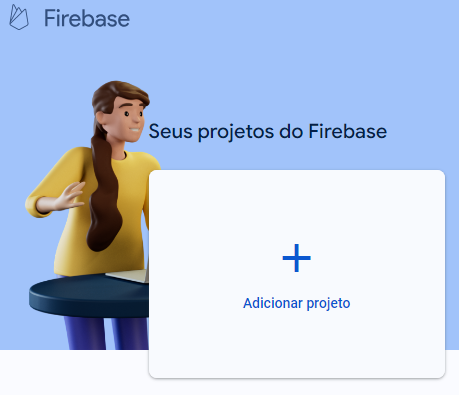
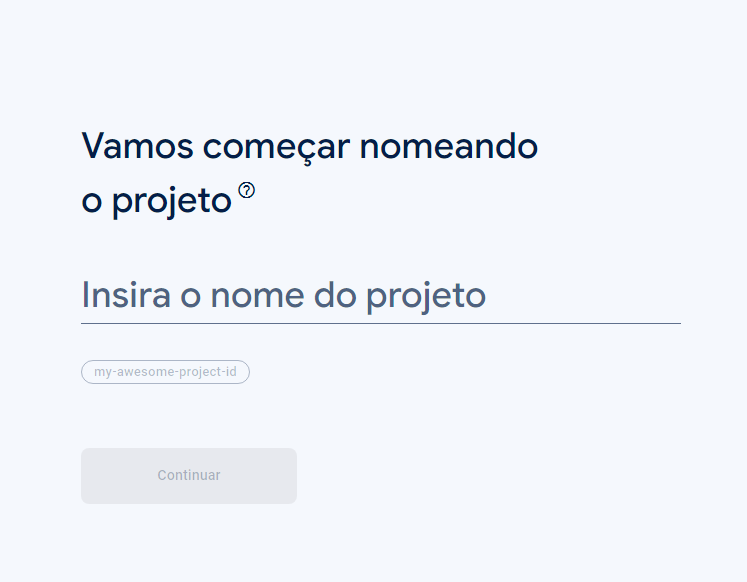
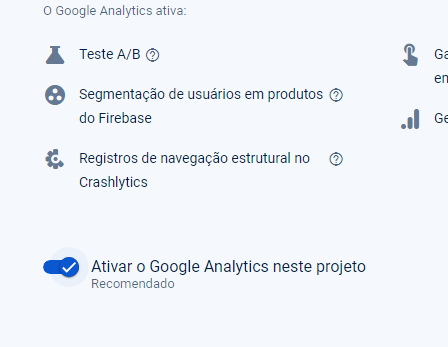
Vamos acessar o firebase: <https://firebase.google.com/>

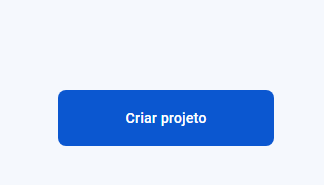
Depois vamos clicar em:

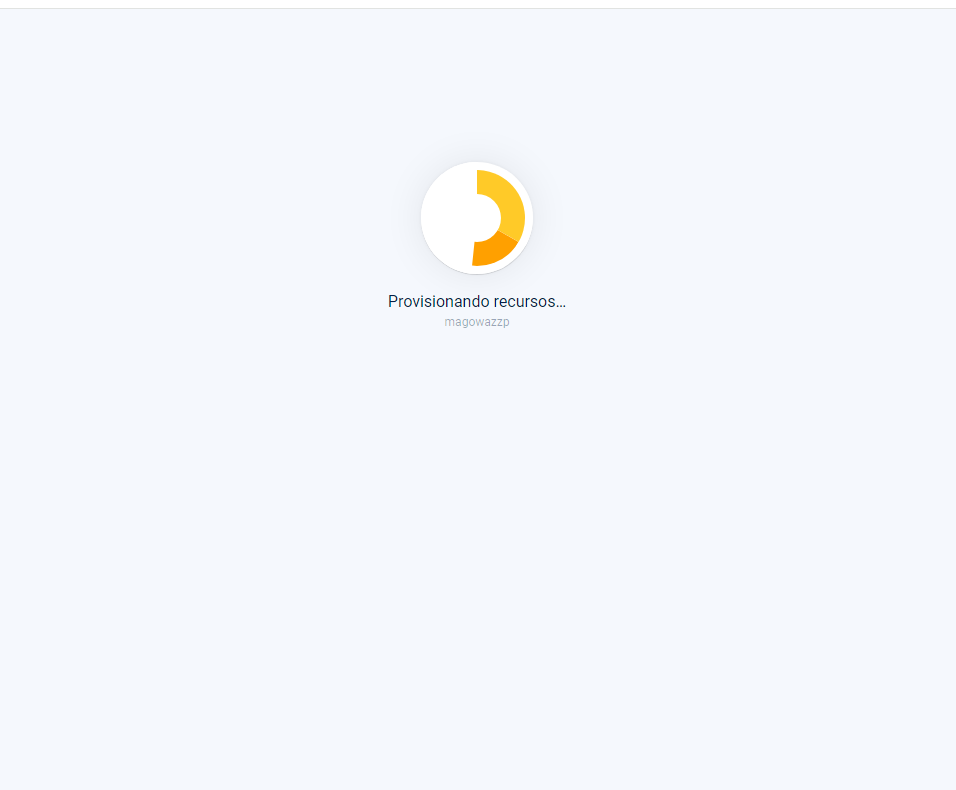


Insira o nome para o projeto:

Retire a opção:



E clique em criar projeto



Na tela do projeto criado vamos clicar em projeto web:

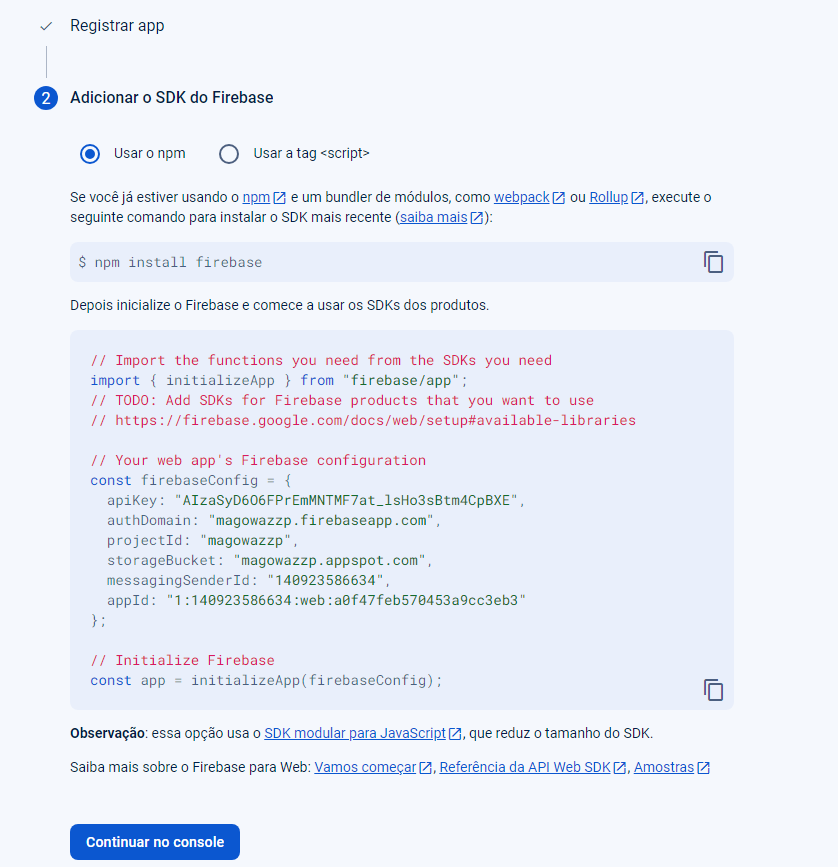


Vamos colocar o nome do app:

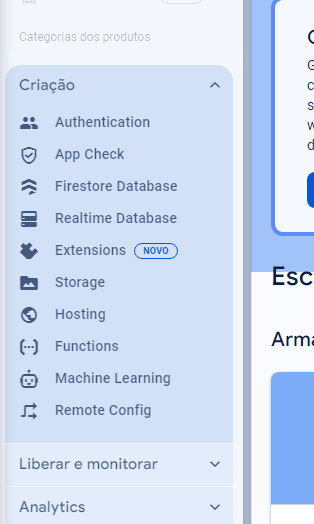


E clicar em registrar app

Depois vamos clicar em continuar no console

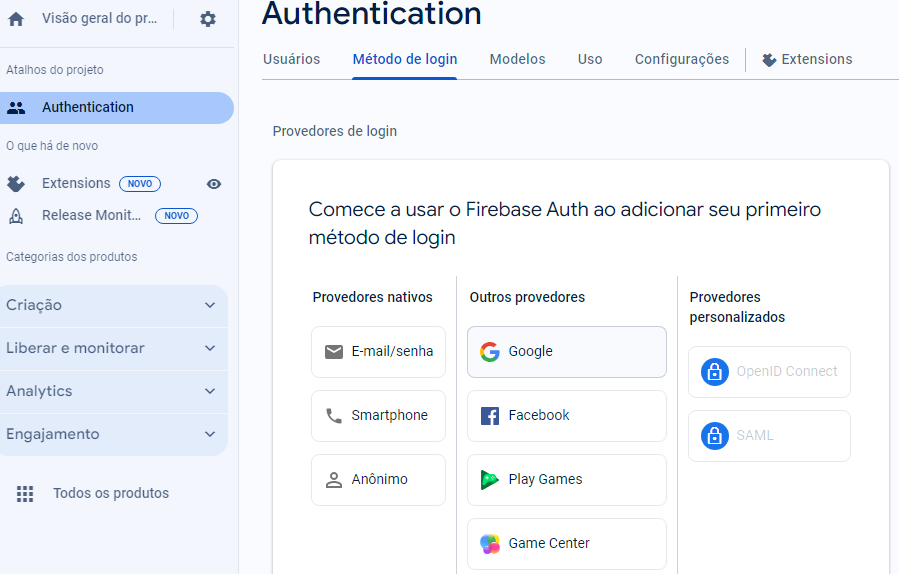


Em criação vamos criar a nossa autenticação que sera por email:

