



INSTITUTO FEDERAL

São Paulo

Câmpus Cubatão

CTII 417

Apostila Aplicativo Octoplus

Allan Ferreira da Silva Moura

Kauanny Santos Silva

Kauã Felipe da Silva Junior

Mariany Generoso da Silva

Victoria Luisa Viana Pontes

CUBATÃO

2023

Apostila - OctoPlus

O que é o OctoPlus?

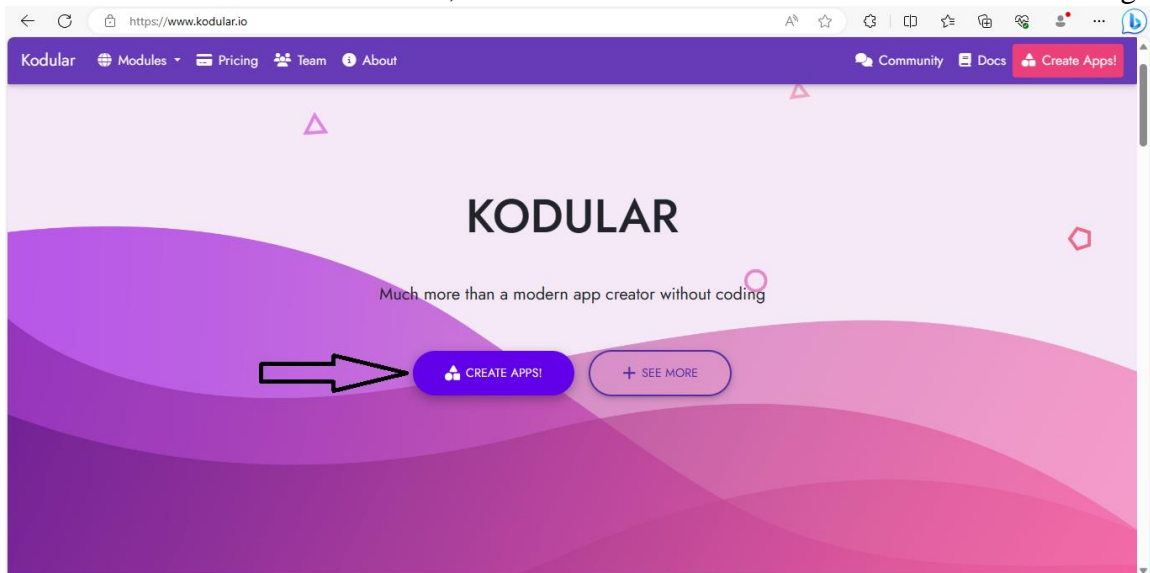
OctoPlus é um aplicativo de gerenciador de tarefas, onde você é capaz de organizar suas tarefas do dia a dia, tudo na palma de sua mão.

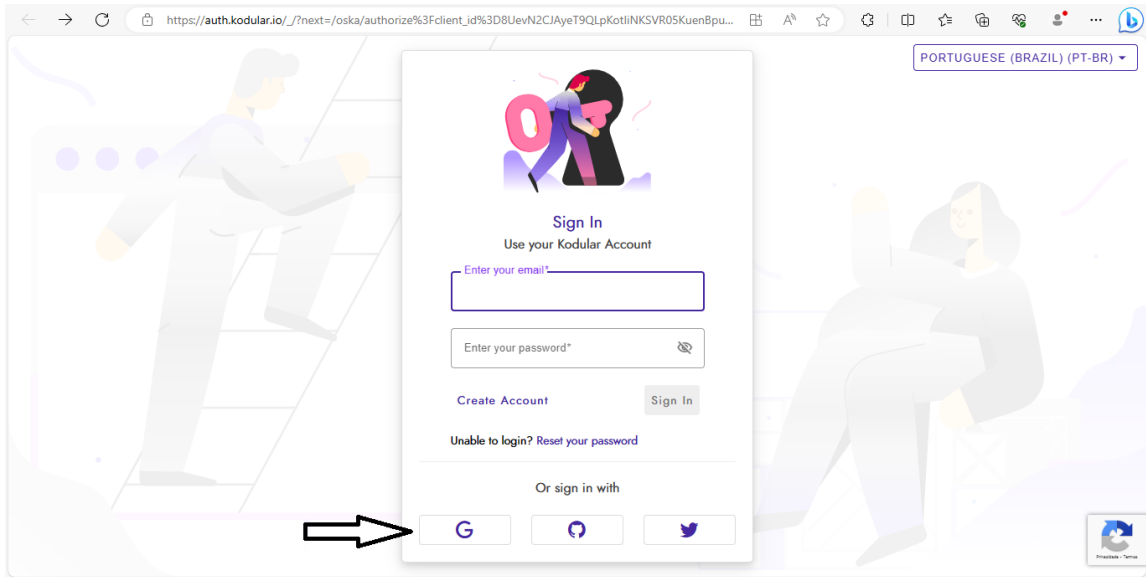
Ferramentas usadas para a criação:

- Computador (com acesso à internet)
- Site da web: www.kodular.io
- Um smartphone com acesso à internet
- App do simulador para Android: [Kodular Companion – Apps no Google Play](#)
- Extensão usada no aplicativo: [Horizontal_Calendar](#)

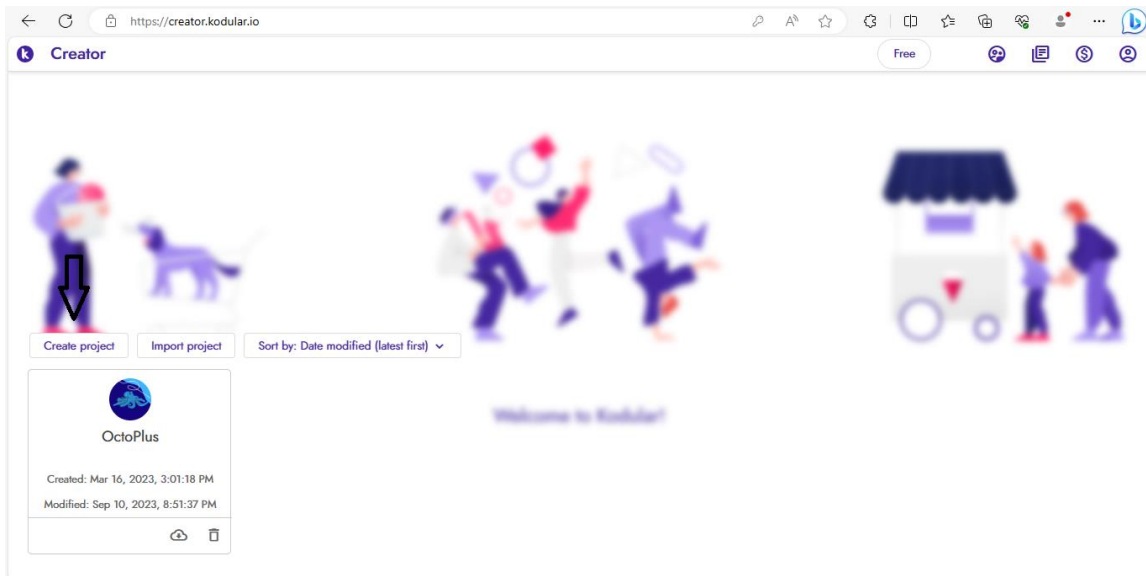
Passo a passo (criação da conta e projeto)

Criaremos nossa conta no Kodular, aconselhamos a criar a conta usando sua conta Google

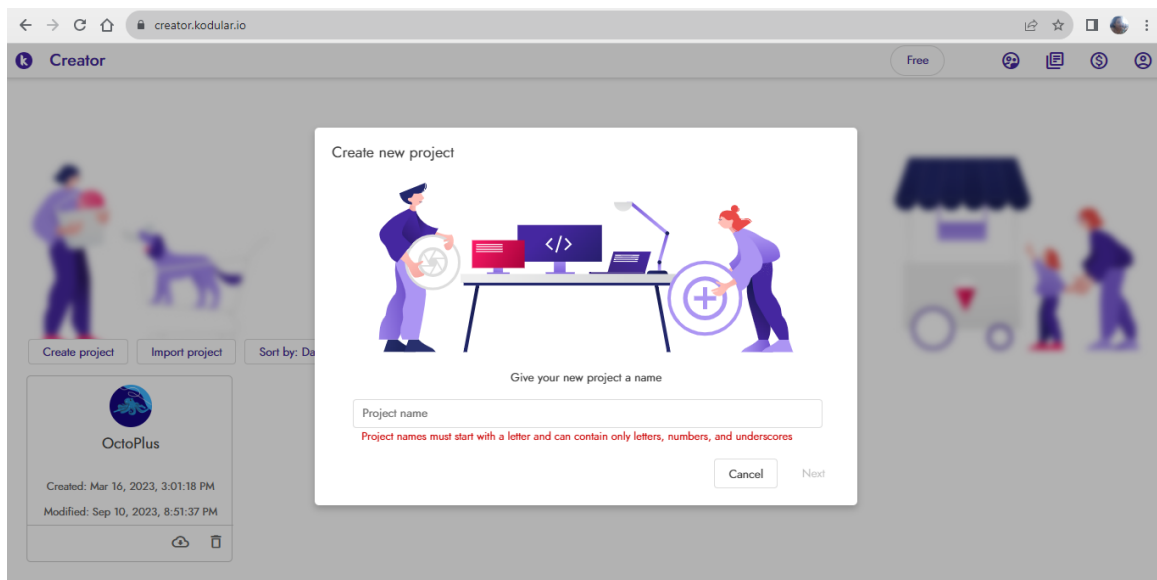




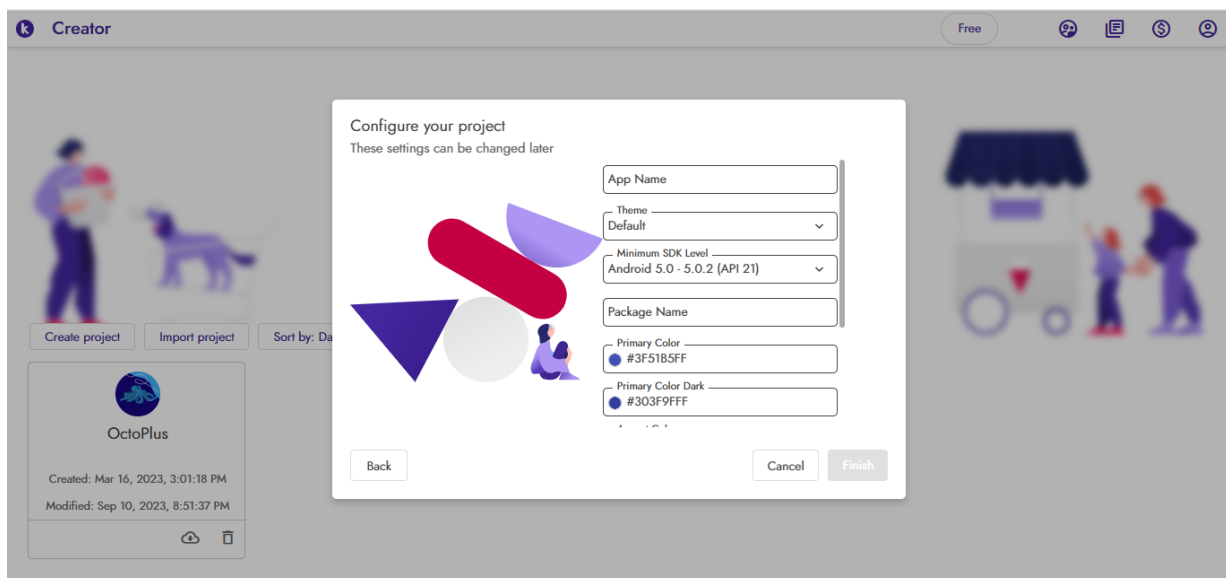
Depois de criarmos a conta iremos criar nosso projeto em Create Project:



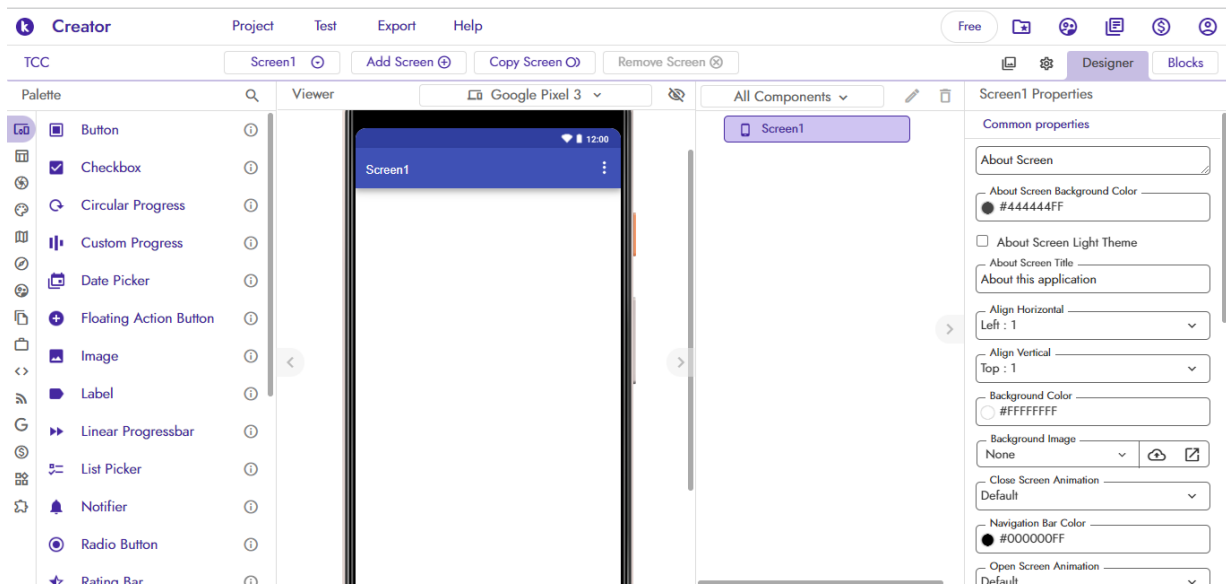
Irá aparecer essa tela para criar o nome desse projeto:



