

JAVA EN ANDROID

Día 1

Oscar Chávez-Bosquez

<http://chavezbosquez.github.io/diplomado/>

Objetivo

Desarrollar una app para el
registro de estudiantes a partipar
en el

Verano de la Investigación Científica
2022

Contenido



1. Diseñando la app
2. *Live testing*
3. Escribiendo código
4. SQLite
5. Base de datos en la nube
6. Ejecución del proyecto



Contenido



- 1. Diseñando la app**
2. Live testing
3. Escribiendo código
4. SQLite
5. Base de datos en la nube
6. Ejecución del proyecto



App Inventor

- ❑ Proyecto del MIT
- ❑ Versión 2.0
- ❑ Completamente en la 
- ❑ Open source
- ❑ [Requerimientos mínimos](#)
- ❑ Cursos gratuitos:
 - ✓ <http://www.appinventor.org/content/CourseInABox/Intro>
 - ✓ https://codeiot.org.br/courses/course-v1:LSI-TEC+IOT105ES+2021_0C/about

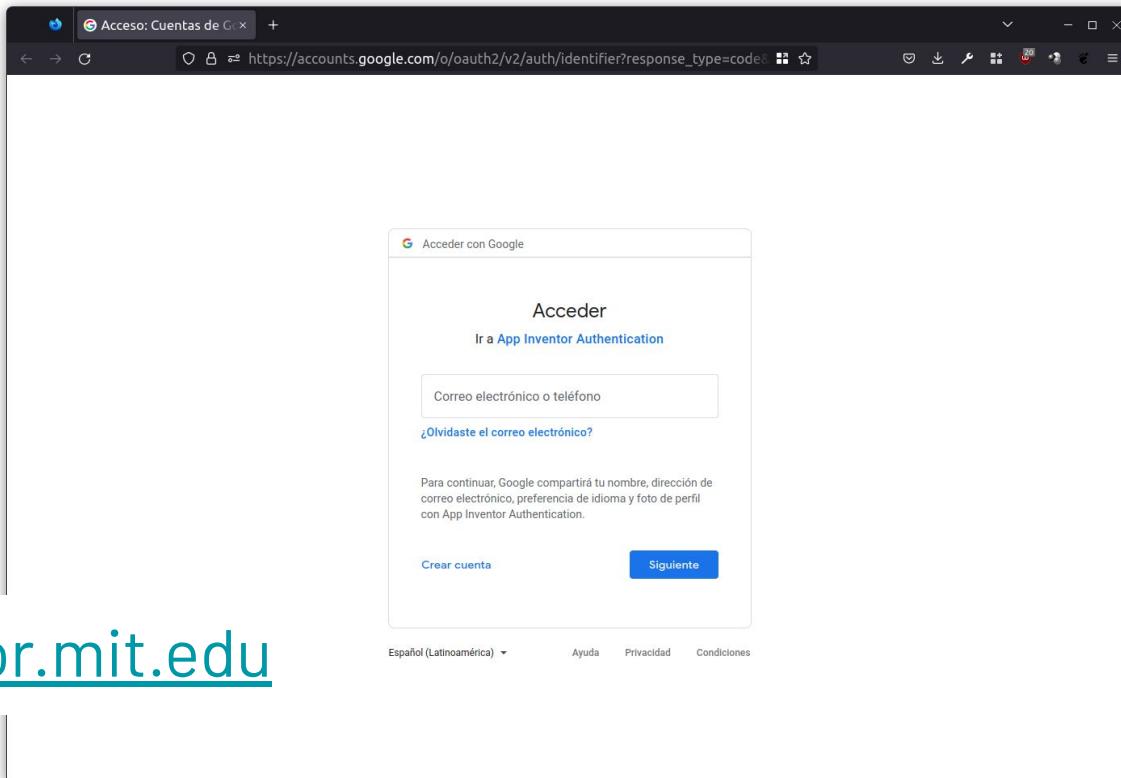
MIT App Inventor 2

Comenzar aquí

The screenshot shows the homepage of the MIT App Inventor 2 website. At the top left is the MIT App Inventor logo featuring a stylized bee icon. To its right is a large orange button labeled "Create Apps!". The top navigation bar includes links for "About", "Educators", "News", "Resources", "Blogs", and a "Give" button. A search bar is located on the far right. The main banner has a purple background with the text "Anyone Can Create iPhone and Android Apps with Global Impact" and a "Get Started" button. Below the banner are four performance metrics: "USERS TODAY: 113.2K", "USERS THIS MONTH: 1.0M", "ALL-TIME USERS: 14.9M", and "APPS BUILT: 67.8M". The bottom of the page features a green decorative footer with various icons related to app development.

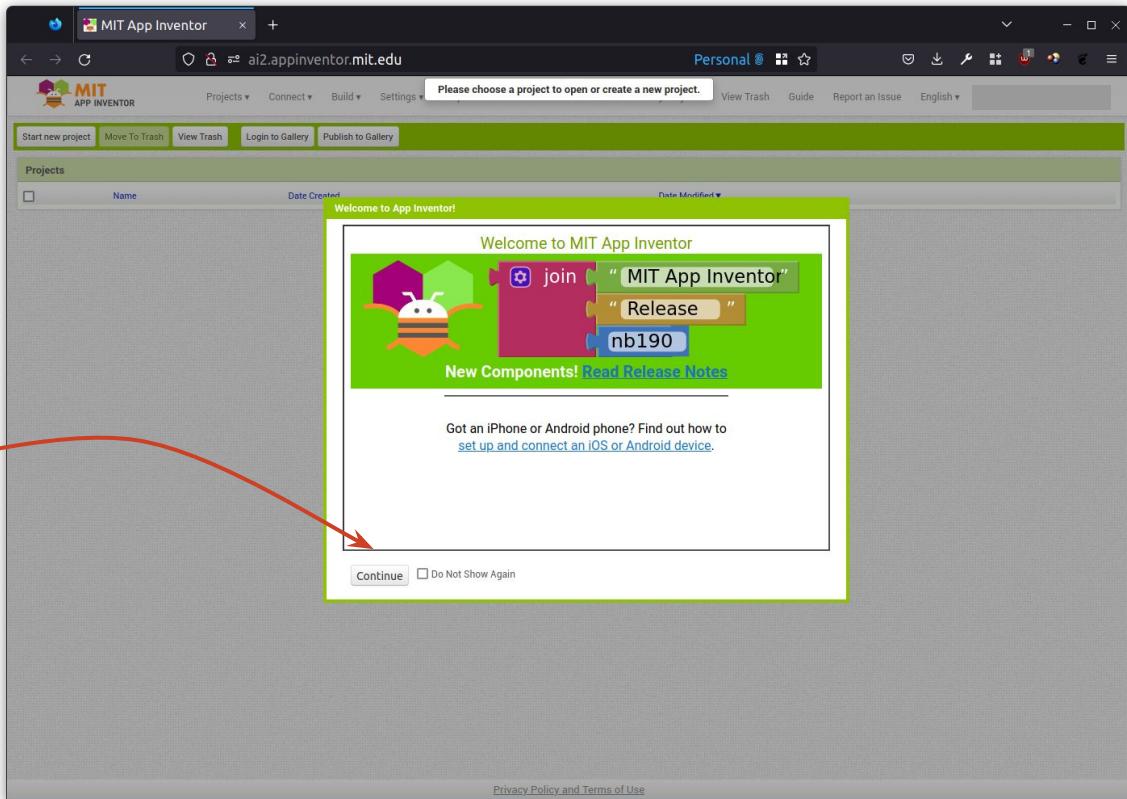
<https://appinventor.mit.edu>

Iniciar sesión



<http://ai2.appinventor.mit.edu>

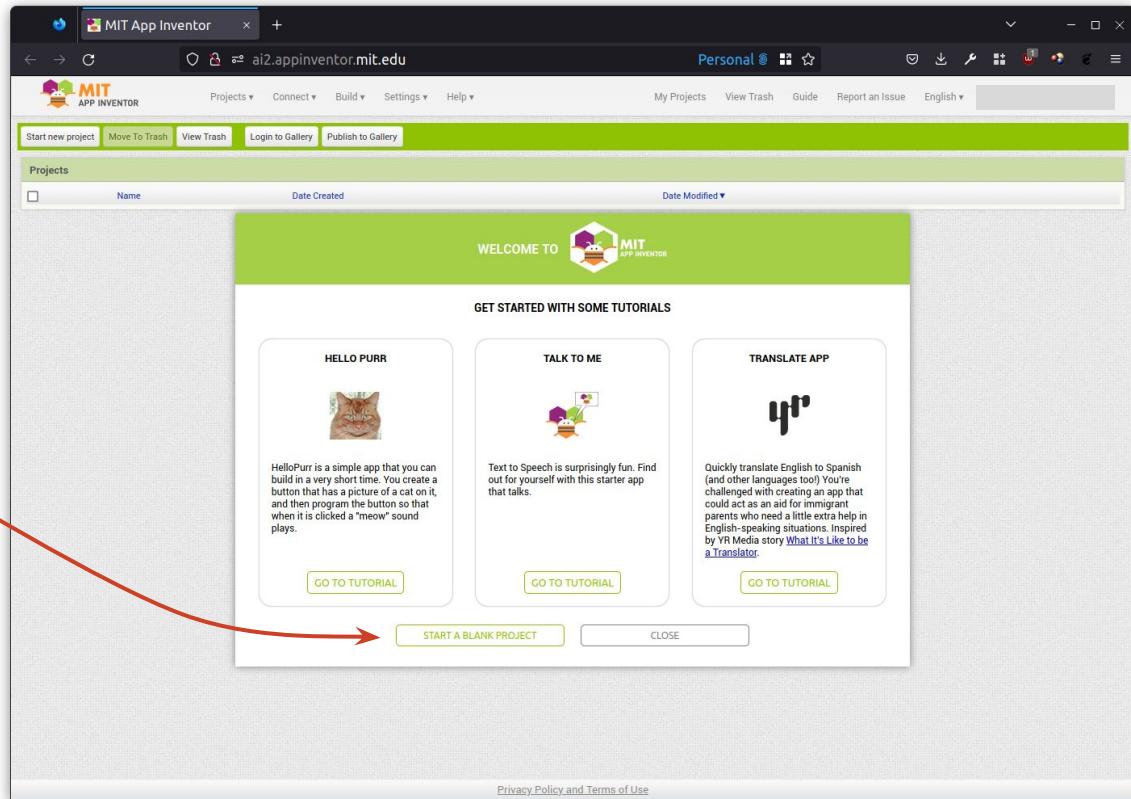
Pantalla de bienvenida



Continuar

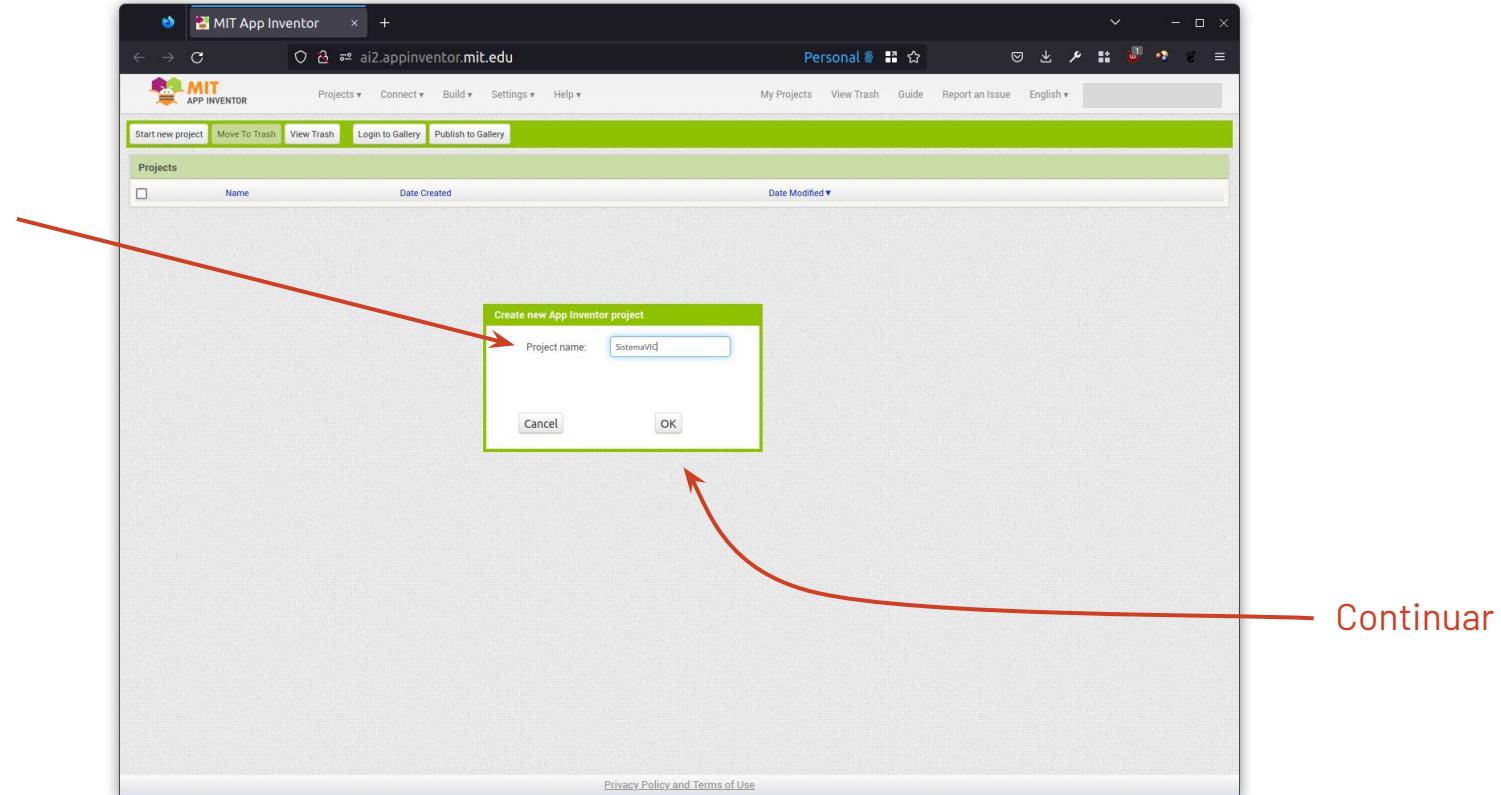
Nuevo proyecto

Iniciar proyecto en blanco



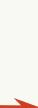
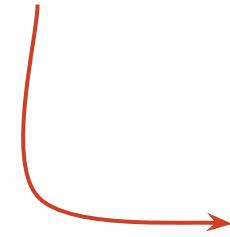
Nombre del proyecto

