



MÓDULO PROYECTO

CFGS Desarrollo de Aplicaciones
Multiplataforma
Informática y Comunicaciones

Sportify

Tutor individual: Cristina Silván Pardo

Tutor colectivo: Cristina Silván Pardo

Año: 2025

Fecha de presentación: 22/05/2025

Nombre y Apellidos: Alejandro de Gregorio Miguel

Email: Alejandro.gremig@educa.jcyl.es



Tabla de contenido

1	Identificación proyecto	4
2	Organización de la memoria	4
3	Descripción general del proyecto.....	4
3.1	Objetivos	4
3.2	Cuestiones metodológicas.....	5
3.3	Entorno de trabajo (tecnologías de desarrollo y herramientas)	5
4	Descripción general del producto	6
4.1	Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.....	6
4.2	Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.....	6
4.3	Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha.....	6
5	Planificación y presupuesto.....	7
6	Documentación Técnica: análisis, diseño, implementación y pruebas.	8
6.1	Especificación de requisitos	8
6.2	Análisis del sistema	8
6.3	Diseño del sistema	10
6.3.1	Diseño de la Base de Datos	10
6.3.2	Diseño de la Interfaz de usuario.....	11
6.3.3	Diseño de la Aplicación.....	16
6.4	Implementación	17
6.4.1	Entorno de desarrollo.....	17
6.4.2	Estructura del código	18
6.4.3	Cuestiones de diseño e implementación reseñables.	20
6.5	Pruebas.....	29

7	Manuales de usuario	36
7.1	Manual de usuario	36
7.2	Manual de instalación	41
8	Conclusiones y posibles ampliaciones	43
9	Bibliografía	43
10	Anexos.....	43

1 Identificación proyecto

Este proyecto tiene como objetivo principal facilitar a los usuarios la reserva de pistas deportivas, integrando el proceso de pago directamente a través de la aplicación. La aplicación es multiplataforma, lo que significa que puede ser utilizada en dispositivos móviles, tabletas y computadoras. Permite a los usuarios registrarse, iniciar sesión y gestionar sus reservas de manera eficiente. La estructura de la aplicación incluye un sistema de cuenta donde los usuarios pueden acceder a una pantalla de reservas, visualizar el historial de sus reservas, y ajustar la configuración de su cuenta.

Para el desarrollo de este proyecto, se utilizarán tecnologías de Microsoft, específicamente PowerApps para el diseño de la interfaz de usuario, PowerAutomate y Make.com para la automatización de flujos y procesos, GitHub Pages para lanzar un par de páginas HTML para la redirección de los pagos de la API de PayPal, API de PayPal para realizar pagos ficticios (sin dinero real de por medio) dentro de la aplicación y SharePoint para el almacenamiento y gestión de datos.

El tiempo estimado para terminar la aplicación es de aproximadamente 100 a 150 horas.

2 Organización de la memoria

Esta memoria consta de 4 apartados clave:

- Descripción del Proyecto: Como su propio nombre indica es una descripción de las herramientas utilizadas, objetivos y cuestiones metodológicas del proyecto.
- Planificación del Proyecto: Una estimación del tiempo que llevaría a hacer el proyecto además de toda la planificación anterior a la realización.
- Documentación Técnica: análisis, diseño, implementación y pruebas.
- Manuales de usuario: Explicación de como instalar y utilizar la aplicación

3 Descripción general del proyecto

Este proyecto consiste en el desarrollo de una aplicación multiplataforma capaz de funcionar en dispositivos móviles, tabletas y ordenadores. La aplicación está construida utilizando herramientas de Microsoft, concretamente **PowerApps**; para el desarrollo principal de la aplicación, **PowerAutomate y Make.com**; para la creación de flujos que faciliten algunos procesos de la aplicación, **GitHub Pages**; para lanzar dos páginas HTML para el direccionamiento de la API de PayPal, **API de PayPal**; como método de pago ficticio en la aplicación, y **SharePoint**; como base de datos.

3.1 *Objetivos*

Los objetivos del proyecto son los siguientes:

- **Accesibilidad:** Desarrollar una aplicación que se pueda usar cómodamente en distintos dispositivos, asegurando que todos los usuarios tengan una experiencia similar y sin complicaciones.
- **Eficiencia en Procesos:** Implementar funciones automatizadas para ahorrar tiempo y reducir las tareas repetitivas.
- **Gestión de Información:** Utilizar SharePoint para centralizar el almacenamiento de datos, permitiendo que los usuarios accedan a la información desde cualquier lugar y en cualquier momento.

3.2 *Cuestiones metodológicas*

3.2.1 APIs

Durante el desarrollo de la aplicación en PowerApps se ha seguido un enfoque basado en la integración de servicios externos mediante el uso de APIs. En lugar de realizar llamadas directas a la API de PayPal desde PowerApps, se ha optado por una arquitectura intermedia utilizando la plataforma **Make.com** lo cual permite gestionar de forma más controlada y segura las llamadas HTTP hacia servicios externos.

¿Qué es una API?

Una **API (Application Programming Interface)** es un conjunto de definiciones y protocolos que permiten que dos aplicaciones se comuniquen entre sí. En este caso, la API de PayPal ofrece una serie de endpoints REST que permiten ejecutar operaciones como la creación de enlaces de pago, la gestión de cobros y la consulta de transacciones.

Tipología de la API

La API de PayPal utilizada es una **API RESTful**, que funciona sobre el protocolo HTTP. Se utiliza principalmente el método POST para generar enlaces de pago, y la autenticación se realiza mediante el estándar **OAuth 2.0**, que proporciona un acceso seguro y controlado.

3.3 *Entorno de trabajo (tecnologías de desarrollo y herramientas)*

En mi proyecto he utilizado varias tecnologías y herramientas para lograr una buena funcionalidad de la aplicación:

- **PowerApps:** Para diseñar la interfaz de usuario y crear pantallas interactivas que

hagan la aplicación intuitiva y fácil de usar.

- **PowerAutomate y Make.com:** Para establecer flujos de trabajo que automatizan tareas, haciendo que la aplicación sea más productiva y eficiente.
- **SharePoint:** Como plataforma para gestionar y almacenar toda la información, permitiendo un acceso organizado y seguro a los datos.
- **GitHub Pages:** Para alojar dos páginas web que servirá como redireccionamiento de la API de PayPal cuando se hay completado o no el pago.
- **API PayPal:** Para realizar pagos ficticios dentro de la aplicación.

4 Descripción general del producto

La aplicación está diseñada para facilitar el acceso y gestión de reservas mediante un sistema de cuentas y pagos integrados a través de la propia aplicación.

4.1 *Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.*

- **Límites del Sistema:** La aplicación se centra en la funcionalidad de reserva de pistas deportivas, procesamiento de pagos, y gestión de cuentas de usuario.
- **Funcionalidades Básicas:** Los usuarios pueden crear y acceder a sus cuentas, realizar reservas de pistas, efectuar pagos, visualizar sus reservas actuales, y gestionar la información de su cuenta.
- **Usuarios y/o Sistemas con los que Interactúa:** Destinada a usuarios individuales que desean reservar pistas deportivas. El sistema también interactúa con plataformas de pago para procesar transacciones seguras.

4.2 *Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.*

- **Técnicas de Gestión de Cuentas:** Integración de un sistema de autenticación seguro que permite inicio de sesión y registro de nuevos usuarios, proporcionando un acceso personalizado.
- **Arquitectura de Pagos Integrados:** La aplicación implementa una arquitectura que facilita pagos seguros y eficientes dentro de la plataforma, utilizando integraciones con proveedores de servicios de pago.

4.3 *Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha*

- **Plataforma Tecnológica:** La aplicación se desarrolla y despliega utilizando

herramientas de Microsoft.

- **Instalación de la Aplicación:** Esta aplicación se distribuye desde el sitio web de la organización, donde puede descargarse y configurarse para su uso en dispositivos móviles y computadoras.
- **Puesta en Marcha:** Tras la instalación, se verifica la funcionalidad completa mediante pruebas de usuario antes de hacerla accesible al público. Se proporciona soporte técnico y guía para que los usuarios puedan adaptarse fácilmente al sistema de reservas y pagos.

5 Planificación y presupuesto

Mi planificación para este proyecto es la siguiente:

1. En primer lugar, se empezará con una idea general de cuantas pantallas habrá que utilizar dependiendo de la función que le quiera dar y como navegar entre ellas. En mi caso serán: 2 pantallas principales, Inicio de Sesión y Registro que se navegará con botones y después 4 pantallas más, una lista con las reservas hechas, otra para hacer reservas con un botón que navegará a otra pantalla con la disponibilidad y otra que será la pantalla de Ajustes con información de la aplicación, recarga de saldo, cerrar sesión... Todo esto último siendo posible navegar entre ellas a través de un menú de navegación inferior.
2. Diseñar y pensar la base de datos, en un principio opté por utilizar MySQL, pero lo cambié por SharePoint porque es lo que más he utilizado en la empresa junto con PowerApps. La base de datos en SharePoint constará de dos listas: una de usuarios y otra de reservas.
3. Una vez pensado todo esto se empezará a diseñar la aplicación visualmente (sin funcionalidad) dentro de las posibilidades de PowerApps para una mejor organización.
4. Ya hecha la parte visual se comenzará a programar los distintos componentes para darles una funcionalidad. También se realizarán algunos flujos con PowerAutomate para automatizar algunos procesos.
5. Durante todo esto se realizarán cambios en la idea original ya que PowerApps es complicado de tener una idea visual clara al principio y poder recrearla una vez la dotas de funcionalidad.
6. Una vez hecho todo esto empezaré con la parte más “difícil” de la aplicación que es la integración de la API de PayPal para gestionar el saldo.

Ya dicho todo esto, este proyecto necesitará entre 100 y 150 horas para ser realizado, aunque seguramente se amplie por la poca fluidez que tengo con PowerApps ya que es una tecnología que todavía estoy aprendiendo a utilizar.

En el desarrollo de esta aplicación no será necesario presupuesto ya que los servicios que he utilizado son gratuitos gracias a la cuenta de la organización de la junta de Castilla y León, ya que PowerApps es de Microsoft y se necesita licencia para ello.

6 Documentación Técnica: análisis, diseño, implementación y pruebas.

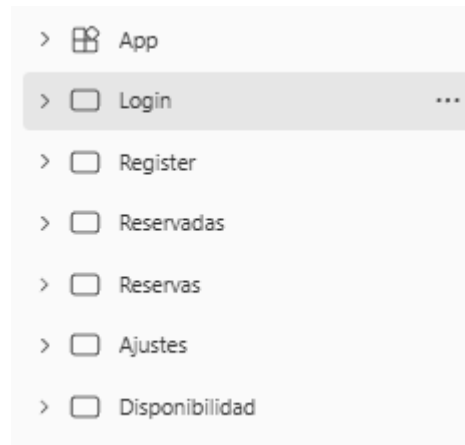
6.1 Especificación de requisitos

- Requisitos Funcionales
 - Inicio de sesión y registro de usuario
 - Visualización de las reservas realizadas por el usuario, con posibilidad de editar o cancelar cada una
- Requisitos No Funcionales
 - Interfaz optimizada para dispositivos móviles, tablets y ordenadores.
 - Tiempo de respuesta adecuado (carga de datos y ejecución de flujos en menos de 3 segundos).
 - Accesibilidad desde cualquier navegador compatible con Microsoft 365.
 - Seguridad en el tratamiento de datos del usuario y en las transacciones.
 - Actualización en tiempo real de disponibilidad de pistas.

6.2 Análisis del sistema

- El funcionamiento de esta aplicación consta de dos partes: El propio uso de un usuario cualquiera y todo el funcionamiento interno de la aplicación (llamadas a API's, flujos, acciones de componentes...):
 - Interfaz de usuario:
 -
 - La aplicación presenta una interfaz intuitiva desarrollada con **PowerApps**, accesible desde navegador o dispositivo móvil.
 - El usuario puede **iniciar sesión** o **registrarse** con sus datos personales y acceder a una vista personalizada.
 - Desde la **pantalla principal**, el usuario puede:
 - Ver sus **reservas activas y pasadas**, ordenadas por fecha.
 - **Editar** una reserva futura si desea cambiar la pista, fecha u hora.
 - **Cancelar** una reserva con un simple botón.
 - En la **pantalla de nueva reserva**, el usuario puede:
 - Seleccionar el **tipo de pista** (tenis, pádel, fútbol, etc.).
 - Escoger una **fecha y franja horaria** disponible.
 - Confirmar la reserva, que se guardará automáticamente en el sistema.



- En la **pantalla de ajustes**, el usuario puede:
 - Consultar y modificar información de su cuenta.
 - Acceder a un **monedero virtual**, donde puede ver su saldo actual.
 - Ingresar dinero desde **PayPal** mediante un botón que genera un enlace de pago.
- Todos los elementos visuales son dinámicos y se adaptan al contenido: por ejemplo, las pistas ya reservadas en un horario no están disponibles al seleccionar.



6.3 Diseño del sistema:

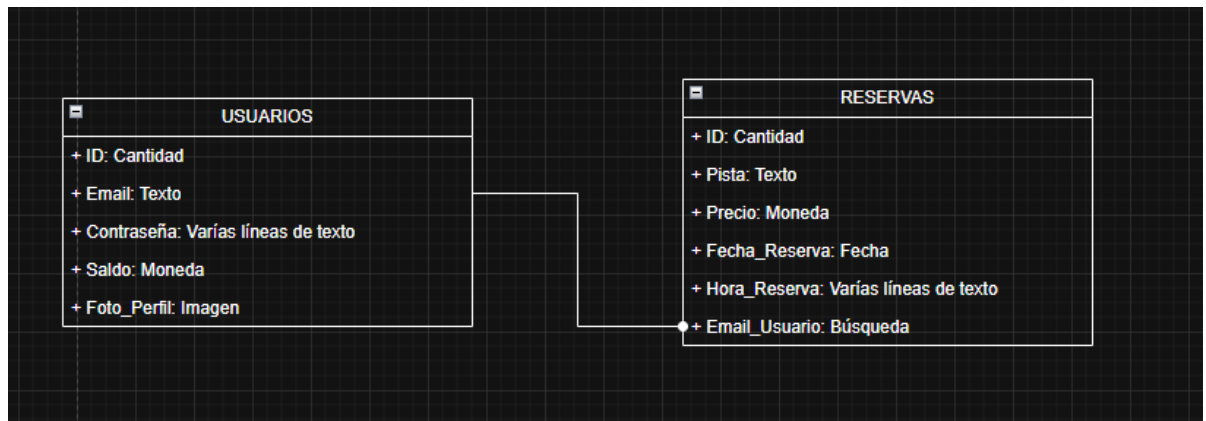
6.3.1 Diseño de la Base de Datos

La base de datos se ha creado en un sitio de SharePoint llamado GestionPistas y tiene dos listas, USUARIOS y RESERVAS relacionadas entre sí ya que un usuario puede hacer varias reservas, pero esas mismas reservas están a nombre del propio usuario.

