****

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** | **Foto actual del alumno** |

Tabla de contenido

[1 Identificación proyecto 4](#_Toc127986556)

[2 Organización de la memoria 4](#_Toc127986557)

[3 Descripción general del proyecto 4](#_Toc127986558)

[3.1 Objetivos 4](#_Toc127986559)

[3.2 Cuestiones metodológicas 4](#_Toc127986560)

[3.3 Entorno de trabajo (tecnologías de desarrollo y herramientas) 4](#_Toc127986561)

[4 Descripción general del producto 4](#_Toc127986562)

[4.1 Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar. 4](#_Toc127986563)

[4.2 Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas. 4](#_Toc127986564)

[4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha 4](#_Toc127986565)

[5 Planificación y presupuesto 4](#_Toc127986566)

[6 Documentación Técnica: análisis, diseño, implementación y pruebas. 4](#_Toc127986567)

[6.1 Especificación de requisitos 4](#_Toc127986568)

[6.2 Análisis del sistema 4](#_Toc127986569)

[6.3 Diseño del sistema: 4](#_Toc127986570)

[6.3.1 Diseño de la Base de Datos 4](#_Toc127986571)

[6.3.2 Diseño de la Interfaz de usuario. 4](#_Toc127986572)

[6.3.3 Diseño de la Aplicación. 4](#_Toc127986573)

[6.4 Implementación: 5](#_Toc127986574)

[6.4.1 Entorno de desarrollo. 5](#_Toc127986575)

[6.4.2 Estructura del código. 5](#_Toc127986576)

[6.4.3 Cuestiones de diseño e implementación reseñables. 5](#_Toc127986577)

[6.5 Pruebas. 5](#_Toc127986578)

[7 Manuales de usuario 5](#_Toc127986579)

[7.1 Manual de usuario 5](#_Toc127986580)

[7.2 Manual de instalación 5](#_Toc127986581)

[8 Conclusiones y posibles ampliaciones 5](#_Toc127986582)

[9 Bibliografía 5](#_Toc127986583)

[10 Anexos 5](#_Toc127986584)

# 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.

# Organización de la memoria

Esta memoria consta de 6 apartados clave:

# 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 redireccionamiento de la API de PayPal, **API de PayPal**; como método de pago ficticio en la aplicación, y **SharePoint;** como base de datos.

## 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.

## Cuestiones metodológicas

### 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.

## 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.

# 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.

## 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.

## 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.

## 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.

# 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.

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

## 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.

## 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.

## Diseño del sistema:

### Diseño de la Base de Datos

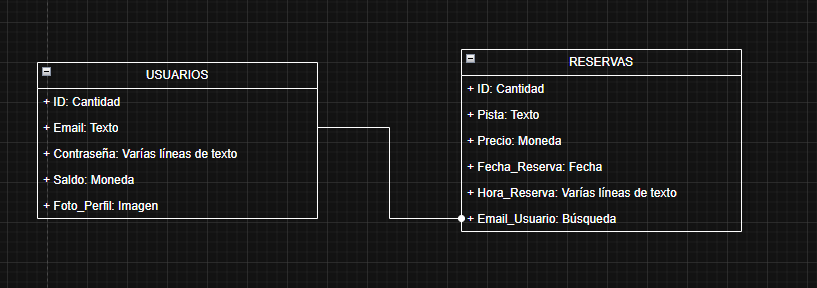
La base de datos se ha creado en un sitio de SharePoint llamado GestionPistas y tiene dos listas, USUARIOS y RESERVAS relacionadas entre sí ya que un usuario puede hacer varias reservas, pero esas mismas reservas están a nombre del propio usuario.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

#### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico El contenido generado por IA puede ser incorrecto.

#### 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.

### 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.



Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

#### Inicio de Sesión

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

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.

#### Registro

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

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.

#### Reservadas

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

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.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

#### Reservas

Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

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.