Nombre del
proyecto



App inventor UI - Modo diseño

Paleta de componentes



Visor

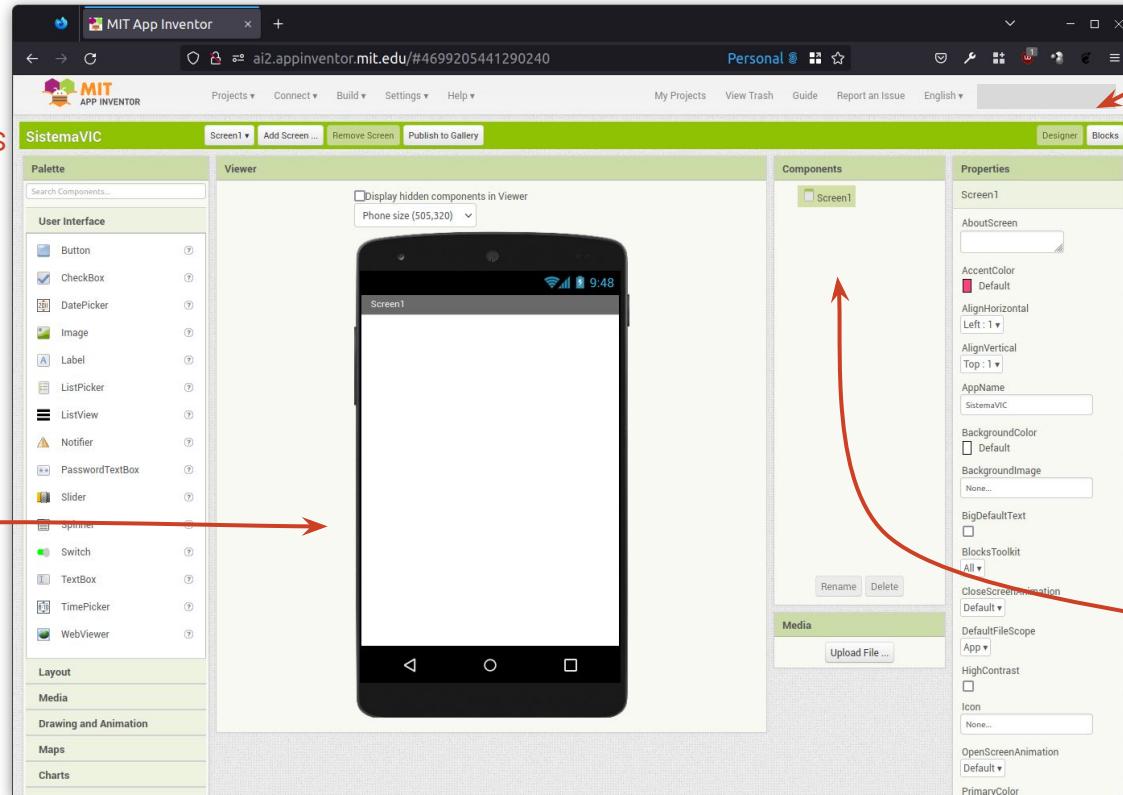
Diseño / Código



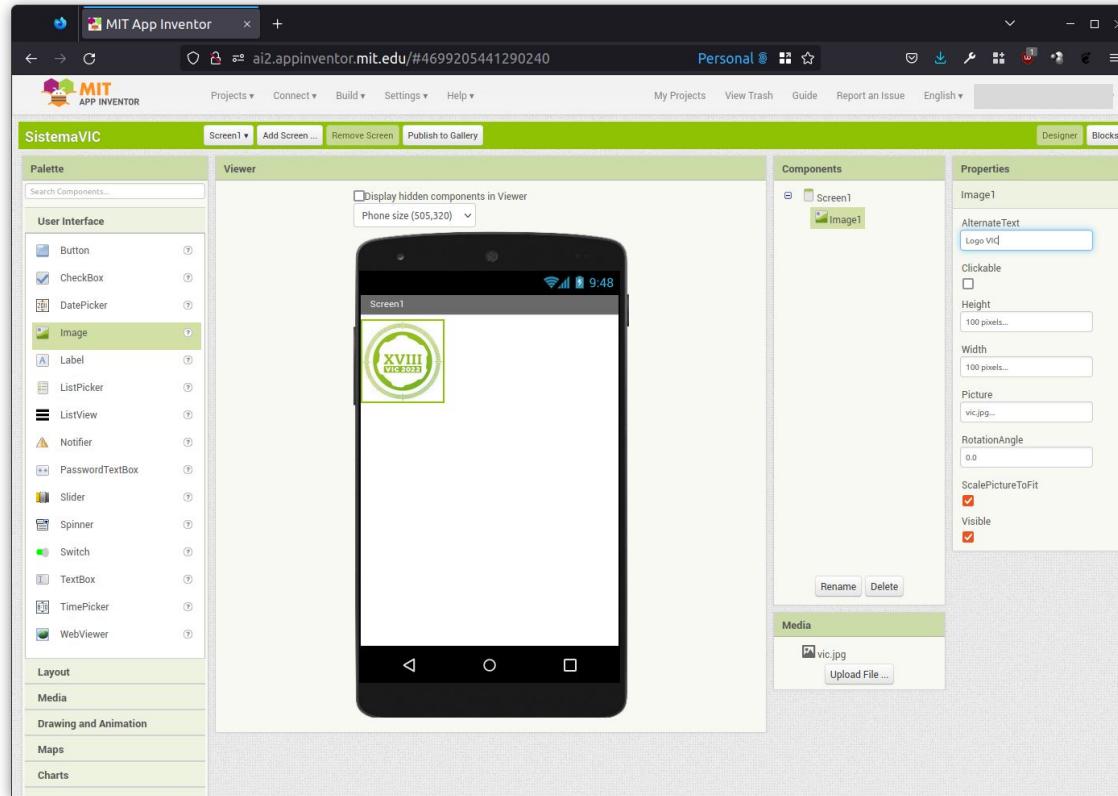
Propiedades



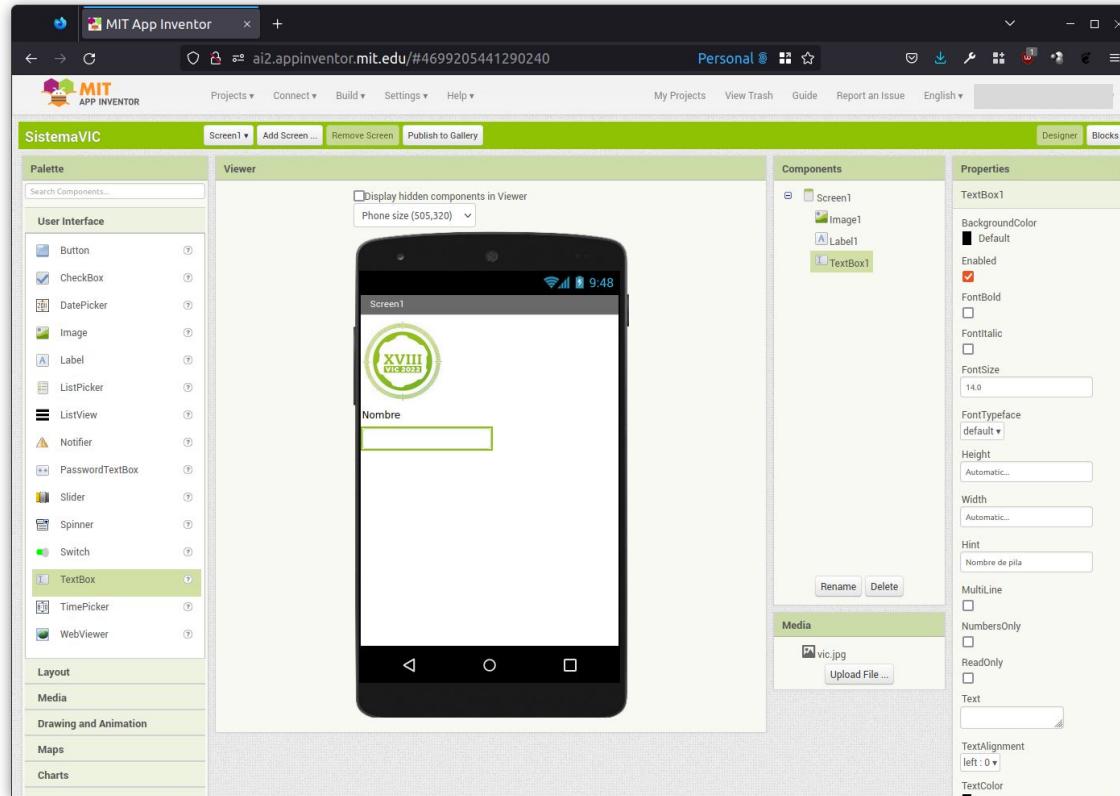
Componentes



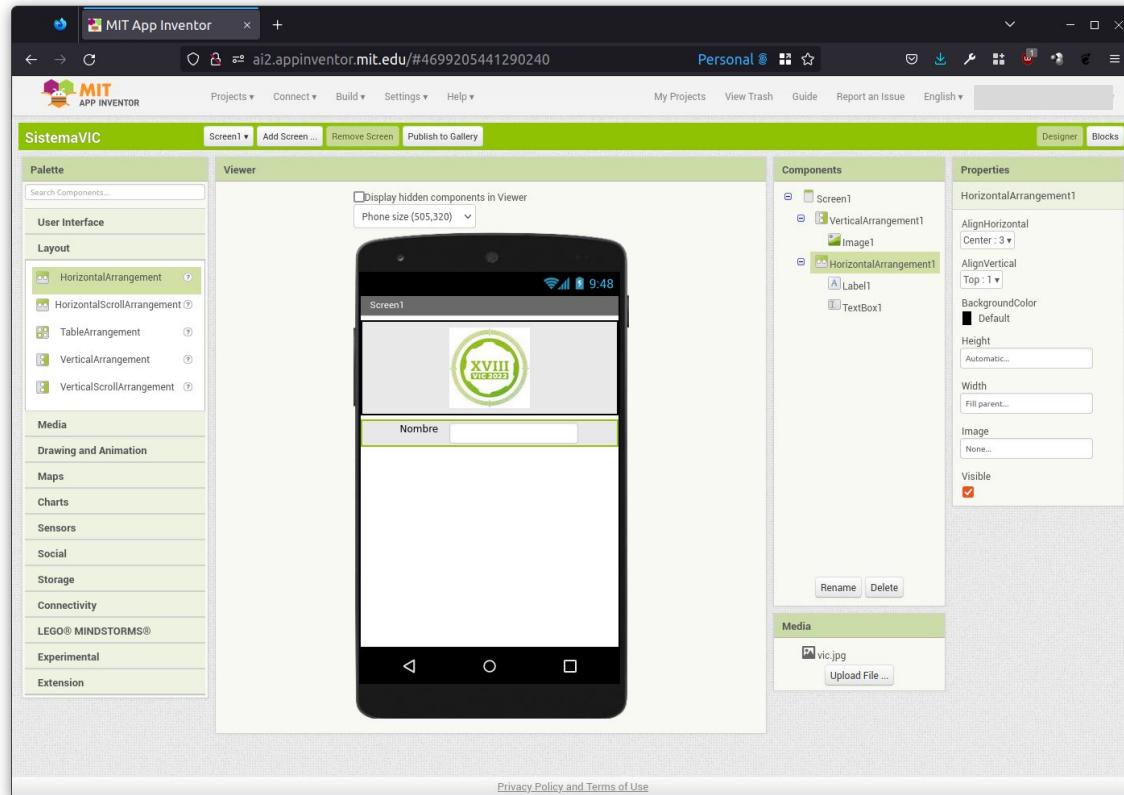
Comenzamos la app agregando una imagen



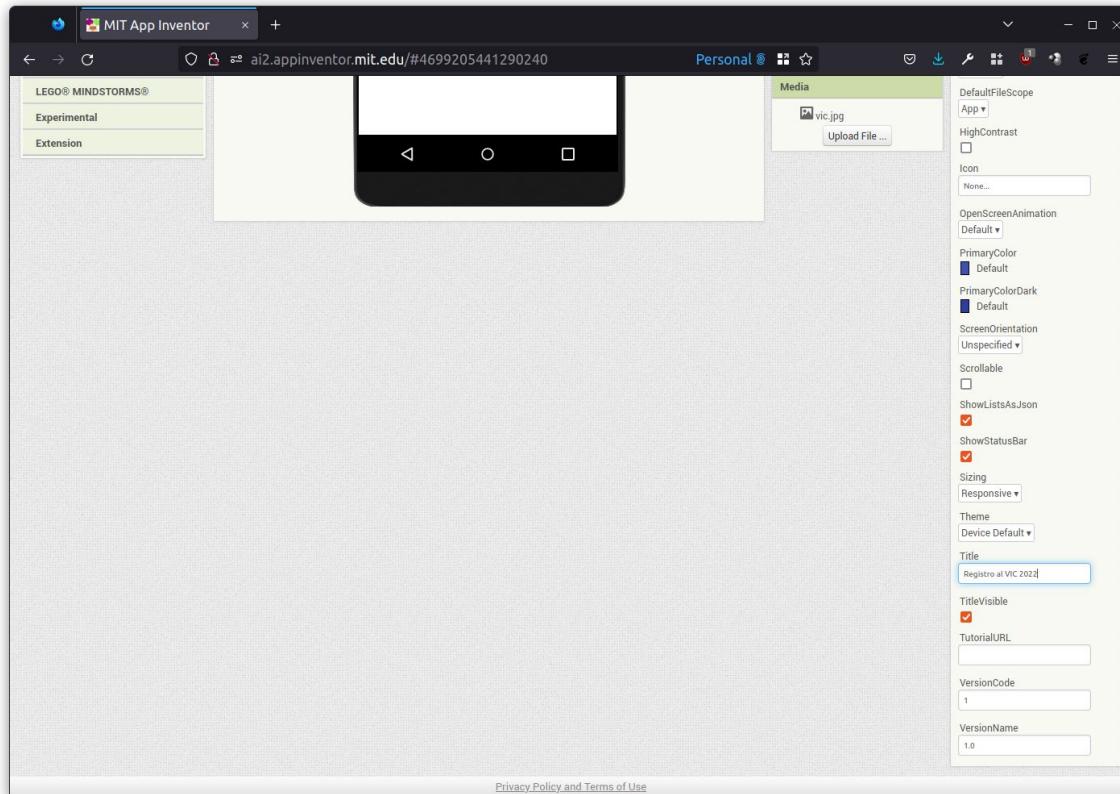
Agregando componentes



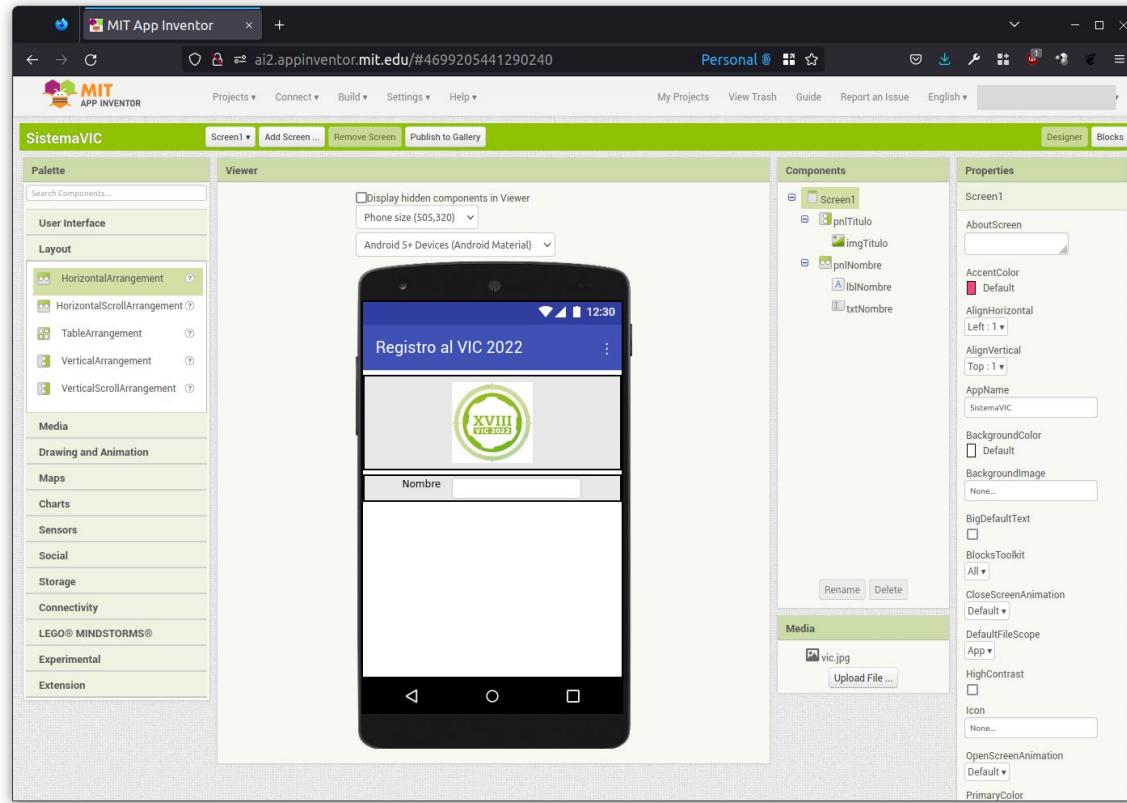
Organizando los componentes



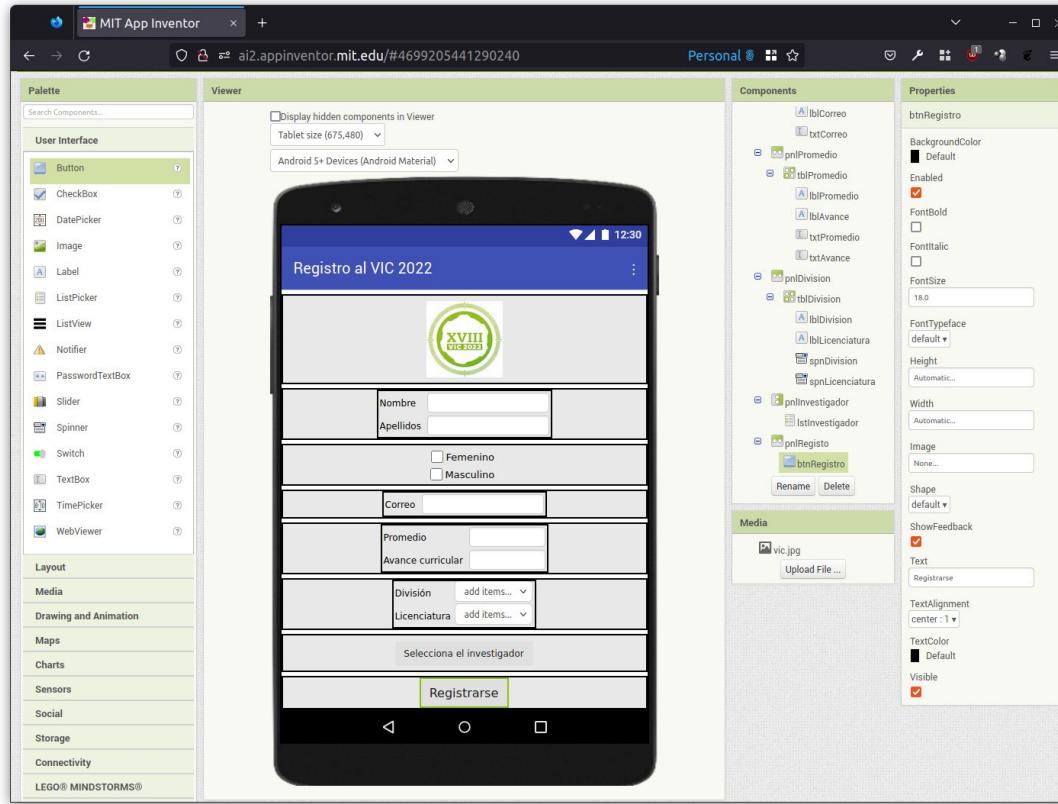
Otras propiedades de la pantalla



iAsignar nombres adecuados!



Mejorando el layout



Contenido



1. Diseñando la app
- 2. *Live testing***
3. Escribiendo código
4. SQLite
5. Base de datos en la nube
6. Ejecución del proyecto



Live testing: <https://appinventor.mit.edu/explore/ai2/setup>

Option One - RECOMMENDED

Test your apps with an iPhone or Android phone and a Wi-Fi connection: [Instructions](#)

If you have a computer, a smartphone, and a Wi-Fi connection, this is the easiest way to create and test your apps. Simply install the MIT App Inventor companion app on your phone and test your apps through a Wi-Fi connection.



Option Two

Test your apps with a Chromebook: [Instructions](#)

Many Chromebooks are capable of running Android apps. That lets you create, test, and run the finished app on the same device.



Option Three

Don't have a device? Test apps with an emulator: [Instructions](#)

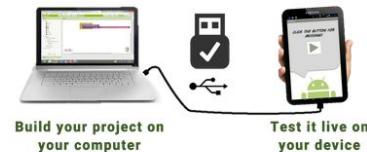
