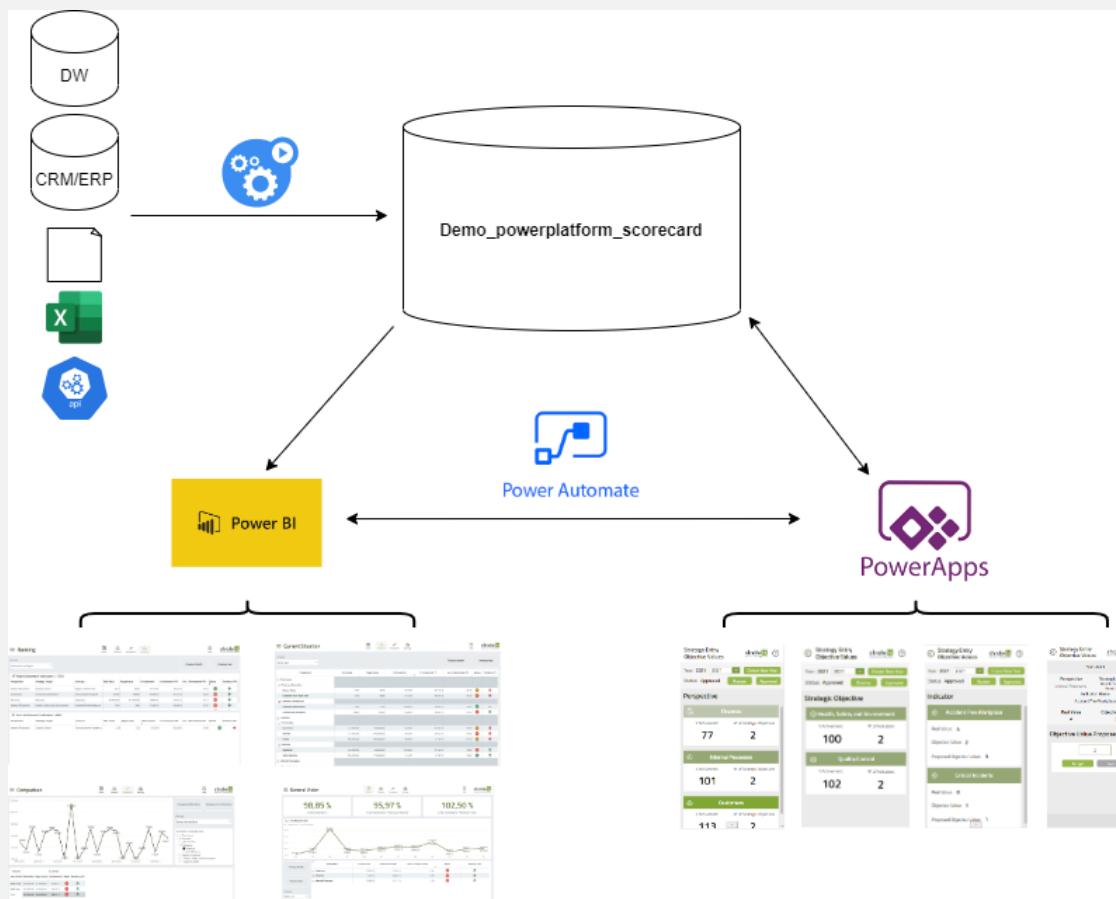


Cómo trabajar con PowerBI, PowerApps y Power Automate



CREACIÓN DE UNA APLICACIÓN DE SCORECARD

CONTENIDO

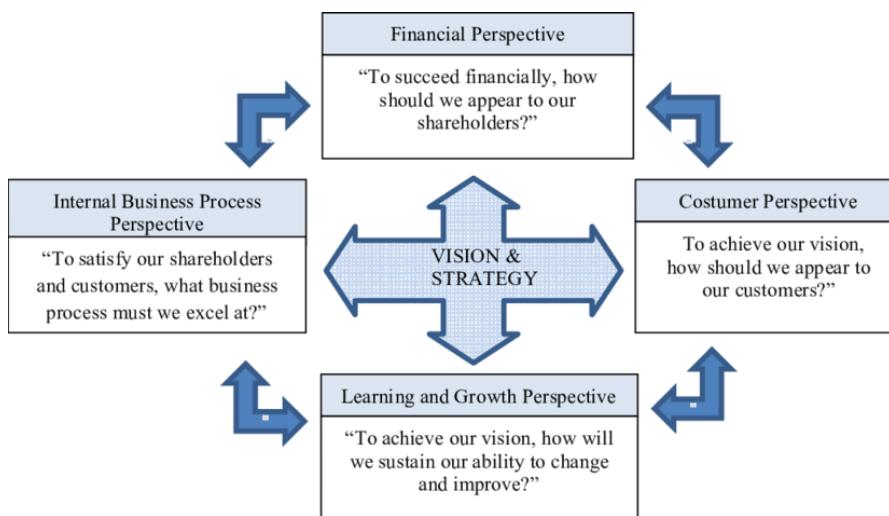
1.	INTRODUCCIÓN	3
2.	INFORMACION FUNCIONAL Y TÉCNICA	5
3.	DESARROLLO	10
4.	CASO DE USO.....	35
5.	CONCLUSIONES.....	40

1. INTRODUCCIÓN

Siguiendo la filosofía de Kaplan y Norton, los cuales definieron el conocido Cuadro de Mando Integral o Balance Scorecard, allá por el año 1990, se ha creado un ejemplo ilustrativo de dicha filosofía de hace ya unos años, con tecnología actual, en este caso concreto, Power Platform.

Como referencia histórica, el objetivo del proyecto era demostrar que los métodos de análisis que existían por aquel entonces, para la medición de los resultados obtenidos por las empresas que dependían únicamente de los indicadores contables financieros, se estaban quedando obsoletos. Es por ello, que era necesario buscar nuevos modelos de medidas para generar valor agregado a futuro, así como para también analizar el desempeño de las organizaciones desde un punto de vista que no fuese el financiero. En resumen, expresa el equilibrio entre objetivos a corto y largo plazo, entre medidas financieras y no financieras, entre indicadores provisionales e históricos y entre perspectivas de actuación internas y externas.

El Cuadro de Mando Integral es un instrumento de gestión que va desde los principios más generales a los más específicos. Está organizado en 4 perspectivas fundamentales: **Financiera, Clientes, Procesos Internos y, Aprendizaje y Crecimiento**. A su vez cada perspectiva engloba una serie de objetivos estratégicos acordes con las metas y valores de la empresa. Y por otro lado, cada objetivo estratégico engloba una serie de indicadores determinados establecidos por la propia organización. Es por ello, por lo que el Cuadro de Mando Integral se utiliza para el desarrollo e implantación de la estrategia corporativa, así como de la gestión de la visión, misión, valores esenciales, factores críticos de éxito, objetivos, medidas de actuación, metas y acciones de mejora.



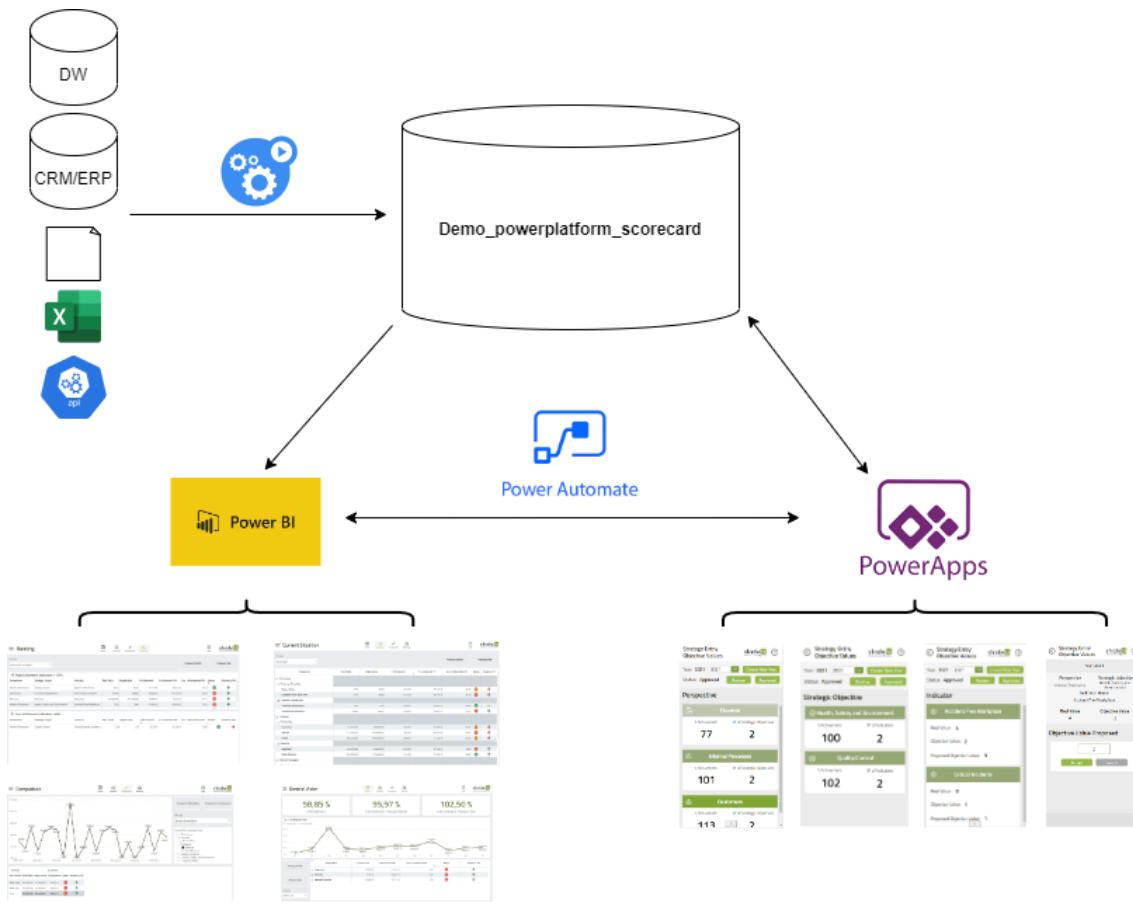
En cuanto a la aplicación desarrollada se ha visto conveniente reflejar 3 **perspectivas** del modelo de Kaplan y Norton: **Financiera** (Finances), **Clientes** (Customers) y **Procesos Internos** (Internal Processes).

