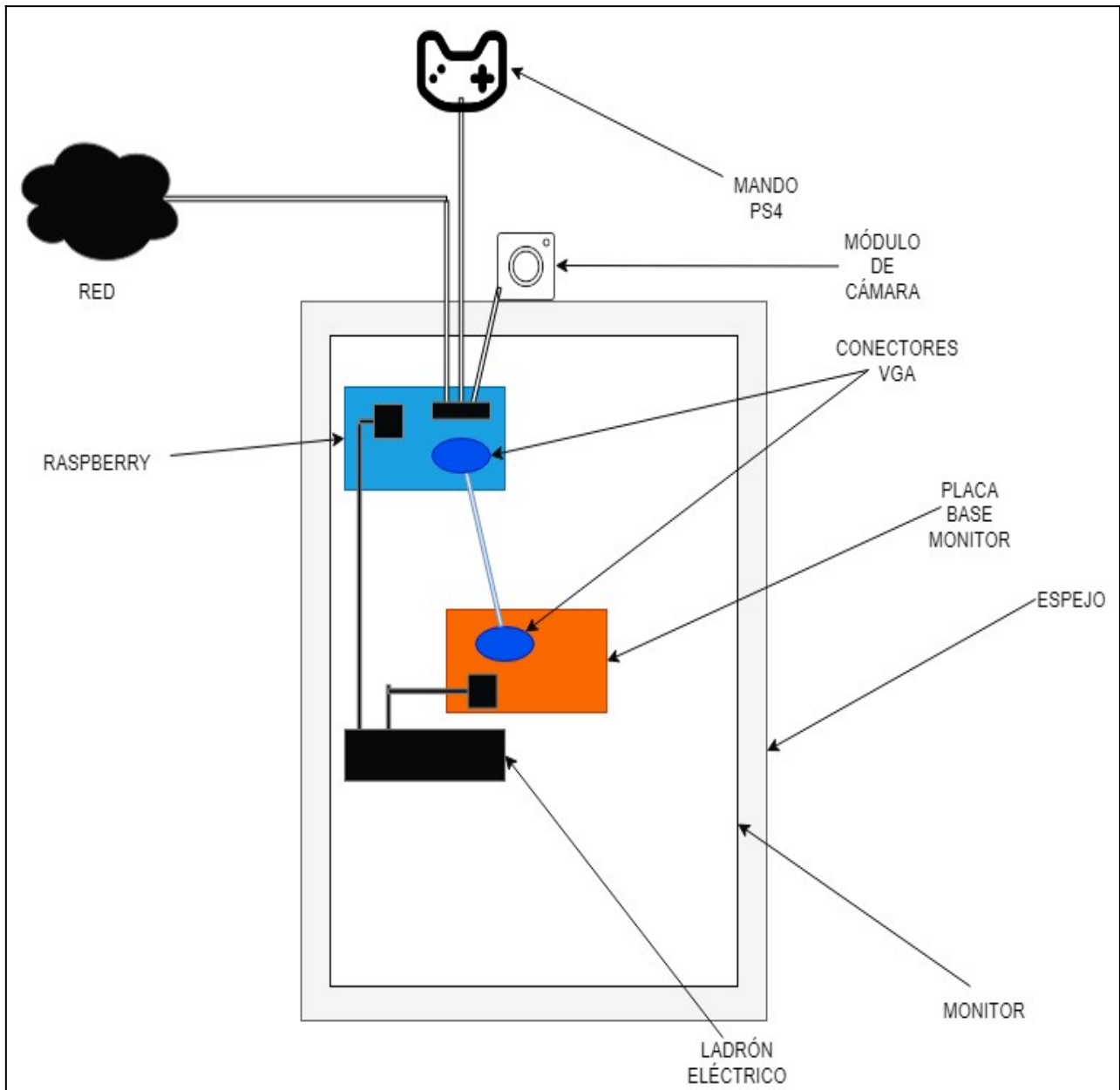


## DESCRIPCIÓN

### ESQUEMA HARDWARE

En este punto explicaremos el esquema hardware que utilizaremos en el proyecto y los elementos elegidos para realizarlo.



Aquí vemos un esquema de la idea que queremos llevar a cabo como podemos ver tenemos toda una serie de componentes, vamos a describir el cometido de cada uno de ellos y el para que lo necesitamos en este proyecto.

