

DCC917A – TÓPICOS ESPECIAIS III: DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

AULA 12

Carlos Bruno Oliveira Lopes

Engenheiro de Computação

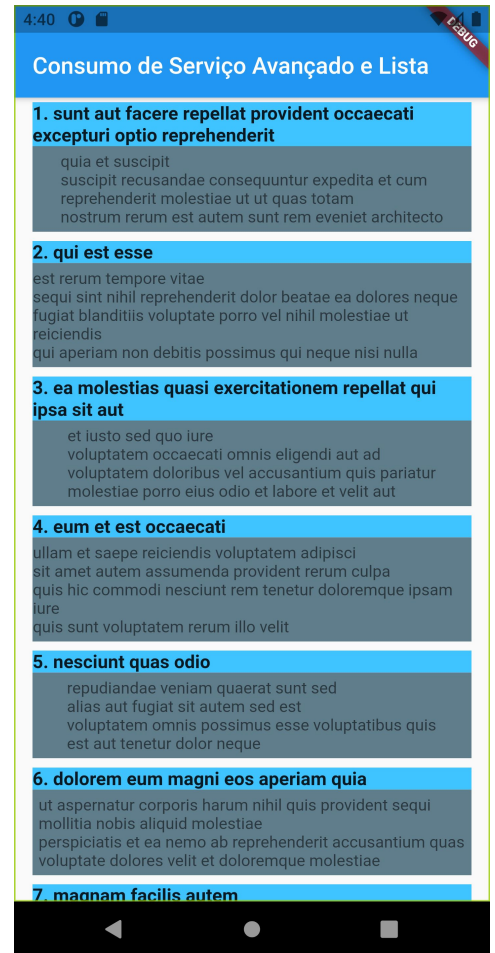
Mestre em Ciência da Computação

Flutter (ListView e requisições)

```
String _urlBase = "https://jsonplaceholder.typicode.com";

Future<List<Post>> _recuperarPostagens() async {
  var url = Uri.parse(_urlBase + "/posts");
  http.Response response = await http.get(url);
  var dadosJason = json.decode(response.body);
  List<Post> postagens = [];

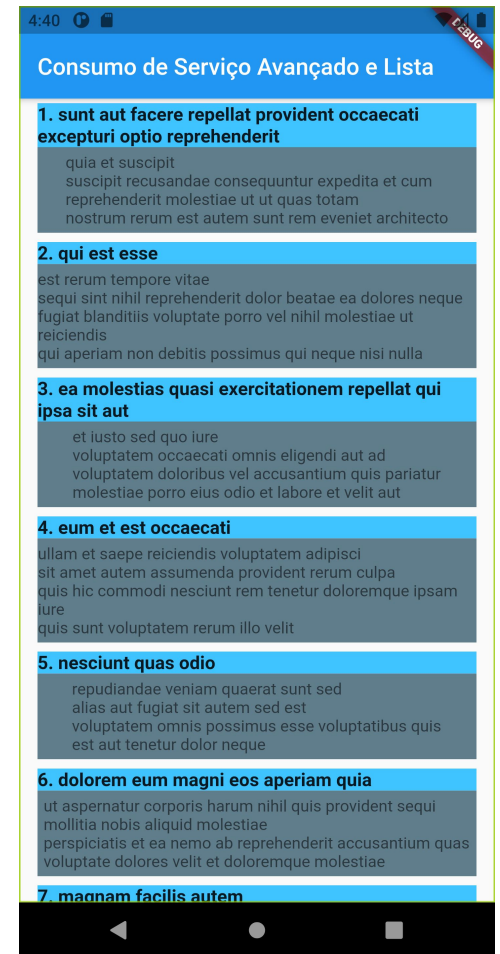
  for (var post in dadosJason) {
    // print("post" + post['title']);
    Post p = Post(post['userId'], post['id'], post['title'],
      post['body']);
    postagens.add(p);
  }
  print(postagens.toString());
  return postagens;
}
```



Flutter (ListView e requisições)

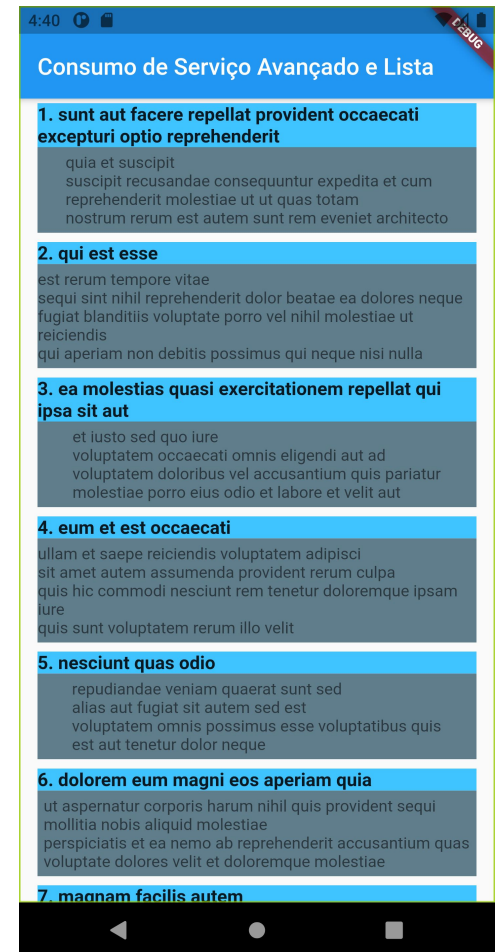
```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Consumo de Serviço Avançado e Lista"),
    ),
    body: FutureBuilder<List<Post>>(
      future: _recuperarPostagens(),
      builder: (context, snapshot) {
        ListView? listView;

        switch (snapshot.connectionState) {
          case ConnectionState.waiting:
            print("conexão waiting");
            return Center(
              child: CircularProgressIndicator(),
            );
```