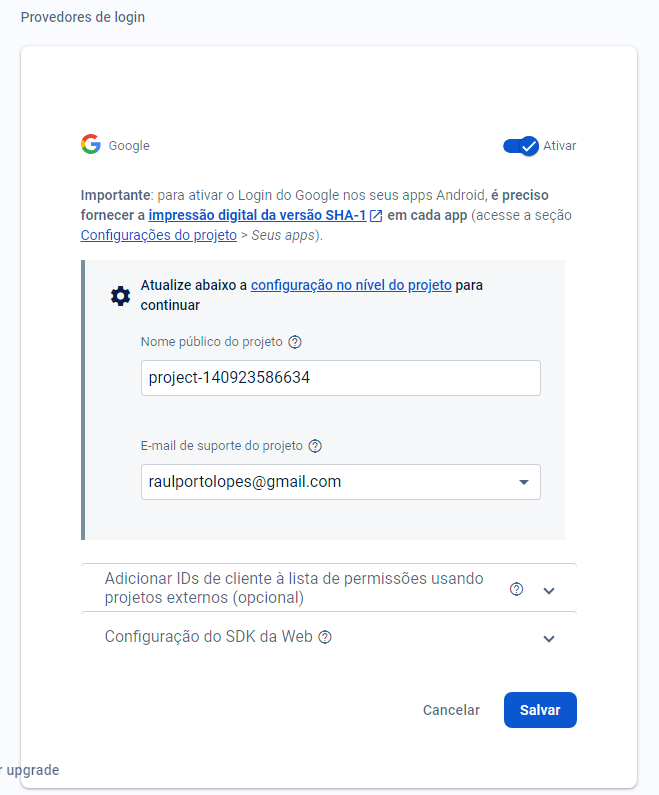


Clicar no botão primeiros passos

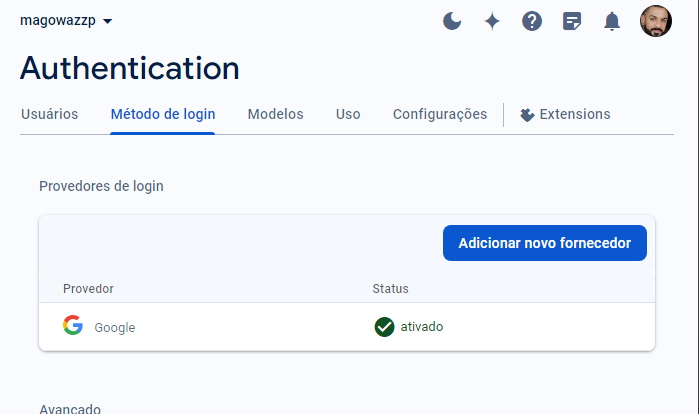
Depois selecione a opção google



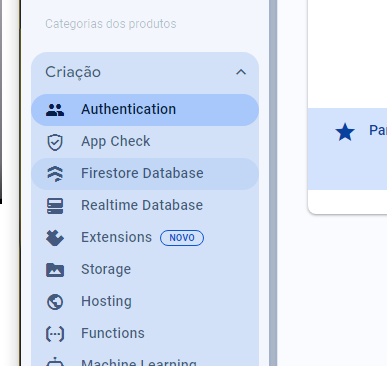
Ligue o botão ativar e coloque o seu email logado e depois clique em salvar

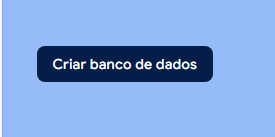


Depois disso foi criado nossa autenticação com o google:



Agora vamos criar o nosso banco de dados em autenticação > firestore database



Clicar em 

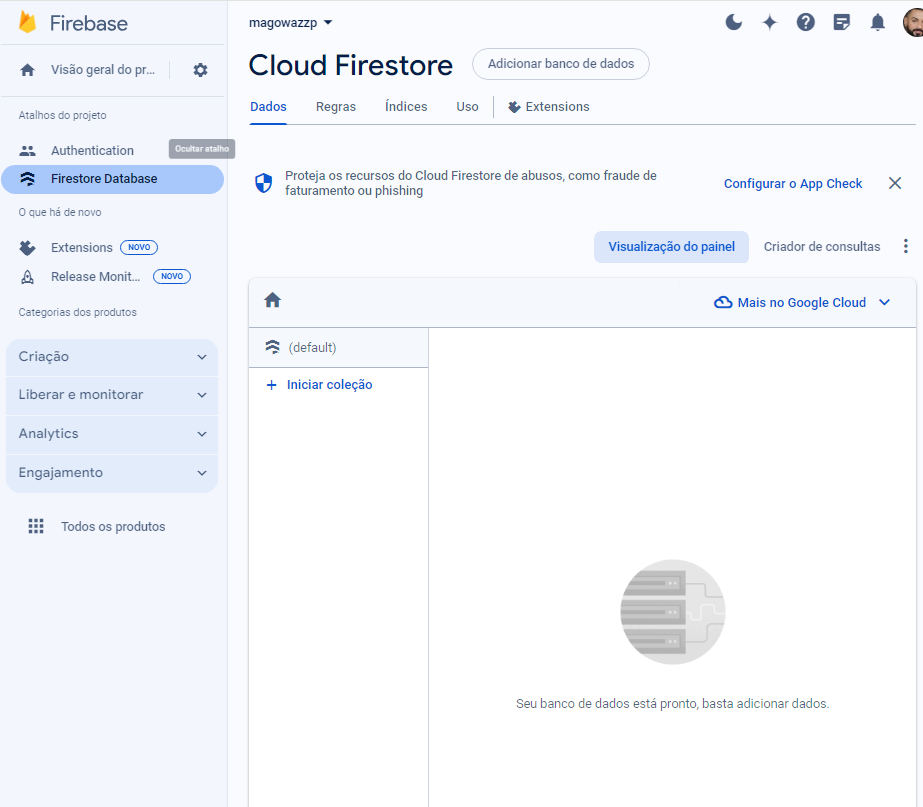
Vamos selecionar o local do banco de dados depois clicar em próximo



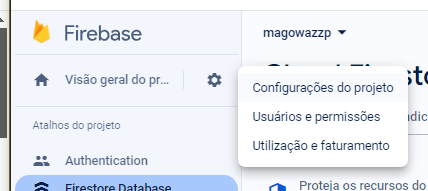
Vamos selecionar a opção iniciar no mode de teste e depois em ativar



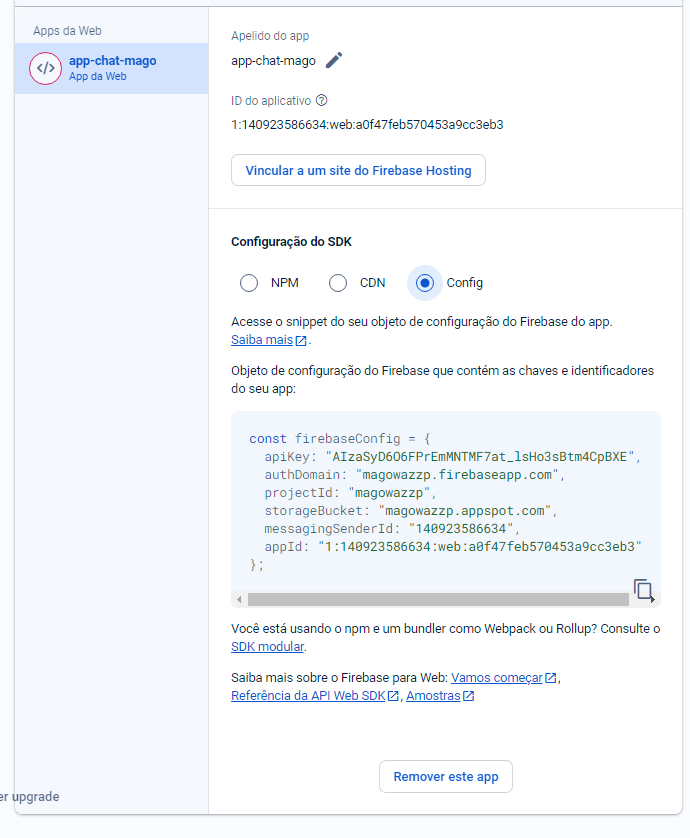
Depois disso o banco sera criado



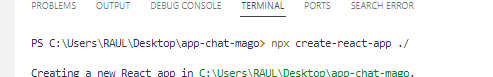
Agora vamos nas configurações do projeto



Agora vamos copiar o código criado para usarmos depois em nosso código:

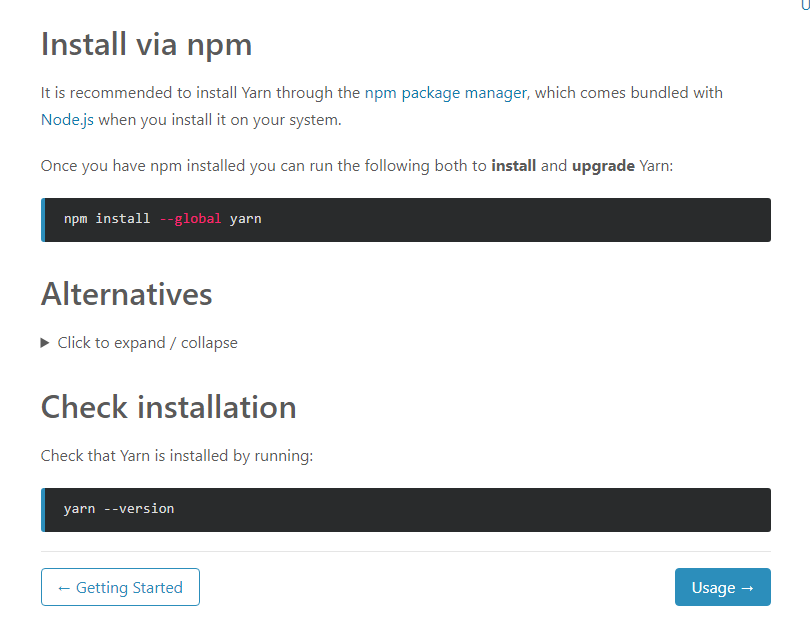


Agora vamos para o visual studio e abrir a nossa pasta criada em branco

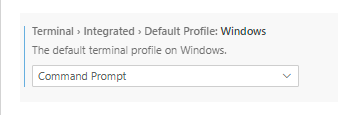


Obs: o ./ significa que vai criar dentro da nossa pasta

Agora vamos instalar o yarn



Alterar a configuração nas configurações do vscode



Agora vamos instalar as bibliotecas

Yarn add

styled-components (para estilização)

react-icons (icones)

firebase (bando de dados)

better-react-spinkit (animações)

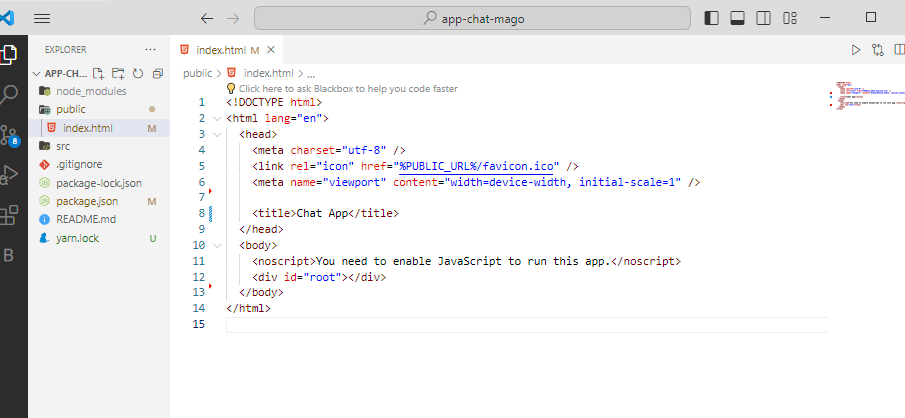
react-firebase-hooks (facilitar na consulta dos dados)

email-validator (validar os email para uso sera valido)

yarn add styled-components react-icons firebase better-react-spinkit react-firebase-hooks email-validator

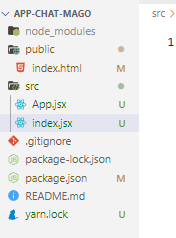
agora na pasta public vamos deixar apenas o index.html

e na parte do código vamos limpar todos os comentarios



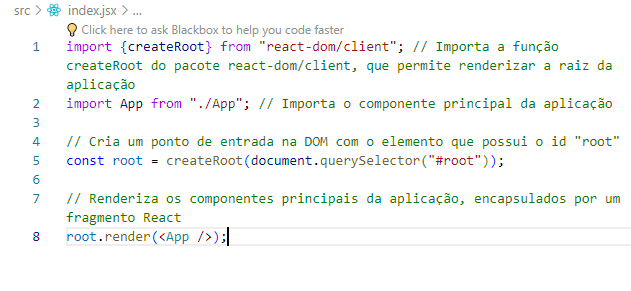
Vamos excluir a pasta src e criar ela do 0