→ **Mando PS4:** Debido a complejidad y falta de presupuesto el usuario no podrá interactuar con la interfaz de manera táctil, por lo que el input del usuario se obtendrá de el mando de PS4 con este mando el usuario podrá interactuar de una manera sencilla con la interfaz gráfica, por ejemplo para cambiar de pantallas o escribir algo a través de un teclado virtual, esto último esta aún por ver.  
Estará conexionado a la Raspberry por USB.

→ **Red:** Necesitaremos conexionado a la red para obtener los datos necesarios de las múltiples APIs que utilizaremos para mostrarlos, también necesitaremos conexión a una red para acceder a la base de datos que guardará los datos de nuestra aplicación, aunque se está pensado en montar la base de datos en local, todavía no sabemos como afectará al rendimiento por lo que de momento la idea es poner la base de datos en remoto y que el dispositivo se conecte a través de una red.  
Estará conectado a la Raspberry por Ethernet.

→ **Módulo de cámara:** Encargada de enviar la señal de lo que está viendo a la aplicación, utilizaremos este elemento para reconocer caras y hacer que los usuarios puedan iniciar sesión por este medio.

→ **Conectores VGA:** Realmente la raspberry no tiene este tipo de conector por lo que para conectar las dos señales de video usaremos un adaptador de VGA a HDMI para poder conectarlo en la raspberry.

→ **Placa base del monitor:** Placa base en el que irán conectados los elementos necesarios para que el monitor funcione y recibir la señal de video de la Raspberry PI, este elemento si tiene un conector VGA y se conectará a la Raspberry con un adaptador de VGA a HDMI.

→ **Ladrón eléctrico:** Con el fin de reducir la cantidad de cables que salen de la carcasa que contiene a todos los elementos hemos decidido introducir un ladrón dentro de la carcasa para poder gestionar las conexiones de alimentación de una manera más cómoda, gracias a esto solo saldrán dos cables fuera de la carcasa, el cable ethernet y el cable de alimentación del ladrón.  
En el estarán conectados el cable de alimentación del monitor y la raspberry.

→ **Monitor:** Gestionado por su placa base mostrará la señal de video que necesitemos.

→ **Raspberry:** Encarga de gestionar los dispositivos que conectaremos y de cargar la aplicación asistente, durante el proyecto trabajaremos sobre ella.

## ESQUEMA SOFTWARE

En este punto veremos como será el diseño de la aplicación a nivel de interfaz. **Estos modelos de momento son prototipos y podrán ir cambiando a lo largo del desarrollo de la aplicación.**

### PANTALLA INICIAR SESIÓN

Pantalla simple de inicio de sesión en la que introduciremos nuestro usuario y contraseña y podremos iniciar sesión de manera tradicional, para iniciar sesión con face ID solo tendremos que introducir el nombre de usuario y que la cámara nos detecte el rostro.

A screenshot of a login screen prototype. The background is black. At the top center is a white octagonal logo containing a stylized 'E' made of two parallel diagonal lines. Below the logo, the word 'BIENVENIDO' is written in white, bold, uppercase letters. Underneath, there are two light gray rounded rectangular input fields. The first field is labeled 'Username' and the second is labeled 'Password'. Below these fields are three yellow rectangular buttons with black text. The first button says 'INICIAR SESIÓN', the second says 'INICIAR SESIÓN CON FACE ID', and the third says 'REGISTRARSE'.

