

*Creació avançada d'aplicacions mòbils*  
**Materials de consulta**

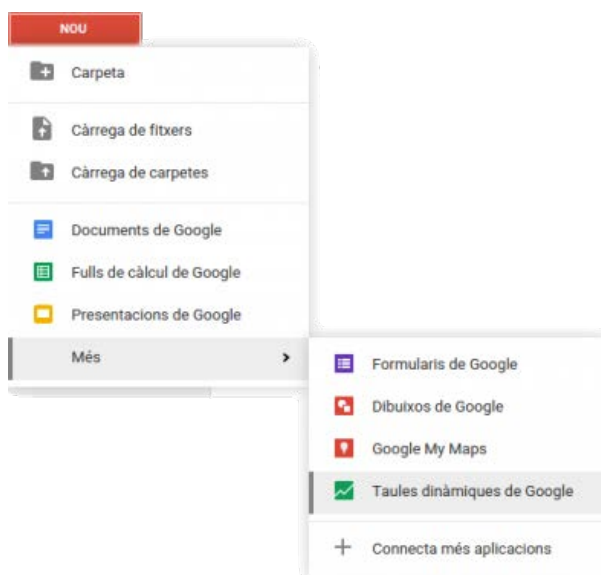
## **Tema 4. Utilitats avançades d'App Inventor**

### **1. Les Fusion Tables com a base de dades**

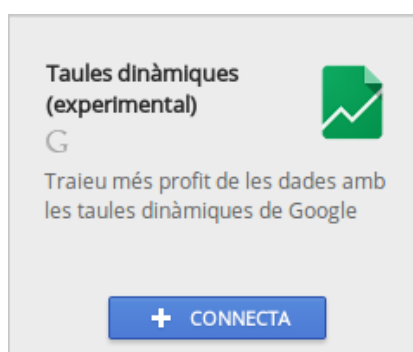
#### **1.1. Crear una Fusion Table**

D'entre les diferents opcions d'accés a dades que ens ofereix l'App Inventor, les Fusion Tables són les més versàtils i fàcils d'utilitzar. Un dels motius principals és que hi podem accedir també des del Drive de Google.

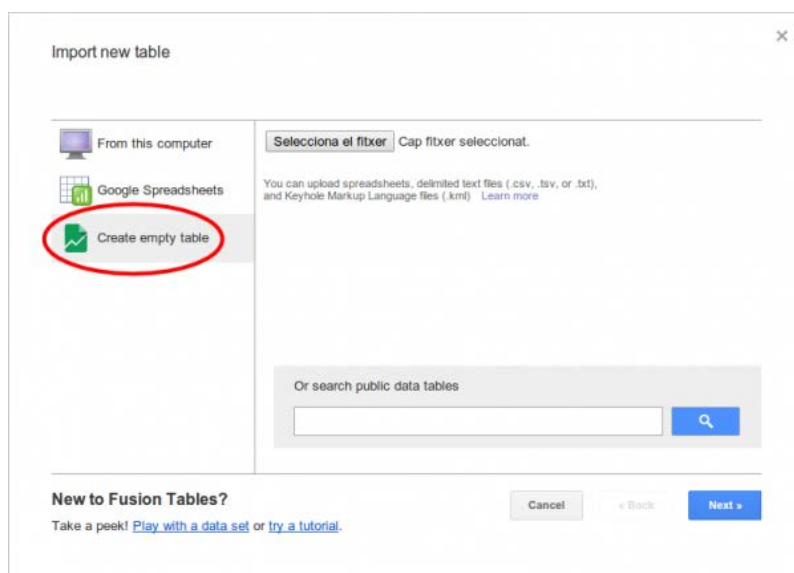
Per crear una Fusion Table només ens cal anar al Drive i crear-la com qualsevol altre tipus de document més:



En cas que no ens aparegui aquesta possibilitat el que hem de fer és anar a l'opció Connecta més aplicacions i triar-la de la llista d'opcions:



Per crear la nostra primera Fusion Table li hem de dir que volem crear un taula buida (Create empty table):



Un cop fet això ens apareixerà una proposta de taula amb quatre camps. Per configurar la taula amb els camps que nosaltres necessitem anem a Edit → Change columns i trobarem que els quatre camps que hi ha corresponen als quatre tipus de dades que les Fusion Tables poden emmagatzemar:






- Text. Li podem modificar el format per tal que hi pugui haver un enllaç a una pàgina web, un enllaç a una imatge o un enllaç a altres tipus de continguts
- Number. Admet diferents formats de nombres tal com podríem fer amb un full de càlcul
- Location. Podem posar-hi una adreça o el podem configurar per tal que faci referència a dos camps de la nostra taula que continguin la Latitud i la Longitud
- Date/time. Podem triar el format en què volem aquest camp






A més a més en tots ells, excepte en els camps de tipus Location, podem incloure-hi una validació de les dades de manera que es pugui triar des d'una llista desplegable.

A continuació hi ha un exemple d'una taula amb cinc camps:

- Nom. Un camp de tipus Text
- DataNaixement. Un camp de tipus Date/time
- Adreça. Un camp de tipus Location
- Nivell. Un camp de tipus Text amb validació de dades on només es poden introduir els valors corresponents a qualsevol nivell de primària: Primer, Segon,...Sisè. Fem que tan sols es puguin introduir aquests valors entrant-los en el quadre List of drop-down items.
- Foto. Un camp de tipus Text amb format Four line image on hi posem la URL d'una imatge i ens la mostra amb una mida que ocupa l'equivalent a quatre línies de text

En aquesta taula hi hem introduït quatre registres, que podem veure en format llista, en format fitxes o, com que hi ha localitzacions, en format mapa:

Nom	DataNaixement	Adreca	Nivell	Foto
Robert	2006-12-06	Plaça dels Corral, 3, Mataró	Sisè	
Joana	2017-09-07	Plaça Catalunya, 12, Barcelona	Tercer	
Pere Ribot Vast	2010-06-01	Sant Salvador 30, Barcelona	Segon	
Violeta	2011-11-11	Plaça dels Tilers, 12, Barcelona	Segon	
Vinyet Valls	2006-08-28	Província de Barcelona	Cinquè	

<p><b>Nom:</b> Robert <b>DataNaixement:</b> 2006-12-06 <b>Adreca:</b> Plaça dels Corral, 3, Mataró <b>Nivell:</b> Sisè <b>Foto:</b></p> 	<p><b>Nom:</b> Joana <b>DataNaixement:</b> 2017-09-07 <b>Adreca:</b> Plaça Catalunya, 12, Barcelona <b>Nivell:</b> Tercer <b>Foto:</b></p> 
<p><b>Nom:</b> Pere Ribot Vast <b>DataNaixement:</b> 2010-06-01 <b>Adreca:</b> Sant Salvador 30, Barcelona <b>Nivell:</b> Segon <b>Foto:</b></p> 	<p><b>Nom:</b> Violeta <b>DataNaixement:</b> 2011-11-11 <b>Adreca:</b> Plaça dels Tilers, 12, Barcelona <b>Nivell:</b> Segon <b>Foto:</b></p> 
<p><b>Nom:</b> Vinyet Valls <b>DataNaixement:</b> 2006-08-28 <b>Adreca:</b> Província de Barcelona <b>Nivell:</b> Cinquè <b>Foto:</b></p> 	

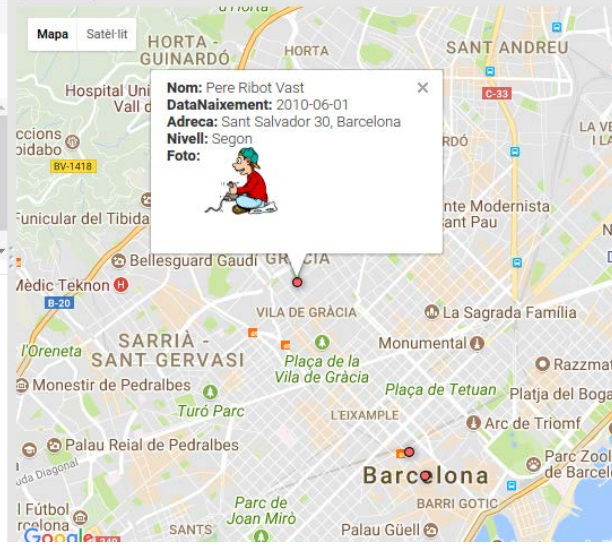
Map of Location

Filter: Adreca IN ('Plaça Catalunya, 12, Barcelona', 'Plaça dels Corral, 3, Mataró', 'Província de Barcelona', 'Sant Salvador 30, Barcelona')


Adreca

5 distinct values

- Plaça Catalunya, 12, Barcelona 1
- Plaça dels Corral, 3, Mataró 1
- Plaça dels Tilers, 12, Barcelona 1
- Província de Barcelona 1
- Sant Salvador 30, Barcelona 1



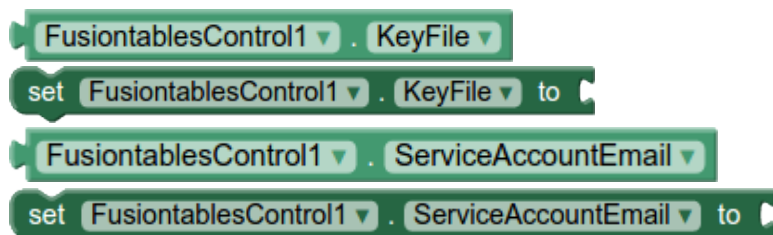
**Nom:** Pere Ribot Vast  
**DataNaixement:** 2010-06-01  
**Adreca:** Sant Salvador 30, Barcelona  
**Nivell:** Segon  
**Foto:**



Per poder utilitzar una Fusion Table és imprescindible fer-la pública per tothom que tingui l'enllaç. Altrament l'App Inventor ens donarà error.

