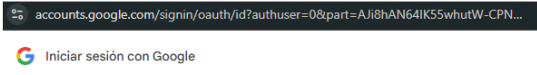




[www.back4app.com](http://www.back4app.com)

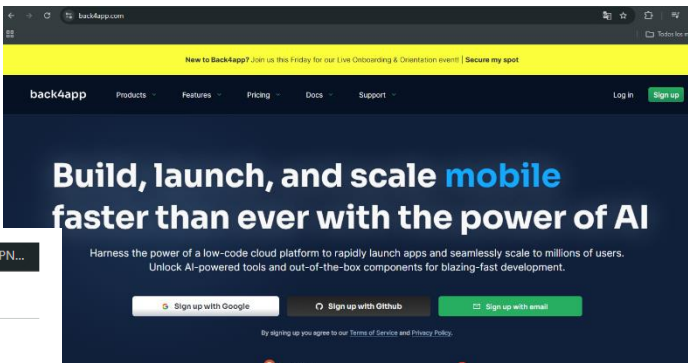


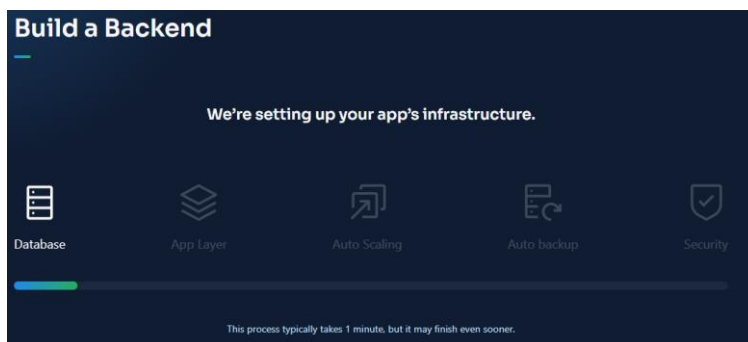
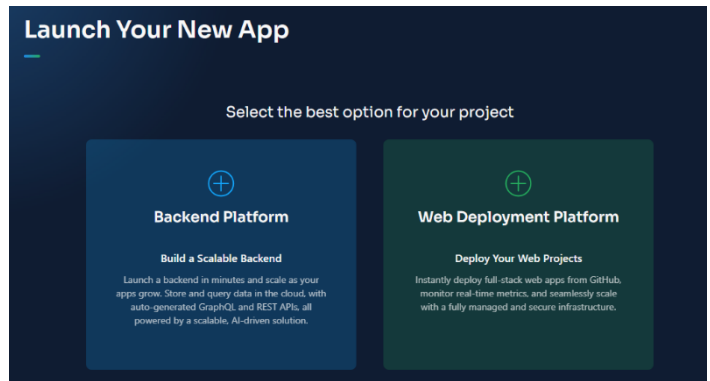
xxxxxxx@gmail.com ▼

Si continúas, Google compartirá tu nombre, tu dirección de correo electrónico, tu preferencia de idioma y tu imagen de perfil con [back4app.com](#). Consulta la [Política de Privacidad](#) y los [Términos del Servicio](#) de [back4app.com](#).

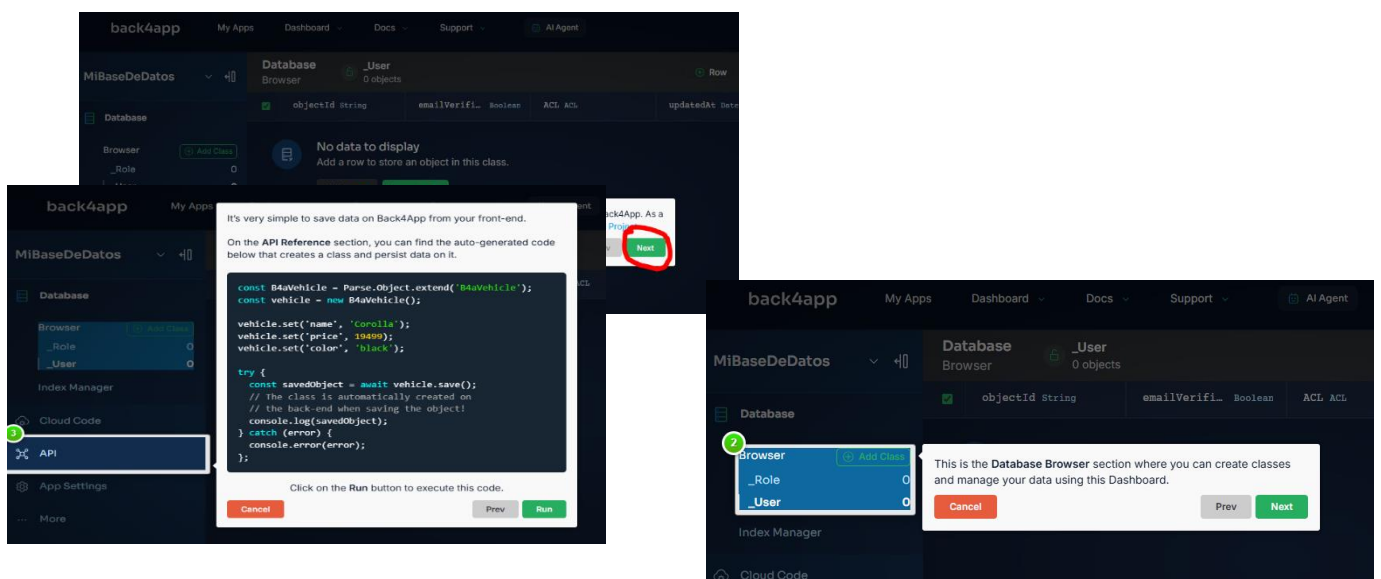
Puedes gestionar [Iniciar sesión con Google](#) en tu [cuenta de Google](#).

