



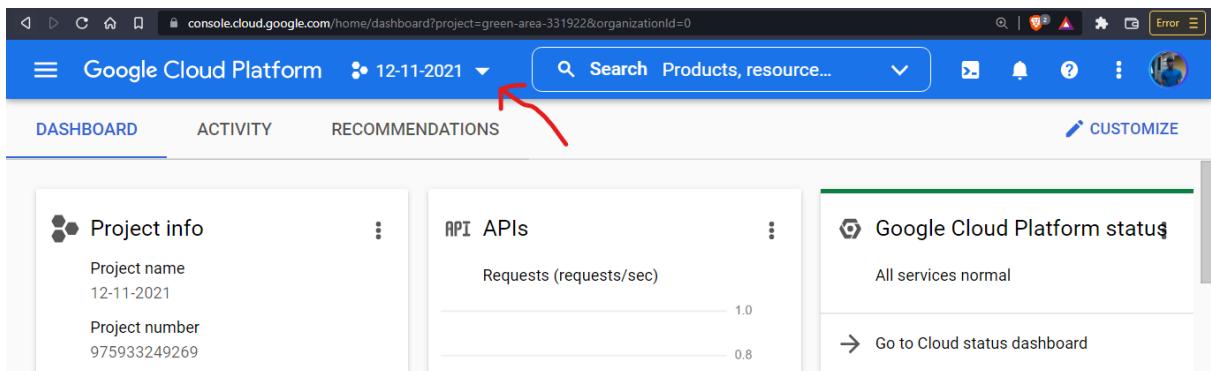
```
1 const { createServer } = require('http')
2 const { parse } = require('url')
3 const next = require('next')
4
5 const dev = process.env.NODE_ENV !== 'production'
6 const app = next({ dev })
7 const handle = app.getRequestHandler()
8
9 app.prepare().then(() => {
10   createServer((req, res) => {
11     const parsedUrl = parse(req.url, true)
12     handle(req, res, parsedUrl)
13   }).listen(3000, (err) => {
14     if (err) throw err
      console.log(`Ready on http://localhost:3000`)
    })
  })
```

