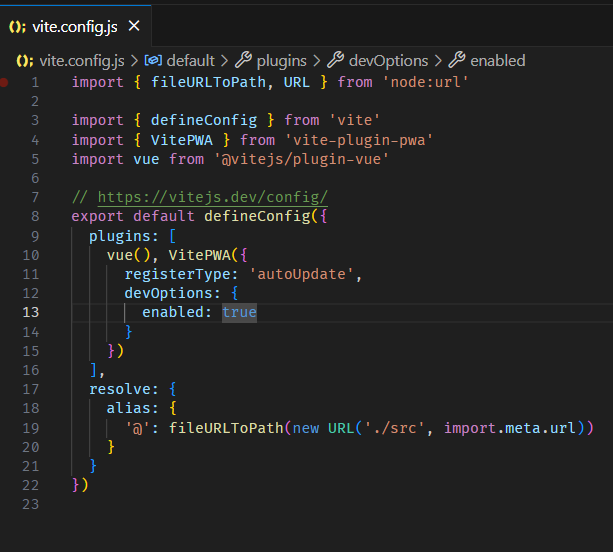
**O QUE SÃO PWAs?**

Permite o desenvolvimento de apps que irão rodar no navegador, utilizando html e javascript, por exemplo. Algumas características: capacidade, confiabilidade e instaláveis. Os PWAs são um conjunto de funcionalidades no navegador.

