**Unidad 1**

**Lección 1.1. Introducción a la plataforma App Inventor**

|  |
| --- |
| http://www.robbokids.com/images/robbokids/cursos/app-inventor.jpg |
| Imagen 1. Logo de Mit App Inventor |

El Mit App inventor es una plataforma de desarrollo de código abierto, creada por Google Lab y administrada por el MIT (instituto tecnológico de Massachusett), esta plataforma permite elaborar aplicaciones de una forma muy fácil y rápida para el sistema operativo Android. El entorno de desarrollo es totalmente visual y está basado en el paradigma de programación orientado a eventos.

Con App Inventor un usuario que tenga principios básicos de programación podrá desarrollar aplicaciones móviles para el sistema operativo android en cuestión de pocas horas.

Para elaborar una aplicación en App inventor solo se necesita un navegador web y un dispositivo móvil con tecnología Android (puede ser un teléfono Smartphone o una Tablet) aunque si no tenemos un celular o tableta podemos probar las aplicaciones por medio de un Emulador.

Cuando se crea un proyecto en App Inventor se usa tres aplicaciones, la primera se conoce como el gestor de proyectos y permite crear, eliminar y modificar los proyectos que construyas en App Inventor. La segunda se conoce como el App Inventor Designer, con este podrás construir o diseñar las pantallas de tu programa, y la tercera es el App Inventor Blocks Editor, en el que se definen y programan los componentes de la aplicación móvil.

**Lección 1.2 ¿Qué es la programación orientada a eventos?**

La Programación orientada a eventos es un paradigma de programación en el que la estructura y la ejecución de los programas van determinados por los sucesos o acciones que ocurren en el sistema, definidos por el usuario o por el propio sistema.

La programación orientada a eventos es muy fácil de usar y es adecuada para aquellas personas que tienen poco conocimiento en programación. Con los lenguajes orientados a eventos se pueden realizar en poco tiempo aplicaciones sencillas y muy funcionales, utilizando interfaces gráficas en las que se insertan componentes o controles a los que se le programan eventos. Dichos eventos permiten al usuario realizar una serie de acciones lógicas para un determinado programa.

Dentro de las herramientas de programación orientadas a eventos se encuentra el Mit App Inventor, este maneja una serie de componentes o controles con propiedades que se pueden cambiar para que el componente se muestre en pantalla de forma diferente o actúe de otra manera dentro de la aplicación. Para cambiar las propiedades de un componente, primero debemos seleccionar el componente deseado en la lista de componentes, también debemos tener en cuenta que hay algunos valores de propiedades de algunos componentes que no son modificables y otros que sí lo son.

Otro aspecto a tener en cuenta en la programación orientada a eventos es el comprender y tener bien claro el concepto de los siguientes términos:

* Evento
* Propiedades
* Métodos

**Lección 1.2.1 ¿Qué son los eventos?**

Los ***Eventos*** son las acciones sobre el programa, como por ejemplo:

* Clic sobre un botón
* Doble clic sobre el nombre de un fichero para abrirlo
* Arrastrar un icono
* Pulsar una tecla o una combinación de teclas
* Elegir una opción de un menú
* Escribir en una caja de texto
* O simplemente mover el ratón

Cuando se produce o dispara un evento sobre un determinado componente, se da inicio a un conjunto de acciones programadas por el usuario para ese evento concreto.

|  |
| --- |
|  |
| Imagen 2. Algunos eventos del componente botón |

**Lección 1.2.2 ¿Qué es una propiedad?**

|  |  |
| --- | --- |
| Una ***propiedad*** es una asignación que describe algo sobre un componente,como por ejemplo:   * Un formulario * Un botón de comando * Una caja de texto * Una etiqueta * Una imagen * etc… | Imagen 3. Ventana de propiedades para el componente imagen |

Dependiendo de la propiedad, esta se puede asignar en tiempo de diseño usando la ventana Propiedades y/o en tiempo de ejecución en el momento de programar el control.

Las propiedades en el momento de programar el control se muestran con color verde

|  |
| --- |
|  |
| Imagen 4. Propiedades en el momento de programar el control para el componente botón |

Las propiedades se usan para cambiar la forma de los componentes o controles de la aplicación, por ejemplo el tamaño de la letra de un control, el tipo de letra, la alineación, etc.

**Lección 1.2.3 ¿Qué es un método?**

Un ***método*** es una función que es llamada desde el programa, a diferencia de los procedimientos, estos no son programados por el usuario, ya que vienen pre-programados en el lenguaje de programación, los métodos realizan tareas típicas y comunes para todas las aplicaciones; por eso están incluidos dentro del lenguaje de programación, esto con el fin de liberar al usuario de la tarea de programarlos. Cada tipo de componente tiene sus propios métodos; debemos tener en cuenta que los métodos solo pueden ser ejecutados en tiempo de ejecución no en tiempo de diseño. Algunos ejemplos de métodos son:

* En una ventana, el método MOVE, que mueve un formulario en un espacio de dos dimensiones en la pantalla
* Otros SetFocus, LostFocus, AddItem etc…

Los métodos en App inventor los encontramos en los bloques de programación de los diferentes componentes y estos aparecen con color violeta.

|  |
| --- |
|  |
| Imagen 5. Algunos métodos del componente Lienzo |

En resumen las aplicaciones creadas con App Inventor no llevan a cabo un conjunto de instrucciones en un orden predeterminado, sino que reaccionan a  eventos. Al hacer clic en un botón, arrastrar el dedo o tocar en la pantalla es lo que conocemos como eventos. Cuando se produce un evento, la aplicación reacciona llamando a una secuencia de instrucciones como establecer el color de fondo de un botón a azul o cambiar el texto de una etiqueta.

Los eventos pueden ser divididos en 2 tipos diferentes:

* automáticos como al abrir una ventana
* Iniciados por el usuario como hacer clic en un botón, tocar o arrastrar en la pantalla, inclinar el teléfono.

**Leccion1.3. El entorno de trabajo en App inventor**

El entorno de trabajo de app inventor es un servicio que se encuentra en la nube y para acceder a este solo necesitas un navegador web (puedes usar Internet Explorer, Firefox o Chrome) y una cuenta de Google, si no tienes una cuenta de correo en google debes suscribirte al servicio de cuenta escribiendo en el navegador web la dirección <http://www.gmail.com> , entrar a la opción de crear cuenta, debes llenar todos los datos que te piden y con eso ya tendrás tu cuenta google para poder acceder al app inventor.

Para entrar al Mit App Inventor solo debes escribir la dirección web <http://www.ai2.appinventor.mit.edu>

El Entorno de trabajo de App Inventor está conformado por tres herramientas:

* El Gestor de Proyectos
* El Diseñador (App Inventor Disigner)
* El Editor de Bloques (App Inventor Blocks Editor)

La siguiente presentación te mostrara como es el entorno de trabajo dentro del Mit App Inventor.

|  |
| --- |
|  |
| Presentación 1. Entorno de Trabajo en App Inventor |

**Leccion1.4. Nuestra primera aplicación en App Inventor**

Vamos a ver cómo crear una aplicación con un botón que al pulsarlo salga un mensaje en pantalla.

1. Creamos un nuevo proyecto
2. En el visor de proyecto insertamos los siguientes componentes.

|  |
| --- |
|  |
| Imagen 6. Diseño de la ventana de mi primera App |

* Agregamos un botón.
* Escribimos en la propiedad texto del botón el texto “Aceptar”
* Agregamos una etiqueta
* Limpiamos la propiedad texto de la etiqueta

3. Vamos al editor de bloques y simplemente agregamos los bloques tal cual están en la imagen.

|  |
| --- |
|  |
| Imagen 6. Bloques de programación para el mi primera app |

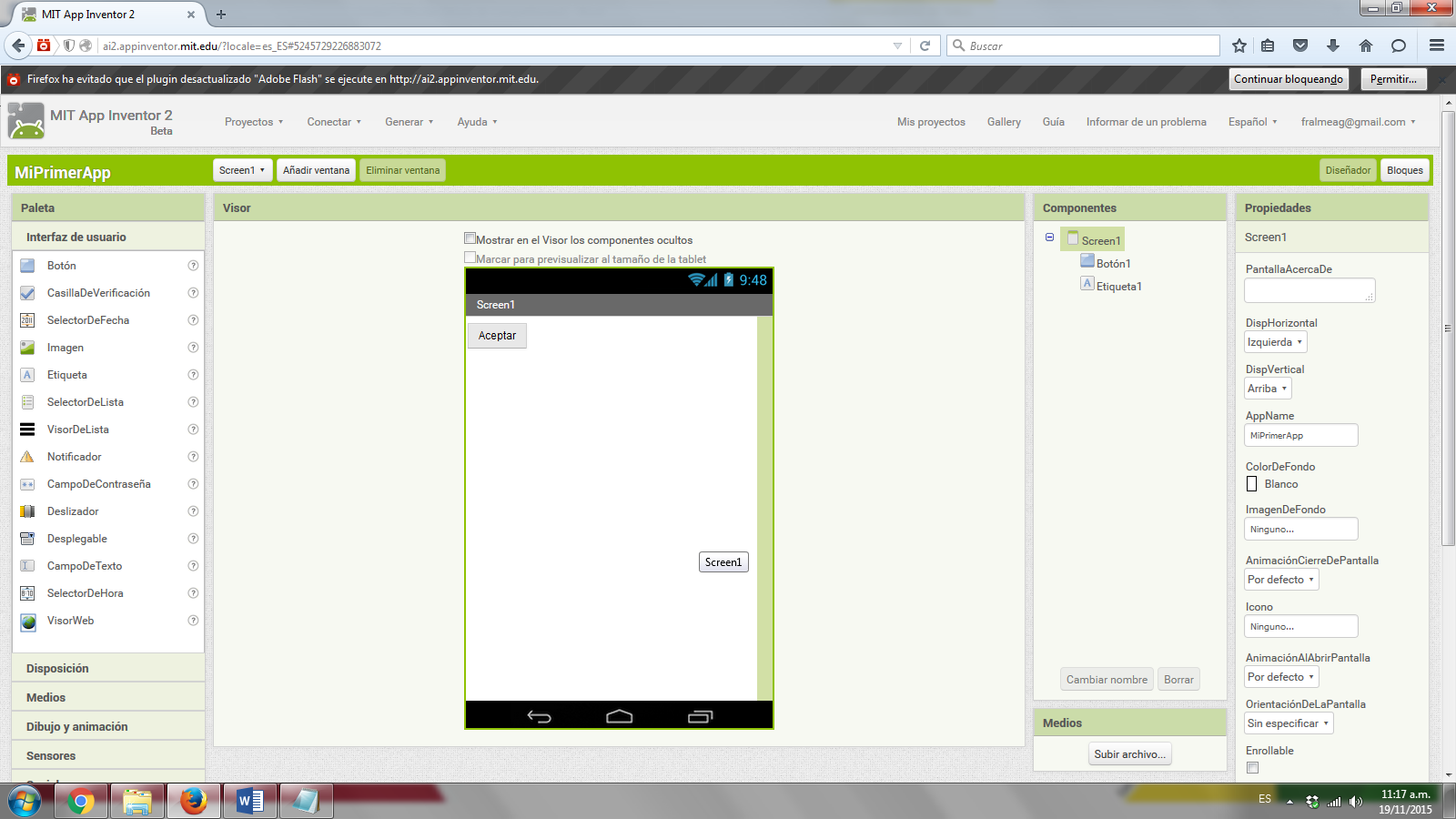
Cuando ejecutes la aplicación en el celular el resultado será el siguiente.

