa les dates amb una variable Timestamp, que té dos camps: seconds i nanoseconds, diferent de la que utilitza JavaScript que és la variable Date.

Això em va obligar a trobar la manera de fer la conversió d'aquestes dues variables. Va ser una tasca complicada que al final es va solucionar multiplicant els seconds de Timestamp per 1000 i llavors cridar a la funció new Date().

8.5 Gestió de les rutes

React és un framework per a construir Single Page Applications. Per poder fer que hi hagi diferents rutes com '/login', '/home', el que es fa és importar la llibreria 'react-router-dom', permeten especificar quin component vols executar quan s'accedeixi a una ruta específica.

```
function App() {
  const router = createBrowserRouter([
    {
      path: '/',
      element: <Home />
    },
    {
      path: '/login',
      element: <Login />
    },
    {
      path: '/register',
      element: <Register />
    },
    {
      path: '/profile',
      element: <Profile />
    },
    {
      path: '/dashboard',
      element: (
        <AdminRoute>
          <Dashboard />
        </AdminRoute>
      )
```

Figura 5 Definició de les rutes de la pàgina web.

8.6 Base de dades Firestore

És una base de dades en temps real orientada a documents, on hi ha documents dins de cada col·lecció. Per al projecte, he requerit cinc col·leccions:

- **Films**: on es guarden totes les dades de les pel·lícules com ara el títol, URL del pòster, duració, sinopsis, etc.
- **Users**: es guarda les dades de l'usuari com el correu, noms, punts que té i si és un usuari administrador de la pàgina.
- **Sessions**: es guarden cada sessió d'una pel·lícula. Hi ha la data i hora de la sessió, el número de la sala, l'identificador únic de la pel·lícula i una cadena de tots els seients que ja han sigut reservats.
- **Reservats**: quan es confirma una reserva, es crea un document a aquesta col·lecció, guardant quins seients s'han reservat, de quina sessió es tracta (identificador de la sessió), l'hora i dia de quan s'ha fet la reserva i l'identificador de l'usuari que ha fet la reserva.

- **Prices** (premis): col·lecció que guarda els premis bescanviats per usuaris, molt similar a reservats però amb premis.

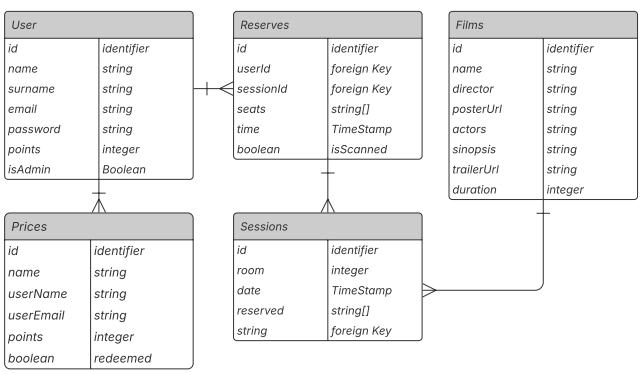


Figura 6 Diagrama de d'objectes de la base de dades no relacional.

8.7 Recompenses amb puntuació

Per incentivar als clients a anar sovint al cinema, vaig implementar un sistema de recompenses per puntuació. Quan un usuari fa una compra a la pàgina, se li dona uns punts. Quan l'usuari té variïs punts acumulats, podrà bescanviar-los a l'apartat del perfil per a premis. Aquesta funcionalitat pot ajudar a fidelitzar clients per al cinema a part d'incentivar a anar més al cinema.

9 Resultats

9.1 Pàgina d'inici

La pàgina d'inici on l'usuari és dirigit a l'inici. En ella es pot veure la barra de navegació, que estarà present a totes les pàgines, un carrusel d'imatges dels pòsters de les pel·lícules que hi ha a la cartellera, que està a sota d'aquest. Per a cada pel·lícula, hi ha el nom, el pòster i les hores de les sessions per a cada dia.

La barra de navegació ofereix un buscador per a buscar les pel·lícules directament a través del títol o nom del director. Hi ha també l'opció de canviar el mode de la pàgina web: mode clar i mode fosc. Finalment, hi ha l'opció d'autenticar-se si no ho està i tanca la sessió si ja s'ha autenticat abans l'usuari.