ID	Email	Contraseña	Saldo	Foto_Perfil
13	alejandro.gremig@edu	MTIzNA==	10,00 €	
16	a@gmail.com	MTIz	25,00 €	

Pista	Precio	Fecha_Reserva	Hora_Reserva	Email_Usuario
Fútbol Sala	15,00 €	19/05/2025	12:00	a@gmail.com
Fútbol Sala	15,00 €	19/05/2025	14:00	alejandro.gremig@...
Baloncesto	15,00 €	19/05/2025	14:00	alejandro.gremig@...

6.3.1.1 Diagrama de clases

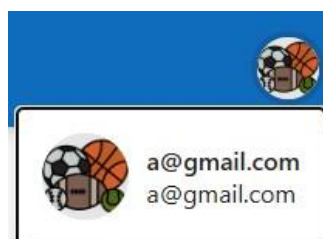


SharePoint tiene muchos tipos de campos. El campo de búsqueda funcionaría como una ForeignKey en la que se puede seleccionar el campo al que se quiere hacer referencia.

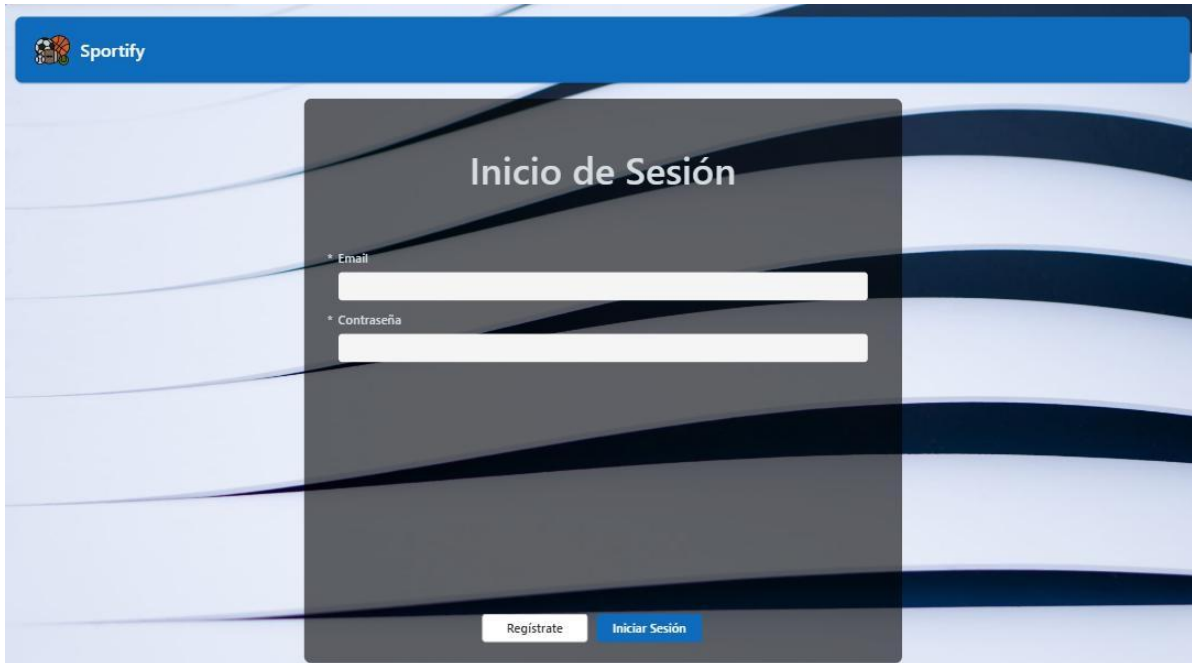
Además, SharePoint cada vez que crea una lista se crean columnas que vienen por defecto como el ID, o el título. En el esquema he puesto las que yo tengo visibles y las que creo que son las más importantes.

6.3.2 Diseño de la Interfaz de usuario.

En todas las páginas de mi aplicación hay un cabecero con el nombre de la aplicación y la foto de perfil y el email del usuario que la esté utilizando.



6.3.2.1 Inicio de Sesión

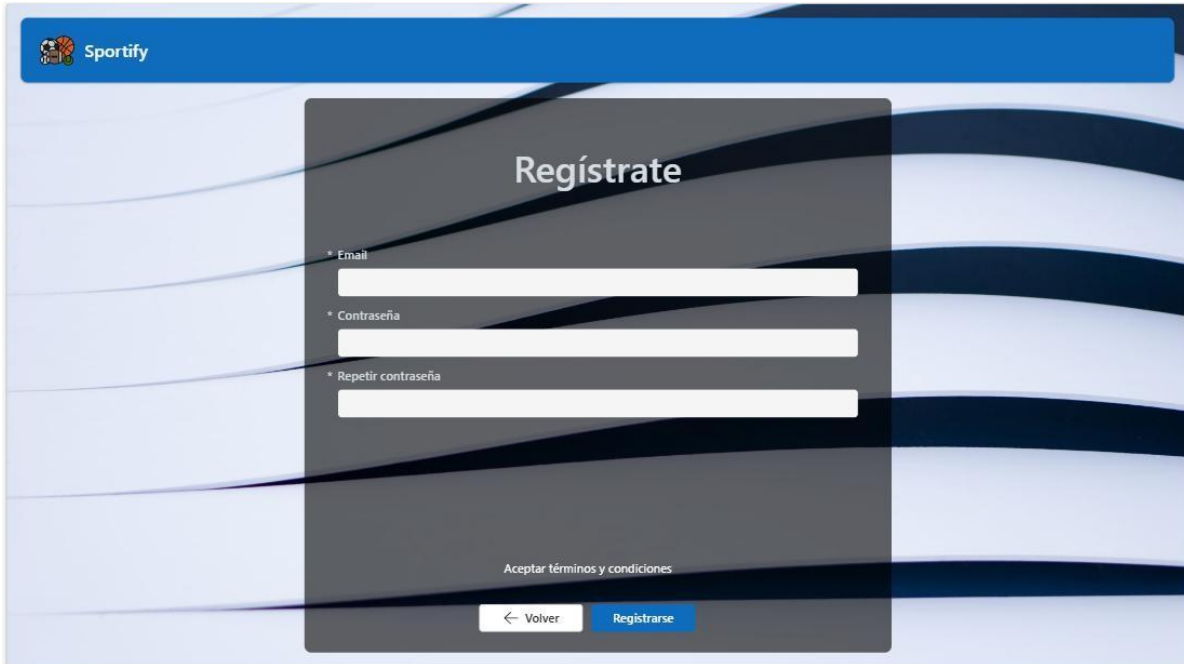


The screenshot shows the 'Inicio de Sesión' (Login) page of the Sportify application. At the top, there is a blue header bar with the Sportify logo and name. The main content area has a dark gray background with a white login form. The form is titled 'Inicio de Sesión' and contains two input fields: '* Email' and '* Contraseña'. Below the input fields, there are two buttons: 'Regístrate' (white with black text) and 'Iniciar Sesión' (blue with white text). The background of the page features a light blue and white wavy pattern.

Esta es la pantalla de Inicio de Sesión, esta pantalla consta de un formulario con dos TextFields, uno para el Email y otro para la contraseña, y dos botones, uno de inicio de sesión y otro para registrarse.

- Si los campos están vacíos al darle al botón de inicio de sesión saldrá un error.
- Si no están vacíos, se buscará en la base de datos las coincidencias:
 - Si el usuario no existe saldrá error.
 - Si el usuario existe, pero la contraseña es errónea, saldrá error
 - Si el usuario existe y la contraseña coincide con la de la base de datos, se pasará a la pantalla de Ajustes.
- Si se pulsa el botón de Registro te redirigirá a la página de Registro.

6.3.2.2 Registro

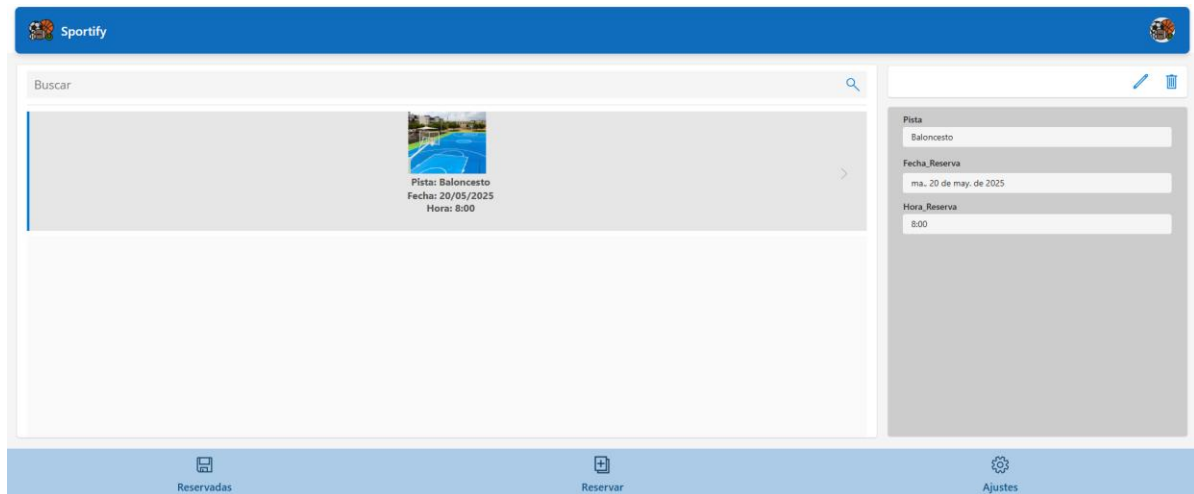


The screenshot shows the 'Sportify' registration interface. At the top, there is a blue header with the 'Sportify' logo. Below this, a dark gray registration form is centered. The form has the title 'Regístrate' at the top. It contains three input fields: 'Email', 'Contraseña', and 'Repetir contraseña'. Below these fields is a checkbox labeled 'Aceptar términos y condiciones'. At the bottom of the form, there are two buttons: a white button with a left arrow and the text 'Volver', and a blue button with the text 'Registrarse'.

Esta es la pantalla de Registro, consta de tres TextFields; uno para el email y otros dos para las contraseñas, dos botones; uno para Volver al Inicio de Sesión y otro para realizar el registro, y un Checkbox de Aceptar Términos y Condiciones.

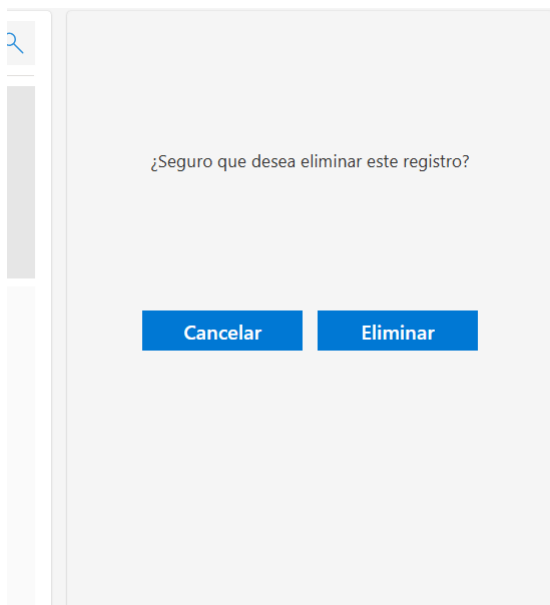
- Si los campos están vacíos saldrá error.
- Si el checkbox no está seleccionado saldrá error.
- Si los campos no están vacíos, se buscará en la base de datos las coincidencias:
 - Si el usuario ya existe saldrá error.
 - Si el usuario no tiene al menos esta estructura: a@dominio.algo, saldrá error.
 - Si el usuario no existe y las contraseñas no coinciden saldrá error.
 - Si todo coincide te redirigirá a la pantalla de Ajustes.

6.3.2.3 Reservadas

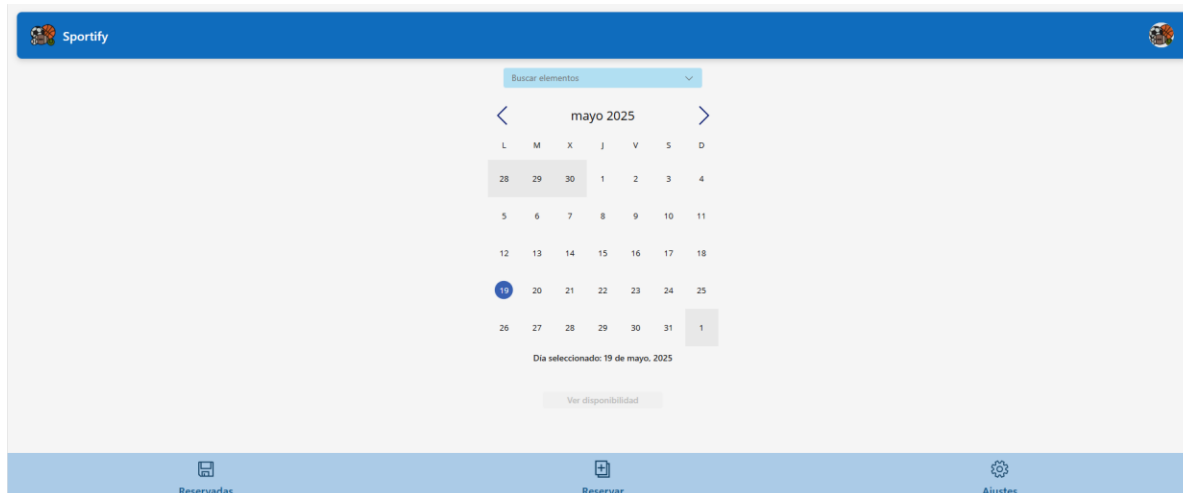


Esta es la pantalla de reservadas, consta de una galería; cuyos Items son las Reservas, un formulario de edición, una Navegación Inferior que navegará entre las diferentes pantallas, y un filtro de búsqueda que solo saldrán las reservas si el tipo de pista contienen esas letras.