A continuación, un mapa de los niveles definidos en el Balance Scorecard:



2. INFORMACION FUNCIONAL Y TÉCNICA

Arquitectura funcional:



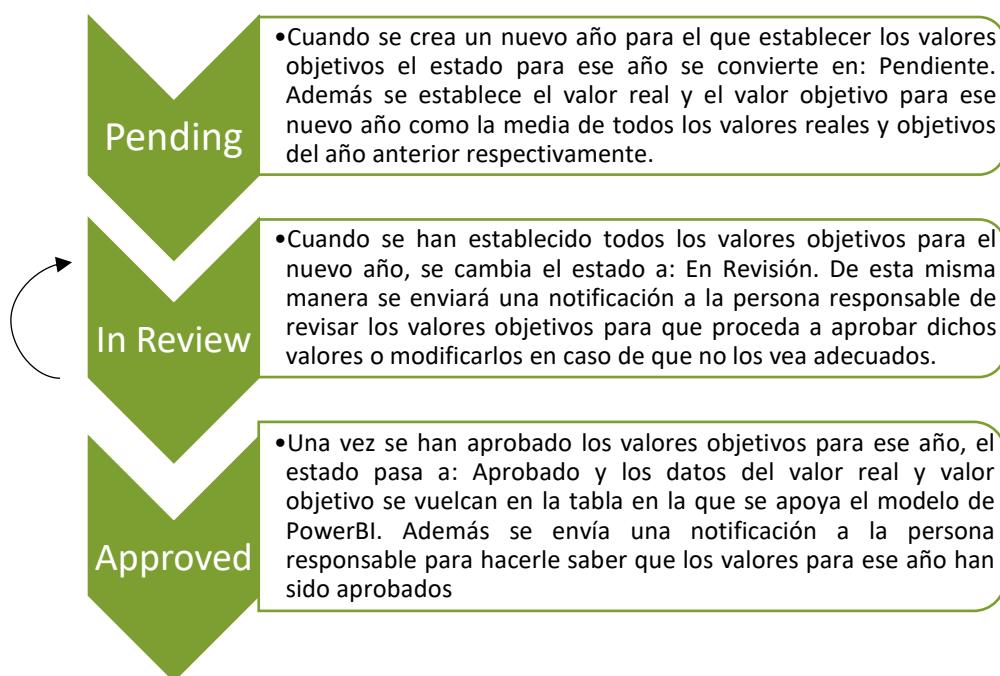
En la imagen se representa el diagrama funcional del sistema:

- El proceso ETL realiza la carga de los datos reales de los indicadores y que son calculados por fuera. Estos datos se obtienen a partir de los diferentes sistemas de la organización, como pueden ser: Data Warehouse, ERP, CRM, otros sistemas,...
- Los datos son almacenados en base de datos.
- Para el análisis de los indicadores, se utiliza Power BI en modo Direct Query, lo cual permite analizar los datos en tiempo real conforme a los valores objetivo que se establecen.
- La aplicación Power Apps permite establecer los valores objetivo de los indicadores.
- Y como pieza clave del proceso, Power Automate, permite definir el flujo operativo de seguimiento en lo que concierne al establecimiento de los valores objetivo.

Como se ha mencionado anteriormente, se han implementado varios flujos con Power Automate, los cuales se clasifican en:

- **Flujos para traspaso de datos:** Estos permiten conducir los datos entre las tablas en las que se apoya PowerBI y las tablas en las que se apoya PowerApps.
- **Flujos de lógica/operativos:** La finalidad de estos automatismos permite cambiar el estado del establecimiento de los valores objetivos propuestos para el conjunto de indicadores.

El flujo operativo para el establecimiento de los valores objetivo, gira en torno a los siguientes estados:



Una vez visto el diagrama funcional y el flujo operativo para la aplicación es importante tener en cuenta el diagrama técnico de la misma. Este diagrama va a mostrar, para PowerBI y PowerApps las tablas, relaciones entre tablas y atributos por cada tabla que se utilizan y para Power Automate el funcionamiento interno de los flujos que conectan las tablas de PowerBI con las de PowerApps y viceversa.

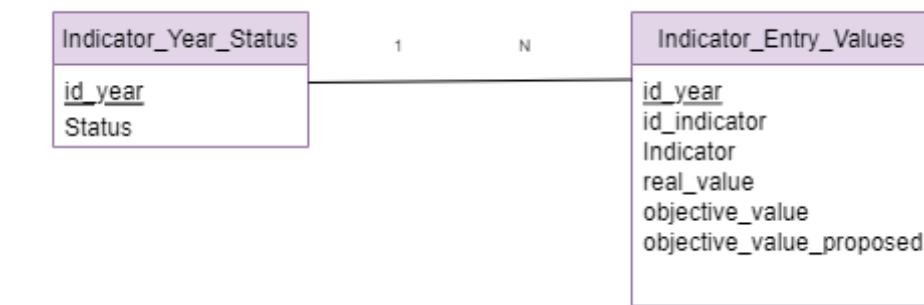
Modelo de Data Mart:



Como se muestra en la imagen, el modelo definido de PowerBI está formado a partir de tres tablas:

- **Dim Time:** Esta tabla se ha creado directamente en Power BI.
- **Fact Scorecard:** Esta tabla es la tabla de hechos del modelo y recoge los datos de medición como valor real del indicador, valor objetivo, % cumplimiento, etc....
- **Dim Indicator:** Esta tabla representa la información específica para el conjunto de indicadores de la compañía.

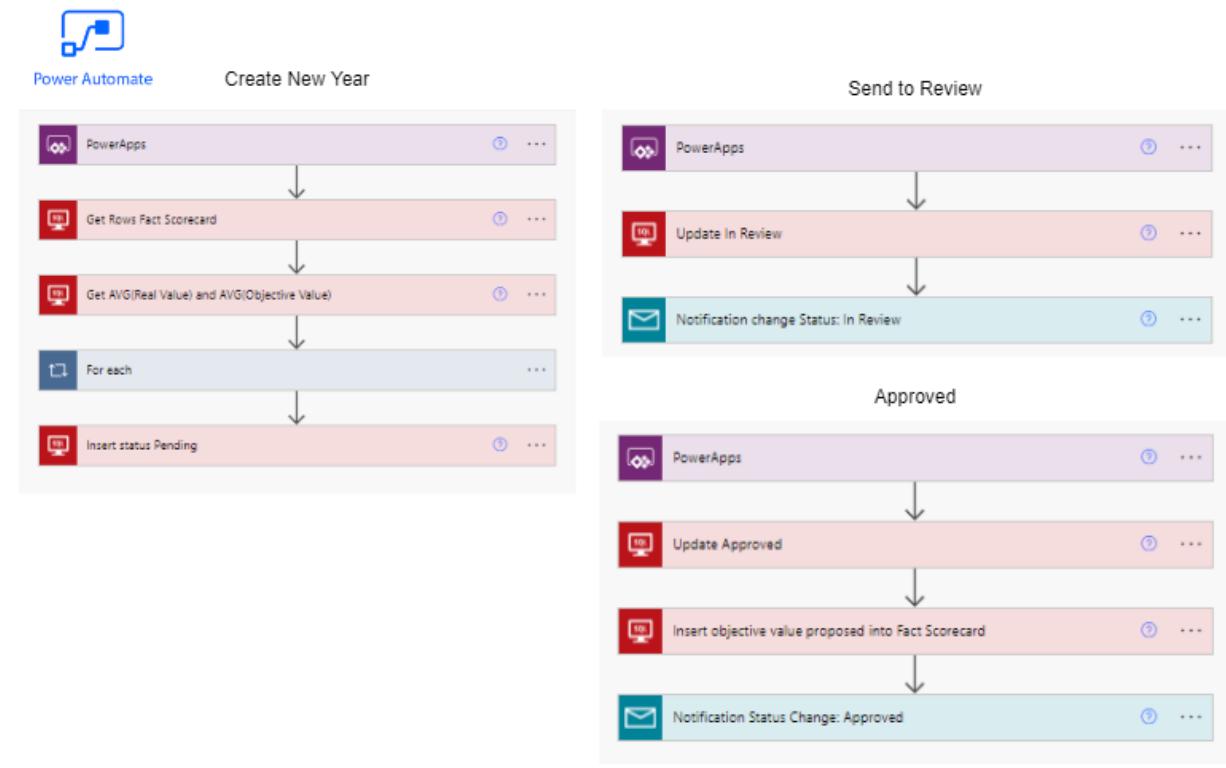
Estructuras utilizadas en la aplicación Power APP:



En este caso se ha visto conveniente la creación de dos tablas dentro de la base de datos. Estas dos tablas son:

- **Indicator_Year_Status:** Utilizada para guardar el estado en el que se encuentra el establecimiento de los valores objetivos.
- **Indicator_Entry_Objective_Values:** Utilizada para recoger por cada año e indicador los valores real y objetivo de referencia y el valor objetivo propuesto.

