



# Practica guiada

## Aplicación con sensor de ubicación

En esta práctica se va a crear una aplicación en la cual va a mostrar la ubicación en latitud y altitud además de mostrar el mapa de ubicación en tiempo real.

1.- Crear un nuevo proyecto en App Inventor y llamarlo ubicación.

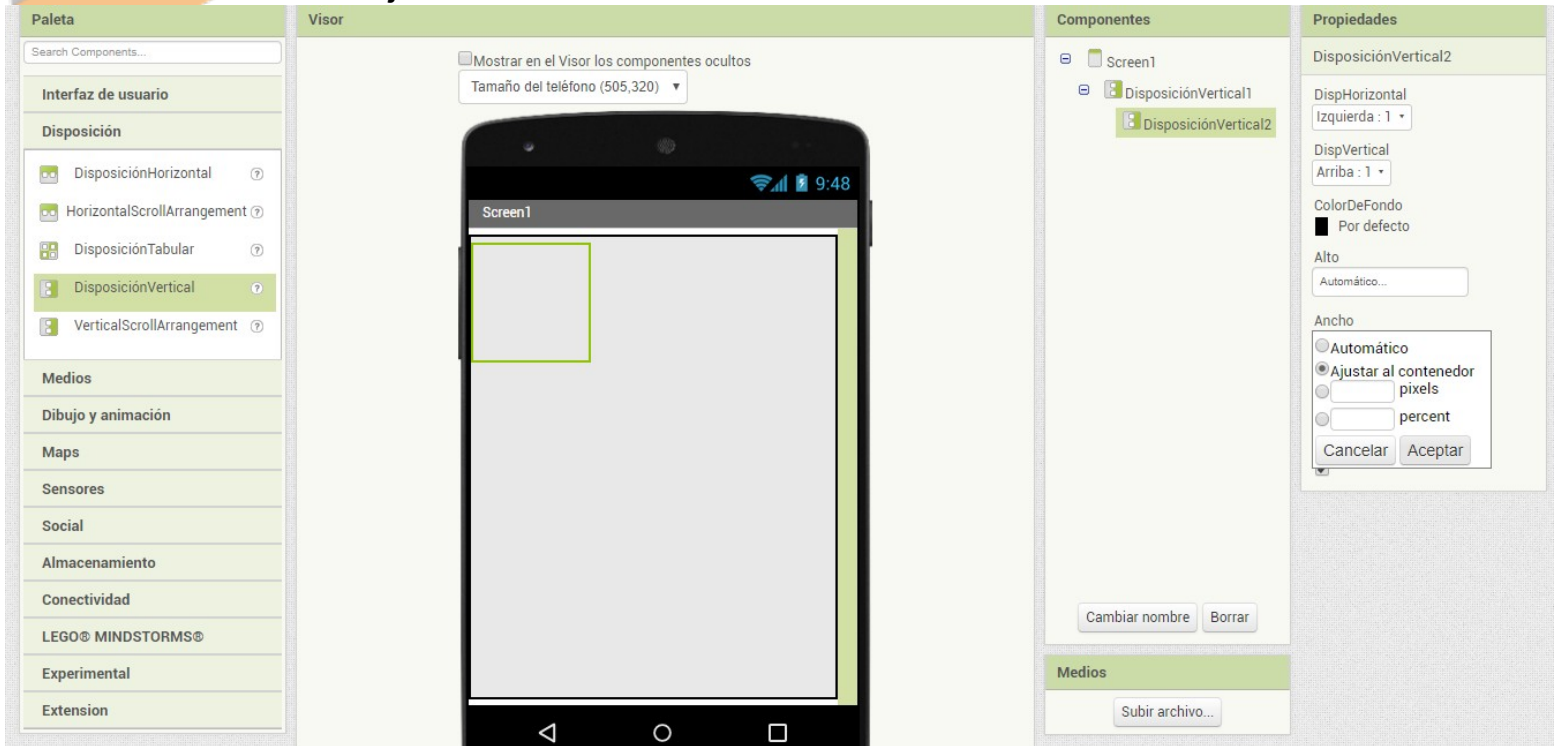
The screenshot shows the MIT App Inventor web interface. At the top, there are navigation tabs: 'Proyectos', 'Conectar', 'Generar', 'Settings', 'Ayuda', 'Mis proyectos', 'View Trash', 'Gallery', 'Guía', and 'Informar de problema'. Below these are buttons: 'Comenzar un proyecto nuevo...', 'Borrar proyecto', 'Publish to Gallery', and 'View Trash'. A table titled 'Proyectos' lists existing projects with columns for 'Nombre', 'Fecha de creación', and 'Fecha modificación'. A dialog box titled 'Crear un nuevo proyecto de App Inventor' is open, with a text input field containing 'Ubicación' and 'Nombre del proyecto:' label. There are 'Cancelar' and 'Aceptar' buttons at the bottom of the dialog.

Nombre	Fecha de creación	Fecha modificación
pelota	29/3/2020 19:45:54	19/4/2020 20:32:04
calculadora_simple	22/3/2020 23:49:00	29/3/2020 19:41:06
versiones	29/2/2020 21:39:15	1/3/2020 22:14:41
Luis_carlos	20/2/2020 8:10:57	27/2/2020 8:31:06
reproductor	23/2/2020 20:10:38	20 10:38:40
prueba	19/2/2020 11:20	20 9:26:43

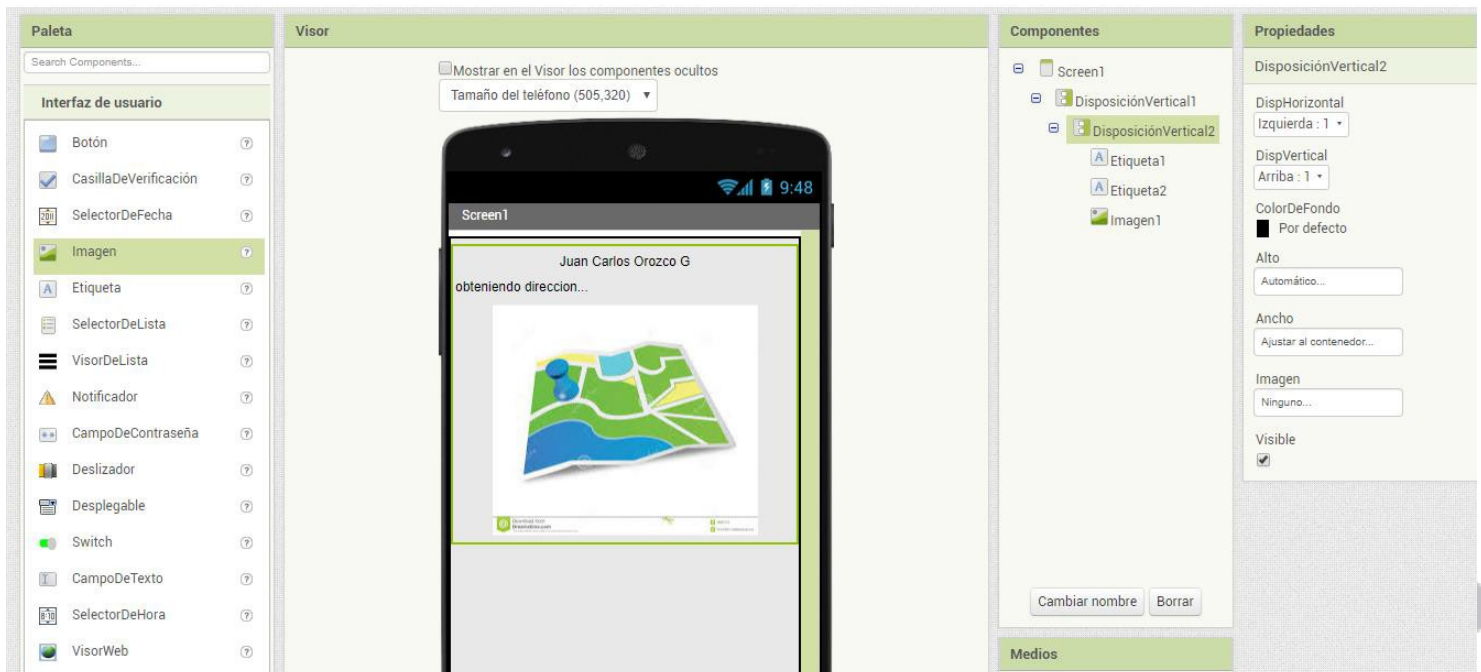
2.- Agregar una disposición Vertical, ajustar al contenedor en ancho y alto.

The screenshot shows the MIT App Inventor interface with a mobile phone simulator in the center. On the left, the 'Interfaz de usuario' palette is open, showing various components. The 'Disposición' section is selected, and 'DisposiciónVertical' is highlighted. On the right, the 'Medios' palette is open, and 'DisposiciónVertical1' is selected. The properties panel for 'DisposiciónVertical1' is visible, showing settings for 'DisposiciónHorizontal', 'DisposiciónVertical', 'ColorDeFondo', 'Alto', and 'Visible'. The 'Alto' property is set to 'Ajustar al contenedor'. The 'Visible' property is checked.

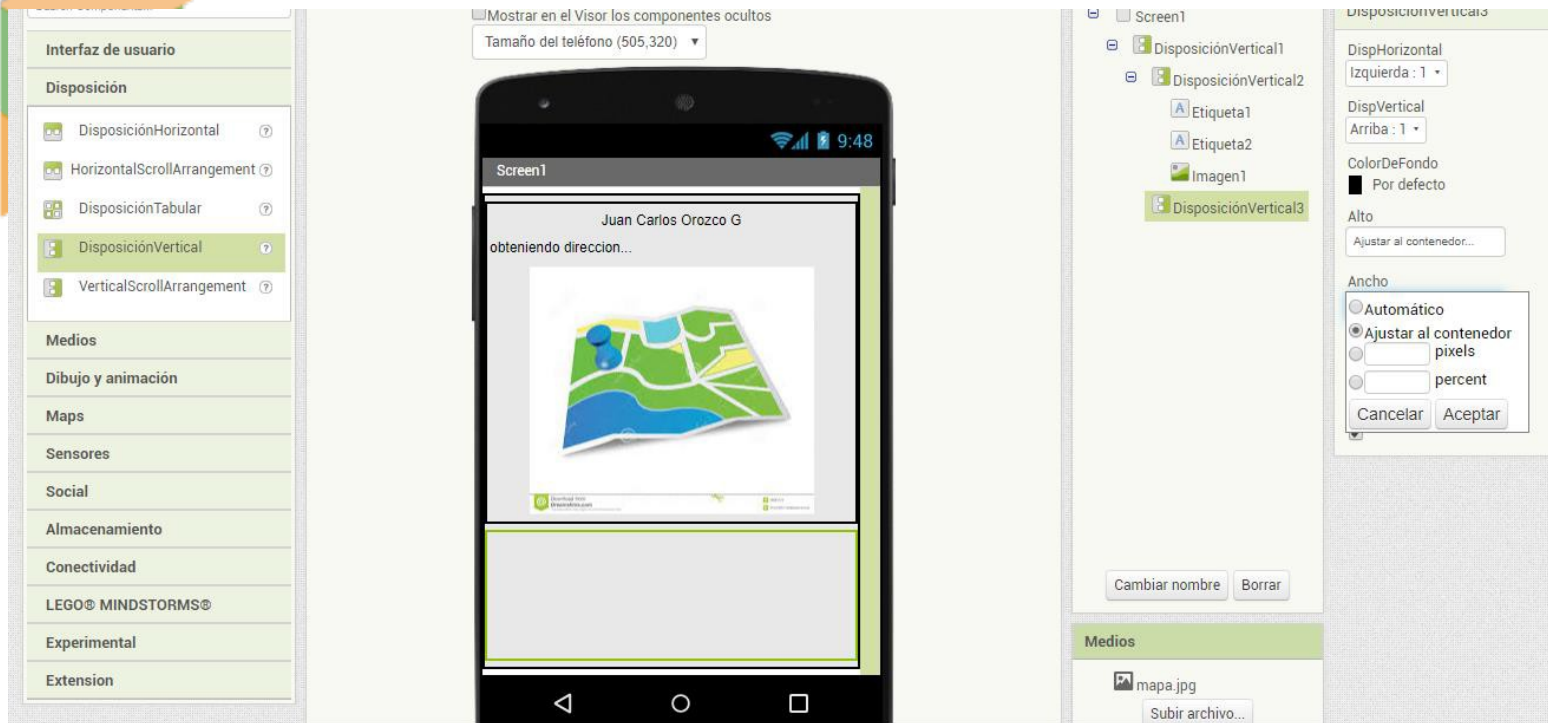
3.-Agregar dentro de la disposición agregada otra disposición vertical, con ancho ajustado al contenedor



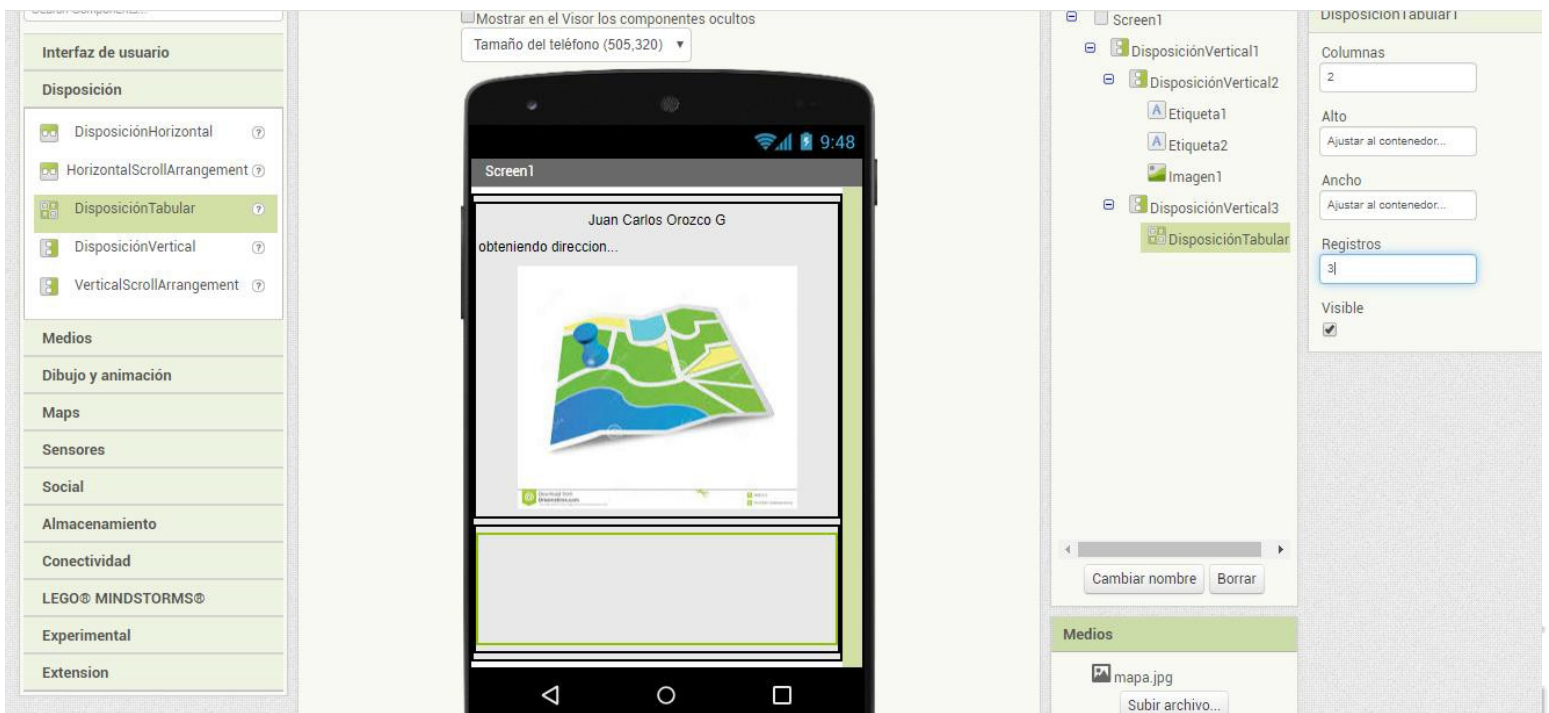
4.- Agregar dentro de esta nueva disposición Vertical, un Label o etiqueta para tu nombre, otro diciendo **Obteniendo dirección** y una imagen con una foto/imagen de mapa



5.-Agregar otra disposición vertical debajo de la ya agregada anteriormente, y ajustar el ancho y el alto al contenedor.