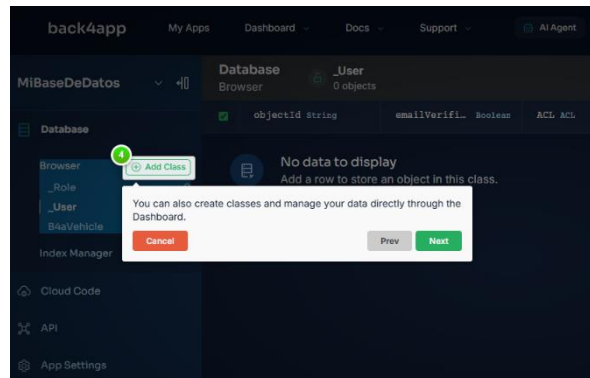
Cancelar

Continuar



ESPERAMOS QUE CARGUE, Y CUANDO TERMINE VAN APARECIENDO LOS CARTELITOS CON NÚMEROS DEL 1 AL 5, NOS VA MOSTRANDO DIFERENTES ACCIONES QUE PODEMOS HACER

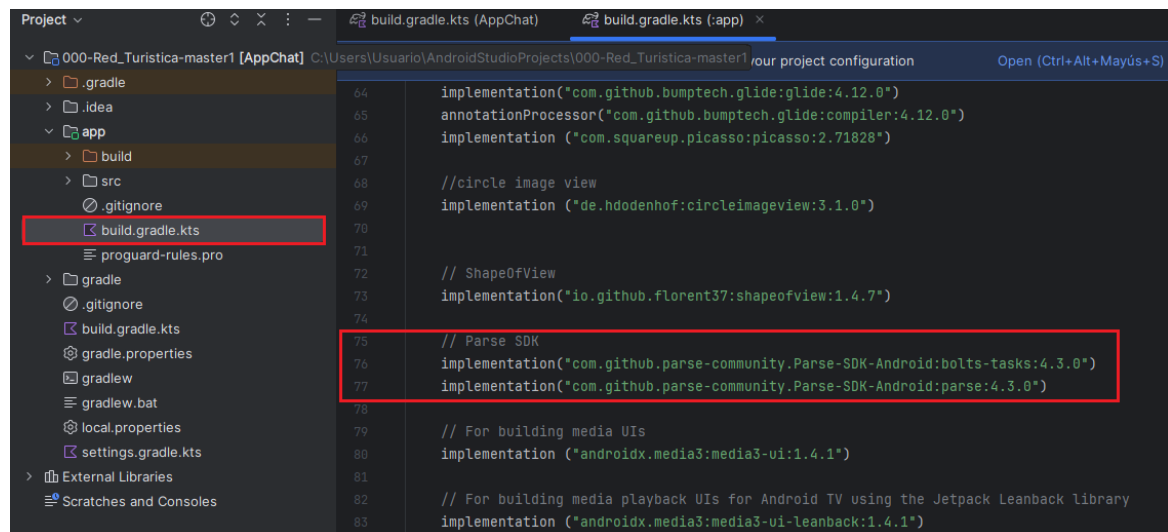




## 2) Configuramos en Android Studio

2.a) Agregamos en el archivo build.gradle las líneas 76 y 77 (de la imagen)

```
// Parse SDK
implementation("com.github.parse-community.Parse-SDK-Android:bolts-tasks:4.3.0")
implementation("com.github.parse-community.Parse-SDK-Android:parse:4.3.0")
```