Agora vamos criar os arquivos

 App.jsx e index.jsx

E preencher eles

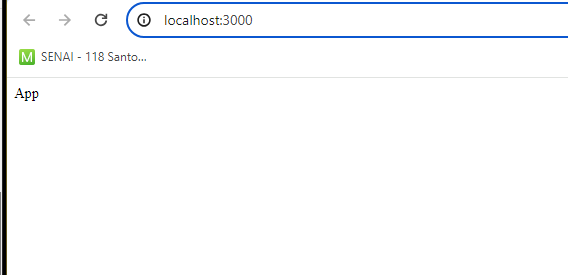
Index.jsx



App.jsx

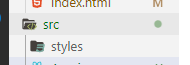


Resultado

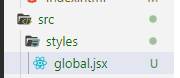


Configuração dos Estilos Globais

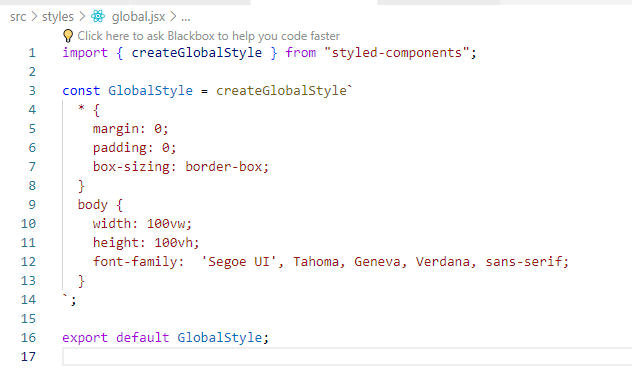
Dentro da pastas src criar a pasta styles



E vamos criar o global.jsx

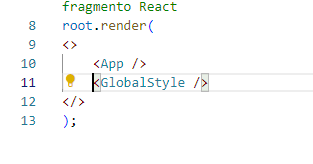


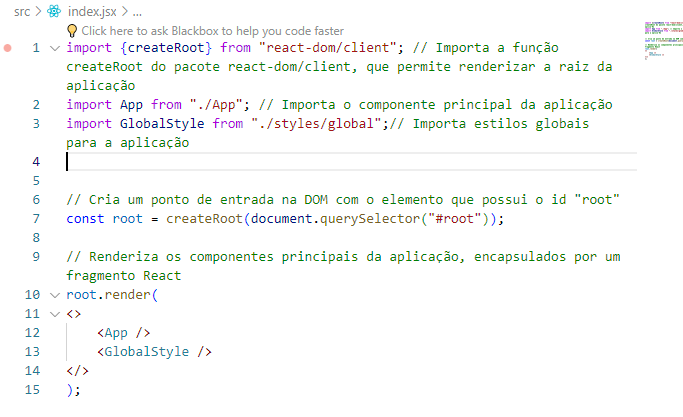
E dentro do global.jsx



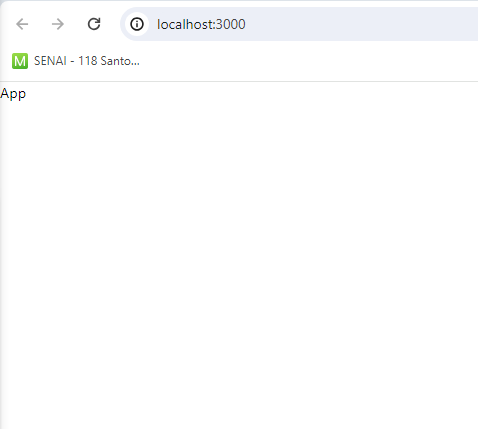
E vamos adicionar o GlobalStyle

Dentro do index.jsx





E conferir que a fonte já mudou:



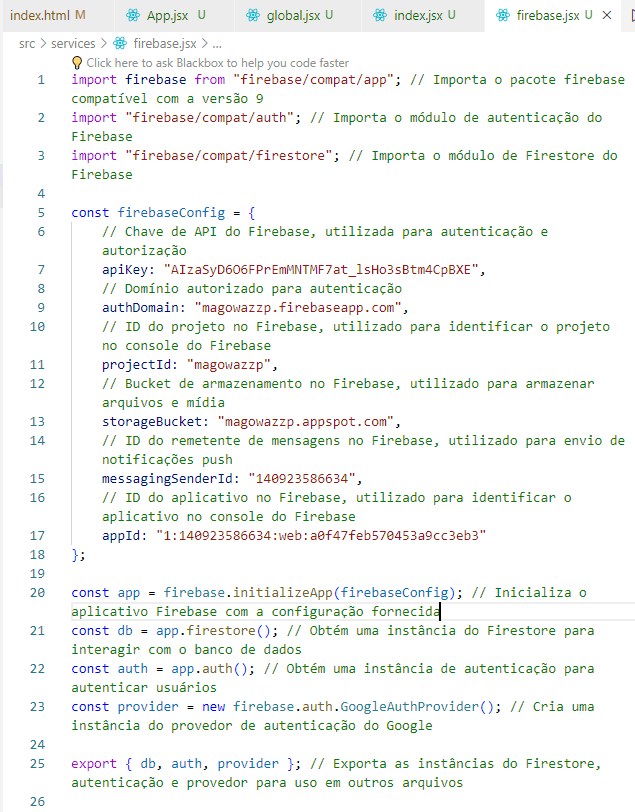
Próximo passo vamos configurar o nosso firebase

Dentro do src criar uma nova pasta chama services

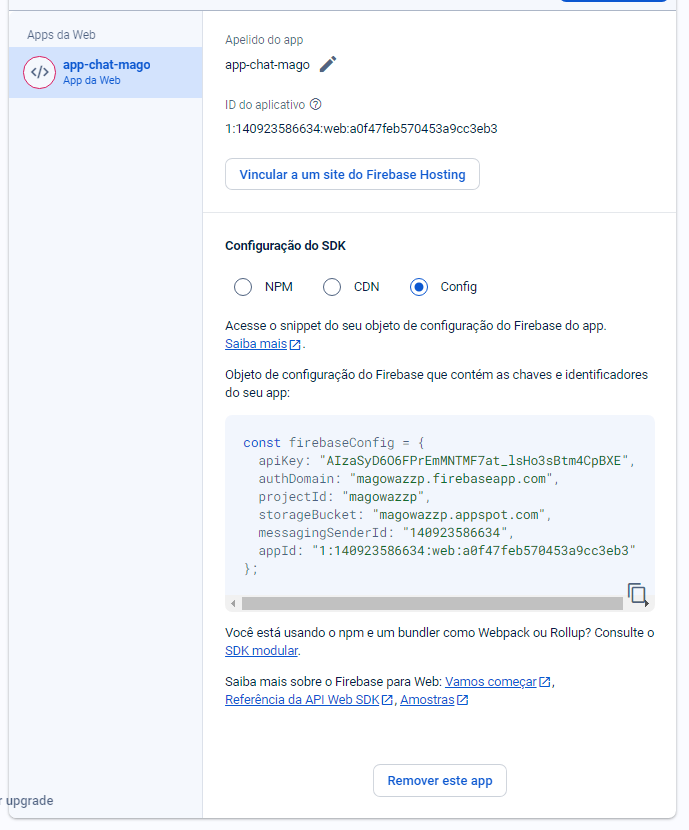


E dentro da pasta services criar um arquivo firebase.jsx



E dentro do firebase.jsx 

O trecho em vermelho esta dentro do firebase que foi criado inicialmente, como mostra abaixo



Agora vamos criar a tela de login

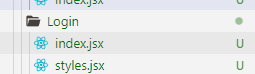
Criar a pasta componentes dentro do src



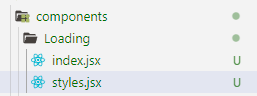
E uma nova pasta de Login dentro do componentes



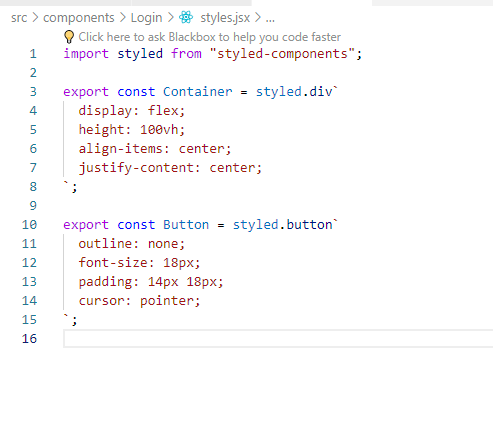
E agora criar um index e um styles.jsx



E depois repetir o processo criando uma pasta Loading

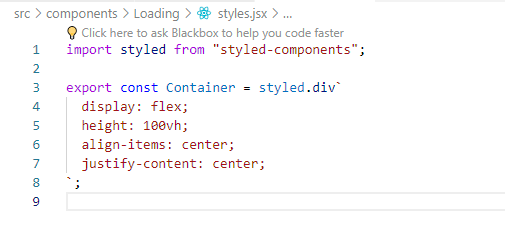


Agora dentro do LOGIN style.jsx:

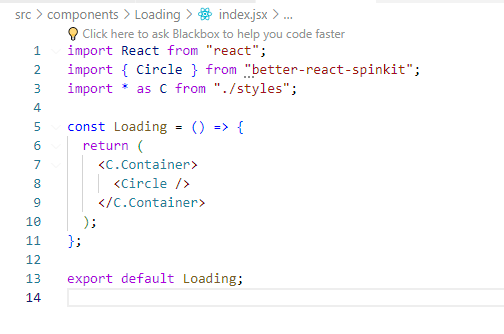


Agora dentro do LOGIN index.jsx:

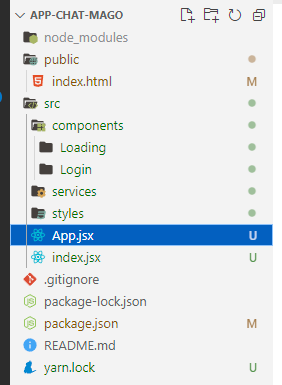
Agora no LOADING styles.jsx



Agora no LOADING index.jsx



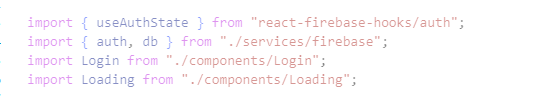
Agora vamos no App.jsx





E no primeiro import adicionar o use Effect e o Use State

E depois vamos importar



Vamos adicionar o parte do loading



Adicionar o trecho de codico com a verificação do usuário

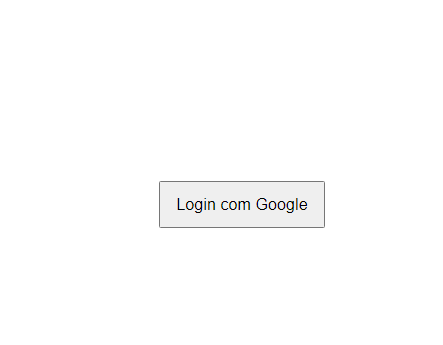


Se localizar o usuário ele faz loading

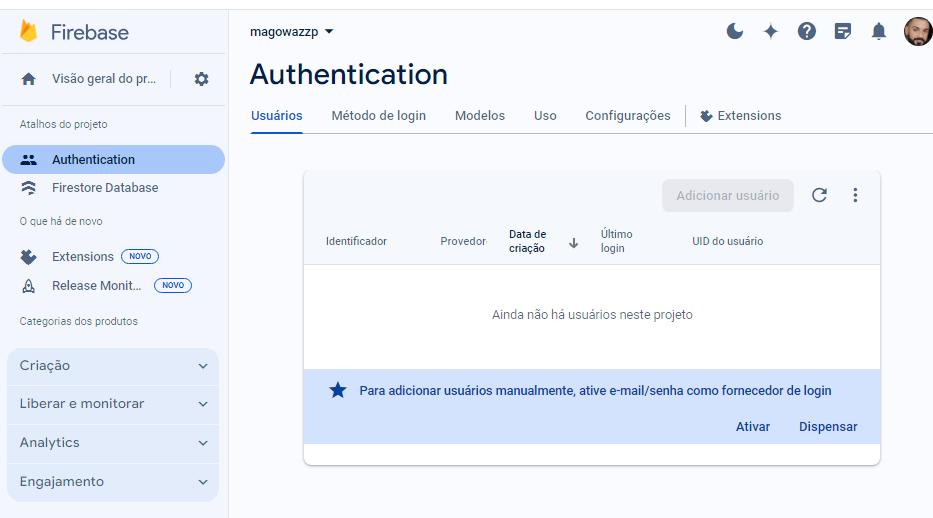
Se não faz apenas o login



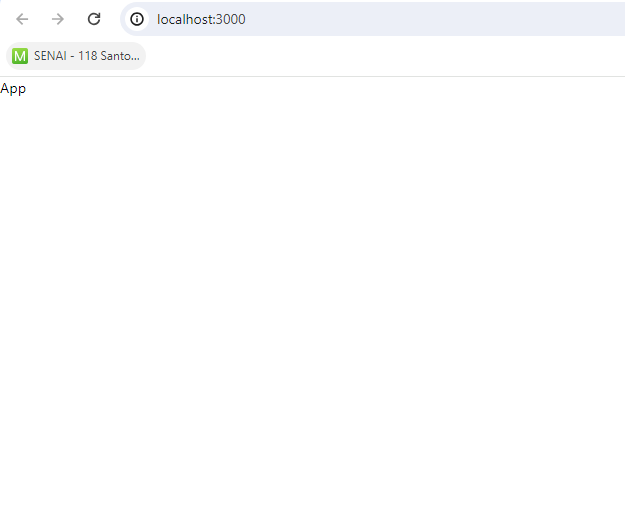
O resultado sera esse:



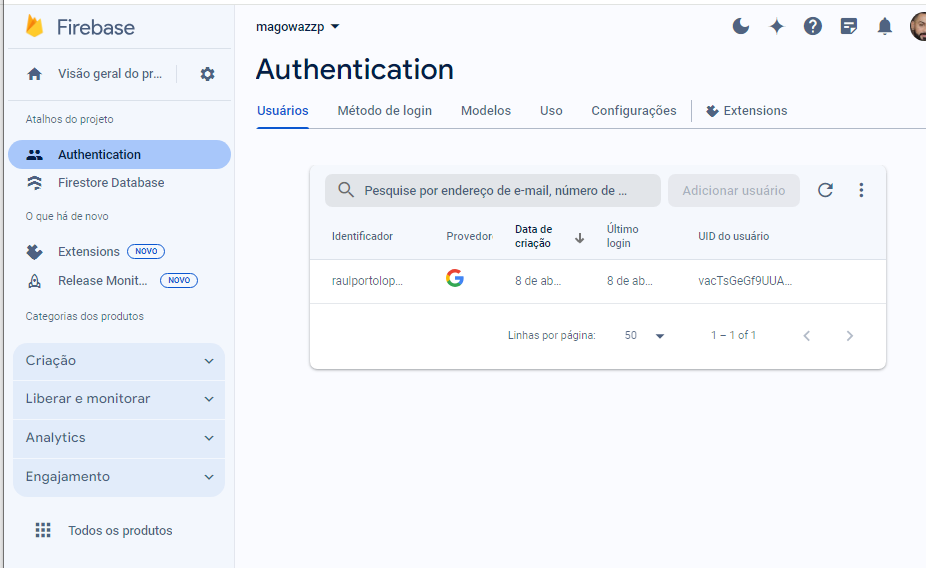
Se agente verificar no firebase, na baba autenticação não tem nenhum usuário cadastrado



Vamos clicar em login com o google e logar com o nosso email gmail



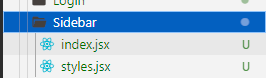
Vai acessar a pagina e depois que atualizarmos o firebase, vai mostrar quem logou



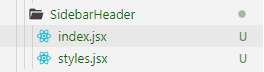
Criação do menu lateral

Dentro da pasta components vamos criar uma nova pasta chamada Sidebar e colocar um index e styles.jsx dentro

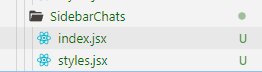




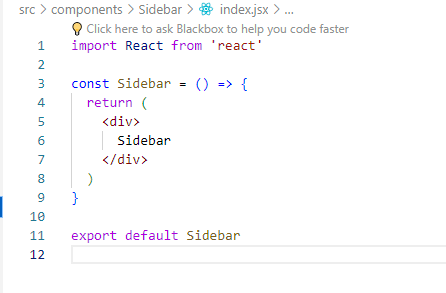
E uma pastas SidebarHeader



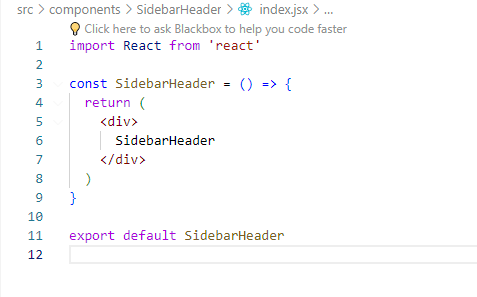
E o SidebarChats



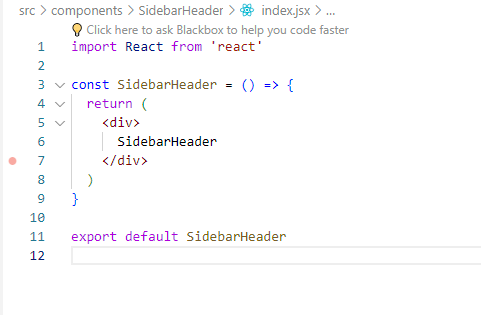
Agora dentro do Sidebar index.jsx com rafce, vamos criar a estrutura do arquivo



E repetir para SidebarChats index.jsx com rafce, vamos criar a estrutura do arquivo

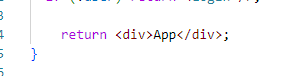


E repetir para SidebarHeader index.jsx com rafce, vamos criar a estrutura do arquivo



Agora vamos no app.jsx principal

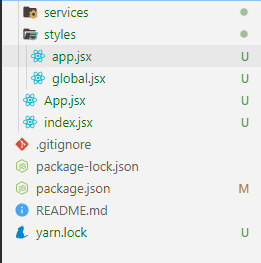
E alterar o return



para os nossos componentes criados



Vamos criar a estilização para o App

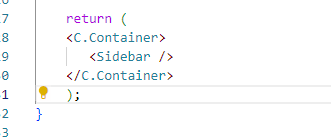


Dentro do styles aonde esta o global vamos criar o app.jsx



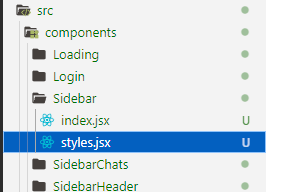
Agora dentro do app,jsx vamos adicionar o import

