

# INTRODUCCIÓN AL SISTEMA DE EVALUACIÓN DINÁMICA

El presente proyecto consiste en una solución integral desarrollada sobre el ecosistema de Microsoft Power Platform (Power Apps, Dataverse y Power Automate), diseñada para automatizar el ciclo de vida de los exámenes corporativos. La aplicación permite a los administradores gestionar un banco de preguntas dinámico y a los alumnos realizar evaluaciones cuya corrección es procesada en tiempo real.

El sistema destaca por su capacidad de corrección híbrida: las preguntas de opción múltiple se califican automáticamente mediante cruce de datos, mientras que las preguntas abiertas activan un flujo de aprobación para su revisión manual. Toda la información reside en una arquitectura relacional en Dataverse, garantizando la integridad de los datos y la trazabilidad de los resultados por alumno.

Tablas > Tabla Examen

Propiedades de la tabla

Propiedades

Herramientas

Nombre	Columna principal	Descripción
Tabla Examen	Id Examen	
Tipo	Última modificación	
Standard	hace 4 meses	

Esquema

Columnas

Relaciones

Claves

Experiencias d

Formularic

Vistas

Gráficos

Paneles

Tabla Examen columnas y datos

Id Examen * ↑	Nombre Examen	Autor	Estado	Informacion sobre Examen	
1	examen prueba 1	Víctor Navarrete Montero	Desactivo	examen sobre las cosas	A
2	examen prueba 2	Víctor Navarrete Montero	Activo	0	A
3	examen prueba 3	Víctor Navarrete Montero	Finalizado	examen sobre las cosas 3	A
5	examen Express	Víctor Navarrete Montero	Desactivo	examen para saber tus conocimie...	A
6	prb1	Víctor Navarrete Montero	Desactivo	cositas	A
	Escribir texto		Seleccionar opción	Escribir texto	

## 1. Tabla Examen (Maestra de Formación)

Es la tabla de configuración de alto nivel donde se definen los exámenes disponibles en la plataforma.

- **Id Examen:** Clave primaria numérica que identifica de forma única cada examen.

- **Nombre Examen:** El título descriptivo que aparecerá en la interfaz del alumno (ej: "examen prueba 1", "examen Express").
- **Autor:** Atribuye la propiedad del examen a un revisor o docente (ej: Víctor Navarrete Montero).
- **Estado:** Controla la visibilidad del examen (Activo, Desactivo o Finalizado).
- **Información sobre Examen:** Campo de texto para añadir instrucciones o descripciones adicionales para el usuario.

Tablas > Tabla Preguntas

Propiedades de la tabla			Propiedades	Herramientas	Esquema
Nombre	Columna principal	Descripción			Columnas
Tabla Preguntas	Id Pregunta				Relaciones
Tipo	Última modificación				Claves
Standard	hace 4 meses				

Tabla Preguntas columnas y datos

Id Pregunta * ↑	Tipo Pregunta	Texto Pregunta	Id Examen	
1	Opciones	funcionara	1	Au
2	V/F	seguro	1	Au
3	Opciones	yya veremos	1	Au
4	V/F	uyfgui	1	Au
5	Opciones	jersyk	1	Au
6	Opciones	ihguiky	1	Au
7	Abierta	dnoefowefj	1	Au
	Seleccionar opción	Escribir texto	Seleccionar búsqueda	

## Tabla Preguntas (Banco de Ítems)

Contiene todas las preguntas individuales vinculadas a los exámenes mediante una relación jerárquica.

- **Id Pregunta:** Identificador único de cada ítem evaluativo.
- **Texto Pregunta:** El enunciado o pregunta que el alumno debe responder.
- **Tipo Pregunta:** Define la naturaleza de la respuesta (Opciones, V/F o Abierta). Esta columna es crucial porque determina qué controles se muestran en Power Apps y qué lógica de corrección sigue Power Automate.

- **Id Examen (Relación):** Columna de búsqueda (Lookup) que asocia la pregunta a un examen específico de la Tabla 1, permitiendo el filtrado dinámico en la App.

Id Registro * ↑	Id Usuario *	Id Examen	Id Pregunta2	Nota Asignada	Id Respuesta	Respuesta texto	+7 m
84	1	1	3		7		
85	1	1	1		2		
86	1	1	3				
87	1	1	2		4		
88	1	1	1		1		
89	1	1	3		7		
9					4		
90	1	1	2		4		
91	1	1	1		1		
92	1	1	4	0	11		
93	1	1	1	0	1		
94	1	1	1	0	2		
95	1	1	1	0	3		
96	1	1	6	1		todo da nota final vi	
97	1	1	6	0		este no va a funcion	
98	1	1	1	1	2		
99	1	1	1	0	1		
	Seleccionar búsqueda	Seleccionar búsqueda	Seleccionar búsqueda	Escribir texto	Seleccionar búsqueda	Escribir texto	

## Tabla ExámenesRealizados (Registro de Respuestas)

Es la tabla transaccional donde se vuelcan todas las interacciones de los alumnos pregunta por pregunta.

