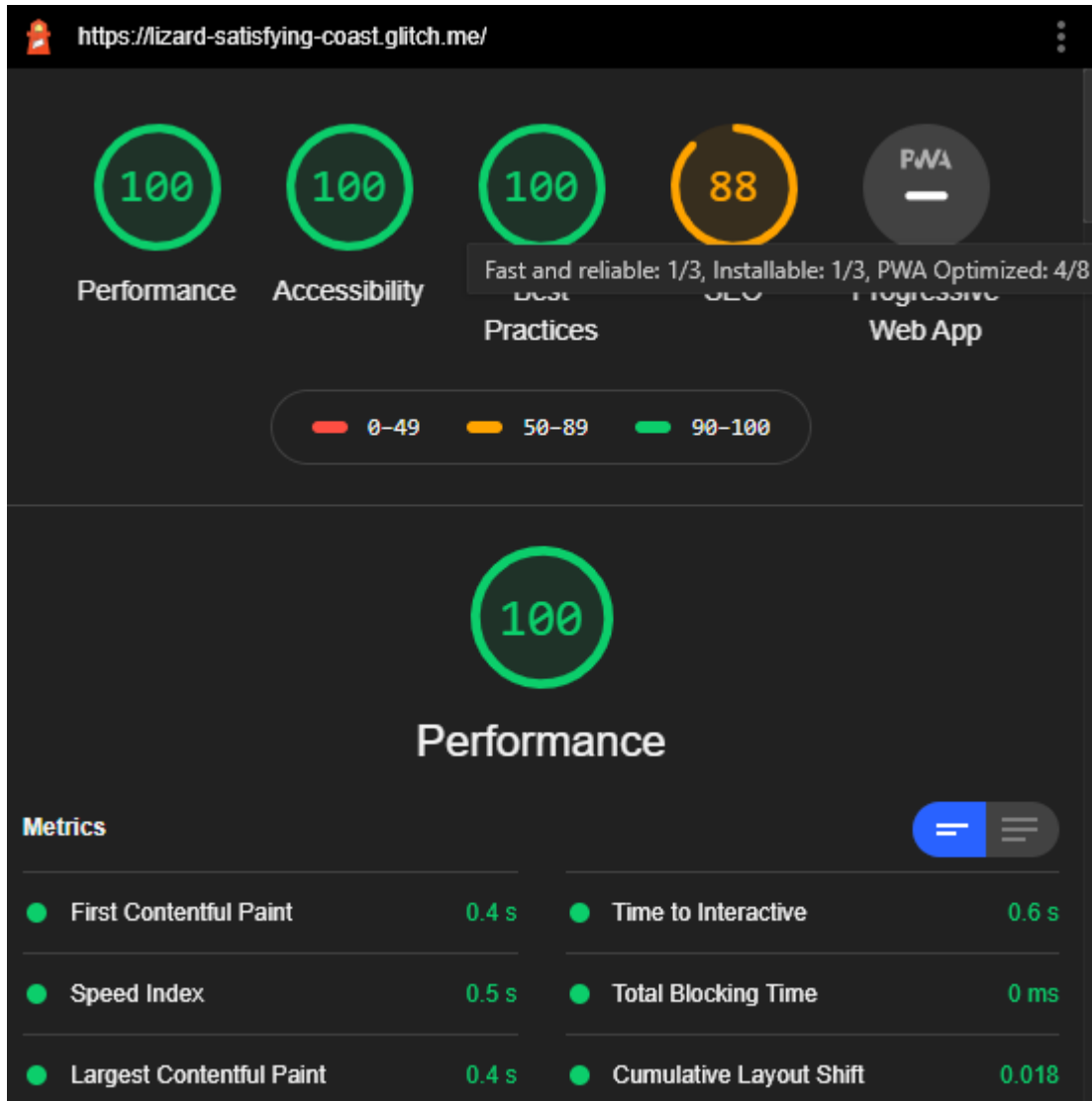


Taller 1 (Semana 2)