If you don't have a phone or tablet handy, you can still use App Inventor by installing the emulator software on your computer. Have a class of 30 students? Have them work primarily on emulators and share a few devices.



Option Four

No Wi-Fi? Test apps with an Android device and a USB Cable: [Instructions](#)

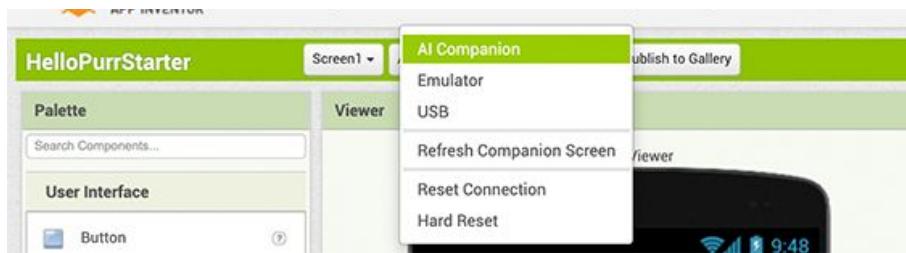
Some firewalls within schools and organizations do not allow the type of Wi-Fi connection required for App Inventor. If Wi-Fi doesn't work for you, try USB (Android only).



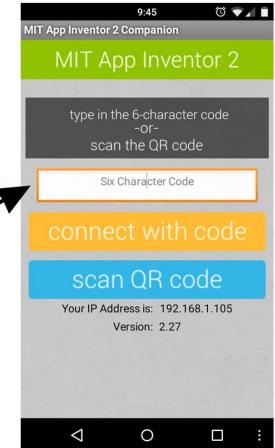
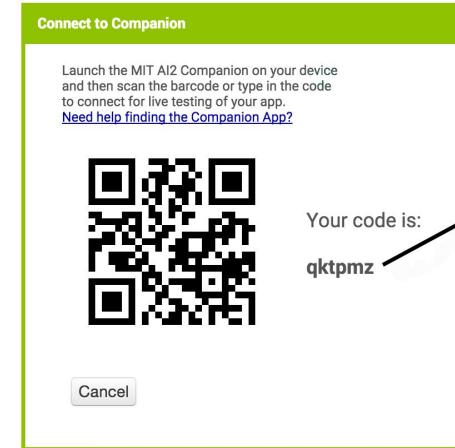
3 maneras de ejecutar la app

AI Companion (<http://appinventor.mit.edu/explore/ai2/setup-device-wifi>):

1. Conectar la PC y el smartphone a la misma WiFi
2. Instalar  MIT AI2 Companion desde la Google Play
3. Desde App Inventor ejecutar en el menú Connect → AI Companion:



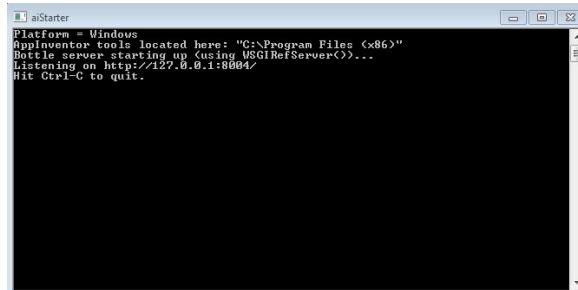
4. Ejecutar la app MIT AI2 Companion y seleccionar Scan QR code:



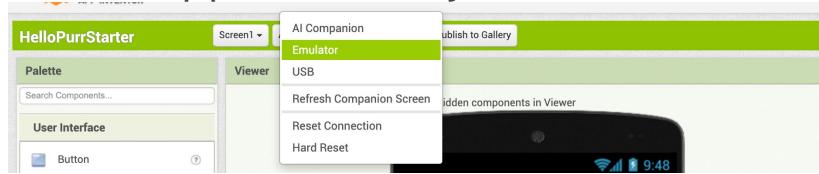
3 maneras de ejecutar la app

Emulator (<https://appinventor.mit.edu/explore/ai2/setup-emulator.html>):

1. Instalar el emulador: https://appinv.us/aisetup_win_30_265.exe (~1.5 GB)
2. Ejecutar *aiStarter*:



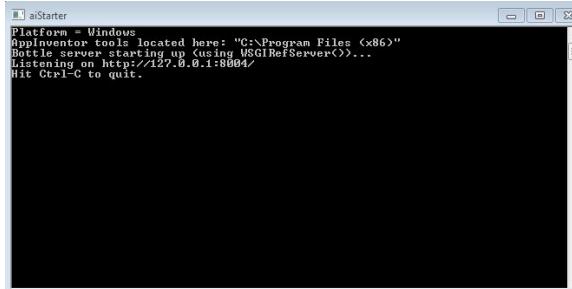
3. ¿Funciona? <http://appinventor.mit.edu/test>
4. Desde App Inventor ejecutar en el menú Connect → AI Companion:



3 maneras de ejecutar la app

USB (<https://appinventor.mit.edu/explore/ai2/setup-device-usb>):

1. Instalar el emulador: https://appinv.us/aisetup_win_30_265.exe (~1.5 GB)
2. Instalar  [MIT AI2 Companion](#) desde la Google Play
3. Ejecutar aiStarter:



4. Configurar smartphone SB Debugging ON
5. ¿Funciona? <http://appinventor.mit.edu/test>

App en ejecución

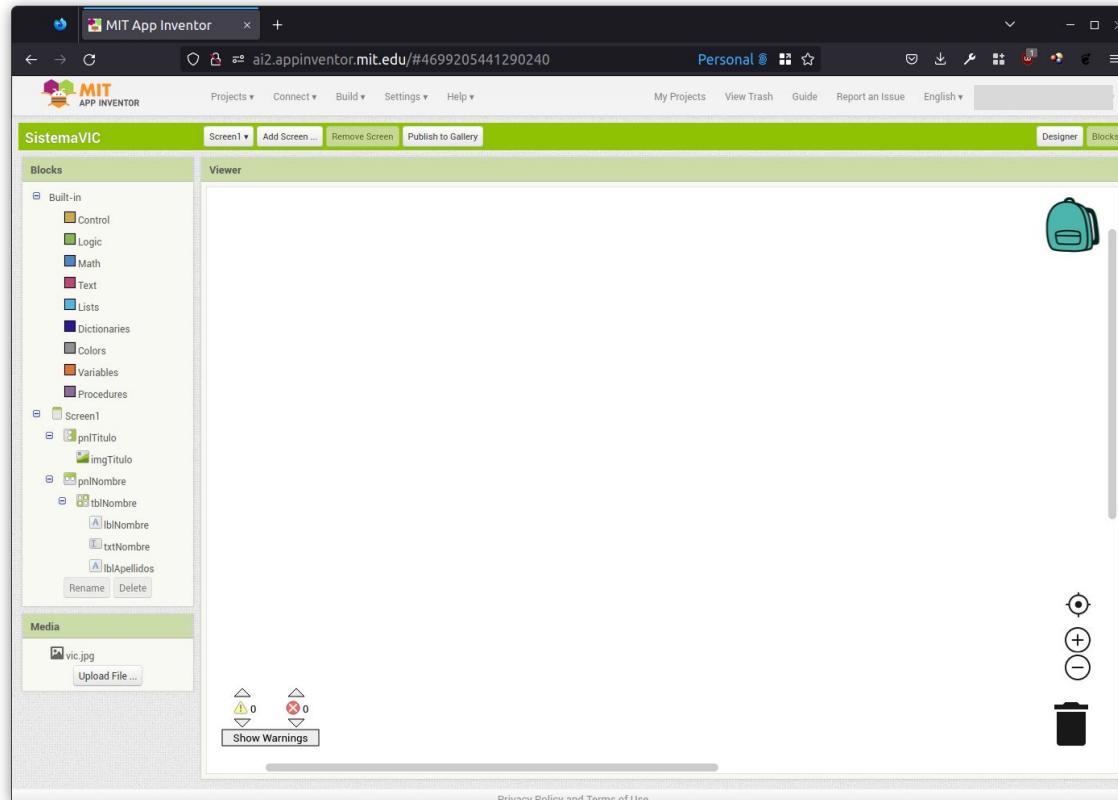
Contenido



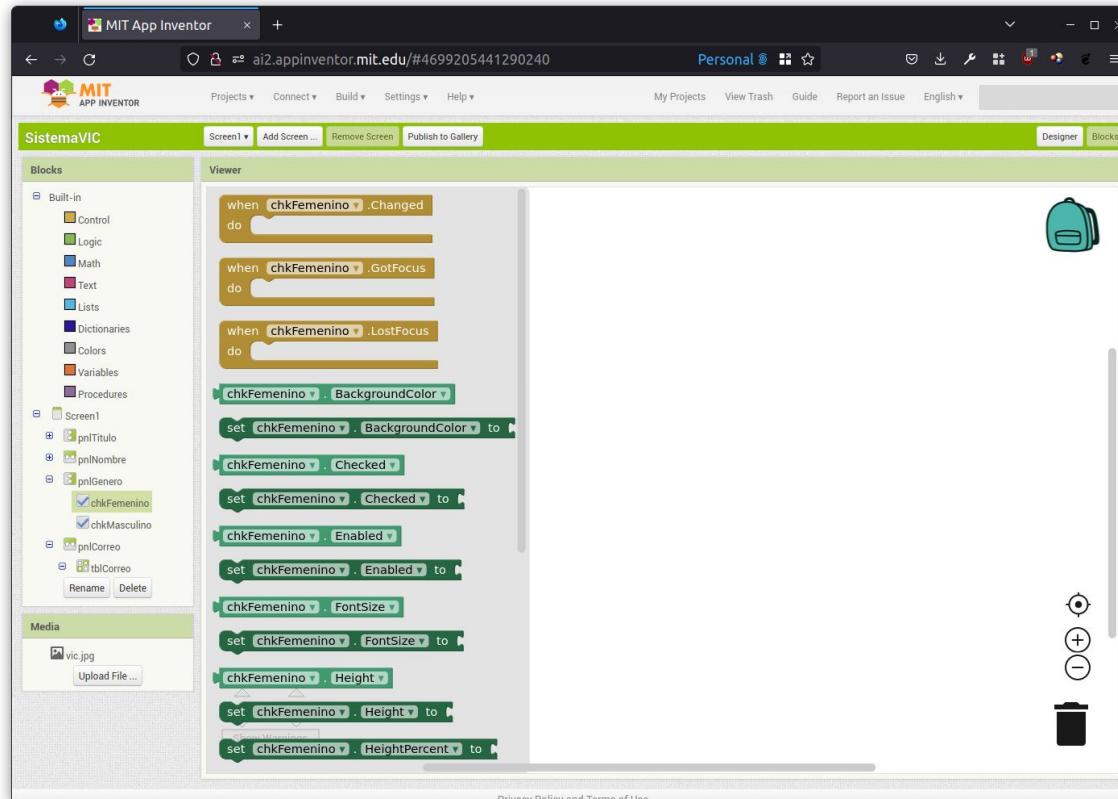
1. Diseñando la app
2. Live testing
- 3. Escribiendo código**
4. SQLite
5. Base de datos en la nube
6. Ejecución del proyecto



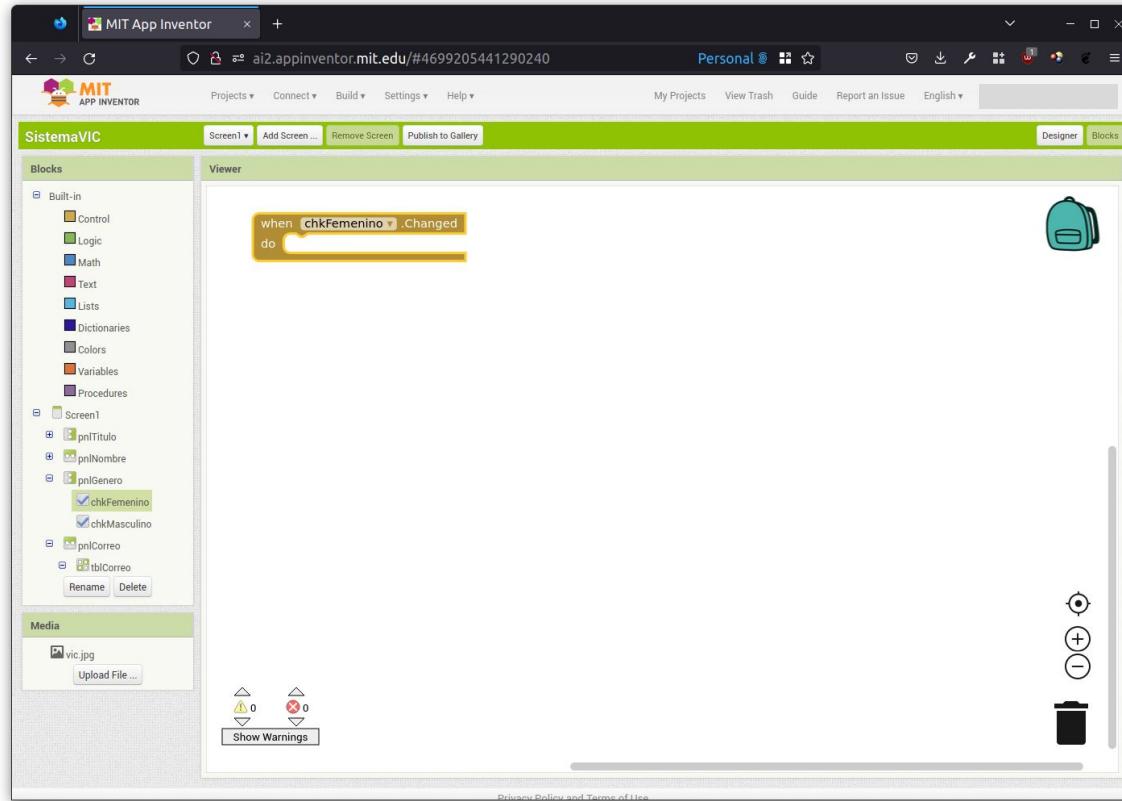
Funcionalidad - Modo bloques



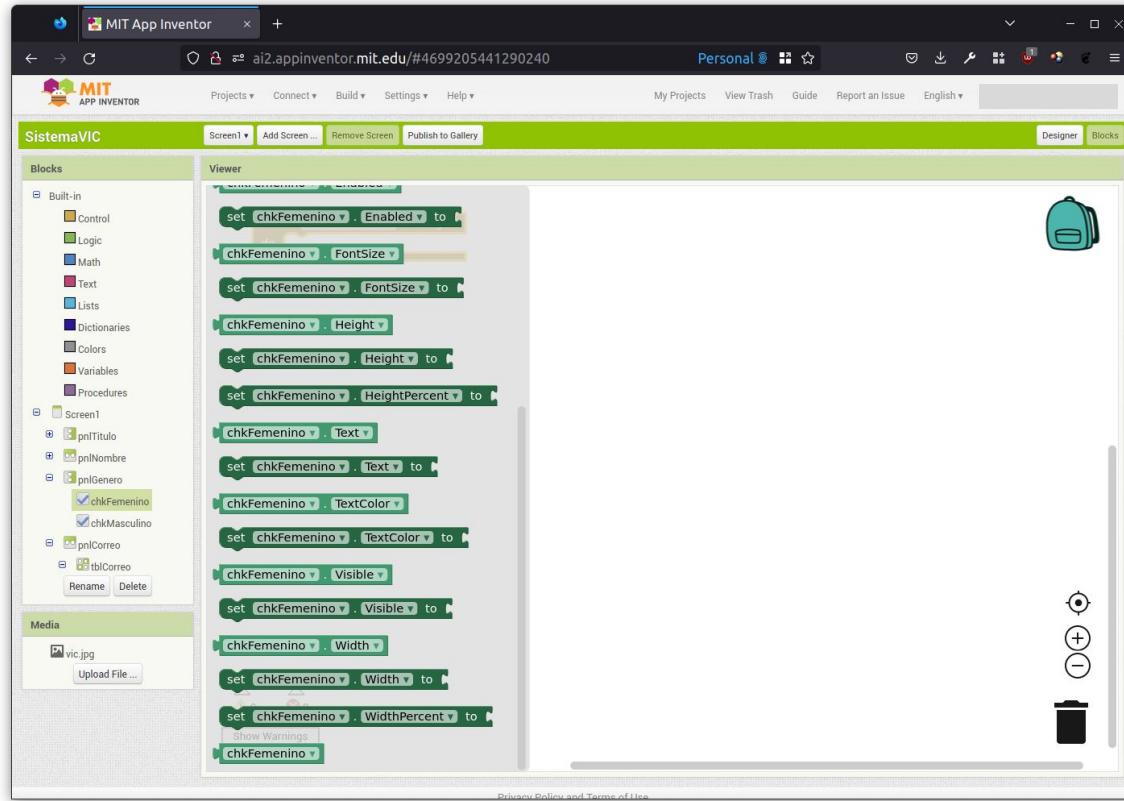
Simulando un radio button en App Inventor



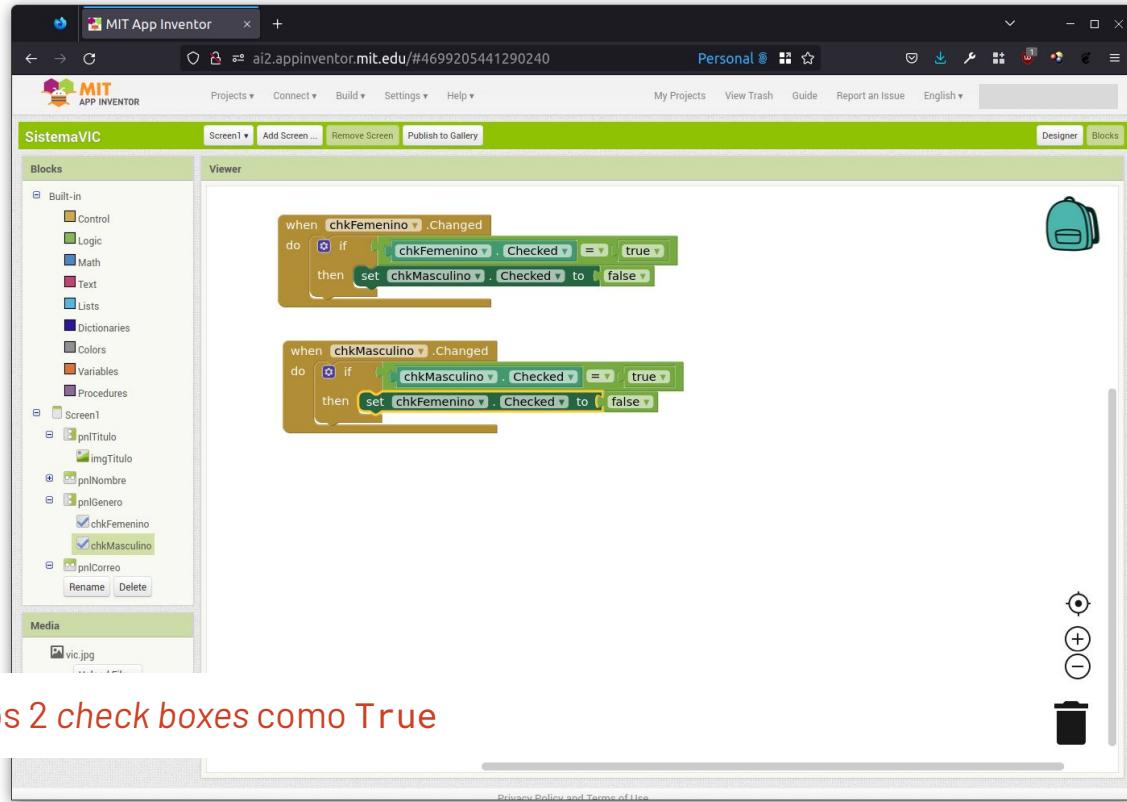
Manejando eventos



Manejando atributos



Funcionalidad tipo *radio button*



Inicializar uno de los 2 *check boxes* como True

Contenido



1. Diseñando la app
2. *Live testing*
3. Escribiendo código
- 4. SQLite**
5. Base de datos en la nube
6. Ejecución del proyecto



Extensión SQLite

<https://github.com/frdfsnlght/aix-SQLite>

The screenshot shows the GitHub repository page for 'frdfsnight/aix-SQLite'. The repository has 63 addtive commits, 22 stars, 3 watching, and 15 forks. It includes sections for Code, Issues, Pull requests, Actions, Projects, Security, Insights, About, Releases, Packages, Contributors, and Languages. The README.md file describes SQLite as a small, fast, self-contained SQL database engine built into Android, used for complex relationships between tables. The Languages section shows Java at 100.0%.

