**Pensamiento computacional**

**y primer programa**

﻿//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

**App inventor**

​

Inicio platicándote de la historia del App inventor y posteriormente describiré el entorno para el desarrollo de aplicaciones.

​

Fue concebido por Google en 2011 y posteriormente, por diversos motivos no muy claros, fue "abandonado". A partir de ese momento su desarrollo lo retomó el MIT (Instituto Tecnológico de Massachusetts) y aunque aún está en su versión beta (de prueba), ya cuenta con muchas características funcionales.

​

App inventor es una aplicación distribuida, es decir, una aplicación donde sus componentes se ejecutan en diferentes plataformas conectadas a través de una red.

​

Para acceder a él hay que ir a la siguiente dirección electrónica: <http://appinventor.mit.edu/explore/>

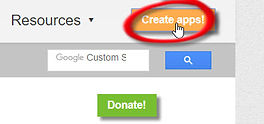
Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

La aplicación se basa en desarrollo visual a través de su editor, por lo que no son necesarios grandes conocimientos de programación, pero sí, el entendimiento del desarrollo de **pensamiento computacional** ya que las aplicaciones que desarrollaremos se realizan principalmente mediante la composición de elementos y bloques que utilizan estructuras de control.

​

Para acceder a la plataforma Web es necesario contar con una cuenta de Gmail:



Interfaz de usuario gráfica, Aplicación

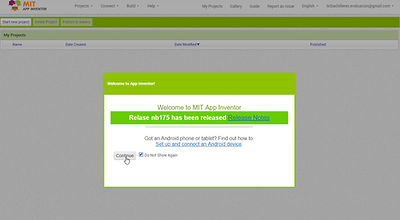
Descripción generada automáticamente

Al iniciar por primera vez en la misma, deberás aceptar los términos del servicio:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Después se te solicitará llenar una breve encuesta (en inglés). Puedes hacerlo en ese momento, más tarde, o si lo deseas puedes no llenarla jamás:



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

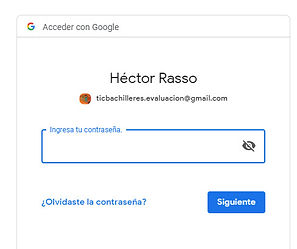
Finalmente te mostrará el siguiente mensaje:

​

No tienes ningún proyecto en **App Inventor 2**. Para aprender a usarlo , haz clic en el enlace "**Guía**" en la parte superior; o para iniciar tu primer proyecto, haz clic en el botón "**Iniciar nuevo proyecto**" (**Star New Proyect**)en la parte superior izquierda. Diviértete inventando!



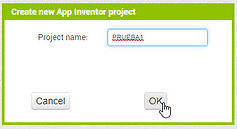
Para las siguientes ocasiones que ingreses se te solicitará únicamente la contraseña de tu cuenta de correo gmail:



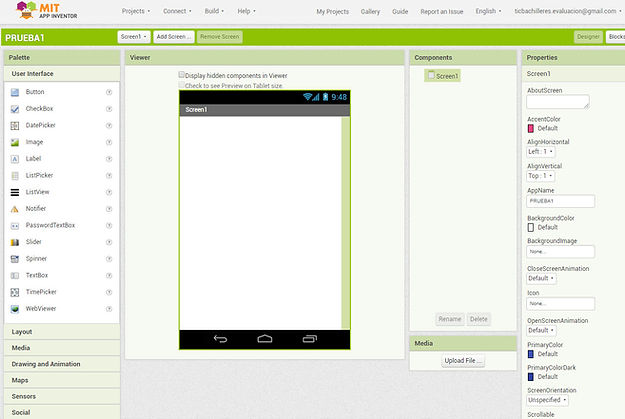
Para crear un nuevo proyecto hay que dar clic en **Star New Proyect.**Los proyectos se almacenan en la propia plataforma y la información se guarda de manera automática y en cualquier momento podremos recuperar un proyecto anterior.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



Una vez determinado el proyecto, la plataforma nos dirige a la pantalla del diseñador App Inventor:



La interfaz del editor Web se divide en varias secciones, en la parte central tenemos el editor de pantallas de nuestra aplicación (**Viewer)**.

Por defecto, la aplicación solo tiene una pantalla/ventana (**Screen 1)**, pero se pueden agregar más.