Agora vamos dar o nome ao nosso aplicativo e clicar no botão “finish”:



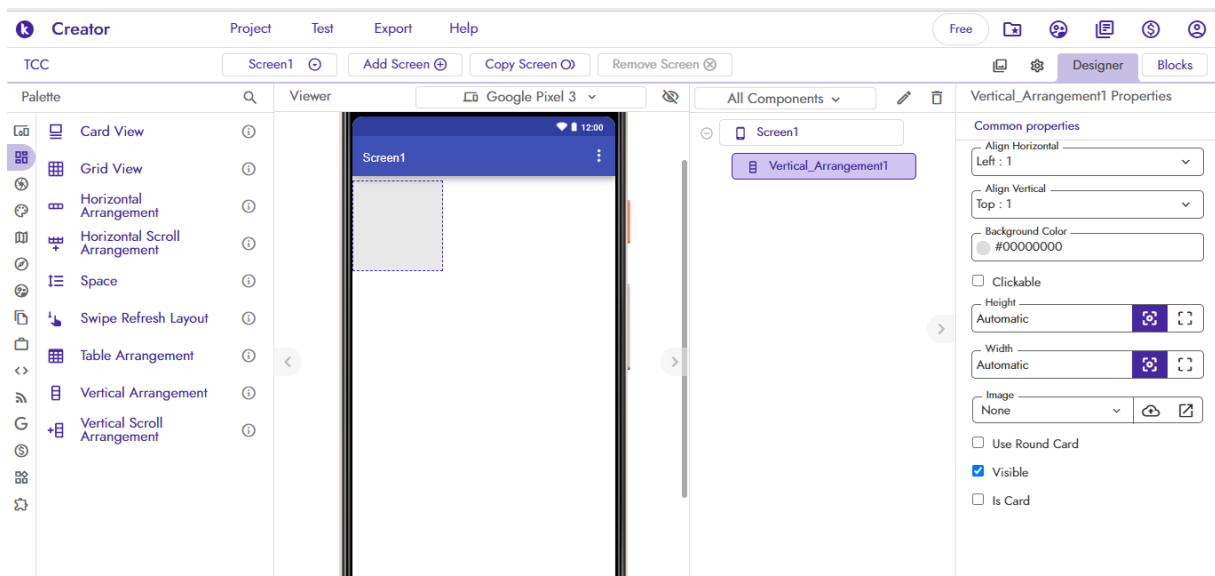
Essa será a tela em que moldaremos o nosso aplicativo:



Como criar a tela de login?

Para criarmos a tela de login precisamos utilizar um banco de dados para armazenar o login e a senha. Para isso iremos usar outra plataforma, o Firebase. Mas antes vamos primeiro criar as funções no aplicativo e depois iremos integrar com o Firebase.

Nosso aplicativo terá várias telas, certo? Então adicionaremos o “Vertical Arrangement” que é só digitar na aba de pesquisa que já irá aparecer.



Iremos arrastar os seguintes itens para dentro desse Vertical Arrangement. É importante ressaltar que todos os itens citados têm que estar dentro desse recurso.

Label – A label será usada como um aviso quando o usuário não digitar nada nos campos de Login e Senha. Essa label deveremos deixar sem o escrito “Text for Label1”. Para isso

clicamos no Recurso Label1 da aba ao lado e logo em seguida terá um campo escrito “Text” onde apagaremos esse “Text for Label1”.

Text Box – Iremos adicionar dois “Text_Box”, um será para o login e outra será para a senha. Como vemos na tela os Text_Box são campos de digitação.

Button – O recurso Button são os botões do aplicativo. Adicionamos três botões para logar, registrar e sair.

Card View - O Card View é usado para melhorar a apresentação de informações do nosso aplicativo, permitindo personalização visual e interações específicas para esse app.

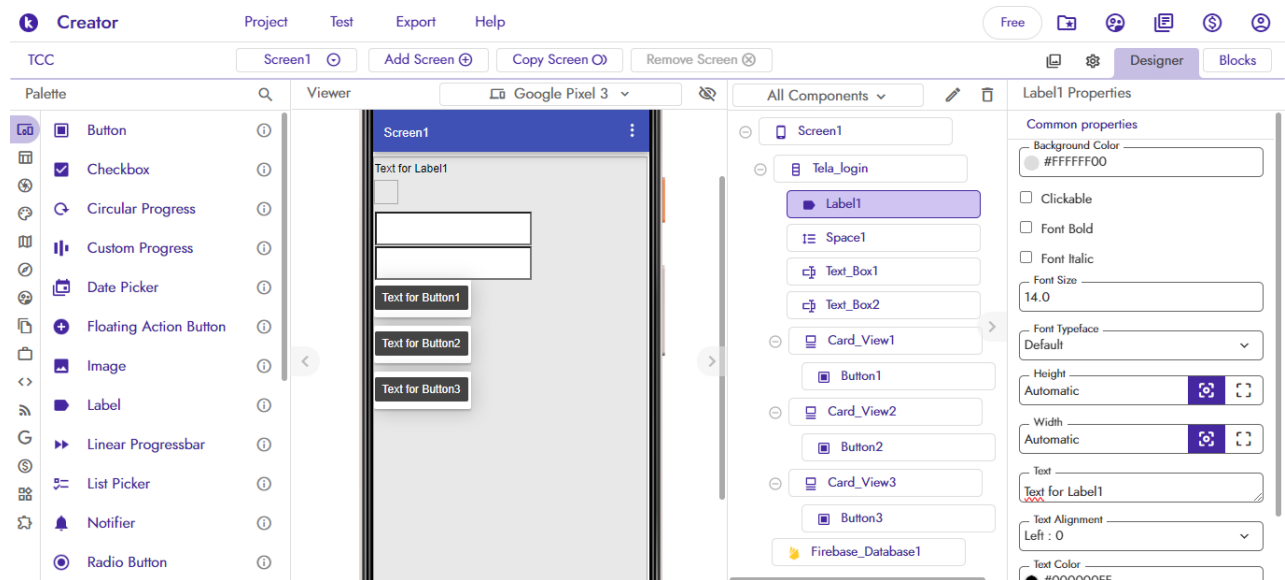
Space – O Space é mais utilizado como uma ferramenta de design.

Adicionamos também o Firebase, mas esse recurso será explicado mais para frente.

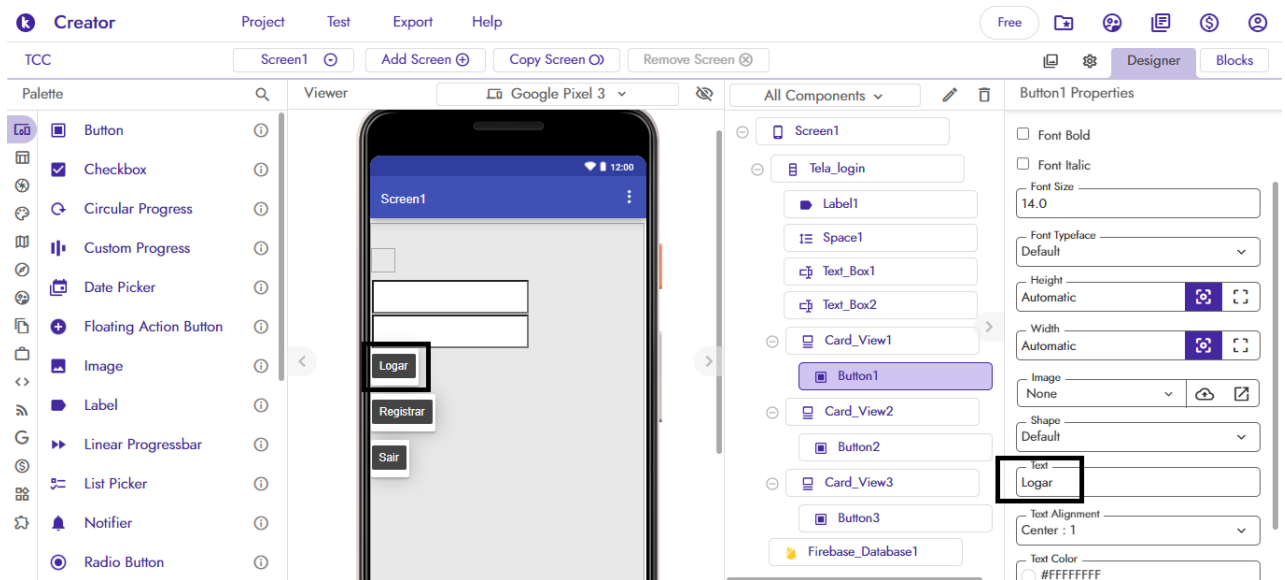
O botão logar servirá para aquele usuário que já fez o registro antes, ou seja, não é para aquele usuário de primeira viagem;

O botão registrar servirá para abrir uma nova tela que iremos elaborar mais para frente. Esse será para a primeira vez que o usuário entrar no aplicativo;

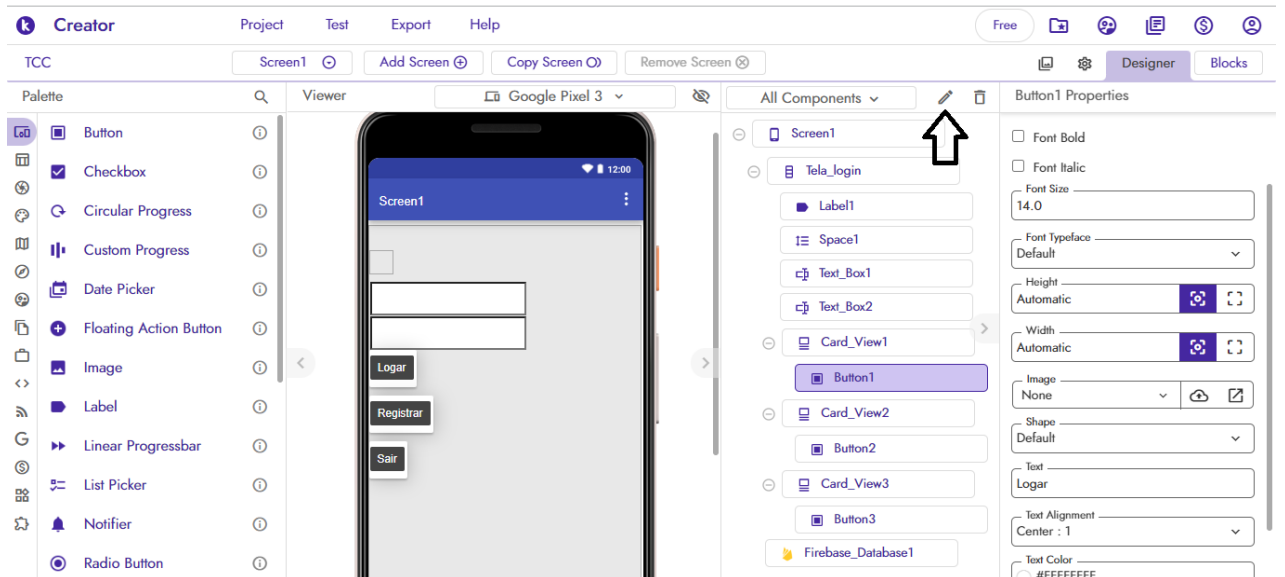
O botão sair será para quando o usuário desejar fechar o aplicativo.



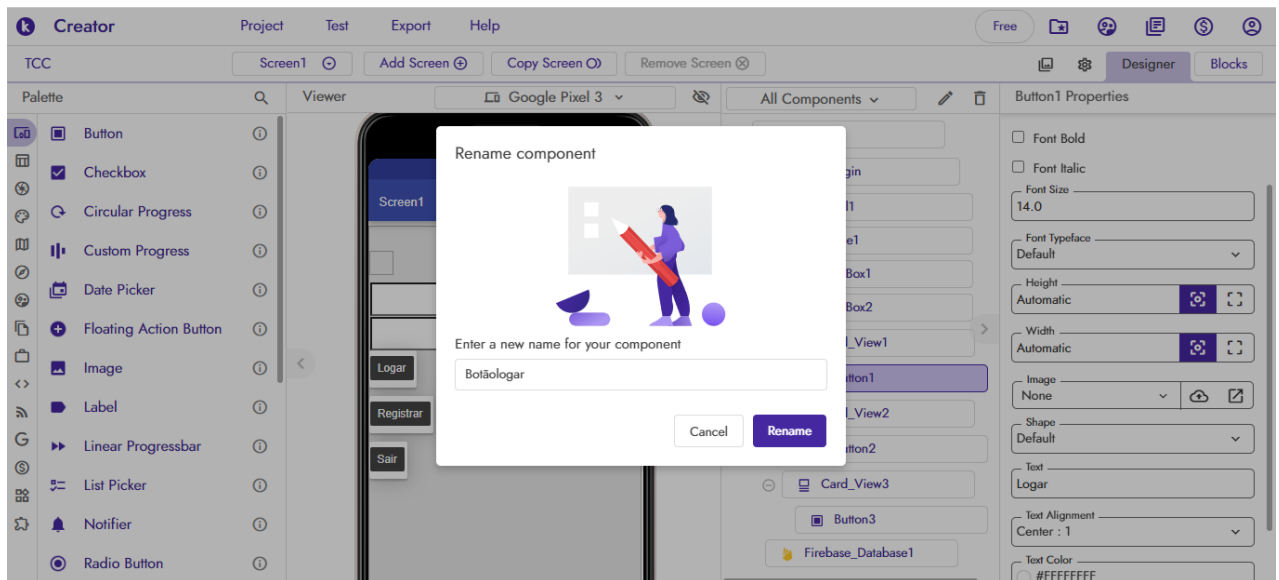
Agora iremos definir os nomes desses recursos para ficar mais fácil de identificar. Iremos fazer o mesmo processo da Label citado anteriormente nos botões, mas ao invés de apagar escreveremos suas funções.