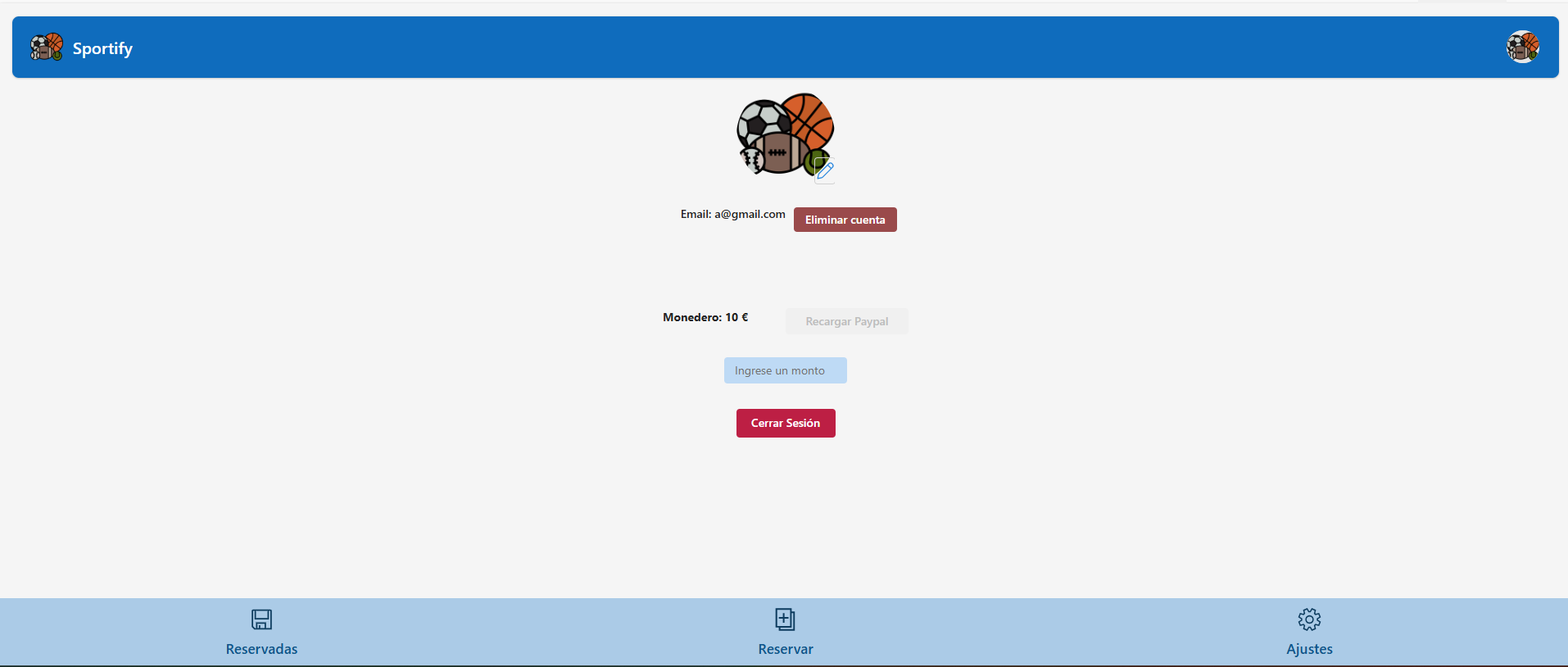
##### Disponibilidad

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

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.

#### 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.

### 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 selccionados anteriormente

## Implementación:

### 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.

### 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

#### Código botón Inicio de Sesión

Texto

El contenido generado por IA puede ser incorrecto.

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.

#### Código botón Registrar en la pantalla de Registro



Este código es bastante parecido al anterior solo que tiene más comprobaciones de campos vacíos, comprobación de que el email sea a@dominio.algo  *!IsMatch(Trim(DataCardValue7\_3.Value); "^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$");* Lo nuevo que tenemos en este código es la función *Patch* que crea un nuevo registro en la base de datos, en este caso con el Email y la contraseña Codificada por el flujo. *ClearCollect* sirve para hacer colecciones en el propio PowerApps para que las consultas a la base de datos no consuman tantos recursos (al hacer la colección de la lista de la base de datos se pueden hacer consultas en esa colección en vez de en el SharePoint para tener una respuesta más rápida)

### Cuestiones de diseño e implementación reseñables.

#### 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.

##### 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

##### Eliminar reservas pasadas

Diagrama

El contenido generado por IA puede ser incorrecto.