|  |
| --- |
| C:\Users\Francisco Alenjadro\Dropbox\Cargas de cámara\Screenshot_2015-12-09-00-17-20.png |
| Imagen 8. Imagen de cómo queda la app en el dispositivo movil |

A continuación te mostramos un video que te mostrará cómo crear esta app.

|  |
| --- |
|  |
| Video 1. Como elaborar nuestro primer programa en app inventor |

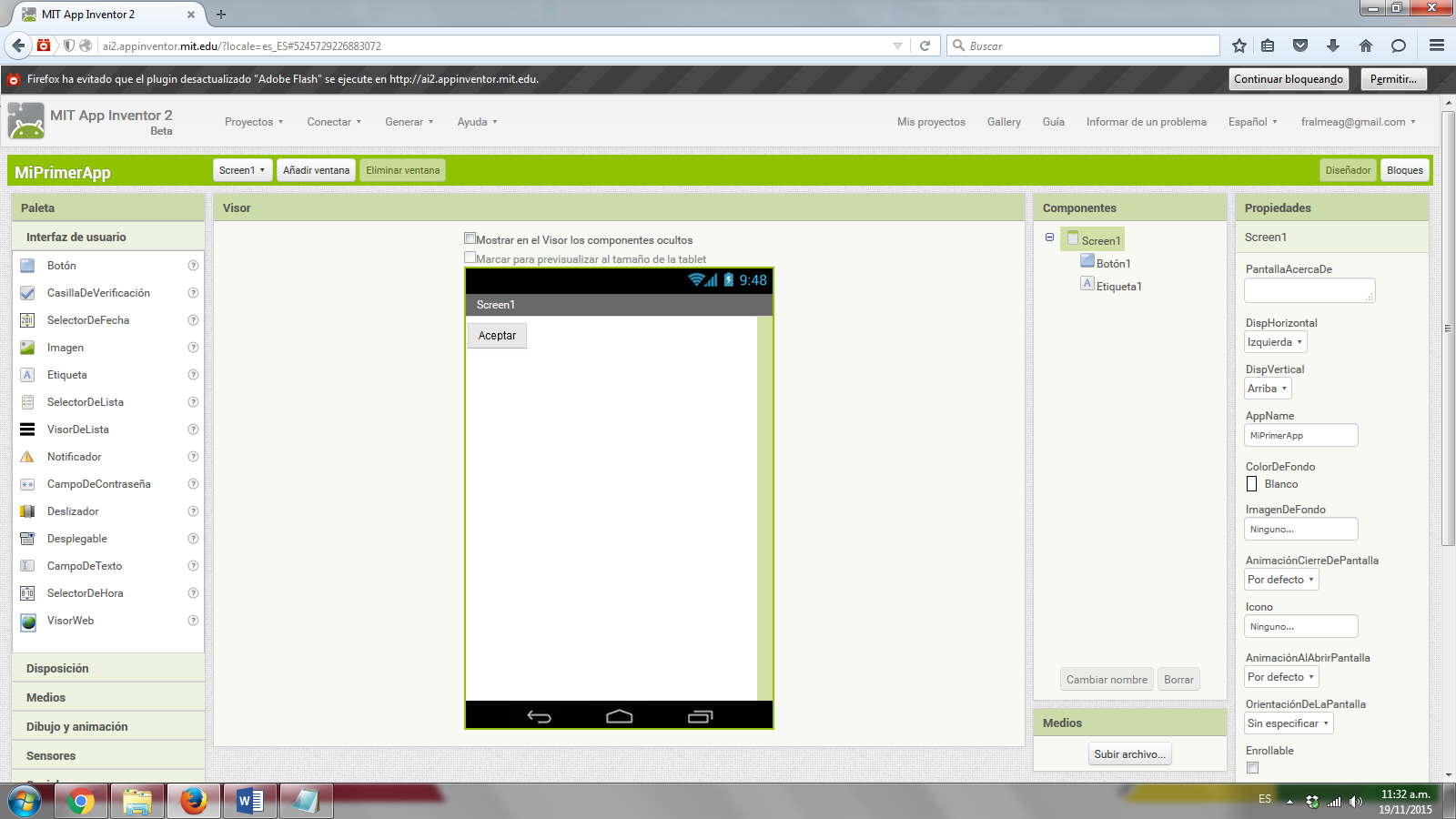
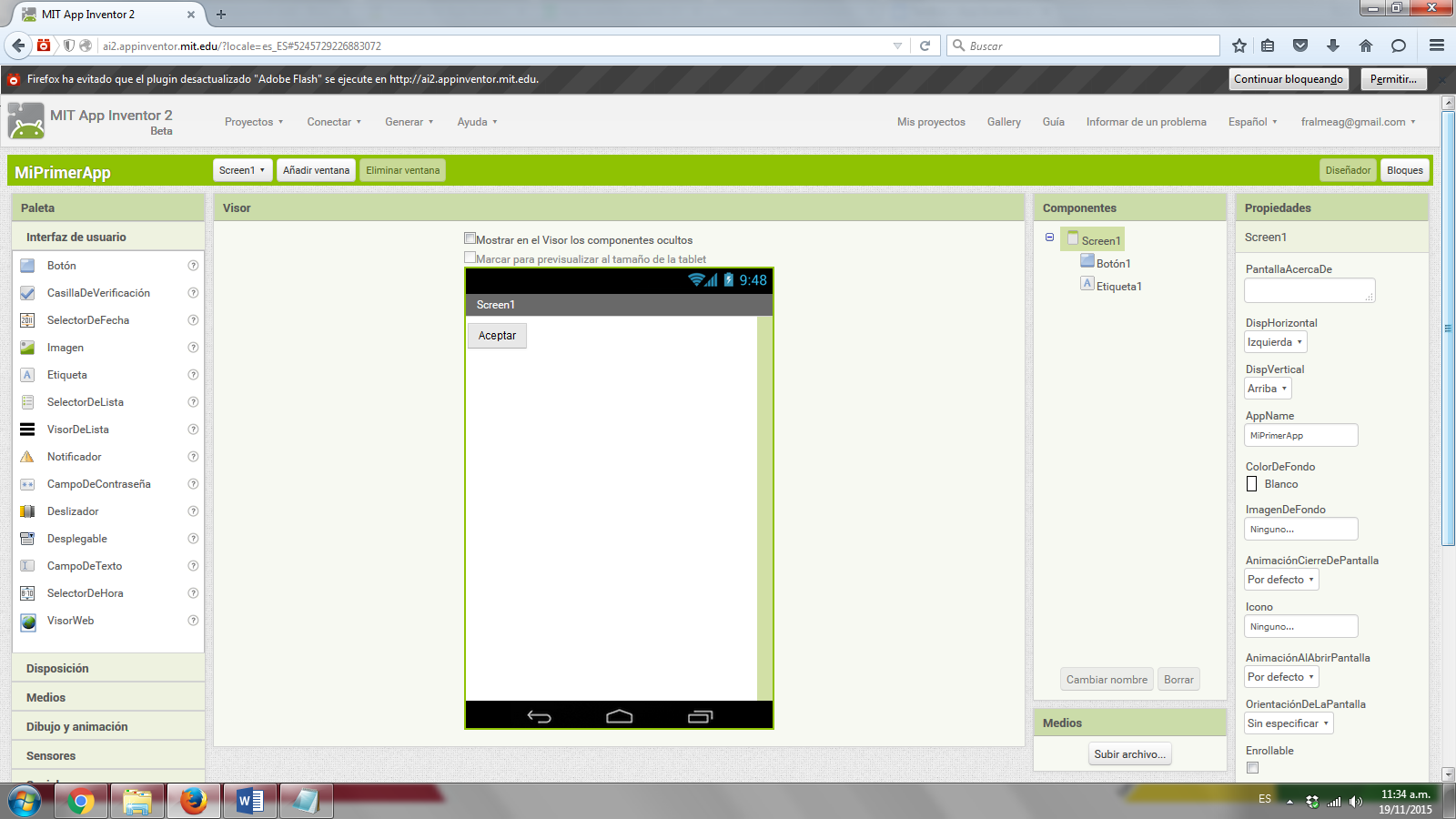
**UNIDAD 2 CONTROLES BASICOS EN APP INVENTOR**

**Lección 2.1.** Control Label (etiqueta).

El control label es un componentes que se utilizan para mostrar texto como el contenido de variables, campos o mensajes en pantalla, este control posee propiedades para cambiar la apariencia del texto en la tabla encontraras las propiedades que usa el control label o etiqueta.

|  |
| --- |
|  |
| Imagen 9. Control Label (Etiqueta) |

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Permite cambiar el color de fondo de la etiqueta. |  |
| FontBold | Muestra el texto de la etiqueta en negrita. |
| FontItalic | Muestra el texto de la etiqueta en cursiva. |
| FontSize | Permite cambiar el tamaño de la letra en el texto de la etiqueta. |
| FontTypeface | Permite cambiar el tipo la letra en el texto de la etiqueta. |
| Height | Alto de etiqueta (y tamaño). |
| Width | Ancho de la etiqueta (x-size). |
| Text | Esta propiedad muestra el Texto que se muestra en la etiqueta. |
| TextAlignment | Permite alinear a la Izquierda, centro o derecha. |
| TextColor | Determina el Color de texto de la etiqueta. |
| Visible | Esta propiedad muestra la etiqueta en pantalla si esta activa. |