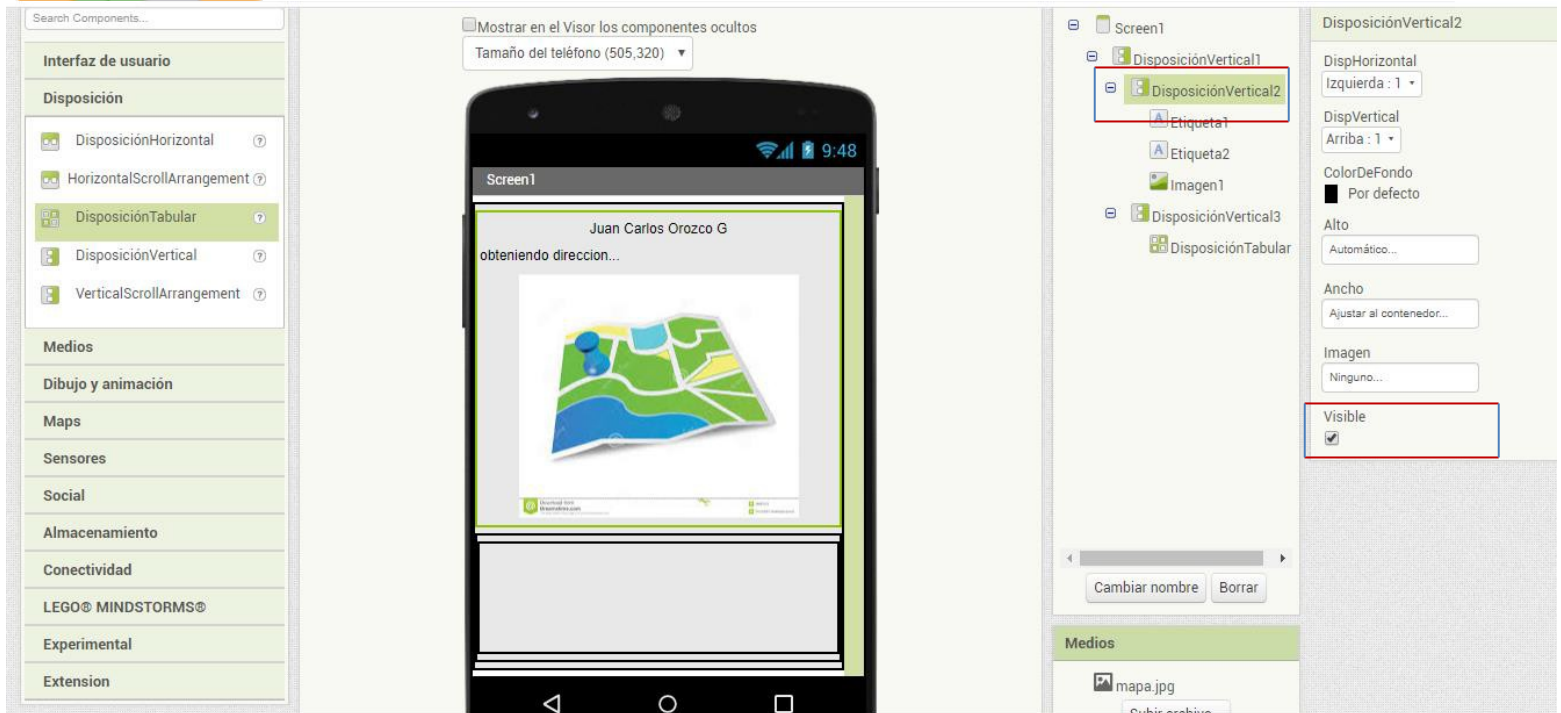
6.- Agregar una disposición tabular dentro la disposición agregada, ajustar ancho al contenedor y el número de registros la cambiamos a 3.



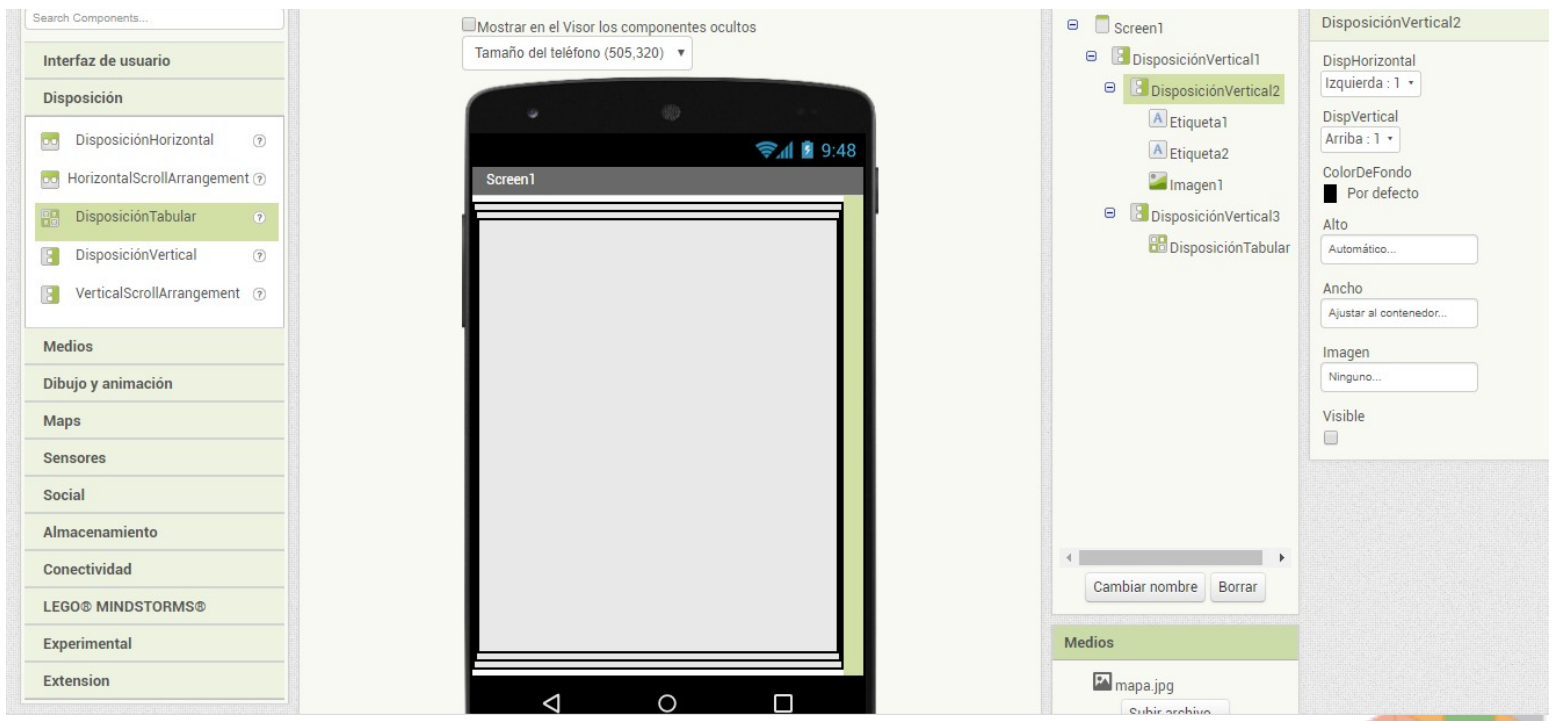


7.-Para continuar trabajando con la disposición tabular, podemos esconder la disposición Vertical que contiene tu nombre y el mapa.

Para eso seleccionamos el elemento DisposicionVertical2 y deseleccionamos la opción Visible



Quedando de la siguiente manera:

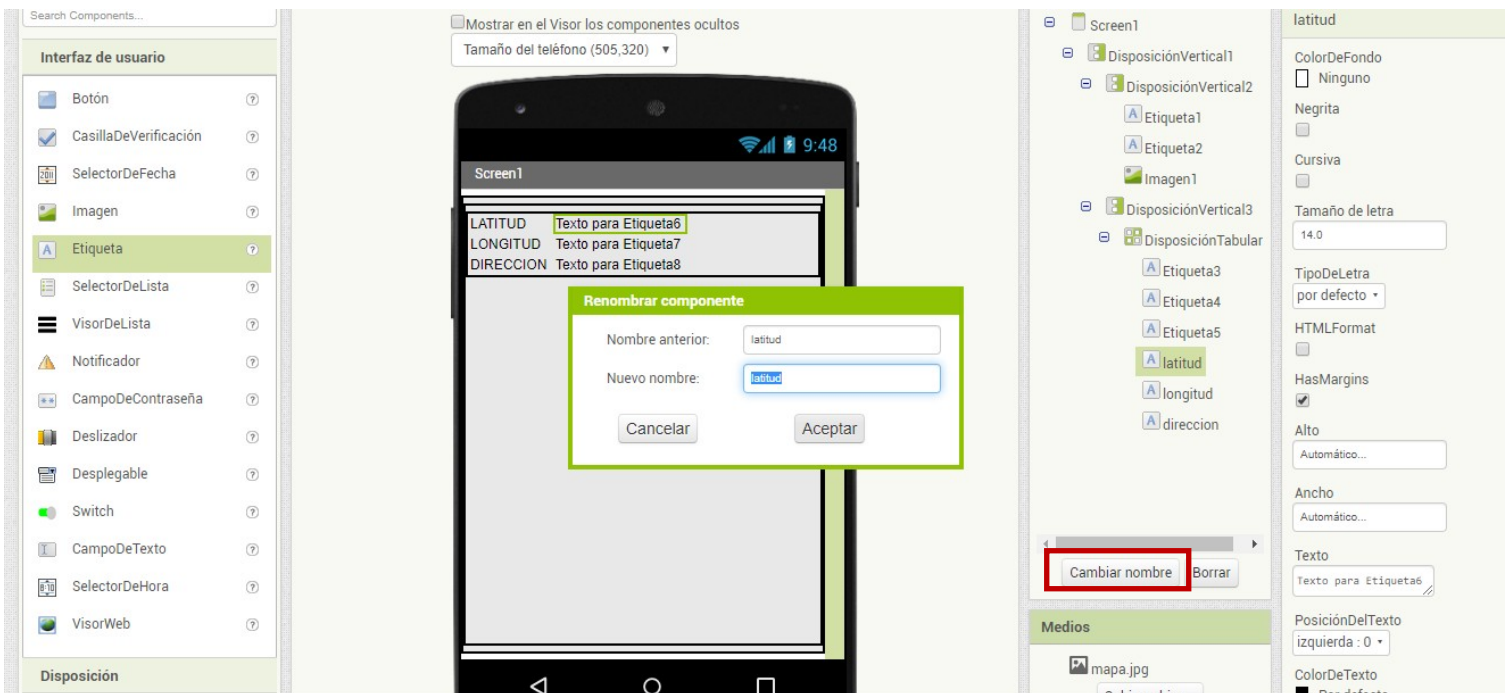


8.- dentro de la disposición tabular, agregar 6 label, nombrarlos de la siguiente manera por columna, El primero como LATITUD, el segundo como LONGITUD y el tercero como DIRECCION.

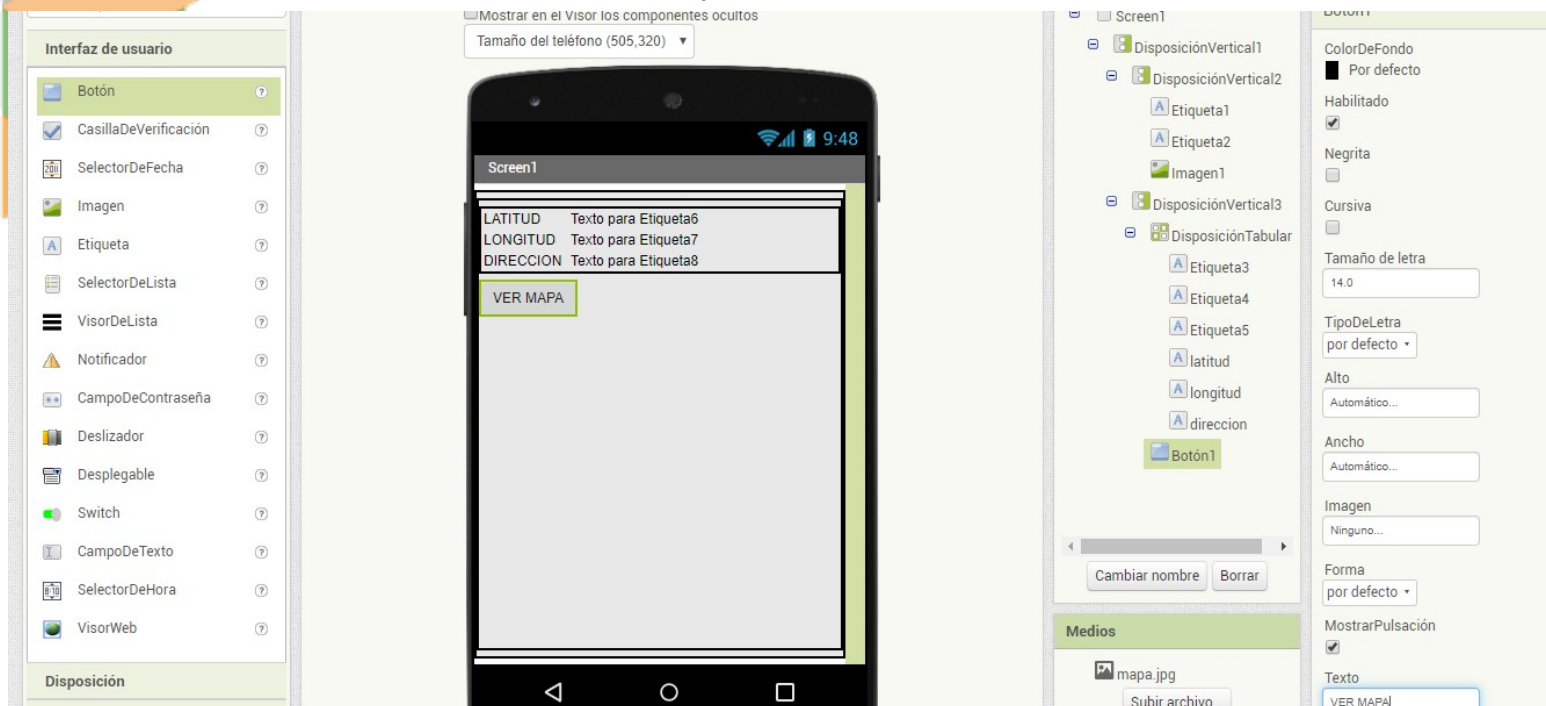
En la otra columna solo es cambiar el nombre de la variable de la siguiente manera.

Seleccionar la etiqueta correspondiente a la latitud y seleccionar la opción cambiar nombre, y renombrar como latitud, dar clic en aceptar.

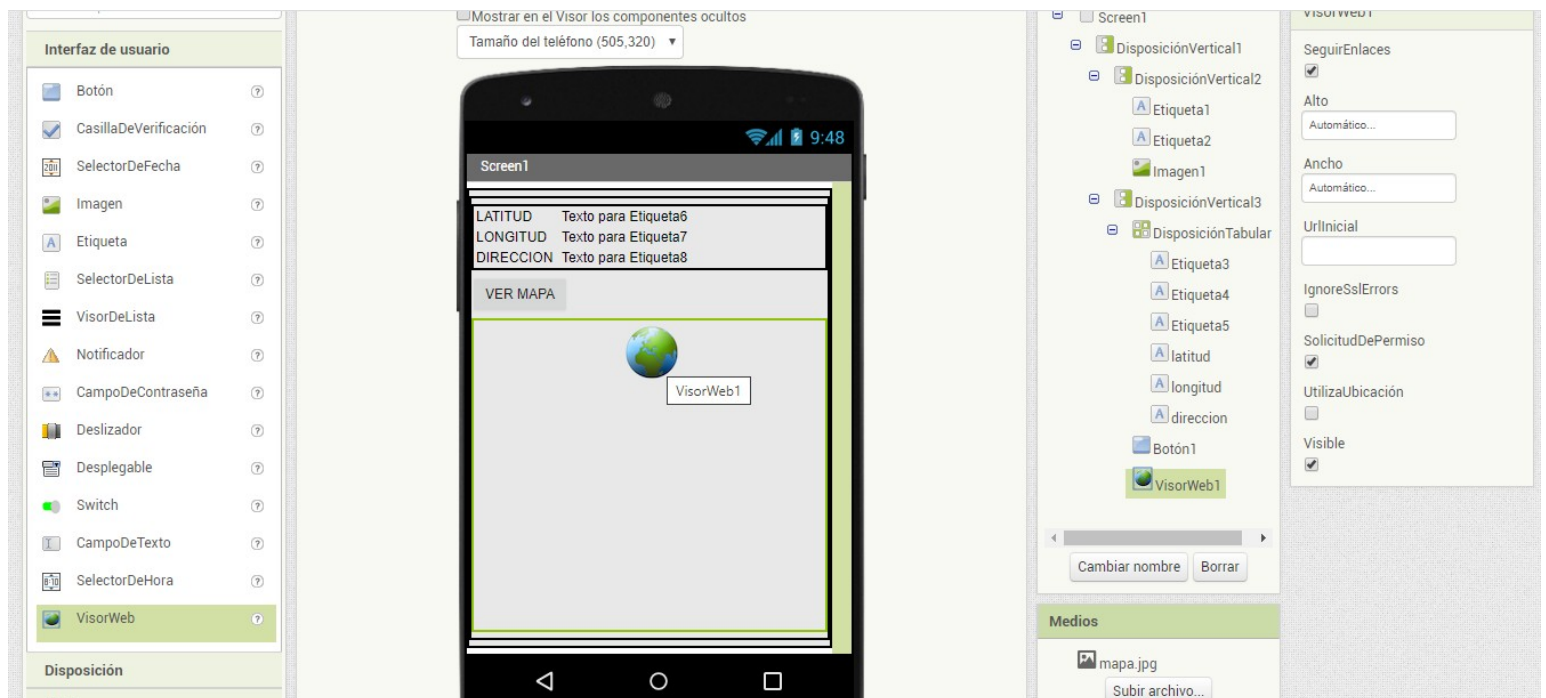
Hacer lo mismo con las otras dos etiquetas, renombrando con los respectivos nombres: longitud y direccion.



