



Capítulo 12: Desarrollo de Formularios

Este capítulo se centra en el uso avanzado de BIZUIT Forms Designer para crear formularios dinámicos, interactivos y personalizados, adaptados a diversas necesidades empresariales. Exploraremos el entorno del diseñador, la configuración y personalización de controles, la integración de fuentes de datos externas e internas, y el uso de subformularios para modularidad y flexibilidad. Aprenderemos a configurar propiedades avanzadas, validar entradas, manejar datos dinámicos y diseñar interfaces atractivas y funcionales.

Audiencia Ideal

Este capítulo está dirigido a Implementadores de proyectos BPM que ya tienen experiencia con el desarrollo de formularios o interfaces de usuario en entornos empresariales. Se recomienda contar con conocimientos básicos en: Diseño de formularios, UX/UI, HTML y JavaScript

Objetivos

1. Explorar el entorno de BIZUIT Forms Designer: Comprender las funcionalidades y herramientas clave para diseñar formularios eficientes y personalizados.
2. Configurar controles básicos y avanzados: Personalizar propiedades, crear validaciones y manejar datos dinámicos para optimizar la interacción con los formularios.
3. Diseñar interfaces atractivas: Utilizar controles como tablas, pestañas y tarjetas para construir formularios organizados y funcionales.
4. Integrar datos externos: Configurar fuentes de datos secundarias (SQL, BIZUIT y RESTAPI) para enriquecer los formularios con información en tiempo real.
5. Implementar subformularios: Dividir flujos complejos en secciones modulares para mejorar la flexibilidad y la experiencia del usuario.



Unidad 1: Exploración del Entorno de BIZUIT Forms Designer

En esta unidad vamos a explorar a fondo el entorno del diseñador de formularios de BIZUIT. Nuestro objetivo es conocer en detalle cada una de sus áreas, entender su funcionalidad y descubrir cómo aprovechar al máximo las herramientas que nos ofrece para construir formularios potentes, personalizados y alineados con los procesos de nuestra organización.

Barra lateral de menú

Comenzamos nuestro recorrido por la barra lateral de menú, un área fundamental desde la cual accedemos a todas las opciones necesarias para crear, editar y gestionar formularios, tanto independientes como asociados a procesos.

Archivo

Dentro del menú Archivo encontramos las herramientas esenciales para manejar nuestros formularios. Algunas de estas opciones varían según el tipo de formulario:

- **Nuevo Formulario** (solo para formularios independientes): nos permite iniciar un diseño desde cero. El sistema nos pedirá un nombre único y, si queremos, también podemos definir categoría, subcategoría y descripción.
- **Abrir Formulario Existente**: accedemos a una lista de formularios guardados. Podemos buscar por nombre, categoría o subcategoría. Desde esta ventana también podemos eliminar formularios, abrir versiones anteriores o exportarlos.
- **Exportar/Importar Formulario**: al exportar, generamos un archivo descargable que podemos importar en otro entorno. Si ya existe un formulario con el mismo nombre, el sistema nos pedirá uno nuevo para evitar conflictos.
- **Guardar Formulario**: podemos sobrescribir la versión actual o guardar una nueva. En el caso de los formularios asociados a procesos, la opción Guardar y Publicar acelera la puesta en marcha del diseño.



-
- **Copiar Formulario:** ideal para reutilizar formularios existentes como base para nuevos diseños.
 - **Configuración del Formulario:** accedemos rápidamente a la hoja de propiedades para ajustar todos los parámetros generales del formulario.

Controles

Desde este menú accedemos a todos los componentes disponibles: cajas de texto, botones, listas desplegables, áreas de texto, etc. Podemos arrastrarlos al área de diseño y ubicarlos según nuestras necesidades. Si tenemos muchos, contamos con una barra de búsqueda para encontrarlos fácilmente.

Controles Agregados

Aquí se listan todos los controles que ya hemos incorporado al formulario. Podemos seleccionarlos para modificar sus propiedades o eliminarlos directamente.

Subformularios

Desde esta sección gestionamos los subformularios vinculados al formulario principal. Podemos crearlos, editarlos o eliminarlos según lo requiera el diseño.

Fuentes de Datos

Configuramos aquí las fuentes que alimentarán el formulario. Para formularios asociados a procesos, además de APIs REST o SQL, también podemos trabajar con parámetros, variables o resultados de actividades previas del proceso.

Área de Diseño

El área de diseño es donde organizamos visualmente nuestro formulario. Colocamos los controles, los distribuimos en cuadrículas de filas y columnas y construimos la estructura lógica y visual de la interfaz. Este espacio nos permite trabajar de forma limpia, estructurada y ágil.

Barra de Herramientas

La barra superior concentra herramientas prácticas para la edición del formulario:



- **Código:** podemos ver y editar el código personalizado asociado a eventos de los controles. También contamos con un espacio de Código Global, donde definimos funciones reutilizables en distintos formularios.
- **Copiar, Pegar, Cortar, Deshacer, Rehacer:** las clásicas funciones de edición.
- **Alternar Cuadrícula:** permite mostrar u ocultar la grilla de filas y columnas para facilitar la organización del diseño.
- **Copiar URL** (para formularios independientes): obtenemos el enlace del formulario, útil para integrarlo en otros sistemas o compartirlo directamente.
- **Ejecutar Formulario:** visualizamos cómo se verá el formulario en tiempo de ejecución, ideal para hacer pruebas rápidas.
- **Ícono de Usuario:** accedemos a configuraciones generales del entorno del editor. Si contamos con permisos, también podemos modificar la apariencia del entorno, al igual que lo hacemos con el Dashboard de BIZUIT.
- **Invitar a Ver/Editar:** seleccionamos usuarios a quienes queremos invitar a colaborar. Al tildar la opción "Se puede Editar", el usuario invitado podrá trabajar en conjunto con nosotros, en tiempo real. Recibirá una invitación por correo y, al aceptarla, podrá comenzar a editar directamente.

Hoja de Propiedades

La hoja de propiedades es el lugar donde configuramos los detalles de cada control y del formulario en general. Ajustamos nombres, estilos, eventos, conexiones con datos y cualquier parámetro necesario para que cada componente cumpla correctamente su función.

Conclusión

En esta unidad recorrimos en detalle el entorno de trabajo del BIZUIT Forms Designer, comprendiendo cada una de sus áreas y herramientas clave para el diseño de formularios. Aprendimos a utilizar el menú lateral para acceder a funciones esenciales, desde la creación, apertura y exportación de formularios hasta la gestión de subformularios y fuentes de datos.

Exploramos el área de diseño como espacio central para estructurar visualmente los formularios, junto con la barra de herramientas que nos permite realizar acciones rápidas, probar diseños y colaborar en tiempo real con otros usuarios.



También profundizamos en la hoja de propiedades, donde es posible ajustar los parámetros generales del formulario y la configuración individual de cada control, optimizando así su apariencia, comportamiento e integración con datos y procesos.

Con estos conocimientos, estamos preparados para aprovechar de manera efectiva el entorno del diseñador, creando formularios potentes, personalizados y alineados con las necesidades operativas de nuestros procesos en BIZUIT.



Unidad 2: Controles

En esta unidad exploraremos los distintos controles disponibles en BIZUIT y aprenderemos a configurarlos para maximizar su funcionalidad y personalización.

Veremos cómo los formularios y controles son herramientas clave para capturar, mostrar y gestionar datos en aplicaciones modernas. Descubrirás cómo configurar propiedades básicas y avanzadas, crear validaciones, trabajar con datos dinámicos y manejar interacciones complejas. Además, aprenderemos a integrar multimedia, geolocalización y firmas digitales, adaptando cada control a las necesidades específicas de nuestros proyectos.

Desde formularios hasta tablas, pestañas y elementos visuales como tarjetas o encabezados, esta unidad nos brindará las bases para diseñar interfaces atractivas, interactivas y altamente funcionales.

1. Formulario

Analizaremos las propiedades básicas de los formularios en BIZUIT, enfocándonos en cómo personalizar su apariencia, comportamiento y acciones asociadas, tanto en formularios independientes como en aquellos vinculados a procesos.

Nuestra meta es aprovechar al máximo las opciones disponibles para adaptar cada formulario a nuestras necesidades funcionales y visuales.

Acceso a las Propiedades del Formulario

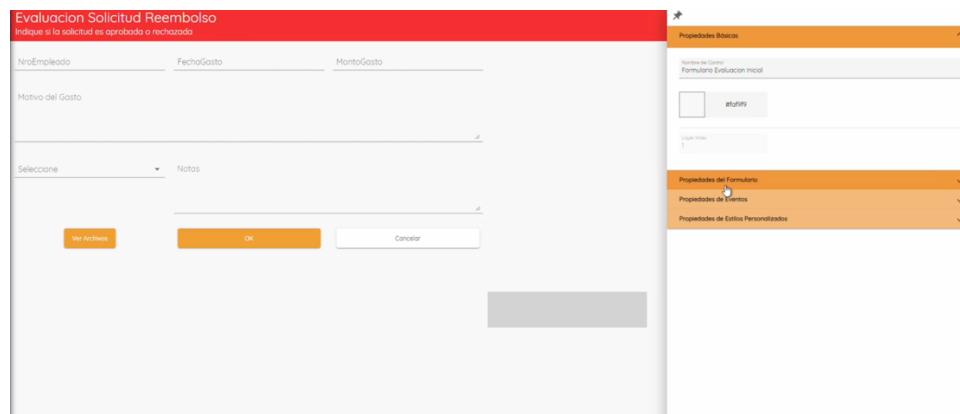
Para comenzar, desde la barra lateral del editor seleccionamos la opción "Editar Propiedades del Formulario". Al hacerlo, se despliegan las opciones de configuración, que varían ligeramente según se trate de un formulario independiente o uno asociado a procesos.

Opciones Comunes a Todos los Formularios

En ambos tipos de formularios encontramos la pestaña Propiedades Básicas, desde donde podemos modificar el nombre del formulario y su color de fondo.

En la pestaña Propiedades del Formulario, accedemos a parámetros de diseño visual como:

- **Estilo de controles:** aplicamos estilos predefinidos, como por ejemplo "Standard".
- **Cantidad de columnas:** definimos la estructura visual del formulario. En este ejemplo, usamos un layout de 6 columnas.
- **Altura de filas:** ajustamos la altura en píxeles, por ejemplo, a 70 px.
- **Tema visual:** elegimos entre temas predefinidos como "Bizuit", "Red" o "Green", unificando el estilo del formulario con el entorno del usuario o la marca.



Propiedades Específicas para Formularios Asociados a Procesos

Cuando trabajamos con formularios asociados a procesos, se habilitan nuevas opciones:

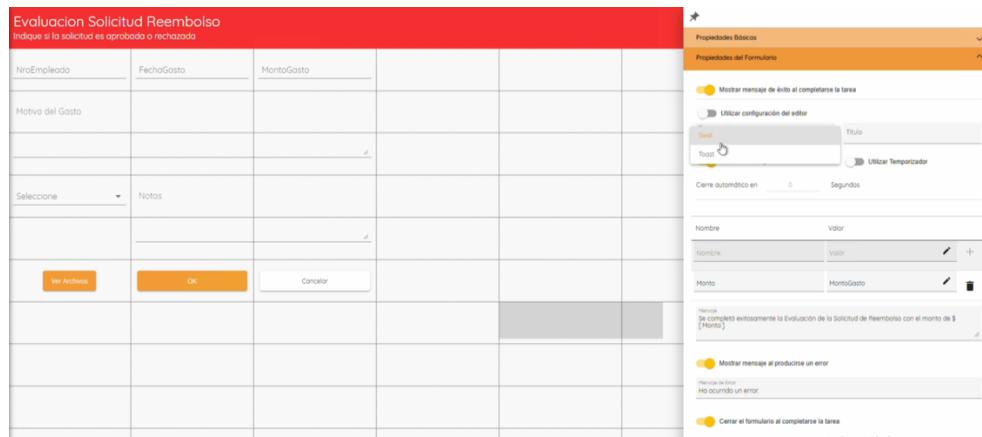
Mensaje de Éxito

Podemos definir un mensaje personalizado que se muestra al completar correctamente la tarea (por ejemplo, iniciar o continuar un proceso). Incluso podemos incluir variables del proceso, como el monto solicitado, en el mensaje.

También elegimos cómo se presentará este mensaje:

- **Swal (ventana modal):** con título, botón de cierre y temporizador opcional.

- **Toast (mensaje emergente):** aparece discretamente en la esquina superior de la pantalla.



Mensaje de Error

Podemos configurar un mensaje que se muestre al usuario en caso de error durante el procesamiento del formulario.

Cierre Automático del Formulario

Decidimos si el formulario se cierra automáticamente una vez completada la tarea. Esta opción es útil en formularios de inicio, donde queremos permitir al usuario iniciar múltiples instancias sin tener que abrir el formulario manualmente cada vez.

Procesamiento Asíncrono

Si activamos esta opción, el formulario puede cerrarse sin esperar a que BIZUIT finalice el procesamiento de la tarea. Una vez completado, el resultado se muestra en la pestaña de Notificaciones del Dashboard.

También podemos configurar un tiempo de espera para que el formulario permanezca abierto algunos segundos antes de cerrarse automáticamente.

Por ejemplo: si sabemos que el proceso puede tardar 30 segundos, configuramos el cierre automático a los 5 segundos. Así evitamos que el usuario tenga que esperar sin necesidad.

Propiedades Exclusivas para Formularios Independientes

En los formularios independientes, disponemos de una pestaña adicional llamada Seguridad del Formulario.



Desde allí podemos activar la opción "Requiere autenticación" y definir qué usuarios o roles tienen acceso. Esta funcionalidad nos permite restringir el uso del formulario según los permisos establecidos.

Resumen

Como vimos, las propiedades de los formularios en BIZUIT nos brindan un alto nivel de personalización, tanto visual como funcional. Podemos ajustar la estética, el comportamiento al completar tareas, la seguridad de acceso y mucho más.

Estas opciones nos permiten crear formularios a medida, más intuitivos, seguros y alineados con los procesos y usuarios a los que están dirigidos.

2. Configuración de controles:

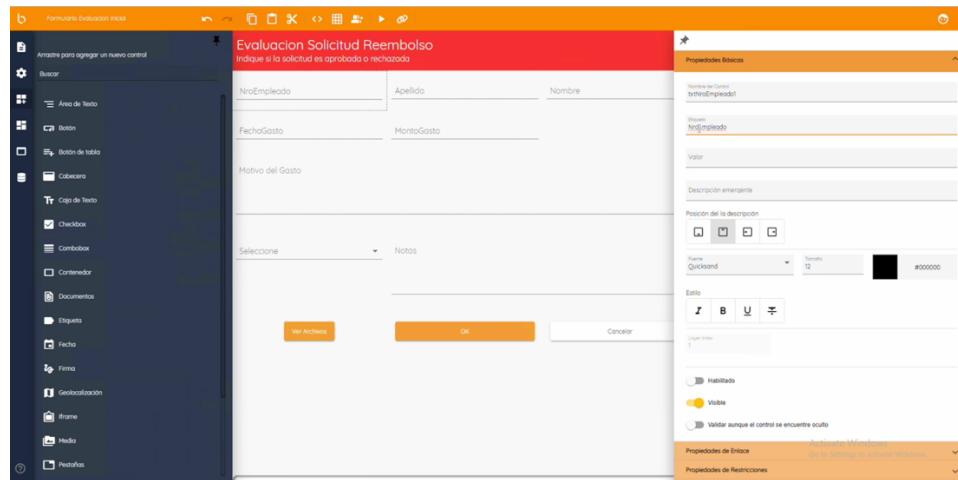
Ahora exploraremos en profundidad todas las posibilidades de configuración de los controles en BIZUIT Forms Designer. Desde propiedades básicas hasta reglas dinámicas y eventos personalizados, veremos cómo cada ajuste contribuye a construir formularios más potentes, claros e interactivos.

Propiedades Básicas

Las propiedades básicas varían según el tipo de control, pero existen ciertos parámetros comunes que podemos ajustar para personalizar el comportamiento y la presentación de cada elemento.

- **Nombre del Control:** asignamos un identificador único, como txtApellido, lo que nos facilita su uso en reglas, enlaces o scripts personalizados.
- **Etiqueta:** muestra un texto descriptivo sobre el control, como "Apellido", ayudando a orientar al usuario.
- **Descripción Emergente:** agregamos un tooltip que aparece al pasar el mouse sobre el control, brindando ayuda contextual (por ejemplo: "Ingrese su Apellido"). Incluso podemos personalizar su posición.
- **Valor:** definimos un valor inicial que se mostrará en el control. Si este está vinculado a una fuente de datos, dicho valor será reemplazado al ejecutarse la fuente.

- **Estilo:** ajustamos tipografía, tamaño, color, negrita, cursiva y subrayado. Podemos aplicar estilos para mejorar la presentación y legibilidad del formulario.
- **Layer Index:** establecemos el orden de superposición entre controles. Es útil cuando varios controles ocupan el mismo espacio y necesitamos que uno esté visible sobre otro, según el contexto.
- **Habilitado y Visible:** decidimos si el control estará activo o visible durante la ejecución del formulario.
- **Validar aunque esté oculto:** permite ejecutar las validaciones del control incluso cuando no es visible, lo cual resulta útil en reglas de negocio más complejas.



Propiedades de Enlace

Los enlaces son fundamentales para conectar controles con datos y procesos:

- **Enlace Primario** (en formularios asociados a procesos): asociamos el control a un parámetro, variable o resultado de una actividad anterior del proceso. Puede ser una vinculación bidireccional (lectura y escritura) o unidireccional (sólo lectura).
- **Enlaces de Entrada (Fuentes de Datos):** conectamos el valor del control con parámetros de entrada de fuentes secundarias, como SQL o APIs. Por ejemplo, vinculamos txtNroEmpleado1 al parámetro pldEmpleado de una fuente



llamada GetDataEmpleado, y configuramos su ejecución al cambiar el valor del control.

- **Enlaces de Salida (Fuentes de Datos):** usamos controles como destino para los valores devueltos por una fuente secundaria. Siguiendo el ejemplo anterior, txtApellido y txtNombre mostrarán los valores LastName y FirstName obtenidos al consultar la fuente GetDataEmpleado.

Propiedades de Restricciones

Estas propiedades nos permiten controlar qué tipo de datos se ingresan:

- **Restricciones del Control:** podemos marcar campos como obligatorios o limitar los valores permitidos. Por ejemplo, en un control de fecha, podemos restringir la selección a fechas posteriores a la actual.

Propiedades de Validaciones

Las validaciones permiten implementar reglas lógicas entre campos para asegurar la coherencia de los datos:

- Podemos definir condiciones usando operadores como “Igual a” o “Diferente de”.
- Las reglas pueden compararse contra valores fijos, parámetros, variables del proceso o incluso contra otros controles del formulario.
- Elegimos si la validación debe cumplirse cuando todas las condiciones sean verdaderas o cuando alguna lo sea.
- Además, podemos configurar mensajes personalizados para ayudar al usuario a corregir los datos ingresados. Por ejemplo: “El nombre no puede ser igual al apellido”.

Propiedades de Formateo

El formateo nos permite alterar dinámicamente la apariencia de los controles en función de condiciones lógicas.

- **Reglas de Formateo:** definimos cuándo un control debe ser visible, habilitado o tener cierto color.



- **Acciones de Formateo:** especificamos qué cambios aplicar (por ejemplo, ocultar un control, cambiar el color del texto o deshabilitarlo).

Ejemplo práctico: ocultamos txtApellido cuando el valor de txtNroEmpleado1 es igual a 2, y lo mostramos en cualquier otro caso.