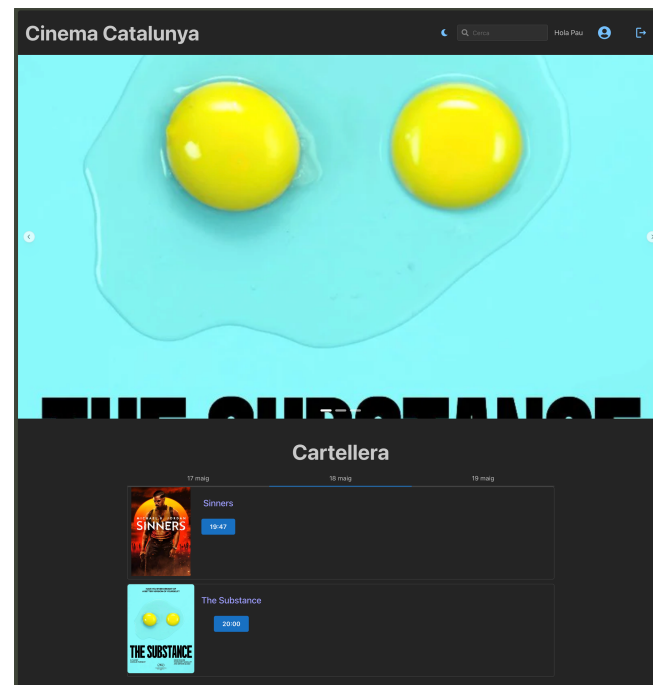


Figura 7 Versió final de la pantalla principal de l'aplicació web.

9.2 Detall de la pel·lícula

Aquesta és una pàgina que mostra els detalls de la pel·lícula que l'usuari selecciona de la cartellera. S'hi mostra informació com el director, actors, duració, sinapsis i el tràiler. També mostra de manera dinàmica consultant a la base de dades les sessions programades que hi ha per a la pel·lícula. (Vegeu Figura 24 a l'Annex).

9.3 Compra d'entrades

És on l'usuari reserva els tiquets de la sessió de la pel·lícula seleccionada. Hi ha tres passos: la selecció dels seients, inici de sessió i confirmació de la reserva.

En el mapa interactiu de la sala (que depenent de la sessió pot ser la Sala 1 o la 2) l'usuari selecciona els seients que vol reservar.

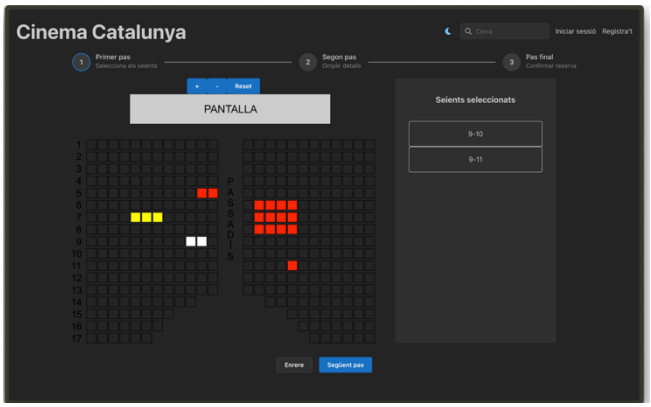


Figura 8 Selecció de seients. En vermell, seients reservats, en groc, seients seleccionats per un usuari contemporani i en blanc els que he seleccionat.

Un cop feta la selecció, anem al segon pas que obliga l'usuari a registrar-se o iniciar sessió. Si l'usuari ja té el compte iniciat, aquest pas no existeix des d'un principi, i va directe al tercer pas. (Vegeu Figura 16 a l'Annex).
L'últim pas de la compra d'entrades és la confirmació de les entrades, no he afegit passarel·la de pagament, ja que no estem venent res, és un prototip.
Finalment, ens surt una pàgina de confirmació amb el detall de la comanda. (Vegeu Figura 17 a l'Annex).

9.4 Autenticació

Per a poder fer l'autenticació d'usuari, hi ha dues pàgines de l'aplicació web: registre i inici de sessió. La de registre, s'utilitzarà per primer cop per donar-se d'alta en la plataforma. L'altre, en canvi, s'utilitzarà quan l'usuari ja estigui donat d'alta i vulgui iniciar la seva sessió.[6], [7], [21], [22].

En el registre es demanen el nom, cognoms, correu electrònic i contrasenya, que s'emmagatzemen a la base de dades de Firebase. (Vegeu Figura 19 a l'Annex).

Per a l'inici de sessió, es demana a l'usuari el correu electrònic i contrasenya. Tot el procés està fet amb les funcions donades per Firebase per tal d'oferir un sistema segur per al client. (Vegeu Figura 18 a l'Annex).

9.5 Perfil

En la pàgina de perfil, que només es pot accedir quan l'usuari ha iniciat sessió amb un compte. Hi ha tres pestanyes que l'usuari veu, i una d'extra que només els administradors de la pàgina veuen.

La primera pestanya és la del 'Perfil', on es veu les dades personals del client i els punts acumulats que té. Més avall, té un menú amb tres opcions per bescanviar els punts obtinguts amb els diferents premis que hi ha. (Vegeu Figura 20 a l'Annex).