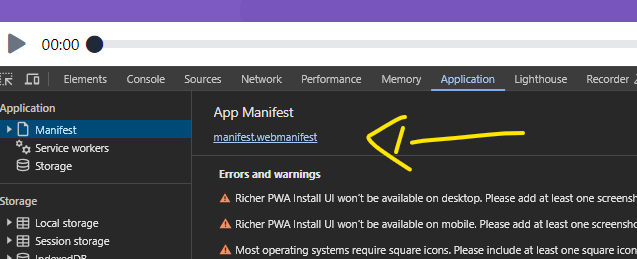
npm i vite-plugin-pwa -D

Em seguida, configurar o arquivo vite.config.js

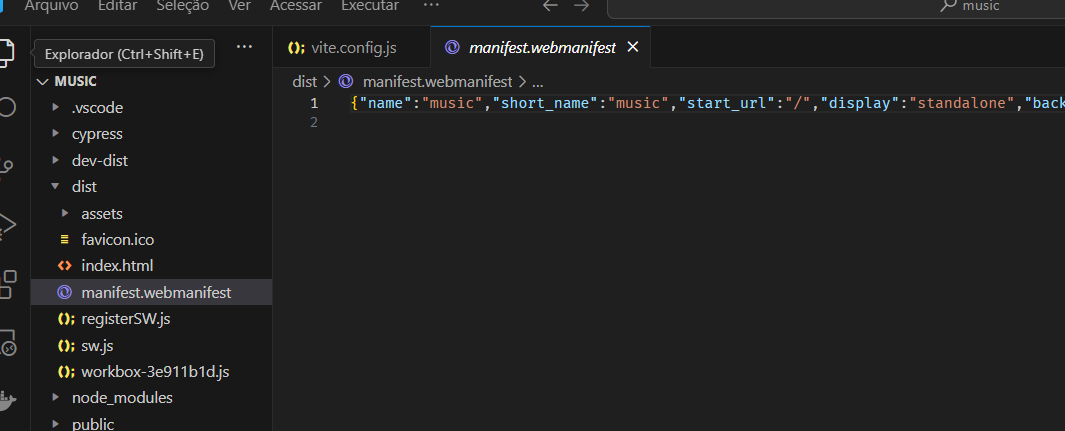


**O ARQUIVO MANIFESTO**

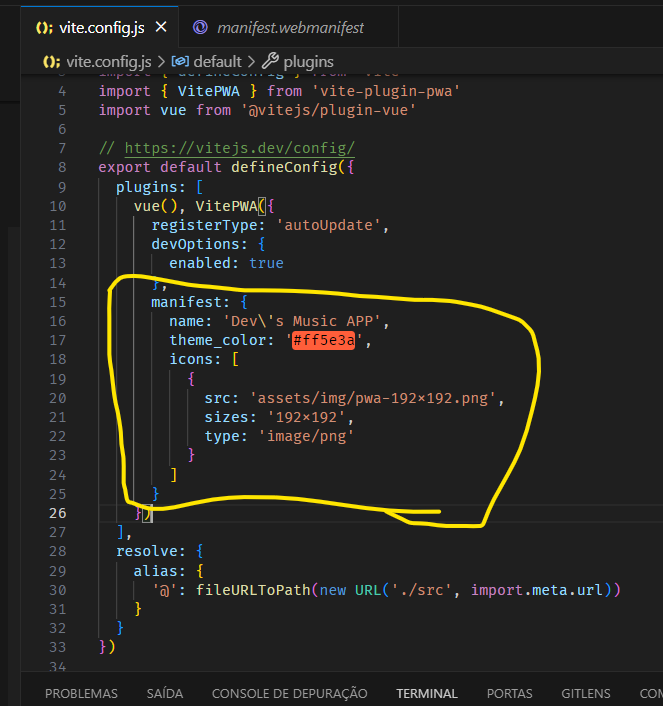
npm run build

Esse comando irá otimizar a aplicação para produção. O arquivo manifesto está no navegador, nesta seção: 

Ou então, após rodar o comando do build acima, ele irá aparecer nessa seção:



**CONFIGURANDO O ARQUIVO MANIFESTO**



**SUPORTE OFFLINE COM PRESTADORES DE SERVIÇO (SERVICE WORKER)**

Service worker é um arquivo javascript rodando em segundo plano. Primariamente utilizado para cache de arquivos, enviar notificações e sincronização em segundo plano. No entanto, não conseguem acessar o DOM e são limitados pelo suporte do navegador. Servicew Worker podem ser criados de duas formas: manualmente ou por meio da workbox.

Para testar nossa aplicação offline sem problemas, rode os seguintes comandos:

npm run build, e em seguida npm run preview. Esse último comando irá criar um servidor para pré-visualizar a construção final de nosso projeto. Diferente do servidor de desenvolvimento, ele não vai assistir nossos arquivos para as alterações.

**ENTENDENDO CACHE**

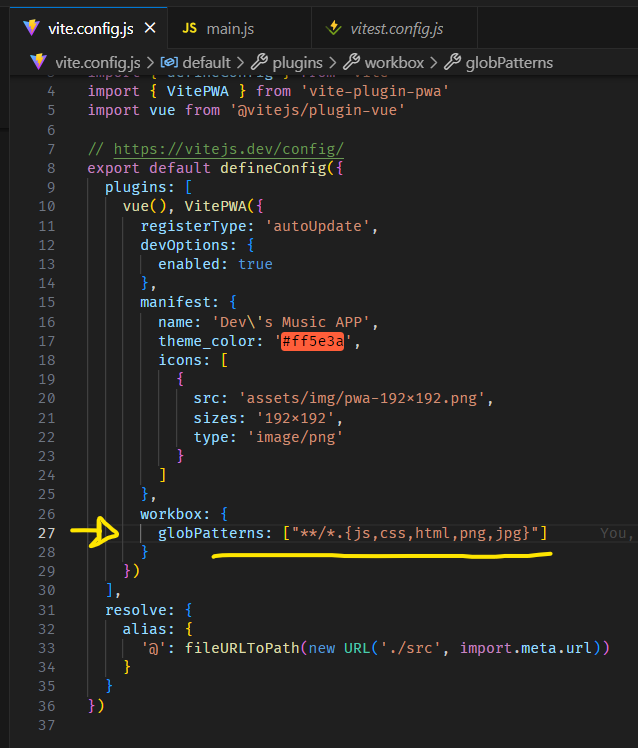
Cache é um conceito onde copias de dados e ativos são armazenados para entrega mais rápida.

