Laravel new apicloud

Modificamos el archivo .ENV

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=dbapicloudx

DB_USERNAME=root

DB_PASSWORD=

Configurar el archivo AppServiceProvider.php (app/Provides/AppServiceProvider.php)

```php
<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;
use Illuminate\Support\Facades\Schema;

class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     *
     * @return void
     */
    public function register()
    {
        //
    }

    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        Schema::defaultStringLength(191);
    }
}
```

AHORA MIGRAMOS LA BASE DE DATOS ANTES PODEMOS MODIFICAR ALGUNAS COSAS

EN DATABASE MIGRATION USER

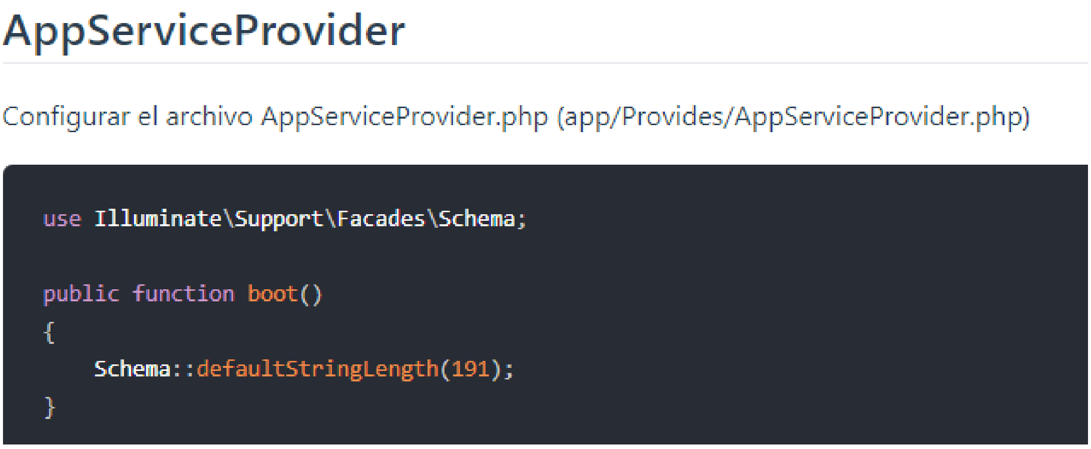
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Ahora vamos a importamos paquete y esta linea
`Schema::defaultStringLength(191);`
Como lo muestra la siguiente imagen



AppServiceProvider

Configurar el archivo AppServiceProvider.php (app/Provides/AppServiceProvider.php)

```
use Illuminate\Support\Facades\Schema;

public function boot()
{
    Schema::defaultStringLength(191);
}
```

AHORA SI php artisan migrate

Ahora

composer require tymon/jwt-auth:dev-develop --prefer-source

php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"

php artisan jwt:secret

modificamos composer.json

```
"laravel/tinker": "^2.0",
        "tymon/jwt-auth": "^1.0.0-beta.3@dev"
```