**Lección 2.2.** Controles TextBox (Campo de Texto)  y PasswordTextbox (campo de contraseña) 

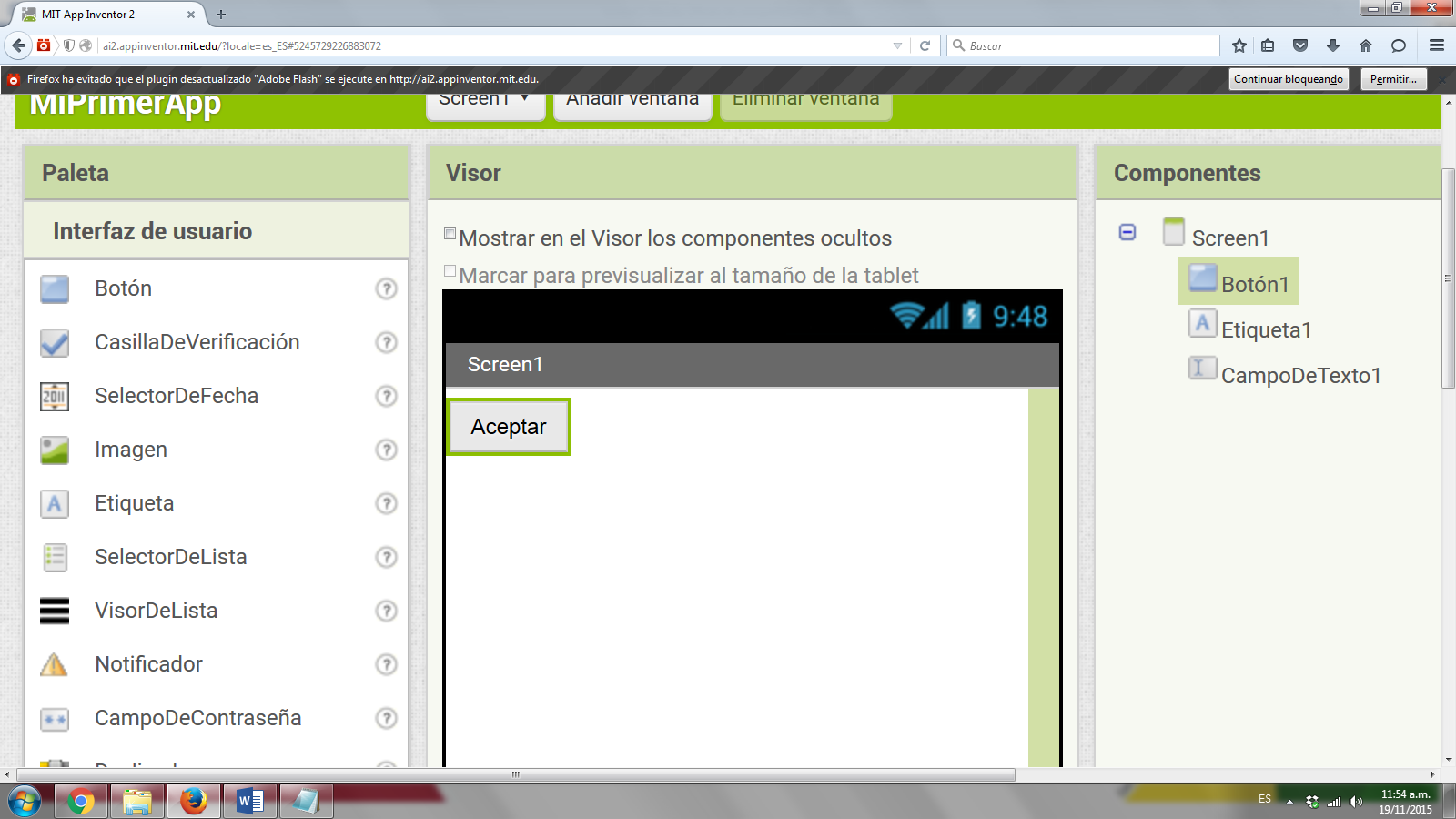
El control Textbox es un campo que se usa para introducir texto, este permite al usuario escribir un texto para su posterior uso en el programa, el texto que se introduzca en el control quedara almacenado en la propiedad Texto. El componente Textbox maneja varias propiedades como por ejemplo, la propiedad “pista que le da una idea al usuario de la aplicación de lo que se debe escribir en el campo. La propiedad *“Multilínea”* permite que el texto este compuesto por varias líneas. La propiedad “*SóloNúmeros*” obliga al usuario a que únicamente pueda introducir datos números.

|  |
| --- |
|  |
| Imagen 9. Controles de Texto y Contraseña |

En la siguiente tabla podemos observar las propiedades del control Textbox.

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Color de fondo de cuadro de texto. |  |
| Enabled | Si se establece, el usuario puede introducir texto en el cuadro. |
| FontBold | Si se activa, se muestra el texto en negrita. |
| FontItalic | Si se activa, se muestra el texto en cursiva. |
| FontSize | Tamaño de la letra para el texto. |
| FontTypeface | Tipo de letra para el texto. |
| Text | Texto que se muestra en la caja antes de que el usuario escriba, puede usarse como sugerencia. |
| Height | Altura de la caja (y de tamaño). |
| Width | Anchura de la caja (x-size). |
| TextAlignment | Izquierda, centro o derecha. |
| TextColor | Color para el texto. |
| Hint | Sugerencia de texto para escribir. |

El componente PasswordTextbox (campo de contraseña) se usa cuando no se quiere que sea visible el texto que está escribiendo, como su nombre lo indica es utilizado especialmente para escribir contraseñas. El componente PasswordTextbox es básicamente un campo de texto en el que no se muestran los caracteres a medida que el usuario los va escribiendo

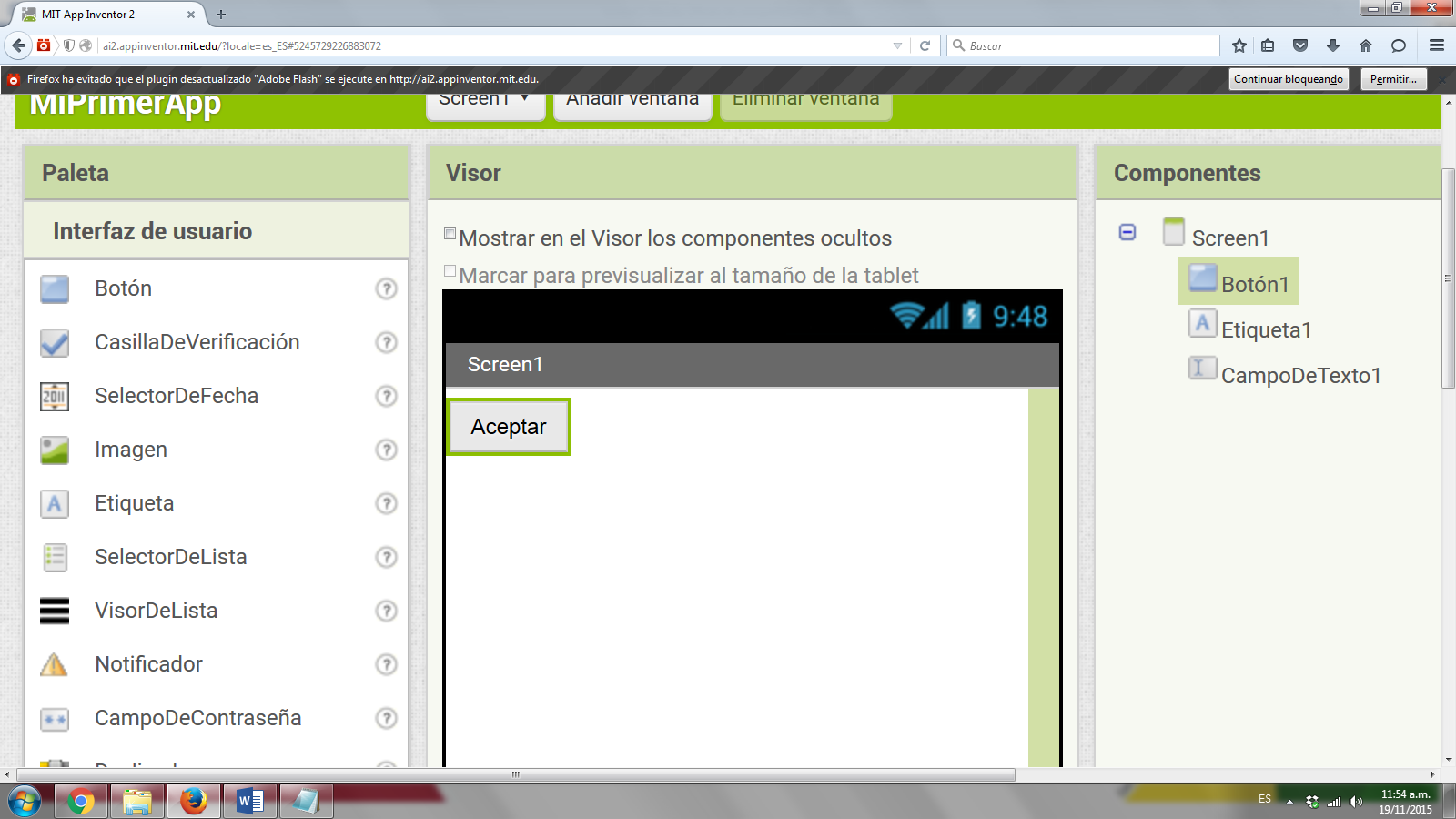
**Lección 2.3.** Control Image (Imagen)

El Control Image es un componente que se usa para mostrar fotos o imágenes en formato gif, jpg, png.

|  |
| --- |
|  |
| Imagen 10. Control Imagen |

Por medio del Diseñador o el Editor de Bloques podemos definir la imagen que queremos mostrar, la propiedades que se usan con el control con las siguientes:

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| Height | Alto de la imagen |  |
| Width | Ancho de la imagen |
| Picture | Permite Insertar una Foto o Imagen |
| Scalepicturetofit | Escala para la imagen a tamaño máximo |
| Visible | Visible |

**Lección 2.4.** Control Buttom (Botón) 

El Control Buttom es un componente que el usuario utiliza para pulsar y disparar un evento que realizara alguna acción en la aplicación.  Es uno de los componentes mas comunes dentro de los programa que usan en interfaces graficas de usuario. Los botones detectan cuando el usuario hace clic sobre estos.

|  |
| --- |
|  |
| Imagen 11. Control Botón de Comando |

Las propiedades que se usan en botones son:

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Establece el color de fondo del botón. |  |
| Enabled | Si se activa, el usuario puede pulsar el botón para provocar la acción. |
| FontBold | Si se activa, el texto del botón se muestra en negrita. |
| FontItalic | Si se activa, el texto del botón aparece en cursiva. |
| FontSize | Determina el tamaño de la letra del el texto del botón. |
| FontTypeface | Permite cambiar el tipo de letra para el texto del botón. |
| Height | Altura del botón (tamaño en Y). |
| Width | Ancho del Botón(tamaño en X). |
| Image | Imagen que se muestra en el botón. |
| Text | Texto que se muestra en el botón. |
| TextAlignment | Permite Alinear el texto ya se a la izquierda, centro o derecha. |
| TextColor | Color para el texto del botón. |

Ahora que conoces las formas, las propiedades y eventos de los componentes text, buttom, image, y label realizaremos algunos ejemplos.