“There are only two hard things in Computer Science: caching and naming things.” - Phil Karlton

Alguns arquivos não precisam ser carregados todo momento no navegador, ou então, não é necessário ser feito um requisição, como por exemplo os assets. Nesse caso, o navegador guarda em cache no browser.

**WORKBOX**

Ferramenta que pode implementar diferentes estratégias de cache.

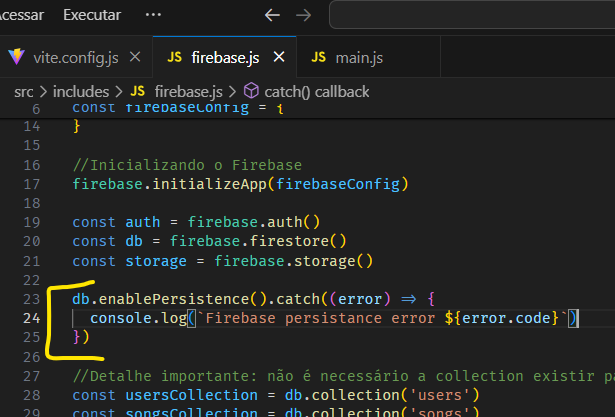


Essa configuração no vite.config.js irá dizer ao navegador que ele deve armazenar em cache os arquivos js, css, html e arquivos de imagem.

Em seguida, rode os comandos: npm run build e em seguida npm run preview. A partir de agora, se em network colocarmos como offline, vamos ver que esses arquivos continuam carregando!

**FIREBASE DATA PERSISTENCE**

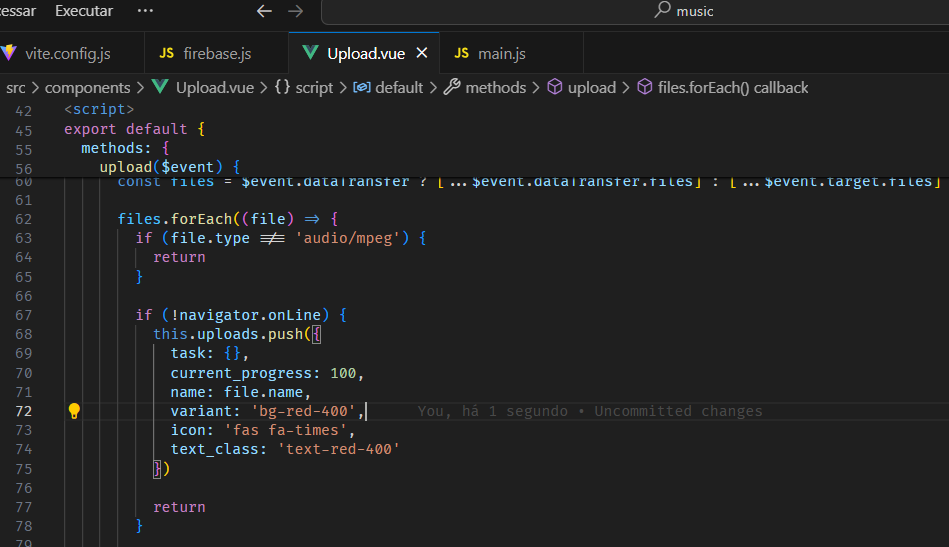
Para habilitar no projeto, basta acrescentar essa linha de código:



Algumas considerações sobre o firebase SDK e a sua feature de persistencia:

* Impossível autenticar ou fazer upload de arquivos.
* Armazenamento é limitado.

**LIDANDO COM UPLOADS OFFLINE**

****

Adicionei essas linhas de código do if (!navigator.onLine) que irá verificar e tratar os dados caso o usuário não esteja online.