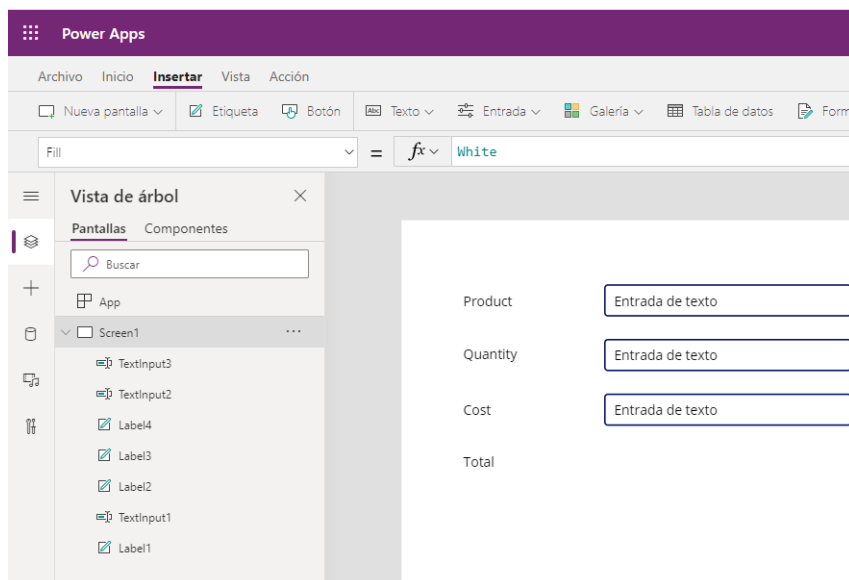


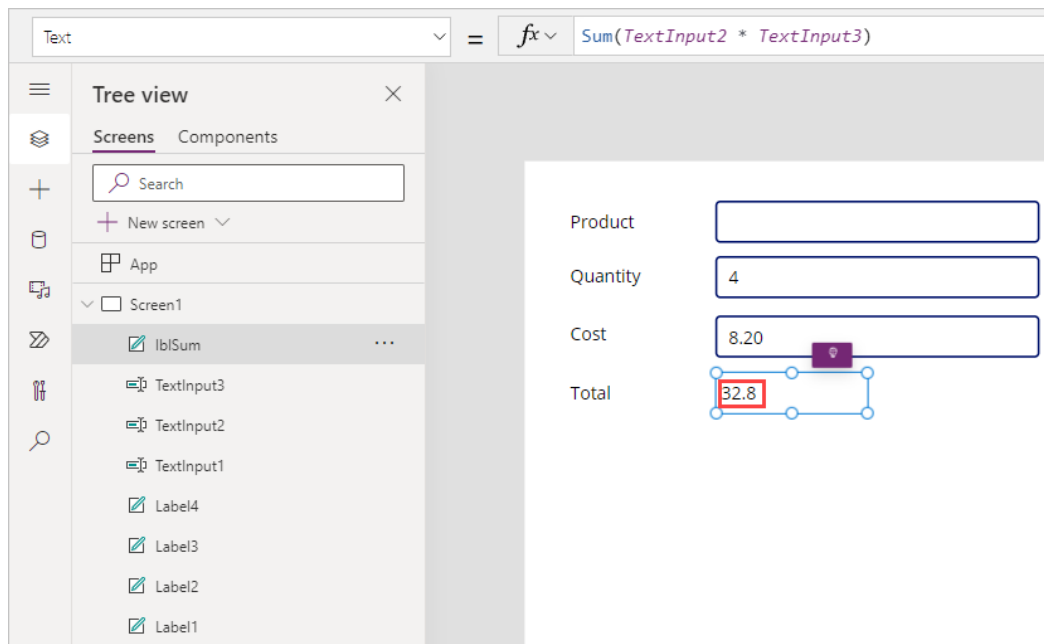
Usar fórmulas para realizar cálculos

En el ejemplo siguiente se muestra una forma de calcular la suma de los bienes y servicios.

