



# Cinet

Pol Bermejo Guzmán  
Franklin Jesús Espino Baquedano  
2º Desenvolupament d'Aplicacions Multiplataforma  
Curs 2021 - 2022

<b>Datos del proyecto</b>	<b>5</b>
<b>Introducción</b>	<b>5</b>
<b>Parte 1 (Gestión del proyecto)</b>	<b>6</b>
<b>Entorno del proyecto</b>	<b>6</b>
Contexto	6
Justificación	6
Stakeholders	6
Soluciones existentes	7
Situación actual	7
Alcance y posibles obstáculos	7
Metodología, validación y herramientas de seguimiento	7
<b>Presupuesto</b>	<b>8</b>
Elementos de la estimación inicial	8
Justificación de los costos estimados	8
Plan de control de los presupuestos	8
Valoración de la viabilidad económica	9
<b>Planificación temporal</b>	<b>9</b>
Fases del proyecto	9
Planificación inicial	9
Punto de control	10
Planificación final	10
<b>Leyes y normativa</b>	<b>11</b>
<b>Parte 2 (Ejecución del proyecto)</b>	<b>12</b>
<b>Análisis</b>	<b>12</b>
Especificación de requisitos	12
Funcionales	12
No funcionales	13
<b>Diseño</b>	<b>14</b>
Arquitectura	14
Seguridad	16
Persistencia	16
Firebase Authentication	16
Firebase Firestore Database	17
Firebase Storage	17
Firebase Realtime Database	17
Interfaz	20
Tecnología	23
Tecnología a desarrollar:	23
Tecnología a utilizar:	24
<b>Desarrollo</b>	<b>25</b>

Estrategia de desarrollo	25
Diagrama de despliegue	26
<b>Pruebas</b>	<b>27</b>
<b>Lanzamiento</b>	<b>30</b>
<b>Conclusiones</b>	<b>31</b>
<b>Bibliografía</b>	<b>31</b>
<b>Anexos</b>	<b>32</b>



Esta obra está sujeta a la licencia de

[Reconocimiento-NoComercial-SinObraDerivada 3.0 España \(CC BY-NC-ND 3.0 ES\)](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## Datos del proyecto

<b>Título</b>	Cinet
<b>Autores</b>	Pol Bermejo Guzmán y Franklin Jesús Espino Baquedano
<b>Tutor</b>	Daniel Martínez Cruz
<b>Descripción</b>	Este proyecto describe una aplicación en que los clientes de Cinet no tengan la necesidad de ir a la taquilla física, sino que directamente reserven a través de la app la hora y película que deseen para así evitar aglomeraciones y que el servicio sea mejor.

## Introducción

En este documento se muestra el desarrollo de una aplicación para los cines Cinet, la aplicación está disponible para dispositivos android.

La idea de la aplicación surge de la necesidad de que no haga falta que los clientes tengan que ir a la taquilla física, sino que a través de la aplicación puedan reservar película, día y hora que mejor les convenga, también se puede reservar menús de comida para que nada más llegar el cine el cliente pueda recoger su comida y entrar a la sala sin necesidad de realizar esperas innecesarias.

Además de la funcionalidad principal de la aplicación que sería hacer reservas también está la sección de administrador, en este apartado solamente podrán acceder los encargados de la taquilla. Otro apartado general que tiene la aplicación es el de promociones que permite comprar menú y entrada a un precio más barato.

La tecnología utilizada ha sido tecnología que hemos ido viendo y utilizado a lo largo de los dos años de grado superior, por lo que los problemas a la hora de integrar las diferentes tecnologías son mínimos. La dificultad encontrada ha sido a la hora de indagar por nuestra cuenta para tratar de conseguir los objetivos que no podíamos cumplir con la experiencia obtenida en los dos años del grado.

# Parte 1 (Gestión del proyecto)

## Entorno del proyecto

### Contexto

El desarrollo de la aplicación surge debido a la colas y esperas que se generan a la hora de comprar entradas y validar promociones, sobre todo los días punta como son los viernes y los sábados.

Como solución se ha pensado en desarrollar una aplicación la cual ayude a gestionar estos problemas permitiendo al usuario realizar sus reservas de forma online sin necesidad de acudir a la taquilla.

Desarrollando esta aplicación, seguramente se evitarán las colas y esperas mejorando así la experiencia del usuario, ya que como hemos comentado anteriormente el cliente puede realizar sus reservas de la película que desee a una hora en la que esté disponible sin necesidad de acudir previamente a la taquilla.

### Justificación

La justificación por la cual nos hemos decidido a realizar este proyecto es por la necesidad de Cinet en gestionar las colas y esperas innecesarias mejorando así la experiencia de los usuarios y ganar fuerza en el mercado cinematográfico.

### Stakeholders

- **El cliente**
  - Este usuario da una idea al equipo de desarrollo de las funciones que quiere que haga su app, además de ayudar en el diseño de la aplicación, dando como requisito principal que la aplicación sea intuitiva y fácil de utilizar.
- **El equipo de desarrollo**
  - Se encargará de analizar, diseñar, desarrollar y testear el proyecto.
- **Usuarios finales**
  - La app está enfocada para un grupo de usuarios de una media de edad entre 18 y 60 años.
- **Testers**
  - Personas externas que ayudarán a la hora de realizar el testeo de la aplicación.

## **Soluciones existentes**

Tras realizar un estudio de soluciones similares a Cinet, se ha llegado a la conclusión de que hay bastantes aplicaciones similares en el mercado (como por ejemplo la app de Cinesa). La diferencia entre Cinet y la competencia es que Cinet está desarrollada cien por cien para teléfonos móviles por lo que en ningún momento la aplicación redirige a una web, esto es un punto de comodidad superior al resto ya que el usuario desde el mismo entorno y con una autenticación puede realizar todas las opciones que ofrece la aplicación.

## **Alcance**

### **Situación actual**

El proyecto lo iniciamos de cero tanto en diseño como en desarrollo. Empezaremos analizando las necesidades del cliente y que requisitos tiene que cumplir la aplicación, comparando con la competencia del sector, se buscará que la aplicación cumpla con al menos todas las funcionalidades que tiene la competencia. Una vez analizado el entorno haremos un prototipo en Figma para tener una idea conjunta a la hora de comenzar con el desarrollo. Por último, una vez el diseño esté realizado iniciaremos el desarrollo de la aplicación en Android Studio.

### **Alcance y posibles obstáculos**

Pensamos que es viable la realización del proyecto debido a que el equipo de desarrollo ha ganado experiencia en la tecnología a implantar. Los únicos inconvenientes previstos son la implementación de Firebase con AndroidStudio, ya que ningún integrante del grupo tiene una alta experiencia utilizando esta base de datos.

### **Metodología, validación y herramientas de seguimiento**

Para organizar todas las fases de la aplicación hemos optado por la metodología Scrum para gestionar el conjunto de tareas del desarrollo, durante este se ha mantenido reuniones diarias en tiempo de clase y por whatsapp. Separaremos cada fase del proyecto en diferentes tareas las cuales se asignaran a los miembros del equipo de desarrollo en base a los conocimientos de los que dispone cada miembro.

Para gestionar quién realiza cada tarea y el tiempo que tarda en realizarla utilizaremos un diagrama de Gantt a través de la herramienta GanttProject.

También se gestiona las horas globales de todo el equipo de desarrollo en un google sheets, en el cual se encuentra el total de horas disponibles para cada tarea y las horas reales que se ha tardado en realizar.

## Presupuesto

### Elementos de la estimación inicial

Para realizar este proyecto tenemos el presupuesto dividido en dos tipos de gastos, tenemos los costes de bienes materiales y los costes humanos. El presupuesto estimado total es de **3823,05 €** aunque conforme vaya evolucionando el proyecto es probable que el coste total del presupuesto aumente o disminuya.

- Costes bienes materiales
  - 2 ordenadores → 1100 €
  - Licencias de software → 0 €
  - Licencia Google Play → 22,55 €
- Costes humanos
  - Analista (16h) → 266,72 €
  - Diseñador (36h) → 387,72 €
  - Desarrollador (135h) → 2.007,45 €
  - Tester (3h) → 38,61 €

### Justificación de los costos estimados

Para justificar el presupuesto hemos calculado lo que cobra un analista, diseñador, desarrollador y tester por horas multiplicado por las horas que tenemos planificadas que nos va a llevar cada parte del proyecto.