Diagrama técnico de flujos Power Automate:

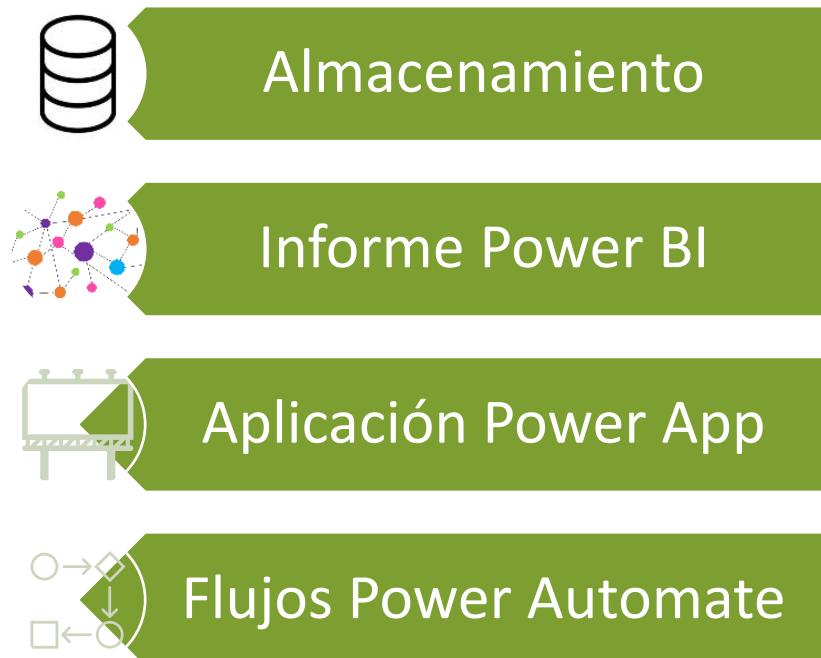


Automatismos implementados:

- **Create New Year:** A partir del último año guardado, se obtienen los datos de este para preparar los datos del nuevo año. La preparación, consiste en obtener como valores de referencia, los valores real y objetivo con vistas a que por parte del usuario se pueda establecer el nuevo valor objetivo a través de la aplicación. Los datos son cargados en la tabla `Indicator_Entry_Objective_Values`. Adicionalmente, en la tabla `Indicator_Year_Status` se crea el registro y se marca el estado “Pending”.
- **Send To Review:** Para el año seleccionado, comprueba si el estado es “Pending” y de ser así, se cambia el estado a “In Review”. Por último, se manda una notificación al responsable de revisar los valores objetivos para que supervise que todo es correcto y en caso necesario hacer modificaciones sobre los datos.
- **Approved:** Tiene una doble finalidad, en primer lugar, se cambia el estado a “Approved”, y posteriormente, se cargan los datos en el modelo de data Mart al cual está conectado el informe Power BI.

3. DESARROLLO

El desarrollo se ha realizado conforme a las siguientes etapas:



Almacenamiento

En primer lugar, se han creado las estructuras sobre las que se soportan los datos que se utilizan desde Power BI y Power Apps. Dichas estructuras están creadas en SQL Server. A continuación, las estructuras creadas para:

- **Power BI:** Las tablas son Fact Scorecard y Indicators. Concretamente la tabla Fact Scorecard representa los datos del % de cumplimiento, el valor real y el valor objetivo, para una serie de indicadores en un tiempo determinado. Y la tabla de Indicators representa los datos relacionados con los indicadores existentes en la organización. Además desde Power BI se ha creado también y a través de fórmulas DAX una tabla de tiempo.
- **Power Apps:** Las tablas son Indicator_Entry_Objective_Values y Indicator_Year_Status. La tabla Indicator_Entry_Objective_Values representa, para cada año, los datos del valor real, el valor objetivo y el valor objetivo propuesto. Y para la tabla de

Indicator_Year_Status el estado en el que se encuentran los indicadores según el flujo establecido.

Informe de Power BI

Una vez creadas las estructuras que utiliza Power BI y Power Apps, así como el modelo de Data Mart en el que se va a sustentar el informe de Power BI, se crea el informe con las siguientes visualizaciones:

- **Visión General:** Centrada en él % de cumplimiento al más alto nivel; organización, perspectivas y objetivos estratégicos, permite además de ver una evolución, y una comparativa a elegir entre el año anterior y el mes anterior.



- **Situación Actual:** En esta otra página se muestra la situación actual de los indicadores. Para cada mes-año y como se llega al nivel máximo de la estructura jerárquica, se refleja el valor real y objetivo del indicador, el % de cumplimiento, el % de cumplimiento con respecto al año pasado, el % de variación del cumplimiento con respecto al año pasado, el estado y la tendencia con respecto al año anterior.

≡ Current Situation

Period				% Achievement	% Achievement PY	Var. % Achievement PY	Status	Tendency PY
2022-jun							Previous Month	Previous Year
Perspective	Real Value	Target value	% Achievement	% Achievement PY	Var. % Achievement PY	Status	Tendency PY	
Customers								
Customer Retention								
Market Value	0,50	0,60	83,33 %	107,14 %	-0,22	🟡	⬇️	
Customer Turn-Over Rate	0,70	0,80	87,50 %	106,25 %	-0,18	🔴	⬇️	
Customer Satisfaction								
Customer Satisfaction	1,00	1,00	100,00 %	100,00 %	0,00	🟢	▬	
Unresolved Complaint	28,00	20,00	140,00 %	100,00 %	0,40	🔴	⬆️	
Finances								
Profitability								
Cash Flow	20.000,00	30.000,00	66,67 %	125,00 %	-0,47	🟡	⬇️	
EBITDA	115.000,00	190.000,00	60,53 %	104,50 %	-0,42	🔴	⬇️	
Profit	80.000,00	100.000,00	80,00 %	97,33 %	-0,18	🟡	⬇️	
Revenue								
Expenses	45.000,00	43.430,00	103,62 %	101,43 %	0,02	🔴	⬆️	
Sales Revenue	160.000,00	175.000,00	91,43 %	93,33 %	-0,02	🟢	⬆️	
Internal Processes								

- **Comparación:** Esta página como su propio nombre indica, permite hacer comparativa de indicadores para así ver su correlación en el tiempo, bien sea atendiendo al valor real o al % de cumplimiento. Si observamos la captura inferior, se aprecia que el selector de indicadores permite seleccionar indistintamente indicadores, pudiendo seleccionar un objetivo estratégico o perspectiva. Todo depende de lo que nosotros podamos llegar a apreciar. Además, en la parte inferior, y para cada indicador y periodo que se selecciona se muestra el valor real y objetivo, el % de cumplimiento, el estado y la tendencia con respecto al mes anterior y la media del total para todos los períodos seleccionados.

≡ Comparison



- **Ranking:** Esta última página permite centrar el análisis en aquellos indicadores tomando en consideración el % de Cumplimiento, que estén fuera de ciertos umbrales, bien sea por encima (100%) o por debajo (60%).

The screenshot shows a dashboard titled "Ranking". At the top, there are four tabs: Vision, Indicator, Comparison, and Ranking (which is selected). Below the tabs are buttons for Period (Periodo), Selección múltiple (Multiple Selection), Previous Month, and Previous Year. The main area contains two tables.

High Achievement indicators > 100%

Perspective	Strategic Target	Indicator	Real Value	Target value	% Achievement	% Achievement PM	Var. % Achievement PM	Status	Tendency PM
Internal Processes	Quality Control	Right the First Time	0,93	0,90	102,78 %	83,33 %	0,23	●	↑
Customers	Customer Satisfaction	Unresolved Complaint	53,00	30,00	195,00 %	131,25 %	0,49	●	↑
Finances	Revenue	Expenses	40.500,00	37.465,00	108,95 %	93,02 %	0,17	●	↑
Internal Processes	Health, Safety and Environment	Accident Free Workplace	3,50	2,00	175,00 %	100,00 %	0,75	●	↑

Low Achievement indicators < 60%

Perspective	Strategic Target	Indicator	Real Value	Target value	% Achievement	% Achievement PM	Var. % Achievement PM	Status	Tendency PM
Internal Processes	Quality Control	Serious Service Incidents	1,00	1,00	50,00 %	100,00 %	-0,50	●	↓

Adicionalmente se han creado otras dos páginas.

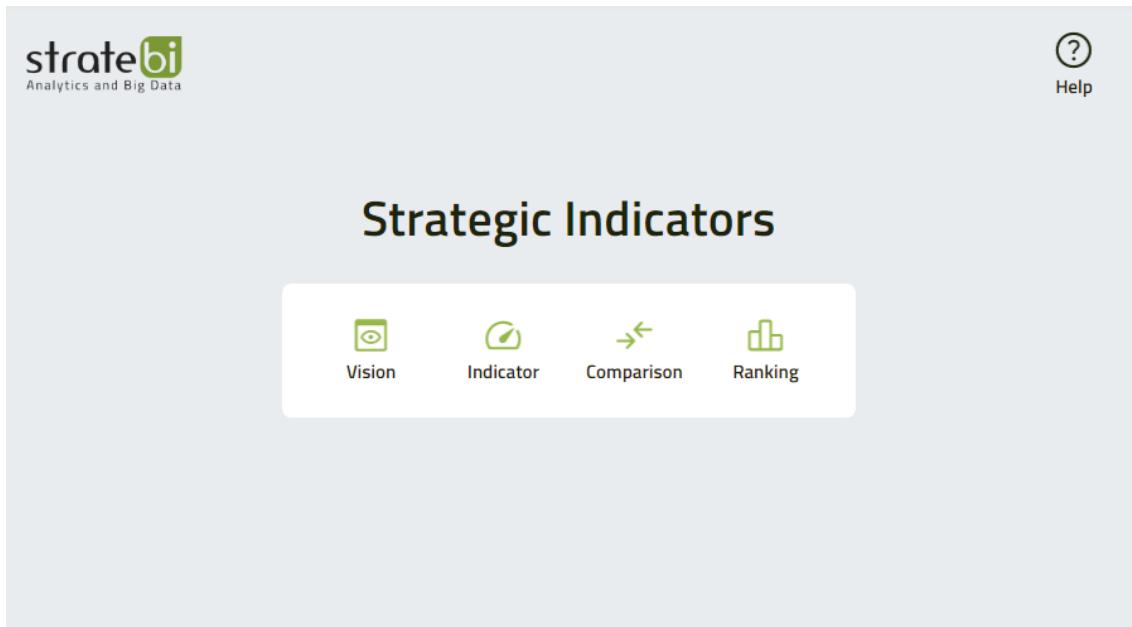
- **Aplicación:** La primera de ellas muestra la aplicación realizada en PowerApps para utilizar la misma dentro del mismo entorno PowerBI:

The screenshot shows a PowerApp application titled "Strategy Entry Objective Values". It has a header with "Year: 2021" and "Create New Year" buttons, and status buttons for "Approved" (green) and "Review" (orange).

The main area displays three cards:

- Finances:** % Achievement: 77, N° of Strategic Objectives: 2
- Internal Processes:** % Achievement: 101, N° of Strategic Objectives: 2
- Customers:** % Achievement: 113, N° of Strategic Objectives: 2

- **Menú:** Para crear una mejor sensación, esta segunda página contiene el menú de navegación entre las diferentes páginas, y el cual está dispuesto en la parte superior en cada una de las páginas del informe.



