

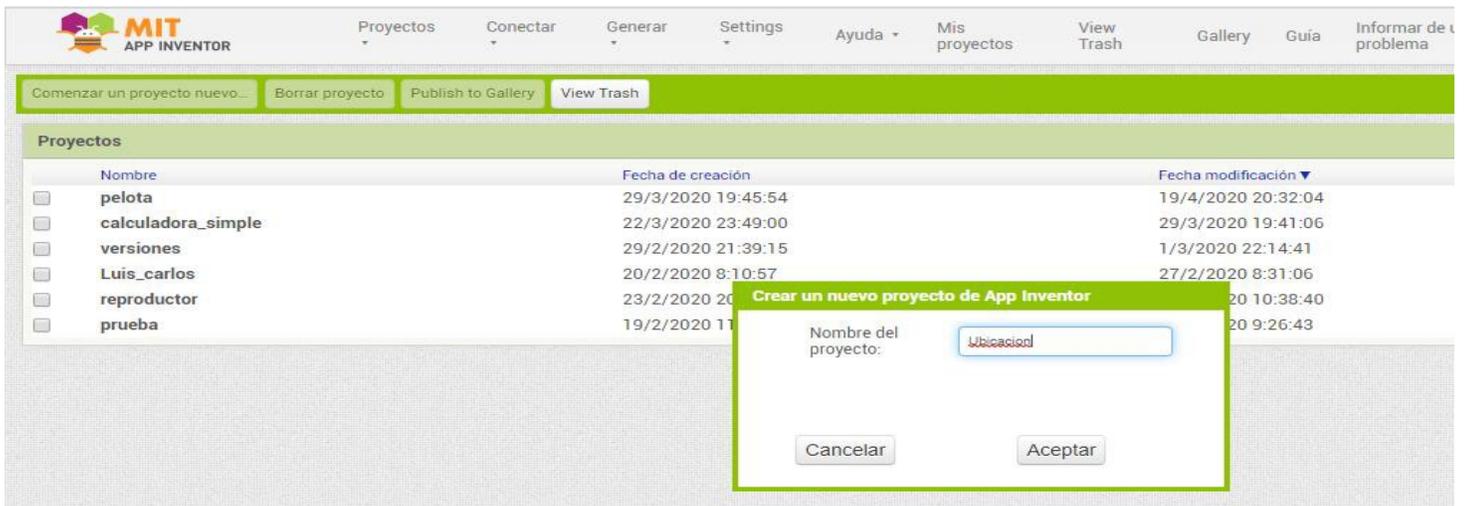


# Practica guiada

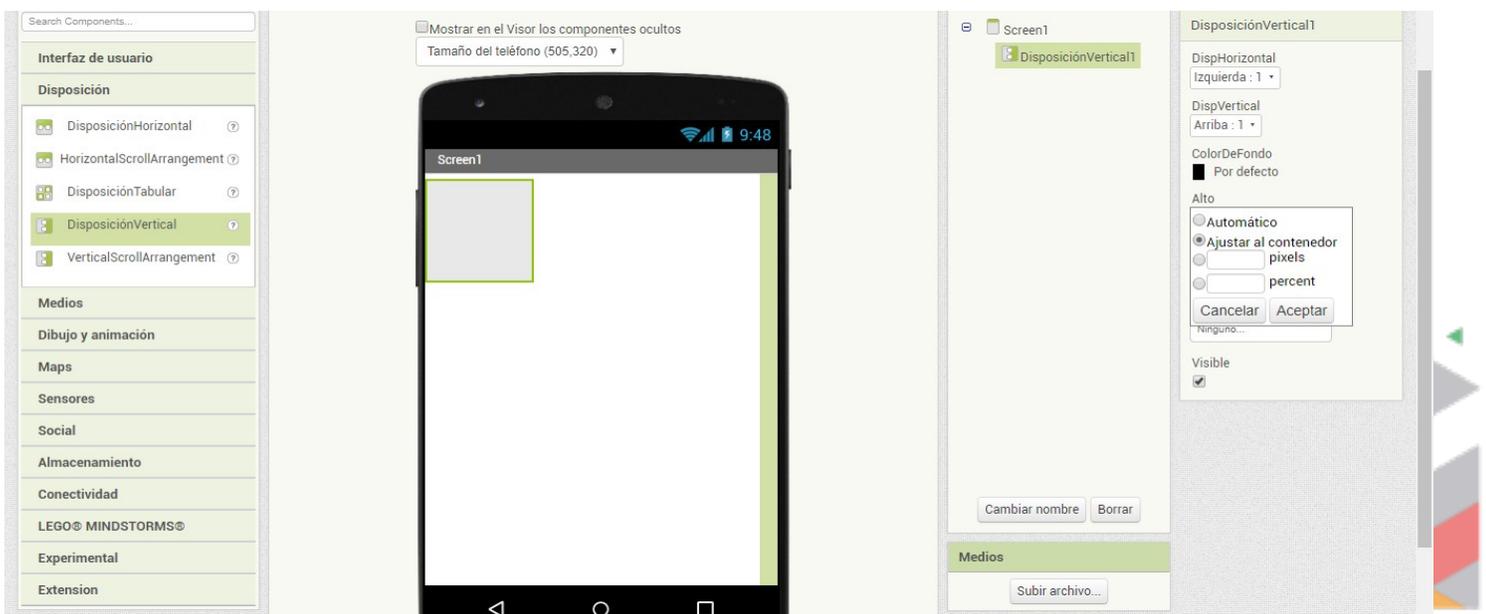
## Aplicación con sensor de ubicación

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.

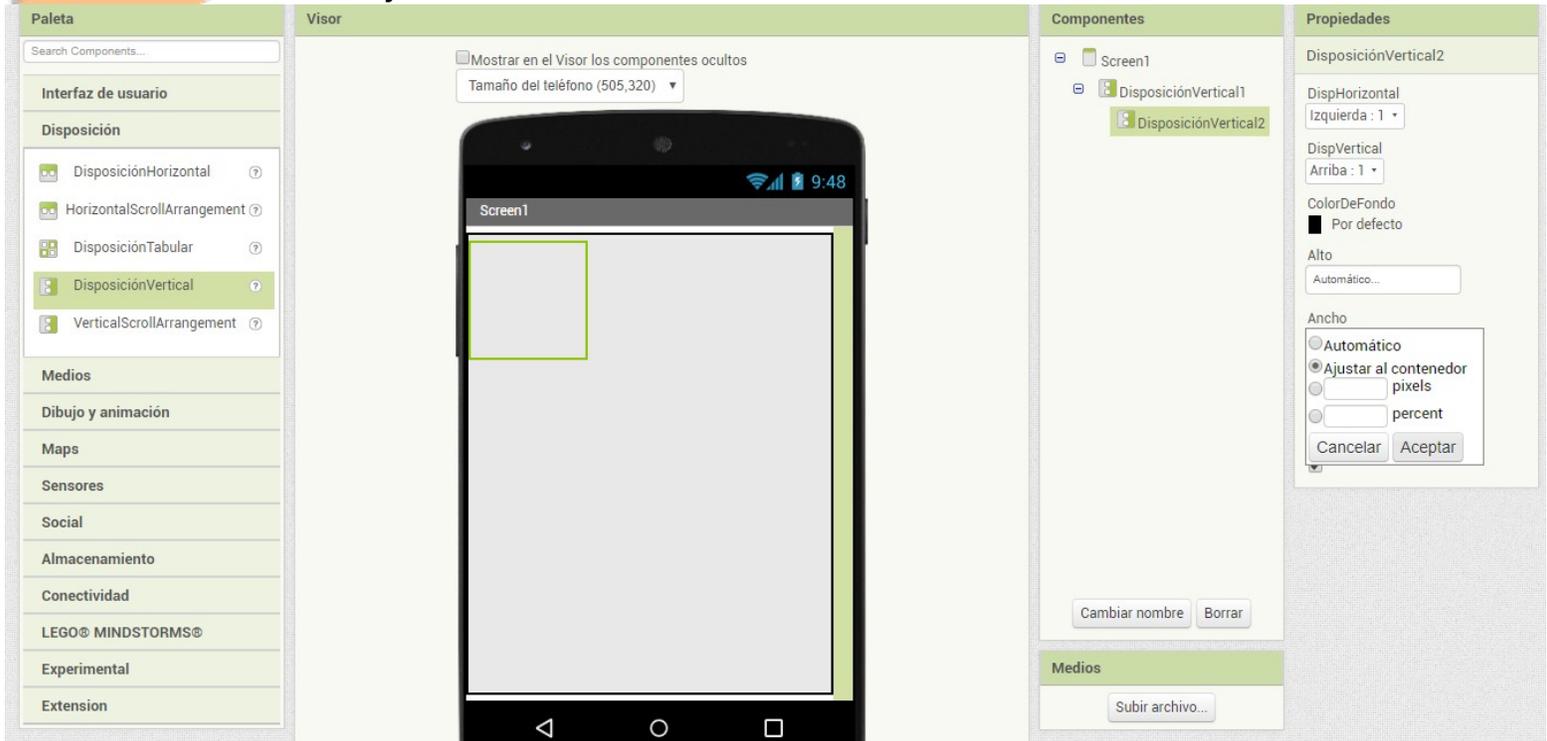
1.- Crear un nuevo proyecto en App Inventor y llamarlo ubicación.



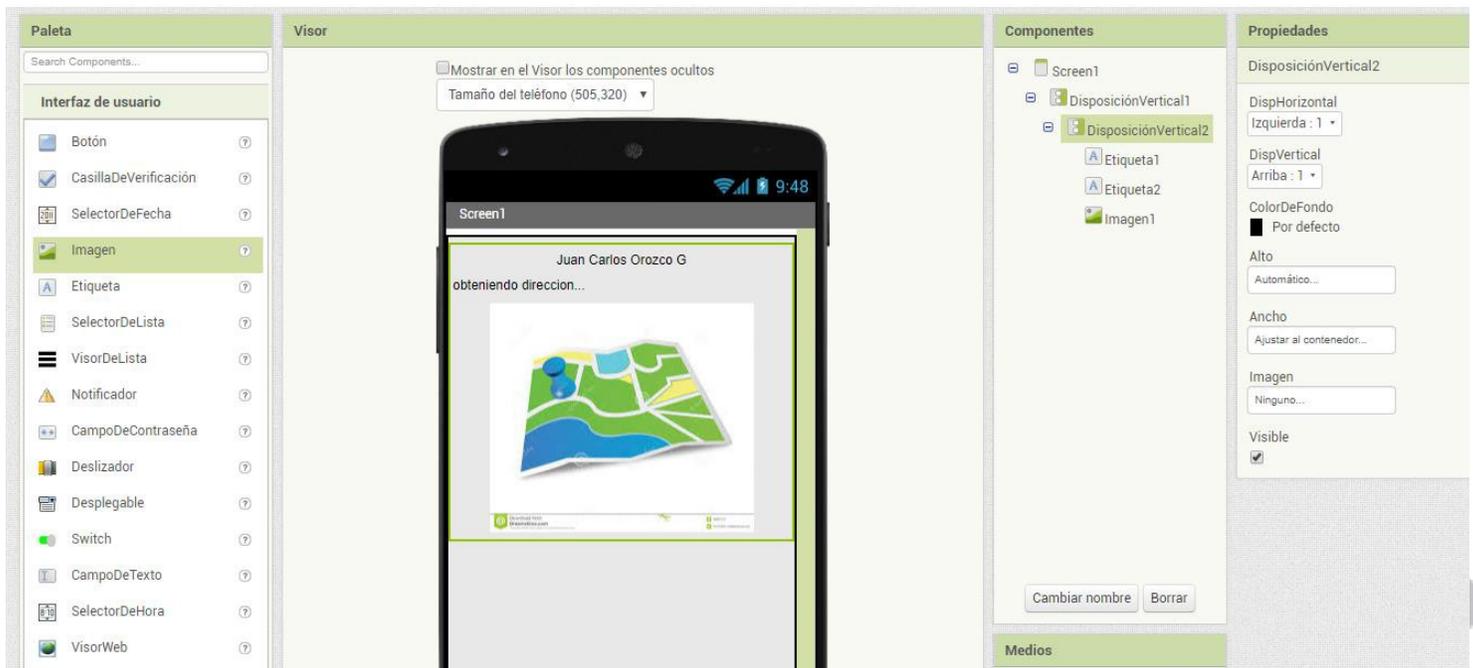
2.- Agregar una disposición Vertical, ajustar al contenedor en ancho y alto.