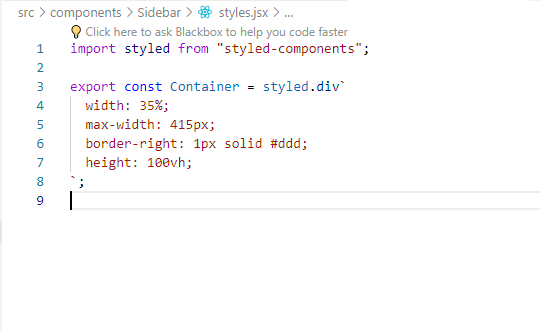




E sinalizar o container dentro do return

Agora dentro do components Sidebar vamos adicionar o Styles





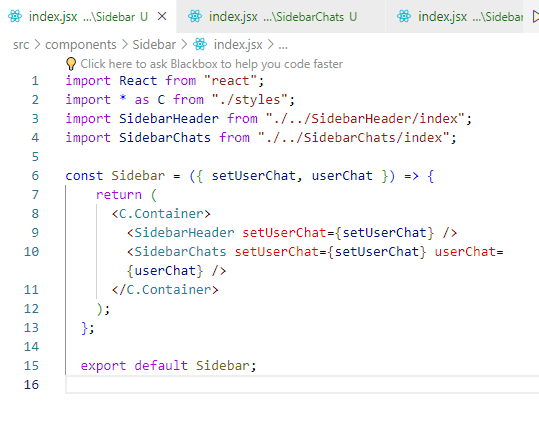
Agora no index do Sidebar



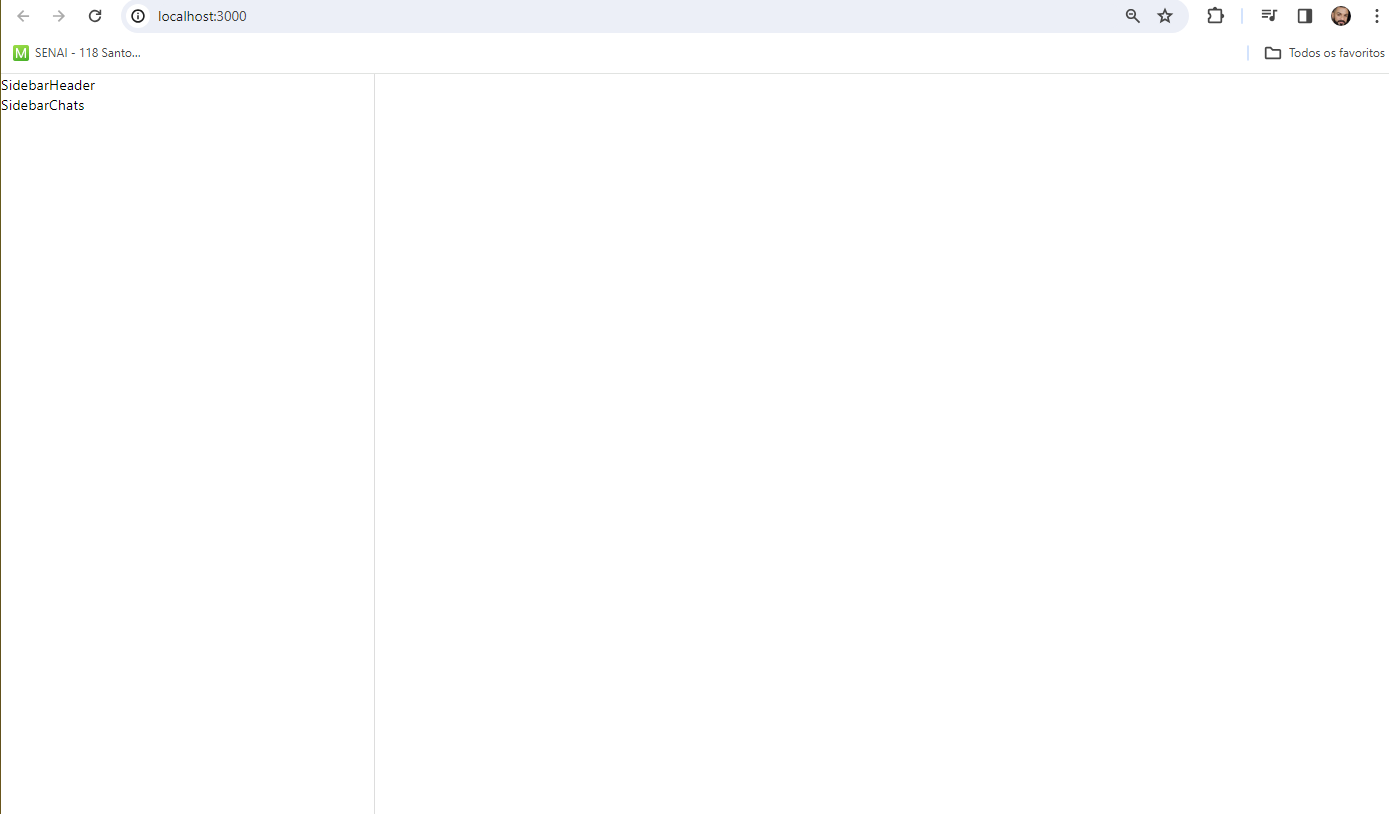
Vamos colocar as importações



E vamos ajustar o código para funcionar as importações



O resultado sera esse

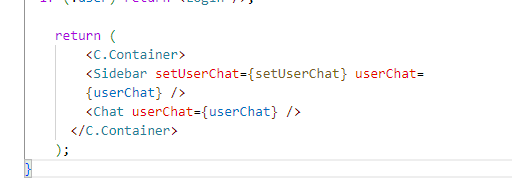


Agora vamos voltar no App.jsx principal

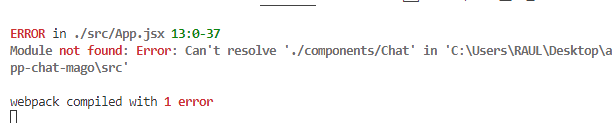
E adicionar a linha do userChat



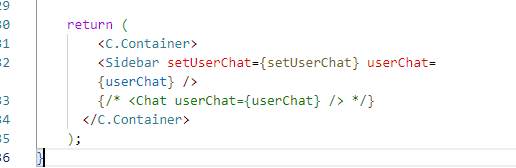
E vamos ajustar o return



Ele apresentou o erro que a pasta Chat não foi criada

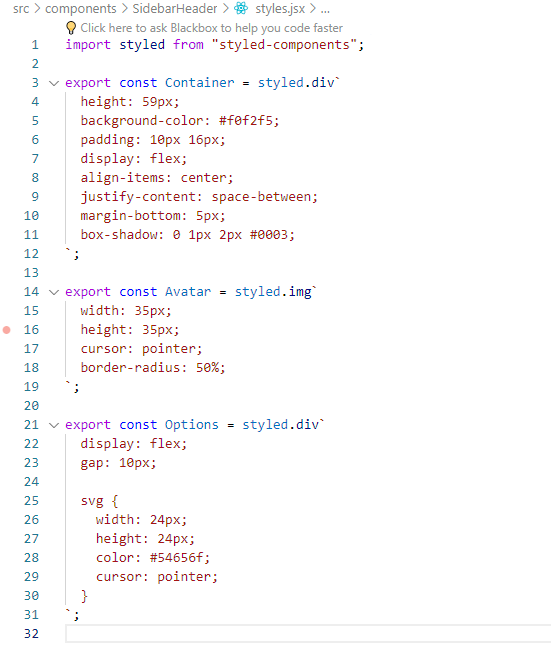


Esse erro é por que não temos ainda a pasta Chat com seu index e seu style em componentes



Vamos comentar a linha de importação do chat E o erro vai sumir

Agora vamos no SideBar header vamos adicionar o style.jsx



No SideBar index.jsx vamos adicionar os imports



Agora vamos adicionar depois do return e importar o container e o avatar



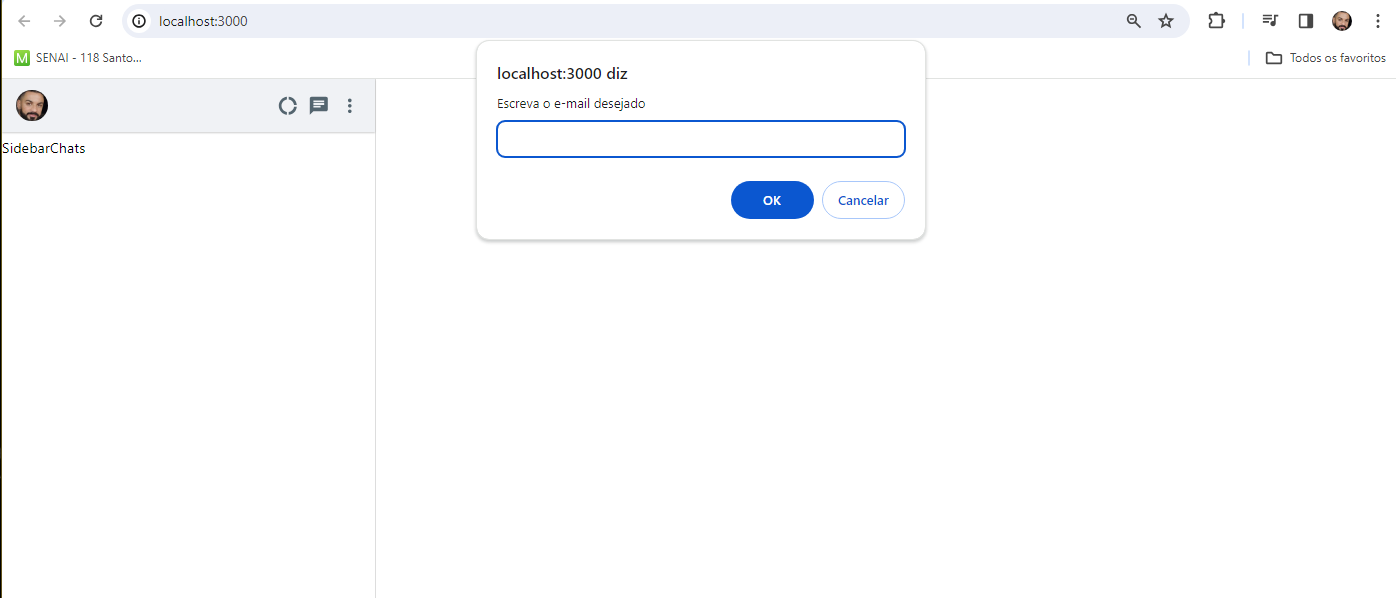
Depois disso criar a importação do useAuthState, para saber se o usuário esta logado



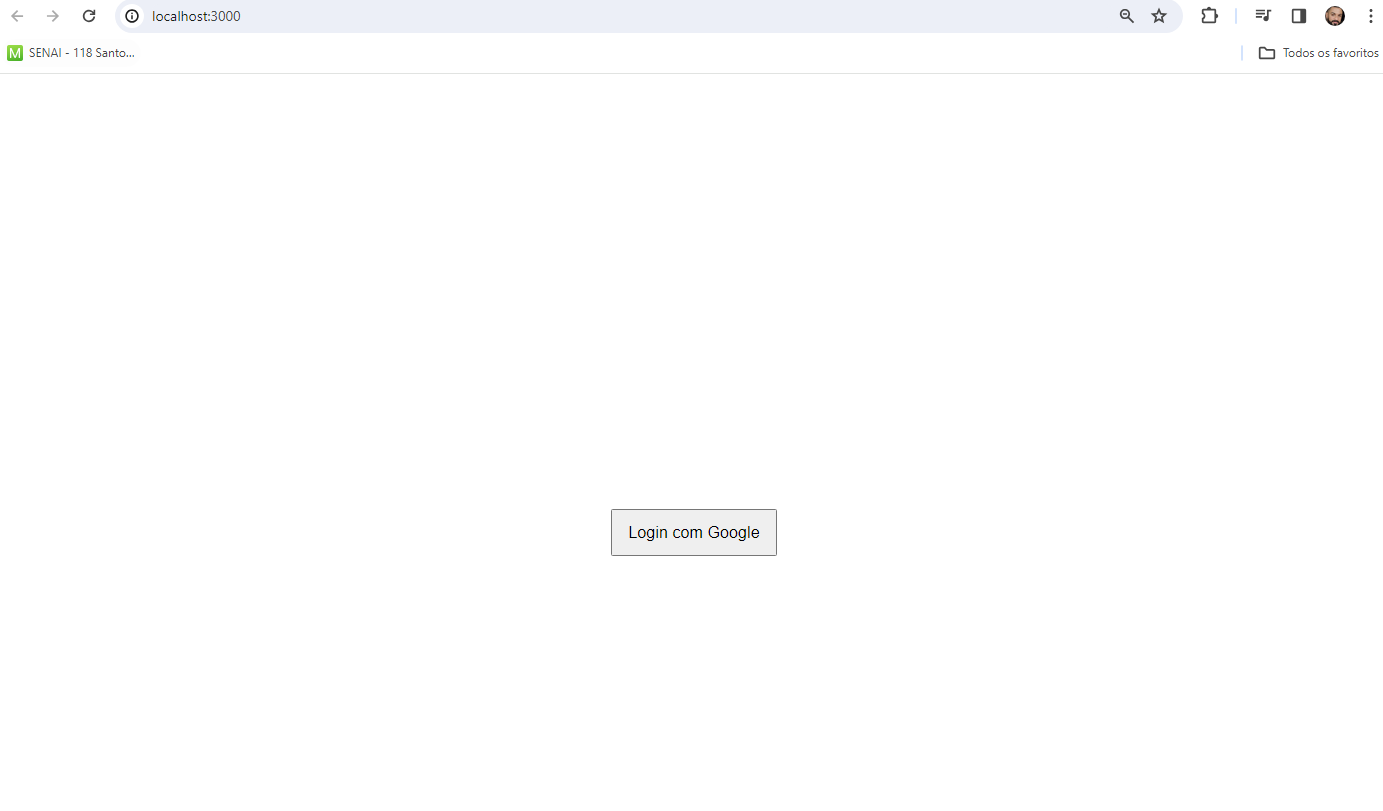
Criar a função handleCreateChat para criar um novo chat



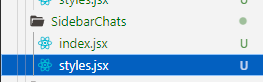
Ao final o resultado será esse:

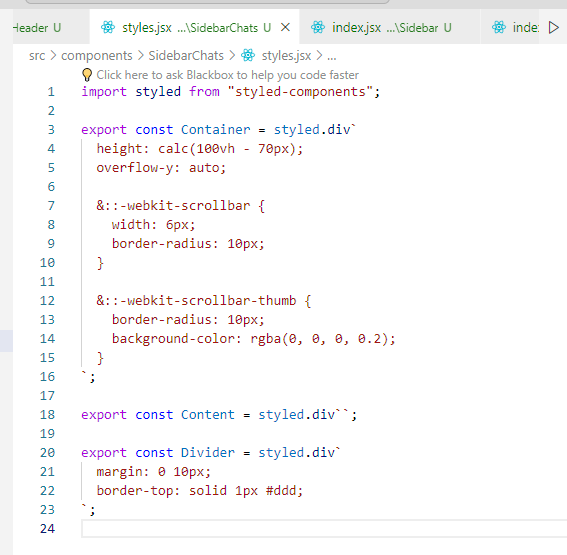


Clicando em cima do avatar ele tem que deslogar



Agora vamos no SidebarChats e criar a stylização no styles.jsx





Agora no index.jsx do SidebarChats

Vamos fazer a importação



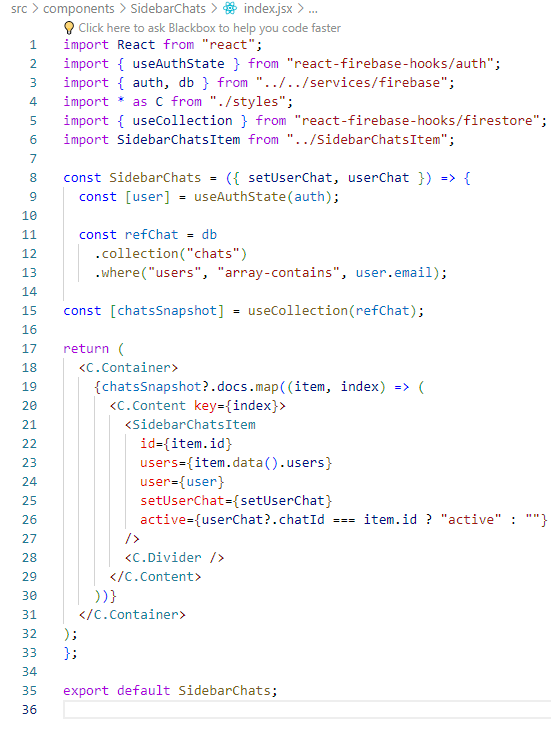
Agora declarar as variáveis para buscar as conversas e verificar as conversas em nosso banco de dados e fazer um filtro para pegar chats que contenham um email logado



