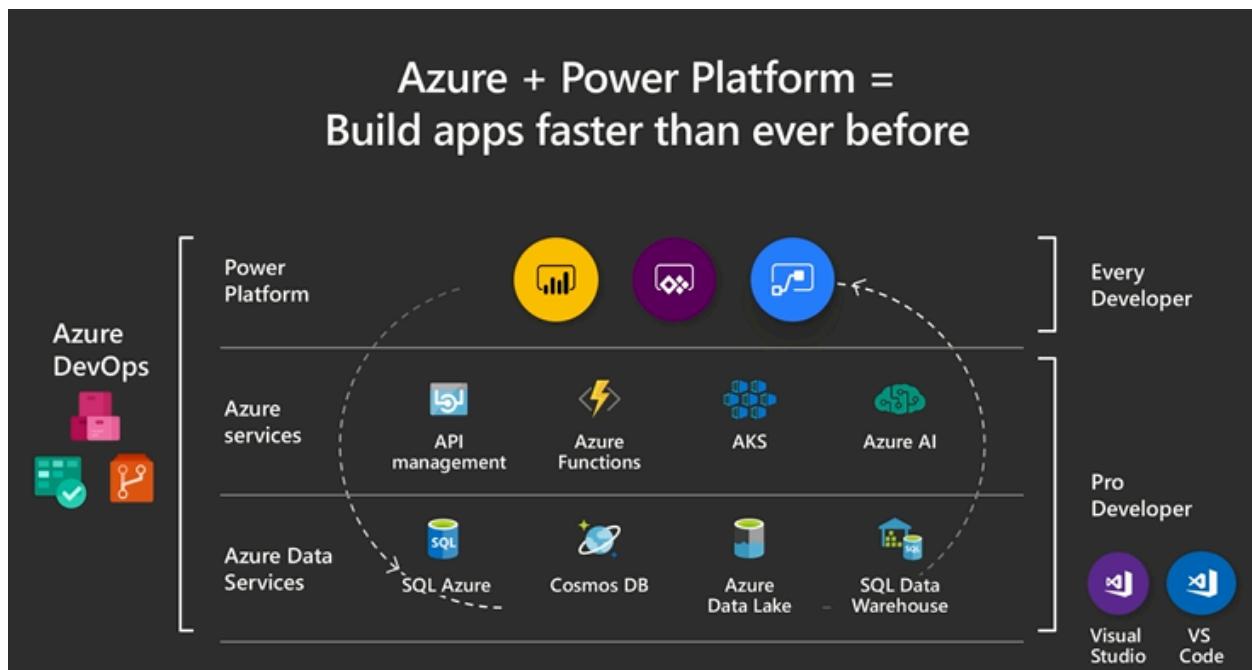


# Desarrollo de una Aplicación de Programación de Citas Médicas utilizando Microsoft Power Platform

25 de Mayo del 2024



Alumno: Sergio Felipe García

Curso: 2º - Desarrollo Aplicaciones Web - DAW

Tutor del Proyecto: Carmelo Escribano

---

<b>Resumen.....</b>	<b>4</b>
Justificación del Proyecto.....	5
Objetivos del Proyecto.....	6
<b>Desarrollo.....</b>	<b>7</b>
Introducción al entorno de desarrollo.....	7
Componentes de la solución:.....	7
¿Es importante para un Desarrollador saber Power Apps?.....	8
<b>Microsoft Dataverse.....</b>	<b>10</b>
Comparativa con otras BBDD del mercado.....	11
TABLAS.....	12
Creación de una tabla.....	13
Tipos de Tablas.....	13
Diseño de las Tablas del Proyecto.....	14
Definir una tabla: Tabla Agenda_2.....	15
CRUD (Create, Read, Update, Delete) en Power Apps.....	17
La función Patch.....	18
Crear un Nuevo Registro.....	19
Actualizar.....	19
Eliminar un Registro.....	19
Tabla Cita.....	20
Tabla Especialistas.....	22
Tabla Especialidades.....	24
Tabla Usuarios.....	25
Siguiente paso tras el modelado de la BBDD.....	26
<b>POWER APPS.....</b>	<b>27</b>
Model Driven Apps.....	28
Beneficios de esta forma de desarrollar:.....	29
Canvas Apps (Aplicaciones de lienzo).....	29
Beneficios de esta forma de desarrollar:.....	29
<b>Aplicación de Reserva de Citas.....</b>	<b>30</b>
Crear la Aplicación.....	31

---

Look & Feel.....	34
Página de inicio.....	35
Página de Registro.....	37
¿Cómo se crea un formulario en power apps?.....	38
Página de inicio de sesión.....	43
Botón Mostrar contraseña.....	43
Botón Iniciar sesión.....	44
La función Lookup.....	45
Página de reserva de cita.....	46
Galería de citas.....	47
Tarjetas de la Galería.....	47
Función Filter.....	48
Organización de la Galería.....	49
Página de tiempo de espera.....	52
Botón de espera.....	54
Pantalla de Citas del Especialista.....	55
Botón de eliminar.....	56
<b>Power Automate.....</b>	<b>57</b>
Plantillas de Power Automate.....	59
Las plantillas de Power Automate.....	59
Flujo Reservar Cita.....	61
Flujo de Recordatorio de cita.....	64
<b>Conclusión.....</b>	<b>73</b>
<b>Webgrafía.....</b>	<b>74</b>
Fuentes:.....	74
Referencias:.....	74
<b>Agradecimientos.....</b>	<b>75</b>

---

## Resumen

Hoy quiero presentarles mi trabajo de fin de grado, titulado "Desarrollo y Funcionamiento de una Aplicación de Citas Médicas utilizando Microsoft Power Platform".

Durante mi formación académica, he tenido la oportunidad de aprender diversas tecnologías y metodologías de desarrollo. Sin embargo, gracias a la empresa **INETUM**, he podido adentrarme en el uso de Microsoft Power Platform, una tecnología totalmente nueva para mí que no había sido cubierta durante el curso.

Esta experiencia ha sido fundamental para mi desarrollo profesional y académico, permitiéndome aplicar los conocimientos adquiridos en un entorno real y aprender nuevas herramientas que están revolucionando la forma en que se crean aplicaciones empresariales.

El objetivo de este documento es explicar detalladamente el desarrollo y funcionamiento de una aplicación de citas médicas, aprovechando las capacidades de Microsoft Power Platform.

Mostraré cómo se integra Power Apps para la interfaz de usuario, Dataverse para la gestión de datos, y Power Automate para la automatización de flujos de trabajo, en una solución que realiza la gestión de citas médicas.



---

## Justificación del Proyecto

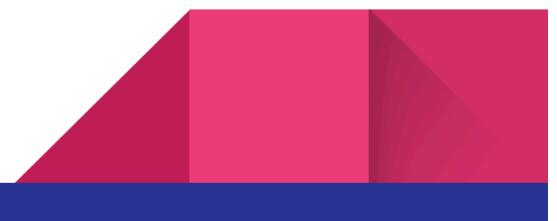
Me gustaría resaltar mi interés, cuando mi tutor de **INETUM** nos propuso realizar esta práctica. Recordé que durante el curso habíamos visto cómo podemos embeber en nuestras aplicaciones de manera muy sencilla Calendly <https://calendly.com/>, que es una herramienta de programación de citas y reuniones que simplifica el proceso de coordinación entre personas.

Durante la práctica realizada en clase, ya me di cuenta de que el conocimiento que podía adquirir en el desarrollo de una aplicación de programación de citas, ahora en **INETUM** y desde cero podía aplicarlo en mi futura vida profesional en múltiples casos de uso comunes e interesantes:

- Reservas de Servicios de Belleza y Spas
- Reservas en Restaurantes
- Reservas de Instalaciones, Gimnasios y Centros Deportivos:
- Reservas en Centros de Salud
- Clínicas y Hospitales Veterinarios.
- Etc.

Y son comunes porque proporcionan un beneficio común y claro bajo mi punto de vista, si son implementados adecuadamente por cualquiera. Algunos beneficios:

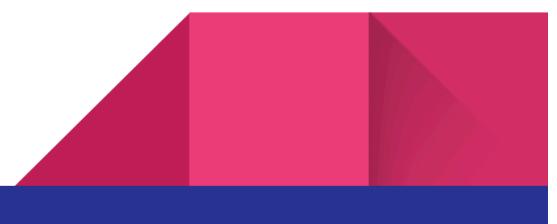
- Optimización del Tiempo de las personas.
- Reducción de Tiempos de Espera lo que mejora la experiencia del usuario.
- Mejora de la satisfacción del Cliente: mediante recordatorios automáticos siempre ayuda a mantener una buena relación con los usuarios.



- 
- Gestión eficiente de recursos o de instalaciones, equipos y espacios. Esto ayuda a reducir costos.
  - Reducir las ausencias y cancelaciones, evitando pérdidas de tiempo y dinero.

## Objetivos del Proyecto

1. **Desarrollar una aplicación de citas médicas utilizando Power Apps:** Crear una interfaz de usuario accesible para pacientes y especialistas, facilitando la gestión de citas.
2. **Implementar una base de datos centralizada en Dataverse:** Gestionar la información de usuarios, especialistas, horarios y citas de manera eficiente.
3. **Automatizar el envío de recordatorios de citas con Power Automate:** Configurar flujos de trabajo que envíen correos electrónicos de recordatorio a los pacientes antes de sus citas, permitiendo la confirmación o cancelación de las mismas.
4. **Integrar y sincronizar los horarios disponibles en tiempo real:** Mostrar los huecos disponibles en el calendario de los especialistas para que los pacientes puedan reservar citas fácilmente.
5. **Evaluar y optimizar la usabilidad de la aplicación:** Realizar pruebas con usuarios reales para recoger feedback y realizar mejoras continuas en la interfaz y funcionalidad de la aplicación.



---

# Desarrollo

## Introducción al entorno de desarrollo.

Antes de adentrarnos en los detalles específicos de la aplicación de citas médicas que he desarrollado y que voy a presentar, quisiera poner en contexto que es Microsoft Power Apps y la Power Platform.

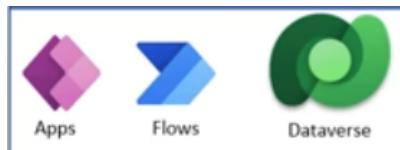
Microsoft lanzó Power Apps en 2016 y lo ideó segun ellos, con varios objetivos en mente, principalmente enfocados en “empoderar” a las organizaciones para mejorar su eficiencia y adaptabilidad en un entorno empresarial cada vez más digital.

Power Apps ha evolucionado durante estos años, incorporando conectores y capacidades que permiten a los desarrolladores crear aplicaciones conectadas a diversas fuentes de datos, tanto en la nube como locales.

De todos las herramientas y componentes que Ofrece Microsoft bajo la Power Platform, hemos utilizado los siguientes

### Componentes de la solución:

- **Power Apps:** Para el desarrollo de la interfaz de usuario.
- **Power Automate:** Para la automatización de flujos de trabajo.
- **Dataverse:** Almacenamiento y gestión centralizada de datos.



## ¿Es importante para un Desarrollador saber Power Apps?

Se ha designado a Microsoft como líder en 2023 *Gartner® Magic Quadrant™* for Enterprise Low-Code Application Platforms.



Fuentes:

- <https://www.microsoft.com/es-es/power-platform/products/power-automate>
- <https://www.gartner.com/doc/reprints?id=1-2F7NELJY&ct=231004&st=sb&culture=es-es&country=es>

Gartner define las plataformas de aplicaciones de código bajo (LCAP) o Low-Code Application Platforms, como plataformas de aplicaciones que se utilizan para desarrollar y

---

ejecutar rápidamente aplicaciones personalizadas minimizando el uso de lenguajes de programación.

Gartner estima que Microsoft Power Apps mantiene **la base de usuarios más grande del mundo** debido a la cantidad de usuarios empresariales de Microsoft 365 y Dynamics 365.

Power Apps admite soluciones empresariales basadas en API con más de 1000 conectores para muchas aplicaciones comunes y fuentes de datos basadas en la nube. Se pueden desarrollar y utilizar conectores personalizados como para API externas que no están disponibles como conectores prediseñados.

Microsoft se encuentra entre los principales actores del mercado de capacidades de IA generativa, con capacidades como AI Copilot, Power Apps Ideas y soporte de complementos de Microsoft 365 Copilot.

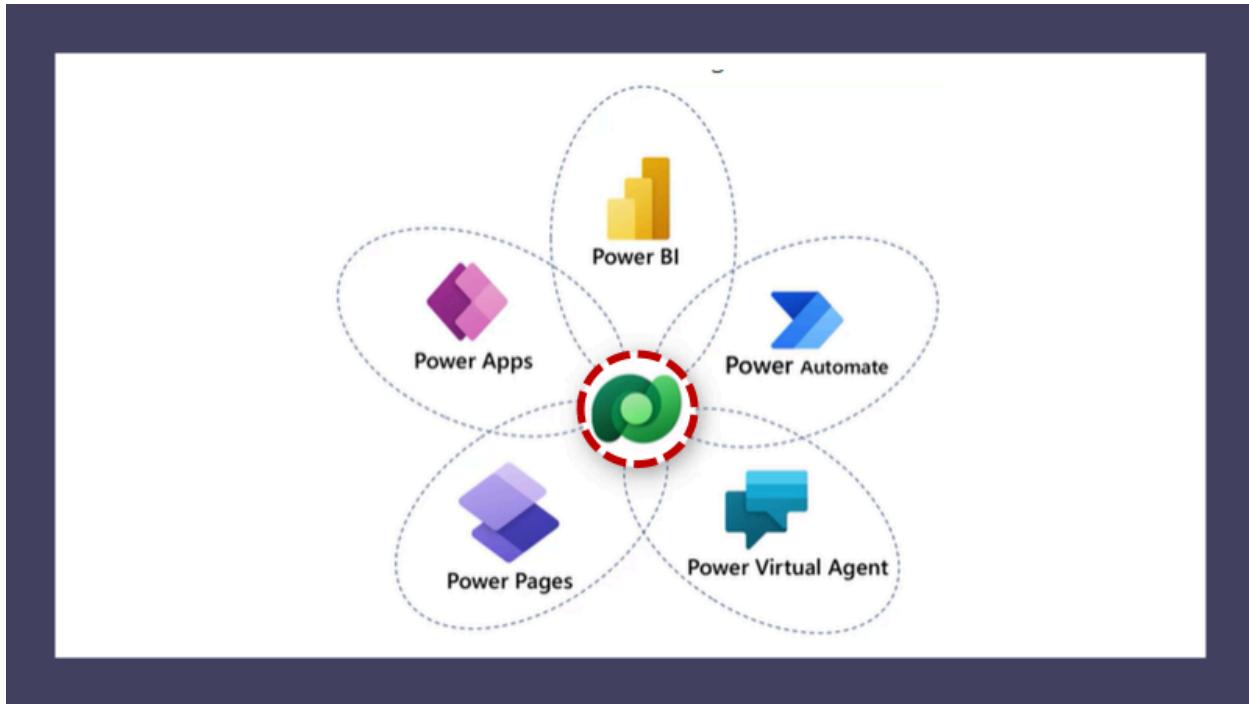
Sin embargo, sus precios son altos comparado con otras plataformas y las licencias complejas, no toda la funcionalidad está disponible con el paquete básico.

Con esta base, espero haber transmitido la relevancia de Power Apps y la Power Platform en mi desarrollo profesional, así como su potencial impacto en el futuro de mi carrera.



# Microsoft Dataverse

Vamos a comenzar por este componente ya que según Microsoft los Datos están en el centro de toda su arquitectura.



Fuente: <https://learn.microsoft.com/en-us/training/modules/build-first-model-driven-app-dataverse/introduction>

Microsoft Dataverse es una plataforma de almacenamiento y gestión de datos basada en la nube que forma parte de Microsoft Power Platform. Está diseñada para facilitar la recopilación, almacenamiento, y manipulación de datos. Es básicamente un motor de base de datos relacional.

En el momento que me comentaron en *INETUM* que es una base de datos relacional y que debía crear las tablas como había hecho en durante el grado. Traté de ver que diferencias había con lo estudiado en el grado, y hacerme una idea de que es Dataverse respecto a otros motores de BBDD. Para ello he hecho un cuadro resumen con algunas características comparativas.

## Comparativa con otras BBDD del mercado.

Característica	Microsoft Dataverse	MySQL	MariaDB	PostgreSQL	MongoDB
<b>Tipo de Base de Datos</b>	Relacional/ Pensada para almacenar Datos en la Nube	Relacional	Relacional	Relacional	NoSQL (Documentos)
<b>Modelo de Datos</b>	Tablas con relaciones y seguridad predefinidas	Tablas relacionales	Tablas relacionales	Tablas relacionales	Documentos BSON (similar a JSON)
<b>Facilidad de Uso</b>	Fácil de usar para usuarios sin experiencia en BD	Requiere conocimientos de SQL	Requiere conocimientos de SQL	Requiere conocimientos de SQL	Para desarrolladores web
<b>Licencia</b>	Propietaria (has de comprar una licencia para usarla)	Open Source (GPL)	Open Source (GPL)	Open Source (PostgreSQL License)	Open Source (SSPL)
<b>Donde se usa comúnmente.</b>	Aplicaciones empresariales integradas con Microsoft	Aplicaciones web y empresariales	Aplicaciones web y empresariales	Aplicaciones complejas con consultas avanzadas	Aplicaciones web en tiempo real, análisis de datos

Dataverse está disponible como un servicio independiente para empresas que quieran desarrollar sus propias soluciones, has de conectarte a la nube de Microsoft llamada Azure y tiene capacidades de integración a otros sistemas a través de webhooks. En resumen Dataverse tiene API para que los datos puedan ser consumidos por otros servicios personalizados siempre y cuando pagues por la licencia.