## 1.2. Configurar l'accés

Des d'App Inventor podem utilitzar el component FusionTablesControl per tal d'accedir a una Fusion Table i utilitzar-la com a base de dades. Per tal d'accedir-hi necessitarem els següents components:



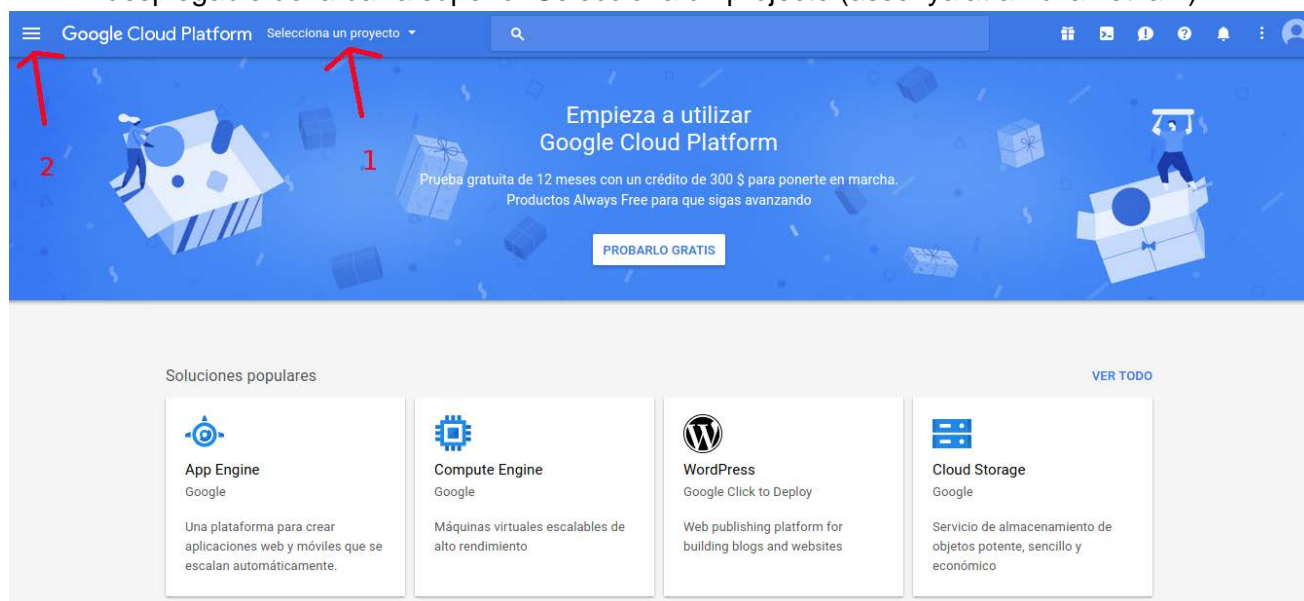
El procediment que mostrem tot seguit ens permetrà obtenir la informació necessària que permetrà que la nostra app accedeixi a les fusion tables. En concret obtindrem:

1- KeyFile: Es tracta d'un arxiu que identificarà la nostra aplicació i que li permetrà accedir a les taules que decidim.

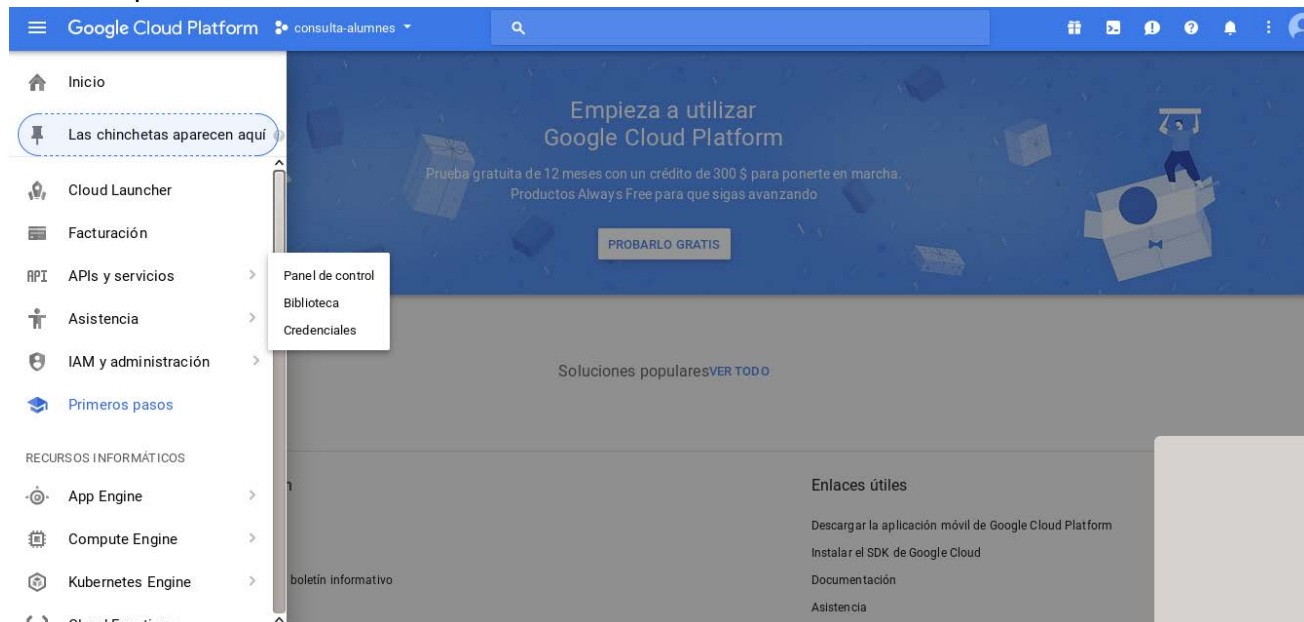
2- ServiceAccountEmail: Generarem un nou usuari de gmail associat a l'API que usarem per accedir a les fusion table compartides.

Per obtenir aquesta informació hem de seguir aquestes instruccions:

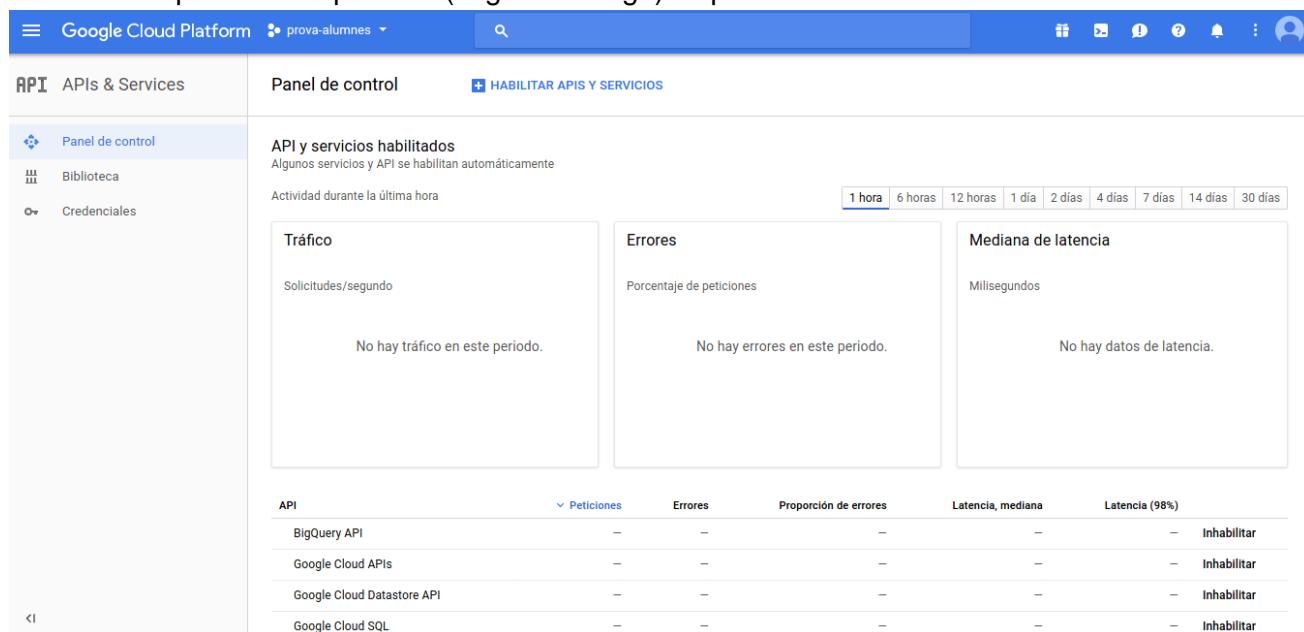
- Hem d'anar a la Consola d'APIs de Google. Cal que estiguem validats amb un usuari gmail. Si és el primer cop que accedim a la consola de desenvolupadors de Google o no hem creat mai cap projecte, ens apareixerà aquesta pantalla on n'hem de crear un mitjançant el desplegable de la barra superior Selecciona un projecte (assenyalat amb la fletxa 1).