En la segona pestanya, és per a poder veure les compres que ha realitzat l'usuari. Es pot veure el pòster, el nom, el director, la data, el número de la sala, els punts obtinguts, els seients reservats i el preu total. (Vegeu Figura 21 a l'Annex).

En clicar el botó 'Veure QR' es genera un QR que els treballadors podran escanejar per confirmar l'entrada.
La tercera pestanya, la de 'Premis', és molt similar a la d'entrades, però amb la diferència que es mostren els premis que té l'usuari.

Figura 9 Selecció de premis per puntuació.

9.6 Quadre de control

Aquesta pantalla (Vegeu Figura 22 a l'Annex), és exclusiva i només poden accedir-hi els qui tenen el rol d'administrador. Això es comprova mirant si al registre de l'usuari el booleà isAdmin és cert. És la pàgina on s'afegeixen totes les pel·lícules a la base de dades i les sessions que hi haurà.
Quan es clica el botó 'Afegir pel·lícula, apareix un desplegable (Vegeu Figura 23 a l'Annex), on es posa tots els detalls de la pel·lícula i es va seleccionant per a cada sessió que es vulgui realitzar la sala on es farà i la data corresponent. El sistema permet afegir i treure sessions d'una pel·lícula.

9.7 Escàner QR

En aquesta pàgina és on els revisors de tiquets podran escanejar les entrades de les persones que entrin en el cinema.

El sistema llegeix a través de la càmera el contingut del QR, que és l'identificador de la reserva. Després, fa diferents crides comprovant si existeix o no en la basa de dades una reserva o premis amb aquest identificador. Si existeix, comprova que no s'hagi escanejat anteriorment i retorna que és vàlida. En canvi, si ja ha estat escanejada

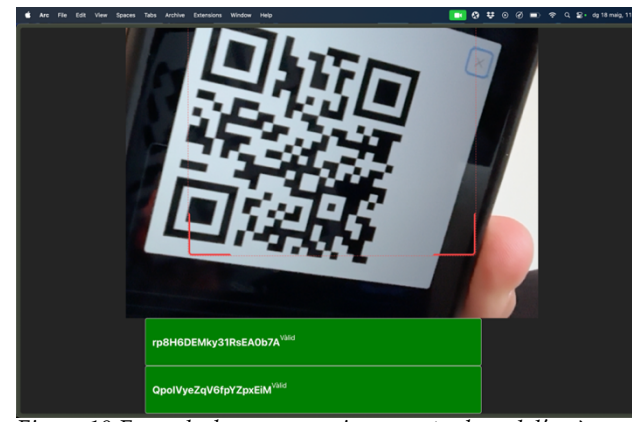


Figura 10 Exemple de com escaneja una entrada amb l'escàner de QR.

anteriorment o no existeix, retorna que no és vàlida i el motiu pel qual no ho és.

9.8 Procés d'enviament de correus de confirmació

Un cop es realitza una reserva de seients exitosa, es fa una crida a un endpoint de Firebase Functions que gestiona amb les dades que rep un enviament d'un correu a l'adreça del client de la confirmació de la reserva. [23]

En aquest correu, es pot veure el nom, dia i hora de la pel·lícula reservada. També inclou un adjunt al correu en format .ics[24], que permet afegir l'esdeveniment al calendari digital si es desitja.

Aquest enviament es realitza amb l'ajuda de Nodemailer[25] i un compte Gmail, que envia els correus de manera automatitzada.

Figura 11 Captura del correu de confirmació que rep el client després de reservar per la pàgina.

9.9 Allotjament al núvol

Tot i que els objectius inicials del projecte no contemplaven necessàriament la publicació en línia del projecte perquè tothom amb connexió a internet, vaig considerar que seria una bona aportació per al treball i un tancament òptim.
Primerament, vaig adquirir el domini pauesmerats.com a través de l'empresa Register.it de manera gratuïta.

Quan ja el vaig tenir operatiu, vaig crear un compte amb Cloudflare per tal de fer la gestió de DNS perquè l'experiència d'usuari és més intuïtiva i fàcil d'utilitzar. Per tant, vaig substituir els nameservers per defecte a Register.it pels servidors de Cloudflare, ja que a part d'això també proporciona certificat TLS automàtic.