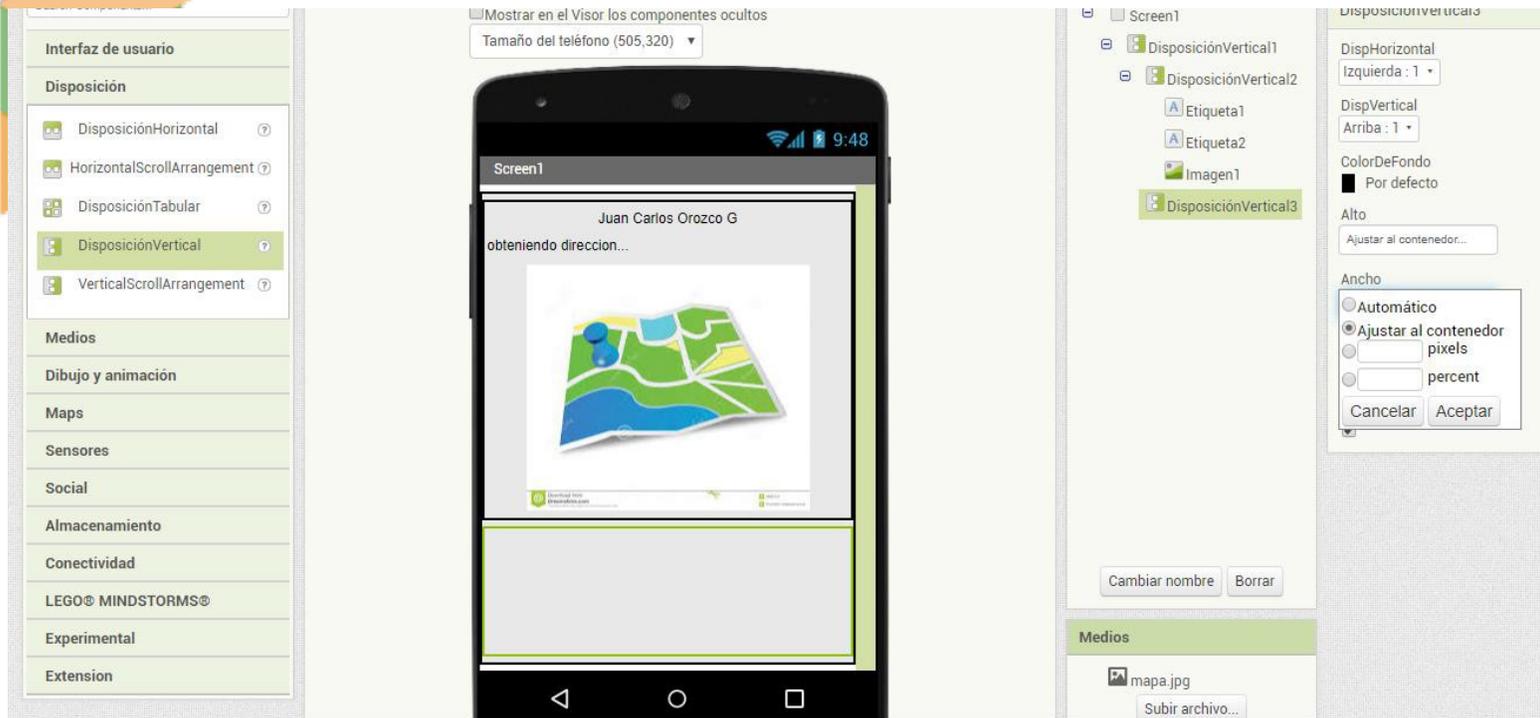
3.-Agregar dentro de la disposición agregada otra disposición vertical, con ancho ajustado al contenedor



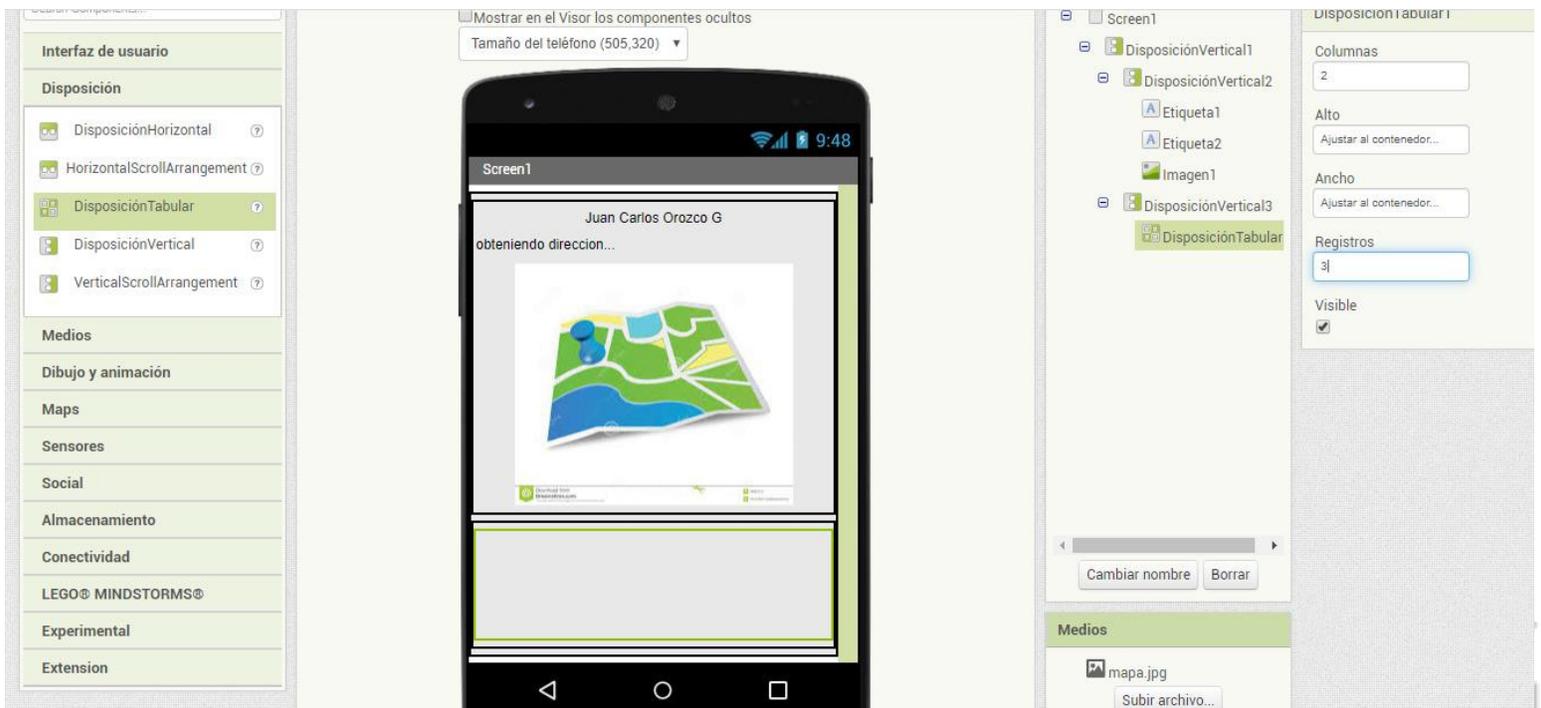
4.- Agregar dentro de esta nueva disposición Vertical, un Label o etiqueta para tu nombre, otro diciendo **Obteniendo dirección** y una imagen con una foto/imagen de mapa



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

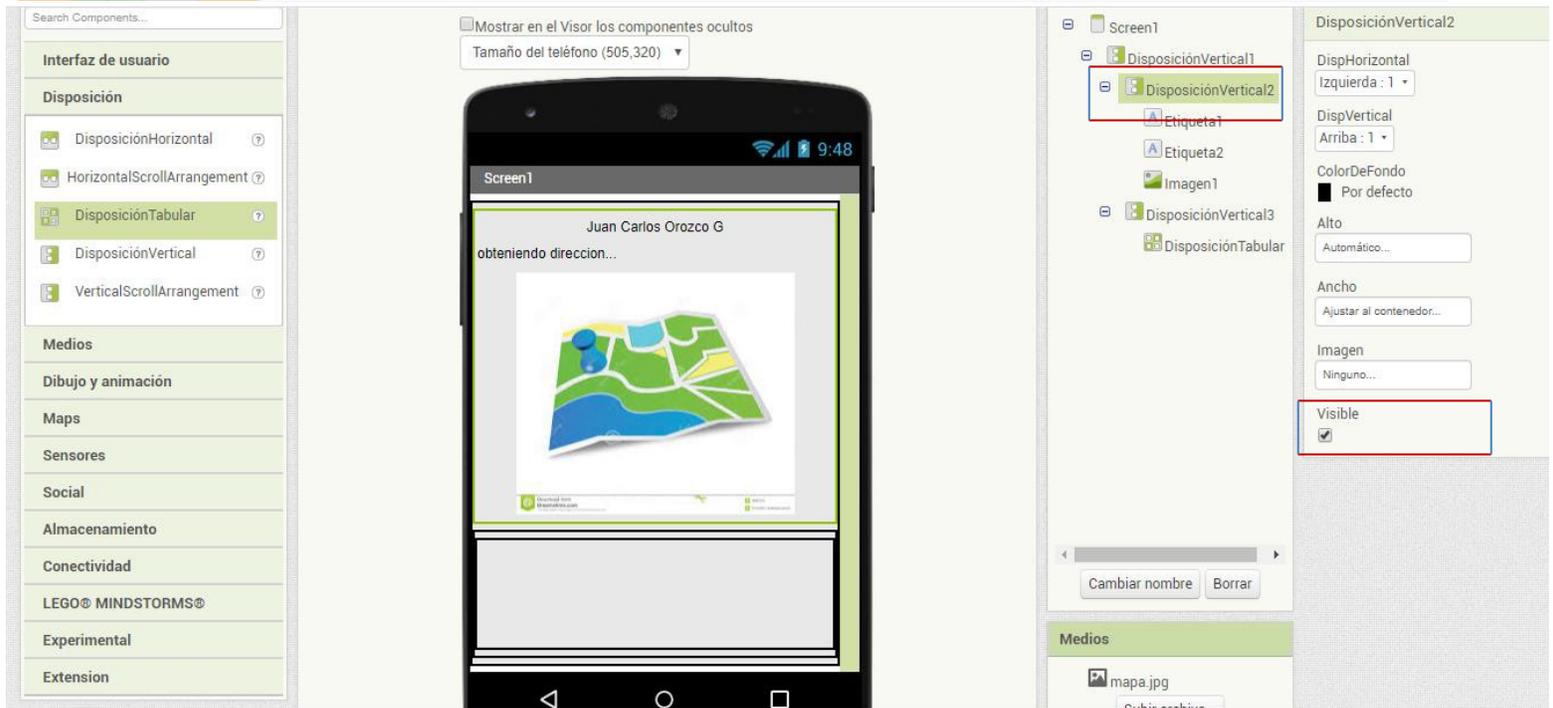


6.- Agregar una disposición tabular dentro la disposición agregada, ajustar ancho al contenedor y el número de registros la cambiamos a 3.

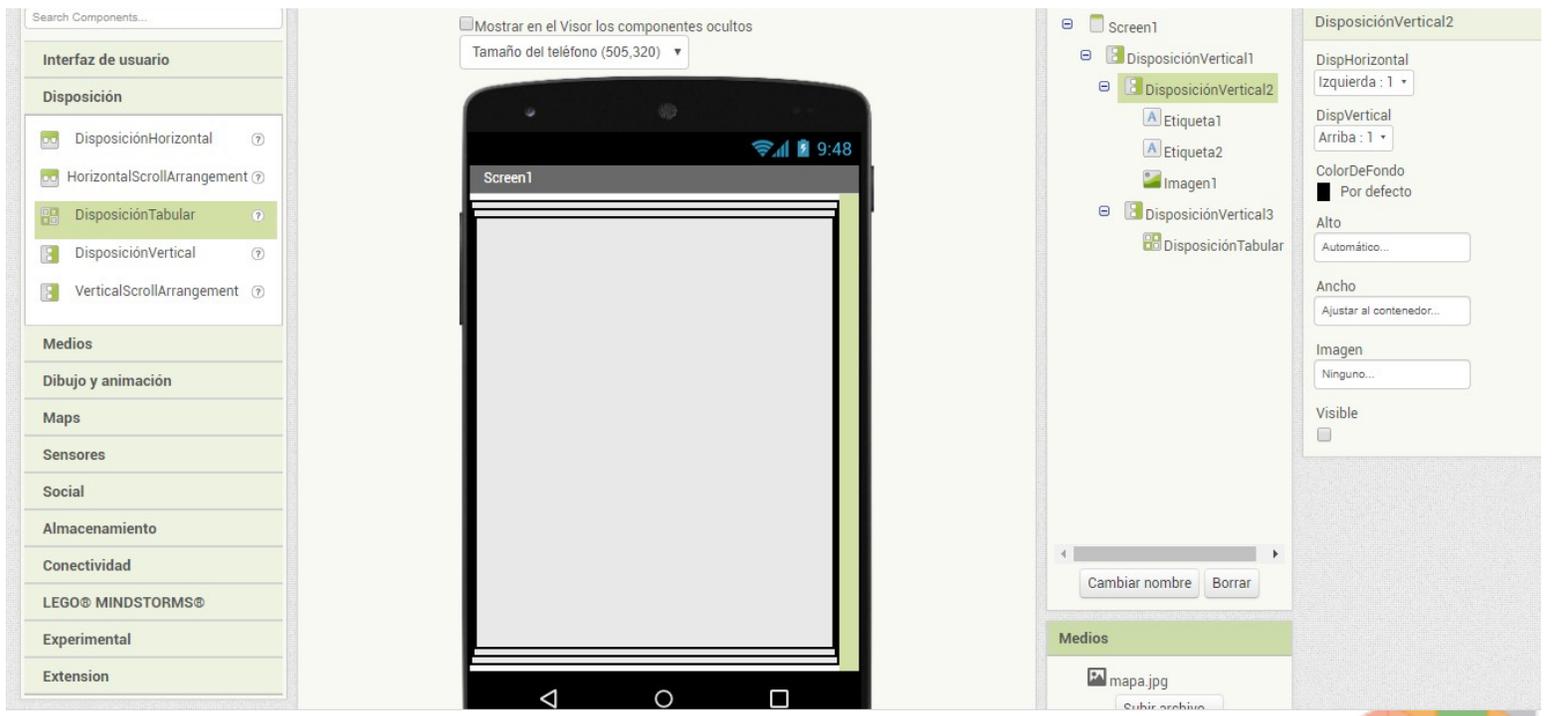


7.-Para continuar trabajando con la disposición tabular, podemos esconder la disposición Vertical que contiene tu nombre y el mapa.