A continuación mostramos una tabla de los salarios medios en España:

	Analista	Diseñador	Desarrollador	Tester
Hora	16,67 €	10,77 €	14,87 €	12,87 €
Día	125 €	80,77 €	112 €	96,54 €
Mes	2.708 €	1.750 €	2.417 €	2.092 €
Año	32.500 €	21.000 €	29.000 €	25.100 €

### Plan de control de los presupuestos

En la hipotética situación de que esta aplicación se desarrollase para un uso comercial este es el presupuesto calculado el cual se controla en cuatro pagos.

Hemos establecido diferentes momentos durante el desarrollo en los que se irá cobrando el presupuesto de manera progresiva, para así mantener el rendimiento del equipo de desarrollo y que se vaya valorando su trabajo a medida que el proyecto avanza:



1. El primer pago se realizará al comienzo del proyecto y será un 15% del total del presupuesto.
2. El segundo pago se hará cuando ya tengamos todo preparado para comenzar con el desarrollo de la aplicación y será un 25% del total del presupuesto.
3. El tercer pago se cobrará cuando la aplicación esté preparada para ser testeada y simplemente quede solucionar errores durante los tests. El pago será de un 25% del total del presupuesto.
4. El cuarto y último pago será cobrado cuando la aplicación ya haya sido lanzada y esté disponible en la Google Play Store, este pago será la cantidad restante del presupuesto marcado (contando si el presupuesto a ido variando o no).

## Valoración de la viabilidad económica

Los ingresos se irán realizando en 4 momentos diferentes del desarrollo. Lo hacemos de esta manera para garantizar que a medida que vamos evolucionando en el proyecto obtenemos recompensa económica, además de que de esta forma podemos evitar malentendidos con el cliente.

## Planificación temporal

### Fases del proyecto

El proyecto se divide en 4 fases principales:

- Análisis
- Diseño
- Desarrollo
- Pruebas

Hemos planificado una cantidad de horas que no són 100% definitivas pero nos hemos basado en la dificultad que pensamos que nos vamos a encontrar, aunque al terminar con el proyecto nos daremos cuenta que la cantidad de horas variará y de normal subirá.

Además de ser la aplicación la parte del proyecto que más tiempo nos ocupa también tenemos horas dedicadas a la redacción de la memoria y a la preparación de la presentación oral.

### Planificación inicial

	Horas Planificadas	Horas Reales	Estado
Proyecto	198	0	-198

Aplicación	181	0	-181
Análisis	7	0	-7
Requisitos	2		-2
Objetivos	2		-2
Tecnología	3		-3
Diseño	36	0	-36
Arquitectura - Diseño de clases	4		-4
Interfaz - Mockups	12		-12
Base de datos	20		-20
Desarrollo	135	0	-135
Estrategia de desarrollo	2		-2
Interfaz	25		-25
Backend	93		-93
BD (SQL)	15		-15
Pruebas	3	0	-3
Usabilidad	3		-3
Memoria	9	0	-9
Redacción	8		-8
Revisión	1		-1
Presentación oral	8	0	-8
Presentació oral	2		-2
Materiales (power point, etc.)	6		-6

## Punto de control

## Planificación final

	Horas Planificadas	Horas Reales	Estado
Proyecto	198	235	37
Aplicación	181	201	20
Análisis	7	9	2
Requisitos	2	1	-1
Objetivos	2	2	0
Tecnología	3	6	3

Diseño	36	38	2
Arquitectura - Diseño de clases	4	3	-1
Interfaz - Mockups	12	10	-2
Base de datos	20	25	5
Desarrollo	135	151	16
Estrategia de desarrollo	2	1	-1
Interfaz	25	17	-8
Backend	93	108	15
BD (SQL)	15	25	10
Pruebas	3	3	0
Usabilidad	3	3	0
Memoria	9	28	19
Redacción	8	26	18
Revisión	1	2	1
Presentación oral	8	6	-2
Presentació oral	2	2	0
Materiales (power point, etc.)	6	4	-2

## Leyes y normativa

Cinet es una aplicación cuyo propósito es gestionar datos del producto que vende y datos personales tanto de los usuarios como de la empresa que nos contrata, por ese motivo, es muy importante asegurarse de que la aplicación cumpla con cuyas leyes que regulan el tratamiento de datos personales y los derechos que los usuarios pueden ejercer. Algunas normativas vigentes que debe de cumplir la aplicación son:

### LOPD y RGPD:

Para cumplir con la ley orgánica de protección de datos y con el reglamento general de protección de datos debemos de cumplir con una serie de requisitos. A continuación algunos de los requisitos:

- Notificar las brechas de seguridad que pongan en riesgo la privacidad de los usuarios
- Establecer un código de conducta
- Realizar un análisis de riesgos y adoptar medidas de seguridad
- Mantener un registro de actividades del tratamiento de datos personales
- Proteger los datos desde el diseño y por defecto

## Parte 2 (Ejecución del proyecto)

### Análisis

#### Especificación de requisitos

##### Funcionales

##### Login de usuario

Al iniciar la aplicación aparece una pantalla para iniciar sesión. Si el usuario tiene una cuenta creada iniciara sesión, si no se tendrá que registrar introduciendo la información que se le pide.



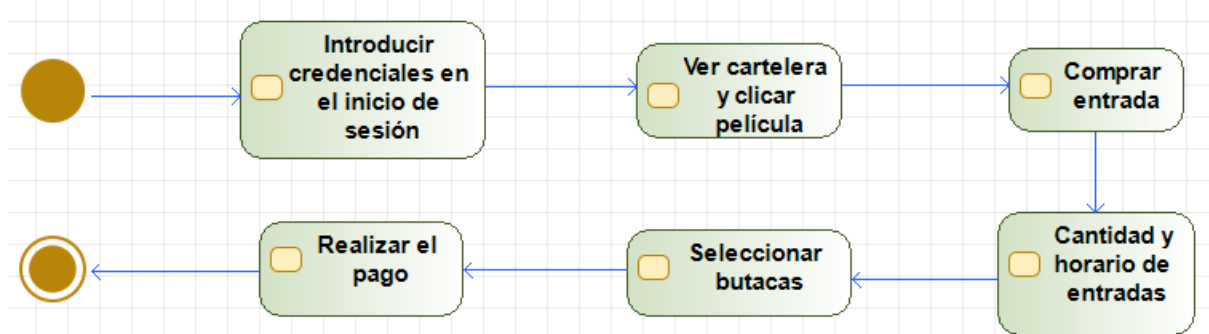
##### Registro de usuario

Esta funcionalidad se utilizará únicamente si el usuario que intenta acceder a la aplicación no tiene una cuenta creada. La información que pedirá esta pantalla para guardarla en la base de datos será: Nombre, Apellidos, Correo electrónico y Contraseña.



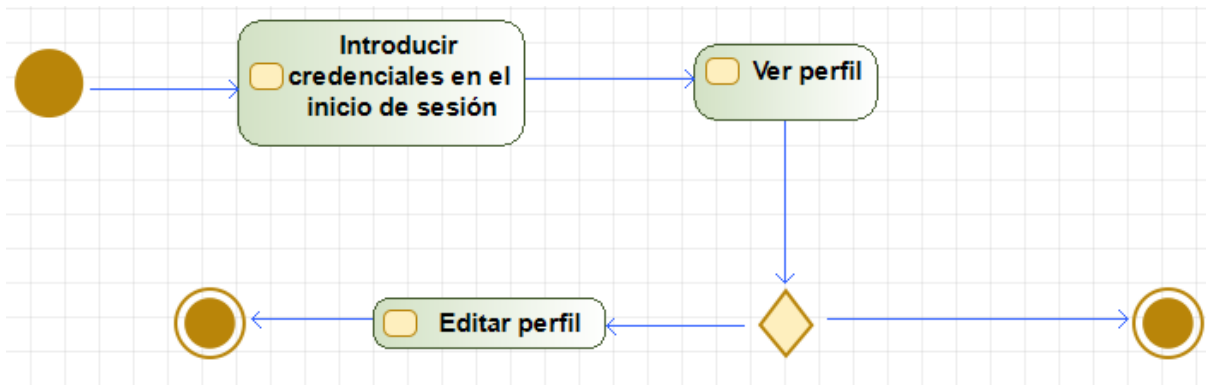
##### Comprar

En la aplicación se puede realizar diferentes compras como comprar entradas, comida para ver la película o diferentes promociones. El usuario añadirá un método de pago previamente para poder realizar las compras.



