
PWA

Web Apps optimizadas para Mobile



¿Qué es una PWA?

Progressive Web App

Es una definición variable



¿Cómo es la web hoy?

Spoiler Alert:

No funciona muy bien en mobile



Mobile First

Más del 50% de nuestros usuarios
está en mobile



Malas Conexiones

Conexiones 3G y LTE no son
particularmente confiables



Sitios muy lentos

Un sitio promedio tarda
14 segundos en cargar



Una mala UX es carísima

Y no sólo tiene que ver con
el diseño de nuestra app



50% de los usuarios se van de un sitio que tarda más de 3 segundos en cargar



**Cada segundo de demora
nos cuesta un 5-10% de
nuestras ventas**



Historias de Éxito

pwastats.com

wpostats.com



Tres Requisitos

En qué nos enfocamos
para hablar de PWA



Performance

¡Mobile First!

Responsive, con foco en UX



Instalable en Home Screen

Web Manifest
Iconos para App



Soporte Offline

Soportar malas conexiones
Confiabilidad en todo momento



Ventajas

Mejor experiencia de usuario

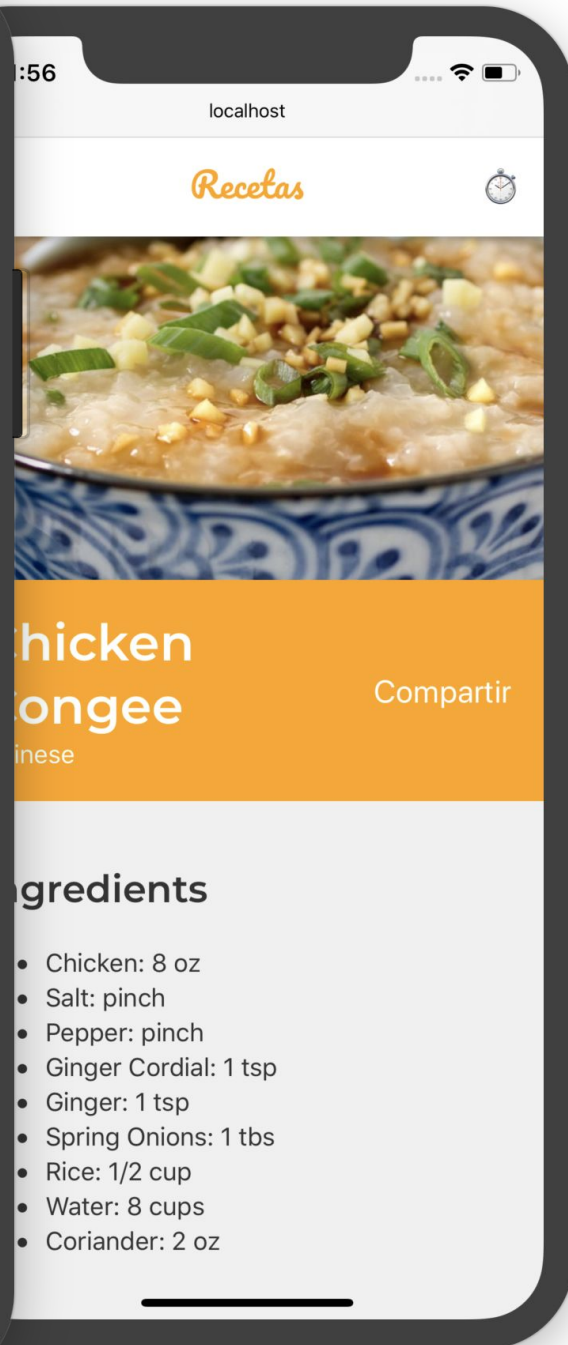
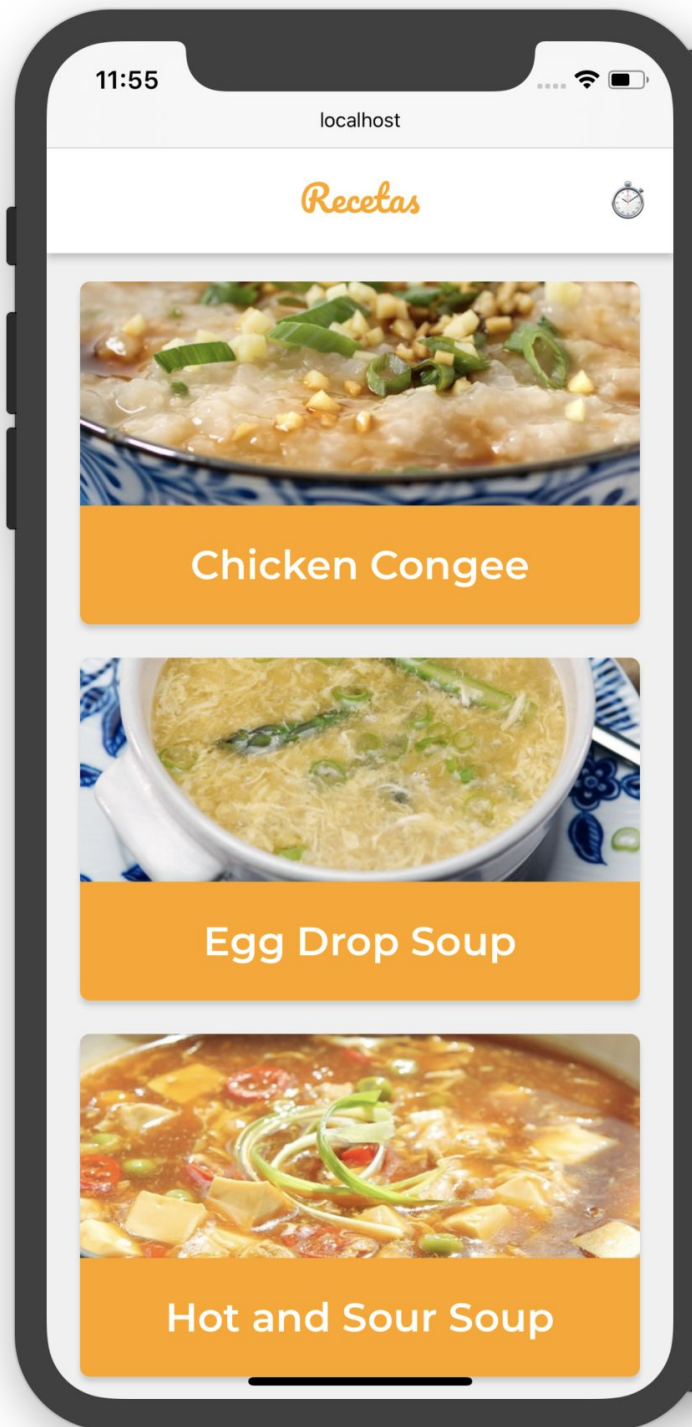
Estar en la Home Screen

