**DESIGN OF AN APPLICATION FOR SCANNING BARCODES IN APP INVENTOR.**

***Version 1.0***

**Produced by: Carlos Fernando Infante**

# OBJ**ECTIVE**

To be able to use the scanner in a space that contains products and thus have control over the acquired elements.

# SCOPE

To determine the number of elements that a warehouse, laboratory, etc. can possess, making it easier to track products.

# DESIGN SPECIFICATIONS AND RESTRICTIONS

**Description**

To develop the application, the platform created by MIT was used. From there, various elements within the platform were used to create a connection with the barcode scanner and the TinyDB database. Products can be added either by scanning the barcode or QR code, and they are stored in a database. Their respective codes and categories to which each element was assigned can be viewed.

**Tools**

The elements used to create the application are quite simple to use. You just need to have some knowledge of general programming logic and become familiar with the App Inventor platform.

**Note:** The final design of this application includes the options to add, view products and scan. It should be noted that this is a first version and it is tentative to continue with more elaborated versions made in a better way.

# STEP-BY-STEP VISUAL INTERFACE

1. Go to the app inventor MIT page through Google by typing app inventor on www.google.com.

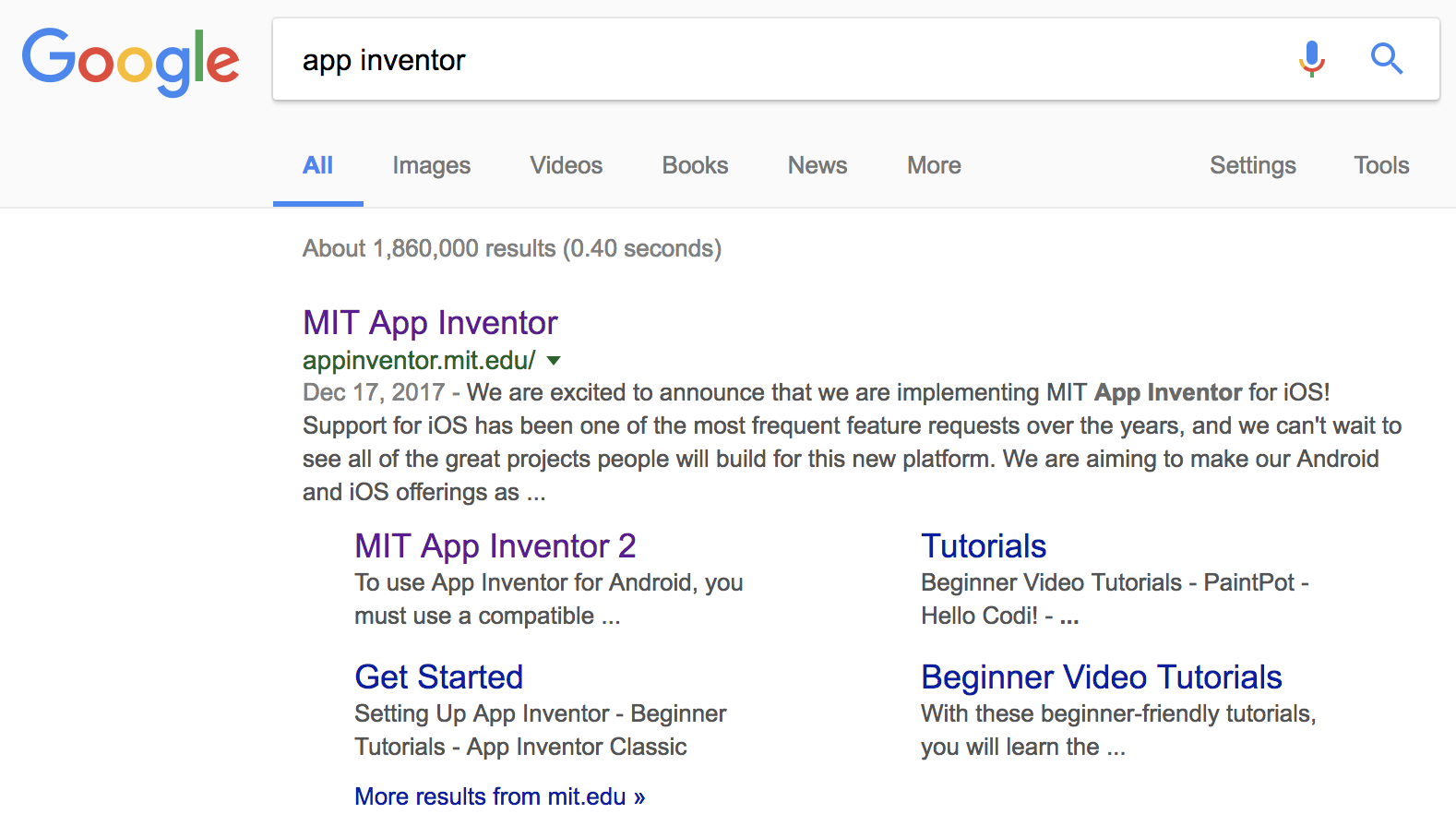


FigurE 1: Accessing app inventor from Google

1. Select the orange button in the upper right corner shown below to create apps.



Figure 2: Button to create apps in the upper right corner.