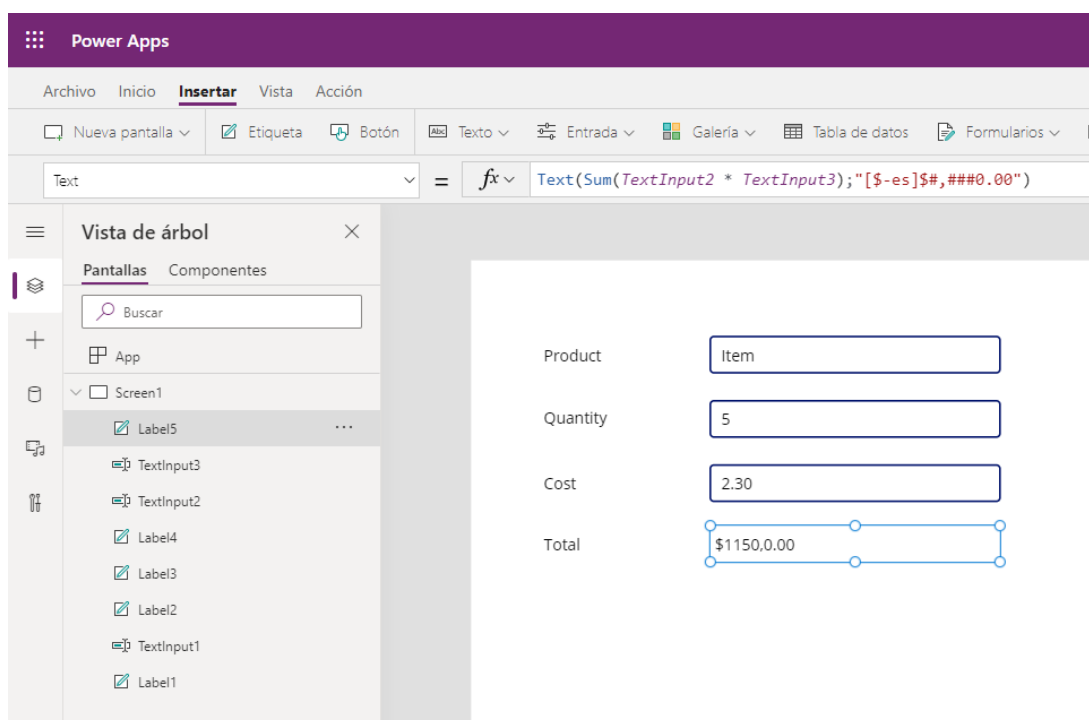
1. Vaya a [Power Apps](#) y seleccione **Aplicaciones** en el panel de navegación izquierdo.
2. Seleccione **+ Nueva aplicación** y **Lienzo**.
3. Asigne un nombre a la aplicación, compruebe que la opción **Tableta** esté seleccionada como formato y seleccione **Crear**.
4. Seleccione el botón **+ Insertar** en la cinta del encabezado y agregue una **Etiqueta de texto**.
5. Cambie la propiedad **Text** por *Product*.
6. Agregue tres **Etiquetas** más y cambie las propiedades **Text** por *Quantity*, *Cost* y *Total*, respectivamente.
7. A continuación, inserte controles **Text Input** en la aplicación de lienzo y ordénelos como se muestra en la siguiente captura de pantalla.



8. En la propiedad **Default** de cada **Text Input**, elimine "Entrada de texto" para que cada **Default** quede en blanco.
9. Inserte otra **Etiqueta de texto** a la derecha de *Total*. Cambie el nombre de este control por *lblSum*.
10. Cambie la propiedad **Text** por lo siguiente. `Sum(TextInput2 * TextInputType3)`
11. Ahora obtenga una versión preliminar de la aplicación seleccionando el botón **Reproducir** situado en la esquina superior derecha.
12. Pruebe la fórmula introduciendo 4 para el campo de entrada **Cantidad** y 2.30 para el **Coste**. Observe que actualmente no mostramos dos decimales (lo cual es normal cuando se representa una divisa) o un signo de divisa, así que modificaremos la fórmula para mostrarlos a continuación.