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

Texto

El contenido generado por IA puede ser incorrecto.

#### 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
   1. Dentro de esto podremos crear una nueva aplicación y nos dará una cuenta tipo SandBox con dinero y credenciales

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. 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

Texto

El contenido generado por IA puede ser incorrecto.

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

##### 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.

Diagrama

El contenido generado por IA puede ser incorrecto.

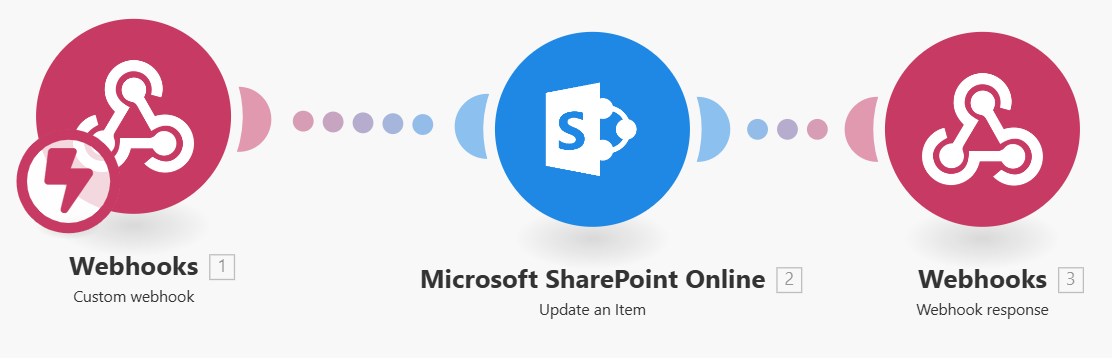
Este flujo lo que hace es: el primer WebHook crea un enlace para poder ejecutar el flujo, en el primer HTTP se llama a <https://api.sandbox.paypal.com/v1/oauth2/token> para poder coger el token de autenticación necesario para poder hacer la transacción, el siguiente HTTP es el que llama a la API de PayPal 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:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

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:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Correo y contraseñas de prueba:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Página de pago una vez iniciada sesión:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Si el pago es realizado redirección a página de success:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

#### Alojamiento de archivos HTML en GitHub Pages

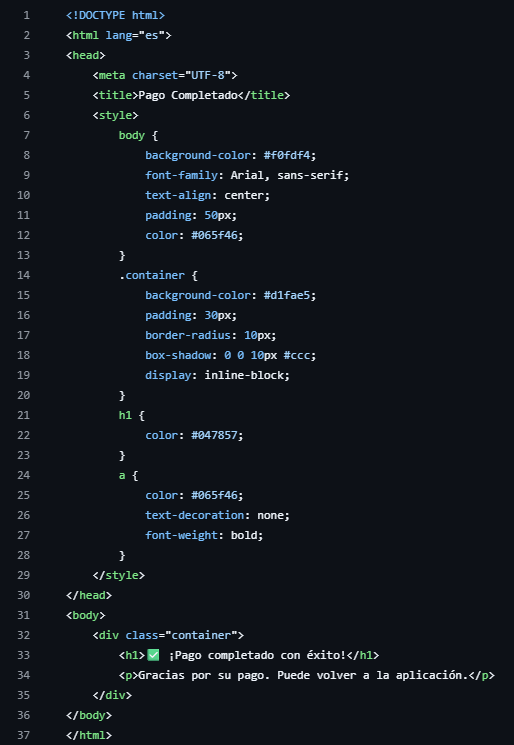
He utilizado GitHub Pages para alojar dos archivos HTML que servirán como páginas de realizado y de cancelación del pago de Paypal. Este alojamiento se crea creando un repositorio de github con un readme por defecto y después te vas a la configuración del repositorio a la sección de Pages, eliges una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

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Estas son las páginas:

* Paypal-success.html



Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

* Paypal-cancel.html

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Pruebas.

# Manuales de usuario

## Manual de usuario

## Manual de instalación

# Conclusiones y posibles ampliaciones

# 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>

# Anexos