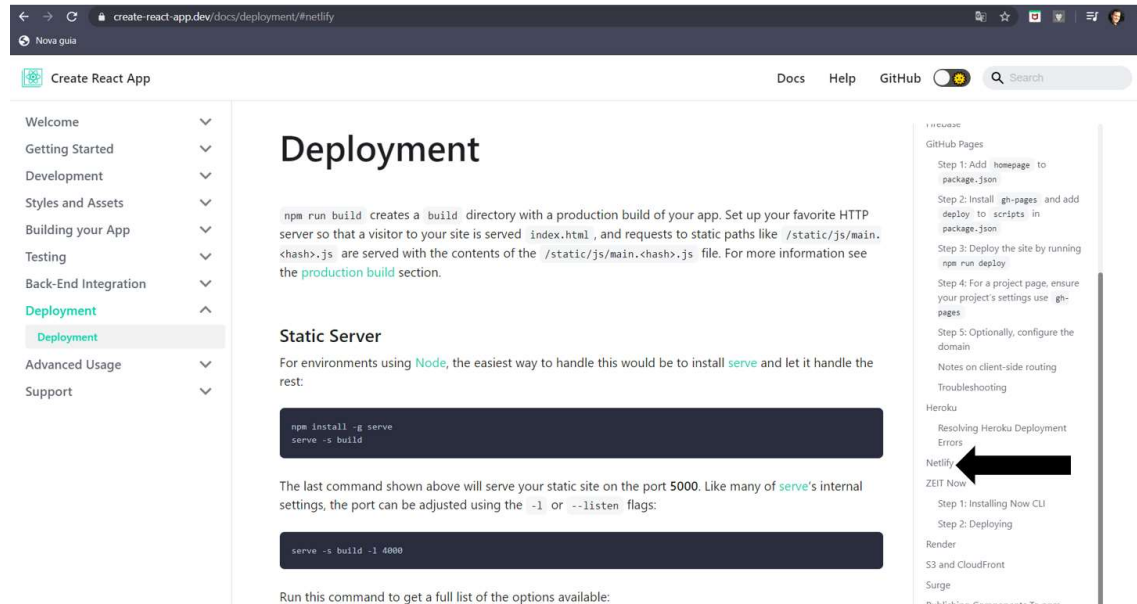


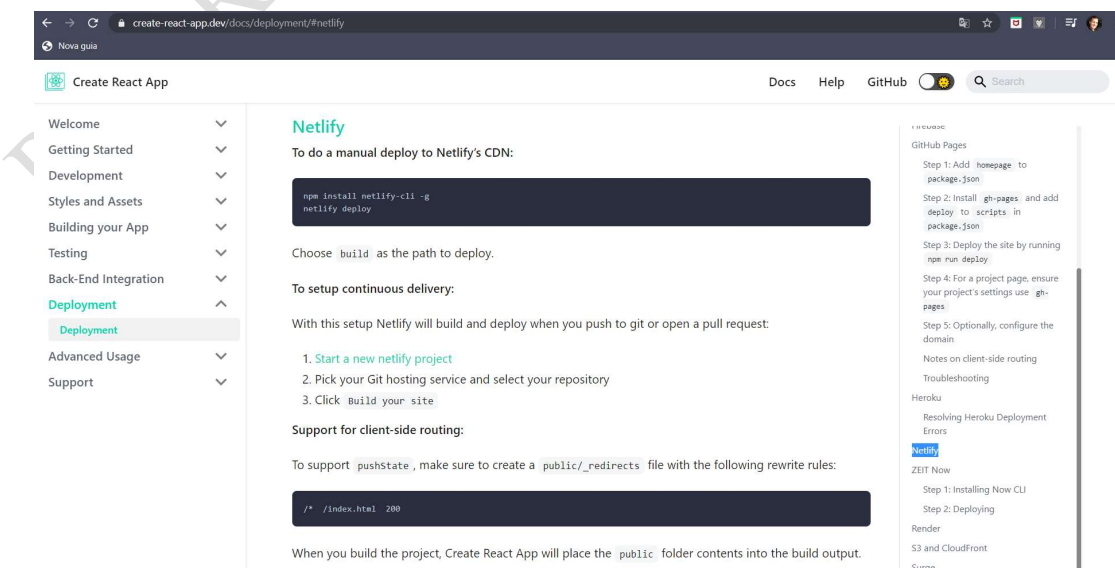
Deploy no ReactJS

Para fazer deploy de uma aplicação React WEB existem inúmeras alternativas, ao entrarmos no site <https://create-react-app.dev/docs/deployment/> existem várias opções.



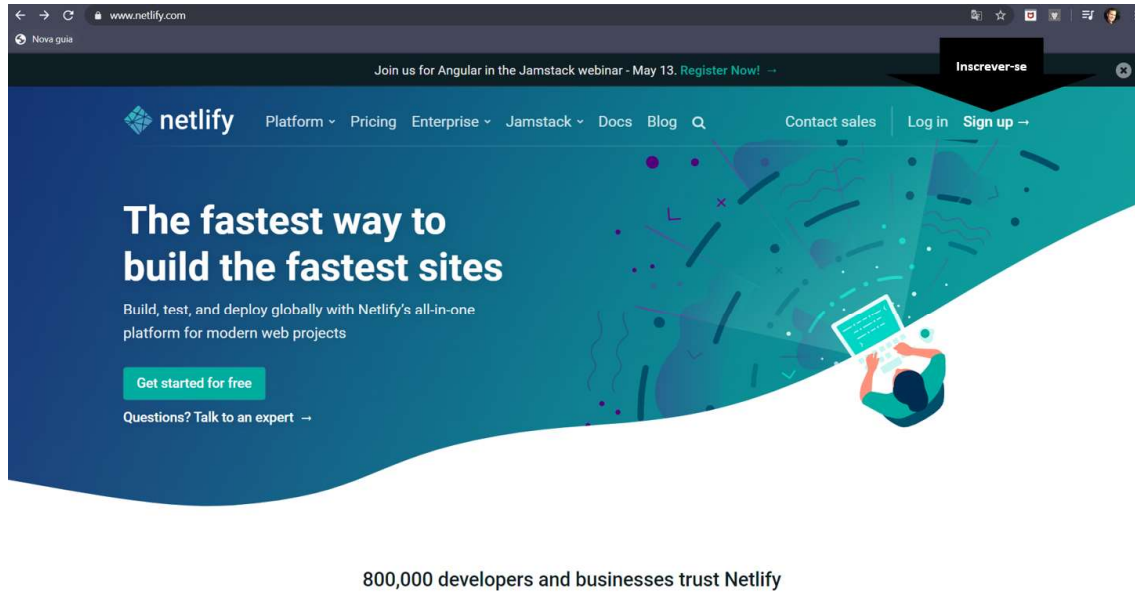
Como o React necessita apenas de recursos front-end, ou seja, tudo o que o usuário irá carregar no browser dele. São arquivos CSS, JS e HTML, não há qualquer tipo de arquivo que seja necessário executar num servidor específico como o Node, PHP ou Java, sistemas que necessitam de um backend. Ele necessita apenas de arquivos estáticos front-end.

Portanto, atualmente, as melhores formas de fazer deploy de uma aplicação React são em serviços que oferecem hospedagem de recursos estáticos. Todos os recursos estáticos são servidos por front-end e processados na máquina do cliente. Diferente, por exemplo, do Node, que se houver um cálculo complexo deverá ser processado pelo lado do servidor. Então não haverá gargalo de performance dentro do front-end da aplicação. Uma opção bastante interessante e gratuita, atualmente é o Netlify, que pode ser vista a seguir.