Las reglas de formateo se pueden probar haciendo clic en Ejecutar Formulario, pero las validaciones solo se ejecutan al procesar el formulario, por lo que deben probarse desde el Dashboard de BIZUIT.

Estilos Personalizados y CSS

Además del formato estándar, podemos aplicar estilos personalizados mediante CSS.

- Desde el editor de estilos del control, escribimos reglas CSS que nos permiten modificar colores, bordes, fuentes, márgenes, etc.
- Esto nos da una capa de personalización avanzada para adaptar visualmente el formulario a criterios específicos de marca o usabilidad.

Propiedades de Eventos

Finalmente, los eventos permiten incorporar lógica dinámica al formulario, haciéndolo verdaderamente interactivo.

Los tipos de eventos varían según el control:

- **Change:** al cambiar un valor.
- **Input:** con cada entrada del usuario.
- **Click:** al hacer clic.
- **MouseOver:** al pasar el puntero.
- **Blur:** cuando se pierde el foco.

En cada evento, podemos escribir código en TypeScript. Por ejemplo, obtenemos el valor de txtNroEmpleado1 y lo usamos para construir un mensaje como "Id del Empleado ingresado: [valor]", que se muestra en otro control (txtDemo).



Resumen

Con esto completamos nuestro recorrido por la configuración de controles en el BIZUIT Forms Designer.

Aprendimos a definir propiedades básicas, conectar controles a datos, aplicar validaciones, modificar estilos, establecer reglas dinámicas y desarrollar comportamientos avanzados con eventos.

Estas herramientas nos permiten construir formularios potentes, inteligentes y ajustados a las necesidades reales de nuestros procesos.

3. Control Caja de Texto

A continuación nos enfocaremos en el control Caja de Texto, uno de los elementos más versátiles y fundamentales dentro de cualquier formulario en BIZUIT. Este control nos permite capturar datos de forma libre o estructurada, y gracias a su flexibilidad puede adaptarse a múltiples escenarios.

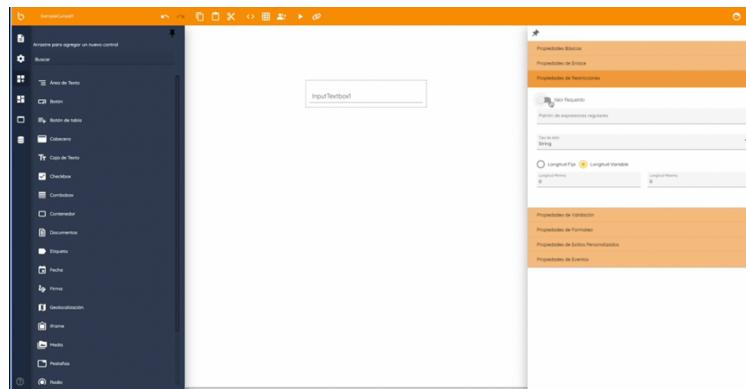
En particular, vamos a centrarnos en las propiedades de restricciones, que nos permiten validar y controlar la información que los usuarios ingresan, garantizando precisión y coherencia en los datos desde el primer momento.

Propiedades de Restricciones

Las restricciones se configuran directamente desde la hoja de propiedades del control y varían según el tipo de dato seleccionado.

Valor Requerido

Podemos activar la opción "Valor requerido" para forzar al usuario a completar ese campo antes de poder continuar con el formulario. Es una manera simple pero efectiva de asegurarnos de que la información clave no quede incompleta.



Patrón de Expresiones Regulares

Este campo nos permite definir un formato específico que debe cumplir el valor ingresado. Utilizando expresiones regulares (regex), podemos validar entradas como correos electrónicos, códigos específicos, números de teléfono, etc.

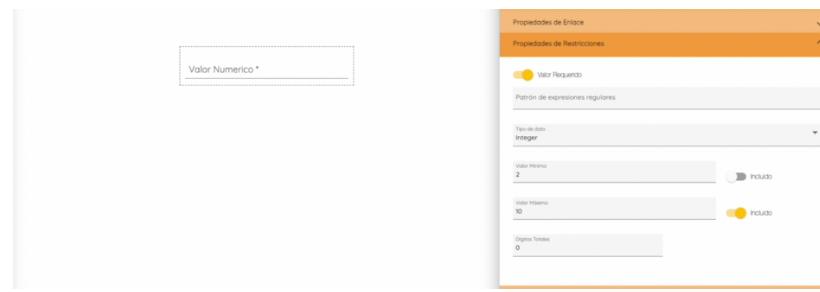
Por ejemplo: si requerimos una dirección de correo válida, configuraremos la expresión regular correspondiente. Si el usuario ingresa un valor que no cumple con el patrón, el control se resaltará y mostrará un mensaje de error contextual.

Selección del Tipo de Dato

El tipo de dato que seleccionemos para el control definirá qué restricciones adicionales podremos aplicar. Estas son las principales opciones:

a. Integer (Número Entero)

- Valor Mínimo y Máximo:** Establecemos un rango permitido (por ejemplo, entre 2 y 10).
- Inclusión de Límites:** Indicamos si el rango incluye o no los valores extremos.
- Dígitos Totales:** Limitamos la cantidad máxima de dígitos. Útil para campos como números de identificación o códigos postales.



b. String (Texto)

- **Longitud Fija:** El texto debe tener exactamente cierta cantidad de caracteres (por ejemplo, 5 caracteres para un código).
- **Longitud Variable:** Establecemos un rango mínimo y máximo (por ejemplo, entre 2 y 10 caracteres para nombres o apellidos).

c. Double (Decimal)

Permite ingresar números con decimales. Podemos establecer rangos mínimos y máximos, así como el número de decimales permitidos, según lo que el campo requiera.

d. DateTime (Fecha y Hora)

Este tipo de dato nos permite restringir fechas dentro de un rango específico. También podemos establecer fechas relativas, como "no permitir fechas anteriores a hoy".

Resumen

El Control Caja de Texto es una herramienta esencial para capturar entradas en formularios. Su capacidad para adaptarse a distintos tipos de datos, combinada con un sistema de validaciones robustas y personalizables, lo convierte en una pieza clave para garantizar la integridad de la información y mejorar la experiencia de los usuarios. Aplicando estas restricciones con criterio, podemos asegurarnos de que los datos sean ingresados correctamente desde el inicio del proceso, evitando errores aguas abajo y aumentando la eficiencia del flujo general.

4. Control Botón

A continuación exploraremos en detalle el control Botón en BIZUIT. Este elemento es esencial en cualquier interfaz, ya que permite ejecutar acciones concretas y adaptarlas al comportamiento que necesitamos dentro del formulario. Desde el tipo de botón hasta los eventos disponibles, veremos cómo personalizarlo para que cumpla su función de forma visualmente efectiva y funcionalmente precisa.

Propiedades Básicas del Botón

Las propiedades básicas definen la apariencia inicial del botón y su comportamiento general. Estos son los principales aspectos configurables:

1. Tipo de Botón

BIZUIT ofrece distintos estilos visuales para los botones, cada uno pensado para un uso específico dentro de la interfaz:

- **mat-button**: botón plano con solo texto.
- **mat-raised-button**: botón elevado con sombra, ideal para destacar.
- **mat-stroked-button**: botón con borde visible.
- **mat-flat-button**: botón plano sin borde ni sombra.
- **mat-icon-button**: botón con solo ícono, ideal para acciones rápidas o repetitivas.
- **mat-fab**: botón flotante grande, perfecto para acciones principales en dispositivos móviles.
- **mat-mini-fab**: versión pequeña del botón flotante.

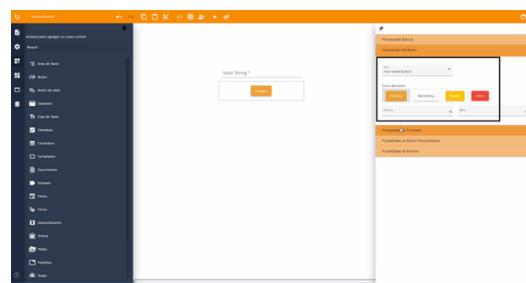


Elegir el tipo correcto ayuda a jerarquizar las acciones visualmente dentro del formulario.

2. Color del Botón

Podemos definir el color del botón según la intención de la acción:

- **Primary**: para acciones principales.
- **Secondary**: para acciones secundarias.
- **Accent**: para destacar interacciones especiales.
- **Warn**: para advertencias o acciones críticas (como eliminar).



Por ejemplo, para un botón "Eliminar", usar el color Warn mejora la claridad de la acción.

Además, podemos ajustar el ancho y alto del botón según el diseño general del formulario o las necesidades del dispositivo (porcentaje o valores fijos), asegurando una experiencia responsiva.

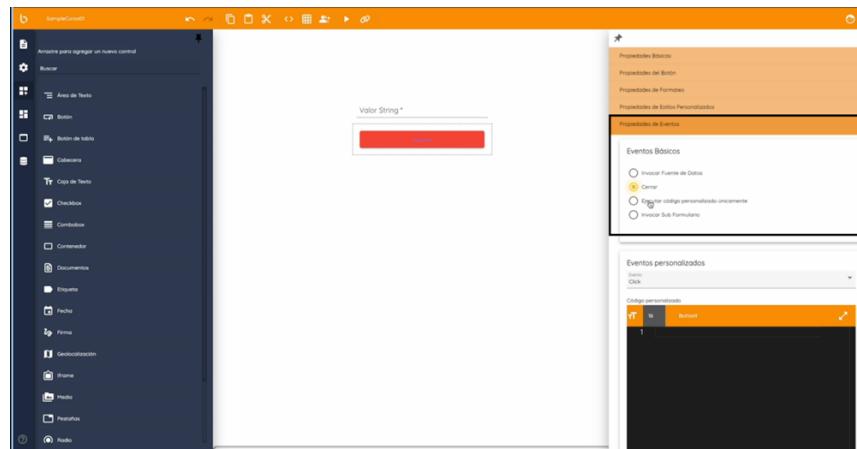
Configuración de Eventos del Botón

Los eventos determinan qué ocurre al hacer clic en el botón. Algunas opciones están disponibles tanto en formularios independientes como en formularios asociados a procesos; otras son exclusivas de formularios de proceso.

Opciones comunes:

- **Cerrar**: cierra el formulario actual.

- **Invocar Fuente de Datos:** ejecuta una fuente secundaria configurada. Podemos incluir validación previa y limpieza de controles después de la ejecución.
- **Ejecutar Código Personalizado:** nos permite agregar lógica avanzada escrita en TypeScript.
- **Invocar Subformulario:** abre un formulario modal dentro del principal, permitiendo intercambiar valores entre ambos.



Opciones específicas para formularios asociados a procesos:

- **Invocar Proceso:**
 - En formularios de inicio: inicia una instancia del proceso.
 - En actividades de interacción: continúa la ejecución del proceso. Los controles vinculados a parámetros o variables pasan automáticamente sus valores.
- **Ventana de Trazabilidad:** abre un modal que muestra el recorrido de la instancia por las actividades de tipo Interacción de Usuario hasta el punto actual.
- **Terminar Tarea:** finaliza la instancia del proceso, independientemente de en qué etapa se encuentre.

Casos de Uso y Buenas Prácticas

Configurar correctamente un botón no es solo una cuestión visual, sino funcional. Algunas recomendaciones clave:



- Usar colores críticos solo para acciones irreversibles (como Warn en eliminación).
- Asegurarse de que los botones tengan etiquetas claras que describan su función.
- Probar cada evento configurado para confirmar que responde como se espera.
- Adaptar el tamaño del botón al dispositivo: los FAB funcionan especialmente bien en pantallas táctiles.

Resumen

El control Botón en BIZUIT nos permite construir interfaces más dinámicas, interactivas y adaptadas a nuestras necesidades funcionales. Su variedad de estilos, su integración con fuentes de datos y procesos, y la posibilidad de ejecutar lógica personalizada lo convierten en una herramienta fundamental dentro del diseño de formularios.

Aplicando correctamente estas configuraciones, podemos crear experiencias de usuario intuitivas, eficientes y alineadas con nuestros objetivos de negocio.

5. Control Fecha

En este punto exploramos el control Fecha en BIZUIT, una herramienta clave para seleccionar, mostrar y validar fechas dentro de los formularios. Es especialmente útil en aplicaciones que gestionan eventos, vencimientos, plazos o cualquier tipo de dato temporal.

Gracias a sus propiedades configurables, podemos limitar rangos, aplicar formatos y establecer valores predeterminados que se adapten a las necesidades de cada flujo.

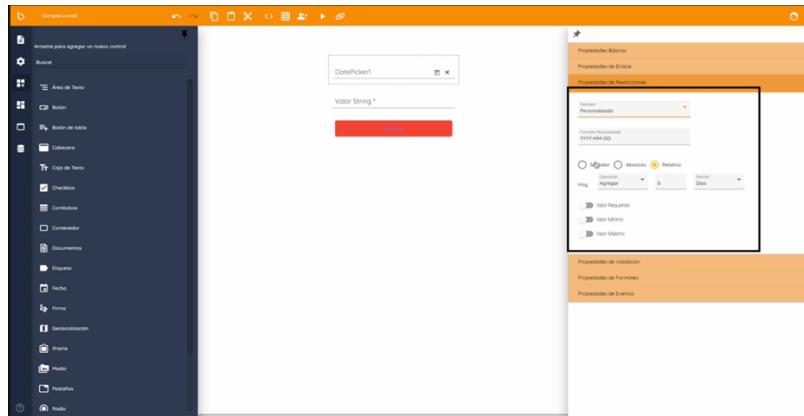
Propiedades de Restricciones

El control Fecha incluye múltiples configuraciones que nos permiten garantizar que el valor ingresado sea coherente y útil para el proceso en el que se enmarca.

1. Formato de Fecha

Podemos elegir cómo se mostrará la fecha seleccionada por el usuario. BIZUIT ofrece distintas opciones:

- **Corta:** formato reducido, por ejemplo MM/DD/YYYY.
- **Larga:** incluye texto completo, como "5 de noviembre de 2024".
- **Hora:** muestra la fecha junto con la hora seleccionada.
- **Personalizada:** definimos un patrón específico como YYYY-MM-DD.



Elegir el formato correcto mejora la comprensión y reduce errores por diferencias regionales o contextuales.

2. Valor Inicial: Sin valor, Absoluto o Relativo

Podemos definir si el campo se presenta inicialmente vacío, con una fecha fija o con una fecha relativa:

- **Sin valor:** el campo queda vacío hasta que el usuario seleccione una fecha.
- **Valor absoluto:** por ejemplo, "05/11/2024".
- **Valor relativo:** por ejemplo, "Hoy + 3 días".



Esto es útil para formularios que requieren una fecha predeterminada (como la fecha actual) o para procesos que anticipan un evento a futuro.



3. Valor Requerido

Al activar esta opción, el usuario deberá seleccionar una fecha obligatoriamente antes de enviar el formulario. Es una forma simple de asegurar que no se omita información crítica.

4. Valor Mínimo

Podemos limitar la fecha más temprana que el usuario puede ingresar. Esta restricción puede configurarse como:

- **Absoluta:** una fecha fija como "01/01/2023".
- **Relativa:** por ejemplo, "Hoy + 1 día".
- **Operaciones relativas:** nos permiten sumar o restar días, meses o años respecto a la fecha actual (por ejemplo: "Hoy - 30 días").

Esto nos permite evitar selecciones inválidas, como fechas pasadas en registros de eventos futuros.

5. Valor Máximo

De forma similar, también podemos establecer la fecha más lejana que se puede seleccionar:

- **Absoluta:** "31/12/2025"
- **Relativa:** "Hoy + 90 días".
- **Operaciones relativas:** al igual que en el valor mínimo, nos permiten adaptar las restricciones a reglas dinámicas o a la lógica del proceso.

Configurar un valor máximo evita que los usuarios ingresen fechas que estén fuera de un rango válido para el negocio.

Resumen

El control Fecha en BIZUIT es una herramienta imprescindible cuando trabajamos con datos temporales. Nos permite ofrecer una experiencia de usuario clara, garantizar que los valores ingresados estén dentro de los márgenes válidos, y adaptar el comportamiento del formulario a las reglas del proceso.



Con sus configuraciones de formato, rangos y valores dinámicos, este control aporta precisión y flexibilidad en cualquier flujo de trabajo.

6. Control ComboBox

El control Combobox de BIZUIT es una herramienta esencial cuando necesitamos que el usuario seleccione una opción de una lista desplegable. Gracias a su versatilidad, podemos poblarlo con datos estáticos o dinámicos, adaptarlo a diferentes flujos y mejorar significativamente la experiencia de uso en formularios.

Propiedades de Población

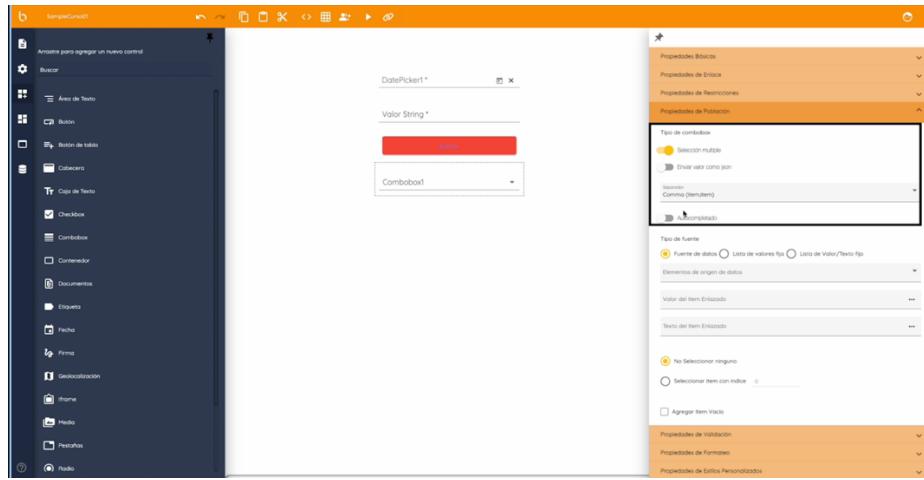
Las propiedades de población determinan cómo se llenan las opciones del Combobox. Estas pueden definirse manualmente (estáticas) o generarse dinámicamente desde fuentes externas.

Veamos en detalle las opciones disponibles.

1. Opciones de Población

Podemos configurar distintos comportamientos que modifican la forma en que el Combobox interactúa con el usuario:

- **Selección múltiple:** habilita al usuario a elegir más de una opción. Además, podemos configurar cómo se almacenan y concatenan las opciones seleccionadas, lo que resulta útil en flujos donde se requiere capturar múltiples entradas desde un solo control.
- **Enviar valor como JSON:** cuando está activa esta opción, el conjunto de opciones seleccionadas se envía en formato JSON al backend. Esto es ideal para integraciones con sistemas externos que esperan estructuras de datos complejas.
- **Autocompletado:** mientras el usuario escribe, el Combobox ofrece sugerencias basadas en la lista disponible. Esta funcionalidad agiliza la selección en listas largas y mejora la usabilidad.



2. Tipo de Fuente

La forma en que se cargan las opciones depende del tipo de fuente que definamos:

a. Fuente de Datos

Esta opción permite poblar el Combobox dinámicamente desde una fuente externa, como una base de datos o una API REST.

- Seleccionamos la fuente de datos secundaria que contiene los valores.
- Indicamos qué campo será utilizado como valor (lo que se almacena) y cuál como texto (lo que se muestra).
- Podemos incluir un ítem vacío para representar el caso en que el usuario no seleccione ninguna opción, asignándole un valor y una descripción personalizada.



