



MIT App Inventor

BETA

Aprende App

Buscar en este sitio

Inventor

- [¿Qué es App Inventor?](#)
- [Instala App Inventor](#)
- [Configura tu teléfono](#)
- [Correr en servidor local](#)
- [Tu primera aplicación](#)
- [Tutoriales](#)
- [Documentación](#)
- [Mapa del sitio](#)

[Documentación](#) >

Componentes Básicos

Tabla de contenidos

- [Botón](#)
- [Lienzo](#)
- [CheckBox](#)
- [Reloj](#)
- [Imagen](#)
- [Etiqueta](#)
- [ListPicker](#)
- [PasswordTextBox](#)
- [Pantalla](#)
- [TextBox](#)
- [TinyDB](#)

Button (Botón)

Los botones son los componentes que los usuarios tocan para realizar alguna acción en su aplicación.

Los botones detectan cuando el usuario lo pulsa. Muchos aspectos de la apariencia de un botón se puede cambiar. Con la propiedad **Enabled** podemos elegir si el botón está activo en la aplicación..

Propiedades

BackgroundColor

Color de fondo del botón.

Enabled

Si está activado, el usuario puede pulsar el botón para provocar la acción.

FontBold

Si está activado, el texto del botón se muestra en negrita.

FontItalic

Si está activado, el texto del botón se muestra en cursiva.

FontSize

Tamaño en puntos del texto del botón.

FontTypeface

Fuente (tipo de letra) del texto del botón.

Height

Altura del botón.

Width

Anchura del botón.

Image

Imagen para mostrar en el botón. En realidad la imagen actuará como botón.

Text

Texto para mostrar en el botón.

TextAlignment

Alineamiento del texto: izquierda, centro o derecha.

TextColor

Color para el texto del botón.

Eventos

Click()

Usuario pulsa y suelta el botón.

GotFocus()

El botón es el elemento enfocado. Lo hemos seleccionado con el dedo (pero mientras no lo soltemos, no es un click).

LostFocus()

El botón dejó de ser el componente focalizado.

Canvas (Lienzo)

Es un panel bidimensional sensible al tacto, en el cual podemos dibujar y mover objetos animados, como bolas o sprites.

El **BackgroundColor** (color del fondo), **PaintColor** (color del objeto que pintamos), **BackgroundImage** (imagen de fondo), **Width** (ancho) y **Height** (alto) del lienzo se puede establecer en el Diseñador o en el Editor de bloques. La **anchura y la altura** se miden en píxeles y deben ser positivas.

Cualquier posición en el lienzo se puede especificar como un par de coordenadas (X, Y), donde

- X es el número de píxeles de distancia desde el borde izquierdo del lienzo
- Y es el número de píxeles de distancia desde el borde superior de la lona

Hay eventos que son llamados cuándo el lienzo ha sido tocado o una animación (**Sprite** de imagen o **Bola**) ha sido arrastrada. También hay métodos para pintar puntos, líneas y círculos.

Propiedades

BackgroundColor

El color del fondo del lienzo.

BackgroundImage

El nombre de un archivo que contiene la imagen de fondo de la tela

Width

Anchura del lienzo

Height

Altura del lienzo

LineWidth

La anchura de las líneas dibujadas en el lienzo.

PaintColor

El color en el que se trazan líneas

Visible

Si el componente es visible

Eventos

Dragged(number startX, number startY, number prevX, number prevY, number currentX, number currentY, boolean draggedSprite)

Cuando el usuario arrastra el dedo de un punto de la pantalla (PrevX, prevY) a otro (X, Y). El par (startX, startY) indica el primer punto donde el usuario tocó la pantalla, y "draggedSprite" indica si un Sprite está siendo arrastrado.

Touched(number x, number y, boolean touchedSprite)

Cuando el usuario toca el lienzo, proporciona la posición (X, Y) del punto tocado respecto a la esquina superior izquierda del lienzo. El valor "touchedSprite" es verdadero si un sprite también estaba en esta posición.

Métodos

Clear()

Borra el lienzo, sin quitar la imagen de fondo, si se ha proporcionado.

DrawCircle(number x, number y, number r)

Dibuja un círculo (en blanco) en las coordenadas indicadas en el lienzo, con un radio dado.