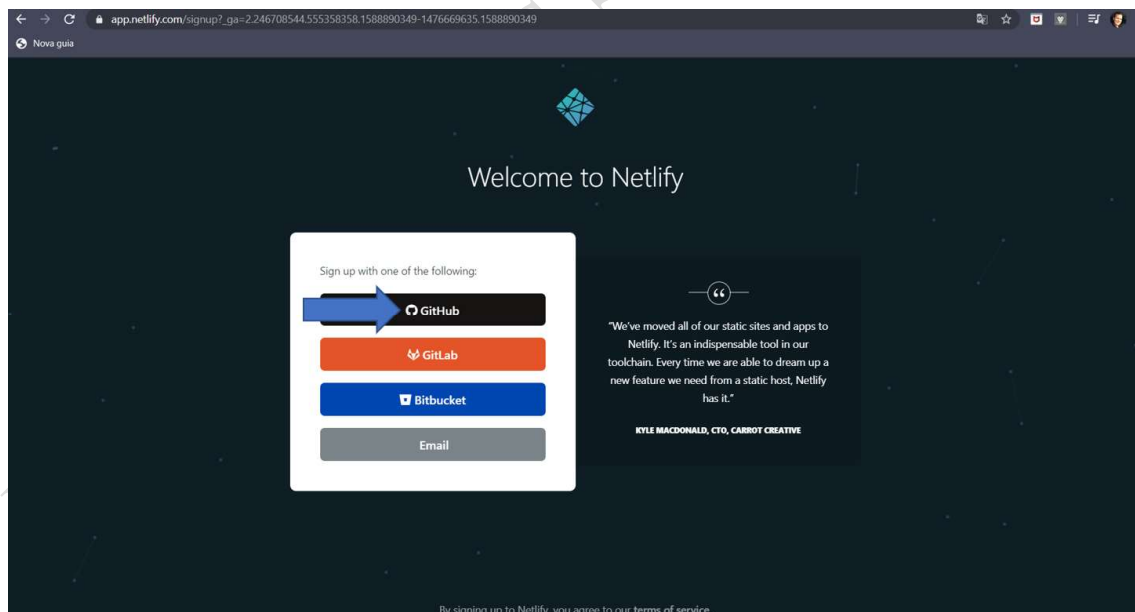


O Netlify nos permite publicar aplicações React, como também qualquer aplicação do tipo front-end, com domínio próprio e com controle de https também gratuito. É algo simplesmente mágico o que se encontra disponível para publicarmos aplicações front-end.

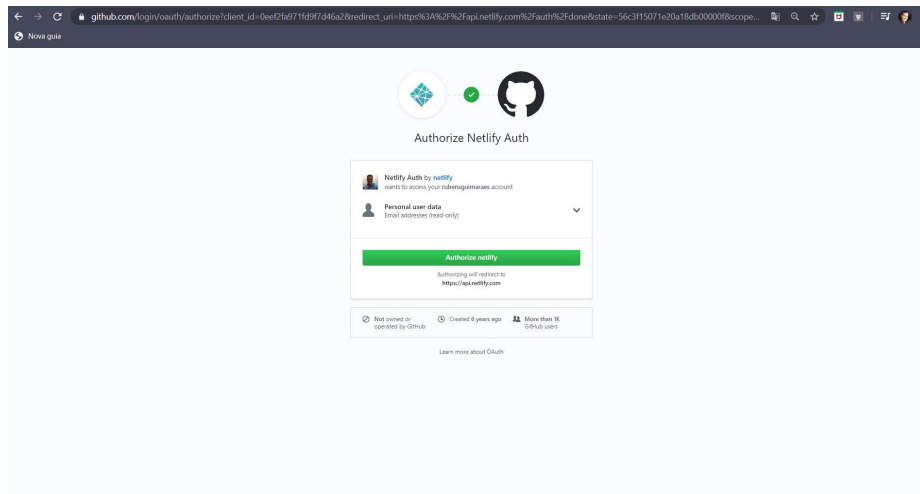
O site Netlify é o seguinte.



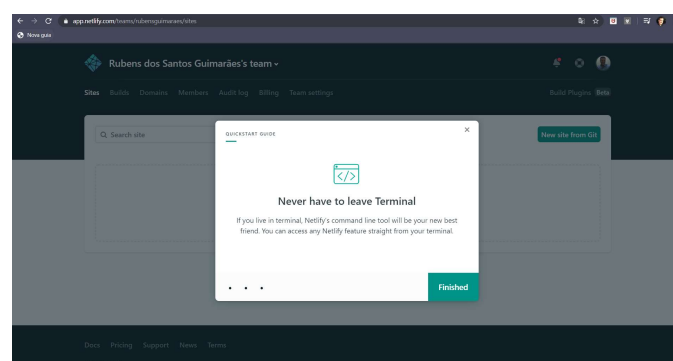
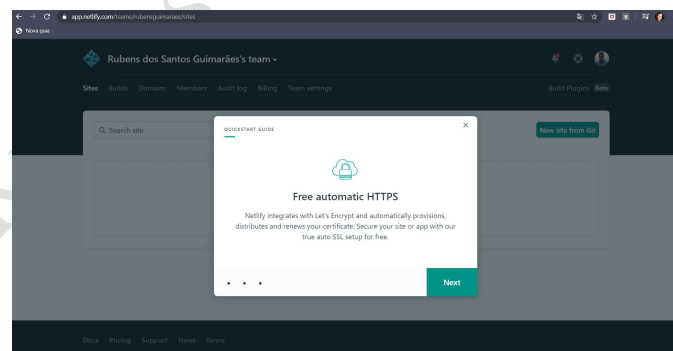
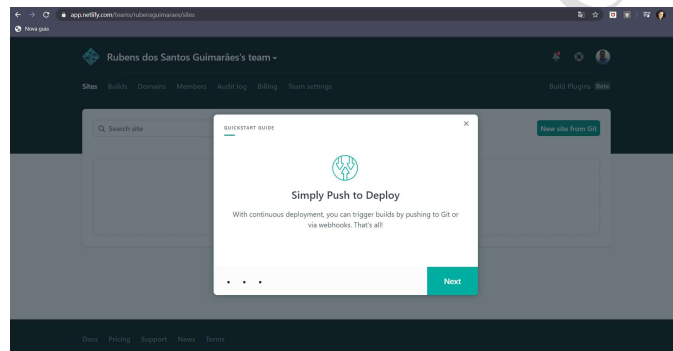
Para publicarmos aplicações no Netlify, é necessário nos inscrevermos.



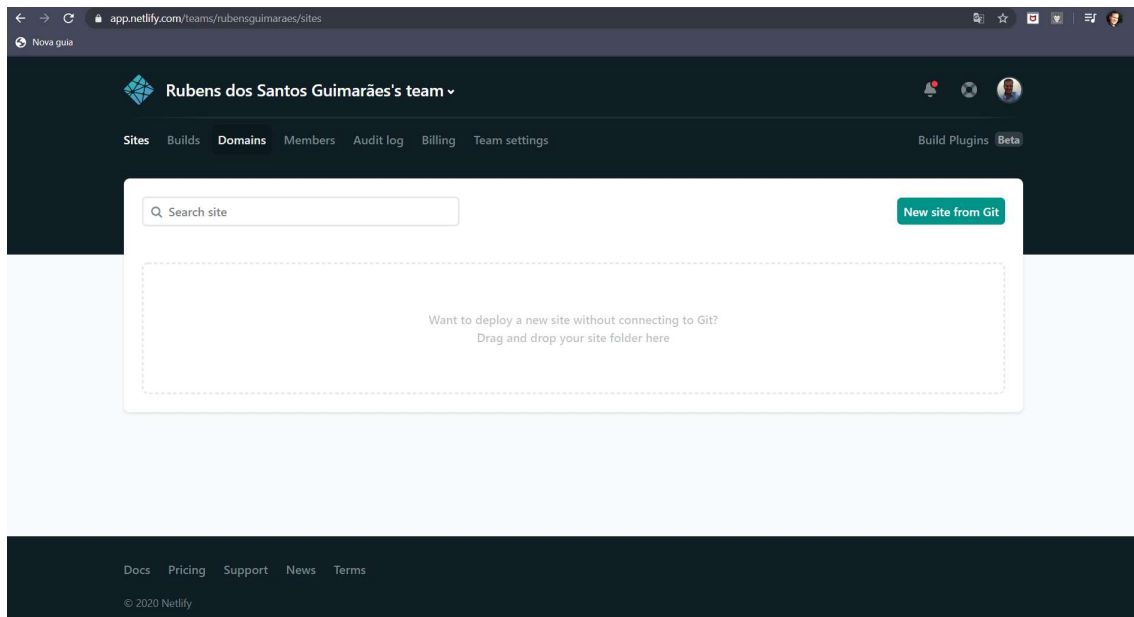
É recomendável que você utilize sua conta do GitHub, pois desta forma suas publicações poderão ser realizadas de forma automática.