13. Para modificar la fórmula para que muestre el valor como *Divisa*, actualice la propiedad **Text** de *lblSum* a `Text(Sum(TextInput2 * TextInput3), "$#,###0.00")`



Observe cómo ahora *Total* muestra el resultado como una *Divisa*.

14. Agreguemos un resumen de transacción. Agregue una última **Etiqueta de texto** bajo los demás controles y estire el ancho para que ocupe al menos la mitad del lienzo de la pantalla. En la propiedad **Text**, introduzca la fórmula siguiente: `"Your " & TextInput1.Text & " total is: " & lblSum.Text`

Donde *TextInput1* es el control de entrada de texto junto a **Producto**

15. Ponga la aplicación en modo de versión preliminar e introduzca un valor en el control de la entrada de texto, junto a **Producto**. Observe cómo obtiene un mensaje de resumen de transacción que incorpora los valores de todos los campos de entrada. En nuestro ejemplo, hemos introducido *Widget* como nombre del producto. Introduzca valores diferentes en los campos de entrada de texto para ver cómo el total y el mensaje de resumen de transacción van cambiando dinámicamente.

Product	<input type="text" value="Widget"/>
Quantity	<input type="text" value="4"/>
Cost	<input type="text" value="8.20"/>
Total	\$32.80
<div>Your Widget total is: \$32.80</div>	

Ahora hemos usado tres controles de entrada con la fórmula **Sum** para calcular un total y mostrar un mensaje de resumen de transacción.

A continuación, presentaremos cómo podemos usar un control para modificar las propiedades de otros controles, específicamente cómo podemos cambiar la posición de visualización de un control.

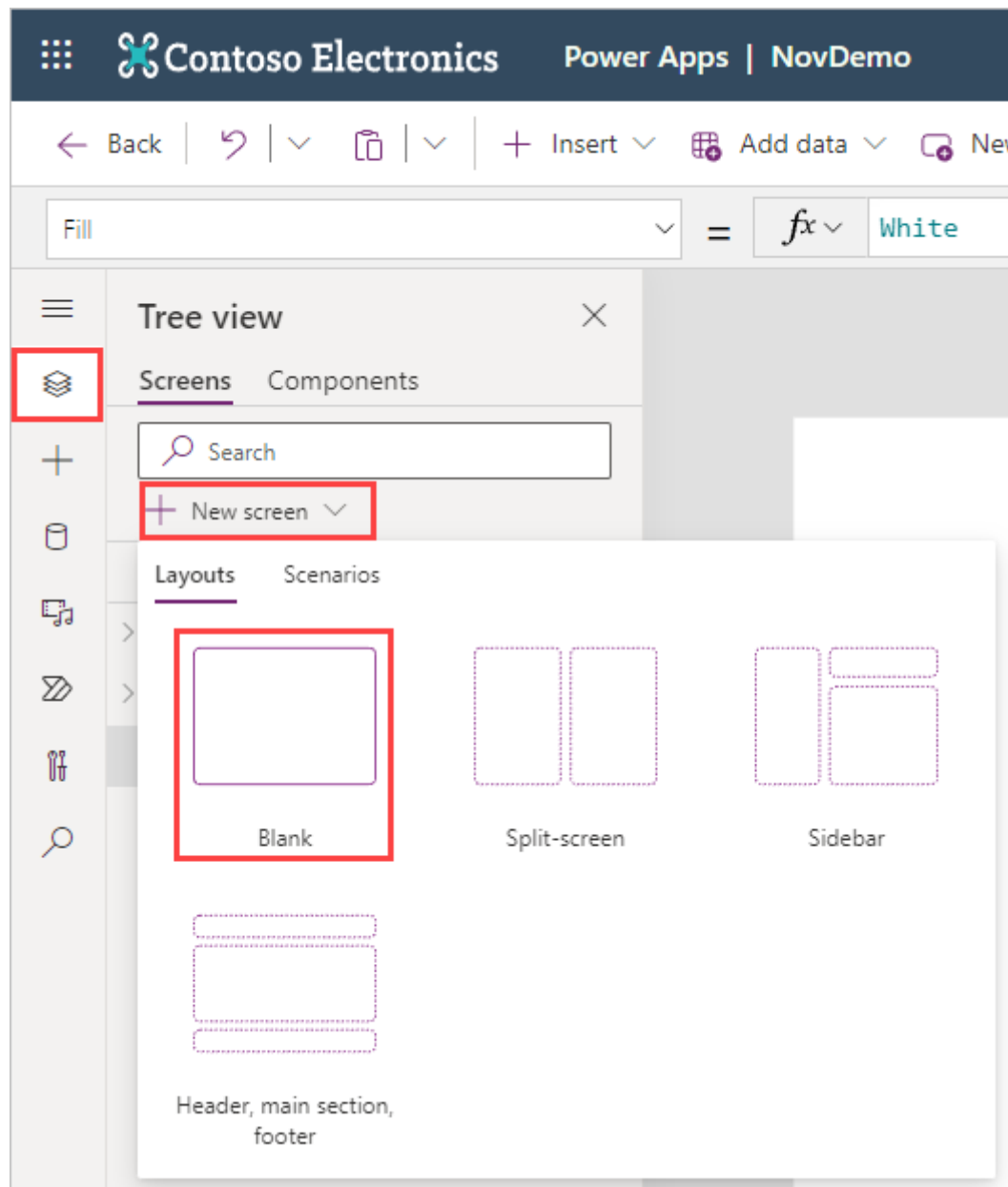
Usar un control para modificar la propiedad de otros controles

