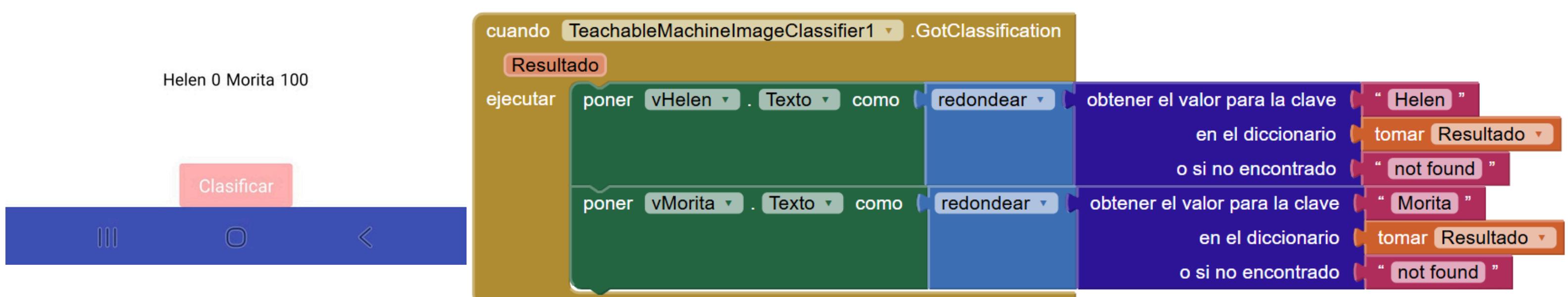
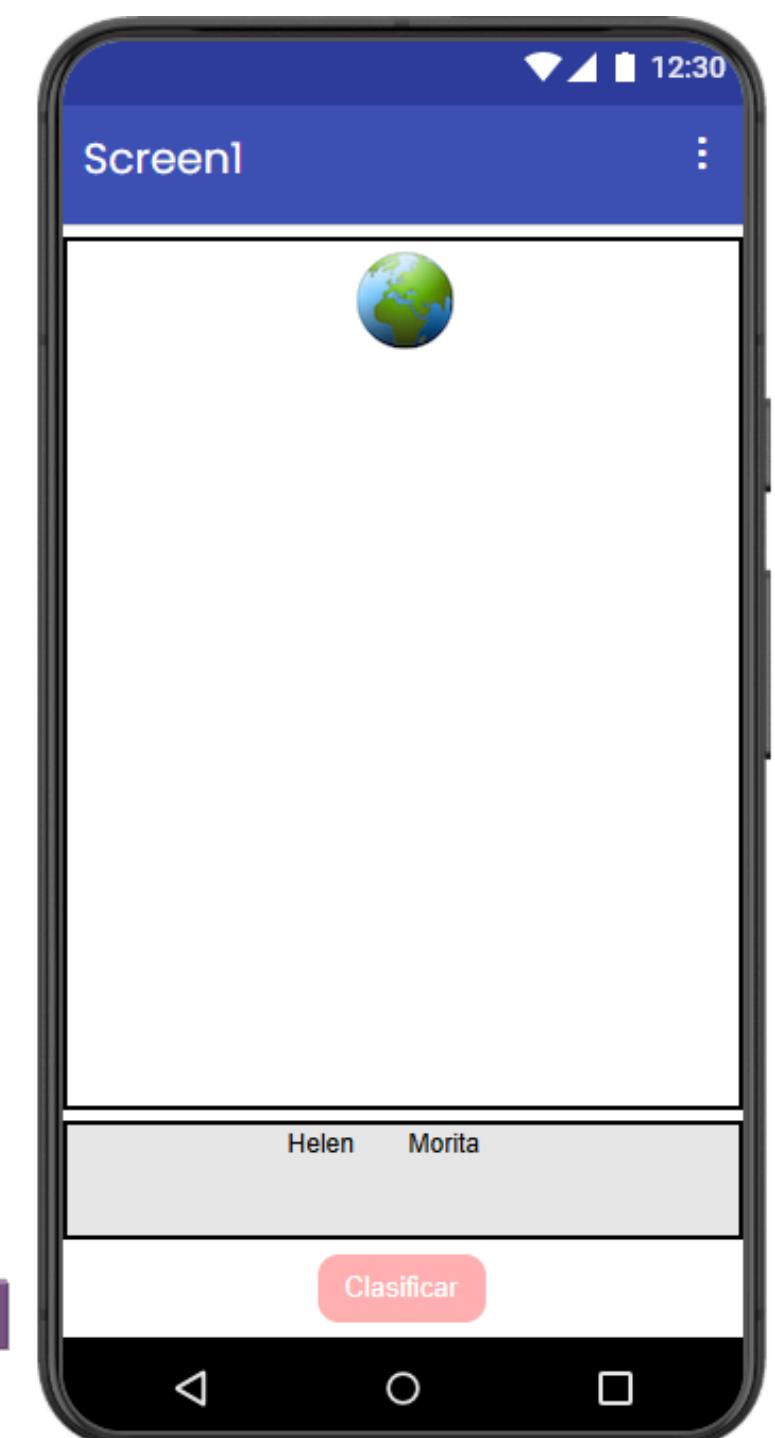
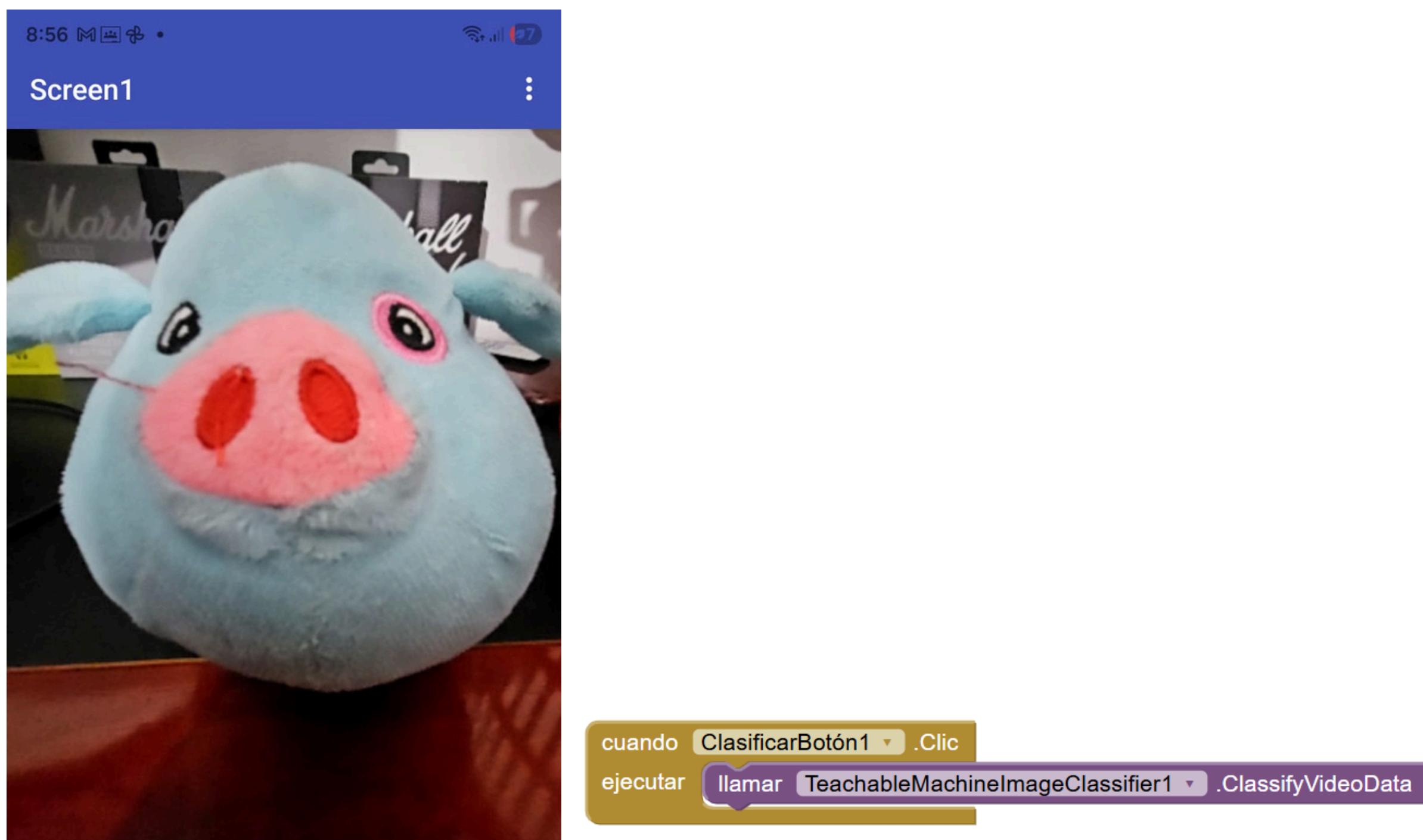