​

Empezaremos a utilizar el App Inventor creando una aplicación muy básica que genere **un saludo personalizado.**

​

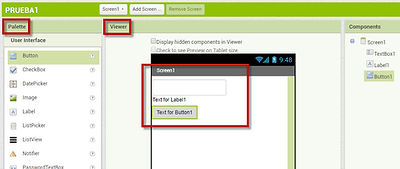
Iniciaremos incluyendo en la pantalla principal (**Screen 1)** los siguientes elementos, que basta arrastrarlos desde la sección **Palette** hasta el **Viewer**:

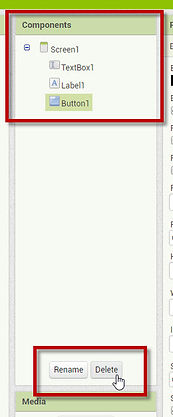
1) Campo de entrada de

texto (**TextBox**)

2) Etiqueta textual (**Label**)

3) Botón (**Button**)





Si deseas eliminar algún elemento, basta con seleccionarlo y pulsar del teclado el botón **SUPRIMIR** o **BORRAR**.

​

Para seleccionar un elemento, puede resultar más útil utilizar la vista **Components**, ya que en ocasiones resulta difícil identificar un elemento concreto en la pantalla, sobre todo si hay muchos de ellos.

​

Por defecto, el editor asigna nombres genéricos a los elementos que se incluyen en pantalla: **Screen 1, TextBox1, Label1, Button1.** Para que la lógica de pensamiento computacional resulte más fácil de comprender, es conveniente modificar los nombres de los controles colocando otros más descriptivos.

​

Para modificar los nombres de los elementos, en la misma vista **Components,** basta seleccionar el elemento en cuestión y dar clic en el botón **Rename.**

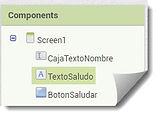
**​**

Para el ejemplo que estamos realizando, modifica los nombres de los siguientes elementos:

​

​

* TextBox   --   CajaTextoNombre
* Label1     --  TextoSaludo
* Button1   --  BotonSaludar



A continuación, modificaremos las propiedades de algunos de los elementos utilizando la vista **Propierties**. Para ello, selecciona el elemento **TextoSaludo**.

​

* Asigna el color de fondo amarillo. (Propiedad **BackgroundColor: Yellow**)

​​

* El texto debe ser letra negrita (Propiedad **FondBold**)

​

* Aumenta el tamaño del texto a 20 (Propiedad **FontSize**)

​

* Cambia el texto por defecto a **"Sin saludo"** (Propiedad **Text**)

​

* Cambia la alineación para que aparezca en el centro (Propiedad **TextAlignment: center**)

​

* Cambia el ancho para que ocupe todo el espacio disponible en la pantalla (Propiedad **Width valor: Fill parent**). **Fill parent** significa que se extiende por todo el espacio disponible en la pantalla, en cambio si seleccionas **Automatic** el elemento ocupará el mínimo espacio posible para su representación

​

Ahora modificaremos las propiedades del elemento **BotonSaludar**. Cambia su texto (Propiedad **Text**) por "**Saluda!**", puedes incluir cualquier otra modificación de aspecto que sea de tu preferencia: cambiar el color de fondo (Propiedad **BackgroundColor**), la forma (Propiedad **Shape**), incluir una imagen (Propiedad **Image**), etcétera.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



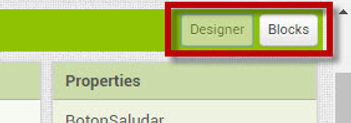
Una vez que hemos definida la interfaz de usuario, es el momento de programar la lógica de operación de la *app*. Para ello, App Inventor utiliza un editor de bloques.

​

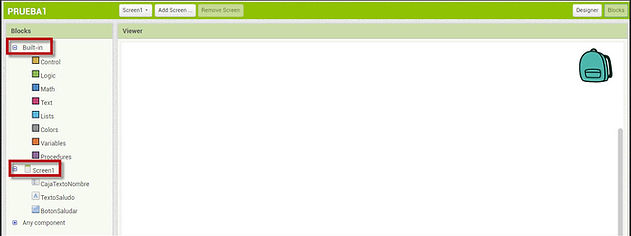
Al editor de bloques se accede pulsando sobre el botón **Blocks** situado en la parte superior derecha de la Web.