Esta configuración es ideal cuando los valores deben mantenerse actualizados o provienen de sistemas externos.

b. Lista de Valores Fija

Podemos definir manualmente una lista estática de valores, útil en formularios donde las opciones no cambian con frecuencia.

Tipo de fuente

Fuente de datos Lista de valores fija Lista de Valor/Texto fijo

Valor	
Elemento fijo	+
Manzana	 
Pera	 

 No Seleccionar ninguno

c. Lista de Valor/Texto Fija

Aquí definimos pares clave-valor, donde:

- La clave es el dato que se guarda en el sistema.
- El texto es lo que ve el usuario en pantalla.

Tipo de fuente

Fuente de datos Lista de valores fija Lista de Valor/Texto fijo

Valor	Texto	
Valor del ítem Enlazado	Texto del ítem Enlazado	+
1	Si	 
0	No	 

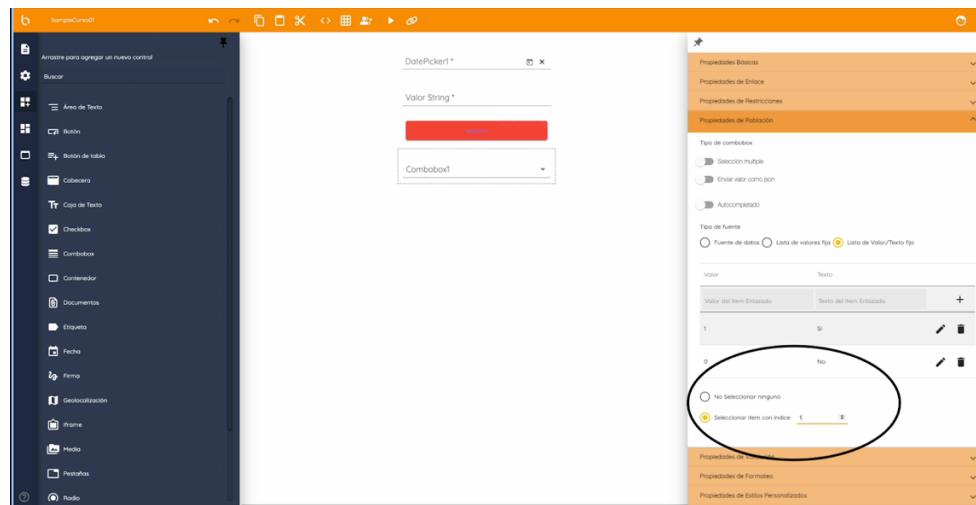
 No Seleccionar ninguno

Este enfoque nos permite manejar códigos internos (como IDs o siglas) sin afectar la claridad de la interfaz.

3. Opciones Adicionales

Además de definir cómo se cargan los datos, el Combobox nos ofrece otras opciones para ajustar su comportamiento:

- **Seleccionar ítem con índice:** podemos preseleccionar una opción por su posición en la lista. Esto es útil para formularios que requieren un valor por defecto o una opción sugerida.
- **Agregar ítem vacío:** permite mostrar una primera opción sin valor asignado, útil para obligar al usuario a tomar una decisión consciente y no dejar el valor preestablecido por accidente.



Resumen

El control Combobox es una herramienta poderosa para mejorar la interacción en formularios. Nos permite ofrecer listas claras, contextualizadas y fáciles de usar, ya sea mediante carga dinámica desde sistemas externos o con listas fijas adaptadas a la lógica del negocio.

Con sus propiedades de población, opciones de autocompletado y soporte para múltiples selecciones, podemos construir interfaces eficientes que respetan tanto las reglas del proceso como las necesidades del usuario final.

7. Control Etiqueta

En este punto vamos a explorar el control Etiqueta en BIZUIT. Aunque no es un componente interactivo, cumple una función clave dentro de los formularios: mostrar información de manera clara, contextual y visualmente integrada. Puede utilizarse como texto estático, mensaje informativo o incluso para exponer datos dinámicos del sistema o del entorno del usuario.



Gracias a su simplicidad y flexibilidad, este control se vuelve indispensable en cualquier diseño funcional.

Propiedades Básicas

El control Etiqueta nos permite definir distintos tipos de contenido a mostrar. Según el tipo seleccionado, podemos presentar datos estáticos o elementos que se actualizan dinámicamente al cargar el formulario.

1. Tipos de Etiqueta

A continuación detallamos los tipos disponibles y sus usos más comunes:

- **Etiqueta de Texto**

Este es el tipo más básico. Permite mostrar un mensaje fijo o un rótulo estático, como por ejemplo: "Bienvenido al sistema" o "Datos del Empleado". Es útil para títulos, instrucciones o secciones dentro del formulario.

- **Id de la Instancia**

Muestra el identificador único de la instancia del proceso en curso. Este dato puede ser útil para auditorías, referencias cruzadas o para que el usuario lo conserve como comprobante.

- **Fecha Actual:** Muestra la fecha del sistema al momento de cargar el formulario, en formato corto. Es útil para validar cuándo se realizó una acción, registrar la fecha de creación o simplemente brindar contexto.

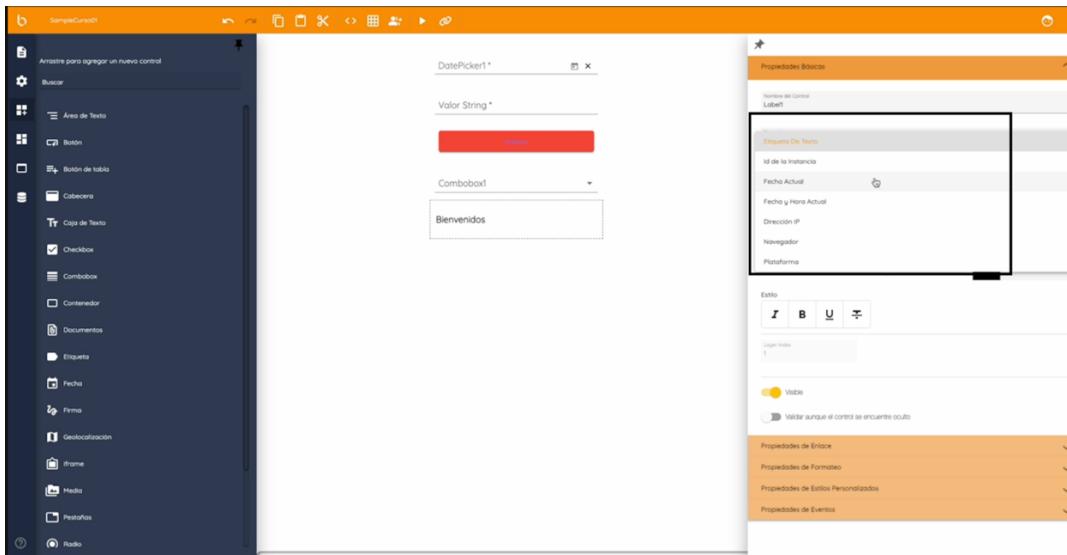
- **Fecha y Hora Actual:** Similar a la anterior, pero agrega el valor horario. Es ideal para formularios en los que se necesita mayor precisión temporal, como registros de actividad, validaciones o entradas críticas.

- **Dirección IP:** Permite mostrar la dirección IP del dispositivo desde el que se accede al formulario. Esto puede ser útil en casos de seguridad, trazabilidad o configuraciones contextuales.

- **Navegador:** Muestra el navegador utilizado por el usuario al acceder al formulario (por ejemplo, Chrome, Firefox). Este dato puede utilizarse para soporte técnico o para adaptar instrucciones según el entorno del usuario.

- **Plataforma:** Muestra el sistema operativo del usuario (por ejemplo, Windows 10, macOS). Esto puede ser útil para registros, auditorías o adaptaciones visuales específicas.

Podemos combinar distintos tipos de etiquetas en un mismo formulario para ofrecer al usuario una experiencia más completa y contextualizada, sin necesidad de interacción adicional.



Resumen

El control Etiqueta en BIZUIT es una herramienta simple pero poderosa para brindar contexto, reforzar la estructura del formulario y mostrar información útil y oportuna.

Ya sea para rótulos estáticos, fechas del sistema, datos técnicos del entorno o identificadores de procesos, este control mejora la presentación general del formulario y aporta claridad al usuario.

Con una configuración adecuada, podemos enriquecer la experiencia sin complicar el diseño ni saturar la interfaz. Una herramienta fundamental para quienes buscan formularios eficientes, organizados y enfocados en la información relevante.

8. Control Área de Texto

El control Área de Texto de BIZUIT es una herramienta pensada para capturar grandes volúmenes de texto, ideal para comentarios, descripciones, notas o cualquier tipo de entrada extensa. A diferencia de un campo de texto simple, este control permite ajustar su tamaño, validar la cantidad de caracteres y controlar la visibilidad del contenido según las necesidades del formulario.

Vamos a recorrer sus propiedades y configuraciones más relevantes para garantizar que el campo sea fácil de usar, visualmente claro y técnicamente validado.

Propiedades de Restricción

Las propiedades de restricción del Área de Texto nos permiten definir condiciones específicas sobre lo que el usuario puede o debe ingresar. Esto es esencial para evitar errores de carga, estandarizar los datos y facilitar su posterior procesamiento.

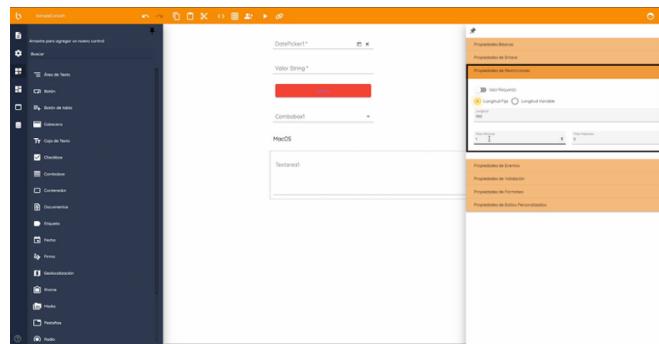
1. Valor Requerido

Podemos habilitar la validación "Valor requerido", que obliga al usuario a ingresar texto antes de continuar. Es una práctica común en campos donde se requiere justificar una acción o dejar asentado un comentario obligatorio.

2. Longitud Fija o Variable

Para asegurar una entrada adecuada, podemos limitar la cantidad de caracteres permitidos. Existen dos variantes:

- **Longitud Fija:** El texto debe tener un número exacto de caracteres. Por ejemplo, podemos exigir que se ingresen exactamente 100 caracteres para un campo de descripción formal.
- **Longitud Variable:** Podemos definir tanto una longitud mínima (ej. al menos 20 caracteres) como una longitud máxima (ej. hasta 500 caracteres). Esta configuración es útil para permitir cierta flexibilidad sin perder el control sobre la extensión del texto.



Estas restricciones se aplican automáticamente al momento de validar el formulario y pueden complementarse con mensajes de error personalizados si el usuario no cumple con los requisitos.

3. Filas Mínimas y Máximas

Además de validar el contenido, podemos controlar el tamaño visual del campo de texto:

- **Filas Mínimas:** Define la altura inicial del área de texto. Por ejemplo, 3 filas visibles al momento de cargar el formulario.
- **Filas Máximas:** Limita el número de líneas visibles antes de que se active el scroll interno. Esto nos permite mostrar más o menos contenido sin comprometer el diseño general del formulario.



Esta configuración mejora la usabilidad y evita formularios extensos innecesariamente si el texto no supera cierto volumen.

Resumen

El control Área de Texto es una solución versátil y personalizable que permite capturar contenido extenso de manera controlada.

Gracias a sus validaciones y ajustes de visualización, se adapta fácilmente a distintos escenarios de negocio, ya sea para ingresar descripciones de productos, observaciones internas o mensajes de los usuarios.

Con una configuración adecuada, no solo mejoramos la experiencia de carga sino que también garantizamos la coherencia y calidad de los datos ingresados.

9. Control CheckBox

El control Checkbox de BIZUIT es una de las herramientas más simples y potentes para capturar decisiones binarias en un formulario. Ya sea para aceptar términos y condiciones, activar funciones, o simplemente registrar un "sí/no", este componente es esencial en escenarios donde una simple selección define el curso de acción.

Vamos a explorar cómo configurar sus propiedades de restricción y validación, para que podamos integrarlo de forma precisa y personalizada en nuestros flujos.

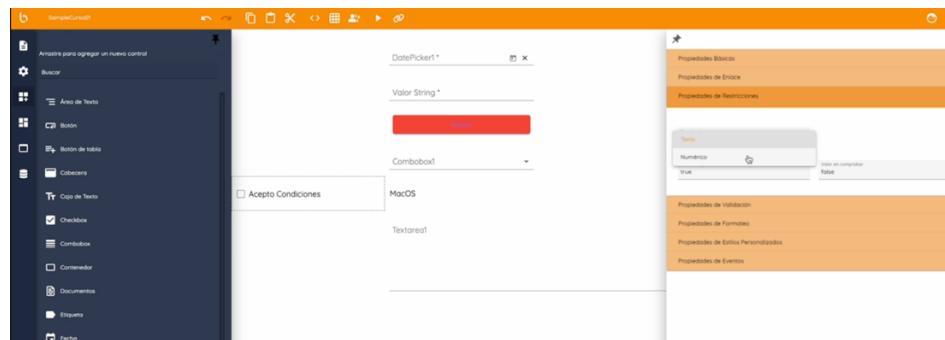
Propiedades de Restricción

El comportamiento del Checkbox se puede ajustar para reflejar con exactitud lo que cada estado (marcado o desmarcado) significa para el proceso o sistema. A continuación, repasamos las principales configuraciones.

1. Tipo de Validación

El tipo de validación define cómo será interpretado el estado del control dentro del formulario o proceso:

- Texto:** El estado del Checkbox se representa con valores como "Aceptado" o "Rechazado". Esta opción es ideal cuando se necesita registrar decisiones en forma de texto legible o almacenar estados más descriptivos.
- Numérico:** Utiliza valores como 1 para "marcado" y 0 para "desmarcado". Este formato es especialmente útil para integraciones con sistemas externos que operan sobre valores booleanos representados en formato numérico.

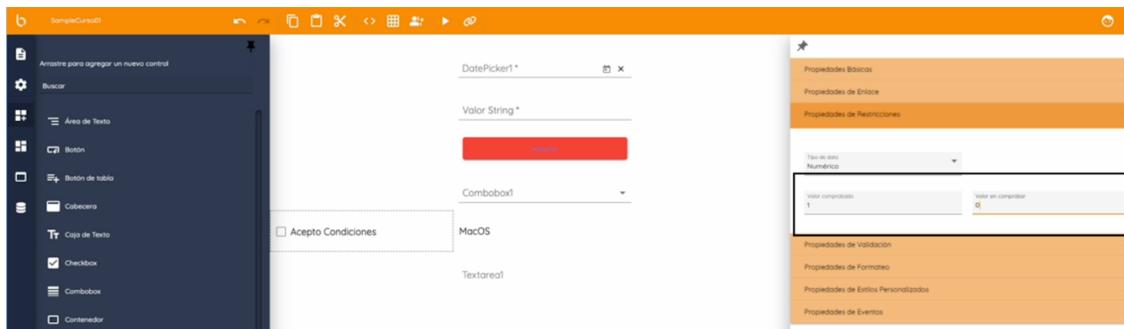


Estas configuraciones pueden combinarse con reglas de validación y formateo para generar comportamientos más complejos, como mostrar u ocultar otros controles dependiendo del estado del Checkbox.

2. Valores Personalizados

BIZUIT permite definir qué valor se enviará al sistema dependiendo del estado del Checkbox. Esto se configura en dos campos clave:

- **Valor Comprobado:** Es el valor que se enviará cuando el Checkbox esté **marcado**. Puede ser true, 1, "Sí", "Aceptado" o cualquier otro string o número según lo que el sistema necesite recibir.
- **Valor Sin Comprobar:** Es el valor que se enviará cuando el Checkbox esté **desmarcado**. De igual manera, puede ser false, 0, "No" o cualquier valor personalizado.



Esto habilita una integración directa y limpia con procesos automatizados, fuentes de datos externas o validaciones personalizadas.

Resumen

El control Checkbox es mucho más que un simple casillero para marcar. Su capacidad de adaptarse a distintos tipos de validación y devolver valores personalizados lo convierte en una herramienta versátil para todo tipo de formularios. Ya sea que estemos construyendo un proceso formal con múltiples condiciones o simplemente necesitemos registrar una preferencia del usuario, este componente cumple con eficiencia y claridad.

10. Control Slide Toggle

El control Slide Toggle en BIZUIT representa una alternativa visual y contemporánea al clásico interruptor binario. Gracias a su diseño intuitivo, permite alternar entre dos estados –como *Encendido* y *Apagado*– de forma clara y eficiente. En esta sección exploraremos sus propiedades de restricción, ideales para validar su comportamiento dentro de un formulario o integrarlo con otros sistemas.

Propiedades de restricción

Las propiedades de restricción del Slide Toggle permiten controlar su valor y definir cómo debe interpretarse en función de la lógica del sistema o los requerimientos del proyecto. Se dividen en dos grandes bloques: el tipo de validación y los valores personalizados.

1. Tipo de validación

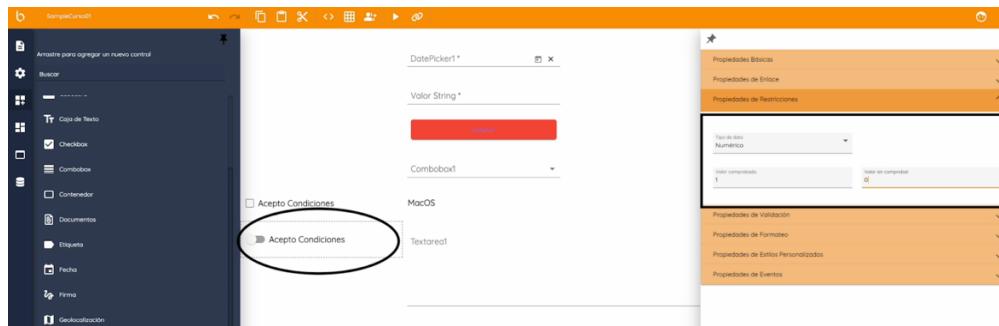
Este campo define cómo se interpretan los valores del Slide Toggle cuando cambia su estado:

- Texto:

Se puede configurar el control para que emita cadenas legibles como "Activado" y "Desactivado". Esta opción resulta útil cuando se necesita mostrar estados comprensibles para el usuario final o cuando se integran sistemas que requieren texto en vez de números.

- Numérico:

En este caso, el control se configura para emitir valores como 1 (para activado) y 0 (para desactivado). Es ideal para integraciones que manejan estados booleanos o condiciones binarias estrictamente numéricas.



2. Valores personalizados

Además del tipo de validación, también es posible definir qué valores específicos deben enviarse cuando el Slide Toggle se encuentra en uno u otro estado:

- **Valor comprobado:** Corresponde al valor que se asigna cuando el Toggle está activado. Puede tratarse de un número (1) o un texto personalizado como "Sí".
- **Valor sin comprobar:** Es el valor que se enviará cuando el Toggle esté desactivado. Al igual que el anterior, puede ser numérico (0) o textual, por ejemplo. "No".



Esta personalización es clave cuando se busca compatibilidad con bases de datos, APIs externas o reglas de negocio que exigen ciertos valores específicos para cada estado.

Resumen

El Slide Toggle es mucho más que un simple interruptor. Gracias a su diseño moderno, mejora la experiencia visual del usuario sin sacrificar funcionalidad.