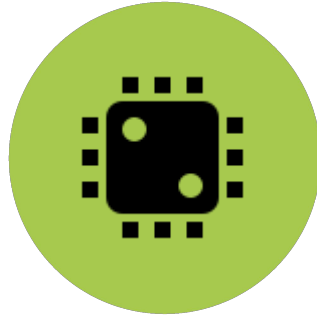
No hay App Stores



Platzi Recetas

PWA para Ver Recetas
Utiliza la API de TheMealDB





El Stack

Create React App
React + React Router



Google Lighthouse

Diagnóstico de Web App

Checklist de PWA



Time To First Paint

¿Cuánto tengo que esperar
para empezar a ver el sitio?



¿Cómo bajar el TTFMP?

Server Side Rendering

Menos HTML y CSS

Servers más rápidos



Time To Interactive

¿Cuánto hay que esperar para interactuar con la app?



¿Cómo bajar TTI?

Reducir la cantidad de JS

Reducir el tiempo de respuesta



Web Manifest

Activa la funcionalidad de
Add to Home Screen



Voice Memos



Record something!



Voice Memos

voice-memos.appspot.com



ADD TO HOME SCREEN





¡Importante!

A partir de Chrome 68
cambia el comportamiento



5:16



Voice Memos



Record something!



Add Airhorner to Home screen



```
{  
  "short_name": "Recetas",  
  "name": "Recetas",  
  "start_url": "/",  
  "scope": "/",  
  "display": "standalone",  
  "icons": [...],  
  "background_color": "#FFA500",  
  "theme_color": "#FFA500"  
}
```



¿Qué pasa con iOS?

Safari es el nuevo
Internet Explorer

<!-- El icono -->

`<link rel="apple-touch-icon" href="/icon.png" />`

<!-- El nombre de la app -->

`<meta name="apple-mobile-web-app-title"
content="Recetas" />`

<!-- ¿Puede correr fullscreen? -->

`<meta name="apple-mobile-web-app-capable"
content="yes"/>`

<!-- ¿Qué diseño de status bar queremos? -->

`<meta name="apple-mobile-web-app-status-bar-style"
content="black-translucent" />`



Detalles de UX

La UI es Fullscreen

Siempre brindar navegación



¿Qué hacemos con iOS?