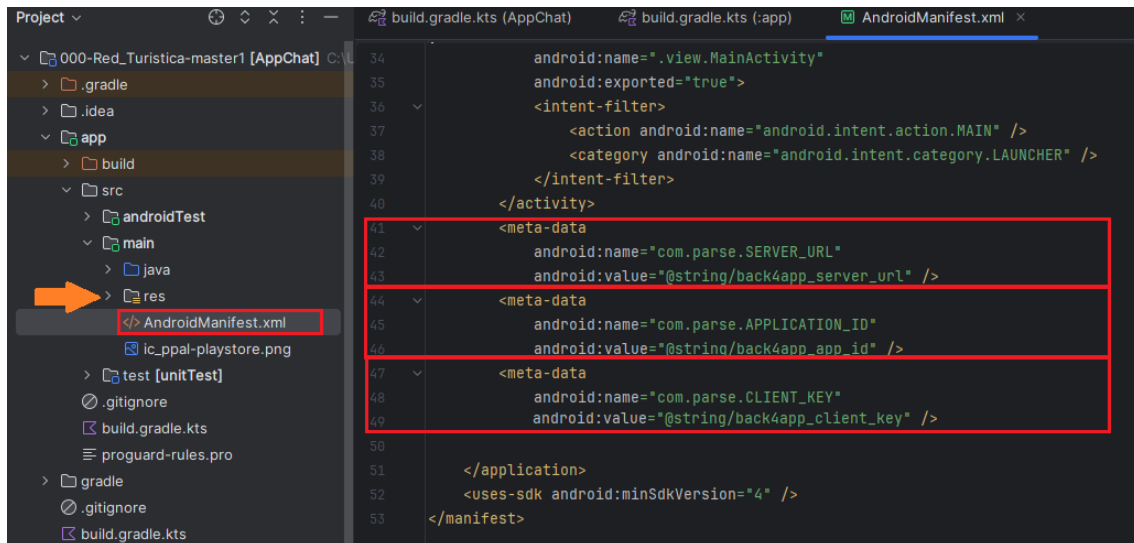
2.b) Agregamos en el archivo AndroidManifest.xml las líneas 41 a la 49 (de la imagen)

```
<meta-data
    android:name="com.parse.SERVER_URL"
    android:value="@string/back4app_server_url" />
<meta-data
    android:name="com.parse.APPLICATION_ID"
    android:value="@string/back4app_app_id" />
<meta-data
```

```

    android:name="com.parse.CLIENT_KEY"
    android:value="@string/back4app_client_key" />

```

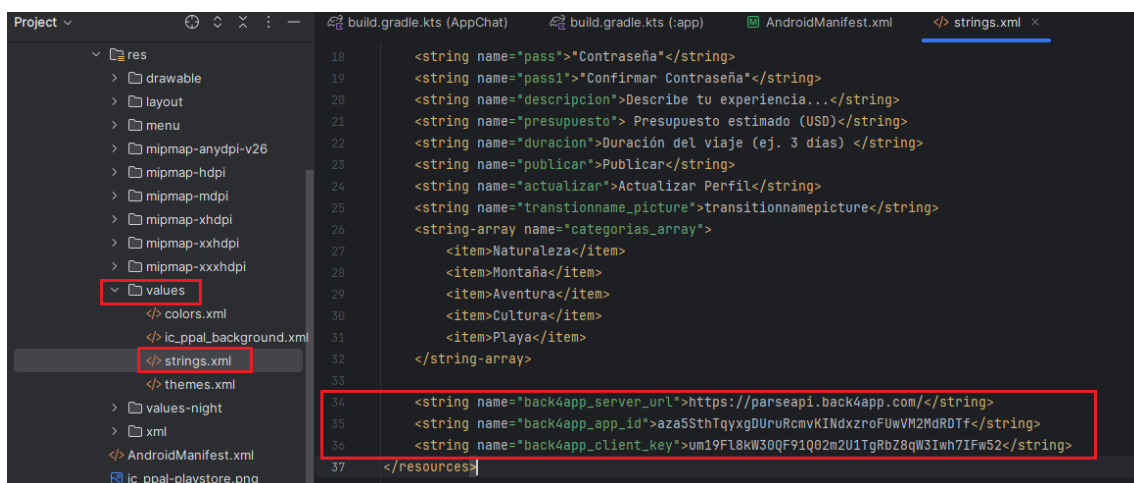


2.c) Agregamos en el archivo string.xml las líneas 34 a la 36 (de la imagen) en el punto 3 se detalla cómo obtener la url, el id y la key.

```

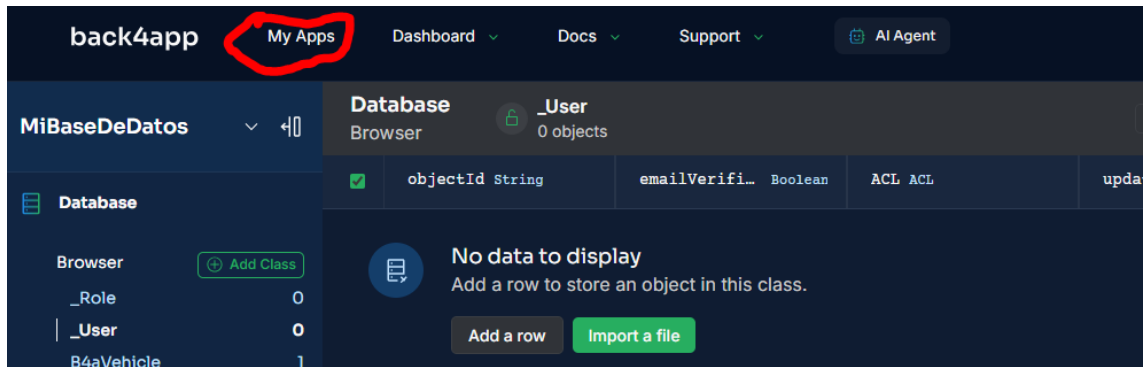
<string name="back4app_server_url">https://parseapi.back4app.com/</string>
<string name="back4app_app_id">aza5SthTqyxgDUruRcmvKINDxzroFUwVM2MdRDTf</string>
<string name="back4app_client_key">um19F18kW30QF91Q02m2U1TgRbZ8qW3Iwh7IFw52</string>