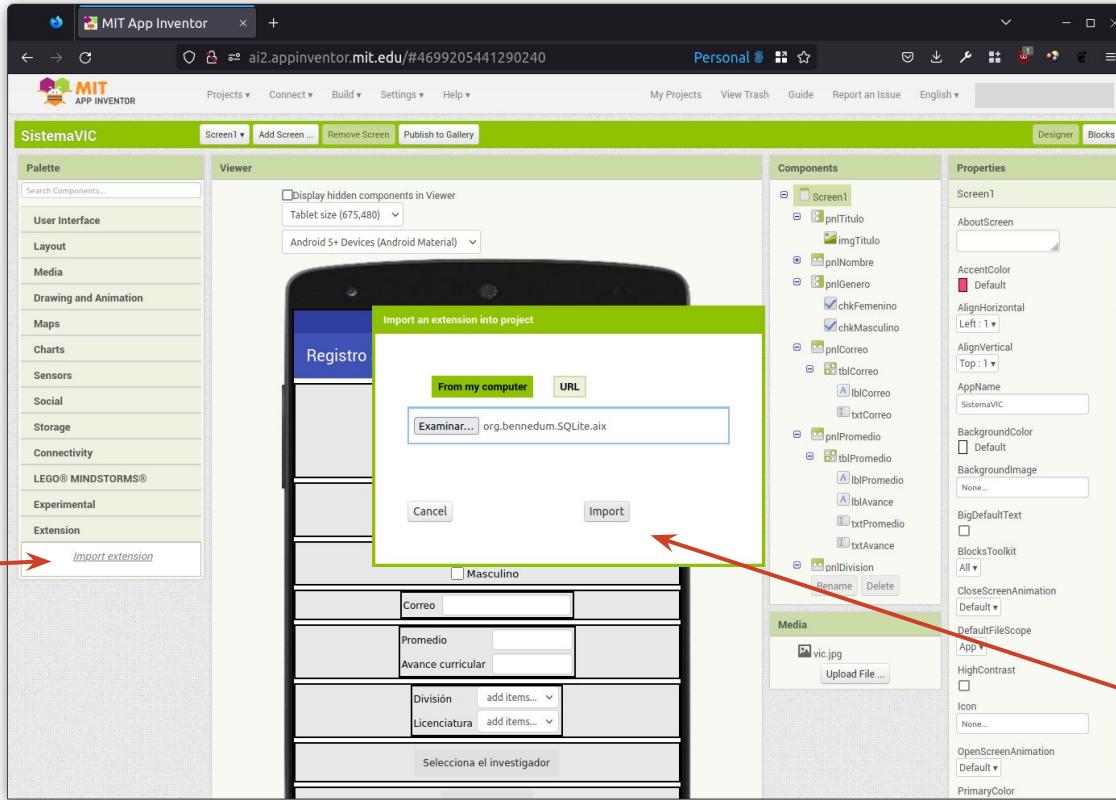
Download

[Download AIX](#)

If you don't see a feature you'd like or need, tell me about it. I can't promise I'll add it but you never know!

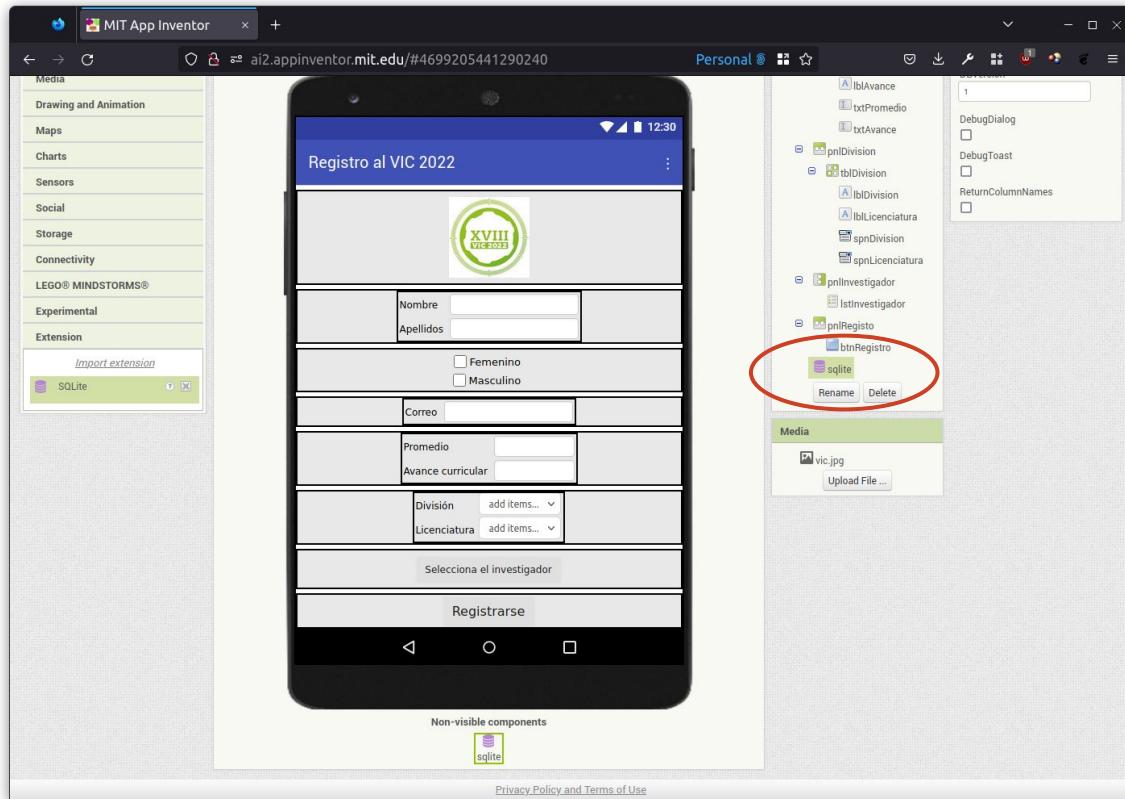
Importar la extensión a nuestro proyecto

Paso 1



Paso 2

Agregando el componente

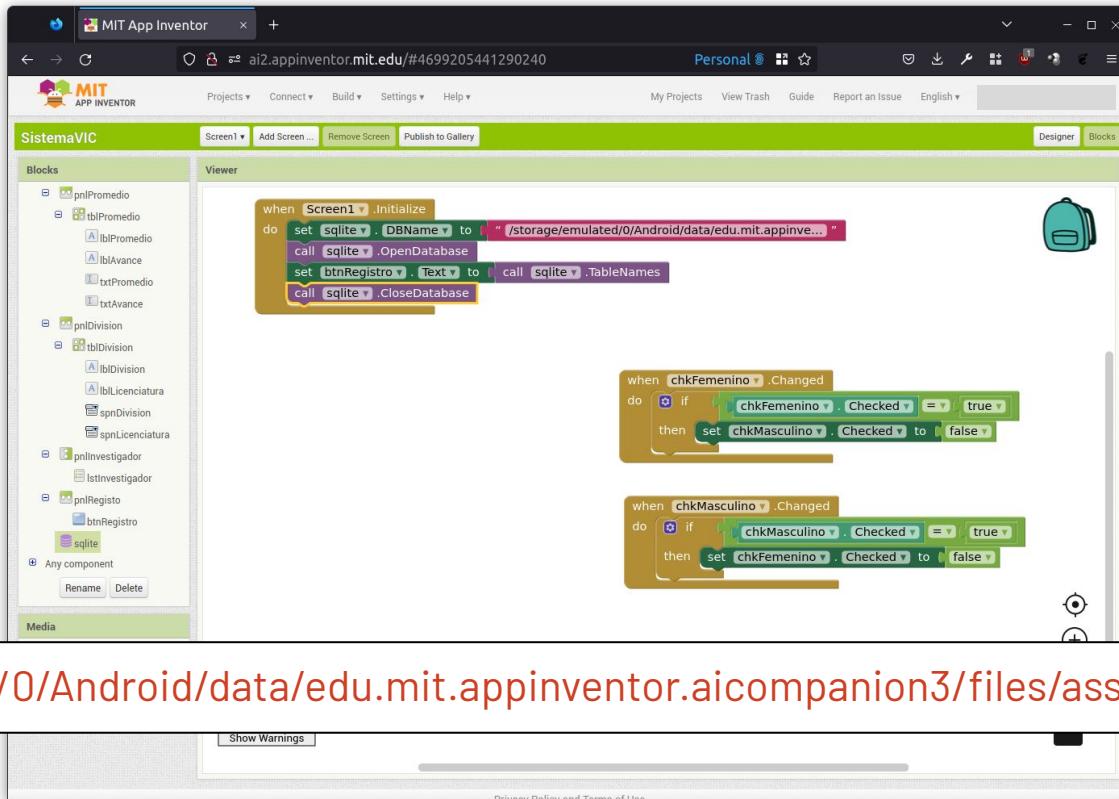


Cargando nuestra BD

The screenshot shows the MIT App Inventor development environment. On the left, the project browser lists various components under the 'Media' category, including 'Drawing and Animation', 'Maps', 'Charts', 'Sensors', 'Social', 'Storage', 'Connectivity', 'LEGO® MINDSTORMS®', 'Experimental', and 'Extension'. A file named 'SQLite' is selected. In the center, a smartphone-style preview window displays a registration form titled 'Registro al VIC 2022'. The form includes fields for Name, Surname, Gender (Female/Male), Email, Grade, and Division/Licenciatura. At the bottom, there's a button labeled 'Registrar'. Below the preview, a section titled 'Non-visible components' shows a single 'sqlite' component. At the bottom of the screen, there are links for 'Privacy Policy and Terms of Use'. On the right, the block editor shows a tree view of components. A red arrow points from the text 'Habilitar el modo depuración' to the 'DebugDialog' component in the tree. A red circle highlights the 'datos_vic.db' file in the 'Media' section of the project browser.

Habilitar el modo depuración

Haciendo una prueba



Testear la app

¿No funciona?

Otras posibles rutas:

/storage/emulated/0/Android/data/edu.mit.appinventor.aicompanion3/files/assets/datos_vic.db

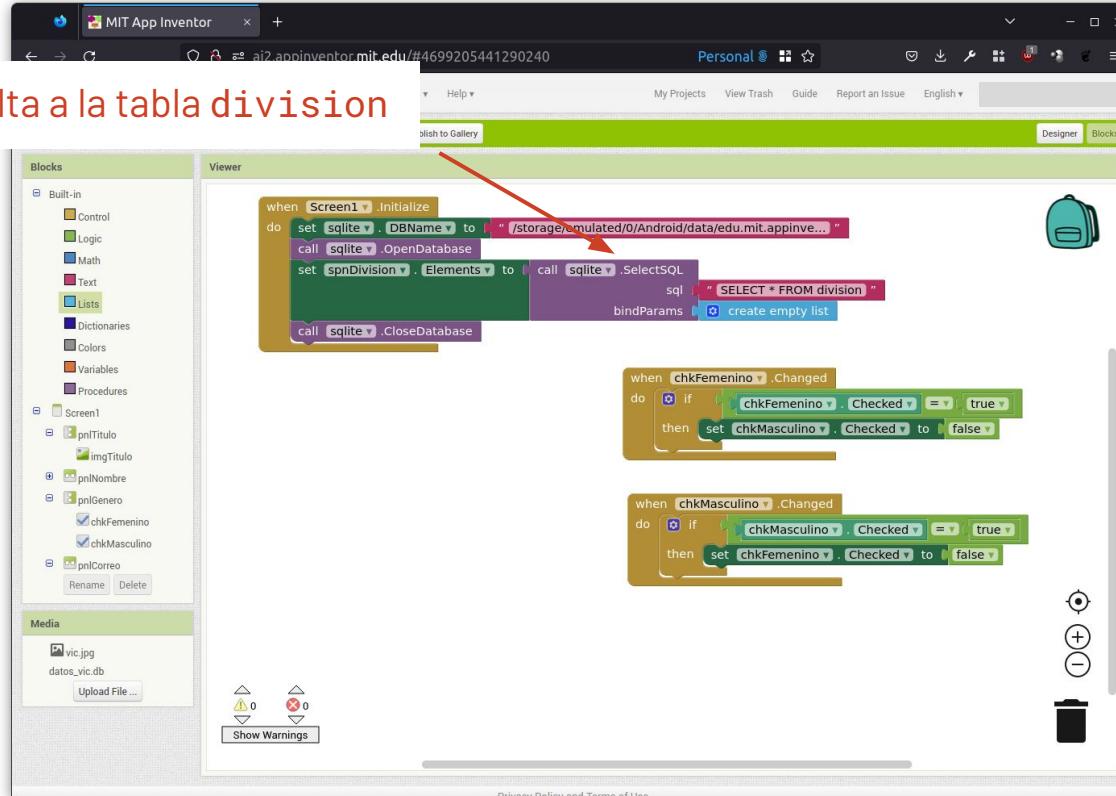
//datos_vic.db

/data/user/0/edu.mit.appinventor.aicompanion3/databases/datos_vic.db

/storage/emulated/0/AppInventor/assets/datos_vic.db//datos_vic.db

Cargar las Divisiones en el Spinner: Intento #1

Consulta a la tabla division



Cargar las Divisiones en el Spinner: Así sí

Solo requerimos el campo acronimo

The screenshot shows the MIT App Inventor Designer interface. On the left, the Blocks palette is open, showing categories like Built-in, Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures, and a specific section for Screen1 containing components pnlTitulo, imgTitulo, pnlNombre, pnlGenero, chkFemenino, chkMasculino, and pnlCorreo. In the center, the Designer view displays the following code:

```
when [Screen1.Initialize]
do
  set [sqlite v].DBName to [/storage/emulated/0/Android/data/edu.mit.appinventor.vic/databases/vic]
  call [sqlite v].OpenDatabase
  set [spnDivision v].Elements to call [sqlite v].SelectSQL
    sql is "SELECT acronimo FROM division"
    bindParams create empty list
  call [sqlite v].CloseDatabase

when [chkFemenino v].Changed
do
  if [chkFemenino v].Checked = true
  then set [chkMasculino v].Checked to false

when [chkMasculino v].Changed
do
  if [chkMasculino v].Checked = true
  then set [chkFemenino v].Checked to false
```

A red arrow points from the text "Solo requerimos el campo acronimo" to the SQL query "SELECT acronimo FROM division" in the code.