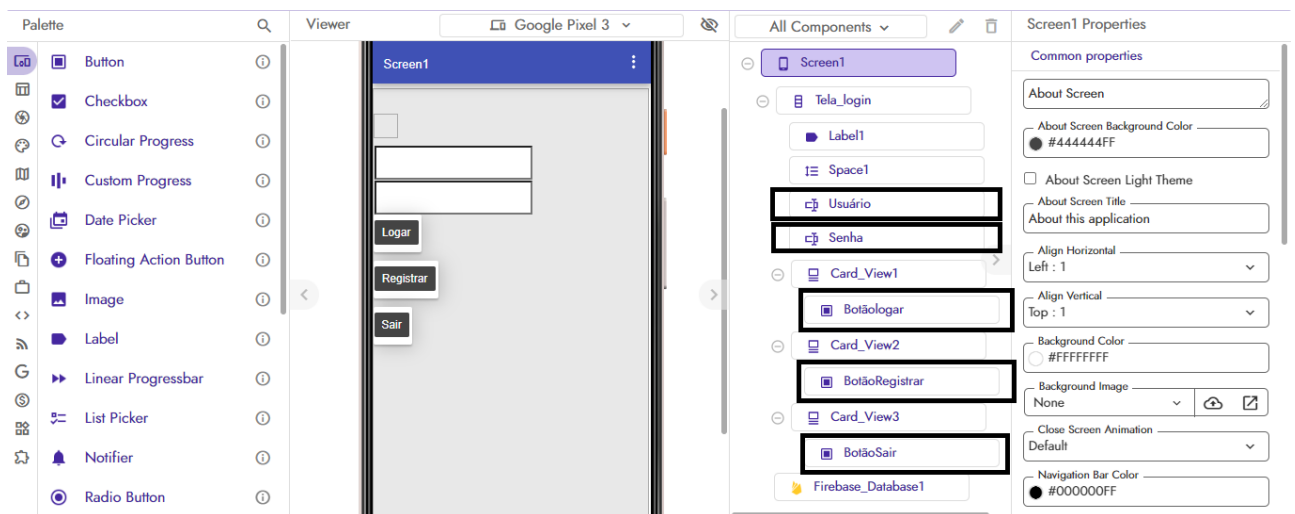
Isso aparecerá na parte de layout, agora iremos mudar os nomes dos recursos para ficar mais fácil de identificar quando montarmos nos Bloks. Para isso basta selecionar um desses recursos e clicar no lápis da aba.



Nesse botão irá aparecer a seguinte tela:

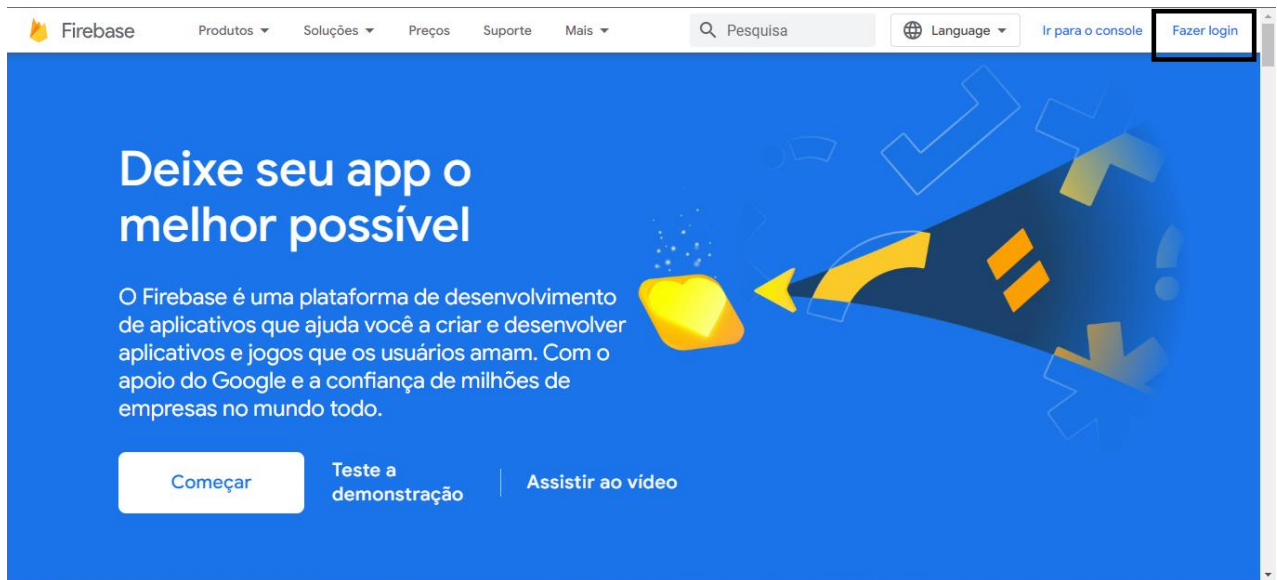


Nós estamos definindo o botão logar na tela acima, mas iremos definir o nome de todos os recursos, isso fica a critério de cada um.

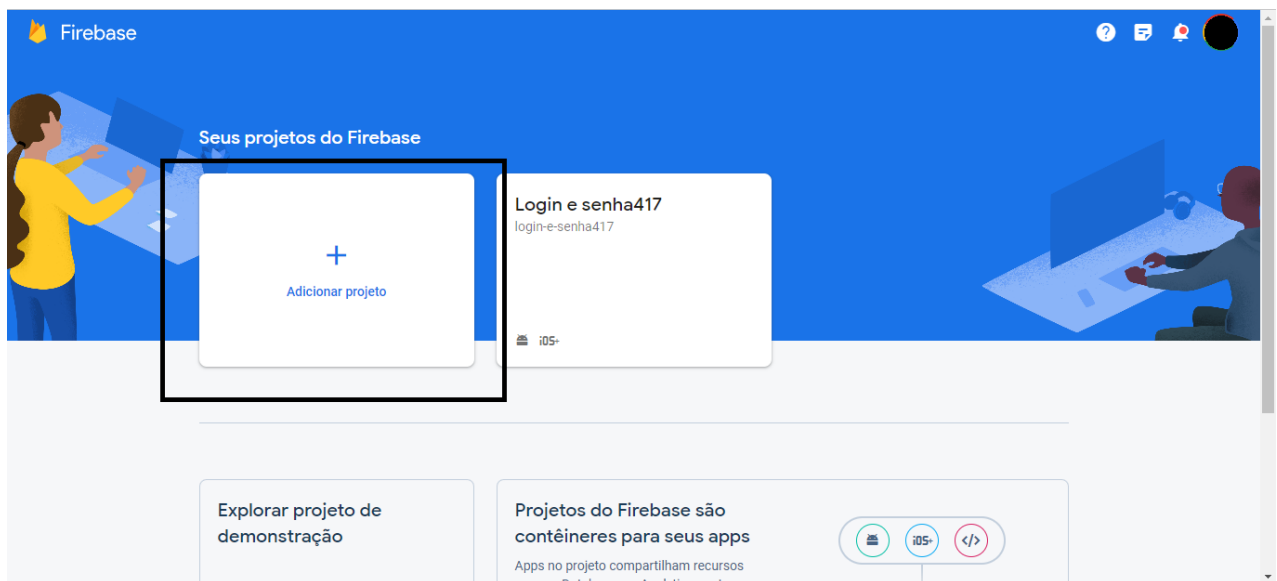


Criando um banco de dados Firebase

No site <https://firebase.google.com/?hl=pt> iremos criar uma conta, para isso basta clicar em “Fazer Login”. Para criar a conta é necessário uma conta Google.



Depois de criar a conta clicaremos em “Adicionar projeto”, no exemplo abaixo há um projeto que usamos no aplicativo Octoplus



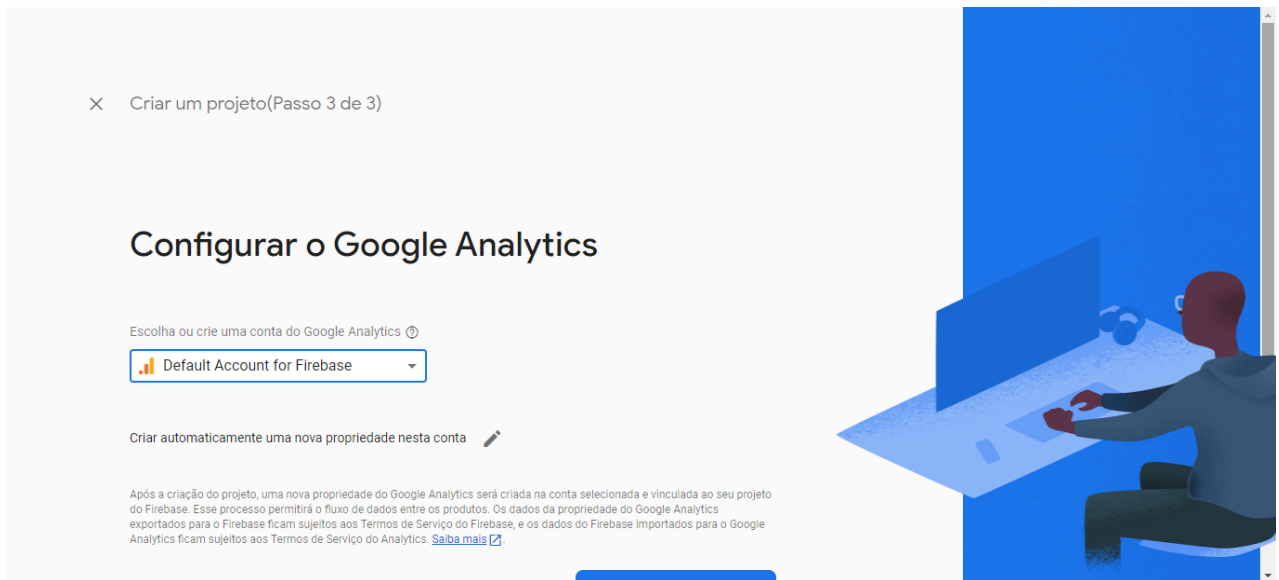
Depois irá aparecer a tela para nomear o nome do projeto, decidimos colocar de AppKodular. Após esse passo basta clicar em continuar.



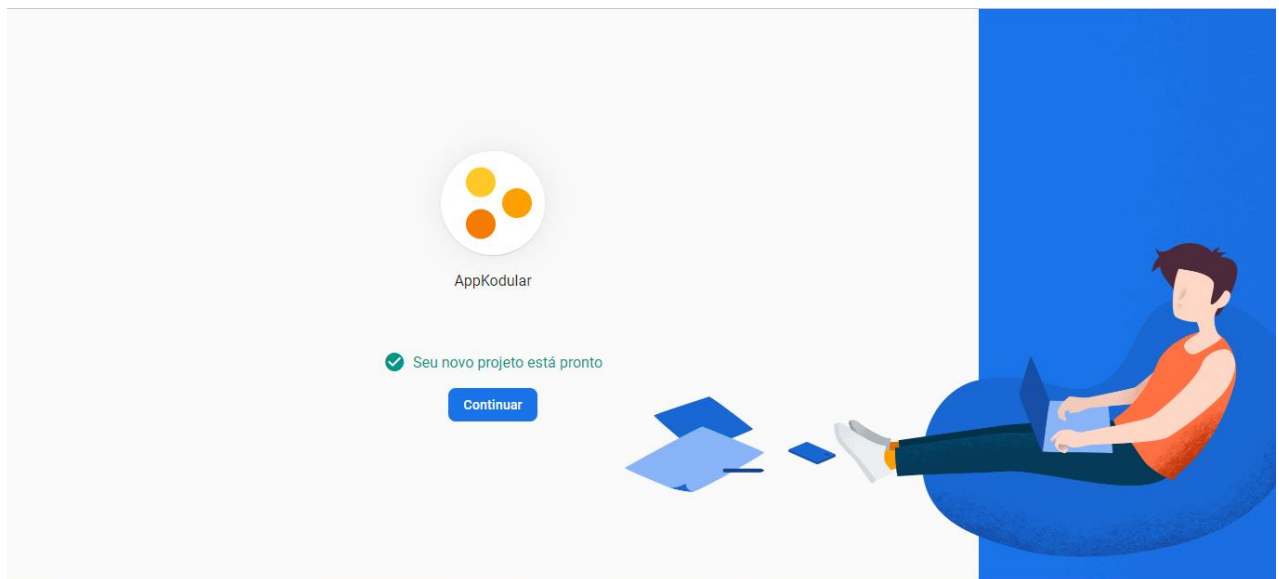
Depois irá aparecer a seguinte tela, basta rolar para baixo e clicar em continuar.