DrawLine(number x1, number y1, number x2, number y2)

Dibuja una línea entre las coordenadas dadas en el lienzo.

DrawPoint(number x, number y)

Dibuja un punto en las coordenadas indicadas en el lienzo.

Save()

Guarda una imagen del lienzo (con su contenido) al dispositivo de almacenamiento externo y devuelve el nombre de ruta completo (en el sistema de carpetas o árbol de directorios) del archivo guardado. Si se produce un error, es llamado el evento de la pantalla **ErrorOccurred**.

SaveAs(nombre del fichero)

Guarda una imagen del lienzo (con su contenido) al dispositivo de almacenamiento externo en el archivo denominado especificado como nombre del fichero. Debe terminar con una de las extensiones ".jpg", ".jpeg", o ".png". (que determina el tipo de archivo: JPEG o PNG). Devuelve el nombre de ruta completo del archivo guardado.

CheckBox (casilla de verificación)

El componente **CheckBox** puede detectar que el usuario lo ha marcado la casilla y puede cambiar su estado en respuesta booleana.

Un componente **CheckBox** provoca un evento cuando el usuario lo toca. Muchas de propiedades que afectan a su aspecto se pueden establecer en el Diseñador o el Editor de bloques.

Propiedades

BackgroundColor

Color para el fondo de la casilla de verificación.

Checked

Verdadero si la casilla está marcada y falso en caso contrario.

Enabled

Si está activado, el usuario puede pulsar la casilla de verificación para cusar una acción.

Height

Altura de la casilla.

Width

Ancho de la casilla.

Text

Texto que se mostrará en la casilla de verificación.

TextColor

Color del texto de la casilla de verificación.

Visible

Si se establece, la casilla de verificación es visible.

Eventos

Click()

El usuario tocó y soltó la casilla de verificación.

GotFocus()

La casilla de verificación es el elemento enfocado.

LostFocus()

La casilla de verificación dejó de ser el elemento enfocado.

Clock(Reloj)

Se puede utilizar un componente del reloj para crear un contador de tiempo que señala los acontecimientos a intervalos regulares. El componente de reloj también hace varias conversiones y manipulaciones con unidades de tiempo.

El formato interno usado por el reloj se llama *instante*. Un instante es un paquete de datos de tiempo diversos, que contiene, entre otros datos, los milisegundos que han pasado desde el inicio de 1970 en el calendario Gregoriano, la zona horaria, el día del año, el día de la semana, el día del mes, el periodo AM o PM, la hora del día, el minuto de la hora, el segundo del minuto, o el milisegundo del segundo.

Uno de los usos de la componente de reloj es el de *temporizador*: permite configurar un intervalo de tiempo al final de cada uno de los cuales se disparará el evento de temporizador.

Un segundo uso del componente de reloj es el de manipular el tiempo, y expresar el tiempo en varias unidades. . El método **Now** devuelve el momento actual como un instante. El reloj proporciona diversos métodos para manipular instantes, por ejemplo, devolver el instante correspondiente a varios segundos, meses o años desde un instante dado. También proporciona métodos para mostrar el día, hora, minuto y segundo, ..., de un instante dado.

Propiedades

TimerInterval

Intervalo de temporizador, en milisegundos

TimerEnabled

Si es cierto, se activa el temporizador.

TimerAlwaysFires

Si es cierto, se activa el temporizador, incluso cuando la aplicación no aparece en pantalla. En caso contrario, cuando la aplicación no está en pantalla no se producen los disparos del temporizador.

Eventos

Timer()

Se produce este evento cuando el temporizador se dispara

Métodos

SystemTime()

Tiempo interno del teléfono, en milisegundos. Proporciona los milisegundos desde la base de tiempo, que es el 1 de enero de 1970.

Now()

El instante correspondiente al momento actual, leído desde el reloj del teléfono

MakeInstant(fecha)

Convierte a un instante el DD / MM / AAAA HH: MM: SS o el DD / MM / AAAA o hh: mm especificado.

MakeInstantFromMillis(ms)

Crea el instante correspondiente al tiempo indicado en milisegundos

GetMillis(instante)

Obtener los milisegundos del instante indicado. Proporciona los milisegundos desde la base de tiempo, que es el inicio de 1970.