1. Create a new project, which in this case is called QReader



Figure 3: Steps to create a new project.

1. Within the project, you can see the screen of a cellphone as a visual aid, and on the left, the elements to design the app.



Figure 4: Steps to create a new project.

1. On the left side of the screen, there are several creation options. First, use the layout, which are like containers on which labels, buttons, etc. can be placed.



Figure 5: Different tools of App Inventor.

1. Inside a vertical panel/layout, place a label, which only contains text, and an image picker in the Media section to add a photo to the project. Then, place a small layout, but in this case horizontal, where the buttons to add, view products, and exit will go. All of this goes in a layout called layout\_principal.

It should be added that in addition to creating all of this, it is also important to rename each element. For example, the button to add can be called btn\_add.



Figure 6: Main screen with layout\_principal

All layouts are going to be created on the same screen, so that the implementation of the interface (visual part of the app) is facilitated. This is why the others will be hidden while the main one is open.

1. Next, create the panel or layout\_leer\_codigos. To make this work, it is necessary to download a separate scanner application since the platform is not compatible with those that do not have their own scanner.



Figure 7: Main screen with layout\_principal

1. Now, create the layout\_ver\_productos, where there will be an option to know all the elements that have been saved in the database, as well as their category and code belonging to the barcode. They are added in a table layout, where there can be rows and columns.



Figure 8: Main screen with layout\_ver\_productos.

1. Finally, we have the vertical layout which will contain the labels and text spaces for the product and entering a category. Below it, enter another horizontal layout and within it.



Figure 9: Main screen with layout\_add

Note that this is the way in which the characteristics of a product are going to be determined when it is inside the database. Thus, it will be added with the product name and the category to which that element belongs.

There is also a section for scanning and returning to the main page. To enter this part of the application, the add button is pressed on the main interface. Then, the desired object can be scanned because an application linked to QReader opens, which in this case is Barcode Scanner by ZXing Team.



Figure 10: Layout\_read\_codes for scanning products

# STEP BY STEP APP INVENTOR CODE

1. The following section shows the creation of the main screen screen1 that calls an empty list that is designed to hold the products. Also, the creation of two variables, one for reading the barcode or QR code and the list of product names that is initialized as an empty list.



Figure 11: Main screen, scanner reading variable and empty product list.

1. As shown in the interface explanation section, there are several buttons that in this case will be implemented with code so that they can be functional. There are several functions; the exit button is used to completely close the application, the add button is to be able to open another section of the screen where the scan button is seen and the current layout is closed, and the return buttons make it go from one layout to the previous one that was on the screen.



Figure 12: Exit, add, and return buttons.

1. Here the scan button is implemented. The App Inventor BarCodeScanner component is called to perform the scanning (DoScan), then the BarcodeScanner application that is located in the Google Play Store is opened with the UseExternalScanner function if the phone does not have a default scanner in the camera. Then the label where the scan result is displayed is disabled and the current layout is closed to put the camera.



Figure 13: Scan button.

1. For the product button, the view\_products layout is opened so that the elements of the database can be viewed. Then the main interface is closed and the elements of the product list are called.



Figure 14: Product button.

1. The save button begins by calling the database, saving the value scanned, and assigning two attributes which are the product name and category; these two are chosen by the user.



Figure 15: Save button.

1. This block of code is for the product list. When one is chosen, the selected element is called, and a search cycle begins; for each name in the database list, it is checked whether the selected name is the same as the one in the list. If so, then that result is returned. If not, then it starts searching for the other elements until it finds the one selected by the user.



Figure 16. Product selection in a list.

1. These blocks are related to scanning and clearing text fields. If the item to be scanned has not been registered in the database, it is added and a message appears saying "This product has already been registered." If what is scanned is already in the database, a message appears saying "This product has already been registered." The other two functions are to clear all text fields and return to the previous screen.



Figure 17: Scan and clear field functions.

1. The next block is intended to clear the text fields where the specifications of a product are located. This is done by calling the element through the database and finding its values.



Figure 18: Clear product names from the list.

1. **CHANGE CONTROL**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHANGE DESCRIPTION** | **DATE** | **VERSION** | **APPROVED BY** |
|  |  |  |  |