​

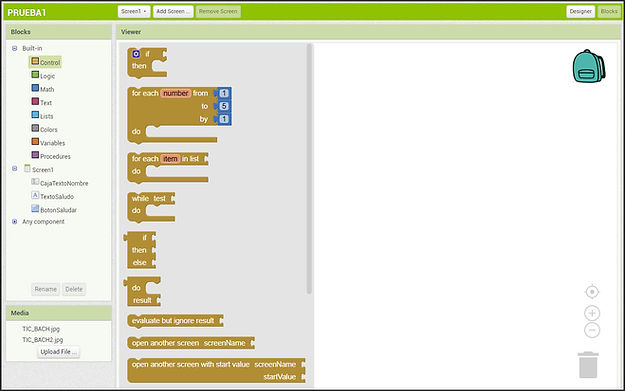
Podemos movernos entre el editor y el diseñador de bloques alternando los botones **Designer** y **Blocks**.



En la parte izquierda del editor de bloques tenemos una lista dónde están todas las estructuras de control que podemos utilizar (**Built-in**) y todos los componentes que tenemos en la pantalla (**Screen1**).



Las estructuras de control son el mismo conjunto de Pensamiento Computacional revisadas en **TIC 3** en el tercer semestre, con la salvedad que App Inventor las designará ahora en inglés.



Al seleccionar cada una de las categorías nos aparecerá un panel desplegable donde se encuentran todos los bloques que se pueden utilizar.

​

* **Control**. Contiene los bloques de control tradiciones: **if / else** (Si / entonces), **for** (Haz / hasta), **while** (Haz / mientras). Además, contiene bloques que permiten realizar evaluaciones y abrir / cerrar ventanas (**Screens**) y la aplicación.   Para más información, consultar [CONTROL](http://appinventor.mit.edu/explore/ai2/support/blocks/control.html).

​

* **Logic**. Contiene bloques para manejar los operadores lógicos y especificar condiciones /comparaciones.   Para más información, consultar [LOGIC](http://appinventor.mit.edu/explore/ai2/support/blocks/logic.html).

​

* **Math**. Contiene los bloques para trabajar con elementos numéricos, estos elementos consisten principalmente en elementos de comparación y diferentes tipos de funciones matemáticas.   Para más información, consultar [MATH](http://appinventor.mit.edu/explore/ai2/support/blocks/math.html).

​

* **Text. Contiene los bloques para definir y manipular textos, permite definir una gran cantidad de operaciones básicas sobre cadenas de texto: unión, evaluación, reemplazos, etcétera.   Para más información, consultar**[**TEXT**](http://appinventor.mit.edu/explore/ai2/support/blocks/text.html)**.**

​

* **Lists**. Contiene los bloques que permiten gestionar listas de elementos, crearlas, manipularlas y realizar operaciones básicas con ellas.    Para más información, consultar [LISTS](http://appinventor.mit.edu/explore/ai2/support/blocks/lists.html).

​

* **Colors**. Contiene los bloques que permiten realizar acciones relacionadas con la selección o definición de Colores. Estos bloques casi no son utilizados.    Para más información, consultar [COLORS](http://appinventor.mit.edu/explore/ai2/support/blocks/colors.html).

​

* **Variables. Contiene los bloques que permiten definir variables globales (en todo el programa) o locales (dentro del procedimiento), además los correspondientes a los valores (*get*) y de modificación (*set*).    Para más información, consultar**[**VARIABLES**](http://appinventor.mit.edu/explore/ai2/support/blocks/variables.html)**.**

​

* **Procedures**. Contiene los bloques que permiten la gestión de procedimientos, los procedimientos permiten "agrupar" un conjunto de bloques bajo un mismo nombre (son similares a las funciones utilizados en otros lenguajes de programación). Ya definidos estos procedimientos pueden ser invocados posteriormente a lo largo del programa.    Para mayor información, consultar [PROCEDURES](http://appinventor.mit.edu/explore/ai2/support/blocks/procedures.html).

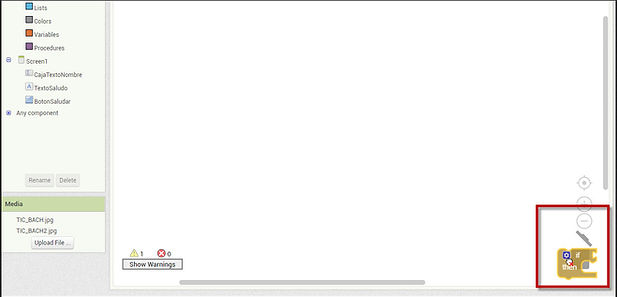
Para programar la lógica de la aplicación, es necesario seleccionar los bloques y "*arrastrarlos*" hasta el área de diseño de la pantalla. Los bloques que se incluyen en el área de diseño definen el comportamiento de la aplicación.