Vamos a usar un ejemplo para entender el posicionamiento de un control en el lienzo y cómo usar la información que especifique el usuario para modificarlo. Como introducción, podemos decir que la ubicación de un control en el lienzo se basa en una combinación de dos propiedades: las propiedades **X** e **Y**.

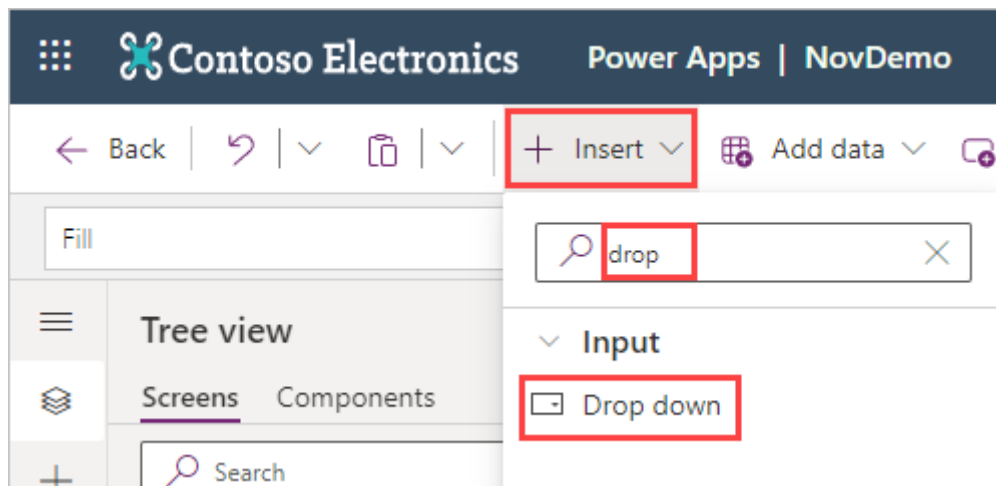
- **X**: distancia entre el borde izquierdo de un control y el borde izquierdo de la pantalla.
- **Y**: distancia entre el borde superior de un control y el borde superior de la pantalla.

Recuerde que las propiedades **X** e **Y** se aplican a todos los controles, excepto las pantallas. Ahora vaya a su aplicación y aprendamos con la práctica.

1. Primero vamos a crear una nueva pantalla para la aplicación seleccionando + **Nueva pantalla** en el panel **Vista de árbol**, luego seleccione **En blanco**.