Clasificador App



Password Purr

Contraseña

Coloca tu contraseña para acceder

Entrar

MiLista

Mi Lista

Recordatorios: Apagar la estufa

Compras: huevo, jamón, queso

Recordatorios: Apagar la estufa

Colocar Tarea	Eliminar Tarea	Eliminar Todo
---------------	----------------	---------------

Código de bloques:

```

cuando BotonContraseña .Clic
ejecutar
  si CampoDeContraseña1 .Texto = " 2244 "
    entonces abrir otra pantalla Nombre de la pantalla MiLista
  else poner CampoDeContraseña1 .Texto como " "
  end

inicializar global Mi_Lista como crear una lista vacía

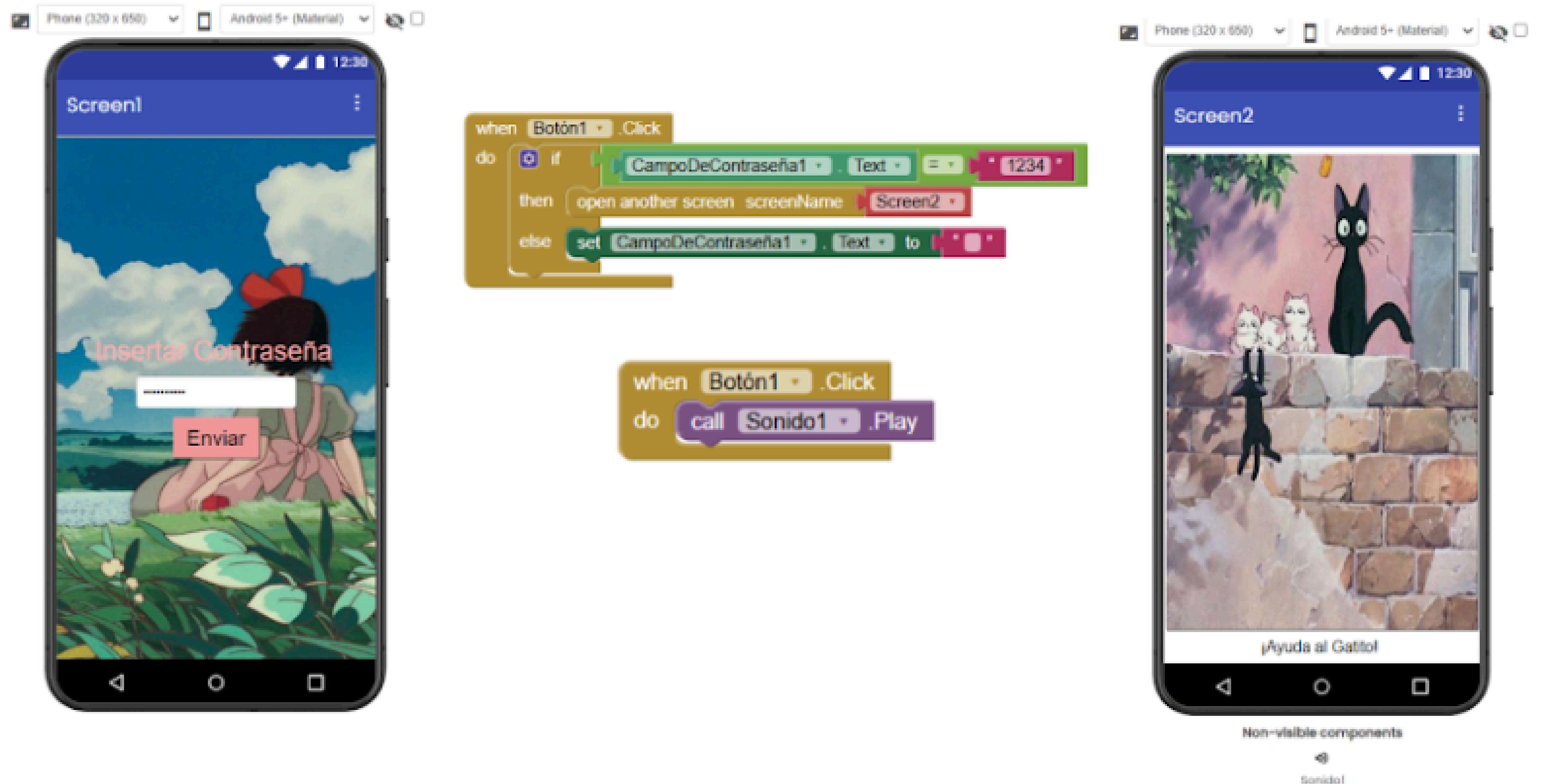
cuando ColocarTareaBotón1 .Clic
ejecutar
  añadir elementos a la lista lista tomar global Mi_Lista
  elemento ToDoCampoDeTexto1 .Texto
  llamar TinyBD1 .GuardarValor
  etiqueta " MiLista "
  valorAGuardar tomar global Mi_Lista
  poner ToDoListVisorDeLista1 .Elementos como tomar global Mi_Lista

cuando EliminarTodoBotón1 .Clic
ejecutar
  El usuario tocó y soltó el botón.
  Lista a crear una lista vacía
  llamar TinyBD1 .LimpiarEtiqueta
  etiqueta " MiLista "
  poner ToDoListVisorDeLista1 .Elementos como crear una lista vacía

cuando BorrarTareaBotón1 .Clic
ejecutar
  si ToDoListVisorDeLista1 .ÍndiceSelección ≠ 0
    entonces eliminar elemento de la lista lista tomar global Mi_Lista
      indice ToDoListVisorDeLista1 .ÍndiceSelección
      llamar TinyBD1 .GuardarValor
      etiqueta " MiLista "
      valorAGuardar tomar global Mi_Lista
      poner ToDoListVisorDeLista1 .Elementos como tomar global Mi_Lista
      poner ToDoListVisorDeLista1 .ÍndiceSelección como 0
    end
  end

cuando MiLista .Inicializar
ejecutar
  poner global Mi_Lista a
  llamar TinyBD1 .ObtenerValor
  etiqueta " MiLista "
  valorSiEtiquetaNoExiste crear una lista vacía
  poner ToDoListVisorDeLista1 .ElementColor como tomar global Mi_Lista
  