```

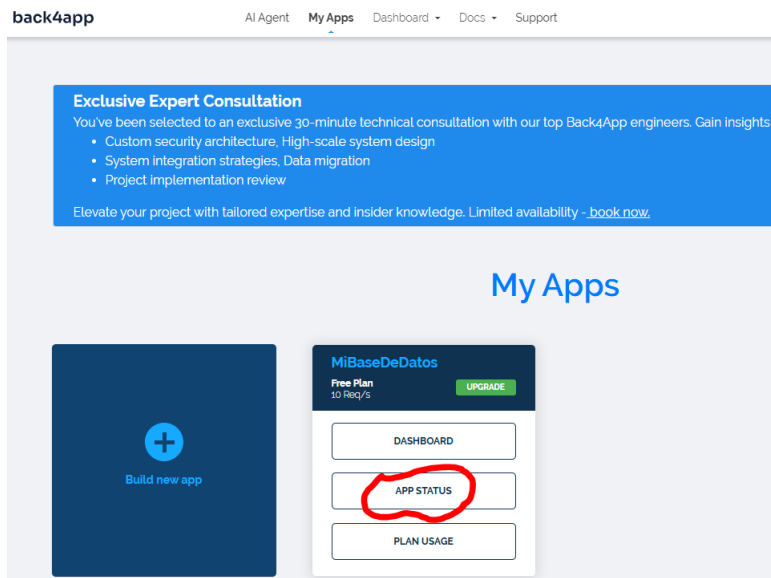


### 3) Buscamos la url, el id y la key

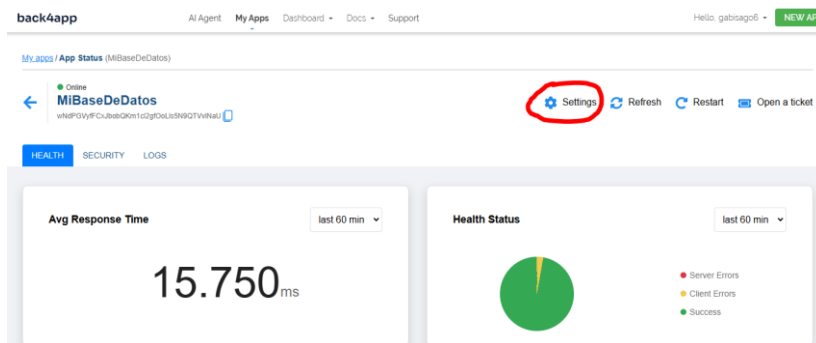
3.a) Volvemos a la página del dashboard de Parse, hacemos clic en “My Apps”



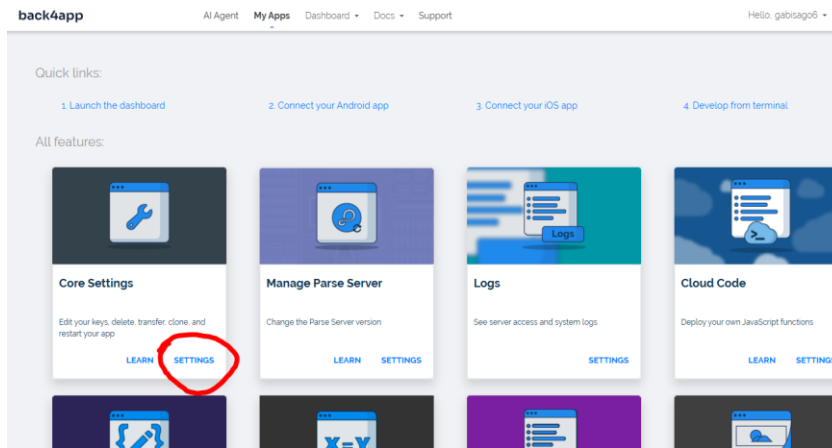
### 3.b) Seleccionamos “APPS STATUS”