Para eso seleccionamos el elemento DisposicionVertical2 y deseleccionamos la opción Visible



Quedando de la siguiente manera:

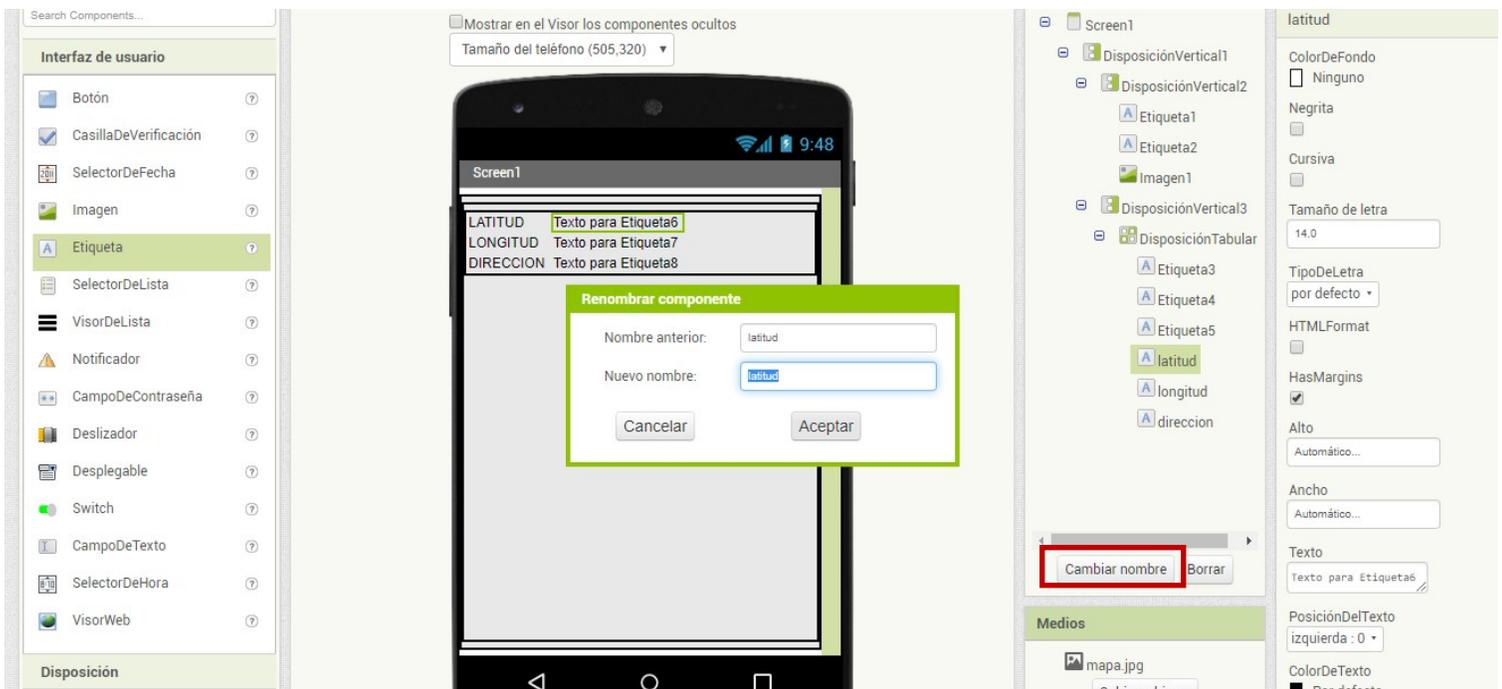


8.- dentro de la disposición tabular, agregar 6 label, nombrarlos de la siguiente manera por columna, El primero como LATITUD, el segundo como LONGITUD y el tercero como DIRECCION.

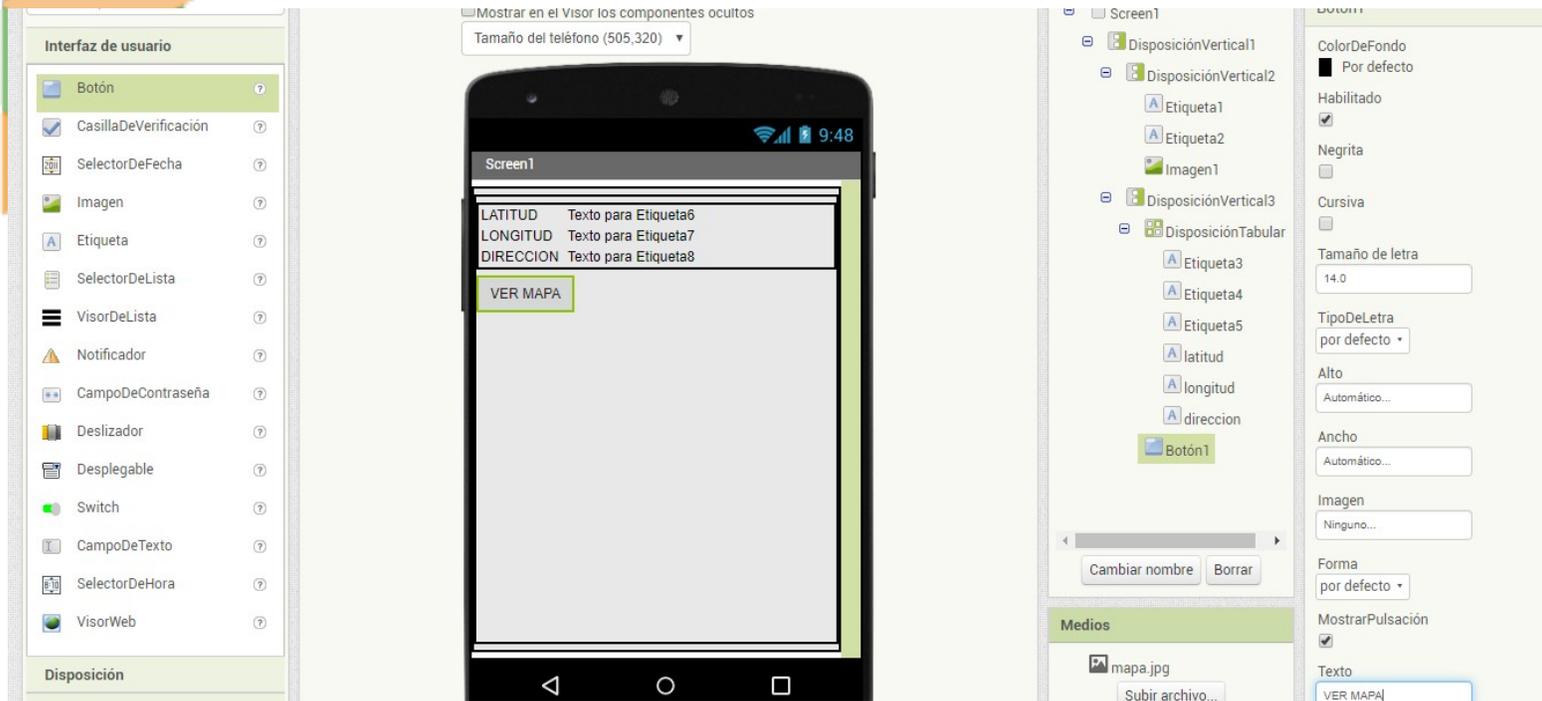
En la otra columna solo es cambiar el nombre de la variable de la siguiente manera.

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

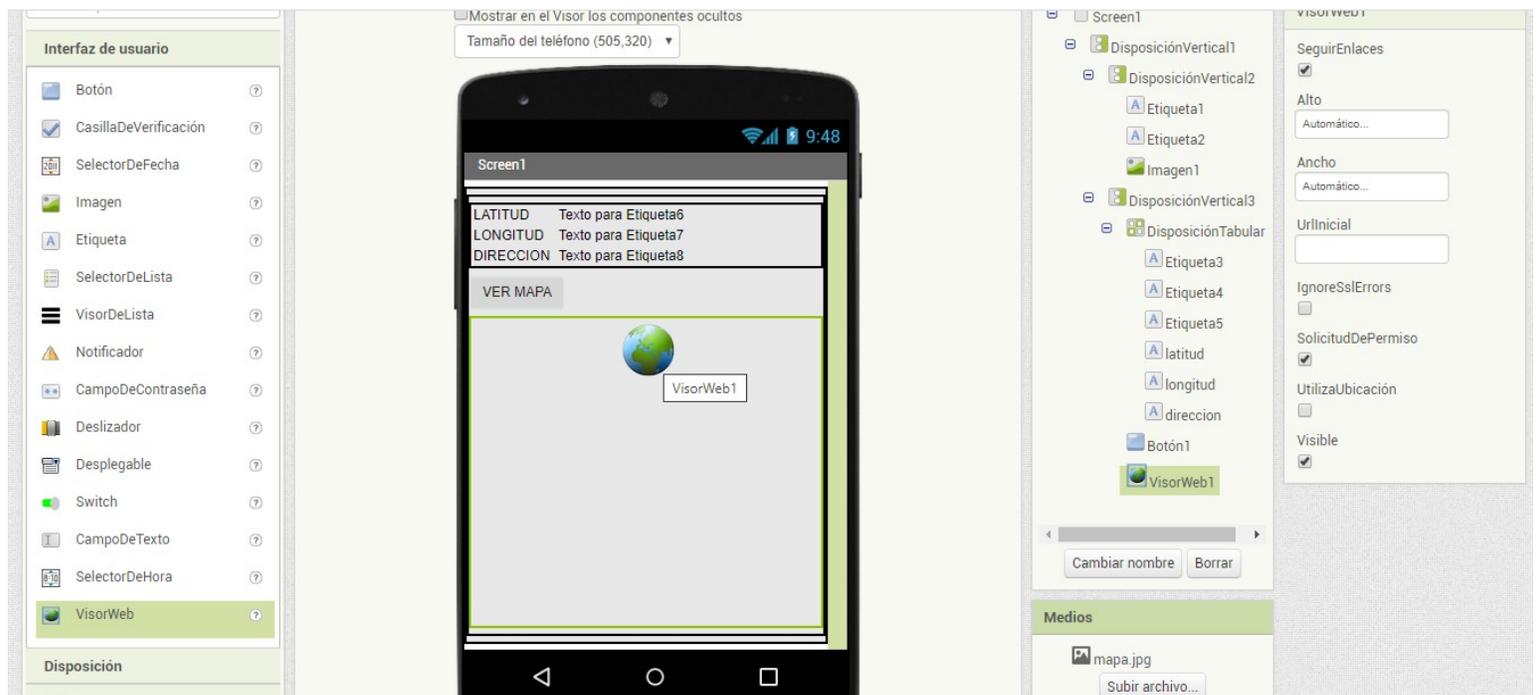
Hacer lo mismo con las otras dos etiquetas, renombrando con los respectivos nombres: longitud y direccion.



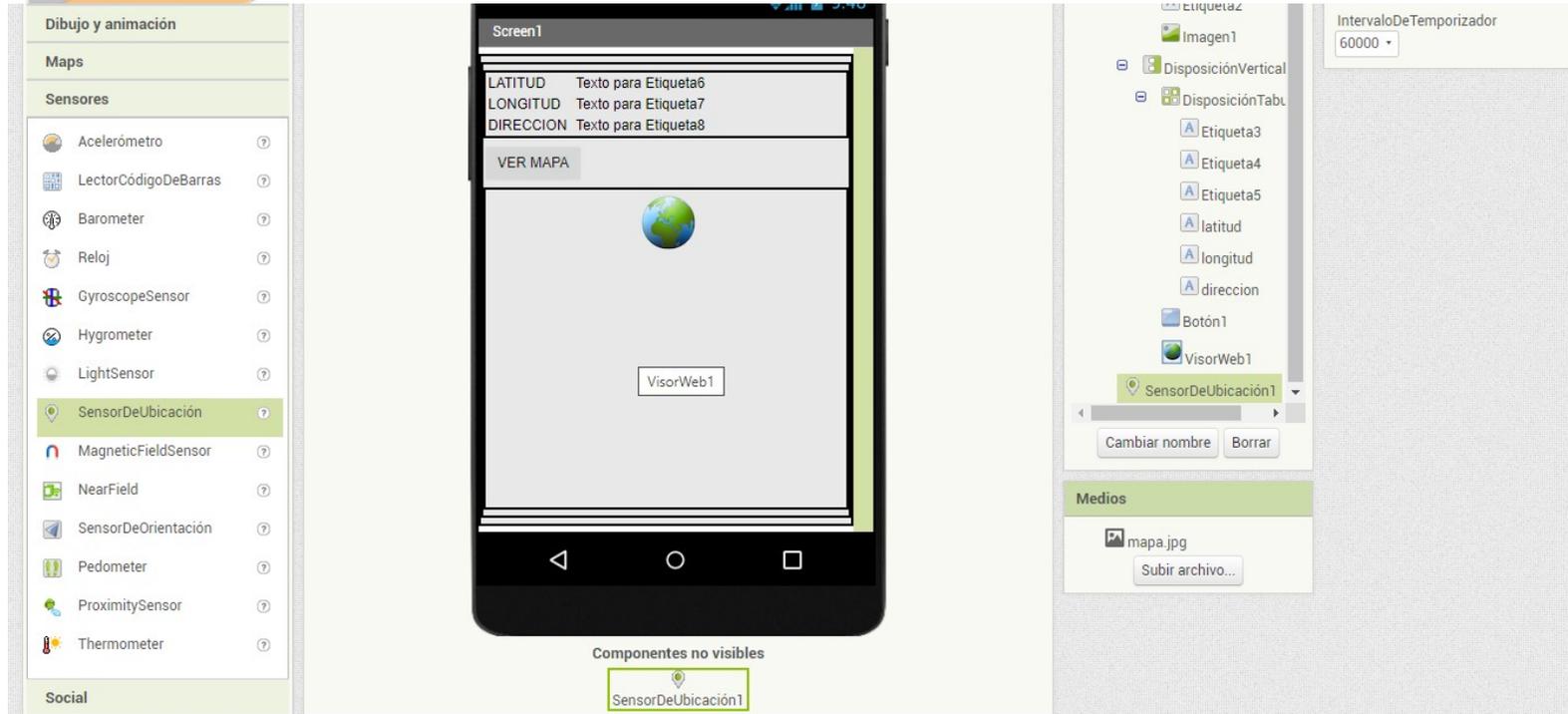
9.- Agregar un botón debajo de la disposición tabular, que le cambiaremos de texto a ver mapa.