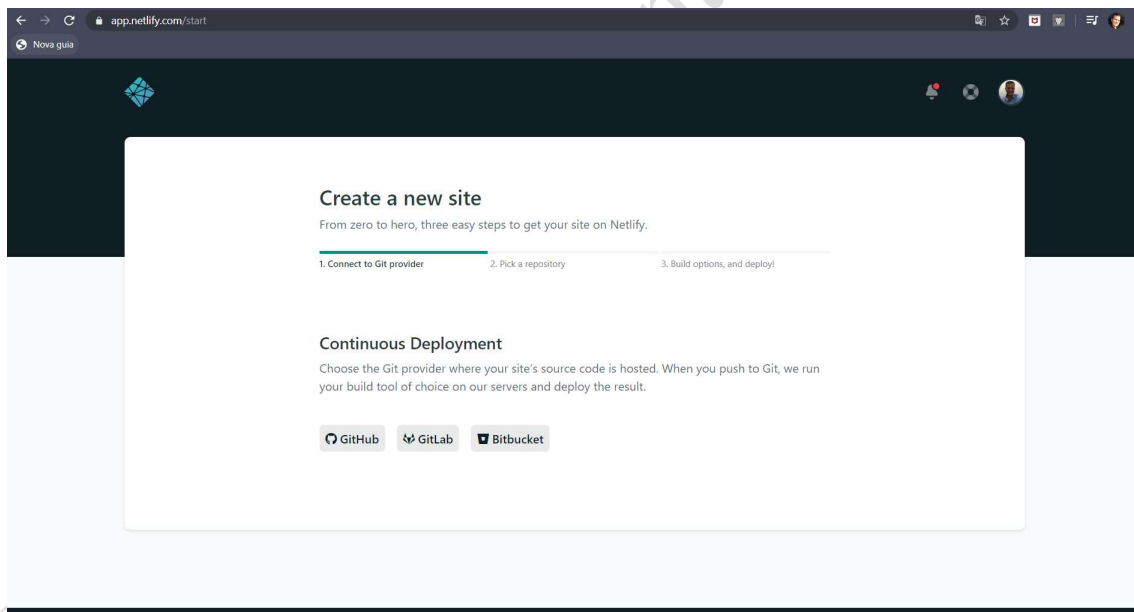
Em seguida será apresentado algumas informações pelo site.



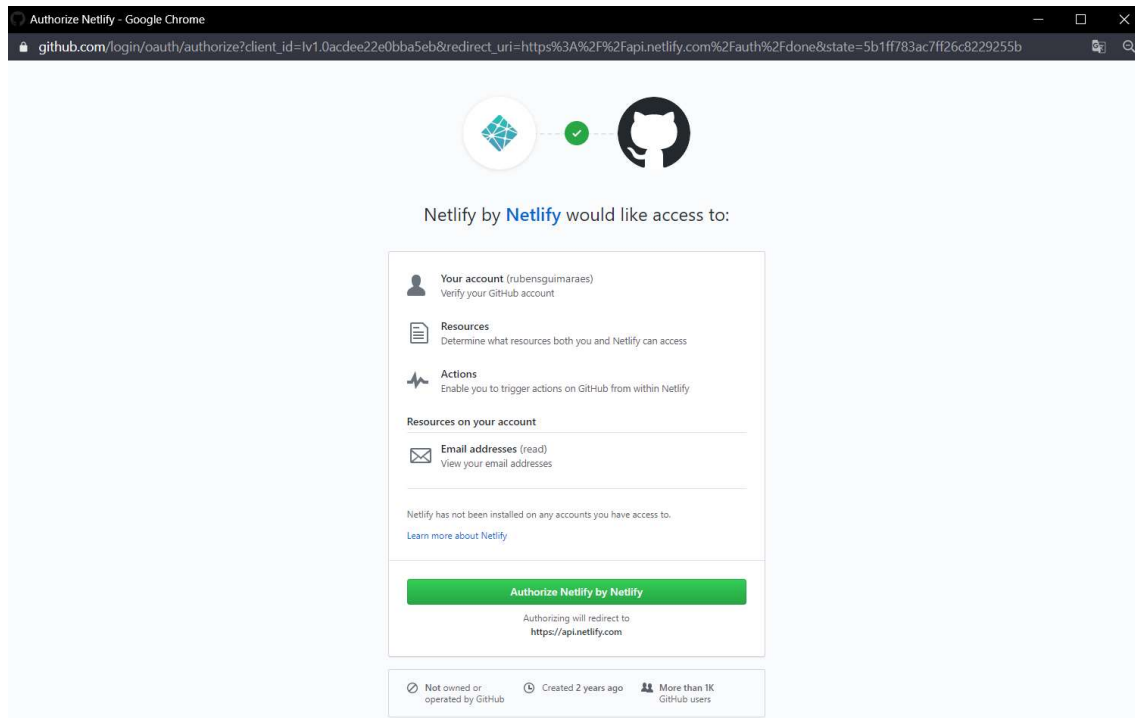
Finalmente você poderá clicar em New site from Git para publicar a sua aplicação.



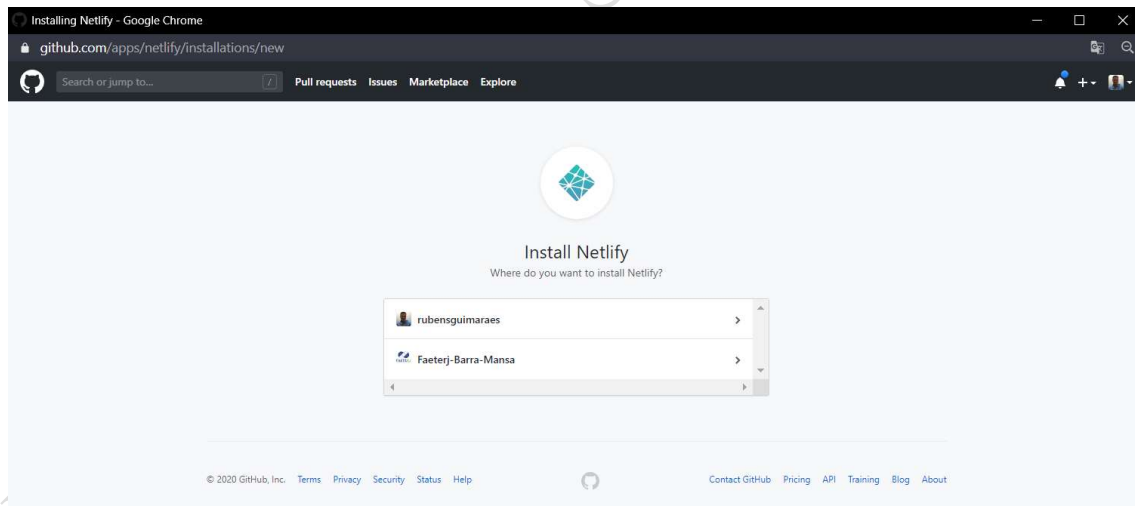
Em seguida, você deverá informar qual o seu provedor Git, selecione GitHub.