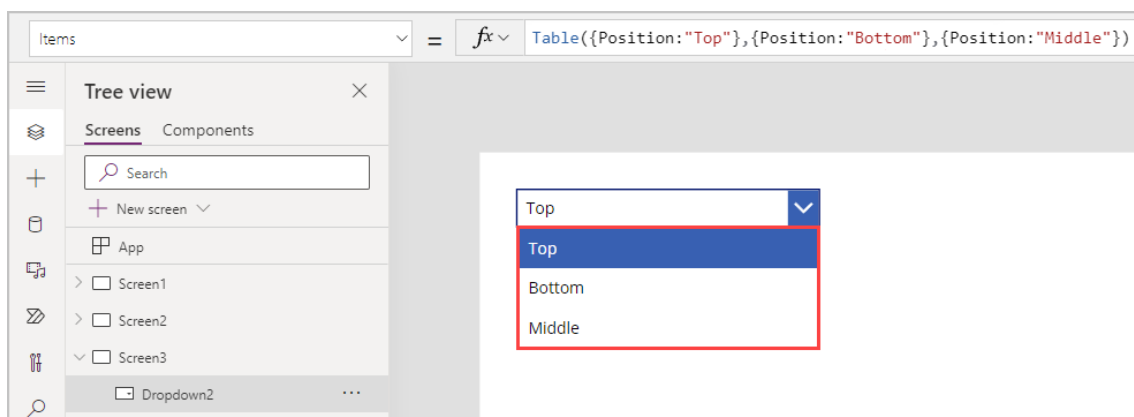
2. Insertemos un control dropdown. En la cinta de la aplicación, seleccione el botón + **Insertar** y escriba *desplegable* en el campo de búsqueda, luego seleccione **Desplegable**.



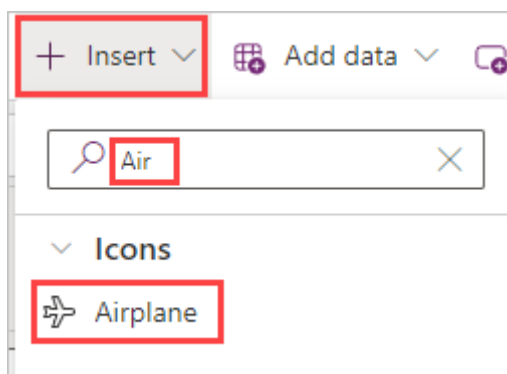
3. Cambie la propiedad **Items** de la lista desplegable de **DropDownSample** a lo siguiente:

```
Table({Position:"Top"},{Position:"Bottom"},{Position:"Middle"})
```

Esto crea una tabla de registros con una única columna denominada **Position** con tres registros denominados *Top*, *Bottom* y *Middle*. Si selecciona el control dropdown mientras mantiene presionada la tecla **Alt**, verá que se muestran estos valores.



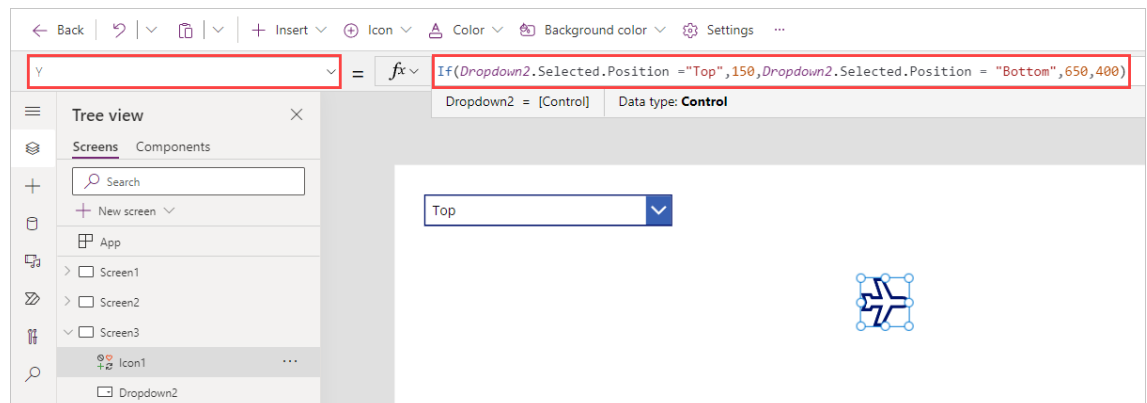
- A continuación, agregaremos un control que podemos reposicionar dinámicamente en la pantalla. Una vez más, seleccione + **Insertar** desde la cinta, escriba *avión* en el campo de búsqueda y seleccione el icono del **avión**. Arrastre el icono del **avión** hasta el centro de la pantalla.