Quan ja estava el domini verificat a Cloudflare, vaig entrar a Vercel per tal de publicar l'aplicació de React. Vercel és una empresa que proporciona suport per a publicar aplicacions web, proporcionant certificat TLS i garantint connexions HTTPS per a l'usuari final. Aquí el que es va fer és connectar el compte de GitHub amb Vercel i seleccionar el repositori de l'aplicació. El que fa Vercel automàticament és fer 'npm run build', que genera la carpeta estàtica /build, que serà la versió que veurà el client, i no el codi font. També es va configurar els DNS entre Vercel i Cloudflare per tal que la web de Vercel es pogués accedir per cinematfg.pauesmerats.com.
Vercel té molts serveis, però la capa gratuïta no em permetia tenir un servidor Node.js per a tenir el servidor que permet el suport de reserva de seients a temps real. Per tant, vaig haver de buscar una alternativa per a això.

Aquesta alternativa va ser Render, una empresa de San Francisco que la seva capa gratuïta si et permet tenir un servidor Node.js. La instal·lació va ser gairebé igual a la de Vercel: iniciar sessió amb GitHub i seleccionar el repositori del servidor i automàticament ja fan el npm run build.

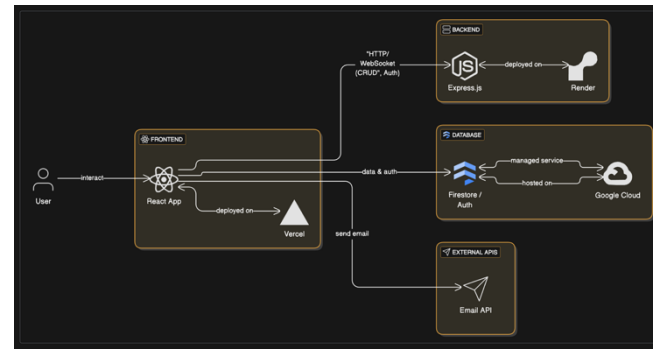


Figura 12 Esquema de tecnologies i on estan allotjades per tal d'oferir servei a Internet.

10 Conclusió

Al llarg del procés de desenvolupament del treball final de grau, m'he trobat amb una sèrie de reptes que han posat a prova tant els aspectes tècnics com la meva capacitat de planificació i adaptació, però que alhora han enriquit profundament la meva formació com a enginyer de software. Des del disseny inicial de l'arquitectura fins al desplegament en producció, cada decisió ha estat condicionada per limitacions de temps, recursos gratuïts i la necessitat de garantir una experiència d'usuari àgil i robusta.

En aquest text repasso amb detall els dubtes i dificultats principals, i com cadascun d'aquests punts ha contribuït a forjar un projecte sòlid, escalable i orientat a futurs avenços.

El primer repte que vaig tenir per al treball va ser decidir quines tecnologies utilitzar: actualment existeixen moltes tecnologies diferents i cada una té els seus avantatges i inconvenients, per la qual cosa em vaig haver d'informar bé de quines utilitzaria. També, vaig haver de planificar l'arquitectura que tindria l'aplicació abans de començar el desenvolupament per tal d'evitar malgastar recursos en desenvolupar coses que després no anirien bé per la web.

Un altre repte important va ser aconseguir una interfície responsiva i accessible en dispositius mòbils. Inicialment, la major part del desenvolupament i proves es van fer en pantalles d'escriptori, i el disseny basat en CSS modules amb disseny de quadrícula va trontollar davant resolucions menors. Per solucionar-ho, vaig introduir media queries específiques a 768 px i 480 px enfocades a aquests dispositius.

Pel que fa al desplegament "full-stack" en plataformes gratuïtes, la interoperabilitat entre Vercel, Render i Cloudflare va requerir diverses investigacions. Vercel, tot i ser l'opció natural per a l'allotjament de l'aplicació React, no allotja processos Node.js persistents, de manera que no podia fer servir directament el servidor Express.js necessàriament per a Socket.io. Després de comparar opcions, vaig triar Render per al back-end, configurant un servei web de tipus "Web Service" que exposa l'endpoint i manté viu el procés de Socket.io.

La sincronització concurrent de selecció de seients va ser, potser, el repte més delicat. Era imprescindible evitar

que dos usuaris reservessin el mateix seient quasi al mateix temps, i alhora garantir que aquells que abandonessin el procés en menys de deu minuts no retinguessin el seient indefinidament. És per això que vaig programar un servidor Express.js que dins seu té un servidor Websocket que fa la gestió de quins són els seients que se seleccionen i desseleccionen a temps real. L'únic problema que es presenta és que el servidor, en ser un servei gratuït, el servidor tarda uns 30 segons a arrancar, fent que el primer usuari que es connecti no tindrà el servidor socket en funcionament de primeres. Això es pot solucionar fàcil posant la instància en un servidor de pagament i que estigui encès les 24 hores del dia, però com que he preferit les opcions gratuïtes hi ha aquest petit inconvenient.