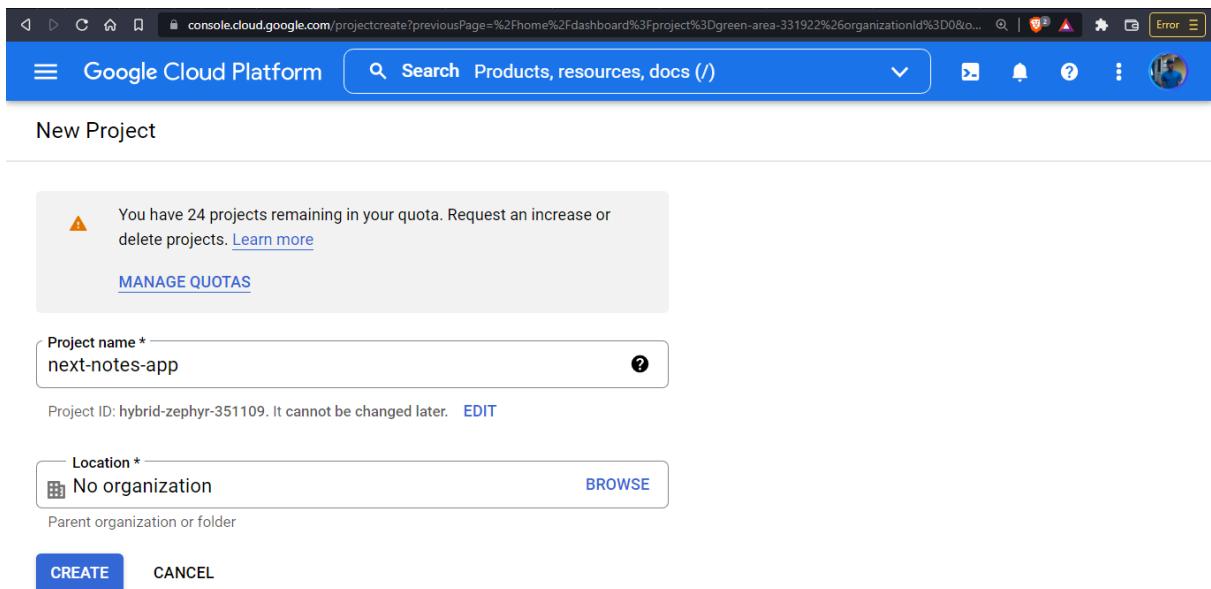
Ajuda configurando google

1 Faça login no console do google:

<https://console.cloud.google.com/home/dashboard>

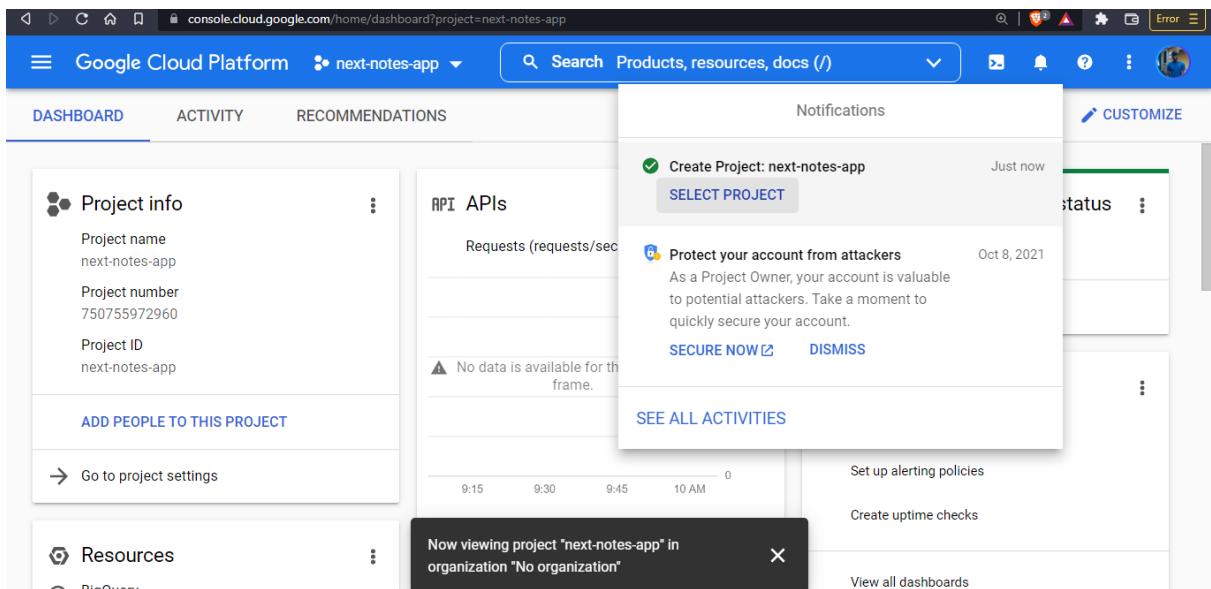
Clique no menu suspenso que abre um modal; no canto superior direito do modal, clique em **Novo Projeto**, para criar um novo projeto.





Insira o nome do projeto e clique em **criar**.

2 Uma vez criado o projeto, clique em **SELECIONAR PROJETO** no modal de notificações.



Agora, em nosso novo aplicativo podemos abrir a barra lateral, ir para **APIs & Services > Credentials**. A primeira coisa que precisamos fazer é configurar a tela de consentimento:

The screenshot shows the Google Cloud Platform API Credentials page for a project named 'next-notes-app'. The left sidebar has 'Credentials' selected. A callout box highlights the 'CONFIGURE CONSENT SCREEN' button, which is underlined with a red arrow.

Em seguida, escolha o tipo de usuário **externo**.

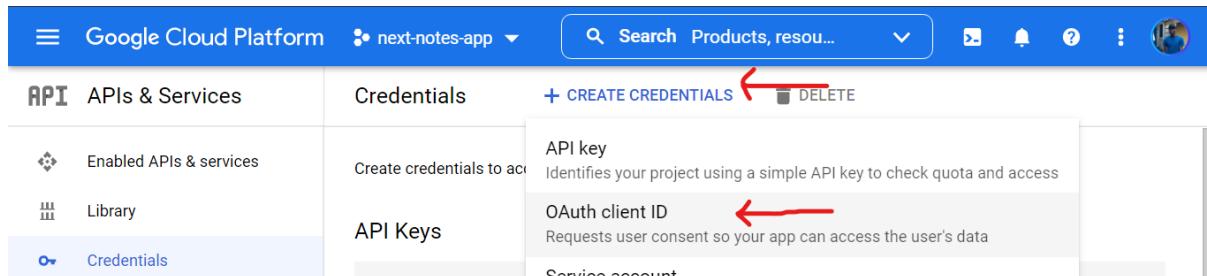
The screenshot shows the 'OAuth consent screen' configuration page. Under 'User Type', the 'External' option is selected (indicated by a red circle). The 'CREATE' button is visible at the bottom.

Na próxima tela, adicionamos as informações do nosso aplicativo.