- Ahora hagamos que el icono del **avión** aparezca a diferentes alturas de la pantalla basándonos en el control dropdown. Para ello, seleccione el icono y busque la propiedad **Y**. Observe que actualmente tiene un valor de número estático. Cambie el valor de la propiedad **Y** en el campo **fx** por el siguiente valor:

```
If(Dropdown2.Selected.Position = "Top", 150, Dropdown2.Selected.Position = "Bottom", 650, 400)
```

Nota: El control dropdown puede tener un nombre que no sea *Dropdown2*. Reemplace *Dropdown2* por el nombre específico del control.]



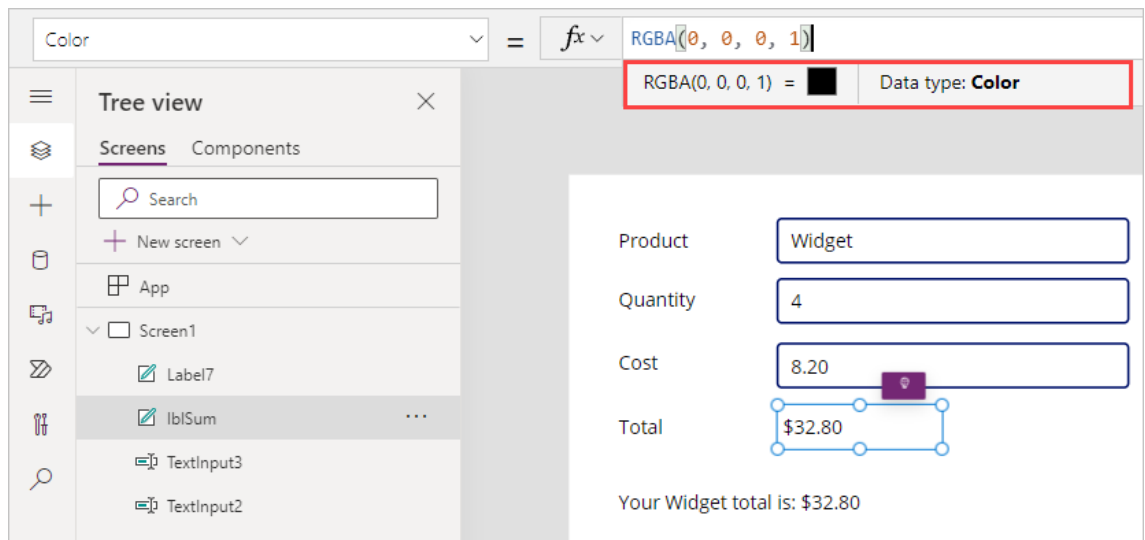
Para divertirnos un poco (y probar la fórmula), ponga la aplicación en modo de versión preliminar. Ahora seleccione cada posición en el menú desplegable. Observe que el icono **Avión** ajusta su "altura" (o coordenada **Y**) a medida que cambia el valor desplegable.

Cambiar las propiedades del control le permite modificar cómo funcionan otros controles. Si lo hace, le brindará una excelente manera de establecer la experiencia de usuario.

Formato condicional para la propiedad Color de un control.

Siguiendo con el ejemplo anterior de la función **Suma**, vamos a crear una fórmula para aplicar un formato condicional a la propiedad **lblSum Color**. Vuelva a la primera pantalla de la aplicación de práctica que creamos en la Unidad 3 (remítase a la captura de pantalla con el número 1 a continuación para recordar cómo era esa pantalla).