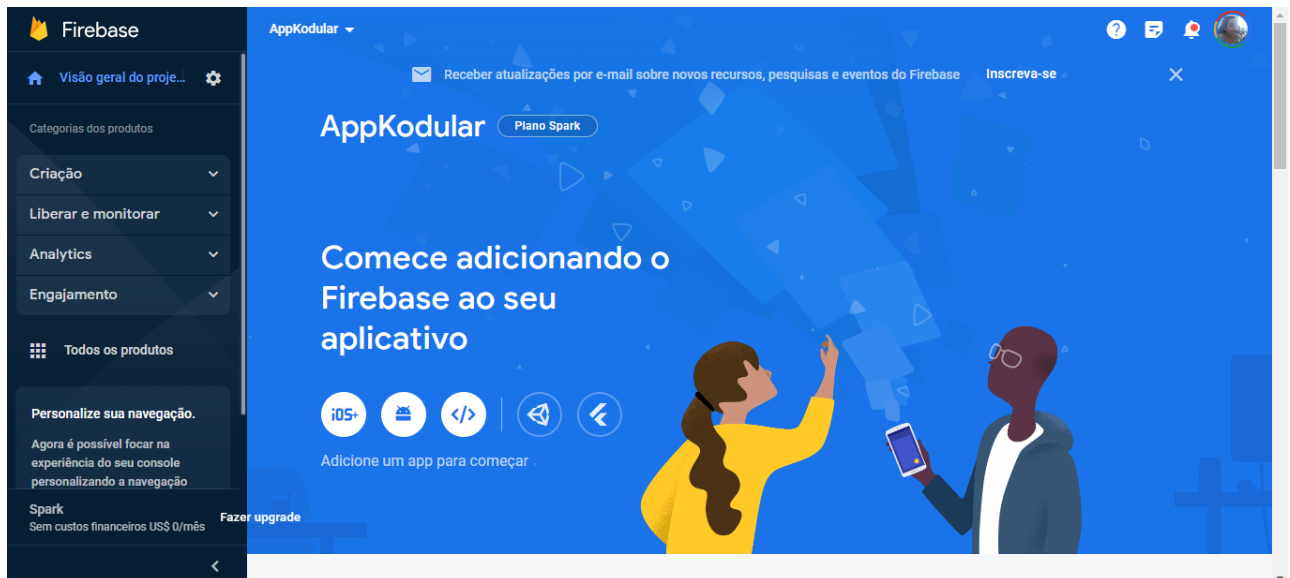
Irá aparecer uma tela para escolher uma conta, como vemos abaixo selecionamos o “Default Account for Firebase”. Logo clicaremos em “Criar projeto” que nos levará a uma página de carregamento, mas isso demora apenas alguns segundos.



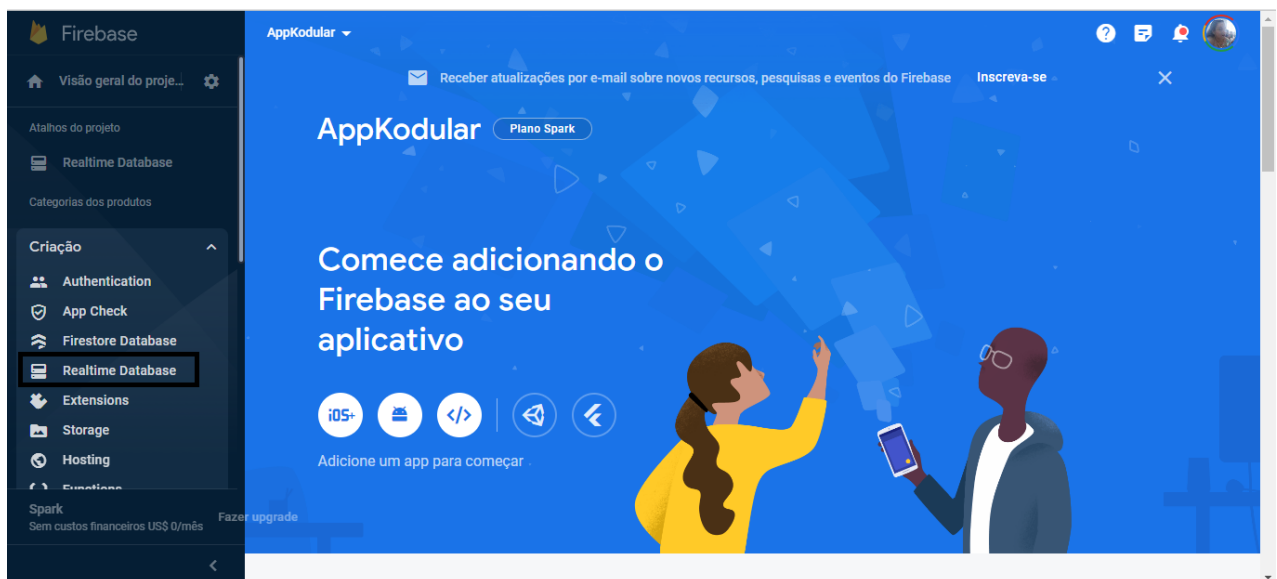
Depois do projeto já carregado devemos clicar no “Continuar”.



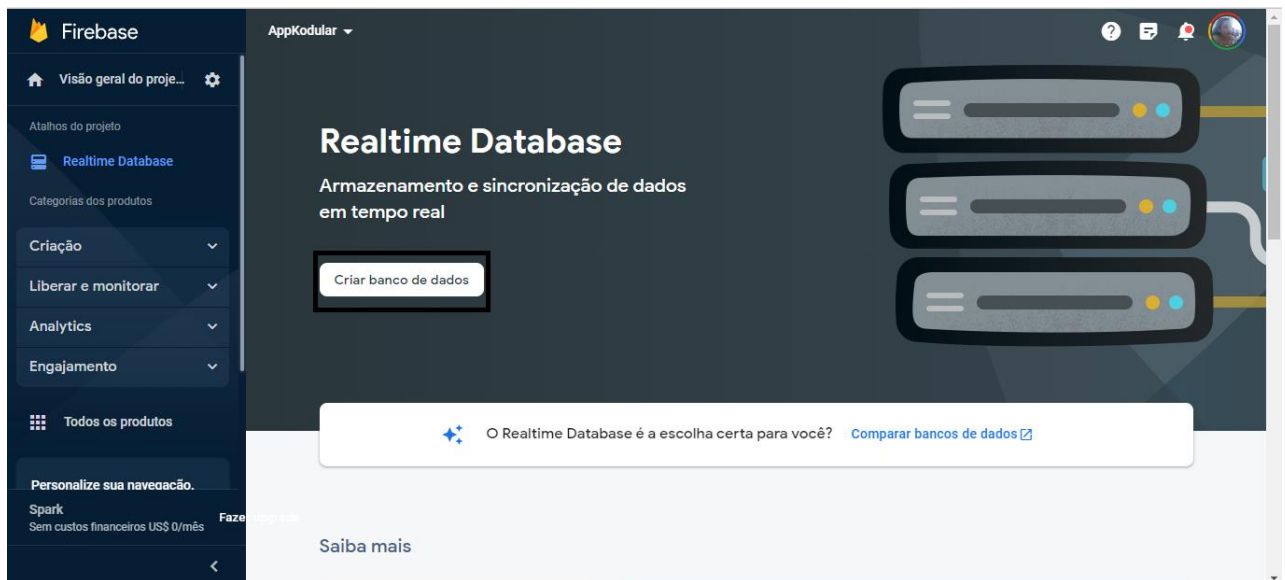
Agora essa será a página do nosso projeto



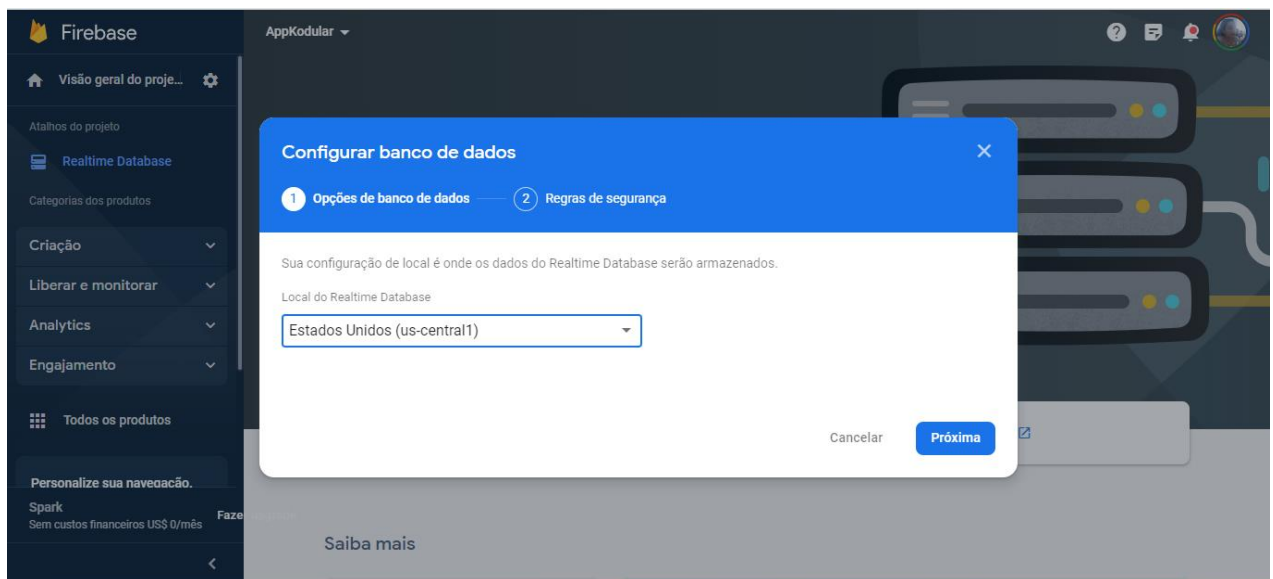
Para criarmos nosso banco de dados vamos em Criação que irá revelar uma aba, nessa aba terá o Realtime Database.



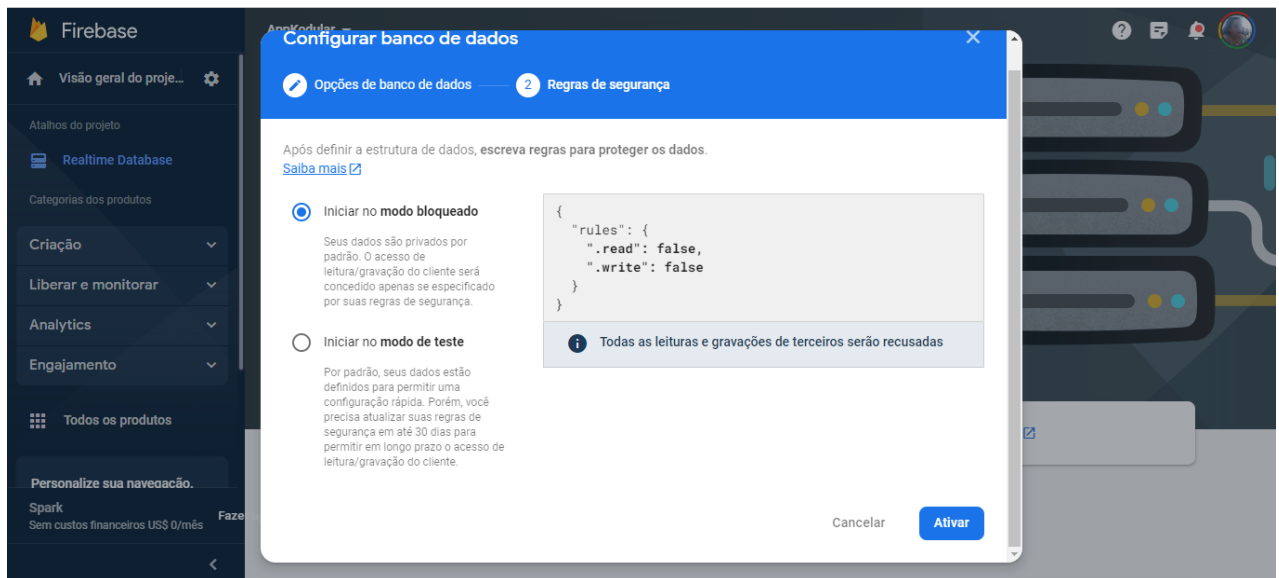
Em Realtime Database irá aparecer o “Criar um banco de dados”, clique nesse botão.



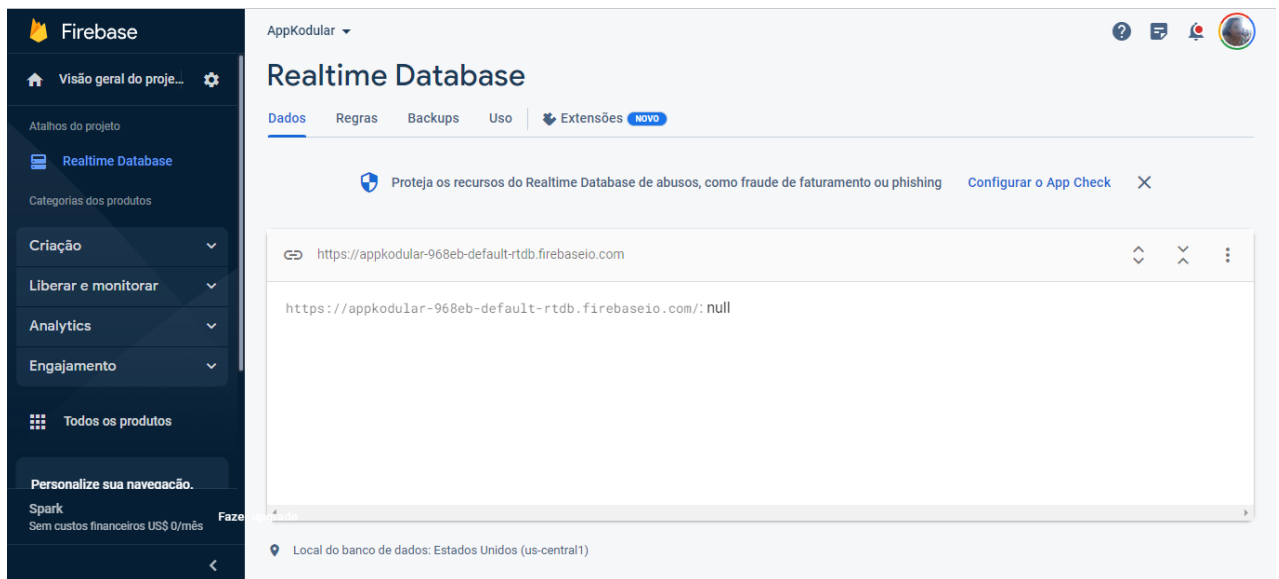
A tela seguinte deixaremos o local do Realtime Database como os Estados Unidos.