9.- Agregar un botón debajo de la disposición tabular, que le cambiaremos de texto a ver mapa.

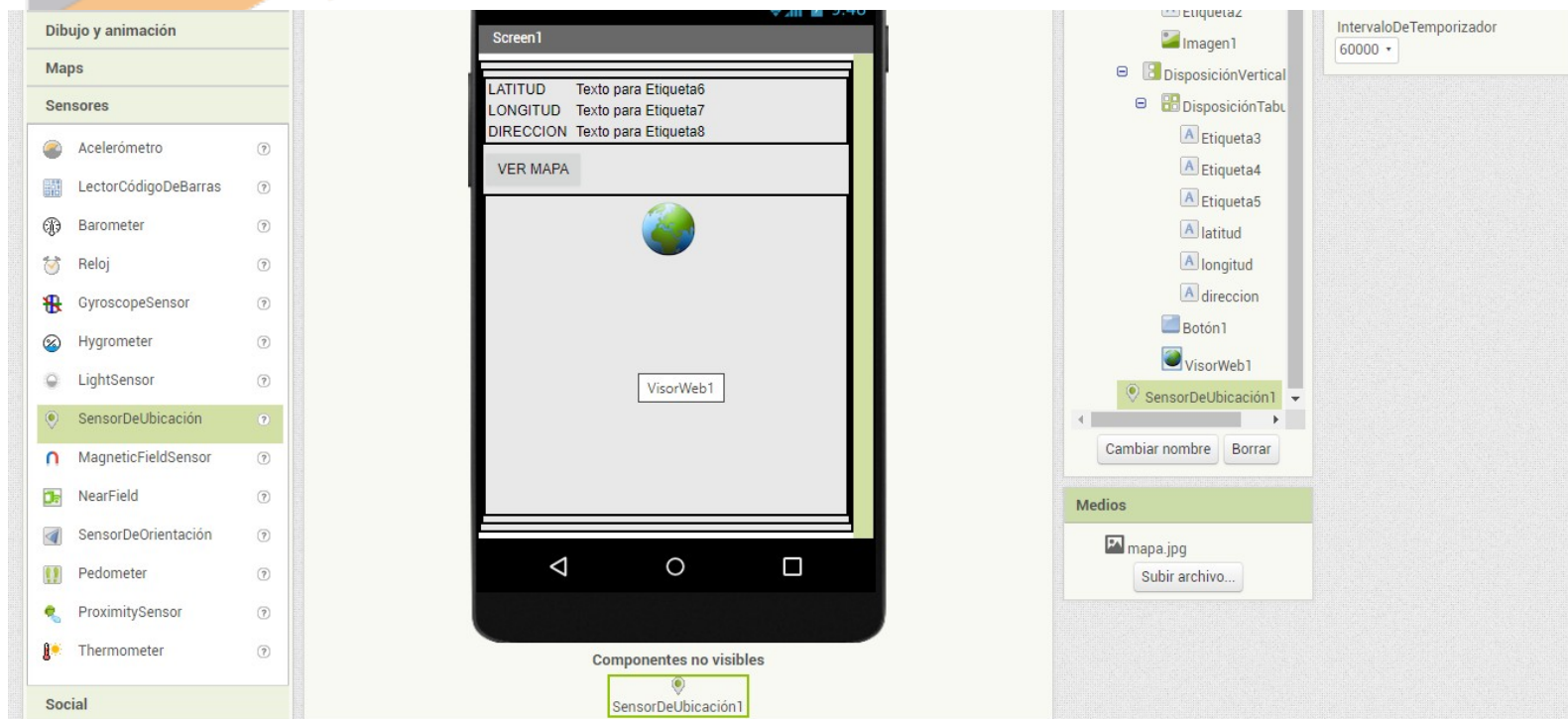


10.- Agregar un Visor web (para poder ver el mapa en tiempo real).

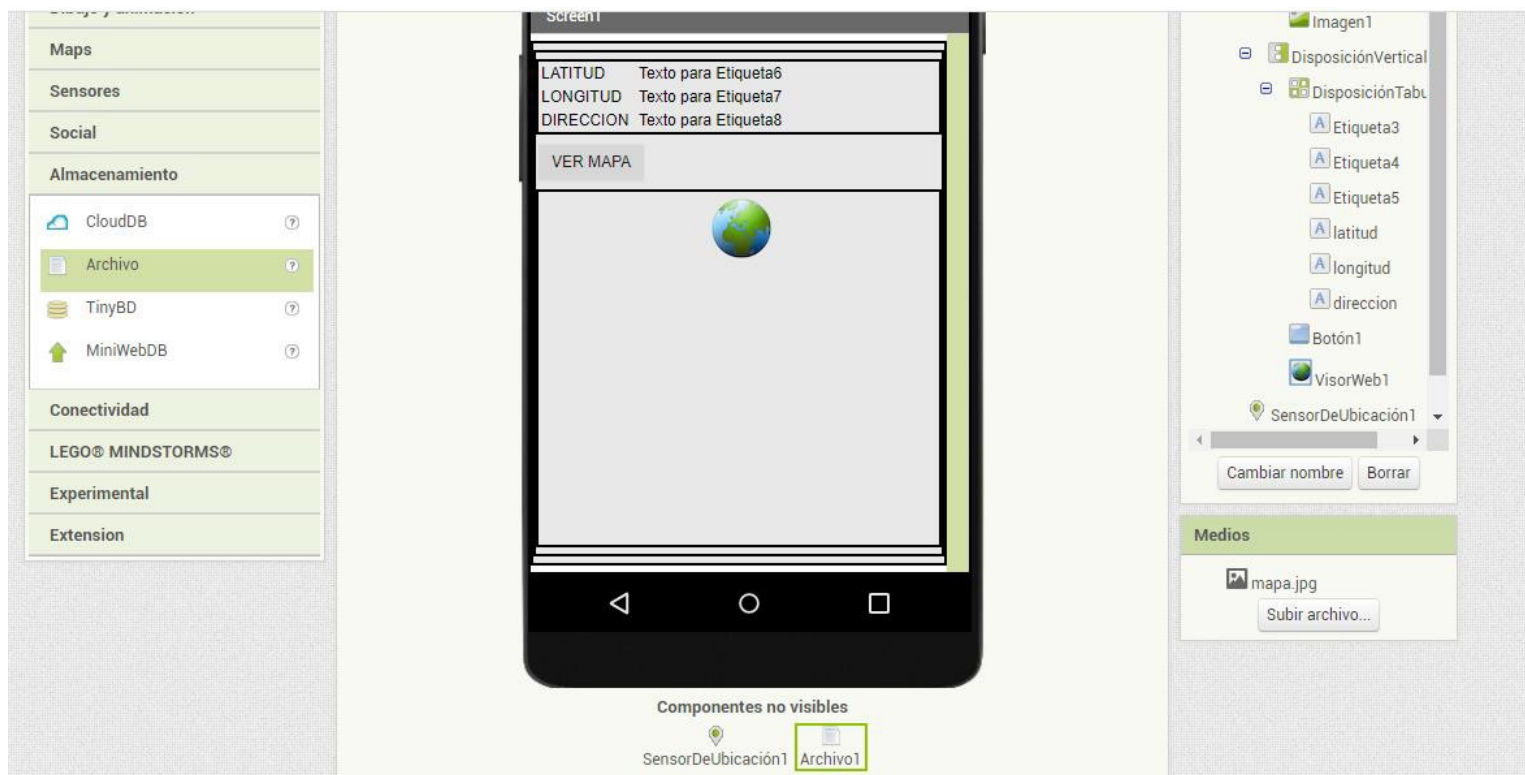




11.- Además vamos a agregar un sensor de ubicación.

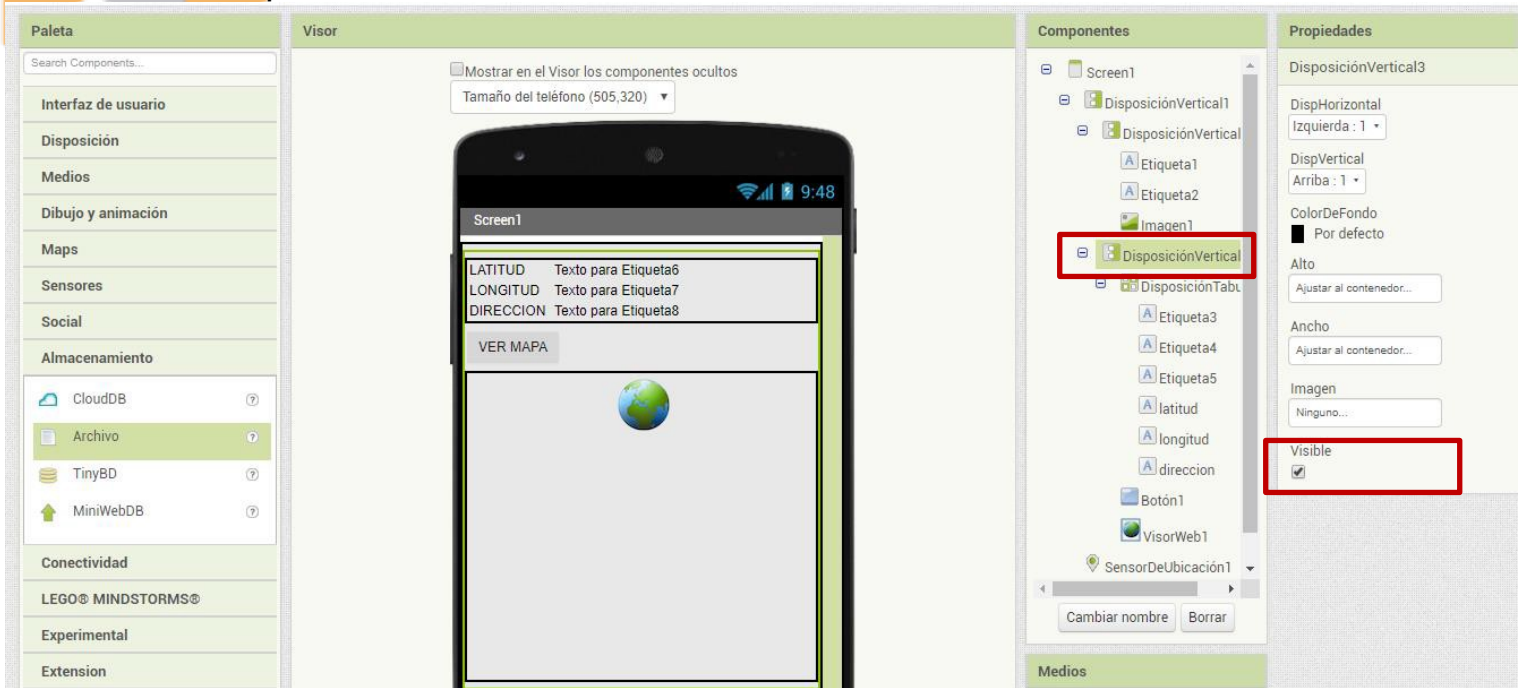


12.- y agregar un archivo de almacenamiento (para generar el documento html y poder mostrar el mapa de google).

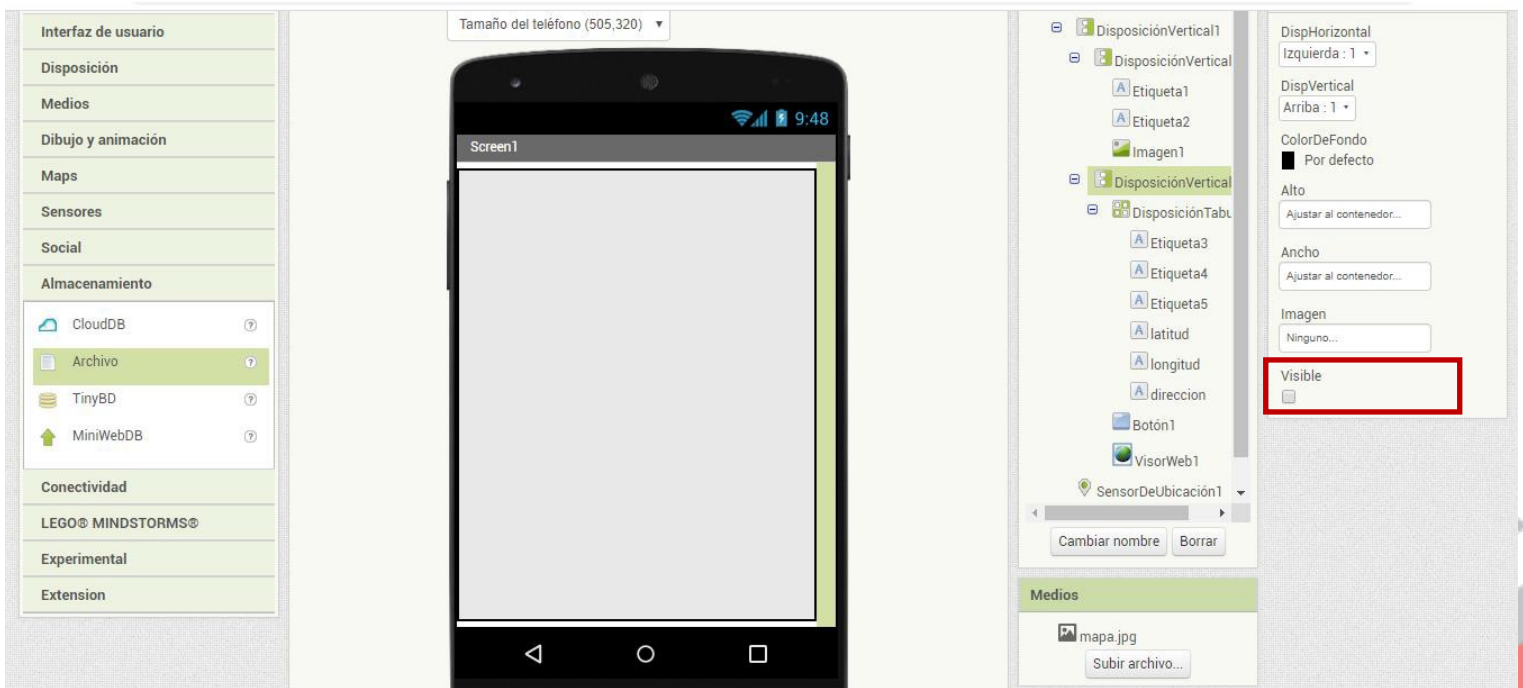


13.- Ahora escondemos la disposición vertical que acabamos de diseñar.

Seleccionamos la disposición que contiene todos los elementos que acabamos de diseñar y quitamos la palomita de visible para esconder la disposición.