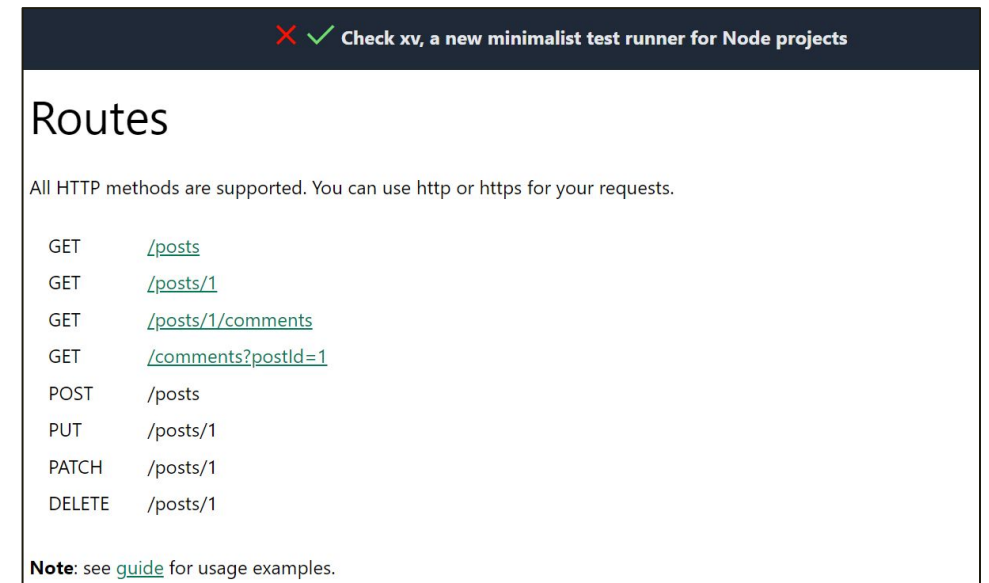
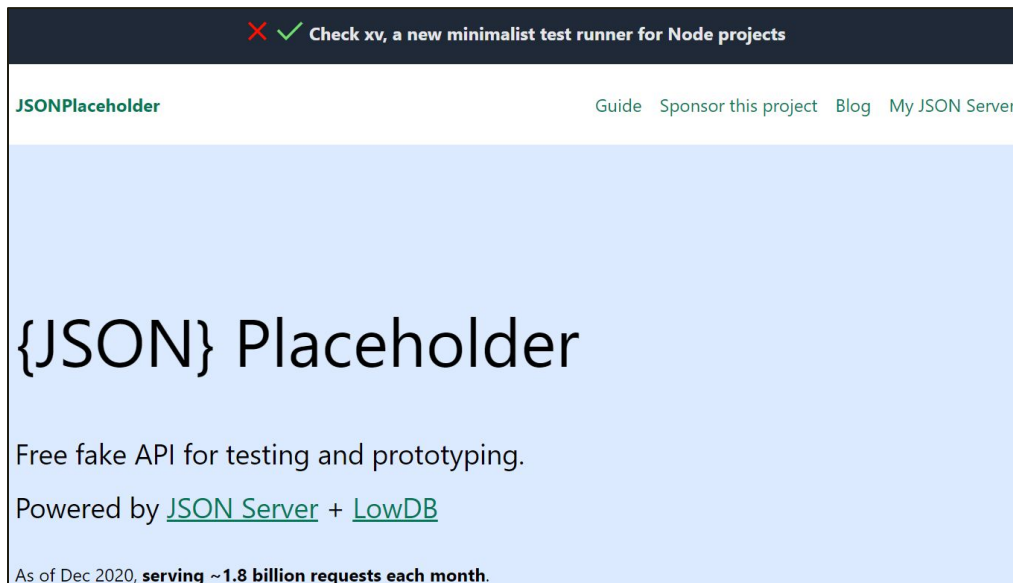
Flutter (ListView e requisições)

```
•      :  
      :  
      }  
      return listView!;  
    },  
  ),  
);  
}  
}
```



Flutter (Post e Widget Expanded)

- Vamos acessar o site JSONPlaceholder pelo link:
<https://jsonplaceholder.typicode.com/>
 - Esse site fornece recursos de API para teste ou simulações de app web



Flutter (Post e Widget Expanded)

Requisições

- **GET**: Realiza a requisição de um recurso.
- **POST**: cria um novo recurso na API.
 - Ex.: criar um novo produto. Fazer um upload de um vídeo.
- **PUT**: Atualiza recursos.
 - Ex.: Consegue atualizar todos os dados como de um produto de vendas, seu nome, preço e estoque.
- **PACTH**: Atualiza recursos específicos.
 - Ex.: Atualização de um preço de um produto.
- **DELETE**: Remove um recurso.

Routes

All HTTP methods are supported. You can use http or https for your requests.

GET	/posts
GET	/posts/1
GET	/posts/1/comments
GET	/comments?postId=1
POST	/posts
PUT	/posts/1
PATCH	/posts/1
DELETE	/posts/1

Note: see [guide](#) for usage examples.