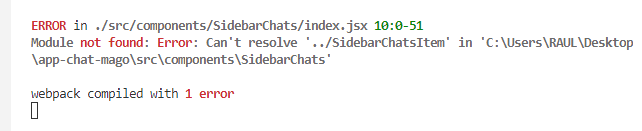
Agora noreturn container para verificar os chats criar um map para conseguir localizar cada item



Código completo

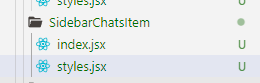


Apareceu um erro



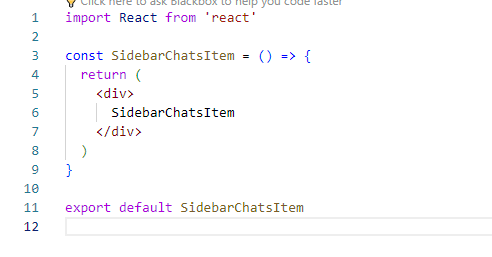
Pela a falta de criação do SidebarChatsItem

Vamos na pasta componentes e criar uma nova pasta chamada SidebarChatsItem, com um index.jsx e um styles.jsx



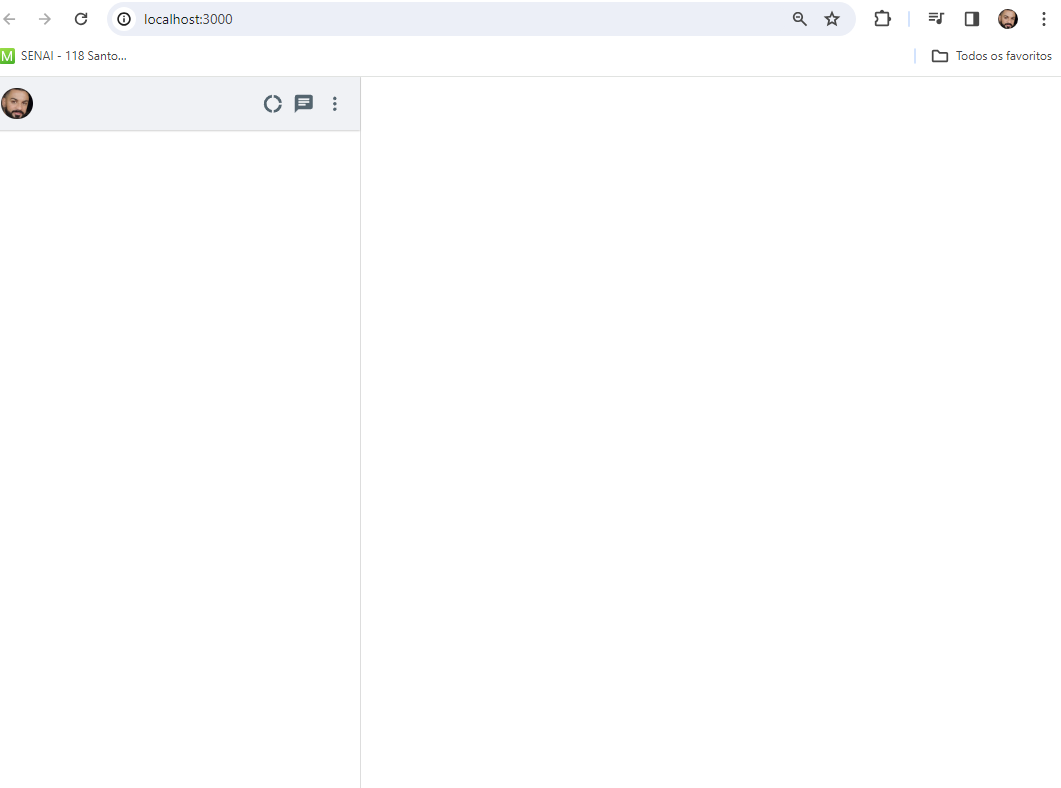
Dentro do index.jsx vamos criar um componente com o atalho 

E vamos chamalo de SidebarChatsItem



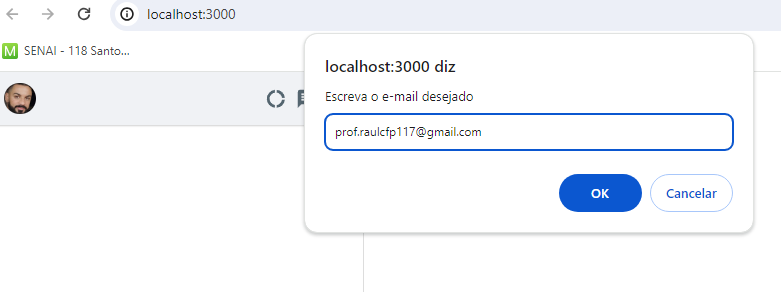
E o erro irá sumir

E o resultado ate agora será esse

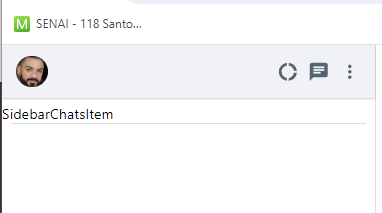


Não está listando nada por que não temos nenhuma conversa ainda

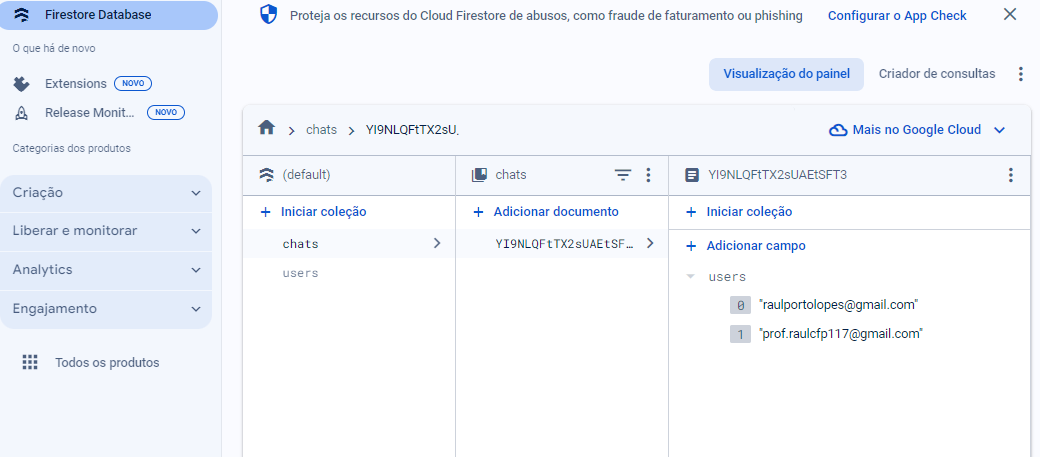
Se adicionarmos uma nova conversa



O resultado será



Agora se formos observar dentro do firebase



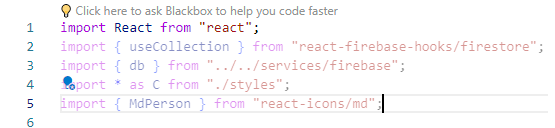
O chat já irá aparecer os 2 e-mails para a conversa

agora vamos voltar no   SidebarChatsItem styles.jsx e estilizar as conversas

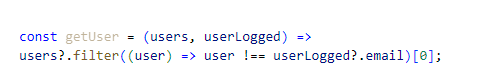


Agora vamos no index.jsx da pasta SidebarChatsItem

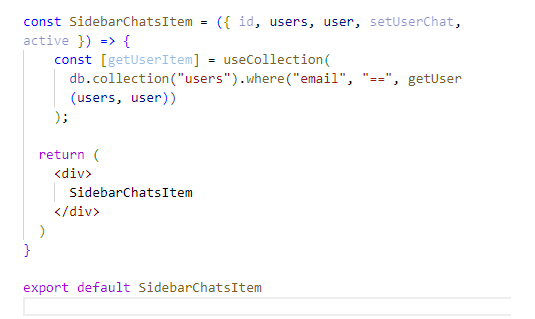
E criar as importações



Criar uma função para pegar o usuário diferente do logado



Agora fazer uma função para identificar os emails e pegar as informações dos usuários

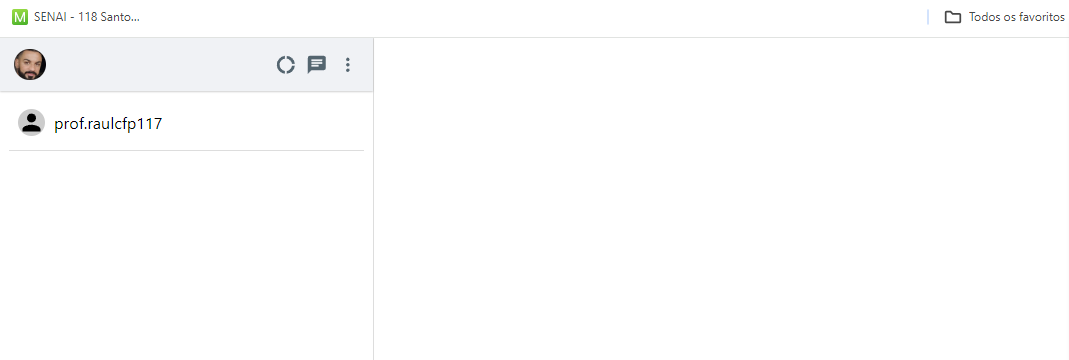


Agora vai verificar as conversas clicadas para importar os dados e mudar as cores do chat deixar meio “cinza”

Continuando o código:



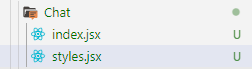
O resultado sera:



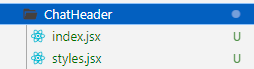
Finalizamos a criação do SIDEBAR

Agora vamos estilizar o chat

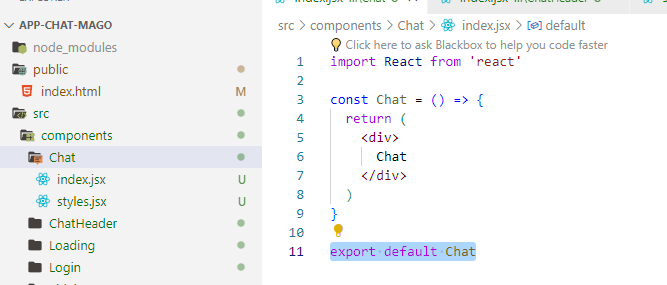
Dentro de components vamos criar a pasta Chat com o index e o styles .jsx