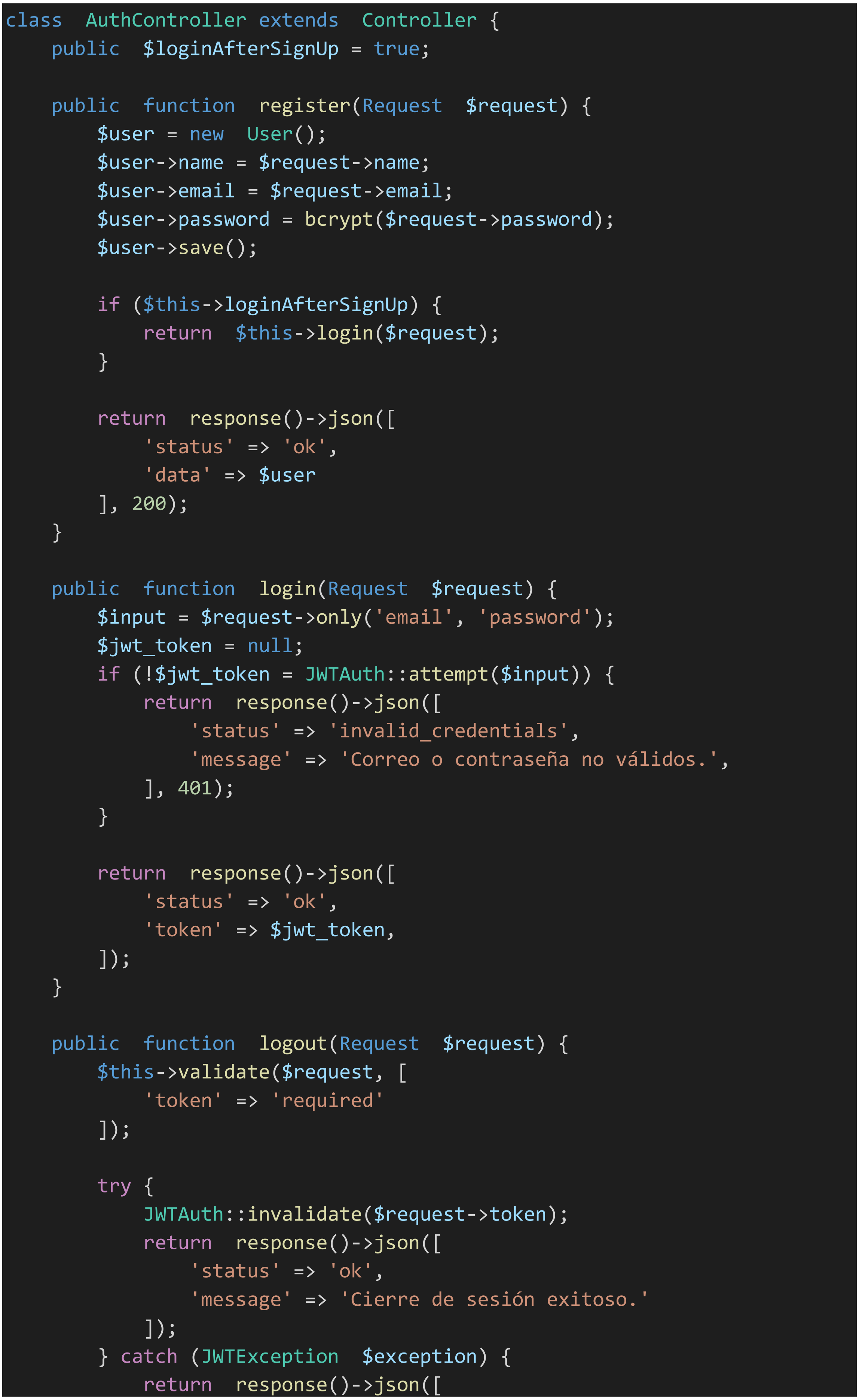
Ahora creamos un controlador

php artisan make:controller AuthController

modificamos este controlador

```
<?php
namespace  App\Http\Controllers;

use App\Http\Requests\RegisterAuthRequest;
use App\User;
use Illuminate\Http\Request;
use  JWTAuth;
use Tymon\JWTAuth\Exceptions\JWTException;
```



```php
class AuthController extends Controller {
    public $loginAfterSignUp = true;

    public function register(Request $request) {
        $user = new User();
        $user->name = $request->name;
        $user->email = $request->email;
        $user->password = bcrypt($request->password);
        $user->save();

        if ($this->loginAfterSignUp) {
            return $this->login($request);
        }

        return response()->json([
            'status' => 'ok',
            'data' => $user
        ], 200);
    }

    public function login(Request $request) {
        $input = $request->only('email', 'password');
        $jwt_token = null;
        if (!$jwt_token = JWTAuth::attempt($input)) {
            return response()->json([
                'status' => 'invalid_credentials',
                'message' => 'Correo o contraseña no válidos.',
            ], 401);
        }

        return response()->json([
            'status' => 'ok',
            'token' => $jwt_token,
        ]);
    }

    public function logout(Request $request) {
        $this->validate($request, [
            'token' => 'required'
        ]);

        try {
            JWTAuth::invalidate($request->token);
            return response()->json([
                'status' => 'ok',
                'message' => 'Cierre de sesión exitoso.'
            ]);
        } catch (JWTException $exception) {