## PANTALLA REGISTRO

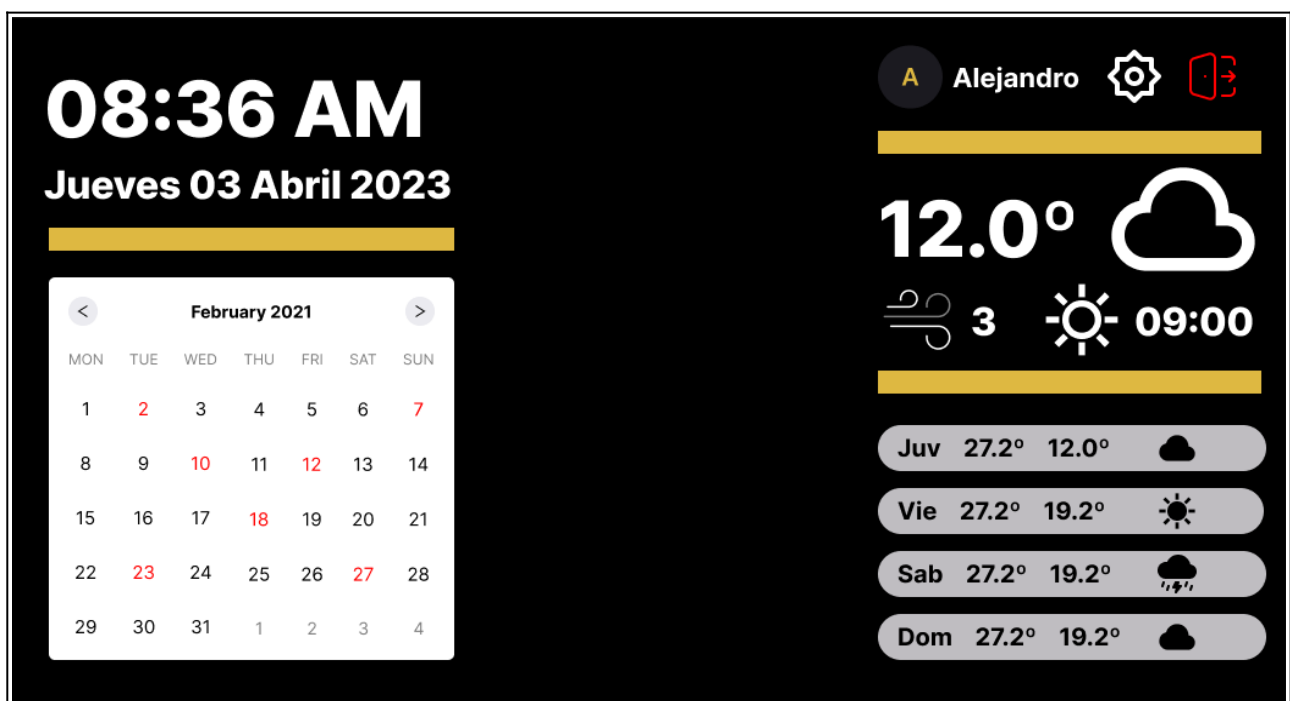
Podremos acceder a esta pantalla a través de la pantalla de inicio de sesión y sirve para registrarnos en el sistema, para ello introducimos los datos que nos solicita el formulario y registramos el rostro.



The registration screen features a dark background with a white octagonal logo at the top center. Below the logo, the word "REGISTRO" is displayed in white. The form includes two input fields for "Username" and "Password". A "FACE ID" section with a green checkmark is followed by a yellow "REGISTRAR ROSTRO" button. Below these are two dropdown menus for "LOCALIZACIÓN" and "EQUIPO FAVORITO". At the bottom is a large yellow "COMPLETAR REGISTRO" button.

## PANTALLA DATOS CLIMÁTICOS Y CALENDARIO

Accederemos a ella cuando el inicio de sesión sea correcto, en ella veremos datos climáticos así como la hora y el día en el que estamos, el usuario que ha iniciado sesión, botones de cerrar sesión, configuración, un calendario en el que podremos marcar días importantes y añadir información.