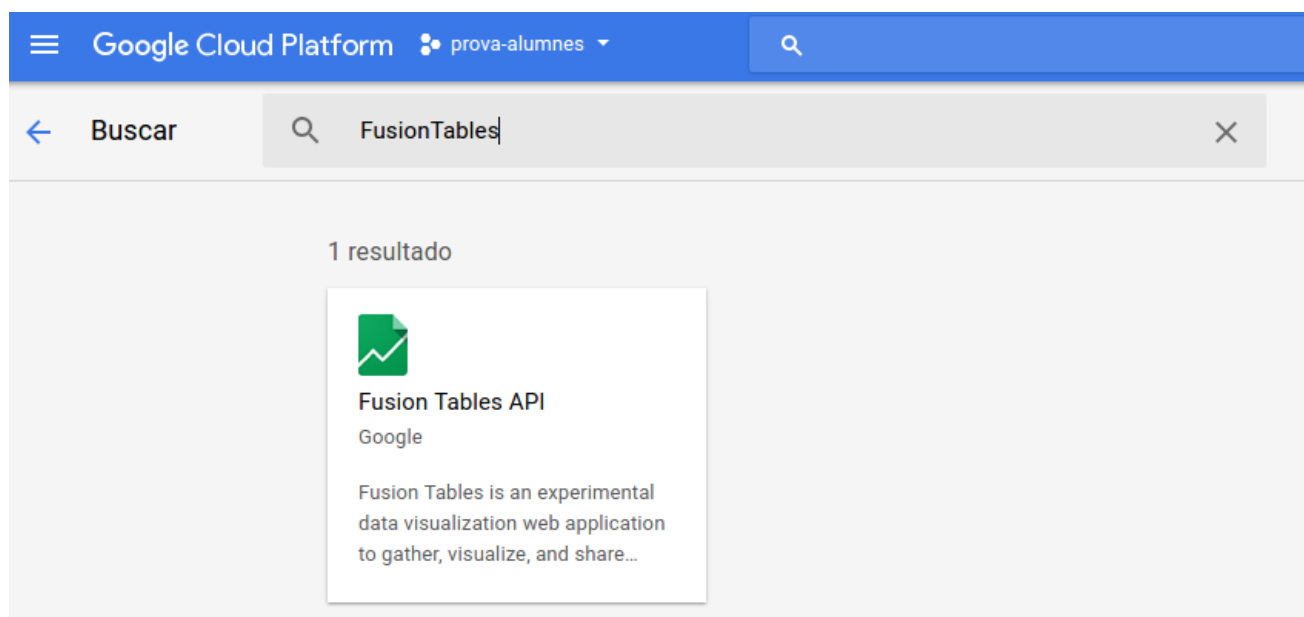
- Un cop creat tornem la mateixa pantalla anterior i si tornem a entrar a l'opció Selecciona un projecte ja podrem escollir el projecte creat. Seguint a la mateixa pantalla cliquem el botó Productes i serveis (assenyalat amb la fletxa 2) i es desplega un menú lateral d'on escollim l'opció API i Serveis → Panell de control.



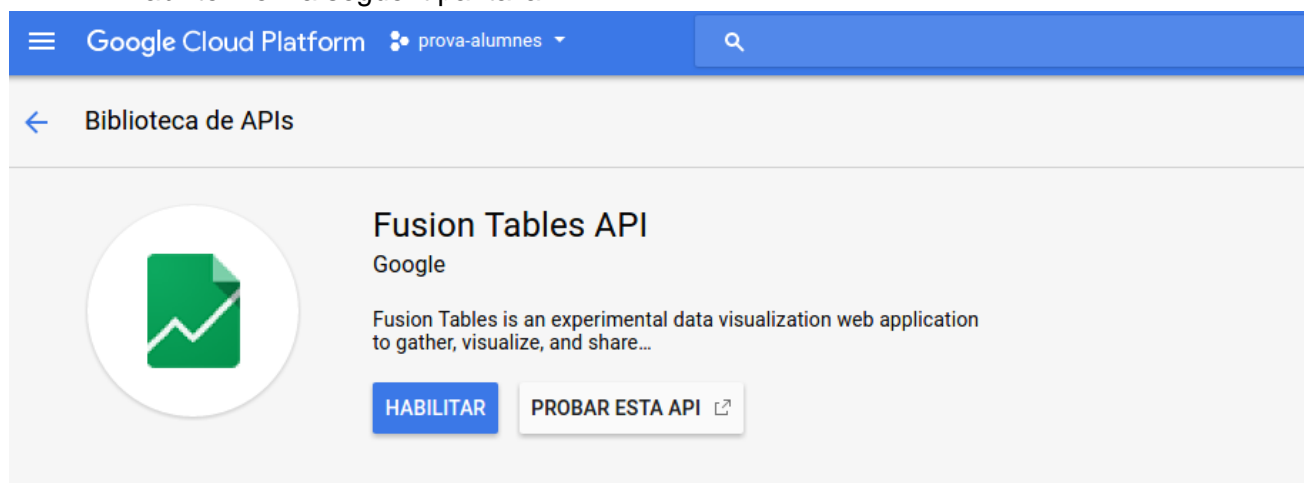
- En aquesta nova pantalla (següent imatge) cliquem Habilitar les APIs i serveis:



- Anem a parar a la biblioteca d'APIS. Haurem de buscar Fusion Tables API i, un cop trobada, seleccionar-la.



- L'habilitem en la següent pantalla:



**Tipo**

[APIs y servicios](#)

**Última actualización**

30/6/17 0:37

**Categoría**

[Otras](#)

**Nombre del servicio**

fusiontables.googleapis.com

**Visión general**

The Fusion Tables API lets you manage your data in Google Fusion Tables.