- **Id Registro:** Identificador correlativo de cada entrada enviada desde la aplicación.
- **Id Usuario:** Vincula la respuesta con el contacto o perfil del alumno que realiza la prueba.
- **Id Examen e Id Pregunta2:** Aseguran la trazabilidad completa, indicando a qué examen y a qué pregunta corresponde cada respuesta.
- **Nota Asignada:** Campo actualizado por Power Automate tras la corrección (automática o manual) con los puntos obtenidos.
- **Respuesta Texto:** Almacena el contenido escrito por el alumno en caso de preguntas abiertas.
- **Id Respuesta:** Identificador del ítem de respuesta seleccionado en las preguntas de opción múltiple.

## Anexo: Lógica de Programación y Código

En esta sección se detallan los scripts utilizados para la lógica de negocio de la aplicación.

### A. Configuración Global (OnStart)

Este código inicializa las variables de sesión y recupera los datos del usuario actual para evitar múltiples consultas a la base de datos durante el uso de la app.

Fragment de codi

```
Set(varCorreoUsuario; UsuariosdeOffice365.MyProfileV2().mail);;  
Set(varRolUsuario; Text(LookUp('Tabla Usuarios'; Correo = varCorreoUsuario; Rol)));;  
Set(varUserEmail; User().Email);;
```

```
Refresh('Tabla Usuarios');;
```

```
Set(varUsuarioActual; LookUp('Tabla Usuarios'; Correo = varUserEmail));;
```

```
If(  
    IsBlank(varUsuarioActual);  
    Notify("No se encontró su registro en Usuarios. Contacta al administrador.");  
    NotificationType.Warning)  
)
```

### B. Lógica de Navegación y Seguridad (OnVisible / OnStartScreen)

Se utiliza la función **With** para optimizar el rendimiento y determinar a qué pantalla debe ser enviado el usuario según su permiso.

Fragment de codi

```
With(  
    { u: LookUp('Tabla Usuarios'; Correo = User().Email) };  
    If(  
        IsBlank(u);  
        Pantalla_Error;  
        If(  
            u.Rol = Rol.Administrador;  
            Home_Admins;  
            If(  
                u.Rol = Rol.Usuario;  
                Home_Usuario  
            )  
        )  
    )  
)
```

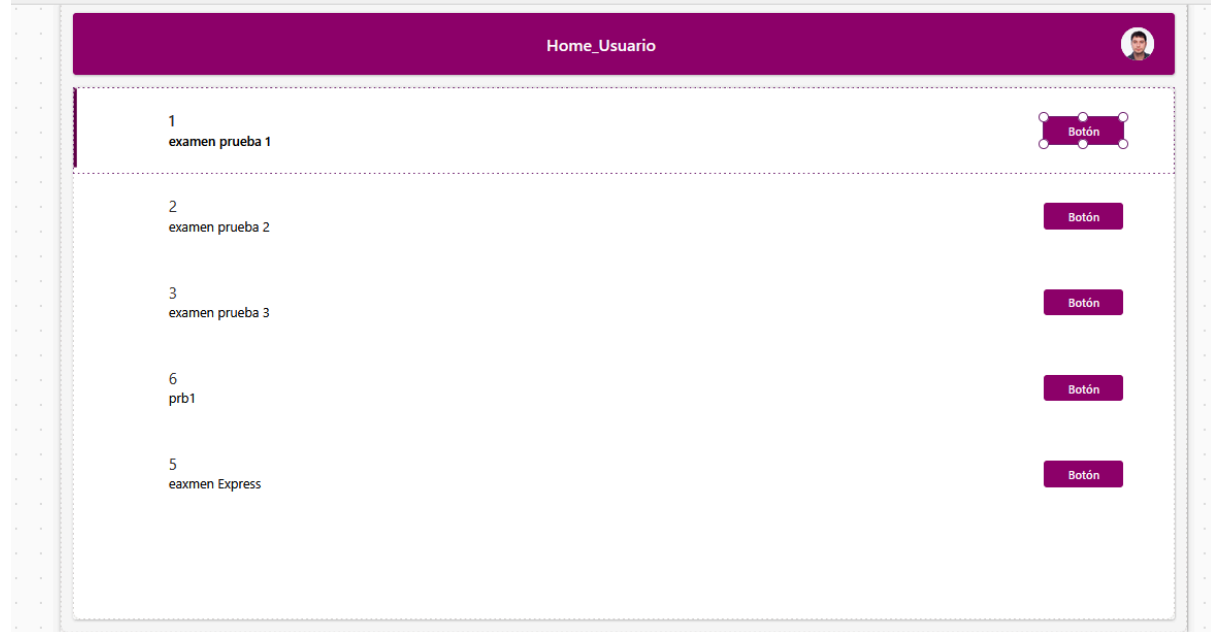
---

```

Set(FormacionSeleccionada; ThisItem);;
Set(ExamenSeleccionadoID; ThisItem.'Id Examen');;
Set(UsuarioSeleccionado; ThisItem.'Id User EX');;
Navigate(Realizar_Examenes; ScreenTransition.Fade; ThisItem)