```

Password Purr



My List



```

cuando [BotonContraseña v].Clic
ejecutar
  si [CampoDeContraseña1 v].Texto = "2244"
    entonces abrir otra pantalla [Nombre de la pantalla v] MiLista
  else poner [CampoDeContraseña1 v].Texto como ""

```



```
inicializar global [Mi_Lista] como [crear una lista vacía]
```

```

cuando [ColocarTareaBotón1 v].Clic
ejecutar
  añadir elementos a la lista [lista v]
    tomar [global Mi_Lista]
    elemento [ToDoCampoDeTexto1 v].Texto
  llamar [TinyBD1 v].GuardarValor
    etiqueta "MiLista"
    valorAGuardar [tomar [global Mi_Lista]]
  poner [ToDoListVisorDeLista1 v].Elementos como [tomar [global Mi_Lista]]

```

```

cuando [EliminarTodoBotón1 v].Clic
ejecutar
  El usuario tocó y soltó el botón.
  [lista v] a [crear una lista vacía]
  llamar [TinyBD1 v].LimpiarEtiqueta
    etiqueta "MiLista"
  poner [ToDoListVisorDeLista1 v].Elementos como [crear una lista vacía]

```

```

cuando [BorrarTareaBotón1 v].Clic
ejecutar
  si [ToDoListVisorDeLista1 v].ÍndiceSelección ≠ 0
    entonces
      eliminar elemento de la lista [lista v]
        tomar [global Mi_Lista]
        índice [ToDoListVisorDeLista1 v].ÍndiceSelección
      llamar [TinyBD1 v].GuardarValor
        etiqueta "MiLista"
        valorAGuardar [tomar [global Mi_Lista]]
      poner [ToDoListVisorDeLista1 v].Elementos como [tomar [global Mi_Lista]]
      poner [ToDoListVisorDeLista1 v].ÍndiceSelección como 0

```

```

cuando [MiLista v].Inicializar
ejecutar
  poner [global Mi_Lista] a []
  llamar [TinyBD1 v].ObtenerValor
    etiqueta "MiLista"
    valorSiEtiquetaNoExiste [crear una lista vacía]
  poner [ToDoListVisorDeLista1 v].ElementColor como [tomar [global Mi_Lista]]

```

Track My Mood

```

when Screen1 .Initialize
do set Web1 . Url to "https://generativelanguage.googleapis.com/v1beta1/documents:analyzeText"
set Web1 . RequestHeaders to make a dictionary | key "x-goog-api-key" value "AlzaSyCEQMQtHCm8ESk7aojcg_fBX6VtA0WNco" | key "Content-Type" value "application/json"

when BtnPreguntar .Click
do call Web1 . PostText
    text make a dictionary | key "contents" value make a dictionary | key "parts" value make a dictionary | key "text" value CajaPregunta . Text

when Web1 . GotText
url responseCode responseType responseContent
do if get responseCode = 200
then set CajaRespuesta . Text to get value at key path "candidates" | 1 | content | parts | 1 | text
in dictionary call Web1 . JsonTextDecodeWithDictionaries jsonText get responseContent
or if not found "not found"
else set CajaRespuesta . Text to get responseContent

when BtnLimpiar .Click
do set CajaPregunta . Text to ""

```



Robot que responde preguntas

cuál es el país más grande del mundo?

Preguntar

Limpiar

El país más grande del mundo por **superficie terrestre** es **Rusia**.

Cubre aproximadamente 17.1 millones de kilómetros cuadrados, abarcando una vasta extensión de Eurasia.

Robot que responde preguntas

Qué es IA?

Preguntar

Limpiar

IA son las siglas de **Inteligencia Artificial**.

En términos sencillos, la Inteligencia Artificial es un campo de la informática que busca crear máquinas o programas capaces de **pensar, razonar, aprender y actuar** de una manera que normalmente asociaríamos con la inteligencia humana.

Imagina que una máquina pudiera:
* **Aprender** de la experiencia (como un niño aprendiendo a sumar).
* **Resolver problemas** (como encontrar la mejor ruta en un mapa).

Robot que responde preguntas

Cuéntame un dato curioso que sea sorprendente (breve)

Preguntar

Limpiar

¡Aquí tienes uno que vuela la cabeza!

Cleopatra vivió más cerca en el tiempo de la invención del iPhone que de la construcción de las Grandes Pirámides de Giza.