Flutter (Post e Widget Expanded)

Código de status

- Link: https://pt.wikipedia.org/wiki/Lista_de_c%C3%B3digos_de_estado_HTTP

1xx Informativa

Indica que a requisição foi recebida e entendida. Essa resposta é despachada provisoriamente, enquanto o processamento da requisição continua. Serve para alertar ao cliente, que espere pela resposta final. A mensagem constitui-se apenas do Status-Line e cabeçalhos opcionais, e é encerrado por uma linha vazia, dado que a versão HTTP/1.0, não definiu nenhum código de status 1xx, os servidores não devem enviar uma resposta 1xx para um cliente HTTP/1.0, exceto sob condições experimentais.

100 Continuar

Isso significa que o servidor recebeu os cabeçalhos da solicitação, e que o cliente deve proceder para enviar o corpo do pedido (no caso de haver um pedido, um corpo deve ser enviado, por exemplo, um POST pedido). Se o corpo do pedido é grande, enviando-o para um servidor, se o pedido já foi rejeitado, com base em cabeçalhos inadequados é ineficiente. Para ter um cheque do servidor se o pedido pode ser aceito com base no pedido de cabeçalhos sozinho, o cliente deve enviar: Esperar 100-continue, como um cabeçalho no seu pedido inicial e verificar se a 100 Continuar, código de status é recebido em resposta antes de permanente (ou receber 417 Falha na expectativa e não continuar).

101 Mudando protocolos

Isso significa que o solicitante pediu ao servidor para mudar os protocolos, e o servidor está reconhecendo que irá fazê-lo.

2xx Sucesso

Esta classe de códigos de status indica a ação solicitada pelo cliente foi recebida, compreendida, aceita e processada com êxito.

```
{
  "Id": 1,
  "Title": "Lista de códigos de estado HTTP"
  "Corpo": "A seguir está uma lista de códigos de resposta em HTTP (HyperText Transfer Protocol). Isso inclui os códigos padrões de internet da IETF, outras especificações e alguns códigos adicionais usados."
}
```

201 Criado

O pedido foi cumprido e resultou em um novo recurso que está sendo criado.

202 Aceito

O pedido foi aceito para processamento, mas o tratamento não foi concluído. O pedido poderá ou não vir a ser posta em prática, pois pode ser anulado quando o processamento ocorre realmente.

203 não-autorizado (desde HTTP/1.1)

O servidor processou a solicitação com sucesso, mas está retornando informações que podem ser de outra fonte.

Flutter (Requisições e Widget Expanded)

```
Future<List<Post>> _recuperarPostagens() async {...}
●
// Usado para criar um novo recurso
void _post() async {...}

void _post1() async {...}

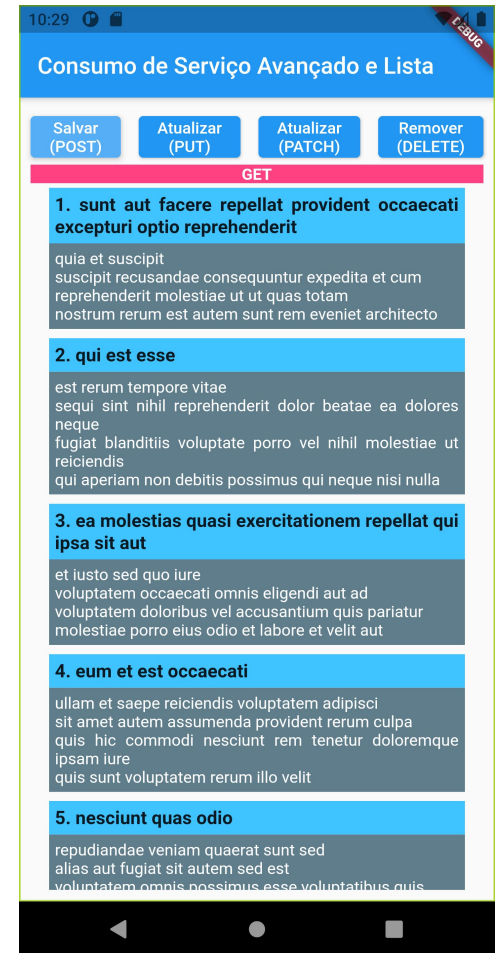
// Usado para atualizar o recurso inteiro
void _put() async {...}

// Usado para atualizar o recurso parcialmete
void _patch() async {...}

// Usado para remover um recurso
void _delete() async {
  var url = Uri.parse(_urlBase + "/posts/2");
  http.Response response = await http.delete(url);

  print("resposta: ${response.statusCode}");
  print("resposta: ${response.body}");
}

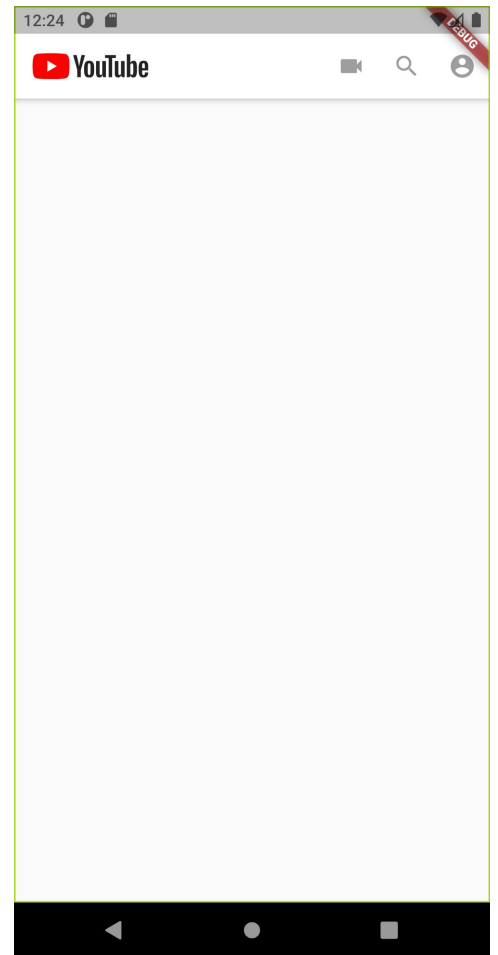
:
Expanded( child: FutureBuilder<List<Post>>(...),),
```