```

ar formato al texto    Quitar formato    Buscar y reemplazar



## (Pantalla de Inicio)

Esta captura muestra la interfaz principal del usuario (**Home\_Usuario**) donde se listan las formaciones o exámenes disponibles.

- **Interfaz Visual:** Se utiliza una galería vertical que muestra el ID y el nombre de los exámenes (ej. "examen prueba 1", "eaxmen Express").
- **Lógica del Botón:** Cada fila tiene un botón para iniciar el examen. Al pulsarlo, se ejecutan las siguientes acciones:
  1. **Set(FormacionSeleccionada; ThisItem):** Almacena todo el registro del examen elegido en una variable global.
  2. **Set(ExamenSeleccionadoID; ThisItem.'Id Examen'):** Guarda específicamente el ID del examen para filtros posteriores.
  3. **Set(UsuarioSeleccionado; ThisItem.'Id User EX'):** Identifica al usuario que realiza la acción.
  4. **Maps(Realizar\_Examenes; ...):** Redirige al usuario a la pantalla de ejecución de la prueba.

The screenshot shows a web application interface. At the top, there is a code editor with a filter function: `Filter('Prueba Preguntas'; IdExamen.'Id Examen' = FormacionSeleccionada.'Id Examen')`. Below the code editor is a toolbar with buttons for 'Aplicar formato al texto', 'Quitar formato', and 'Buscar y reemplazar'. The main content area displays a list of questions, each with a title, a timestamp, and a right-pointing arrow. The questions are: 'numero de pokemons' (08/10/2025 17:05), 'me gustan los macarrones' (08/10/2025 17:08), 'tipos de ordenadores' (08/10/2025 17:09), 'el jesus la ...' (08/10/2025 17:09), and 'prrimer pregunta abierta' (23/10/2025 13:36). The list is enclosed in a dashed purple border.

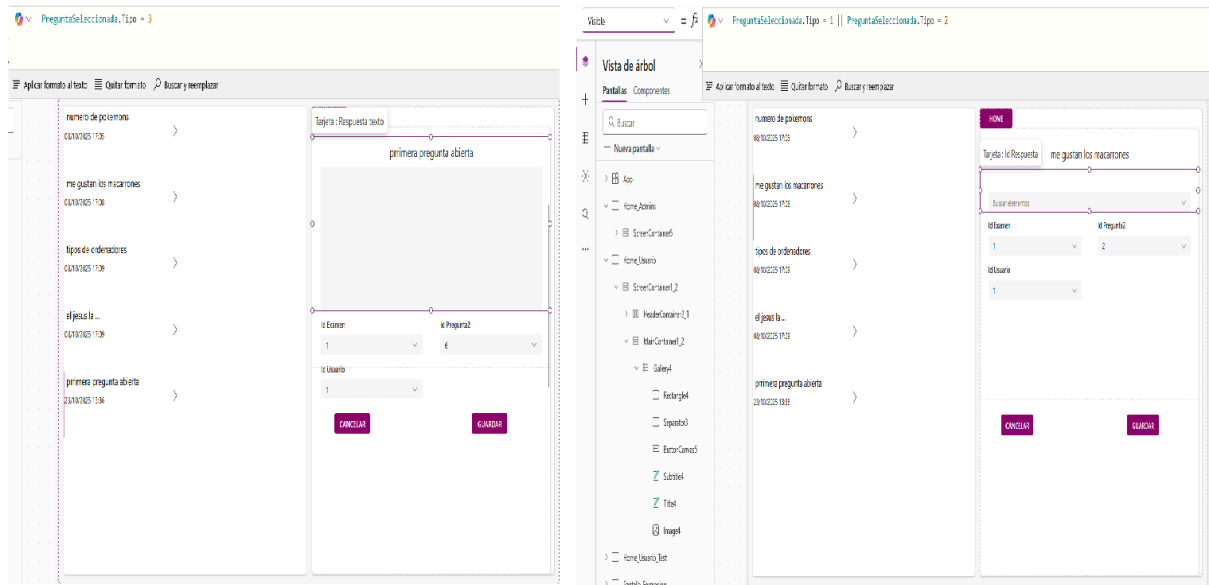
Question Title	Timestamp
numero de pokemons	08/10/2025 17:05
me gustan los macarrones	08/10/2025 17:08
tipos de ordenadores	08/10/2025 17:09
el jesus la ...	08/10/2025 17:09
prrimer pregunta abierta	23/10/2025 13:36