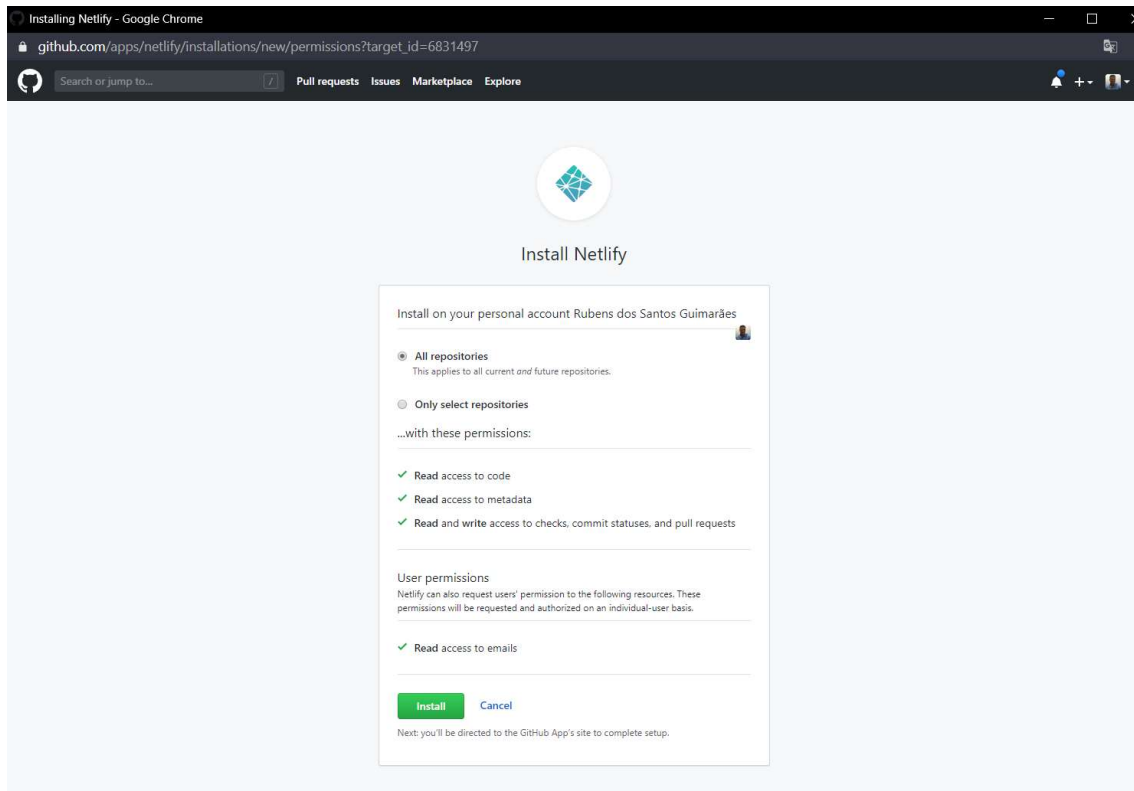
Em seguida, ele solicitará autorização para acessar os meus repositórios. Clique em **Authorize**.



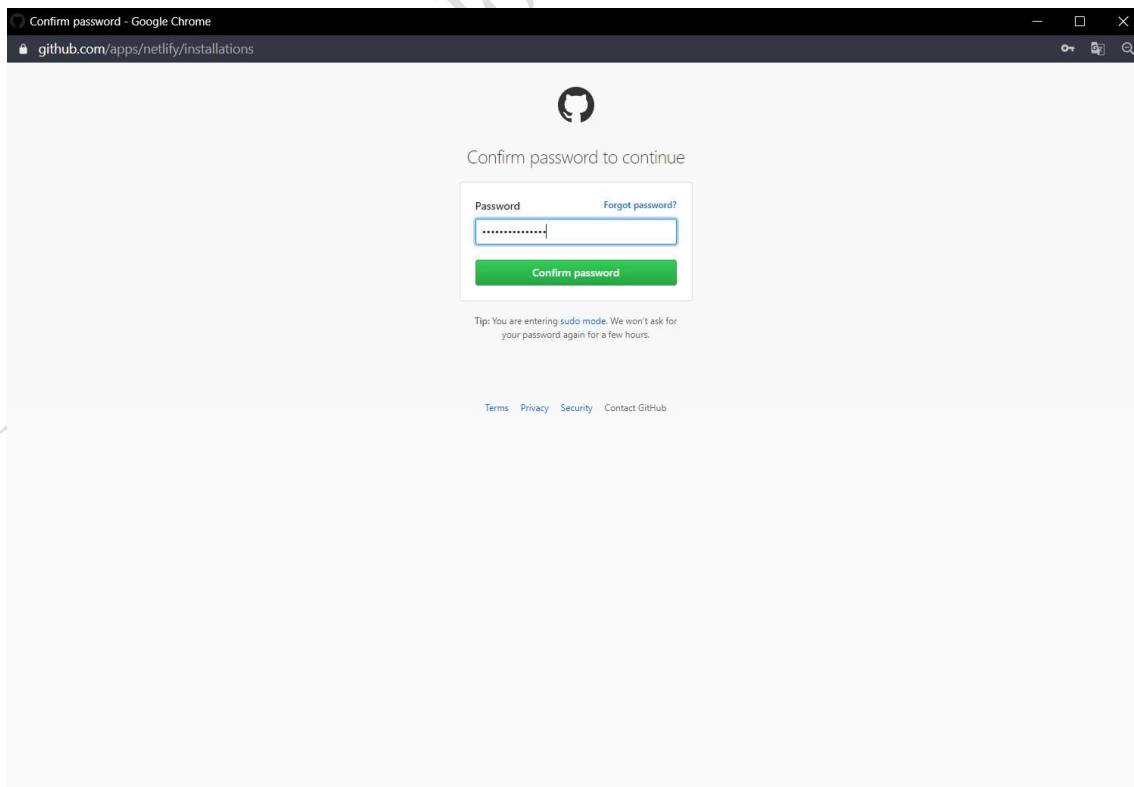
Em seguida, eu darei acesso aos meus repositórios. Vou clicar em rubensguimaraes.



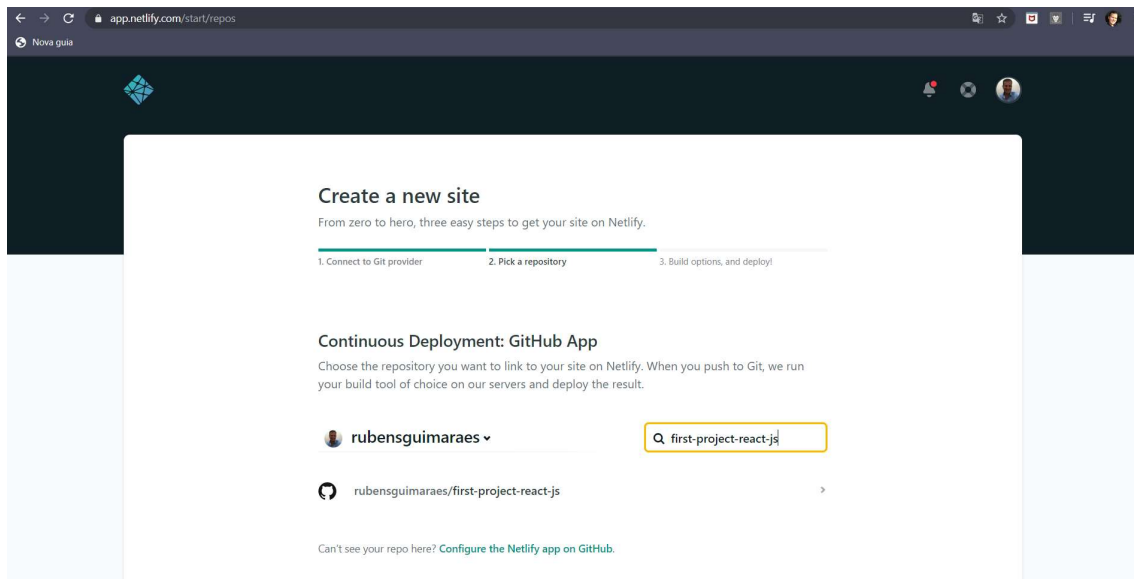
Em seguida, darei acesso a todos os meus repositórios, confirmando “All repositories” e clicando em Install.