Flutter (AppBar – botões de ação)

App - Youtube

```
appBar: AppBar(  
  • iconTheme: IconThemeData(  
    color: Colors.grey,  
    opacity: 1  
  ),  
  backgroundColor: Colors.white,  
  title: Image.asset( 'imagens/youtube.png', width:100),  
  actions: [  
    IconButton( icon: Icon(Icons.videocam),  
      onPressed: () {print("Icons.videocam");},  
    ),  
    :  
    IconButton( icon: Icon(Icons.account_circle),  
      onPressed: () {print("Icons.account_circle");},  
    ),  
  ],  
)
```



Flutter (bottomNavigationBar – Navegação)

App - Youtube

```
bottomNavigationBar: BottomNavigationBar(  
  currentIndex: _indexBar,  
  onTap: (value) { setState(() { _indexBar = value;}); },  
  type: BottomNavigationBarType.fixed,  
  fixedColor: Colors.red,  
  items: [  
    BottomNavigationBarItem(  
      title: Text("Início"),  
      icon: Icon(Icons.home)),  
    BottomNavigationBarItem(  
      title: Text("Em alta"),  
      icon: Icon(Icons.whatshot)),  
    BottomNavigationBarItem(  
      title: Text("Inscrições"),  
      icon: Icon(Icons.subscriptions)),  
    BottomNavigationBarItem(  
      title: Text("Biblioteca"),  
      icon: Icon(Icons.folder)),  
  ],  
) ,
```



Flutter (API Youtube)

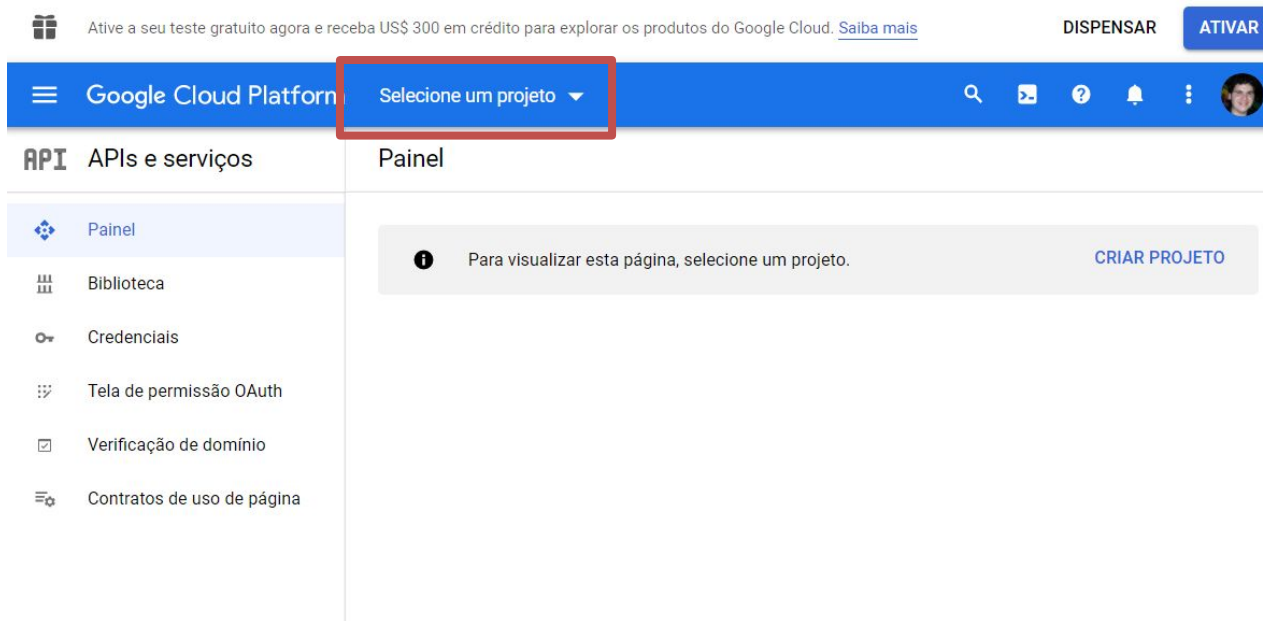
App - Youtube

- Link de descrição da API do Youtube:
<https://developers.google.com/youtube/v3/>
- Para usar as API's que pertence ao google é necessário ativar o seu acesso por meio de uma conta do google.
 - Para ativar acesse o link:
<https://console.cloud.google.com/apis/dashboard>
 - Entre com a sua conta google

Flutter (API Youtube)

App - Youtube

- Em seguida, clique em selecionar um projeto para criar um projeto de desenvolvimento para que possamos ativar um API do google:
 - Clique em novo projeto e dê um nome para ele.