##### Perfil

En el menú lateral hay un apartado en el que podemos encontrar toda la información de nuestro perfil y editarla.



### Consultar la cartelera

En este apartado se puede consultar las películas disponibles. Para comprar entradas hay tres secciones diferentes (Cartelera, Estrenos, Próximos), en cada sección hay diferentes películas.



### Valorar película

En el momento de mostrar información de la película, el cliente puede dejar una puntuación sobre la película que va del 1 al 5 con una barra de estrellas, siendo 1 la peor puntuación.



## No funcionales

### Fiabilidad

La aplicación tiene que mostrarse como una aplicación segura en la que al usuario le de confianza, ya que se trata información sensible y privada de cada usuario como sus datos de pago o su información personal.

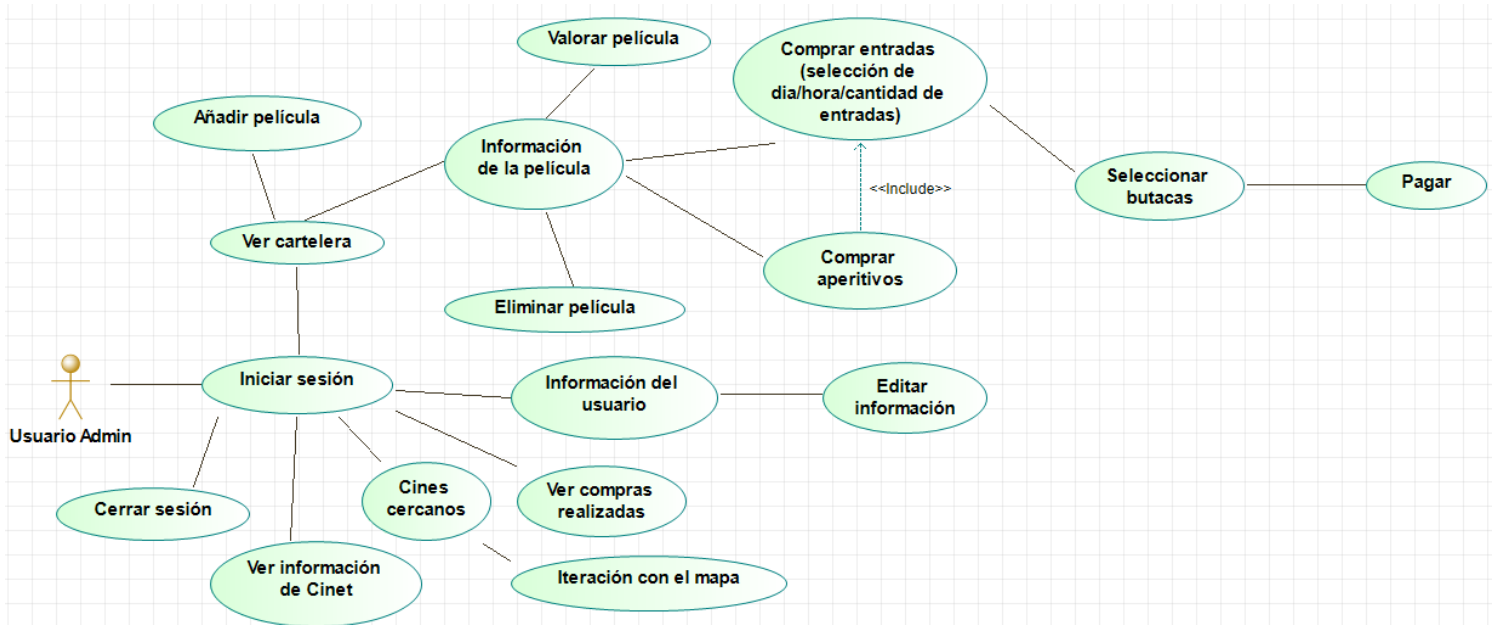
### Eficacia

La aplicación debe de cumplir con todas sus funciones, las cuales serán, tanto para el cliente que es nuestro cliente directo (Cinet), como los usuarios quienes son los que utilizarán la aplicación. Para tener una mejor experiencia y que la app sea eficaz realizaremos varias versiones añadiendo nuevas funcionalidades.

### Fácil aprendizaje



En este diagrama vemos todo lo que puede realizar el usuario administrador dentro de la aplicación. Este usuario tiene disponibilidad total de la aplicación, por lo que además de poder realizar todas las funcionalidades de la aplicación, también puede gestionar las películas (tanto añadir películas como eliminarlas de la cartelera).



## Seguridad

Lo más importante en un proyecto es la seguridad y privacidad de la información de los usuarios, por eso la aplicación utiliza Firebase para el almacenamiento de todos los datos de la aplicación.

Para obtener acceso a la aplicación el usuario se registrará utilizando el sistema de autenticación que nos ofrece Firebase llamado Firebase authentication. Una vez el usuario haya estado registrado y verifique el correo electrónico, podrá acceder a toda la información de la aplicación. Cada vez que el usuario quiera entrar a la app, esta verificará si existe el usuario.

La información de la cuenta de cada usuario, de las películas, de las compras y la gestión de las entradas se registra en Firebase de forma segura.

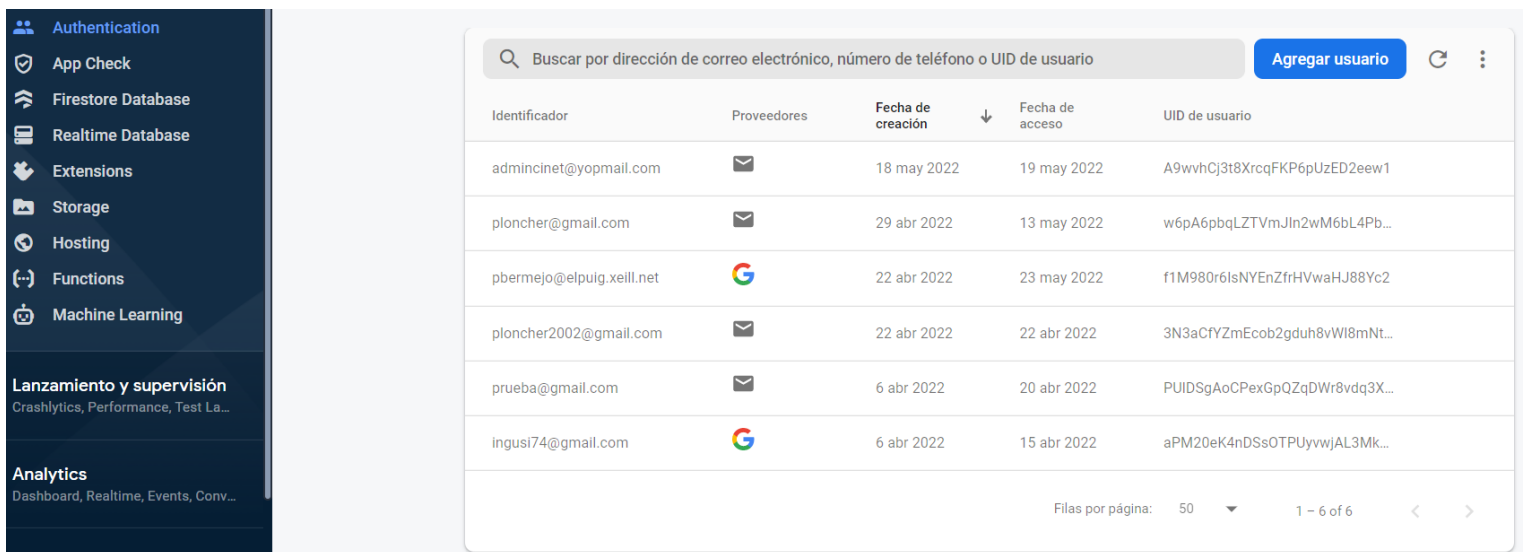
## Persistencia

Aprovechando que **Firestore** dispone de diferentes funcionalidades de almacenamiento, hemos utilizado varias de ellas dependiendo de la información que

se necesite guardar. Por lo que toda la persistencia de datos de la aplicación se ha gestionado a través de Firebase. A continuación veremos una breve explicación de lo que hace cada una de ellas y sus formatos de almacenamiento.

## Firebase Authentication

En este apartado se gestiona toda la información de los usuarios, en el panel de control se puede ver la información de cada usuario, cuando ha sido su último login o que tipo de cuenta tiene (correo/contraseña o google sign In). También permite gestionar correos electrónicos y sms a la hora de verificar un usuario nuevo.