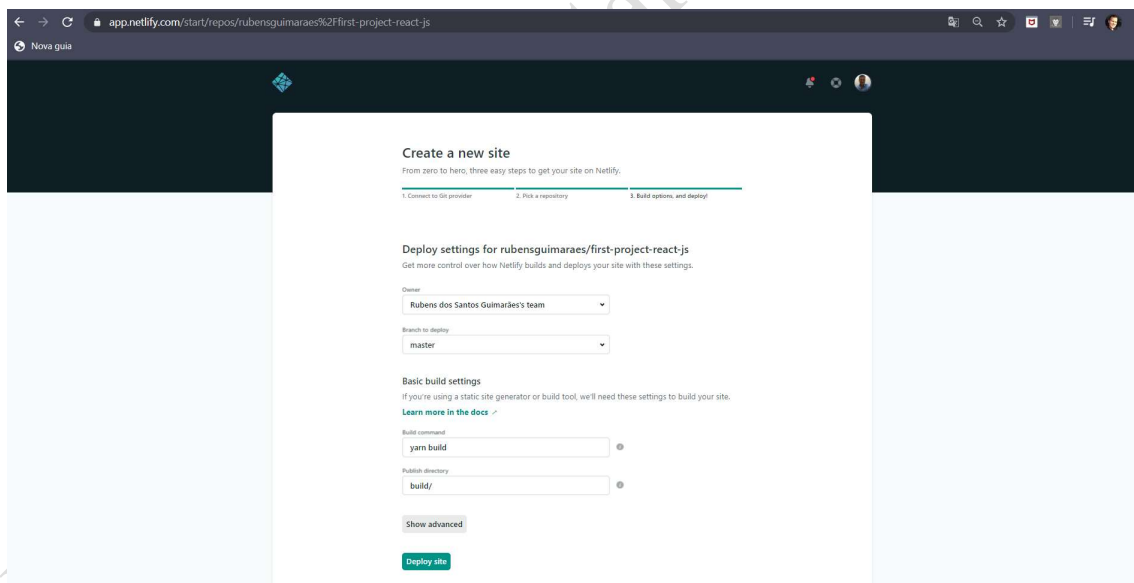
Em seguida, informarei minha senha do Github para acessar.



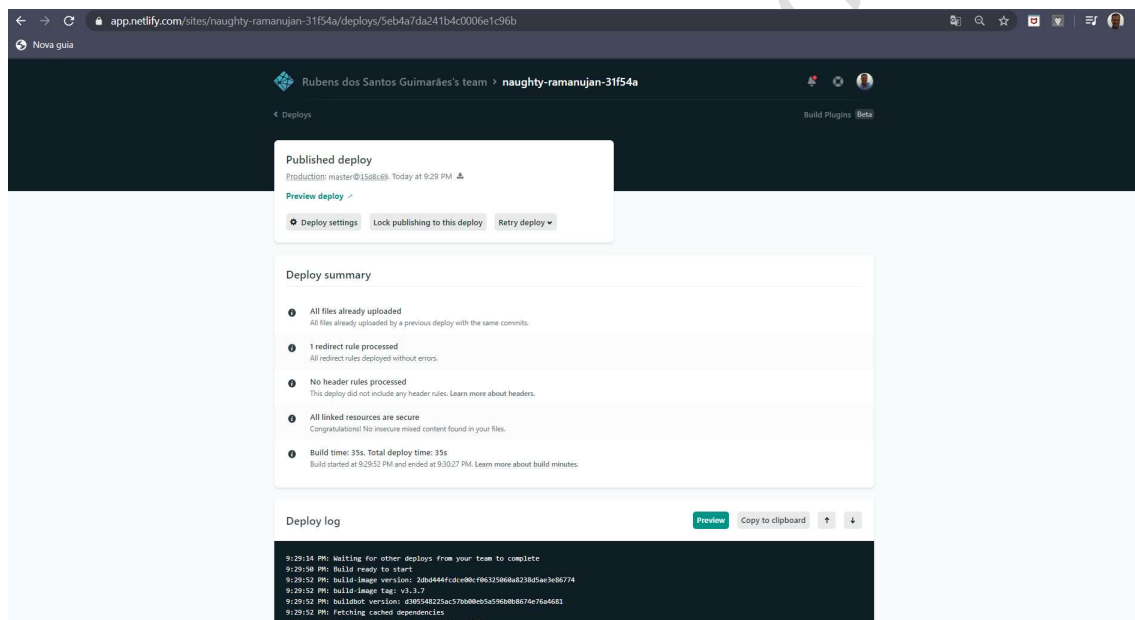
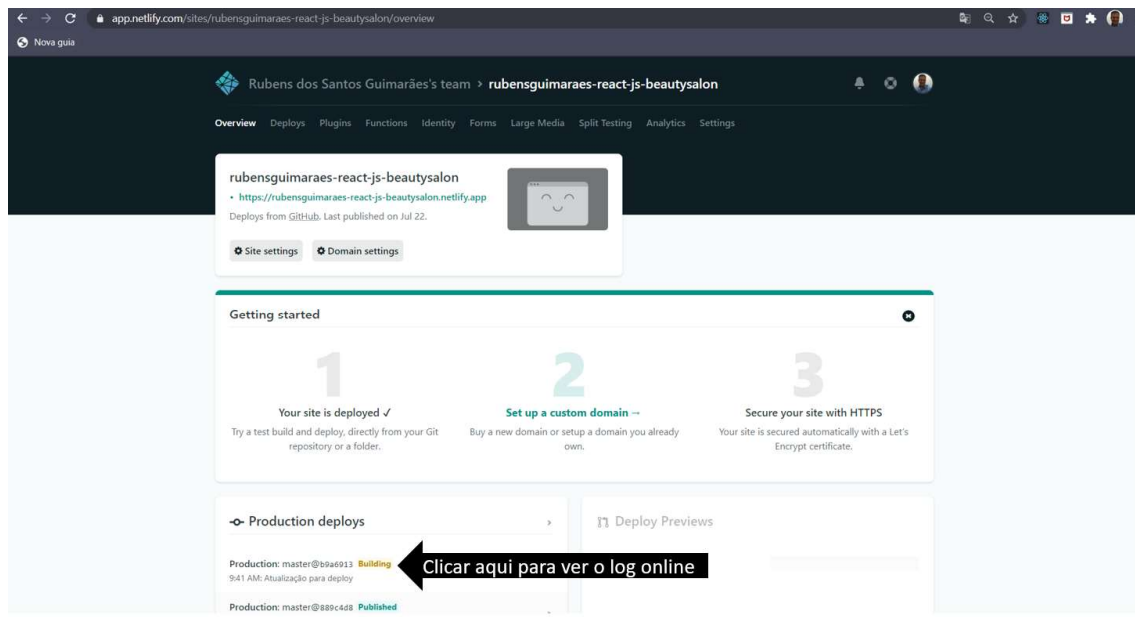
Vou pesquisar o projeto first-project-react-js e vou clicar neste projeto para seleccioná-lo.