Además, su capacidad para adaptarse a diferentes formatos de datos lo convierte en una herramienta versátil para cualquier tipo de proyecto dentro de BIZUIT.

11. Control Radio

El control Radio Button en BIZUIT está diseñado para capturar una única opción entre varias alternativas. Esto lo convierte en la herramienta ideal cuando se trata de decisiones mutuamente excluyentes dentro de un formulario. En esta sección, exploran sus propiedades principales, con énfasis en las opciones de población disponibles.

Propiedades de población

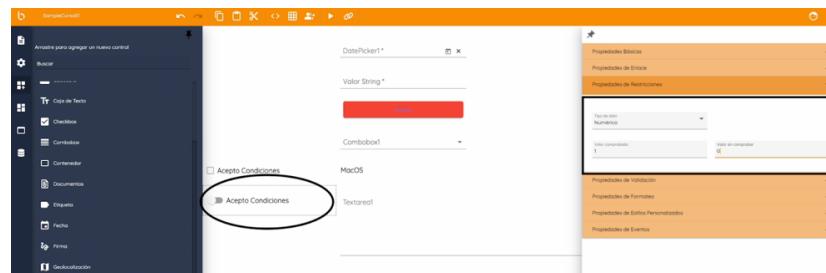
El comportamiento del Radio Button puede ajustarse según cómo se desea que se llenen sus opciones. BIZUIT ofrece distintas formas de definir esta lista, dependiendo del contexto o la fuente de los datos.

1. Tipo de fuente

Las opciones que verá el usuario pueden provenir de distintas fuentes:

- **Fuente de datos:** Esta opción permite vincular el control a una fuente de datos secundaria. Se puede seleccionar qué campo actuará como valor (lo que se guarda) y cuál como texto (lo que se muestra). Es especialmente útil en formularios que deben mantenerse sincronizados con bases de datos externas o que dependen de información dinámica.
- **Lista de valores fija:** Consiste en definir una lista estática de opciones. Es la opción ideal cuando las alternativas posibles no cambian y son conocidas de antemano (por ejemplo, días de la semana, niveles de prioridad, etc.).

- **Lista de valor/texto fijo:** Permite especificar pares clave-valor. Por ejemplo: 1: Rojo, 2: Amarillo, 3: Verde. En este caso, se almacena la clave (por ejemplo, 1) mientras que el usuario ve el texto correspondiente ("Rojo"). Esta separación entre lo que se muestra y lo que se guarda otorga mayor flexibilidad al momento de manejar datos internamente.



2. Seleccionar ítem con índice

Otra funcionalidad importante es la posibilidad de definir una opción predeterminada dentro del conjunto de botones. Esto se hace a través del índice de la lista, es decir, su posición.

Por ejemplo, si se quiere que "Rojo" esté seleccionado desde el inicio, basta con asignarle el índice 0 (asumiendo que es el primer elemento). Esto permite preconfigurar formularios de forma intuitiva y rápida.

Resumen

El control Radio Button en BIZUIT no solo simplifica la selección de opciones únicas, sino que además ofrece una gran versatilidad en la forma de poblar sus datos. Ya sea con listas fijas o mediante fuentes externas, permite mantener formularios claros, consistentes y alineados con la lógica del sistema.

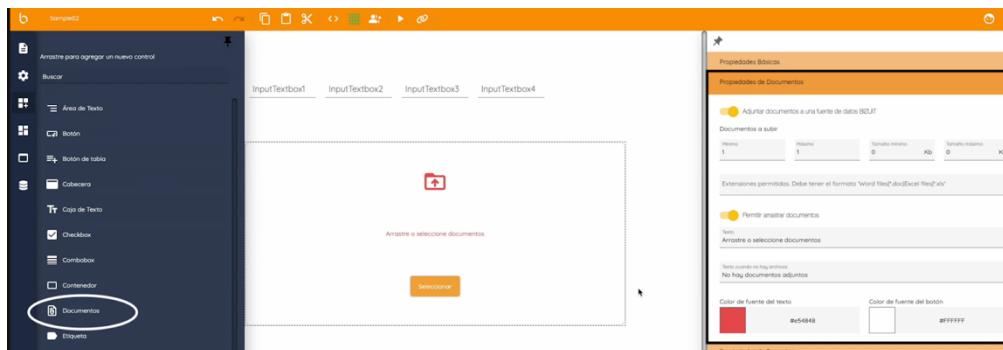
13. Control Documentos

El control Documentos de BIZUIT está diseñado para facilitar la carga, visualización y gestión de archivos dentro de formularios. Su configuración se adapta tanto a formularios independientes como a aquellos integrados en procesos, ofreciendo flexibilidad en múltiples escenarios. En esta sección, analizaremos sus propiedades según el contexto de uso.

1. En formularios independientes

Cuando el control Documentos se utiliza dentro de un formulario independiente, las opciones de configuración se centran en cómo y dónde se almacenarán los archivos.

- **Adjuntar documentos a una fuente de datos BIZUIT:** Si esta opción no está seleccionada, es posible especificar manualmente el ID de instancia BIZUIT al cual se asociarán los documentos. En cambio, si está seleccionada, los archivos se vincularán automáticamente a una fuente de datos secundaria del tipo BIZUIT, la cual debe configurarse previamente.
- **Documentos a subir:** Esta propiedad permite establecer la cantidad mínima y máxima de archivos que el usuario podrá o deberá cargar, así como el tamaño máximo permitido por archivo.
- **Extensiones permitidas:** Se puede restringir el tipo de archivos que se aceptarán, por ejemplo .pdf, .docx, .jpg, entre otros. Esto garantiza que se carguen solo los formatos compatibles con el proceso o sistema asociado.
- **Permitir arrastrar documentos:** Si se habilita esta opción, el usuario podrá arrastrar archivos directamente al control, además de usar el botón tradicional de carga. En este caso, también es posible personalizar el aspecto visual del área de carga.



2. En formularios asociados a procesos

Cuando el control Documentos se utiliza en un formulario que forma parte de un proceso, cambia ligeramente su configuración:



- **Mostrar documentos de la instancia actual:** Se puede indicar que los archivos estén vinculados a la instancia activa del proceso, o bien a otra instancia específica mediante el uso de otro control, parámetro o variable.
- **Permitir agregar documentos:** Si se habilita esta opción, los usuarios podrán cargar nuevos archivos durante la ejecución del proceso.
- **Permitir eliminar documentos existentes:** También es posible permitir que los usuarios eliminen archivos previamente cargados, brindando mayor control sobre el contenido vinculado a la instancia

Resumen

El control Documentos ofrece una solución robusta y flexible para la gestión de archivos dentro de formularios. Ya sea en un contexto autónomo o como parte de un proceso estructurado, permite definir límites claros, establecer tipos de archivo permitidos, y controlar acciones como agregar, eliminar o visualizar documentos.

14. Control Cabecera

El Control Cabecera en BIZUIT permite estructurar y personalizar la parte superior de un formulario o pantalla, integrando logotipos, títulos, subtítulos y menús. Esta herramienta no solo cumple un rol estético, sino también funcional, ya que facilita la organización del contenido y mejora la experiencia de usuario. En esta sección, recorremos sus propiedades principales y cómo configurarlas para adaptarlas a cada proyecto.

Propiedades del encabezado

El control está compuesto por cinco elementos configurables: logotipo, título, subtítulo, menú y el orden de los componentes. Cada uno puede ajustarse en contenido, estilo y disposición.

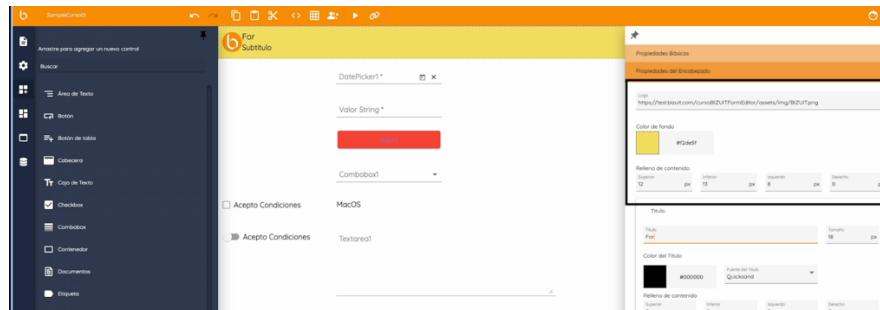
1. Logotipo

Este elemento permite identificar visualmente la aplicación o formulario:

- **Origen de la imagen:** Se puede cargar un logotipo mediante una URL o insertar una imagen codificada en formato base64.



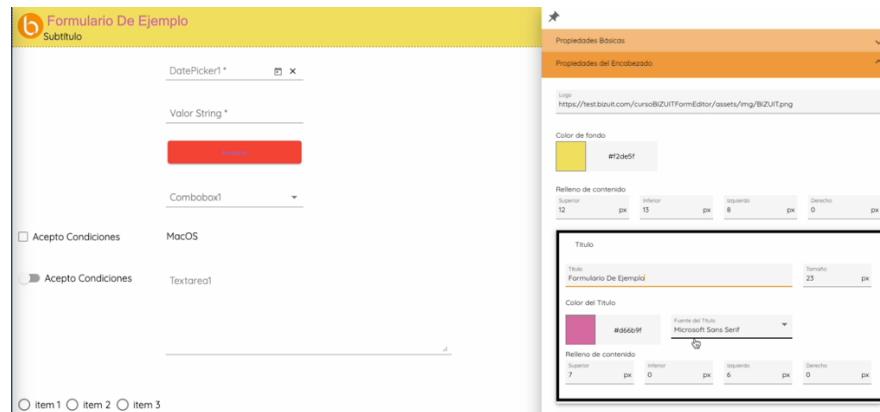
- **Color de fondo:** Define el fondo del encabezado, útil para destacar el logotipo o alinear la estética con la identidad institucional.
- **Relleno de contenido:** Permite ajustar los márgenes superiores, inferiores y laterales en píxeles, mejorando el espaciado visual.



2. Título

Ubicado generalmente junto al logotipo, presenta el nombre o tema principal del formulario:

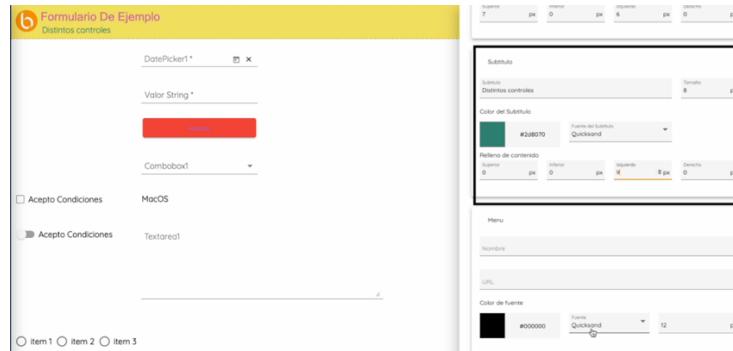
- **Texto:** Se define el contenido del título principal.
- **Estilo:** Se pueden personalizar el color, la fuente y el tamaño del texto. También es posible ajustar los márgenes para controlar el espaciado en relación con los demás elementos del encabezado.



3. Subtítulo

Actúa como complemento del título, proporcionando información adicional o contextual:

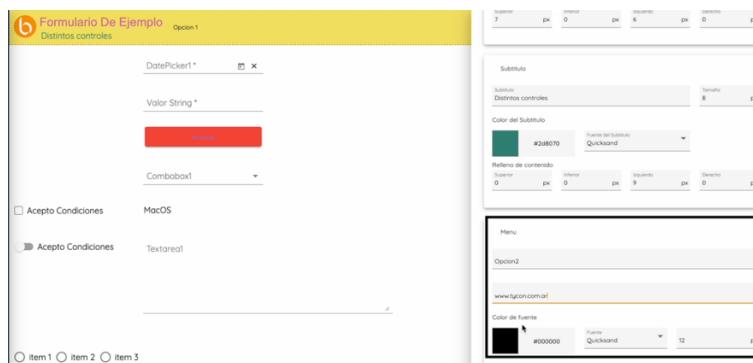
- **Texto:** Sirve para añadir detalles relevantes que aclaren el propósito del formulario o sección.
- **Estilo:** Al igual que el título, permite ajustar fuente, color, tamaño y márgenes.



4. Menú

Este componente ofrece acceso rápido a secciones clave o recursos externos:

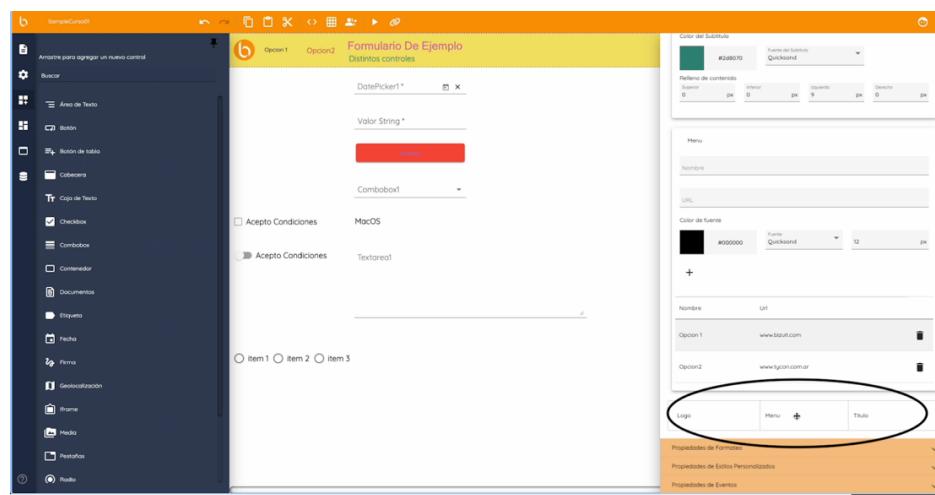
- **Opciones del menú:**
 - **Nombre:** Es el texto visible en cada opción.
 - **URL:** Especifica la dirección de destino. Es importante que incluya `http://` o `https://`, ya que de lo contrario se interpretará como una ruta relativa a la base del sistema.
 - **Estilo:** Color, fuente y tamaño del texto totalmente personalizables.
- **Gestión dinámica:** Las opciones pueden añadirse, editarse, eliminarse o reordenarse mediante simple arrastre en el editor.





5. Orden de los componentes

BIZUIT permite reorganizar fácilmente los elementos del encabezado para adaptarlos al diseño deseado. Solo es necesario arrastrar los componentes (logotipo, título, subtítulo, menú) en el editor visual hasta lograr la disposición ideal.



Resumen

El Control Cabecera es una pieza fundamental en la construcción de interfaces claras, accesibles y coherentes. Su capacidad de integrar elementos visuales, textos informativos y enlaces funcionales permite consolidar una experiencia de usuario profesional desde el primer vistazo. Ya sea en formularios simples o pantallas más complejas, este control asegura identidad, orden y navegabilidad.

15. Control Tabla

El Control Tabla en BIZUIT es una herramienta fundamental para presentar grandes volúmenes de información estructurada dentro de formularios. Gracias a su flexibilidad y nivel de personalización, permite construir desde simples listados hasta tablas interactivas, adaptadas a las necesidades específicas de cada proyecto.

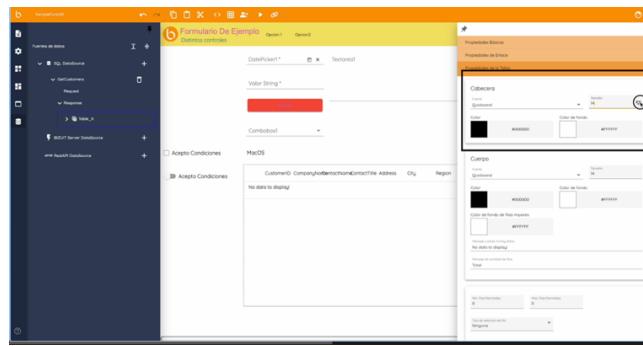
Creación y vinculación

El proceso comienza al arrastrar al formulario una estructura de datos proveniente de una fuente secundaria (como una consulta SQL o una variable de actividad anterior). Al hacerlo, se genera automáticamente un control Tabla con las columnas correspondientes, ya enlazado a esa fuente de datos.

1. Propiedades de Cabecera

La cabecera permite definir el estilo visual de los títulos de columna, mejorando la organización y legibilidad del contenido.

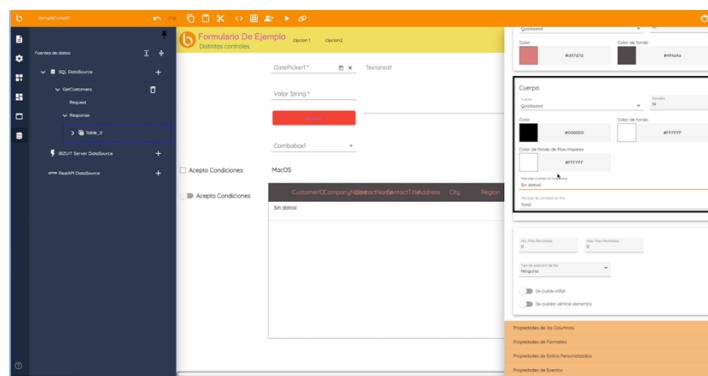
- **Fuente y tamaño:** Posibilita seleccionar el tipo de letra y su tamaño.
- **Colores de texto y fondo:** Se pueden establecer colores personalizados para alinear visualmente la tabla con el diseño del formulario.



2. Propiedades del Cuerpo

Esta sección se enfoca en el contenido de la tabla: las filas y celdas donde se presentan los datos.

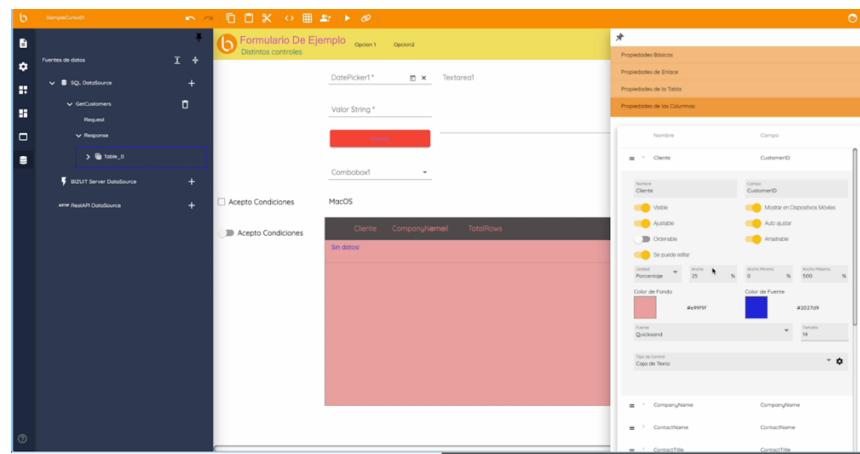
- **Fuente y tamaño del texto:** Permite ajustar el estilo del contenido mostrado.
- **Colores personalizados:** Se pueden definir tanto para el texto como para el fondo, incluyendo esquemas de color alternados para facilitar la lectura.
- **Mensajes personalizados:**
 - **Sin datos:** Texto que se muestra cuando la tabla está vacía.
 - **Cantidad de filas:** Texto que indica el número total de registros visibles.