## Componente: Formulario de Entrada de Datos (Forms)

Esta parte de la interfaz es la encargada de recolectar la información del alumno y enviarla a la base de datos para iniciar el proceso de corrección.

- **Vínculo de Datos:** El formulario está conectado directamente a la entidad de Dataverse donde se registran las respuestas.
- **Campos de Control (Metadata):**
  - **Id Examen:** Campo automático que vincula la respuesta al examen específico seleccionado en el Home.

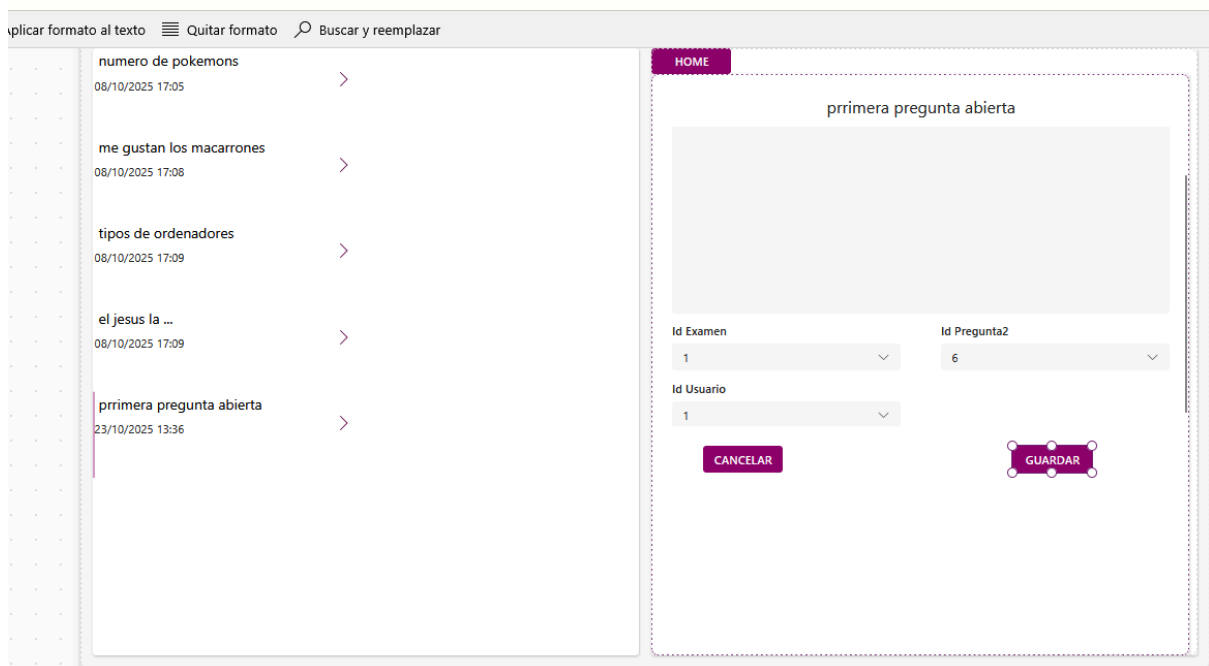
- **Id Pregunta2:** Identificador que relaciona la respuesta con la pregunta activa de la galería izquierda.
- **Id Usuario:** Registra el identificador del alumno para asegurar que la nota se asigne a la persona correcta.



```

✓ SubmitForm(Form1);;
NewForm(Form1);;
UpdateContext({ShowForm: false});;
Notify("✅ Guardado")

```





## Componente: Formulario de Registro Dinámico

El formulario está diseñado para procesar y enviar los datos de cada respuesta de forma individual a **Dataverse**, asegurando que el progreso se guarde inmediatamente después de cada interacción.