**Ejemplo 1**. Elaboraremos un programa que dado la base y la altura de un triángulo rectángulo se calcule su área.

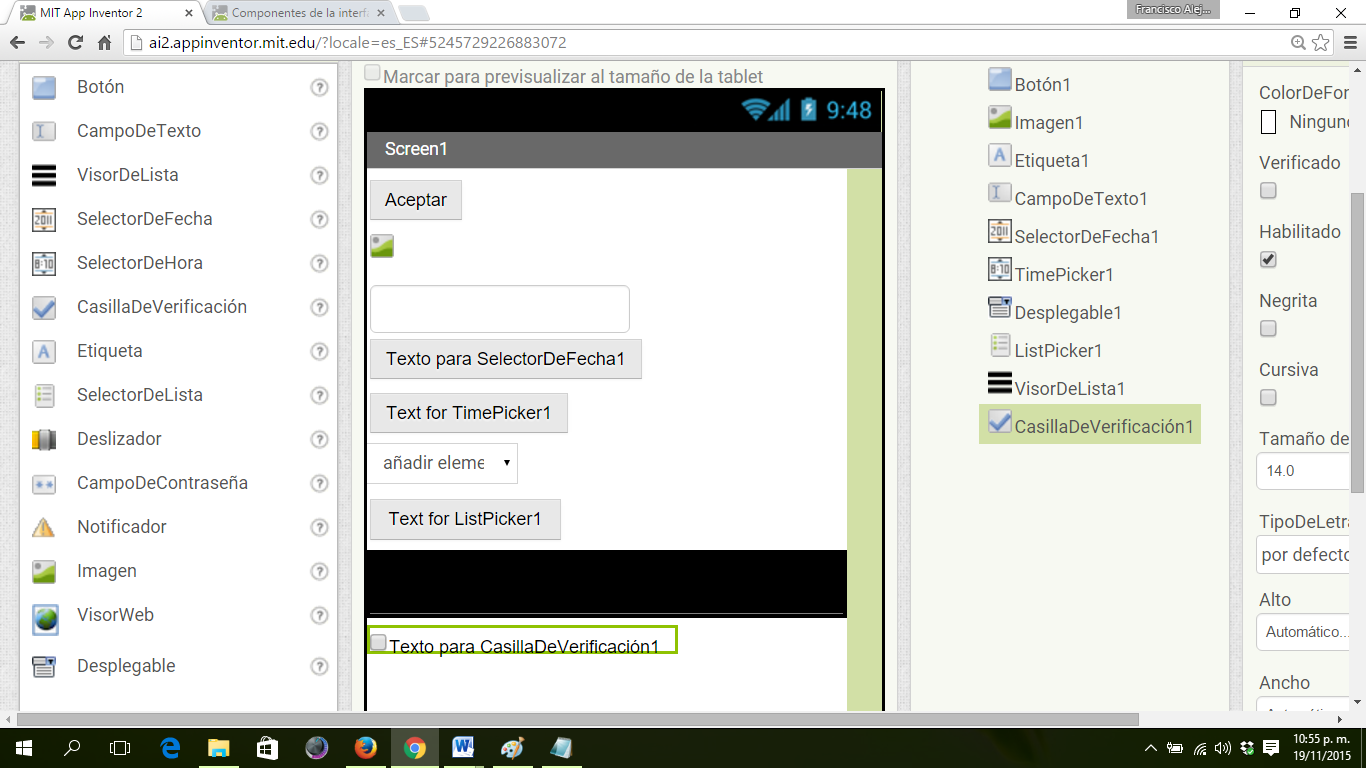
|  |
| --- |
| C:\Users\Francisco Alenjadro\AppData\Local\Microsoft\Windows\INetCache\Content.Word\imgVideoAreaT.PNG |
| Video 2. Ejemplo de app para calcular el área de un triangulo |

**Ejemplo 2.** El siguiente video muestra cómo hacer un botón de pánico en App Inventor

|  |
| --- |
| C:\Users\Francisco Alenjadro\AppData\Local\Microsoft\Windows\INetCache\Content.Word\imgImagen.png |
| Video 3. Ejemplo de botón de pánico |

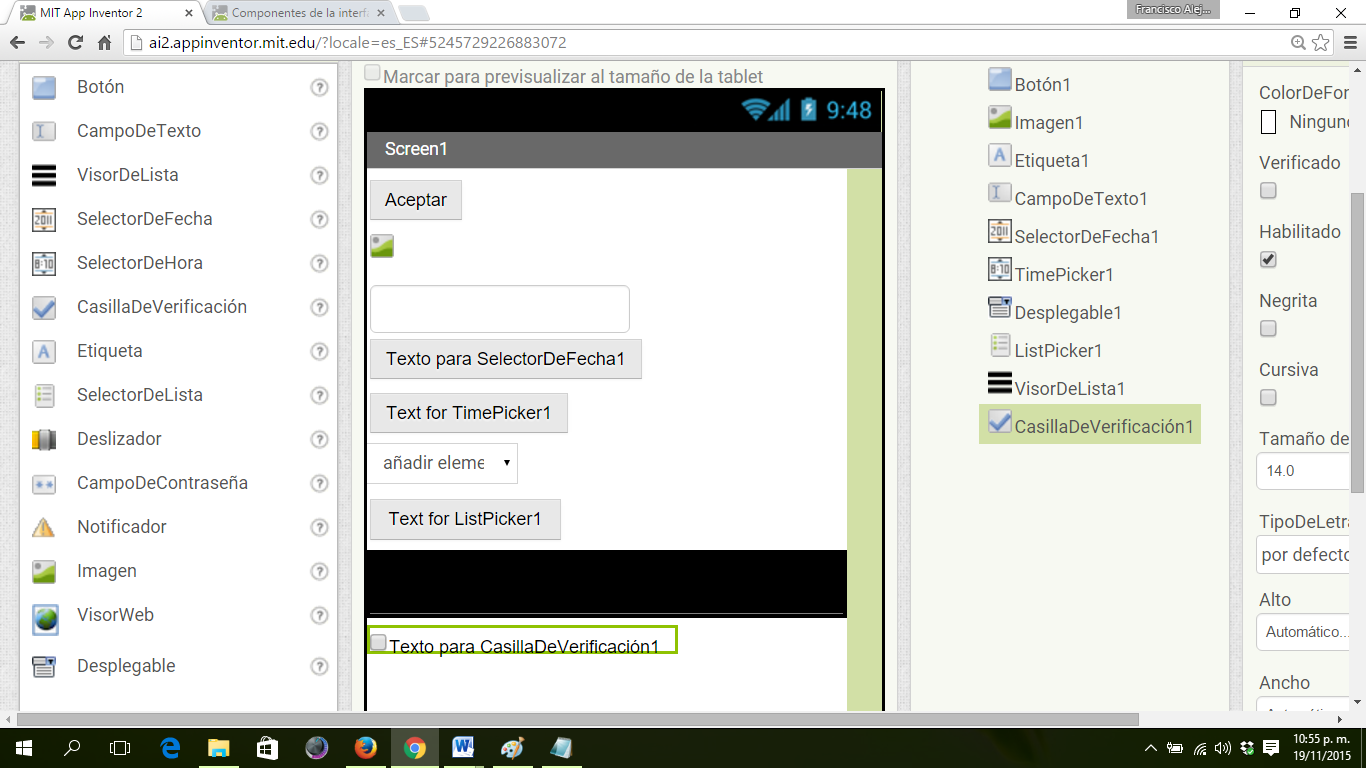
**UNIDAD 3 CONTROLES PARA MANEJO DE DATOS**

**Lección 3.1.** Controles DatePicker y TimePicker (Selectores de fecha y hora)

El control DatePicker es un botón que al pulsarlo, abre una ventana de diálogo que permite al usuario seleccionar una fecha.

|  |
| --- |
|  |
| Imagen 12. Control de Fecha |

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Determina el color de fondo |  |
| Enabled | Indica si el control está habilitado |
| FontBold | Establece la Negrita en la letra del control |
| FontItalic | Establece la Cursiva en la letra del control |
| FontSize | Determina el tamaño de letra |
| FontTypeface | Permite cambiar el tipo de letra |
| Height | Configura el alto de control |
| Width | Configura el ancho del control |
| Imagen | Inserta una imagen de fondo al componente |
| Shape | Permite cambiar la forma del control |
| ShowFeedback | Mostrar Pulsación |
| Text | Texto para el control |
| TextAlignment | Determina la Posición del texto |
| TextColor | Permite cambiar el color de texto |
| Visible | Indica si el control esta visible para el usuario |

El control TimePicker Es un botón que cuando se pulsa muestra una ventana que permite al usuario definir una hora.

|  |
| --- |
|  |
| Imagen 13. Control de Hora |

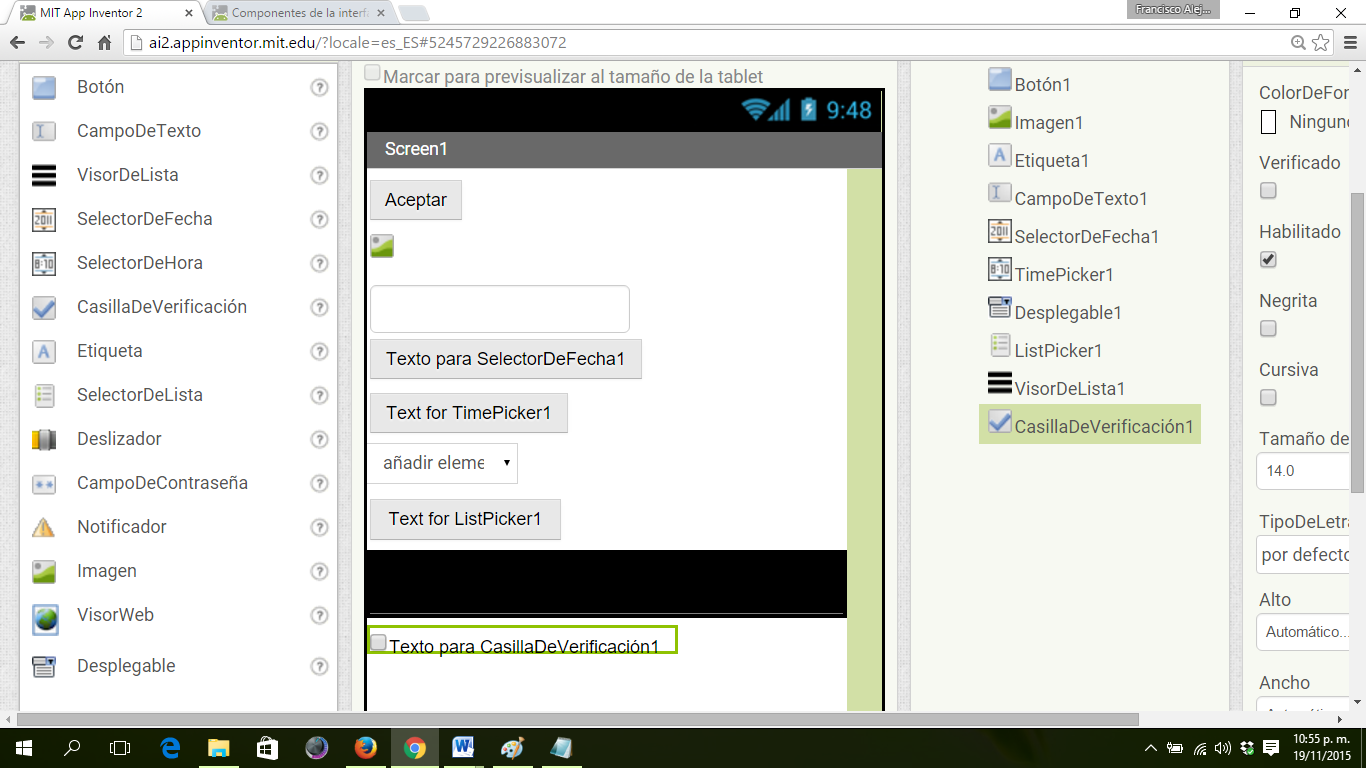
|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Establede el color de fondo de control |  |
| Enabled | Determina si el control esta habilitado para el uso del usuario |
| FontBold | Formato de negrita para el control |
| FontItalic | Formato de cursiva para el control |
| FontSize | Permite cambiar el tamaño de letra |
| FontTypeface | Permite cambiar el tipo de letra |
| Height | Cambia el alto del componente |
| Width | Cambia el ancho del componente |
| Imagen | Inserta una imagen de fondo en el control |
| Shape | Cambia la forma del componente o control |
| ShowFeedback | Mostrar Pulsación |
| Text | Texto para el control |
| TextAlignment | Cambia la posición del texto |
| TextColor | Cambia el color de texto |
| Visible | Indica si el control es visible para el usuario |