No hay Notificaciones

No hay aviso para home screen

La app se reinicia en cada carga


```
<!-- ￣\_ (ツ) _/  -->
```

```
<meta
```

```
name="apple-mobile-web-app-capable"
```

```
content="no" />
```



¿Cómo Diferir el Banner?

Podemos lanzar el banner
en el momento ideal

```
window.on('beforeinstallprompt',  
  function(e) {  
    // Mostramos el prompt  
    e.prompt()  
  }  
)
```

```
var eventoAddToHome
```

```
window.on('beforeinstallprompt',  
  function(e) {  
    eventoAddToHome = e  
  })
```

```
// Más tarde...
```

```
eventoAddToHome.prompt()
```



Service Workers

Soporte Offline

Proxy In-Browser



Service Workers

¿Cómo implementarlos
en nuestra aplicación?



SW-Precache

Viene con Create React App
Precarga assets automáticamente



SW-Precache

Pero si tenemos algo más complejo
que un one pager no sirve



**¿Debería escribir mi propio
Service Worker?**

NOOOOOOOOOOO



Workbox

Librería de Service Workers

Estrategias de Red

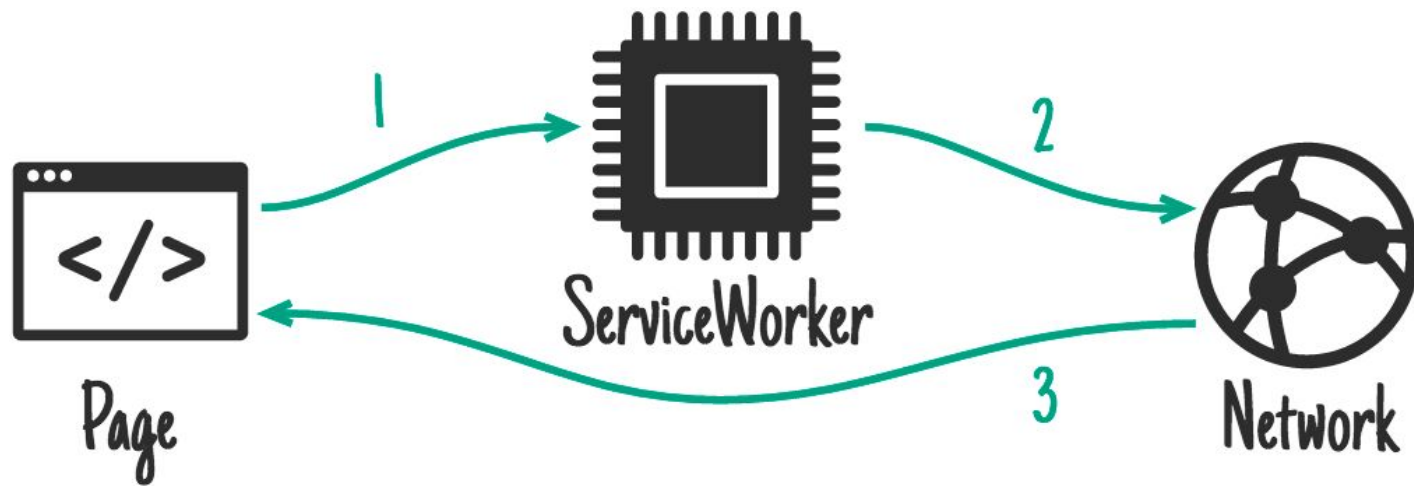
Google Analytics Offline



Estrategias de Carga

¿Cómo queremos que cargue nuestra aplicación?