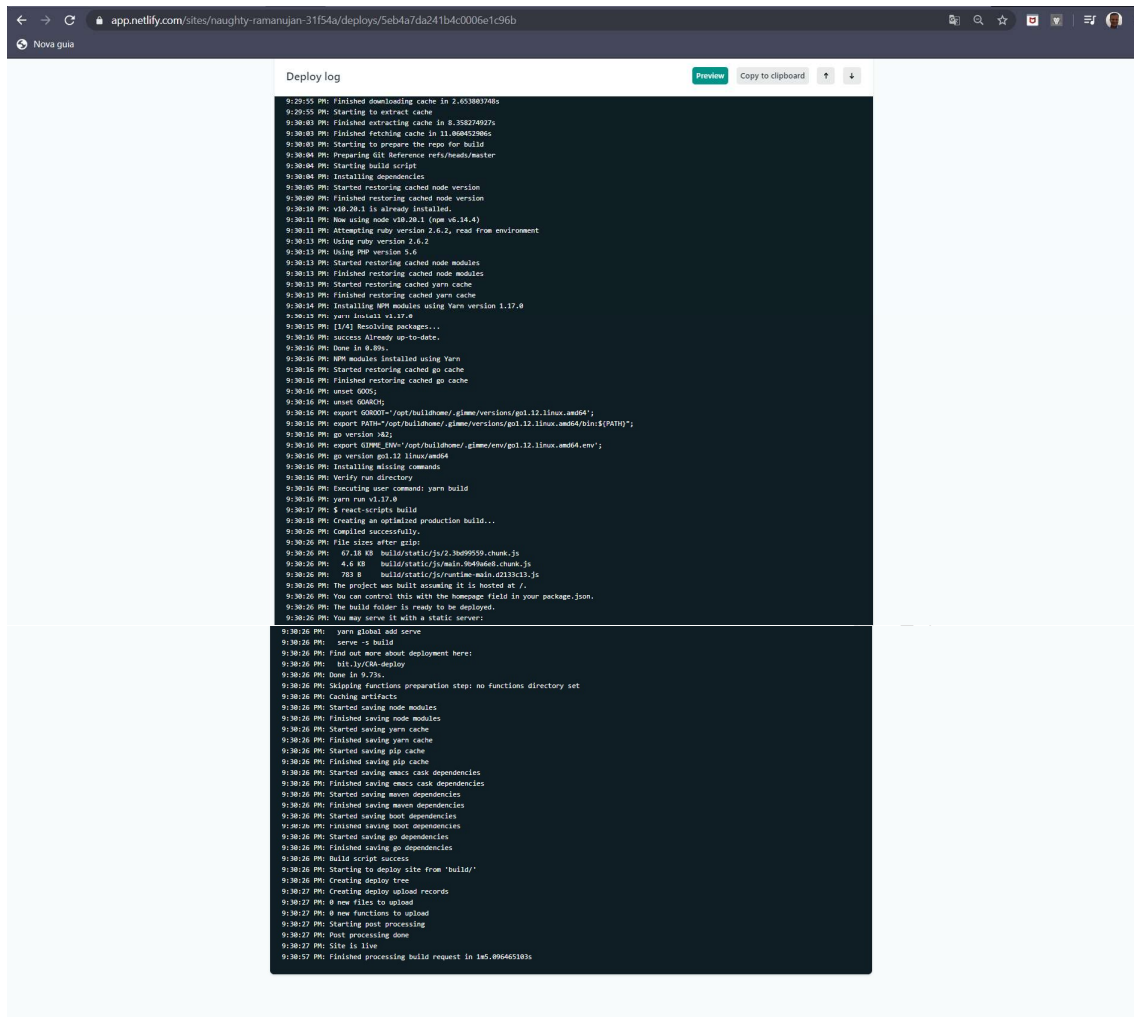


Ele carregará os branches, o meu já está seleccionado o master. Cliquei no botão Deploy site.



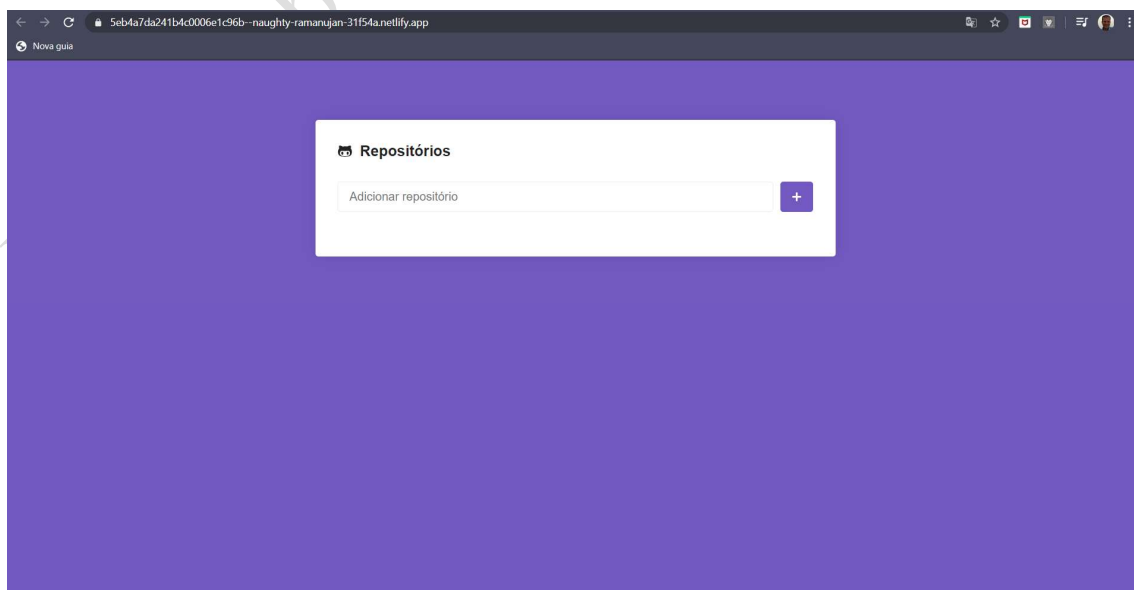
Após clicar em Deploy site, você poderá clicar numa opção que aparece durante o deploy, para que você possa acompanhar o log do deploy, conforme é apresentado nas figuras a seguir.