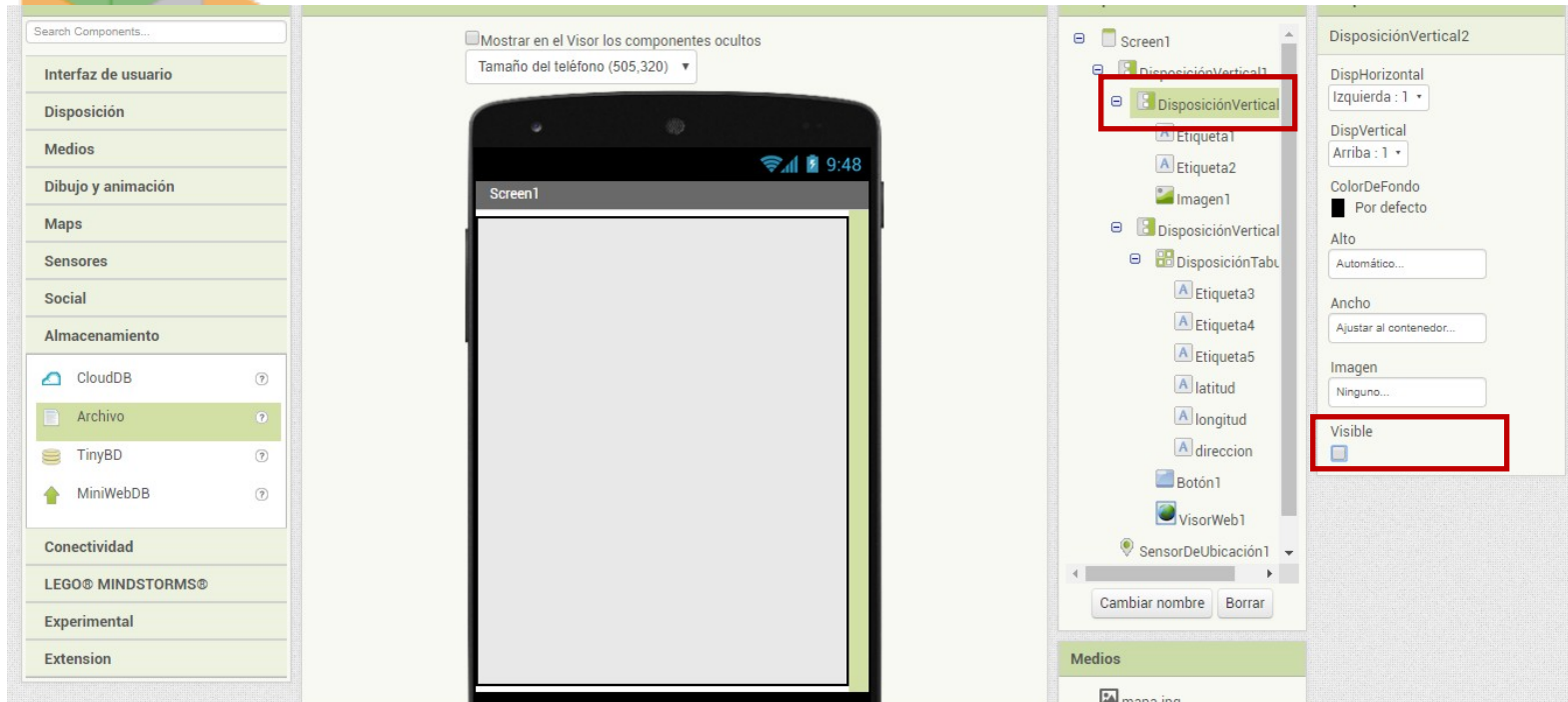
Quedando de la siguiente manera:



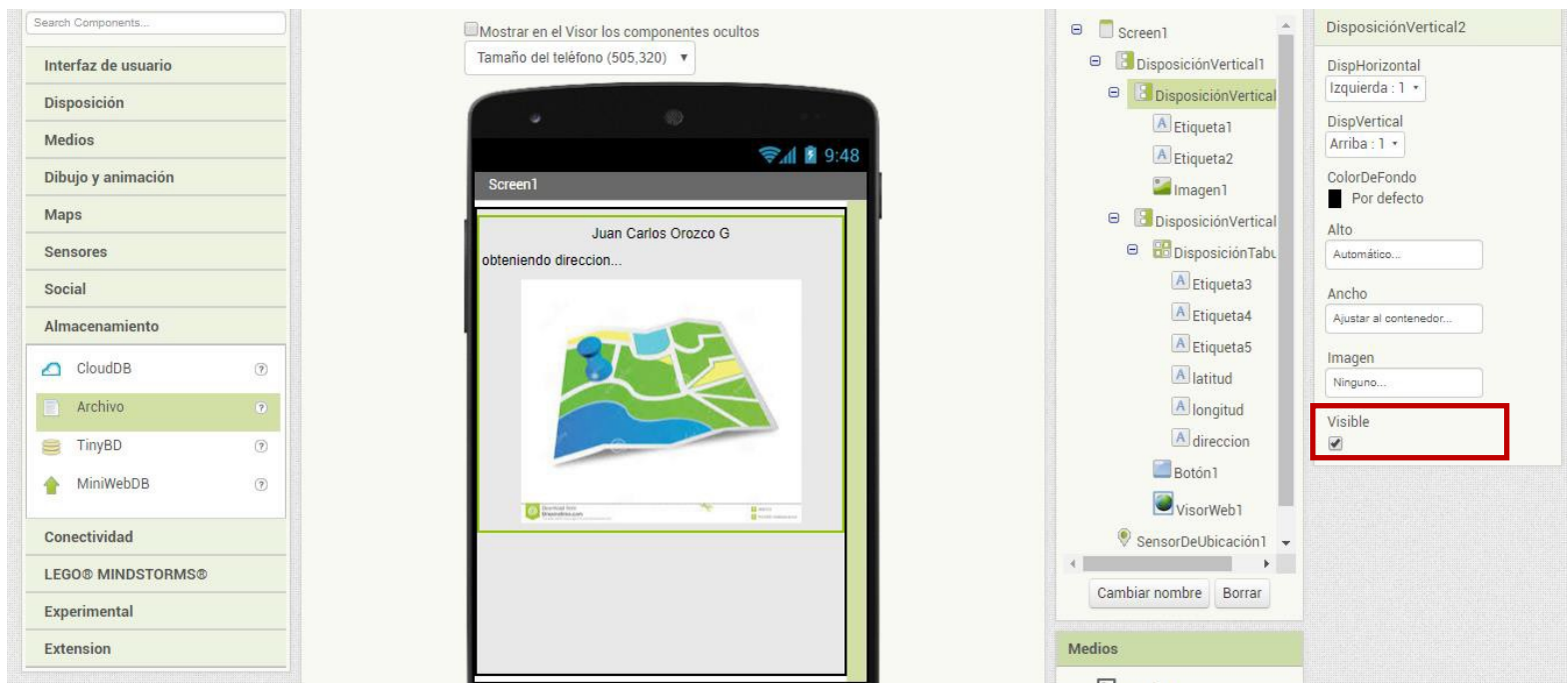


14.- ahora hacemos visible nuevamente la disposición que contiene tu nombre y la imagen del mapa

Seleccionar la disposición que contiene esos elementos y seleccionar la opción visible



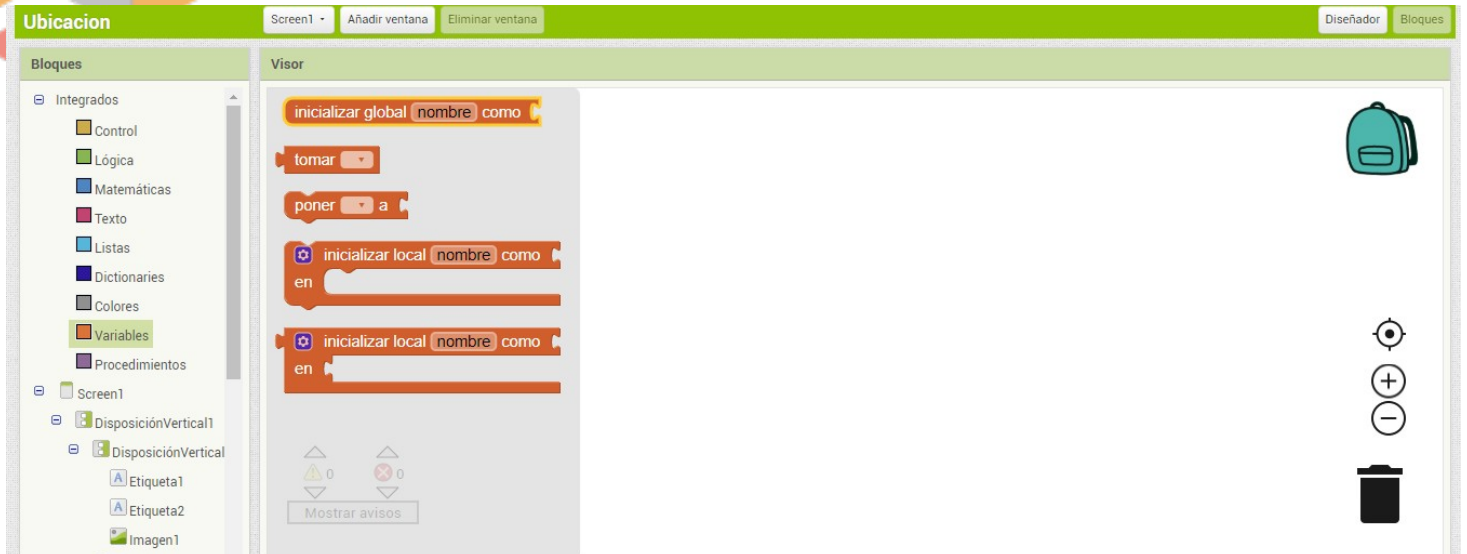
Quedando de la siguiente manera:



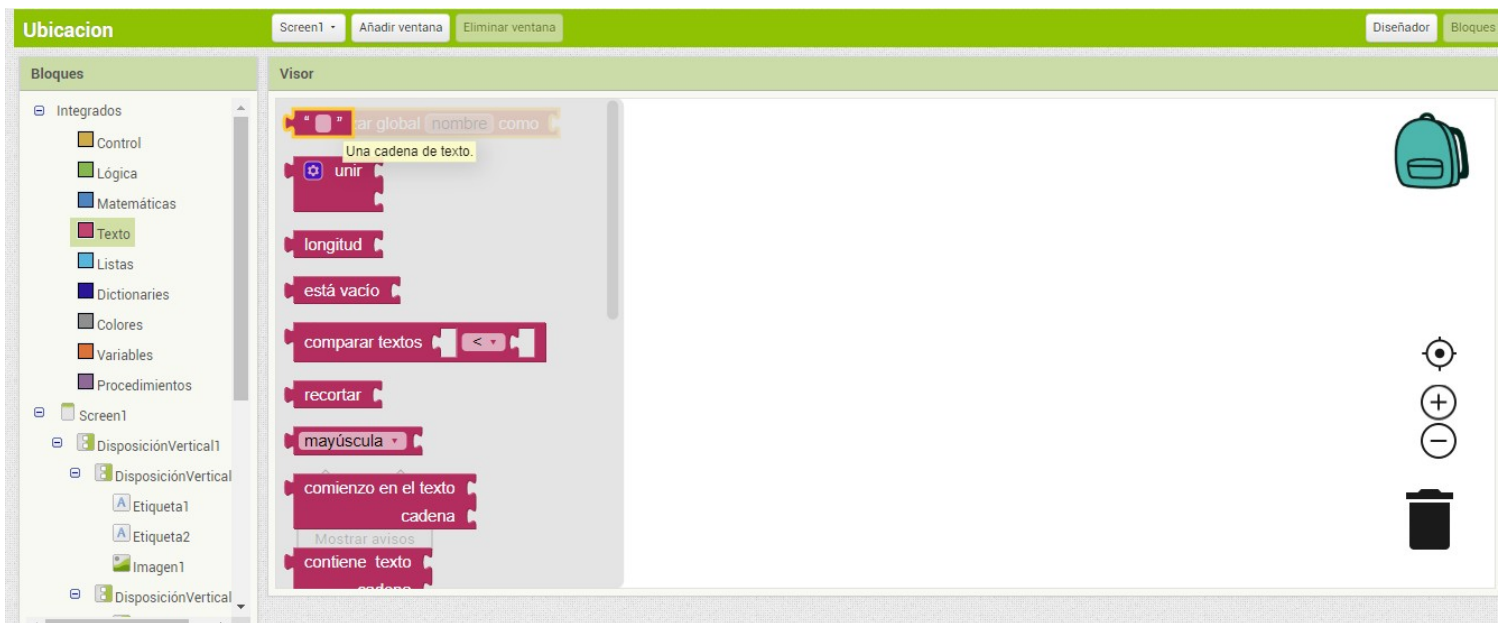
# Codificación en bloques

1.- Crear una variable global para después ser utilizada.

\*Ir al bloque Variable y seleccionar la opción **Inicializar global** nombre como



\*agregar un texto en los integrados text, elegir la primer opción.



\*escribimos en el nombre de la variable `api_key` y en el texto copiamos el key del bloc de notas archivo mapa 1.

```
mapa 1: Bloc de notas
Archivo Edición Formato Ver Ayuda
<div id="map"></div>
<script>
// Initialize and add the map
function initMap() {
  // The location of Uluru
  var uluru = {lat:
20.502887
, lng:
-103.366065};
  // The map, centered at Uluru
  var map = new google.maps.Map(
    document.getElementById('map'), {zoom: 16, center: uluru});
  // The marker, positioned at Uluru
  var marker = new google.maps.Marker({position: uluru, map: map});
}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AlzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4&callback=initMap">
</script>
</body>
</html>
```

Y lo agregamos en el texto quedando de la siguiente manera:

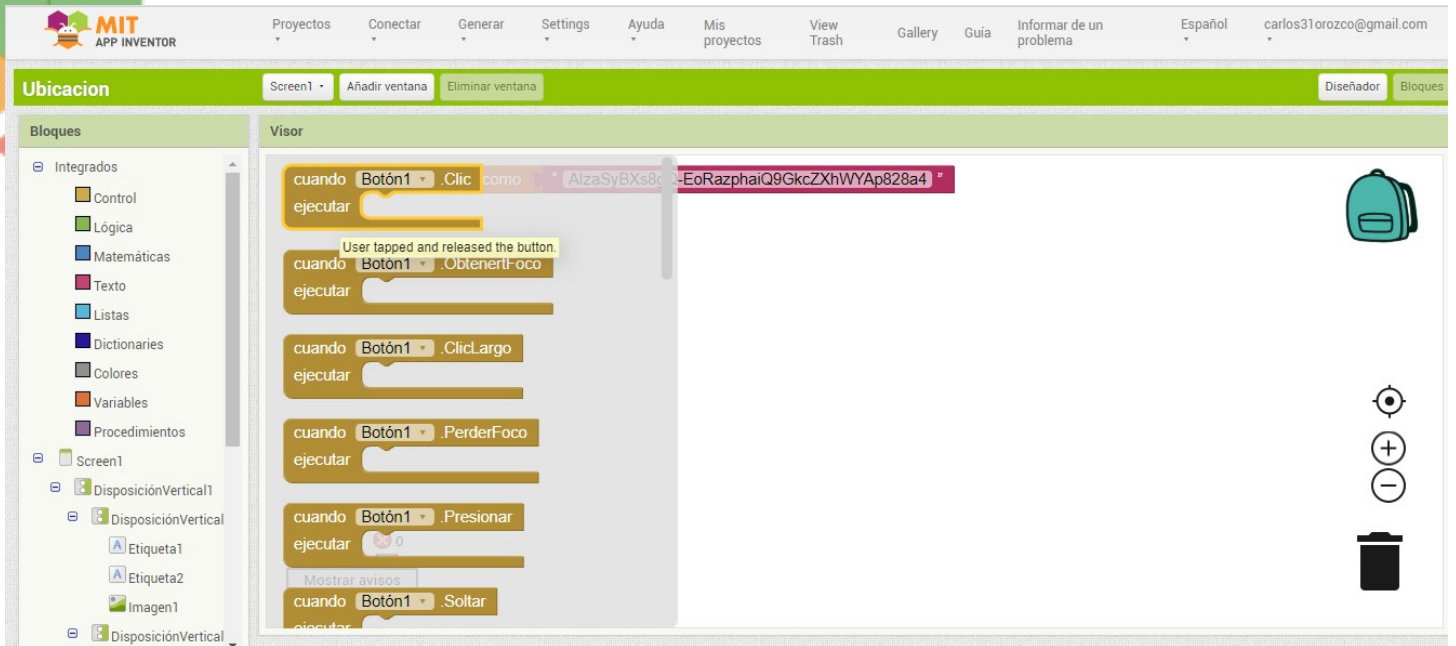
The screenshot shows the Android Studio IDE interface. The top bar is green and contains the title 'Ubicación' and several menu items: 'Screen1', 'Añadir ventana', 'Eliminar ventana', 'Diseñador', and 'Bloques'. The left sidebar is titled 'Bloques' and shows a tree view of components. Under 'Integrados', 'Texto' is highlighted. The main workspace is titled 'Visor' and shows a text field with the following content: 'inicializar global api\_key como 'AlzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4''. The text field is highlighted in pink. Below the text field, there are several icons: a backpack, a location pin, a plus sign, a minus sign, and a trash can. At the bottom left of the workspace, there are two warning icons (a yellow triangle and a red circle with an X) and a button labeled 'Mostrar avisos'.