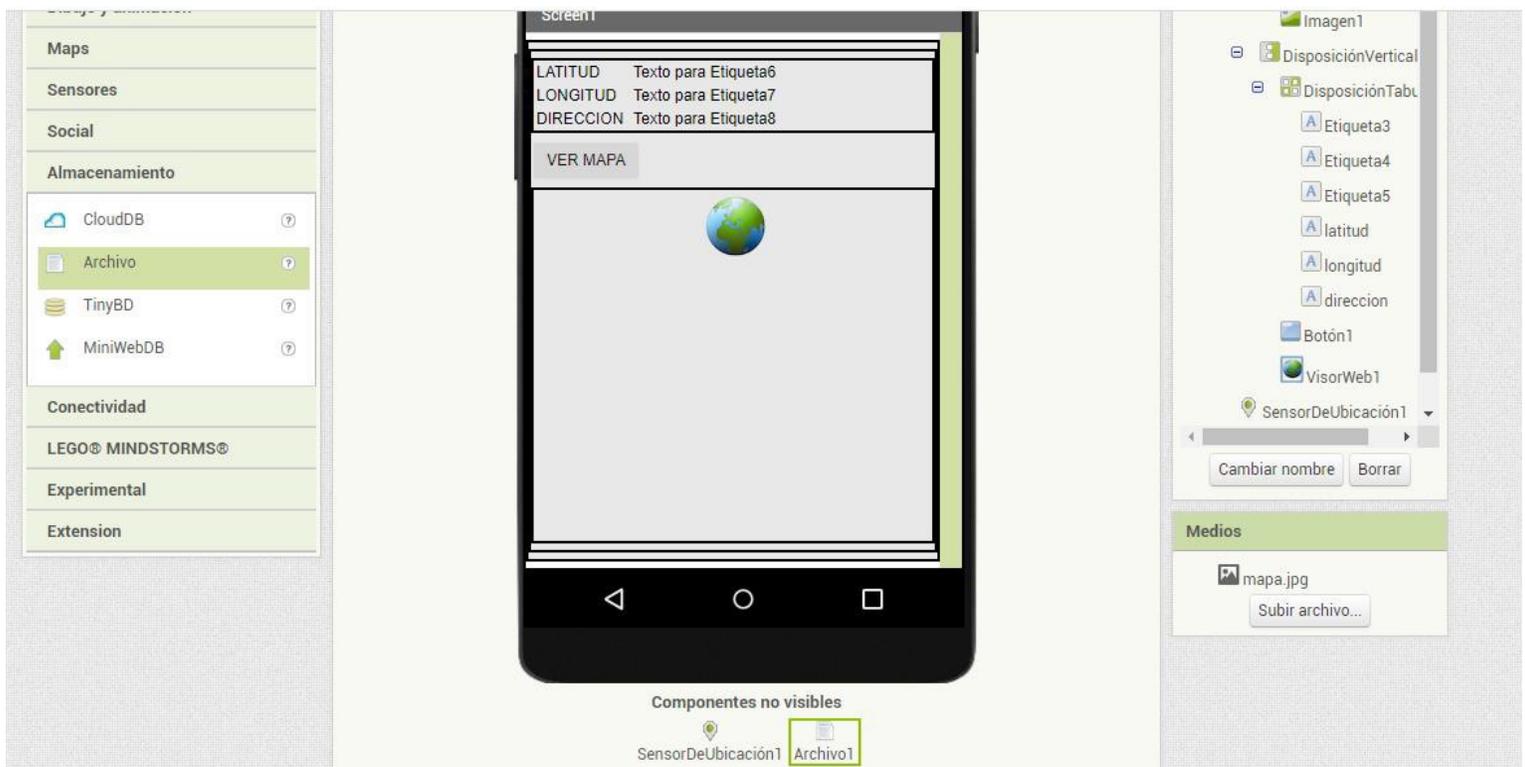
10.- Agregar un Visor web (para poder ver el mapa en tiempo real).



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

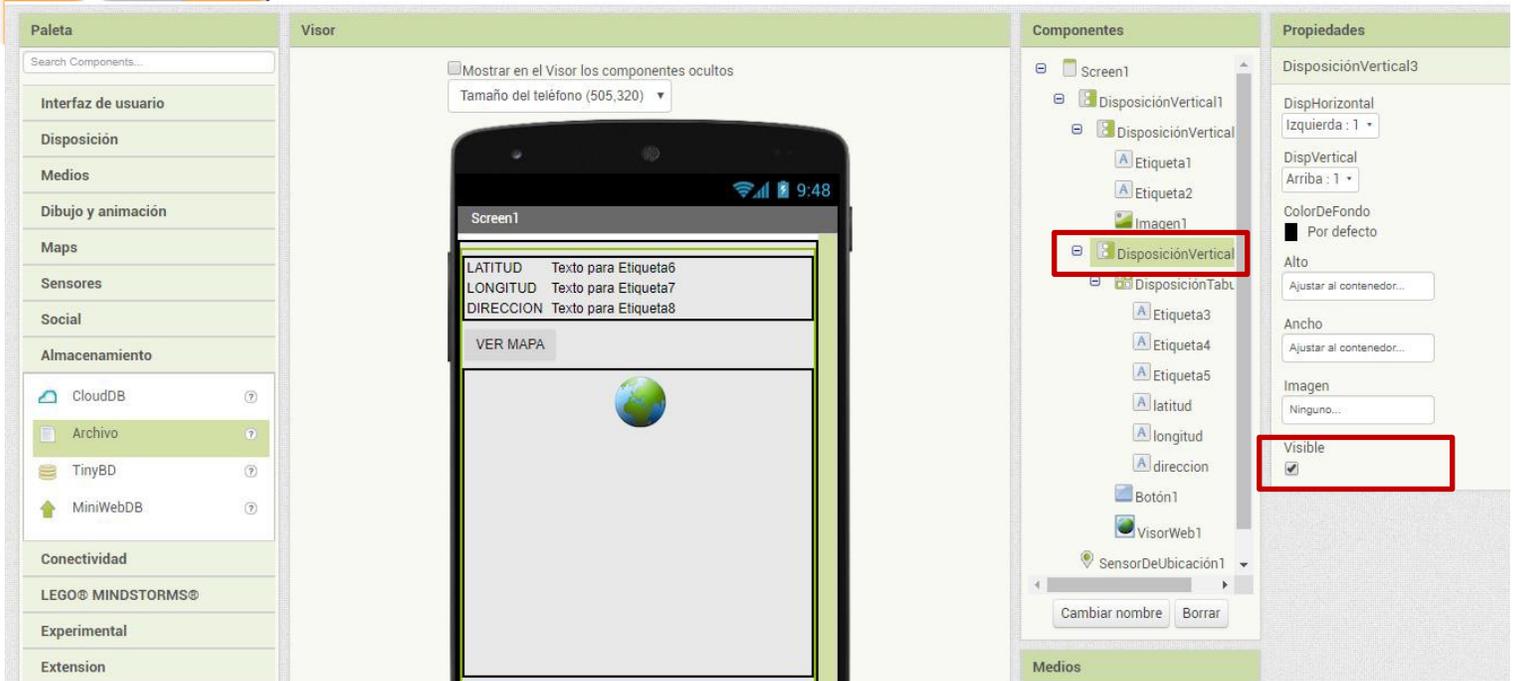


12.- y agregar un archivo de almacenamiento (para generar el documento html y poder mostrar el mapa de google).

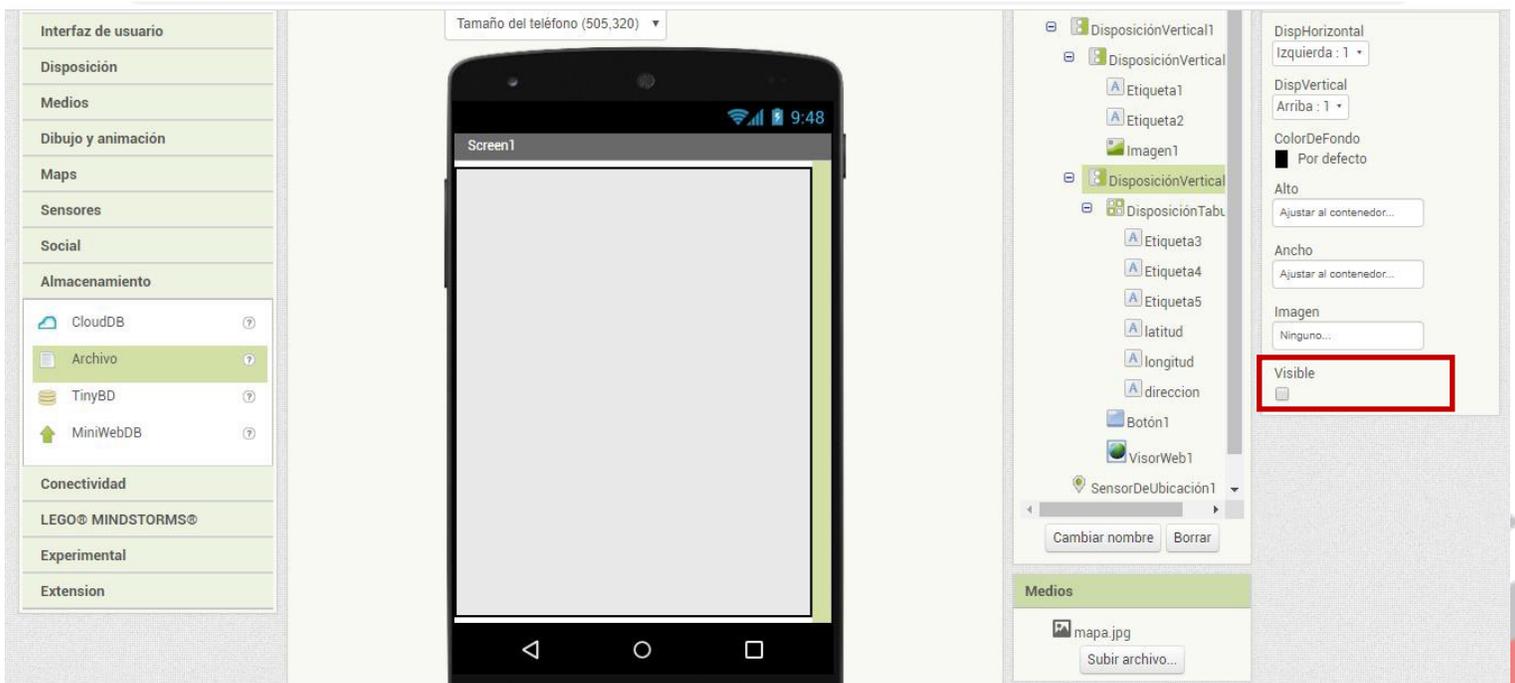


13.- Ahora escondemos la disposición vertical que acabamos de diseñar.