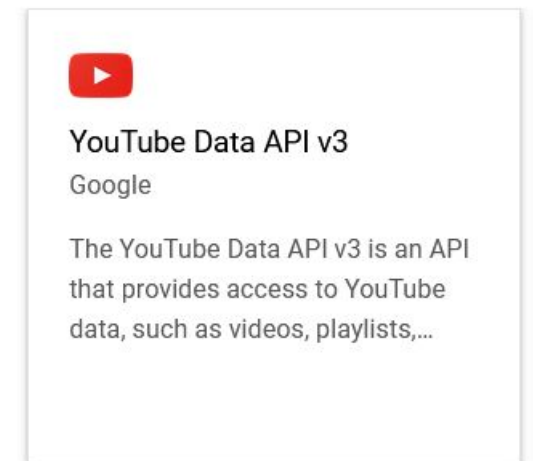
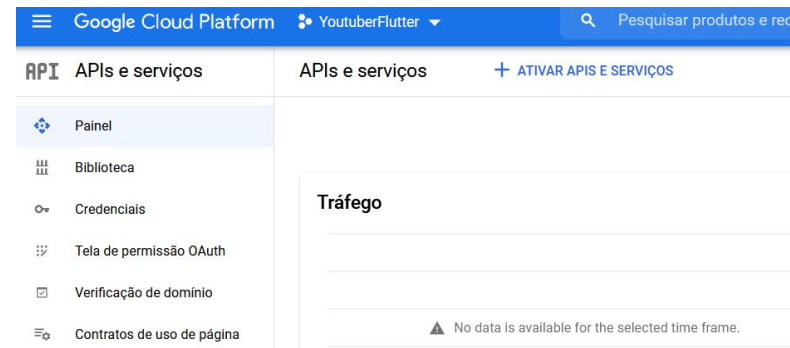
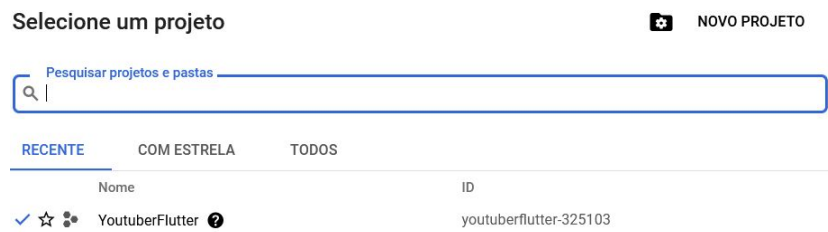


CANCELAR ABRIR

Flutter (API Youtube)