Si se le da al icono de la papelera te saldrá una confirmación para eliminar.



6.3.2.4 Reservas



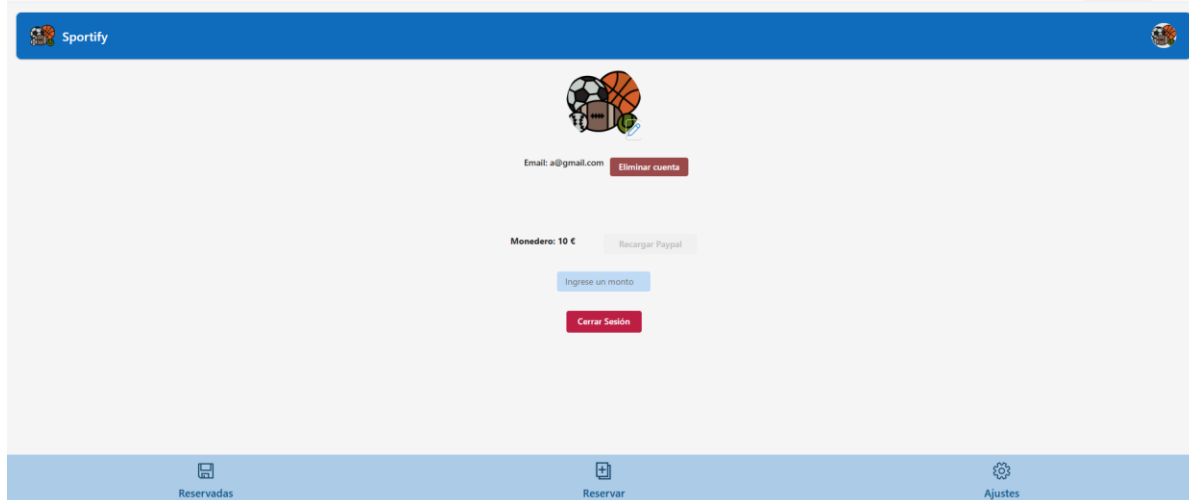
Esta es la pantalla de reservas, consta de un Combobox; para elegir el tipo de pista, un calendario; para escoger la fecha de la reserva (no se pueden escoger días anteriores a hoy), y un botón; para ver la disponibilidad de la pista el día de la fecha seleccionada que te redireccionará a la pantalla de Disponibilidad.

6.3.2.4.1 Disponibilidad



Esta es la pantalla de Disponibilidad, consta de un botón; para volver a la página de Reservas, una etiqueta; que te muestra la fecha que hayas seleccionado en el calendario, y una galería; con las horas que están disponibles para ese día, si esa hora ya está seleccionada para otra reserva saldrá el botón de esa hora como deshabilitado.

6.3.2.5 Ajustes



Esta es la pantalla de Ajustes, aquí saldrá toda la información relacionada con la cuenta del usuario. Consta de una imagen de perfil; que se puede modificar, una etiqueta; con el email del usuario, un botón; para eliminar a cuenta, una etiqueta; que sirve como monedero de la aplicación, un botón; desde aquí se realizan la llamada a la API para realizar el recargo del monedero, un TextField; para insertar el monto a recargar, y un botón; para cerrar sesión.

6.3.3 Diseño de la Aplicación.

Mi aplicación tiene un total de 6 pantallas con funcionalidades distintas:

- La primera pantalla que sale al iniciar la aplicación es la de Inicio de Sesión la cual contiene dos botones: Uno de Registro que navegará a la pantalla de Registro y otro de Iniciar Sesión, que redirige al usuario a la pantalla de Ajustes si el usuario y las contraseñas coinciden con las de la base de datos.
- La segunda pantalla es la de Registro que contiene dos botones: uno para volver a la pantalla de Inicio de Sesión y otro de Registro que navegará a la pantalla de Ajustes si se cumplen todas las condiciones.
- En las siguientes pantallas se navegará a través de un menú de navegación inferior:
 - La tercera pantalla es la de Ajustes que contiene la opción de poner y modificar la foto de perfil de tu cuenta además de un botón que llama a la API de PayPal para recargar el monedero de tu cuenta. También tiene dos botones: uno para eliminar la cuenta y otro para cerrar sesión que te redirigirá a la primera pantalla.
 - La cuarta pantalla es la de Reservadas, aquí saldrán las reservas que hayas hecho a tu nombre. Consta de una galería para visualizar tus reservas y un formulario de edición para modificar o eliminar tus reservas.

- La quinta pantalla es la de Reservas, para realizar reservas. Consta de un combobox para la selección del tipo de pista, un calendario para seleccionar un día y un botón para ver la disponibilidad que te llevará a la siguiente página
 - La sexta pantalla es la de Disponibilidad, saldrá una galería con las horas disponibles para reservar en el día y en el tipo de pista seleccionados anteriormente

6.4 Implementación:

6.4.1 Entorno de desarrollo.

Para desarrollar la aplicación utilicé **PowerApps**, una herramienta de desarrollo low-code que forma parte de la plataforma Microsoft Power Platform. Elegí PowerApps porque lo he utilizado en las prácticas y me pareció interesante realizar una aplicación con una herramienta que no hemos dado en clase.

El entorno de desarrollo es completamente online, por lo que pude trabajar desde cualquier navegador sin tener que instalar nada adicional. PowerApps ofrece una interfaz muy intuitiva, basada en componentes visuales como botones, galerías, etiquetas y formularios. Además, utiliza fórmulas parecidas a las de Excel, lo que facilitó bastante el aprendizaje y la implementación de funciones dentro de la aplicación.

Desarrollé la aplicación en formato “canvas app”, lo que me permitió tener total libertad para diseñar las pantallas a mi gusto. Organicé la aplicación en varias secciones principales: una para ver las reservas, otra para realizar nuevas reservas y otra para ajustes, a las que se accede mediante un menú de navegación inferior.

Para el almacenamiento de datos, utilicé colecciones (Collect, ClearCollect) que me permitieron trabajar con la información de forma temporal dentro de la propia aplicación. También conecté la aplicación con fuentes de datos externas como listas de SharePoint, que usé como base de datos para guardar las reservas, los usuarios y demás información relevante.

6.4.2 Estructura del código.

Como he dicho anteriormente PowerApps es una herramienta low-code con lo cual no tiene gran cantidad de código. Voy a hablar aquí de algunas partes del código que me han parecido un desafío y un poco de las propiedades de los componentes visuales

6.4.2.1 Código botón Inicio de Sesión

```
If(
  IsBlank(CrdEmail.Value) || IsBlank(CrdContra.Value);

  Set(ErrorEmail; IsBlank(Trim(CrdEmail.Value)));;
  Set(ErrorContra; IsBlank(CrdContra.Value));;
  Set(MsgErrorEmail; "Email no puede estar vacío");;
  Set(MsgErrorContra; "Contraseña no puede estar vacía");

  // Si ambos campos están rellenos, seguir
  With(
    { usuario: LookUp(_colUsuarios; Email = Trim(CrdEmail.Value)) };

    If(
      IsBlank(usuario);
      Set(ErrorEmail; true);;
      Set(MsgErrorEmail; "El usuario no existe");

      If(
        usuario.Contraseña <> CodificarPassword.Run(CrdContra.Value).salida;
        Set(ErrorContra; true);;
        Set(MsgErrorContra; "Contraseña incorrecta");

        // Todo correcto
        Set(ErrorEmail; false);;
        Set(ErrorContra; false);;
        Set(MsgErrorEmail; "");;
        Set(MsgErrorContra; "");;
        Notify("Sesión iniciada con éxito"; NotificationType.Success);;
        Navigate(Ajustes; ScreenTransition.Fade; {locUsuario: LookUp(_colUsuarios; Email = Trim(CrdEmail.Value))})
      )
    )
  )
)
```

Esta parte de mi aplicación es la que comprueba en la base de datos si el usuario existe, si la contraseña no coincide...

Primero comprueba si los campos del formulario están en blanco *IsBlank(CrdEmail.Value) || IsBlank(CrdContra.Value)*; y pone los mensajes correspondientes de error que saldrán debajo de los TextFields de Email y contraseña. La función *With* evalúa un solo registro es decir *usuario: LookUp(_colUsuarios; Email = Trim(CrdEmail.Value))*. La función *LookUp* sirve para hacer una consulta en la base de datos y devuelve el valor o valores que coincidan con lo que le estás pasando. Esta parte *usuario.Contraseña <> CodificarPassword.Run(CrdContra.Value).salida*; lo que hace es comprobar si la contraseña del usuario de la base de datos es distinta de la llamada del flujo que codifica la contraseña.

```
// Codificar contraseña
Set(contraseñaCod; CodificarPassword.Run(DataCardValue7_1.Value));

// Validaciones iniciales
If(
    !ChxFerminos.Checked;
    Set(ErrorFerminos; true);;
    Set(MsgErrorFerminos; "Debes aceptar los términos y condiciones");;
    Notify("Debes aceptar los términos y condiciones"; NotificationType.Error)
); // Continuar con las demás validaciones
If(
    IsBlank(Trim(DataCardValue7_3.Value)) || IsBlank(DataCardValue7_1.Value) || IsBlank(DataCardValue1.Value);

    // Errores individuales
    Set(ErrorEmail; IsBlank(Trim(DataCardValue7_3.Value)));;
    Set(MsgErrorEmail; If(IsBlank(Trim(DataCardValue7_3.Value)); "Email no puede estar vacío"; ""));;

    Set(ErrorContra; IsBlank(DataCardValue7_1.Value));;
    Set(MsgErrorContra; If(IsBlank(DataCardValue7_1.Value); "Contraseña no puede estar vacía"; ""));;

    // Validar formato del email
    If(
        !IsMatch(Trim(DataCardValue7_3.Value); "[\w-\.]+@[([\w-]+\.)+[\w-]{2,4}$");
        Set(ErrorEmail; true);;
        Set(MsgErrorEmail; "Formato de email inválido")
    ); // Validar si el email ya existe
    If(
        !IsBlank(LookUp(USUARIOS; Email = Trim(DataCardValue7_3.Value)));
        Set(ErrorEmail; true);;
        Set(MsgErrorEmail; "Este correo ya está registrado")
    ); // Verificar si las contraseñas coinciden
    If(
        DataCardValue7_1.Value <> DataCardValue1.Value;
        Set(ErrorContraRep; true);;
        Set(MsgErrorContraRep; "Las contraseñas no coinciden")
    ); // Registro válido
    With(
        {
            nuevoUsuario: Patch(
                USUARIOS;
                Defaults(USUARIOS);
                {
                    Email: Trim(DataCardValue7_3.Value);
                    Contraseña: contraseñaCod.salida
                }
            );
        };
        Set(ErrorEmail; false);;
        Set(ErrorContra; false);;
        Set(ErrorContraRep; false);;
        Set(ErrorTerminos; false);;
        Set(MsgErrorEmail; "");;
        Set(MsgErrorContra; "");;
        Set(MsgErrorContraRep; "");;
        Set(MsgErrorTerminos; "");;
        Notify("Usuario registrado correctamente"; NotificationType.Success);;
        Refresh(USUARIOS);;
        ClearCollect(_colUsuarios; USUARIOS);;
        Navigate(Ajustes; ScreenTransition.Fade; (locUsuario: LookUp(_colUsuarios; Email = Trim(DataCardValue7_3.Value))))
    )
)
```

pág. 19

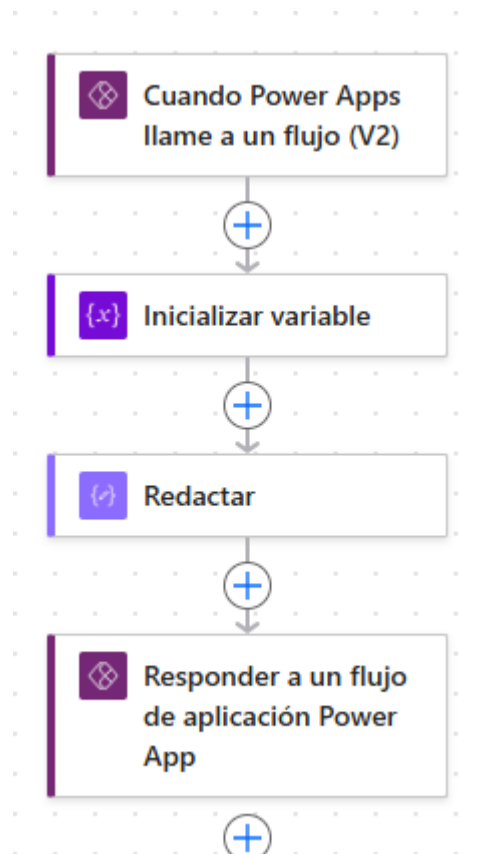
6.4.3 Cuestiones de diseño e implementación reseñables.