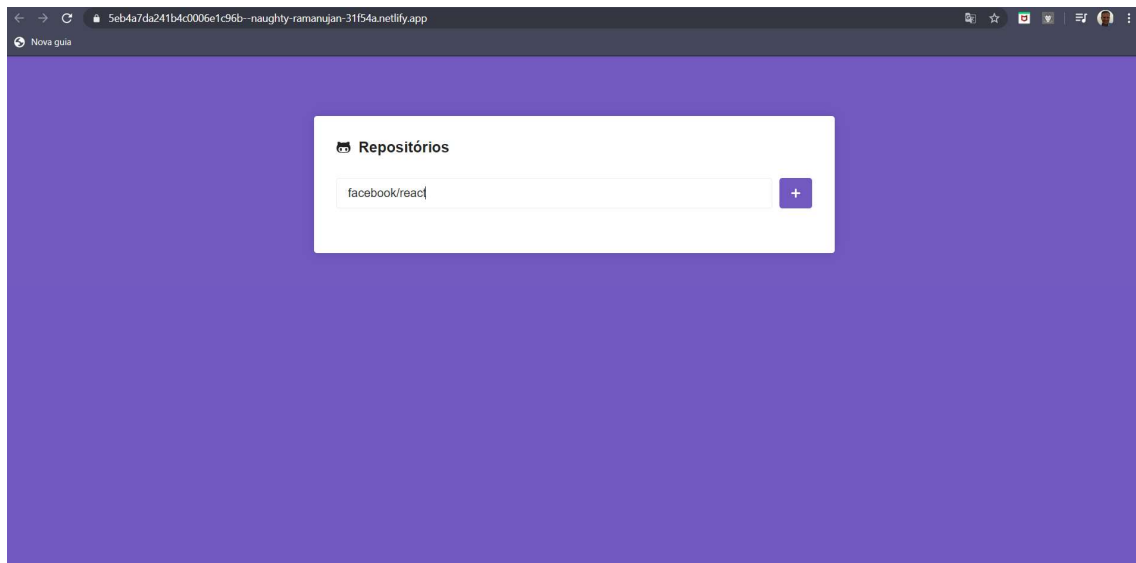


```
Deploy log
9:29:55 PM: Finished downloading cache in 2.453883748s
9:29:55 PM: Starting to extract cache
9:29:58 PM: Finished extracting cache in 8.338278927s
9:30:03 PM: Finished fetching cache in 11.08452986s
9:30:03 PM: Starting to prepare the repo for build
9:30:04 PM: Preparing Git Reference refs/heads/master
9:30:04 PM: Starting build script
9:30:04 PM: Installing dependencies
9:30:05 PM: Started restoring cached node version
9:30:05 PM: Finished restoring cached node version
9:30:10 PM: v18.20.1 is already installed.
9:30:11 PM: Now using node v18.20.1 (npm v8.14.0)
9:30:11 PM: Installing Ruby version 2.6.2, read from environment
9:30:13 PM: Using ruby version 2.6.2
9:30:13 PM: Using PHP version 5.6
9:30:13 PM: Started restoring cached node modules
9:30:13 PM: Finished restoring cached node modules
9:30:13 PM: Started restoring cached yarn cache
9:30:13 PM: Finished restoring cached yarn cache
9:30:14 PM: Installing NPM modules using Yarn version 1.17.0
9:30:15 PM: yarn install v1.17.0
9:30:15 PM: [1/0] Resolving packages...
9:30:16 PM: success Already up-to-date.
9:30:16 PM: Done in 0.09s.
9:30:16 PM: NPM modules installed using Yarn
9:30:16 PM: Started restoring cached go cache
9:30:16 PM: Finished restoring cached go cache
9:30:16 PM: unset GOOS;
9:30:16 PM: unset GOARCH;
9:30:16 PM: export GOPATH="/opt/buildhome/.gimme/versions/go1.12.linux.amd64";
9:30:16 PM: export PATH="/opt/buildhome/.gimme/versions/go1.12.linux.amd64/bin:$PATH";
9:30:16 PM: go version 1.21
9:30:16 PM: export GOROOT="/opt/buildhome/.gimme/versions/go1.12.linux.amd64";
9:30:16 PM: go version 1.21
9:30:16 PM: Installing missing commands
9:30:16 PM: Verify run directory
9:30:16 PM: Executing user command: yarn build
9:30:16 PM: yarn run v1.17.0
9:30:17 PM: $ react-scripts build
9:30:18 PM: Creating an optimized production build...
9:30:20 PM: Compiled successfully.
9:30:20 PM: File sizes after gzip:
9:30:20 PM: 67.18 kB build/static/js/2.3b009550.chunk.js
9:30:20 PM: 4.6 kB build/static/js/main.9d49a6e8.chunk.js
9:30:20 PM: 781 B build/static/js/runtime-main.d213c1c1.js
9:30:20 PM: The project was built assuming it is hosted at /.
9:30:20 PM: You can control this with the homepage field in your package.json.
9:30:20 PM: The build folder is ready to be deployed.
9:30:20 PM: You may serve it with a static server.
9:30:26 PM: yarn global add serve
9:30:26 PM: serve -s build
9:30:26 PM: Find out more about deployment here:
9:30:26 PM: https://cra.link/ deploy
9:30:26 PM: Done in 9.73s.
9:30:26 PM: Skipping functions preparation step: no functions directory set
9:30:26 PM: Caching artifacts
9:30:26 PM: Started saving node modules
9:30:26 PM: Finished saving node modules
9:30:26 PM: Started saving yarn cache
9:30:26 PM: Finished saving yarn cache
9:30:26 PM: Started saving pip cache
9:30:26 PM: Finished saving pip cache
9:30:26 PM: Started saving emacs cask dependencies
9:30:26 PM: Finished saving emacs cask dependencies
9:30:26 PM: Started saving maven dependencies
9:30:26 PM: Finished saving maven dependencies
9:30:26 PM: Started saving boot dependencies
9:30:26 PM: Finished saving boot dependencies
9:30:26 PM: Started saving rust dependencies
9:30:26 PM: Finished saving rust dependencies
9:30:26 PM: Started saving go dependencies
9:30:26 PM: Finished saving go dependencies
9:30:26 PM: Build script success
9:30:26 PM: Starting to deploy site from 'build/'
9:30:26 PM: Creating deploy tree
9:30:27 PM: Creating deploy without records
9:30:27 PM: 0 new files to upload
9:30:27 PM: 0 new functions to upload
9:30:27 PM: Starting post processing
9:30:27 PM: Post processing done
9:30:27 PM: Site is live
9:30:27 PM: Finished processing build request in 145.696465183s
```

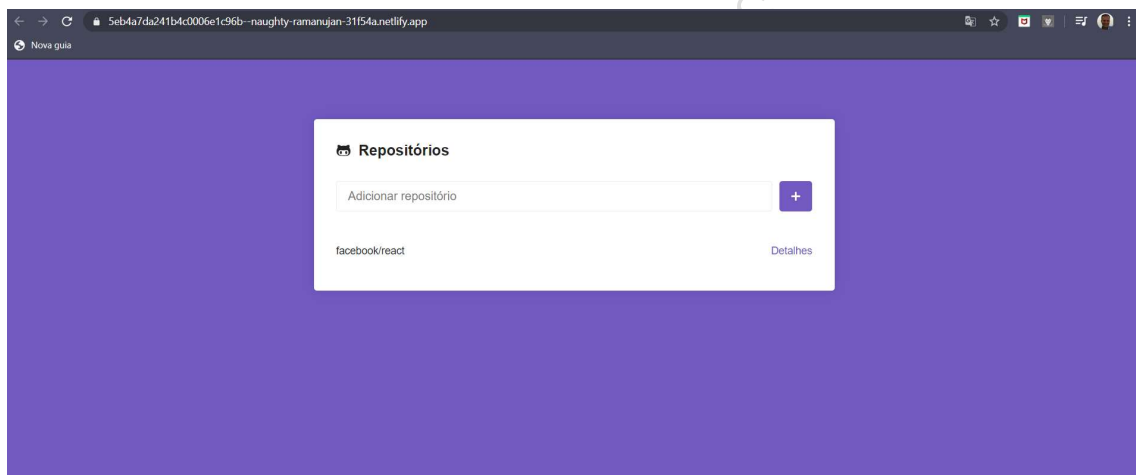
Observe a mensagem no log: “site is live”, ou seja, “o site está no ar”. Podemos clicar no botão verde Preview, para ver um preview deste deploy. Será gerado um link somente para preview.