El siguiente video nos muestra cómo podemos trabajar con los controles de fecha y hora en app inventor

|  |
| --- |
|  |
| Video 4. Manejo de controles de fecha y hora |

**Ejercicios para practicar**

1. Realiza un código en donde se seleccione una fecha, después se seleccione otra y se calcule la cantidad de días entre esas dos fechas.
2. Realiza un código en donde se seleccione una hora venidera, y cuando llegue esa hora que suele una alarma.

**Lección 3.2.** Control CheckBox 

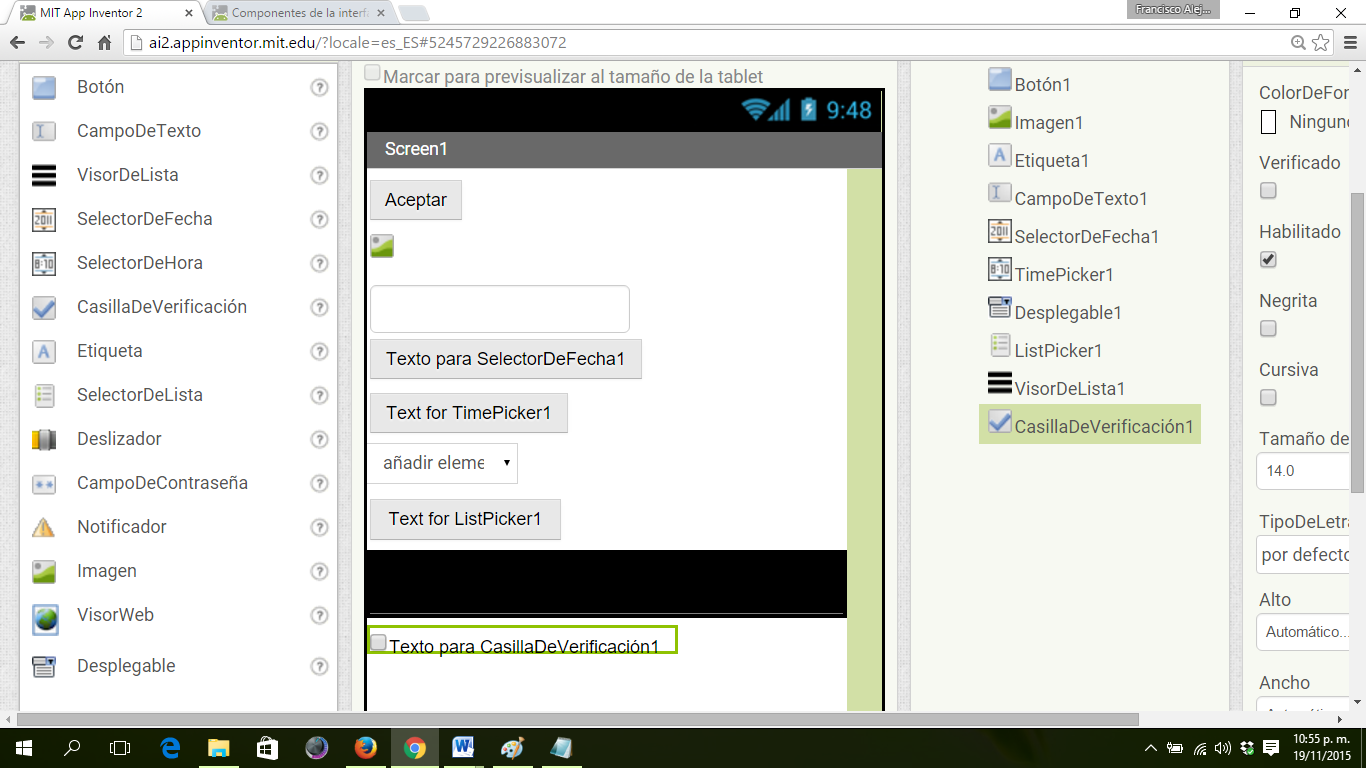
El Control CheckBox es una casilla de verificación permiten chequear listar de opciones, estos poseen un cuadro que permite seleccionar la opción, cuando el cuadro es pulsado equivale a un "cierto" y sin pulsar a un "falso" o. Al igual que los otros componentes las casillas de verificación poseen propiedades que permiten definir su aspecto, y sus propiedades pueden gestionarse desde el Diseñador o el Editor de Bloques.

|  |
| --- |
|  |
| Imagen 14. Control para manejar casillas de verificación |

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Determina el color de fondo |  |
| Enabled | Indica si el control esta habilitado |
| FontBold | Establece el formato Negrita para el control |
| FontItalic | Establece el formato Cursiva para el control |
| FontSize | Permite cambiar el tamaño de letra |
| FontTypeface | Permite cambiar el tipo de letra |
| Height | Determina el alto del componente |
| Width | Determina el ancho del componente |
| Text | Establece un Texto para el control |
| TextColor | Permite cambiar el color de texto |
| Visible | Indica si el control es visible para el usuario |

El siguiente video muestra un ejemplo un ejemplo sobre cómo implementar una aplicación manejando casillas de verificación.

|  |
| --- |
|  |
| Video 5. Manejo de Casillas de Verificación |

**Lección 3.3.** Control ListPicker 

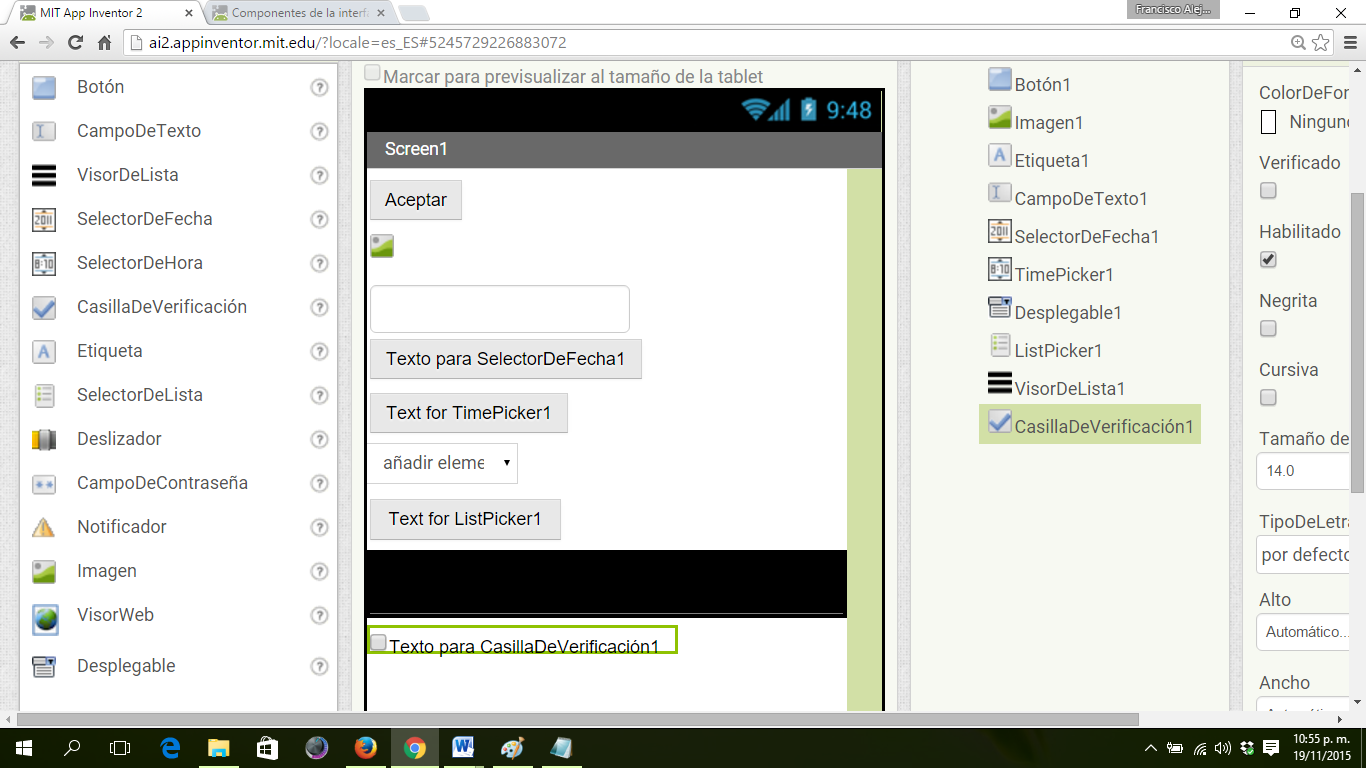
El Control ListPicker es un botón que al pulsarlo muestra una lista de textos entre los cuales el usuario puede elegir una opción. Los textos se pueden definir a través del Diseñador especificando en la propiedad ElementosDesdeCadena la lista de textos delimitados (por ejemplo, *opción 1, opción 2, opción 3*) también se puede definir la lista por medio del Editor de Bloques poniendo en la propiedad “Elementos” el nombre de una Lista e indicando el valor Verdadero en la propiedad “Mostrar Barra De Filtrado”, esta propiedad permite al usuario buscar en la lista; Este control posee otras propiedades que permiten cambiar la apariencia del botón (PosiciónDelTexto, ColorDeFondo, etc.) y que determinan si el botón se puede pulsar o no.