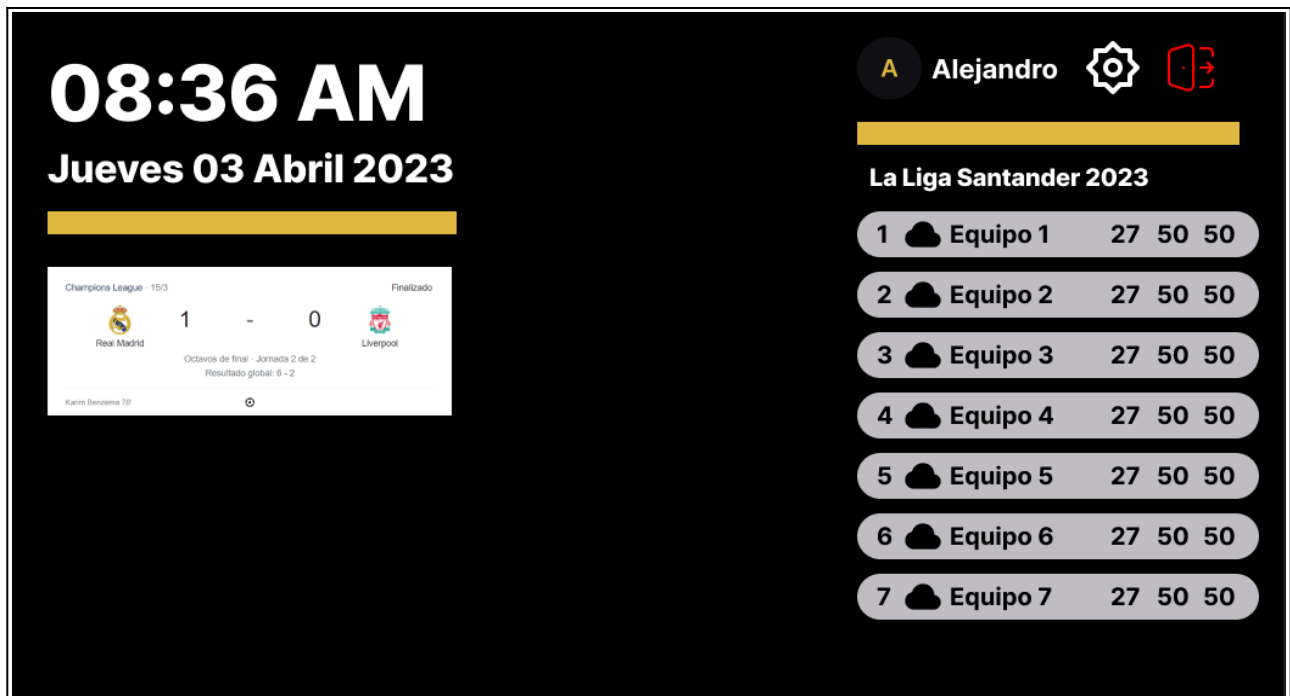


The home screen displays the time "08:36 AM" and the date "Jueves 03 Abril 2023" in large white text. A yellow bar separates this from the main content. On the left is a calendar for February 2021. On the right, the user's name "Alejandro" is shown with a profile icon, a settings gear, and a logout icon. Below this, the current temperature "12.0°" is shown with a cloud icon. Further down, wind speed "3" and sunrise time "09:00" are displayed. At the bottom, a 5-day weather forecast is shown in yellow bars.

Day	Temperature	Weather
Juv	27.2° / 12.0°	Cloudy
Vie	27.2° / 19.2°	Sunny
Sab	27.2° / 19.2°	Cloudy with rain
Dom	27.2° / 19.2°	Cloudy

## PANTALLA DATOS FUTBOLÍSTICOS

Otra de las pantallas navegables del sistema, podremos acceder a ella desde la ventana de datos climáticos (desde esta también podremos volver a la ventana de datos climáticos) en ella veremos datos futbolísticos de nuestro equipo favorito, así como resultados recientes o partidos próximos, también veremos los componentes generales como el usuario que ha iniciado sesión el botón de configuración y el botón de cerrar sesión.



## PANTALLA CONFIGURACIÓN DE LA APLICACIÓN

Pantalla de configuración del usuario, en ella podremos cambiar el usuario, la contraseña, nuestros datos faciales, la localización del sistema y nuestro equipo favorito.



## JUSTIFICACIÓN, ALCANCE Y ALTERNATIVAS

Este proyecto se ha realizado con el fin de reducir el tiempo que necesitamos para consultar datos básicos juntos antes de salir los días que tenemos prisa o vamos ajustados de tiempo, mientras nos arreglamos delante del espejo podremos consultar datos básicos como la hora, tiempo del día, a que hora jugará nuestro equipo favorito...

El alcance del proyecto es bastante corto en lo que se refiere a comercializarlo debido a todos los materiales que necesitaremos para hacerlo, la aplicación eso sí podría usarse por separado con una adaptación móvil o algo parecido, aún así este proyecto está siendo comercializando por otras entidades poco a poco, eso sí, con precios bastante altos.

Las alternativas que existen del software que vamos a realizar es: Magic Mirror este proyecto fue desarrollado para Raspberry en 2021 este tiene varios módulos diferentes pero nosotros le vamos a añadir un par de funcionalidades más para diferenciarlo.

## STACK TECNOLÓGICO

La aplicación utilizará el patrón de diseño de Modelo Vista Controlador, el lenguaje principal será Java y se integrarán varias tecnologías y frameworks para el desarrollo del backend y el frontend.