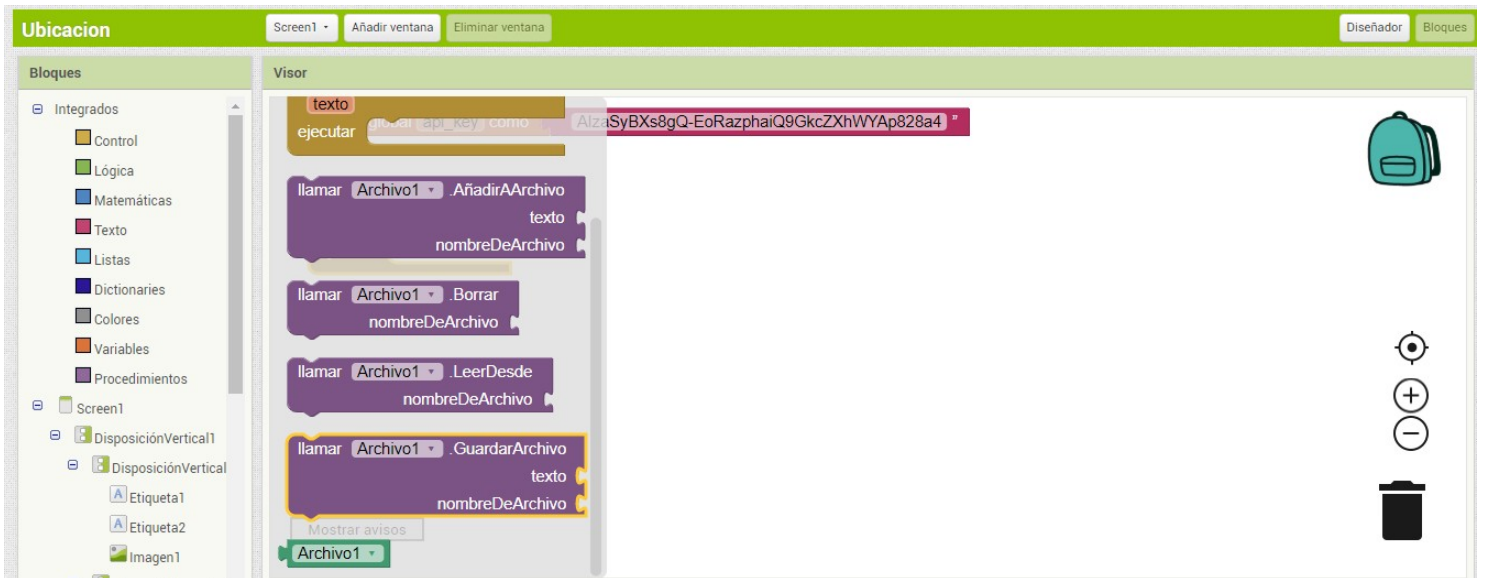


2.- Cuando demos clic al botón ver mapa nos va a mostrar el mapa según la ubicación en la que nos encontremos.

\*Seleccionamos el botón, y la opción **cuando botón .clic hacer**



\*Seleccionamos el elemento archivo y la opción **llamar Archivo .GuardarArchivo texto nombreDeArchivo**



\*Seleccionamos de integrados Texto y seleccionamos **Unir**

The screenshot shows the Scratch IDE interface. On the left, the 'Bloques' (Blocks) palette is open to the 'Integrados' (Integrated) section, specifically the 'Texto' (Text) category. The 'unir' (Join) block is highlighted. The main workspace, labeled 'Visor', shows a script starting with 'inicializar global api\_key como' followed by a text block containing an API key. Below this, the 'unir' block is being added to the script. A tooltip for the 'unir' block is visible, stating: 'Concatena todas las entradas para formar una única cadena de texto. Si no hay entradas, crea un texto en blanco.' Other blocks like 'longitud', 'está vacío', 'comparar textos', 'recortar', 'mayúscula', 'comienzo en el texto', and 'contiene texto' are also visible in the workspace.

\*Agregamos 2 cadenas más. Dando clic en el engrane y arrastrar la opción cadena al espacio.

This screenshot shows the same Scratch IDE workspace as the previous one. The 'unir' block now has two 'cadena' (string) options added to its input slots. A mouse cursor is shown clicking on the gear icon of the 'unir' block, and a small menu is open showing the 'cadena' option being dragged into the input slots. The workspace also shows a 'cuando Botón1 .Clic' event block, an 'ejecutar' block, and a 'llamar Archivo1 .GuardarArchivo' block. The 'Mostrar avisos' (Show warnings) button at the bottom left indicates 2 warnings and 0 errors.



\*agregar de integrados un texto en blanco.

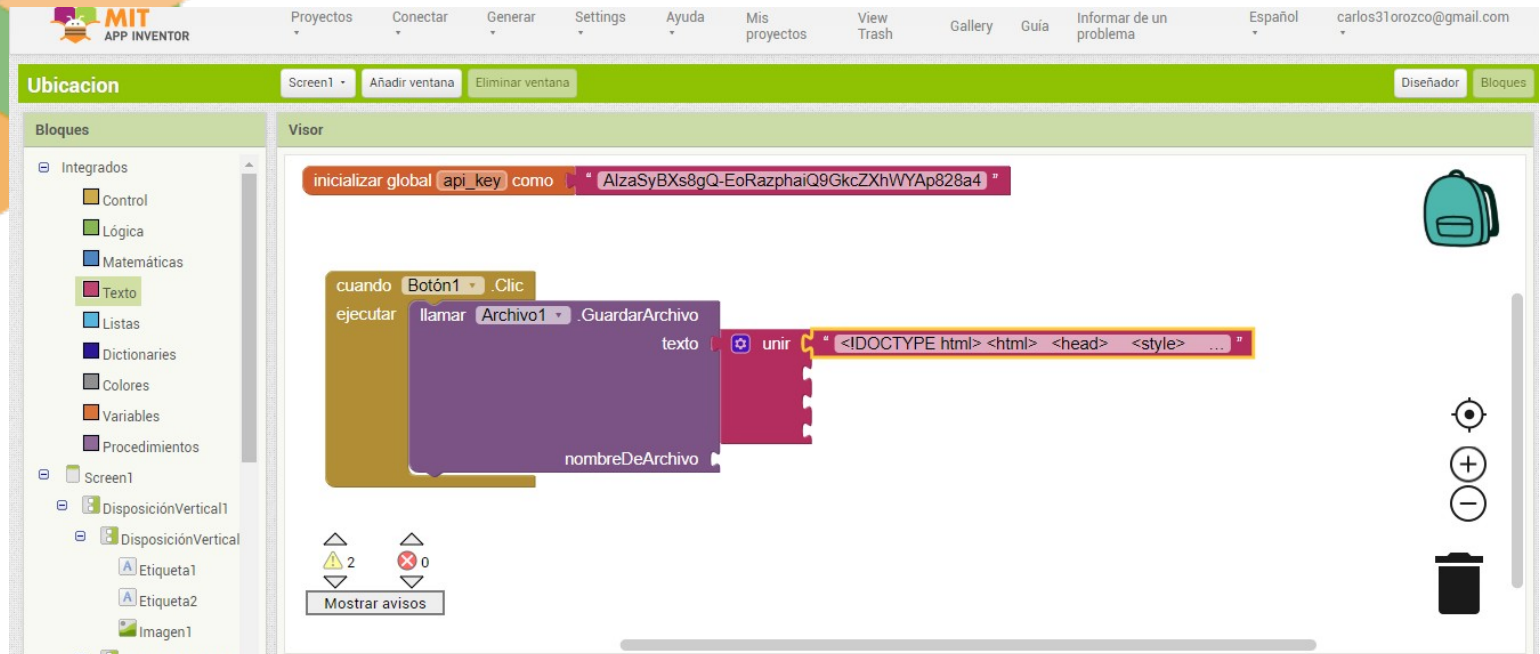
The screenshot shows the MIT App Inventor web interface. The 'Bloques' (Blocks) palette on the left is open to the 'Text' category. In the 'Visor' (Viewer) area, a code block is visible: 'cuando Botón1 .Clic' (when Button1 clicked) triggers 'ejecutar llamar Archivo1 .GuardarArchivo' (execute call file1 save file). The 'texto' (text) block is set to 'unir' (concatenate) and contains the string 'Una cadena de texto.' (A text string). The 'nombreDeArchivo' (filename) block is set to 'AlzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4', which is the API key from the 'inicializar global' (initialize global) block above it.

\*Copiar del bloc de notas (mapa 1) el siguiente texto

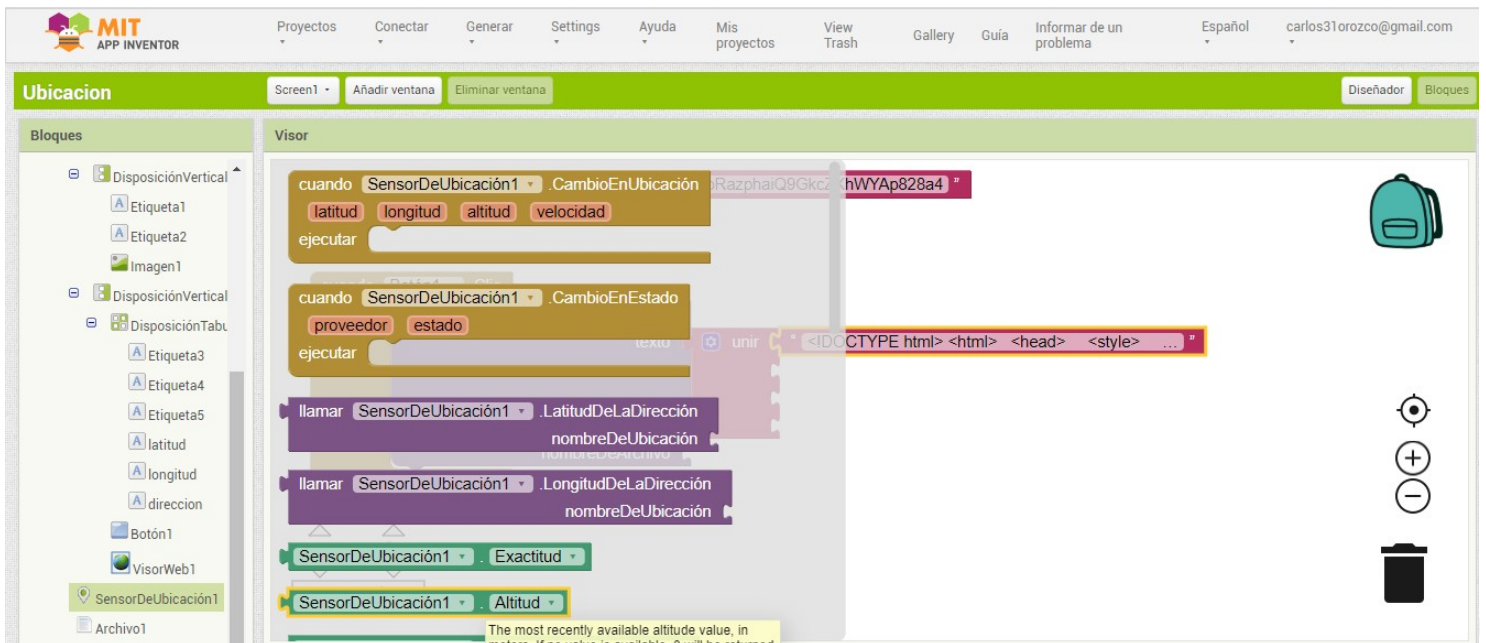
```
mapa 1: Bloc de notas
Archivo Edición Formato Ver Ayuda
<!DOCTYPE html>
<html>
  <head>
    <style>
      /* Set the size of the div element that contains the map */
      #map {
        height: 400px; /* The height is 400 pixels */
        width: 100%; /* The width is the width of the web page */
      }
    </style>
  </head>
  <body>
    <h3>Mi mapa</h3>
    <!--The div element for the map -->
    <div id="map"></div>
    <script>
      // Initialize and add the map
      function initMap() {
        // The location of Uluru
        var uluru = {lat:
20.502887
, lng:
```



\*pegarlo en el bloque en texto en blanco, quedando de la siguiente manera:



\*Debemos agregar en la unión la latitud de nuestro dispositivo móvil, para ello debemos seleccionar el elemento **SensorDeUbicación** y seleccionar la opción **SensordeUbicación1 .Altitud**



## \*Cambiamos de Altitud a Latitud

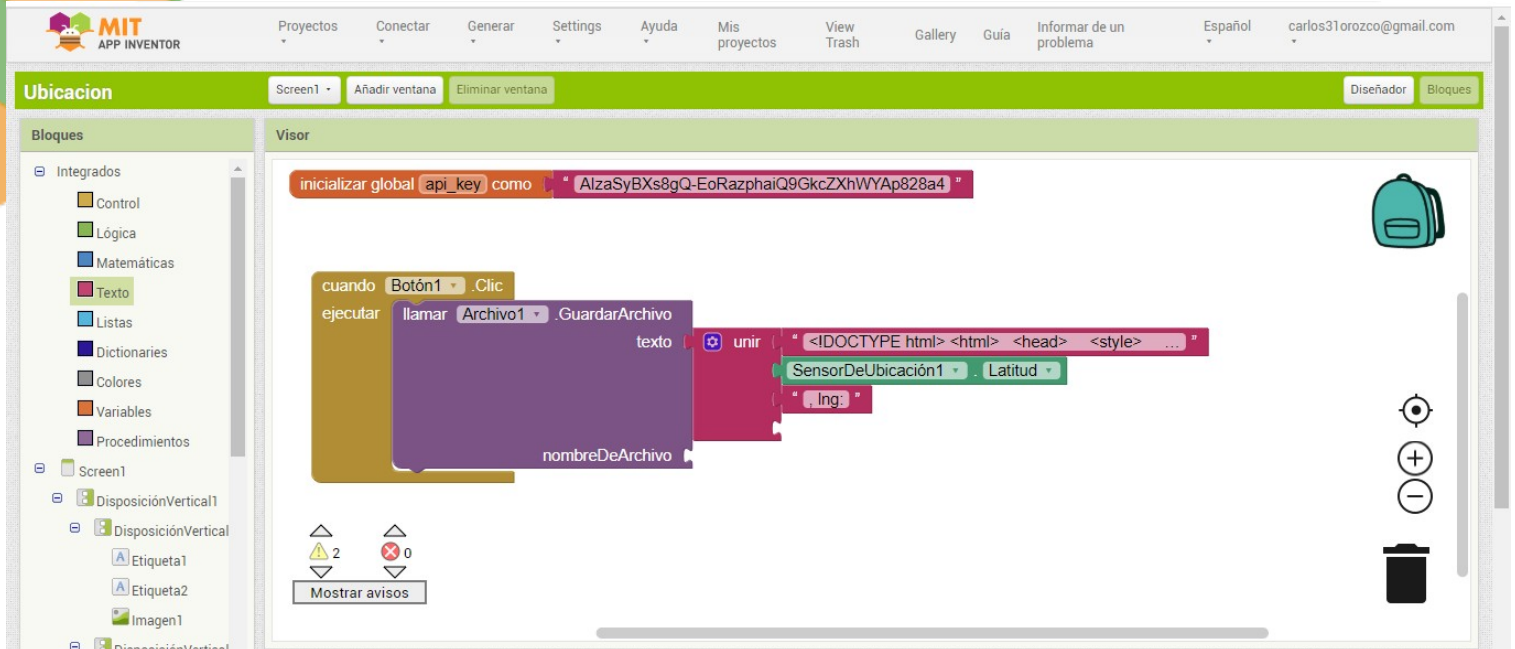
The screenshot shows the MIT App Inventor interface. In the center, a code block for a button click event is visible. It contains the following code: `inicializar global api_key como "AlzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4"`, `llamar Archivo1 .GuardarArchivo texto`, and `unir "<IDCTYPE html> <div id='map'></div> </script> // Initialize and add the map function initMap() { // The location of Uluru var uluru = {lat: 20.502887, lng: -103.366065}; // The map, centered at Uluru var map = new google.maps.Map(document.getElementById('map'), {zoom: 16, center: uluru}); // The marker, positioned at Uluru var marker = new google.maps.Marker({position: uluru, map: map}); } </script> <script async defer src='https://maps.googleapis.com/maps/api/js?key=AIzaSvBXs8g0-EoRazphai09GkcZXhWYAp828a4&callback=initMap'></script>`. A dropdown menu is open over the `Altitud` property of the `SensorDeUbicación1` block, with `Latitud` selected. The interface also shows a `Bloques` panel on the left and a `Visor` panel on the right.