3. Configuración de Filas

Esta sección permite controlar cómo se comportan y visualizan las filas de la tabla:

- **Cantidad mínima y máxima de filas:** Se puede limitar el número de registros visibles.
- **Tipo de selección de fila:**
 - *Ninguna:* Sin interacción.
 - *Singular:* Solo se puede seleccionar una fila.
 - *Múltiple:* Permite seleccionar varias filas (Ctrl+Click, Click múltiple o CheckBoxes).
- **Personalización visual:** Se pueden definir colores específicos para filas seleccionadas y para el efecto hover (cuando el cursor pasa sobre ellas).



4. Acciones Habilitadas

La tabla puede configurarse para que los usuarios no solo visualicen, sino también interactúen con los datos:

- **Edición de filas:** Habilita la modificación directa de los registros desde el formulario.
- **Agregar elementos:** Permite sumar nuevas filas manualmente.
- **Eliminar elementos:** Brinda la opción de quitar registros innecesarios o erróneos.



5. Propiedades de las Columnas

Cada columna del control puede configurarse de forma independiente:

- **Configuraciones generales:**

- *Nombre y campo:* Identifican la columna y su origen de datos.
- *Visible:* Determina si la columna debe mostrarse.
- *Mostrar en dispositivos móviles:* Optimiza la tabla para pantallas reducidas.

- **Ajuste del ancho:**

- *Ajustable:* El usuario puede modificar el ancho manualmente.
- *Auto ajustar:* Se adapta al contenido de forma automática.
- *Unidad de medida:* Se puede usar píxeles o porcentajes.
- *Ancho mínimo y máximo:* Define límites para evitar columnas demasiado anchas o estrechas.

- **Interactividad:**

- *Ordenable:* Permite ordenar los datos haciendo clic en el encabezado.
- *Arrastrable:* Posibilita reorganizar las columnas mediante arrastre.

- **Edición en línea:** Las celdas pueden volverse editables y asociarse a diferentes controles como cajas de texto, selectores, o checkboxes.

Resumen

El Control Tabla es indispensable para construir formularios potentes y usables en BIZUIT. Desde encabezados estilizados hasta celdas editables, cada aspecto puede ser ajustado para adaptarse a diferentes escenarios de negocio. Su capacidad para gestionar datos de forma clara, dinámica y eficaz lo convierte en uno de los componentes más completos del sistema.

16. Control Botón de Tabla

Una de las herramientas más prácticas que ofrece BIZUIT cuando trabajamos con formularios que incluyen tablas es el control Botón de Tabla. Este componente nos permite gestionar registros de manera rápida y ordenada, ya sea para agregar, editar o eliminar datos directamente desde el formulario. Gracias a su diseño, optimiza el flujo de trabajo y mejora notablemente la experiencia del usuario.

Veamos cómo funciona y qué posibilidades nos brinda.

Acciones sobre la Tabla

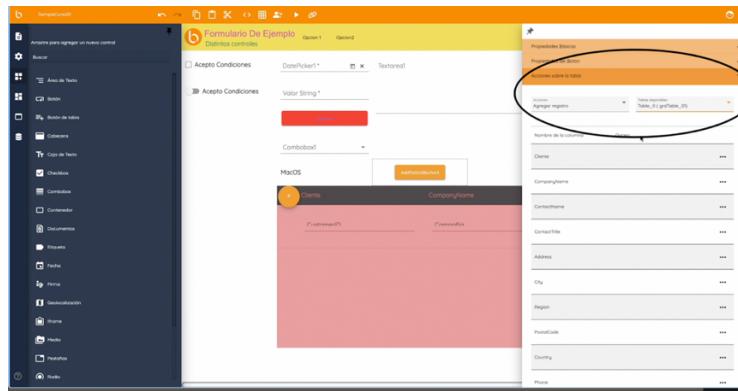
Cuando incorporamos un Botón de Tabla, lo primero que debemos hacer es vincularlo a una tabla específica del formulario. Esto se hace desde una lista desplegable, y es un paso clave: cada acción que configuremos con ese botón estará limitada exclusivamente a esa tabla, lo que nos da un control más preciso y evita errores.

A partir de ahí, podemos configurar tres tipos de acciones:

1. Agregar Registro

Esta acción nos permite añadir nuevos datos a la tabla de forma automática. Para configurarla:

- Indicamos qué valores queremos asignar a cada columna de la tabla.
- Esos valores pueden venir de constantes, de otros controles del formulario o de parámetros del sistema.



Por ejemplo, podríamos establecer un valor constante como "TEST" para el campo CustomerId, y al mismo tiempo vincular el campo CompanyName con lo que ingrese el usuario en un InputTextbox1.

Al ejecutar el formulario, al hacer clic en el botón, se agrega un nuevo registro con esos valores definidos.

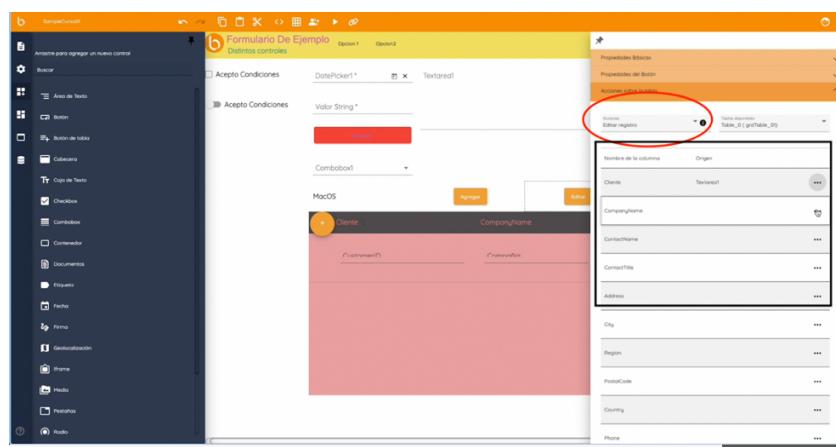
2. Editar Registro

Con esta acción, podemos modificar un registro existente. Su funcionamiento es muy intuitivo, pero hay un requisito fundamental: Primero debemos seleccionar el registro que queremos editar.

Una vez seleccionado, podemos:

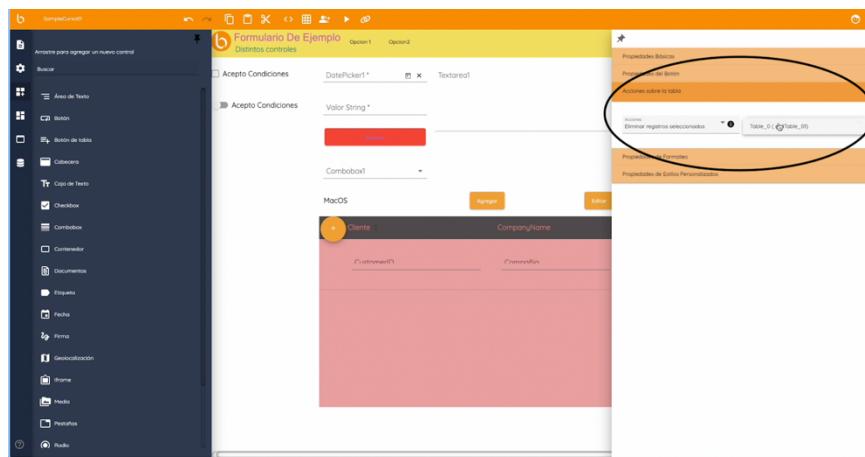
- Vincular los valores de las columnas a controles del formulario.
- Al seleccionar el registro, los controles toman automáticamente el valor correspondiente.
- Si modificamos el contenido del control, ese nuevo valor se actualiza automáticamente en la tabla.

Por ejemplo, podemos hacer que el campo CustomerId se vincule a un TextArea1 y que CompanyName se relacione con InputTextbox1.



3. Eliminar Registros Seleccionados

Esta acción nos permite eliminar uno o varios registros de la tabla de forma eficiente. Solo es necesario seleccionar las filas que queremos borrar y hacer clic en el botón. Ideal para mantener la tabla limpia y actualizada.





Asociar el Botón a una Tabla

Es importante recordar que cada Botón de Tabla debe estar asociado a una tabla específica, lo cual se define desde una lista desplegable en las propiedades del control. Esto nos garantiza que las acciones que configuremos no interfieran con otras tablas del formulario.

Resumen

El control Botón de Tabla es una pieza clave cuando necesitamos trabajar con registros dentro de un formulario. Nos permite:

- Agregar datos nuevos
- Editar registros existentes.
- Eliminar registros seleccionados.

Además, al poder vincular valores a constantes, controles o parámetros del sistema, y limitar cada botón a una tabla específica, ganamos en flexibilidad y control.

Una herramienta sencilla, pero poderosa, que no puede faltar en nuestros formularios dinámicos.

17. Control Pestaña

Cuando trabajamos con formularios complejos, necesitamos una forma práctica y visual de organizar la información. Ahí es donde entra en juego el control de pestañas. Este componente nos permite dividir el formulario en secciones diferenciadas, facilitando la navegación y mejorando la experiencia del usuario.

Propiedades Básicas

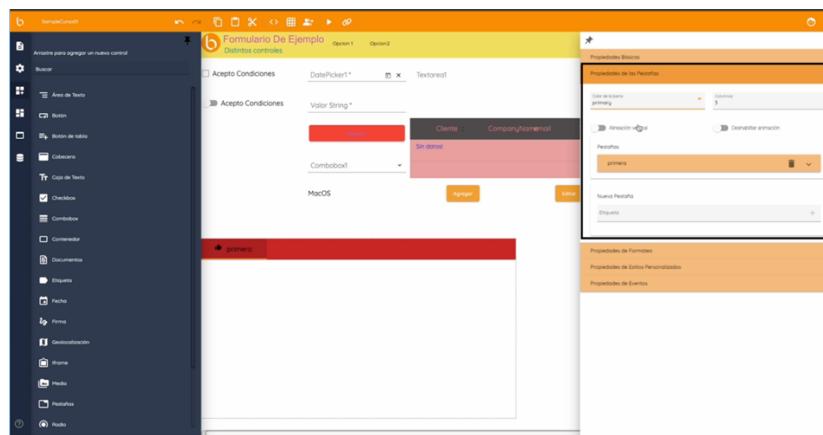
Apenas agregamos el control de pestañas, accedemos a una serie de propiedades que nos permiten personalizar su apariencia general:

- **Color de fondo:** podemos definir un color que destaque el área de las pestañas o que armonice con la estética general del formulario. Esto ayuda a separar visualmente las secciones y guiar al usuario.

Propiedades de las Pestañas

Cada pestaña se puede personalizar de forma independiente para que se adapte al diseño que estamos construyendo. Algunas de las opciones más relevantes son:

- **Color de la barra:** elegimos entre valores predefinidos del tema como Primary, Secondary, Accent o Warn, según la intención visual que queramos transmitir.
- **Columnas:** podemos definir cuántas columnas estarán disponibles dentro de cada pestaña. Esto es muy útil para organizar los controles de manera ordenada.
- **Alineación vertical:** nos permite ajustar cómo se distribuyen los elementos en la pestaña, lo que puede ser clave para ciertas interfaces.
- **Deshabilitar animación:** cuando necesitamos una experiencia más rápida o minimalista, podemos eliminar las transiciones suaves entre pestañas.

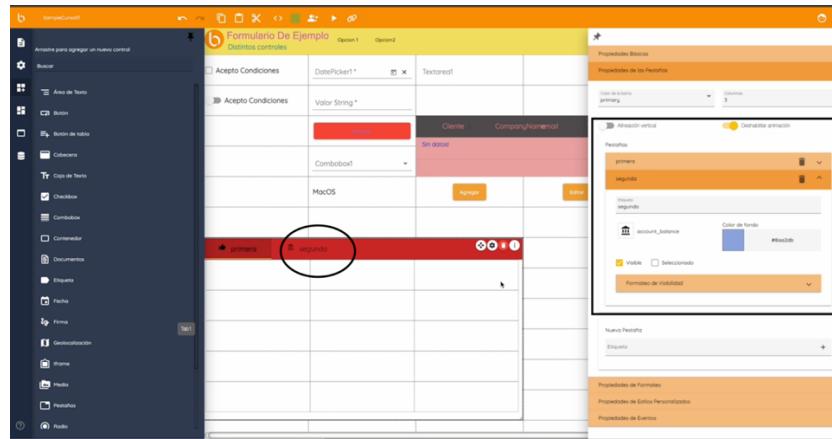


Gestión de Pestañas

Además del diseño, cada pestaña tiene una serie de configuraciones individuales que determinan su comportamiento y visibilidad:

- **Etiqueta:** es el texto que se muestra al usuario para identificar la pestaña.
- **Ícono:** podemos sumar un ícono al lado del texto para reforzar la idea o función de la pestaña (por ejemplo, un ícono de usuario para una sección de perfil).
- **Color de fondo:** otra capa de personalización para diferenciar visualmente cada pestaña.
- **Visible:** define si esa pestaña se muestra o no. Ideal cuando trabajamos con reglas condicionales.

- **Seleccionado:** indica qué pestaña queremos que esté activa al cargar el formulario.



Dentro de cada pestaña, simplemente arrastramos y soltamos los controles que vamos a necesitar. Esto nos permite construir secciones completas y funcionales de manera modular.

Formato de Visibilidad

Una de las funcionalidades más potentes de este control es la posibilidad de configurar reglas dinámicas de visibilidad.

Podemos hacer que ciertas pestañas aparezcan u oculten en función de condiciones que definimos según los datos del formulario.

Las reglas se construyen combinando:

- **Condiciones:** basadas en controles, operadores (como igual, diferente, mayor que, etc.) y valores definidos.
- **Unión de condiciones:** podemos combinar múltiples condiciones con lógica AND o OR.
- **Acción:** definimos si la pestaña debe mostrarse o esconderse cuando se cumplan esas condiciones.

Un ejemplo sencillo sería ocultar una pestaña si el usuario escribe la palabra "HIDE" en un campo específico. Este tipo de lógica nos permite adaptar la interfaz en tiempo real, mejorando la usabilidad del formulario.

Resumen

En definitiva, el control de pestañas es una herramienta esencial cuando queremos ordenar información y mejorar la experiencia del usuario dentro de formularios más complejos. Nos permite dividir, ocultar, mostrar y personalizar secciones de forma muy flexible.

Gracias a sus amplias posibilidades de configuración –desde la estética hasta el comportamiento dinámico–, podemos construir formularios más limpios, intuitivos y eficaces.

18. Control Tarjeta

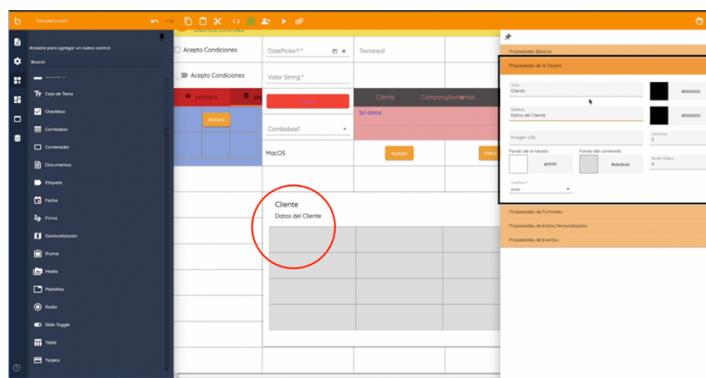
En esta sección vamos a conocer el control Tarjeta, un elemento visual muy útil cuando queremos agrupar y destacar información clave dentro de un formulario. Gracias a su diseño estructurado y atractivo, nos permite organizar datos de manera clara y jerarquizada.

Propiedades del control Tarjeta

Al incorporar una tarjeta en un formulario, veremos que contamos con varias propiedades configurables. Vamos a repasarlas una por una para entender cómo aprovecharlas al máximo.

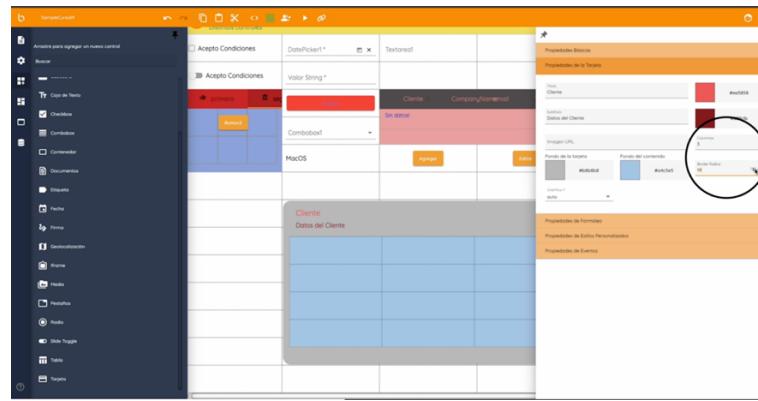
1. Título y subtítulo

Podemos asignar un título principal que actúe como encabezado de la tarjeta. También es posible incluir un subtítulo, útil para brindar información complementaria. En ambos casos, podemos personalizar el color del texto utilizando valores hexadecimales. Por ejemplo, #000000 para negro.



2. Columnas

Esta propiedad nos permite definir la cantidad de columnas internas dentro de la tarjeta. Así, podemos distribuir en paralelo diferentes controles (como texto, botones o imágenes), logrando un diseño más ordenado y visualmente equilibrado.



3. Fondo de la tarjeta y del contenido

El control nos ofrece dos opciones para trabajar con colores de fondo:

- **Fondo de la tarjeta:** modifica el color general del componente.
- **Fondo del contenido:** afecta únicamente al área donde colocamos los elementos internos.



Esto nos da la posibilidad de jugar con contrastes y jerarquías visuales.

4. Bordes redondeados (Border Radius)

Mediante esta propiedad podemos definir el estilo de las esquinas:

- 0 px: esquinas rectas.
- 4 px, 8 px o más: esquinas redondeadas, que suavizan el diseño general.



5. Overflow Y

Cuando el contenido dentro de la tarjeta supera su tamaño visible, podemos controlar su comportamiento vertical con la propiedad Overflow Y:

- **auto**: agrega una barra de desplazamiento.
- **hidden**: oculta el contenido excedente.
- **visible**: permite que el contenido se desborde sin restricciones.

6. Imagen

También podemos incluir una imagen especificando una URL. Esto es muy útil si queremos destacar visualmente una sección o complementar la información textual con contenido gráfico.

Resumen

El control Tarjeta es una herramienta muy versátil dentro del ecosistema BIZUIT. Nos permite presentar información de forma clara, atractiva y adaptable al diseño general de nuestros formularios. Si ajustamos correctamente sus propiedades, podemos transformar una sección básica en una experiencia visual más ordenada y profesional.

19. Control Contenedor

Cuando diseñamos formularios en BIZUIT, muchas veces necesitamos agrupar, ordenar o simplemente estructurar visualmente los distintos elementos que lo componen. Para eso, contamos con una herramienta fundamental: el control Contenedor.

Este componente nos permite no solo agrupar controles dentro de una misma sección, sino también personalizar su apariencia y disposición para mejorar la organización y la experiencia visual del formulario.

Vamos a recorrer una por una sus propiedades configurables para entender cómo usarlo de forma efectiva.