AddSeconds(instante, Número de segundos)

Se obtiene el instante resultante de añadir el número de segundos especificado al instante dado.

AddMinutes(instante, Número de minutos)

Se obtiene el instante resultante de añadir el número de minutos especificado al instante dado.

AddHours(instante, Número de horas)

Se obtiene el instante resultante de añadir el número de horas especificado al instante dado.

AddDays(instante, Número de días)

Se obtiene el instante resultante de añadir el número de días especificado al instante dado.

AddWeeks(instante, Número de semanas)

Se obtiene el instante resultante de añadir el número de semanas especificado al instante dado.

AddMonths(instante, Número de meses)

Se obtiene el instante resultante de añadir el número de meses especificado al instante dado.

AddYears(instante, Número de años)

Se obtiene el instante resultante de añadir el número de años especificado al instante dado.

Duration(instante de inicio, instante posterior)

Milisegundos entre los instantes

Second(instante)

El segundo del instante dado. Si es el instante de ahora (now) nos indica el segundo actual.

Minute(instante)

El minuto del instante dado. Si es el instante de ahora (now) nos indica el minuto actual.

Hour(instante)

La hora del instante dado. Si es el instante de ahora (now) nos indica la hora actual.

DayOfMonth(instante)

El día del instante dado, de 1 a 31. Si es el instante de ahora (now) nos indica el día de hoy.

Weekday(instante)

El día de la semana, como un número del 1 (domingo) al 7 (sábado), del instante dado. Si es el instante de ahora (now) nos indica el día de hoy.

WeekdayName(instante)

El nombre del día de la semana del instante dado. Si es el instante de ahora (now) nos indica el día de hoy.

Month(instante)

El mes del instante dado, del 1 al 12. Si es el instante de ahora (now) nos indica el mes actual.

MonthName(instante)

El nombre del mes del instante dado. Si es el instante de ahora (now) nos indica el mes actual.

Year(*instante*)

El año del instante dado. Si es el instante de ahora (now) nos indica el año actual.

FormatDateTime(*instante*)

Devuelve la fecha y la hora del instante dado en el formato del teléfono. En España como dd/mm/aaaa hh:mm:ss

FormatDate(*instante*)

Devuelve la fecha del instante dado en el formato de fecha del teléfono. En España como dd/mm/aaaa.

FormatTime(*instante*)

Devuelve la la hora del instante dado en el formato hh:mm:ss.

Image (Imagen)

Se usa el componente de imagen para representar imágenes que los usuarios seleccionan y manipulan.

Un componente de la imagen muestra una imagen. Se puede especificar la imagen para mostrar y otros aspectos de la apariencia de la imagen en el diseñador o en el Editor de bloques.

Propiedades

Picture

Imagen que muestra el componente de imagen.

Visible

Si es cierto, la imagen se muestra.

Height

Altura de la imagen.

Width

Ancho de la imagen.

Label (Etiqueta)

Las etiquetas son componentes que se utilizan para mostrar texto.

Una etiqueta muestra el texto especificado en la propiedad **Text**. Otras propiedades, que se pueden establecer tanto en el diseñador como en el editor de bloques, controlan la apariencia y la colocación de ese texto.

Propiedades

BackgroundColor

Color para el fondo de la etiqueta.

FontBold

Si se establece, el texto de la etiqueta se muestra en negrita.

FontItalic

Si se establece, el texto de la etiqueta se muestra en cursiva.

FontSize

Tamaño en puntos del texto de la etiqueta.

FontTypeface

Fuente del texto de la etiqueta.

Height

Altura de la etiqueta.

Width

Ancho de etiqueta.

Text

Texto para mostrar en la etiqueta.

TextAlignment

Alineamiento a la izquierda, centro o derecha.

TextColor

Color para el texto de la etiqueta.

Visible

Si está activado, la etiqueta es visible.

ListPicker (Lista de selección)

Los usuarios pueden tocar un componente de lista de selección para elegir un elemento de una lista de cadenas de texto.

Cuando un usuario toca el botón selector, se muestra una lista de elementos de texto para que el usuario pueda elegir. Los elementos de texto se puede especificar a través del diseñador o el Editor de bloques mediante el establecimiento de la propiedad `ElementsFromString` mediante la concatenación de los valores separados por comas (por ejemplo, `selección1, selección2, selección3`) o mediante el establecimiento de la propiedad `Elements` de una `List` (`Lista`) en el editor de bloques.