Además de los datos relacionales, Dataverse también admite almacenamiento de archivos y tiene la posibilidad de aplicar detección de duplicados, campos calculados, campos acumulativos, etc.

## TABLAS

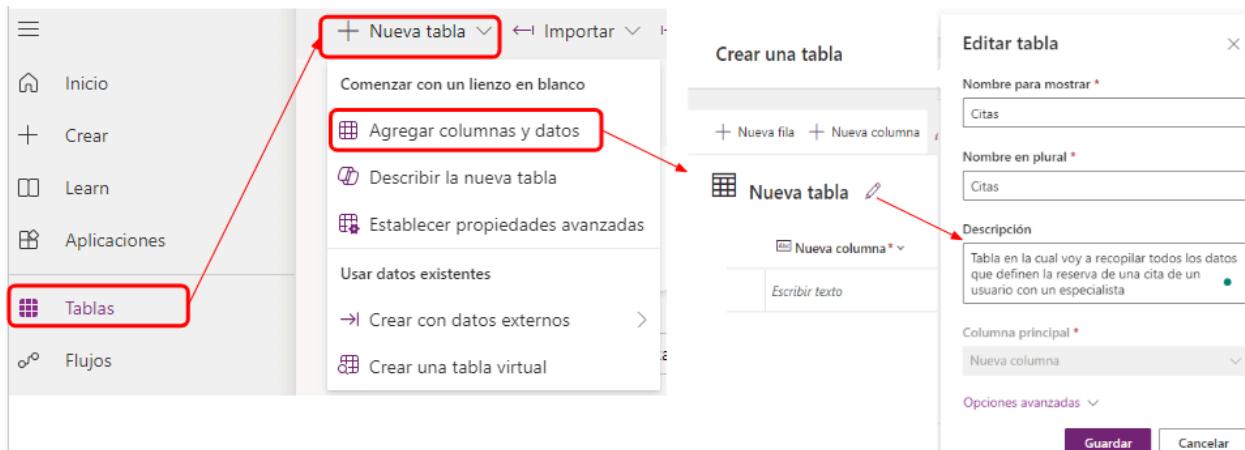
En Dataverse, los datos se almacenan en tablas que tienen una estructura de filas y columnas, parecido a lo que hacen las tablas para almacenar datos dentro de una base de datos de las que hemos visto en grado.

The screenshot shows the Microsoft Dataverse interface for managing tables. On the left, there's a sidebar with navigation links for Power Apps, Tables, Flows, Solutions, and Power Platform. The main area is titled 'Tablas' and contains three buttons: 'Cargar un archivo de Excel', 'Comenzar con un lienzo en blanco', and 'Crear una tabla virtual'. Below these buttons is a filter bar with 'Recomendado', 'Personalizado', and 'Todos' options. The main content area displays a table with columns: 'Nombre', 'Tipo', 'Administrada', 'Personalizable', and 'Etiquetas'. The table lists various standard entities such as Artículo de conocimientos, Buzón, Carta, Cita, Cita periódica, Comentarios, Contacto, Correo electrónico, Cuenta, Datos adjuntos, Dirección, and Divisa. Each entity has its name, type (e.g., knowledgearticle, mailbox, letter), management status, personalizability, and standard tags.

Nombre	Tipo	Administrada	Personalizable	Etiquetas
Artículo de conocimientos	knowledgearticle	Standard	Sí	Estándar
Buzón	mailbox	Standard	Sí	Estándar
Carta	letter	Activity	Sí	Estándar
Cita	appointment	Activity	Sí	Estándar
Cita periódica	recurringappointmentmas...	Activity	Sí	Estándar
Comentarios	feedback	Standard	Sí	Estándar
Contacto	contact	Standard	Sí	Estándar
Correo electrónico	email	Activity	Sí	Estándar
Cuenta	account	Standard	Sí	Estándar
Datos adjuntos	activitymimeattachment	Standard	Sí	Estándar
Dirección	customeraddress	Standard	Sí	Estándar
Divisa	transactioncurrency	Standard	Sí	Estándar

## Creación de una tabla

La interfaz de PowerApps proporciona una manera sencilla de definir las tablas. Desde el propio menú simplemente, seleccionas la opción > Tablas > para crear una tabla desde el principio, seleccionamos la opción de agregar columnas y datos.



## Tipos de Tablas

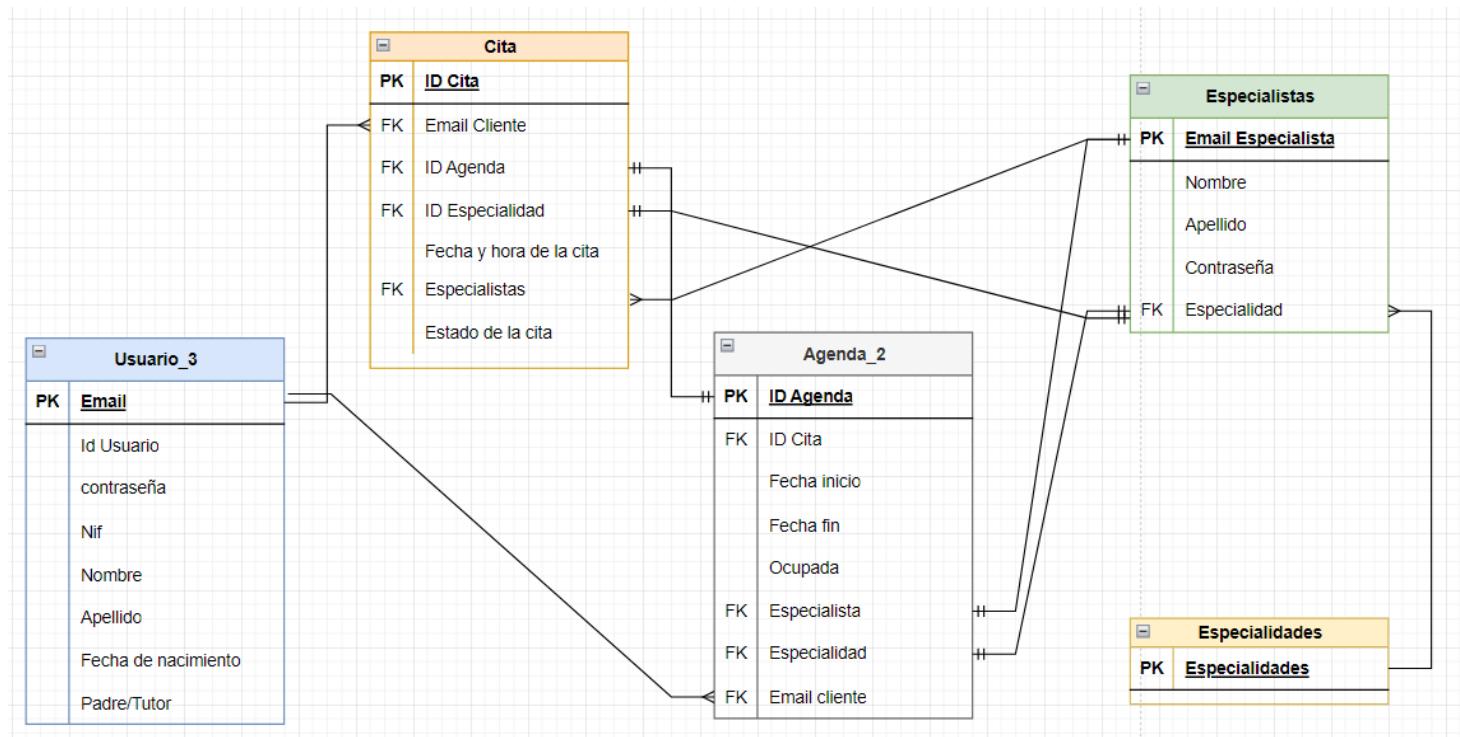
Existen varias opciones a la hora de crear una tabla, por ejemplo:

- **Tablas Virtuales:** que son aquellas que toman datos de una bbdd externa de forma dinámica en tiempo de ejecución. Por ejemplo de Azure SQL y tienen un tratamiento diferente ya que al editarlas hay que tener en cuenta que contienen los datos de un origen externo.
- **Tablas elásticas:** que están enfocadas a conjuntos de datos muy grandes, millones de filas.
- **Tabla de actividad:** son un tipo especial de tabla y son mejores para filas que tienen elementos, que puede incluir un asunto, hora de inicio, hora de finalización, fecha de

vencimiento y duración. Dataverse ya viene con varias tablas de actividades listas para usar, como cita, tarea, correo electrónico y llamada de teléfono.

A nosotros nos han pedido en las prácticas que utilicemos **Tablas estándar**.

## Diseño de las Tablas del Proyecto



## Definir una tabla: Tabla Agenda\_2

Esta tabla contiene el calendario con las fechas y horas disponibles que tienen los especialistas en un determinado periodo de tiempo.

Esta imagen muestra cómo podemos definirla desde el interfaz de power apps.

The screenshot shows the Power Apps Studio interface with the 'Agenda\_2' table selected. The top navigation bar includes 'Gestion\_reserva\_cita2 > Tablas > Agenda\_2'. The main area is divided into several sections:

- Propiedades de la tabla:** Shows basic properties like 'Nombre' (Agenda\_2), 'Columna principal' (ID Agenda), 'Tipo' (Standard), and 'Descripción' (Esta tabla recoge los huecos que tienen los especialistas en su agenda para que los clientes puedan reservar su cita).
- Esquema:** Contains links to 'Columnas', 'Relaciones', and 'Claves'.
- Experiencias de datos:** Contains links to 'Formularios', 'Vistas', 'Gráficos', and 'Paneles'.
- Personalizaciones:** Contains links to 'Reglas de negocio' and 'Comandos'.

Below this is a detailed data view:

ID A...	ID Cita	Fecha inicio	Fecha fin	Ocupada	Especialista	Especialidad	Email cliente	+18 más	Añadir
1108		15/05/2024 12:15	15/05/2024 12:45	No	alejandro@inetum.com	Radiología	sergio.felipe.ext@ine		
1109		15/05/2024 12:45	15/05/2024 13:15	Si	alejandro@inetum.com	Radiología			
1110		15/05/2024 13:15	15/05/2024 13:45	Si	alejandro@inetum.com	Radiología			
1111		15/05/2024 13:45	15/05/2024 14:15	Si	alejandro@inetum.com	Radiología			
1112		15/05/2024 14:15	15/05/2024 14:45	Si	alejandro@inetum.com	Radiología			
1113		15/05/2024 14:45	15/05/2024 15:15	Si	alejandro@inetum.com	Radiología			
1114		14/05/2024 9:15	14/05/2024 9:45	Si	alejandro@inetum.com	Radiología			
1115		14/05/2024 9:45	14/05/2024 10:15	Si	alejandro@inetum.com	Radiología			
1130		14/05/2024 11:15	14/05/2024 11:45	Si	paco.felipe@gmail.com	Medicina_Gral	sergio.felipe.ext@ine		

- ID\_Agenda(Primary Key): Identificador único para cada entrada en la agenda.
- Id Cita (Foreign Key, referencia a la tabla Cita): Vincula la entrada de la agenda con una cita específica.
- Fecha inicio (Fecha y hora): Indica cuándo comienza la cita.
- Fecha fin (Fecha y hora): Indica cuándo termina la cita.

- Ocupada (Texto): Estado de la cita (ocupada: Si o No).
- Especialista (Foreign Key, referencia a la tabla Especialistas): Identifica al especialista encargado de la cita.
- Especialidad (Foreign Key, referencia a la tabla Especialidades): Indica la especialidad del servicio.
- Email cliente (Foreign Key, referencia a la tabla Usuario\_3): Email del cliente que reservó la cita.

La programación de fechas del calendario, especialistas y horarios, las carga en BBDD un administrador, que define una serie de parámetros a tener en cuenta para completar las horas disponibles. Para ello la aplicación cuenta con un formulario que genera dichos datos.

The screenshot shows a user interface for appointment scheduling. At the top left is the logo "inetum." with the tagline "Positive digital flow". Below the logo are two dropdown menus labeled "Seleccione Especialidad" and "Seleccione Especialista", both with "Buscar elementos" placeholder text. To the right of these is a text input field with a blue border and a note: "Establezca la duración de la cita. 30 recomendado". Below this section are two rows of input fields. The first row contains "Fecha de inicio" (26/05/2024) and "Hora inicio" (1). The second row contains "Fecha Fin" (26/05/2024) and "Hora fin" (1). To the right of these rows are two blue buttons: "Generar Citas" and "Guardar en BBDD". At the bottom left is the text "Número de citas Generadas: 0". At the bottom right is the text "Fechas y horas disponibles". A large blue arrow points to the right at the bottom right corner of the form area.

## CRUD (Create, Read, Update, Delete) en Power Apps

Como ejemplo, aquí muestro el extracto de código que permite hacer la inserción en la tabla *Agenda\_2*, tomando como referencia los parámetros obtenidos del formulario.

```
ForAll(
    colFechas;
    Patch(
        Agenda_2;
        Defaults(Agenda_2);
        {
            Especialidad: ComboBox1_1.Selected;
            Especialista: ComboBox1.Selected;
            'Fecha inicio': ThisRecord.dateIni;
            'Fecha fin': ThisRecord.dateEnd;
            Ocupada: "No"
        }
    )
)
```

El lenguaje se llama **Power Fx**, que es un lenguaje utilizado en Power Apps. **Power Fx** se utiliza para escribir expresiones y lógica en aplicaciones creadas con Power Apps, permitiendo a los usuarios manipular datos y definir comportamientos de la aplicación de manera declarativa.

Referencia: <https://learn.microsoft.com/es-es/power-platform/power-fx/overview>

---

## La función Patch

La función **Patch** es esencial en Power Fx para gestionar datos dentro de Power Apps, ofreciendo la forma de realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre fuentes de datos compatibles.

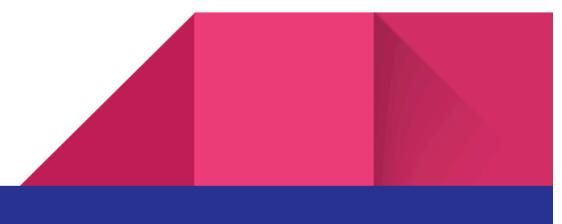
La sintaxis básica de **Patch** es:

*Patch(DataSource, BaseRecord, ChangeRecord1 [, ChangeRecord2, ...])*

- **DataSource** es la fuente de datos en la que deseas realizar la operación.
- **BaseRecord** es el registro que deseas modificar o el valor predeterminado para un nuevo registro.
- **ChangeRecord1, ChangeRecord2**, etc. son uno o más registros que contienen los cambios que deseas aplicar.

Como hemos explicado, con una sola función, **Patch** podemos realizar todas las operaciones (Crear, Leer, Actualizar, excepto Eliminar) .

Como me ha costado entender su funcionamiento al estar acostumbrado a la sintaxis de SQL voy a explicar al menos el código que he proporcionado, ya que me servirá de guía en el futuro para recordar cómo funciona.



## Crear un Nuevo Registro

Recordemos la sintaxis: `Patch(DataSource, BaseRecord, ChangeRecord1 [, ChangeRecord2, ...])`

1. `DataSource` = `Agenda_2` → tabla donde vamos a crear un nuevo registro.
2. `BaseRecord` = `Defaults(Agenda_2)` es el registro que deseas modificar o el valor predeterminado para un nuevo registro.
3. La función `Defaults` → devuelve un registro (vacío) que tiene los campos del `DataSource`, tabla `Agenda_2`, cada uno con su tipo de dato para cada campo.
4. Campo → Especialidad: Valor ; Especialista: Valor. Ocupada:"NO"; etc.

```
Patch(  
    Agenda_2;  
    Defaults(Agenda_2);  
    {  
        Especialidad: ComboBox1_1.Selected;  
        Especialista: ComboBox1.Selected;  
        'Fecha inicio': ThisRecord.dateIni;  
        'Fecha fin': ThisRecord.dateEnd;  
        Ocupada: "No"  
    }  
)
```

## Actualizar

Si queremos actualizar un registro, en lugar de utilizar la función `Defaults` , usaremos la función `LookUp` Por ejemplo: `Lookup(Agenda_2, ID = 123), // Busca el registro existente por ID`

## Eliminar un Registro

Para eliminar un registro, utilizaremos la función `Remove` en lugar de `Patch`:

## Tabla Cita

La tabla **Cita** contiene información detallada sobre cada cita que ha sido reservada por los usuarios. A continuación, se detalla cómo está estructurada esta tabla.

Gestion\_reserva\_cita2 > Tablas > Cita

Propiedades de la tabla			Propiedades	Herramientas	Esquema	Experiencias de datos	Personas
Nombre Cita	Columna principal ID Cita	Descripción Tabla en donde se registran las citas que han sido reservadas por los usuarios			<a href="#">Columnas</a> <a href="#">Relaciones</a> <a href="#">Claves</a>	<a href="#">Formularios</a> <a href="#">Vistas</a> <a href="#">Gráficos</a> <a href="#">Paneles</a>	<a href="#">F</a> <a href="#">C</a>
Tipo Standard	Última modificación hace 12 minutos						

**Cita columnas y datos**

ID Cita ↑	Email clientes	ID_Agenda	ID Especialidad	Fecha y hora de la cita
1011	sergio.felipe.ext@inetum.com	1179	Medicina_Gral	21/05/2024 13:30
1012	sergio.felipe.ext@inetum.com	1204	Medicina_Gral	22/05/2024 16:00
	Seleccionar búsqueda	Seleccionar búsqueda	Seleccionar búsqueda	Escriba o elija una fecha

- **Id cita (Primary Key):Propósito:** Proporciona un identificador único para cada cita, asegurando que cada cita pueda ser referenciada de manera única en el sistema. Este campo es de tipo autonumeración.