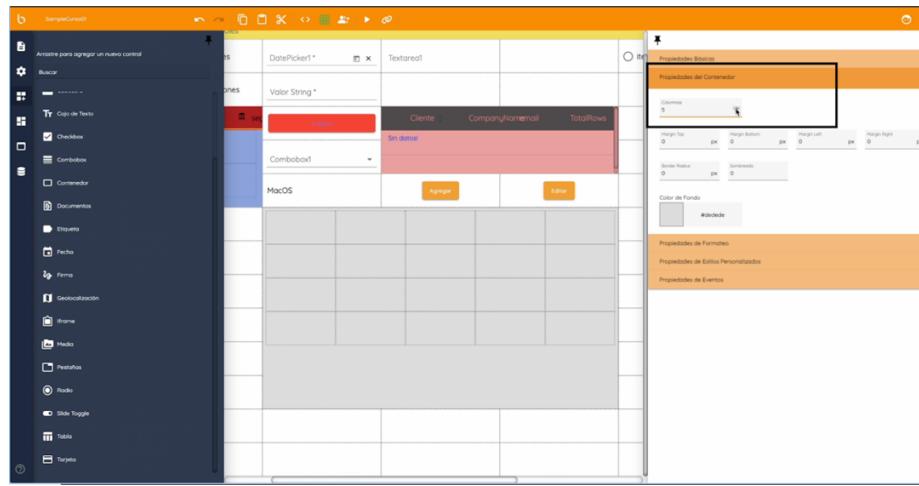
Propiedades del Contenedor

A través de sus opciones de configuración, el control Contenedor nos da mucha flexibilidad para definir tanto la estética como la funcionalidad del espacio que agrupa.

1. Columnas

Una de las propiedades más útiles es la posibilidad de definir cuántas columnas tendrá el contenedor. Esto nos permite distribuir los controles internos en un formato de grilla, de forma ordenada y adaptable.

Por ejemplo, si configuramos el contenedor con 5 columnas, los controles que agreguemos se acomodarán automáticamente según esa estructura.



2. Márgenes (Margin Top, Bottom, Left, Right)

Podemos controlar el espacio exterior que rodea al contenedor mediante los márgenes:

- **Top y Bottom:** ajustan el espacio vertical.
- **Left y Right:** controlan el espacio horizontal respecto a otros elementos.

Esto resulta muy útil para mantener una separación visual clara entre secciones o componentes.

3. Radio de Borde (Border Radius)

El radio de borde define cuán redondeadas estarán las esquinas del contenedor.

- Si usamos un valor de 0 px, el diseño tendrá esquinas rectas.
- A medida que aumentamos el valor (por ejemplo, 8 px), las esquinas se redondean, dando una apariencia más moderna y suave.



4. Sombreado

Otra propiedad estética interesante es el sombreado, que agrega profundidad al contenedor.

- Un valor de 0 implica sin sombra, ideal para diseños planos.
- Valores más altos aumentan la intensidad de la sombra, generando un efecto tridimensional que hace que el contenedor se destaque sobre el fondo.

Aplicar un sombreado sutil puede ayudar a guiar visualmente al usuario hacia bloques importantes del formulario.

5. Color de Fondo

Podemos personalizar el color de fondo del contenedor mediante un código hexadecimal, lo cual nos permite adaptarlo al diseño general del formulario.

Por ejemplo:

- #dedede para un gris claro.
- #ffffff para blanco.
- O cualquier otro color que necesitemos según la identidad visual del proyecto.

Esto también ayuda a distinguir visualmente secciones o grupos de controles.

Resumen

El control Contenedor es mucho más que un simple agrupador de controles. Nos permite:

- Organizar visualmente los formularios.
- Aplicar estilos coherentes.
- Trabajar con estructuras en columnas.
- Reutilizar o mover grupos de controles con facilidad.

Gracias a sus propiedades personalizables, se convierte en una herramienta esencial cuando buscamos formularios ordenados, claros y visualmente atractivos.

20. Control IFrame

En ciertos proyectos, necesitamos incrustar contenido externo dentro de nuestros formularios: desde una página web hasta gráficos, dashboards o incluso aplicaciones embebidas. Para eso, en BIZUIT contamos con el control IFrame.

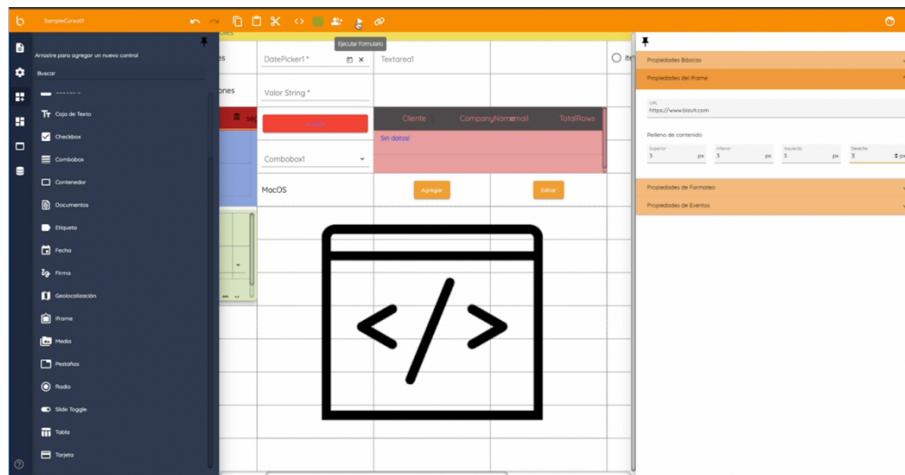
Este componente nos permite integrar ese contenido sin que el usuario tenga que abandonar la interfaz del formulario, haciendo que la experiencia sea mucho más fluida y profesional.

Propiedades del IFrame

Una vez que agregamos un control IFrame al formulario, podemos definir dos propiedades clave para que funcione correctamente:

1. URL

Esta es la propiedad más importante: nos permite especificar qué contenido se va a mostrar dentro del IFrame.



Al hacerlo, el IFrame cargará esa página directamente dentro del formulario, permitiendo que el usuario interactúe con ella sin salir del entorno de BIZUIT.

2. Relleno de Contenido

Otra opción útil es la posibilidad de configurar el relleno interno del IFrame. Esto define el espacio entre los bordes del componente y el contenido que se visualiza.



Por ejemplo, podemos aplicar un padding de 3 píxeles en todos los lados para que el contenido no quede pegado al borde y se vea más limpio.

Esto es especialmente útil cuando embebemos recursos con interfaces propias, como gráficos, mapas o formularios externos, ya que mejora la estética general.

Resumen

El control IFrame es una herramienta sencilla, pero muy potente, que nos permite ampliar las posibilidades de nuestros formularios con contenido externo dinámico. Desde páginas informativas hasta soluciones de terceros, podemos integrarlo todo sin perder cohesión en el diseño.

Gracias a su fácil configuración, se convierte en una excelente opción cuando necesitamos mostrar información externa de forma integrada y profesional.

21. Control Geolocalización

Cuando necesitamos trabajar con mapas, ubicaciones o coordenadas específicas dentro de un formulario, el control Geolocalización se convierte en una herramienta indispensable.

Este componente nos permite visualizar posiciones geográficas, vincular datos con ubicaciones y capturar coordenadas ingresadas o seleccionadas por el usuario.

Propiedades del Control Geolocalización

El control cuenta con varias opciones de configuración que determinan cómo se visualiza y cómo interactúa el usuario con el mapa.

1. Marcador e Información

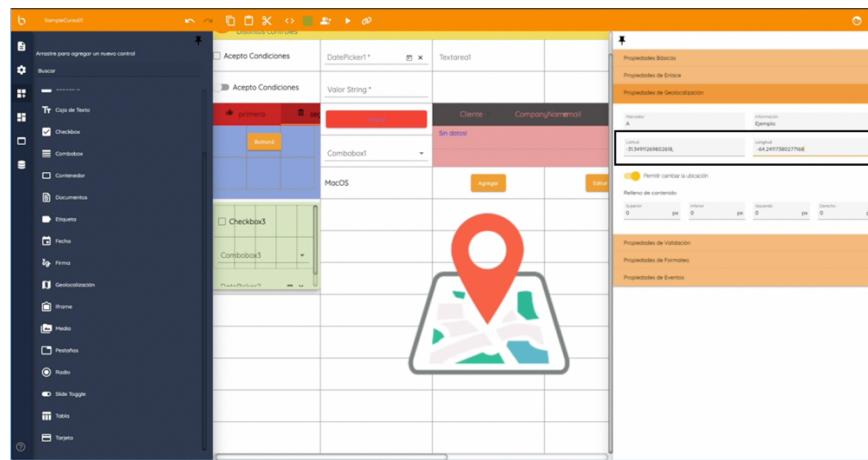
Una vez insertado el control, podemos configurar un marcador que señale una ubicación específica dentro del mapa.

- Marcador: representa visualmente el punto exacto en el mapa. Por ejemplo, podríamos ubicar una empresa como "TYCON SA" en sus coordenadas reales.
- Información: nos permite asociar un texto descriptivo al marcador, como "Oficina Principal" o una dirección específica. Esto enriquece la experiencia visual y aporta contexto

2. Latitud y Longitud

Otra de las configuraciones clave es la posibilidad de definir coordenadas exactas para posicionar el marcador.

- Latitud: valor entre -90 y 90 que determina la ubicación vertical.
- Longitud: valor entre -180 y 180 para la ubicación horizontal.



Podemos escribir manualmente estos valores o asociarlos a controles o variables, lo cual resulta útil cuando el mapa debe actualizarse según datos ingresados previamente.

3. Permitir Cambiar la Ubicación

Esta opción nos permite definir si el usuario puede mover el marcador directamente sobre el mapa:

- Si está activado, el usuario puede arrastrar el marcador, y las coordenadas se actualizan automáticamente.
- Si está desactivado, el marcador permanece fijo, útil cuando queremos solo mostrar una ubicación sin posibilidad de modificarla.

4. Relleno de Contenido

Por último, podemos configurar el relleno interno del control, es decir, los márgenes que separan el mapa del borde del componente.

Esto se define en píxeles para cada lado (superior, inferior, izquierdo y derecho), y nos permite lograr un diseño más limpio y cómodo visualmente.



Casos de Uso Comunes

El control Geolocalización puede aplicarse en múltiples contextos. A continuación, repasamos algunos de los más habituales:

- Aplicaciones de Logística: para ubicar puntos de entrega o recolección, permitiendo al usuario seleccionar la dirección exacta.
- Sistemas de Rastreo: mostrar ubicaciones en tiempo real, como la posición de un vehículo, un dispositivo o una persona.
- Mapeo Interactivo: permitir que el usuario marque ubicaciones relevantes como sucursales, propiedades, sitios de interés, etc.

Resumen

El control Geolocalización aporta una capa visual y funcional muy potente a nuestros formularios. Nos permite trabajar con datos geográficos de forma sencilla, interactiva y adaptable.

Ya sea para mostrar una ubicación, permitir que el usuario seleccione una, o asociar coordenadas con información dinámica, este componente es clave para proyectos que requieren integración con mapas.

22. Control Firma

Cuando un formulario requiere algún tipo de validación personal, ya sea para aprobar condiciones, confirmar entregas o firmar digitalmente un documento, el control adecuado es el control de Firma.

Este componente permite que el usuario dibuje su firma directamente sobre el formulario, ofreciendo una solución simple, moderna y totalmente digital para procesos que antes requerían papel o instancias externas.

Veamos cómo se configura y en qué casos resulta especialmente útil.

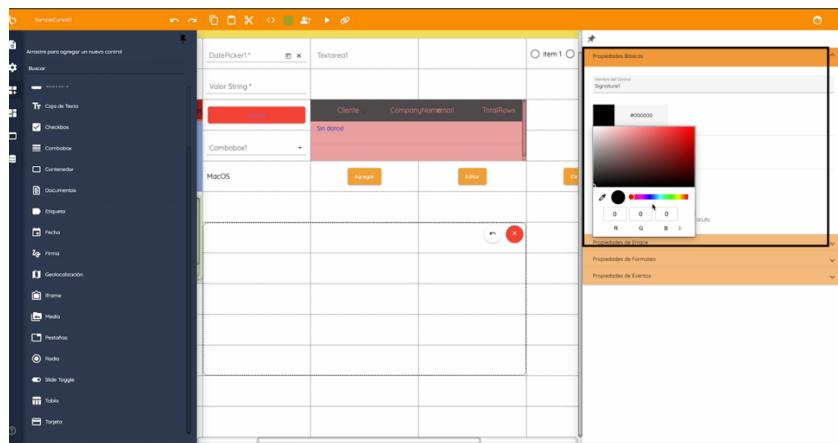
Propiedades Básicas

El control de Firma es fácil de configurar y se adapta bien a distintas situaciones. Entre sus propiedades más relevantes encontramos:

Color de Tinta

Podemos personalizar el color de la tinta utilizada para la firma.

- Por defecto, se utiliza el color negro.
- Sin embargo, es posible elegir otros colores –por ejemplo, azul o rojo– según el estilo visual del formulario o las preferencias del cliente.



Esto puede parecer un detalle menor, pero ayuda a reforzar la estética general o a diferenciar visualmente las firmas entre distintos usuarios.

Casos de Uso Comunes

El control de Firma se integra perfectamente en numerosos flujos de trabajo, especialmente aquellos donde se requiere algún tipo de autenticación o validación directa del usuario. A continuación, repasamos algunos ejemplos:

1. Aprobaciones y Consentimientos

Ideal para situaciones donde el usuario debe aceptar términos y condiciones, validar información o autorizar un procedimiento. Con este control, puede firmar directamente en el formulario, lo que agiliza el proceso y le da validez inmediata.

2. Procesos de Entrega

También resulta muy útil en procesos logísticos: por ejemplo, cuando necesitamos que el receptor confirme la entrega de un producto o servicio. La firma digital actúa como constancia de recepción, reemplazando el papel tradicional.

3. Formularios Electrónicos



En formularios generales, como:

- Inscripciones a eventos.
- Autorizaciones médicas.
- Encuestas con secciones de validación.

El control de Firma se convierte en una herramienta versátil que mejora la presentación del formulario y facilita la recolección de datos firmados sin necesidad de salir del entorno digital.

Resumen

El control de Firma nos permite incorporar validaciones personales de forma profesional, simple y completamente digital. Aporta fluidez a los procesos, mejora la experiencia del usuario, nos ayuda a reducir el uso de papel y los tiempos de procesamiento.

Gracias a su facilidad de uso y a sus opciones de personalización, se integra sin problemas en todo tipo de formularios modernos, ya sea para aprobaciones, entregas o registros.

23. Control Media

Cuando queremos enriquecer nuestros formularios con contenido multimedia –ya sean imágenes, videos, audios o incluso acceso a la cámara–, el control Media de BIZUIT es la herramienta indicada.

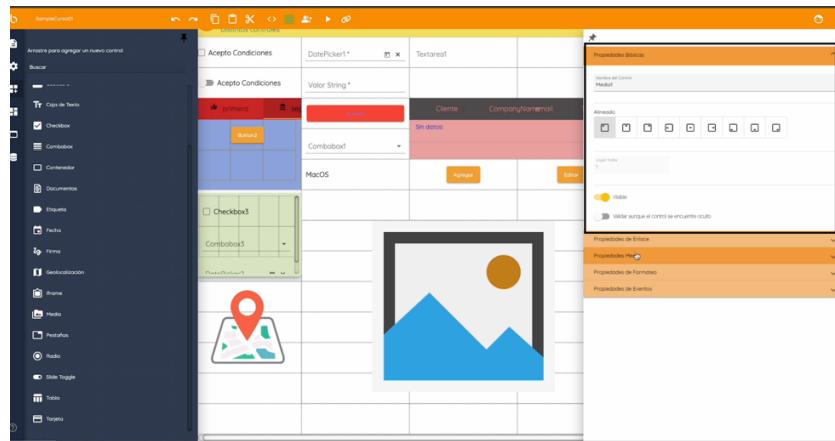
Este componente nos permite integrar elementos visuales y sonoros de forma directa en la interfaz, aportando dinamismo y mejorando la experiencia del usuario.

Propiedades Básicas

Antes de definir el tipo de medio que queremos incorporar, el control Media nos permite ajustar algunas propiedades generales relacionadas con su presentación:

- **Alineado:** Define la posición del contenido dentro del área del control (izquierda, centro o derecha), útil para lograr una disposición precisa en el diseño.

- **Layer Index:** Establece la prioridad del control sobre otros elementos en la misma pantalla. Cuanto mayor el valor, más arriba se muestra.
- **Visible:** Permite indicar si el contenido se mostrará desde el inicio o si su visibilidad dependerá de una condición dinámica.



Propiedades según el tipo de contenido

Una de las ventajas principales del control Media es que admite múltiples tipos de contenido, cada uno con sus propias propiedades específicas. Vamos a verlas una por una.

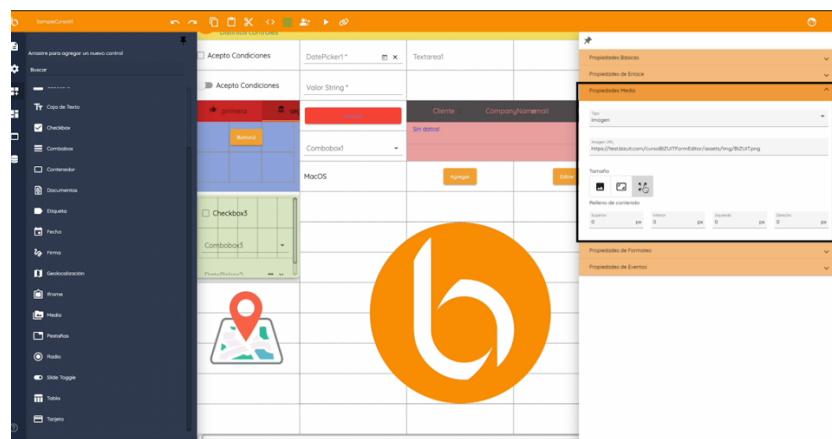
1. Imagen

Ideal para mostrar elementos visuales estáticos como logos, banners, gráficos o ilustraciones.

Propiedades específicas:

- **Imagen URL:** Ruta del archivo de imagen a mostrar (puede ser una URL externa o un archivo cargado en el sistema).
- **Tamaño:** Define cómo se adapta la imagen al contenedor:
 - Ajustar contenido.
 - Escalar proporcionalmente
 - Llenar completamente

- Relleno de contenido:** Margen interno entre la imagen y los bordes del control, útil para lograr un diseño más limpio.

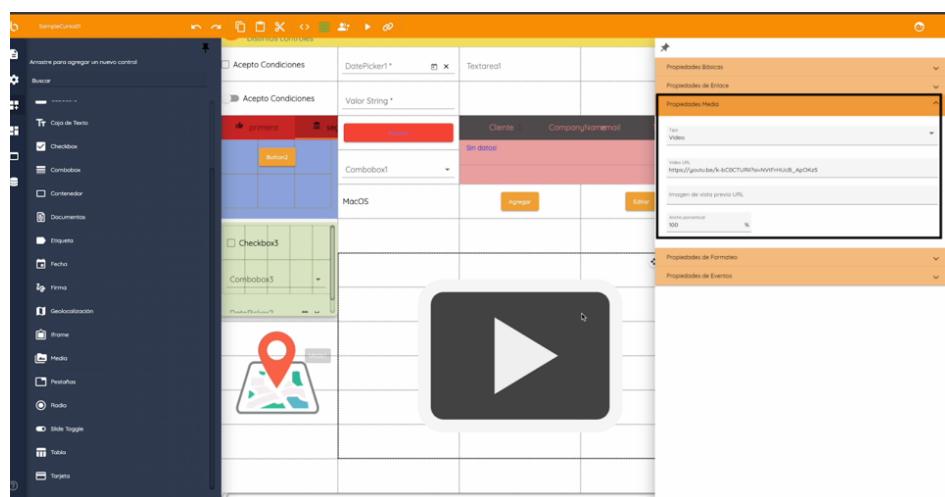


2. Video

Muy útil para incorporar contenido dinámico como tutoriales, demostraciones o presentaciones visuales.