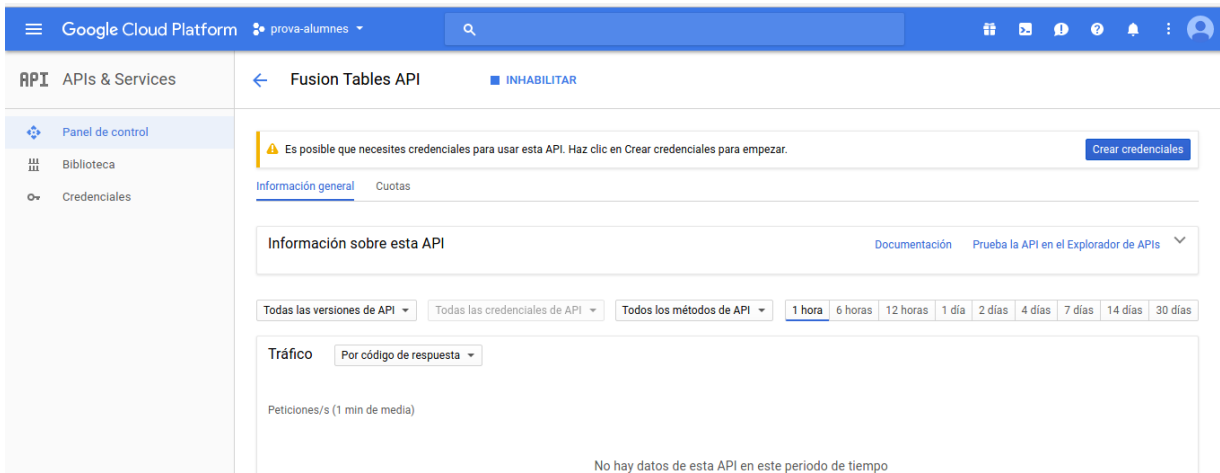
**Información sobre Google**

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

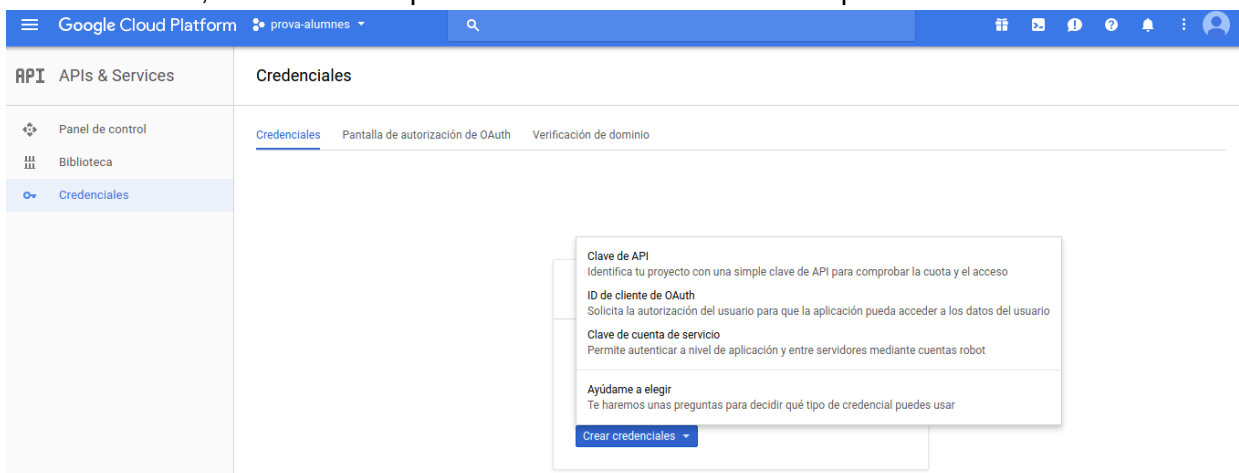
**Tutoriales y documentación**

[Learn more](#)

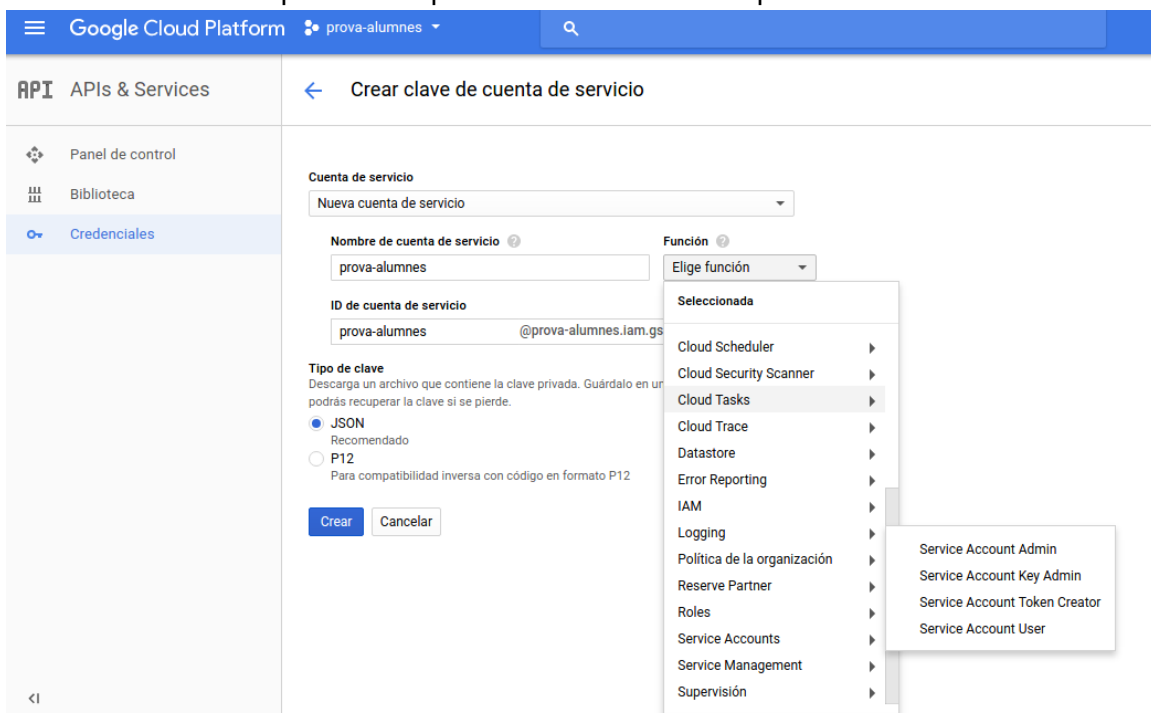
- Un cop habilitada l'API, hem de crear unes credencials que ens permetran utilitzar-la. Hem d'anar al menú de l'esquerra i triar l'opció Credenciales.



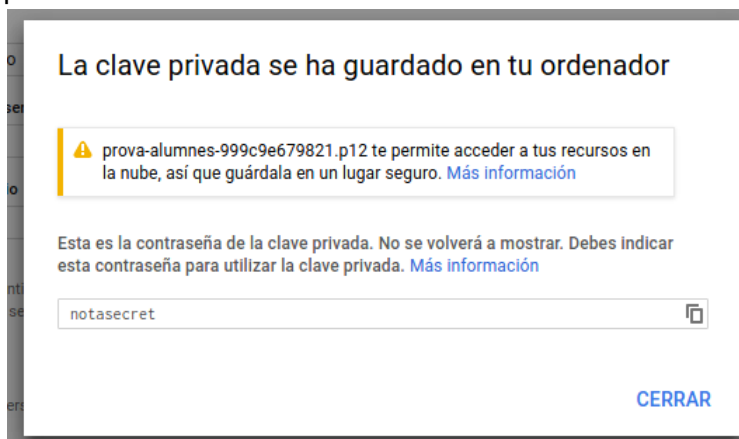
- Quan ho haguem fet ens trobarem aquesta nova pantalla on, al desplegable Crear credencial, hem de triar l'opció Clave de cuenta de servicio per crear una nova credencial:



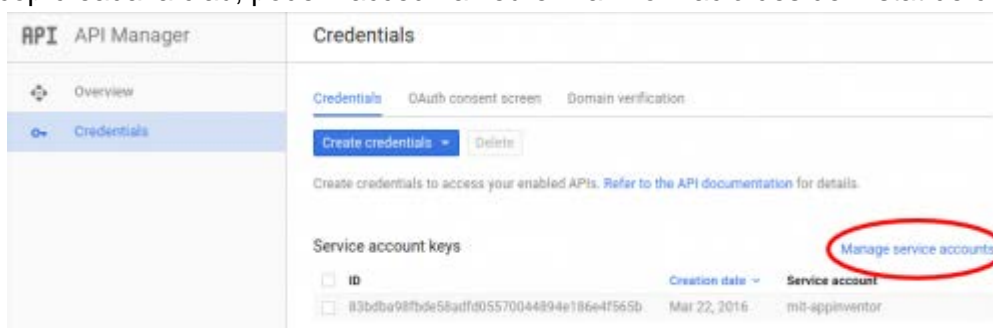
- Ens apareixerà aquesta nova pantalla on hem d'indicar que volem crear un nou Service account (o cuenta de servicio), escollim la funció Service Accounts Admin. Li hem de donar nom i hem de triar que la clau que estem creant és del tipus P12:



- El KeyFile s'ha generat i descarregat a l'ordinador. Aquest fitxer l'hem de desar ja que ens farà falta per tenir accés a les Fusion Tables. A més a més ens apareix aquesta finestra que ens avisa d'aquest fet:



- Un cop creada la clau, podem accedir a veure'n la informació des del llistat de credencials:



### 1.3. Accedir a les dades

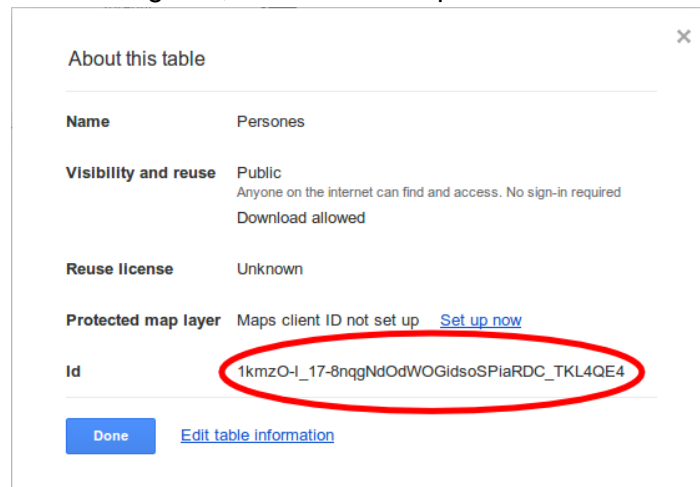
Per accedir a una Fusion Table ho farem via Service account tal i com hem vist a l'apartat anterior. Aquesta informació la utilitzarem des de les propietats del component FusionTable:

Properties	
FusiontablesControl1	
ApiKey	<input type="text"/>
KeyFile	APIProject-83bdba98fb
Query	<input type="text"/>
ServiceAccountEmail	mit-appinventor@api-pi
UseServiceAuthentication	<input checked="" type="checkbox"/>

De manera que a KeyFile hi pujarem el fitxer amb les claus que hem creat des dels Service accounts, al ServiceAccountEmail hi posarem l'adreça de correu electrònic associada i, finalment, marcarem la casella UseServiceAuthentication.