A tela seguinte devemos deixar no modo bloqueado e ativar o banco de dados.



Pronto, criamos o nosso banco de dados!

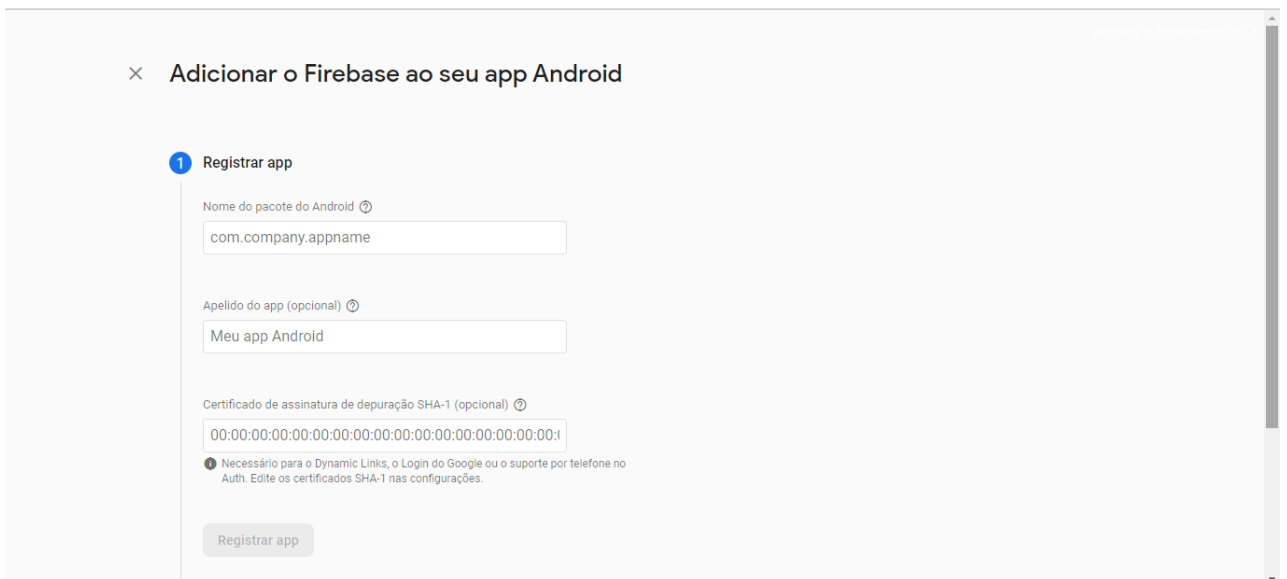


Agora vamos definir o banco de dados para iOS e Android, para isso ao voltarmos na página principal e iremos adicionar um por vez o Android e o iOS.

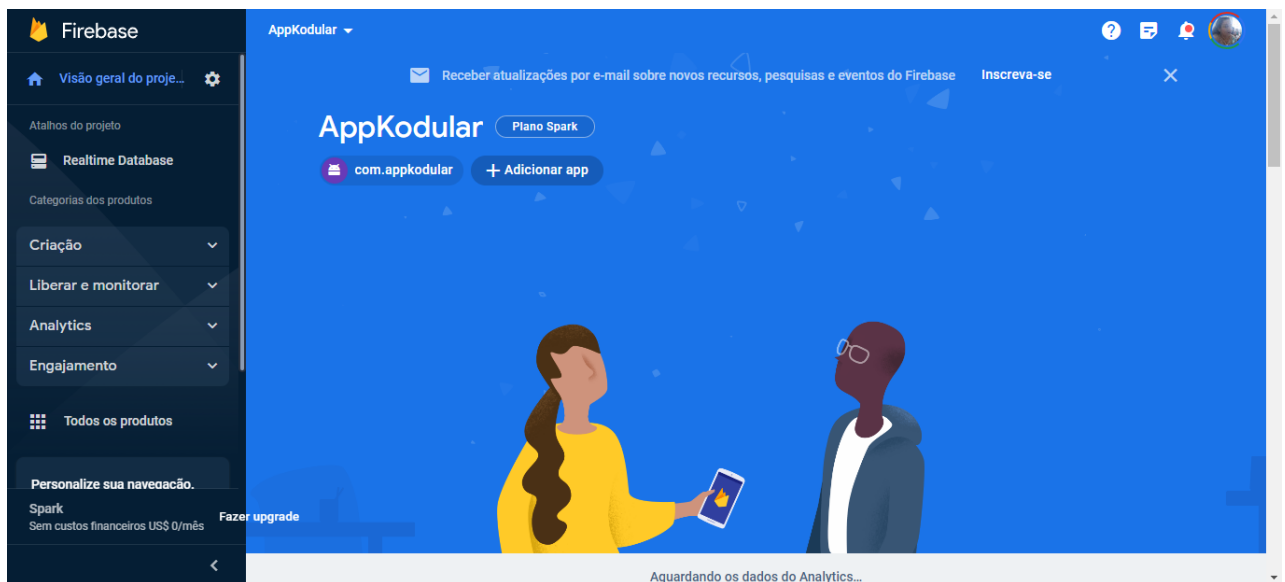
Vamos começar com o Android.



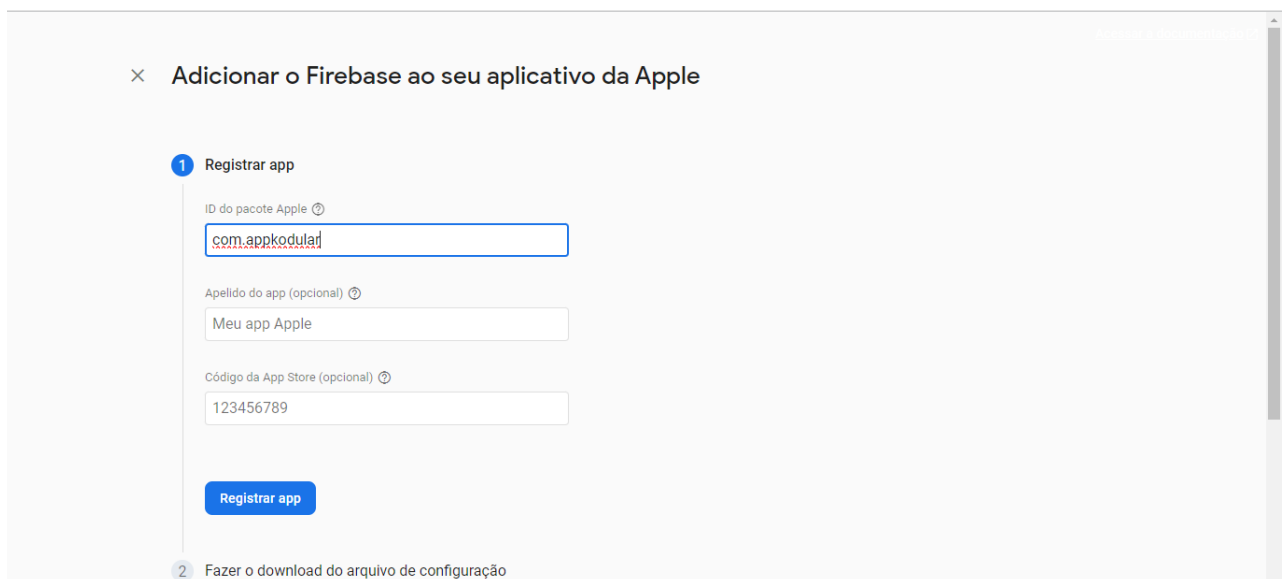
Ao clicar no ícone com o símbolo Android irá aparecer a seguinte tela.



Vamos preencher o nome do pacote do Android com um “com.” na frente, no exemplo a seguir preenchemos como “com.appkodular”. Logo em seguida podemos registrar o app.

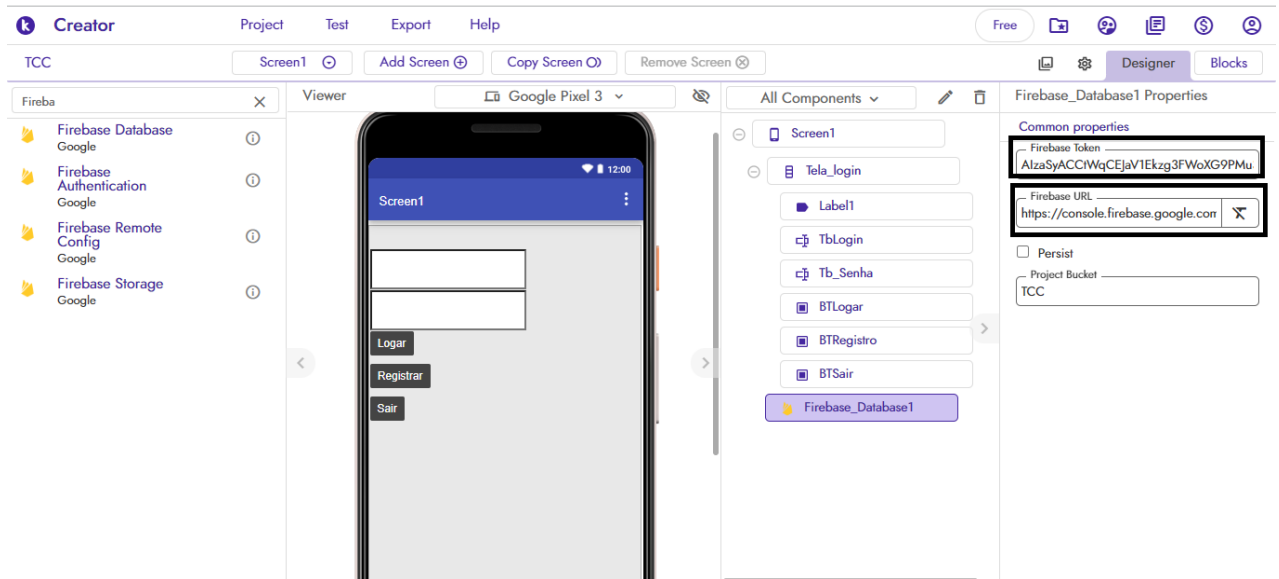


O ID do pacote Apple nomeamos da mesma forma que o Android. Depois só “registrar app” e prosseguir até no “Continuar no console”



Firestore para o Kodular

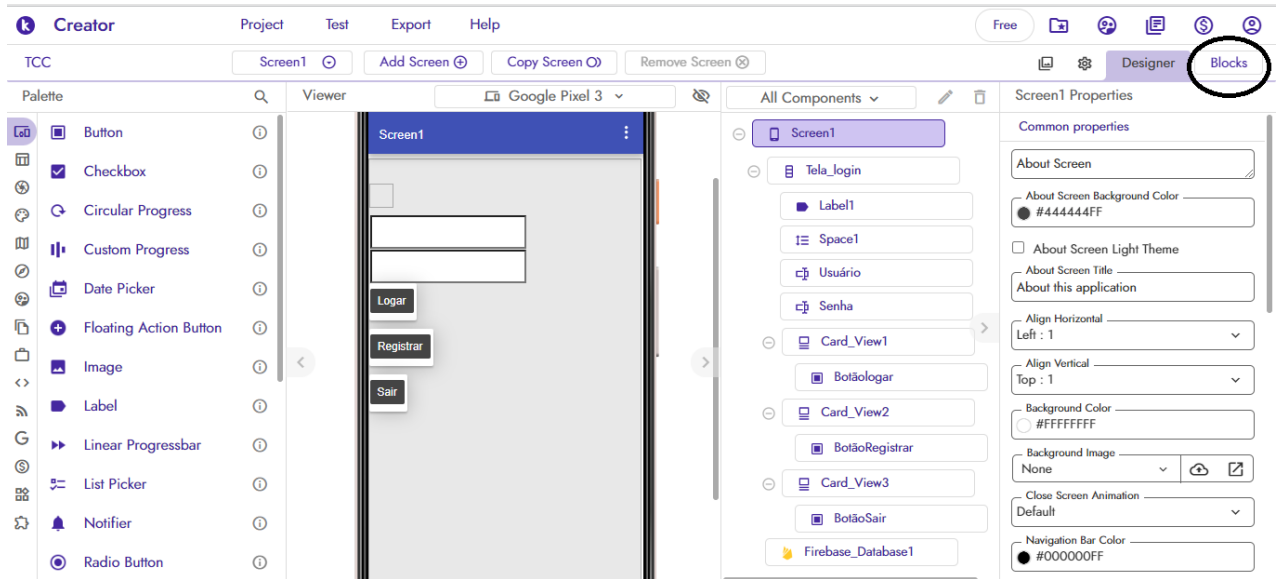
No kodular ao selecionarmos o Firebase irá aparecer dois campos na aba da lateral a direita. Esses campos iremos preencher com as informações do banco de dados que acabamos de criar. São o Firebase URL e o Firebase Token, ambas informações são encontradas nas “Configurações do projeto” visto anteriormente.