Propiedades específicas:

- Video URL:** Ruta del archivo de video.
- Imagen de Vista Previa URL:** Imagen estática que se muestra antes de iniciar la reproducción, funcionando como una miniatura.
- Ancho porcentual:** Permite establecer qué porcentaje del ancho disponible ocupará el video.

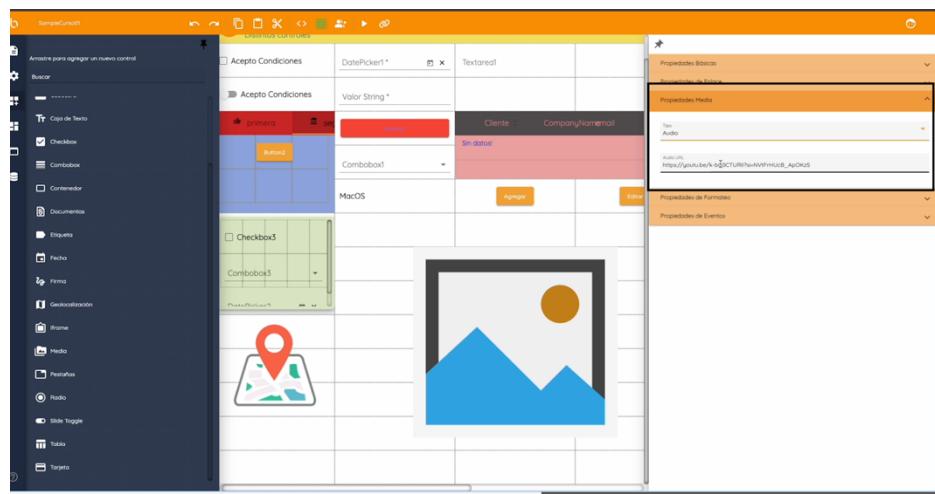


3. Audio

Cuando queremos incorporar música, mensajes de voz, podcasts o efectos de sonido, esta es la opción indicada.

Propiedad específica:

- **Audio URL:** Ruta del archivo de audio que se va a reproducir. El control incluye un reproductor incorporado para que el usuario pueda iniciar, pausar o controlar el volumen directamente.

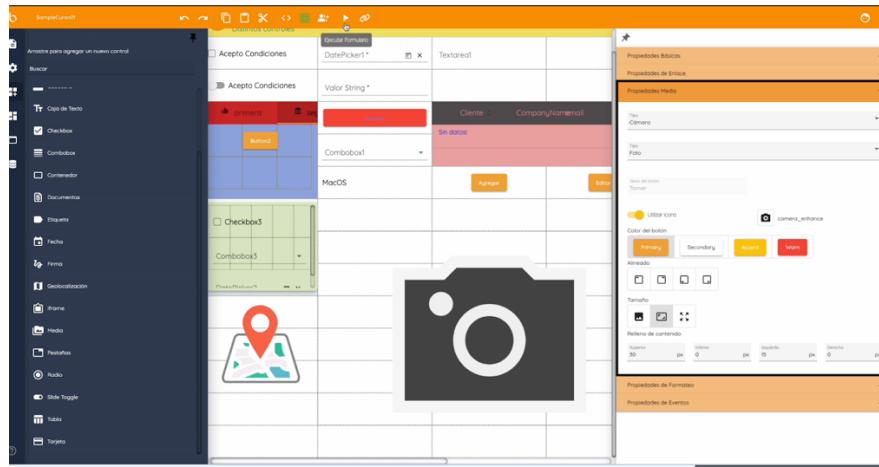


4. Cámara

Una opción avanzada que nos permite activar la cámara del dispositivo, ya sea para capturar imágenes o leer códigos QR directamente desde el formulario.

Propiedades específicas:

- **Tipo:** Indicamos si el uso será para capturar una foto o para escanear códigos QR.
- **Tamaño:** Al igual que en las imágenes, permite definir cómo se adapta el contenido al área disponible.
- **Relleno de contenido:** Podemos añadir márgenes internos para una mejor integración visual.
- Además, podemos incluir botones personalizados con íconos para iniciar la cámara o capturar la imagen, mejorando la interacción del usuario.



Resumen

El control Media es una herramienta sumamente versátil que nos permite integrar contenido multimedia en nuestros formularios de forma ordenada, funcional y visualmente atractiva.

Ya sea que necesitemos mostrar una imagen, reproducir un video, incluir un audio o acceder a la cámara del dispositivo, este componente nos ofrece todas las opciones necesarias para construir experiencias interactivas dentro de nuestras aplicaciones.

Conclusión

A lo largo de esta unidad, recorrimos en detalle una amplia variedad de controles disponibles en BIZUIT, entendiendo cómo configurarlos, personalizarlos y combinarlos para construir formularios potentes, visualmente atractivos y altamente funcionales.

Aprendimos a trabajar con elementos básicos como cajas de texto, encabezados y formularios, y avanzamos hacia componentes más complejos como tablas, botones de acción, pestañas y controles de geolocalización, todos ellos fundamentales para gestionar datos e interacciones de forma eficiente.

También exploramos controles visuales que elevan el diseño, como tarjetas, medios multimedia e IFrames, y herramientas específicas para capturar información, como el control de Firma o la lectura de códigos QR mediante la cámara.



Durante todo el recorrido, vimos cómo usar propiedades, enlaces, validaciones y condiciones para adaptar cada componente a las necesidades de cada proyecto.

Con este conocimiento, estamos preparados para diseñar formularios dinámicos y profesionales, integrando datos, automatizando procesos y mejorando la experiencia del usuario final.

BIZUIT nos da la flexibilidad para responder a casi cualquier escenario, y ahora contamos con las herramientas para aprovecharlo al máximo.

¡Felicitaciones por completar esta unidad! Nos vemos en la próxima, donde exploraremos en profundidad el uso de fuentes de datos secundarias y cómo aprovecharlas para alimentar nuestros formularios con información dinámica y contextual.



Unidad 3: Fuentes de Datos Secundarias

En esta unidad abordaremos las Fuentes de Datos Secundarias en BIZUIT, explorando cómo utilizar y configurar distintos tipos de fuentes de datos para integrar información externa y procesos internos directamente en los formularios.

Las fuentes de datos secundarias en BIZUIT son herramientas que permiten interactuar con bases de datos externas, procesos nativos de la plataforma o APIs directamente desde los formularios, facilitando consultas, validaciones y visualización de datos en tiempo real. Estas fuentes enriquecen la experiencia del usuario al integrar información dinámica sin depender de la lógica del proceso principal. Pueden ser de tipo SQL, BIZUIT o RestAPI, y su uso optimiza la eficiencia y flexibilidad en la gestión de datos dentro de los formularios.

Veremos cómo utilizar:

- Fuentes SQL, para consultar y trabajar con bases de datos externas.
- Fuentes BIZUIT, que permiten ejecutar lógica nativa de la plataforma dentro del formulario.
- Fuentes RestAPI, ideales para conectarnos con servicios web externos mediante métodos HTTP.

Además, aprenderemos a optimizar el rendimiento de nuestras consultas mediante caché, evitando llamadas innecesarias.

Las fuentes de datos secundarias son clave para construir formularios más dinámicos, integrados y eficientes, sin necesidad de modificar el flujo del proceso.

Fuente de datos secundarias SQL

En BIZUIT, las fuentes de datos secundarias SQL nos permiten interactuar con bases de datos externas directamente desde el formulario, sin depender de la lógica del proceso. Esto nos da la posibilidad de consultar, insertar, actualizar o eliminar datos en tiempo real, integrando información dinámica y enriqueciendo la experiencia del usuario.



Este tipo de fuente es ideal para:

- Cargar listas dinámicas.
- Validar datos ingresados por el usuario.
- Mostrar información externa sin alterar la estructura del proceso

En esta sección, vamos a ver paso a paso cómo configurar y utilizar una fuente SQL con un ejemplo práctico.

Ejemplo: Obtener Información de un Cliente

1. Configuración inicial

Creamos un control de texto donde el usuario pueda ingresar un identificador de cliente. Luego, añadimos una fuente de datos secundaria SQL, que llamaremos GetCustomerInfo.

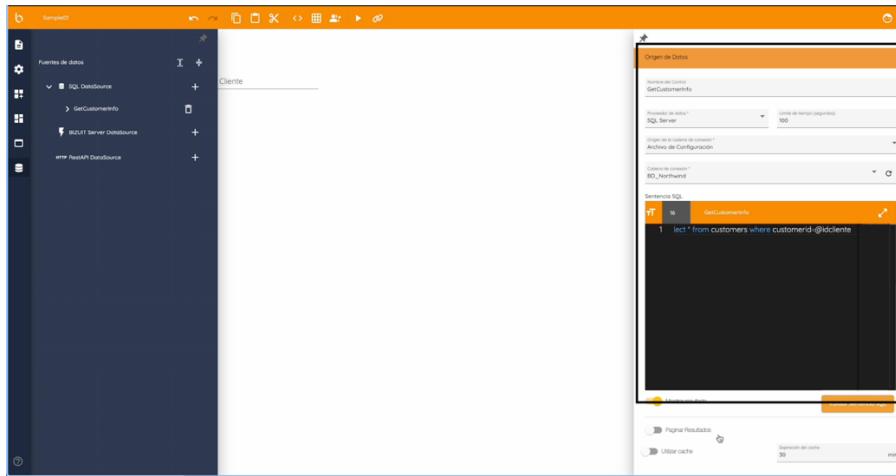
Desde la barra lateral “Fuentes de Datos”, seleccionamos SQL y configuramos la conexión, ya sea cargando un archivo o escribiendo manualmente la cadena de conexión.

2. Consulta SQL

Escribimos una consulta con parámetros, por ejemplo:

```
SELECT * FROM Customers WHERE CustomerId = @IdCliente
```

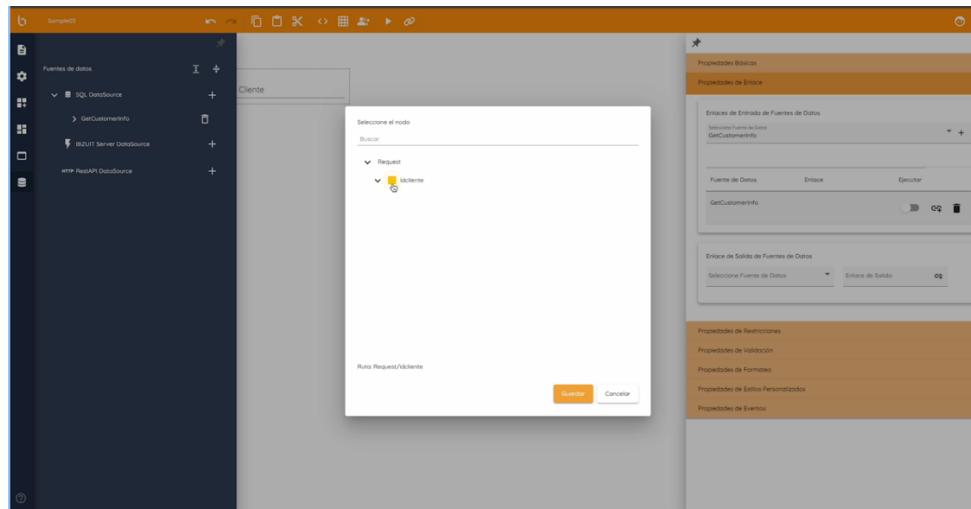
BIZUIT reconocerá automáticamente los parámetros y la estructura de salida de la consulta.



3. Enlace del control al parámetro

En las propiedades del campo de entrada:

- Usamos el ícono de enlace para asociar el parámetro @IdCliente con el valor del control.
- Indicamos que la fuente de datos se ejecute al cambiar el valor del campo.



4. Visualización de los resultados

Arrastramos los campos de respuesta al formulario. Por ejemplo:

- Si arrastramos CompanyName, se creará automáticamente un campo enlazado a ese valor.
- También podemos usar controles ya existentes y enlazarlos manualmente a cualquier campo devuelto, como ContactName.



Resumen

En esta sección aprendimos a usar las fuentes de datos secundarias SQL en BIZUIT para interactuar con bases de datos externas directamente desde los formularios. Exploramos cómo estas permiten mostrar información en tiempo real sin necesidad de modelarla en los procesos mediante parámetros o variables.

Resumen de lo Visto:

1. Obtener Información de un Cliente: Configuramos una fuente SQL para consultar datos según un identificador ingresado por el usuario y mostramos los resultados automáticamente en pantalla.
2. Obtener una Lista de Resultados: Creamos una fuente SQL para cargar y visualizar registros en una tabla interactiva con paginación.
3. Optimización: Habilitamos el uso de caché para mejorar el rendimiento en casos de datos que cambian poco.

Con estas configuraciones, ahora podemos integrar datos externos de forma dinámica y eficiente en BIZUIT, optimizando tanto la experiencia del usuario como el rendimiento de nuestros formularios.

Fuente de Datos Secundaria BIZUIT

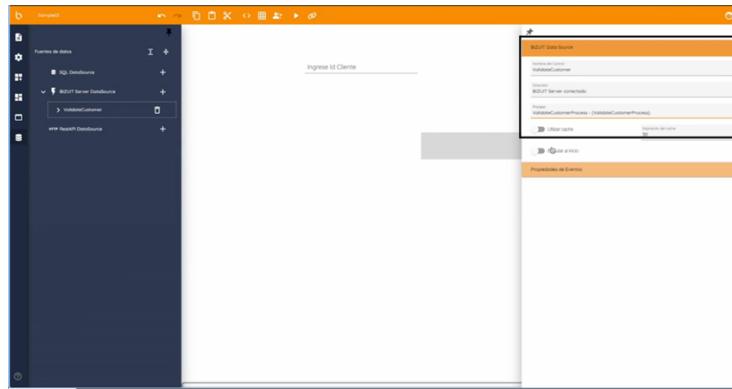
En BIZUIT, las fuentes de datos secundarias de tipo BIZUIT nos permiten interactuar directamente con procesos nativos de la plataforma desde los formularios. Con ellas podemos validar información, consultar datos o ejecutar operaciones sobre instancias sin depender de integraciones externas.

A diferencia de otros tipos de fuentes, estas están optimizadas para trabajar de forma directa con el servidor BIZUIT local o remoto, ofreciendo una experiencia completamente integrada.

Caso práctico: Validación de Cliente

1. Configuración inicial

Creamos un formulario donde el usuario ingrese un ID de Cliente y añadimos una fuente de datos secundaria BIZUIT llamada ValidateCustomer.



2. Selección de servidor y proceso

La fuente puede conectarse:

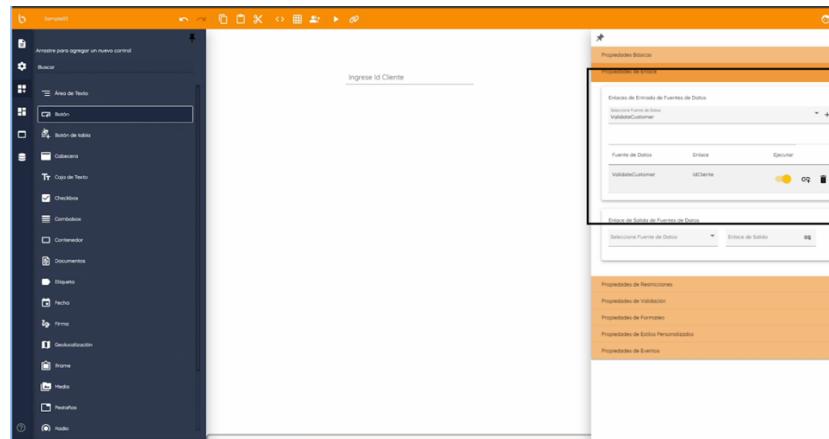
- Al servidor local de BIZUIT.
- A un servidor remoto (requiere URL y credenciales de API).

En este ejemplo usamos el servidor actual y seleccionamos el proceso ValidateCustomerProcess, el cual proceso recibe el parámetro IdCliente y devuelve el parámetro de salida llamado "Habilitado" junto con el Instanceld de ejecución.

3. Enlace del control

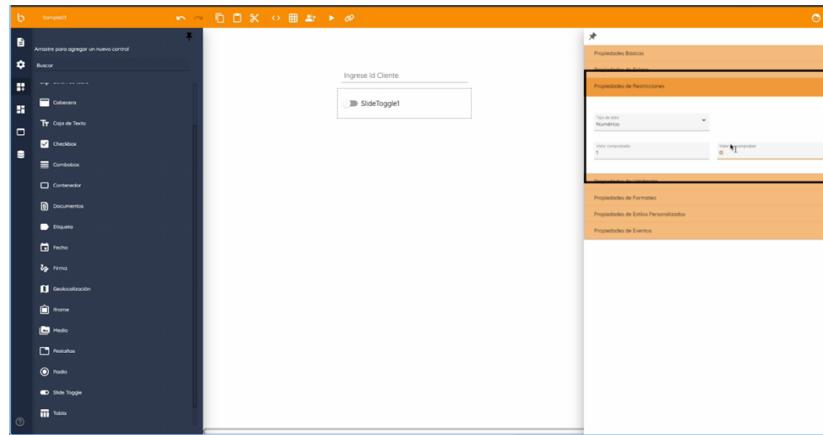
En el formulario:

- Asociamos el campo de entrada al parámetro IdCliente.
- Configuramos que la fuente se ejecute al cambiar el valor.



4. Visualización del resultado

Enlazamos el campo Habilitado a un control SlideToggle y configuramos que interprete 1 como habilitado y 0 como deshabilitado.



5. Ejecución en contexto seguro

Para que el proceso pueda ejecutarse, el formulario debe estar dentro de un contexto de seguridad:

1. En BIZUIT Dashboard, abrimos una página con el módulo Formularios Independientes.
2. Configuramos ese módulo para usar nuestro formulario.

Ahora, al ingresar distintos IDs de cliente, el formulario ejecuta el proceso, valida automáticamente y muestra el resultado en el SlideToggle.

Optimización con caché

Si la validación no cambia con frecuencia, activamos la opción de caché. Esto evita ejecuciones repetitivas, mejora el rendimiento y reduce la carga del servidor.

Resumen

Con las fuentes de datos BIZUIT podemos:

- Ejecutar procesos nativos desde un formulario.
- Obtener resultados en tiempo real.
- Optimizar el rendimiento con caché.

Esta funcionalidad nos permite integrar la lógica de negocio directamente en los formularios, logrando procesos más ágiles y una mejor experiencia de usuario.

Fuente de Datos Secundaria RestAPI

En BIZUIT, las fuentes de datos secundarias de tipo RestAPI permiten conectar formularios con servicios web externos, ya sean públicos o privados.

Con ellas podemos enviar solicitudes HTTP (GET, POST, PUT, DELETE) para obtener, actualizar o eliminar datos en tiempo real y enriquecer la experiencia del usuario.