|  |
| --- |
|  |
| Imagen 15. Control manejar listas |

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| BackgroundColor | Determina el color de fondo del botón |  |
| Elements | Permite determinar la lista de opciones de visualización para el componte |
| ElementsFromString | lista de opciones para utilizar separadas por coma |
| Enabled | Indica si el ListPicker puede ser usado |
| FontBold | Formato Negrita para el control. |
| FontItalic | Formato Cursiva para el control. |
| FontSize | Determina el tamaño de la letra del texto del selector de lista. |
| FontTypeface | Familia de fuentes de texto del Selector de lista. |
| Height | Altura de la caja (tamaño en Y). |
| Image | Especifica la ruta de la imagen del botón. |
| SelectionIndex | Indica el elemento seleccionado.  Cuando es cambiado directamente por el programador, la propiedad SelectionIndex también se cambia al primer elemento del componente ListPicker con el valor dado.  La numeración de los elementos de la lista es a partir de 1 (1, 2, 3, …, n) hasta el final de los “n” elementos que posee la lista.  Si el valor no aparece, SelectionIndex se establece en 0. |
| Shape | Especifica la forma del botón (por defecto, redondeada, rectangular, ovalada). La forma no será visible si se visualiza una imagen. |
| ShowFeedback | Especifica si un feedback visual se debe mostrar un botón que como una imagen como fondo. |
| ShowFilterBar | Devuelve el estado actual de ShowFilterBar indicando si el Filtro de la barra de búsqueda se mostrará en ListPicker o no |
| Text | Texto Título para mostrar en la lista de selector. |
| TextAlignment | Alineación del control (Izquierda, centro o derecha.) |
| TextColor | Determina Color para el texto. |
| Title | Esta propiedad es Título opcional y aparece en la parte superior de la lista de opciones. |
| Visible | Especifica si el componente debe ser visible en la pantalla. El valor es true si el componente está mostrando y falso si oculta. |
| Width | Recuadro anchura (x-size). |
| ItemTextColor | El color del texto de los Elementos del ListPicker. |
| ItemBackgroundColor | El color de fondo de los elementos ListPicker. |

El siguiente video nos muestra cómo elaborar un sencillo conversor de unidades de longitud usando un selector de lista en app inventor.

|  |
| --- |
|  |
| Video 6. Listas en App Inventor |

**Lección 3.4.** Componente Spinner Este control es una lista desplegable, que muestra una ventana emergente con un conjunto de elementos. Dichos elementos se definen en el Diseñador o en el Editor de Bloques poniendo en la propiedad “Elementos Desde Cadena” una sucesión de elementos separados por comas (por ejemplo, opción 1, opción 2, opción 3), o en el Editor de Bloques, indicando en la propiedad Elementos el nombre de una Lista.

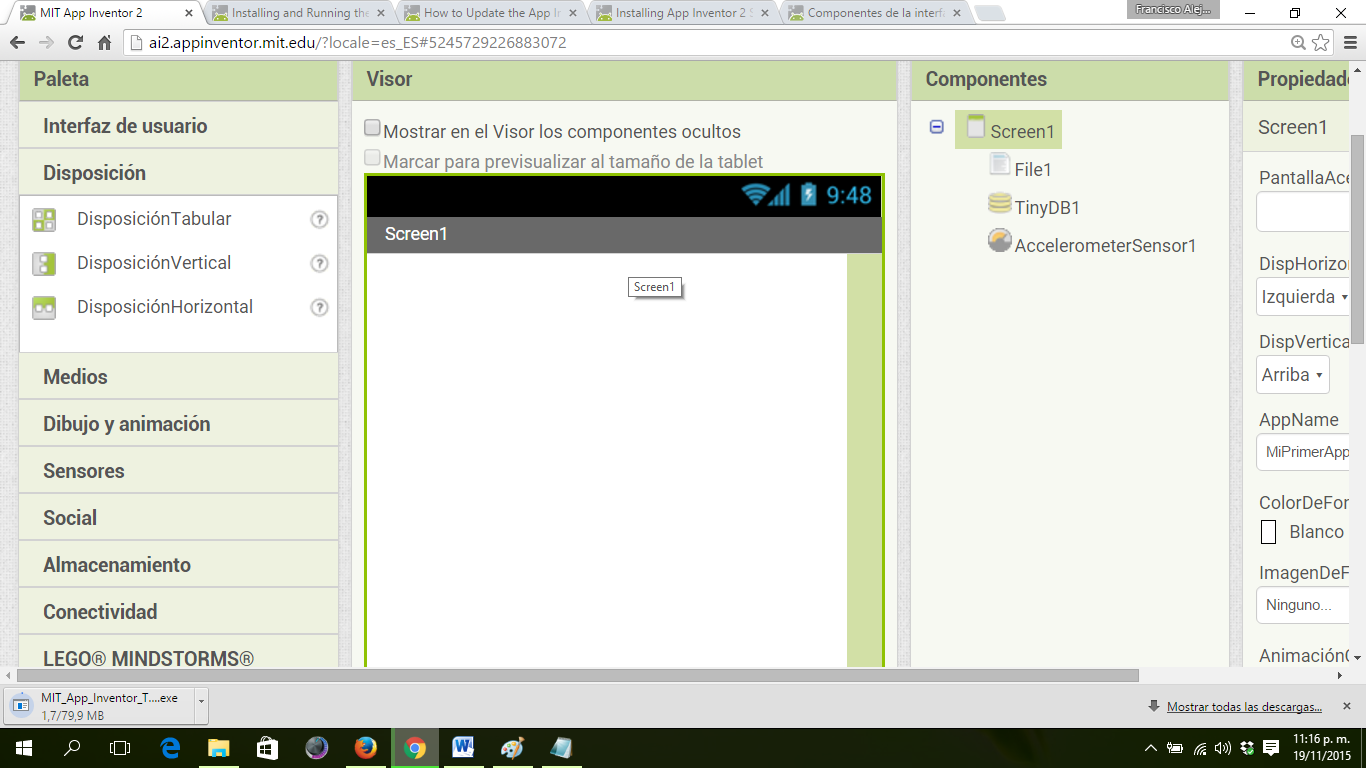
|  |
| --- |
|  |
| Imagen 16. Control para listas desplegables |

|  |  |  |
| --- | --- | --- |
| **Propiedades** | | |
| ElementsFromString | Elementos Desde Cadena |  |
| Width | Ancho del control |
| Prompt | Prompt |
| Selection | Indica cual es el elemento seleccionado por el usuario en el control |
| Visible | Especifica si el componente debe ser visible en la pantalla. El valor es true si el componente está mostrando y falso si oculta. |

En el siguiente video realizaremos un programa que permita convertir un valor a su equivalente en otras monedas (pesos, dólares, y euros)

|  |
| --- |
|  |
| Video 7. Listas desplegables |

**Unidad 4 COMPONENTES AVANZADOS.**

**Lección 4.1.** Uso de Layout 

|  |
| --- |
|  |
| Imagen 17. Manejo de Layouts |

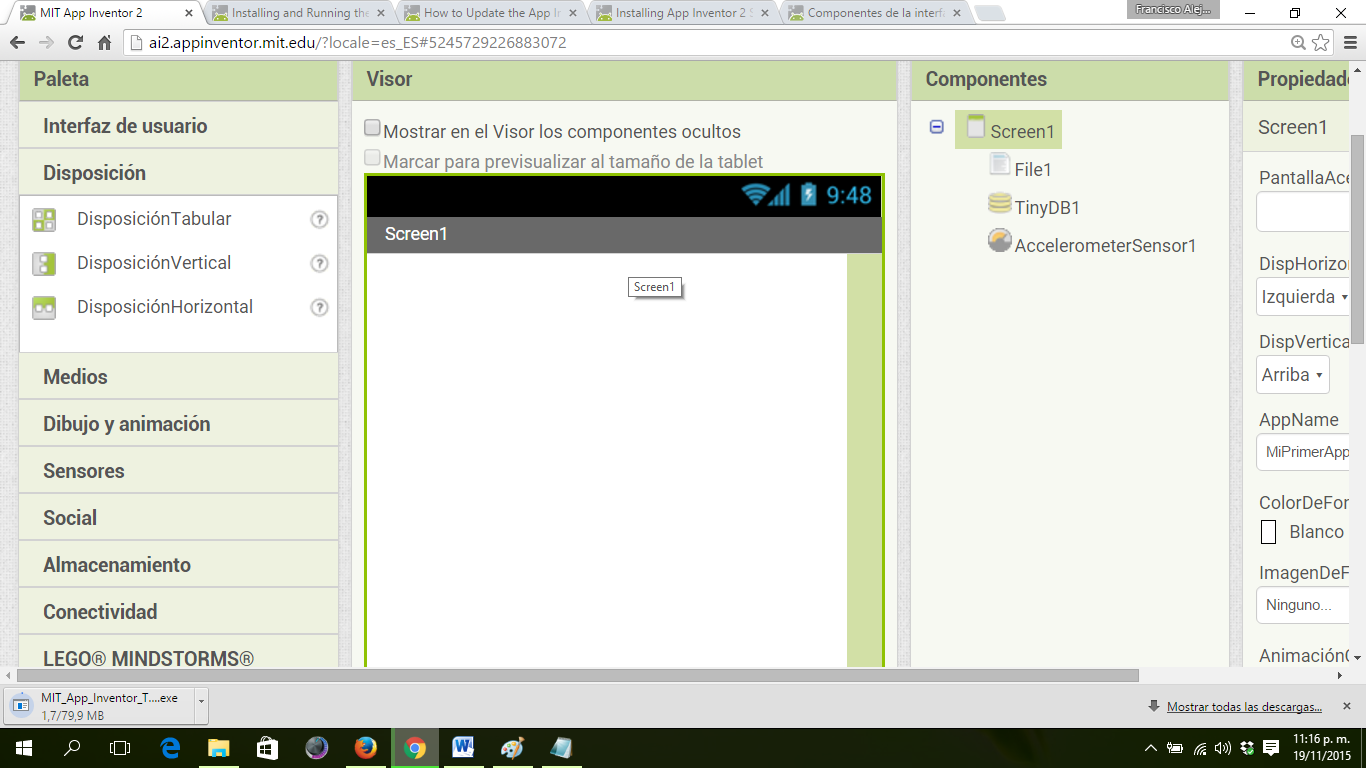
Los Layouts (Disposiciones) son elementos que permiten ubicar la posición de los controles en la pantalla.