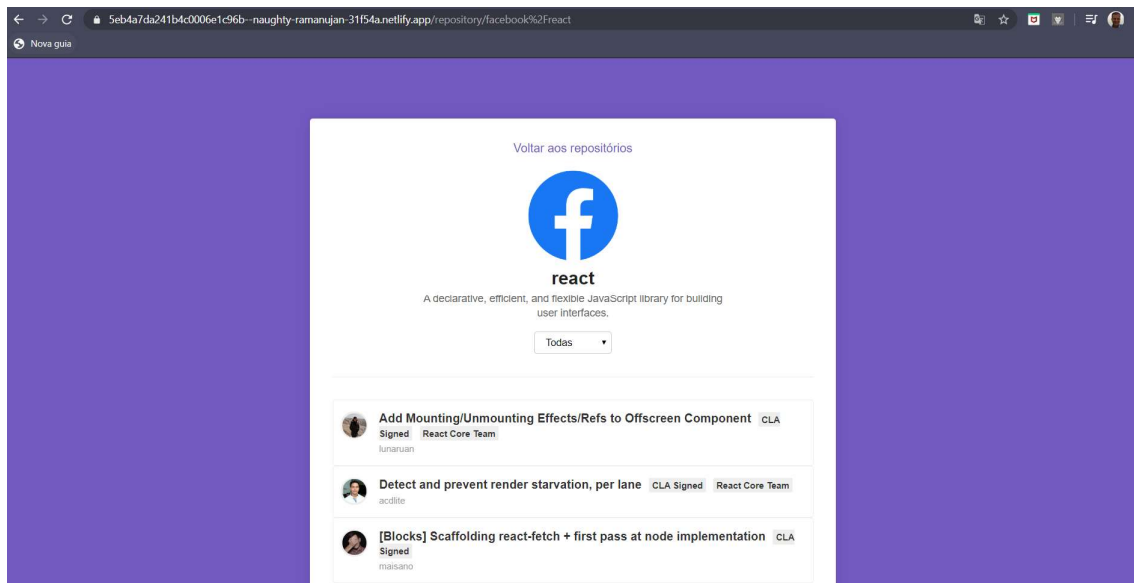
Posso testar a aplicação:



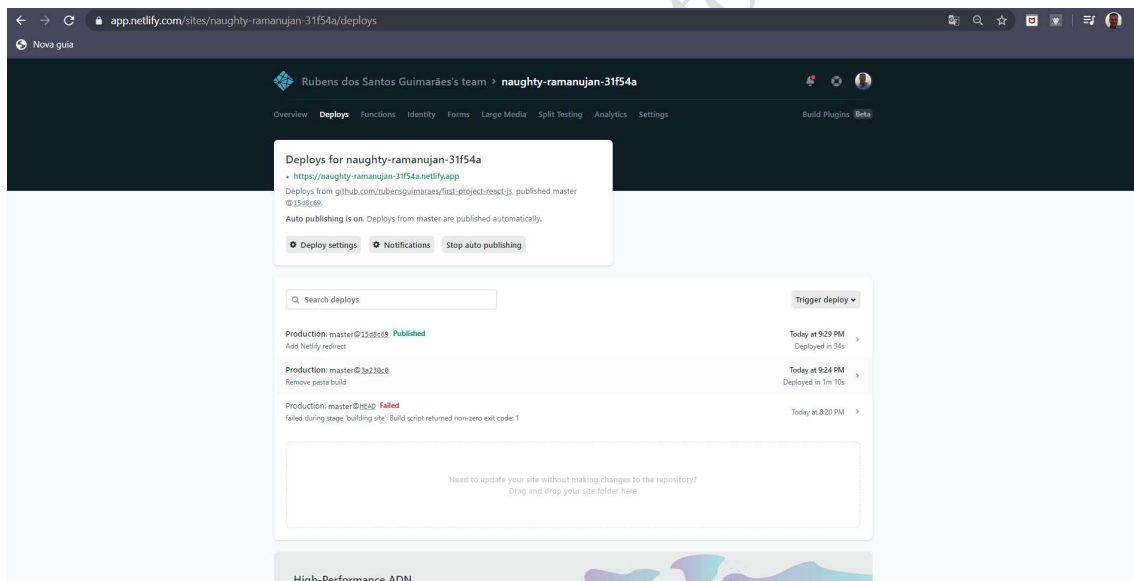
Clico no botão



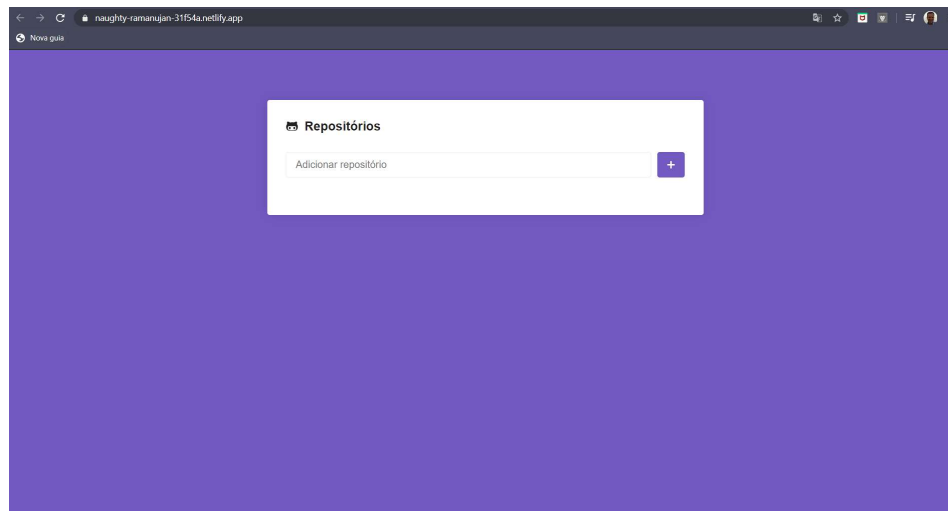
Clico em detalhes



Ao voltar ao Netlify, posso ver que a aplicação já foi publicada, você pode observar o domínio gerado em produção.



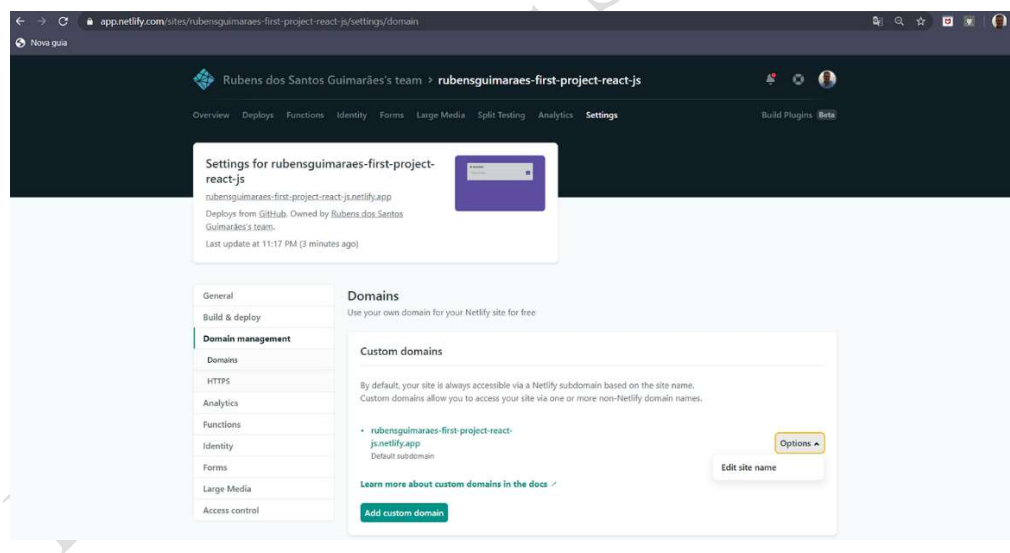
Ao clicar no domínio gerado, veja a página a seguir.



Portanto, o domínio gerado para a aplicação foi o seguinte:

<https://naughty-ramanujan-31f54a.netlify.app/>

Também é possível configurar um domínio próprio ou alterar o nome do site, selecionando o menu settings, opção Domain management, botão Options e Edit site name. Veja como apresenta-se abaixo que alterei o nome do site para <https://rubensguimaraes-first-project-react-js.netlify.app/>



O resultado apresentado é o seguinte