            return response()->json([
                'status' => 'unknown_error',
                'message' => 'Al usuario no se le pudo cerrar la sesión.'
            ], 500);
        }
    }

    public  function  getAuthUser(Request  $request) {
        $this->validate($request, [
            'token' => 'required'
        ]);

        $user = JWTAuth::authenticate($request->token);
        return  response()->json(['user' => $user]);
    }

    protected function jsonResponse($data, $code = 200)
{
    return response()->json($data, $code,
        ['Content-Type' => 'application/json;charset=UTF-
8', 'Charset' => 'utf-8'], JSON_UNESCAPED_UNICODE);
}
}
```

Ahora modificamos el modelo USer

Modelo User

```
<?php
namespace  App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Foundation\Auth\User  as Authenticatable;

class  User  extends  Authenticatable  implements  JWTSubject {
    use  Notifiable;

    protected  $fillable = [
        'name', 'email', 'password',
    ];

    protected  $hidden = [
        'password', 'remember_token',
    ];
    protected  $casts = [
        'email_verified_at' => 'datetime',
    ];

    public  function  getJWTIdentifier() {
        return  $this->getKey();
    }

    public  function  getJWTCustomClaims() {
        return [];
    }
}
```

Modificamos el archivo routes api

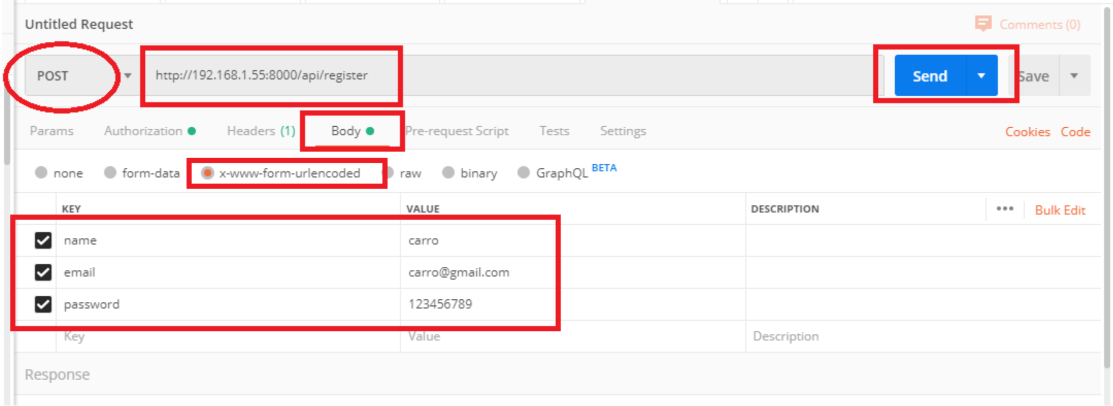
```
<?php
use Illuminate\Http\Request;

// estas rutas se pueden acceder sin proveer de un token válido.
Route::post('/login', 'AuthController@login');
Route::post('/register', 'AuthController@register');
// estas rutas requiren de un token válido para poder accederse.
Route::group(['middleware' => 'auth.jwt'], function () {
Route::post('/logout', 'AuthController@logout');
});
```