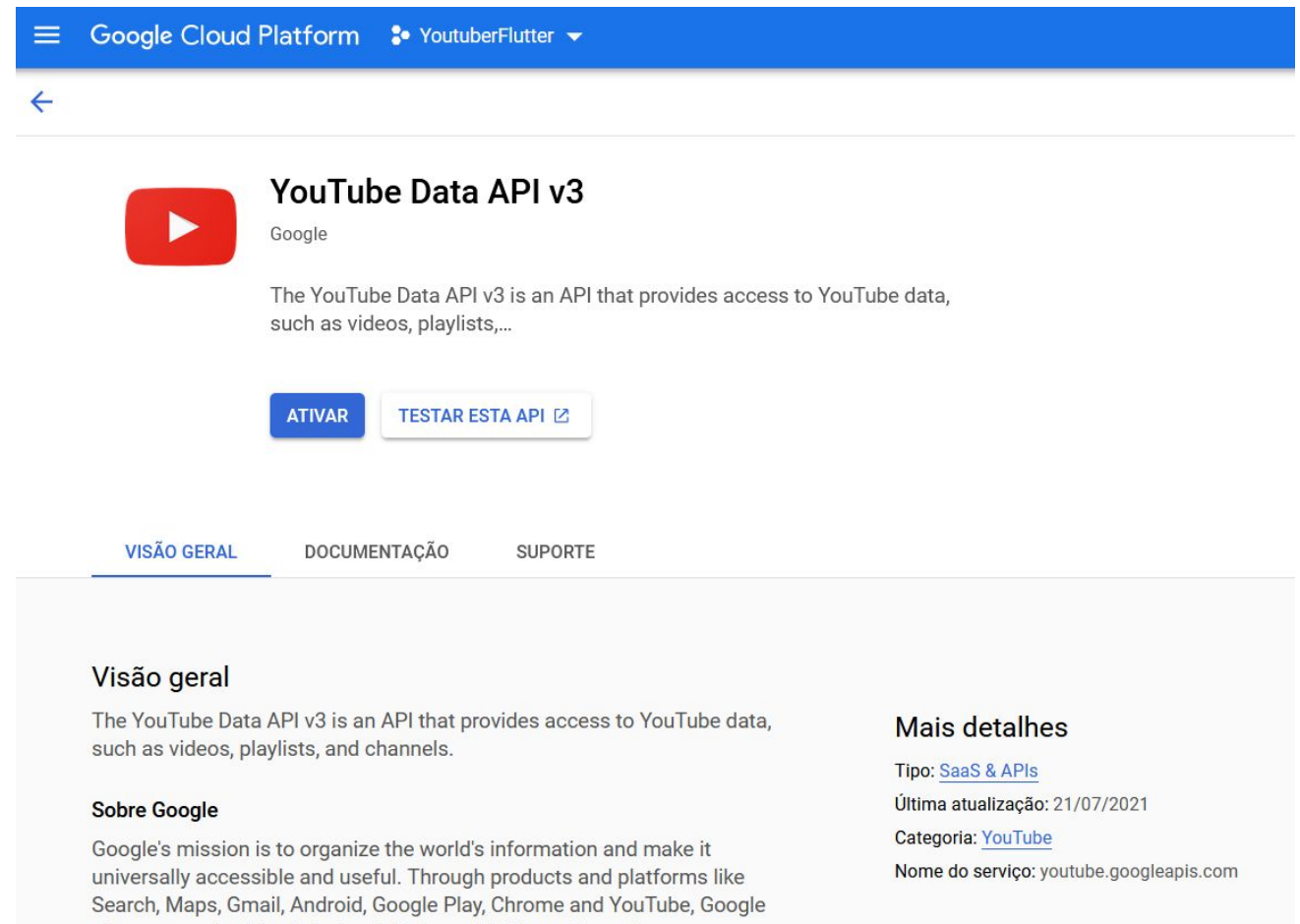
App - Youtube

- Selecione o projeto criado.
- Em seguida vá APIs e serviços, e selecione Biblioteca.
- Então, selecione a API YouTube Data API v3



Flutter (API Youtube) App - Youtube

- Para finalizar, ative a API:



The screenshot shows the Google Cloud Platform console interface. At the top, there's a blue header with the Google Cloud Platform logo and a dropdown menu showing 'YoutuberFlutter'. Below the header, there's a back arrow icon. The main content area features the YouTube Data API v3 card. It includes a red YouTube play button icon, the title 'YouTube Data API v3', and the provider 'Google'. A description states: 'The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...'. Below the description are two buttons: 'ATIVAR' (Activate) and 'TESTAR ESTA API' (Test this API). At the bottom, there's a navigation bar with three tabs: 'VISÃO GERAL' (Overview), 'DOCUMENTAÇÃO' (Documentation), and 'SUPORTE' (Support). The 'VISÃO GERAL' tab is selected, showing the 'Visão geral' (Overview) section. This section contains a description of the API and a 'Sobre Google' (About Google) section. To the right, there's a 'Mais detalhes' (More details) section with links for 'Tipo: SaaS & APIs', 'Última atualização: 21/07/2021', 'Categoria: YouTube', and 'Nome do serviço: youtube.googleapis.com'.

Google Cloud Platform YoutuberFlutter

YouTube Data API v3

Google

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...

ATIVAR TESTAR ESTA API

VISÃO GERAL DOCUMENTAÇÃO SUPORTE

Visão geral

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists, and channels.

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

Mais detalhes

Tipo: [SaaS & APIs](#)

Última atualização: 21/07/2021

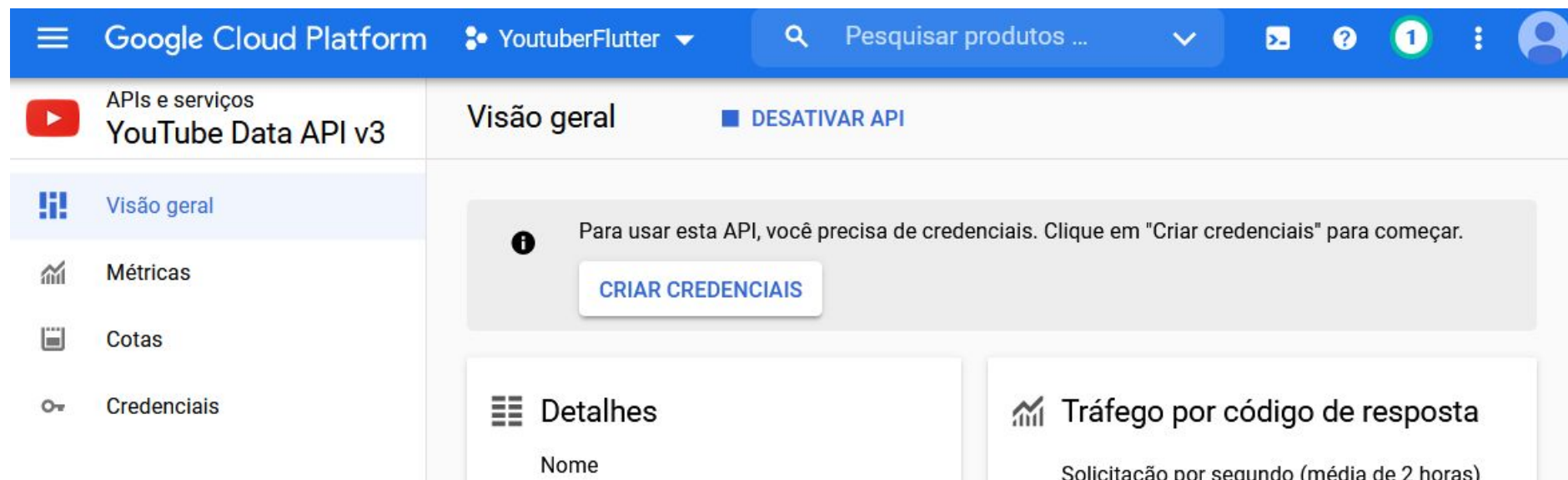
Categoria: [YouTube](#)

Nome do serviço: youtube.googleapis.com

Flutter (API Youtube)

App - Youtube

- Agora, é necessário criar as credencias de acesso da API para que o google reconheça que você está autorizado a usa-las:



Flutter (API Youtube)

App - Youtube

- Em Criar Credenciais, selecione a API **YouTube Data API v3**. Em seguida, escolha **Dados Públicos** em “Que dados você acessará? ”, clique em próxima:

Google Cloud Platform YoutuberFlutter Pesquisar produtos e recursos

API APIs e serviços

Painel

Biblioteca

Credenciais

Tela de permissão OAuth

Verificação de domínio

Contratos de uso de página

1 Tipo de credencial

Qual API você usa?