Robot que responde preguntas

Cuéntame un dato curioso que sea sorprendente

Preguntar

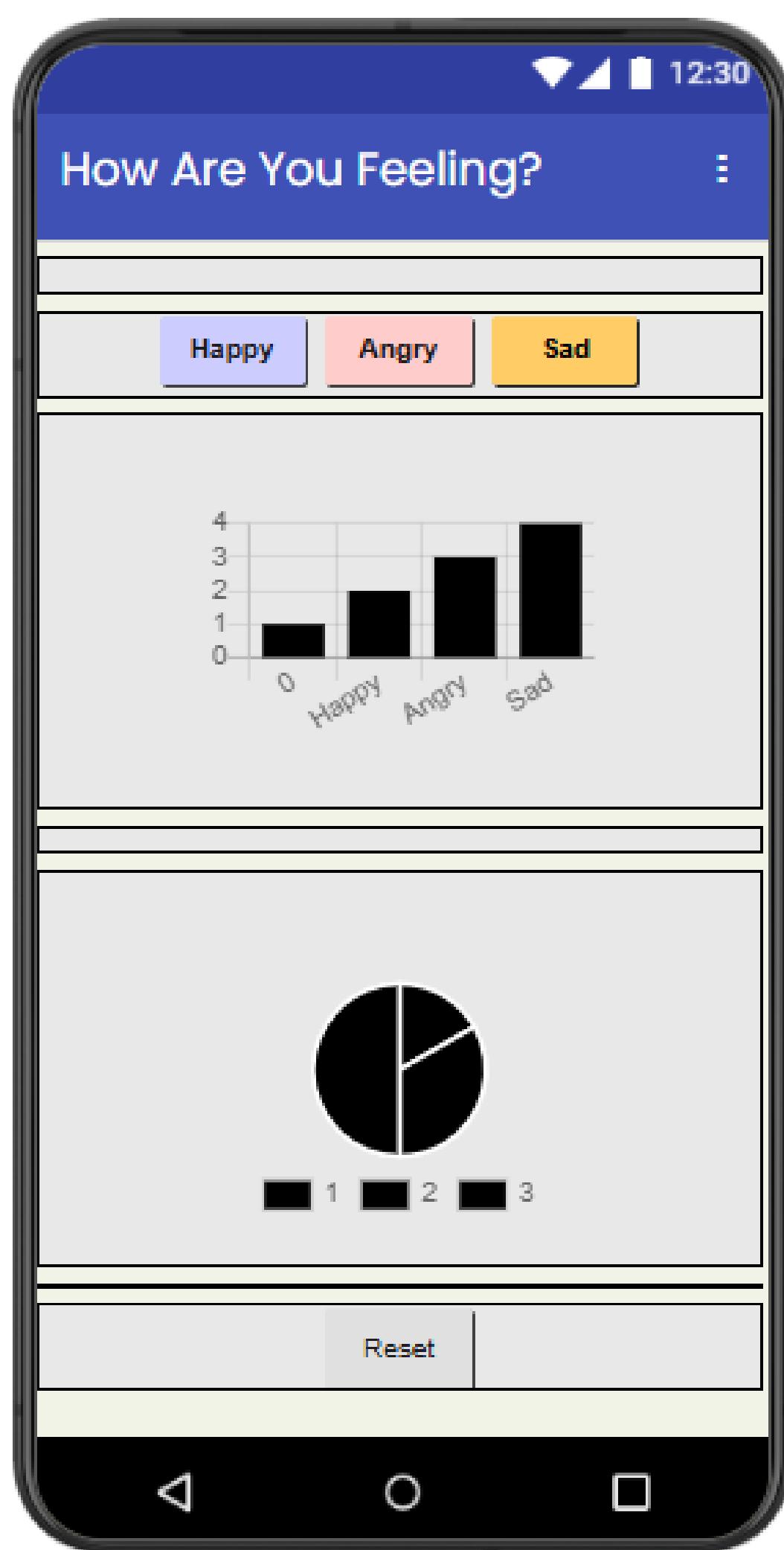
Limpiar

¡Aquí tienes uno que te volará la cabeza!

Los pulpos tienen TRES corazones.
* **Dos de ellos** bombean sangre a las branquias (para oxigenarla).
* Y el **tercero** la bombea al resto del cuerpo.

Además, su sangre no es roja como la nuestra, sino de **color azul**. Esto se debe a que, en lugar de hierro (como la hemoglobina en nuestra sangre), usan

Track My Mood



```
when Screen1.Initialize
do set BarGraphData2D .Colors to make a list
set pieGraphData2D .Colors to make a list
```

```
initialize global happyButtonTotal to 0
initialize global angryButtonTotal to 0
initialize global sadButtonTotal to 0
```

```
when happyButton .Click
do set global happyButtonTotal to get global happyButtonTotal + 1
call UpdateCharts

when angryButton .Click
do set global angryButtonTotal to get global angryButtonTotal + 1
call UpdateCharts

when sadButton .Click
do set global sadButtonTotal to get global sadButtonTotal + 1
call UpdateCharts

when resetButton .Click
do call BarGraphData2D .Clear
call pieGraphData2D .Clear
set global happyButtonTotal to 0
set global angryButtonTotal to 0
set global sadButtonTotal to 0
```

```
to UpdateCharts
do call BarGraphData2D .Clear
call BarGraphData2D .ImportFromList
list make a list
make a list 1 get global happyButtonTotal
make a list 2 get global angryButtonTotal
make a list 3 get global sadButtonTotal

call pieGraphData2D .Clear
call pieGraphData2D .ImportFromList
list make a list
make a list "Happy" get global happyButtonTotal
make a list "Angry" get global angryButtonTotal
make a list "Sad" get global sadButtonTotal
```

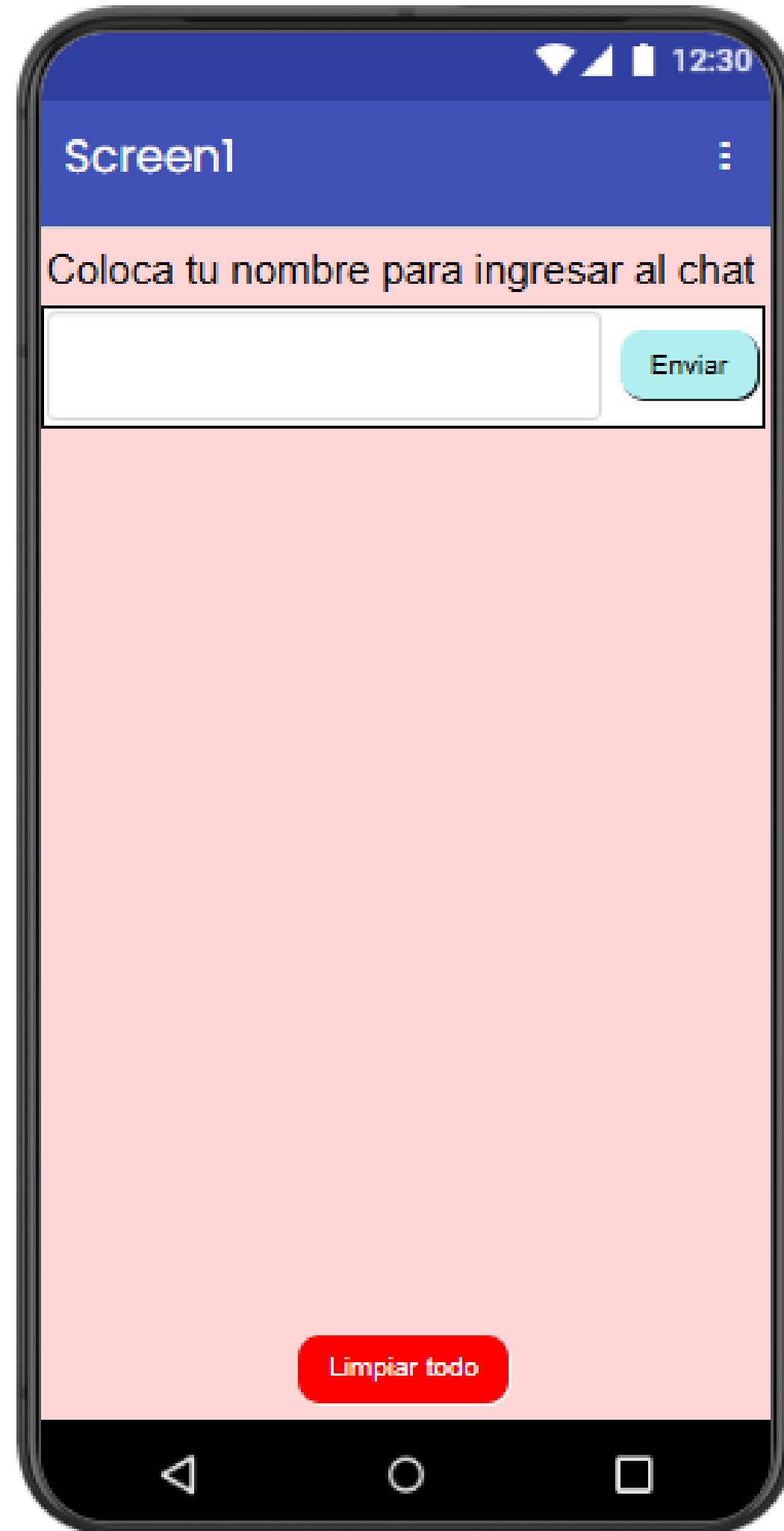
CloudDB chatAPP

```

initialize global [UserName] to [" "]
initialize global [SehaUnido] to [false]
initialize global [ChatList] to [create empty list]

when [EnviarBotón1].Click
do
  if [not get [global UserName]]
    then
      if [not is empty [CampoDeTexto1].Text]
        then set [global UserName] to [CampoDeTexto1].Text
        set [global SehaUnido] to [true]
        set [Etiqueta1].Text to ["Escribe tu mensaje y presiona enviar"]
      else call [CloudDB1].AppendValueToList
        tag: "Chat"
        itemToAdd: [join [get [global UserName]] [" "]] [CampoDeTexto1].Text
    else call [CloudDB1].AppendValueToList
      tag: "Chat"
      itemToAdd: [join [get [global UserName]] [" "]] [CampoDeTexto1].Text
  set [CampoDeTexto1].Text to [" "]