Y por ultimo procedemos a levantar el servidor

Ahora la ponemos a correr el servidor php artisan serv --host 192.168.1.55

Probamos con postman

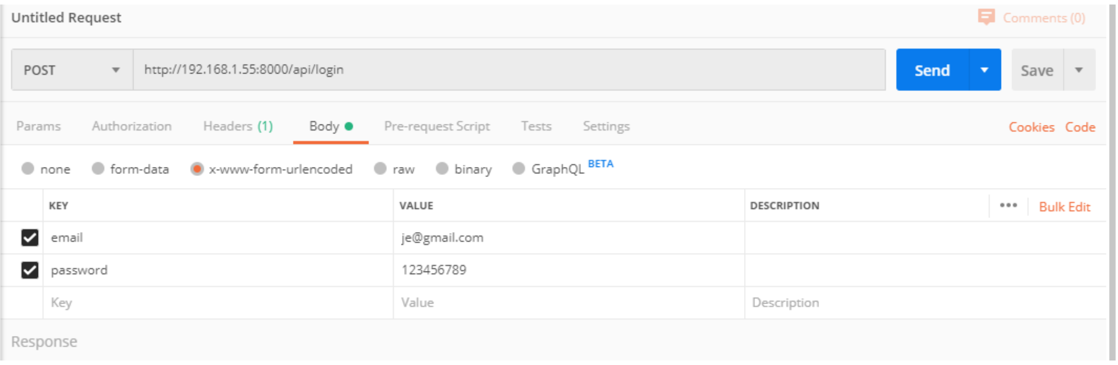


POST http://192.168.1.55:8000/api/register

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL BETA

	KEY	VALUE	DESCRIPTION
☑	name	carro	
☑	email	carro@gmail.com	
☑	password	123456789	
	Key	Value	Description

Ahora el login



Untitled Request

POST http://192.168.1.55:8000/api/login

Body: ○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL BETA

	KEY	VALUE	DESCRIPTION
☑	email	je@gmail.com	
☑	password	123456789	
	Key	Value	Description

Response

El devulve el token

Bien ahora vamos con VSCODE Dart Flutter

Create new project flutter appbatman

Modificamos el archivo pucspec.yalm

```yaml
dependencies:
  flutter:
    sdk: flutter
  http: any
  shared_preferences: any
```