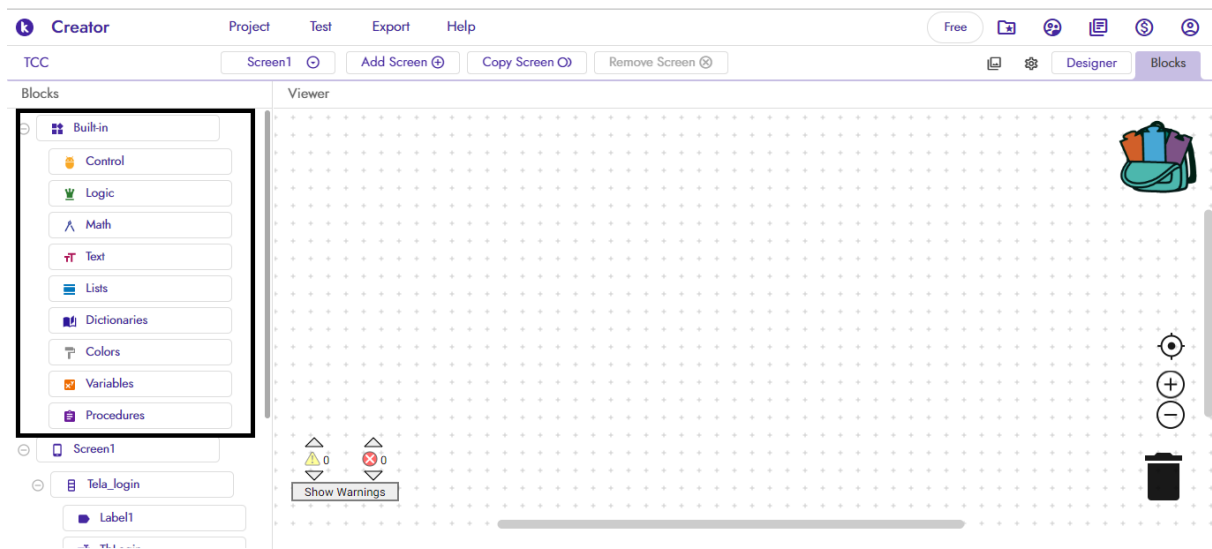
Blocks

Agora iremos focar em colocar algumas funções para depois de estar pronto mudarmos o design.

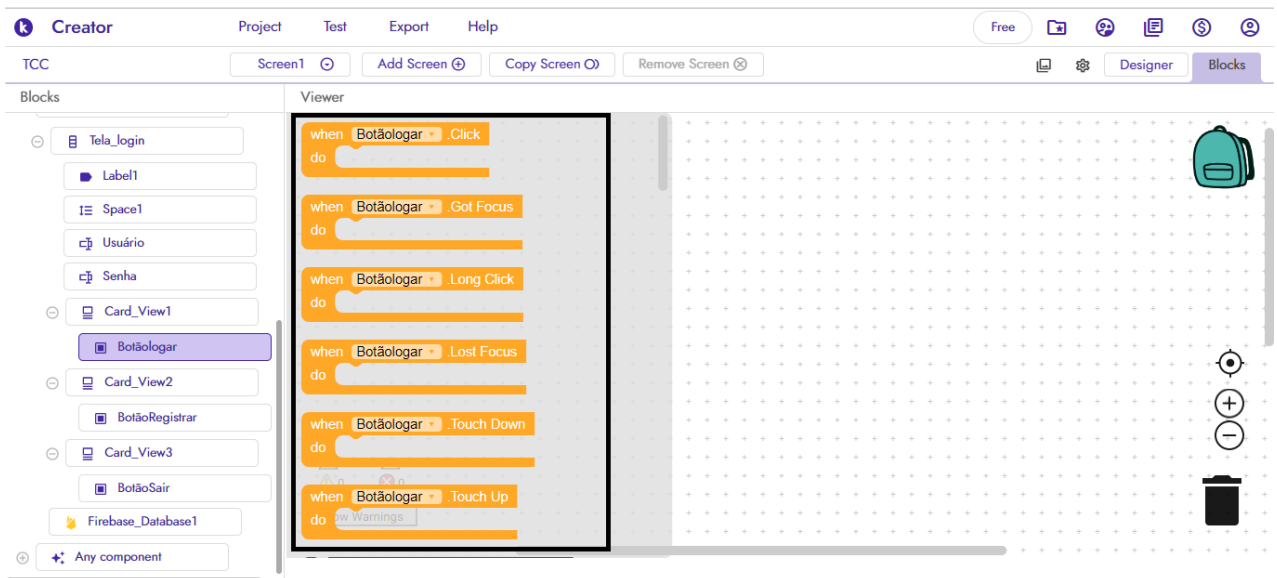
Há um botão superior esquerdo escrito “Blocks”, clicando nele irá aparecer a seguinte tela para começarmos a programar o nosso app.



Logo em seguida irá aparecer essa tela onde iremos montar toda a programação do nosso aplicativo. Na aba lateral irá ter as funções necessárias para isso, além de aparecer todos os recursos que adicionamos na outra página “Designer”.

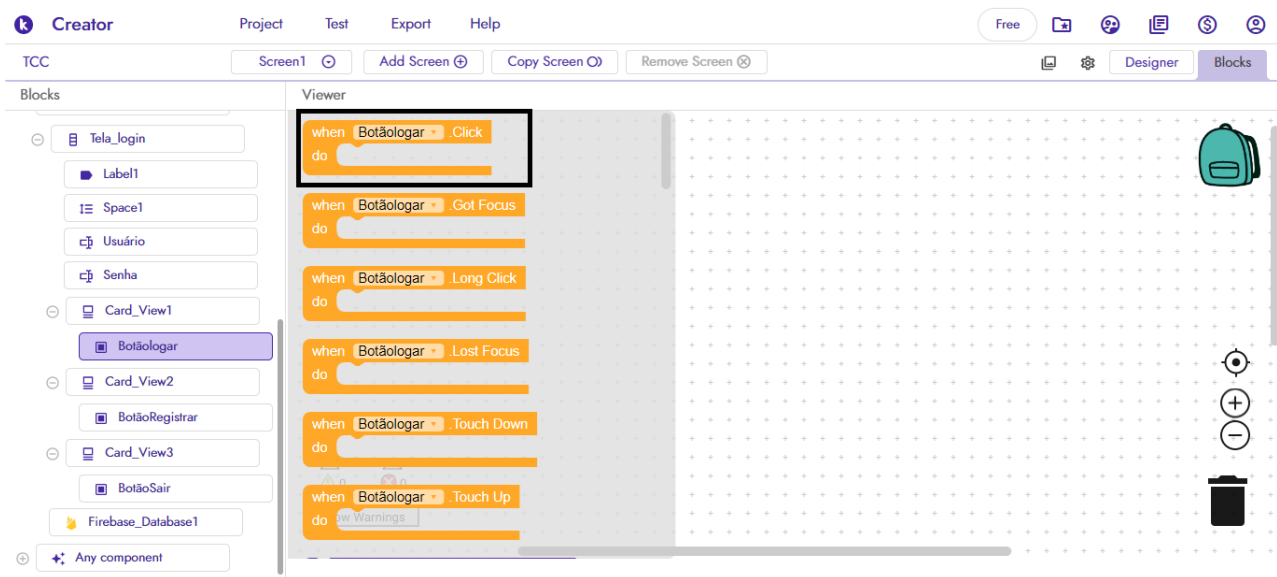


Iremos usar esses recursos também que darão para nós as funções que têm como fazer por exemplo como um “Button”. No exemplo abaixo estamos usando o “BotãoLogar”.



Criando a função para o botão Logar

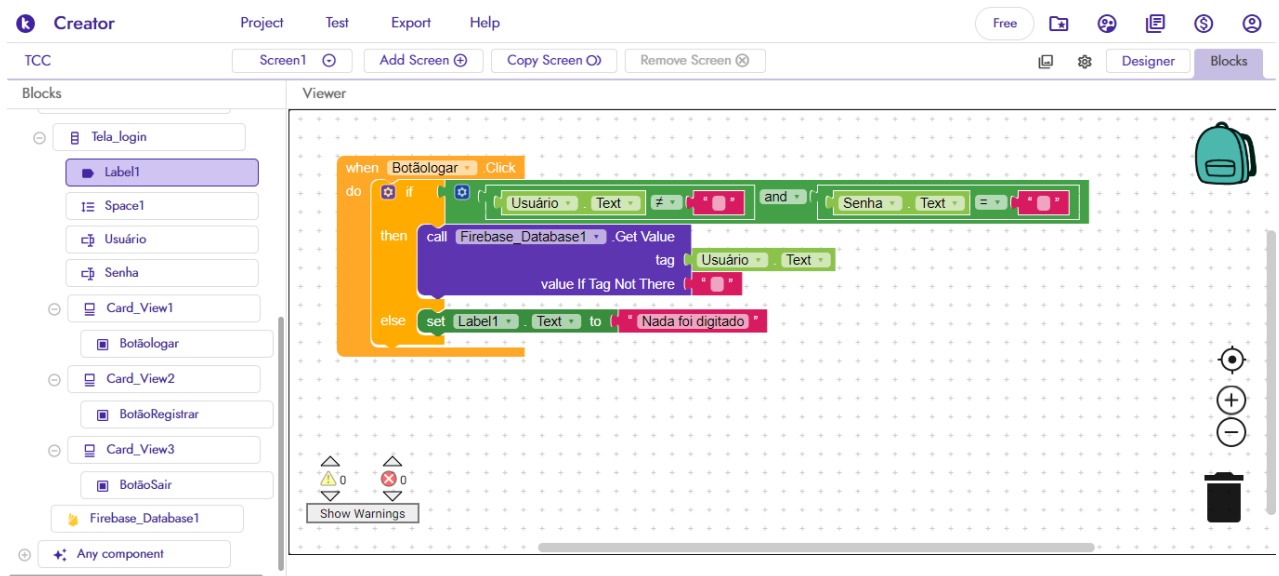
Como mostrado anteriormente, iremos clicar na aba lateral esquerda no recurso “BTLogar” e selecionar na primeira função mostrada na imagem. Essa função diz que quando o usuário clicar no botão Logar irá fazer a função que ainda iremos definir.



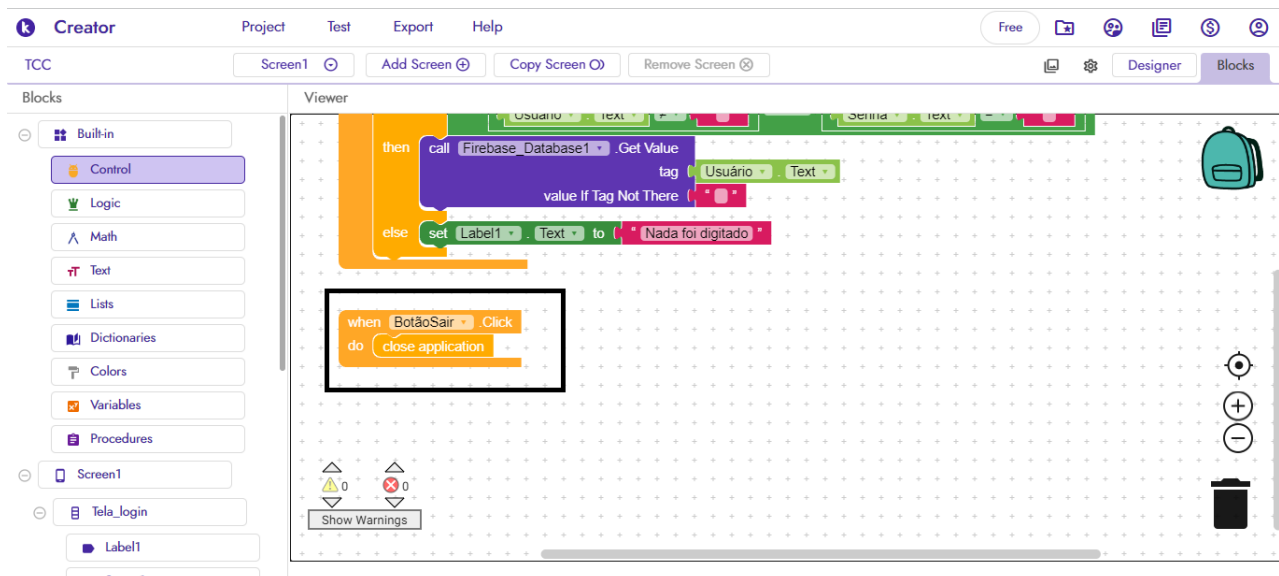
Agora vamos avançar um pouco na montagem dos Blocks para explicarmos suas funções.

Como visto na imagem abaixo nós já montamos todo o esquema da função do botão logar.

Quando o usuário clicar no botão “BotãoLogar”, colocamos o evento do usuário colocando suas informações em “Usuário” e na “Senha” será chamado o nosso banco de dados Firebase para armazenar, caso este não digite nada nos campos mencionados é chamado no “else” a Label1 trazendo a mensagem de “Nada foi digitado” como vemos no começo da apostila quando estávamos colocando os recursos em “Designer”.



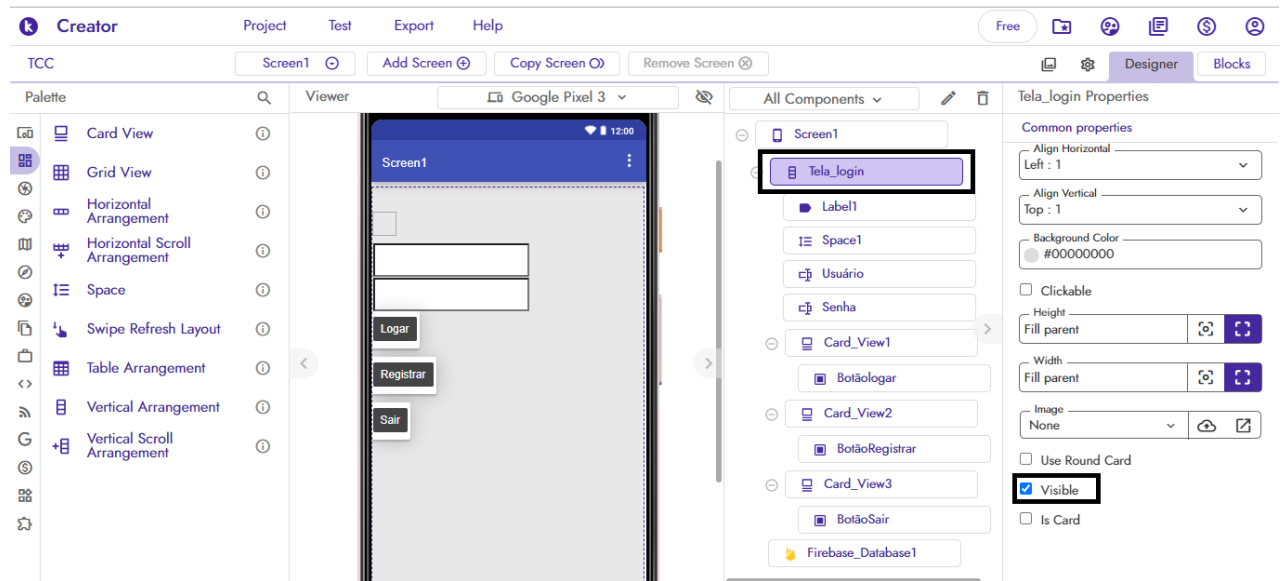
Agora vamos criar a função do “Botão sair”. Essa é a mais simples, vamos chamar o evento de clicar o botão e adicionar o “close application” para fechar o aplicativo.