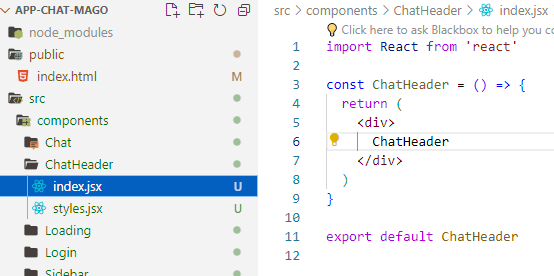


E também Dentro de components vamos criar a pasta ChatHeader com o index e o styles .jsx

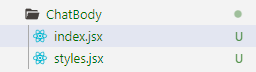


Agora vamos dentro do index do chat e criar a aerofunciton com o atalho rafce

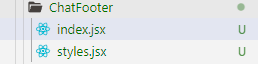


E repetimos o processo dentro da ChatHeader

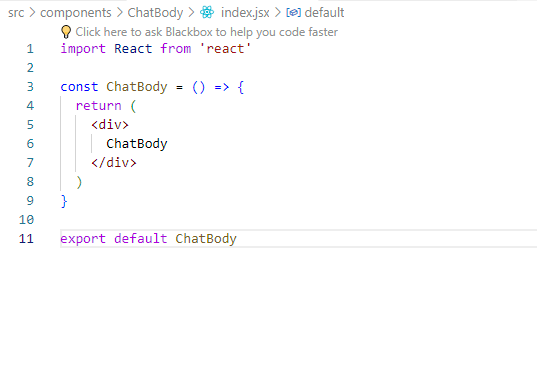
Vamos criar também uma pasta Dentro de components vamos criar a pasta ChatBody com o index e o styles .jsx

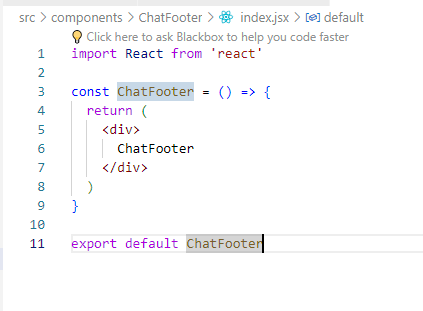


E também Dentro de components vamos criar a pasta ChatFooter com o index e o styles .jsx

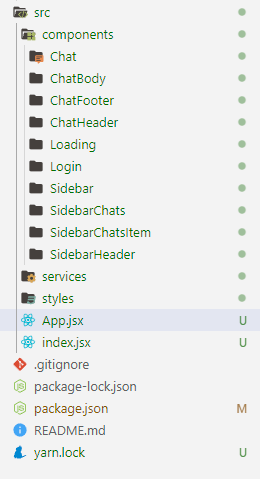


Agora vamos dentro do index do ChatBody e criar a aerofunciton com o atalho rafce

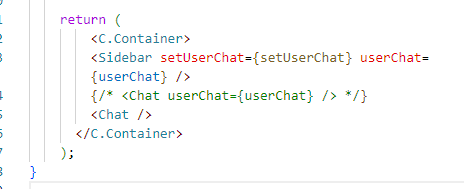
 E também dentro do index do ChatFooter e criar a aerofunciton com o atalho rafce



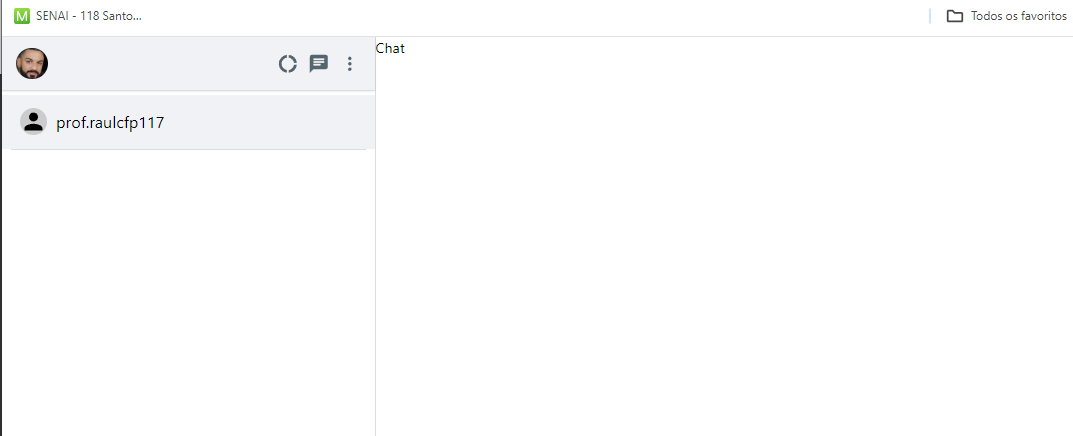
Agora vamos no App.jsx



e em baixo da sidebar importar o Chat



O resultado será



Agora na tag Chat vamos adicionar o userChat para saber em qual chat estamos



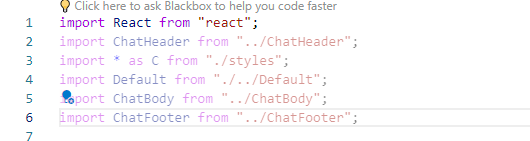
Agora vamos estilizar o chat

Dentro a pasta Chat em styles.jsx



Agora no index.jsx do chat

Vamos fazer os imports



Agora no index.jsx

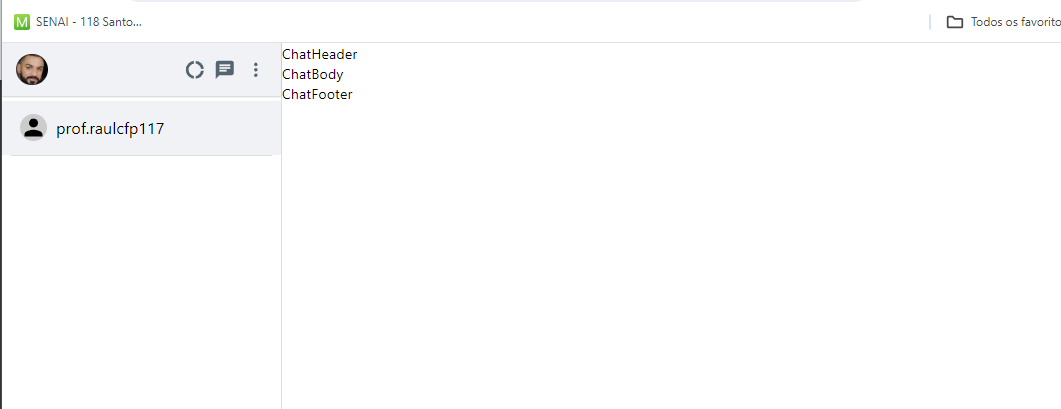
De acordo com o mapeamento feito anteriormente vamos buscar o nome e a foto e a conversar para o chat



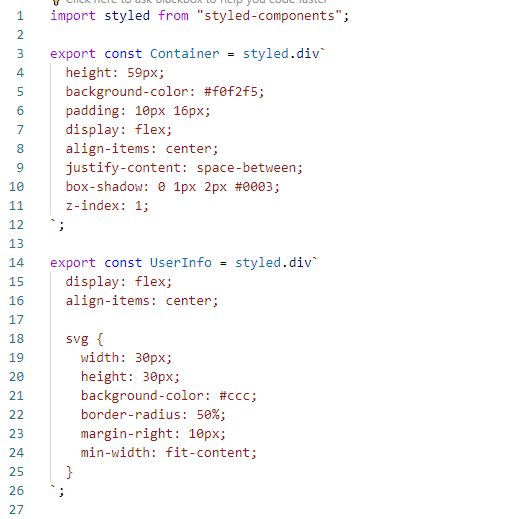
O código completo fica assim:



E o resultado sera a importação na parte do chat:

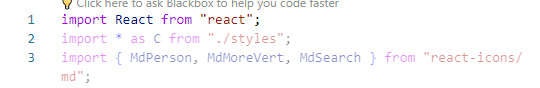


Vamos começar a configurar o ChatHeader pelo styles.jsx





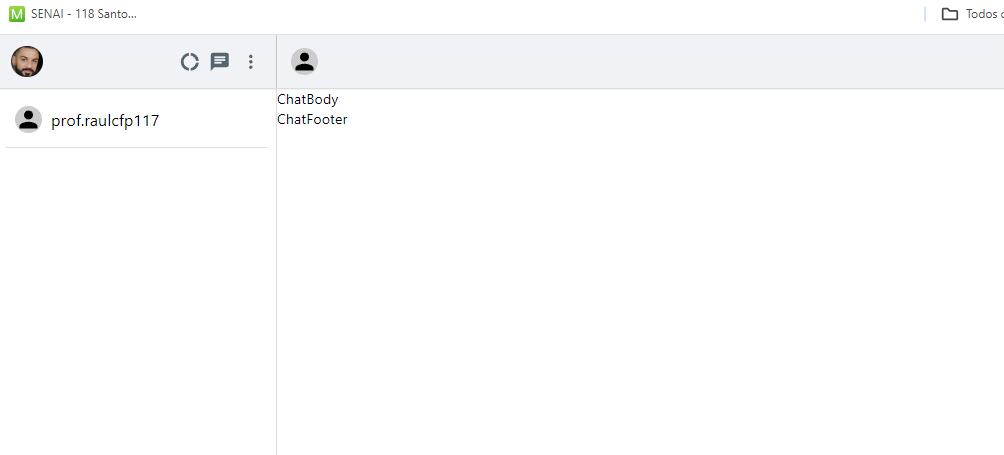
Agora no index.jsx do ChatHeader vamos fazer o import



E depois vamos criar um container para informações do usuário e carregar as informações como foto:

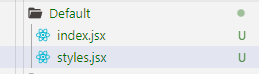


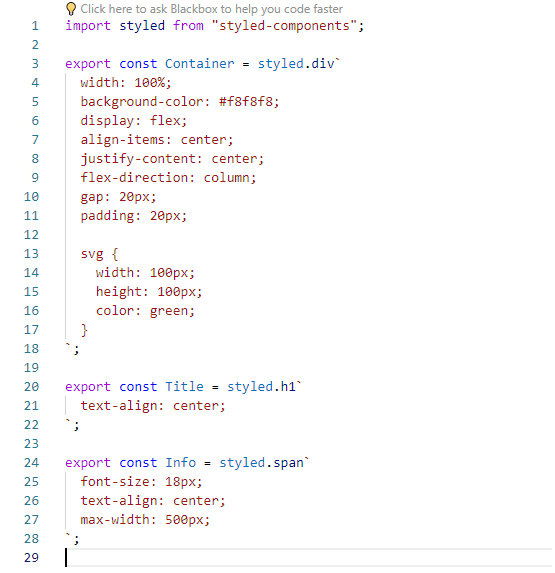
O resultado sera:

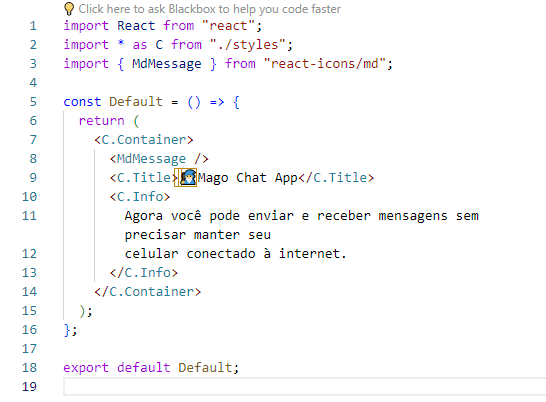


Antes de seguir com a configuração do chat

Vamos em componentes e criar uma pasta chamada Default com um index e um styles.jsx