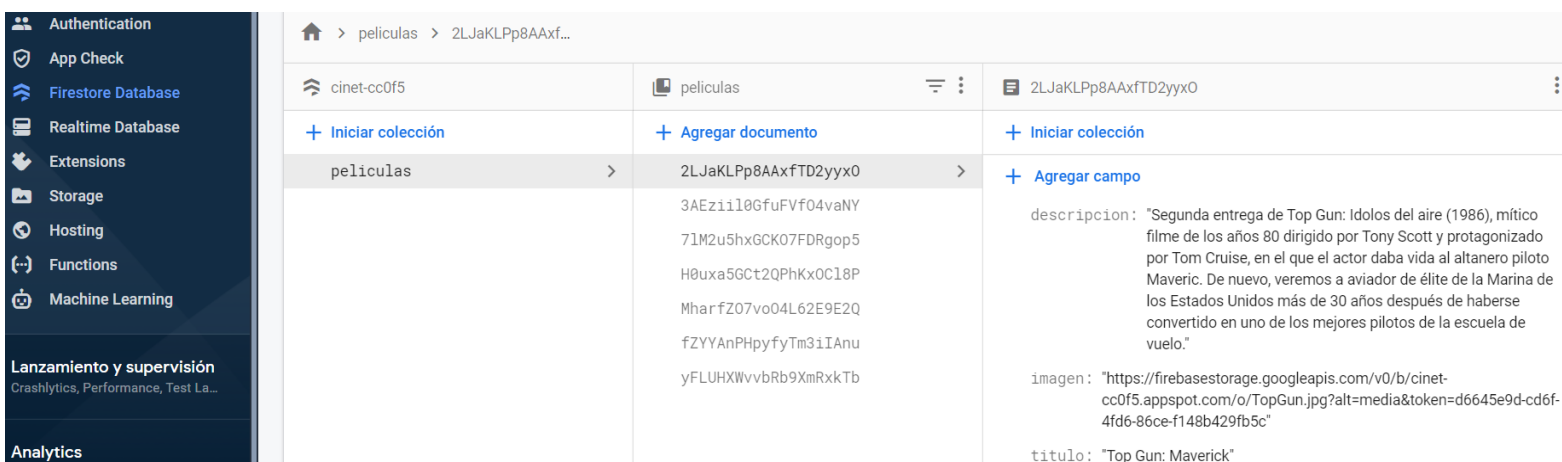
The screenshot shows the Firebase Authentication console. On the left is a navigation menu with options like Authentication, App Check, Firestore Database, Realtime Database, Extensions, Storage, Hosting, Functions, Machine Learning, Lanzamiento y supervisión, and Analytics. The main area displays a table of users with the following columns: Identificador, Proveedores, Fecha de creación, Fecha de acceso, and UID de usuario.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
admincinet@yopmail.com	✉	18 may 2022	19 may 2022	A9vvhCj3t8XrcqFKP6pUzED2eew1
ploncher@gmail.com	✉	29 abr 2022	13 may 2022	w6pA6pbqLZTVmJln2wM6bL4Pb...
pbermejo@elpulg.xelll.net	🌐	22 abr 2022	23 may 2022	f1M980r6IsNYEnZfrHVwaHJ88Yc2
ploncher2002@gmail.com	✉	22 abr 2022	22 abr 2022	3N3aCfYZmEcob2gduh8vWI8mNt...
prueba@gmail.com	✉	6 abr 2022	20 abr 2022	PUIDSgAoCPexGpQZqDWr8vdq3X...
ingusi74@gmail.com	🌐	6 abr 2022	15 abr 2022	aPM20eK4nDSs0TPuyvwjAL3Mk...

At the bottom of the table, it shows 'Filas por página: 50' and '1 - 6 of 6'.

## Firebase Firestore Database

En este apartado almacenamos toda la información relacionada con las películas. Cada película dispone de un título, una descripción, y un link a su imagen de portada gestionado con Firebase Storage a través de un google.api. Si se requiere información adicional sobre la película se guarda en su document "nodo/id".



The screenshot shows the Firebase Firestore Database console. The left sidebar is the same as in the previous image. The main area shows a document view for a collection named 'películas'. The document ID is '2LJaKLPp8AAxfTD2yxx0'. The document content is as follows:

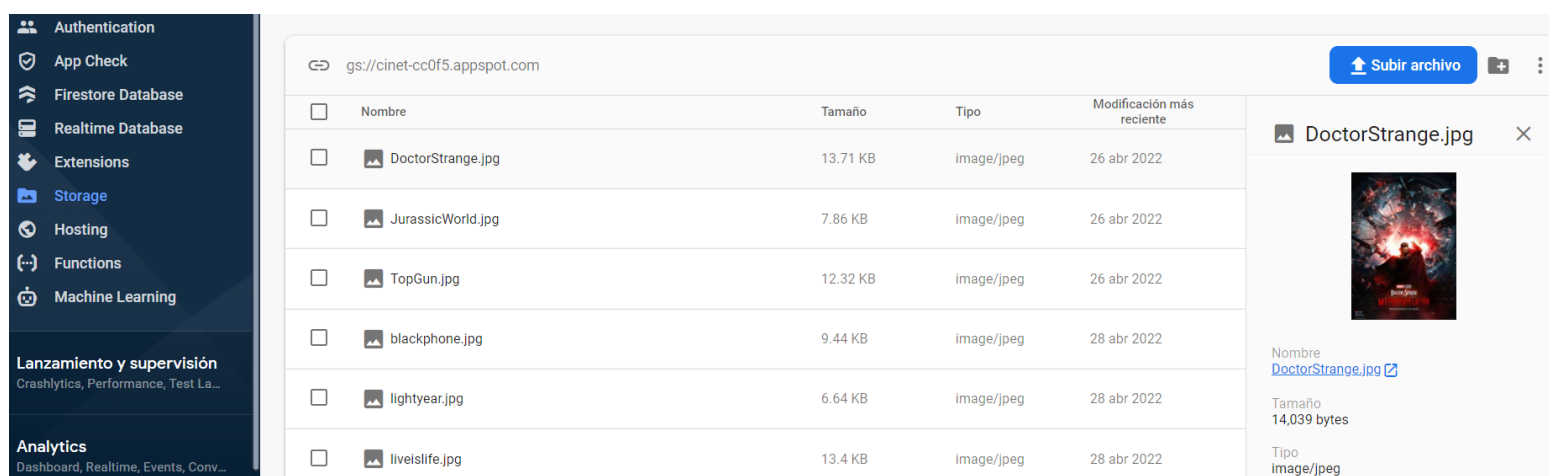
```

descripcion: "Segunda entrega de Top Gun: Idolos del aire (1986), mítico filme de los años 80 dirigido por Tony Scott y protagonizado por Tom Cruise, en el que el actor daba vida al altanero piloto Maverick. De nuevo, veremos a aviador de élite de la Marina de los Estados Unidos más de 30 años después de haberse convertido en uno de los mejores pilotos de la escuela de vuelo."
imagen: "https://firebasestorage.googleapis.com/v0/b/cinet-cc0f5.appspot.com/o/TopGun.jpg?alt=media&token=d6645e9d-cd6f-4fd6-86ce-f148b429fb5c"
titulo: "Top Gun: Maverick"
  
```



## Firestore Database

En este apartado almacenamos todas las imágenes de portada de las películas. Al almacenarla Firebase genera un link google.api el cual guardamos en el campo imagen de cada película registradas en el apartado anterior.



The screenshot shows the Firebase Storage console interface. On the left is a navigation sidebar with categories like Authentication, App Check, Firestore Database, Realtime Database, Extensions, Storage (highlighted), Hosting, Functions, and Machine Learning. Below these are sections for 'Lanzamiento y supervisión' and 'Analytics'. The main area displays a table of files stored in the bucket 'gs://cinet-cc0f5.appspot.com'. The table has columns for 'Nombre', 'Tamaño', 'Tipo', and 'Modificación más reciente'. Below the table, a detailed view of the file 'DoctorStrange.jpg' is shown, including its thumbnail, name, size (14,039 bytes), and type (image/jpeg).