A continuación, se muestran las diferentes medidas creadas en Power BI en la tabla Fact Scorecard:

Nombre Medida	Descripción	Formula DAX
%Achievement	Representa el % de cumplimiento del BSC	<pre>if(ISINSCOPE('Dim Indicator'[Indicator]), AVERAGE('Fact_Scorecard'[achievement_indicator]) ,if(ISINSCOPE('Dim Indicator'[Strategic_target]), average('Fact_Scorecard'[achievement_target]) ,if(ISINSCOPE('Dim Indicator'[Perspective]),AVERAGE ('Fact_Scorecard'[achievement_perspective]),AVERAGE('Fact_Scorecard'[achievement_scorecard]))))</pre>
%Achievement PM	Representa el % de cumplimiento del BSC del mes inmediatamente anterior al seleccionado	<code>CALCULATE([%Achievement],PREVIOUSMONTH('Dim Time'[Date]))</code>
%Achievement PY	Representa el % de cumplimiento del BSC del año inmediatamente anterior al seleccionado	<code>CALCULATE([%Achievement],SAMEPERIODLASTYEAR('Dim Time'[Date]))</code>
Lower_threshold_status	Representa el estado del indicador	<code>sum('Dim Indicator'[Lower_threshold_state])/100</code>

Multiplicator	Intercambiará el valor Max por 1 y el valor Min por -1 dependiendo del valor Goal de la tabla Dim Indicator	<pre>SWITCH (SELECTEDVALUE ('Dim Indicator'[Goal]), "Max", 1, "Min", -1)</pre>
Real Value	Muestra el valor real de la tabla Fact Scorecard	<pre>if(ISINSCOPE('Dim Indicator'[Indicator]),SWITCH (SELECTEDVALUE ('Dim Indicator'[Aggregation]), "Sum", SUM ('Fact_Scorecard'[Real_Value]), "Avg", AVERAGE('Fact_Scorecard'[Real_Value]), "Count", COUNT('Fact_Scorecard'[Real_Value]), "Max", MAX ('Fact_Scorecard'[Real_Value]), "Min", MIN ('Fact_Scorecard'[Real_Value])),0)</pre>
Real Value PM	Muestra el valor real de la tabla Fact Scorecard para el mes	<pre>CALCULATE([Real Value],PREVIOUSMONTH('Dim Time'[Date]))</pre>

	inmediatamente anterior al seleccionado	
Real Value PY	Muestra el valor real de la tabla Fact Scorecard para el año inmediatamente anterior al seleccionado	<code>CALCULATE([Real Value],SAMEPERIODLASTYEAR('Dim Time'[Date]))</code>
Status	Muestra el estado del indicador	<code>If([Multiplicator]=1,If([%Achievement]*[Multiplicator] < [Lower_threshold_status]*[Multiplicator], -1, if([%Achievement]*[Multiplicator]> [Upper_threshold_status]*[Multiplicator],1,0)), If([%Achievement]*[Multiplicator]>[Upper_threshold_status]*[Multiplicator],1, If([%Achievement]*[Multiplicator]>[Lower_threshold_status]*[Multiplicator],0,-1)))</code>
Target Value	Muestra el valor objetivo de la tabla Fact Scorecard	<code>if(ISINSCOPE('Dim Indicator'[Indicator]),SWITCH (SELECTEDVALUE ('Dim Indicator'[Aggregation]), "Sum", SUM ('Fact_Scorecard'[Target_Value]), "Avg", AVERAGE('Fact_Scorecard'[Target_Value]), "Count", COUNT('Fact_Scorecard'[Target_Value]), "Max", MAX ('Fact_Scorecard'[Target_Value]), "Min", MIN ('Fact_Scorecard'[Target_Value])),0)</code>

Target Value PM	Muestra el valor objetivo de la tabla Fact Scorecard para el mes inmediatamente anterior al seleccionado	<code>CALCULATE([Target Value],PREVIOUSMONTH('Dim Time'[Date]))</code>
Target Value PY	Muestra el valor real de la tabla Fact Scorecard para el año inmediatamente anterior al seleccionado	<code>CALCULATE([Target Value],SAMEPERIODLASTYEAR('Dim Time'[Date]))</code>
Tendency PY	Muestra la tendencia del indicador para el año inmediatamente anterior al seleccionado	<code>if([%Achievement]>[%Achievement PY],1,if([%Achievement]<[%Achievement PY],-1,0))</code>
Tendency PM	Muestra la tendencia del indicador para el mes inmediatamente anterior al seleccionado	<code>if([%Achievement]>[%Achievement PM],1,if([%Achievement]<[%Achievement PM],-1,0))</code>

Upper_threshold_status	Representa el estado del indicador	<code>sum('Dim Indicator'[Upper_threshold_state]) /100</code>
Var. % Achievement PY	Representa el % de variación entre el % de cumplimiento de un año y el año anterior	<code>([%Achievement]/[%Achievement PY])-1</code>
Var. % Achievement PM	Representa el % de variación entre el % de cumplimiento de un mes y el mes anterior	<code>([%Achievement]/[%Achievement PM])-1</code>

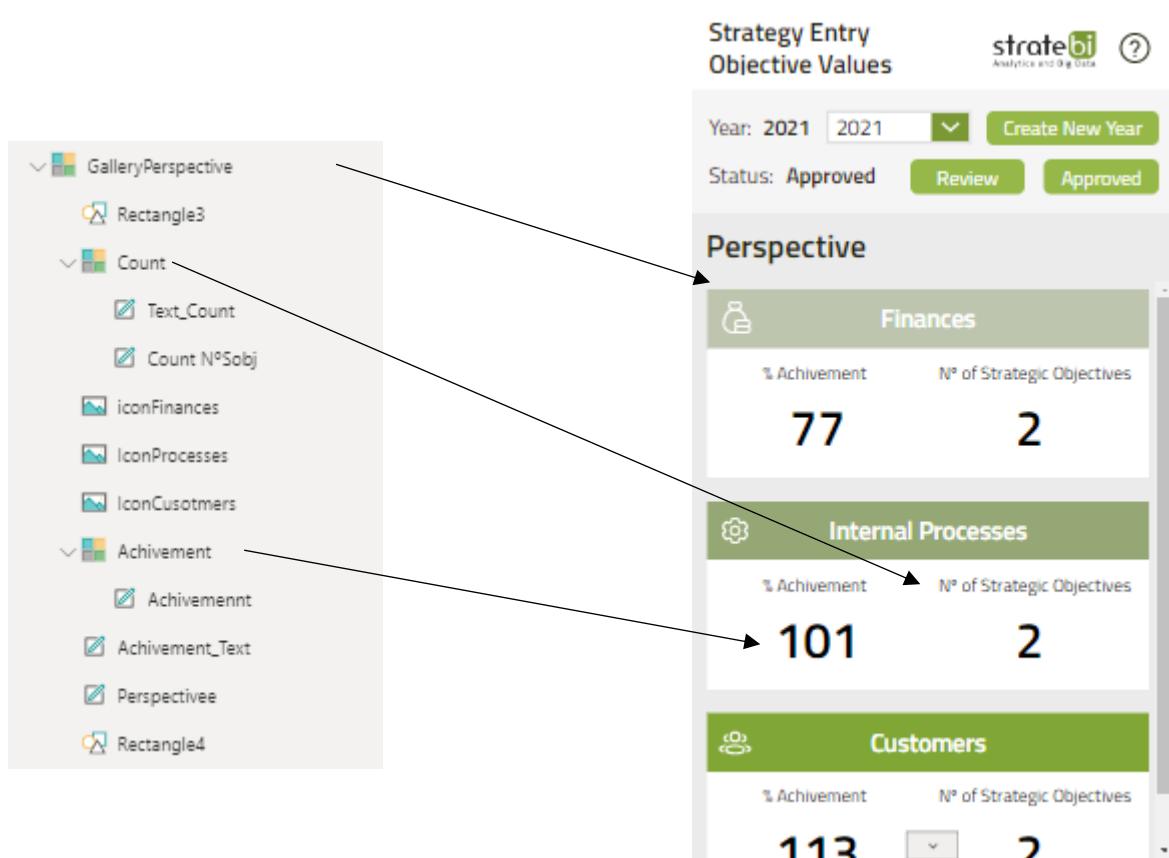
Aplicación Power Apps

Una vez desarrollado el informe Power BI, el siguiente paso es comenzar con la implementación de la aplicación en Power Apps.

Dentro de Power Apps se crea una conexión a SQL Server para poder utilizar las estructuras de datos definidas previamente.

Una vez creada la conexión se crean las diferentes pantallas en las que se divide la aplicación:

- La primera pantalla muestra todas las perspectivas que recoge la organización. Para cada perspectiva se muestra además, su % de cumplimiento y el número de objetivos estratégicos que tiene asociados. Esta página se ha programado con la ayuda de tres galerías, en las que la galería padre es la galería que muestra la perspectiva y las galerías hijos son las galerías que muestran el % de cumplimiento y el número de objetivos estratégicos que tiene asociados.



De esta manera las galerías para esta pantalla se han programado de la siguiente manera:

- La galería que muestra las perspectivas se ha programado para que partiendo de la tabla Indicator_Entry_Objective_Values, se recuperen los valores del objetivo estratégico y la perspectiva. Estos valores pertenecen a la tabla Indicators y se recuperan con la función LookUp. Una vez recuperadas dichos registros se añaden a la tabla Indicator_Entry_Objective_Values temporalmente y se procede a filtrar por el valor que se necesita mostrar, en este caso la perspectiva. Como al hacer la función LookUp por el id del indicador se devuelven las perspectivas repetidas es necesario aplicar la función GroupBy para agrupar los valores y que de esta manera se muestren una sola vez.

```
GroupBy(
    Filter(
        AddColumns(
            Indicator_Entry_Objective__Values;
            "Strategic__Target";
            LookUp(
                Indicators;
                Id = Indicator_Entry_Objective__Values[@id_indicator];
                Strategic_target
            );
            "Perspective";
            LookUp(
                Indicators;
                Id = Indicator_Entry_Objective__Values[@id_indicator];
                Perspective
            );
            Perspective
        );
        "Perspective";
        "Perspe"
    )
)
```

- Una vez configurada la galería para mostrar el valor que se devuelve por pantalla, es necesario programar una etiqueta de texto que nos indique exactamente que registros de la tabla que se devuelven en dicha galería se quieren mostrar. En este caso el nombre de las perspectivas.