\*Agregamos otro texto en blanco a la unión y copiamos el siguiente texto del bloc de notas (mapa 1).

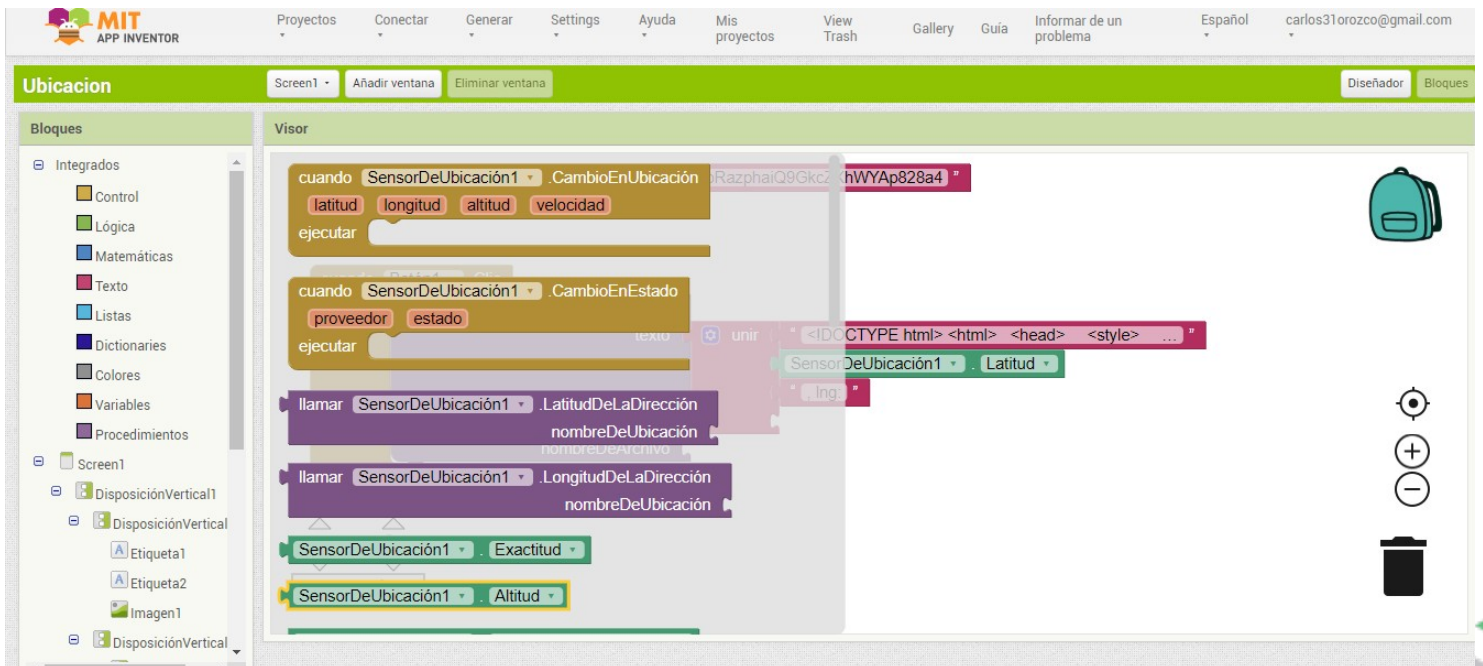
```
mapa 1: Bloc de notas
Archivo Edición Formato Ver Ayuda
</head>
<body>
  <h3>Mi mapa</h3>
  <!--The div element for the map -->
  <div id="map"></div>
  <script>
// Initialize and add the map
function initMap() {
  // The location of Uluru
  var uluru = {lat:
20.502887
, lng:
-103.366065};
  // The map, centered at Uluru
  var map = new google.maps.Map(
    document.getElementById('map'), {zoom: 16, center: uluru});
  // The marker, positioned at Uluru
  var marker = new google.maps.Marker({position: uluru, map: map});
}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSvBXs8g0-EoRazphai09GkcZXhWYAp828a4&callback=initMap">
```



\*lo pegamos en el texto en blanco, quedando de la siguiente manera:

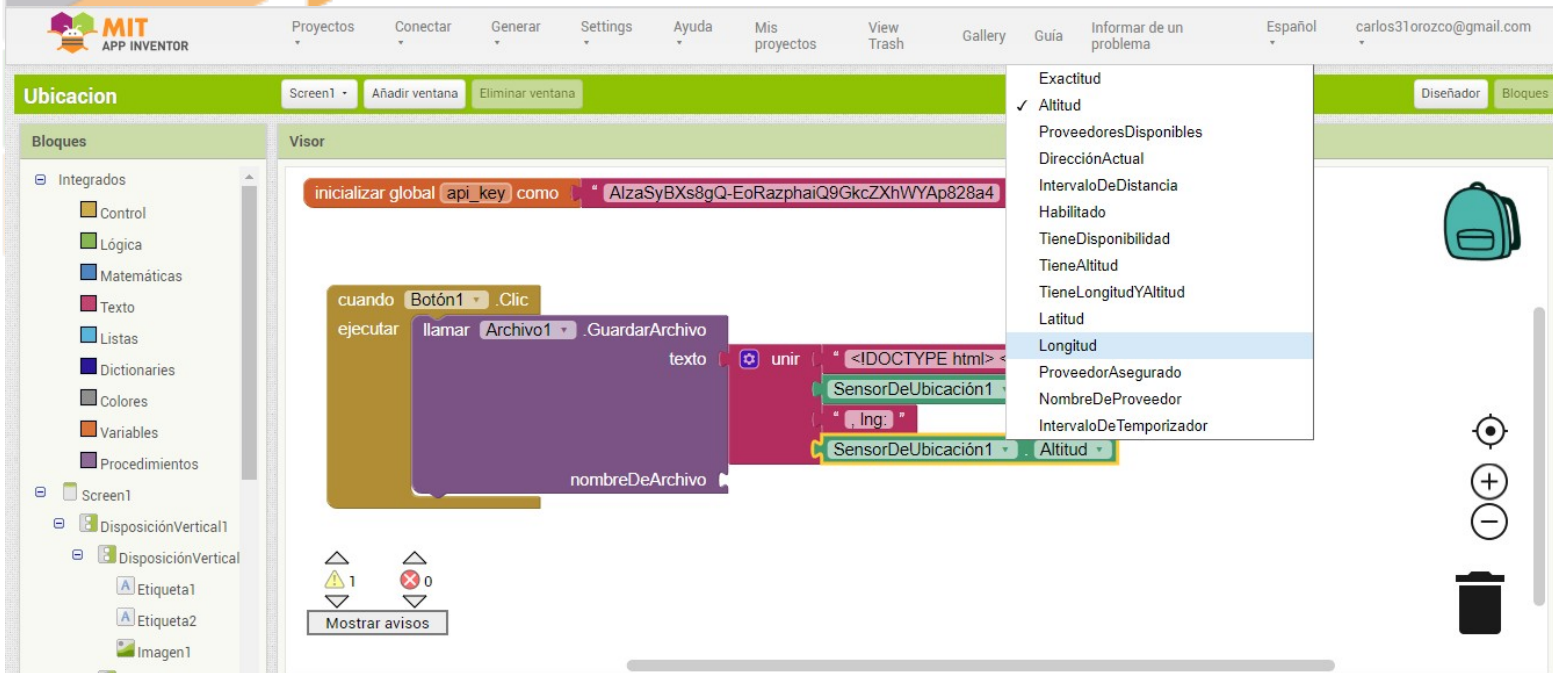


\*Debemos agregar en la unión la longitud de nuestro dispositivo móvil, para ello debemos seleccionar el elemento `SensorDeUbicación1` y seleccionar la opción `SensordeUbicación1 .Altitud`



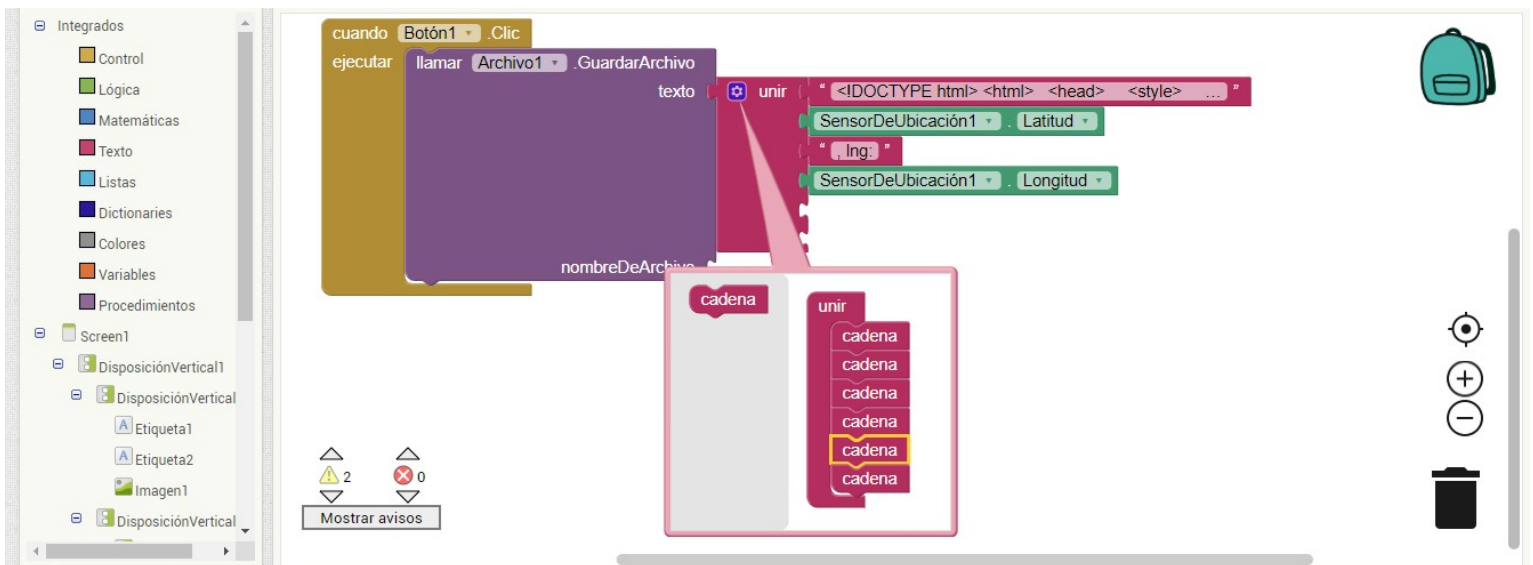


## \*Cambiar altitud por longitud



The screenshot shows the MIT App Inventor interface. The top menu bar includes 'Proyectos', 'Conectar', 'Generar', 'Settings', 'Ayuda', 'Mis proyectos', 'View Trash', 'Gallery', 'Guía', 'Informar de un problema', 'Español', and 'carlos31orozco@gmail.com'. The main workspace is titled 'Ubicacion' and shows a code block for 'Botón1.Clic' that calls 'Archivo1.GuardarArchivo' with a text block containing a 'unir' block. The 'unir' block currently contains three items: '<DOCTYPE html><...', 'SensorDeUbicación1.Lng:', and 'SensorDeUbicación1.Altitud'. A dropdown menu is open over the 'unir' block, listing various location-related properties. 'Longitud' is selected and highlighted in blue.

\*agregar otras dos cadenas a la unión, seleccionando el engrane y arrastra la cadena al final de la unión quedando de la siguiente manera:



The screenshot shows the same MIT App Inventor interface as the previous one. The 'unir' block now contains four items: '<DOCTYPE html><html><head><style>...', 'SensorDeUbicación1.Latitud', ', lng:', and 'SensorDeUbicación1.Longitud'. A callout box is shown over the 'unir' block, illustrating how to add 'cadena' (string) blocks. The callout shows a 'cadena' block being dragged into the 'unir' block, which now contains five 'cadena' blocks in a vertical stack.



\*agregamos otro texto en blanco y copiamos el siguiente texto del bloc de notas (mapa 1)

\*mapa 1: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
<div id="map"></div>
<script>
// Initialize and add the map
function initMap() {
// The location of Uluru
var uluru = {lat:
20.502887
, lng:
-103.366065
};
// The map, centered at Uluru
var map = new google.maps.Map(
document.getElementById('map'), {zoom: 16, center: uluru});
// The marker, positioned at Uluru
var marker = new google.maps.Marker({position: uluru, map: map});
}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=
```

AIzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4

\*pegarlo en el texto en blanco, quedando de la siguiente manera:

The screenshot shows a block editor interface with a 'Bloques' sidebar on the left and a 'Visor' workspace on the right. The workspace contains a script starting with 'cuando Botón1 .Clic' and 'ejecutar'. The main block is 'llamar Archivo GuardarArchivo' with a 'nombreDeArchivo' field. This is followed by a 'unir' block containing the following code:

```
<!DOCTYPE html> <html> <head> <style> ...
SensorDeUbicación1 . Latitud
, lng:
SensorDeUbicación1 . Longitud
}; // The map, centered at Uluru var map = n...
```



\*Es momento de utilizar la variable `ap_key` agregada al inicio. Para ellos vamos a integrados variables y seleccionamos la opción **tomar**

The screenshot shows the Scratch IDE interface. On the left, the 'Bloques' (Blocks) palette is open to 'Integrados' (Integrated) > 'Variables'. The 'tomar' block is highlighted. The main workspace shows a script for a button click event. The script consists of the following blocks: 'ejecutar llamar Archivo1 GuardarArchivo', 'tomar' (highlighted), 'poner a', 'inicializar local nombre como', 'en', 'inicializar local nombre como', 'en', and 'unir'. The 'unir' block contains HTML code: '<!DOCTYPE html> <html> <head> <style> ...'. The 'poner a' block is connected to the 'tomar' block. The 'inicializar local nombre como' blocks are connected to the 'poner a' block. The 'unir' block is connected to the 'ejecutar' block.

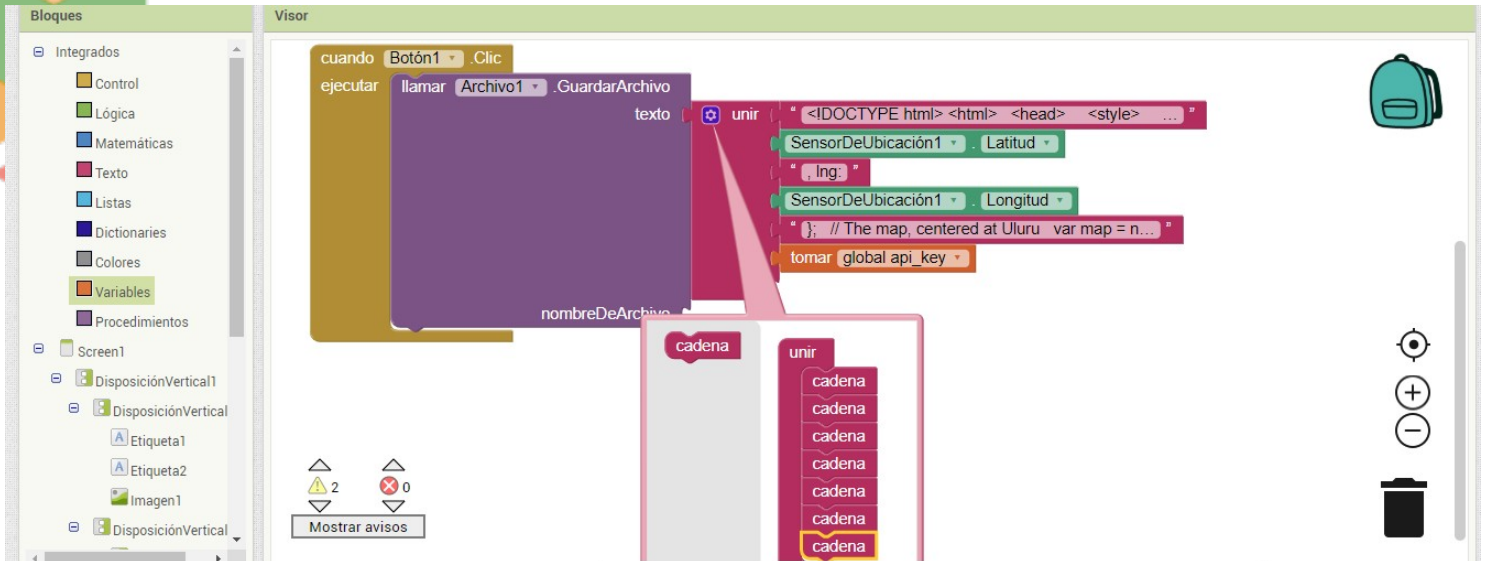
\*seleccionamos la variable que agregamos

The screenshot shows the same Scratch IDE interface as the previous one. The 'tomar' block is now connected to the 'poner a' block and is selecting the 'global api\_key' variable from the 'Variables' palette. The rest of the script remains the same.





\*agregamos otro campo a la unión, seleccionando el engrane y arrastrando una cadena al final, quedando de la siguiente manera:



\*agregamos un texto en blanco y copiamos el siguiente código del bloc de notas (mapa 1)