Otras propiedades, incluyendo `TextAlignment` y `BackgroundColor`, afectan a la apariencia del botón y si puede ser desplegado (`Enabled`).

Propiedades

`Selection`

Elemento de la lista seleccionado.

`Items`

Lista separada por comas de elementos a mostrar cuando se despliegue la lista.

`ElementsFromString`

Aparentemente, idéntica a la anterior. Lista separada por comas de elementos para mostrar cuando se despliegue la lista.

`BackgroundColor`

Color para el fondo del botón selector.

`FontBold`

Si se establece, el texto del botón selector se muestra en negrita.

`FontItalic`

Si se establece, el texto del botón selector se muestra en cursiva.

`FontSize`

Tamaño en puntos del texto del botón selector.

`FontTypeface`

Fuentes del texto del botón selector.

`Height`

Altura del botón selector.

`Width`

Ancho del botón selector.

`Text`

El texto que se mostrará en el botón selector de la lista.

`TextAlignment`

Alineamiento a la izquierda, centro o derecha.

`TextColor`

Color del texto que se mostrará en el botón selector de la lista.

`Visible`

Si se establece, el botón selector es visible.

Eventos

`AfterPicking()`

Se dispara cuando el usuario ha seleccionado un elemento de la lista.

`BeforePicking()`

El usuario ha pulsado el botón selector de la lista, pero aún no se ha seleccionado un elemento.

`GotFocus()`

El botón selector de la lista se convirtió en el elemento enfocado.

`LostFocus()`

El botón selector de la lista dejó de ser el elemento enfocado.

Screen (Pantalla)

La pantalla no aparece en la paleta igual que los restantes componentes, pero aparece automáticamente en el proyecto. Cada proyecto tiene una pantalla, llamada `Screen1`. Este nombre no puede ser cambiado.

Propiedades

`BackgroundColor`

Color del fondo de la pantalla.

`BackgroundImage`

La imagen del fondo de la pantalla.

`Height`

Altura de la pantalla,

`icon`

Imagen para ser utilizada como icono de la aplicación instalada en el teléfono. Este debe ser un PNG o una imagen JPG, 48x48 es un buen tamaño. *Advertencia:* Especificar otros formatos diferentes a PNG o JPG, tales como ico, por ejemplo, pueden impedir que App Inventor empaquete la aplicación.

Scrollable

Si se establece mediante la casilla de verificación que aparece en el diseñador, habrá una barra de desplazamiento vertical en la pantalla, y la altura de la aplicación puede exceder la altura física del dispositivo. Cuando no se controla, la altura aplicación está limitada a la altura del dispositivo.

Title

Título de la pantalla (texto). Este aparecerá en la parte superior izquierda del teléfono cuando se ejecuta la aplicación. Una elección natural para el título es el propio título de la aplicación, pero se puede usar otra cosa, o incluso cambiar el título mientras la aplicación se está ejecutando.

Width

Ancho de la pantalla.

Eventos

Initialize()

Se dispara cuando se inicia la aplicación. Se puede utilizar para establecer los valores iniciales y realizar otras operaciones adicionales.

ErrorOccurred(component, functionName, number errorNumber, message)

Señala cuando se produce un error. El evento ErrorOccurred se utiliza actualmente para un pequeño conjunto de errores, incluyendo:

- Errores que se producen el componente NXT de LEGO Mindstorms
- Errores que se producen en los componentes Bluetooth
- Errores que se producen en el componente de Twitter
- Errores que se producen en el componente de SoundRecorder
- ActivityStarter - cuando se llama a startActivity, pero no hay ninguna actividad que se corresponde a las propiedades definidas.
- LocationSensor - cuando LatitudeFromAddress o LongitudeFromAddress fallan.
- Player - al establecer la fuente en propiedades, falla.
- Sonido - al establecer la fuente en propiedades, falla, o bien, falla durante la reproducción.
- MediaPlayer - al establecer la fuente en propiedades, falla.