Hem de tenir també l'identificador de la Fusion Table en concret a la que volem accedir. Per obtenir aquesta informació, des de la Fusion Table hem d'anar al menú File i triar l'opció About this table. Obtindrem una pantalla com la següent, d'on hem de copiar la Id:

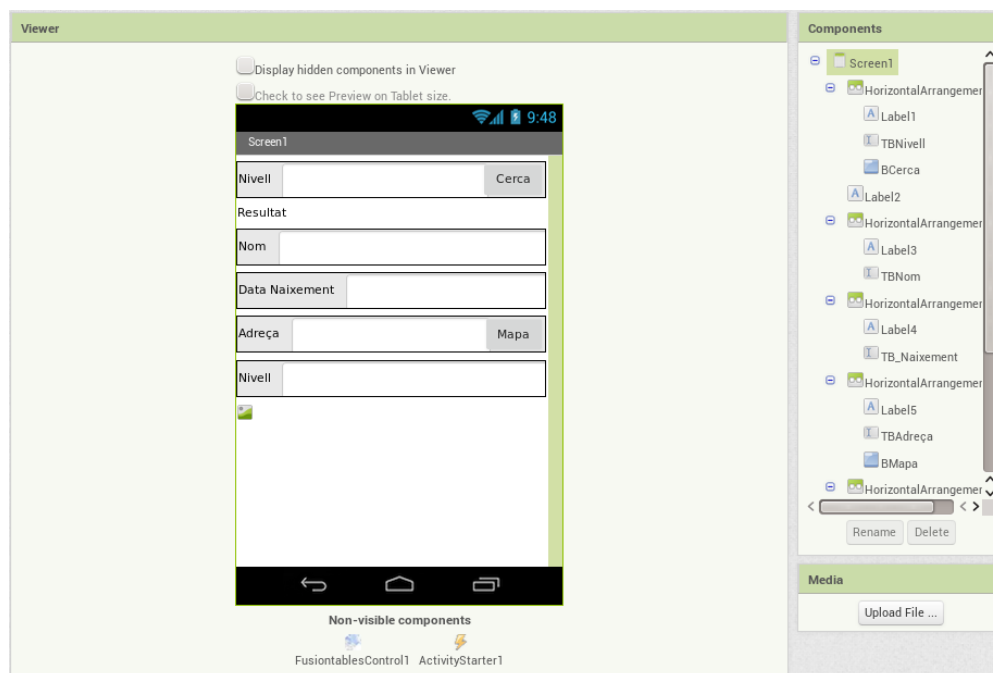


Quan tenim tota aquesta informació introduïda, el més pràctic és definir una constant a App Inventor per fer-ne referència sempre que la necessitem:

```
initialize global TAULA_ID to "1kmzO-I_17-8nqgNdOdWOGidsoSPiaRDC_TKL4QE4"
```

## Disseny de la pantalla

Què farem ara? Farem una consulta a la Fusion Table amb una sentència SQL, com a una base de dades, de manera que ens mostri un llistat de tots els registres que compleixin la condició que haguem definit, en triarem un i ens l'ensenyarà en una fitxa. El disseny de la pantalla a App Inventor serà la següent:



A part de les etiquetes i quadres de text que ens permetran visualitzar informació, també utilitzarem un objecte imatge mitjançant el qual mostrarem la foto corresponent a l'alumne consultat i un parell d'objectes no visibles: un objecte de tipus FusionTablesControl i un altre de tipus ActivityStarter.

Un altre element que no es visualitza a la pantalla és un UIPickerView, amb la propietat Visible a fals, que utilitzarem per recollir el resultat que ens retorni la consulta a la fusion table.

### Consulta de la base de dades

Per fer consultes sobre una base de dades s'utilitzen sentències SQL (Structured Query Language). La sintaxi d'una sentència SQL és la següent:

SELECT nom dels camps separats per comes, o un asterisc per tots els camps  
FROM nom de la taula  
WHERE criteris de selecció

Per exemple: SELECT Nom, Cognoms FROM Alumnes WHERE AnyNaixement=2000

En el nostre cas, veiem que a la part superior de la pantalla ens demana el nivell educatiu. Haurem de construir una sentència SQL de forma dinàmica en funció del contingut d'aquest TextBox i fer la consulta a la nostra Fusion Table. Això ho farem de la següent forma:

```
initialize global sql to ""
when BCerca.Click
do
  set global sql to join ["SELECT * FROM",
    get global TABLE_ID,
    "WHERE Nivell=",
    TBNivell.Text,
    ""]
  set FusiontablesControl1.Query to get global sql
  call FusiontablesControl1.SendQuery
```

A la sentència SQL que comença per SELECT \*, aquest asterisc vol dir tots els camps. Quan construïm aquesta sentència SQL és especialment important que anem amb compte a deixar els espais en blanc necessaris per tal de separar els termes de forma adequada. Així doncs, després del FROM i abans del WHERE hi haurà d'haver un espai:

```
initialize global registres to []
when FusiontablesControl1.GotResult
  result
do
  set global registres to list from csv table text get result
  remove list item list get global registres
  index 1
```

Com que volem poder seleccionar un registre, els afegirem tots a un UIPickerView. Un UIPickerView és una llista desplegable on es pot triar un element. El UIPickerView es pot omplir dinàmicament a partir

d'un origen de dades, o també es pot omplir amb valors escrits directament al codi amb la propietat Elements. Ara bé, en lloc d'afegir tots els camps, visualitzarem només el nom i l'etapa, que corresponen als camps 1 i 4. Això ho farem creant una llista buida en el ListPicker i, per a cadascun dels registres, hi afegirem el contingut dels camps seleccionats. Finalment obrirem el ListPicker per poder triar la persona que vulguem.

```

when FusionablesControl1 .GotResult
  result
do
  set global registres to list from csv table text get result
  remove list item list get global registres
  index 1
  set ListPicker1 . Elements to create empty list
  for each item in list get global registres
  do
    add items to list list ListPicker1 . Elements
    item
    join
    select list item list get item
    index 1
    " - "
    select list item list get item
    index 4
  call ListPicker1 .Open
  
```

Val a dir que si mirem el disseny de la pantalla no trobarem cap ListPicker, però si ens fixem bé en el llistat de components allà sí que n'hi ha un. Hem dissenyat l'app de manera que no hi ha cap necessitat d'accedir al ListPicker manualment, per la qual cosa l'hem definit com un component ocult. Això ho hem fet establint com a hidden la seva propietat Visible.

### Visualització de la informació

Un cop hem triat la persona que volíem del llistat, el que hem de fer ara és mostrar tota la seva informació en els camps corresponents. Això ho fem agafant el SelectionIndex del ListPicker i utilitzant-lo en la llista on tenim tots els registres.

```

when ListPicker1 .AfterPicking
do
  initialize local element to select list item list get global registres
  index ListPicker1 . SelectionIndex
  in
  set TBNom . Text to select list item list get element
  index 1
  set TB_Naixement . Text to select list item list get element
  index 2
  set TBAdreça . Text to select list item list get element
  index 3
  set TBNivell2 . Text to select list item list get element
  index 4
  set Image1 . Picture to select list item list get element
  index 5
  
```

La visualització de l'aplicació en un dispositiu mòbil quedaria de la següent forma:

Screen1

Nivell Segon  Cerca

Resultat

Nom Pere

Data Naixement 2010-06-01

Adreça Sant Salvador 30, Barcelona  Mapa

Nivell Segon

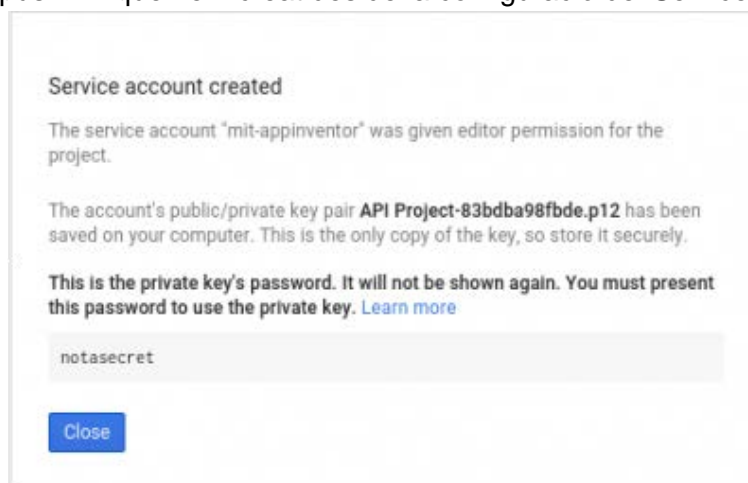


Finalment, i com que tenim l'adreça, hem afegit un botó en el disseny de la pantalla per situar en un mapa de Google aquesta adreça. Això ho fem de la següent manera:

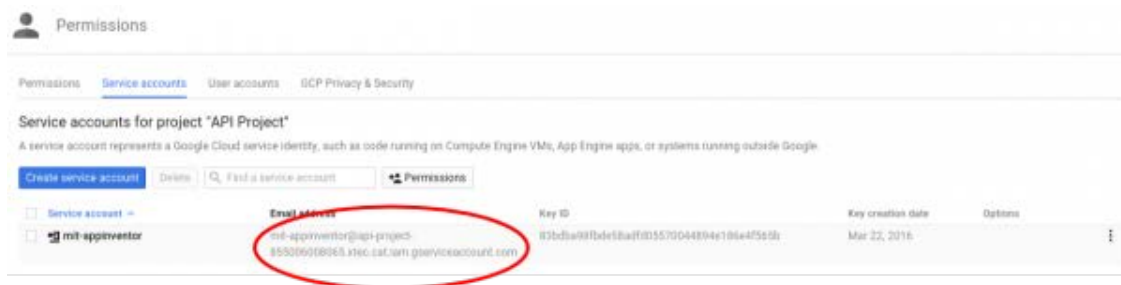
```
when BotoMapa .Click
do
  set ActivityStarter1 . Action to " android.intent.action.VIEW "
  set ActivityStarter1 . ActivityClass to " com.google.android.maps.MapActivity "
  set ActivityStarter1 . ActivityPackage to " com.google.android.apps.maps "
  set ActivityStarter1 . DataUri to join ( " geo:0,0?q= "
  TextAdressa . Text
  "?z=10 "
  call ActivityStarter1 .StartActivity
```

### Resum dels punts imprescindibles per accedir a les Fusion Tables

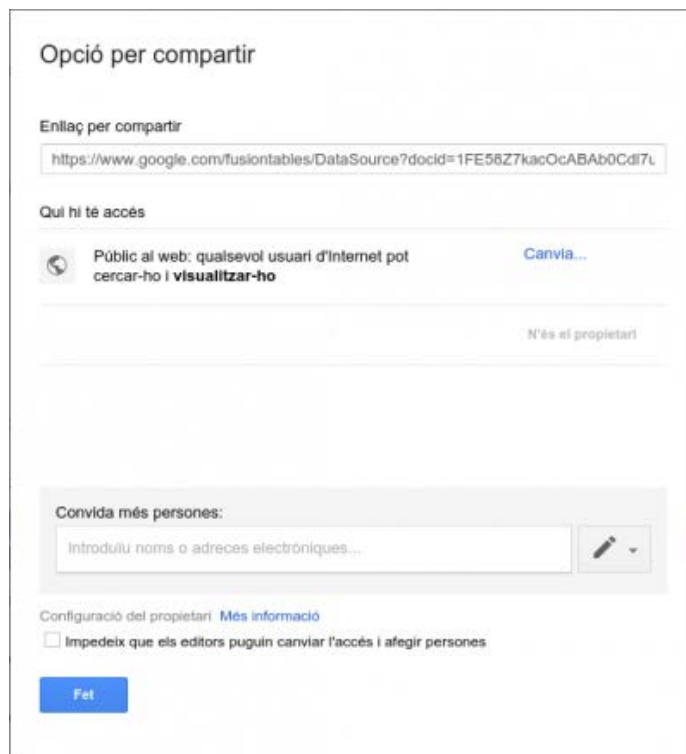
1. Guardar el fitxer tipus P12 que hem creat des de la configuració del Service Account:



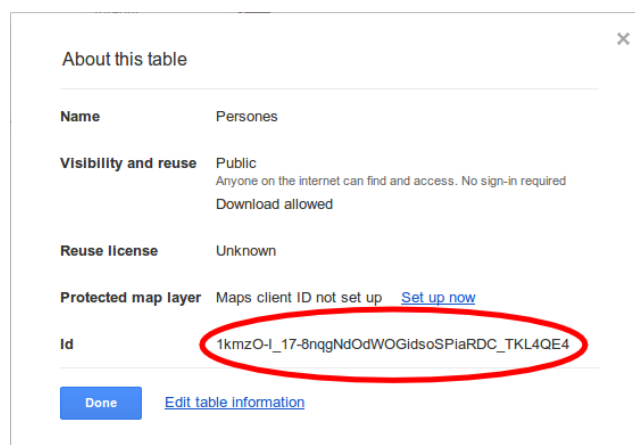
2. Copiar l'adreça de correu associada a la credencial que hem creat des de la configuració del Service Account:



3. Compartir la Fusion Table de manera que sigui pública a la web:



4. Des de la Fusion Table anar a File - About this table, verificar la visibilitat i copiar la Id, que serà el nom de la taula que utilitzarem des de l'App Inventor:



5. Des de l'App Inventor, anar a les propietats del component FusionTable:

- A la KeyFile pujar-hi el fitxer que hem generat en el punt 1 d'aquest resum.
- Al ServiceAccountEmail posar-hi l'adreça de correu que hem creat en el punt 2 d'aquest resum.
- Marcar l'opció UseServiceAuthentication.

6. Des de l'App Inventor utilitzar l'Id de la Fusion Table que hem copiat al punt 4 d'aquest resum com a nom de la taula sobre la que farem la consulta:

```
initialize global TAULA_ID to "1kmzO-I_17-8nqgNdOdWOGidsoSPiaRDC_TKL4QE4"
```

## 2. Dades Obertes

El concepte Dades Obertes (Open Data en anglès) són tots aquells conjunts de dades que les administracions posen a disposició del públic i poden ser reutilitzats i tornats a publicar sense cap restricció.

Les diferents administracions tenen un catàleg de dades obertes al servei de la ciutadania:

[Unió Europea](#)

[Generalitat de Catalunya](#)

[Diputació de Barcelona](#)

[Ajuntament de Barcelona](#)

### 2.1. Un exemple concret

La Diputació de Barcelona ofereix gratuïtament un llistat de totes les biblioteques de la província amb informació sobre ubicació, horaris, telèfons, etc...

Aprofitarem aquestes dades per implementar una app, que, a partir d'una població de la província de Barcelona o barri de Barcelona, mostri el llistat de les biblioteques que hi ha, i en triar-ne una, mostri tota la informació disponible per poder-hi contactar.

Aquestes dades en concret les podem trobar aquí:

[Dades obertes de la Diputació de Barcelona](#)

Altres diputacions també ofereixen dades similars del seu territori:

[Dades obertes de la Diputació de Girona](#)

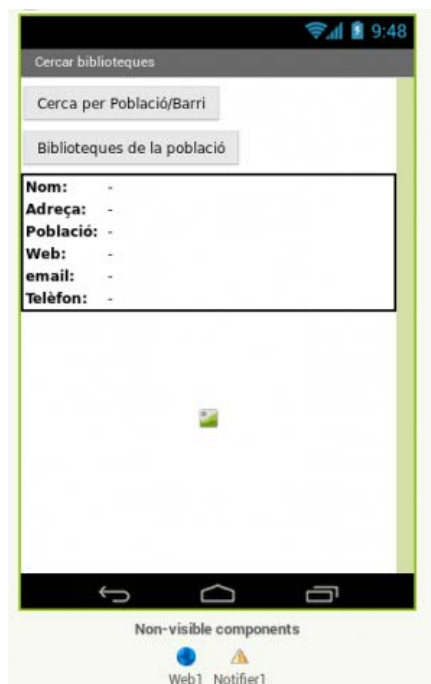
[Dades obertes de la Diputació de Lleida](#)

Dels diferents formats de fitxer disponibles, el format més fàcil de tractar des d'App Inventor és el CSV (Comma Separated Values), on cada línia és un registre, els camps estan separats per comes i entre cometes per si dins del contingut d'un camp hi ha una coma.