Network Only

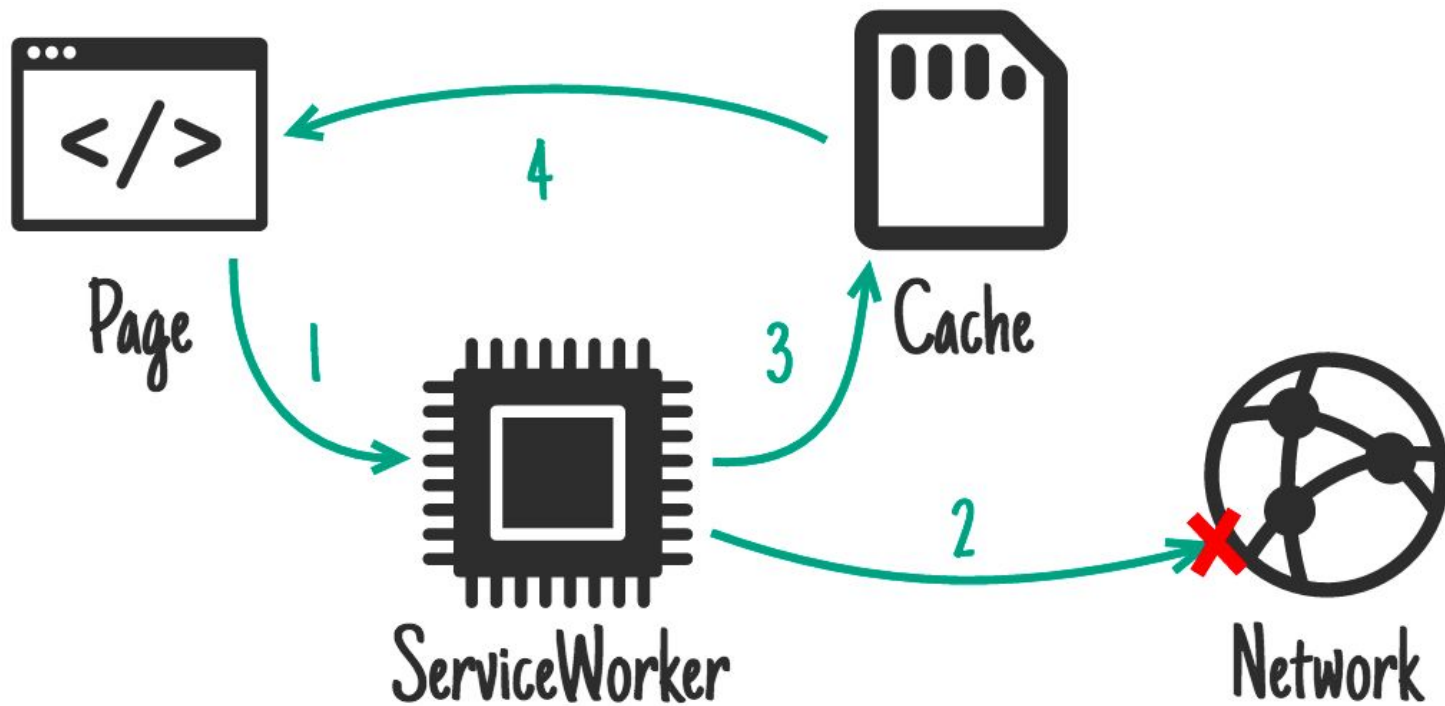




¿Cuándo usar Network Only?

Por defecto si no queremos cache
Información en tiempo real

Network First





¿Cuándo usar Network First?