APIs diferentes usam plataformas de autenticação distintas. Algumas credenciais podem ser restritas para fazer chamadas a somente determinadas APIs.

Selecionar uma API *
YouTube Data API v3

Que dados você acessará? *

É necessário usar credenciais diferentes para autorizar o acesso dependendo do tipo de dados solicitado. [Saiba mais](#)

☐ Dados do usuário ?
Dados de um usuário do Google, como o endereço de e-mail ou a idade. É obrigatório ter o consentimento do usuário. Isso criará um cliente OAuth.

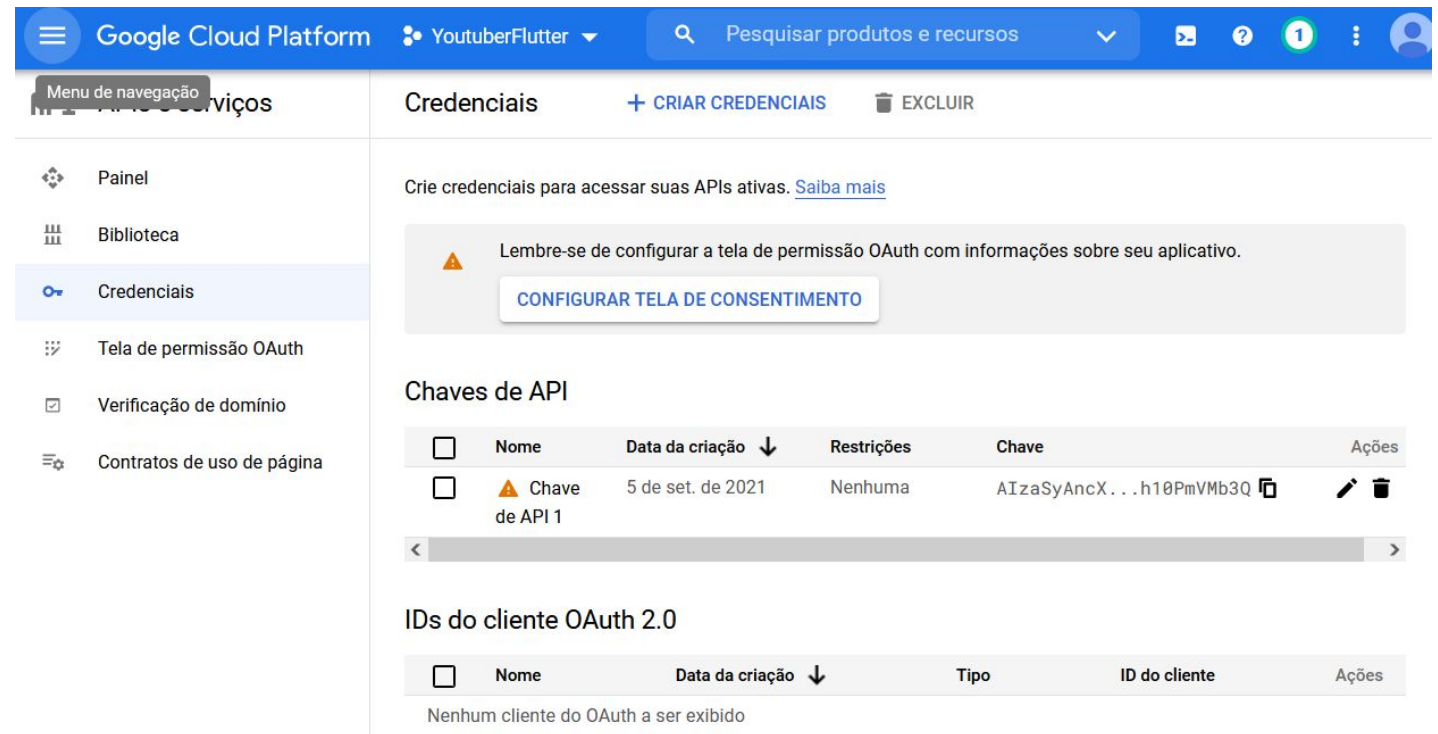
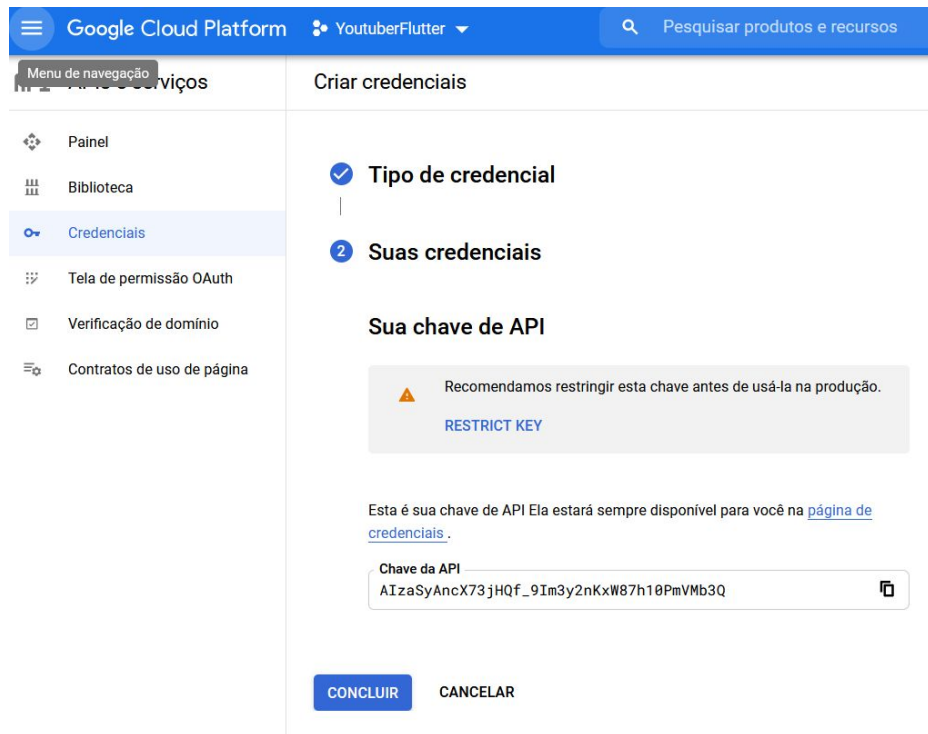
☒ Dados públicos
Dados do Google disponibilizados publicamente, como dados públicos do Maps que exibem informações sobre restaurantes. Isso criará uma chave de API.