Por esos errores, el sistema le mostrará una notificación por defecto, con un número de error y un mensaje. Se puede utilizar este controlador de eventos para prescribir un comportamiento de error diferente al predeterminado, poniendo a prueba ErrorNumber y tomar la acción apropiada.

PasswordTextBox (cuadro de texto para contraseña)

Los usuarios introducen las contraseñas en un componente de cuadro de texto para contraseña, que oculta el texto que se ha escrito en él.

Un cuadro de texto para contraseña es el mismo que el componente `TextBox` ordinario, excepto en que no muestra los caracteres tecleados por el usuario.

Se puede obtener o establecer el valor del texto en la caja con la propiedad `Text`. Si `Text` está en blanco, puede utilizar la propiedad `Hint` para proporcionar al usuario una sugerencia de lo que puede escribir. La `Hint` aparece como texto débil en el cuadro.

El componente de cuadro de texto para contraseña se utiliza generalmente con un componente de botón. El usuario pulsa el botón después de la introducción de texto.

Propiedades

BackgroundColor

Color para el fondo del cuadro de texto.

Enabled

Si está activado, el usuario puede introducir una contraseña en el cuadro.

FontBold

Si se establece, se muestra el texto en negrita.

FontItalic

Si se establece, se muestra el texto en cursiva.

FontSize

Tamaño en puntos para el texto.

FontTypeface

Fuente para el texto.

Height

Altura de la caja de texto.

Width

Ancho de la caja.

TextAlignment

Alineamiento a la izquierda, centro o derecha.

TextColor

Color para el texto.

Hint

Sugerencia sobre la contraseña.

Eventos

GotFocus()

La caja se convirtió en el componente seleccionado.

LostFocus()

La caja ya no tiene el foco.

TextBox

Los usuarios pueden introducir texto en un componente de cuadro de texto.

El valor del texto inicial o el que el usuario haya introducido en un componente de cuadro de texto está en la propiedad **Text**. Si **Text** está en blanco, puede utilizar la propiedad **Hint** para proporcionar al usuario una sugerencia de lo que escribir. La **sugerencia** aparece como texto débil en el cuadro.

La propiedad **MultiLine** determina si el texto puede tener más de una línea. Para una caja de una única línea de texto, el teclado se cerrará automáticamente cuando el usuario presiona la tecla Listo. Para cerrar el teclado para cuadros de texto de varias líneas, la aplicación debe utilizar el método **HideKeyboard** o confiar en que el usuario pulse la tecla Atrás.

La propiedad **NumbersOnly** restringe al teclado numérico, de modo que sólo puedan introducirse números.

Otras propiedades afectan a la apariencia del cuadro de texto (**TextAlignment**, **BackgroundColor**, etc) y si se puede utilizar (**Enabled**).

Los cuadros de texto se suelen utilizar con el componente **Button**, para que el usuario haga clic en el botón cuando se haya completado la entrada del texto.

Si el texto introducido por el usuario no se debe mostrar, usar **PasswordTextBox** en su lugar.

Propiedades

BackgroundColor

El color de fondo de la caja de entrada. Usted puede elegir un color por su nombre en el Diseñador o en el Editor de bloques. El color de fondo por defecto es 'default' (aspecto 3-D sombreado).

Enabled

Cuando se selecciona, el usuario puede introducir texto en el cuadro de entrada. De forma predeterminada, está seleccionado.

FontBold (exclusivamente en el diseñador)

Si se selecciona, la fuente para el texto aparece en negrita. Por defecto, no está seleccionado.

FontItalic (exclusivamente en el diseñador)

Si se selecciona el texto aparece en cursiva. Por defecto, no lo hace.

FontSize (exclusivamente en el diseñador)

El tamaño de la fuente para el texto. De forma predeterminada, es de 14,0 puntos.

FontTypeface (exclusivamente en el diseñador)

La fuente para el texto.

Height

Altura de la caja de texto.

Hint

Si la propiedad **Text** está vacía, mostrará una sugerencia sobre lo que el usuario debe ingresar.

MultiLine

Si es verdad, entonces este cuadro de texto acepta varias líneas de entrada, que se introducen con la tecla de retorno. Para una caja de una única línea de texto, el teclado

se cerrará automáticamente cuando el usuario presiona la tecla Listo. Para cerrar el teclado para cuadros de texto de varias líneas, la aplicación debe utilizar el método HideKeyboard o confiar en que el usuario pulse la tecla Atrás.