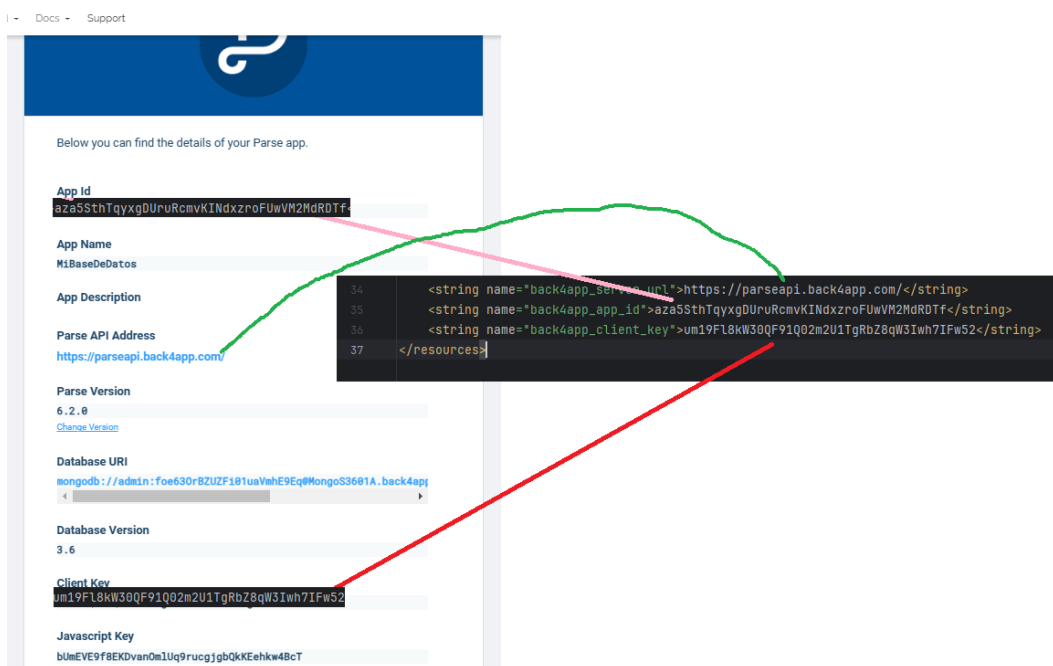
### 3.c) Seleccionamos “Settings”



### 3.d) Vamos a la tarjeta “Core Settings” y seleccionamos “Settings”

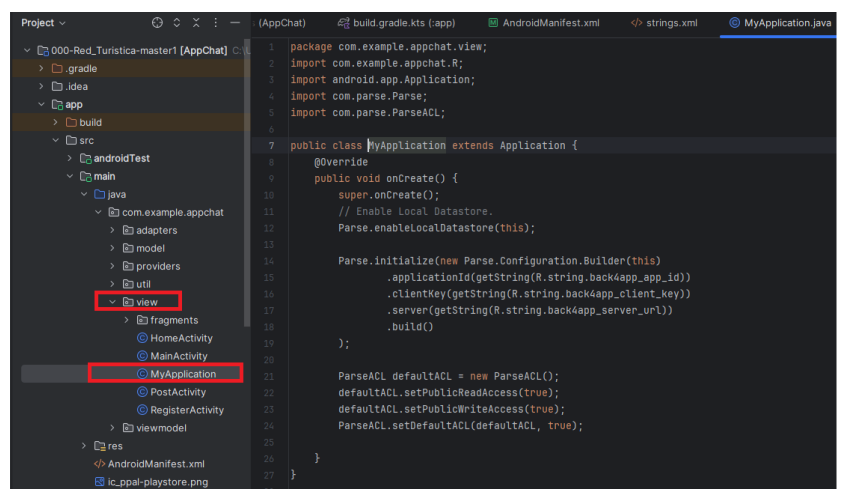


3.e) Obtenemos la información para completar los valores asignados a las variables de la imagen, que encontramos dentro de la carpeta strings, del punto 2c)

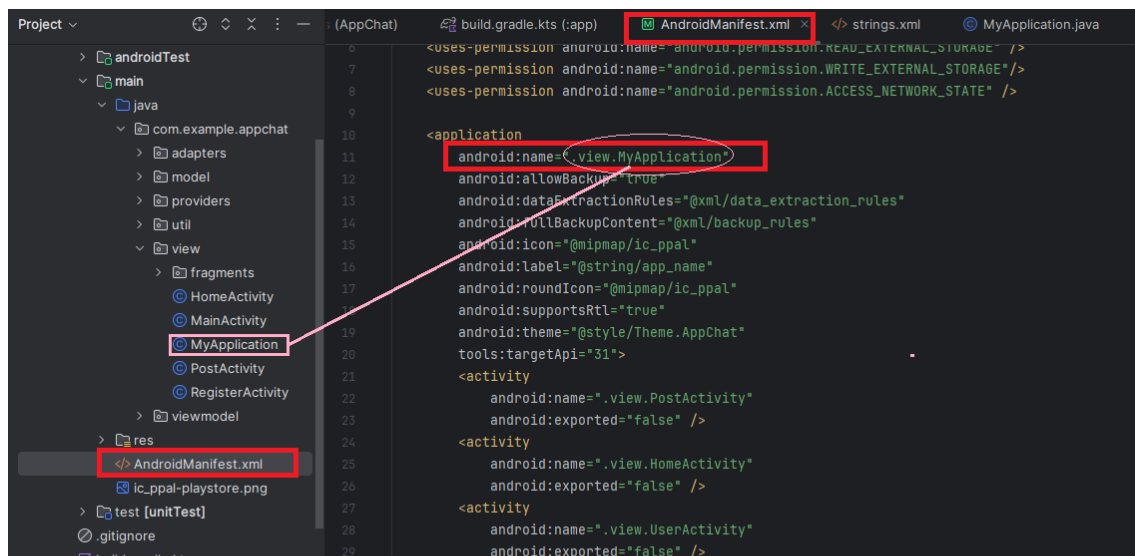


## 4) Agregamos la clase Java que

4.a) Creamos la clase "MyApplication.java", ver punto 5) para analizar el código.