`ThisItem.Perspective`

- La galería que muestra el porcentaje de cumplimiento asociado a cada perspectiva se ha programado para que partiendo de la tabla Indicator_Entry_Objective_Values se recupere el valor del porcentaje de cumplimiento de la perspectiva y el nombre de la perspectiva. Esto se consigue con la función LookUp. Una vez recuperados, se añaden temporalmente a la tabla Indicator_Entry_Objective_Values con la función AddColumns. El siguiente paso sería filtrar los datos para sacar el % de cumplimiento de cada perspectiva, pero en este caso acorde a una perspectiva en específico.

De esta manera, la galería mostrará para cada perspectiva su porcentaje de cumplimiento asociado. Como al hacer la función LookUp por el id del indicador se devuelven las perspectivas repetidas es necesario aplicar la función GroupBy para agrupar los valores y que de esta manera se muestren una sola vez.

```

    GroupBy(
        Filter(
            AddColumns(
                Indicator_Entry_Objective__Values;
                "Achievement_Pers";
                LookUp(
                    Fact_Scorecard;
                    Id_indicator = Indicator_Entry_Objective__Values[@id_indicator];
                    achievement_perspective
                );
                "Perspective";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    Perspective
                )
            );
            Achievement_Pers;
            Perspective = Perspective.Text;
            id_year = YearDropDown.Selected.Result
        );
        "Perspective";
        "Achievement_Pers";
        "Pers"
    )
)

```

- Una vez configurada la galería, para mostrar el valor que se devuelve por pantalla, es necesario programar una etiqueta de texto que nos indique exactamente que registros de la tabla que se devuelven en dicha galería se quieren mostrar. En este caso, el porcentaje de cumplimiento asociado a cada perspectiva. Se ha utilizado la función Trunc, para truncar los valores de los decimales y el resultado se ha multiplicado por 100 para que resulte en forma de porcentaje.

`Trunc((ThisItem.Achievement_Pers)*100)`

- La galería que muestra el número de objetivos estratégicos asociados a cada perspectiva se programa únicamente estableciendo como origen la tabla principal en la que se apoya Power Apps

`Indicator_Entry_Objective__Values`

- La etiqueta de la galería se ha programado para que partiendo de la tabla Indicator_Entry_Objective_Values, se recuperen los valores del objetivo estratégico y la perspectiva. Estos valores pertenecen a la tabla Indicators y se recuperan con la función LookUp. Una vez recuperadas se añaden a la tabla Indicator_Entry_Objective_Values temporalmente y se procede a filtrar por el valor que se necesita mostrar, en este caso que la perspectiva que se devuelve sea igual al valor de la etiqueta de texto de la perspectiva que se programó en la galería perspectiva. Como al hacer la función LookUp por el id del indicador se devuelven datos repetidos es necesario aplicar la función GroupBy para agrupar los valores y que de esta manera se muestren una sola vez. Por último, la función CountIf va a ser la encargada de contar el número de objetivos estratégicos para los que el nombre de la perspectiva es igual al valor de la etiqueta perspectiva en la galería perspectiva. De esta manera la galería mostrará el número de objetivos estratégicos asociados a dicha perspectiva.

```

CountIf(
    GroupBy(
        Filter(
            AddColumns(
                Indicator_Entry_Objective__Values;
                "Strategic__Target";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    Strategic_target
                );
                "Perspective";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    Perspective
                )
            );
            Perspective = Perspectivee.Text
        );
        "Strategic__Target";
        "Perspective";
        "Perspe_A"
    );
    Perspective = Perspectivee.Text
)

```

El siguiente paso es programar las tarjetas que hay dentro de cada galería. Estas tarjetas se han creado en base a dos rectángulos. Sobre el rectángulo más grande, se incluyen los datos y en el rectángulo más pequeño se muestra el nombre de la perspectiva. Para este último se ha programado que se rellene de un color diferente dependiendo del nombre de la misma.

```

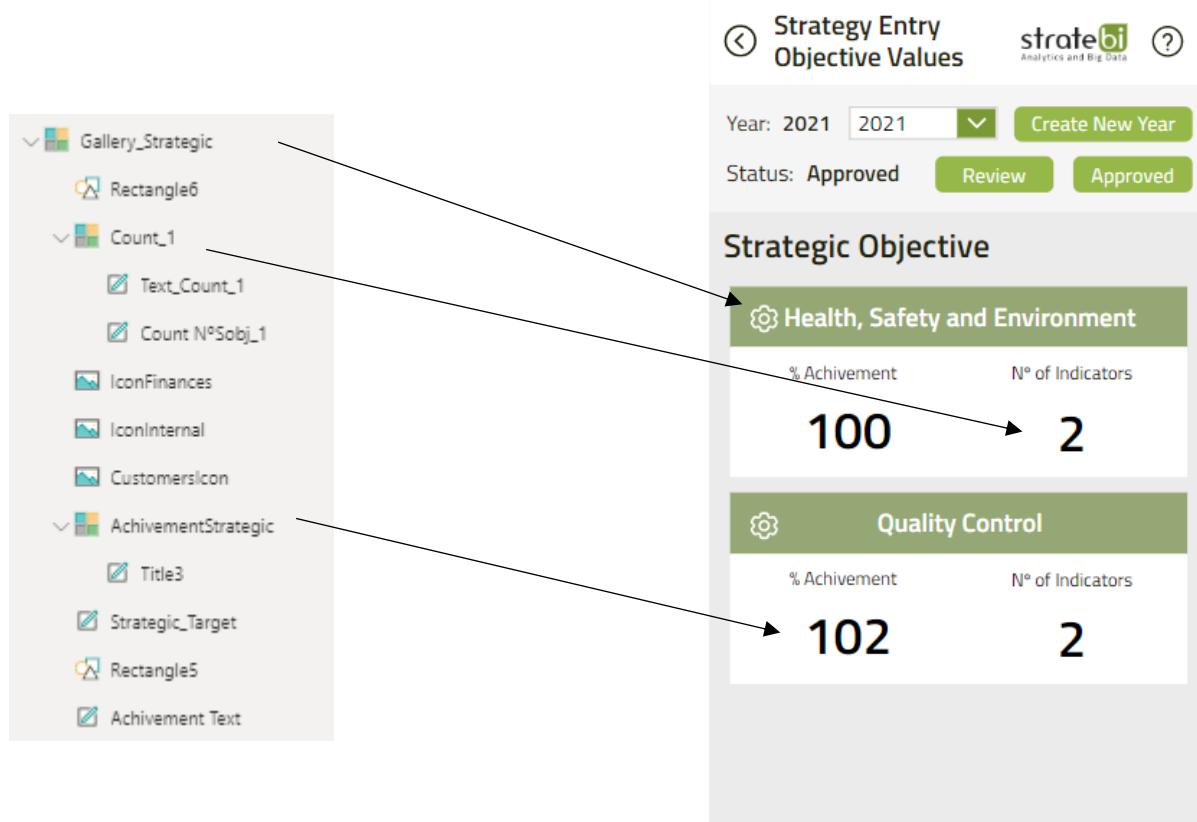
Switch(Perspectivee.Text; "Finances"; RGBA(189;197;175;1);
      "Internal Processes"; RGBA(149;167;116;1); "Customers"; RGBA(128;166;54;1))

```

Es Importante mencionar que las tarjetas se han programado para que cuando se pinche sobre ellas, se pueda navegar y bajar un nivel dentro de la estructura, concretamente al nivel de objetivo estratégico.

Navigate('Strategic Objective Form';Cover)

- La segunda pantalla muestra los objetivos estratégicos asociados a la perspectiva seleccionada en la pantalla anterior. Se recogerá además el % de cumplimiento y el número de indicadores que tiene asociados. De manera análoga, se ha programado con la ayuda de tres galerías, donde la galería padre es la galería que muestra los objetivos estratégicos y las galerías hijos son las galerías que muestran el % de cumplimiento y el número de indicadores que tiene asociados.



De esta manera las galerías para esta pantalla se han programado de la siguiente forma:

- La galería que muestra los objetivos estratégicos se ha programado para que partiendo de la tabla Indicator_Entry_Objective_Values, se recuperen los valores del objetivo estratégico y la perspectiva. Estos valores pertenecen a la tabla Indicators y se recuperan con la función LookUp. Una vez recuperados se añaden a la tabla Indicator_Entry_Objective_Values temporalmente. El siguiente paso sería filtrar por el valor que se necesita para mostrar, en este caso que la columna perspectiva que se recupera sea igual al valor de la etiqueta de la galería perspectiva que se había programado en la pantalla anterior. Como al hacer la función LookUp por el id del indicador se devuelven por cada perspectiva los objetivos estratégicos repetidos, es necesario aplicar la función GroupBy para agrupar los valores por el objetivo estratégico y que de esta manera se muestren una sola vez.

```

    GroupBy(
        Filter(
            AddColumns(
                Indicator_Entry_Objective__Values;
                "Strategic_Target";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    Strategic_target
                );
                "Perspectiv";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    Perspective
                )
            );
            Perspectiv = Perspectivee.Text
        );
        "strategic_Target";
        "Target_S"
    )
)

```

- Una vez configurada la galería, para mostrar el valor que se devuelve por pantalla es necesario programar una etiqueta de texto que nos indique exactamente que registros de la tabla que se devuelven en dicha galería se quieren mostrar. En este caso el nombre del objetivo estratégico asociado a la perspectiva que se ha seleccionado previamente.

`ThisItem.Strategic_Target`

- La galería que muestra el porcentaje de cumplimiento asociado a cada objetivo estratégico sigue un procedimiento es similar. En este caso, se recupera el valor del porcentaje de cumplimiento del propio objetivo estratégico y el nombre de la perspectiva con la función LookUp y se añaden temporalmente dichas columnas a la tabla Indicator_Entry_Objective__Values con la función AddColumns. Ahora, se filtran los datos para sacar el % de cumplimiento de cada objetivo estratégico, pero acorde a un objetivo estratégico específico. De esta manera la galería mostrará por cada objetivo estratégico su porcentaje de cumplimiento asociado. Como al hacer la función LookUp por el id del indicador se devuelven los objetivos estratégicos repetidos es necesario aplicar la función GroupBy para agrupar los valores y que de esta manera se muestren una sola vez.

```

    GroupBy(
        Filter(
            AddColumns(
                Indicator_Entry_Objective__Values;
                "Achievement_targ";
                LookUp(
                    Fact_Scorecard;
                    Id_indicator = Indicator_Entry_Objective__Values[@id_indicator];
                    achievement_target
                );
                "Strategic";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    strategic_target
                )
            );
            Achievement_targ;
            Strategic = Strategic_Target.Text;
            id_year = YearDropDown_1.Selected.Result
        );
        "Strategic";
        "Achievement_targ";
        "Pers"
    )
)

```

- Una vez configurada la galería, y para mostrar el valor que se devuelve por pantalla, es necesario programar una etiqueta de texto que nos indique exactamente que registros de la tabla que se devuelven en dicha galería se quieren mostrar. En este caso, el porcentaje de cumplimiento asociado a cada objetivo estratégico. Se ha utilizado la función Trunc, para truncar los valores de los decimales y el resultado se ha multiplicado por 100 para que resulte en forma de porcentaje.