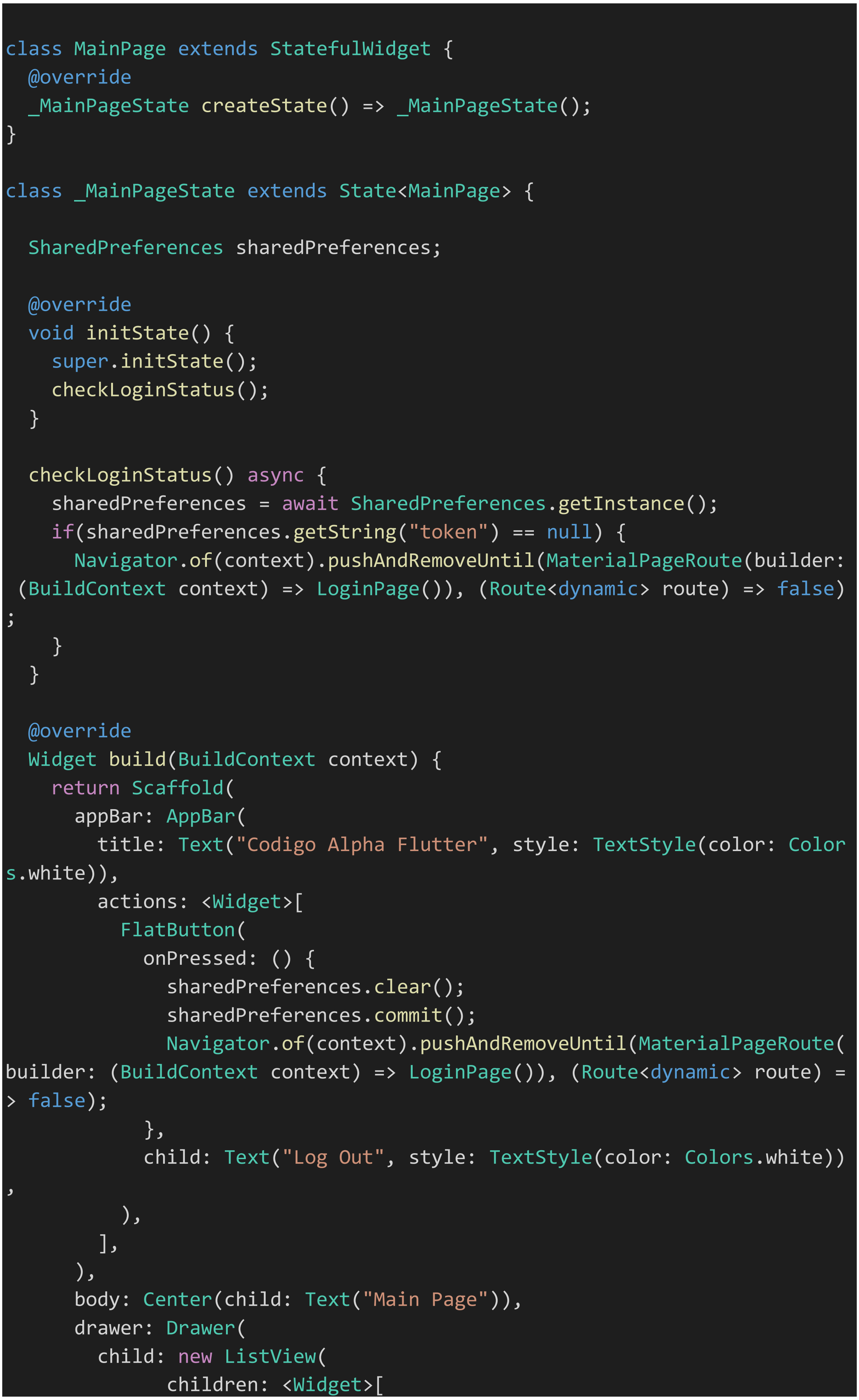
modificamos el main.dart

```dart
import 'dart:convert';

import 'package:appphp/login.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Codigo Alpha Flutter",
      debugShowCheckedModeBanner: false,
      home: MainPage(),
      theme: ThemeData(
        accentColor: Colors.white70
      ),
    );
  }
}
```



```
class MainPage extends StatefulWidget {
  @override
  _MainPageState createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {

  SharedPreferences sharedPreferences;

  @override
  void initState() {
    super.initState();
    checkLoginStatus();
  }

  checkLoginStatus() async {
    sharedPreferences = await SharedPreferences.getInstance();
    if(sharedPreferences.getString("token") == null) {
      Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (BuildContext context) => LoginPage()), (Route<dynamic> route) => false);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Codigo Alpha Flutter", style: TextStyle(color: Colors.white)),
        actions: <Widget>[
          FlatButton(
            onPressed: () {
              sharedPreferences.clear();
              sharedPreferences.commit();
              Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (BuildContext context) => LoginPage()), (Route<dynamic> route) => false);
            },
            child: Text("Log Out", style: TextStyle(color: Colors.white)),
          ),
        ],
      ),
      body: Center(child: Text("Main Page")),
      drawer: Drawer(
        child: new ListView(
              children: <Widget>[
                new UserAccountsDrawerHeader(
                  accountName: new Text('Ejercicios'),
                  accountEmail: new Text('codigoalphacol@gmail.com'),
                  // decoration: new BoxDecoration(
                  //   image: new DecorationImage(
                  //     fit: BoxFit.fill,
                  //    // image: AssetImage('img/estiramiento.jpg'),
                  //   )
                  // ),
                ),
                new Divider(),
                // new ListTile(
                //   title: new Text("Add data"),
                //   trailing: new Icon(Icons.fitness_center),
                //   onTap: () => Navigator.of(context).push(new MaterialPageRoute(
                //     builder: (BuildContext context) => AddData(),
                //   )),
                // ),
                // new Divider(),
                // new ListTile(
                //   title: new Text("Mostrar listado"),
                //   trailing: new Icon(Icons.help),
                //   onTap: () => Navigator.of(context).push(new MaterialPageRoute(
                //     builder: (BuildContext context) => ShowData(),
                //   )),
                // ),
              ],
            ),
      ),
    );
  }
}
```