- 
- **Email Clientes (Foreign Key, referencia a la tabla Usuario\_3): Propósito:** Vincular la cita con el cliente que la ha reservado.
  - **ID Agenda (Foreign Key, referencia a la tabla Agenda\_2): Propósito:** Vincula la cita con una entrada específica en la tabla Agenda\_2.
  - **ID Especialidad (Foreign Key, referencia a la tabla Especialidades): Propósito:** Indica la especialidad para la cual se ha reservado la cita.
  - **Fecha y hora de la cita (Fecha y hora): Propósito:** Registra la fecha y la hora exacta en que se llevará a cabo la cita.
  - **Especialistas (Foreign Key, referencia a la tabla Especialistas): Propósito:** Indica el email del especialista que atenderá la cita.
  - **Estado de la cita (Opción): Propósito:** Indica el estado actual de la cita.

Cuando un usuario reserva una cita, se crea una nueva entrada en la tabla Cita con los detalles proporcionados (email del cliente, ID de agenda, ID de especialidad, fecha y hora, email del especialista, estado de la cita).

## Tabla Especialistas

La tabla Especialistas contiene la información de los especialistas que ofrecen sus servicios en el sistema. A continuación, se detalla la estructura de la tabla y el propósito de cada columna.

Gestion\_reserva\_cita2 > Tablas > Especialistas

Propiedades de la tabla			Propiedades	Herramientas	Esquema	Experiencias de datos
Nombre	Columna principal	Descripción				
Especialistas	Email Especialista	En esta tabla se registran los especialistas				
Tipo	Última modificación					
Standard	hace 1 minuto					

柱状图 Especialistas columnas y datos 刷新 Actualizar formulario:

Nombre	Apellido	Email Especialista*	Contraseña	Especialidades
Alejandro	Garcia	alejandro@inetum.com	555	Radiologia
Felipe	Jimenez	felipe@inetum.com	666	Cardiologia
Francisco	Felipe	paco.felipe@gmail.com	1234	Medicina_Gral
Marcos	Perez	sergio.felipe.ext@inetum.com	123456	Enfermeria
Pepe	Gonzalez	sergio.felipe.ext@inetum.com	333	Pediatria
Mara	Perez	sergio.felipe.ext@inetum.com	444	Fisioterapia

- 
- **Nombre (Texto):** nombre del especialista. Utilizado para identificar al especialista en el sistema.
  - **Apellido (Texto):** apellido del especialista, información de identificación personal.
  - **Email Especialista (Primary Key):** identificador único para cada especialista basado en su dirección de correo electrónico. Asegura que el especialista único en el sistema.
  - **Contraseña (Texto):** contraseña del especialista para el acceso al sistema.
  - **Especialidad (Foreign Key, referencia a la tabla Especialidades):** Indica la especialidad del especialista.

## Tabla Especialidades

La tabla Especialidades se utiliza para gestionar las distintas especialidades o tipos de servicios que ofrecen los especialistas en el sistema. A continuación, se detalla la estructura de la tabla y el propósito de la columna.

Gestion\_reserva\_cita2 > Tablas > **Especialidades**

Propiedades de la tabla Propiedades Herramientas

Nombre	Columna principal	Descripción
Especialidades	Especialidad	
Tipo	Última modificación	
Standard	hace 1 mes	

**Especialidades columnas y datos**

ABC	Especialidad * ↑ ↓	+24 más	+
	Cardiologia		
	Enfermeria		
	Fisioterapia		
	Medicina_Gral		
	Pediatrica		
	Radiologia		

**Especialidad (Primary Key):Propósito:** Proporciona un identificador único para cada especialidad.

## Tabla Usuarios.

La tabla Usuarios\_3 se utiliza para registrar la información de los usuarios del sistema. A continuación, se detalla la estructura de la tabla y el propósito de cada columna.

Gestion\_reserva\_cita2 > Tablas > Usuario\_3

Propiedades de la tabla			Propiedades	Herramientas	Esquema	Experiencias de datos
Nombre	Columna principal	Descripción				
Usuario_3	Email	Esta tabla se utiliza para registrar los datos de los clientes				
Tipo	Última modificación		Columnas	Formularios		
Standard	hace 19 segundos		Relaciones	Vistas		
			Claves	Gráficos		
				Paneles		

Usuario\_3 columnas y datos

Id Usuario	Email*	Contraseña*	Nombre	Apellido
1001	pepe.felipe@gmail.com	678	Pepe	Garcia
1000	sergio.felipe.ext@inetum.com	123	Sergio	Felipe

Actualizar formulario

- **ID Usuario (Autonumeración):**identificador único y secuencial para cada usuario, cada usuario debe único en el sistema.
- **Email (Primary Key):**identificador único basado en la dirección de correo electrónico del usuario. Este campo se utiliza para el inicio de sesión y para enviar notificaciones y recordatorios relacionados con las citas.
- **Contraseña (Texto):** contraseña del usuario para el acceso al sistema.
- **Nombre (Texto):** nombre del usuario.

- **Apellido (Texto):Propósito:** apellido del usuario, información de identificación personal.
- **Nif (Texto):** Nif del usuario, información de identificación personal.
- **Padre/tutor (Opción - Sí/No):** Indica si el usuario es un parent o tutor.
- **Fecha de nacimiento (Fecha):** fecha de nacimiento del usuario. información de identificación personal.

## Siguiente paso tras el modelado de la BBDD.

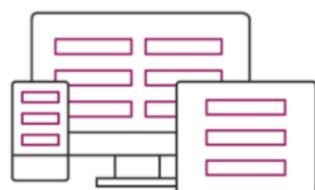
Una vez que hemos creado las BBDD con sus diferentes tablas, ya podemos comenzar a diseñar las pantallas de la aplicación. Conectaremos los datos a nuestra app y comenzaremos a diseñar las pantallas y el comportamiento de sus elementos.



Connect to data & systems you're already using *easily*



Create apps, forms, and workflows *without writing code*

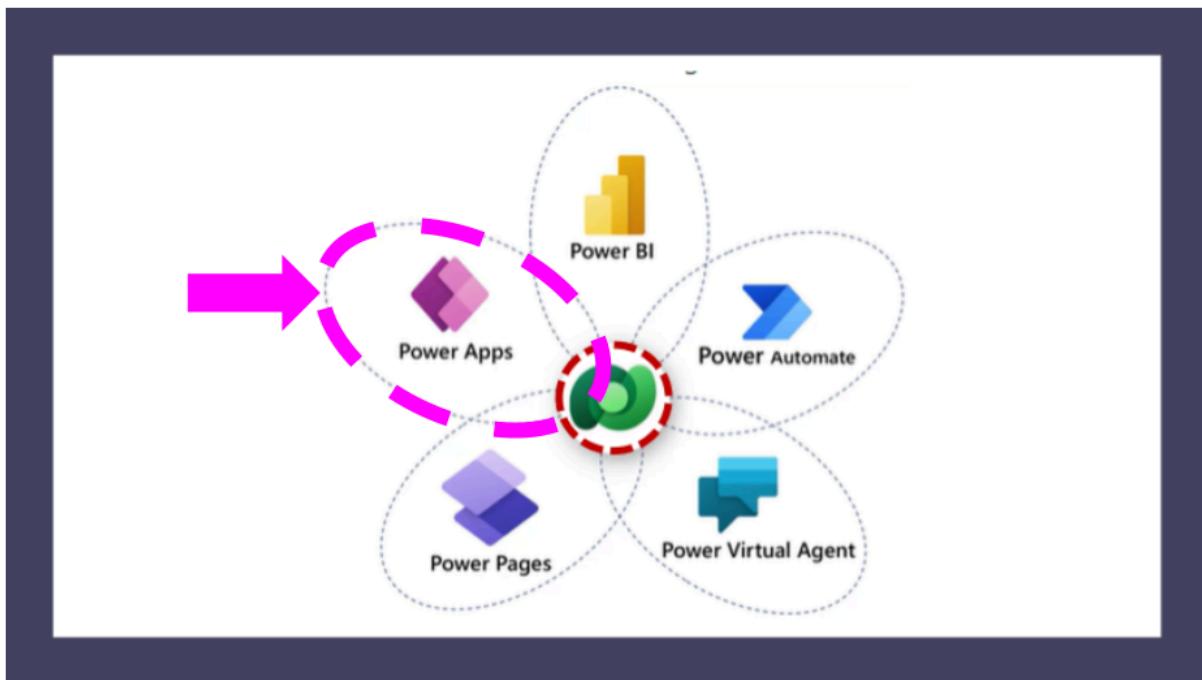


Use apps *on any device* – both web and mobile

Fuente: <https://learn.microsoft.com/en-us/training/modules/get-started-with-powerapps/1-powerapps-introduction>

## POWER APPS

Una vez que he explicado Dataverse como soporte de datos de la aplicación que he desarrollado, ahora me voy a centrar en el 2º componente



Microsoft hace una diferenciación entre tipos de apps que se pueden crear. aplicaciones basadas en modelos de datos que llaman (Model Driven) y aplicaciones de lienzo que llaman (Canvas). Aunque existe esta diferenciación en mi opinión y así lo he hecho en mi App de citas médicas, se pueden combinar ambas.

# Model Driven Apps

Las aplicaciones basadas en modelos **empiezan con un modelo de datos que ya existe** en Dataverse. Este tipo de app **genera automáticamente UI** (user interface) y se pueden emplear en diferentes dispositivos ya que **ofrece un diseño responsivo** que funciona en navegadores y dispositivos móviles. **No se puede construir** un app basada en modelos **sin un modelo de datos previamente almacenado en Microsoft Dataverse**.

Cuando creamos una nueva columna en nuestra BBDD, podemos incluirla en Formularios y Vistas de forma automática. Aquí muestro una hoja de referencia con todos los tipos de columnas estándar y la UI automática que genera,

The screenshot displays a comprehensive cheatsheet for Power Apps fields, organized into several sections:

- Single line of text:** Includes Text, Text area, and Rich text examples.
- Multiple lines of text:** Includes Text and Rich text examples.
- Choice:** Includes Choice, Multiple choices, Yes/no, Checkbox, Option set, and Toggle examples.
- Date and time:** Includes Date only, Date and time, and Time without Date examples.
- Lookup:** Includes Lookup, Customer, and Polymorphic lookup examples.
- File:** Includes File and Image examples.
- Whole number:** Includes None, Duration, Language code, Time zone, Number input, and Star rating examples.
- Decimal & Float:** Includes Decimal and Float examples.
- Currency:** Includes Currency value and Currency examples.

Each section shows a preview of the field type in a form, along with its properties and a visual representation of the generated user interface.

Referencia: <https://i0.wp.com/danikahil.com/wp-content/uploads/2022/08/Power-Apps-Model-Driven-Apps-Fields-Cheatsheet-1-scaled.jpg?ssl=1>

---

### **Beneficios de esta forma de desarrollar:**

**Proceso de construcción rápido:** la construcción de aplicaciones es rápida una vez que se han creado el modelo de datos y las relaciones. Recordar que no se puede crear una app de este estilo si no existe en Dataverse el modelo de datos.

**Interfaz de usuario estandar:** Las aplicaciones tienen una interfaz de usuario similar en varios dispositivos, desde el escritorio hasta el móvil. Los componentes para visualizar los datos o para insertarlos en Dataverse ya están disponibles.

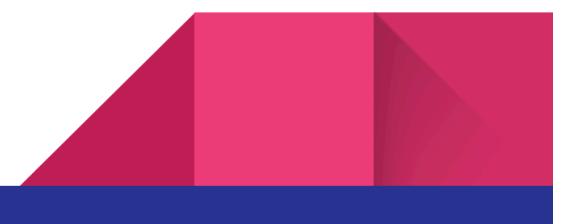
## **Canvas Apps (Aplicaciones de lienzo)**

Canvas Apps permite tener un control total sobre el diseño de las pantallas que vas a mostrar a los usuarios finales. La ventaja de diseñar una App del tipo canvas, es que puedes arrastrar y soltar elementos de la interfaz de usuario como si fuese un powerpoint y personalizar el aspecto y comportamiento de cada componente. Este es el **tipo de Aplicación que me han dicho en INETUM que debo seleccionar**.

### **Beneficios de esta forma de desarrollar:**

**Control Total sobre el diseño de la interfaz:** Parece que trabajas con Powerpoint. Es lo que llaman en Microsoft WYSIWYG (del inglés "What You See Is What You Get", que significa "lo que se ve es lo que se obtiene"). Permite modificar la interfaz del usuario sin código.

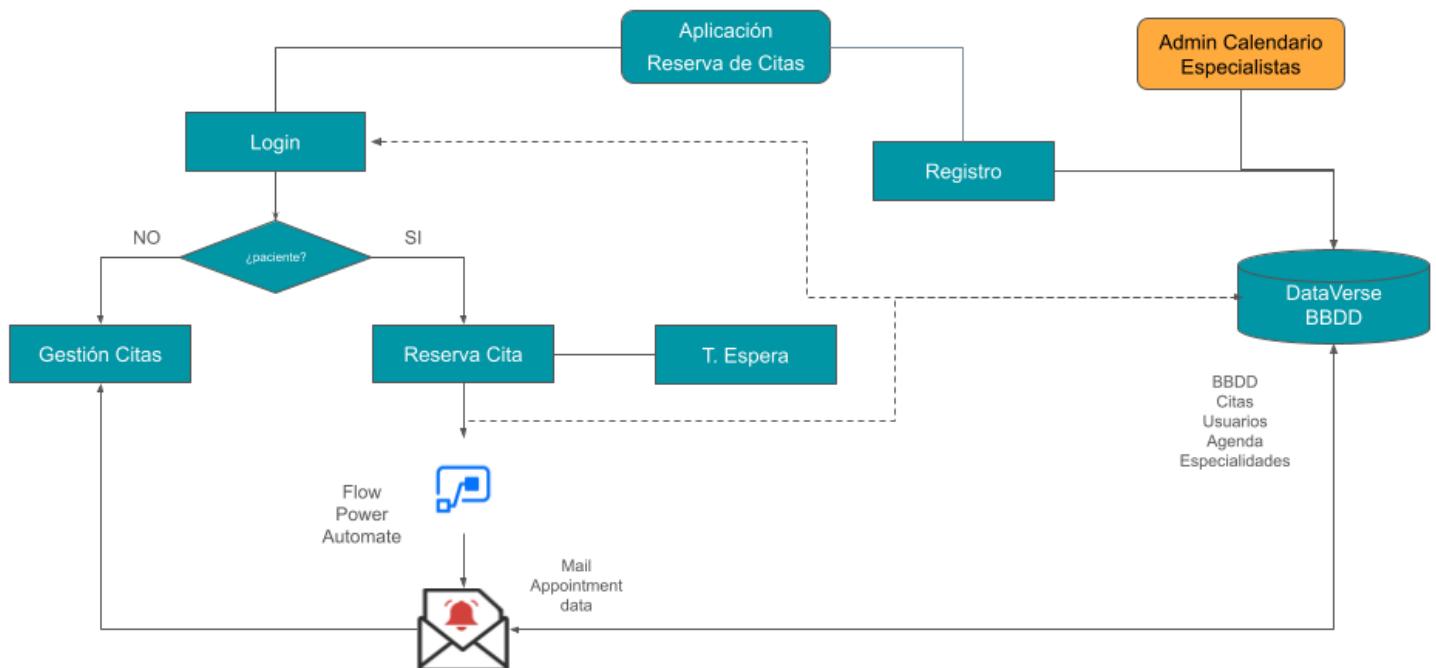
Solo te preocupas de codificar el comportamiento de los componentes que insertas en la página.



# Aplicación de Reserva de Citas.

A modo de vista aérea, la siguiente figura, describe de manera básica los componentes principales de la aplicación y su flujo.