- **Gestión de Respuestas Individuales:** Debido a la arquitectura del sistema, el usuario debe guardar cada pregunta por separado. Esto garantiza que, incluso si la sesión se interrumpe, las respuestas anteriores ya hayan sido procesadas por el motor de automatización.
- **Campos de Trazabilidad:** El formulario captura automáticamente metadatos esenciales para cada registro:
  - **Id Examen:** Vincula la respuesta al examen específico que se está cursando.
  - **Id Pregunta2:** Identifica exactamente qué pregunta de la base de datos se está respondiendo.
  - **Id Usuario:** Registra la identidad del alumno para la posterior suma de su nota final.



## Lógica del Botón GUARDAR (OnSelect)

Al pulsar el botón **GUARDAR**, se ejecuta una cadena de funciones que coordina la aplicación con la base de datos y la interfaz:

Fragment de codi

```
SubmitForm(Form1);;  
NewForm(Form1);;  
UpdateContext({ShowForm: false});;  
Notify("✅ Guardado")
```

1. **SubmitForm(Form1)**: Envía los datos actuales a la tabla **Tabla ExámenesRealizados** en Dataverse. Esta acción es el disparador (Trigger) que inicia el flujo de **Power Automate** para calificar la respuesta.
2. **NewForm(Form1)**: Restablece el formulario a su estado inicial, permitiendo que el usuario seleccione y responda la siguiente pregunta sin conflictos de datos previos.
3. **UpdateContext({ShowForm: false})**: Oculta el formulario de la interfaz hasta que el usuario seleccione una nueva pregunta de la galería.
4. **Notify("✅ Guardado")**: Proporciona feedback visual instantáneo al usuario, confirmando que su respuesta ha sido registrada con éxito.



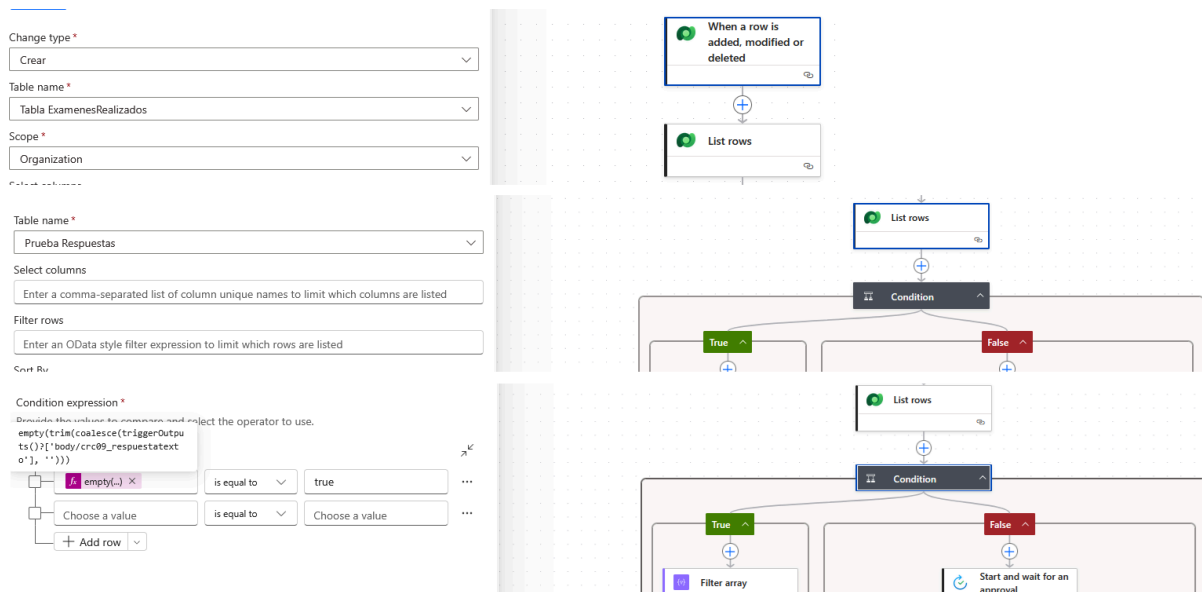
## Control de Visibilidad según el Tipo de Pregunta

El sistema adapta dinámicamente qué campos mostrar basándose en la configuración de la pregunta seleccionada.



- **Preguntas Abiertas:** Cuando el tipo de pregunta es de texto (Tipo 3), el formulario muestra una tarjeta de entrada de texto grande para respuestas extensas.
- **Preguntas de Selección:** Para tipos de opción múltiple (Tipo 1 o 2), la interfaz cambia para mostrar un selector de elementos o búsqueda, facilitando la elección de la respuesta correcta.