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

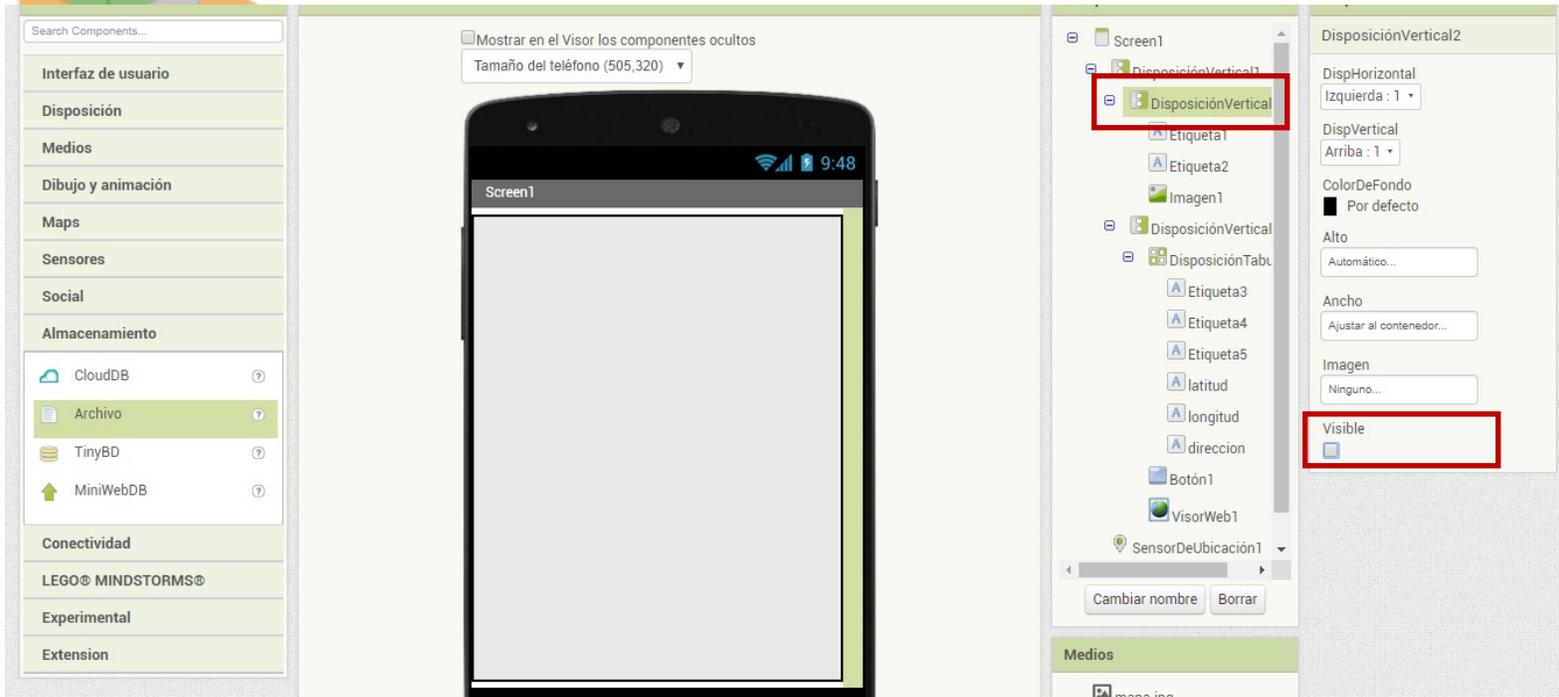


Quedando de la siguiente manera:

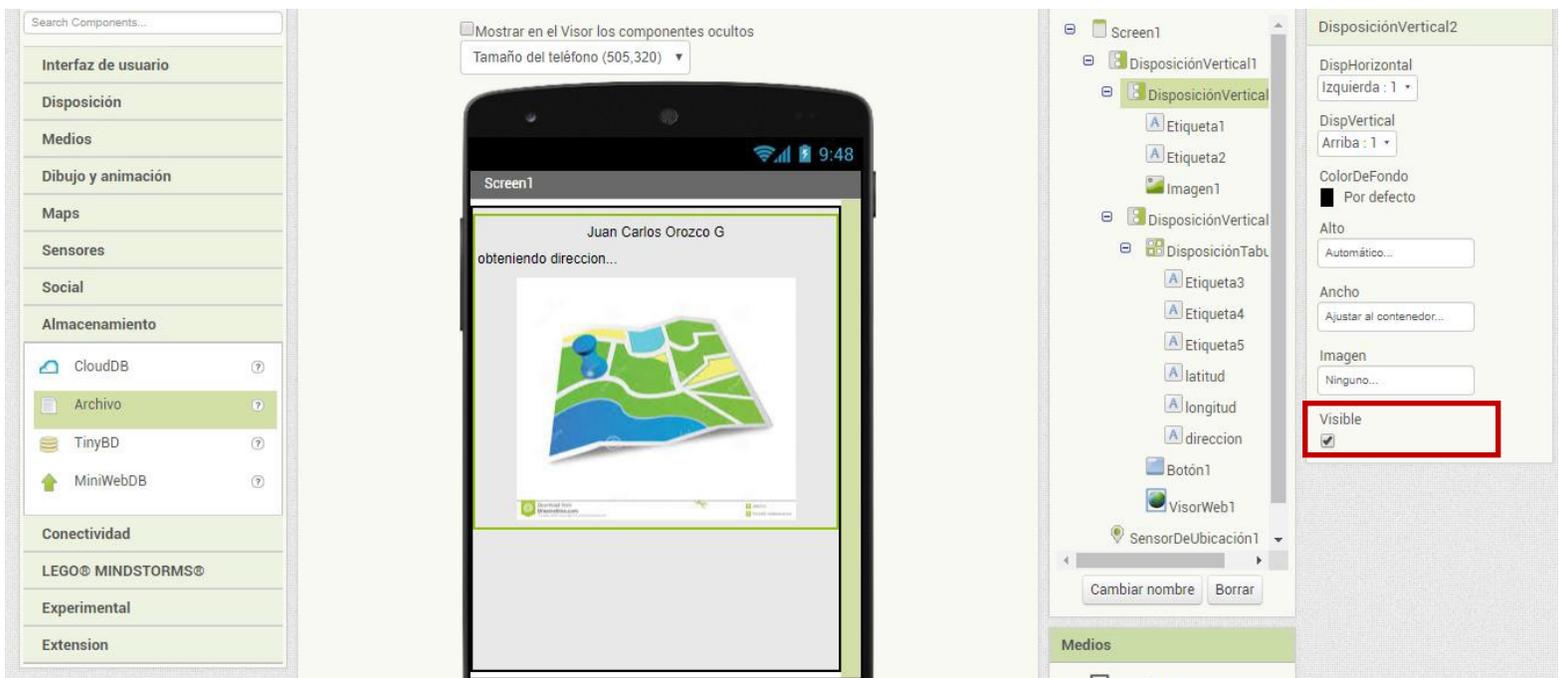


14.- ahora hacemos visible nuevamente la disposición que contiene tu nombre y la imagen del mapa

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



Quedando de la siguiente manera:



# Codificación en bloques

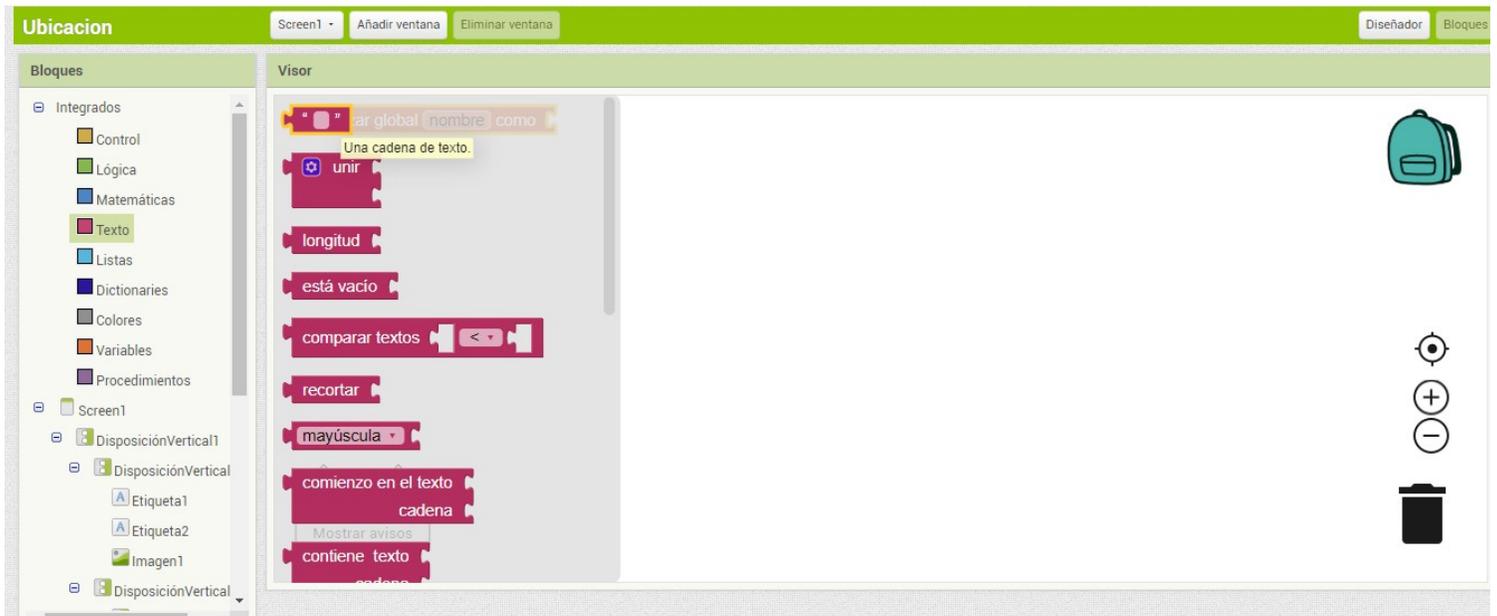
1.- Crear una variable global para después ser utilizada.

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



The screenshot shows the Scratch 'Ubicación' (Location) interface. The 'Bloques' (Blocks) panel on the left has 'Variables' selected. The 'Visor' (Viewer) panel on the right shows a script with the following blocks: 'inicializar global nombre como', 'tomar', 'poner a', 'inicializar local nombre como en', and another 'inicializar local nombre como en' block. The interface includes a top bar with 'Screen1', 'Añadir ventana', and 'Eliminar ventana' buttons, and a right sidebar with a backpack icon and zoom controls.

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



The screenshot shows the Scratch 'Ubicación' interface with 'Texto' (Text) selected in the 'Bloques' panel. The 'Visor' panel shows a script with the following blocks: 'inicializar global nombre como', 'Una cadena de texto.', 'unir', 'longitud', 'está vacío', 'comparar textos', 'recortar', 'mayúscula', 'comienzo en el texto cadena', and 'contiene texto'. The interface includes a top bar with 'Screen1', 'Añadir ventana', and 'Eliminar ventana' buttons, and a right sidebar with a backpack icon and zoom controls.