4.b) Vamos al archivo AndroidManifest.xml y agregamos la línea 11 de la imagen, el código sería `android:name=".view.NombreDeLaClaseJava"`, en nuestro caso reemplazamos "NombreDeLaClaseJava" por MyApplication, que como muestra la imagen, es la clase que acabamos de crear.



## 5) Análisis del código de la clase "MyApplication"

5.a) Importamos...

```
import com.example.appchat.R;
import android.app.Application;
import com.parse.Parse;
import com.parse.ParseACL;
```

5.b) Extendemos de Application

```
public class MyApplication extends Application
```

5.c) Método onCreate de la clase padre, o sea de la clase "Application"

```
public void onCreate() {
    super.onCreate();
```

5.d) Parse.enableLocalDatastore

```
Parse.enableLocalDatastore(this);
```

- Esta línea habilita la función de **almacén de datos local** proporcionada por Parse SDK.
- El almacén de datos local le permite almacenar objetos Parse localmente en el dispositivo, lo que permite el acceso sin conexión a los datos y un mejor rendimiento en escenarios donde la conectividad de red es intermitente o no funciona.

5.e) **Conectamos nuestra aplicación al servidor de**

**análisis** los applicationId, clientKey, y serverId URL garantizan que la aplicación se comuniquen con el backend de Parse correcto (por ejemplo, Back4App) y habilita la funcionalidad del SDK de Parse.

```
Parse.initialize(new Parse.Configuration.Builder(this)
    .applicationId(getString(R.string.back4app_app_id))
    .clientId(getString(R.string.back4app_client_key))
    .server(getString(R.string.back4app_server_url))
    .build()
);
```

5.f) **SDK de Parse** permite conectarse a un servidor de Parse (por ejemplo, Back4App)

5.g) **Parse.initialize** inicializa el SDK de Parse en nuestra aplicación. Hay que llamarlo antes de utilizar cualquier funcionalidad relacionada con Parse, como guardar datos, consultar la base de datos o autenticar usuarios.

5.g) **new Parse.Configuration.Builder(this)** crea un nuevo objeto de configuración para el SDK de Parse, "this" se refiere al contexto actual.

5.h) **applicationId(getString(R.string.back4app\_app\_id))** establece el **ID de la aplicación** para el servidor Parse, esta identificación identifica de forma única nuestra aplicación y garantiza que las solicitudes se dirijan al backend de Parse correcto. Para facilitar su gestión al valor R.string.back4app\_app\_id ya lo almacenamos en el archivo res/values/strings.xml

5.i) **clientId(getString(R.string.back4app\_client\_key))** establece la **clave de cliente** o **clientId** para el servidor Parse, la clave del cliente autentica la aplicación con el servidor y garantiza que solo las aplicaciones autorizadas puedan acceder al backend. Para facilitar su gestión al valor R.string.back4app\_client\_key ya lo almacenamos en el archivo res/values/strings.xml

5.j) **server(getString(R.string.back4app\_server\_url))** especifica la URL del servidor Parse al que se conectará nuestra aplicación. Para Back4App, la URL del servidor normalmente es <https://parseapi.back4app.com/>, también ya lo almacenamos en el archivo

5.k) **build()** construye el objeto de configuración con los parámetros especificados, luego, este objeto se pasa para Parse.initialize() inicializar el SDK.

```
ParseACL defaultACL = new ParseACL();
defaultACL.setPublicReadAccess(true);
defaultACL.setPublicWriteAccess(true);
ParseACL.setDefaultACL(defaultACL, true);
```

5.l) **ParseACL defaultACL = new ParseACL();**

Crea una nueva instancia de ParseACL, una **lista de control de acceso (ACL)**, es una característica de seguridad que especifica permisos para usuarios individuales o para el público por objeto.

5.m) **defaultACL.setPublicReadAccess(true);** da **acceso de lectura** al público para todos los objetos que utilizan esta ACL predeterminada, por lo que **cualquier usuario** puede leer los datos sin necesidad de permisos explícitos.

5.n) **defaultACL.setPublicWriteAccess(true);** da **acceso de escritura** al público para todos los objetos que utilizan esta ACL predeterminada, por lo que **cualquier usuario** puede modificar, actualizar o eliminar los datos sin restricciones.