Mit App inventor maneja tres tipos de Layouts:

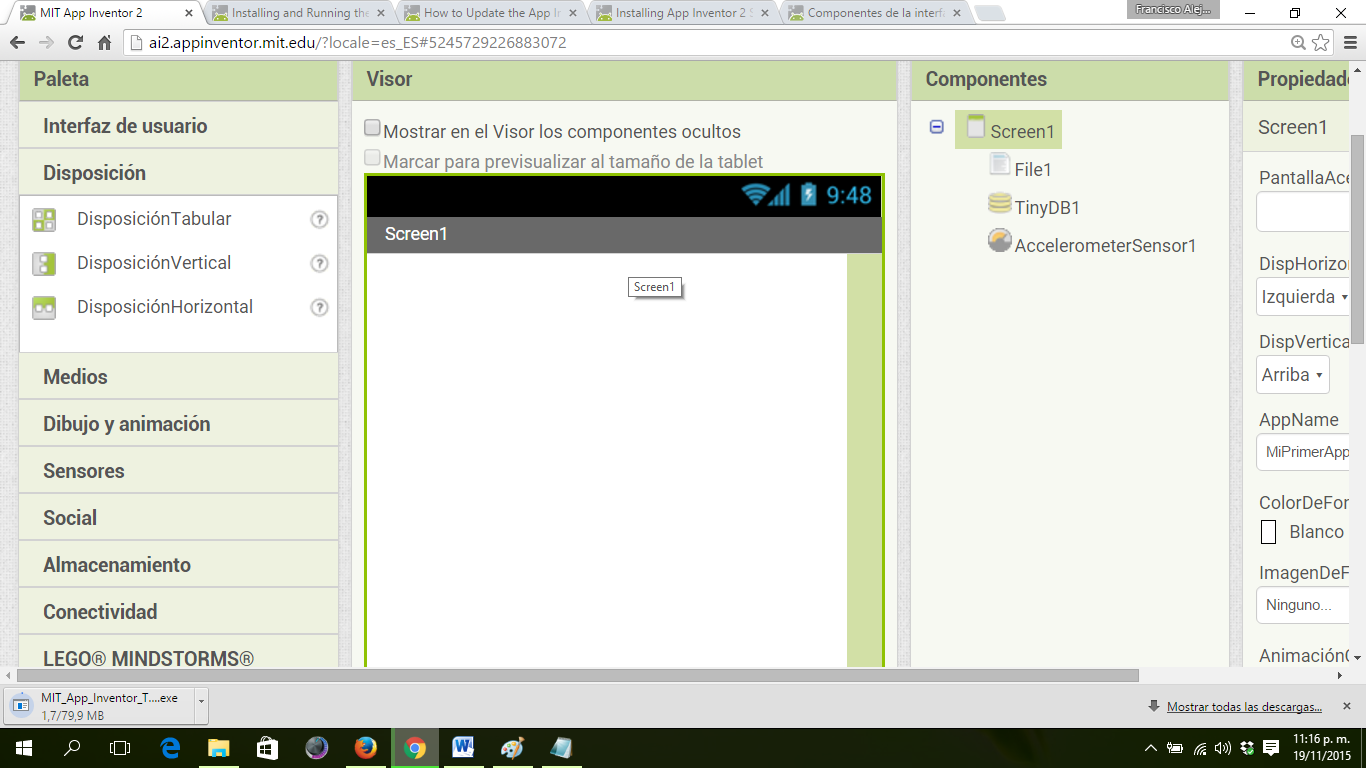
* Tabulares
* Horizontales
* Verticales

La Disposición tabular permite ubicar los componentes en una tabla que maneja filas y columnas. Los componentes están ubicados en una cuadrícula, con no más de un componente visible en cada celda. Si varios componentes ocupan la misma celda, sólo el último componente será el que se visualice.

|  |  |  |
| --- | --- | --- |
| **Propiedades del Layout Tabular** | | |
| Visible | Si está activo, los componentes y sus contenidos son visibles. |  |
| Rows | Numero de Filas de la tabla. |
| Columns | Número de columnas en la tabla. |
| Height | Altura de la tabla (de tamaño y) |
| Width | Ancho de la tabla (de tamaño x ) |

El Layout Verticares un elemento de formato en el que se pueden colocar otros elementos que deben aparecer uno debajo de otro (el primero estará arriba, el siguiente debajo, etc).

|  |  |  |
| --- | --- | --- |
| **Propiedades del Layout Vertical** | | |
| AlignHorizontal | Permite Alinear de forma horizontal los controles que se pongan dentro del Layout (izquierdo, centro, derecha) |  |
| AlignVertical | Permite Alinear de forma Vertical los controles que se pongan dentro del Layout (superior, centrado verticalmente, inferior) |
| BackgroundColor | Determina el color del fondo del Layout |
| Image | Permite insertar una imagen de fondo al layout |
| Visible | Especifica si el Layout debe ser visible en la pantalla. El valor es “Cierto” si el componente está mostrando y “falso” si oculta. |
| Height | Determina el alto del Layout (Tamaño en Y) |
| Width | Determina el ancho del Layout (Tamaño en X) |

El layout Horizontal permute ubicar elementos que deban mostrarse de izquierda a derecha.

|  |  |  |
| --- | --- | --- |
| **Propiedades del Layout Horizontal** | | |
| AlignHorizontal | Permite Alinear de forma horizontal los controles que se pongan dentro del Layout (izquierdo, centro, derecha) |  |
| AlignVertical | Permite Alinear de forma Vertical los controles que se pongan dentro del Layout (superior, centrado verticalmente, inferior) |
| BackgroundColor | Determina el color del fondo del Layout |
| Image | Permite insertar una imagen de fondo al layout |
| Visible | Especifica si el Layout debe ser visible en la pantalla. El valor es “Cierto” si el componente está mostrando y “falso” si oculta. |
| Height | Determina el alto del Layout (Tamaño en Y) |
| Width | Determina el ancho del Layout (Tamaño en X) |

Si realizas las combinaciones adecuadas con los layouts podrás ubicar los controles de tus aplicaciones como tú quieras, el siguiente video te muestra cómo manejar los tres tipos de layouts.

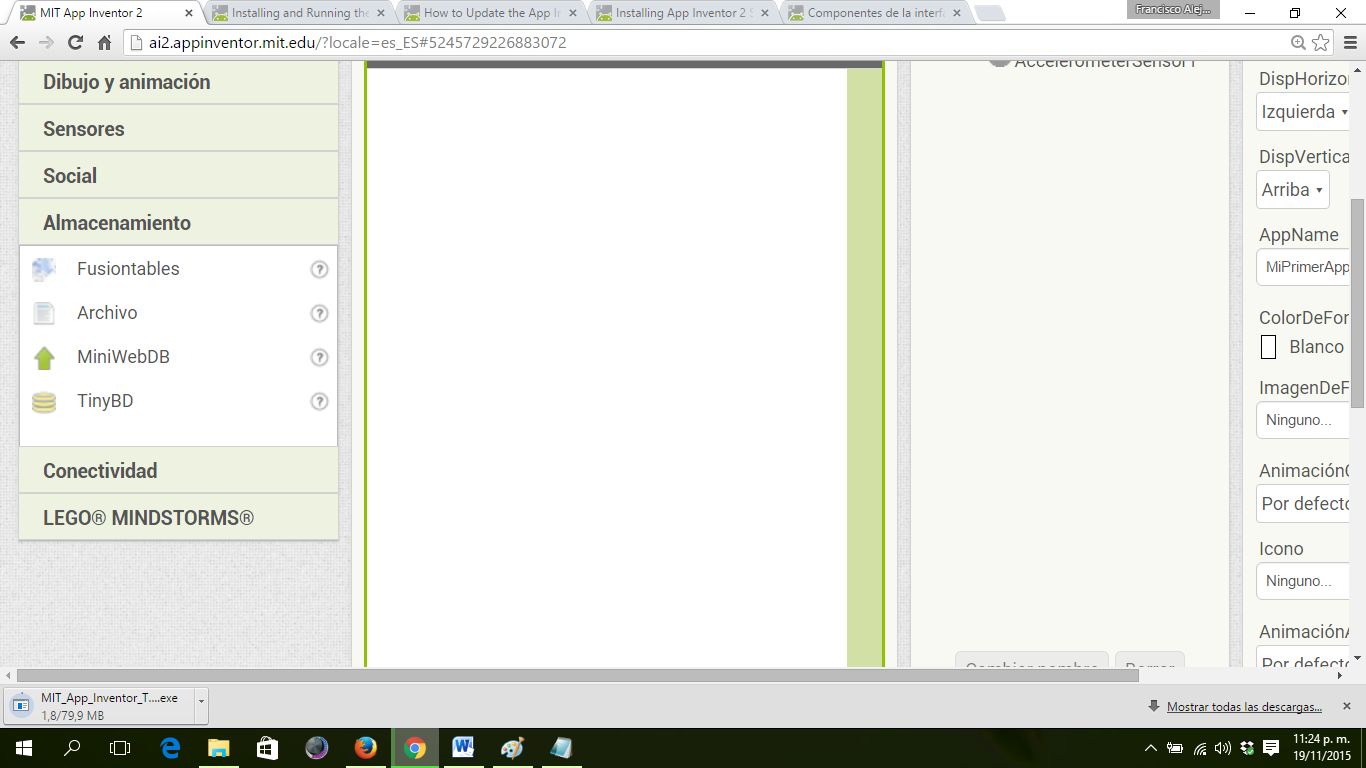
|  |
| --- |
|  |
| Video 8. Uso de Layouts |

Ahora que sabes manejar layouts podrás hacer los siguientes ejercicios y mostrárselos a tu profesor.

|  |
| --- |
|  |
| Imagen 18. Ejercicio de Layouts |

El siguiente video te muestra los principios básicos para elaborar una calculadora sencilla en App Inventor, sigue los pasos del video, termina de programar la calculadora y muéstrasela a tu profesor