Nombre	Tamaño	Tipo	Modificación más reciente
DoctorStrange.jpg	13.71 KB	image/jpeg	26 abr 2022
JurassicWorld.jpg	7.86 KB	image/jpeg	26 abr 2022
TopGun.jpg	12.32 KB	image/jpeg	26 abr 2022
blackphone.jpg	9.44 KB	image/jpeg	28 abr 2022
lightyear.jpg	6.64 KB	image/jpeg	28 abr 2022
liveslife.jpg	13.4 KB	image/jpeg	28 abr 2022

## Firestore Database

En este apartado es más antiguo que Firestore Database, pero aun así su función en la aplicación es gestionar la disponibilidad de las entradas de cada película, además de la información de todos los usuarios registrados y gestionar todas las compras realizadas en la aplicación.

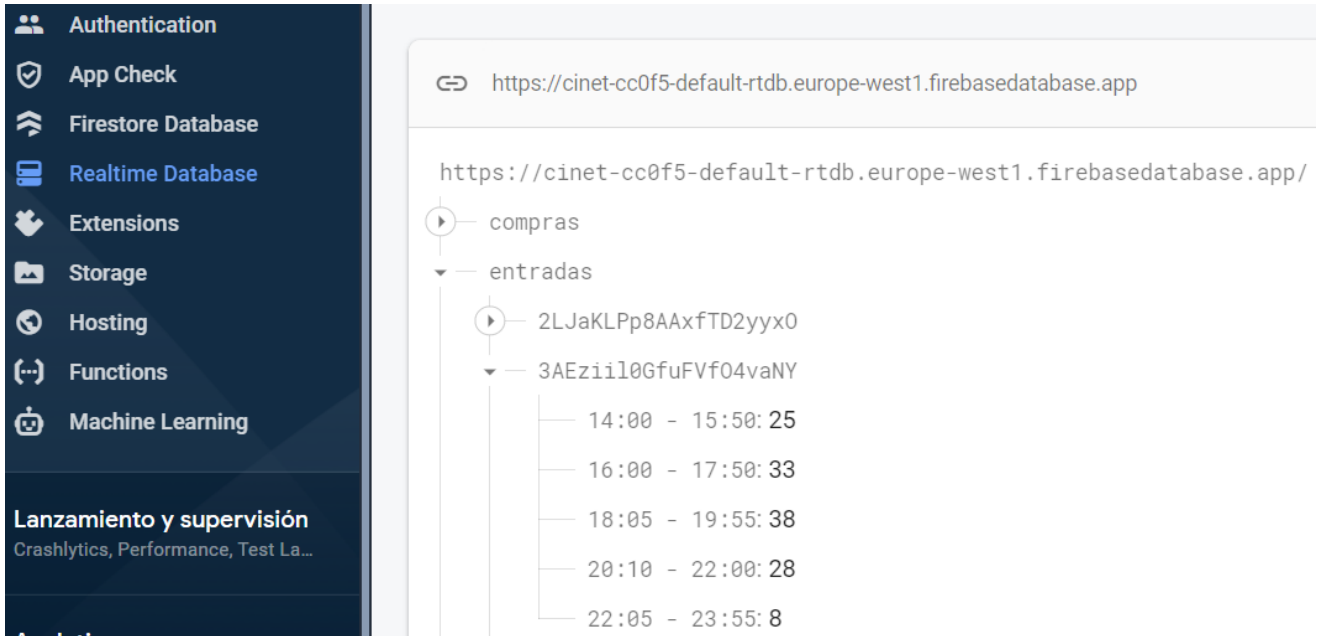


The screenshot shows the Firebase Realtime Database console. The left sidebar is the same as in the previous image, with 'Realtime Database' highlighted. The main area displays the database URL: 'https://cinet-cc0f5-default-rtdb.europe-west1.firebaseio.com'. Below the URL, a tree view shows the database structure with three nodes: 'compras', 'entradas', and 'usuarios'.

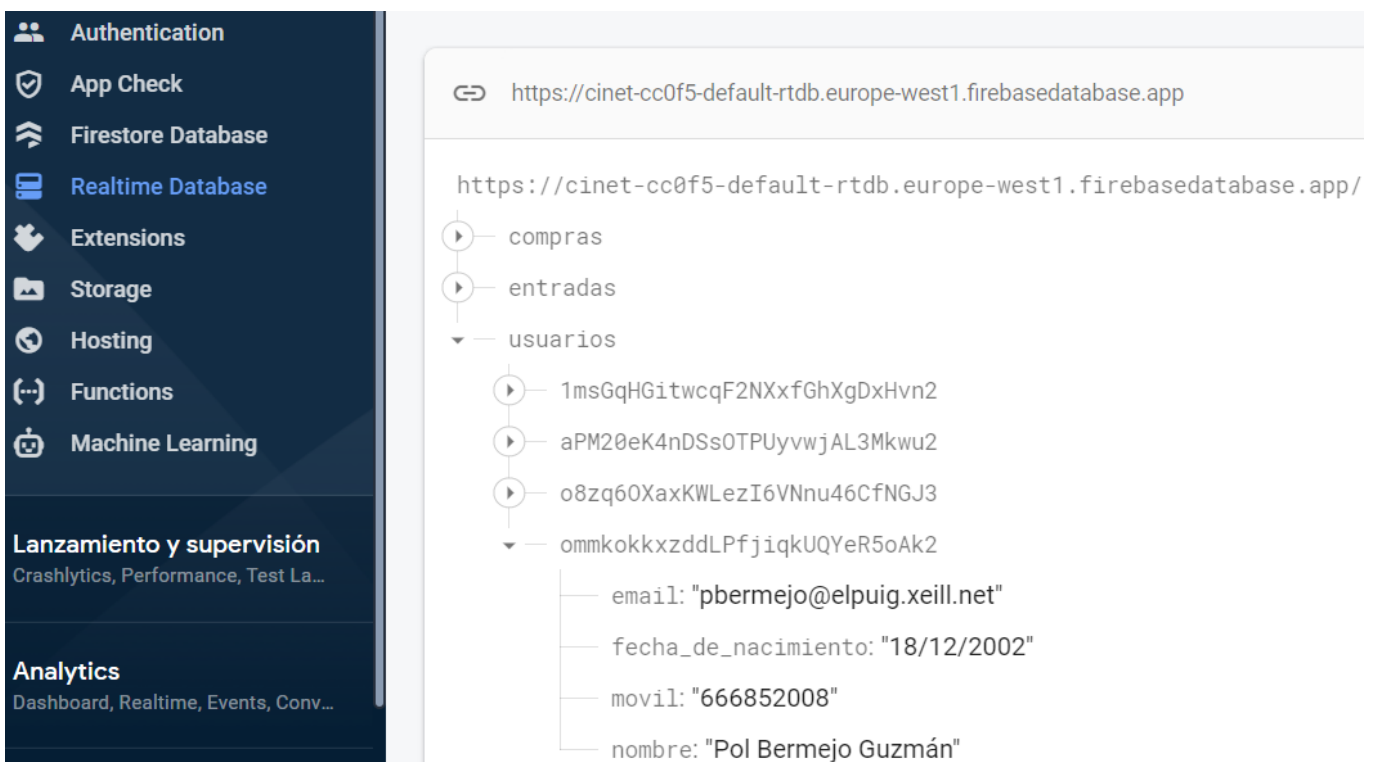
```

https://cinet-cc0f5-default-rtdb.firebaseio.com/
├── compras
├── entradas
└── usuarios
  