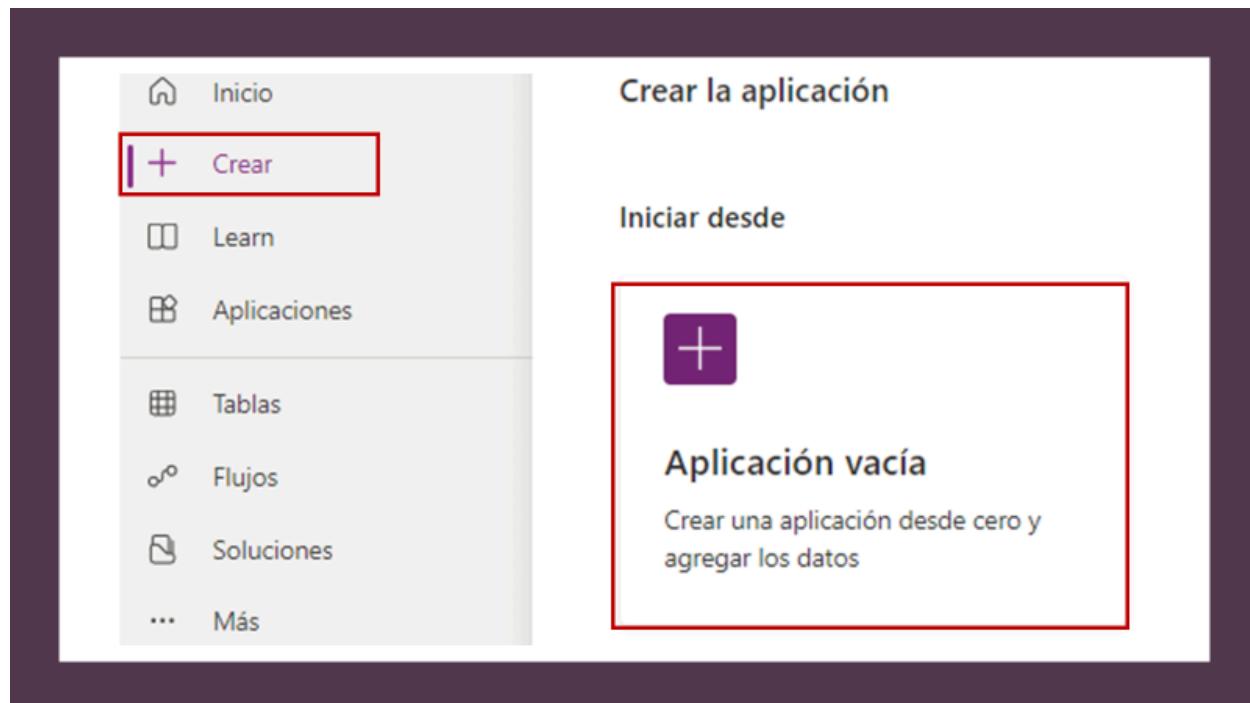
## ARQUITECTURA BÁSICA - APP RESERVA DE CITAS



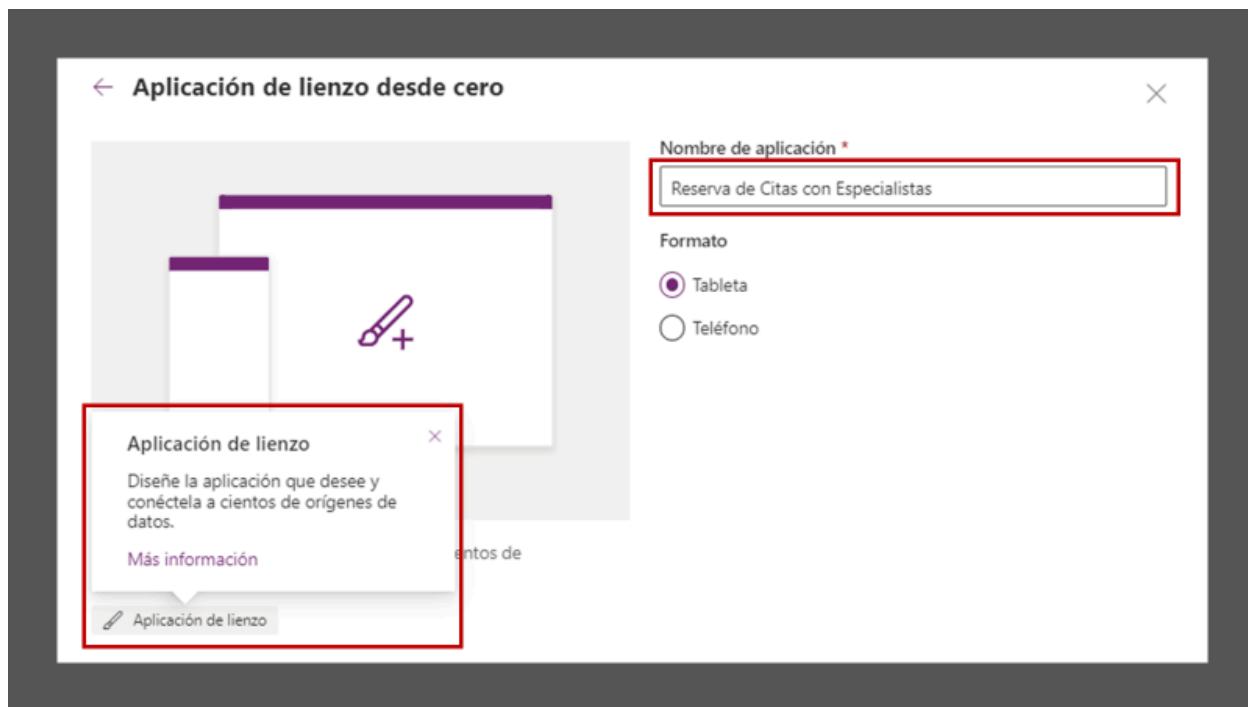
## Crear la Aplicación

Tal y como he explicado anteriormente en INETUM me han pedido que cree la aplicación desde cero, en formato Canvas o Lienzo y en modo Tableta.

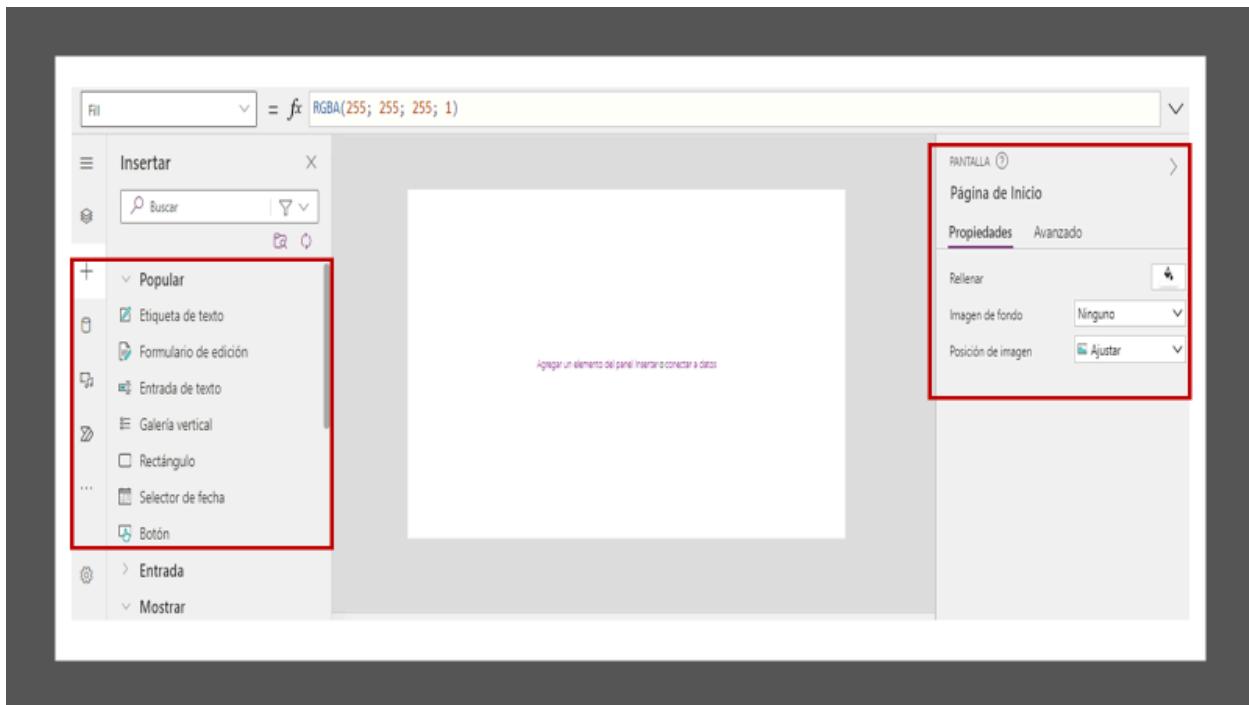
Para ello desde el panel de Power Apps seleccionamos Crear>Aplicacion



A continuación definimos el nombre de la aplicación y en que formato vamos a construirla. En INETUM me han solicitado que la haga en formato Tableta.

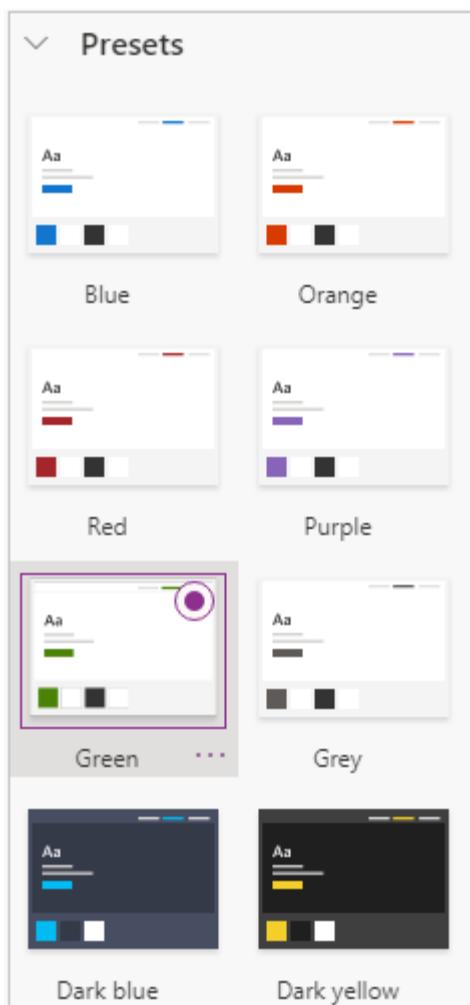


Una vez creada la aplicación ya podemos comenzar a diseñar seleccionando los elementos que deseamos incluir, conectarlo con los datos y modificar su comportamiento.



## Look & Feel

Mi tutor en **INETUM** comenta, que existen Temas Preconstruidos, de gran ayuda si deseas utilizarlos ya que tienen una apariencia bastante buena.



Sin embargo asegura que cuando el mismo realiza algun desarrollo, **todas la empresas le piden que personalize la aplicación**, básicamente para diferenciarse del resto. Le facilitann la paleta de colores de esa marca, los logos, etc.

Asi pues, me solicita que usemos los colores corporativos y Logo de **INETUM** para el desarrollo de la aplicación.

Me ha llamado poderosamente la atención que Power apps permite personalizar los temas e incluso **cargar tu propio CSS**. Cada vez soy más consciente de que CSS es uno de los lenguajes más importantes que se utilizan para ordenar y presentar los contenidos de una página de forma atractiva.

Fuente: <https://learn.microsoft.com/es-es/power-apps/maker/portals/theme-overview>

## Página de inicio



Esta es la página de inicio, en ella hay dos botones:

- Si el usuario no está registrado podrá navegar a la página de registro, donde habrá un formulario para registrarse.
- Si el usuario ya está registrado, el botón navega a la página de inicio de sesión, donde puede acceder con su cuenta de acceso.

Para redirigir al usuario a las páginas correspondientes al hacer clic en los botones de "Acceder" o "Registrarse", he utilizado el siguiente código:

```
Navigate(Login; ScreenTransition.Cover)
```

Este código se encarga de decirle a la aplicación que debe navegar a la pantalla de inicio de sesión, **es similar a poner un enlace href a otra página, tal y como hemos visto en clase**. Su funcionamiento es:

- **Navigate:** Esta es una función que se usa para cambiar de una pantalla a otra dentro de una aplicación.
- **Login:** Este es el argumento de la función Navigate ,al pasar este argumento, le estoy indicando a la función Navigate que debe llevar al usuario a la pantalla de inicio de sesión que yo he llamado “Login”.
- **ScreenTransition.Cover:** Con este argumento lo que estoy haciendo es darle un efecto de transición para cuando dirija a la pantalla de inicio de sesión.

## Página de Registro

The screenshot shows a registration form for Inetum. At the top, there is a logo with the text "inetum." and "Positive digital flow". Below the logo, there are four input fields: "Nombre" (Name) and "NIF" (Tax Identification Number), both with white input boxes. Next are "Apellidos" (Last Name) and "Fecha\_Nacimiento" (Date of Birth), where "28/05/2024" is entered in a date picker field. Below these are "Email" and a checkbox labeled "Marque si es Ud. Padre/Tutor" (Check if you are Parent/Guardian). A password field is labeled "Introduzca contraseña deseada" (Enter desired password) with a "Mostrar" (Show) button next to it. At the bottom is a large blue "REGISTRARME" (Register) button.

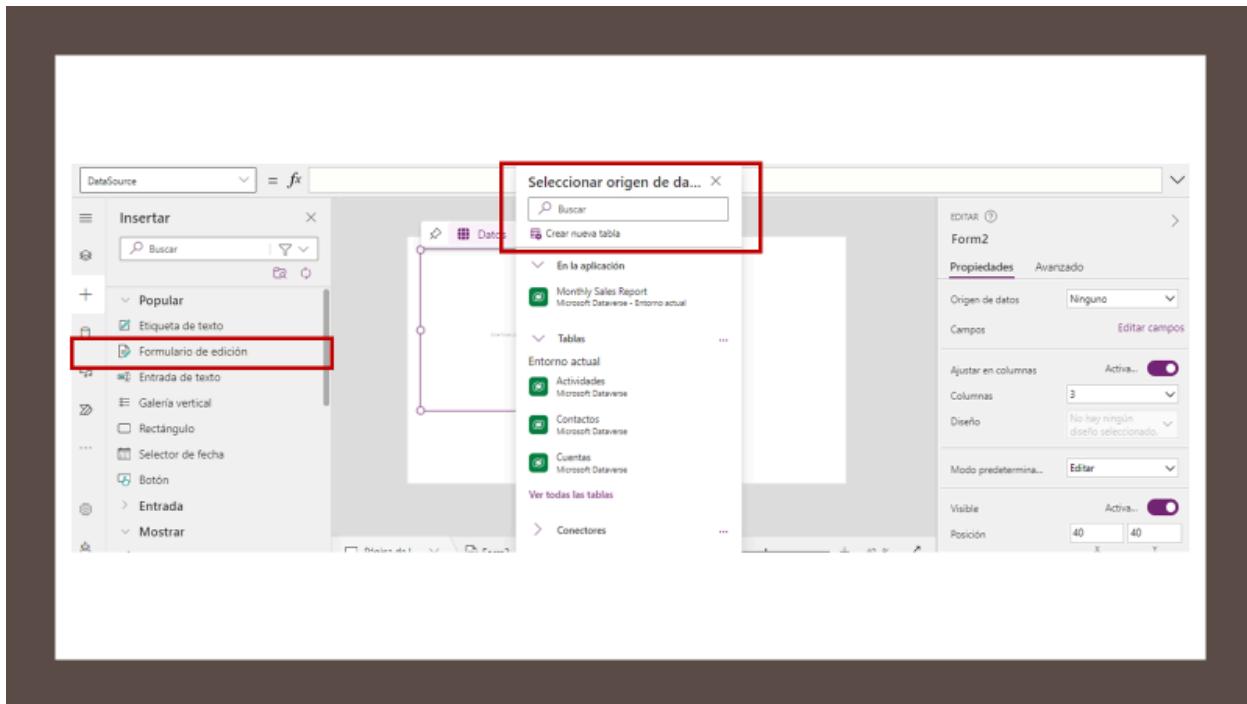
Si un usuario no está registrado y hace clic en el botón de la página de inicio para registrarse llega a esta nueva página.

He creado un formulario, donde el usuario debe completar los datos para crear su cuenta.

Una vez el usuario haya llenado los datos este le dará al botón de "Registrarse" y los datos se grabarán en la base de datos de dataverse y pasará a la pantalla de inicio de sesión.

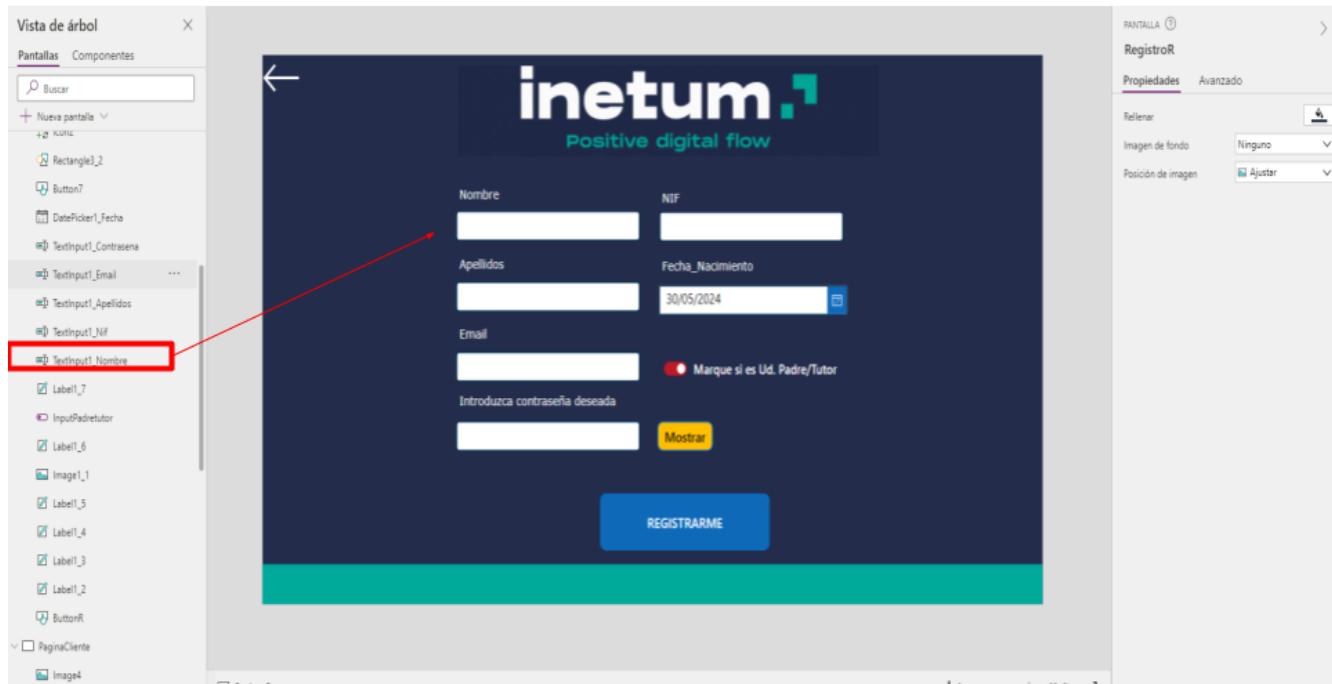
## ¿Cómo se crea un formulario en power apps?