Cargar las licenciaturas al seleccionar una División: Intento #1

The screenshot shows the MIT App Inventor Designer interface with the project titled "SistemaVIC". The code blocks are organized into several sections:

- Screen1.Initialize:** A sequence of blocks to initialize the database:

```
when Screen1.Initialize
do set sqlite1 .DBName to "/storage/emulated/0/Android/data/edu.mit.appinventor.vic/datos_vic.db"
call sqlite1 .OpenDatabase
set spnDivision .Elements to call sqlite1 .SelectSQL
sql "SELECT acronimo FROM division"
bindParams create empty list
```
- spnDivision.AfterSelecting:** A sequence of blocks to load degrees for the selected division:

```
when spnDivision .AfterSelecting
selection
do call sqlite1 .OpenDatabase
set spnLicenciatura .Elements to call sqlite1 .SelectSQL
sql "SELECT nombre FROM licenciatura WHERE division = ?"
bindParams get selection
call sqlite1 .CloseDatabase
```
- chkFemenino.Changed:** A sequence of blocks to handle gender checkbox changes:

```
when chkFemenino .Changed
do if chkFemenino .Checked = true
then set chkMasculino .Checked to false
```
- chkMasculino.Changed:** A sequence of blocks to handle gender checkbox changes:

```
when chkMasculino .Changed
do if chkMasculino .Checked = true
then set chkFemenino .Checked to false
```

A red arrow points from the "get selection" block in the `spnDivision.AfterSelecting` section to the `bindParams` block in the same section.

Cargar las licenciaturas al seleccionar una División: Ahora sí

The screenshot shows the MIT App Inventor Designer interface for a project titled "SistemaVIC". The code is written in the积木 (Blocks) language.

Code Snippet 1 (Top):

```
when Screen1.Initialize
do set sqlite v. DBName to "/storage/emulated/0/Android/data/edu.mit.appinventor.vic/databases/vic"
call sqlite v. OpenDatabase
set spnDivision v. Elements to call sqlite v. SelectSQL
sql "SELECT acronimo FROM division"
bindParams create empty list
```

Code Snippet 2 (Middle):

```
when spnDivision v. AfterSelecting selection
do call sqlite v. OpenDatabase
set spnLicenciatura v. Elements to call sqlite v. SelectSQL
sql "SELECT nombre FROM licenciatura WHERE division = ?"
bindParams make a list get selection
call sqlite v. CloseDatabase
```

Code Snippet 3 (Bottom):

```
when chkFemenino v. Changed
do if chkFemenino v. Checked = true
then set chkMasculino v. Checked to false
when chkMasculino v. Changed
do if chkMasculino v. Checked = true
then set chkFemenino v. Checked to false
```

A red arrow points from the text "Lista de parámetros en bindParams" to the "make a list" block in the middle code snippet.

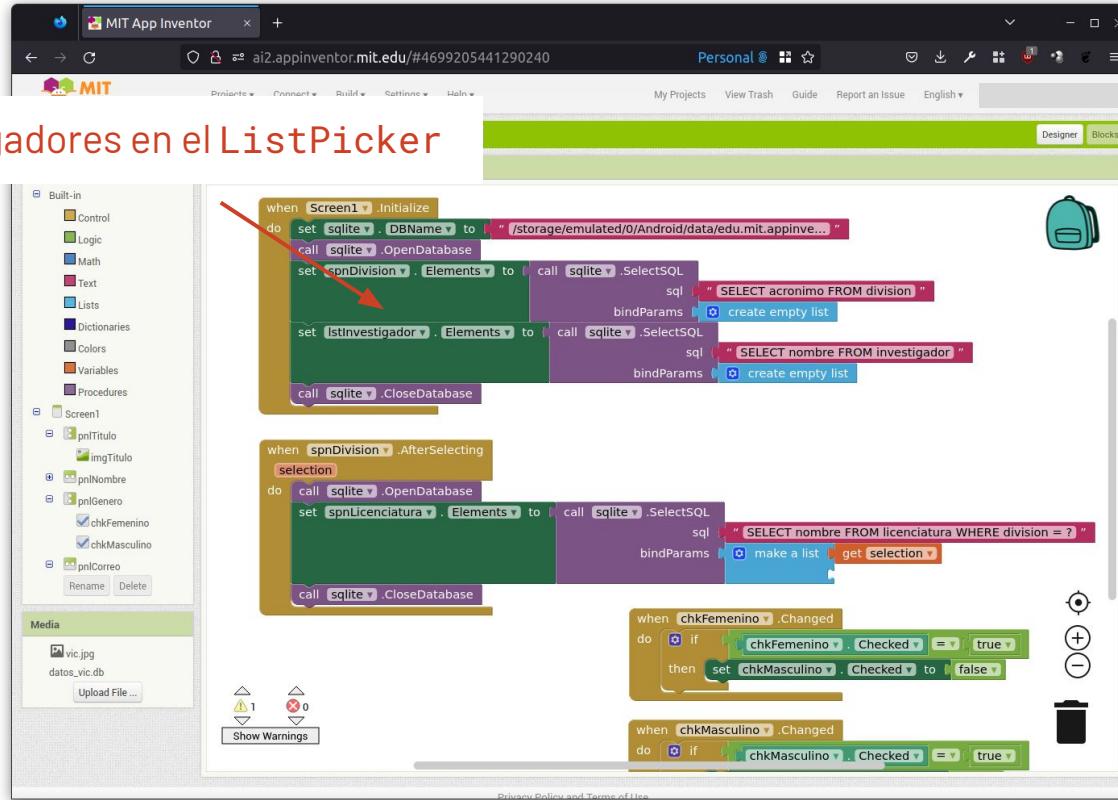
Project Structure:

- Blocks: Built-in, Control, Logic, Math
- Viewer: Screen1, Add Screen..., Remove Screen, Publish to Gallery
- Designer: Personal, Projects, Connect, Build, Settings, Help, My Projects, View Trash, Guide, Report an Issue, English
- Media: vic.jpg, datos_vic.db
- Upload File...
- Show Warnings

Lista de parámetros en bindParams

Mostrar lista de investigadores

Cargar investigadores en el ListPicker



Mejorando la interfaz

The screenshot shows the MIT App Inventor Designer interface. On the left, the 'Blocks' panel lists components: pnIPromedio, tblPromedio, lblPromedio, lblAvance, txtPromedio, txtAvance; pnDivision, tblDivision, lblDivision, lblLicenciatura, spnDivision, spnLicenciatura; pnInvestigador, lstInvestigador; pnRegistro, btnRegistro; and sqlite. The 'Viewer' panel displays three script blocks:

- A main block under 'when Screen1.Initialize':

```
when Screen1.Initialize
do set sqlite . DBName to "/storage/emulated/0/Android/data/edu.mit.appinve...
call sqlite .OpenDatabase
set spnDivision . Elements to call sqlite .SelectSQL
sql "SELECT acronimo FROM division "
bindParams create empty list
```
- A block under 'when spnDivision.AfterSelecting':

```
when spnDivision.AfterSelecting
selection
do call sqlite .OpenDatabase
set spnLicenciatura . Elements to call sqlite
```
- A final block under 'when lstInvestigador.AfterPicking':

```
when lstInvestigador.AfterPicking
do set lstInvestigador . Text to lstInvestigador . Selection
```

A red arrow points from the text 'Propiedad title del ListPicker' at the bottom left to the 'Text' property of the final block's 'Set Text' action.

Propiedad **title** del ListPicker

Cambiar el texto que muestra el ListPicker

iListo!

Contenido



1. Diseñando la app
2. Live testing
3. Escribiendo código
4. SQLite
- 5. Base de datos en la nube**
6. Ejecución del proyecto

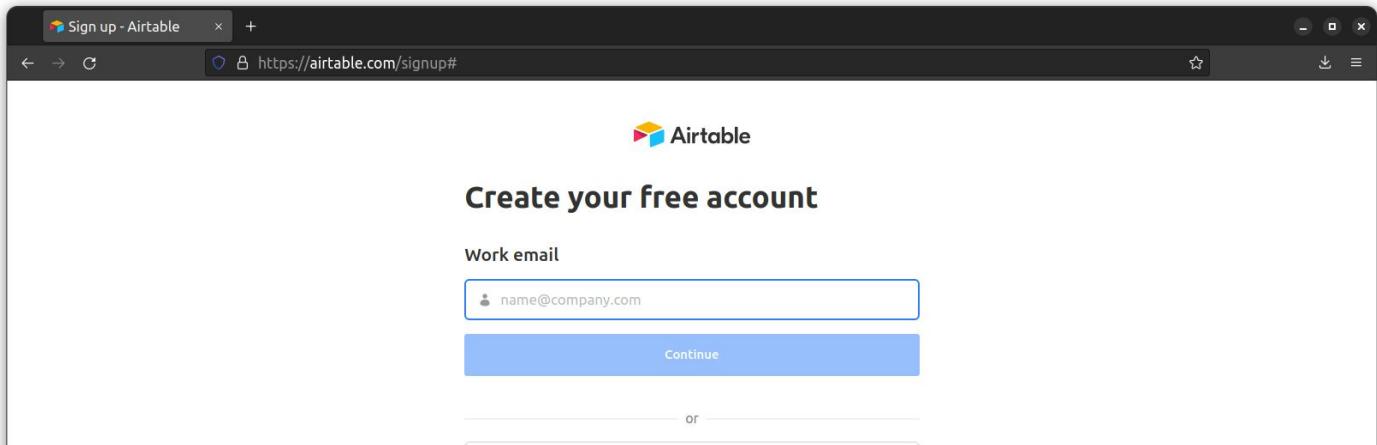


Base de datos en la nube

The screenshot shows the Airtable website (<https://www.airtable.com>) with a focus on its cloud integration features. The main interface displays a "Product Management" base titled "Customer Feedback". The table has columns for "Title", "Priority", "Source", and "Actions". Several rows of feedback are listed, each with a "Create Jira issue" checkbox and a "Open in Jira" button. To the left of the table, there are icons for Microsoft Office 365, Zendesk, Google Drive, and a dotted line indicating integration. To the right, there are icons for GitHub, a blue diamond shape, Salesforce, and Google Sheets, also connected by dotted lines. The top navigation bar includes links for Product, Solutions, Pricing, Enterprise, Resources, Contact sales, Sign up for free, and Sign in.

<https://www.airtable.com>

Registrarse



https://airtable.com/invite/I?inviteId=invMe61hjW6rhQ8Zb&inviteToken=d1d01cc e13d8837716fc6cd35b69e8610e691a205ce605cba792a85713d4c60c&utm_medium=email&utm_source=product_team&utm_content=transactional-alerts

Crear la base de datos

The screenshot shows the Airtable homepage with a trial upgrade notice and workspace management.

Upgrade to Pro before your trial expires

Keep the power you need to manage complex workflows, run reports, and more with tools like:

- ✓ Multiple extensions per base
- ✓ More sync integrations to quickly bring in outside data
- ✓ More record limits and attachment space
- ✓ Team permissions

[Upgrade now](#)

Please verify your email address by clicking the link sent to [redacted]
[Resend verification email](#)

Bases de datos Pro trial 11 days left

[SHARE](#)

Workspaces listed: **datos_vic**, **cicoti**, Add a base

Search bar: Find a base or interface

Navigation menu:

- Interfaces
- WORKSPACES
 - Bases de datos Trial
 - Add a workspace
 - Trash
- LEARNING AND RESOURCES
 - Guide to Airtable
 - Contact sales
 - Webinars
 - Help center

Start from scratch
Create a new blank base with custom tables, fields, and views.

Quickly upload
Easily migrate your existing projects in just a few minutes.

Start with templates
Select a template to get started and customize as you go.

Tabla estudiante

The screenshot shows the Airtable interface for the 'estudiante' table. The top navigation bar includes tabs for 'Data', 'Automations', and 'Interfaces'. The main header shows the table name 'estudiante' and a '+' button. The URL in the address bar is <https://airtable.com/appC6zu8aX66lxcID/tblTpKBBe12x8BpHH/viwwaMIFF8qYAJyNh?blocks=hide>. The top right corner displays a '14 days left on Pro' badge, notifications, and user account options.