6.4.3.1 Implementación de flujos con PowerAutomate

Para ayudarme en algunos procesos de mi aplicación he utilizado PowerAutomate que también pertenece a Microsoft y sirve para realizar distintos tipos de flujos:

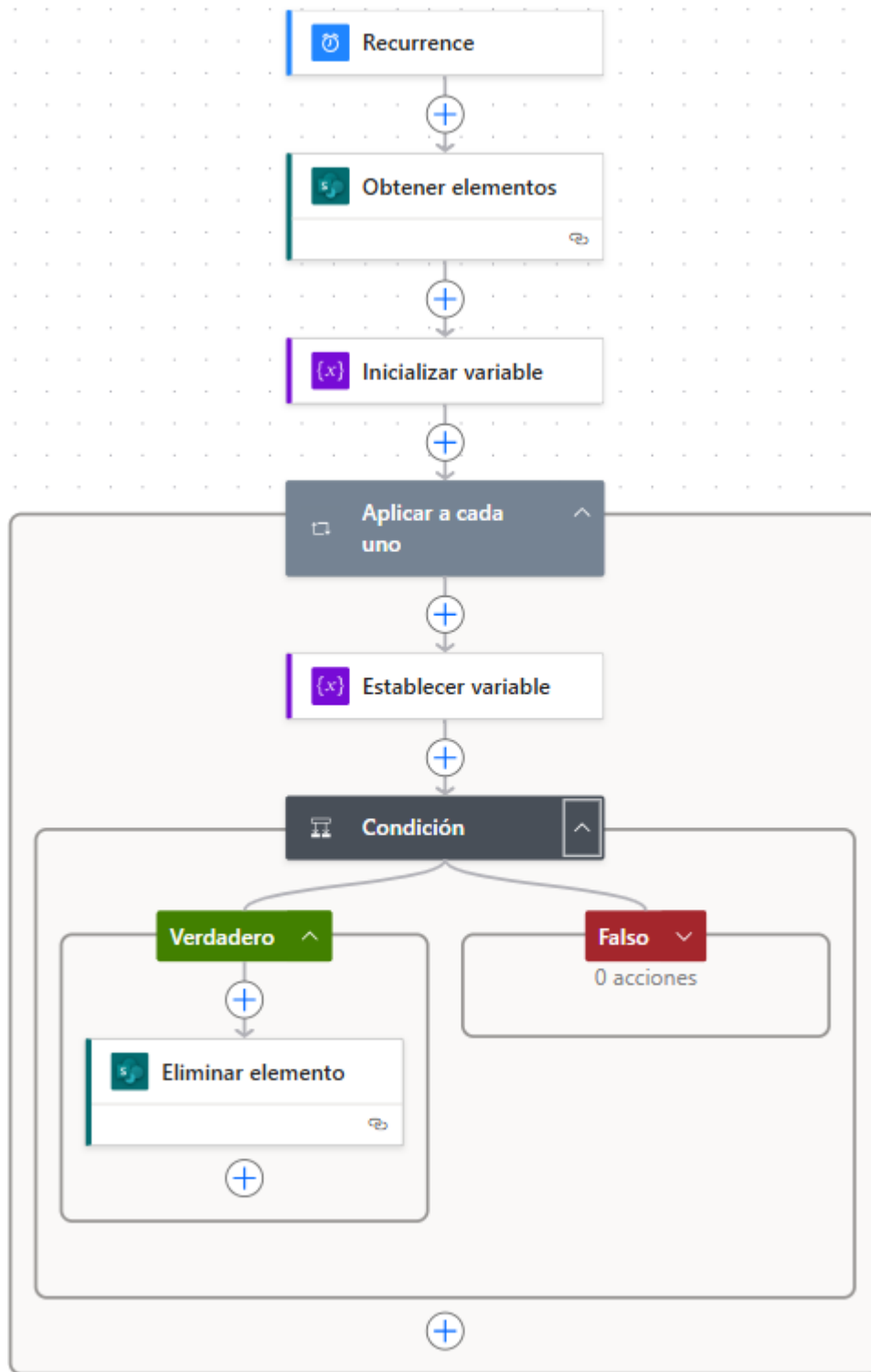
- Los hay instantáneos, que se desencadena de manera manual siempre que sea necesario
- Los hay automatizados, que se desencadenan por un evento designado.
- Los hay programados; que se puede elegir cuándo y con qué frecuencia se ejecuta.

6.4.3.1.1 CodificarPassword



Este flujo es instantáneo, el desencadenador es una llamada al flujo desde PowerApps y lo que hace es con base64() codifica la contraseña para hacer más segura el tema de los usuarios, una vez codificada la devuelve a PowerApps

6.4.3.1.2 Eliminar reservas pasadas



Este flujo es de tipo programado, se ejecuta cada 15 minutos y lo que hace es eliminar las reservas que se hayan pasado del día y la hora de hoy. Se obtienen los elementos de la lista de SharePoint, se inicializa una variable de tipo Cadena vacía,

se comprueba en todos los registros si la fecha y la hora de la reserva son anteriores a hoy, si lo es, elimina ese elemento y si no es, no hace nada

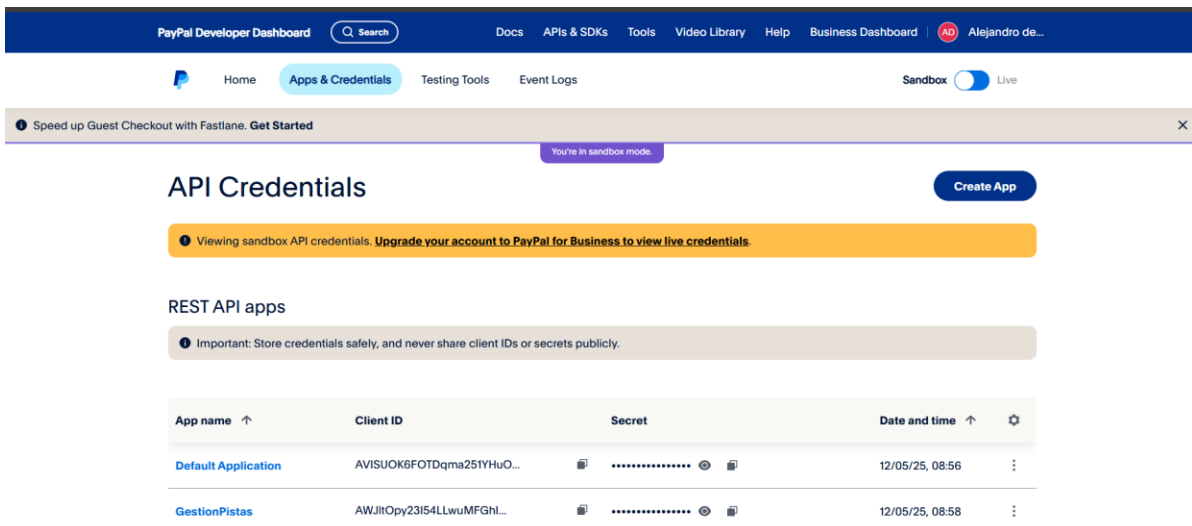
```
addMinutes(  
  addHours(  
    items('Aplicar_a_cada_uno')?['Fecha_Reserva'],  
    int(split(items('Aplicar_a_cada_uno')?  
      ['Hora_Reserva'], ':')[0])  
  ),  
  int(split(items('Aplicar_a_cada_uno')?  
    ['Hora_Reserva'], ':')[1])  
)
```

{x} Tiempo × es menor o igual que {x} utcNow() ×

6.4.3.2 Implementación de la API REST de PayPal

Para realizar un recargo del monedero de mi aplicación y del saldo de la base de datos para poder realizar reservas en la aplicación he optado por añadir la API REST de PayPal. Para poder hacer esto he tenido que realizar varios pasos.

1. Crear una cuenta en PayPal Developer
2. Una vez dentro dirigirnos a Apps y Credentials
 - a. Dentro de esto podremos crear una nueva aplicación y nos dará una cuenta tipo SandBox con dinero y credenciales



The screenshot shows the PayPal Developer Dashboard. The top navigation bar includes links for Docs, APIs & SDKs, Tools, Video Library, Help, Business Dashboard, and a user profile. The main content area is titled 'API Credentials' and features a 'Create App' button. A yellow banner indicates that the user is viewing sandbox API credentials and suggests upgrading to PayPal for Business to view live credentials. Below this, a section titled 'REST API apps' contains an important note about storing credentials safely. A table lists the created applications:

App name	Client ID	Secret	Date and time
Default Application	AVISUOK6FOTDqma251YHuO...	12/05/25, 08:56
GestionPistas	AWJltOpY23i54LLwuMFGHl...	12/05/25, 08:58

3. Si nos vamos a Testing Tools tendremos cuentas de prueba para poder realizar los pagos, en el caso de querer pagar escogemos la de tipo personal

Sandbox test accounts

Create bulk accounts

Create account

You can use sandbox accounts to test your apps and mimic live transactions. [Read more about sandbox testing](#)

Missing a test account? [Link other sandbox accounts to this developer account](#). Have your sandbox logins ready.

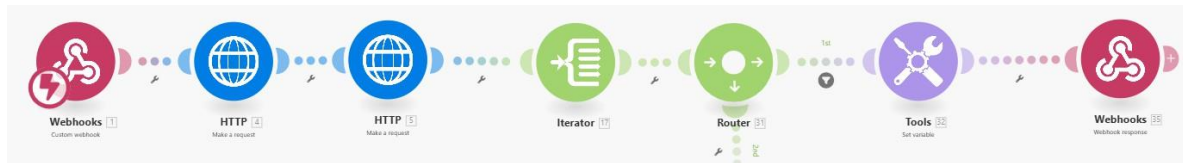
Showing 2 sandbox accounts:

Account name ↑		Apps	Type	Country	Date created ↑	Settings
sb-kckt4342031324@business.example.com	Default	2	Business	ES	12/05/25, 08:56	⋮
sb-0dhy542051191@personal.example.com	Default	0	Personal	ES	12/05/25, 08:56	⋮

- Y dentro de esta tendremos el correo y la contraseña para poder realizar el pago ficticio

6.4.3.2.1 Uso de Make.com para llamar a la API

Make.com es una herramienta online para crear flujos al igual que PowerAutomate. He optado por utilizar esto porque los controles de PowerAutomate de tipo HTTP eran de pago y esto es gratis.



Este flujo lo que hace es: el primer WebHook crea un enlace para poder ejecutar el flujo, en el primer HTTP se llama a <https://api.sandbox.paypal.com/v1/oauth2/token> para poder coger el token de autenticación necesario para poder hacer la transacción, el siguiente HTTP es el que llama a la API de PayPal con el monto que le pasemos desde el enlace con PowerApps y nos da un array de links con la redirección a la página de PayPal, este también, si el pago es realizado o es cancelado redirigirá a unas páginas alojadas en GitHub Pages (más tarde hablaré de ello), el Iterator recoge ese Array de links y el router recoge el tipo que necesitamos para la llamada, en mi caso esta condición:

Label

Set the route as a fallback. A fallback route is a backup route that is used if the source data didn't go through any other route. One router can have only one fallback route.

☐ Yes ☒ No

Condition

17. rel

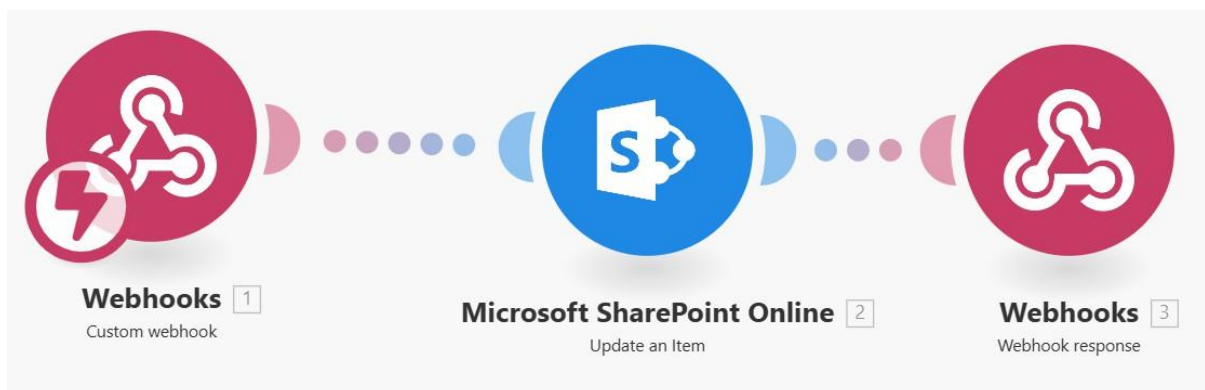
Text operators: Equal to

approve

Add AND rule Add OR rule

El último webhook es el que redirecciona a la página de pago gracias a la condición anterior.

El segundo HTTP, al ser el pago realizado, en vez de redireccionar a la página de pago realizado llama a otro flujo de Make.com que actualiza el saldo de la lista de SharePoint con el monto introducido.