The screenshot shows the 'Edit app registration' page. Under 'App information', the 'App name' is set to 'next-notes-app' and the 'User support email' is set to 'miracleficient@gmail.com'. An 'App logo' field contains 'memo_1f4dd.png'.

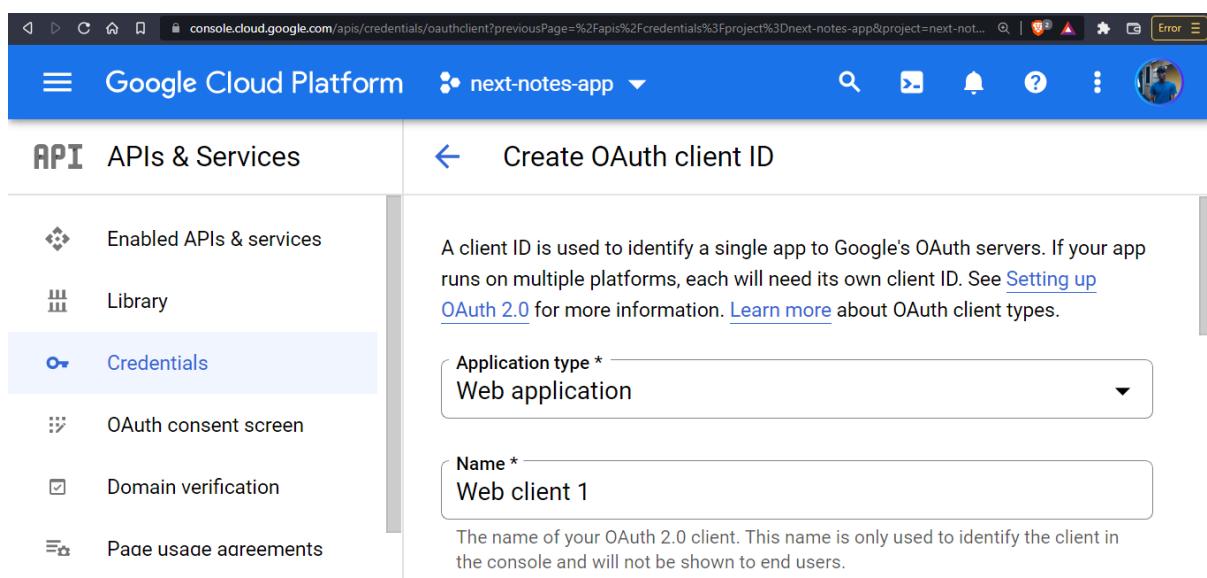
Em seguida, adicionamos as informações do desenvolvedor e clicamos em **Salvar e Continuar** nesta tela e nas telas seguintes até que seja concluído,

e voltamos ao painel. No menu lateral, em **API e CREDENCIAIS**, clique em **Credenciais** para criar um novo **OAuth Client ID.



The screenshot shows the Google Cloud Platform API & Services Credentials page. The left sidebar has 'Credentials' selected. In the main area, there's a 'Create credentials to access' dropdown with 'API key' and 'OAuth client ID' options. A red arrow points to the 'OAuth client ID' option, which is described as 'Requests user consent so your app can access the user's data'. There are also 'Service account' and 'Delete' buttons.

Na próxima tela, selecionaremos o tipo e o nome do aplicativo,



The screenshot shows the 'Create OAuth client ID' form. The left sidebar has 'Credentials' selected. The main form has 'Application type *' set to 'Web application' and 'Name *' set to 'Web client 1'. A note below says 'The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.'

E adicione um URI autorizado do Google para desenvolvimento -

`http://localhost:3000/api/auth/callback/google`

The screenshot shows the Google Cloud Platform API & Services Credentials page. On the left sidebar, 'Credentials' is selected. In the main area, it says 'Create OAuth client ID'. Under 'Authorized redirect URIs', there is a field containing 'http://localhost:3000/api/auth/callback/google'. A note below says 'Note: It may take 5 minutes to a few hours for settings to take effect'. At the bottom are 'CREATE' and 'CANCEL' buttons.

Clique em **Criar** e, após a criação do ID do cliente, teremos nosso ID do cliente e o segredo do cliente. Crie um `.env` arquivo para salvar essas variáveis

```
GOOGLE_CLIENT_ID=<client_ID_here>
GOOGLE_CLIENT_SECRET=<client_secret_here>
DATABASE_URL=mongodb+srv://username:password@cluster0.ba9ic.m
NEXTAUTH_SECRET=somesecret
NEXTAUTH_URL=http://localhost:3000
```