Per accedir a la informació sobre biblioteques que ens facilita la Diputació de Barcelona, utilitzarem el següent fitxer:

<http://do.diba.cat/api/dataset/biblioteques/format/csv>

Un cop ja tenim clar d'on obtindrem les dades, dissenyem l'aparença de l'aplicació. Proposem una interfície com la que segueix:

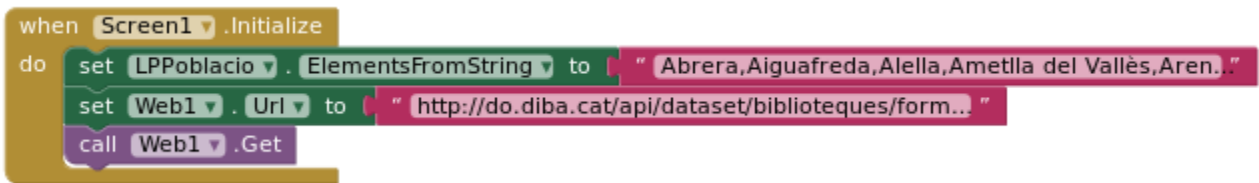


En la interfície podem veure un parell d'objectes ListPicker que ens permetran escollir de quina població o barri de Barcelona volem consultar les biblioteques que existeixen, i posteriorment un altre que ens permetrà escollir sobre quina biblioteca de la població o barri volem fer la consulta en cas que n'hi hagi més d'una.

En iniciar l'aplicació inicialitzarem les dades amb les quals treballarem:

- Emmagatzemarem en un ListPicker els pobles de la província de Barcelona i els barris de la seva capital.
- Farem una crida al procediment Get de l'objecte Web per obtenir el fitxer en format CSV amb les dades de les biblioteques de tota la província de Barcelona. A la propietat url de

l'objecte Web hi haurem emmagatzemat l'adreça d'on podem obtenir el fitxer CSV amb les dades.



Un cop obtenim la resposta a la petició Get, caldrà dir-li què volem que faci when Web GotText.

### Tractem les dades

Per poder utilitzar les dades de l'arxiu CSV amb App Inventor, ens caldrà editar el format per tal que el puguem utilitzar i cercar-hi les dades que ens interessin.

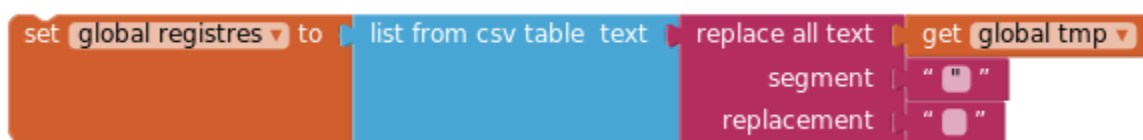
Per poder tractar aquest fitxer, primer n'eliminarem el màxim nombre possible de comes dins els camps i les dobles cometes a l'inici i final de cada camp:



Fixeu-vos que per eliminar tan sols les comes de dins els camps, deixem un espai darrera la coma. D'aquesta forma evitem esborrar les comes que separen els camps ja que porten tant al davant com al darrera dobles cometes.

### Mostrem el resultat

Un cop ja tenim el CSV amb els camps separats per comes, utilitzem la funció que proporciona App Inventor per convertir aquest CSV en una llista:



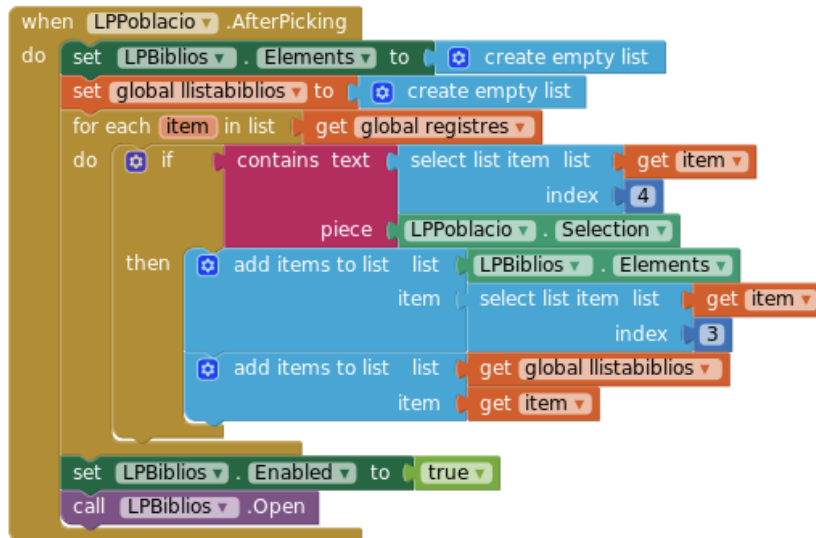


Per acabar d'adequar les dades eliminem el primer element de la llista ja que contindrà les capçaleres dels camps:



```
remove list item list index 1
get global registres
```

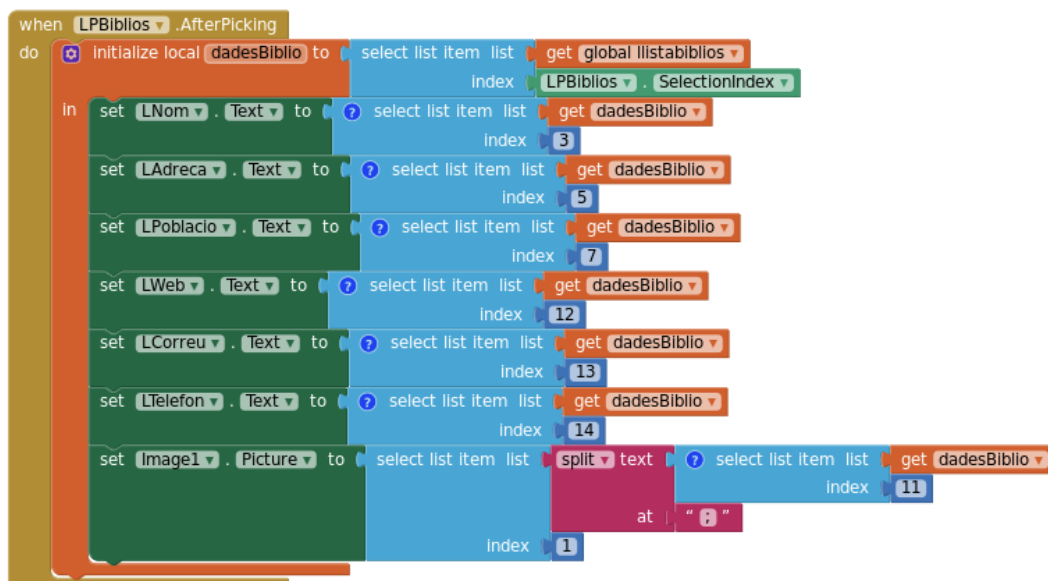
Un cop tinguem les dades carregades, esperarem a que l'usuari esculli una població per mostrar-li tan sols les biblioteques d'aquell municipi:



```
when LPPoblacio .AfterPicking
do
  set LPBiblios .Elements to create empty list
  set global llistabiblios to create empty list
  for each item in list
    get global registres
    do
      if contains text select list item list index 4 piece LPPoblacio .Selection
      then
        add items to list list LPBiblios .Elements item select list item list index 3
        add items to list list global llistabiblios item get item
    end
  end
  set LPBiblios .Enabled to true
  call LPBiblios .Open
```

En la imatge anterior veiem com en funció del municipi seleccionat, recorrem la llista de biblioteques emmagatzemada a la variable global registres i creem una llista amb les dades de les biblioteques de la població escollida. En el mateix moment, afegim els noms de les biblioteques del municipi a un ListPicker que ens permetrà escollir una biblioteca en concret i que obrim en el moment que ja tingui les biblioteques carregades.

Un cop que l'usuari ja ha escollit quina biblioteca vol visualitzar, aprofitant la propietat LPBiblios.SelectionIndex, podem cercar en la llista de biblioteques de la població la informació de la biblioteca en qüestió i mostrar-la mitjançant diferents etiquetes de text i un objecte imatge que tenim a la pantalla principal.



```
when LPBiblios .AfterPicking
do
  initialize local dadesBiblio to select list item list index LPBiblios .SelectionIndex get global llistabiblios
  in
    set LNom .Text to select list item list index 3 get dadesBiblio
    set LAdreca .Text to select list item list index 5 get dadesBiblio
    set LPoblacio .Text to select list item list index 7 get dadesBiblio
    set LWeb .Text to select list item list index 12 get dadesBiblio
    set LCorreu .Text to select list item list index 13 get dadesBiblio
    set LTelefon .Text to select list item list index 14 get dadesBiblio
    set Image1 .Picture to split text select list item list index 11 at : get dadesBiblio
```

### 3. Extensions d'App Inventor

Com hem vist en els blocs anteriors, App Inventor ofereix moltíssimes possibilitats gràcies a la gran varietat d'objectes dels quals disposa i que es van actualitzant. Si ens trobem però que hi ha alguna tasca en concret que volem resoldre però no disposem de cap objecte que ens permeti fer-ho, tenim dues opcions:

- Cercar una API que ens pugui resoldre la tasca i cridar-la mitjançant l'objecte Web.
- Utilitzar les Extensions d'App Inventor.