→ Java JDK 18: Kit de desarrollo de aplicaciones Java

→ Raspberry PI OS with desktop 32 bits 5.15: Sistema operativo de la Raspberry PI.

→ Spring framework 3.0.5: De Spring utilizaremos varias herramientas de las que disponen aparte de sus funcionalidades básicas como la IoC o ID.

→ Spring Boot: Para facilitar las configuraciones, el patrón de diseño y las dependencias

→ Spring Data JPA: Para facilitar tareas de acceso a datos.

→ Base de datos: H2 u Oracle (aún no decidido) la idea de incluir H2 es que es una base de datos muy muy ligera y en este proyecto no pretendemos guardar muchos datos.

→ APIs

→ OpenWeather: <https://openweathermap.org/> La utilizaremos para extraer datos meteorológicos varios.

→ Api-football: <https://www.api-football.com/> La utilizaremos para extraer datos futbolísticos varios.

→ Maven: Para gestionar las librerías de una manera más cómoda.

→ Librería JINPUT: Necesario para extraer el input del usuario y poder controlar la aplicación.

→ Librería Open CV: Necesario para poder reconocer rostros con el módulo de cámara.

→ Librería JavaFX y SDK Versión 20: Necesario para realizar toda la interfaz de la aplicación.

# ANÁLISIS DE LA APLICACIÓN

## OBJETIVOS GENERALES

- El usuario tendrá acceso a información climática, futbolísticos y datos del día.
- El usuario podrá gestionar su cuenta, cambiando los datos que se muestran en todo momento.

## REQUISITOS FUNCIONALES

- El usuario podrá registrarse en la aplicación.
- El usuario podrá iniciar sesión si tiene una cuenta registrada, por face id o por contraseña y usuario.
- El usuario podrá cerrar sesión.
- El usuario podrá cambiar de contraseña.
- La base de datos cargará todos los datos del usuario al iniciar sesión.
- El usuario podrá consultar datos climáticos y futbolísticos
- El usuario podrá configurar la localización y equipo para que aparezcan otros datos.
- El usuario dispondrá de un calendario básico en el que podrá añadir notas.

## REQUISITOS NO FUNCIONALES

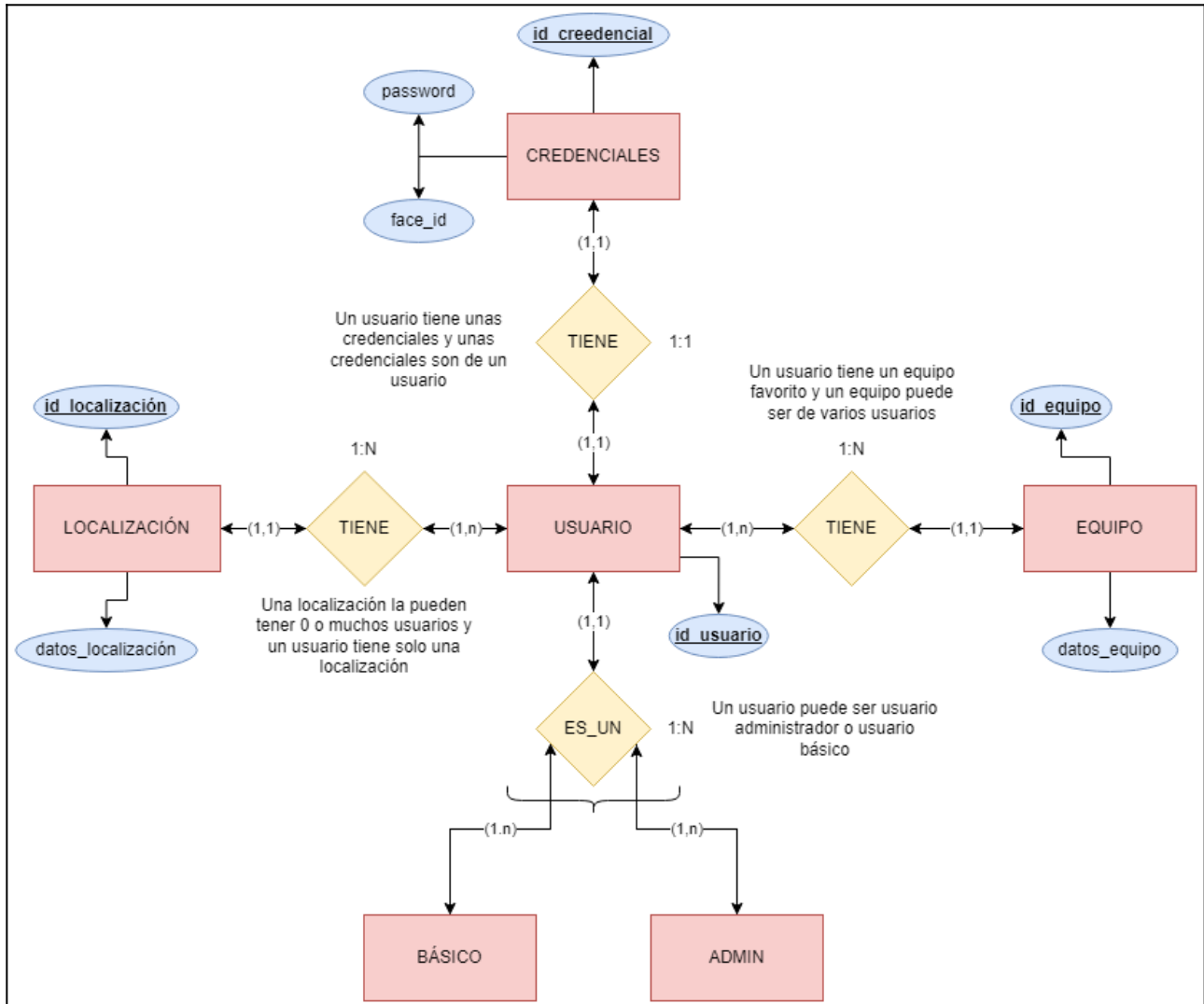
- El sistema podrá guardar más de 10 usuarios en la base de datos y que todos sean perfectamente funcionales.
- La aplicación debe intentar consumir lo mínimo posible en la Raspberry PI.
- Todos o casi todos los datos de la cuenta del usuario se guardarán en una base de datos.
- La aplicación debe controlar todas las excepciones posibles que puedan suceder en su uso.
- La aplicación obtendrá datos de 2 APIs diferentes y será capaz de mostrarlo al usuario de manera sencilla.
- La aplicación será capaz de reconocer rostros y guardarlos en la base de datos.
- La interfaz del usuario será controlada por este con un mando de PS4.