```

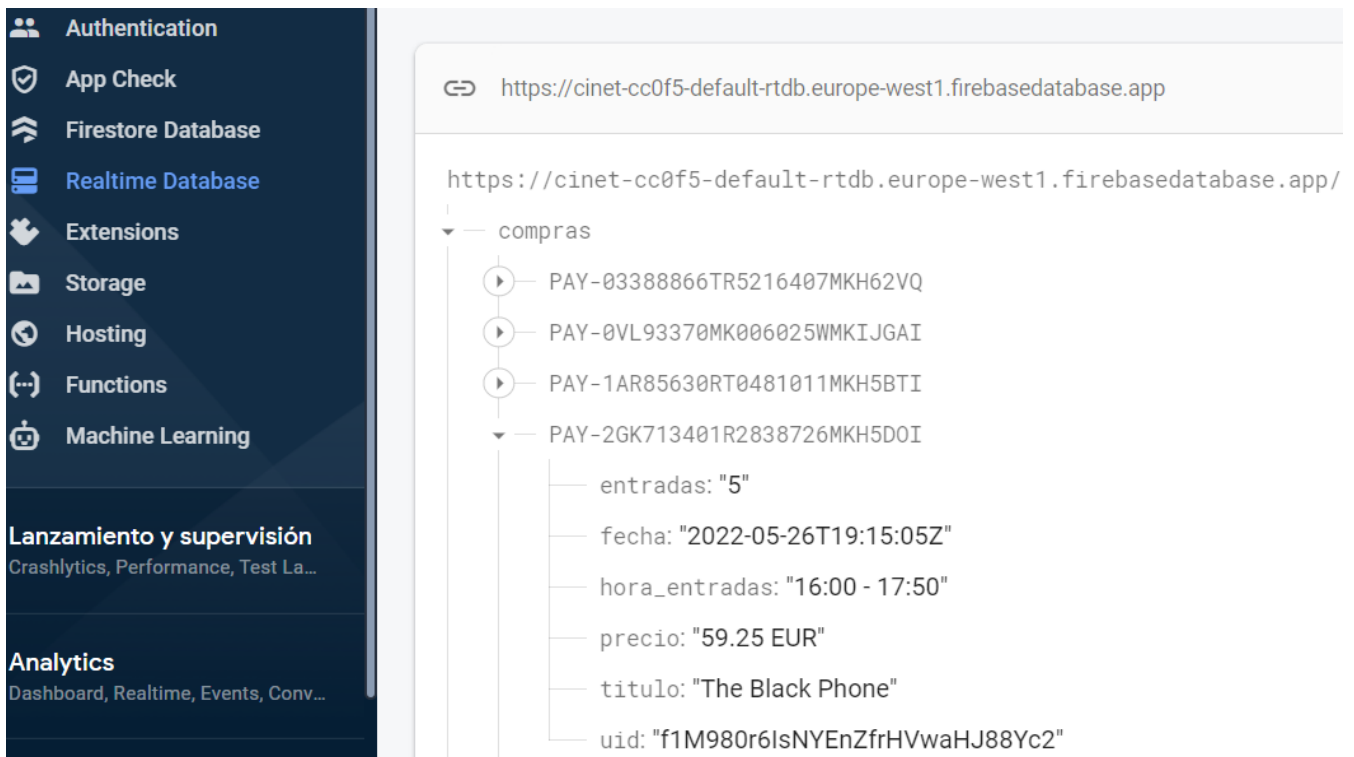
Por cada película hay un registro con los diferentes horarios y la cantidad de entradas en cada uno. Al seleccionar entradas la aplicación hace una consulta para saber cuántas entradas hay disponibles y proseguir con la compra. Si hay suficientes entradas en el momento de pagar, las entradas seleccionadas se descuentan de la base de datos.



Los usuarios a la hora de registrarse, la aplicación lanza una query para insertar todos los campos de registro en el campo `"usuarios"`. Si el usuario inicia sesión con Google se insertará la información del usuario en el primer inicio de sesión que realice.



Cada registro de compra se genera automáticamente gracias a PayPal sdk, por lo que la información que se guarda en la base de datos se extrae de la información seleccionada en la selección de entradas y de la información de compra obtenida de PayPal.



The screenshot displays the Firebase console interface. On the left, a dark sidebar contains navigation options: Authentication, App Check, Firestore Database, Realtime Database, Extensions, Storage, Hosting, Functions, and Machine Learning. Below these are sections for 'Lanzamiento y supervisión' and 'Analytics'. The main content area shows the URL `https://cinet-cc0f5-default-rtdb.europe-west1.firebaseio.com` and a tree view of the database. The 'compras' collection is expanded, showing a list of purchase records with their details.

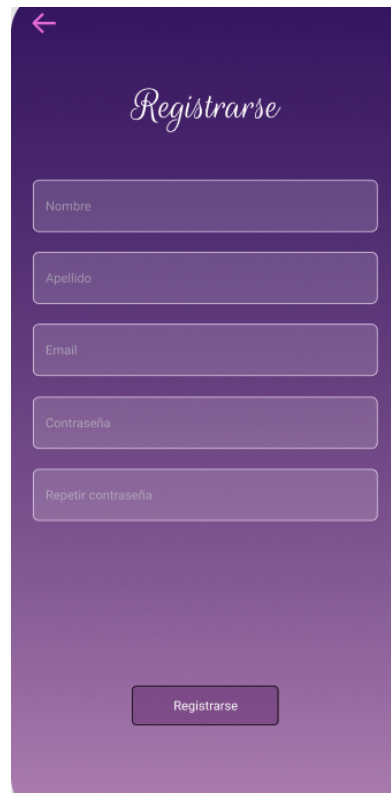
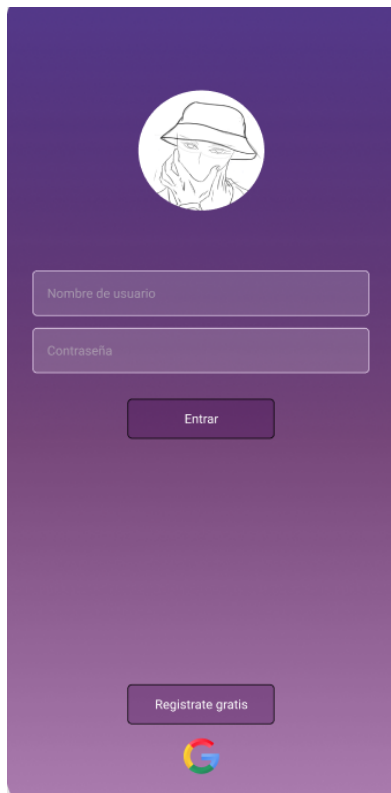
Key	Value
PAY-03388866TR5216407MKH62VQ	
PAY-0VL93370MK006025WMKIJGAI	
PAY-1AR85630RT0481011MKH5BTI	
PAY-2GK713401R2838726MKH5D0I	<ul style="list-style-type: none"><li>entradas: "5"</li><li>fecha: "2022-05-26T19:15:05Z"</li><li>hora_entradas: "16:00 - 17:50"</li><li>precio: "59.25 EUR"</li><li>titulo: "The Black Phone"</li><li>uid: "f1M980r6lsNYEnZfrHVwaHJ88Yc2"</li></ul>

## Interfaz

A continuación podemos ver un diseño muy simple realizado con **Figma**, sobre la idea principal de la aplicación (Pantallas más importantes y colores del entorno). A partir de este concepto se comenzará el desarrollo de la aplicación. Es posible que durante el desarrollo se omitan algunas ideas principales, ya sea por dificultad o por mejorar la interacción con la app.



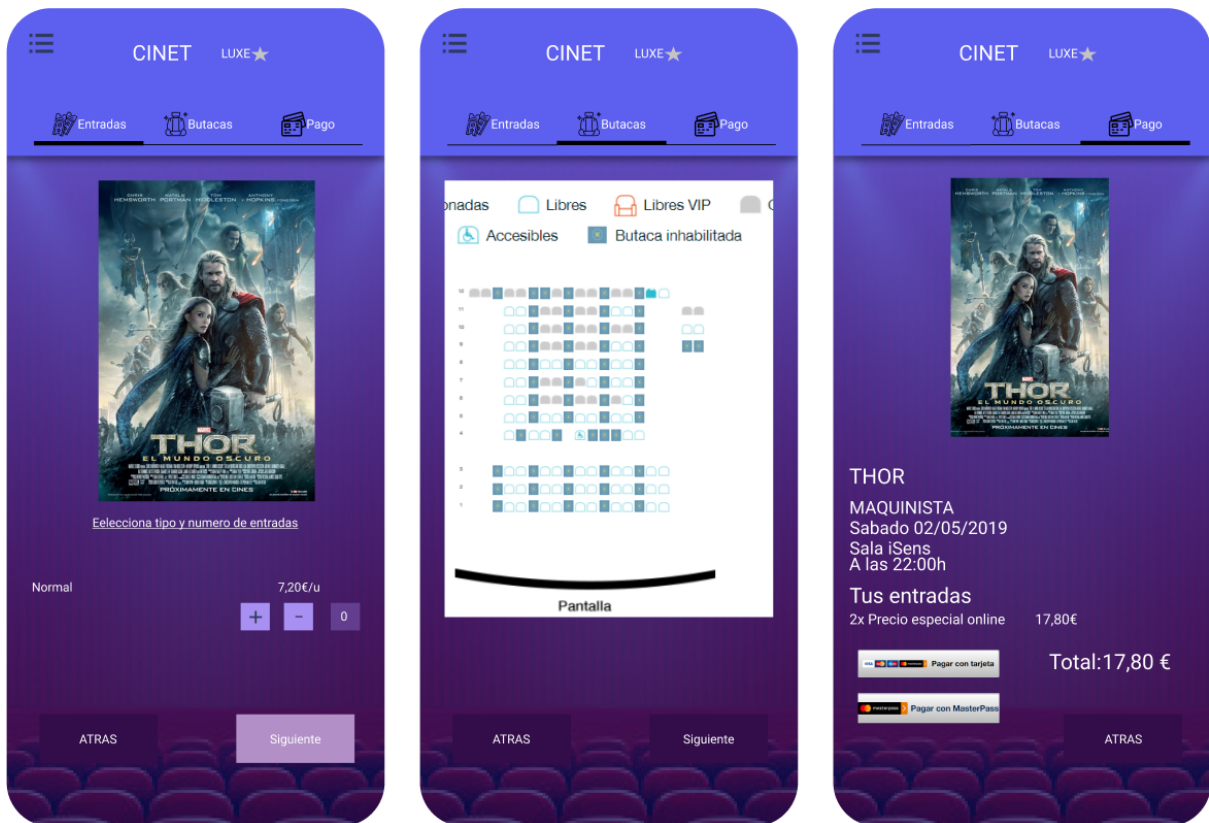
### Autenticación del usuario:



### Consultar cartelera:





**Compra de entradas:**

Link: <https://www.figma.com/file/A5oSq1UJFASG65wo1Bu8wq/Untitled?node-id=0%3A1>

**Tecnología****Tecnología a desarrollar:****Android**

Es un sistema operativo basado en Linux, inicialmente pensado para teléfonos móviles. Hemos utilizado este sistema operativo para desarrollar la aplicación porque es el sistema que hemos utilizado a lo largo del curso para programar aplicaciones móviles.

**Java**

Como lenguaje principal utilizamos Java ya que es el lenguaje que hemos utilizado durante estos dos años de grado superior, además de ser el lenguaje principal de Android.



### **XML**

XML es la manera de maquetar en android y es la que hemos utilizado durante el segundo año del grado, por lo que estamos bastante familiarizados con ella.



### **Firebase**

Es una plataforma que nos ofrece gestionar los datos de nuestra aplicación y la información que ésta utiliza. En ella almacenamos la información de películas, la gestión de entradas y el registro de los usuarios tanto de correo/contraseña como los usuarios que inician sesión con google.

## **Tecnología a utilizar:**



### **Herramientas ofimáticas (Google Docs, Google Sheets, Google Slides)**

Estas herramientas tienen como utilidad ayudar en la creación de toda la documentación relacionada con el proyecto (Memoria, Seguimiento de horas invertidas, Presentación).



### **Google Drive**

Google Drive es una herramienta de Google que permite almacenar todo tipo de ficheros en sus servidores (nube). En ella almacenaremos los documentos creados con las herramientas ofimáticas.



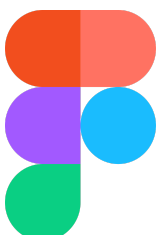
### **Modelio**

Modelio es una herramienta que permite hacer diagramas UML, esta herramienta es de código abierto y sigue los estándares UML. La usaremos para crear los diagramas necesarios para la fase de análisis.



### **GitHub**

GitHub es un sistema de gestión de versiones que tiene como utilidad alojar el proyecto y todos los cambios que se vayan realizando durante el desarrollo.



### **Figma**

Hemos utilizado Figma para desarrollar las interfaces de la aplicación antes de iniciar el desarrollo. Con Figma hemos plasmado el diseño de la interfaz que posteriormente hemos utilizado de partida para la fase de desarrollo.



### **Android Studio**

Android Studio es un IDE que nos permite desarrollar Android. En este IDE hemos desarrollado el frontend y el backend de la aplicación, además nos permite inicializar la aplicación en un móvil android.



### **GanttProject**

GanttProject es una herramienta que permite hacer diagramas de Gantt, esta herramienta es de código abierto y su función en el proyecto será ayudar con la gestión de las tareas a realizar.



### **PayPal Developer**

PayPal Developer es un servicio de PayPal que permite la integración (sdk) de PayPal durante el desarrollo para comprobar todo lo relacionado con transacciones a través de cuentas ficticias (SandBox)

## **Desarrollo**

### **Estrategia de desarrollo**

La estrategia de desarrollo seleccionada ha sido “Pair Programming”, un miembro del equipo va desarrollando la aplicación, mientras que otro miembro del equipo revisa el código y da ideas para poder mejorar la efectividad del código. Hemos optado por esta estrategia para que la integración en el desarrollo del proyecto sea más o menos de forma equitativa y además que al ser programadores juniors está bien que siempre haya un miembro del equipo de desarrollo revisando, para corregir posibles errores o dar ideas que pueden mejorar el funcionamiento de la aplicación.

El plan de ruta planteado y que hemos seguido es el siguiente:

1. Crear los principales Fragments (SignInFragment, RegisterFragment, ...)
2. Vincular Firebase Authentication con el proyecto en AndroidStudio y realizar un Login y un Registro funcionales desde el primer momento
3. Introducir la navegación sobre las diferentes pantallas, añadiendo un Draw Menu y un Tab Menu.
4. Vincular Firebase Storage y Firebase Firestore Database para tener almacenadas la información de todas las películas vinculadas con su imagen de portada almacenadas en el storage. Desde Firebase Storage se genera un link con una Google api, y este link lo registramos en el Firebase Firestore Database para recuperarlo en el Android Studio.
5. Vincular Firebase Realtime Database con Android Studio para gestionar las entradas de cada película. A través de una query saber cuantas entradas hay



disponibles para proseguir con la compra o avisar al usuario de que no puede continuar. Se utiliza el id del Document de cada película como referencia entre Firebase Realtime Database y Firebase Firestore Database.

6. Aprovechar la vinculación con Firebase Realtime Database para gestionar más información sobre los usuarios a través del UID (User ID) creado por Firebase Authentication.
7. Vincular Paypal Sdk con Android para realizar la gestión de las compras de cada usuario en Firebase Realtime Database. En el registro de compras se almacena información que nos facilita Paypal Sdk como: id transacción o fecha de la compra.

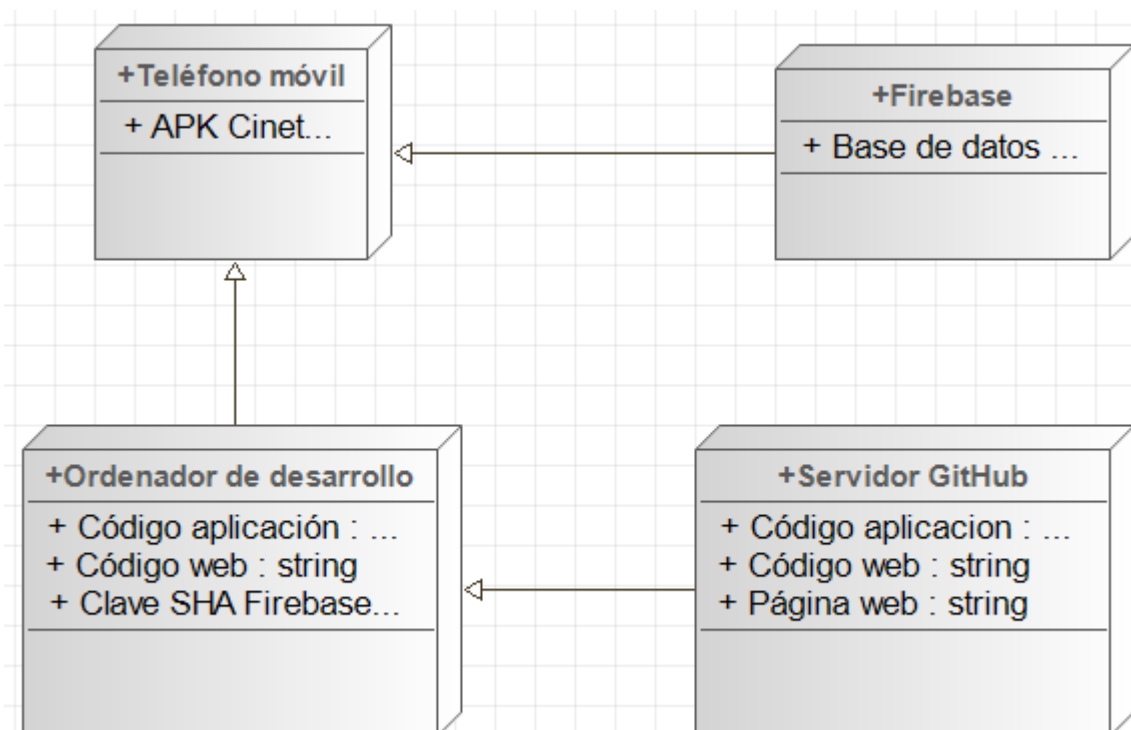
Por otro lado almacenamos recogida de otros fragments como son: el número de entradas, hora de las entradas, precio final, título de la película.