NumbersOnly

Si es verdad, entonces se despliega exclusivamente el teclado numérico, por lo que el usuario sólo podrá entrar números, que pueden incluir un punto decimal y un signo menos opcional. Esto se aplica a la entrada del teclado solamente. Incluso si NumbersOnly es cierto, se puede utilizar [SetTextAt] para introducir cualquier texto.

Text

El texto en el cuadro de entrada, que se puede establecer por el programador en el diseñador o el Editor de bloques, o puede ser introducido por el usuario.

TextAlignment (exclusivamente en el diseñador)

El texto puede ser justificado a la izquierda, centrado o justificado a la derecha. De forma predeterminada, el texto se justifica a la izquierda.

TextColor

El color para el texto. Usted puede elegir un color por su nombre en el Diseñador o en el Editor de bloques. El color de texto predeterminado es negro.

Visible

Si es cierto, entonces el componente es visible

Width

Eventos

GotFocus()

Se dispara este evento cuando el cuadro obtiene el foco, al tocarlo.

LostFocus()

Este evento se dispara cuando el cuadro pierde el foco, por ejemplo, si el usuario toca un cuadro de texto diferente.

Métodos

HideKeyboard()

Oculto el teclado. Sólo los cuadros de texto multilinea lo necesitan.

TinyDB

Se puede utilizar un componente TinyDB para almacenar los datos que estarán disponibles cada vez que la aplicación se ejecuta.

TinyDB es un componente no visible.

Las aplicaciones creadas con App Inventor se inicializan cada vez que se ejecutan. Si una aplicación establece el valor de una variable y el usuario cierra la aplicación, el valor de esa variable no será recordado la próxima vez que se ejecuta la aplicación. TinyDB es un almacén de datos *persistente* para la aplicación, es decir, los datos almacenados estarán disponibles cada vez que se ejecute la aplicación. Un ejemplo podría ser un juego que guarda la puntuación más alta, y la recuperaa cada vez que se juega el juego.

Los elementos de datos se almacenan en *las etiquetas*. Para guardar un elemento de datos, se debe especificar la etiqueta bajo la que debe ser almacenado bajo. Posteriormente, se puede recuperar el elemento de datos que se almacena en una etiqueta determinada. Si no hay ningún valor almacenado en una etiqueta, entonces el valor devuelto es el texto vacío. En consecuencia, para ver si una variable tiene un valor almacenado en ella, comprobar si el valor de retorno es igual a la de texto vacío (es decir, un cuadro de texto sin texto, en blanco).

Sólo hay un almacén de datos por aplicación. Si usted tiene múltiples componentes TinyDB, se utilizará el mismo almacén de datos. Para obtener el efecto de almacenes separados, utilice claves diferentes. Además, cada aplicación tiene su propio almacén de datos. No se puede utilizar TinyDB para pasar los datos entre dos aplicaciones diferentes en el teléfono.

Propiedades

ninguno

Eventos

ninguno

Métodos

StoreValue(text tag, valueToStore)

Almacenar el valor en la etiqueta dada. La **tag** debe ser una cadena de texto, el valor puede ser una cadena o una lista.

GetValue(text tag)

Obtiene el valor que se almacena bajo la etiqueta dada. Si el valor no se almacena, se devuelve el texto vacío.

Para limpiar la base de datos para una aplicación, vaya en el teléfono bajo Ajustes → Aplicaciones → Administrar aplicaciones, selecciona la aplicación y pulsa el botón "Borrar datos".

Los datos de TinyDB son persistentes sólo cuando se ha empaquetado y descargado la aplicación en el dispositivo. Si está desarrollando la aplicación conectado al teléfono, y se reinicia la aplicación AppInventor, o si se desconecta y vuelve a conectar el teléfono, entonces la base de datos se reinicializa. Este es un caso donde la aplicación no se limita a ser detenida y reiniciada, sino que se elimina del teléfono y luego vuelve a cargar.

Comentarios

No tienes permiso para agregar comentarios.



Contenidos bajo [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0](#).

Buena parte de los contenidos son traducciones y adaptaciones de [Learn](#), publicados bajo licencia de Creative Commons Attribution 3.0 Unported License.