Vamos editar primeiro o styles.jsx da pasta Default

Agora no index.jsx da pasta Default

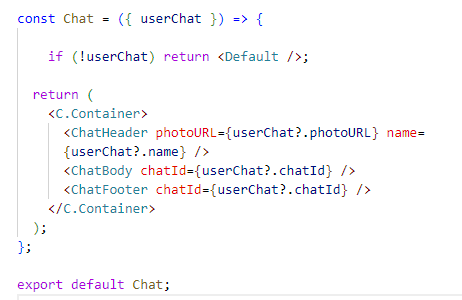
Agora vamos importar o default no chat

E fazer uma verificação se tiver se tiver um usuário vai carregar o chat se não a pagina criada

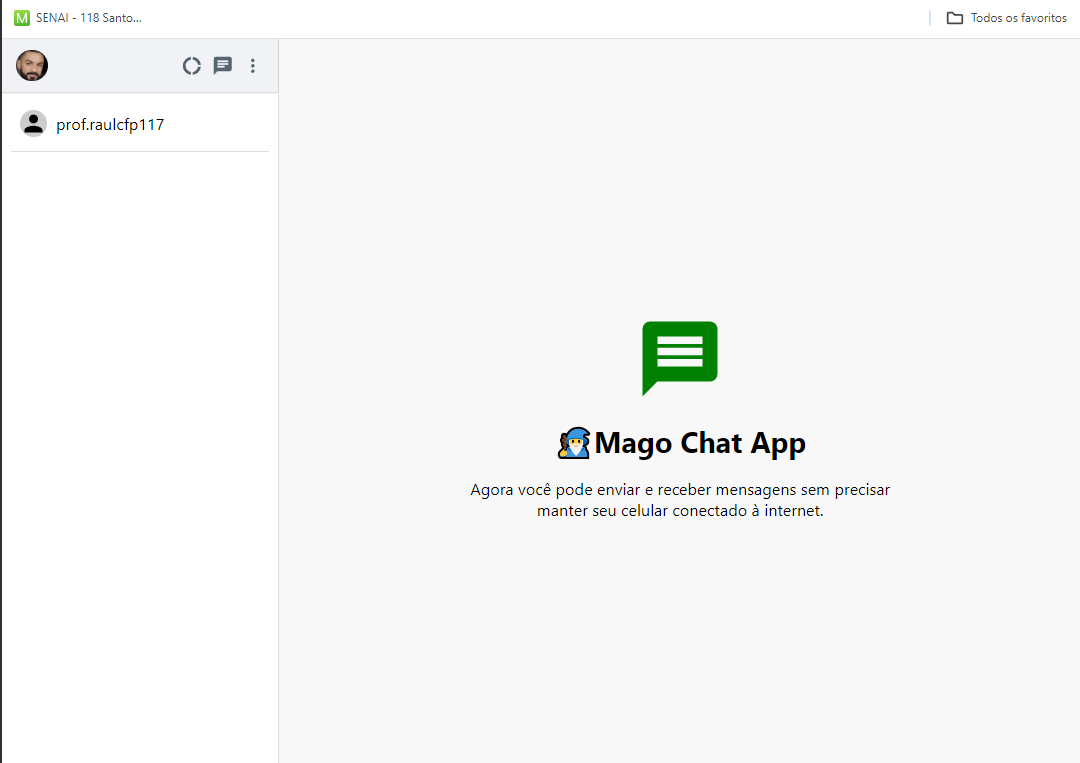
Primeiro vamos no index.jsx da pasta Chat e importar o default



E agora vamos inserir a verificação If



Agora quando não tiver nenhum chat ativo esse será o resultado



Agora vamos finalizar o ChatBody

Em styles



E no index.jsx vamos realizar as importações do react user efect do banco de dados e estilos



Como propriedade vamos receber o chatId para realizar a consulta no banco de dados

Cada chat vai ter um array messageRes que será ordenada pelo tempo



Agora vamos importar o container no return e dentro desse container sera feito uma verificação na consulta que foi feita

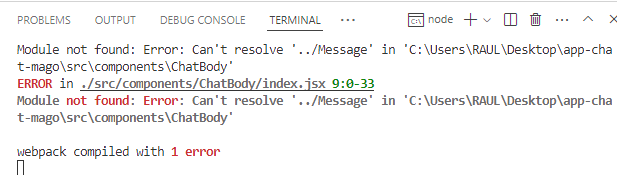


Agora antes do return vamos criar um userEffect para uma verificação da barra de rolagem, para quando der o f5 as mensagem ficam na ordem correta

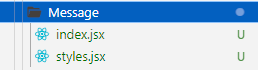


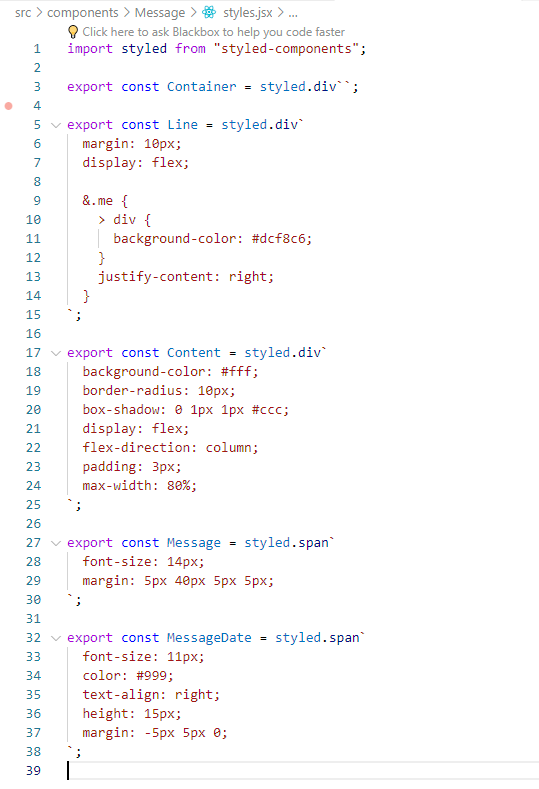
O código completo ficara assim: 

Vai aparecer o erro abaixo por que falta criar um component de Message



Então dento de componentes vamos criar a pasta Message, com o index e o styles.jsx



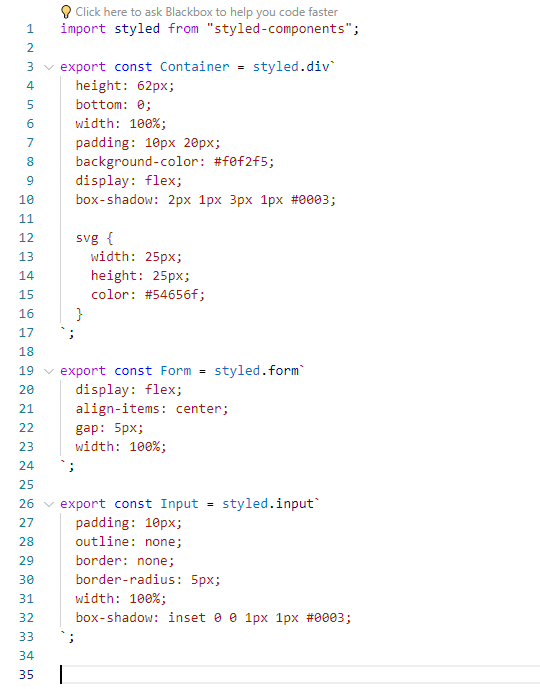
Dentro do styles.jsx do Message: 

Agora no index.jsx fazer as importações e verificar se oque esta sendo importado do body para saber se a mensagem é minha e ficara com a cor verde ou da outra pessoa em branco:



Agora vamos fazer o footer para enviar as mensagens

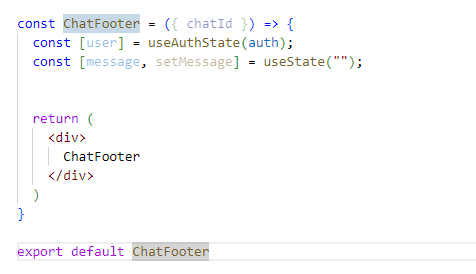
Na pasta ChatFooter em styles.jsx



Agora no index.jsx do ChatFooter vamos criar as importações



Vamos declarar o chatId e declarar as variáveis do user e menssage



Agora dentro do Return retornar os container que vai diferenciar o estado da mensagem

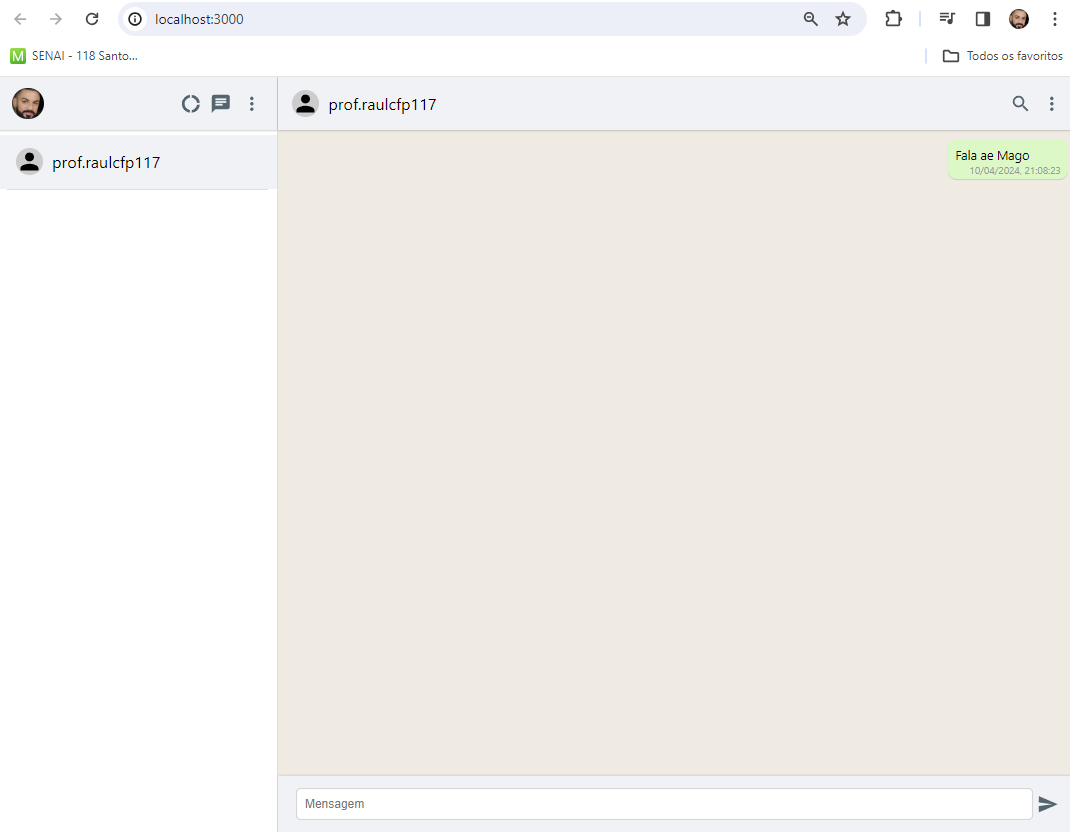


Agora vamos criar a função Handle e declarar o (e) event para carregar a pagina e declarar o chatId para enviar as mensagens com o tempo e no final vai limpar o campo.



O código completo ficara assim

O resultado final



Conseguimos enviar mensagem para responder, logamos na outra conta e responderemos.