En definitiva, anant pas a pas amb cada repte durant el projecte —des de la selecció d'eines i el processament dels fitxers SVG fins a la integració amb React i la gestió precisa dels identificadors de cada seient— he pogut crear un sistema sòlid i fàcil de manteniment. Aquest procés no només ha garantit una aplicació web fiable i adaptable, sinó que també s'han fomentat les bones pràctiques de modularitat, fent possible qualsevol futura extensió com afegir una plataforma de pagament o integració a un cinema real.

També, per finalitzar, donaré algunes idees per a millorar i continuar expandint el projecte:

1. Integració de TMDB pel detall de les pel·lícules:
En comptes que l'administrador del cinema col·loqui un per un els diferents camps de cada pel·lícula com el nom, director, URL del tràiler, etc., es podria lligar la plataforma amb l'API de TMDB (The Movie Database)[26]Aquesta proporciona tots els detalls possibles d'una pel·lícula sabent només el títol o ID de la pel·lícula. No ho he pogut implementar per falta de temps, ja que aquesta API, tot i que és gratuïta, conté una verificació a través d'una clau que et donen ells i és bastant complexa d'implementar.

2. Optimització de l'SVG Viewer per a dispositiu mòbils:
El SVG Viewer, el component que permet fer zoom al mapa de la sala de cinema, tot i anar prou bé per a ordinadors, quan es carrega als dispositius mòbils, la manera de com es veu és molt petita i difícil de seleccionar amb precisió els seients. Una idea a futur seria veure com es podria fer per millorar l'experiència dels usuaris amb telèfons mòbils amb els SVG de les sales.

3. Evolució de la base de dades:
Firebase és una eina molt completa i fàcil d'utilitzar i molt ràpida d'implementar, per això l'he utilitzat per al treball. És una eina molt madura i dona moltes facilitats pel desenvolupat, però, també té els seus inconvenients. Proporciona poca flexibilitat en consultes complexes com ara joins o agregacions i estàs lligat als preus que Google posi pel seu servei.

És per això, que si hi hagués de fer un projecte amb més temps, triaria implementar una base de dades MongoDB per fer consultes amb agregacions, més control de les dades, índexs i còpies de seguretat i costos més barats.

4. Afegir passarel·la de pagaments:
Per donar com a finalitzat el projecte i poder-se presentar per distribuir, el més important i que manca del meu treball és la passarel·la per a fer pagaments amb targeta directament des de la pàgina. Per això, utilitzaria segurament Stripe[27] que ja et ve donat el component i no t'has de preocupar de temes de seguretat, ja que es tracta d'informació confidencial. A més a més, ofereix diferents tipologies de pagament com el pagament en targeta, Apple Pay o Google Pay, PayPal entre d'altres, oferint una extensa amplitud d'opcions al client per tal que utilitzi la que ell prefereixi.

Agraïments

Vull agrair al meu tutor Sergio Bachiller Rubia pel seu temps i dedicació. Gràcies al seu suport i consells he pogut realitzar aquest projecte amb claredat i seguretat.
A la Universitat Autònoma de Barcelona per tot l'aprenentatge obtingut durant aquests anys.
Finalment, a la meva família i la Núria, per tots els ànims i suport mentre realitzava el treball.

Bibliografia

[1] Cinema Catalunya, «Cinema Catalunya». Consulta: 8 març 2025. [En línia]. Disponible a: https://cinemacatalunya.cat/

[2] Socket.io, «Tutorial - Introduction | Socket.IO», https://socket.io/docs/v4/tutorial/introduction.

[3] Cinesa, «Cartelera, horarios y compra | Cinesa». Consulta: 8 març 2025. [En línia]. Disponible a: https://www.cinesa.es/

[4] GanntProject, «GanttProject: Free Project Management App». Consulta: 27 juny 2025. [En línia]. Disponible a: https://www.ganttproject.biz/

[5] Dan Radigan, «Kanban», https://www.atlassian.com/agile/kanban.

[6] Cosden Solutions, «Authentication in React with JWTs, Access & Refresh Tokens (Complete Tutorial)».

[7] Code Radiance, «Setting Up Firebase Auth with React: Step-by-Step Tutorial».

[8] Wikipedia, «React (software) - Wikipedia», https://en.wikipedia.org/wiki/React_(software).