```

*mapa 1: Bloc de notas
Archivo Edición Formato Ver Ayuda
var uluru = {lat:
20.502887
, lng:
-103.366065
};
// The map, centered at Uluru
var map = new google.maps.Map(
  document.getElementById('map'), {zoom: 16, center: uluru});
// The marker, positioned at Uluru
var marker = new google.maps.Marker({position: uluru, map: map});
}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=
AIzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4
&callback=initMap">
</script>
</body>
</html>

```



\*y lo pegamos en el texto en blanco, quedando de la siguiente manera:

The screenshot shows the Scratch IDE interface. On the left is the 'Bloques' (Blocks) palette with categories like 'Integrados', 'Control', 'Lógica', 'Matemáticas', 'Texto', 'Listas', 'Dictionaries', 'Colores', 'Variables', and 'Procedimientos'. The 'Screen1' workspace contains a 'DisposiciónVertical1' group with 'Etiqueta1', 'Etiqueta2', and 'Imagen1'. The main 'Visor' (Stage) area shows a script for 'Botón1 .Clic' with the following blocks:

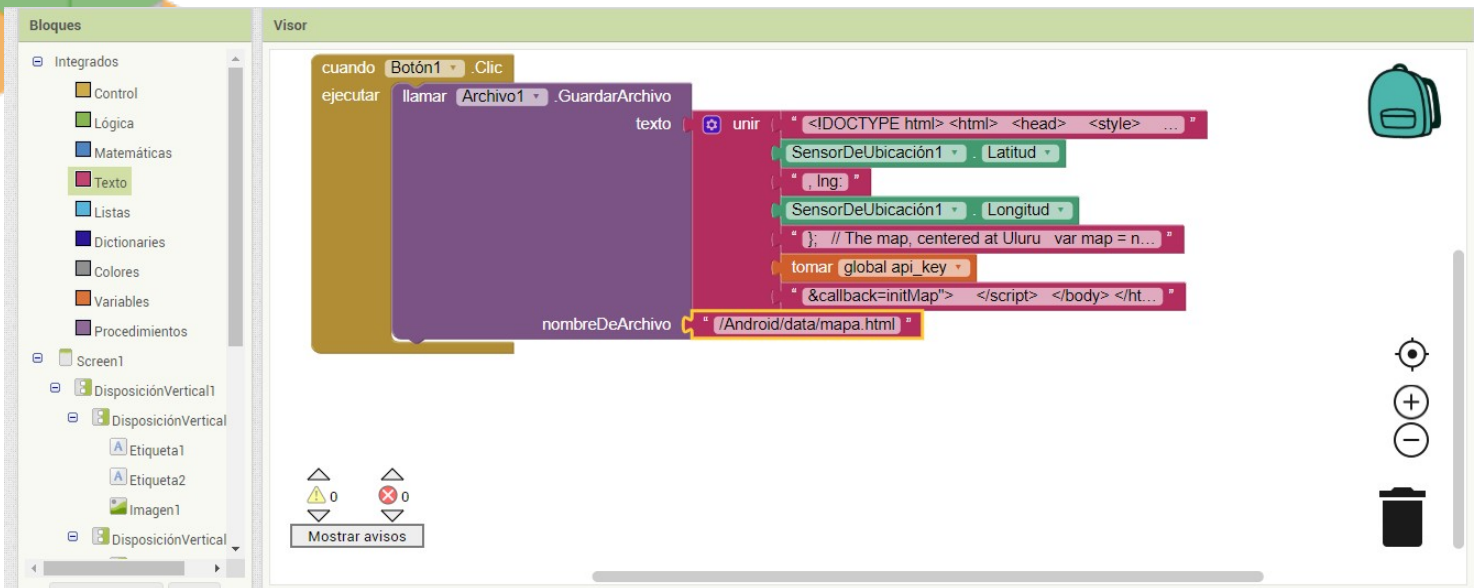
- 'ejecutar' block
- 'llamar Archivo1 .GuardarArchivo' block with a 'texto' field and a 'nombreDeArchivo' field.
- 'unir' block containing:
  - '<!DOCTYPE html> <html> <head> <style> ...'
  - 'SensorDeUbicación1 . Latitud'
  - ', lng:'
  - 'SensorDeUbicación1 . Longitud'
  - '}; // The map, centered at Uluru var map = n...'
  - 'tomar global api\_key'
  - '&callback=initMap"> </script> </body> </ht...'

\*agregar un texto en blanco para guardar el archivo html que estamos creando.

This screenshot is identical to the one above, but the 'unir' block now includes an additional empty text block at the end of its content, represented by a small pink box with a white space inside. The rest of the script and the IDE interface remain the same.

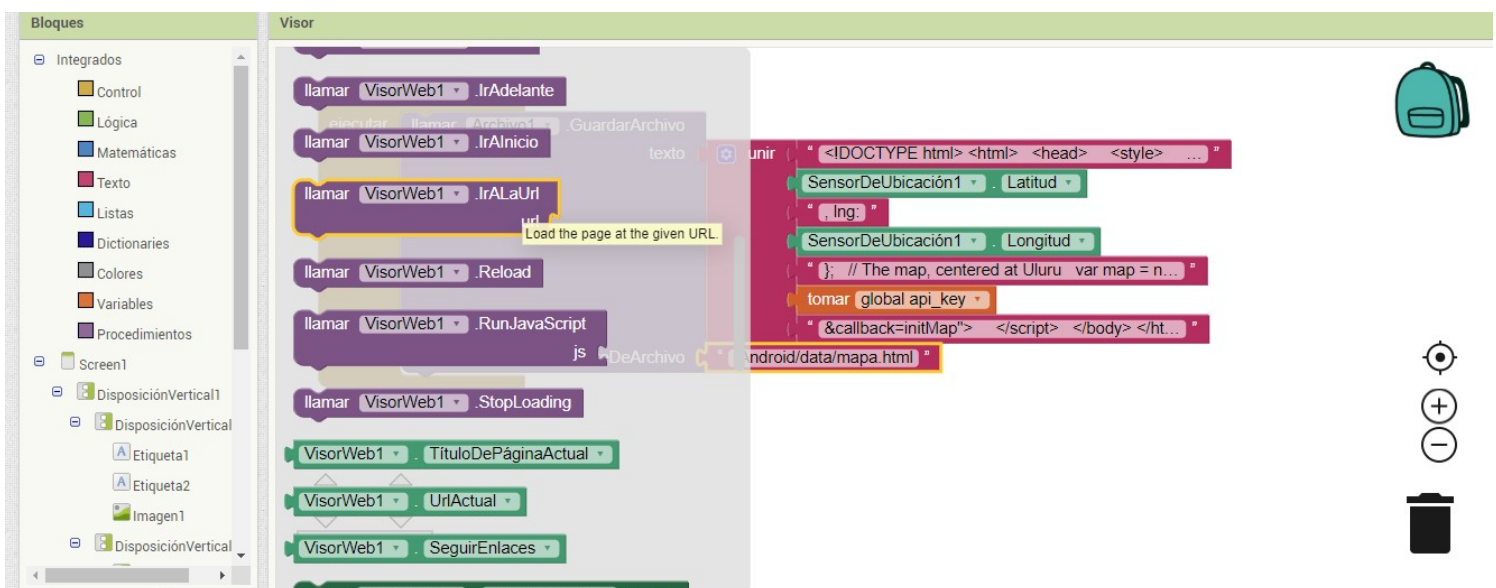


\*el nombre del archivo lo vamos a guardar en nuestro dispositivo móvil, en la carpeta **Android** y carpeta **data** y con el nombre del archivo llamado **mapa.html**  
Quedando de la siguiente manera:

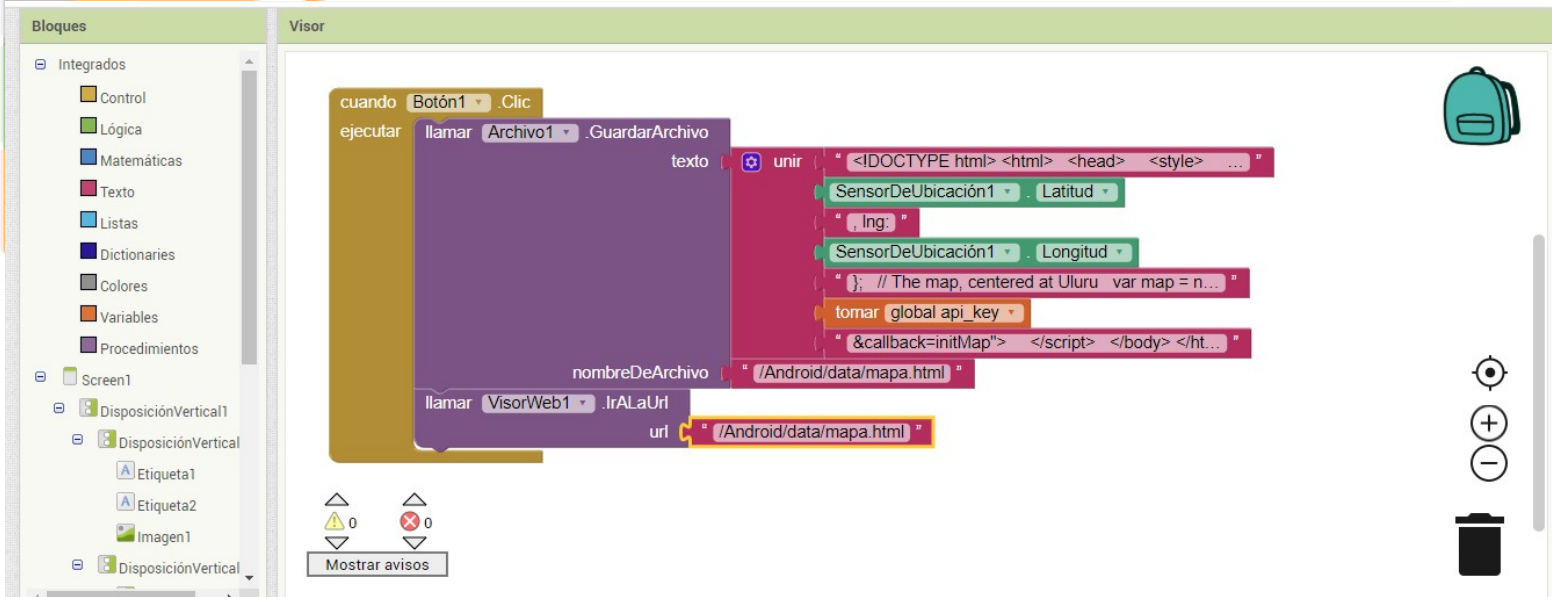


3.-Es necesario mostrar el mapa que se acaba de guardar en el archivo mapa.html en nuestro dispositivo móvil.

\*Mostrar en el visor web, para ello seleccionamos el elemento **VisorWeb** y seleccionamos la opción llamar **VisorWeb1 .IrALaUrl**



\*copiamos la dirección en donde guardamos el archivo en el celular.

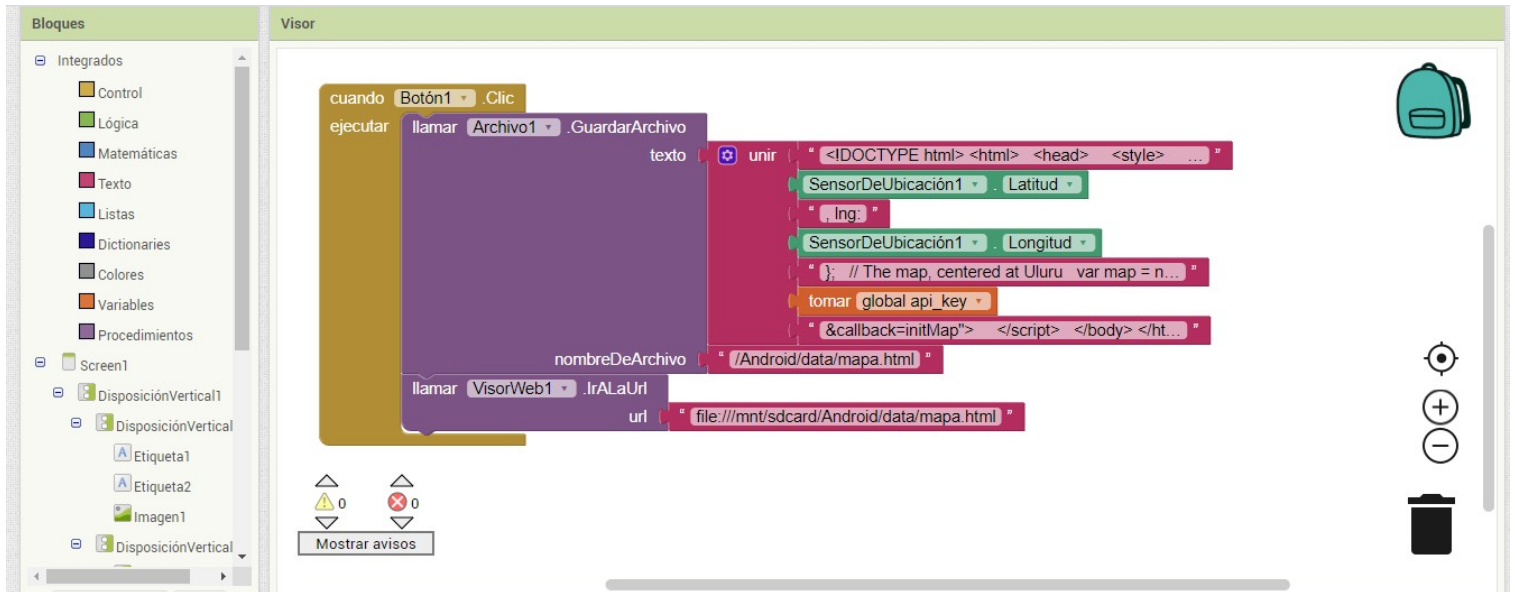


The screenshot shows the Scratch IDE interface. On the left, the 'Bloques' (Blocks) palette is visible, with 'Integrados' (Integrated) and 'Screen1' categories. The main workspace shows a script for a button click event. The script consists of the following blocks:

- cuando Botón1 .Clic
- ejecutar
- llamar Archivo1 .GuardarArchivo
- texto unir " <IDOCTYPE html> <html> <head> <style> ... "
- SensorDeUbicación1 . Latitud
- , lng: "
- SensorDeUbicación1 . Longitud
- } ; // The map, centered at Uluru var map = n...
- tomar global api\_key
- &callback=initMap"> </script> </body> </ht...
- nombreDeArchivo " /Android/data/mapa.html "
- llamar VisorWeb1 .IrALaUrl
- url " /Android/data/mapa.html "

At the bottom of the workspace, there are navigation controls (up, down, left, right arrows) and a 'Mostrar avisos' (Show warnings) button.

\*agregamos al inicio de la dirección **file:///mnt/sdcard/** de la siguiente manera:



The screenshot shows the Scratch IDE interface, similar to the previous one. The script is updated to include the file path:

- cuando Botón1 .Clic
- ejecutar
- llamar Archivo1 .GuardarArchivo
- texto unir " <IDOCTYPE html> <html> <head> <style> ... "
- SensorDeUbicación1 . Latitud
- , lng: "
- SensorDeUbicación1 . Longitud
- } ; // The map, centered at Uluru var map = n...
- tomar global api\_key
- &callback=initMap"> </script> </body> </ht...
- nombreDeArchivo " /Android/data/mapa.html "
- llamar VisorWeb1 .IrALaUrl
- url " file:///mnt/sdcard/Android/data/mapa.html "

The 'url' block now includes the full file path: " file:///mnt/sdcard/Android/data/mapa.html ".





4.-Para obtener la latitud, longitud y dirección del dispositivo cuando haya cambios debe mostrarlos para ver el mapa de ubicación.