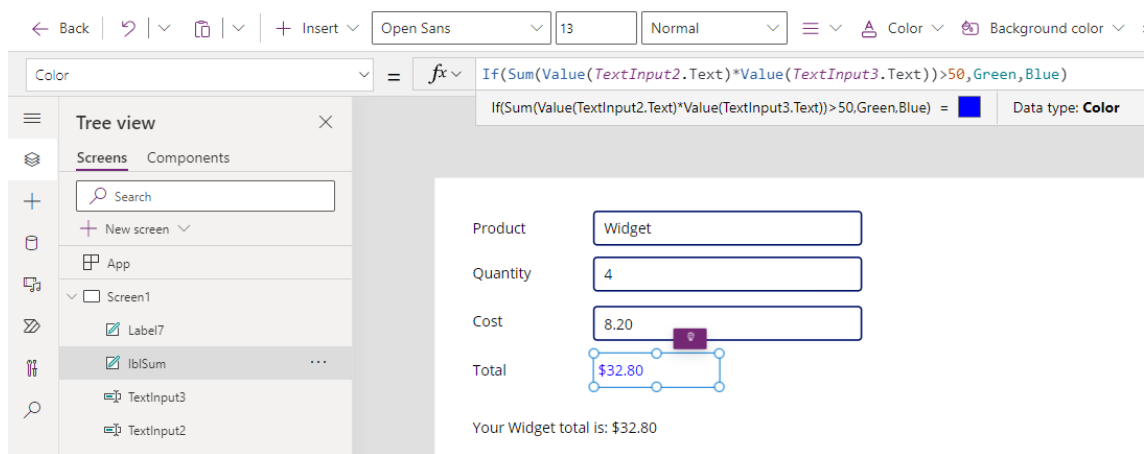
1. Seleccione **lblSum** y vaya a la propiedad **Color**. Si coloca el cursor dentro del campo **fx**, podrá observar cómo el campo de información justo debajo de este muestra que el valor de RGBA(0, 0, 0, 1) es negro(color) con **Tipo de datos: Color**.



2. Borre el valor actual y especifique lo siguiente en el campo **fx** para **Color**:

`If(Sum(Value(TextInput2.Text)*Value(TextInput3.Text))>50,Green,Blue)`

La propiedad **Color** define el color del texto en el control **Label**. En este ejemplo, si el cálculo **Suma** de los dos controles **TextInput** es mayor que 50, el color será verde; de lo contrario, el color será azul. Además, observe cómo, si coloca el cursor en la barra de fórmulas después de cambiar la fórmula, el tipo de datos de salida es **Color**.



Funciones para validar datos

Utilizar funciones de validación en la aplicación puede ayudarle a influir en los datos que introduzca el usuario y en la interacción general con la aplicación.

A continuación, se ofrecen algunas funciones de validación que se pueden usar en Power Apps (recuerde que el resultado es siempre *true* o *false*):

- **IsBlank**: comprueba un campo, propiedad o variable para ver si el resultado está *en blanco* o es una cadena vacía.

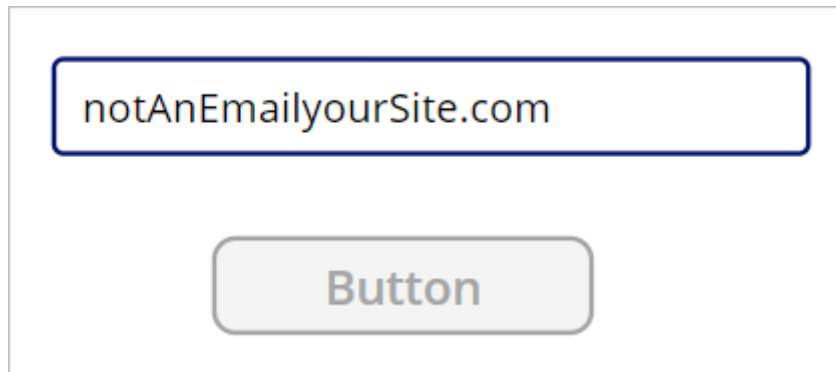
- **IsBlankOrError**: comprueba si una expresión está en blanco o genera un error, y devuelve true o false.
- **IsEmpty**: comprueba una tabla de datos para ver si contiene registros.
- **IsError**: prueba un valor de error.
- **IsMatch**: comprueba una cadena de texto para determinar si el resultado coincide con el patrón especificado. Por ejemplo, puede usar esta función para probar entradas de correo electrónico o números de teléfono válidos. Para obtener más información y ejemplos sobre los diversos patrones de coincidencia, consulte [Función IsMatch en Power Apps](#).
- **IsNumeric**: comprueba un campo, propiedad o variable para determinar si el valor es numérico (como en el ejemplo anterior).
- **IsToday**: comprueba una fecha o valor de hora para ver si se encuentra entre hoy a medianoche y mañana a medianoche según la hora local.
- **IsUTCToday**: es similar a **isToday**, pero utiliza la hora universal coordinada (UTC) en lugar de la hora local y no depende de las zonas horarias.