5.o) **ParseACL.setDefaultACL(defaultACL, true);** establece la ACL especificada ( defaultACL) como la **ACL predeterminada** para todos los objetos nuevos en nuestra



aplicación, el parámetro ( true) garantiza que esta ACL se aplique **incluso a los objetos creados antes de que un usuario haya iniciado sesión**

## 6) Código completo de la clase “MyApplication”

```
import com.example.appchat.R;
import android.app.Application;
import com.parse.Parse;
import com.parse.ParseACL;

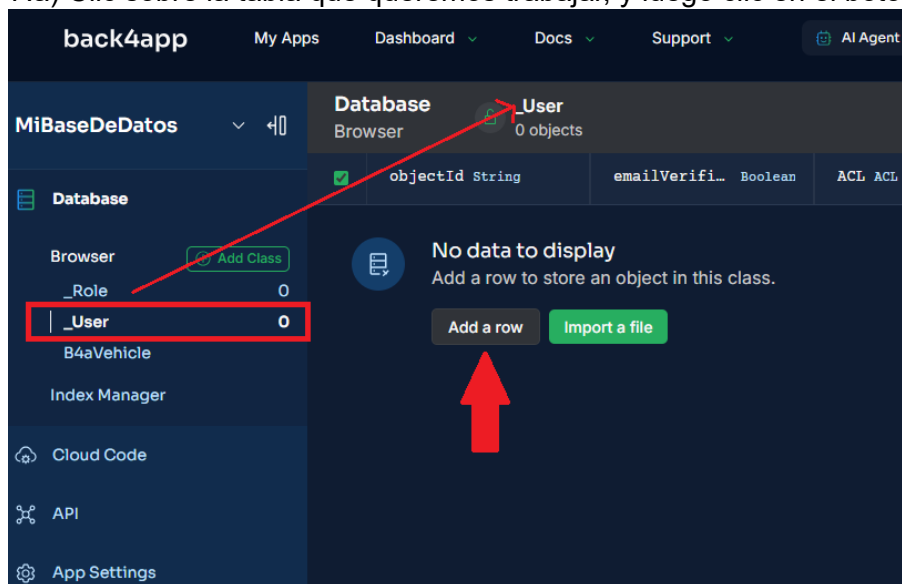
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        // Enable Local Datastore.
        Parse.enableLocalDatastore(this);

        Parse.initialize(new Parse.Configuration.Builder(this)
            .applicationId(getString(R.string.back4app_app_id))
            .clientId(getString(R.string.back4app_client_key))
            .server(getString(R.string.back4app_server_url))
            .build()
        );

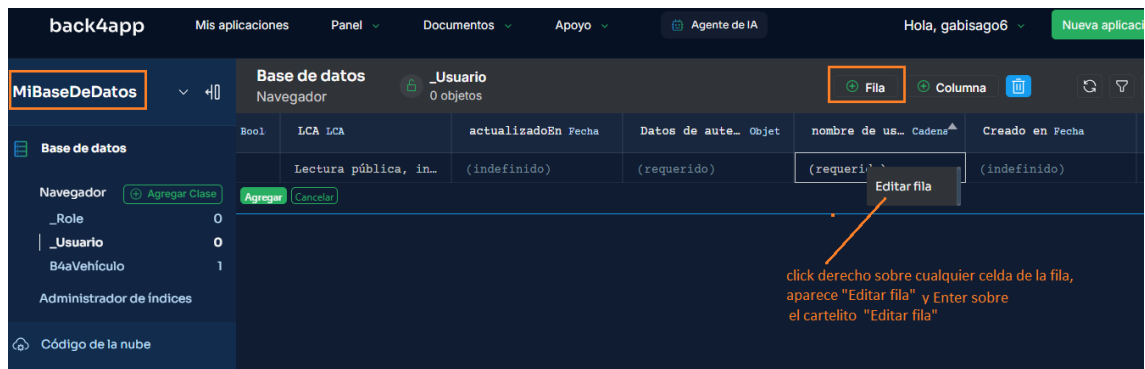
        ParseACL defaultACL = new ParseACL();
        defaultACL.setPublicReadAccess(true);
        defaultACL.setPublicWriteAccess(true);
        ParseACL.setDefaultACL(defaultACL, true);
    }
}
```