[9] «react-zoom-pan-pinch - npm». Consulta: 30 març 2025. [En línia]. Disponible a: https://www.npmjs.com/package/react-zoom-pan-pinch

[10] Google, «Firebase | Google's Mobile and Web App Development Platform». Consulta: 29 juny 2025. [En línia]. Disponible a: https://firebase.google.com/

[11] @mds, «Learn Figma with Shift Nudge», https://shiftnudge.com/figma. Consulta: 24 febrer 2025. [En línia]. Disponible a: https://shiftnudge.com/figma

[12] X.com, «MDS (@mds)». Consulta: 27 juny 2025. [En línia]. Disponible a: https://x.com/mds

[13] Vercel, «Vercel: Build and deploy the best web experiences with the AI Cloud». Consulta: 29 juny 2025. [En línia]. Disponible a: https://vercel.com/home

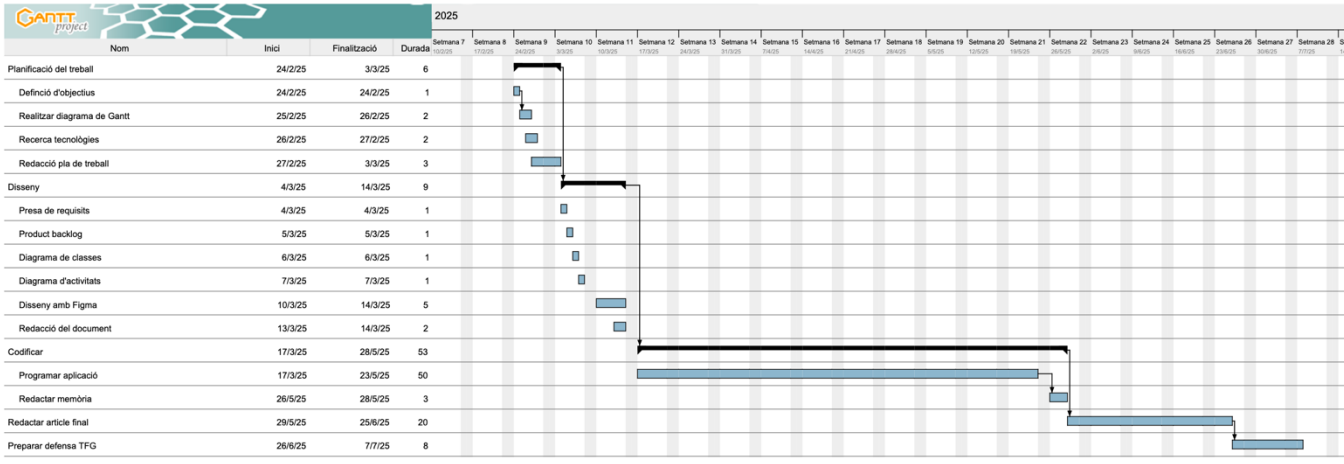
[14] Render, «Cloud Application Platform | Render». Consulta: 29 juny 2025. [En línia]. Disponible a: https://render.com/

[15] Wikipedia, «SVG - Wikipedia», https://en.wikipedia.org/wiki/SVG. Consulta: 24 febrer 2025. [En

línia]. Disponible a: https://en.wikipedia.org/wiki/SVG
[16] davep1970, «Alternate vector based illustration software?», https://www.reddit.com/r/AdobeIllustrator/comments/18z1xmw/alternate_vector_based_illustration_software/.
[17] turtleProphet, «How do I implement seat selection for a theater using svg layout of the theater : r/react», https://www.reddit.com/r/react/comments/1bj5vsl/how_do_i_implement_seat_selection_for_a_theater/.
[18] D. N. Adelaide, «Creating a Seat Selector with jQuery and SVG», https://medium.com/@digitalnoir/creating-a-seat-selector-with-jquery-and-svg-13e3cda73014.
[19] The University of Melbourne, «Accessible SVGs», https://www.unimelb.edu.au/accessibility/techniques/accessible-svgs.
[20] SVGR, «SVGR Playground - SVGR». Consulta: 9 març 2025. [En línia]. Disponible a: https://react-svgr.com/playground/?expandProps=start&namedExport=Sala1&svgo=false
[21] Firebase, «Primeros pasos con Firebase Authentication en sitios web | Firebase Authentication», https://firebase.google.com/docs/auth/web/start?hl=es-419.
[22] Code Radiance, «Step-by-Step Guide to Implementing an Authentication Context in React JS - YouTube». Consulta: 13 març 2025. [En línia]. Disponible a: https://www.youtube.com/watch?v=2-6K-TMA-nw
[23] «(13) Cómo enviar correos con Firebase functions, Nodemailer y Gmail. - YouTube». Consulta: 5 maig 2025. [En línia]. Disponible a: https://www.youtube.com/watch?v=b3u9UgjJjqM
[24] «ics - npm». Consulta: 5 maig 2025. [En línia]. Disponible a: https://www.npmjs.com/package/ics
[25] «https://nodemailer.com/».
[26] TMDB, «TMDB API». Consulta: 27 juny 2025. [En línia]. Disponible a: https://developer.themoviedb.org/docs/getting-started
[27] «Stripe: acepta más métodos de pago en tu negocio». Consulta: 24 maig 2025. [En línia]. Disponible a: https://stripe.com/es/payments/payment-methods