`Trunc((ThisItem.Achievement_targ)*100)`

- La galería que muestra el número de indicadores asociados a cada objetivo estratégico se programa únicamente estableciendo como origen la tabla principal en la que se apoya Power Apps

Indicator_Entry_Objective__Values

- La etiqueta de la galería se ha programado para que partiendo de la tabla Indicator_Entry_Objective_Values, se recuperen los valores del objetivo estratégico y los indicadores. Estos valores pertenecen a la tabla Indicators y se recuperan con la función LookUp. Una vez recuperadas se añaden a la tabla Indicator_Entry_Objective_Values temporalmente y se procede a filtrar por el valor que se necesita para mostrar. En este caso, que el objetivo estratégico que se devuelve sea igual al valor de la etiqueta de texto que se programó en la galería objetivo estratégico. Como al recuperar los objetivos estratégicos por el id del indicator se devuelven datos repetidos es necesario aplicar la función GroupBy para agrupar los valores y que de esta manera se muestren una sola vez. Por último la función CountIf va a ser la encargada de contar el número de indicadores para los que el nombre del objetivo estratégico es igual al valor de la etiqueta objetivo estratégico en dicha galería.

```

CountIf(
    GroupBy(
        Filter(
            AddColumns(
                Indicator_Entry_Objective__Values;
                "Strategic__Target";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    Strategic_target
                );
                "Indicators";
                LookUp(
                    Indicators;
                    Id = Indicator_Entry_Objective__Values[@id_indicator];
                    id_indicator
                )
            );
            Strategic__Target = Strategic_Target.Text
        );
        "Strategic__Target";
        "Indicators";
        "Count_ST"
    );
    Strategic__Target = Strategic_Target.Text
)

```

El siguiente paso será programar las tarjetas que hay dentro de cada galería. Estas tarjetas se han creado en base a dos rectángulos. Dentro del rectángulo más grande, se incluirán los datos, y en el rectángulo más pequeño se muestra el nombre del objetivo estratégico. Para este último se ha programado que se rellene de un color diferente dependiendo del nombre de la perspectiva.

```
Switch(Perspectivee.Text; "Finances"; RGBA(189;197;175;1);
      "Internal Processes"; RGBA(149;167;116;1); "Customers"; RGBA(128;166;54;1))
```

Importante mencionar que las tarjetas se han programado para que cuando se pinche sobre ellas puedas navegar a la siguiente pantalla con el siguiente código

```
Navigate('Indicator Form';Cover)
```

- La tercera pantalla mostrará los indicadores asociados a cada objetivo estratégico seleccionado. En esta página, a diferencia de las anteriores se recoge el valor real que toma el indicador, el valor objetivo y el valor objetivo propuesto que se introducirá manualmente en la siguiente pantalla. De manera análoga, se ha programado con la ayuda de cuatro galerías. La galería padre es la galería que muestra los indicadores y las galerías hijos son las galerías que muestran el valor real, el valor objetivo y el valor objetivo propuesto.



The diagram illustrates the structure of the Stratebi object library and its connection to the 'Strategy Entry Objective Values' application interface.

Stratebi Object Library:

- GalleryIndicator:** Contains items like IconCustomer, InconFinan, IconProce, Rectangle9, Proposed_Value, Real_Value, Objective_Value, Proposed_Text, Real_Text, Objective_Text, Indicator_Name, and Rectangle8.
- GalleryObjective:** Contains the ObjectiveValue item.
- GalleryOProposed:** Contains the Proposed_Value item.
- GalleryVReal:** Contains the Real_Value item.

Strategy Entry Objective Values Application:

This application displays two indicators:

- Accident Free Workplace:**
 - Real Value: 4
 - Objective Value: 2
 - Proposed Objective Value: 3
- Critical Incidents:**
 - Real Value: 0
 - Objective Value: 1
 - Proposed Objective Value: 1

Arrows from the object library items point to the corresponding fields in the application interface, indicating how each library item maps to a specific data component or UI element.

De esta manera las galerías para esta pantalla se han programado de la siguiente forma:

- La galería que muestra los indicadores se ha programado para que partiendo de la tabla Indicator_Entry_Objective_Values, se recuperen los valores del objetivo estratégico y la perspectiva. Estos valores pertenecen a la tabla Indicators y se recuperan con la función LookUp. Una vez recuperadas dichos registros se añaden dichas columnas a la tabla Indicator_Entry_Objective_Values temporalmente y se procede a filtrar por el valor que se necesita para mostrar, en este caso se necesitan mostrar los indicadores para los que el objetivo estratégico y la perspectiva sean iguales al valor de las etiquetas de las galerías de objetivo estratégico y perspectiva programadas en las pantallas anteriores. Además, también es necesario seleccionar únicamente los valores de un año determinado. Por último, como al hacer el LookUp por el id del indicador se devuelven valores repetidos es necesario aplicar la función GroupBy para agrupar los valores y que de esta manera se muestren una sola vez.

```
GroupBy(
    Filter(
        AddColumns(
            Indicator_Entry_Objective__Values;
            "Strategic__Target";
            Lookup(
                Indicators;
                Id = Indicator_Entry_Objective__Values[@id_indicator];
                Strategic_target
            );
            "Perspective";
            Lookup(
                Indicators;
                Id = Indicator_Entry_Objective__Values[@id_indicator];
                Perspective
            )
        );
        Strategic__Target = Strategic__Target.Text && Perspective = Perspective.Text && id_year = YearDropDown_2.Selected.Result
    );
    "indicator";
    "Indi"
)
```

- Una vez configurada la galería, para mostrar el valor que se devuelve por pantalla, es necesario programar una etiqueta de texto que nos indique exactamente que registros de la tabla que se devuelven en dicha galería se quieren mostrar. En este caso el nombre de los indicadores.

`ThisItem.indicator`

- Las galerías que muestran el valor real, el valor objetivo y el valor objetivo propuesto se han programado para que partiendo de la tabla Indicator_Entry_Objective_Values, se recuperen los datos del valor real, el valor objetivo y el valor objetivo propuesto por año seleccionado y nombre del indicador. Por último, una vez se hayan filtrado los valores es importante establecer la función Distinct para que te aparezcan valores únicos por cada indicador.

```
Distinct(
    Filter(
        Indicator_Entry_Objective__Values;
        id_year = YearDropDown_2.Selected.Result && indicator = Indicator_Name.Text
    );
    real_value
).Result

Distinct(
    Filter(
        Indicator_Entry_Objective__Values;
        id_year = YearDropDown_2.Selected.Result && indicator=Indicator_Name.Text
    );
    objective_value_proposed
).Result

Distinct(
    Filter(
        Indicator_Entry_Objective__Values;
        id_year = YearDropDown_2.Selected.Result && indicator=Indicator_Name.Text
    );
    objective_value
).Result
```

- Una vez configurada la galería será necesario programar las tres etiquetas de texto, para que en cada galería se muestre el valor real, el valor objetivo o el valor objetivo propuesto. Al anteriormente haber utilizado la función Distinct, esta, siempre devuelve un tipo de datos .Result, que es el que hay poner en la etiqueta de datos de cada galería respectivamente.

ThisItem.Result

El siguiente paso será programar las tarjetas que hay dentro de cada galería. Estas tarjetas se han creado en base a dos rectángulos. Dentro del rectángulo más grande, se incluirán los datos y dentro del rectángulo más pequeño, se mostrará el nombre del indicador. Para este último se ha programado que se rellene de un color diferente dependiendo del nombre de la perspectiva.

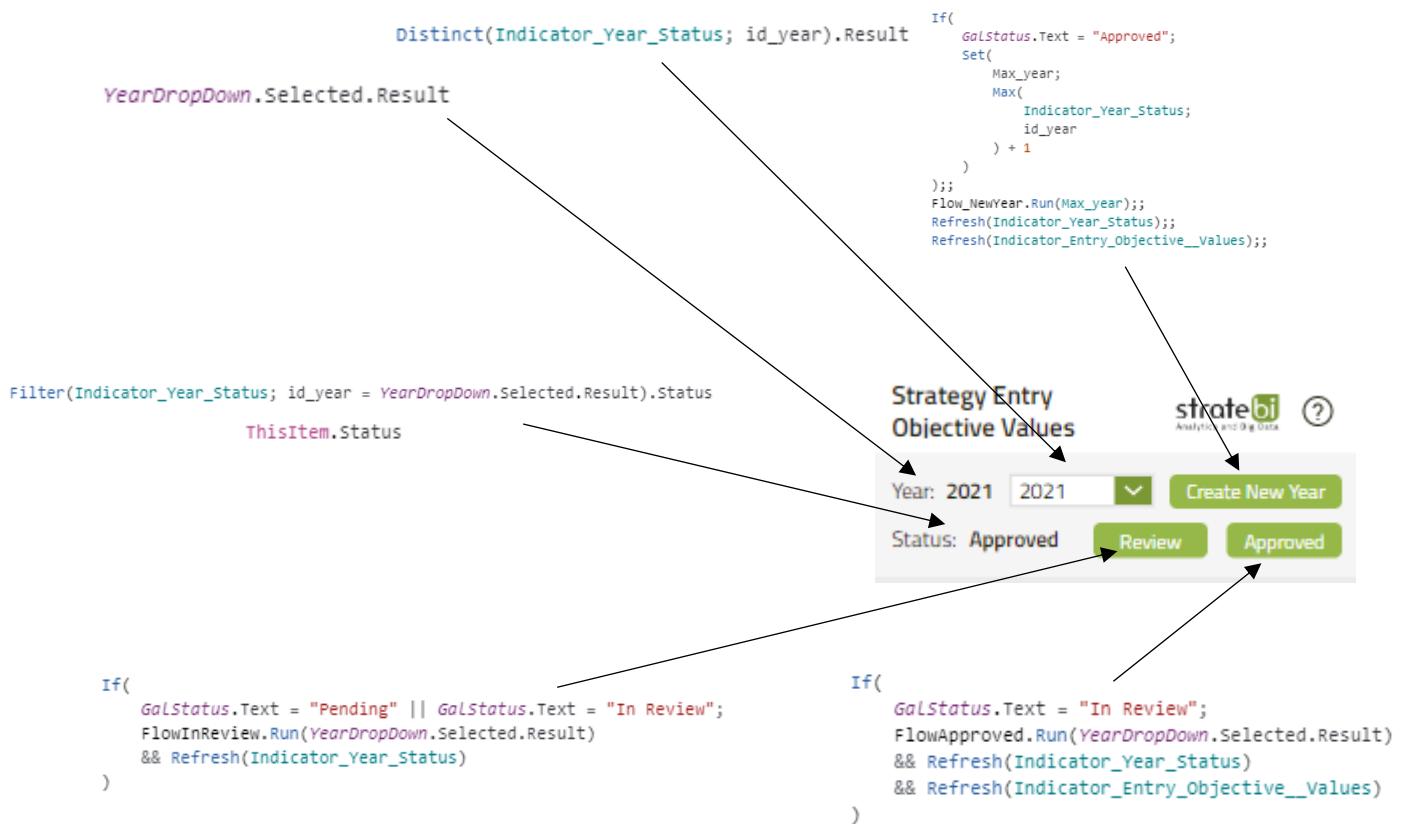
```
Switch(Perspectivee.Text; "Finances"; RGBA(189;197;175;1);
       "Internal Processes"; RGBA(149;167;116;1); "Customers"; RGBA(128;166;54;1))
```