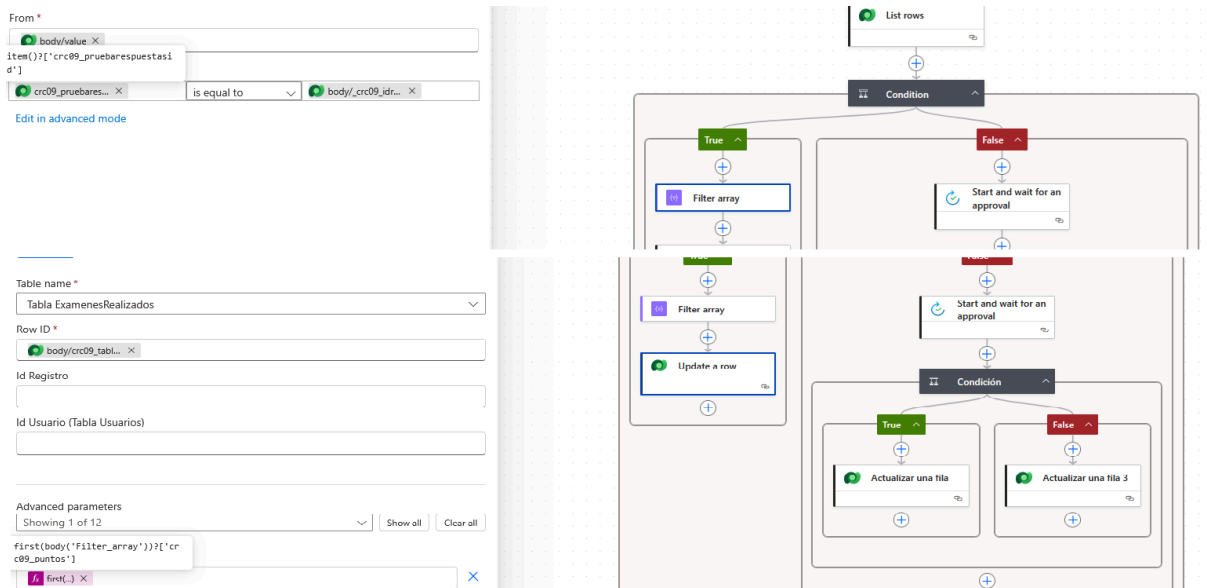
## Aquí empieza el Power Automate



### Archivo: **cp1.png** - El Disparador y la Validación Inicial

Este es el inicio del flujo donde el sistema detecta la nueva respuesta.

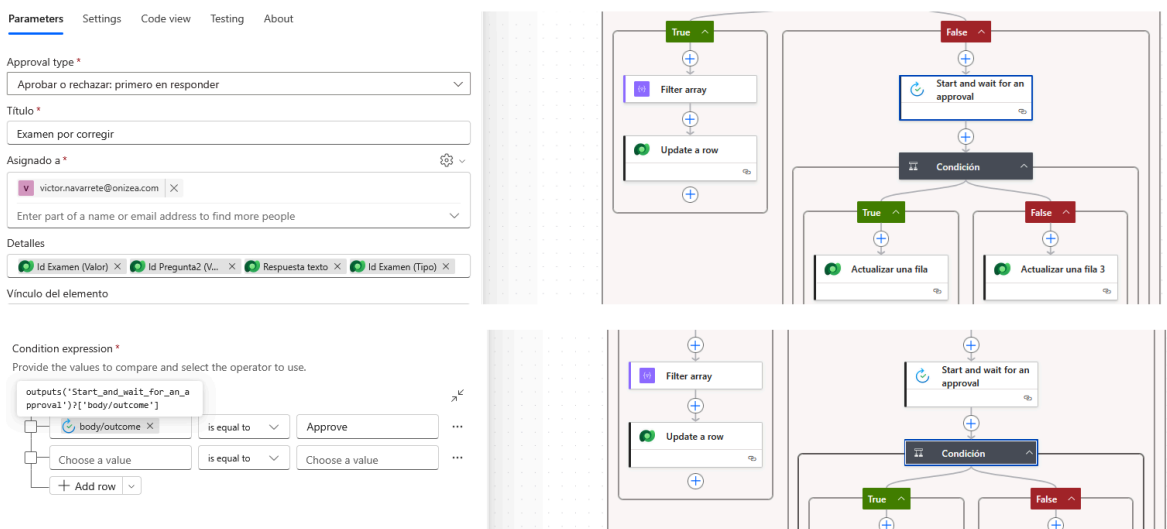
- **Trigger (When a row is added):** Se activa cuando el usuario pulsa "GUARDAR" en la App y se crea un registro en la tabla **Tabla ExámenesRealizados**.
- **Configuración del Trigger:** El cambio es de tipo "Crear" y el alcance es "Organization" para que funcione con cualquier usuario.
- **Condición de Tipo de Pregunta:** Utiliza la expresión `empty(trim(coalesce(triggerOutputs()?['body/crc09_respuestatexto'], '')))` para verificar si la respuesta es de texto (pregunta abierta) o si viene de una selección de opciones.