```



```

when [CloudDB1].GotValue
  tag: value
do
  if [get [tag] = "chat"]
    then set [global ChatList] to [get [value]]
  set [VisorDeLista1].Elements to [get [global ChatList]]

```

```

when [CloudDB1].DataChanged
  tag: value
do
  if [get [tag] = "chat"]
    then set [global ChatList] to [get [value]]
  set [VisorDeLista1].Elements to [get [global ChatList]]

```

```

when [LimpiarTodoBotón1].Click
do
  set [global ChatList] to [create empty list]
  set [global UserName] to [" "]
  set [global SehaUnido] to [false]
  set [Etiqueta1].Text to ["Coloca tu nombre para ingresar al chat"]
  set [CampoDeTexto1].Text to [" "]
  set [VisorDeLista1].Elements to [create empty list]
  call [CloudDB1].ClearTag
    tag: "chat"

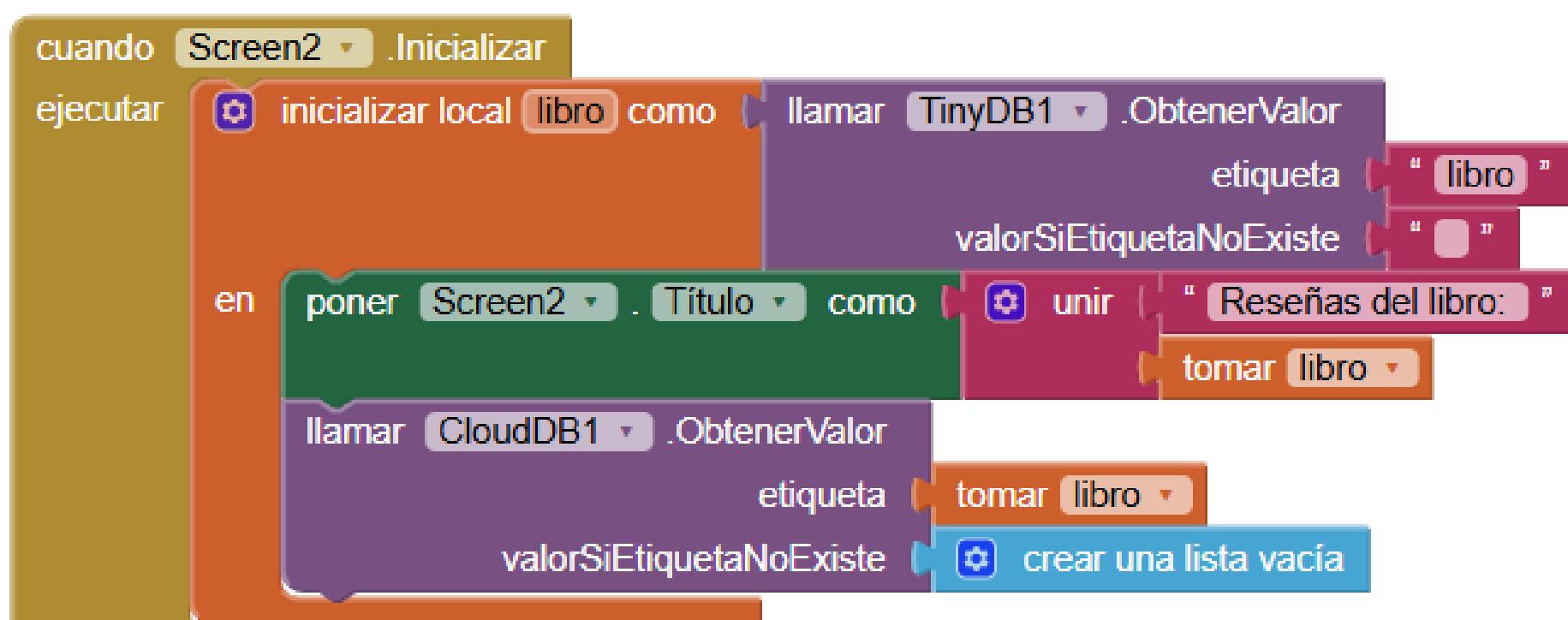
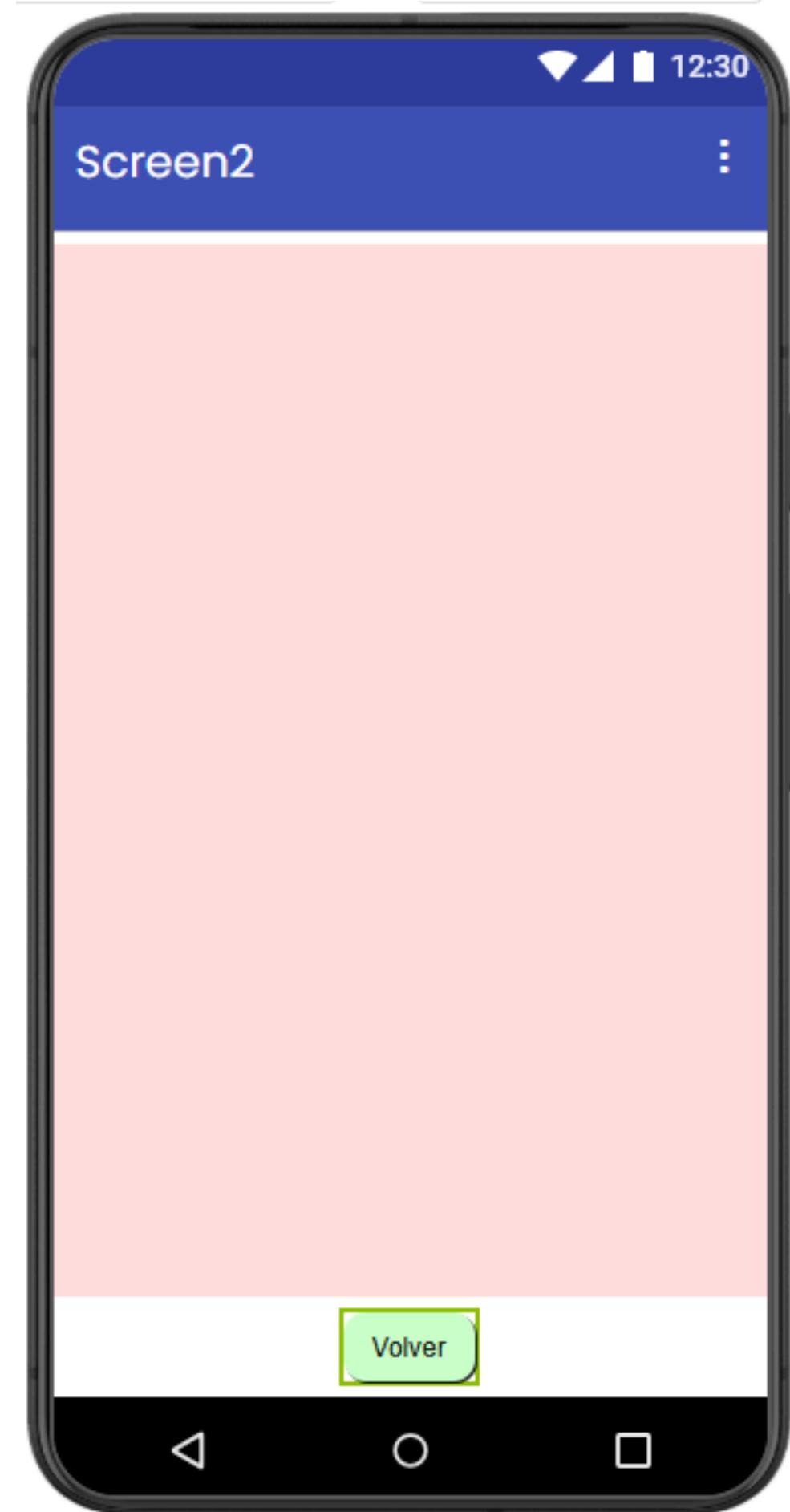
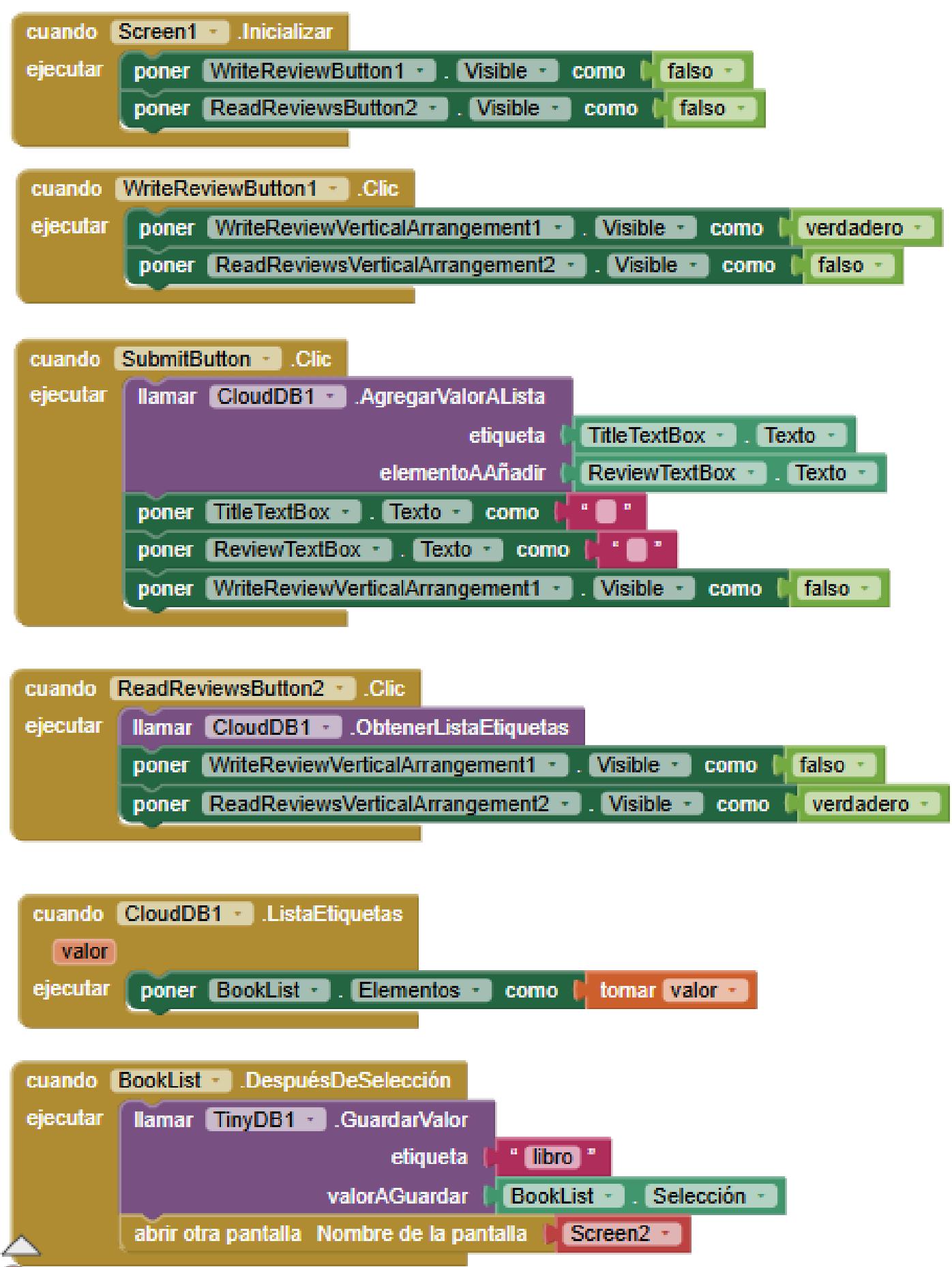
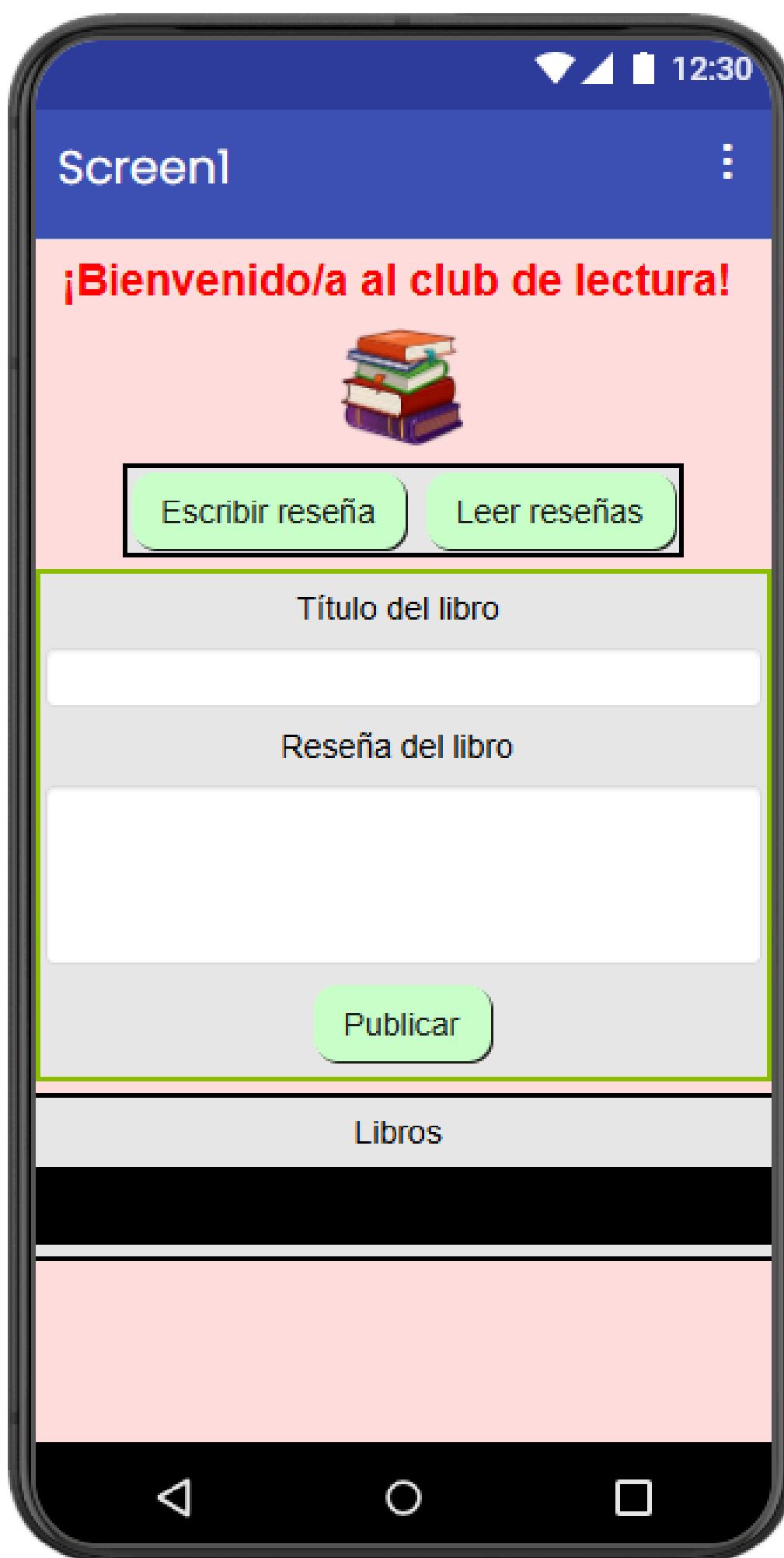
```

```

when [Screen1].Initialize
do
  call [CloudDB1].GetValue
    tag: "chat"
    valueIfTagNotThere: [create empty list]