The table structure is visible below the header:

A matrícula	A nombre	genero	A correo	## promedio	## avance	A división	A licenciatura	A investigador
-------------	----------	--------	----------	-------------	-----------	------------	----------------	----------------

A modal window is open for the 'matrícula' field, which is defined as a 'Single line text' type. The modal contains the following text:

This is the table's primary field. It cannot be hidden or moved. The **name** is meant to be a short, unique representation of each record in your table.

A single line of text.

Buttons at the bottom of the modal include '+ Add description', 'Cancel', and a highlighted 'Save' button.

At the bottom of the table view, there are buttons for '+', 'Add...', '0 records', 'Sum 0.00', 'Sum 0.00', and a 'Getting started' link.

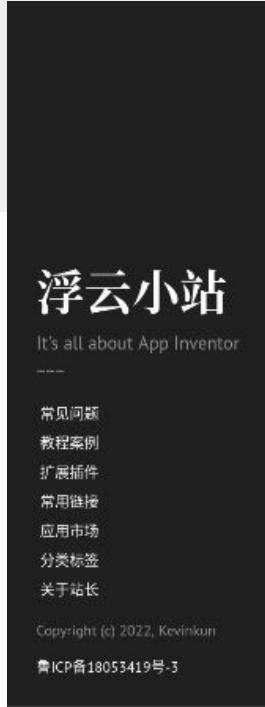
Tabla estudiante

The screenshot shows a browser window for the Airtable 'estudiante' database. The URL is <https://airtable.com/appC6zu8aX66lxcID/tblTpKBBe12x8BpHH/viwwaMIFF8qYAJyNh?blocks=hide>. The interface includes a top navigation bar with 'Data', 'Automations', 'Interfaces', and a 'Saving...' status. A modal dialog is open over the grid view, centered on the 'genero' field. The dialog title is 'genero' and the type is 'Single select'. It contains two options: 'Femenino' and 'Masculino', both with small dropdown arrows. Below the options are buttons for '+ Add an option', '+ Add description', 'Cancel', and a prominent blue 'Save' button. At the bottom of the dialog is a link '+ Add a Kanban view organized by this field'. The main grid view shows columns for 'matricula', 'nombre', 'genero', 'correo', 'promedio', 'avance', 'division', 'licenciatura', and 'investigador'. The 'genero' column header is currently selected. The bottom of the screen shows a footer with '0 records', 'Sum 0.00', 'Getting started', and a help icon.

Tabla estudiante

The screenshot shows a screenshot of an Airtable interface for a table named "estudiante". The table has columns for matrícula, nombre, genero, correo, promedio, avance, division, licenciatura, and investigador. A modal dialog is open over the table, focusing on the "promedio" field. The dialog is titled "promedio" and is set to "Number" format. Under "Format", "Decimal (1.0)" is selected, and "Precision" is set to "1.00". There is an unchecked checkbox for "Allow negative numbers". At the bottom of the dialog are "Add description", "Cancel", and a blue "Save" button.

Extensión App Inventor



<https://kevinkun.cn/airtable/>

Airtable extension

2022-09-16 15:21 | 阅读量: 47 | 标签: extension, database, airtable

Why another extension about Airtable? Because this one is different. This one can select/ insert/ update/ delete according to your filter/condition.

V2 Update history 2020/12/24

- fix bug of 'can not update value with space'; now in the formula, use " and " to quote the value with space.
- add property of FetchAndTime. if set to true, in the SelectFinished event you will get 'id' and 'createdTime' for each record.

set `KevinkunAirtable1` .FetchAndTime to `(true)`

- add method 'UpdateById' and 'DeleteById' for update or delete one record, which is faster than by filter.

call `KevinkunAirtable1` .UpdateById
 `id` `data`

call `KevinkunAirtable1` .DeleteById
 `id`

V1:

Add a picture of all the blocks

Initialize the extension

when `Screen1` .Initialize
do
 set `KevinkunAirtable1` . API Key to `"Your API Key"`
 set `KevinkunAirtable1` . BaseID to `"Your Base ID"`
 set `KevinkunAirtable1` . TableName to `"Your Table name"`

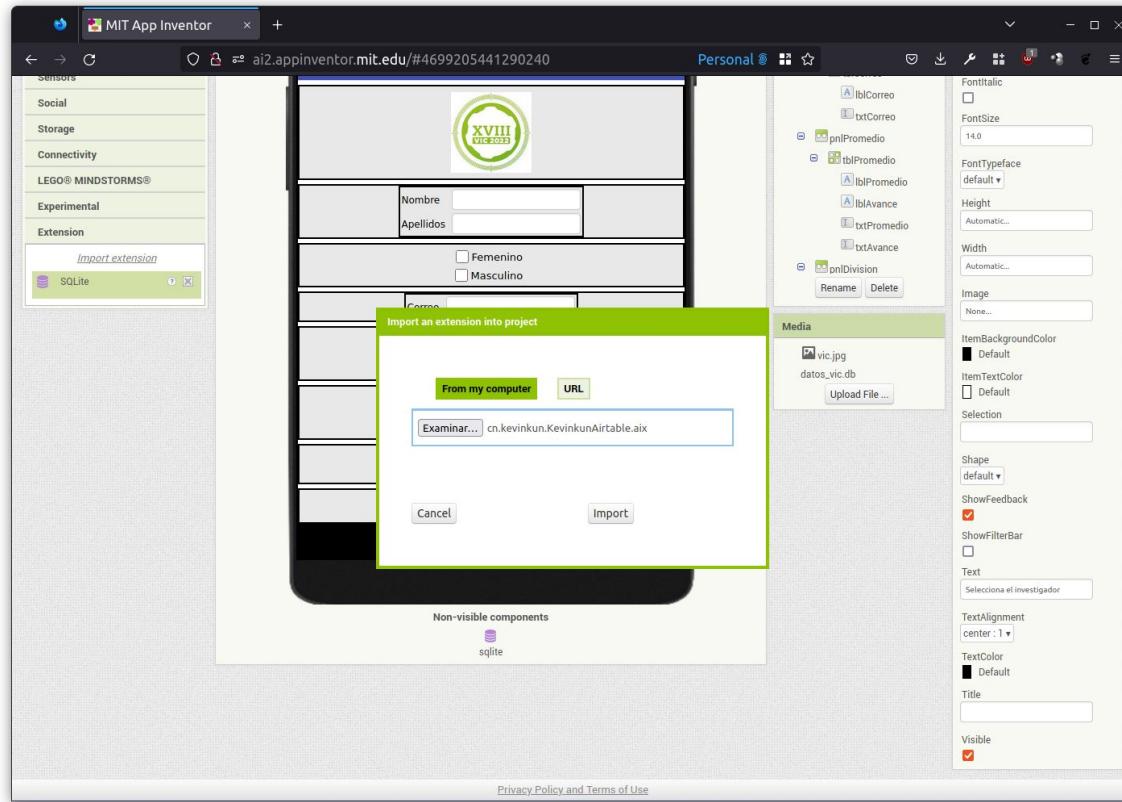
Same like other Airtable component, get your API Key, BaseID, TableName from the airtable website.

Select records

call `KevinkunAirtable1` .Select
 `fields`
 `filter`
 `pageSize`
 `offset`
 `sort`

when `KevinkunAirtable1` .SelectFinished

Instalando la extensión



Configurando el componente

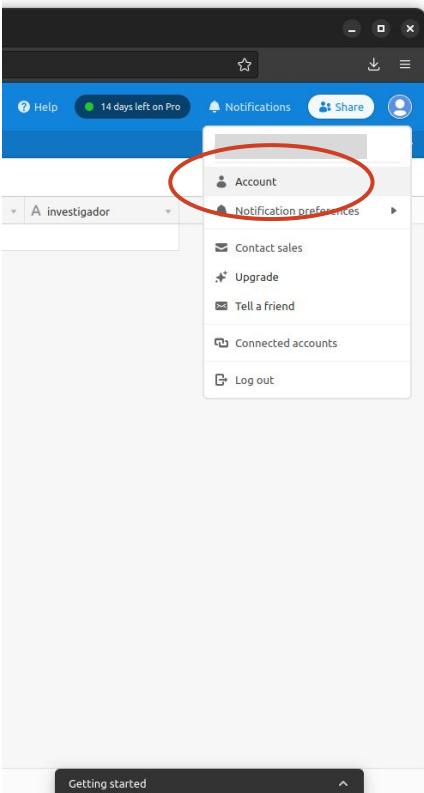
The screenshot shows the MIT App Inventor web-based development environment. On the left, the 'Palette' sidebar lists various component categories like User Interface, Layout, Media, etc. In the center, the 'Viewer' pane displays a smartphone screen with a form titled 'Registro al VIC 2022'. The form includes fields for Nombre, Apellidos, gender selection (Femenino/Masculino), Correo, Promedio, Avance curricular, División, and Licenciatura. At the bottom, it says 'Selecciona el investigador'. On the right, the 'Components' and 'Properties' panes are visible. A red box highlights the 'Properties' pane for the 'tblPromedio' component, which contains settings for 'airtable', 'ApiKey', 'BaseID', 'FetchIdAndTime', and 'TableName'. Below the components, there's a 'Media' section with an image file named 'vic.jpg'. A large text box at the bottom right asks: '¿De donde tomamos estos valores?' (Where do we get these values?).

Properties pane details:

- airtable
- ApiKey
- BaseID: appC6cu8aXG6ixcD
- FetchIdAndTime
- TableName: estudiante

Text at the bottom right: ¿De donde tomamos estos valores?

Configurando la API de Airtable



A screenshot of the Airtable account overview page. The URL in the browser is https://airtable.com/account. The page has a dark header with the Airtable logo and navigation links for HELP, LOG OUT, and other account settings. The main content area is titled 'Account overview' and features a profile picture of Oscar Chávez-Bosquez with his email address chavez.bosquez@gmail.com. It includes sections for 'CREDITS' (showing \$0) and 'API'. A note states: 'There is no personal API key associated with your account. It's required in order to use the Airtable API.' A prominent red circle highlights the 'Generate API key' button. Other sections include 'WORKSPACES' (Bases de datos) and 'GOOGLE CONTACTS INTEGRATION' (Allow Airtable to import your Google Contacts). The bottom section is 'GOOGLE DRIVE INTEGRATION' (Link your Airtable account to Google Drive).

Copiar la API key

The screenshot shows the 'Account overview' page in Airtable. On the left, there's a sidebar with 'ACCOUNT' and 'WORKSPACES' sections. Under 'ACCOUNT', there are links for 'Overview', 'Referrals and credits', and 'Recent activity'. Under 'WORKSPACES', there's a link for 'Bases de datos' which is currently selected, indicated by a yellow PRO PLAN badge.

The main content area is titled 'Account overview'. It displays the user's profile picture, name (Oscar Chávez-Bosquez), email (chavez.bosquez@gmail.com), and options to 'Edit name', 'Edit email', 'Update password', and 'Set up two-factor authentication'.

Below the profile, there's a section for 'CREDITS' with a note that the user has '\$0 in credit'. It offers incentives for inviting people and installing the mobile app.

The 'API' section is highlighted with a red oval. It explains that this is the personal API key required for using the Airtable API. It states that the key has access to all data in the user's bases and should only be shared with trusted third-party services. It also provides instructions for creating a read-only API key.

At the bottom of the API section, there's a button with a redacted API key value followed by 'Regenerate API key' and 'Delete key' buttons.

At the very bottom, there's a 'GOOGLE CONTACTS INTEGRATION' section with a checkbox for 'Allow Airtable to import your Google Contacts.' and a note about making it easy to invite contacts.

A large red text overlay 'Copiar la llave' is positioned on the left side of the API key area.

Id de la base de datos

The screenshot shows a web browser window with the URL <https://airtable.com/api>. The page title is "REST API - Airtable". A banner at the top says "Just released: new ways to connect your apps with shared data [Read blog post →](#)". The main heading is "REST API". Below it, a text block explains: "After you've created and configured the schema of an Airtable base from the graphical interface, your Airtable base will provide its own API to create, read, update, and destroy records." A link "Click here for Airtable's metadata API documentation." is provided. A section titled "Select a base to view API documentation" lists two bases: "Bases de datos" (with icons for "datos_vic" and "cicoti") and "Bases de datos" (with icons for "cicoti"). At the bottom, there is a navigation bar with links to "Product", "Solutions", "Pricing", "Enterprise", "Contact Sales", and social media icons for Facebook, LinkedIn, Twitter, and YouTube. There is also a footer with links to "Popular Templates", "Popular Extensions", "Popular Solutions", "Learn More", "Company", "Content Calendar", "Miro", "Marketing", "Community", and "About".

<https://airtable.com/api>

Id de la base de datos

Copiar el id

The screenshot shows a web browser window with two tabs: 'datos_vic: estudiante' and 'Airtable API - datos_vic'. The main content is the 'Airtable API for "datos_vic"' documentation. On the left, there's a sidebar with navigation links like INTRODUCTION, METADATA, RATE LIMITS, AUTHENTICATION, ESTUDIANTE TABLE, and ERRORS. The 'INTRODUCTION' section contains text about integrating the 'datos_vic' data into Airtable. A green box highlights the line 'The ID of this base is `appC6zu8aX66Ixc1D`'. This line is circled in red. Below it, there's a note about changing field names or types. The right side of the screen has code examples for curl, JavaScript, and a 'show API key' button.

The ID of this base is `appC6zu8aX66Ixc1D`.

Please note: if you make changes to a field (column) name or type, the API interface for those fields will change correspondingly. Therefore, please make sure to update your API implementation accordingly whenever you make changes to your Airtable schema from the graphical interface.