Importante mencionar que las tarjetas se han programado para que cuando se pinche sobre ellas puedas navegar a la siguiente pantalla. En este caso además de navegar hacia la siguiente pantalla se indica que se va a editar un formulario.

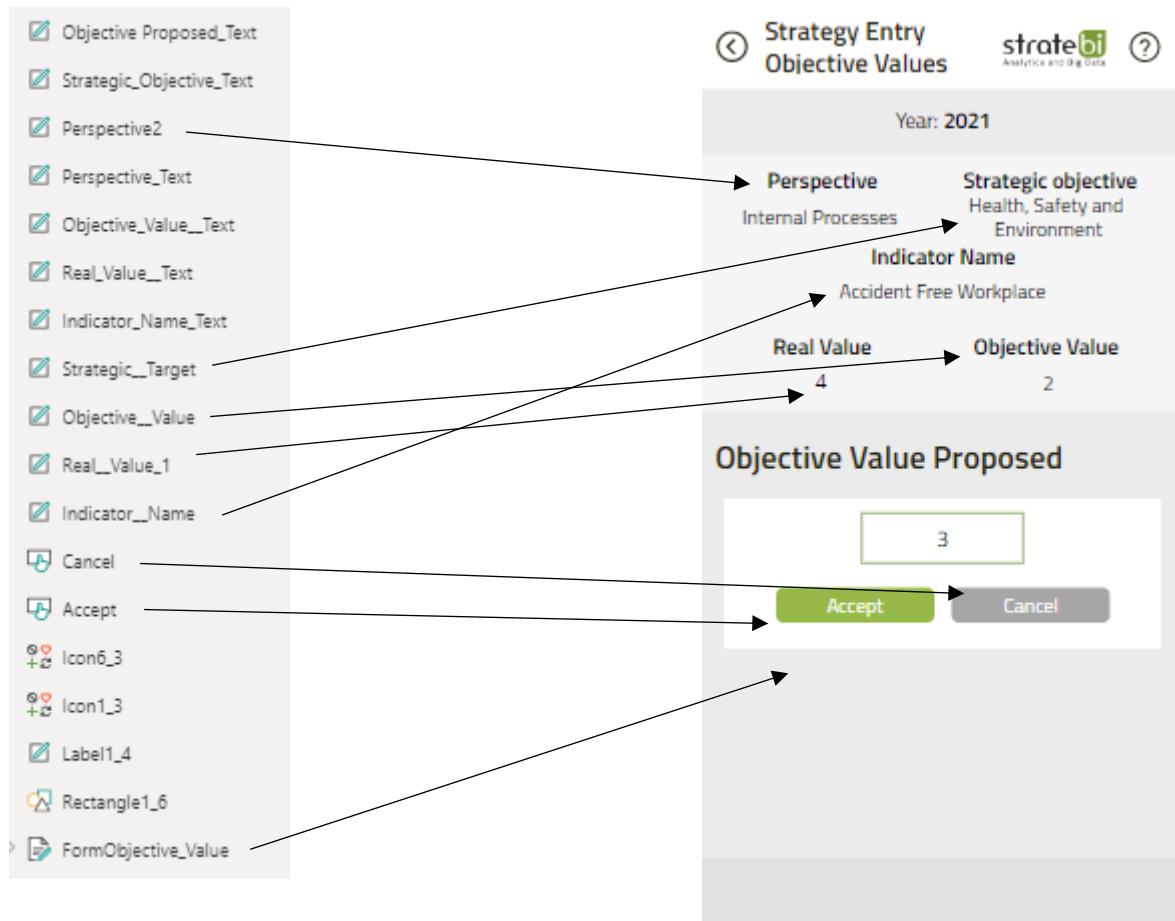
EditForm(FormObjective_Value);;Navigate(FormScreen; Cover)

Todas las pantallas mencionadas anteriormente comparten una serie de objetos visuales:

- La etiqueta que muestra el año seleccionado.
- La galería que muestra el estado de los valores objetivos propuestos para el año seleccionado.
- La lista desplegable a través de la cual vamos a poder seleccionar el año en que queramos mostrar los datos del valor real y objetivo.
- Los tres botones que están programados para desencadenar un flujo en Power Automate. El botón de Create New Year, el botón de Review y el botón de Approved.



- La última pantalla mostrará los valores de la perspectiva, objetivo estratégico, indicador, valor real y valor objetivo. Además en esta pantalla se programará también un formulario que nos permitirá conectarnos con la tabla Indicator_Entry_Objective_Values y modificar el valor objetivo propuesto para el indicador seleccionado. En esta pantalla, a diferencia de las anteriores, no se ha programado ninguna galería sino que se han programado etiquetas de texto que devuelvan los valores que se han ido seleccionando.



Para programar las etiquetas que te muestren los valores mencionados anteriormente lo que hay que hacer es recuperar el valor de la etiqueta seleccionada en cada galería respectivamente:

- De tal manera que para recuperar el valor seleccionado de la perspectiva se utilizaría el siguiente código

```
GalleryPerspective.Selected.Perspectivee
```

- Para recuperar el valor seleccionado del objetivo estratégico se utilizaría el siguiente código

```
Gallery_Strategic.Selected.strategic_Target
```

- Para recuperar el valor seleccionado del indicador se utilizaría el siguiente código

```
GalleryIndicatoor.Selected.Indicator_Name
```

- Para recuperar el valor real para el indicador seleccionado se utilizaría el siguiente código. Además tener en cuenta que en estos casos para que se muestren únicamente dos decimales se utiliza la función Round.

```
Round(GalleryVReal.Selected.Real_Value;2)
```

- Para recuperar el valor objetivo para el indicador seleccionado se utilizaría el siguiente código. Además tener en cuenta que en estos casos para que se muestren únicamente dos decimales se utiliza la función Round.

```
Round(GalleryObjective.Selected.ObjectiveValue;2)
```

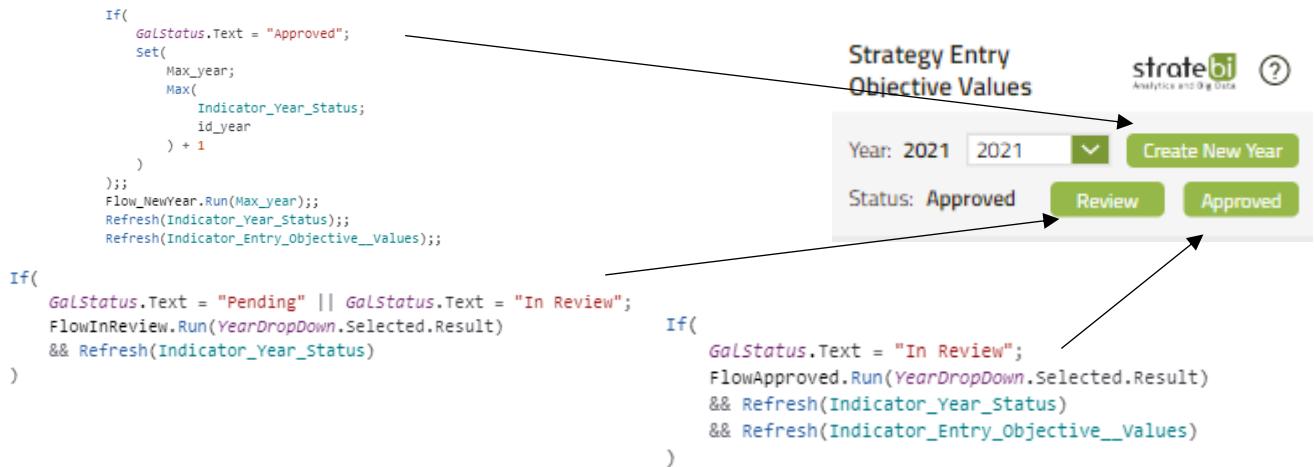
- Por último, quedaría programar el formulario de edición para que cuando se introdujese un valor objetivo por la aplicación, este valor quedase grabado en la tabla correspondiente de la base de datos de SQL server.

```
Lookup(  
    Indicator_Entry_Objective_Values;  
    ThisRecord.indicator = GalleryIndicatoor.Selected.indicator && id_year = YearDropDown_2.Selected.Result  
)
```

Flujos de Power Automate

Una vez programada la aplicación en Power Apps, el último paso sería conectar Power Apps con los flujos de datos de Power Automate.