```

DIY Book Club



Components not visible
CloudDB1 TinyDB1

Proyecto IceMelt

Scratch Script (Top Left):

```

when [homeButton v].Click
do [open another screen screenName v Screen1]
when [showDataButton v].Click
do [call [topChartData2D v].Clear]
[call [spreadsheet1 v].ReadSheet
sheetName v "Spirit Lake"]
[call [spreadsheet2 v].ReadSheet
sheetName v "Otsego Lake"]
when [spreadsheet1 v].GotSheetData
sheetData
do [set [topChartLabel v].Text v to "Spirit Lake, Orleans, Iowa"]
[call [topChartData2D v].ImportFromSpreadsheet
spreadsheet v spreadsheet1
xColumn v "Year"
yColumn v "Ice"
useHeaders v true]
when [spreadsheet2 v].GotSheetData
sheetData
do [set [topChartLabel v].Text v to "Otsego Lake, Cooperstown, NY"]
[call [bottomChartData2D v].ImportFromSpreadsheet
spreadsheet v spreadsheet2
xColumn v "Year"
yColumn v "Ice"
useHeaders v true]
when [homeButton v].Click
do [open another screen screenName v Screen1]
when [showDataButton v].Click
do [call [topChartData2D v].Clear]
[call [topChartData2D v].Clear]
[call [spreadsheet1 v].ReadSheet
sheetName v "Spirit Lake"]
when [spreadsheet1 v].GotSheetData
sheetData
do [set [topChartLabel v].Text v to "Spirit Lake, Orleans, Iowa"]
[call [topChartData2D v].ImportFromSpreadsheet
spreadsheet v spreadsheet1
xColumn v "Year"
yColumn v "Ice"
useHeaders v true]
[call [bottomChartData2D v].ImportFromSpreadsheet
spreadsheet v spreadsheet1
xColumn v "Year"
yColumn v "Temp"
useHeaders v true]

```

Visualize Climate Data Application (Top Right):

Scratch Script (Bottom Left):

```

when [singleCaseScreenButton v].Click
do [open another screen screenName v singleCaseScreen]
when [multicaseScreenButton v].Click
do [open another screen screenName v MultiCaseScreen]

```



```

when homeButton .Click
do open another screen screenName [Screen1]

when showDataButton .Click
do call cleanedChartData2D .Clear
call dataCleaningChartData2D .Clear
call spreadsheet1 .ReadSheet
sheetName [Spirit Lake]

when spreadsheet1 .GotSheetData
sheetData
do set topChartLabel .Text to [Spirit Lake, Orleans, Iowa]
call cleanedChartData2D .ImportFromSpreadsheet
spreadsheet [spreadsheet1]
xColumn [Year]
yColumn [Ice]
useHeaders [true]
call dataCleaningChartData2D .ImportFromSpreadsheet
spreadsheet [spreadsheet1]
xColumn [Year]
yColumn [Ice]
useHeaders [true]

when drawLineOfBestFitButton .Click
do set Trendline1 .ChartData to [cleanedChartData2D]

when Trendline1 .Updated
results
do set SlopeValueLabel .Text to [Trendline1 .LinearCoefficient]
set Y_intValueLabel .Text to [round [Trendline1 .YIntercept]]
set Cor_coefValueLabel .Text to [Trendline1 .CorrelationCoefficient]
set X_intValueLabel .Text to [round [Trendline1 .XIntercept]]
call cleanedChartData2D .ExtendDomainToInclude
x [Trendline1 .XIntercept]