PRÓXIMA

Flutter (API Youtube)

App - Youtube

- Em seguida suas credenciais serão criadas, basta copiar as credenciais, salva-las e em seguida clicar em concluir:



Flutter (API Youtube)

App - Youtube

- Instalação e configuração das dependência para uso do Youtube em App:
 - HTTP
 - Acesse o site: <https://pub.dev/>
 - Digite http no buscador e confirme a busca. Selecione Flutter e clique em http;
 - Em seguida, selecione “**Installing**”, procure “**dependencies:**” e copie:
`http: ^0.13.3`
 - Finalize a configuração do http acessando, o arquivo “pubspec.yaml” no seu projeto e adicionando a dependência.

Flutter (API Youtube)

App - Youtube

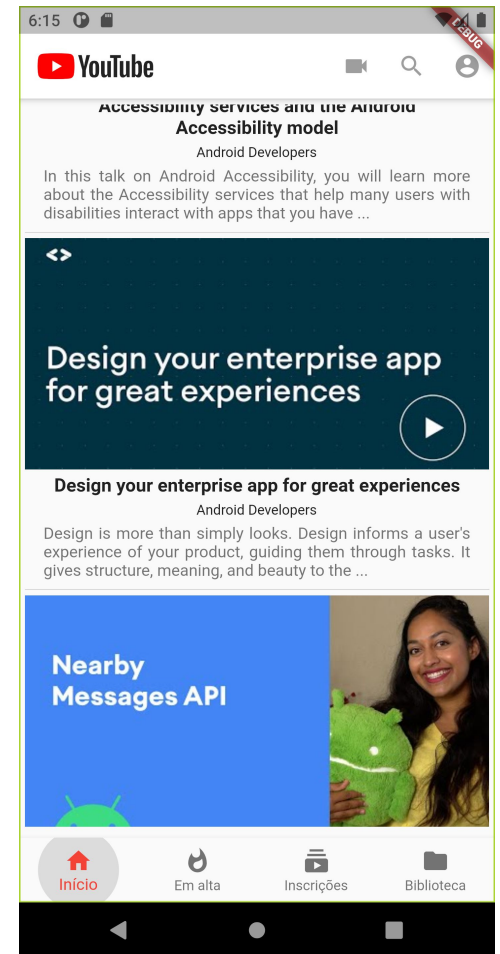
- Instalação e configuração das dependência para uso do Youtube em App:
 - YOUTUBE
 - Acesse o site: <https://pub.dev/>
 - Digite http no buscador e confirme a busca. Selecione Flutter e clique em youtube_player_flutter;
 - Em seguida, selecione “**Installing**”, procure “**dependencies:**” e copie:

```
youtube_player_flutter: ^8.0.0
```
 - Finalize a configuração do http acessando, o arquivo “pubspec.yaml” no seu projeto e adicionando a dependência.

Flutter (API Youtube)

App - Youtube

```
Widget builderListView(Video video){
  return Column(
    children: [
      Container(
        height: 200,
        decoration: BoxDecoration( image: DecorationImage(fit: BoxFit.cover,
          image: NetworkImage(video.imagem!), ),),),
      ListTile( title: Column(children: [
        Text(video.titulo!, textAlign: TextAlign.center,
          style: TextStyle( fontWeight: FontWeight.bold,)),
        Padding( padding: EdgeInsets.only(top: 5, bottom: 5),
          child: Text( video.canal!, textAlign: TextAlign.start,
            style: TextStyle( fontSize: 12,
              fontWeight: FontWeight.bold,
              color: Color.fromRGBO(90, 90, 90, 1),),),),),
        subtitle: Text( video.descricao!, textAlign: TextAlign.justify,),
      ],),
    ],
  );
}
```



Flutter (API Youtube)

App - Youtube

State

- Representam o tempo de vida da widget, ou seja, seus estados que variam desde sua construção até a sua “destruição”;
- **initState**: É executado uma única vez para inicializar as dependências ou configurações para construção da widget.
- **didChangeDependencies**: Chamado há mudanças nas dependências no estado do objeto. (Em geral mudanças relações de herança das Widgets).
- **build**: constrói a widget.
- **didUpdateWidget**: Chamado após ocorrer uma mudança na árvore de widget geralmente associado a abertura ou fechamento de uma segunda tela.
- **deactivate**: Chamado quando um objeto é removido temporariamente da árvore. Permite que o objeto possa ser reinserido em outra parte da árvore.
- **dispose**: Chamado quando um objeto é removido da árvore.