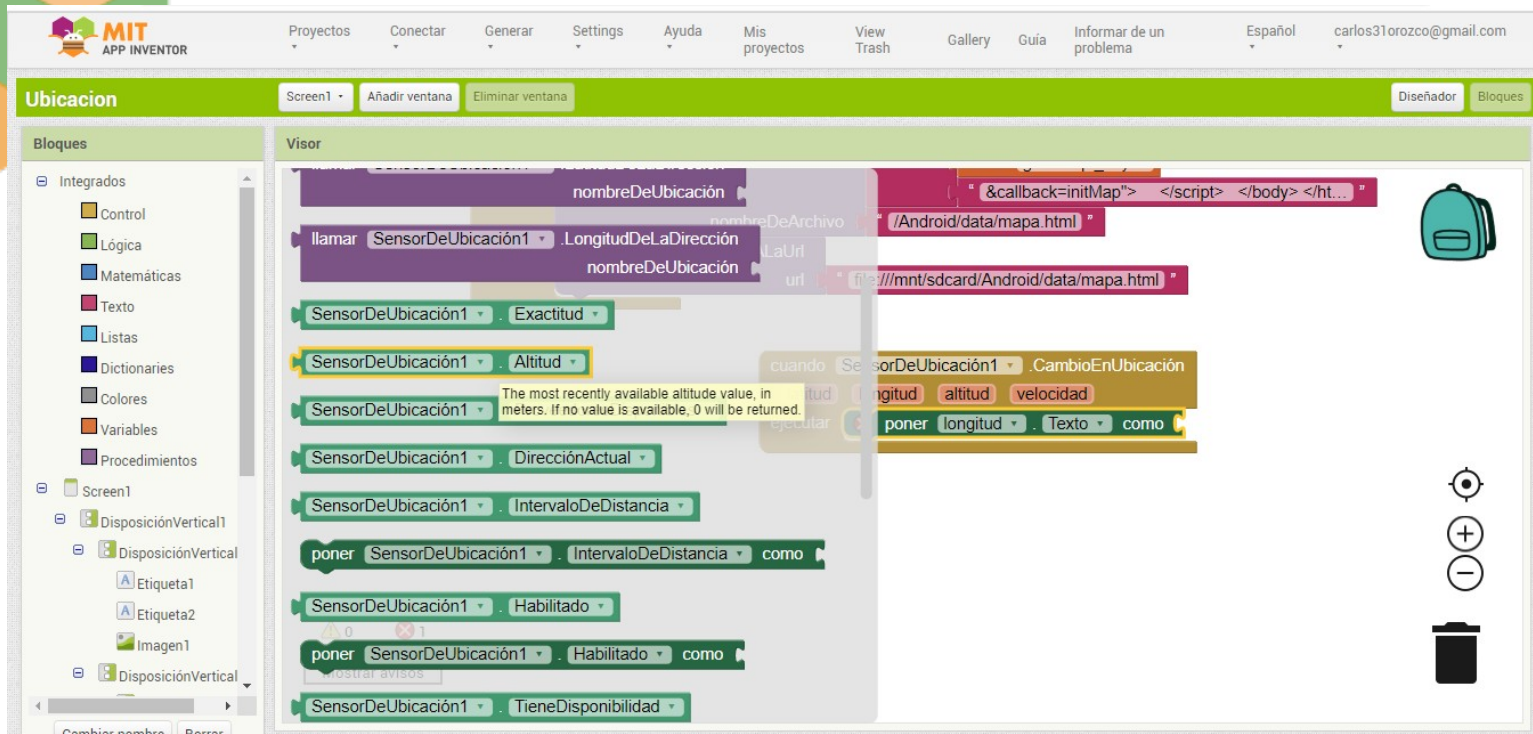
\*para ello debemos seleccionar el elemento SensorDeUbicacion seleccionamos la opción **Cuando SensorDeUbicacion .CambioEnUbicacion** ejecutar

The screenshot shows the MIT App Inventor interface. On the left, the 'Bloques' (Blocks) palette is visible, showing various components like 'Etiqueta1', 'Etiqueta2', 'Imagen1', and 'SensorDeUbicacion1'. The main workspace, labeled 'Visor', contains a logic block for 'SensorDeUbicacion1 . CambioEnUbicacion'. This block has a 'cuando' (when) trigger and an 'ejecutar' (do) section. The 'ejecutar' section contains several blocks: 'llamar' (call) blocks for 'SensorDeUbicacion1 . LatitudDeLaDirección' and 'SensorDeUbicacion1 . LongitudDeLaDirección', and 'SensorDeUbicacion1 . Exactitud', 'SensorDeUbicacion1 . Altitud', and 'SensorDeUbicacion1 . ProveedoresDisponibles' blocks. A code editor on the right shows the generated HTML and JavaScript code for the event.

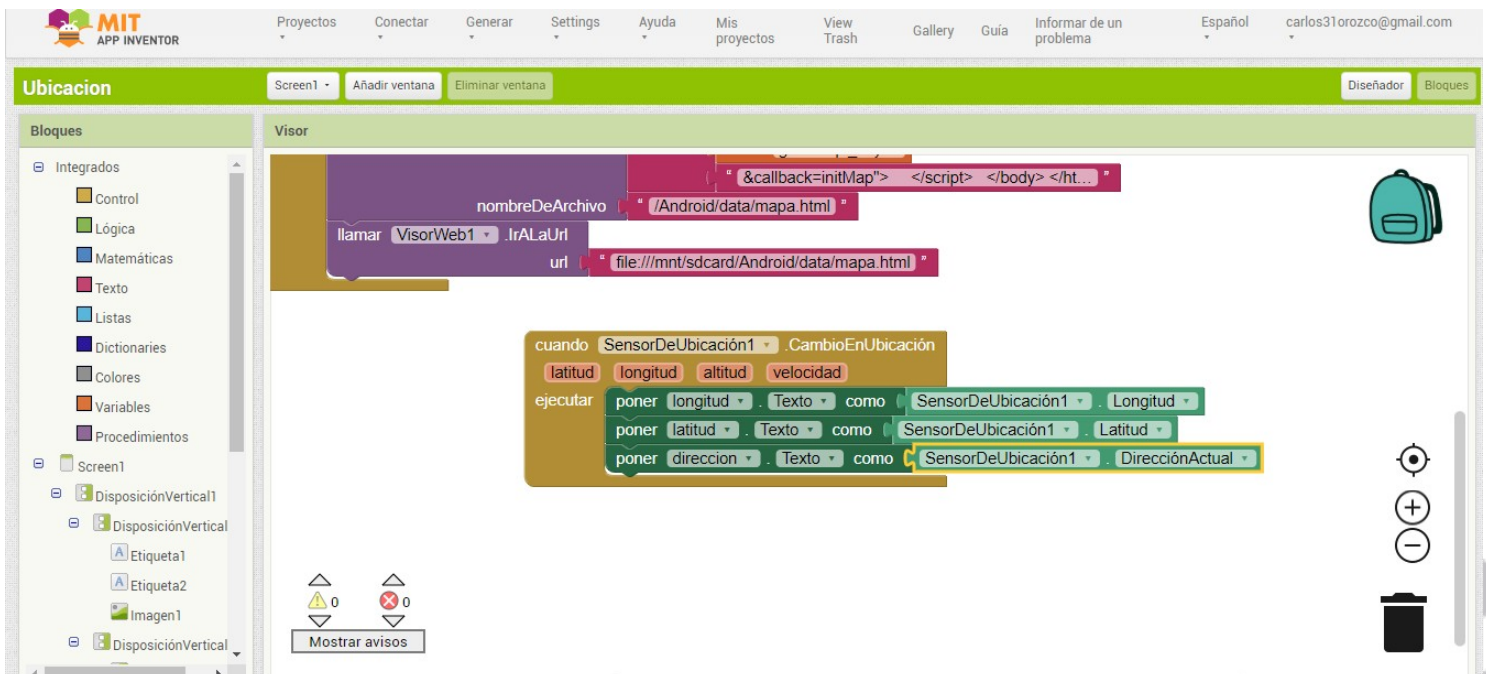
\*Seleccionamos la etiqueta longitud y seleccionamos la opción **poner longitud .texto** como

The screenshot shows the MIT App Inventor interface for a project named 'Ubicacion'. The 'Bloques' palette on the left includes 'Integrados' (Integrated) blocks like 'Control', 'Lógica', 'Matemáticas', 'Texto', 'Listas', 'Diccionarios', 'Colores', 'Variables', and 'Procedimientos'. The 'Visor' workspace shows a logic block for 'SensorDeUbicacion1 . CambioEnUbicacion' with an 'ejecutar' section. The 'ejecutar' section contains a 'poner longitud . Texto' block, which is highlighted. Other blocks in the 'ejecutar' section include 'poner longitud . HasMargins', 'poner longitud . Alto', 'poner longitud . Alto como', 'poner longitud . HeightPercent como', 'poner longitud . ColorDeTexto', 'poner longitud . ColorDeTexto como', 'poner longitud . Visible', 'poner longitud . Visible como', and 'poner longitud . Ancho'. The code editor on the right shows the generated HTML and JavaScript code for the event.

\*seleccionamos el elemento SensorDeUbicacion y elegimos la opción **SensorDeUbicacion1 .Altitud**



\*cambiar altitud por longitud y agregar lo mismo para latitud y dirección quedando de la siguiente manera:



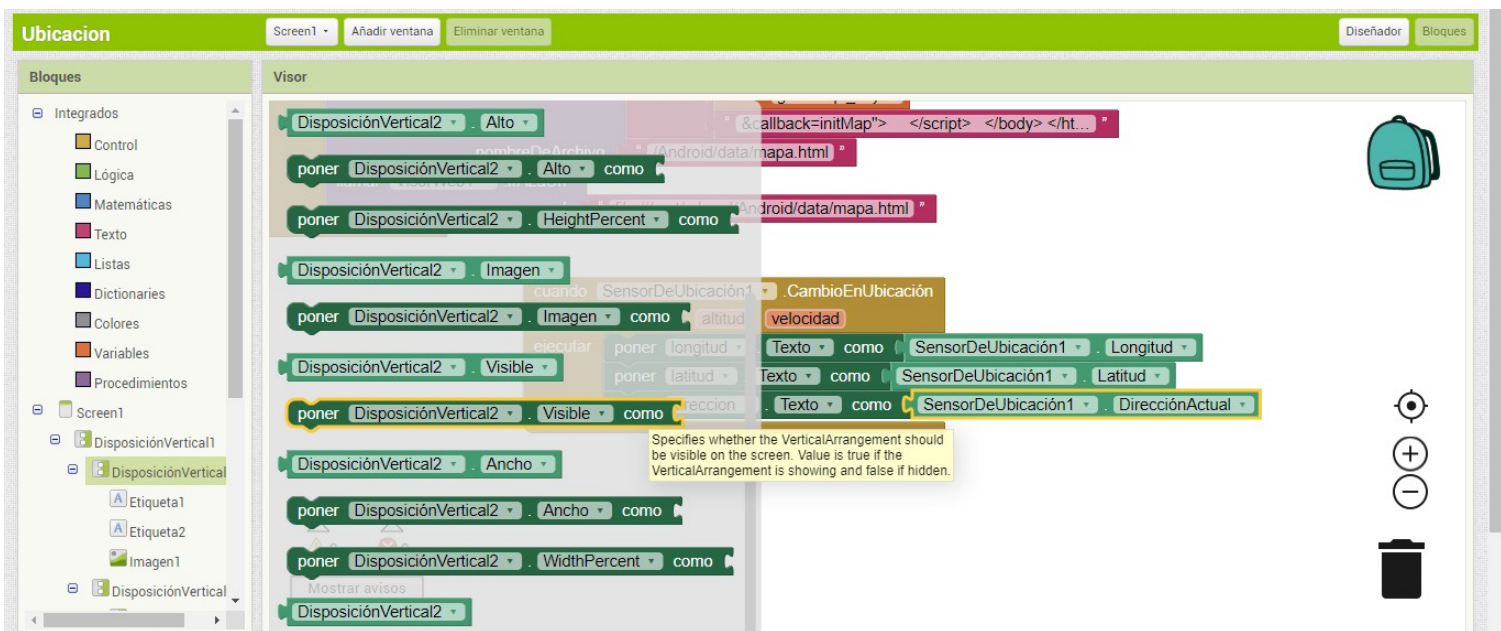


\*Cuando encuentre cambio de dirección, va a poner la disposición donde tiene tu nombre y dibujo del mapa lo va a deshabilitar y va a mostrar la disposición donde se encuentra el botón para ver el mapa de ubicación

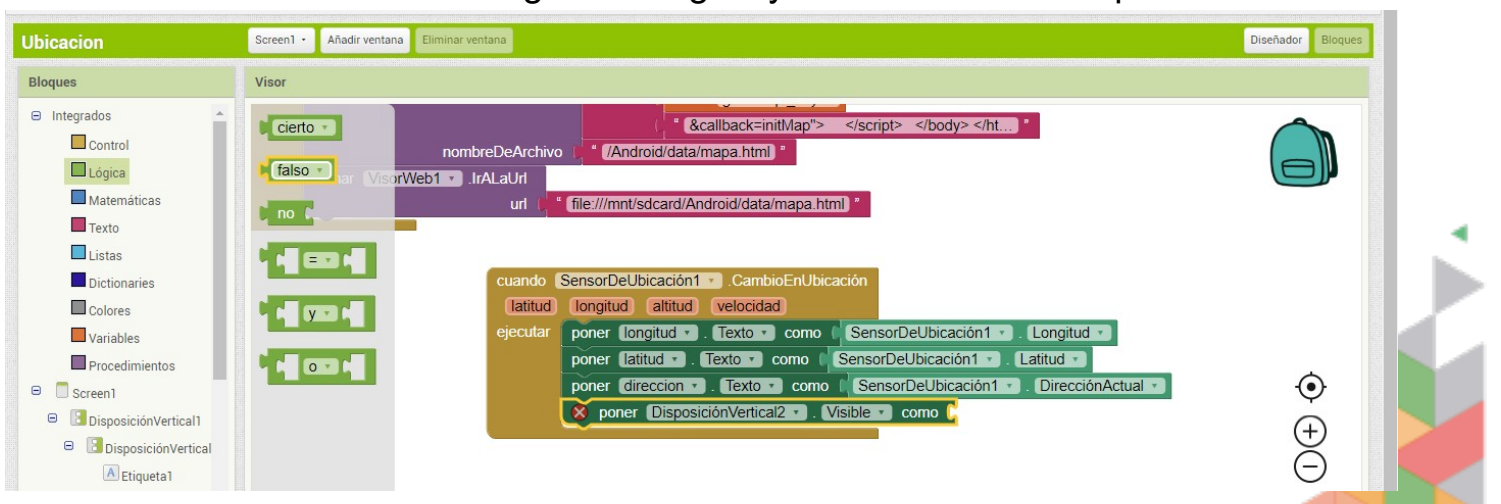
Para ello nos vamos a diseño para verificar que la DisposicionVertical2 sea donde contiene tu nombre y la DisposicionVertical3 sea donde contiene los elementos de ver mapa con datos actuales.

\*una vez confirmado que son esas disposiciones regresamos a bloques para seguir programando.

Seleccionamos la DisposicionVertical2 y seleccionamos la opción poner DisposicionVertical2 . Visible como



\*seleccionamos los integrados Lógica y seleccionamos la opción falso



\*Hacemos lo mismo para la disposición vertical3 pero en el integrado lógico ponemos **Cierto**

Quedando de la siguiente manera:

The screenshot shows the Android Studio IDE with the following components:

- Bloques (Blocks):** A sidebar on the left containing categories like Integrados, Control, Lógica, Matemáticas, Texto, Listas, Dictionaries, Colores, Variables, and Procedimientos. It also shows a project tree with 'Screen1' and 'DisposiciónVertical1'.
- Visor (Viewer):** The central area displaying the logic blocks:
  - llamar VisorWeb1 . IrALaUrl:** nombreDeArchivo: "/Android/data/mapa.html", url: "file:///mnt/sdcard/Android/data/mapa.html"
  - cuando SensorDeUbicación1 . CambioEnUbicación:** ejecutar
    - poner longitud . Texto como SensorDeUbicación1 . Longitud
    - poner latitud . Texto como SensorDeUbicación1 . Latitud
    - poner direccion . Texto como SensorDeUbicación1 . DirecciónActual
    - poner DisposiciónVertical2 . Visible como falso
    - poner DisposiciónVertical3 . Visible como cierto
- Diseñador (Designer):** The right side showing a map view with a backpack icon and navigation controls (compass, zoom in, zoom out, and trash).

Solo queda CREAR el APK e instalar la aplicación en el dispositivo móvil.