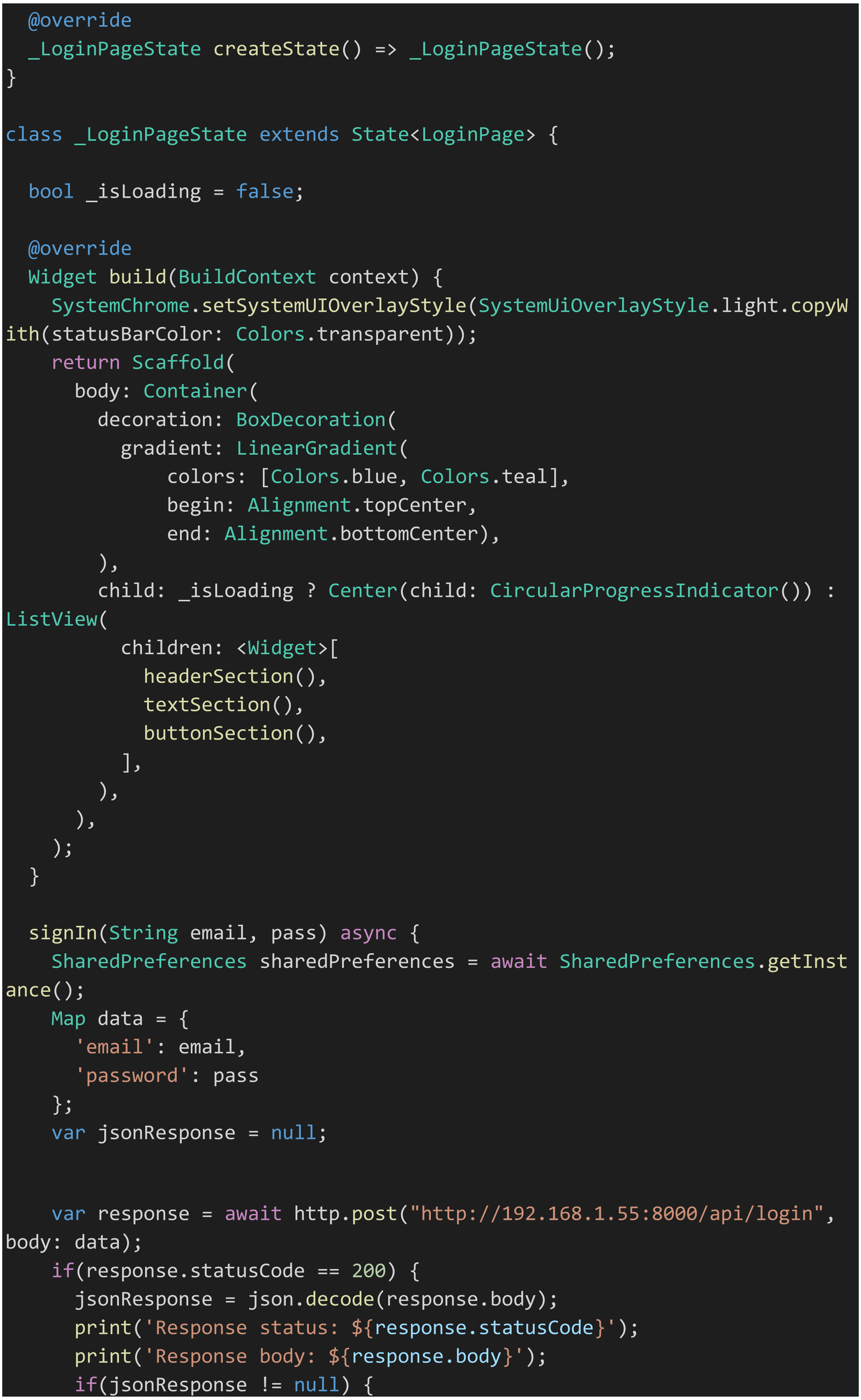
Creamos el login.dart

```
import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';

import 'main.dart';

class LoginPage extends StatefulWidget {
```



```
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {

  bool _isLoading = false;

  @override
  Widget build(BuildContext context) {
    SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle.light.copyWith(statusBarColor: Colors.transparent));
    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
              colors: [Colors.blue, Colors.teal],
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter),
        ),
        child: _isLoading ? Center(child: CircularProgressIndicator()) : ListView(
          children: <Widget>[
            headerSection(),
            textSection(),
            buttonSection(),
          ],
        ),
      ),
    );
  }

  signIn(String email, pass) async {
    SharedPreferences sharedPreferences = await SharedPreferences.getInstance();
    Map data = {
      'email': email,
      'password': pass
    };
    var jsonResponse = null;


    var response = await http.post("http://192.168.1.55:8000/api/login", body: data);
    if(response.statusCode == 200) {
      jsonResponse = json.decode(response.body);
      print('Response status: ${response.statusCode}');
      print('Response body: ${response.body}');
      if(jsonResponse != null) {
```

```
    setState(() {
     _isLoading = false;
    });
    sharedPreferences.setString("token", jsonResponse['token']);
    Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (BuildContext context) => MainPage()), (Route<dynamic> route) => false);
   }
  }
  else {
   setState(() {
    _isLoading = false;
   });
   print(response.body);
  }
 }

 Container buttonSection() {
  return Container(
   width: MediaQuery.of(context).size.width,
   height: 40.0,
   padding: EdgeInsets.symmetric(horizontal: 15.0),
   margin: EdgeInsets.only(top: 15.0),
   child: RaisedButton(
    onPressed: emailController.text == "" || passwordController.text == "" ? null : () {