APÈNDIX

A1. Diagrama de Gannt



Nom	Inici	Finalització	Durada
Planificació del treball	24/2/25	3/3/25	6
Definció d'objectius	24/2/25	24/2/25	1
Realitzar diagrama de Gantt	25/2/25	26/2/25	2
Recerca tecnològies	26/2/25	27/2/25	2
Redacció pla de treball	27/2/25	3/3/25	3
Disseny	4/3/25	14/3/25	9
Presa de requisits	4/3/25	4/3/25	1
Product backlog	5/3/25	5/3/25	1
Diagrama de classes	6/3/25	6/3/25	1
Diagrama d'activitats	7/3/25	7/3/25	1
Disseny amb Figma	10/3/25	14/3/25	5
Redacció del document	13/3/25	14/3/25	2
Codificar	17/3/25	28/5/25	53
Programar aplicació	17/3/25	23/5/25	50
Redactar memòria	26/5/25	28/5/25	3
Redactar article final	29/5/25	25/6/25	20
Preparar defensa TFG	26/6/25	7/7/25	8

Figura 13 Diagrama de Gannt del projecte.

A2. Captures de les maquetes

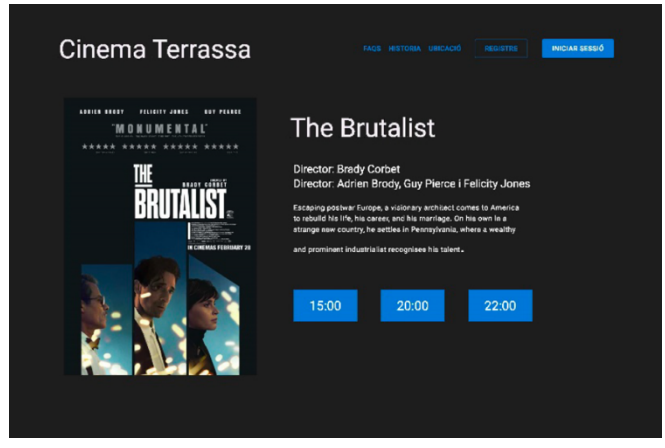


Figura 14 Maqueta de la pàgina del detall d'una pel·lícula, feta amb Figma.

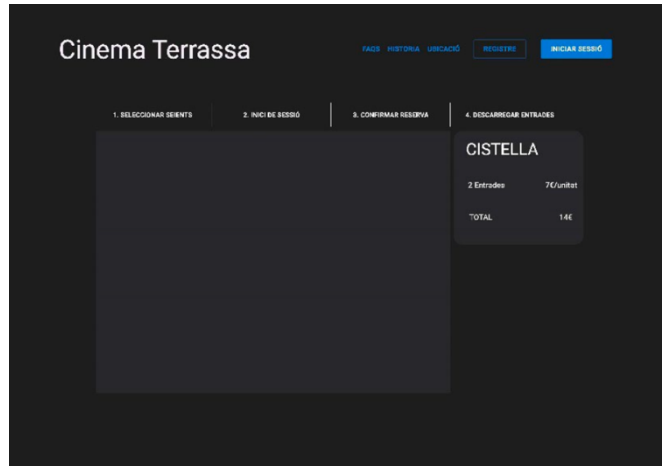


Figura 15 Maqueta de la selecció de seients, feta amb Figma.

A3 CAPTURES DE RESULTATS

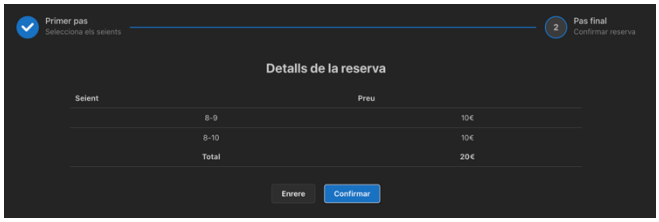


Figura 16 Segon pas de comprar entrades.

Figura 17 Pantalla de confirmació exitosa de la reserva de seients.

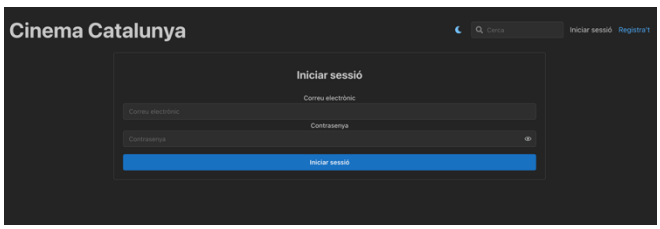


Figura 18 Pantalla d'inici de sessió.