Official API client:

- JavaScript: [airtable.js](#) (Node.js + browser)

Community-built API clients:

- Ruby: [airrecord](#)
- .NET: [airtable.net](#)
- Python 3: [pyairtable](#)
- Python 2/3: [airtable.py](#)

METADATA

This API gives you the ability to list all of your bases, tables, fields, and views. You can [register](#) for access

¿Cómo se guardan los datos en Airtable?

```
Getting tag values indicated by [red box]
the "list" tag denotes an array that has 40 elements

{
  "cod": "200",
  "message": 0,
  "cnt": 40,
  "list": [
    {
      "dt": 1575547200,
      "main": {
        "temp": -5.88,
        "temp_min": -5.88,
        "temp_max": -4.6,
        "pressure": 1020,
        "sea_level": 1020,
        "grnd_level": 1011,
        "humidity": 68,
        "temp_kf": -1.28
      },
      "weather": [
        {
          "id": 600,
          "main": "Snow",
          "description": "light snow",
          "icon": "13n"
        }
      ],
      "clouds": {},
      "wind": {},
      "snow": {},
      "sys": {},
      "dt_txt": "2019-12-05 12:00:00"
    },
    ...
    // 39 more items
    {
      "dt": 1575968400,
      "main": {},
      "weather": [],
      "clouds": {},
      "wind": {},
      "sys": {},
      "dt_txt": "2019-12-10 09:00:00"
    }
  ]
}
```

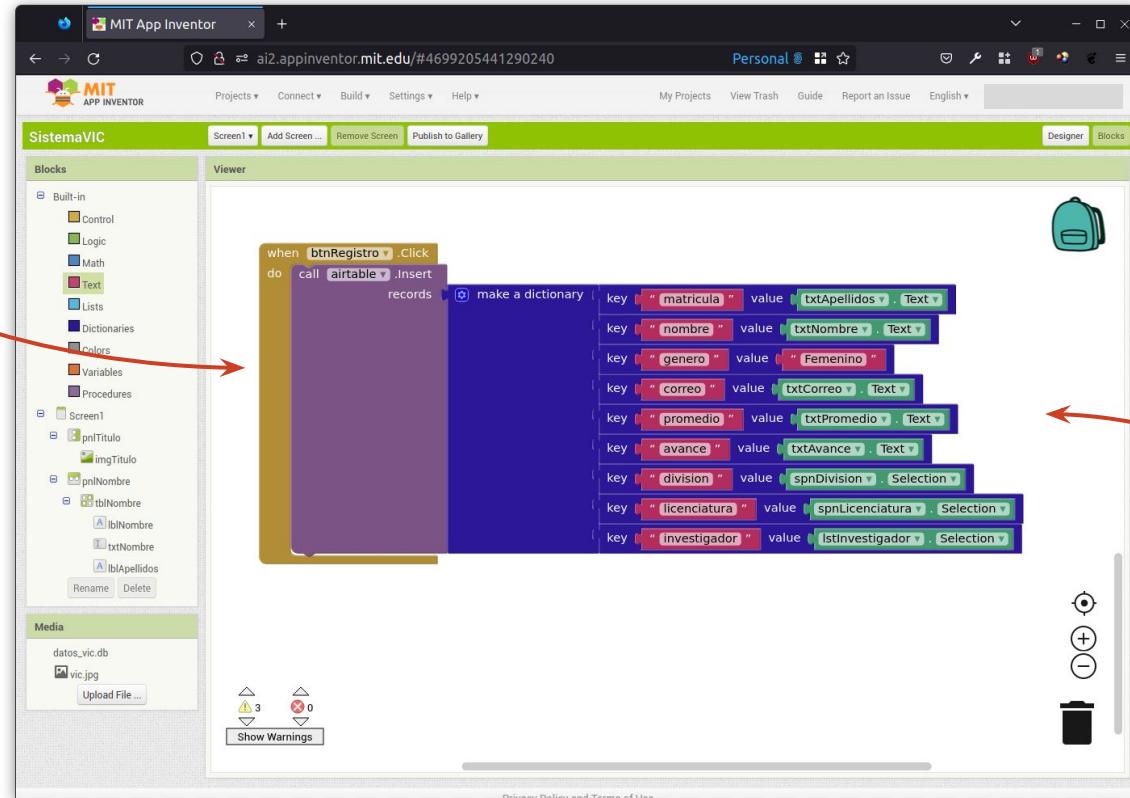
Fig.1 Weather forecast JSON file from the OpenWeather

Formato JSON == Diccionario:

[http://ai2.appinventor.mit.edu/reference\(blocks/dictionaries.html](http://ai2.appinventor.mit.edu/reference(blocks/dictionaries.html)

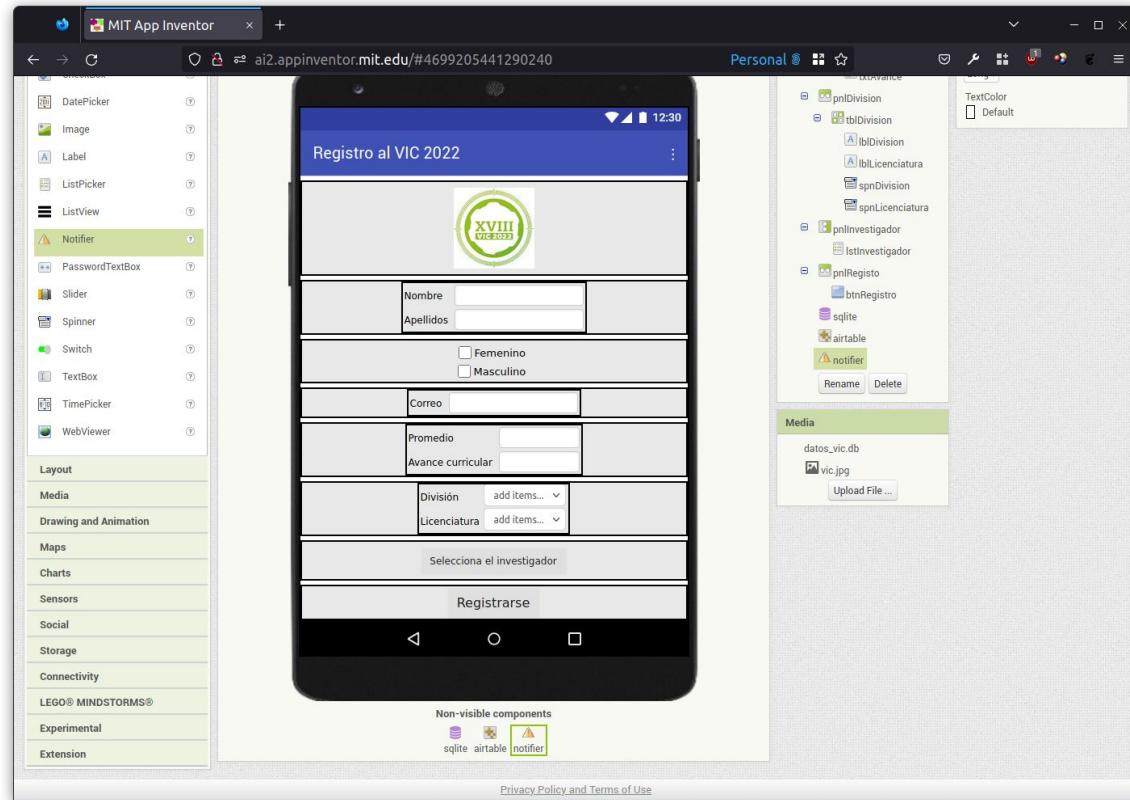
Guardar los datos del veraneante

Componente
airtable



Componente
diccionario

Notificación del resultado



Guardar el registro completo

Variable genero



initialize global [genero] to "Femenino"

Actualizando la variable



when chkFemenino .Changed
do if chkFemenino .Checked = true
then set chkMasculino .Checked to false
set global genero to "Femenino"

when chkMasculino .Changed
do if chkMasculino .Checked = true
then set chkFemenino .Checked to false
set global genero to "Masculino"



Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
pnITitulo
imgTitulo
pnINombre
tblNombre
lblNombre
txtNombre
lblApellidos
txtApellidos
Rename / Delete

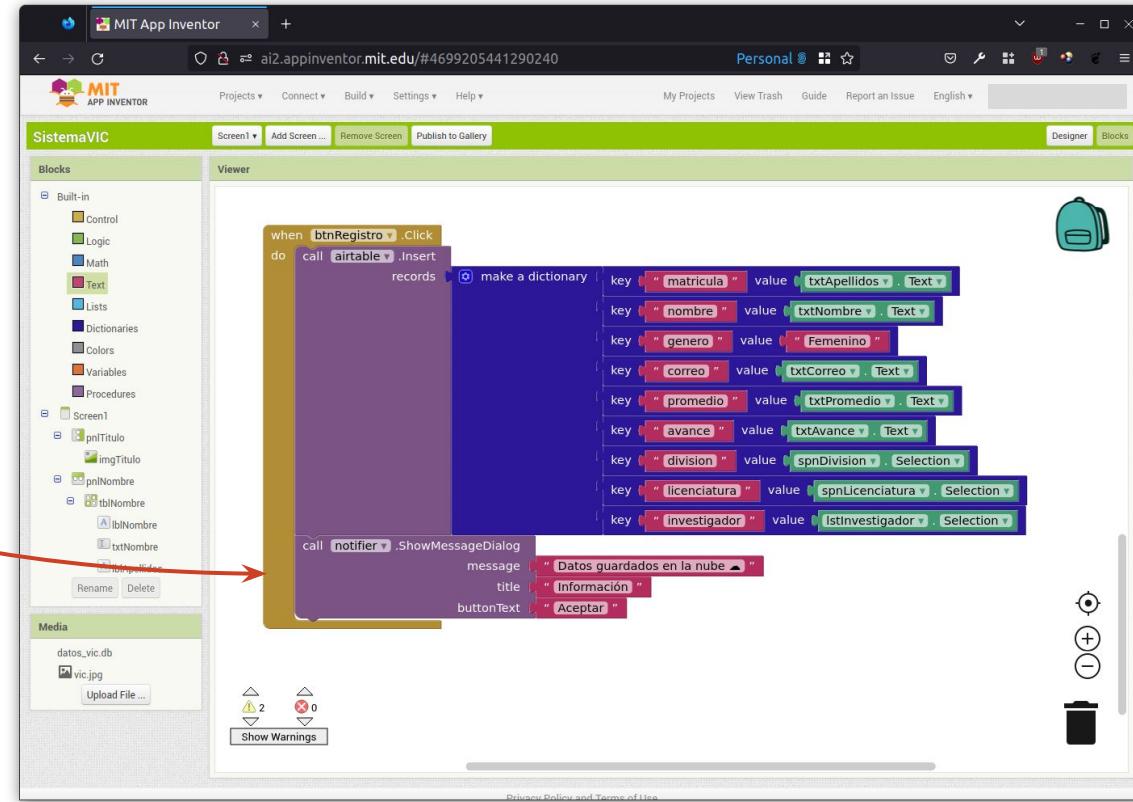
Media
datos_vic.db
vic.jpg
Upload File ...

when btnRegistro .Click
do call airtable .Insert
records make a dictionary
key "matricula" value txtApellidos .Text
key "nombre" value txtNombre .Text
key "genero" value get global genero
key "correo" value txtCorreo .Text
key "promedio" value txtPromedio .Text
key "avance" value txtAvance .Text
key "division" value spnDivision .Selection
key "licenciatura" value spnLicenciatura .Selection
key "investigador" value lstdInvestigador .Selection
call notifier .ShowMessageDialog
message " Datos guardados en la nube "
title " Información "
buttonText " Aceptar "



Guardando la variable

Notificación del resultado



Método
showMessageDialog

Contenido



1. Diseñando la app
2. Live testing
3. Escribiendo código
4. SQLite
5. Base de datos en la nube
- 6. Ejecución del proyecto**



¿No funciona?

The screenshot shows the Airtable interface for a 'estudiante' database. At the top, there are tabs for 'Data', 'Automations', and 'Interfaces'. Below the header, there are buttons for 'Views', 'Grid view', 'Hide fields', 'Filter', 'Group', 'Sort', 'Color', 'Share view', 'Extensions', and 'Tools'. The main area displays a grid of student records. Two modal dialogs are overlaid on the grid:

- Field Type: promedio**

A single line text. You can optionally prefill each new cell with a default value:
Default text
Enter default text

+ Add description Cancel Save
- Field Type: avance**

A single line text. You can optionally prefill each new cell with a default value:
Default text
Enter default text

+ Add description Cancel Save

Cambiar el tipo de dato de los campos **promedio** y **avance**

The screenshot shows the 'estudiante' view in Airtable with a new record added. The bottom left corner of the grid shows the text '0 records'. A new row is visible at the bottom of the grid, indicated by a plus sign icon and an 'Add...' button.

Registro al VIC 2022



Nombre Oscar

Apellidos Chávez

 Femenino Masculino

Correo scar.chavez@ujat.mx

Promedio 8.9

Avance curricular 72

División DACB

Licenciatura Licenciatura en Actuaría

ADDY MARGARITA BOLIVAR CIME

Registrarse

Información

Datos guardados en la nube

Aceptar

Ejecución del proyecto

datos_vic: estudiante

https://airtable.com/appC6zu8aX66IxcID/tbTpKBBe12x8BpHH/viwwaMIFF8qYAJyNh?blocks=hide

120% Notifications Share

estudiante Data Automations Interfaces

Add or import

Views Grid view Hide fields Filter Group Sort Color Share view

	A matricula	A nombre	genero	A correo	A promedio	A avance	A division	A licenciatura	A investigador
1	Chávez	Oscar	Masculino	oscar.chavez@uja...	8.9	72	DACB	Licenciatura en Actuaría	ADDY MARGARITA BOLIV...

+ Add... 1 record

¡Listo!

1. Desarrollamos la app en App Inventor
2. Importamos datos de SQLite
3. Base de datos en Airtable
4. Sistema VIC en el móvil 