\*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>
```

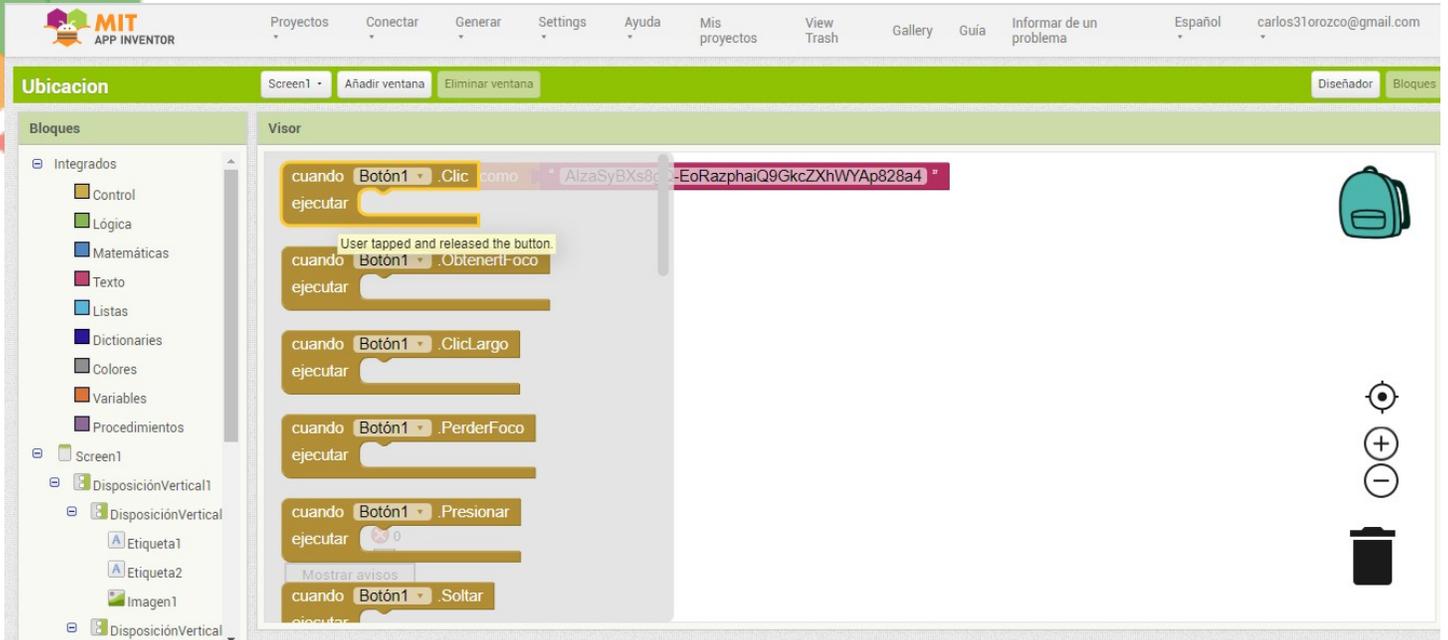
Y lo agregamos en el texto quedando de la siguiente manera:

The screenshot shows the Android Studio interface. The top bar is green and contains the title 'Ubicación' and several window management buttons. Below the top bar, there are two main panels: 'Bloques' (Blocks) on the left and 'Visor' (Preview) on the right. The 'Bloques' panel shows a tree view of the UI components, with 'Text' selected. The 'Visor' panel shows a preview of the text block, with the API key 'AlzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4' highlighted in a purple box. The code snippet in the preview is: 'inicializar global api\_key como 'AlzaSyBXs8gQ-EoRazphaiQ9GkcZXhWYAp828a4''.

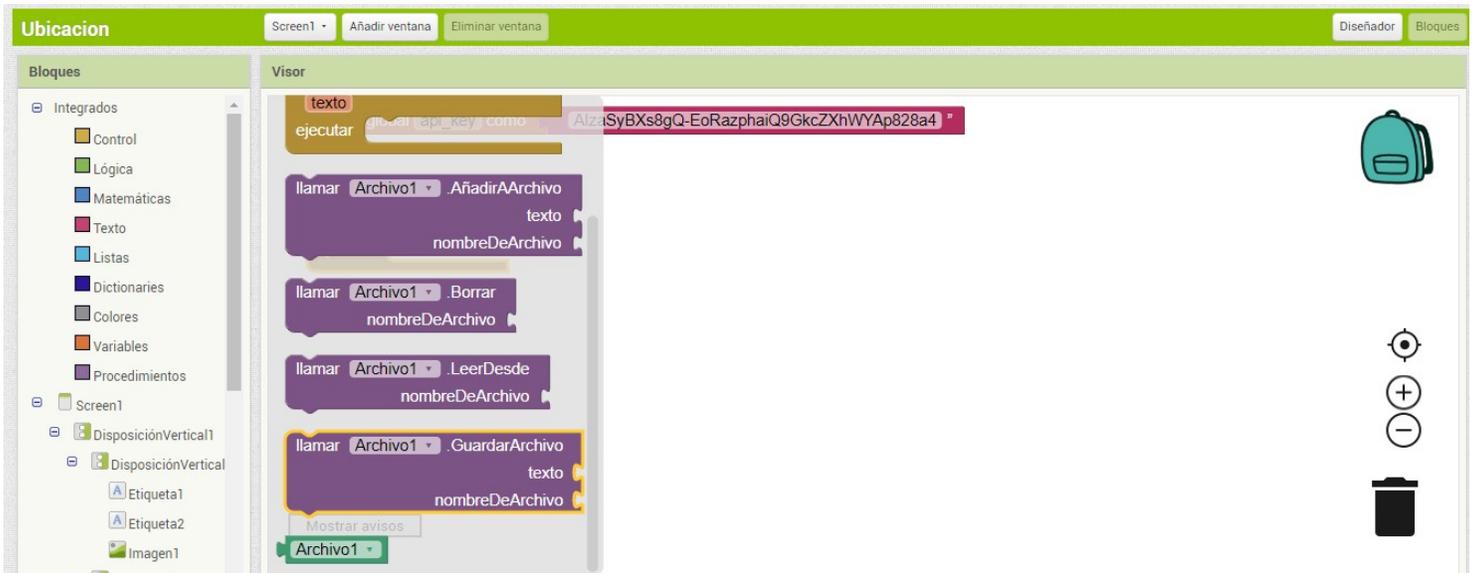


2.- Cuando demos clic al botón ver mapa nos va a mostrar el mapa según la ubicación en la que nos encontremos.

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



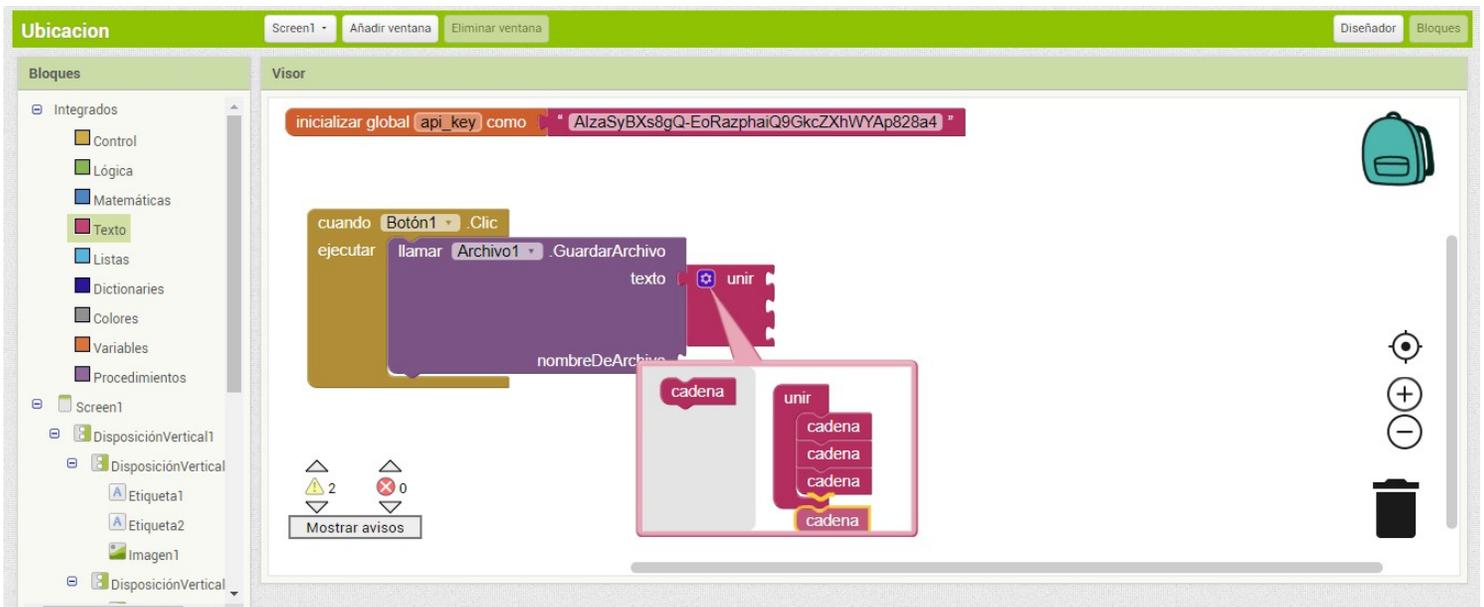
\*Seleccionamos el elemento archivo y la opción **llamar Archivo .GuardarArchivo texto nombreDeArchivo**



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



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



\*agregar de integrados un texto en blanco.

The screenshot shows the MIT App Inventor web interface. The top navigation bar includes 'Proyectos', 'Conectar', 'Generar', 'Settings', 'Ayuda', 'Mis proyectos', 'View Trash', 'Gallery', 'Guia', 'Informar de un problema', 'Español', and 'carlos31orozco@gmail.com'. The main workspace is titled 'Ubicación' and shows a code block for 'Botón1' with the event 'Clic'. The code block contains a 'llamar' block for 'Archivo1' with the method '.GuardarArchivo'. The 'texto' parameter is set to a 'unir' block with two empty string blocks. The 'nombreDeArchivo' parameter is set to 'Una cadena de texto.'. The left sidebar shows a 'Bloques' panel with categories like 'Integrados', 'Control', 'Lógica', 'Matemáticas', 'Texto', 'Listas', 'Dictionaries', 'Colores', 'Variables', and 'Procedimientos'. The 'Screen1' folder is expanded, showing 'DisposiciónVertical1' and 'Etiqueta1', 'Etiqueta2', and 'Imagen1'.

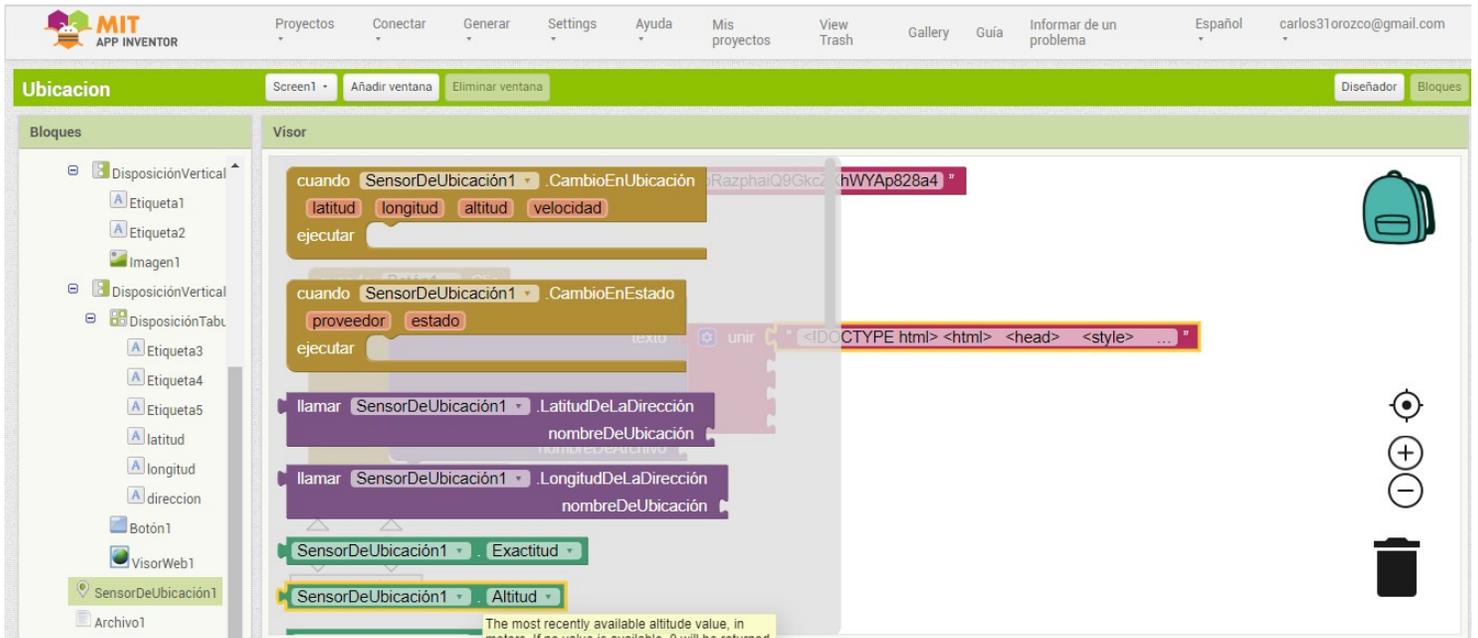
\*Copiar del bloc de notas (mapa 1) el siguiente texto

```
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 SensorDeUbicacion y seleccionar la opción **SensordeUbicacion1 .Altitud**



## \*Cambiamos de Altitud a Latitud

The screenshot shows the MIT App Inventor interface. In the center, a code block is being edited. The block is a 'SensorDeUbicación1' block with a dropdown menu open, showing 'Altitud' selected. The code block contains the following logic:

```

cuando Botón1 . Clic
ejecutar
  llamar Archivo1 . GuardarArchivo
  texto unir "<DOCTYPE html> <div id='map'></div>"
  nombreDeArchivo "SensorDeUbicación1 . Altitud"
  
```

The interface also shows a 'Bloques' panel on the left with various UI components like 'Etiqueta1', 'Etiqueta2', 'Imagen1', etc. A 'Mostrar avisos' button is visible at the bottom of the code editor.