Todo esto se ejecuta después de pulsar el botón de Recargar Paypal en mi pantalla de Ajustes:

Monedero: 10 €

Recargar Paypal

20

Correo y contraseñas de prueba:

PayPal

Al disponer de una cuenta PayPal, las compras que cumplan los requisitos estarán cubiertas por nuestra política de Protección del comprador. También tendrás la posibilidad de activar nuestro programa Reembolso de gastos de devolución. [Consultar condiciones](#)

Correo electrónico o número de móvil

sb-0dhy542051191@personal.example.com

Contraseña

••••••••

Mostrar

[¿Has olvidado la contraseña?](#)

Iniciar sesión

O

Pagar con tarjeta de débito o crédito

[Cancelar y volver a Test Store](#)

[Español](#) | [English](#)

Página de pago una vez iniciada sesión:

JD

PayPal

20,00 EUR

Enviar a John Doe

calle Vilamari 76993- 17469, 02001 ALBACETE, ALBACETE

Cambiar

Pagar con

☒

Saldo de PayPal

20,00 EUR

☐ Establecer como mi forma de pago preferida

☐

CaixaBank SA

Cuenta corriente ****2857

☐

Visa

Crédito ****8877

+ Añadir tarjeta

☐ Paga en 3 plazos

No disponible para esta compra

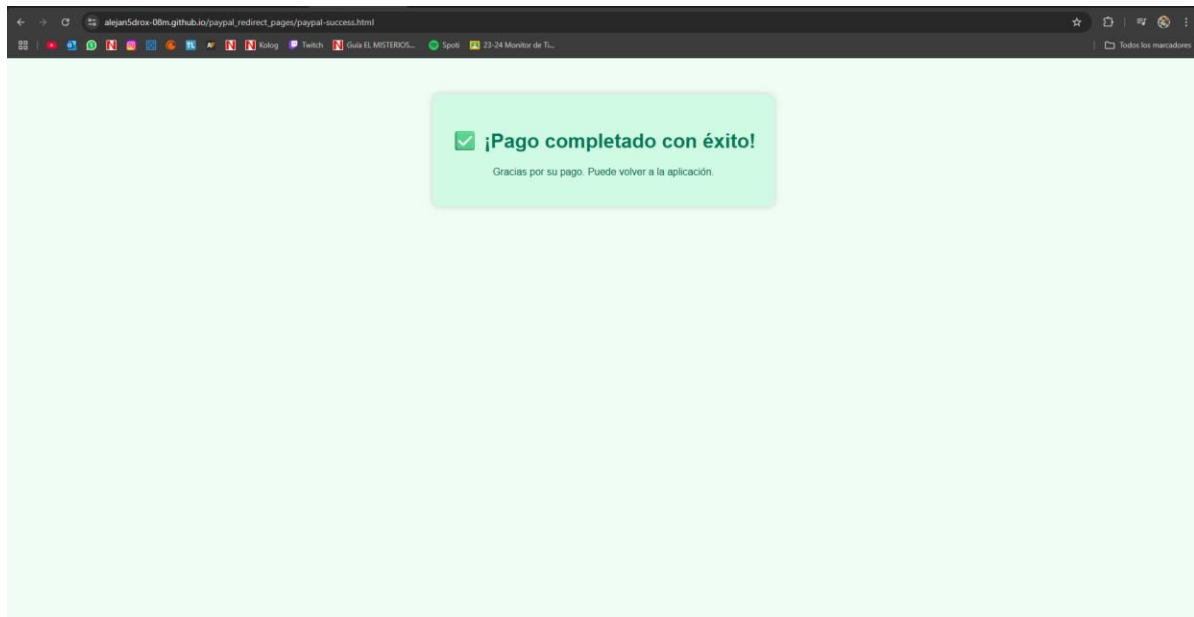
3 plazos sin intereses para compras elegibles de 30,00 EUR a 2.000,00 EUR.

Continuar y revisar pedido

Puede haber tarifas adicionales por gastos de envío o impuestos. Aceptas pagar hasta 25,00 EUR. Verás el importe final cuando vuelvas al vendedor.

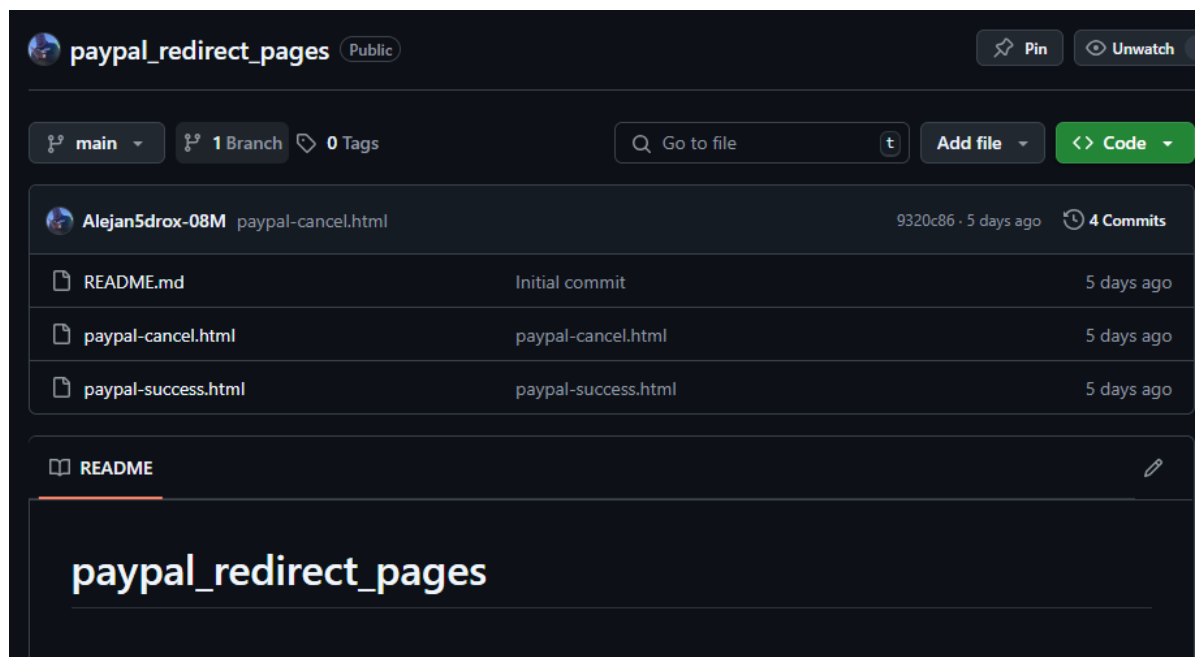
[Cancelar y volver a Test Store](#)

Si el pago es realizado redirección a página de success:



6.4.3.3 Alojamiento de archivos HTML en GitHub Pages

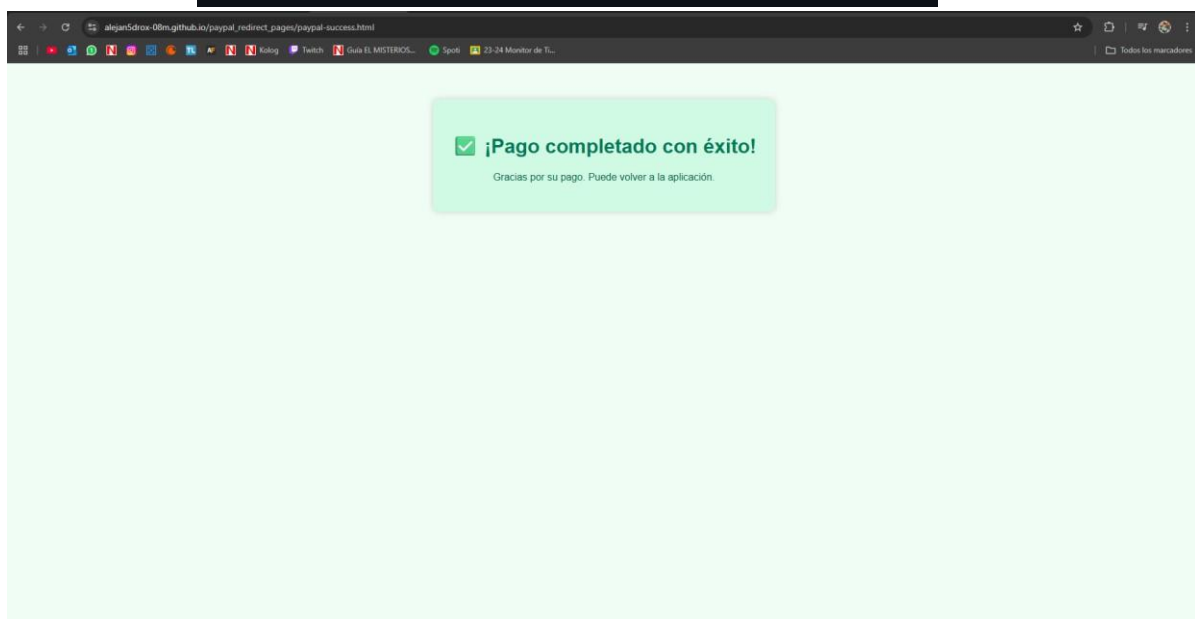
He utilizado GitHub Pages para alojar dos archivos HTML que servirán como páginas de realizado y de cancelación del pago de Paypal. Este alojamiento se crea creando un repositorio de github con un readme por defecto y después te vas a la configuración del repositorio a la sección de Pages, eliges una rama y te dará la opción de crear un enlace, y ya estarían alojadas, preparadas para ser llamadas cuando sea necesario.



Estas son las páginas:

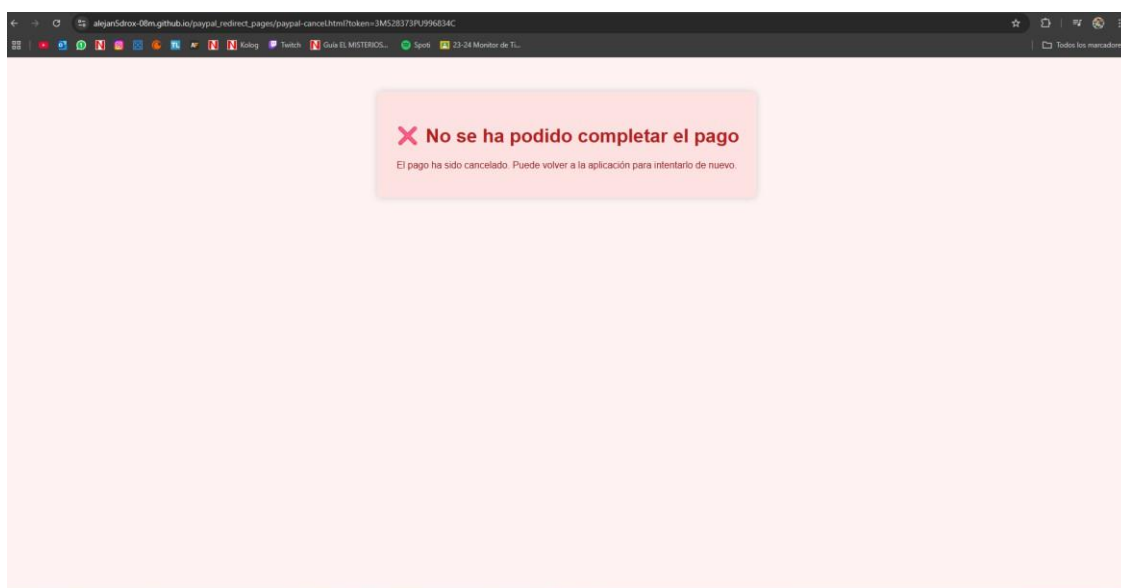
- Paypal-success.html

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <title>Pago Completado</title>
6    <style>
7      body {
8        background-color: #f0fdf4;
9        font-family: Arial, sans-serif;
10       text-align: center;
11       padding: 50px;
12       color: #065f46;
13     }
14     .container {
15       background-color: #d1fae5;
16       padding: 30px;
17       border-radius: 10px;
18       box-shadow: 0 0 10px #ccc;
19       display: inline-block;
20     }
21     h1 {
22       color: #047857;
23     }
24     a {
25       color: #065f46;
26       text-decoration: none;
27       font-weight: bold;
28     }
29   </style>
30 </head>
31 <body>
32   <div class="container">
33     <h1>✅ ¡Pago completado con éxito!</h1>
34     <p>Gracias por su pago. Puede volver a la aplicación.</p>
35   </div>
36 </body>
37 </html>
```



- Paypal-cancel.html

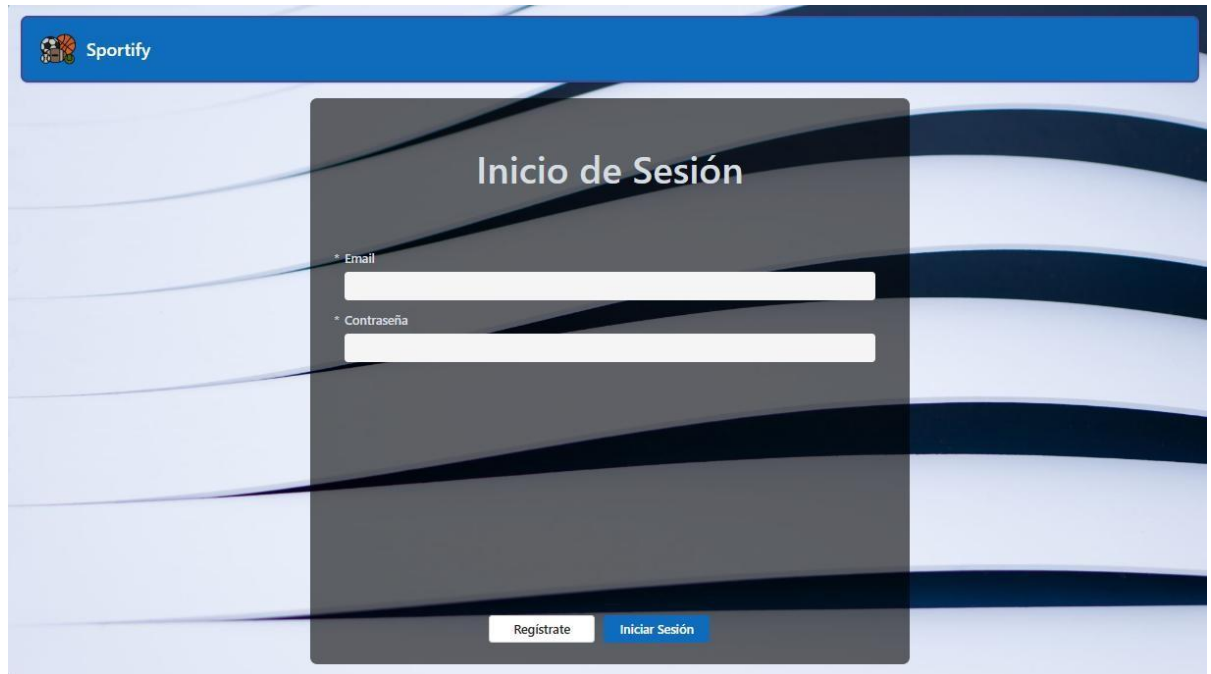
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Pago Cancelado</title>
  <style>
    body {
      background-color: #fef2f2;
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 50px;
      color: #991b1b;
    }
    .container {
      background-color: #fee2e2;
      padding: 30px;
      border-radius: 10px;
      box-shadow: 0 0 10px #ccc;
      display: inline-block;
    }
    h1 {
      color: #b91c1c;
    }
    a {
      color: #991b1b;
      text-decoration: none;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>✖ No se ha podido completar el pago</h1>
    <p>El pago ha sido cancelado. Puede volver a la aplicación para intentarlo de nuevo.</p>
  </div>
</body>
</html>
```



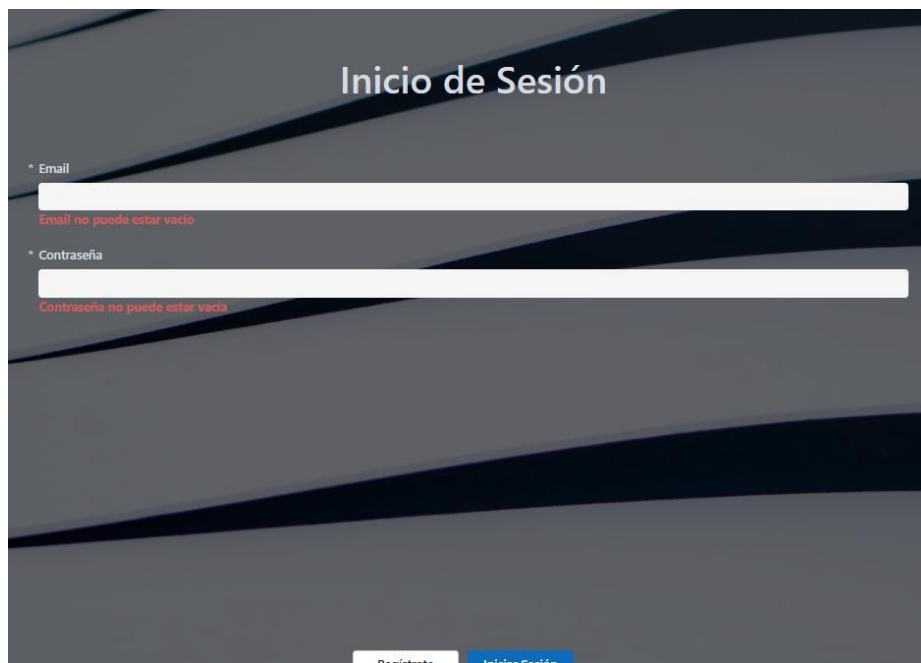
6.5 Pruebas.