## Archivo: **cp2.png** - Filtro de Respuestas Cerradas

Si la pregunta es de opción múltiple, el flujo busca la puntuación automáticamente.

- **Acción Filter Array:** Cruza los datos de la tabla **Prueba Respuestas** con la respuesta enviada por el usuario para encontrar la coincidencia exacta.
- **Acción Update a row:** Una vez encontrada la respuesta, el flujo actualiza la fila original de **Tabla ExámenesRealizados** insertando los puntos correspondientes mediante la expresión `first(body('Filter_array'))?['crc09_puntos']`.



## Archivo: **cp3.png** - Gestión de Preguntas Abiertas (Aprobaciones)

Si la respuesta es de texto, el sistema requiere intervención humana.

- **Acción Start and wait for an approval:** Crea una solicitud de aprobación titulada "Examen por corregir" dirigida al administrador (en este caso, Víctor Navarrete).
- **Detalles del Approval:** En la tarjeta de aprobación se envía el **Id Examen**, el **Id Pregunta** y el contenido de la **Respuesta texto** para que el revisor sepa qué está evaluando.
- **Condición de Resultado:** El flujo se detiene hasta que el administrador marca "Approve" o "Reject".

Table name \*

Tabla ExámenesRealizados

Row ID \*

body/crc09\_tabl...

Id Registro

Id Usuario (Tabla Usuarios)

Advanced parameters

Showing 1 of 12

Show all Clear all

Parameters Settings Code view Testing About

Table name \*

Tabla ExámenesRealizados

triggerOutputs()?['body/crc09\_tabla\_exámenesrealizadosid']

body/crc09\_tabl...

Id Registro

Id Usuario (Tabla Usuarios)

Condición

True

Actualizar una fila

False

Actualizar una fila 3

## Archivo: **cp4.png** - Actualización tras Corrección Manual

Muestra cómo se graban los puntos una vez el administrador ha decidido la nota.

- **Acciones Actualizar una fila:** Dependiendo de si el administrador aprobó o rechazó (o la nota que puso), el flujo identifica el registro mediante `triggerOutputs()?['body/crc09_tabla_exámenesrealizadosid']` y guarda el resultado final en la base de datos.

Table name \*

Tabla ExámenesRealizados

Select columns

Enter a comma-separated list of column unique names to limit which columns are listed

Filter rows

Enter an OData style filter expression to limit which rows are listed

Parameters Settings Code view About

outputs('List\_rows\_1')['body/value']

body/value

Filter Query \*

@and(  
@equals(item()['Id Usuario (Valor)'], triggerOutputs()['body/Id Usuario']),  
@equals(item()['Id Examen (Valor)'], triggerOutputs()['body/Id Examen'])  
)