     setState(() {
      _isLoading = true;
     });
     signIn(emailController.text, passwordController.text);
    },
    elevation: 0.0,
    color: Colors.purple,
    child: Text("Sign In", style: TextStyle(color: Colors.white70)),
    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0)),
   ),
  );
 }

 final TextEditingController emailController = new TextEditingController();
 final TextEditingController passwordController = new TextEditingController();

 Container textSection() {
  return Container(
   padding: EdgeInsets.symmetric(horizontal: 15.0, vertical: 20.0),
   child: Column(
        children: <Widget>[
          TextFormField(
            controller: emailController,
            cursorColor: Colors.white,

            style: TextStyle(color: Colors.white70),
            decoration: InputDecoration(
              icon: Icon(Icons.email, color: Colors.white70),
              hintText: "Email",
              border: UnderlineInputBorder(borderSide: BorderSide(color:
Colors.white70)),
              hintStyle: TextStyle(color: Colors.white70),
            ),
          ),
          SizedBox(height: 30.0),
          TextFormField(
            controller: passwordController,
            cursorColor: Colors.white,
            obscureText: true,
            style: TextStyle(color: Colors.white70),
            decoration: InputDecoration(
              icon: Icon(Icons.lock, color: Colors.white70),
              hintText: "Password",
              border: UnderlineInputBorder(borderSide: BorderSide(color:
Colors.white70)),
              hintStyle: TextStyle(color: Colors.white70),
            ),
          ),
        ],
      ),
    );
  }

  Container headerSection() {
    return Container(
      margin: EdgeInsets.only(top: 50.0),
      padding: EdgeInsets.symmetric(horizontal: 20.0, vertical: 30.0),
      child: Text("Codigo Alpha Flutter",
          style: TextStyle(
              color: Colors.white70,
              fontSize: 40.0,
              fontWeight: FontWeight.bold)),
    );
  }
}
```