6.5.1 Prueba de inicio de sesión

Tenemos la pantalla de Inicio de Sesión:

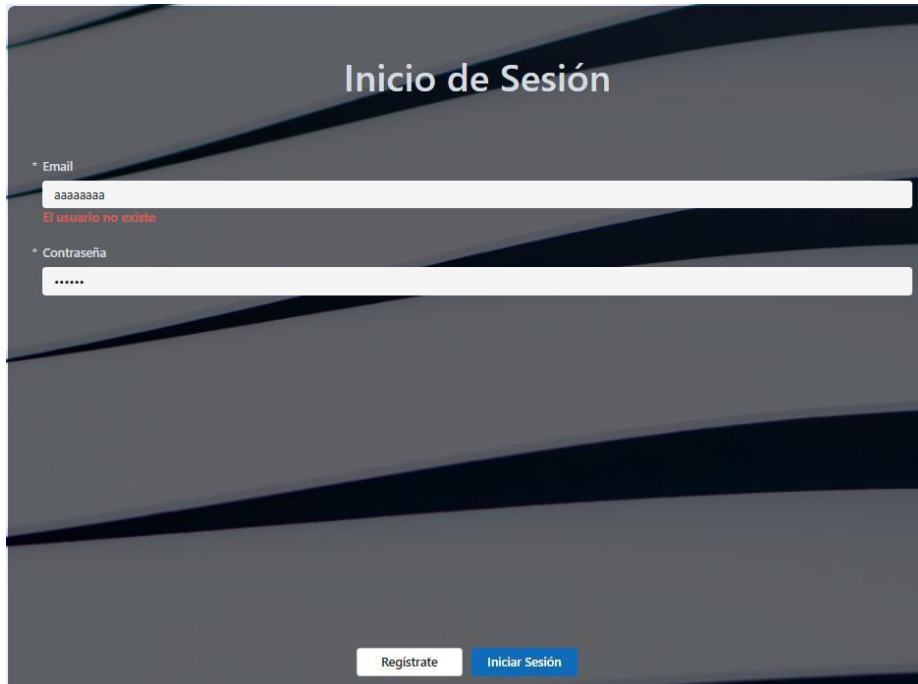


Le damos al botón de Inicio de Sesión:



Salen errores.

Rellenamos email y contraseña con datos que no existen en la base de datos:



Inicio de Sesión

* Email
aaaaaaa
El usuario no existe

* Contraseña

Registrate Iniciar Sesión

Saldrá error.

Rellenamos Email que existe y contraseña que no es correcta:



Inicio de Sesión

* Email
a@gmail.com

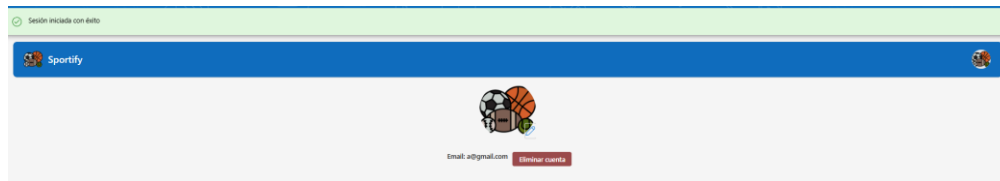
* Contraseña

Contraseña incorrecta

Registrate Iniciar Sesión

Saldrá error.

Rellenamos todo correctamente:

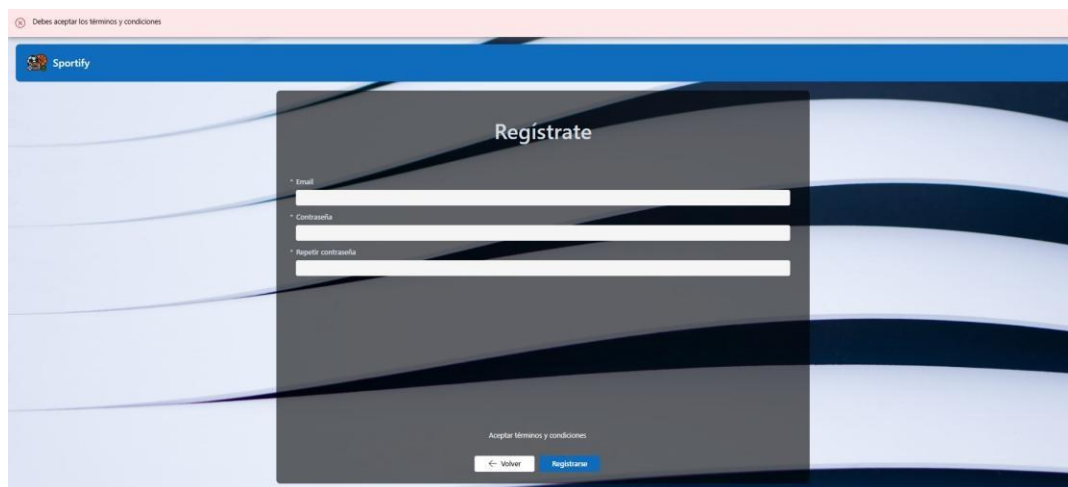


Navegamos a pantalla de Ajustes y nos sale mensaje correcto.

6.5.2 Prueba de registro

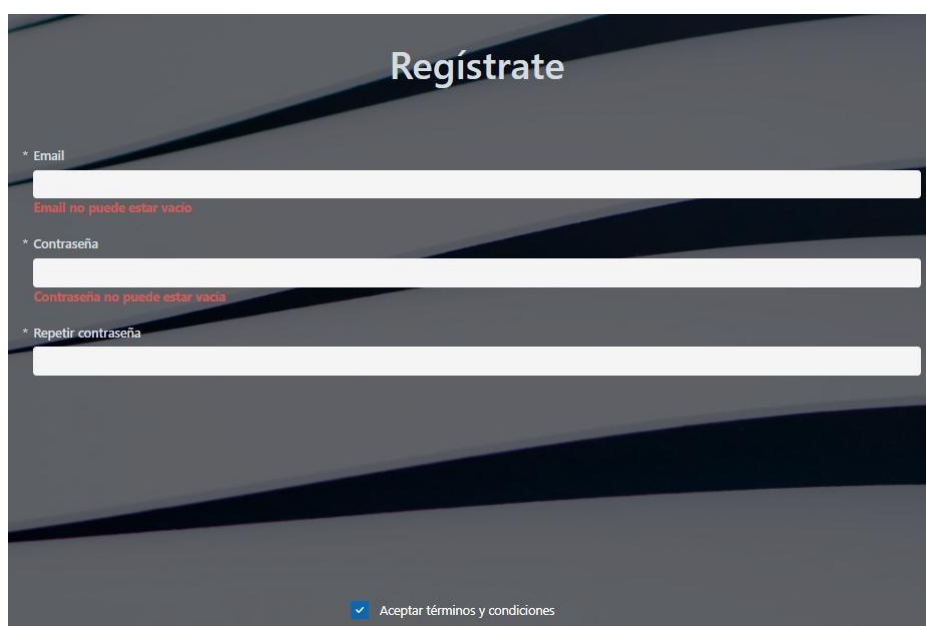
En la pantalla de Registro:

Si le damos al botón de Registrarse con todo vacío:



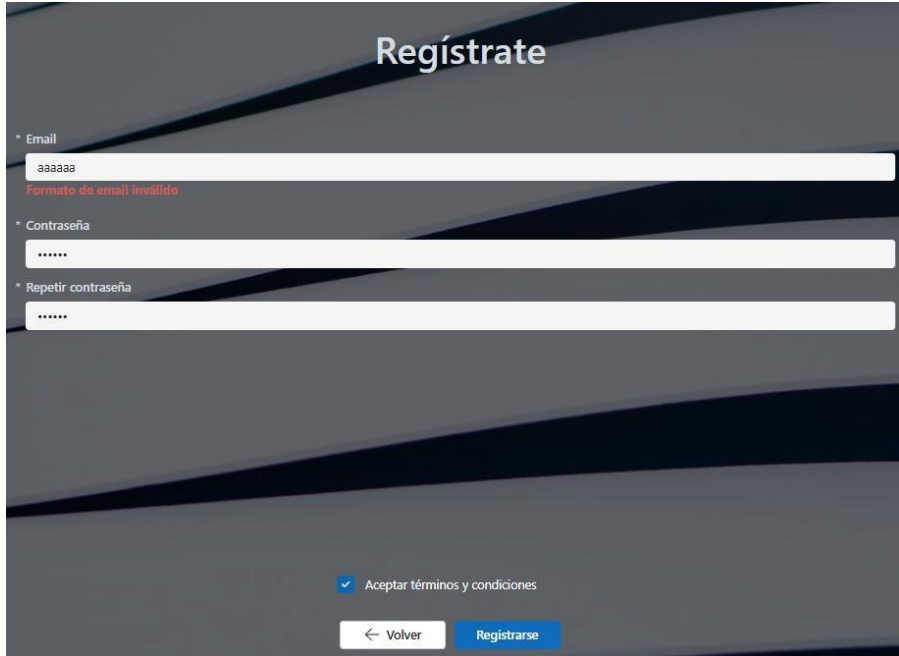
Saldrá error.

Si aceptamos las condiciones:



Saldrá error.

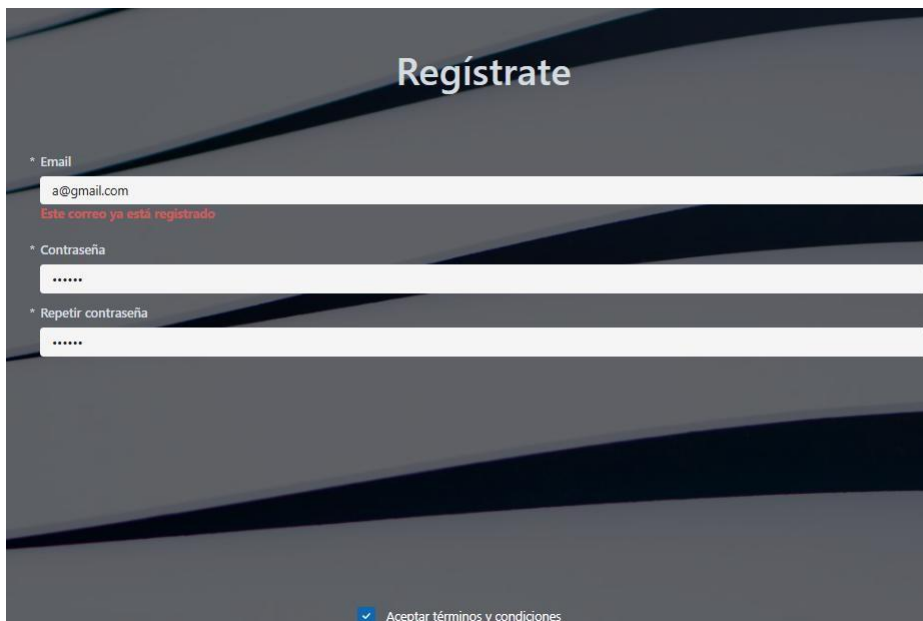
Si rellenamos el Email con un formato que no es válido:



The screenshot shows a registration form titled "Regístrate". It has three input fields: "Email" with the text "aaaaaa" and a red error message "Formato de email invalido" below it; "Contraseña" with masked characters "*****"; and "Repetir contraseña" with masked characters "*****". At the bottom, there is a checked checkbox for "Aceptar términos y condiciones" and two buttons: "Volver" and "Registrarse".

Saldrá error.

Si rellenamos Email con un correo que ya está registrado:



The screenshot shows the same registration form titled "Regístrate". The "Email" field now contains "a@gmail.com" and has a red error message "Este correo ya está registrado" below it. The "Contraseña" and "Repetir contraseña" fields remain masked with "*****". The "Aceptar términos y condiciones" checkbox is still checked, and the "Volver" and "Registrarse" buttons are at the bottom.

Saldrá error.

Si rellenamos Email con un correo nuevo pero las contraseñas no coinciden:

Regístrate

* Email
b@gmail.com

* Contraseña
...