Desde el panel donde hemos creado nuestra aplicación, seleccionamos el elemento, en este caso formulario de edición y lo conectamos al origen de datos, es decir a la tabla donde vamos a insertar los registros.



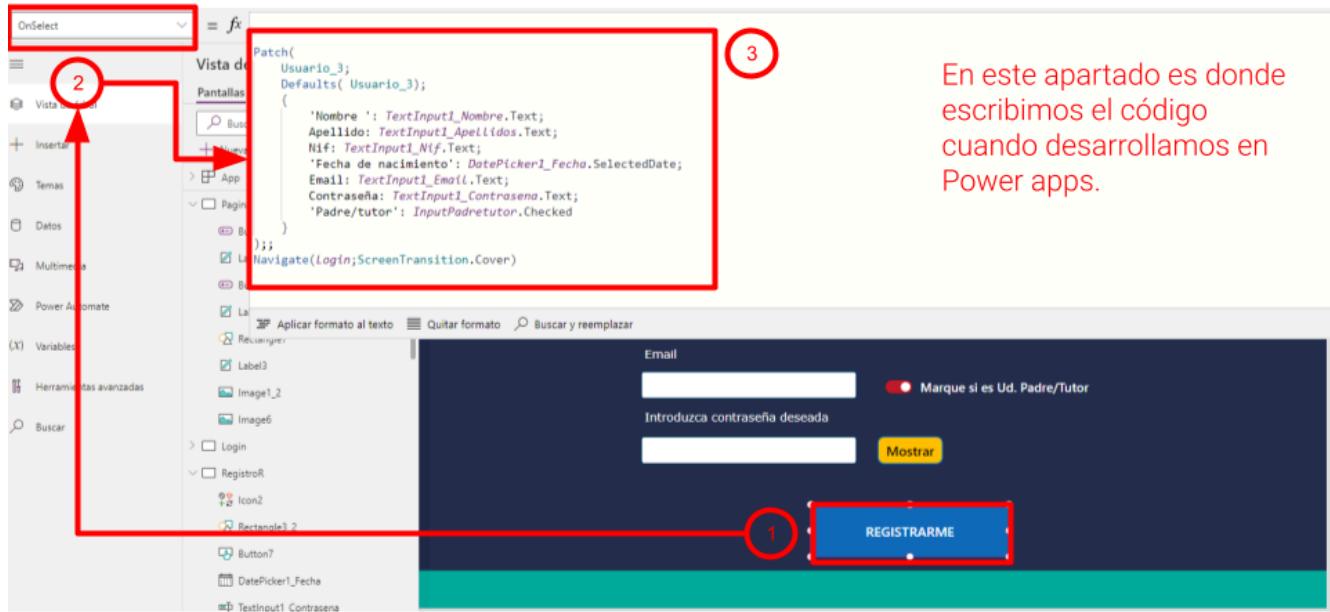
Posteriormente vamos a personalizar el comportamiento de este formulario con un ejemplo.

Si se desea tener un control mayor sobre cada uno de los campos del formulario, se pueden insertar elementos de forma individual. Esto es lo que he hecho yo por recomendación y de mis tutores de INETUM.



Ahora veamos con un ejemplo como funciona este formulario cuando un cliente rellena los datos y hace clic en el botón “Registrarse”.

Primero debemos modificar el comportamiento del botón “registrarse” escribiendo el código correspondiente. Para ello seleccionamos el botón, escribimos el código tal y como muestra la siguiente imagen.

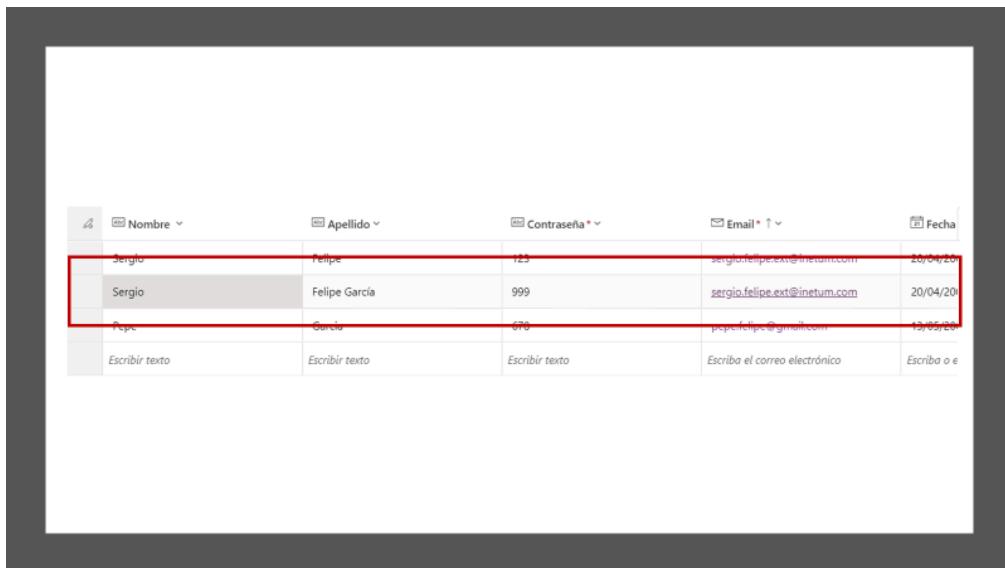




The screenshot shows a registration form for 'inetum Positive digital flow'. The form fields include:

- Nombre: Sergio
- NIF: 11111111D
- Apellidos: Felipe García
- Fecha\_Nacimiento: 20/04/2004
- Email: sergio.felipe.ext@inetum.com
- Marque si es Ud. Padre/Tutor
- Introduzca contraseña deseada: 999
- Mostrar button
- REGISTRARME button

Para comprobar que se realiza el registro, abrimos nuestra base de datos de dataverse y nos dirigimos a la tabla correspondiente. Ahí, podemos ver que el registro se ha hecho exitosamente.



The screenshot shows a Dataverse table with the following data:

Nombre	Apellido	Contraseña	Email	Fecha
Sergio	Felipe	123	sergio.felipe.ext@inetum.com	20/04/20
Sergio	Felipe García	999	sergio.felipe.ext@inetum.com	20/04/20
Pepe	García	678	pepe.felipe.g@gmail.com	13/05/20

El Código para el registro en la bbdd de los campos del formulario es el siguiente

```
Patch(  
    Usuario_3;  
    Defaults( Usuario_3);  
{  
    'Nombre ': TextInput1_Nombre.Text;  
    Apellido: TextInput1_Apellidos.Text;  
    Nif: TextInput1_Nif.Text;  
    'Fecha de nacimiento': DatePicker1_Fecha.SelectedDate;  
    Email: TextInput1_Email.Text;  
    Contraseña: TextInput1_Contrasena.Text;  
    'Padre/tutor': InputPadretutor.Checked  
}  
);;  
Navigate(Login;ScreenTransition.Cover)
```

Para hacer el registro lo que utilizo es la función **Patch** descrita en un apartado anterior, para recordar su sintaxis y funcionamiento [vaya aquí](#).

## Página de inicio de sesión

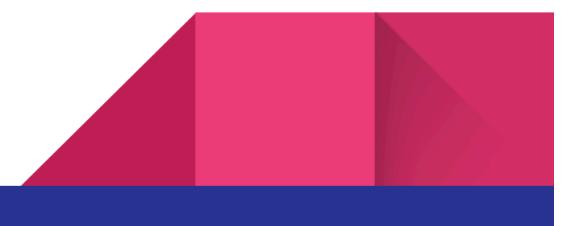


Clásica página de inicio de sesión donde el usuario debe introducir su email y la contraseña.  
El comportamiento de los elementos de la página son los siguientes:

### Botón Mostrar contraseña.

Si el usuario presiona el botón, le permite visualizar la contraseña introducida en claro.

```
If (Button7.Pressed; TextMode.SingleLine; TextMode.Password)
```



## Botón Iniciar sesión.

```
/* Declaro una variable para almacenar el correo electrónico */
Set(
    EmailUsuario;
    TextInputEmail.Text
);
/* Verificar primero si hay alguna entrada del formulario en blanco*/
If(
    IsBlank(TextInputEmail.Text) || IsBlank(TextInputContraseña.Text);
/* Si alguna entrada está en blanco, mostrar un mensaje de error */
    Notify("Por favor, completa todos los campos.");
    /* En caso contrario, hago una búsqueda en la tabla de usuarios. */
    If(
        !IsBlank(
            LookUp(
                Usuario_3;
                Email = TextInputEmail.Text && Contraseña =
TextInputContraseña.Text
            )
        );
        /* Si existe, navegar a la página de clientes */
        Navigate(PaginaCliente);
        /* Si no existe, compruebo si es un Especialista */
        If(
            !IsBlank(
                LookUp(
                    Especialistas;
                    'Email Especialista' = TextInputEmail.Text &&
Contraseña = TextInputContraseña.Text
                )
            );
            /* Si existe, navegar a la página de especialistas */
            Navigate(PaginaEspecialista);
            /* Si no se encuentra una coincidencia en ninguna tabla,
mostrar un mensaje de error */
            Notify("Credenciales incorrectas. Por favor, inténtalo de
nuevo.")
        )
    )
)
```

---

## La función Lookup

La función `Lookup` en Power Fx se usa para buscar datos dentro de Power Apps, sobre fuentes de datos compatibles.

La sintaxis básica de `Lookup` es:

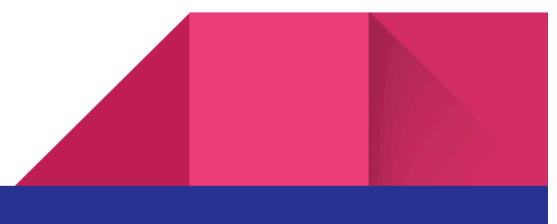
*LookUp(Source, Condition [, ReductionFormula])*

- **Source:** La tabla o colección en la que deseas buscar.
- **Condition:** La condición que debe cumplir el registro buscado.
- **ReductionFormula** (opcional): Una fórmula que se aplica al registro encontrado para devolver un solo valor en lugar del registro completo.

Explicamos su funcionamiento, con el código que he mostrado anteriormente para el botón iniciar sesión:

```
LookUp( Especialistas; 'Email Especialista' = TextInputEmail.Text &&
Contraseña = TextInputContraseña.Text)
```

Realizamos una búsqueda en la tabla `Especialistas` para encontrar la fila cuyo contenido de la columna '`Email Especialista`' conocida con el texto que hemos recogido del formulario `TextInputEmail.Text` y que la columna contraseña coincida con el contenido del campo del formulario `TextInputContraseña.Text`.



## Página de reserva de cita

The screenshot shows a web page for Inetum medical appointments. At the top left is the Inetum logo with the tagline "Positive digital flow". To the right, the heading "Horarios y Especialistas Disponibles" and the sub-instruction "Seleccione el que convenga" are displayed. On the left, there's a sidebar with a house icon and sections for selecting the specialty ("Seleccione la Especialidad que desea") and date ("Seleccione a partir de que fecha desea concertar su Cita con el especialista"). The date selected is "16/05/2024". Below this, a section titled "Consulte cuánto tiempo tiene que esperar a ser atendido" is visible. The main content area lists appointment slots in a grid format. Each slot includes the specialist's email (Espe: sergio.felipe.ext@inetum.com), start time (Inicio: 16/05/2024 9:00 or 17/05/2024 10:00/11:00/12:00/13:00), end time (Fin cita: 16/05/2024 9:30 or 10:30/11:30/12:30/13:30), and a blue "Reservar" button.

Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com
Inicio:	16/05/2024 9:00	Inicio:	17/05/2024 9:00	Inicio:	17/05/2024 9:30
Fin cita:	16/05/2024 9:30	Fin cita:	17/05/2024 9:30	Fin cita:	17/05/2024 10:00
<button>Reservar</button>					
Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com
Inicio:	17/05/2024 10:00	Inicio:	17/05/2024 10:30	Inicio:	17/05/2024 11:00
Fin cita:	17/05/2024 10:30	Fin cita:	17/05/2024 11:00	Fin cita:	17/05/2024 11:30
<button>Reservar</button>					
Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com
Inicio:	17/05/2024 11:30	Inicio:	17/05/2024 12:00	Inicio:	24/05/2024 9:00
Fin cita:	17/05/2024 12:00	Fin cita:	17/05/2024 12:30	Fin cita:	24/05/2024 9:30
<button>Reservar</button>					
Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com
Inicio:	24/05/2024 9:30	Inicio:	24/05/2024 10:00	Inicio:	24/05/2024 10:30
Fin cita:	24/05/2024 10:00	Fin cita:	24/05/2024 10:30	Fin cita:	24/05/2024 11:00
<button>Reservar</button>					
Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com	Espe:	sergio.felipe.ext@inetum.com
Inicio:	24/05/2024 11:00	Inicio:	24/05/2024 11:30	Inicio:	24/05/2024 12:00
Fin cita:	24/05/2024 11:30	Fin cita:	24/05/2024 12:00	Fin cita:	24/05/2024 12:30
<button>Reservar</button>					

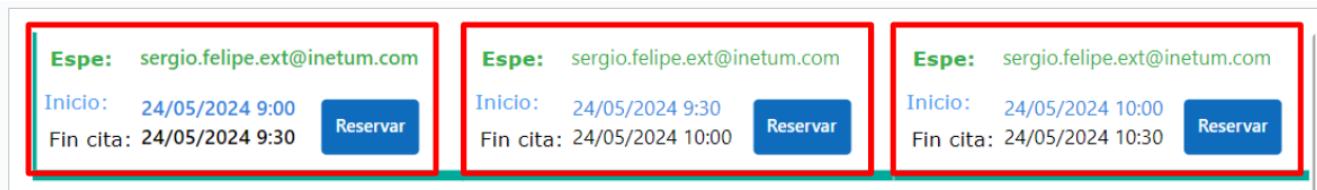
Para no extenderme hablando de los campos de un formulario, en esta página cuya finalidad es ofrecer al usuario final la posibilidad de reservar una cita médica, eligiendo la especialidad y la fecha de la cita. Hablaré únicamente de los componentes que entiendo son distintos y de cómo he modificado su comportamiento.

## Galería de citas

En Power Apps, una Galería es un elemento que sirve para mostrar una colección de elementos. Es útil para presentar listas de datos que provienen de bases de datos.

Este elemento permite mostrar una serie de registros de datos. Cada registro se muestra como una "tarjeta" dentro de la galería.

### Tarjetas de la Galería



En la figura podemos ver cada una de las tarjetas que son los datos contenidos en cada registro de la base de datos que tenemos. Una galería puede conectar varias fuentes diferentes.

Cada tarjeta o registro de la galería puede contener varios controles como etiquetas, imágenes, botones, etc.

En nuestro caso hemos incluido las etiquetas "Inicio:" y "Fin Cita" y el botón "Reservar" para que cuando un usuario seleccione el horario que más convenga, al hacer clic en el botón quede la cita registrada.

Este elemento también tiene la capacidad de filtrado y ordenación de registros mostrados en la galería, mejorando la usabilidad y personalización de los datos mostrados.

## Función Filter

Este código lo que hace es mostrar en la galería las citas que estén disponibles de la especialidad que hayan elegido y en ese día o después de la fecha seleccionada.

```
Filter(  
    Agenda_2;  
    Especialidad.Especialidad = Dropdown3.Selected.Especialidad &&  
    Ocupada = "No" &&  
    'Fecha inicio' >= DatePicker1.SelectedDate  
)
```

La función **Filter** selecciona de la Tabla Agenda\_2 las filas que cumplen con las condiciones especificadas en el formulario.

- **Especialidad.Especialidad = Dropdown3.Selected.Especialidad:** Filtra las citas para que solo se muestran aquellas cuya especialidad coincide con la seleccionada por el usuario en **Dropdown3** (es la lista desplegable ).
- **Ocupada = "No":** Filtra las citas para que solo se muestren aquellas que no están ocupadas.
- **'Fecha inicio' >= DatePicker1.SelectedDate:** Filtra las citas para que solo se muestran aquellas cuya fecha de inicio es igual o posterior a la fecha seleccionada en **DatePicker1**.

## Organización de la Galería

Otro aspecto que me gustaría mencionar es que este elemento permite **organizar las tarjetas de diferentes formas** para que se muestren de forma ordenada. Es un aspecto visual. En mi caso, al disponer de muchos registros cada uno de los días, se pueden realizar reservas de citas en diferentes horarios.

He organizado el elemento para que se vean en 3 columnas y ordenados por horas de izquierda a derecha.