​

De forma parecida como se hace en Scratch, los bloques se combinan y encajan entre sí para definir la lógica computacional de una aplicación.

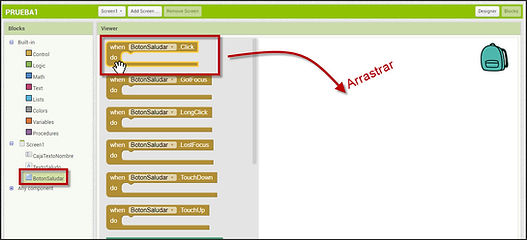
​

Toma en cuenta que en todo momento podemos eliminar los bloques que no necesitemos arrastrándolos hacia el "**cesto de basura**" que aparece en la parte inferior izquierda de la pantalla.



Iniciaremos la funcionalidad que ejecutará nuestra *app* cuando el usuario pulse el botón **BotonSaludar.**

Primero selecciona de la lista de elementos de la parte izquierda el  **BotonSaludar**y a continuación asígnale la funcionalidad **WhenBotonSaludar.Click do**:



A continuación, definiremos las funcionalidades que realizará la app sobre el botón saludar:

* Obtener el texto que el usuario escriba en el componente **CajaTextoNombre**.

​

* Formar una nueva cadena con "**Hola**" + el texto introducido en **CajaTextoNombre**

**​**

* Asignar esa nueva cadena a la etiqueta **TextoSaludo**

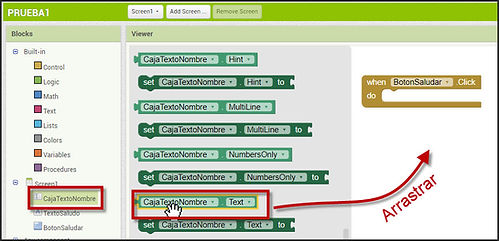
**​**

**(Es importante notar que después de la palabra**Hola**, hay un espacio en blanco)**

**​**

​

Para lograr lo anterior, primero selecciona del lado izquierdo de la pantalla el elemento **CajaTextoNombre** y busca la caja que permite obtener el valor de su propiedad **Text**y arrástrala dentro de la pantalla de trabajo **Viewer** sin realizar ningún tipo de conexión.



Para que la *app* opere como se desea, será necesario "concatenar" el valor del texto de la **CajaTextoNombre** con la cadena de texto

"**Hola**"

(recuerda que tiene un espacio final).

Para ello utilizaremos el elemento de tipo

**Text - Join**.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

El **Join** permite concatenar cadenas de texto.

​

Vamos a concatenar el texto "**Hola** " (con un espacio al final) de tipo **Text,** con el valor del texto de **CajaTextoNombre**.

​

El bloque donde se permite escribir texto literal es:



Concatenación de cadenas (Join):

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

Ahora solo falta asignar la concatenación al elemento **TextoSaludo**.

​

Para esto selecciona **TextoSaludo** y busca el bloque que permite modificar el valor de su atributo Text (**setTextoSaludo.Text**)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El texto que se le asigna es el resultado de **Join**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Todo eso se realizará cuando el usuario haga Click sobre el botón **BotonSaludar**, por lo que tendremos que encajar el bloque dentro del bloque Click

Escala de tiempo

Descripción generada automáticamente

Con esto finalizamos la implementación de la lógica del botón **BotonSaludar** y podemos probar la aplicación en el dispositivo.

​

Desde la pantalla **Designer** hay que dar clic en la opción del menú principal **Connect** y elegir la primera opción **Al companion:**

​

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Esta acción te generará un **código QR** que deberás escanear desde la aplicación App Inventor 2 instalada en tu móvil:

Interfaz de usuario gráfica, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Después de que se establece la conexión entre la aplicación y tu móvil...

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

...podrás ver su funcionamiento directamente en pantalla:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**VIDEO DE APOYO**

Imagen que contiene objeto, reloj

Descripción generada automáticamente

[*App inventor (tutorial 1)*](https://www.youtube.com/watch?v=30OcisYzjCc&t=135s)

Referencias:

- Pascual, J. García V. & González R. (2014). *Desarrollo de aplicaciones android con App Inventor, Paso a paso*. México: Jordán Pascual Estrada.

- Cabello, J.M. (2014). *Crea tus aplicaciones Android con App Inventor 2*. España: IC