\*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">

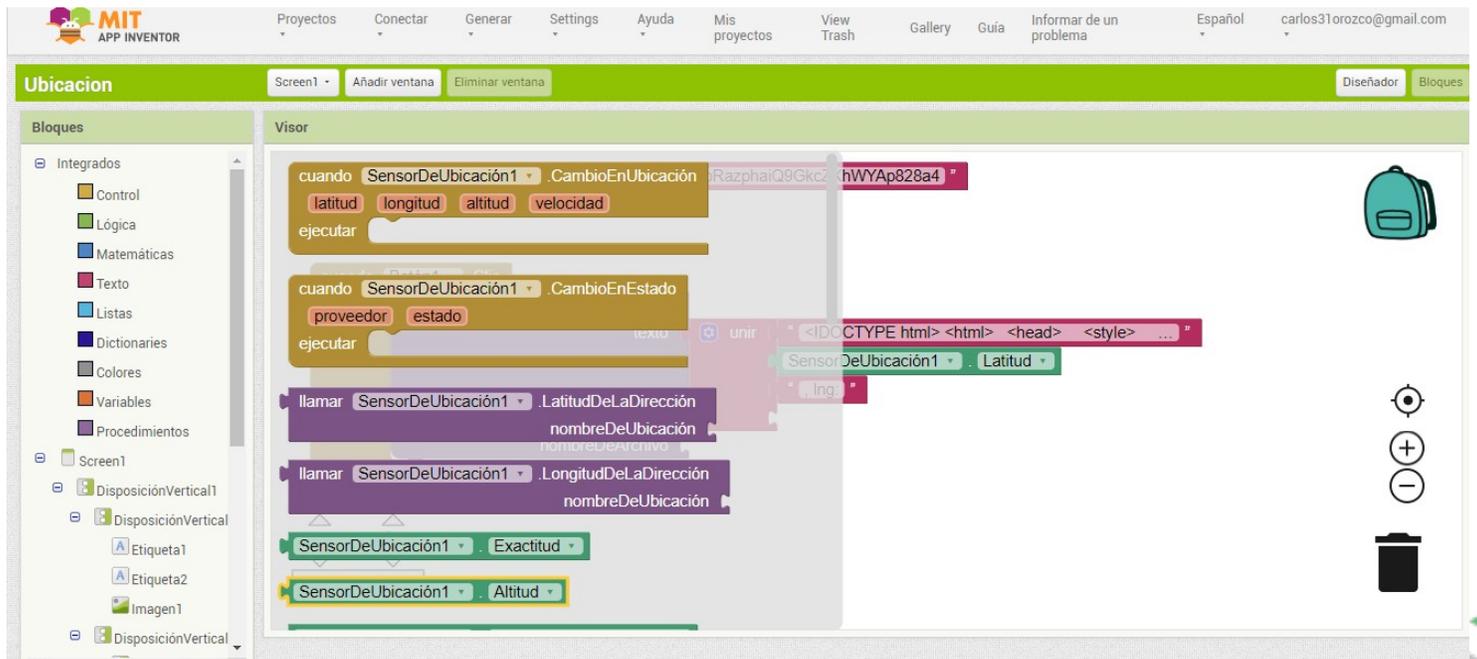
```



\*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**

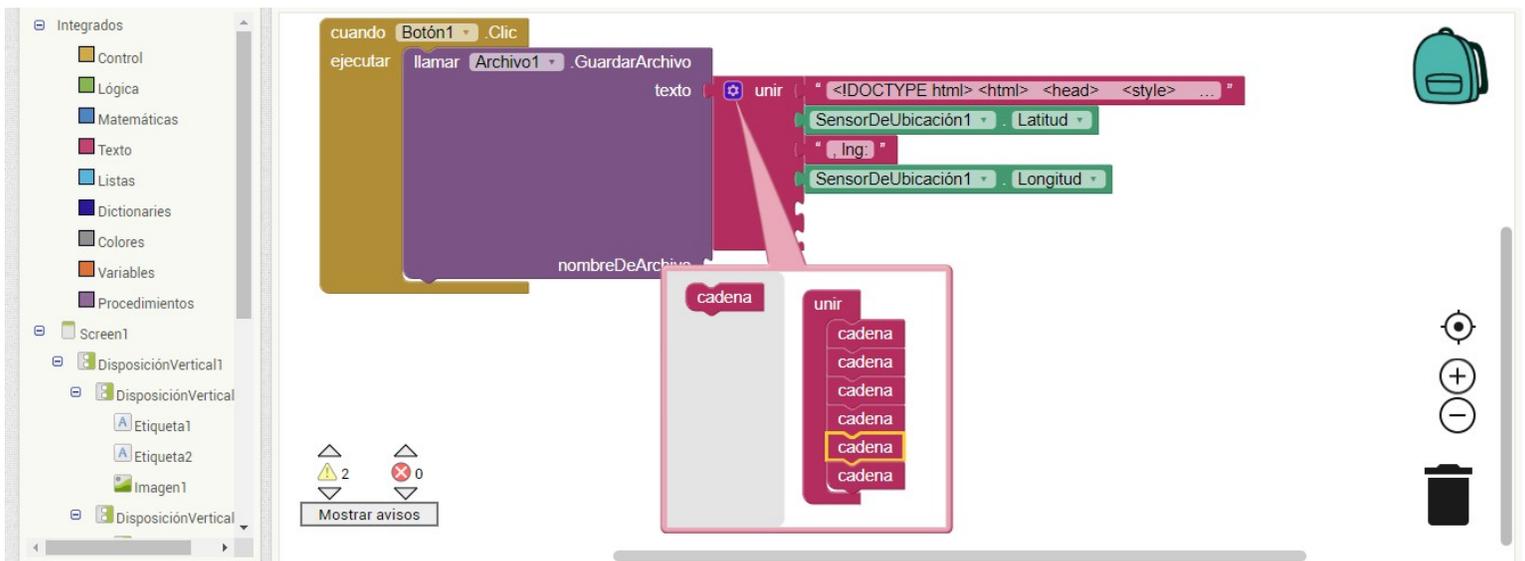


## \*Cambiar altitud por longitud



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 concatenated string. The string is: `"<IDOCTYPE html>< <SensorDeUbicación1.Latitud> <SensorDeUbicación1.Lng:> <SensorDeUbicación1.Altitud>"`. A dropdown menu is open over the 'unir' block, listing various location-related properties, with 'Longitud' selected.

\*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 MIT App Inventor interface with the same code block as above. The 'unir' block now contains: `"<IDOCTYPE html><html><head><style> ... <SensorDeUbicación1.Latitud> <SensorDeUbicación1.Lng:> <SensorDeUbicación1.Longitud>"`. A callout box highlights the 'unir' block, showing a 'cadena' block being dragged to the end of the string. The callout box contains a 'cadena' block and a 'unir' block with five 'cadena' blocks stacked vertically, the bottom one highlighted in yellow.



\*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 Archivo1 .GuardarArchivo' with a 'texto' block containing the following HTML code:

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



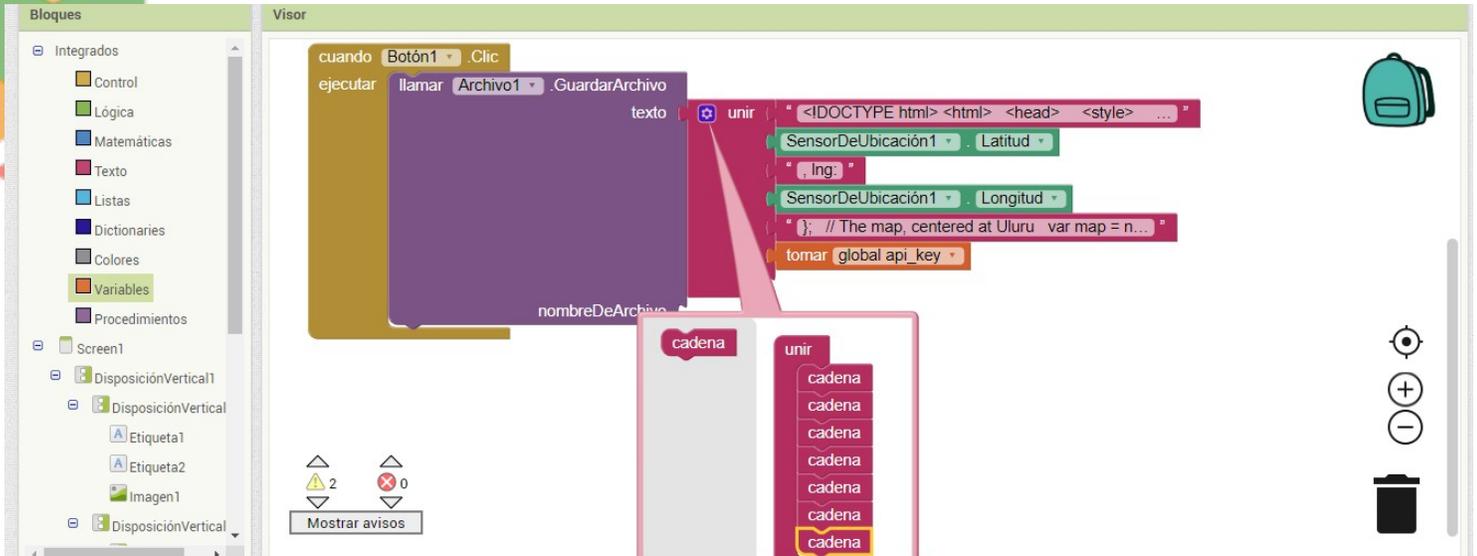
\*Es momento de utilizar la variable `ap_key` agregada al inicio. Para ellos vamos a integrados variables y seleccionamos la opción **tomar**

The screenshot shows the Scratch IDE interface. On the left, the 'Bloques' (Blocks) palette is open, showing the 'Integrados' (Integrated) category with 'Variables' selected. The main workspace, labeled 'Visor', contains a script starting with 'inicializar global nombre como', followed by 'ejecutar llamar Archivo1 GuardarArchivo', 'tomar Devuelve el valor de esta variable.', 'poner a', 'inicializar local nombre como', 'en', 'inicializar local nombre como', 'en', and finally a 'unir' block containing HTML and JavaScript code for a map centered at Uluru.

\*seleccionamos la variable que agregamos

The screenshot shows the Scratch IDE interface. On the left, the 'Bloques' (Blocks) palette is open, showing the 'Integrados' (Integrated) category with 'Variables' selected. The main workspace, labeled 'Visor', contains a script starting with 'cuando Botón1 .Clic', followed by 'ejecutar llamar Archivo1 GuardarArchivo', 'texto nombreDeArchivo', a 'unir' block containing HTML and JavaScript code for a map centered at Uluru, and finally a 'tomar global api\_key' block.

\*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:

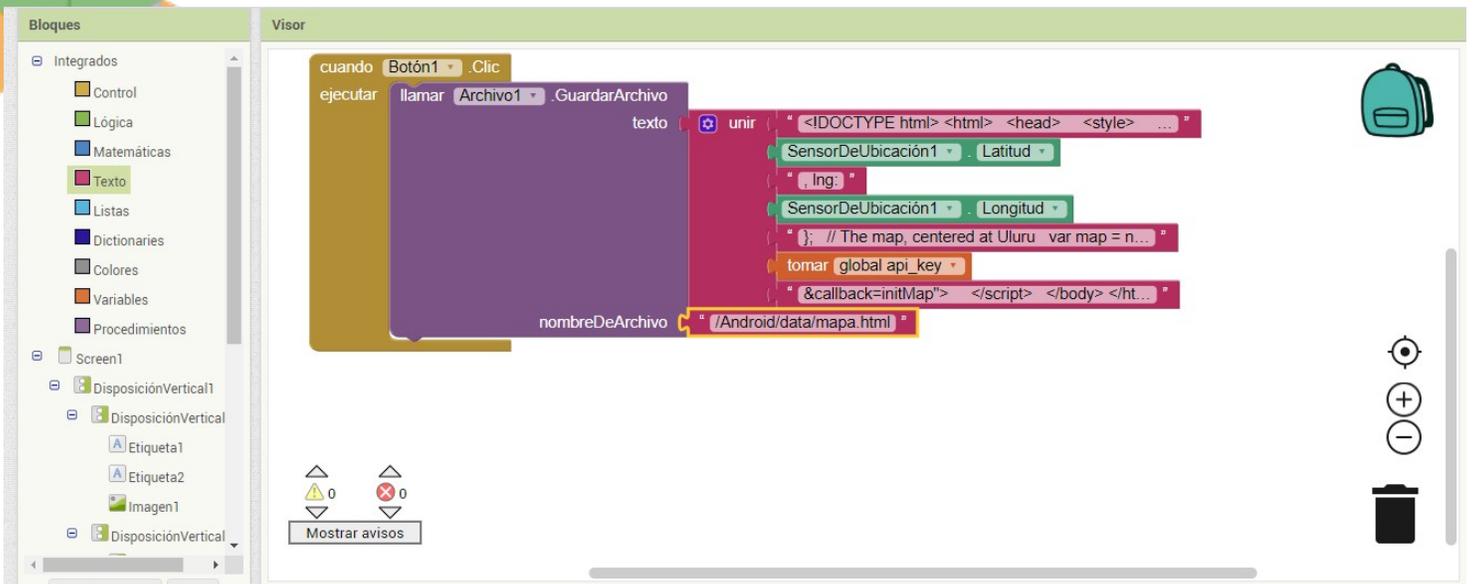
The screenshot shows the Scratch IDE interface. On the left, the 'Bloques' panel is visible with categories like 'Integrados', 'Control', 'Lógica', 'Matemáticas', 'Texto', 'Listas', 'Dictionaries', 'Colores', 'Variables', and 'Procedimientos'. The 'Visor' panel shows a script starting with 'cuando Botón1 .Clic' and 'ejecutar'. The main script consists of a 'llamar Archivo1 .GuardarArchivo' block with a 'texto' field. The 'unir' block contains the following HTML code: '<!DOCTYPE html> <html> <head> <style> ...', 'SensorDeUbicación1 . Latitud', ', lng:', 'SensorDeUbicación1 . Longitud', and a comment '// The map, centered at Uluru var map = n...'. It also includes 'tomar global api\_key' and '&callback=initMap'.

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

The screenshot shows the same Scratch IDE interface as above. The script is identical, but a small white text box is now visible in the 'texto' field of the 'llamar Archivo1 .GuardarArchivo' block, indicating that a blank space has been added for saving the HTML file.



\*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) blocks expanded. The main workspace shows a script for a button click event. The script consists of the following blocks:

- cuando Botón1 .Clic (when button 1 is clicked)
- ejecutar (do)
- llamar Archivo1 .GuardarArchivo (call Archivo1's save file block)
- texto (text) block containing:
  - unir (join) block with the following code:

```
<IDOCTYPE html> <html> <head> <style> ...  
SensorDeUbicación1 . Latitud  
, lng:  
SensorDeUbicación1 . Longitud  
}; // The map, centered at Uluru var map = n...  
&callback=initMap"> </script> </body> </ht...
```
- nombreDeArchivo (filename) block containing: "/Android/data/mapa.html"
- llamar VisorWeb1 .IrALaUrl (call VisorWeb1's go to URL block)
- url (url) block containing: "/Android/data/mapa.html"

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

This screenshot shows the same Scratch IDE script as above, but with an update to the URL block. The 'url' block now contains 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.

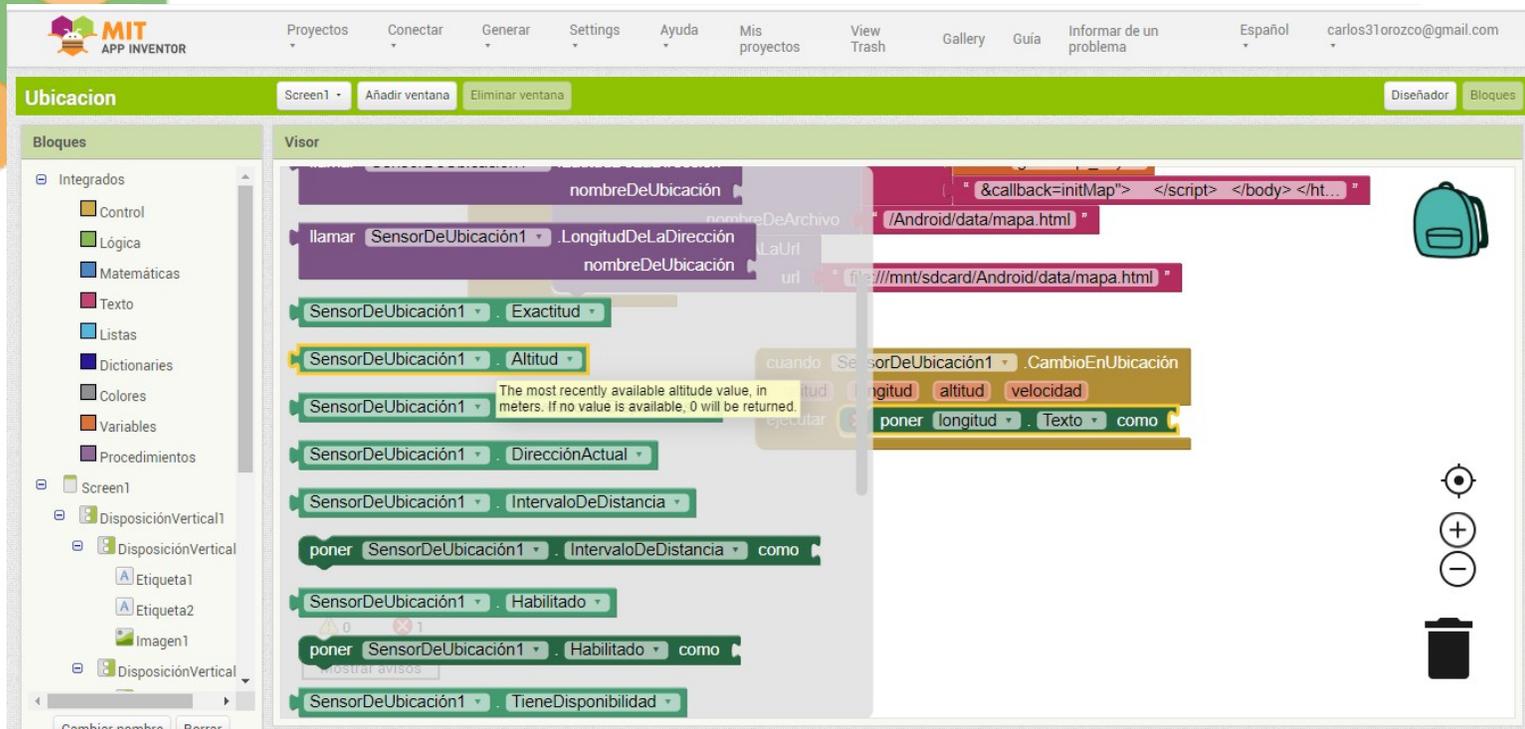
\*para ello debemos seleccionar el elemento SensorDeUbicacion seleccionamos la opción **Cuando SensorDeUbicacion .CambioEnUbicacion** ejecutar

The screenshot shows the MIT App Inventor interface. On the left, the 'Bloques' panel lists various components like 'Etiqueta1', 'Imagen1', and 'SensorDeUbicacion1'. The main workspace shows a visual programming block for 'SensorDeUbicacion1 . CambioEnUbicacion'. This block is connected to a 'llamar' block that calls 'SensorDeUbicacion1 . LatitudDeLaDirección' and 'SensorDeUbicacion1 . LongitudDeLaDirección'. Below these are blocks for 'Exactitud', 'Altitud', and 'ProveedoresDisponibles'. The visualizer shows a map centered on Uluru with a red location pin.

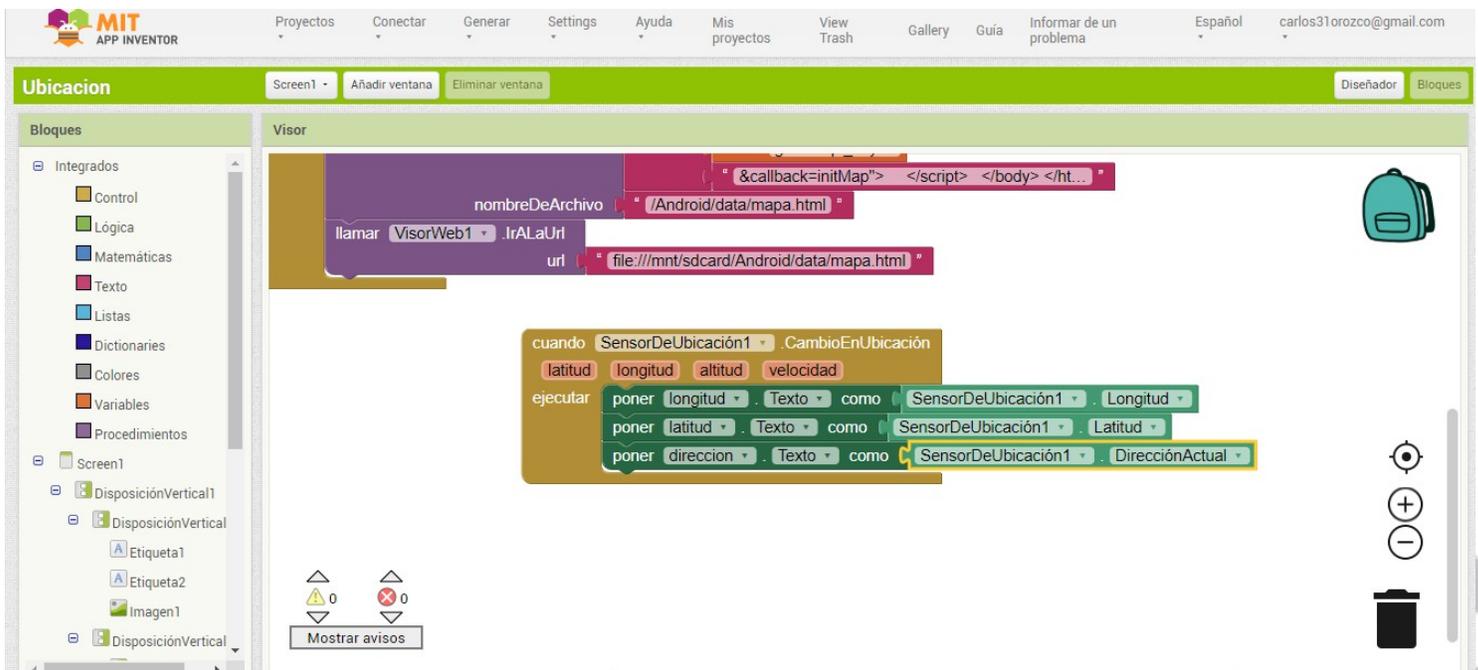
\*Seleccionamos la etiqueta longitud y seleccionamos la opción **poner longitud .texto** como

The screenshot shows the MIT App Inventor interface. On the left, the 'Bloques' panel lists various components like 'Control', 'Lógica', 'Matemáticas', 'Texto', and 'Etiqueta1'. The main workspace shows a visual programming block for 'poner longitud . Texto' connected to a 'SensorDeUbicacion1 . CambioEnUbicacion' block. The visualizer shows a text label with the word 'longitud'.

\*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:

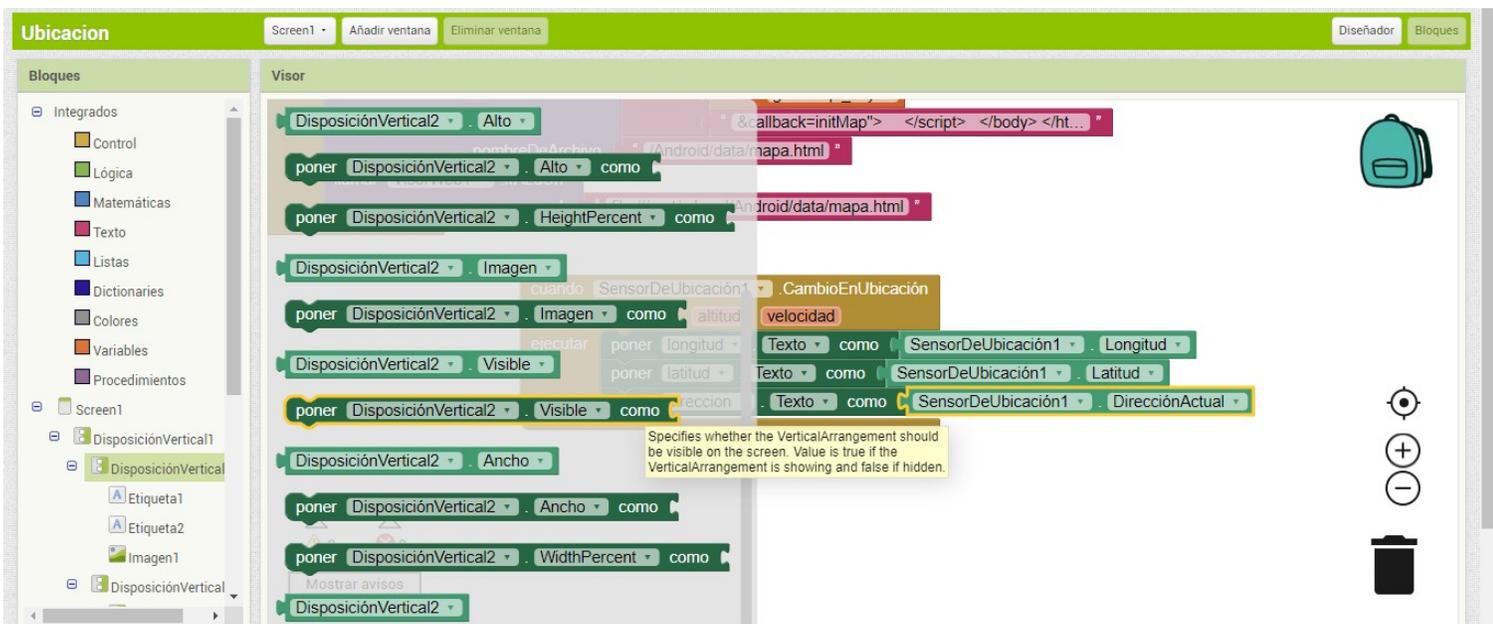


\*Cuando encuentre cambio de dirección, va a poner la disposición donde tiene tu nombre y dibujo del mapa lo va a deshabilitar y va a mostrar la disposición donde se encuentra el botón para ver el mapa de ubicación

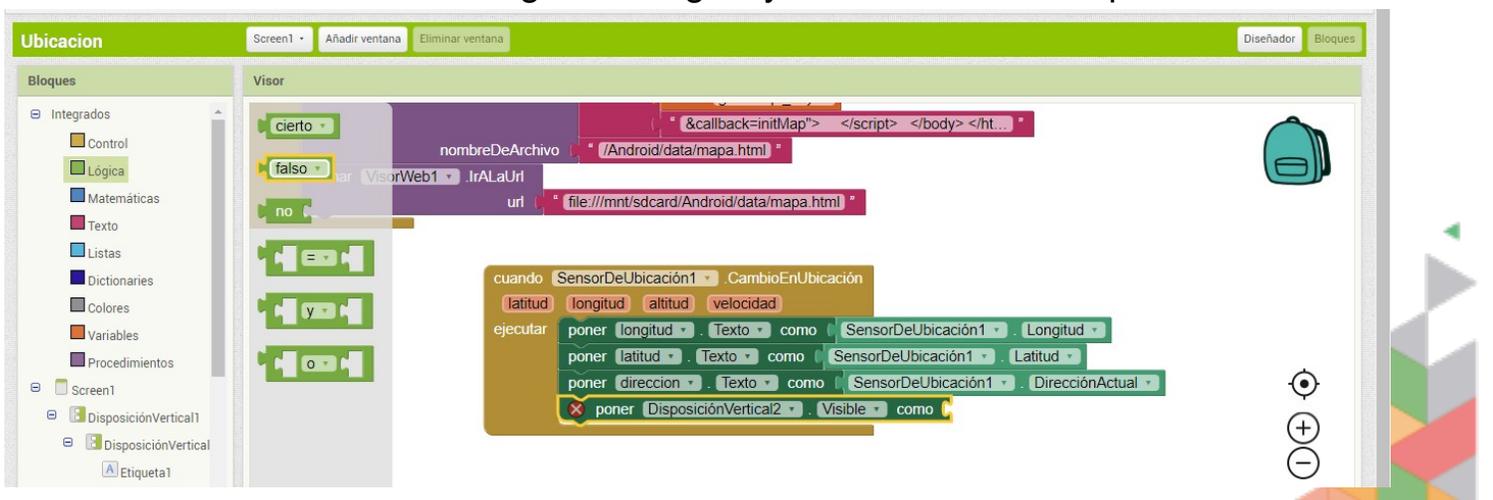
Para ello nos vamos a diseño para verificar que la DisposicionVertical2 sea donde contiene tu nombre y la DisposicionVertical3 sea donde contiene los elementos de ver mapa con datos actuales.

\*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:

Ubicación

Screen1 Añadir ventana Eliminar ventana Diseñador Bloques

Bloques

Integrados

- Control
- Lógica
- Matemáticas
- Texto
- Listas
- Diccionarios
- Colores
- Variables
- Procedimientos

Screen1

- DisposiciónVertical1
- DisposiciónVertical
- Etiqueta1
- Etiqueta2
- Imagen1
- DisposiciónVertical

Visor

```
<script> </script> </body> </ht...>
```

nombreDeArchivo "/Android/data/mapa.html"

llamar VisorWeb1 . IrALaUrl

url "file:///mnt/sdcard/Android/data/mapa.html"

cuando SensorDeUbicación1 . CambioEnUbicación

latitud longitud altitud velocidad

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

Mostrar avisos

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