Name \*

SumaNotaAsign

Type \*

Float

Value

0

List rows 1

Filter array 1

Initialize variable

Filter array 1

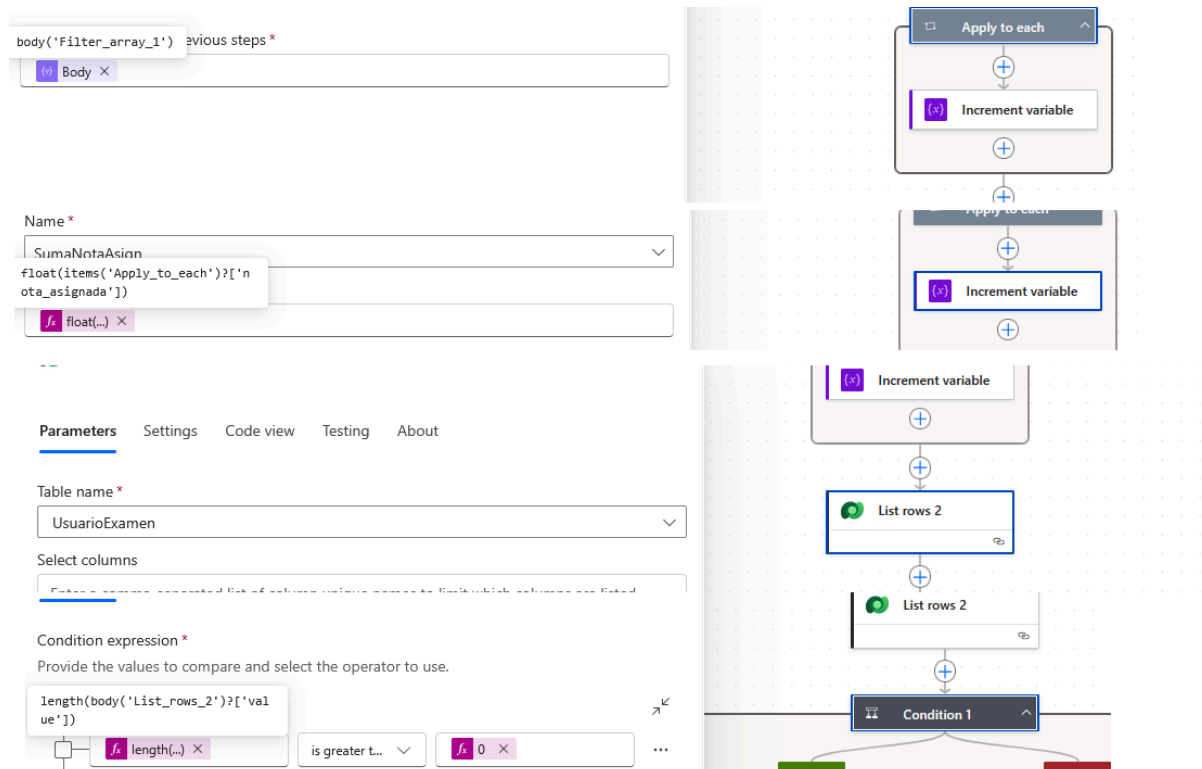
Initialize variable

Apply to each

## Archivo: cp5.png - Recopilación de Resultados Totales

Una vez corregida la pregunta actual, el flujo calcula el total del examen.

- **List rows 1:** El flujo busca en la tabla **Tabla ExámenesRealizados** todos los registros que pertenezcan al mismo **Id Usuario** y al mismo **Id Examen** mediante un filtro OData.
- **Initialize variable:** Se crea la variable **SumaNotaAsign** de tipo Float (número decimal) iniciada en 0 para empezar el conteo.



## Archivo: cp6.png - Bucle de Sumatoria

Aquí es donde se hace el cálculo matemático de la nota.

- **Apply to each:** El flujo recorre cada una de las respuestas encontradas en el paso anterior.
- **Increment variable:** Suma los puntos de cada respuesta a la variable mediante la expresión `float(items('Apply_to_each')?['nota_asignada'])`.
- **Condición de Existencia:** Al terminar la suma, el flujo comprueba si el usuario ya tiene una fila en la tabla **UsarioExamen** usando la expresión `length(body('List_rows_2')?['value'])`.

Parameters Settings Code view About

outputs('List\_rows\_2')?['body/value']

body/value

Table name \*

UsuarioExamen

Row ID \*

@odata.id

Id Registro

Advanced parameters

Showing 4 of 11

Show all

Clear all

IdExamen (Tabla Examen)

\_crc09\_idexame...

IdUsuario (Contactos)

\_crc09\_idusuario...

IdUsuario (Cuentas)

\_crc09\_idusuario...

Nota

SumaNotaAssign

True

For each

False

For each 1

Condition 1

True

For each

Update a row 1

False

For each 1

Add a new row

## Archivo: cp7.png y cp8.png - Finalización del Expediente

El paso final para mostrar la nota en la App.

- **Si el registro existe (True):** Ejecuta un **Update a row 1** en la tabla **UsuarioExamen**, actualizando el campo "Nota" con el valor final de la variable **SumaNotaAssign**.
- **Si el registro no existe (False):** Ejecuta un **Add a new row**, creando por primera vez el expediente del usuario para ese examen con su nota correspondiente.

Parameters Settings Code view About

outputs('List\_rows\_2')?['body/value']

body/value

True

For each

False

For each 1

UsuarioExamen

Id Registro \*

Tabla Examenes...

Advanced parameters

Showing 4 of 13

Show all

Clear all

IdExamen (Tabla Examen)

\_crc09\_idexame...

IdUsuario (Contactos)

\_crc09\_idusuario...

IdUsuario (Cuentas)

\_crc09\_idusuario...

Nota

SumaNotaAssign

Condition 1

True

For each

Update a row 1

False

For each 1

Add a new row