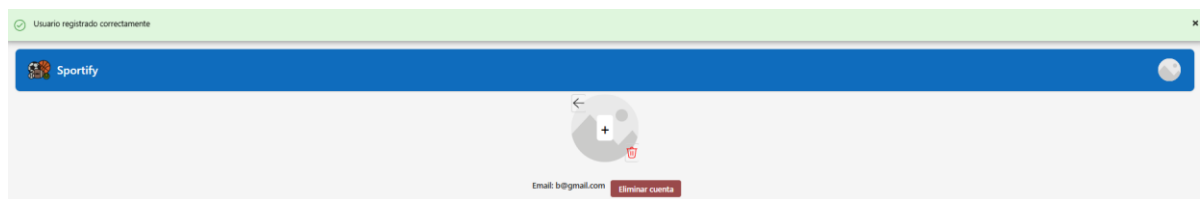
* Repetir contraseña
....

Las contraseñas no coinciden

☒ Aceptar términos y condiciones



Saldrá error.

Si todo está correcto:



Navegará a la pantalla de Ajustes y el usuario se habrá creado.

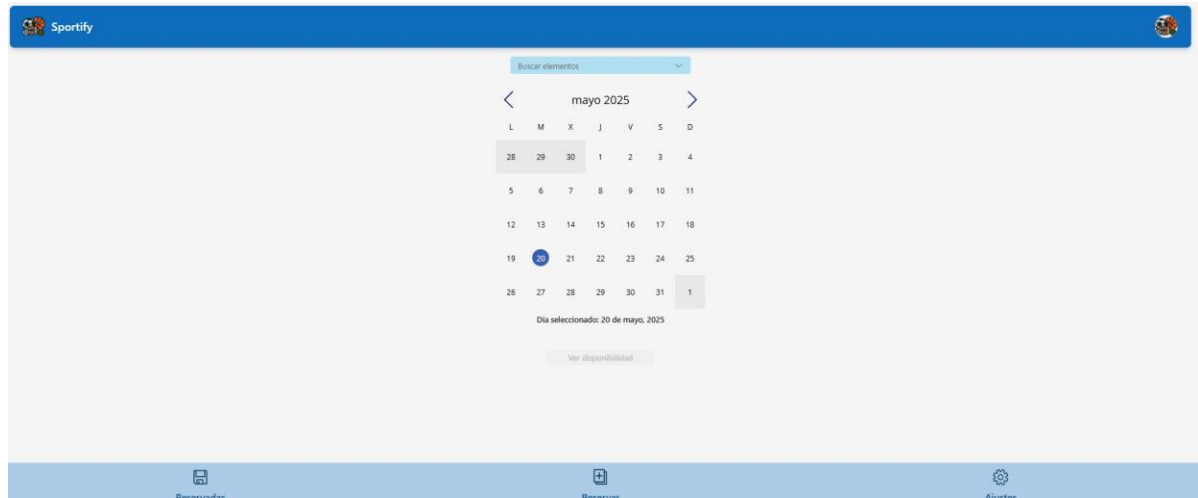
USUARIOS ☆

ID	Email	Contraseña	Saldo	Foto_Perfil	Agregar columna
13	alejandro.gremig@edu	MTIzNA==	10,00 €		
16	a@gmail.com	MTIz	30,00 €		
17	b@gmail.com	YWFh	0,00 €		

6.5.3 Prueba de creación de reserva

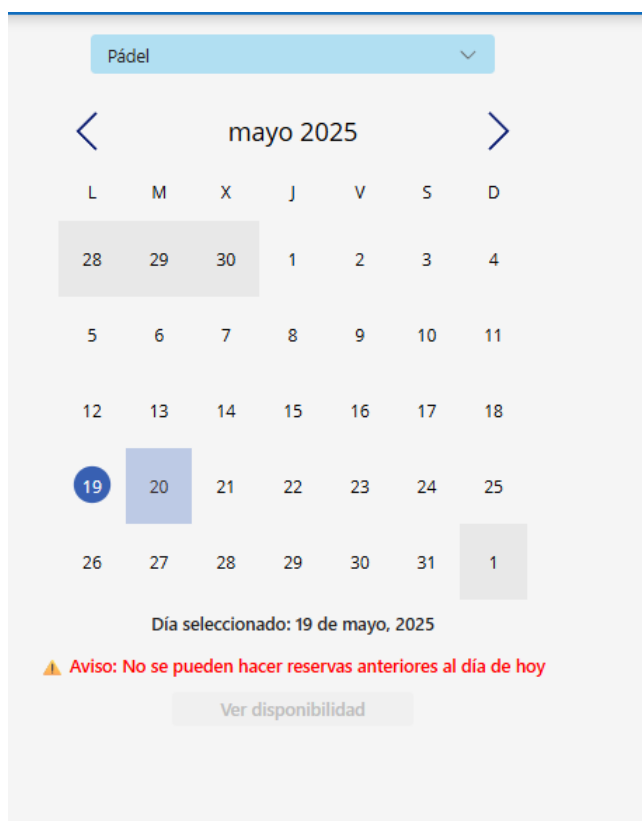
En la pantalla de Reservas:

Si no elegimos una pista en el combobox:



No se activará el botón.

Si ponemos un día anterior al día de hoy en el calendario:



Saldrá error y no se activará el botón.

Si está todo bien:

Se activará el botón y navegamos a la pantalla de disponibilidad:

Si las horas a reservar ya están reservadas o ya han pasado saldrán desactivadas esas horas. Si le damos a una de las horas disponibles:

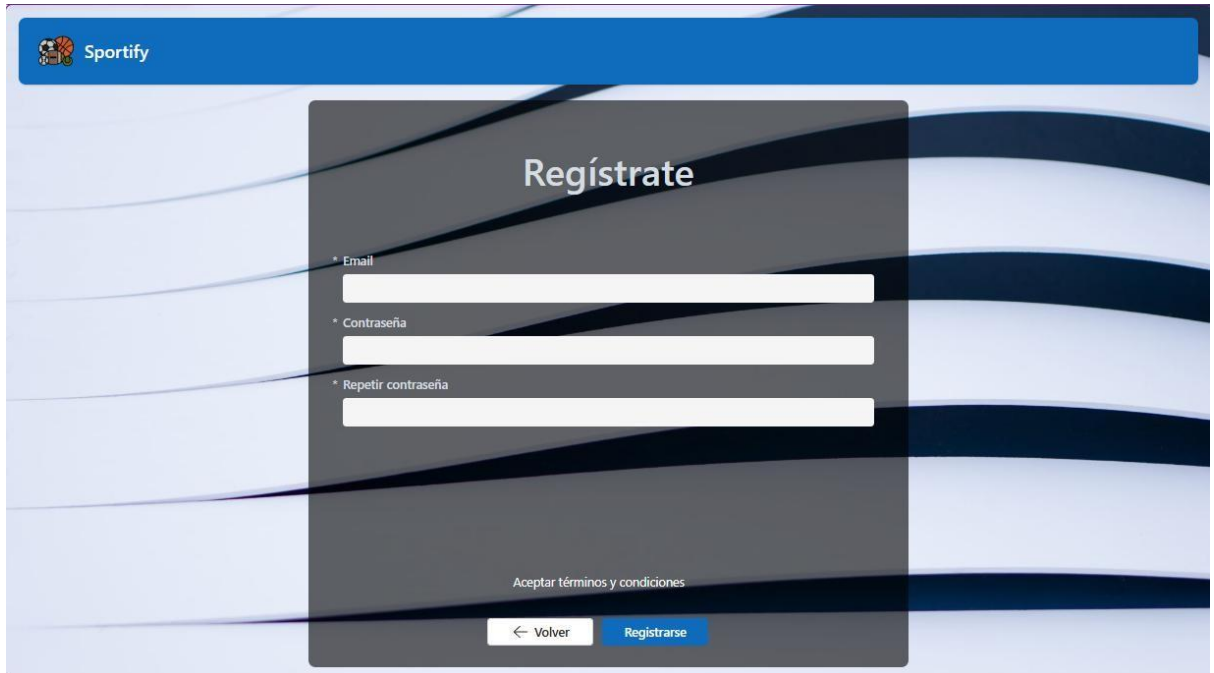
pág. 35

Se creará la reserva y navegaremos a la página de Reservadas para visualizar la reserva.

7 Manuales de usuario

7.1 Manual de usuario

7.1.1 Pantalla Registro



The screenshot shows the 'Regístrate' (Register) screen of the Sportify application. At the top, there is a blue header with the Sportify logo and name. The main content area is a dark gray card with the title 'Regístrate' in white. Below the title, there are three input fields, each preceded by an asterisk: '* Email', '* Contraseña' (Password), and '* Repetir contraseña' (Repeat password). At the bottom of the card, there is a link 'Aceptar términos y condiciones' (Accept terms and conditions) and two buttons: a white button with a left arrow and the text 'Volver' (Back), and a blue button with the text 'Registrarse' (Register).

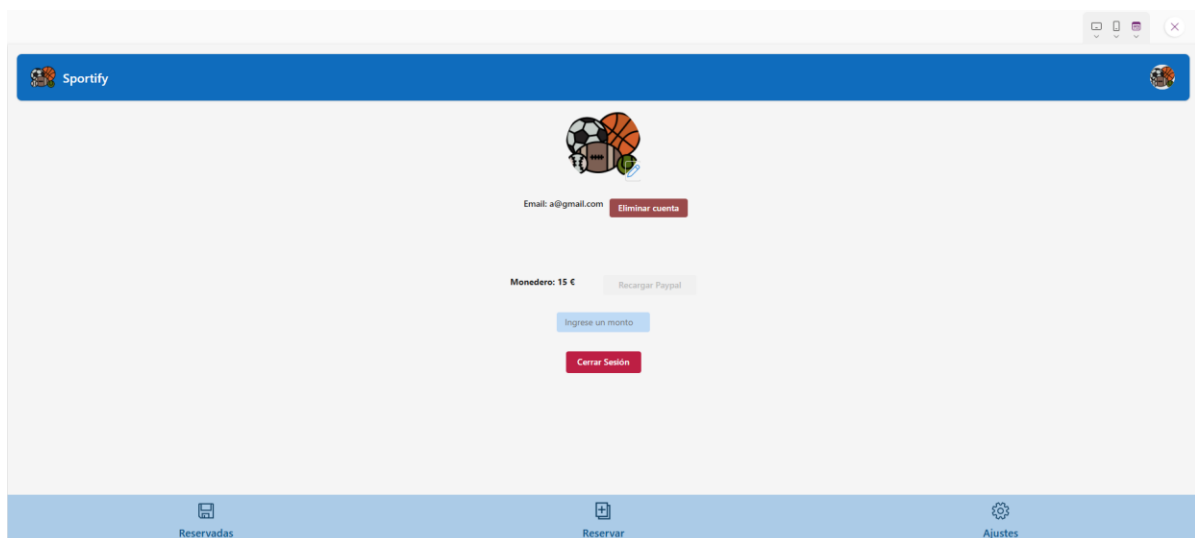
La primera pantalla que se encontrará el usuario será la de Inicio de Sesión al entrar a la aplicación. Al ser un usuario nuevo no tendrá cuenta así que tendrá que Registrarse, rellenando los campos con un Email que no exista ya en la base de datos, el email será de tipo `example@example.com/es...`, tendrá que poner dos veces su contraseña por tema extra de seguridad y aceptar el Checkbox de términos y condiciones. Una vez realizado todo esto tendrá que pulsar el botón de Registrarse y, si todo es correcto, ya podrá entrar en la aplicación.

7.1.2 Pantalla Inicio de Sesión



La primera pantalla que se encontrara el usuario. Al ser primerizo en la aplicación tendrá que ir primero a la pantalla de Registro dándole al botón de Regístrate. Si el usuario ya ha entrado anteriormente y ya tiene una cuenta creada, tendrá que rellenar el formulario con los datos de su cuenta, si comete algún error al introducir el correo o la contraseña el propio formulario lo dirá al dar al botón Iniciar Sesión. Si toda la información es correcta entrará dentro de la aplicación.

7.1.3 Pantalla Ajustes



La pantalla a la que te redirige después del Inicio de Sesión o del Registro. Contiene la información relacionada con el usuario. Dentro de la imagen de

foto de Perfil hay un botón para editar la foto de perfil. Una vez pulsado saldrá la opción de subir una foto de tu ordenador, volver a la foto anterior o incluso eliminar la que tenías.



El botón eliminar cuenta, elimina tu cuenta y las reservas del usuario y te redirige a la pantalla de inicio de sesión.



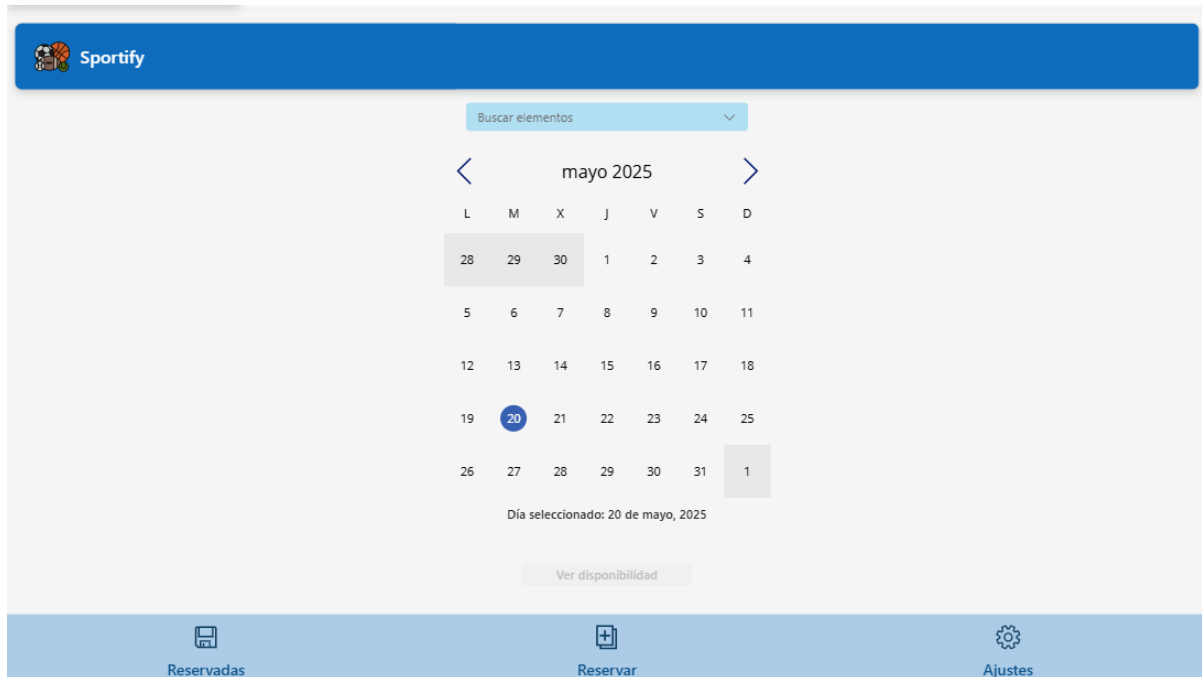
El monedero es el saldo que tienes en tú cuenta, es necesario para hacer reservas. Si quieres recargar ese monedero tendrás que pulsar el botón Recargar Paypal, inicialmente deshabilitado hasta que se introduzca un monto en el TextField de ingresar monto.