Por último también registramos el UID (User ID) como vinculación de la compra al usuario.

## Diagrama de despliegue

En este proyecto disponemos de cuatro dispositivos hardware:

- El servidor de Github donde guardamos el código de la aplicación, además del código de la web y el hosting de esta última.
- Los ordenadores donde hemos desarrollado el proyecto. En estos tenemos el código del proyecto y de la web además de la clave SHA generada como permiso para el servidor de Firebase.
- Teléfono móvil donde se ha ido comprobando los resultados del desarrollo. En este simplemente se encuentra el APK de la aplicación, ya sea descargada desde la página web de descarga o lanzada desde el ordenador.
- Por último tenemos el servidor de Firebase, el cual almacena toda la información de la aplicación (usuarios, compras, películas, entradas ...).



## Pruebas

Durante el desarrollo hemos realizado pruebas de seguimiento a la app para comprobar su correcto funcionamiento. Estos son los puntos que se han ido comprobando:

### Interfaz

Al nivel de interfaz, hemos realizado pruebas para mejorar el aspecto visual de la aplicación, además de que se muestre tal y como se espera algunas de estas pruebas son:

- Comprobar que la fusión de colores fuese adecuada y no que resultase demasiado.
  - Comprobar el tamaño de los botones, para saber si son adecuados y si están en la comodidad que corresponde, también su color y el de su letra para adaptarlos al entorno de la aplicación
  - Comprobar la navegación de la aplicación para ver si se puede navegar por todas las pantallas añadidas, también el comportamiento de los dos menú (DrawMenu y TabMenu).
- Comprobar el diseño, para ver si se adapta a todo tipos de pantallas

### Inicio de Sesión:

Las pruebas realizadas en este apartado son las siguientes:

- Comprobar que el usuario este en la base de datos cuando intente iniciar sesion
- Comprobar que el usuario acceda a la aplicación al verificar su estado en la base de datos.

### Registro:

- Comprobar que el usuario este en la base de datos cuando intente iniciar sesión y si no lo esta que lo añada con la información que se le pide
- Comprobar que el usuario acceda a la aplicación al acabar y verificar su registro en la base de datos

#### **Perfil de usuario:**

Con este paso comprobamos que el usuario tenga dentro de la app su informacion de forma grafica y ordenada

### **Back-end**

#### **Inicio de Session:**

Con respecto al inidio de sesion se han realizado las siguientes pruebas para asegurar el correcto funcionamiento de la app:

- Intentar acceder con un usuario y contraseña do validas
- Intentar accedesr con una cuenta registrada poniendo el correo correcto pero una contraseña erronea
- Intentar acceder con un usuario y contraseñas registrados correctamente
- probar el inicio de sesion dejando los parametros vacios
- inicuar sesion con un correo incorrecto pero pero utilizando una contraseña registrada correctamente

#### **Registro:**

Con la parte de registro se han realizado las siguientes pruebas en relacion a este apartado:

- añadir un usuario ya existente
- probra el registro con los parametros vacios
- intentar acceder con un usuario y contraseña registradas correctamente

### **Añadir películas:**

Durante el proceso de desarrollo de la aplicación en lo que a añadir películas se refiere se han realizado las siguientes pruebas:

- añadir una película con todos los campos necesarios
- añadir una película la cual no consta en la base de datos
- añadir una película utilizando un ID que no corresponde
- Garantizar que solo se puede hacer si estas desde modo Administrador

### **Eliminar películas:**

Para llevar a cabo esta funcionalidad hemos realizado las siguientes pruebas:

- eliminar una película inexistente
- Eliminar una película correctamente con todos sus campos
- Garantizar que solo se puede hacer si estas desde modo Administrador

### **Usabilidad:**

Para comprobar el nivel de usabilidad de la aplicación, se solicitó a un usuario que probara la aplicación durante un corto tiempo, evaluando así la aplicación dándonos datos necesarios sobre el uso de la app, los detalles se mostrarán a continuación:

## Lanzamiento

El principal objetivo de esta aplicación es el de llegar a l máximo posible de usuarios por lo cual se opto para hacer de una forma mas intuitiva, seguidamente se explicaran las tareas que se haran para su distribución con el fin de que la aplicacioinn desa accesible a todos los usuarios.

### **Publicacion:**

Para la publicación de la aplicación se deben cumplir varios requisitos tando en la App store de Apple y Google play store:

### **App Store:**

Para poder publicar la aplicación en app store se necesita disponer de una cuenta en esta plataforma que se obtiene de manera gratuita, registrándote en la plataforma con un correo electrónico, después de ya tener la cuenta creada, se deberá de dar de alta en el programa de desarrolladores de Apple, el cual a diferencia de Google Play Store, tendrá un coste elevado, en concreto el coste anual seria de 99 euro, Tras la verificación por parte de App conforme esté todo bien, tanto el pago como los datos, pasadas 48 horas después de la solicitud, se activará la cuenta y podrá ser funcional para poder subir las aplicaciones en la plataforma.

### **Google Play Store:**

En el caso de Play Store, una vez terminada la aplicación, es necesario crear un certificado si es la primera vez que se hace por parte del desarrollador, el cual contendrá las claves de firma de la aplicación o aplicaciones creadas. Al acabar con este proceso y teniendo ya el certificado, se debe generar un APK con la clave creada en el certificado hecho previamente, y con esto, el usuario se cerciora de tener los derechos de dicha aplicación.

Tras terminar el proceso descrito anteriormente y subir la aplicación a la plataforma Play Store, es necesario hacerse una cuenta en la consola de desarrollador Google Play Store. Dicha cuenta, si es la primera vez que se hace, se debe abonar un pago único de por vida de 25€. Al finalizar el abono de la cantidad descrita previamente, se tiene que rellenar el perfil de desarrollador con los datos personales del propietario de mencionada cuenta.

## Conclusiones

Para la realización de este proyecto apostamos por una idea que consideramos necesaria y con futuro, pero creemos que el desarrollo del proyecto es muy grande como para tener el proyecto terminado por completo y comercialmente viable, después de un periodo de desarrollo bastante agitado, tenemos la conclusión que el proyecto a supuesto toda una serie de desafíos a los cuales no nos habíamos enfrentado, por lo que salir de algunos de ellos ha sido gratificante. Todo y que algunos de ellos no los hemos podido solucionar ya sea por dificultad o por limitación de la versión gratuita utilizada en Firebase.

Más o menos hemos conseguido los objetivos previstos y podemos afirmar que ha sido un proyecto más ambicioso de lo previsto en la preparación, gracias a su complejidad hemos aprendido tecnologías y técnicas que no habíamos utilizado hasta el momento como son **PayPal Developer** y **Firestore Realtime Database**.

Para finalizar, creemos que con un poco más de experiencia en el sector y con la experiencia conseguida durante el desarrollo del proyecto, tenemos unas bases sólidas para emplear en el futuro y algún día publicar Cinet en el mercado.

## Bibliografía

Iconos dentro del proyecto:

<https://www.flaticon.es/resultados>

Cálculo de presupuestos:

<https://es.talent.com/salary?job=analista+de+software>

<https://es.talent.com/salary?job=dise%C3%B1ador>

<https://es.talent.com/salary?job=desarrollador>

<https://es.talent.com/salary?job=tester>

Información gestión Firebase Real Time Database:

<https://firebase.google.com/docs/database/android/read-and-write?hl=es-419>

Uso de PayPal sdk:

<https://www.desarrollolibre.net/blog/android/como-integrar-paypal-en-android>

## Anexos

Repositorio Cinet: <https://github.com/pbermejo18/Cinet>

Link de descarga Cinet: <https://pbermejo18.github.io/cinet.github.io/>

Link PayPal Developer: <https://developer.paypal.com/>