Criar tela registro

Para adicionarmos um evento ao “BotãoRegistro” devemos criar uma nova tela para quando ele for clicado ser direcionado para essa nova tela. Para isso iremos fazer o mesmo processo que fizemos ao criar a nossa primeira página, mas deixaremos essa que criamos invisível para

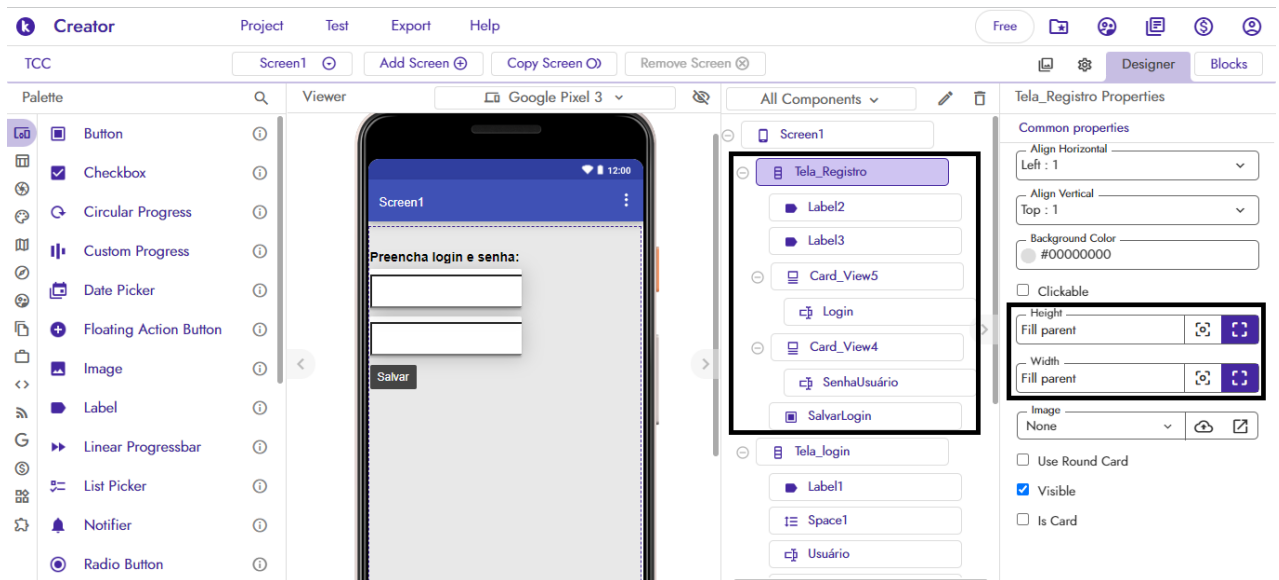
não sobrepor a nova página. Por isso devemos selecionar a página e desativar a opção “visible”.



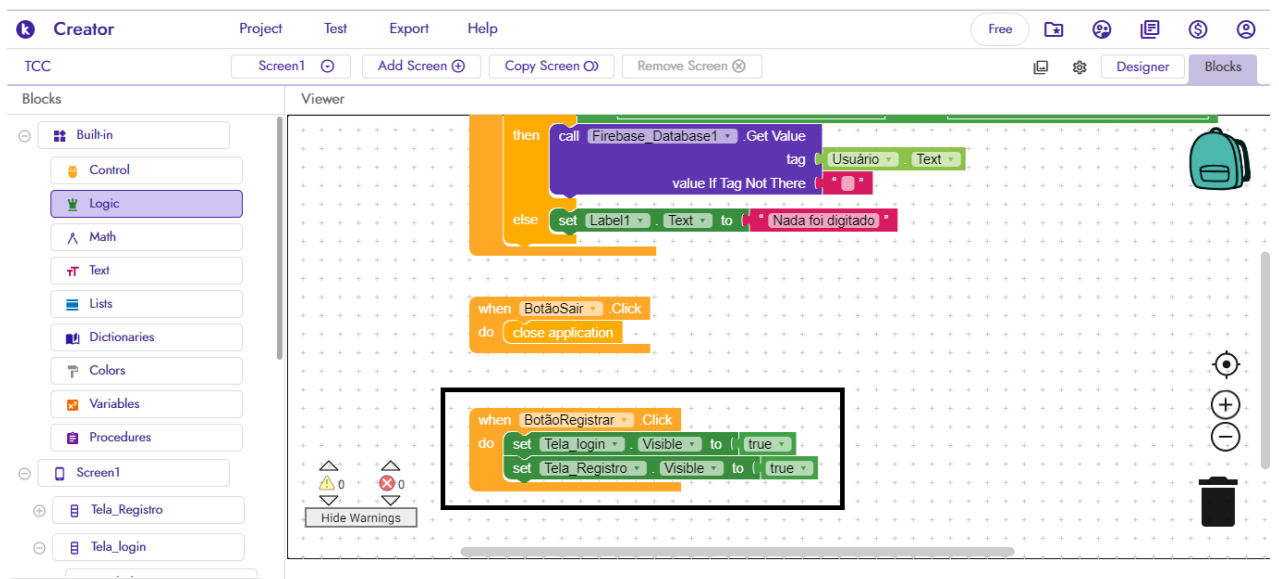
Agora adicionaremos um novo “Vertical Arrangement” e outros recursos.

Abaixo irá perceber que já definimos suas legendas, além de usar uma label (Label3) com as instruções do que deverá ser preenchido.

No canto esquerdo veremos o “Height” e o “Width”, essas são funções onde configura a largura e altura da nossa tela, para cobrir a tela do celular toda, ambas estão selecionadas nos ícones à esquerda.



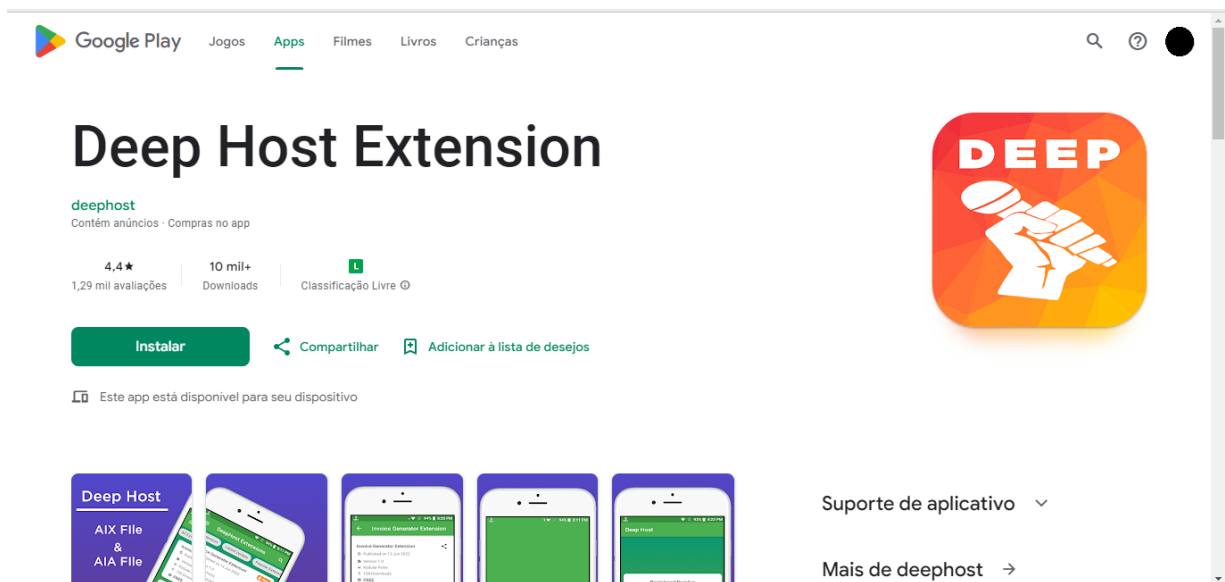
Voltando ao “Blocks” nós definiremos que o “BotãoRegistrar” irá direcionar o usuário para uma página em que criará o login e sua senha do aplicativo.



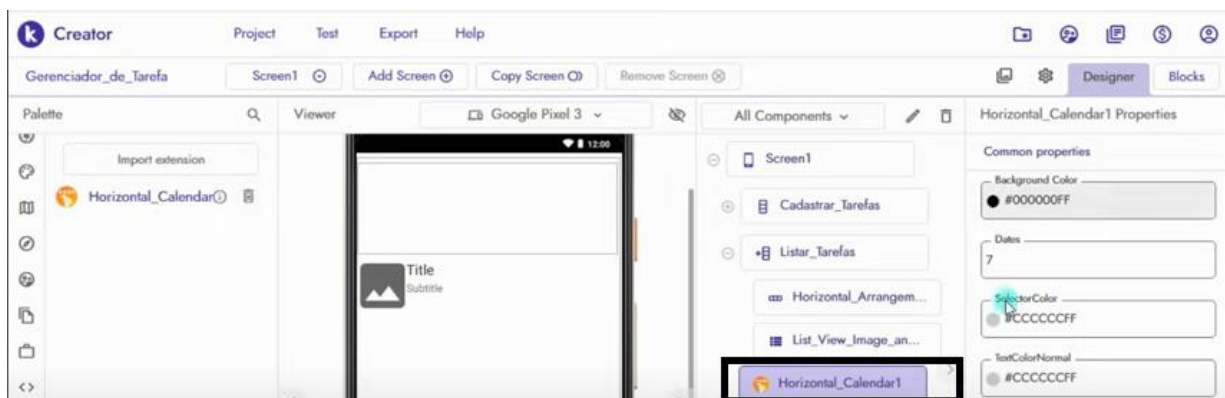
Praticamente já terminamos a tela login do aplicativo.

Calendário

Como já criamos as telas de login e registro, vamos para a mais importante o calendário. Para isso iremos fazer o mesmo processo de “Vertical Arrangement” para criarmos a tela que vai listar nossas tarefas e um “Vertical Arrangement” para cadastrar as tarefas. Iremos colocar outros elementos e instalar uma extensão chamado de “Deep Host Extension” para o calendário.

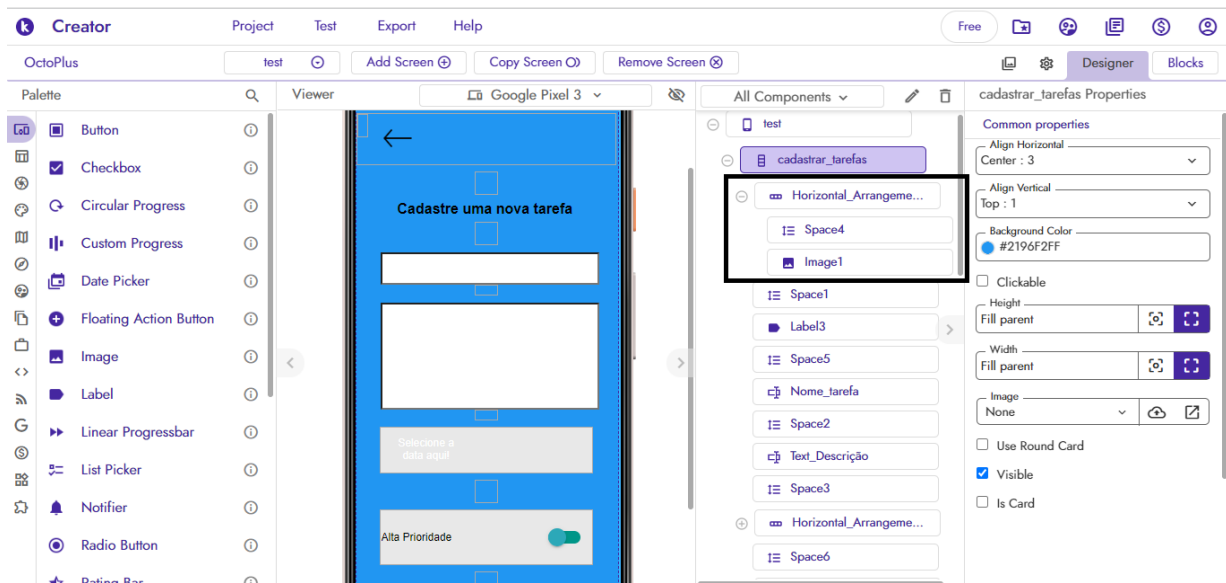


Na tela abaixo mostra que conseguimos adicionar o “Horizontal_Calendar”. Além disso colocamos em nossa tela de “Listar_Tarefas” um “Horizontal_Arragement” e um “List_View”.