## REQUISITOS DE INTERFAZ

- La pantalla 1 tendrá un login tradicional con tres botones uno para iniciar sesión otro para iniciar sesión con face ID y otro para registrarnos.
- En la primera manera de iniciar sesión, el usuario tendrá que escribir en un campo de texto, su usuario y contraseña y pulsar el botón de iniciar sesión.
- En la segunda manera de iniciar sesión, el usuario deberá de mostrar su cara a la cámara y que esta le reconozca, no tendrá que pulsar ningún botón.
- Para registrarse, el usuario deberá pulsar el botón de registro en la pantalla de inicio de sesión, rellenar el formulario de datos que aparezca en la siguiente pantalla y pulsar el botón de completar el registro.
- Una vez iniciado sesión el usuario podrá navegar entre las 2 pantallas disponibles pulsando los botones del mando R1 (para ir a la siguiente pantalla) y L1 (para ir a la anterior).
- La interfaz general que veremos siempre en la parte superior de cada pantalla será: Avatar del usuario, nombre, un botón de cerrar sesión y un botón para acceder a la ventana de configuración.
- La pantalla 2 contendrá datos del tiempo del día y un calendario básico en esta pantalla el usuario podrá navegar entre la interfaz general y el calendario usando las flechas ← → del mando.
- La pantalla 3 contendrá datos generales de nuestro equipo favorito, partidos próximos, resultados anteriores, clasificación...
- La pantalla 4 a la cual se accederá por el botón de configuración contendrá opciones de configuración básica como la localización de los datos del clima, nuestro nombre de usuario, cambio de contraseña, cambio de equipo favorito...

## MODELO Y DISEÑO DE LA BASE DE DATOS

Este es el primer diseño, este podrá ir cambiando a lo largo del desarrollo dependiendo de nuestras necesidades, pero esta será la idea general debido a que no tendremos que guardar muchos datos en la bd.



## REPOSITORIO DE GITHUB

[https://github.com/alejandrosp166/TFG\\_ESPEJO\\_ASISTENTE](https://github.com/alejandrosp166/TFG_ESPEJO_ASISTENTE)