```

```

when detectAnomaliesButton .Click
do call dataCleaningChartData2D .HighlightDataPoints
dataPoints [color red]
call AnomalyDetection1 .DetectAnomaliesInChartData
chartData [dataCleaningChartData2D]
threshold [2]

when dataCleaningChartData2D .EntryClick
x [x get x]
y [y get y]
do call dataCleaningChartData2D .RemoveEntry
x [get x]
y [get y]
call cleanedChartData2D .Clear
call cleanedChartData2D .ImportFromList
list [call dataCleaningChartData2D .GetAllEntries]

when AiAnalysisButton .Click
do set dataCleaningChart .Visible to [false]
set dataCleaningHorizontalArrangement .Visible to [false]
set AIResponseHorizontalArrangement .Visible to [true]
call ChatBot1 .Converse
question [join [Dado los siguientes datos para el número anual] [Número de días que el lago de agua dulce estuvo ...] [call dataCleaningChartData2D .GetAllEntries] [El coeficiente de correlación para la línea de m...] [Trendline1 .CorrelationCoefficient] [La pendiente de la línea de mejor ajuste es...] [Trendline1 .LinearCoefficient] [La intersección en "Y" para la línea de mejor aj...] [Trendline1 .YIntercept] [Primero, predice el año que probablemente no se ...] [¿Cómo se relaciona esta tendencia con el cambio ...] [¿Qué pasa con las personas que viven cerca?] [Límita tus respuestas a 120 palabras]]
responseText
do set AIResponseTextBox .Text to [get responseText]

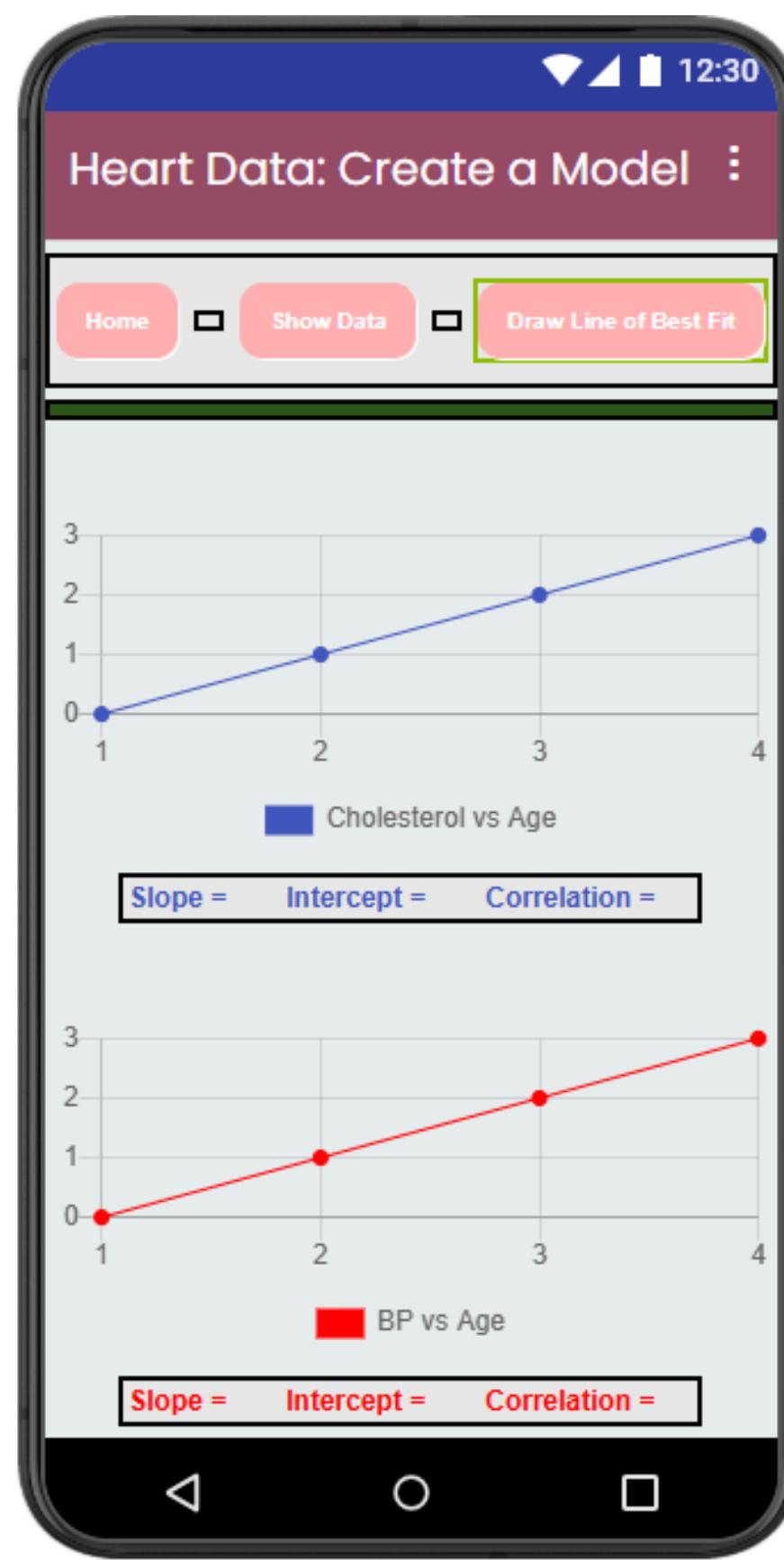
when ChatBot1 .GetResponse
responseText
do set AIResponseTextBox .Text to [get responseText]

```

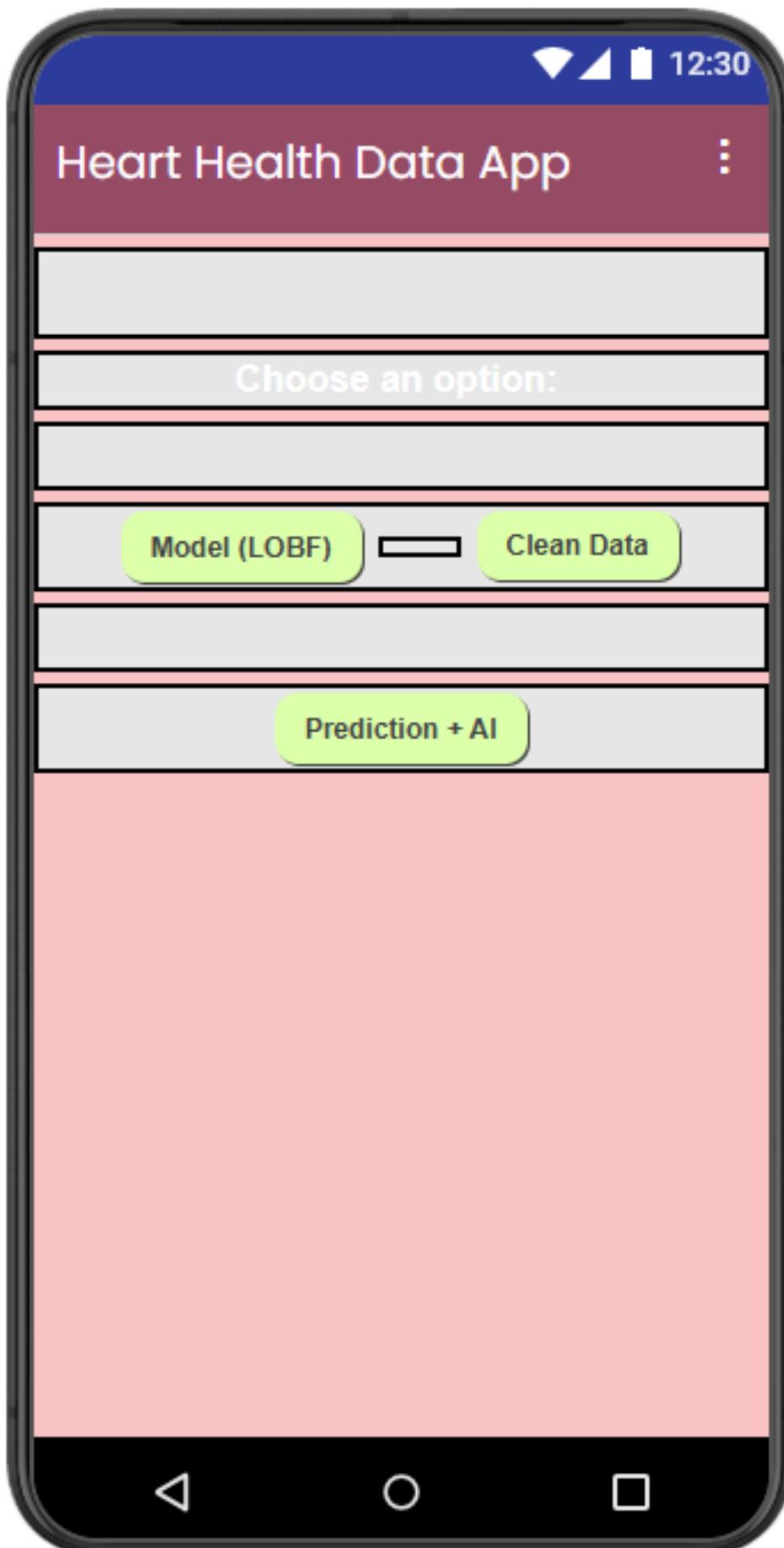
Proyecto Final



Non-visible components



Non-visible components



Non-visible components