Y obviamente el botón de Cerrar Sesión cerrara sesión de tu cuenta y volverás a la pantalla de inicio de sesión.

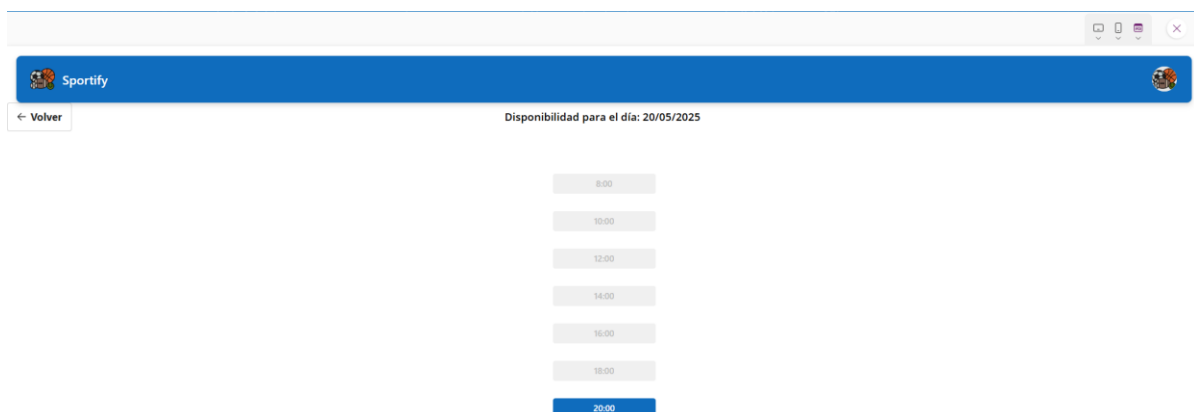
La pantalla también tiene un menú de navegación inferior que navegará entre las distintas pantallas de la aplicación.

7.1.4 Pantalla Reservas



La pantalla de Reservas contiene un selector de Pista; ya puede ser pista de Pádel, Fútbol Sala, Tenis o baloncesto, un calendario para seleccionar la fecha de tu reserva. No puedes escoger un día que sea anterior a hoy. Al escoger una pista y un día se desbloqueará el botón de Ver Disponibilidad que te llevará a otra pantalla con un listado de horas para reservar en uno de ellos.

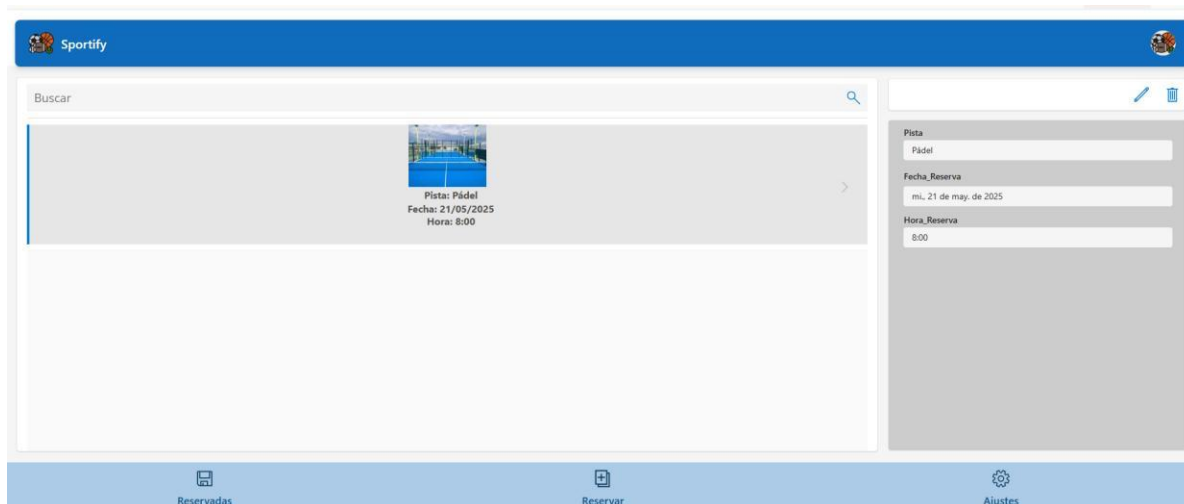
7.1.4.1 Pantalla Disponibilidad



Esta es la pantalla de Disponibilidad, aquí tenemos una lista de botones que servirán para escoger la hora de la reserva de la pista que hayas escogido anteriormente en la pantalla de Reservas. Si algún usuario ya tiene reservada esa hora para esa pista no

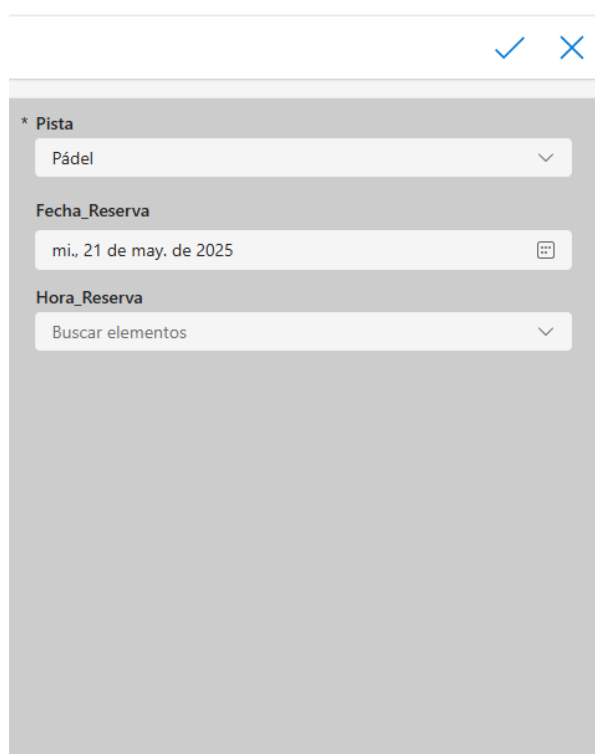
sé podrá seleccionar. También no se podrá seleccionar una reserva si la hora ya se ha pasado. Una vez escogida una hora para reservar y, por supuesto, si tienes saldo suficiente para realizar la reserva se realizará a tu nombre.

7.1.5 Pantalla Reservadas



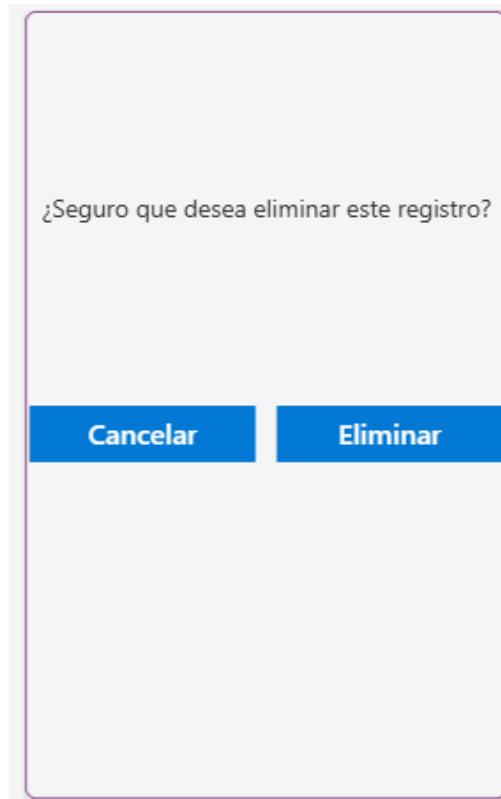
Esta es la pantalla Reservadas, aquí saldrán las reservas hechas del usuario. Si pulsas sobre las diferentes Reservas, el formulario de la derecha cargará los datos de esa tarea.

Si pulsas sobre el icono del lapicero podrás editar la tarea (aquí también se tiene en cuenta las reservas realizadas a las horas escogidas para que no hay solapamiento).



Una vez hecha la edición, pulsas al icono del tick, si no estás de acuerdo con los cambios que has hecho, pulsas la cruz y los cambios se deshacen.

Si quieres eliminar tu reserva simplemente pulsas sobre el icono de la papelera.



Si le das al botón eliminar eliminarás esa Reserva y se te reembolsará el dinero que costó. El botón Cancelar simplemente cancela el estado de eliminación.

7.2 Manual de instalación

7.2.1 Requisitos Previos

Para poder acceder y utilizar correctamente la aplicación desarrollada en PowerApps, el usuario debe contar con los siguientes elementos:

- Cuenta activa de Microsoft 365 con acceso a PowerApps.
- Permisos de uso otorgados por el desarrollador o administrador de la aplicación.
- Conexión a internet estable.
- (Opcional) Dispositivo móvil con la aplicación PowerApps instalada.

7.2.2 Acceso a la Aplicación desde Navegador Web

La aplicación puede ser ejecutada directamente desde cualquier navegador moderno a través de la plataforma oficial de PowerApps:

1. Acceder a <https://make.powerapps.com>
2. Iniciar sesión con las credenciales de Microsoft 365.
3. En el menú lateral izquierdo, seleccionar la opción “Aplicaciones”.
4. Localizar la aplicación compartida utilizando el buscador si es necesario.
5. Hacer clic sobre la aplicación para iniciarla.
6. Aceptar los permisos solicitados si es la primera vez que se accede.

Opcionalmente, se puede fijar la aplicación al panel lateral izquierdo mediante la opción “Anclar a la barra de navegación”.

7.2.3 Instalación en Dispositivo Móvil (Opcional)

Para una mayor accesibilidad, es recomendable instalar la aplicación PowerApps en el teléfono móvil del usuario.

7.2.3.1 Android

1. Abrir la aplicación Google Play Store.
2. Buscar “PowerApps” (Microsoft Corporation).
3. Instalar y abrir la aplicación.
4. Iniciar sesión con las credenciales de Microsoft 365.
5. Localizar la aplicación en la lista y pulsar para acceder.

7.2.3.2 iOS

1. Acceder a App Store.
2. Buscar “PowerApps” (Microsoft).
3. Instalar y abrir la aplicación.
4. Iniciar sesión con la cuenta Microsoft.
5. Seleccionar la aplicación deseada para abrirla.

En ambos sistemas, se puede marcar la aplicación como favorita o anclarla al inicio para facilitar futuros accesos.

8 Conclusiones y posibles ampliaciones

8.1 Conclusión

Como conclusión este trabajo ha sido duro de roer por querer hacer tantas cosas en un programa que he usado solo en mi periodo de prácticas, que he aprendido a usar solo, aunque he tenido un poco de ayuda de unas guías que nos dieron en la empresa. La verdad que este proyecto al principio fue ambicioso y quise poner muchas cosas, pero por temas de licencias premium no he podido aplicarlas. Finalmente, este trabajo me ha servido para aprender más a utilizar PowerApps, PowerAutomate y las tecnologías vistas en la empresa.

8.2 Posibles ampliaciones

- Añadir más métodos de pago en la Aplicación, como GooglePay
- Hacer la codificación de la contraseña más segura, en vez de con base64 con sha256
- Mejorar visualmente alguna parte de la aplicación (hacerla más vistosa, poner menos componentes, repartirlos entre más pantallas...)
- Añadir un historial de reservas

9 Bibliografía

<https://learn.microsoft.com/es-es/power-apps/>

<https://learn.microsoft.com/es-es/power-automate/>

<https://www.make.com/en>

<https://pages.github.com/>

<https://developer.paypal.com/api/rest/>

https://github.com/Alejan5drox-08M/paypal_redirect_pages.git

<https://learn.microsoft.com/es-es/power-apps/maker/canvas-apps/test-studio>

10 Anexos

10.1 OnStart

```
ClearCollect(_colUsuarios;USUARIOS);;  
ClearCollect(_colReservas;RESERVAS);;  
ClearCollect(
```

```
_ColWeekDays;  
{Dia:"L"};  
{Dia:"M"};  
{Dia:"X"};  
{Dia:"J"};  
{Dia:"V"};  
{Dia:"S"};  
{Dia:"D"}  
);;  
Set(_dateSelected; Today());;  
Set(_firstDayOfMonth; DateAdd(Today(); 1 - Day(Today());  
TimeUnit.Days));;  
Set(_firstDayInView; DateAdd(_firstDayOfMonth; -  
(Weekday(_firstDayOfMonth) - 2); TimeUnit.Days));;  
Set(_lastDayOfMonth; DateAdd(DateAdd(_firstDayOfMonth; 1;  
TimeUnit.Months);-1; TimeUnit.Days));;  
Set(_calendarVisible; false);;  
Set(_minDate; DateAdd(_firstDayOfMonth; -(Weekday(_firstDayOfMonth) -  
2); TimeUnit.Days));;  
Set(_maxDate; DateAdd(DateAdd(_firstDayOfMonth; -  
(Weekday(_firstDayOfMonth) - 2); TimeUnit.Days); 40; TimeUnit.Days));;  
Set(_calendarVisible; true)
```

https://github.com/Alejan5dnrox-08M/TFG_AdGM.git