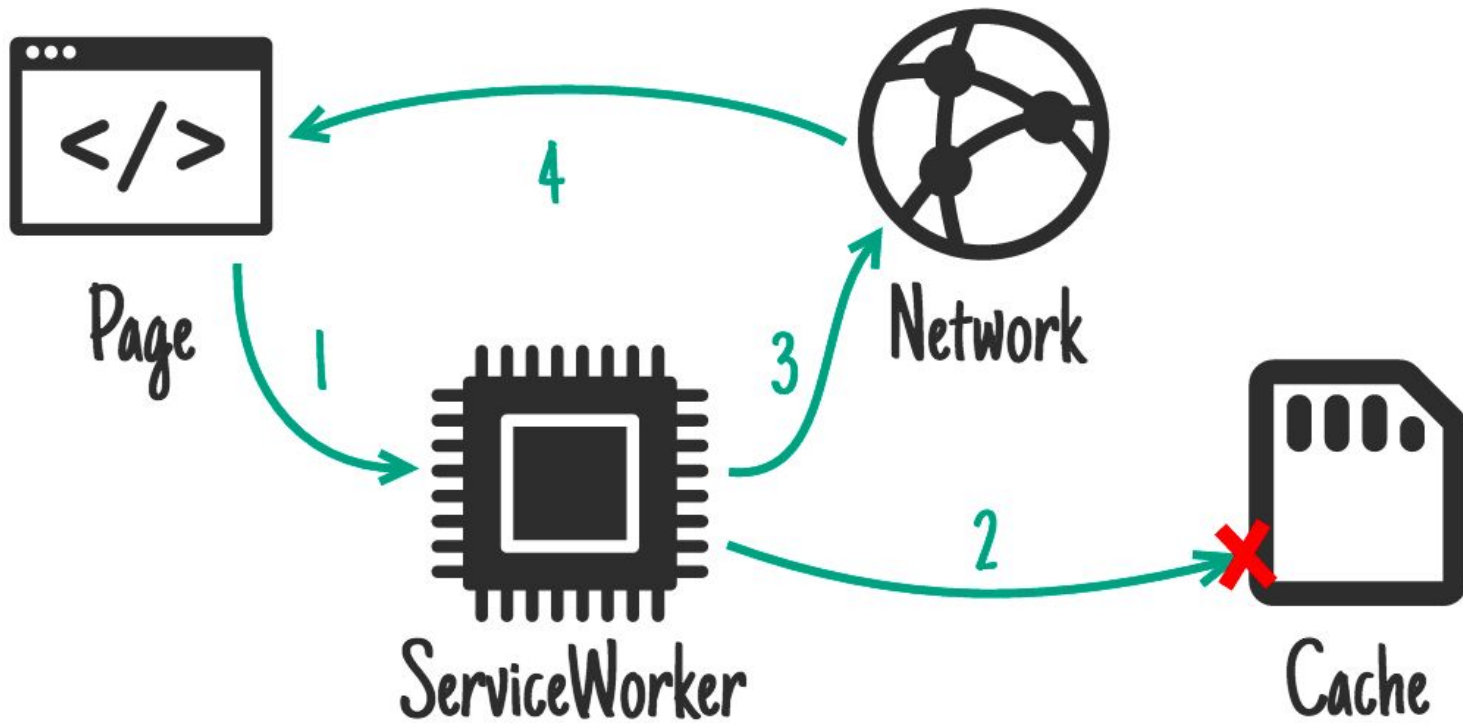
Cuando queremos la última versión
Y tener soporte offline



Estrategias de Carga

¿Cómo queremos que cargue nuestra aplicación?