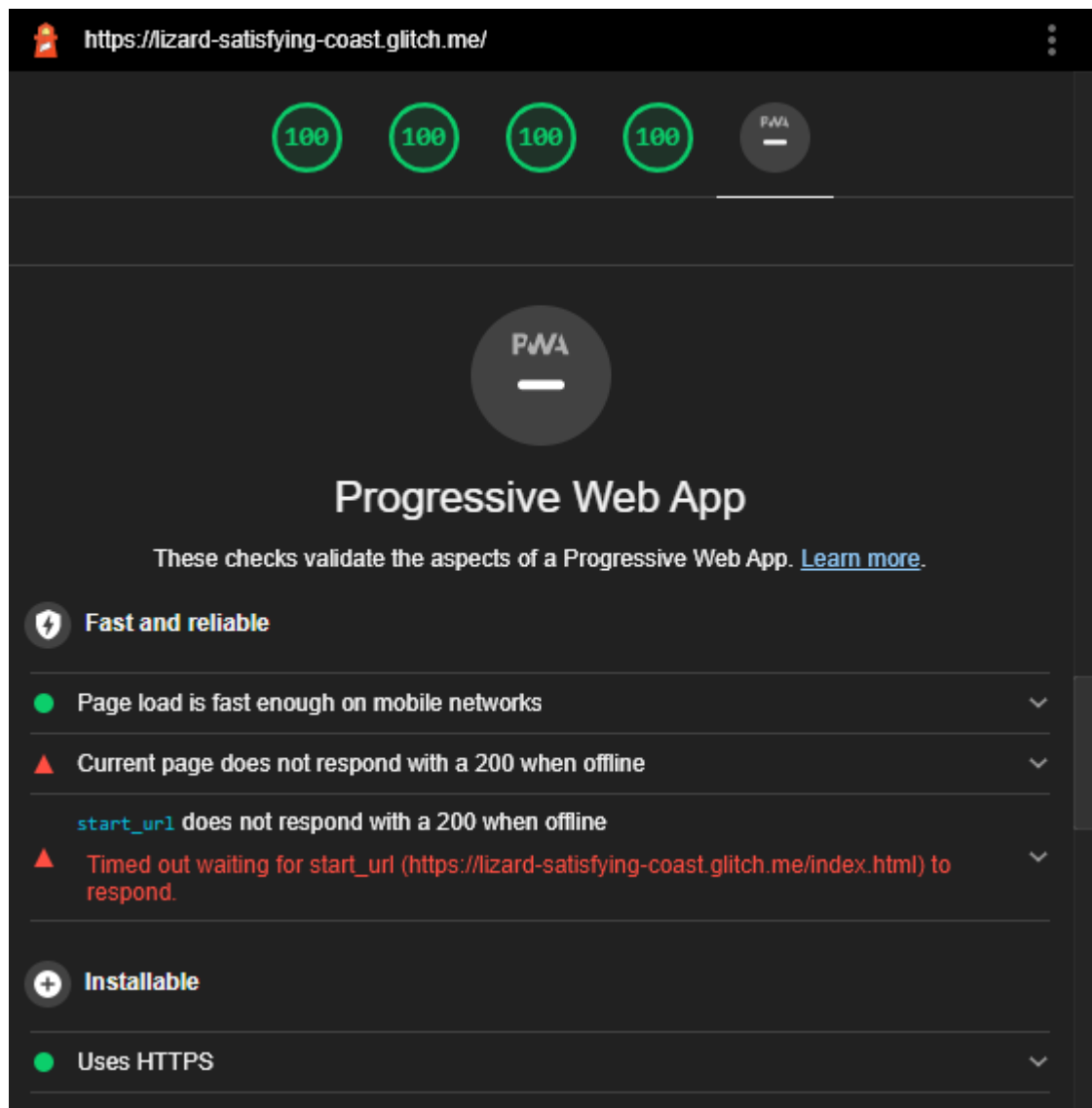
Andres Felipe Varon Maya 201626174

Parte 1:

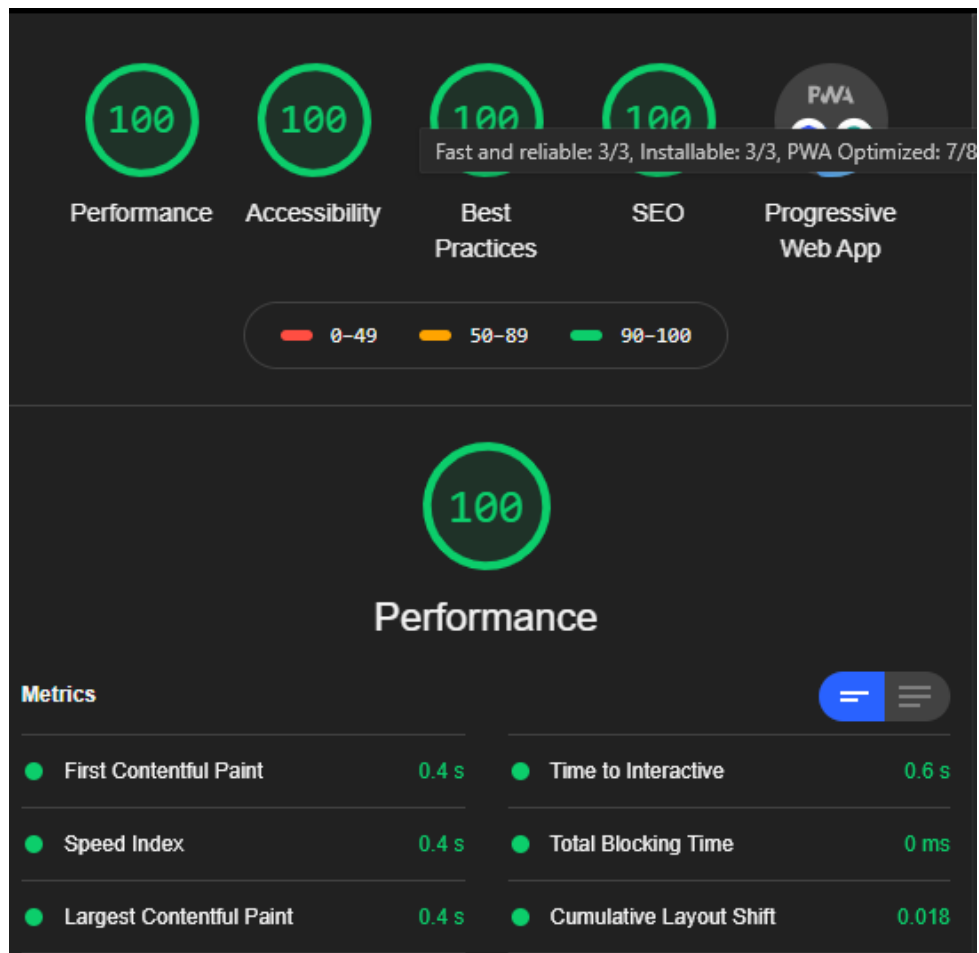
Versión Inicial:



Al agregar el manifest.json y los headers en html:



Con serviceworker configurado:



Al optimizar el cache en serviceworker.js



https://lizard-satisfying-coast.glitch.me/



Performance



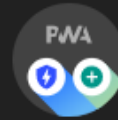
Accessibility



Best Practices



SEO



Progressive Web App

0-49

50-89

90-100



Performance

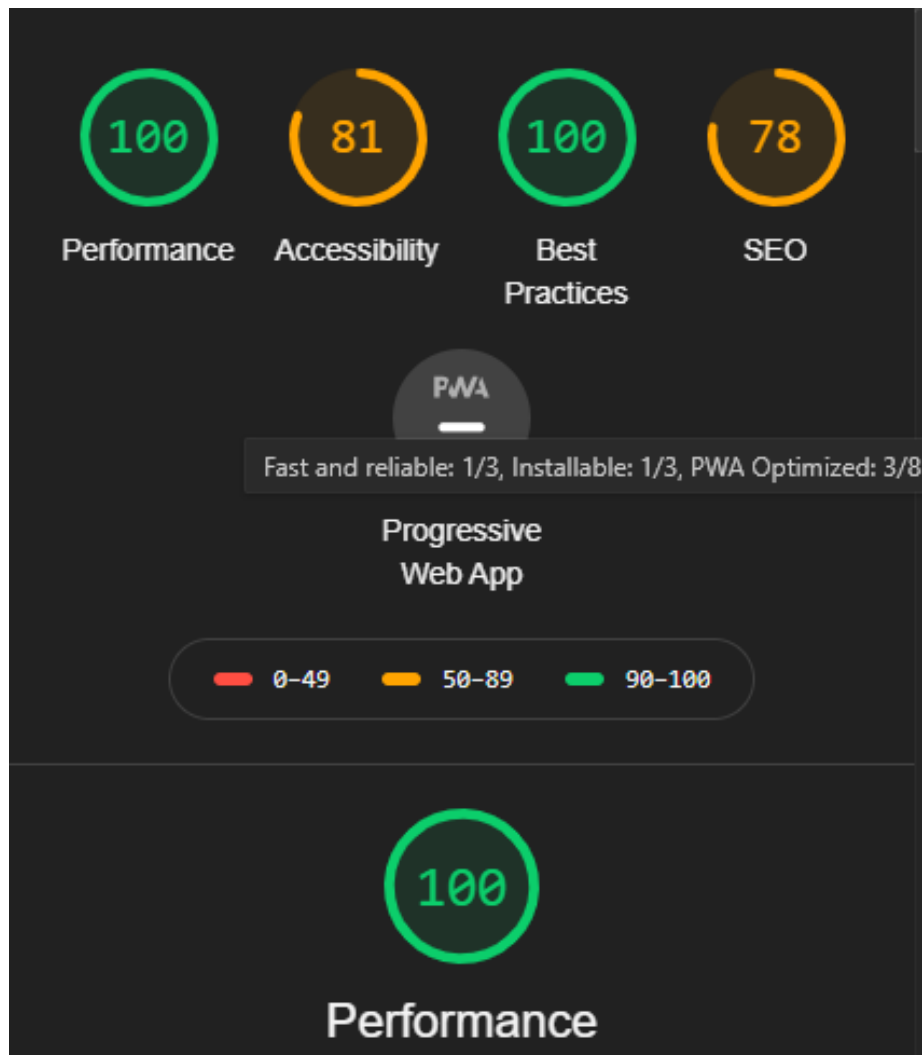
Metrics



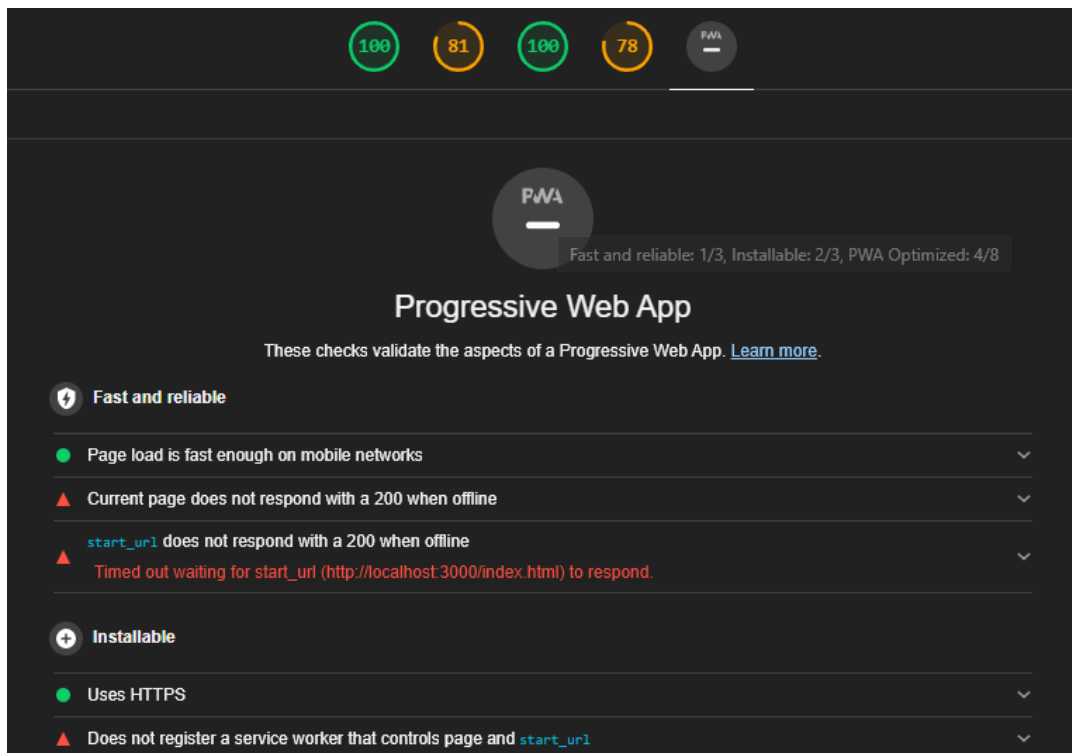
● First Contentful Paint	0.4 s	● Time to Interactive	0.7 s
● Speed Index	0.4 s	● Total Blocking Time	0 ms
● Largest Contentful Paint	0.4 s	● Cumulative Layout Shift	0.018