<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 9:00 Reservar Fin cita: 24/05/2024 9:30	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 9:30 Reservar Fin cita: 24/05/2024 10:00	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 10:00 Reservar Fin cita: 24/05/2024 10:30
<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 10:30 Reservar Fin cita: 24/05/2024 11:00	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 11:00 Reservar Fin cita: 24/05/2024 11:30	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 11:30 Reservar Fin cita: 24/05/2024 12:00
<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 12:00 Reservar Fin cita: 24/05/2024 12:30	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 12:30 Reservar Fin cita: 24/05/2024 13:00	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 13:00 Reservar Fin cita: 24/05/2024 13:30
<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 13:30 Reservar Fin cita: 24/05/2024 14:00	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 14:00 Reservar Fin cita: 24/05/2024 14:30	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 14:30 Reservar Fin cita: 24/05/2024 15:00
<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 15:00 Reservar Fin cita: 24/05/2024 15:30	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 15:30 Reservar Fin cita: 24/05/2024 16:00	<b>Espe:</b> sergio.felipe.ext@inetum.com Inicio: 24/05/2024 9:00 Reservar Fin cita: 24/05/2024 9:30

Comportamiento del Botón de Reserva en una Galería.

```
Set(newCita;
Patch(
    Cita;
    Defaults(Cita);
    {
        'Email clientes': LookUp(Usuario_3; EmailUsuario = EmailUsuario);
        Cliente: EmailUsuario;
    // Tomo el valor del ID del usuario actual
        'Fecha y hora de la cita ': Gallery3.Selected.'Fecha inicio';
    // Tomo la fecha de inicio del elemento seleccionado por el usuario en
    Gallery3
        Especialistas: Gallery3.Selected.Especialista;
        'ID Especialidad': Gallery3.Selected.Especialidad;
        'Estado de la cita':'Estado de la cita (Cita)'.Pendiente;

        ID_Agenda: LookUp(Agenda_2; 'ID Agenda' = Gallery3.Selected.'ID
        Agenda')
    // Tomo el registro completo de la cita y lo asignar a 'ID_Agenda'
    }
));
// Tengo que cambiar el estado del campo ocupación de la cita para que no
pueda volver a ser seleccionable por otro usuario.
Patch(
    Agenda_2;
    LookUp(Agenda_2; 'ID Agenda' = Gallery3.Selected.'ID Agenda');
    {
        Ocupada: "Si";
        'Email cliente': LookUp(Usuario_3; EmailUsuario = EmailUsuario);
        'ID Cita':LookUp(Cita; 'ID Cita' = newCita.'ID Cita')
    }
)
```

---

Estas son las operaciones principales relacionadas con la gestión de citas.

1. **Primero, crea una nueva cita y almacena su información en la tabla `Cita`**. Para ello, utiliza la función `Patch` explicado anteriormente, y actualiza varios campos, como el email del cliente, la fecha y hora de la cita, el especialista, la especialidad, el estado de la cita (**marcándose como pendiente**), y el ID de la agenda correspondiente a la cita seleccionada en `Gallery3`. El resultado de esta operación se guarda en la variable `newCita`.
2. **Actualizo el estado de ocupación** de la cita seleccionada en la tabla `Agenda\_2`. Buscando el registro de la agenda que coincide con el ID de la cita seleccionada y cambio el estado de ocupación a "Sí".
3. Actualizo el email del cliente y el ID de la nueva cita creada (`newCita.ID Cita`) con el registro de la agenda correspondiente. Y la relacionocon el cliente y la cita recién creada.

## Página de tiempo de espera

The screenshot displays a user interface for managing appointments. At the top, there's a teal header bar with the title "Sus Citas con Especialistas". Below this, the main content area has a dark blue background. On the left, there's a back arrow icon. On the right, the "inetum" logo is visible. The central part of the screen lists two appointment slots:

Especialista:	sergio.felipe.ext@inetum.	Especialista:	sergio.felipe.ext@inetum
Especialidad:	Enfermería	Especialidad:	Enfermería
Fecha cita:	01/06/2024 10:00	Fecha cita:	01/06/2024 10:30

Below the appointment details, a message indicates a waiting time: "Tiempo de espera: 4 minutos".

Esta página tiene como finalidad ofrecer al usuario final visualizar las citas que ha reservado y ver el tiempo que tiene que esperar para ser atendido.

Primeramente esta página obtiene las citas reservadas por el usuario con sus especialistas:

The screenshot shows a mobile application interface titled "Sus Citas con Especialistas". It displays two appointment cards. Each card contains the following information:

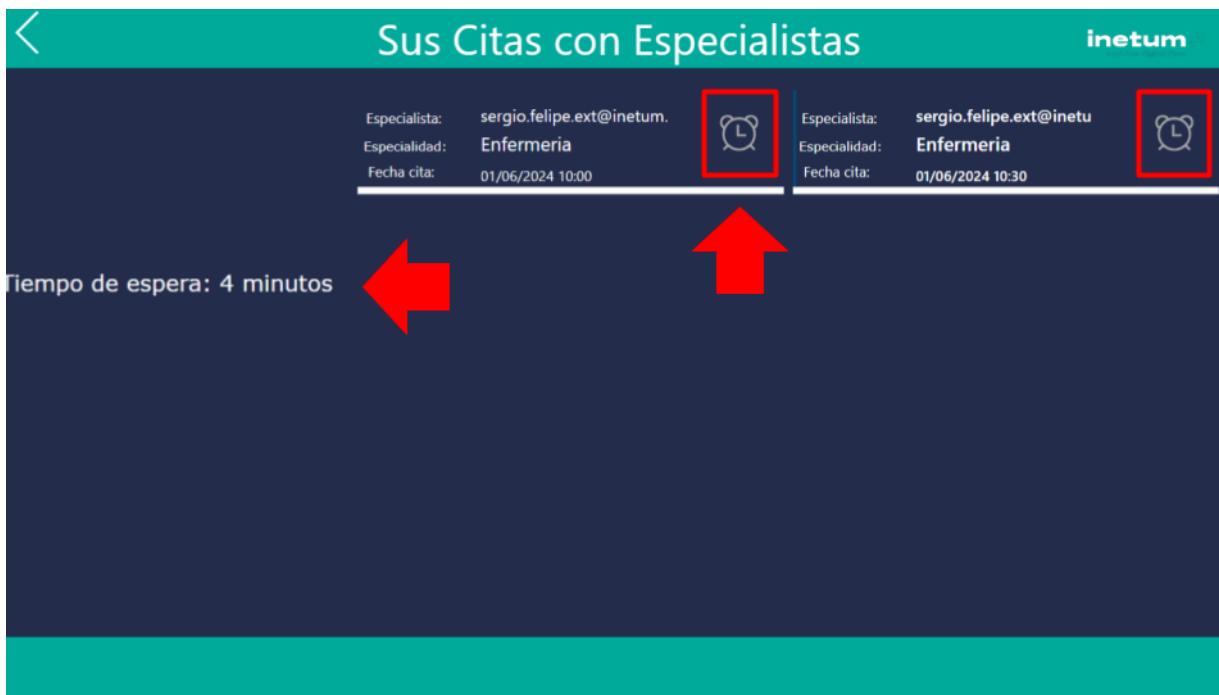
Especialista:	sergio.felipe.ext@inetum.
Especialidad:	Enfermería
Fecha cita:	01/06/2024 10:00

Two large red arrows point upwards from the text "Tiempo de espera:" to the top of each appointment card, highlighting the data being retrieved.

Esta información la obtenemos usando la función `Filter`, de la tabla Cita donde el cliente es el usuario que ha iniciado sesión en la aplicación., La función `Filter` fue explicada anteriormente.

```
Filter(  
    Cita;  
    Cliente=EmailUsuario  
)
```

El usuario, cuando asista a la consulta, podrá visualizar el tiempo que debe esperar para que comience su consulta. Podrá pulsar el icono del reloj y se lanzará el cálculo que describo a continuación.



### Botón de espera

```
// Recuperamos la hora actual del sistema
Set(horaActual; Now());
// Tomo la hora de la cita seleccionada en la galería
Set(horaCita; ThisItem.'Fecha y hora de la cita ');
// Calcula la diferencia en minutos entre ambas horas
Set(tiempoEsperaMinutos;DateDiff(horaActual;horaCita;TimeUnit.Minutes));
```

Y en el Label de la izquierda hago una concatenación de un string mostrando el resultado.

```
"Tiempo de espera: " & tiempoEsperaMinutos & " minutos"
```

## Pantalla de Citas del Especialista

The screenshot shows a web interface titled "Citas Reservadas de sus Pacientes" (Reserved Appointments for your Patients). At the top left is a house icon, and at the top right is the "inetum" logo. The main content area displays two appointment cards. Each card contains the following information:

Paciente :	sergio.felipe.ext@inetum.com
Fecha :	01/06/2024 10:00
Especialidad :	Enfermeria

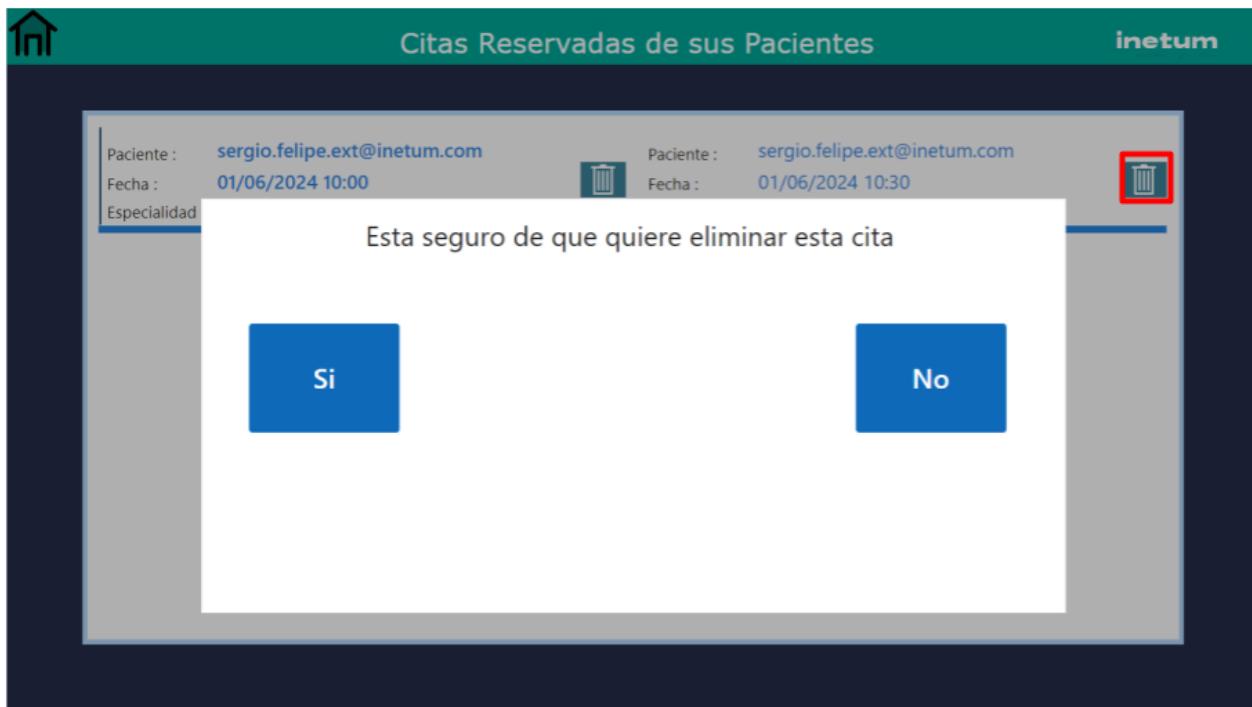
Each row has a small trash can icon on its right side. The first and second rows of the table are highlighted with a red rectangular box.

Muestro la galería filtrando la tabla Cita, donde tengo los especialistas, utilizando el siguiente código:

```
Filter(Cita; Especialistas.'Email Especialista' = EmailUsuario)
```

Con este filtro lo que hago es asegurarme de que la galería solo muestre las citas correspondientes al especialista cuyo correo electrónico coincide con el del usuario actual. Esto permite que cada especialista vea únicamente sus propias citas y no las de otros especialistas.

## Botón de eliminar



Cuando el especialista le da al botón de eliminar le sale un Popup para eliminar esa cita que ya ha sido atendida si le da al botón "No" el Popup desaparece y si le da al botón "Si" la cita se elimina de la tabla **Cita** con este código:

```
Remove(Cita; itemAEliminar);;
```

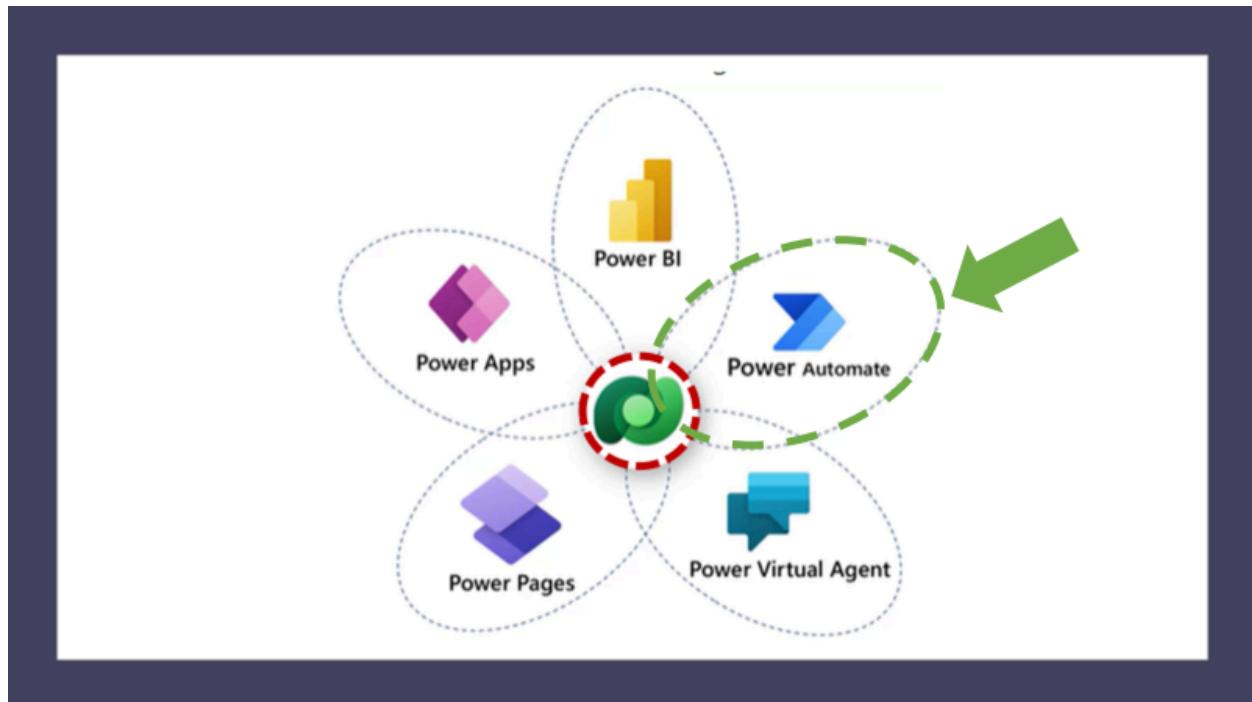
```
UpdateContext({mostrarConfirmacion: false})
```

Aquí lo que hago es que utilizo la función `Remove` para eliminar el registro de la tabla **Cita** y utilizó la variable `itemAEliminar` para especificar el registro que se va a eliminar.

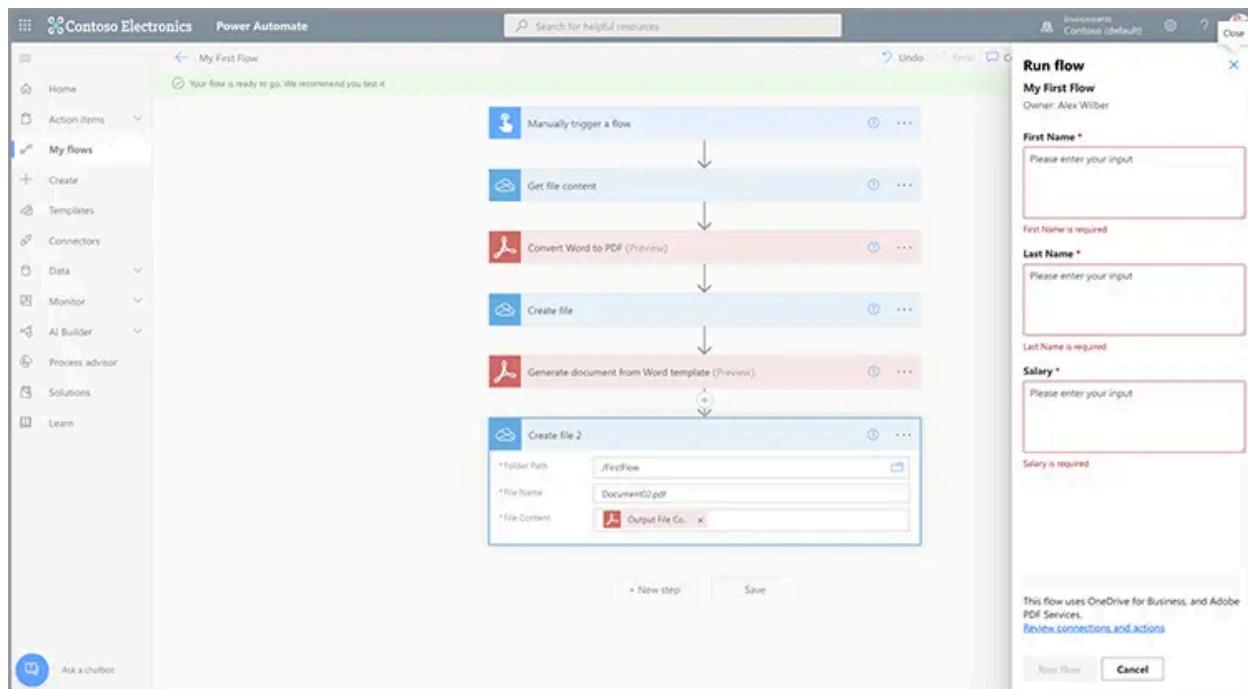
## Power Automate

Ahora voy a centrarme en introducir brevemente el último componente de mi aplicación.

Power automate, es un software desarrollado por Microsoft principalmente para automatizar tareas repetitivas.