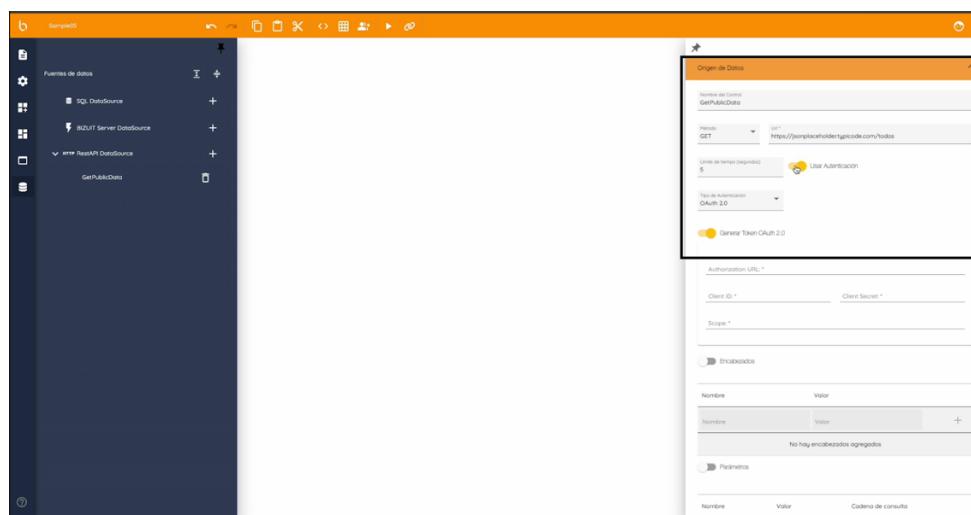
Caso práctico: Consultar una API pública y mostrar datos

1. Configuración inicial

Creamos un formulario que consultará información desde una API pública y añadimos una fuente de datos secundaria RestAPI llamada GetPublicData.

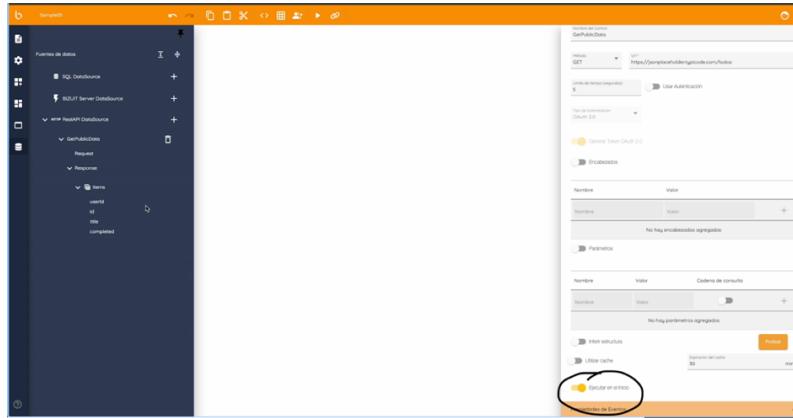
2. Configuración de la fuente

- Seleccionamos el método HTTP **GET** e indicamos la URL.
- Si la API requiere autenticación:
 - **Basic Auth:** ingresar usuario y contraseña.
 - **OAuth 2.0:** usar un token existente o generar uno en el momento.



Si es necesario, añadimos encabezados HTTP.

Luego, configuramos que la fuente se ejecute automáticamente al iniciar el formulario (no requiere parámetros de entrada).



3. Prueba y análisis de respuesta

- Ejecutamos una prueba para confirmar que la API responde correctamente.
- Revisamos la estructura de la respuesta para identificar los campos que mostraremos en el formulario.

4. Visualización en tabla

- Arrastramos los campos deseados a una tabla en el formulario.
- Personalizamos las columnas y formato de visualización.

5. Ejecución del formulario

Al iniciar el formulario, la tabla se llena automáticamente con los datos obtenidos de la API.

Optimización con caché

Si la información no cambia con frecuencia, activamos caché para evitar consultas innecesarias, mejorar la velocidad y reducir la carga en la API externa.

Resumen

Con las fuentes RestAPI podemos:



-
- Integrar datos externos en tiempo real.
 - Mostrar la información en tablas dinámicas.
 - Optimizar rendimiento con caché.

Esta funcionalidad amplía las posibilidades de integración en BIZUIT, permitiéndonos construir formularios conectados con cualquier servicio web.

Conclusión

En esta unidad profundizamos en el uso de las fuentes de datos secundarias en BIZUIT, entendiendo cómo conectarlas a bases de datos externas, procesos nativos de la plataforma y servicios web, para enriquecer formularios con información dinámica en tiempo real.

Aprendimos a trabajar con fuentes SQL, configurando consultas parametrizadas que permiten obtener, validar y mostrar datos externos directamente en el formulario. También exploramos las fuentes BIZUIT, que nos brindan la posibilidad de ejecutar procesos internos y devolver resultados de forma segura y optimizada. Finalmente, analizamos las fuentes RestAPI, que amplían las posibilidades de integración conectando nuestros formularios con cualquier servicio web mediante solicitudes HTTP.

En todos los casos, revisamos cómo optimizar el rendimiento utilizando caché para reducir la carga en los sistemas, así como las mejores prácticas para enlazar datos, manejar parámetros y mostrar resultados en diferentes controles.

Con estas capacidades, ahora contamos con las herramientas necesarias para crear formularios inteligentes, integrados y altamente eficientes, capaces de interactuar con múltiples orígenes de datos y responder de forma ágil a las necesidades del negocio.



Unidad 4: SubFormularios

En esta unidad exploraremos en profundidad el uso de **subformularios** en BIZUIT Forms Designer, una funcionalidad clave para modularizar y optimizar el diseño de formularios complejos. Los subformularios permiten dividir un flujo extenso en secciones independientes, reutilizables y más fáciles de mantener, sin perder coherencia con el proceso principal.

Analizaremos cómo crear, vincular y configurar subformularios para que interactúen de manera fluida con el formulario principal, intercambiando datos mediante parámetros y variables. También revisaremos casos de uso frecuentes, como formularios anidados para capturar información detallada, ventanas modales para completar datos complementarios y estructuras modulares que faciliten la escalabilidad de los proyectos.

La funcionalidad de Subformularios en BIZUIT permite gestionar secciones dinámicas dentro de un formulario principal.

Son ideales para manejar datos relacionados, dividir procesos complejos en partes más pequeñas y optimizar la experiencia del usuario.

Con ellos podemos:

- Mostrar o capturar información relacionada sin recargar el formulario principal.
- Controlar parámetros de entrada y salida para mantener la coherencia de los datos.
- Personalizar diseño, botones y eventos para adaptarlos a cada flujo.

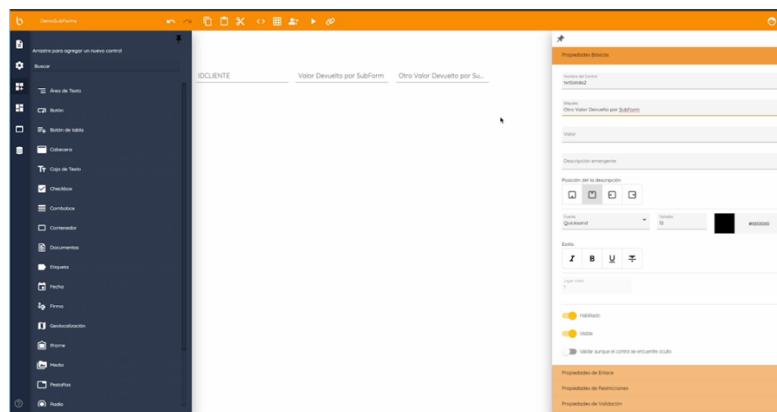
Al finalizar esta unidad, tendrás la capacidad de implementar subformularios que mejoren la experiencia del usuario, reduzcan la complejidad del mantenimiento y potencien la eficiencia en la gestión de datos dentro de tus procesos en BIZUIT.

Ejemplo práctico: Subformulario conectado a un formulario llamador

1. Creación del formulario llamador

En el formulario principal:

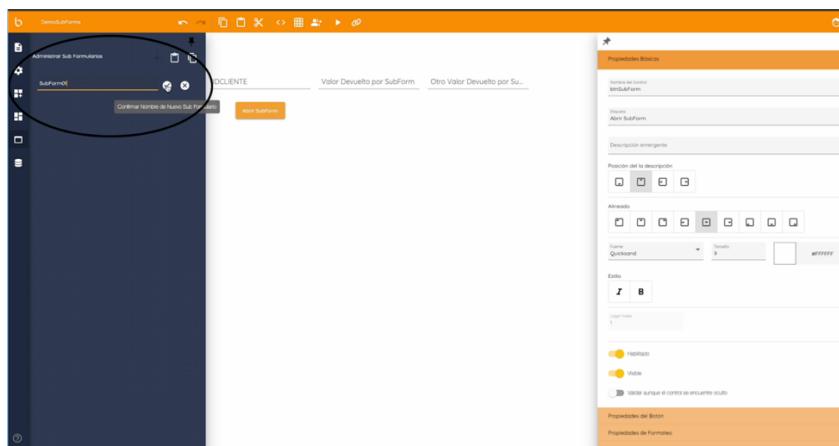
- Añadimos tres cajas de texto:
 - txtIdCliente (etiqueta: IDCLIENTE): valor que se enviará al subformulario.
 - txtSalida1 y txtSalida2: mostrarán valores devueltos por el subformulario.
- Añadimos un botón btnSubForm (etiqueta: Abrir SubForm) para abrir el subformulario.



2. Creación del subformulario

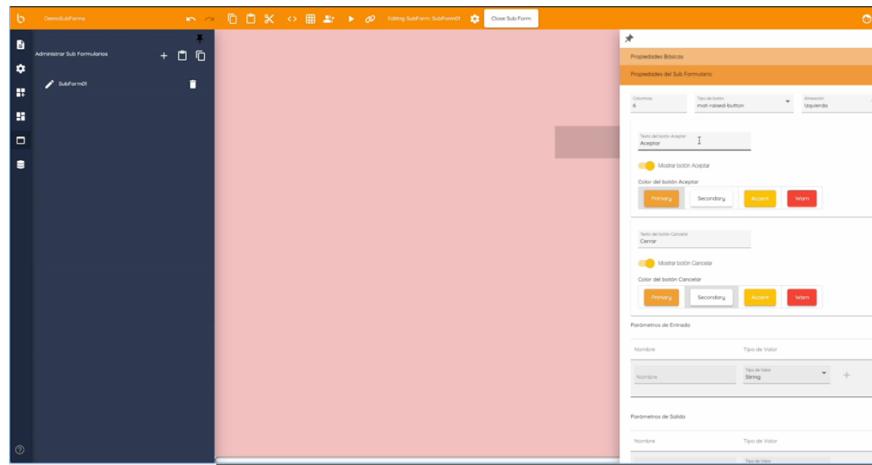
Desde la opción Subformularios:

1. Creamos un nuevo Subformulario llamado SubForm01.



2. Editamos su configuración:

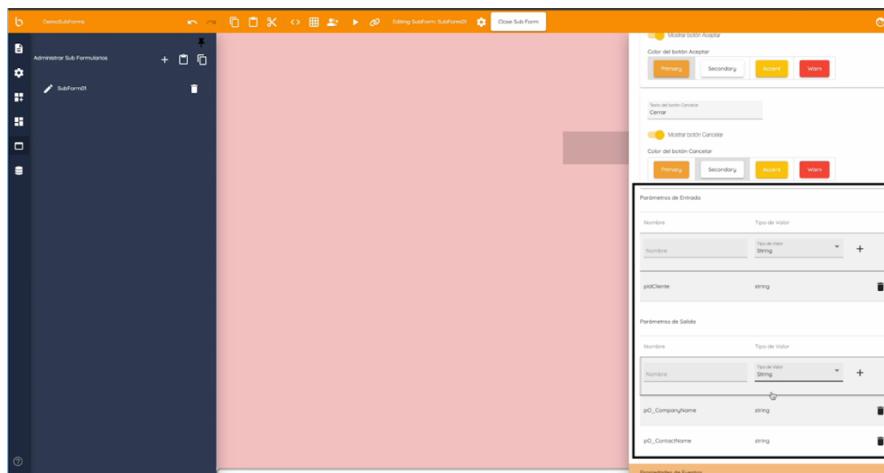
- Cambiamos color de fondo para distinguirlo del formulario principal.
- Activamos o personalizamos botones Aceptar y Cancelar.



3. Configuración de parámetros

En Propiedades del Subformulario:

- **Parámetros de Entrada:**
 - pIdCliente (String): recibe el valor desde el formulario llamador.
- **Parámetros de Salida:**
 - pO_CompanyName (String)
 - pO_ContactName (String)



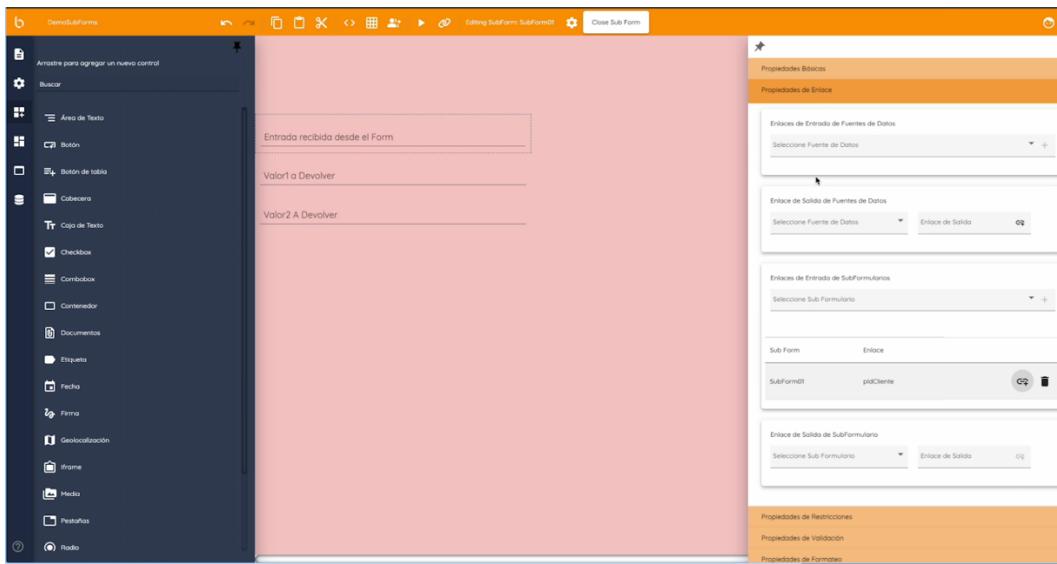
4. Controles dentro del subformulario

- txtEntrada: enlazado a pIdCliente (deshabilitado para no modificarlo manualmente).
- txtOutput1 y txtOutput2: enlazados a pO_CompanyName y pO_ContactName respectivamente.

5. Integración con el formulario principal

En el botón btnSubForm del formulario llamador:

1. Evento: Invocar Subformulario → seleccionar SubForm01.
2. Request Mapper: mapear txtIdCliente → pIdCliente.
3. Response Mapper: mapear pO_CompanyName → txtSalida1 y pO_ContactName → txtSalida2.
4. Deshabilitar txtSalida1 y txtSalida2 para evitar modificaciones manuales.



6. Ejecución

- Ingresar un valor en IDCLIENTE y abrir el subformulario.
- Completar valores de salida y aceptar: los campos del formulario principal se llenan automáticamente.
- Si se cancela, no se actualizan los valores.

Beneficios

- **Centralización:** agrupa lógicas específicas en un componente separado.



- **Interacción dinámica:** actualiza el formulario principal según la actividad en el subformulario.
- **Optimización de diseño:** mejora la usabilidad evitando sobrecargar el formulario principal.
- **Menos errores:** aísla lógicas complejas.
- **Consistencia visual:** estilos coherentes en todos los subformularios.
- **Escalabilidad:** fácil añadir nuevas secciones sin modificar todo el formulario.
- **Integración externa:** compatible con SQL, BIZUIT o RestAPI.
- **Rendimiento:** reduce la carga inicial del formulario principal.
- **Seguridad:** controla el acceso a datos sensibles mediante parámetros.
- **Flujos por etapas:** ideal para procesos que requieren pasos definidos.

Casos de uso

- **Detalles de pedido:** capturar artículos y precios sin sobrecargar el formulario principal.
- **Confirmaciones:** validar acciones críticas antes de ejecutarlas.
- **Validaciones en tiempo real:** verificar datos antes de procesar.
- **Flujos de aprobación:** gestionar aprobaciones o rechazos.
- **Notas y observaciones:** registrar comentarios relacionados.
- **Preferencias de usuario:** personalización sin interrumpir el flujo principal.
- **Maestro-detalle:** manejar relaciones como facturas y líneas de detalle.
- **Datos temporales:** capturar información relevante sin guardarla en la base principal.
- **Reportes secundarios:** mostrar reportes o gráficos relacionados.
- **Simulaciones:** mostrar resultados potenciales antes de confirmar acciones.

Conclusión

A lo largo de esta unidad, profundizamos en el uso de los subformularios dentro de BIZUIT Forms Designer como una solución clave para estructurar y optimizar formularios complejos. Aprendimos a crearlos, configurarlos y vincularlos con el formulario principal, asegurando un intercambio de datos fluido mediante parámetros, variables y eventos.



Vimos cómo los subformularios pueden mejorar la modularidad del diseño, permitiendo reutilizar componentes en distintos contextos, y cómo adaptarlos a diferentes necesidades mediante validaciones, reglas de visibilidad y actualización dinámica de información.

También exploramos casos de uso concretos, como la incorporación de formularios anidados o modales, que facilitan la captura de información complementaria sin sobrecargar la interfaz principal.

Con estas herramientas, ahora contamos con la capacidad de diseñar formularios más ordenados, escalables y fáciles de mantener, potenciando tanto la experiencia del usuario como la eficiencia operativa de nuestros procesos.



Resumen del Capítulo

En este capítulo recorrimos, paso a paso, las principales herramientas y técnicas para diseñar formularios potentes, personalizados y alineados con los procesos en BIZUIT Forms Designer, abordando desde la exploración del entorno de trabajo hasta la implementación de estructuras modulares y reutilizables.

En la Unidad 1, nos familiarizamos con el entorno del diseñador, comprendiendo la función de cada una de sus áreas: el menú lateral con sus opciones de creación, edición y gestión; el área de diseño para estructurar visualmente los formularios; la barra de herramientas con funciones de edición, prueba y colaboración; y la hoja de propiedades para configurar tanto parámetros generales como detalles específicos de cada control.

La Unidad 2 nos llevó a un recorrido completo por los controles disponibles en BIZUIT. Aprendimos a configurarlos, personalizarlos y combinarlos para capturar, mostrar y gestionar datos de forma eficiente. Revisamos propiedades básicas y avanzadas, validaciones, enlaces de datos, formateo dinámico, estilos personalizados y eventos, abordando desde elementos simples como cajas de texto hasta componentes complejos como tablas, pestañas y controles multimedia.

En la Unidad 3, nos enfocamos en las fuentes de datos secundarias, que permiten integrar información externa y lógica interna en tiempo real dentro de los formularios. Trabajamos con tres tipos: SQL: para interactuar con bases de datos externas mediante consultas parametrizadas, BIZUIT: para ejecutar procesos nativos y obtener resultados integrados y RestAPI: para conectarnos con servicios web y mostrar datos dinámicos.

En todos los casos, incorporamos técnicas de optimización con caché y enlazado de parámetros para mejorar rendimiento y experiencia de usuario.

Finalmente, en la Unidad 4 exploramos los subformularios como herramienta clave para modularizar diseños, dividiendo formularios extensos en secciones independientes y reutilizables. Aprendimos a crearlos, vincularlos y configurar su comunicación bidireccional con el formulario principal, aplicando validaciones, reglas



de visibilidad y actualización automática de datos, lo que permite construir interfaces más ordenadas, escalables y fáciles de mantener.

Con lo visto en este capítulo, disponemos de un conjunto sólido de recursos para crear formularios profesionales, integrados y adaptados a cualquier escenario de negocio, maximizando la flexibilidad, la escalabilidad y la experiencia de usuario en BIZUIT.