Cache First

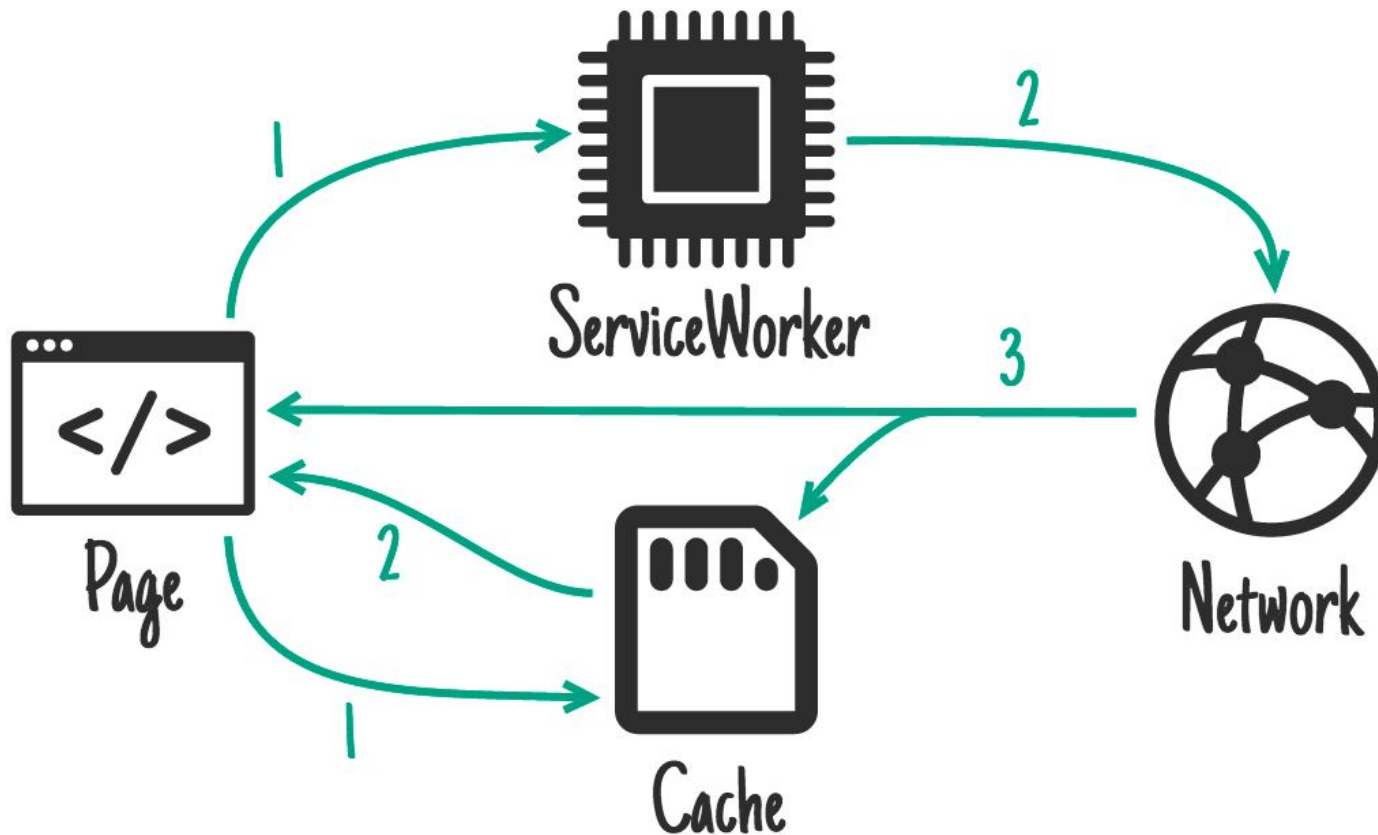




¿Cuándo usar Cache First?

Cuando queremos máxima velocidad
y es un recurso que no cambia
(Fuentes, Imágenes, Estáticos)

Stale While Revalidate





¿Cuándo usar Stale & Revalidate?

Cuando queremos mucha velocidad
Y es un recurso que puede estar
levemente desactualizado



¡Sólo funciona con HTTPS!

Podemos conseguir certificados
gratis con Let's Encrypt



Google Analytics

Implementando Analytics
para usuarios offline



Reto: Workbox

Probemos estrategias de carga
en nuestra aplicación

1. Imágenes con Cache First
2. Vencimiento de 1 semana
3. Probar Plugins y Estrategias



Web Share API

Utilizar el Share Nativo
en Android



My recording #60



UNPUBLISH RECORDING

Share via

Tap an icon below to share your content directly.



+39 339 247
3698



omarmizdaq



damiandizeo



romain



Gmail



Message+



Add to Firefox



Slack



WhatsApp



Facebook



ProtonMail



Add to Dropbox



```
link.on('click', function(e) {  
    e.preventDefault()  
    if( ! navigator.share ) return  
  
    navigator.share({  
        title, text, url  
    })  
})
```



¡Requiere HTTPS!

Como casi todas las features,
Web Share requiere HTTPS



Comunicando Offline

Avisando a nuestros usuarios
que no tenemos conexión

```
if( navigator.onLine ) {  
    // Estamos online  
} else {  
    // Estamos offline  
}
```

```
// No hagas esto
setInterval(function(e) {
    if( navigator.onLine ) {
        // Code
    }
}, 10)
```

```
window.addEventListener('online',  
  function(e) {  
    // Estamos online  
  })
```

```
window.addEventListener('offline',  
  function(e) {  
    // Estamos offline  
  })
```




Notificaciones

Enviando Notificaciones
en nuestra Web App



UX de Notificaciones

Hay que asegurarse de que el usuario quiere recibirlas

```
// Pedimos permiso al usuario  
await Notification.requestPermission()  
  
if( Notification.permission === 'granted' ) {  
    // Podemos enviar notificaciones  
}
```

```
if( ('Notification' in window) &&  
    ('serviceWorker' in navigator) ) {
```

```
    // Tenemos Notificaciones
```

```
}
```

```
// En Android se envían por ServiceWorker  
var sw = navigator.serviceWorker  
var registration = await sw.getRegistration()
```

```
registration.showNotification("Título", {  
  body: "Descripción",  
  img: "/icon.png"  
})
```



¡Requiere HTTPS!

Notificaciones también
requiere HTTPS



Publicando en Github

Algunos consejos y tips
para mejorar tu portfolio



¡El Readme!

Incluir información del proyecto

Cómo levantar entornos

Quién lo hizo



¡A shipear!

Pongamos el sitio en producción



now

Nos permite publicar nuestra
aplicación con un solo comando



Protip: Docker

Estandarizar Deployments
Curso Fundamentos de Docker



Documentación

developers.google.com

serviceworker.rs