Power Automate, permite agilizar procesos mediante flujos de trabajo automatizados. Esta herramienta proporciona una interfaz basada en diagramas de flujo tal y como se muestra en la imagen, y básicamente está enfocada a que los usuarios puedan crear tareas y resolver procesos repetitivos.



Pongamos algún ejemplo práctico para explicar que se puede conseguir con Power Automate :

- Puede guardar automáticamente archivos adjuntos de correos en OneDrive o SharePoint, así puedes organizar y almacenar documentos como facturas, contratos o informes.
- Permite generar tareas en Microsoft Planner al recibir solicitudes por correo electrónico.

---

En mi caso, hemos utilizado Power Automate para enviar **convocatorias de cita por email y gestionar confirmaciones**. Enviaremos automáticamente convocatorias de cita a los usuarios, y ellos mismos podrán gestionar sus confirmaciones de asistencia o cancelarla si no pueden asistir finalmente, simplificando la coordinación con los usuarios y los especialistas.

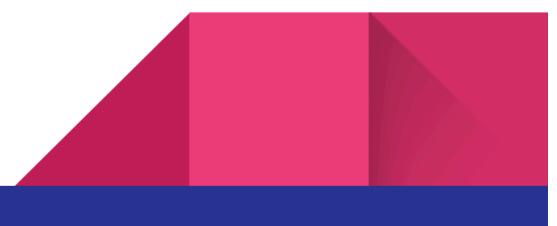
## Plantillas de Power Automate

No es fácil construir un flujo desde cero sin tener conocimientos de power automate, al menos eso es lo que me ha parecido a mi. Sin embargo, Microsoft tiene configurados bastantes flujos. Son de uso común y están en forma de plantillas. Si alguno de ellos hace exactamente lo que deseas, puedes implementarlo rápidamente. En caso contrario al menos tienes una plantilla desde la que partir para adaptarla a lo que necesitas en tu proyecto.

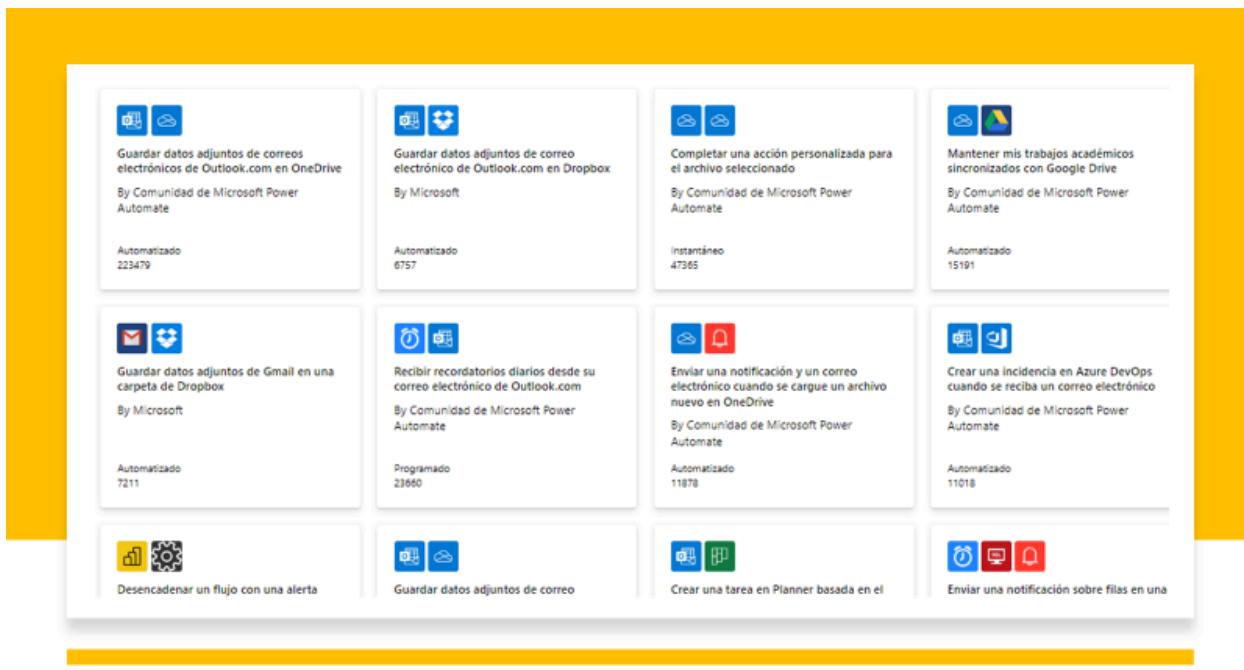
### **Las plantillas de Power Automate**

Para acceder a las plantillas , en la barra lateral izquierda “Plantillas”. Dentro están los flujos de trabajo preconfigurados que, además, los podemos encontrar separados por categorías. Hay un buscador.

Para empezar, hay que hacer clic encima de la plantilla y se nos abrirá una nueva página. En esta nueva página, encontraremos las herramientas con las que se conectarán el flujo.

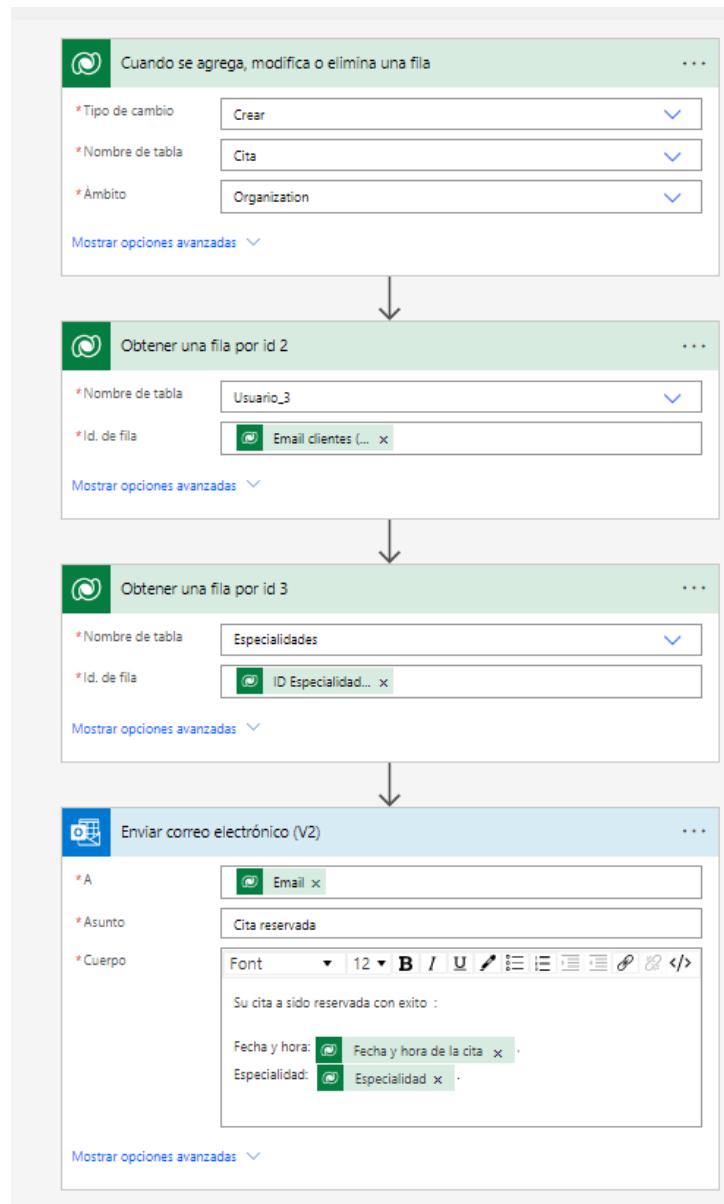


Hay una página de Microsoft, donde puedes explorar las plantillas existentes para ver cuál se adaptaría mejor al flujo que quieras montar. Yo he tenido que mirar varias, para encontrar las claves necesarias para montar mi flujo ya que no había una que hacía expresamente lo que necesitaba.



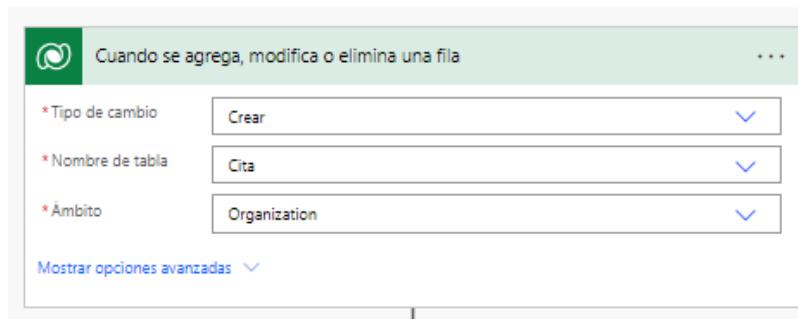
Fuente: <https://powerautomate.microsoft.com/es-es/templates/>

## Flujo Reservar Cita



Este flujo lo que hace es enviar un correo electrónico al cliente cuando éste reserva su cita ahora voy a detallar qué hace cada componente del flujo.

### Disparador:



Este sería el disparador del flujo aquí cuando se crea un registro en la tabla Cita comienza el flujo ahora específico que hace cada campo del flujo:

#### Tipo de cambio: Crear

- Esto hace que el flujo se active cuando se crea una nueva fila en la tabla especificada.

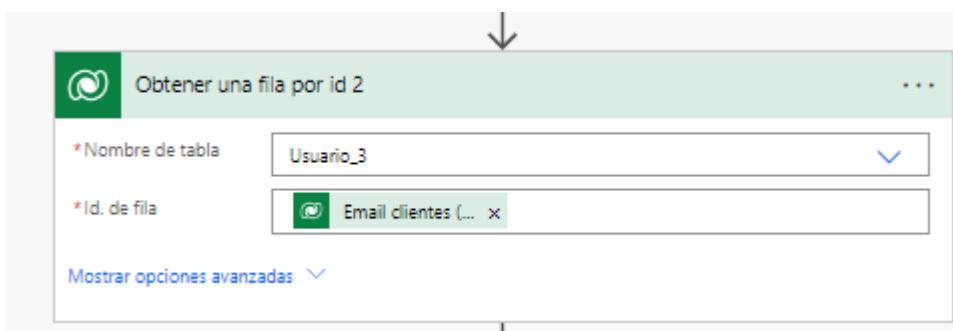
#### Nombre de tabla: Cita

- La tabla sobre la que se está monitoreando el cambio es **Cita**.

#### Ámbito: Organization

- Esto indica que el cambio se monitorea a nivel de toda la organización.

## Obtener una fila por ID 2:



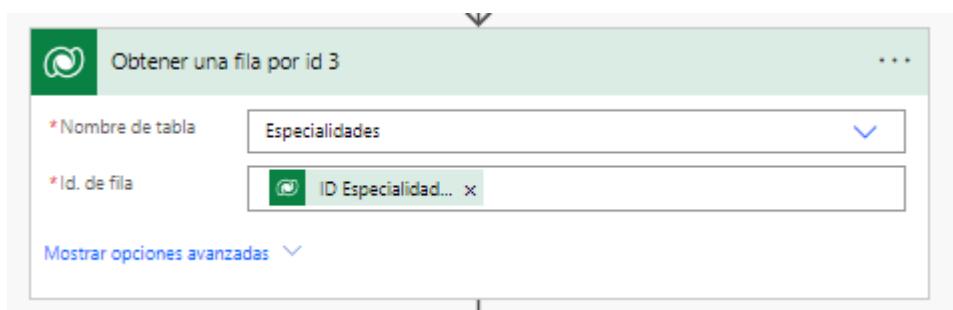
Este componente lo que hace es buscar en la tabla Usuario\_3 que corresponde al cliente que tiene el correo electrónico igual al valor especificado en Email clientes. Esto lo hago para coger el correo del cliente para poder mandarle el correo.

### Nombre de tabla: Usuario\_3

Este campo indica que la consulta se realizará en la tabla llamada Usuario\_3.

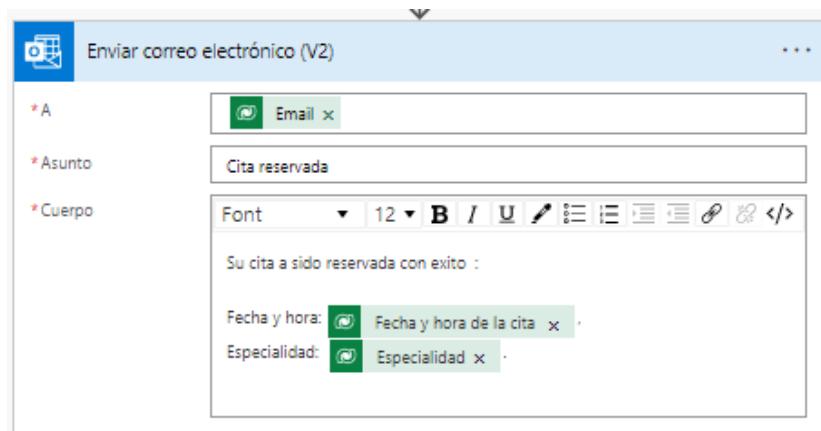
### ID de fila: Email clientes

Este campo define el identificador (ID) de la fila que se va a obtener de la tabla Usuario\_3.



Este componente hace lo mismo que el anterior y lo utiliza para coger la especialidad que el cliente ha elegido para poder mostrarle estos datos al cliente.

### Enviar correo:



Este componente lo que hace es enviar un correo electrónico al cliente mostrándole la fecha y la especialidad de la cita.

Correo nuevo

Favoritos

Prioritarios Otros

Cita cancelada

Su cita ha sido cancelada con exito...

Confirmación de cita

Cita reservada

Felipe-Garcia Sergio

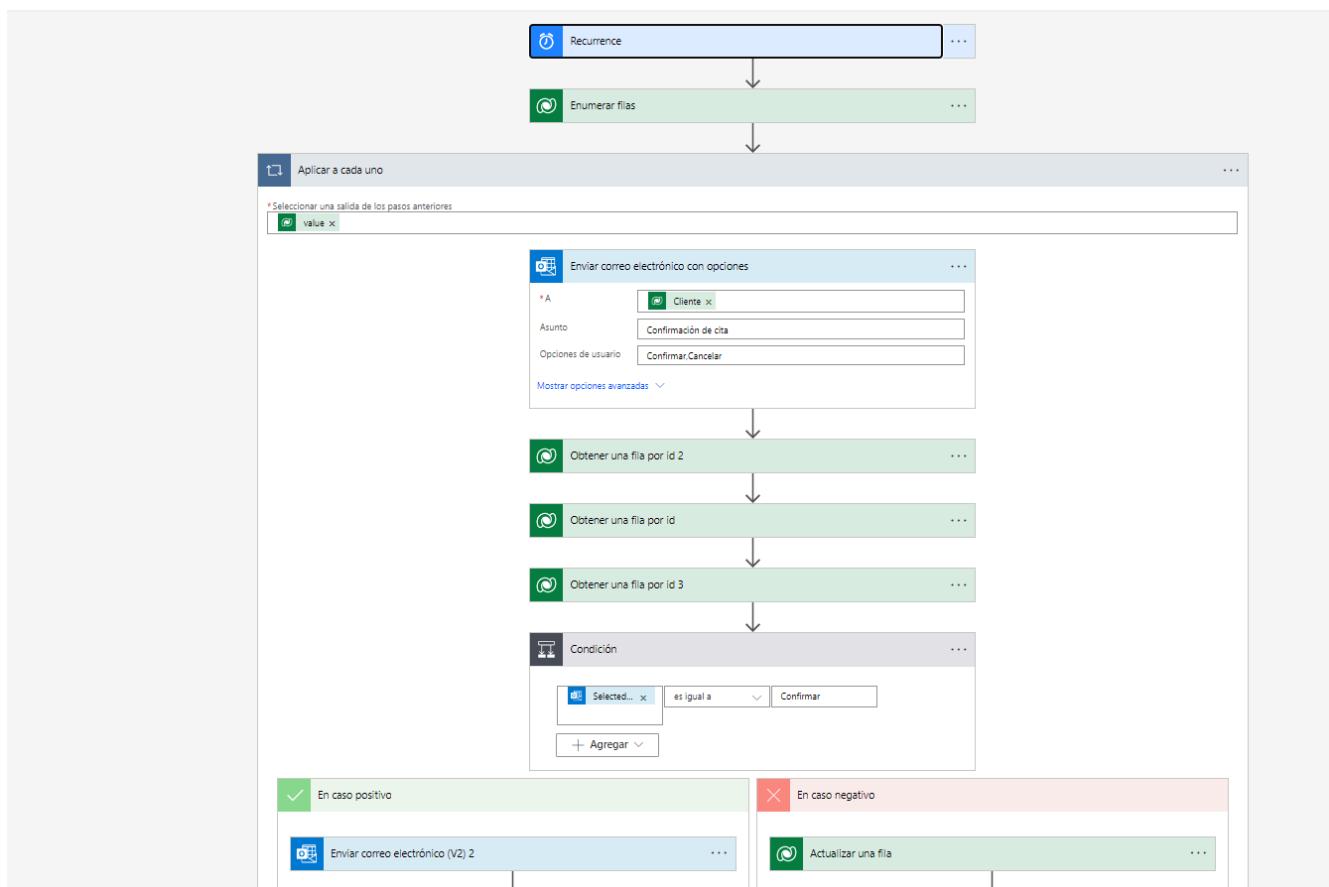
Para: Felipe-Garcia Sergio

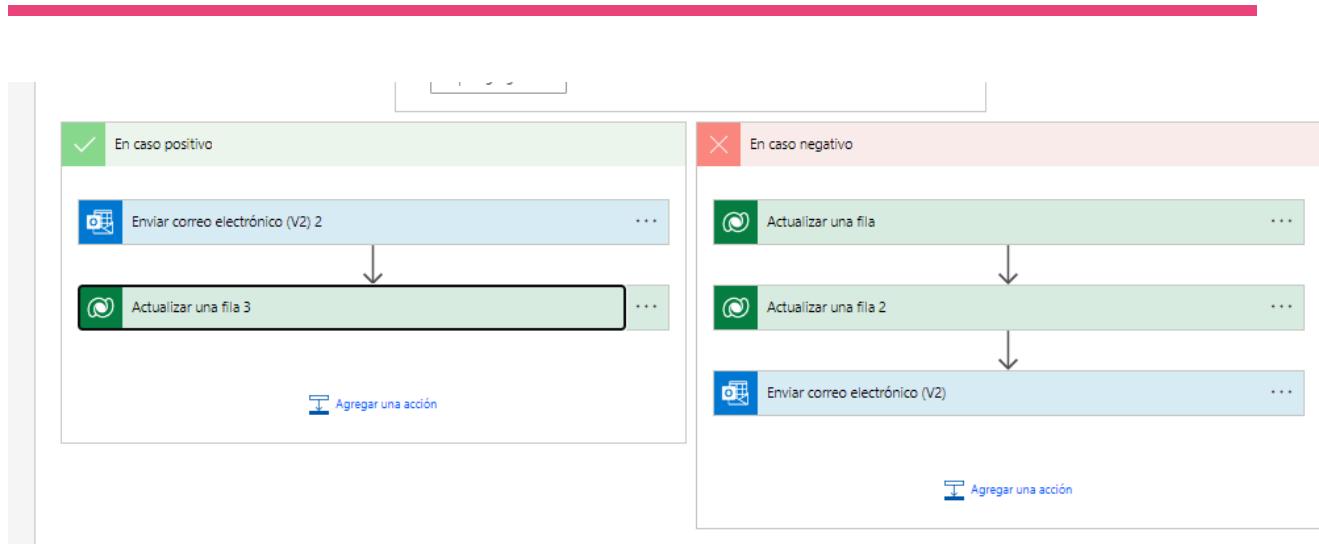
Su cita a sido reservada con exito :

Fecha y hora: 2024-06-01T10:00:00Z, Especialidad: Enfermería.