#### 3.1. Què són les extensions d'App Inventor?

Les extensions són objectes que podem agregar a un projecte d'App Inventor i que ofereixen funcionalitats que els objectes dels quals disposa App Inventor no tenen. Aquestes extensions estan desenvolupades en JAVA, tot i que per utilitzar-les, nosaltres les veurem com qualsevol altre objecte de l'entorn de programació. Un cop incorporades al nostre projecte, les podrem trobar a la llibreria Extensions de la barra lateral esquerra.

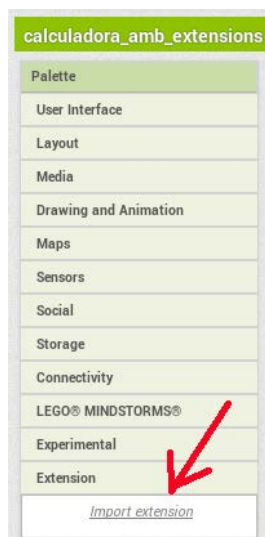
El propi MIT ofereix una llibreria d'extensions que podem afegir a les nostres aplicacions. Si hi accediu, veureu que és un llista força curta tot i que totes elles estan comprovades amb la darrera versió d'App Inventor.

Per mostrar el funcionament de les extensions proposem el desenvolupament d'una calculadora que utilitza una extensió que interpreta i resol expressions matemàtiques molt diverses.

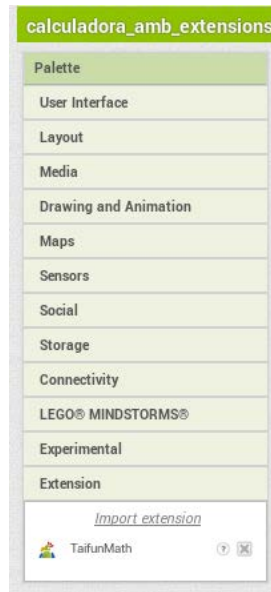
#### Preparem els objectes

Utilitzarem però una extensió que supera les funcions que ofereix aquesta biblioteca pròpia d'App Inventor. L'extensió que utilitzarem serà [Math Extensions](#) de [Pura Vida Apps](#). Per poder-la agregar al nostre projecte cal descarregar l'arxiu [com.puravidaapps.TaifunMath.aix](http://com.puravidaapps.TaifunMath.aix) o emmagatzemar-ne la url.

Un cop ho hagem fet, cal obrir un nou projecte i hi afegirem l'extensió tal com mostra la següent pantalla:



Cliquem l'opció assenyalada, Import extension, i cerquem l'arxiu com.puravidaapps.TaifunMath.aix que hem descarregat en el nostre ordinador per incorporar-lo en el projecte. També podem facilitar la url per tal que el sistema carregui directament l'extensió. Al cap d'uns instants, veurem que disposem ja d'un nou objecte anomenat TaifunMath:



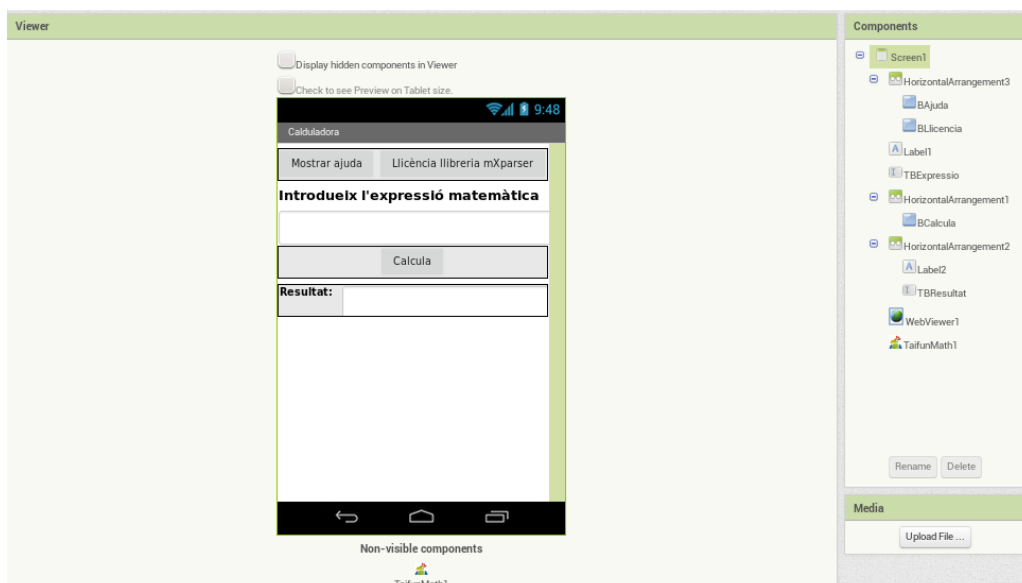
Aquest objecte rebirà una expressió matemàtica, i utilitzant la llibreria [mXparser](#), l'analitzarà, la resoldrà i ens en retornarà el resultat. Si mirem les funcions que ens ofereix, són molt senzilles però suficients donada la potència de la biblioteca que utilitza:



La funció TaifunMath1.Expression ens retornarà el resultat d'una expressió matemàtica que li facilitem, i TaifunMath1.License ens mostrarà les condicions d'ús de la biblioteca mXparser.

## Dissenyem la interfície

Bàsicament ens caldrà un quadre de text on introduir expressions matemàtiques, un botó que ens servirà per enviar l'operació i un altre espai on rebrem la resposta. La proposta d'interfície que fem és la següent:

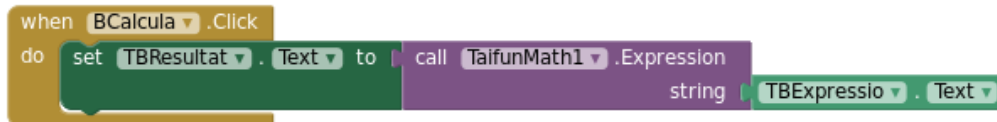


Adicionalment tenim també un botó que ens permetrà mostrar l'ajuda en línia de la biblioteca a través d'un objecte WebView i un altre que ens mostrarà les condicions d'ús de la biblioteca. També podem veure que hem incorporat un objecte de tipus TaifunMath no visible.

## Programem els objectes

La programació d'aquests objectes també és d'allò més senzilla ja que tan sols hem de captar els clicks sobre els diferents botons per actuar en conseqüència.

Quan cliquem el botó de calcular,



```
when BCalcula .Click
do
  set TBResultat . Text to
  call TaifunMath1 .Expression
  string TBEexpressio . Text
```

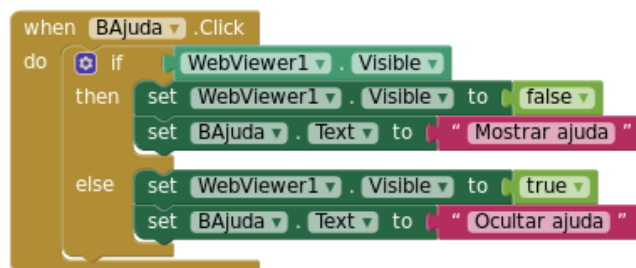
cridem a la funció TaifunMath1.Expression facilitant-li la expressió entrada per l'usuari. El resultat de la funció el mostrem en el camp de text TBResultat.

Quan cliquem el botó per veure la llicència la mostrem també en el camp de text TBResultat:



```
when BLlicencia .Click
do
  set TBResultat . Text to
  TaifunMath1 . License
```

I per acabar, quan cliquem el botó d'ajuda podem veure indicacions sobre les funcions matemàtiques que es poden resoldre i la seva sintaxi.



```
when BAJuda .Click
do
  if
  then
    set WebView1 . Visible to false
    set BAJuda . Text to "Mostrar ajuda"
  else
    set WebView1 . Visible to true
    set BAJuda . Text to "Ocultar ajuda"
```

Aquest darrer botó serveix per fer visible o invisible l'objecte WebView1 gràcies al qual podrem navegar per la pàgina amb informació sobre les possibilitats de la biblioteca mXparser..

Si voleu experimentar amb altres extensions, en el següent enllaç a [Pura Vida apps](#) hi podeu trobar extensions molt diverses, i si voleu més informació sobre el seu funcionament i com es programen, ho podeu consultar a [App Inventor Extensions](#).