Agora vamos ver como vai ficar a nossa tela “Cadastrar_Tarefas”

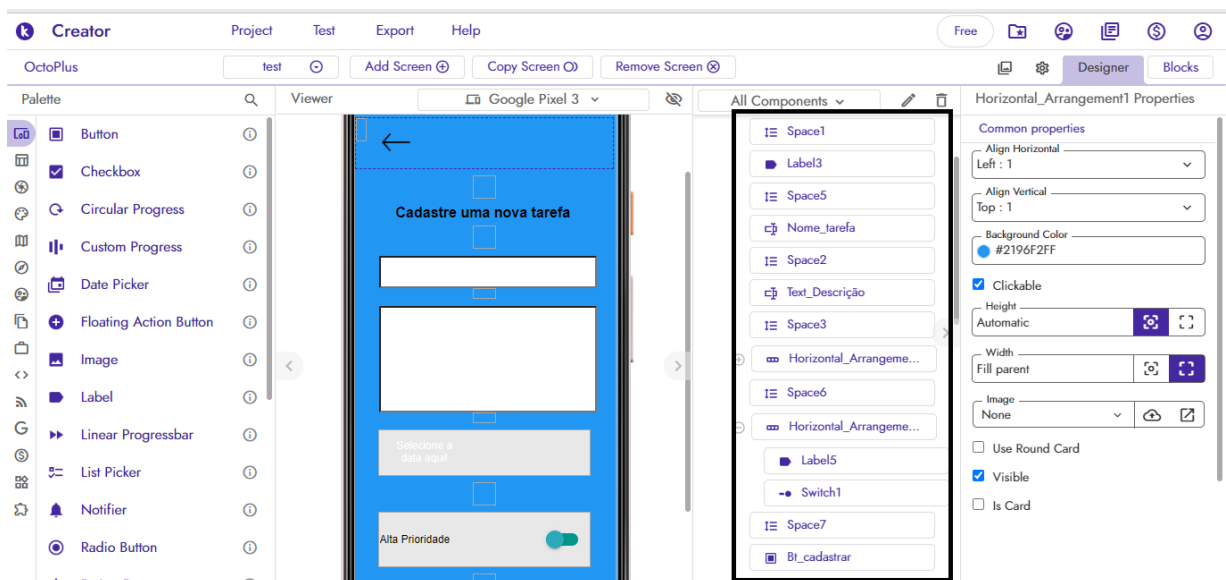
Colocamos do lado um “Horizontal_Arragement” para adicionarmos um Space e uma imagem que quando clicada irá voltar para a tela de Listar Tarefas que criamos anteriormente.



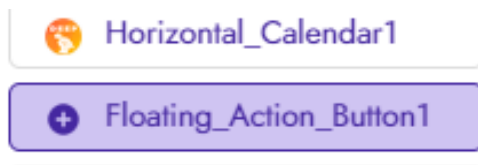
Agora focando vamos focar em outras funções. Além de ficar adicionando os “Spaces” para ficar um design, colocamos uma label com a descrição para o usuário entender para o que aquela tela serve.

Colocamos os “Text_Box” que alteramos os nomes para “Nome_Tarefa” e “Text_Descrição”.

Adicionamos dois “Horizontal_Arragament” uma com o “Data_Picker” para o usuário decidir o dia que aquela tarefa deve ser cumprida e outra para termos aquela tarefa definida ou não como “Alta Prioridade” com o “Swich1”.



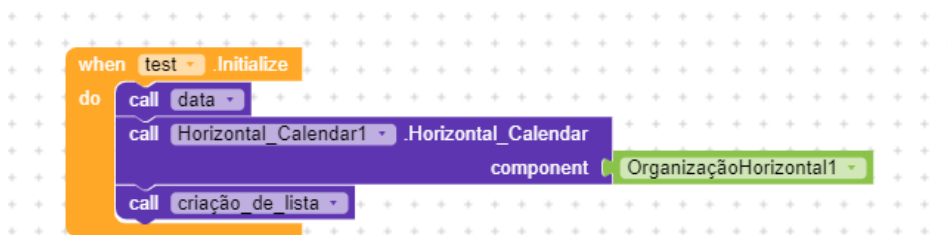
Antes de finalizarmos as telas vamos adicionar um “Floating_Action_Button” para quando o usuário quiser cadastrar suas tarefas na tela que colocaremos como tela de inicialização depois do login.



Com as duas telas prontas vamos para a parte dos “Blocks”

Blocks

Abaixo colocamos que quando a tela de login estiver completa será aberta o nosso “Horizontal_Calendar”.



Iremos colocar a função “create empty list” em “nome_tarefa”, “descrição_tarefa” e “data_tarefa”. Essa função serve para criarmos uma lista vazia em nosso aplicativo, a fim de essas listas ficarem manipuláveis, como adicionar, remover, itens a ela ou até mesmo verificar o tamanho dessa lista.



Agora iremos adicionar o que o botão “Bt_Cadastrar” irá fazer, no nosso caso esse botão serve para cadastrar as tarefas diárias do usuário, como o nome da tarefa, a sua descrição do que deve ser feito e a data que deve ser cumprida. Além disso na tela “Cadastrar_Tarefas”

tem a função de Alta Prioridade caso a tarefa seja muito importante, caso não seja ela terá uma “Prioridade_Normal”, quando tudo estiver preenchido é chamada a função de “criação_de_lista”.

```

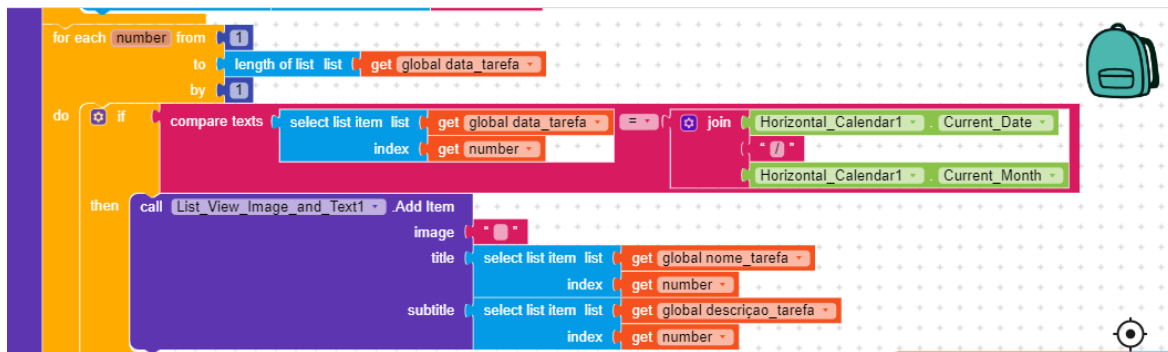
when Bt_cadastrar .Click
do
  initialize local tarefas to
  in
    add items to list list
    item (
      make a list (
        make a list "nome"
        Nome_tarefa .Text
      )
      make a list "Descrição"
      Text_Descrição .Text
      make a list "Data"
      Date_Picker1 .Text
      "Prioridade"
      if Switch1 .Enabled = true
      then "Prioridade: alta"
      else "Prioridade: Normal"
    )
  call criação_de_lista
  
```

A “criação_de_lista” terá com certeza um espaço maior no “Blocks”.

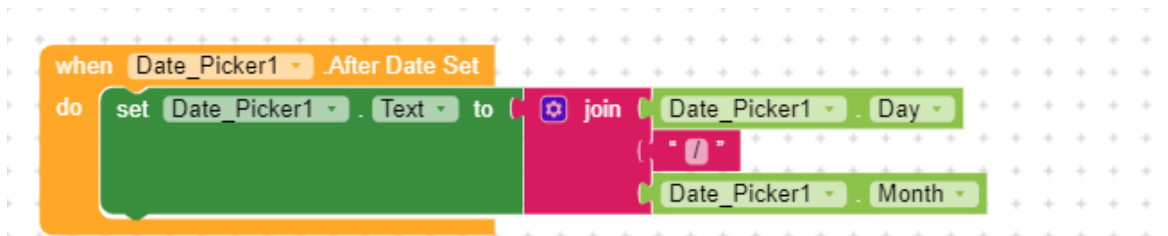
Vamos colocar as funções de “global_nome_tarefa”, “global_descrição_tarefa”, “global_data_tarefa” dentro delas. Essas funções estarão atreladas com o nosso Horizontal_Calendar.

```

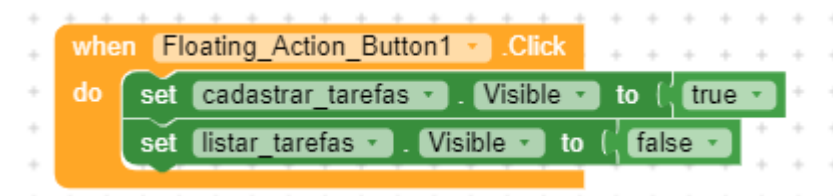
to criação_de_lista
do
  for each item in list
  do
    add items to list list
    item
    look up in pairs key "nome"
    pairs
    get item
    notFound "not found"
    add items to list list
    item
    look up in pairs key "Descrição"
    pairs
    get item
    notFound "not found"
    add items to list list
    item
    look up in pairs key "Data"
    pairs
    get item
    notFound "not found"
  for each number from 1
  to length of list list
  by 1
  compare texts
  select list item list
  index
  get global data_tarefa
  =
  join
  Horizontal_Calendar1 .Current_Date
  
```



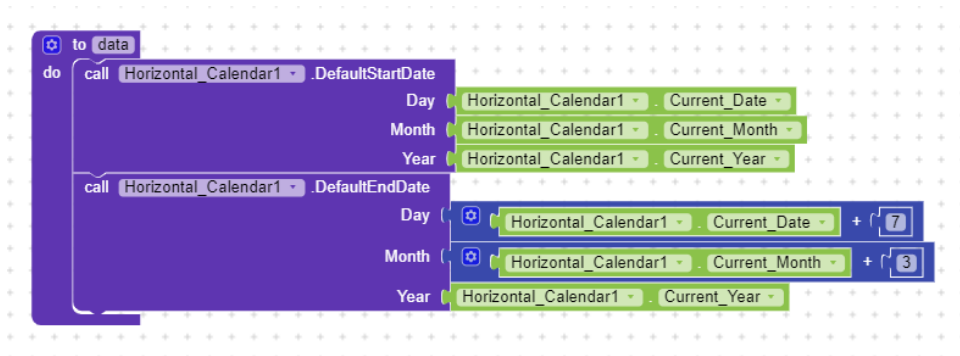
O “Data_Picker” é a função que permitirá que o usuário de selecionar a data da tarefa, por isso já deixaremos que essa função será organizada por dia e mês como vemos abaixo.



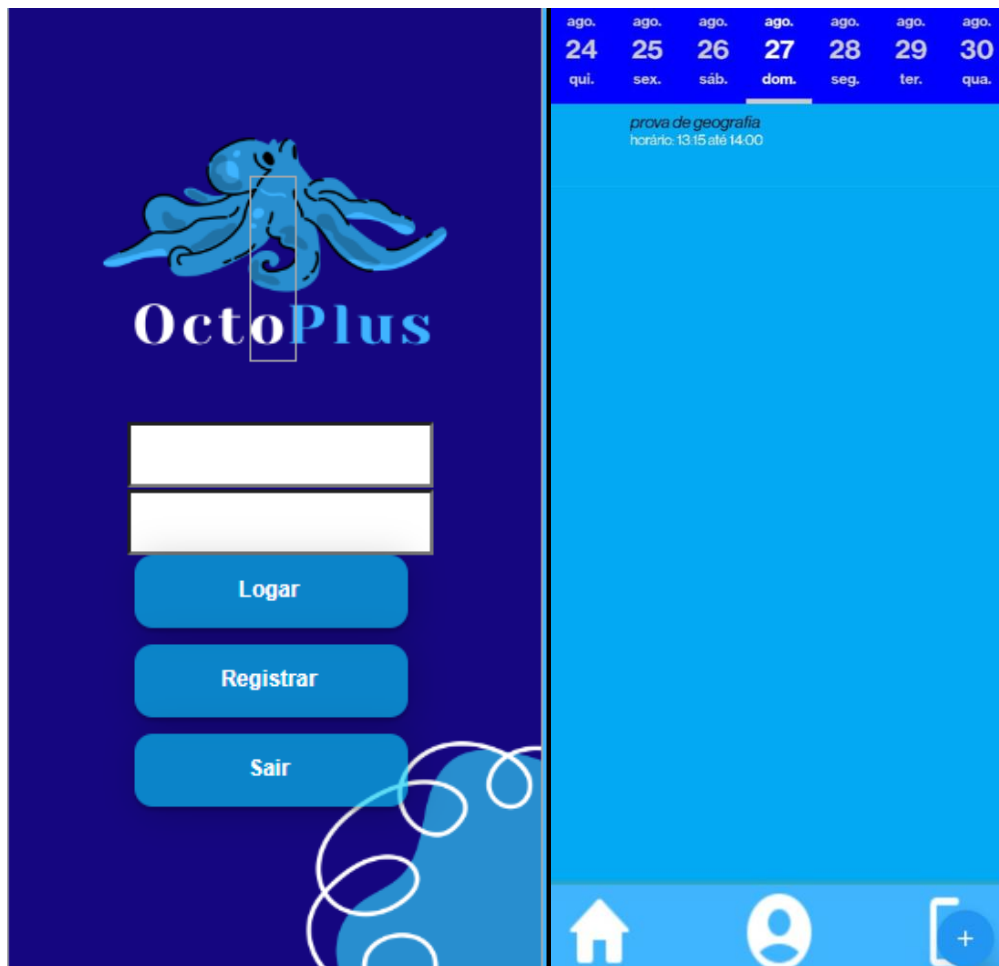
Com o Aplicativo quase pronto vamos definir o que o “Floatin_Action_Button” irá fazer como mencionarmos anteriormente. Quando esse clicado irá abrir a tela “Cadastrar_Tarefas” que está sendo definida como “True” logo abaixo.



Vamos definir o que o “data” irá fazer, como sabemos essa função irá usar o “Horizontal_Calendar” que vamos definindo em “Day”, “Month” e “Year”. Ou seja, dia, mês e ano.



Agora é só decorar de acordo com o seu gosto o design do aplicativo, o nosso aplicativo ficou assim:



Essa foi a base do nosso aplicativo OctoPlus, como viram desenvolvemos a tela de “Login”, “Registro”, “ Listar tarefas” e “Cadastrar Tarefas”.