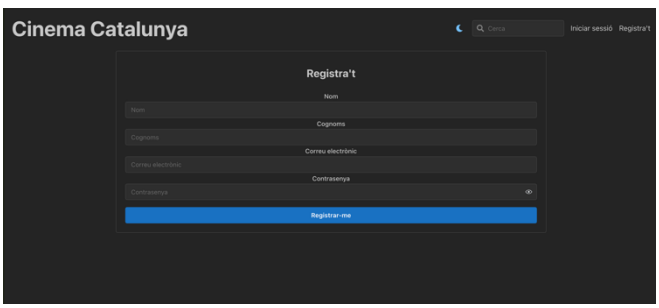


Figura 19 Pantalla per on l'usuari és pot donar d'alta.

Figura 20 Pantalla de perfil d'usuari.

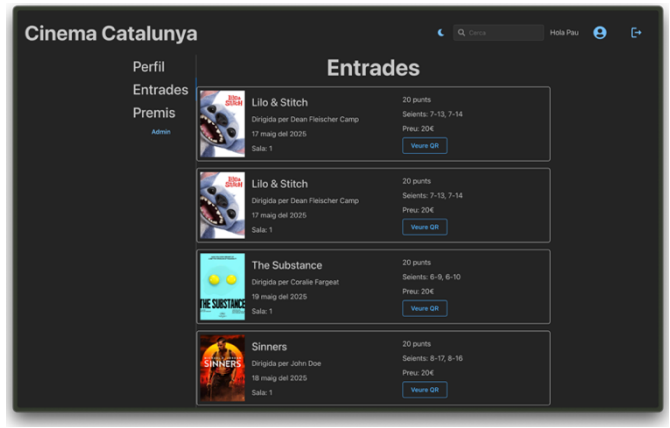


Figura 21 Pantalla del perfil, dins del subapartat d'Entrades

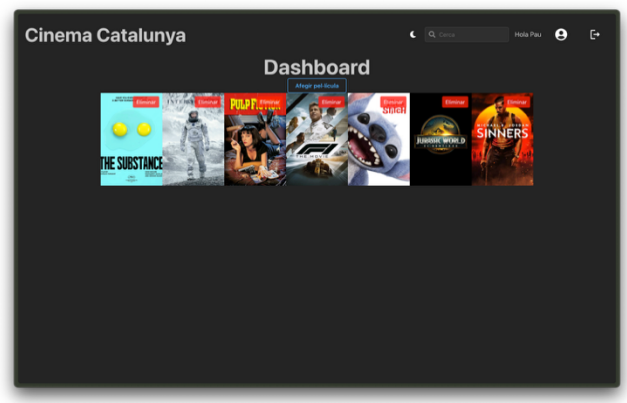


Figura 22 Pantalla per a la gestió de la cartellera, reservada per a administradors.

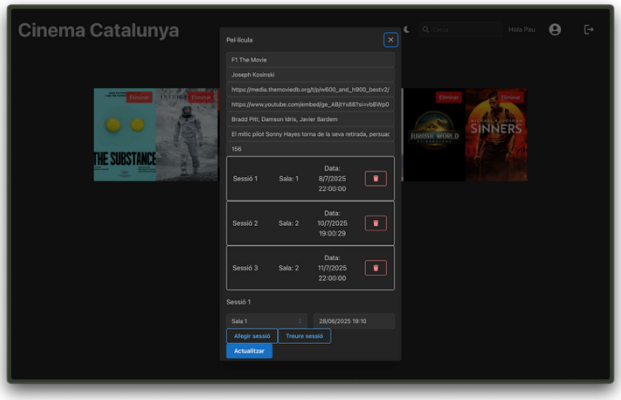


Figura 23 Detall per editar informació i sessions d'una pel·lícula.

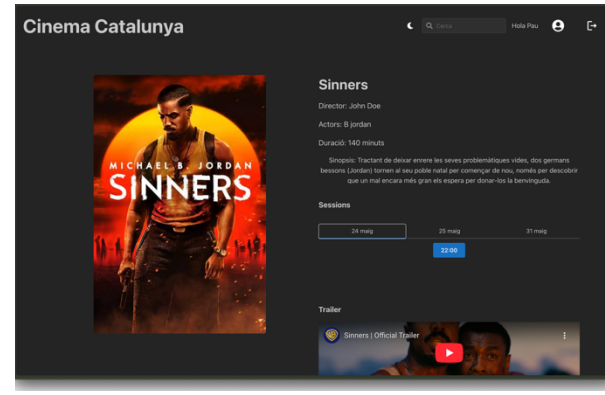


Figura 24 Pàgina del detall de la pel·lícula.