Responder Reenviar

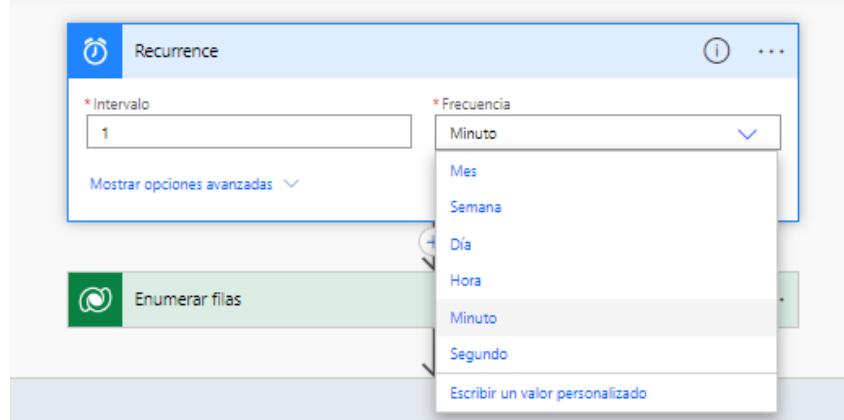
## Flujo de Recordatorio de cita





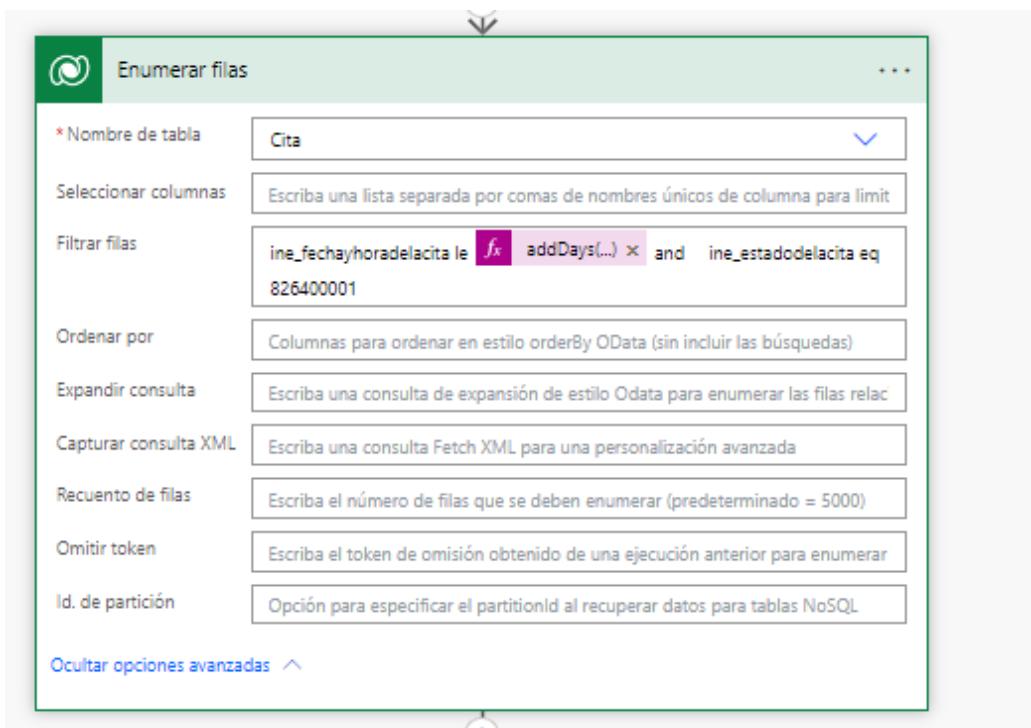
Este flujo lo que hace es enviar correos electrónicos a los clientes y tomar acciones según sus respuestas. El flujo se ejecuta de forma recurrente en intervalos de tiempo específicos, si el cliente confirma la cita se actualiza la columna de estado de la cita pasando de estar pendiente a confirmada y se envía un correo adicional confirmando que el usuario va a asistir a la cita , pero si el cliente cancela, se actualiza la base de datos y la columna estado de la cita pasa a cancelada y la cita vuelve a estar disponible para ser reservada por otro cliente y se le envía un correo para confirmar que su cita a sido cancelada.

## Recurrence (Recurrencia):



Este paso indica que el flujo se ejecutará de manera recurrente, es decir, en intervalos de tiempo especificados, pueden ser meses, días ,horas , minutos y segundos.

## Enumerar filas:

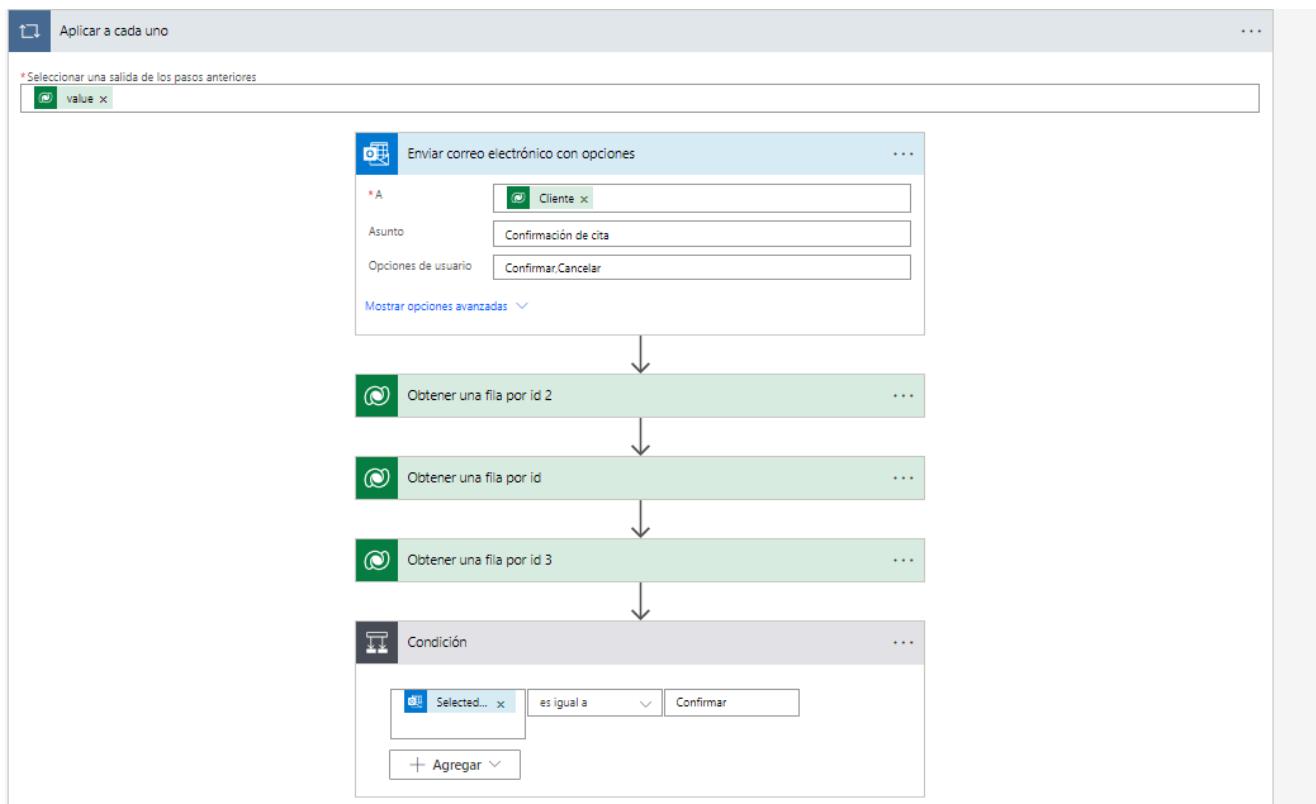


Este componente lo que hace es obtener obtener todas las citas de la tabla "Cita" que cumplen con las siguientes condiciones:

- La fecha/hora de la cita es anterior o igual a la fecha de la cita elegida por el cliente.
- El estado de la cita es igual a Pendiente.

## Filtrar filas:

- **ine\_fecha/hora de la cita le addDays(utcNow(), 2):** Filtra las citas cuya fecha y hora son menores o iguales a la fecha actual más dos días. Esto selecciona las citas que ocurren hoy o en los próximos dos días.
- **ine\_estado de la cita eq 826400001:** Filtra las citas cuyo estado es igual a **826400001** que significa que el estado sea pendiente.



En esta parte del flujo lo que hago es que envío un correo electrónico al cliente de tipo opción para que elija confirmar o cancelar la cita luego obtengo varios id para obtener los datos de la cita para mostrárselo al cliente cuando confirme la cita.

**Condición**

Selected... x es igual a Confirmar

+ Agregar

**En caso positivo**

Enviar correo electrónico (V2) 2

\* A Email x

\* Asunto Cita Confirmada

\* Cuerpo

Su cita ha sido confirmada estos son los datos:  
 Fecha y hora de la cita: Fecha y hora de la cita x  
 Su nombre: Nombre x  
 Su Email: Email x  
 Especialidad: Especialidad x

Mostrar opciones avanzadas v

**Actualizar una fila**

\* Nombre de tabla Agenda\_2

\* Id. de fila ID\_Agenda (Val... x)

Código de zona horaria de conversión UTC Código de la zona horaria que estaba en uso cuando se creó el registro.

Email cliente (Usuario\_3)

Especialidad (Especialidades)

Especialista (Especialistas)

Estado Estado del Agenda\_2

Fecha fin

Fecha inicio

ID Agenda Columna autonumerada que actúa como Primary Key de la tabla Agenda.

ID Cita (Cita) identificador que actúa como Foreign key de la tabla Cita

Número de versión de regla de zona horaria Solo para uso interno.

Ocupada No

Propietario (Propietarios) Id. del propietario

Razón para el estado Razón para el estado del Agenda\_2

Ocultar opciones avanzadas ^

**Actualizar una fila 3**

\* Nombre de tabla Cita

\* Id. de fila ID Cita (Valor) x

Cliente

Código de zona horaria de conversión UTC Código de la zona horaria que estaba en uso cuando se creó el registro.

Duración

Email clientes (Usuario\_3)

Especialistas (Especialistas)

Estado Estado del Cita

Estado de la cita Confirmada

Fecha y hora de la cita

Fin de semana

Hora de fin de atención

**Actualizar una fila 2**

\* Nombre de tabla Cita

\* Id. de fila ID Cita (Valor) x

Cliente

Código de zona horaria de conversión UTC Código de la zona horaria que estaba en uso cuando se creó el registro.

Duración

En la condición lo que hago es dependiendo de si se confirma o se cancela se envía un correo u otro y se actualiza las columnas de la base de datos por ejemplo si se cancela se actualiza la tabla Agenda\_2 y se cambia la columna ocupada de Sí a No y en la tabla Cita el estado de la cita cambia de Pendiente a Cancelada

Traducir mensaje a: Español | Nunca traduzca de: Inglés

Felipe-Garcia Sergio  
Para: Felipe-Garcia Sergio

Responder | Responder a todos | Reenviar | ...  
Sáb 01/06/2024 14:17

Confirmación de cita

Confirmar | Cancelar

**Confirmación de cita**

Select one of the options below to respond

Confirmar | Cancelar

Message sent via Microsoft Power Automate, enabling you to create automated workflows between your favorite apps and services.  
© Microsoft Corporation 2024

Responder | Reenviar

FS

Felipe-Garcia Sergio

Para: Felipe-Garcia Sergio

Su cita ha sido cancelada con éxito, tenga un buen día

Responder

Reenviar

---

## Conclusión.

Al evaluar mi trabajo y la investigación para llevar adelante el desarrollo de mi aplicación de citas médicas utilizando Microsoft Power Apps y la Power Platform, puedo decir que estas herramientas son utilizadas en una proporción muy grande de empresas. Creo que esta plataforma es significativamente adoptada por en el mundo la plataforma es versátil ya que permite hacer una gran variedad desarrollos para resolver muchos problemas. Me hubiera gustado disponer de más tiempo para tener unos conocimientos más profundos.

Este proyecto me ha permitido familiarizarme con Power Apps. Lo aprendido en clase me ha permitido entender que con Power Apps se pueden simplificar tareas que podrían haber sido complejas en otros entornos que hemos visto, me he dado cuenta que la curva de aprendizaje para hacer cosas esenciales no ha sido extremadamente dura y me ha permitido alcanzar resultados más allá de mis expectativas iniciales. Teniendo en cuenta que ni había oído hablar de power apps.

Creo que debo explorar aún más las posibilidades de innovación y personalización que ofrecen este tipo de plataformas, ya que creo que tienen el potencial de impulsar significativamente mi carrera profesional.



# Webgrafía

## Fuentes:

- Microsoft. (n.d.). Visión general del tema de los portales de Power Apps. Microsoft Learn. <https://learn.microsoft.com/es-es/power-apps/maker/portals/theme-overview>
- Microsoft. (n.d.). Introducción a Power Apps. Microsoft Learn. <https://learn.microsoft.com/en-us/training/modules/get-started-with-powerapps/1-powerapps-introduction>
- Microsoft. (n.d.). Introducción a la creación de la primera aplicación basada en modelos con Dataverse. Microsoft Learn. <https://learn.microsoft.com/en-us/training/modules/build-first-model-driven-app-dataverse/introduction>
- Microsoft. (n.d.). Power Automate. Microsoft. <https://www.microsoft.com/es-es/power-platform/products/power-automate>
- Gartner. (2023, 4 de octubre). Reimpresión del informe de Gartner. Gartner. <https://www.gartner.com/doc/reprints?id=1-2F7NELJY&ct=231004&st=sb&culture=es-es&country=es>

## Referencias:

- Kahil, D. (2022). Power Apps Model-Driven Apps Fields Cheatsheet. Danikahil. <https://i0.wp.com/danikahil.com/wp-content/uploads/2022/08/Power-Apps-Model-Driven-Apps-Fields-Cheatsheet-1-scaled.jpg?ssl=1>
- Microsoft. (n.d.). Power Fx Overview. Microsoft Learn. <https://learn.microsoft.com/es-es/power-platform/power-fx/overview>

---

## Agradecimientos

En primer lugar, quiero expresar mi más profundo agradecimiento a mis padres, quienes han sido mi pilar fundamental a lo largo de este largo viaje desde que era pequeño. Gracias por su amor incondicional, su apoyo constante y su confianza en mis capacidades. Su sacrificio y dedicación me han proporcionado las herramientas necesarias para superar los desafíos que se han presentado en mi camino. Agradezco los consejos y el ánimo que me han dado en los momentos más difíciles. Han sido una motivación constante, y no hubiera llegado hasta aquí sin ellos.

Agradecer al equipo de profesores de Campus FP. Su paciencia, dedicación han sido básicas en mi formación. Gracias por haberme enseñado una profesión. Cada uno de ustedes ha contribuido a mi desarrollo profesional y personal. Sus clases, me han preparado para enfrentar el mundo laboral.

Mi gratitud también para los tutores de Inetum por haberme dado la oportunidad de conocer un entorno laboral profesional. Gracias por recibirme con los brazos abiertos y por brindarme un espacio donde pude aplicar y expandir mis conocimientos.

Este trabajo de fin de grado no sería posible sin el apoyo de todas estas personas. A todos y cada uno, les debo mi más sincero agradecimiento.