|  |
| --- |
|  |
| Video 9. Ejercicio Calculadora |

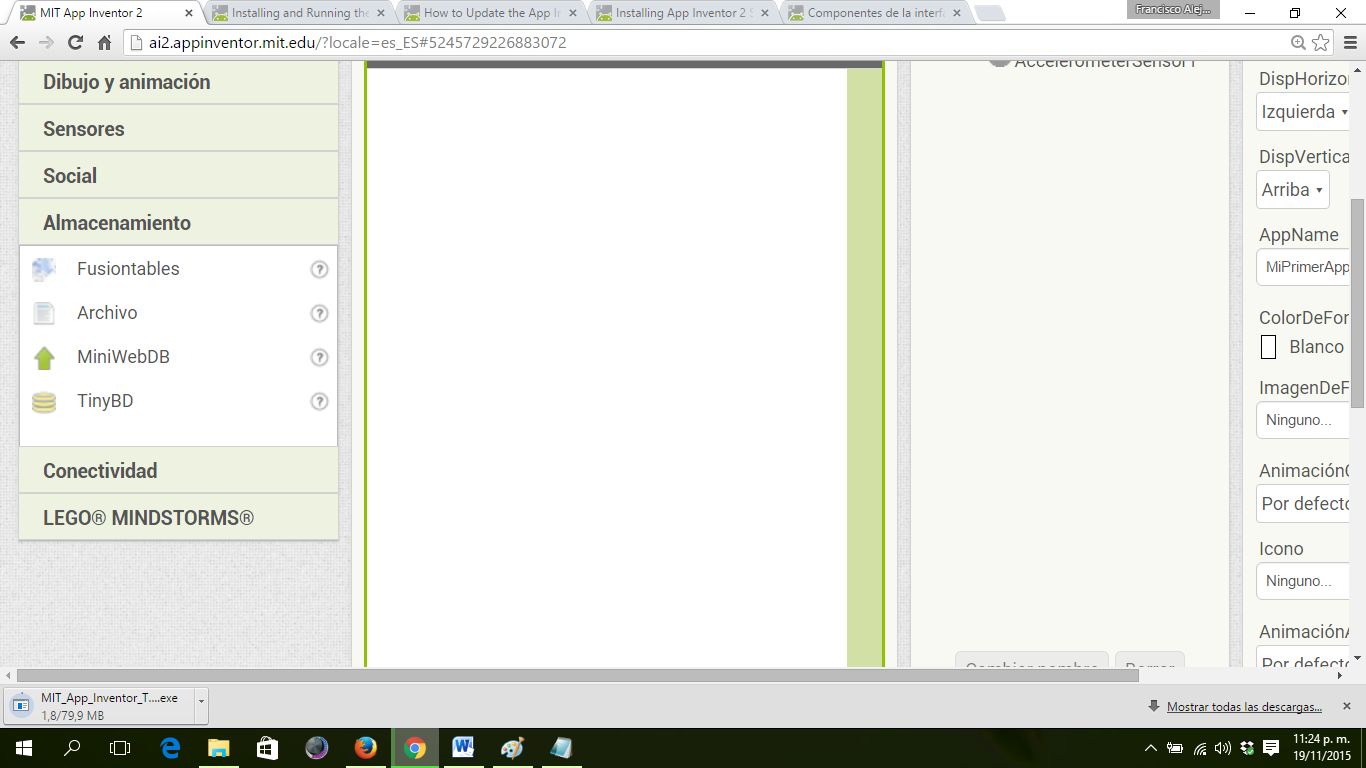
**Lección 4.2.** El componente File. Es un control no visible que sirve para guardar o recuperar archivos en formato TXT. Utiliza este componente para escribir o leer archivos en el dispositivo.

|  |
| --- |
|  |
| Imagen 19. Archivos en App Inventor |

Inicialmente sirve para escribir archivos en el directorio privado asociado a la aplicación. Como caso especial, Companion escribe los archivos en /sdcard/AppInventor/data para facilitar la depuración. Si la dirección comienza con una barra invertida (/), el archivo se escribirá a partir de /sdcard. Por ejemplo, si se especifica el archivo com /miArchivo.txt, se almacenará realmente en /sdcard/miArchivo.txt.

El siguiente video te muestra cómo elaborar un programa que almacene información en un cuadro de texto y posteriormente la envía a un archivo.

|  |
| --- |
|  |
| Video 10. Manejo de Archivos |

**Lección 4.3.** TinyDB. 

TinyDB es un componente no visible que puede almacenar datos.

Las aplicaciones creadas con App Inventor se inicializan cada vez que se ejecutan; si una aplicación inicializa el valor de una variable y el usuario termina la ejecución de la aplicación, el valor de esa variable se perderá, y no estará disponible la próxima vez que se ejecute la aplicación. TinyDB es un almacén de datos *persistente* para la aplicación, lo que significa que los datos estarán disponibles cada vez que se abra la aplicación. Un ejemplo de uso puede ser un juego que guarda las puntuaciones más altas cada vez que se juega.

Los datos son cadenas de texto que se almacenan en *etiquetas (tags)*. Para almacenar datos, se debe de especificar la etiqueta que corresponde a ese dato. Posteriormente, los datos se pueden recuperar con esa misma etiqueta.

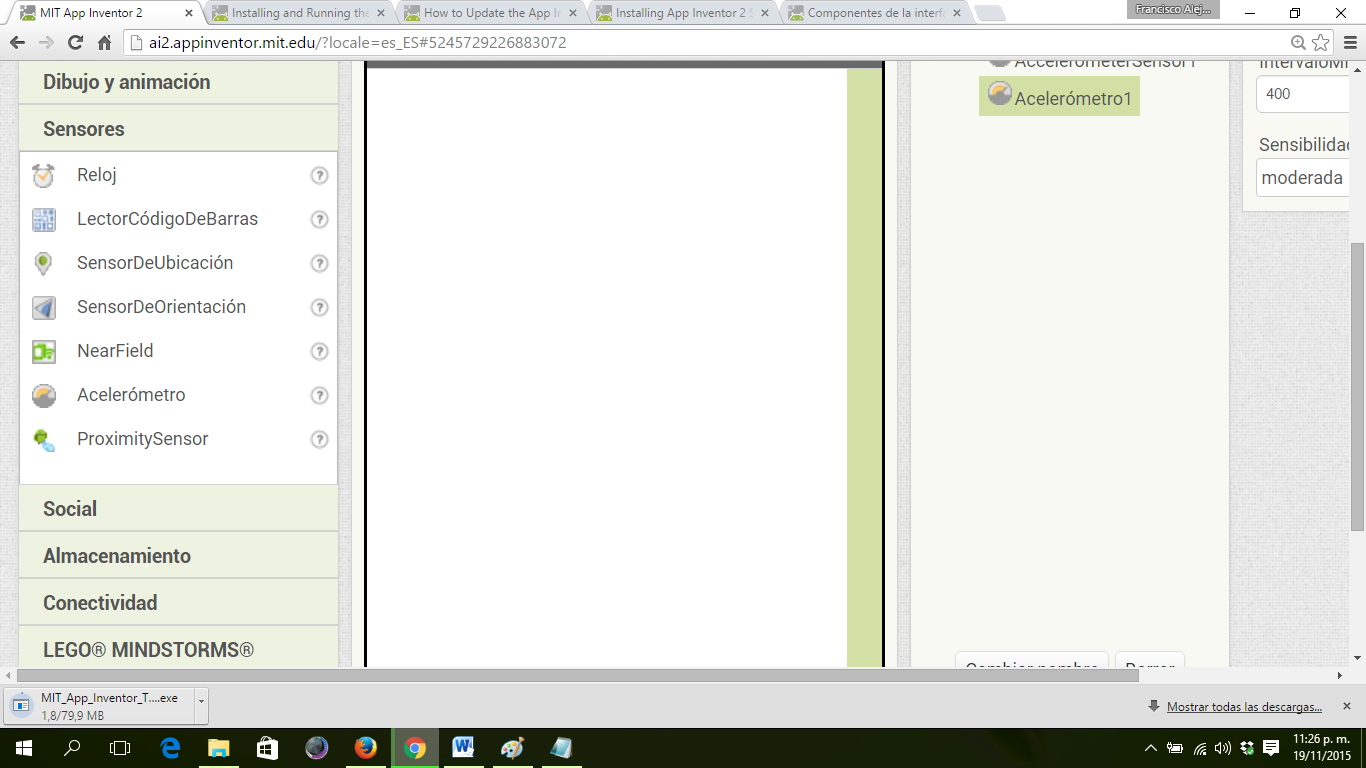
|  |
| --- |
|  |
| Imagen 20. Bases de datos (TinyDB) |

Solamente existe una base de datos para cada aplicación. Aunque se utilicen distintas TinyDBs en una aplicación, todas ellas apuntan al mismo almacén. Se deben utilizar distintas etiquetas para guardar diferentes datos, y no usar diferentes almacenes. TinyDB es única para cada aplicación, y no se puede utilizar para pasar datos a otras aplicaciones instaladas en el dispositivo, pero se sí puede utilizar para pasar datos entre pantallas en la misma aplicación.

Cuando TinyDB se utiliza con la aplicación Companion, todas las aplicaciones comparten la misma DB. Esto no ocurre cuando la aplicación se empaqueta como un archivo apk. Pero mientras se desarrolla, se debe ser cuidadoso en situaciones en las que las mismas etiquetas se usan para distintas aplicaciones.

El siguiente video nos muestra como almacenar datos usando TinyDB, y posteriormente abrimos otra ventana donde recuperamos la información almacenada en la base de datos

|  |
| --- |
|  |
| Video 11. Bases de datos en App Inventor |

**Lección 4.4.** AcelerometerSensor

AcelerometerSensor es un componente no visible que detecta movimiento del dispositivo móvil y mide aceleración aproximada en tres dimensiones utilizando unidades del sistema internacional (m/s2).

|  |
| --- |
|  |
| Imagen 21. El Sensor de acelerómetro |

Las componentes son:

* **xAccel**: 0 cuando el dispositivo no está en movimiento en una superficie plana, positivo cuando el dispositivo se inclina hacia la derecha (por ejemplo, cuando la se levanta por la parte izquierda), y negativo cuando el dispositivo se inclina hacia la izquierda (por ejemplo, levantándolo por la parte derecha).
* **yAccel**: 0 cuando el dispositivo se encuentra en una superficie plana (sin movimiento), positivo cuando se levanta la base, y negativo cuando se levanta la parte superior.
* **zAccel**: Tiene el valor de -9.8 (la gravedad de la tierra en metros por segundo) cuando el dispositivo no está en movimiento y paralelo al suelo con la pantalla hacia arriba, 0 cuando el dispositivo está en posición perpendicular al suelo, y +9.8 con la pantalla hacia abajo. Este valor puede verse afectado si se acelera en contra de la gravedad

|  |  |  |
| --- | --- | --- |
| **Propiedades del Acelerometro** | | |
| Enabled | Si está activo, habilita el acelerómetro |  |
| MinimumInterval | Establece el intervalo mínimo entre batidos de teléfono |
| Sensitivity | Un número que codifica la sensibilidad del acelerómetro es.  Las opciones son :  1 = débil  2 = moderado  3 = fuerte |

El siguiente video nos muestra como capturar los datos del acelerómetro en Mit app inventor.

|  |
| --- |
|  |
| Video 12. Como capturar los datos del acelerómetro |

**Web grafía Spbre App Inventor**

* Página Oficial de App Inventor

<http://appinventor.mit.edu/explore/index-2.html>

* Tu App Inventor

<http://www.tuappinvetorandroid.com/blog/cursos-para-novatos/>

* KIo4

<http://kio4.com/appinventor/>

* Aprende App Inventor

<https://sites.google.com/site/aprendeappinventor/>

* App Inventor en Español

<https://sites.google.com/site/appinventormegusta/>

* 100 Best AppInventor Videos

<http://meta-guide.com/videography/100-best-appinventor-videos>

* Curso de App Inventor

<http://webpub.esi.uclm.es/img/upload/plugin/ESI-TechLab-AppInventor2-2015beta.pdf>

* guía de iniciación a app inventor

<http://codeweek.eu/resources/spain/guia-iniciacion-app-inventor.pdf>

* Háblame: Tu primera app de App Inventor

<http://www.technovationchallenge.org/wp-content/uploads/2015/02/H%C3%A1blame.pdf>