## 7) Completamos los campos de la tabla \_User

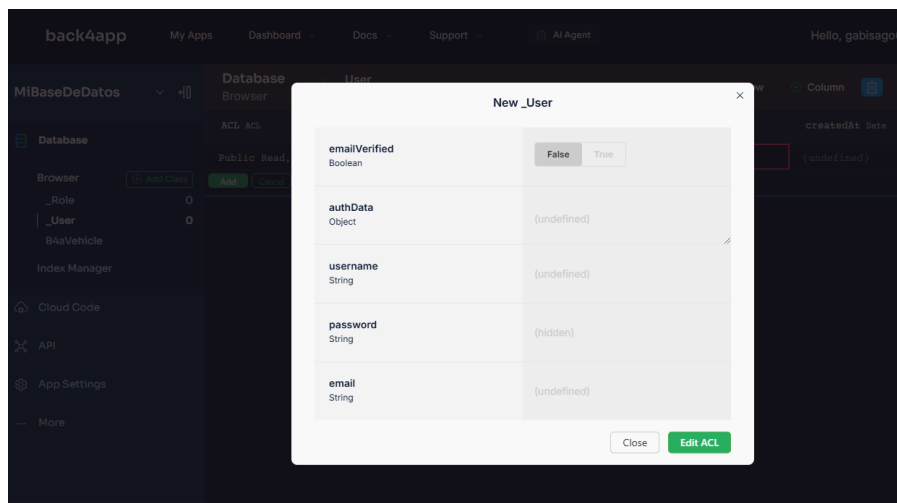
7.a) Clic sobre la tabla que queremos trabajar, y luego clic en el botón “Add row”



7.b) Se agrega una fila automáticamente

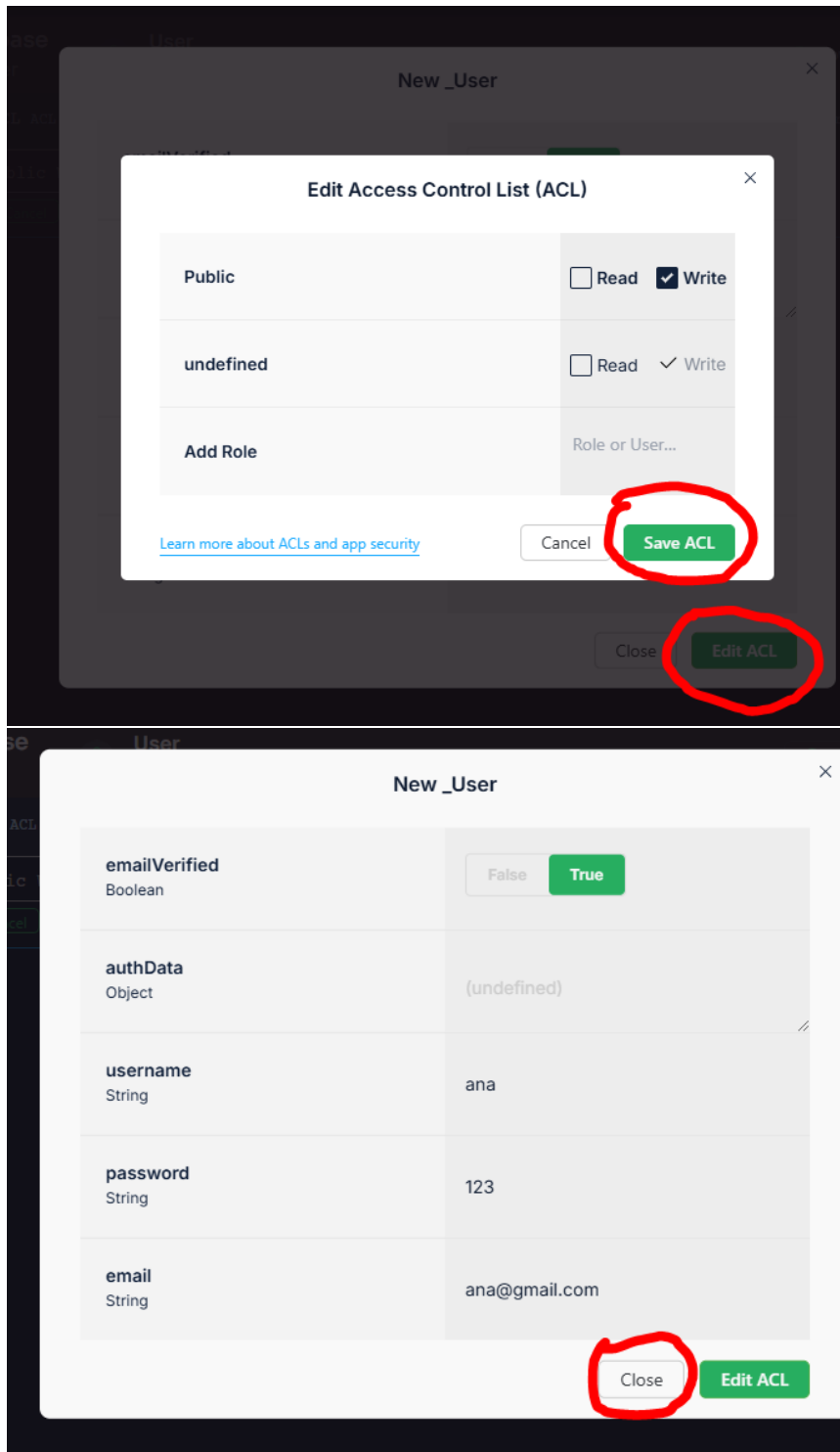


7.c) Aparece una ventana con los atos que debemos agregar



7.d) Completamos y hacemos clic en los botones que muestran las imágenes





7.e) Podemos ver los campos que completamos en la fila, el password, aparece oculto (hidden)