Como se ha mencionado anteriormente la aplicación va a contar con tres botones que programados van a llamar a un automatismo en Power Automate:



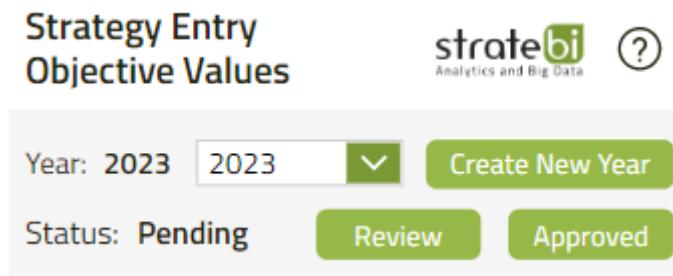
- El botón de **Create New Year**: Se ha programado para que únicamente pueda iniciarse cuando el estado de los valores del año anterior esté aprobado. Además se pasa por parámetro a la función que desencadena el flujo el año máximo que existe en la tabla Indicator_Year_Status + 1. Este flujo tendrá como función crear un nuevo año y establecer el valor para ese nuevo año a Pendiente.
- El botón **Review**: Se ha programado para que únicamente pueda iniciarse cuando el estado de los valores de ese año esté en pendiente o en revisión. Además se pasa por parámetro a la función que desencadenará el flujo el valor del año seleccionado en la lista desplegable. Este flujo tendrá como función pasar del estado Pendiente a En revisión y enviar una notificación al responsable de revisar los valores objetivos para aprobarlos o en caso de creerlo conveniente modificarlos.
- El botón **Approved**: Se ha programado para que únicamente pueda iniciarse cuando el estado de los valores de ese año está en revisión. Además se pasa por parámetro, a la función que desencadenará el flujo, el valor del año seleccionado en la lista desplegable. Este flujo tendrá como función pasar del estado En revisión a aprobado, enviar una notificación al responsable para comunicarle que los valores objetivos han sido aprobados y volcar los datos de la tabla Indicator_Entry_Objective_Values a la tabla Fact Scorecard para que puedan visualizarse en Power BI.

4. CASO DE USO

Una vez visto el desarrollo, en este punto, se pretende mostrar un ejemplo de su funcionamiento.

Supongamos que estamos en el mes de diciembre, y nuestra organización asocia el cierre de ejercicio al año natural. Ha llegado el momento de establecer los valores objetivo de los indicadores, para ello y como primer paso, desde Power App, sería necesario preparar el nuevo ejercicio, y para ello se debe crear un nuevo año con el botón Create New Year.

Como podemos observar se ha creado el año 2023 y para este año el estado por defecto es Pending:



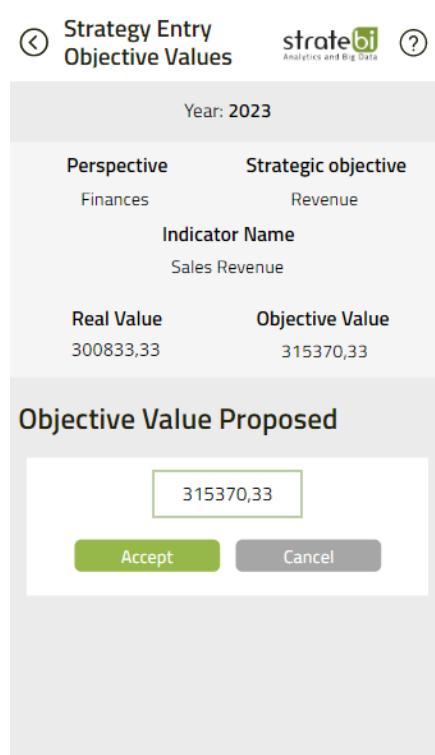
Como consecuencia de ello, en la tabla Indicator_Entry_Objective_Values de la base de datos demo_powerplatform_scorecard se insertan los registros necesarios para el nuevo año:

2023	1	Sales Revenue	300833...	315370,3333...	NULL
2023	2	Expenses	48500	46476,66666...	NULL
2023	3	EBITDA	249500	280166,6666...	NULL
2023	4	Profit	174250	174250	NULL
2023	5	Cash Flow	52166.6...	53750	NULL
2023	6	Accident Free Workplace	2,66666...	2	NULL
2023	7	Critical Incidents	2,5	1,25	NULL
2023	8	Serious Service Incidents	1,41666...	1,166666666...	NULL
2023	9	Right the First Time	0,79083...	0,875	NULL
2023	10	Customer Satisfaction	1	1	NULL
2023	11	Unresolved Complaint	17,25	12,58333333...	NULL
2023	12	Customer Turn-Over Rate	0,72916...	0,795833333...	NULL
2023	13	Market Value	0,7625	0,6875	NULL

A partir de este momento, lo tenemos todo listo para ir estableciendo el valor objetivo de los indicadores. Para ello se puede ir navegando por las perspectivas y objetivos estratégicos hasta llegar al indicador deseado. En la captura inferior, vemos como para el indicador Sales Revenue, aparece la siguiente información: Valor real, Valor Objetivo y Valor Objetivo Propuesto.

The screenshot shows a user interface for managing indicators. At the top, a dark blue header bar contains the text "Indicator". Below it, a light green header bar displays the indicator name "Sales Revenue" next to a small icon of a document with a dollar sign. The main content area is white and contains three lines of text: "Real Value: 300833,33", "Objective Value: 315370,33", and "Proposed Objective Value: 315370,33".

Seleccionando el indicador, se abrirá una pantalla de edición, en la que se proporcionará la posibilidad de introducir el valor objetivo propuesto:



Una vez establezcamos el valor objetivo debería aparecernos en el recuadro del formulario y en la página de indicadores. En este caso se ha establecido que el valor objetivo propuesto para ese año sea 350000.

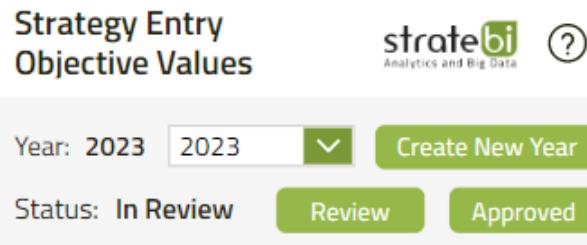
The screenshot shows two panels. On the left, the "Indicator" panel for "Sales Revenue" displays:

- Icon: Sales Revenue
- Real Value: 300833,33
- Objective Value: 315370,33
- Proposed Objective Value: 350000

On the right, a modal dialog titled "Objective Value Proposed" shows the value "350000" in a green-bordered input field, with "Accept" and "Cancel" buttons.

Una vez establecidos todos los valores objetivos para los indicadores en el año seleccionado se pasará al estado de revisión con el botón Review. De esta manera, al encargado de supervisar los valores objetivos le llegará una notificación y podrá aprobar los valores o en su defecto a modificarlos.

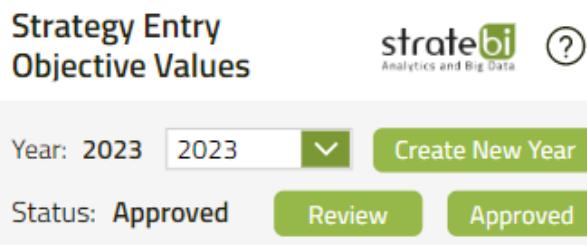
Hello,
The objective values of the indicators have been established for the year 2023
Please review the values entered so they can be approved



Por último, una vez revisados todos los valores se procedería a aprobarlos con el botón Approved. De esta manera el estado pasaría a aprobado y le llegaría una notificación al responsable para avisarle de que para ese año ya se han aprobado los valores objetivos.

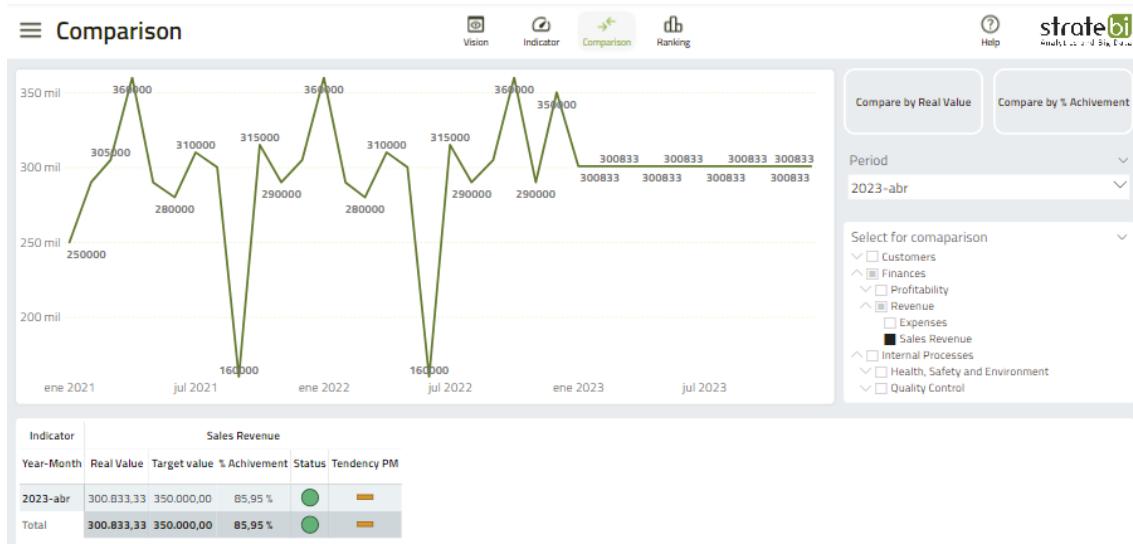
Hello,

The objective values of the indicators have been Approved for the year 2023



Es importante ver el funcionamiento de la aplicación no sólo desde el propio PowerApps si no también como interactúa con el informe de PowerBI. Como se ha mencionado anteriormente el informe está diseñado en DirectQuery de tal manera que al introducir un nuevo año, modificar el valor objetivo y aprobar su valor automáticamente quede reflejado en el propio informe.

Debajo se puede ver en el informe el resultado de establecer el valor objetivo para el indicador Sales Revenue:



5. CONCLUSIONES

Como conclusión, este ejercicio realizado trata de ilustrar como fácilmente se pueden poner en marcha planteamientos teóricos como el que nos ocupa, un sistema de Balance Scorecard o Cuadro de Mando Integral, utilizando tecnología que está al alcance de cualquier organización. Este sistema provee a las organizaciones de diferentes capacidades, como la analítica, la funcional y operativa, todo en uno e integrado con mucha facilidad.