Parte 2:

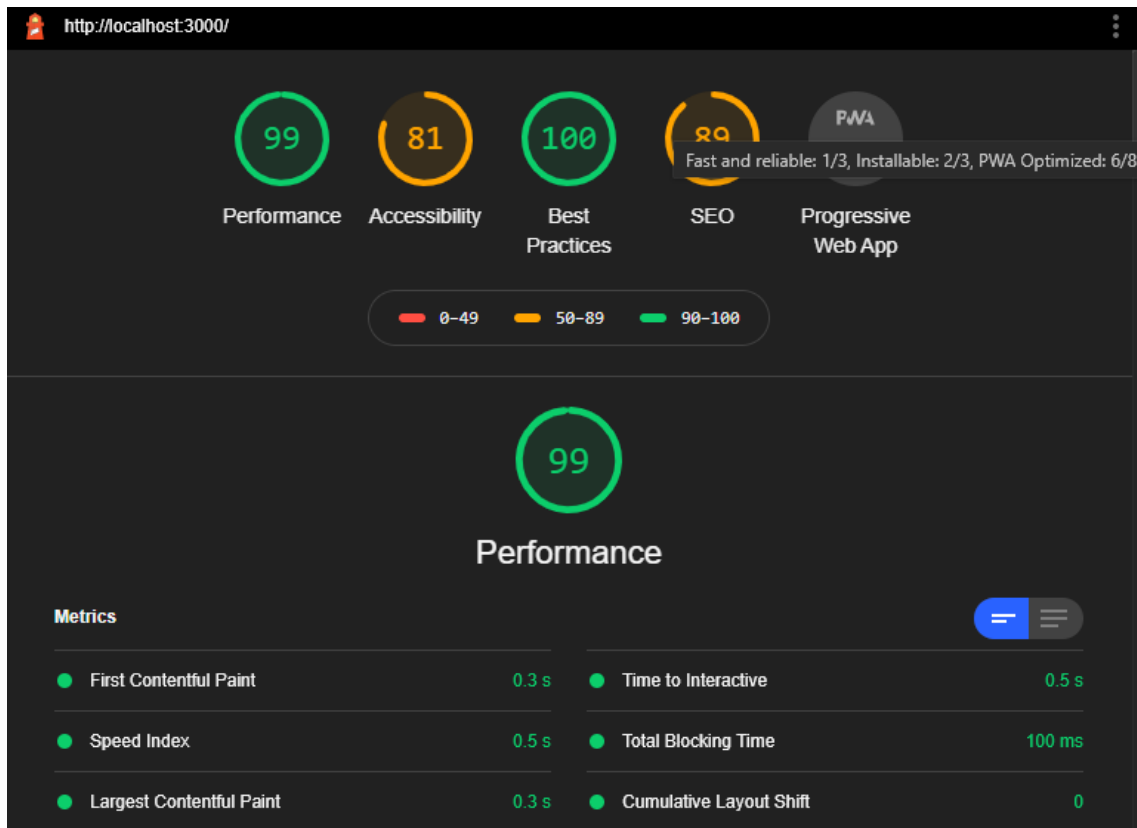
Inicial:



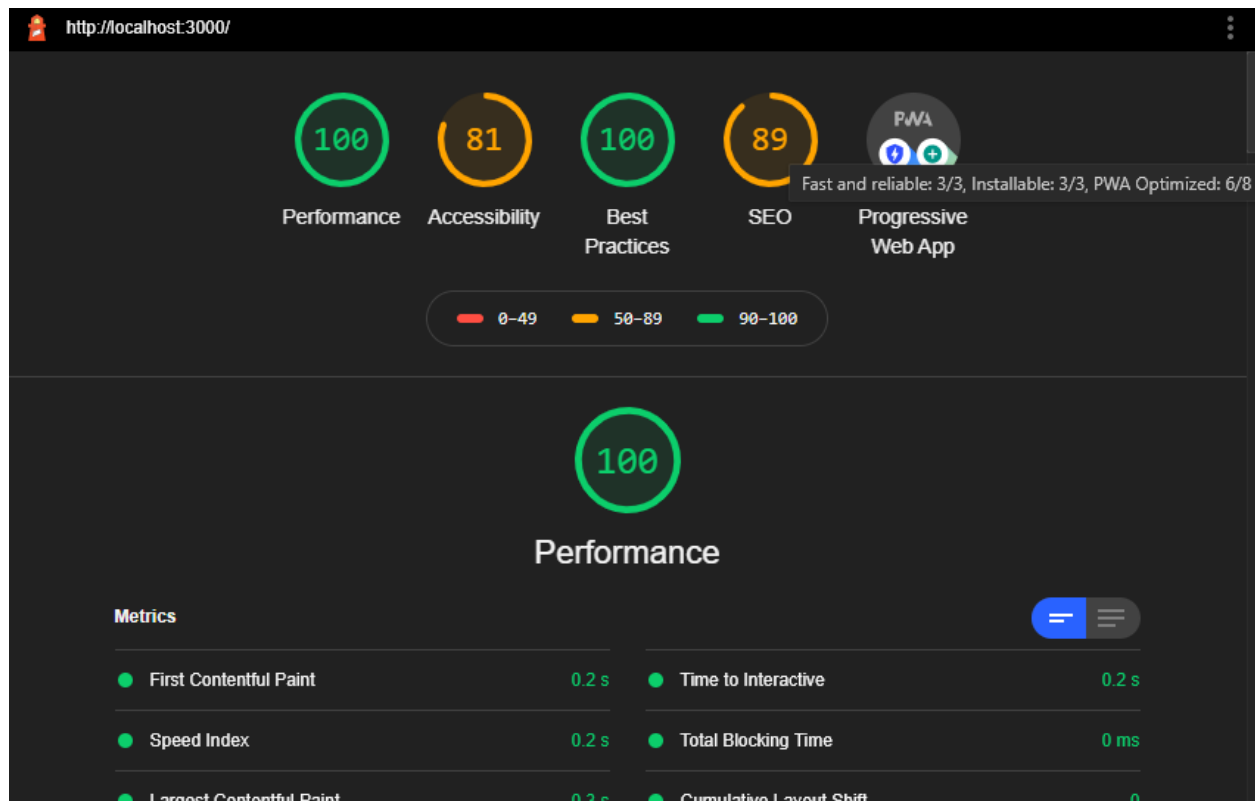
Al agregar el manifest y los iconos de Android:



Al agregar los meta tags de ios, su icono y tags de descripción y theme:



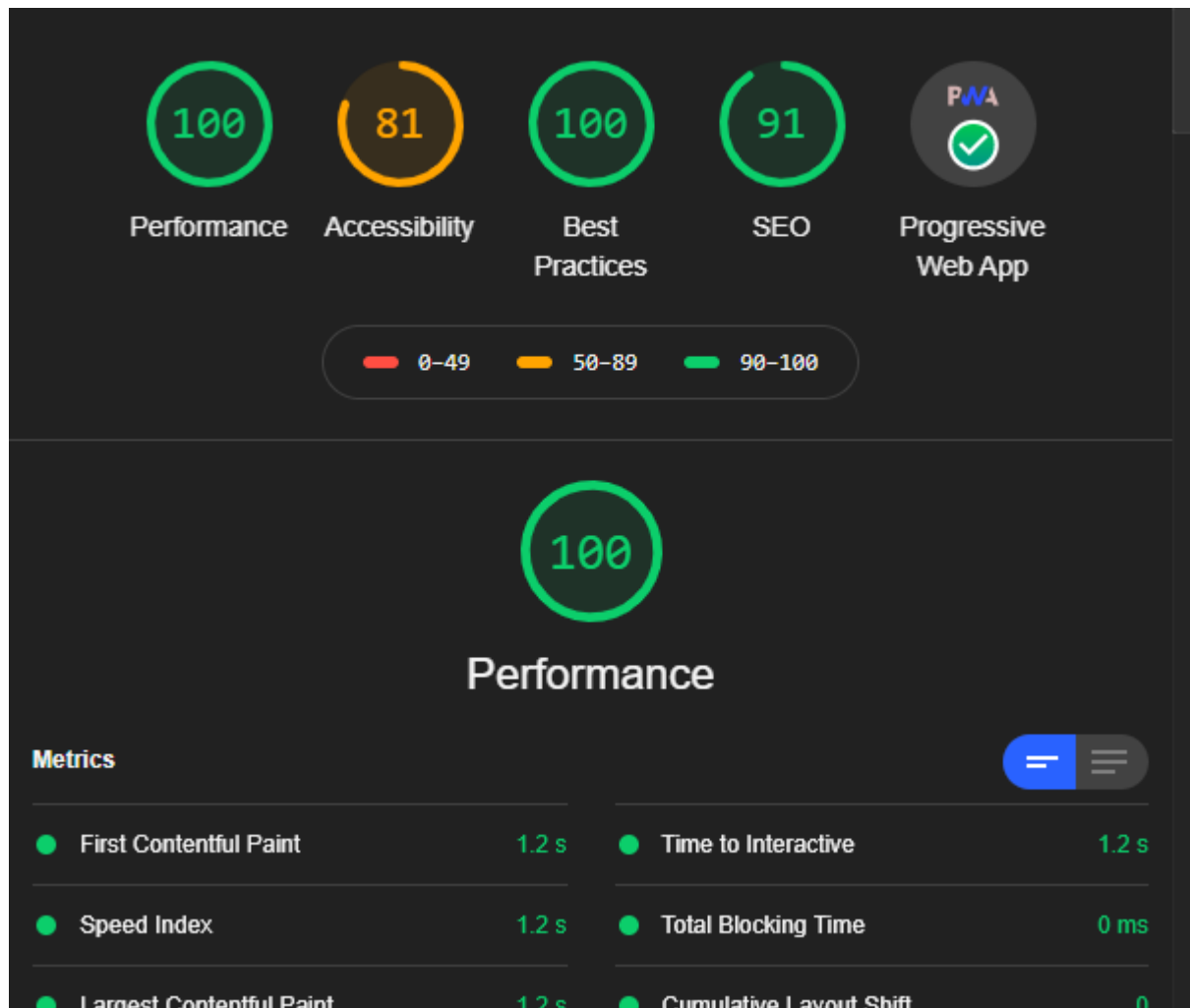
Al configurar el worker con workbox para webpack:



Luego se le agregó `StaleWhileRevalidate` a los urls del api lo que permite hacer exactamente lo que se necesitaba:

The **stale-while-revalidate** pattern allows you to respond to the request as quickly as possible with a cached response if available, falling back to the network request if it's not cached. The network request is then used to update the cache. May 7, 2020

Luego se le agregó el icono maskable y se agregó el paquete express-http-to-https y se desplegó en glitch:



Finalmente se utilizó idb para configurar el indexeddb que guardara los datos al crear y al borrar en el navegador.