Echemos un vistazo a un ejemplo más sobre la función **IsMatch**. Puede probar esta funcionalidad para *deshabilitar* un botón si el control **Text Input** no tiene especificada una dirección de correo electrónico válida. *Correo electrónico* es un patrón predefinido que se puede usar fácilmente. Sin embargo, recuerde que está buscando solo un *patrón*; no* se valida si la dirección de correo electrónico realmente funciona o no. Pasemos a utilizar la aplicación de demostración para lo siguiente:

1. Agregue un control **Text Input** al lienzo. Haga que la propiedad **Default** quede en blanco eliminando "*Entrada de texto*". Cambie el nombre del control: *tiEmailEntry*.
2. Agregue un control de **botón** justo debajo del campo de entrada de texto.
3. Modifique la propiedad **DisplayMode** del control **Button** a la siguiente fórmula:

```
If(IsMatch(tiEmailEntry.Text, Email),
DisplayMode.Edit, DisplayMode.Disabled)
```

3. Con la fórmula de coincidencia *Correo electrónico*, Power Apps puede probar un valor de texto en el campo de entrada con este patrón: una cadena alfanumérica, seguida del símbolo @, otra cadena alfanumérica, un punto (.) y se cierra con una cadena de varias letras. Si la cadena de entrada no coincide con el patrón especificado, el botón permanecerá en modo "deshabilitado" y el usuario no podrá interactuar con él. Probemos la funcionalidad.
4. Obtenga una vista previa de la aplicación seleccionando el botón **Reproducir** situado en la esquina superior derecha.
5. Si el campo está en blanco o tiene una dirección de correo electrónico incompleta, se deshabilitará el botón y se mostrará en un color gris. Pruebe algunos ejemplos diferentes sin @ o sin el punto (.).

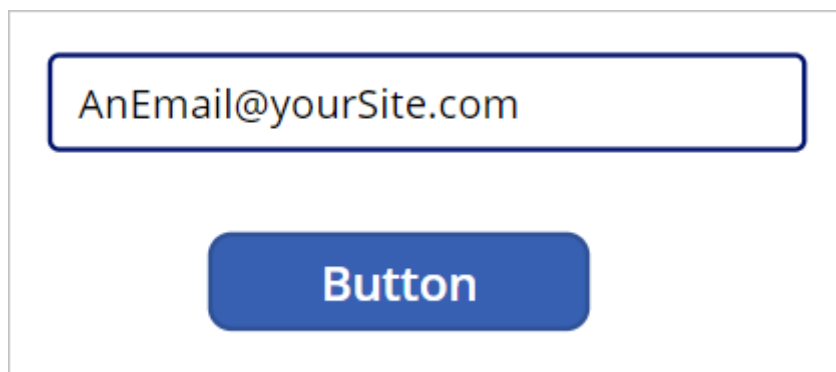


notAnEmailyourSite.com

Button

A form with a text input field containing the text "notAnEmailyourSite.com" and a button labeled "Button". The button is light gray and disabled.

Cuando escriba una dirección de correo electrónico con formato correcto, el botón pasará a color azul y se podrá seleccionar.



AnEmail@yourSite.com

Button

A form with a text input field containing the text "AnEmail@yourSite.com" and a button labeled "Button". The button is blue and enabled.

Las funciones de validación de datos son herramientas útiles para dar forma a los datos introducidos por sus usuarios. Emplearlos puede ayudar a eliminar errores de introducción de datos y mejorar aún más la experiencia del usuario.