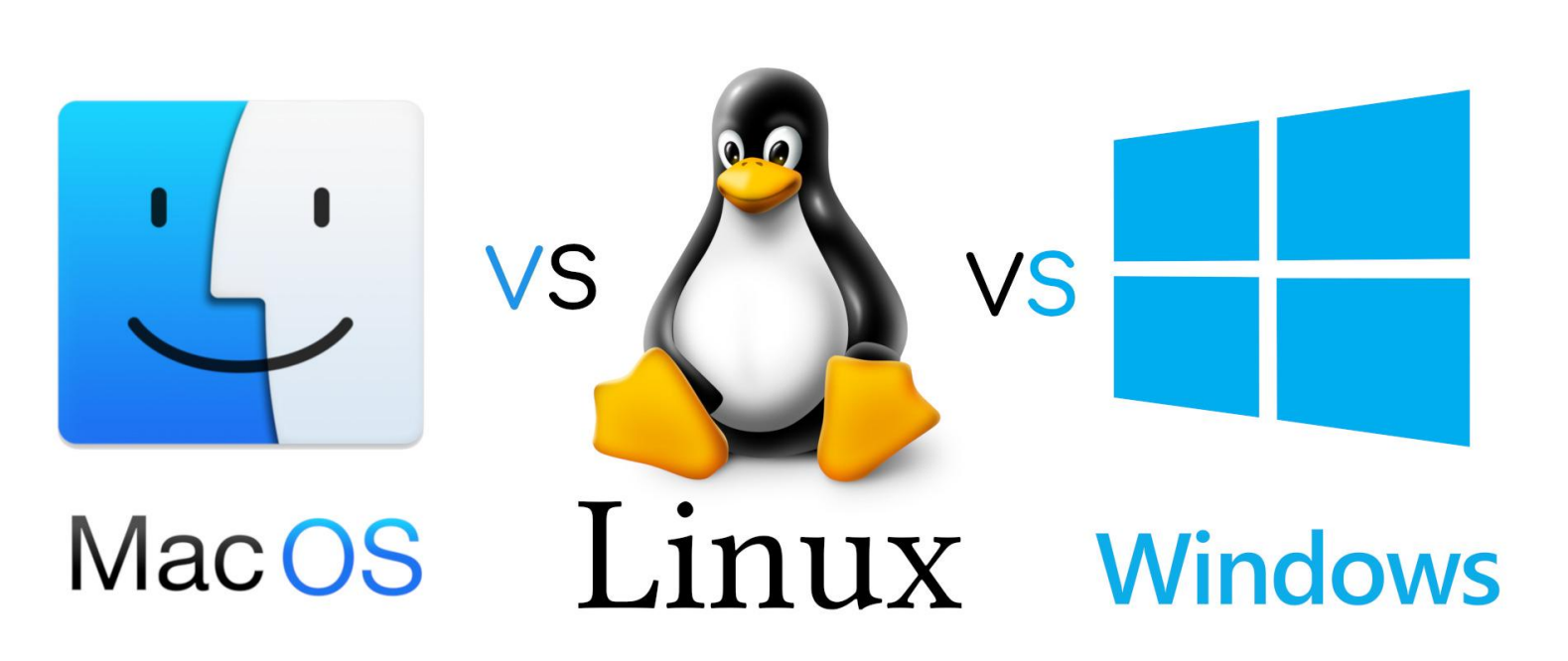
PWAs

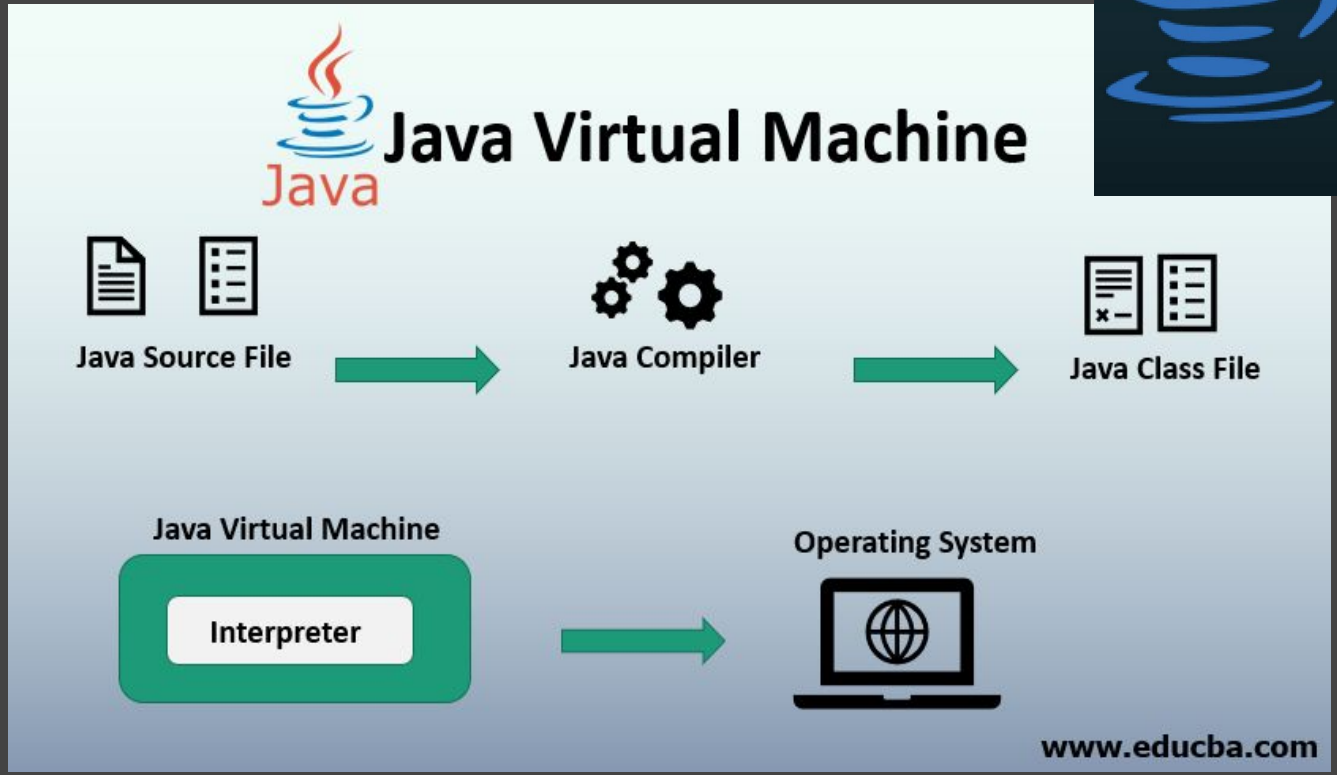
Aplicaciones Web Progresivas

CONTEXTO

Aplicaciones de Escritorio vs Web Apps.



Aplicaciones de Escritorio vs Web Apps.

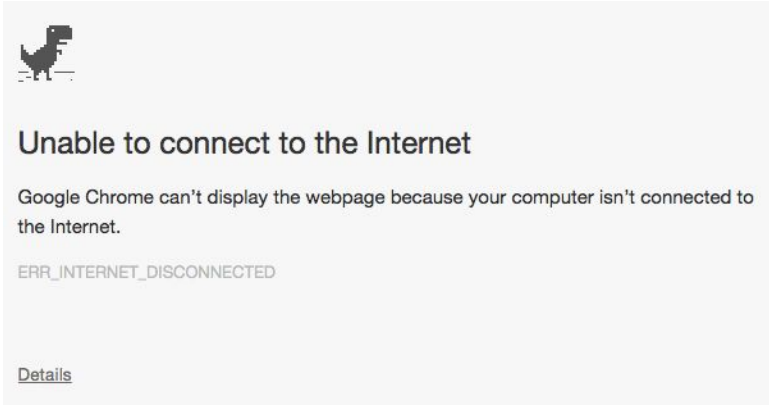


Aplicaciones de Escritorio vs Web Apps.

1. **Distribución**: Nada que descargar. Nada que instalar.
2. Se acabaron las **actualizaciones**.
3. Acceso a **Analytics.**
4. **Desarrollo más económico**: multiplataforma

Aplicaciones de Escritorio vs Web Apps.

1. ¿Siempre conectado?
2. Mala arquitectura (código web)
3. ¿Demasiadas tabs?



4. Necesidad de hosting (alojamiento)

Desarrollo móvil vs Web Apps

Nativo

Híbrido

Apache Cordova

Adobe PhoneGap

jQuery mobile

Flutter

NativeScript

Ionic

¿Y la web?

Desarrollo móvil vs Web Apps

¿Y la web?

- 52.2% Tráfico Web vía Móvil
- Experiencia de usuario lenta e incómoda
- 90% usuarios no vuelve a una web debido a un rendimiento pobre.

Google Empire

DEFINICIÓN

¿Qué es una PWA?

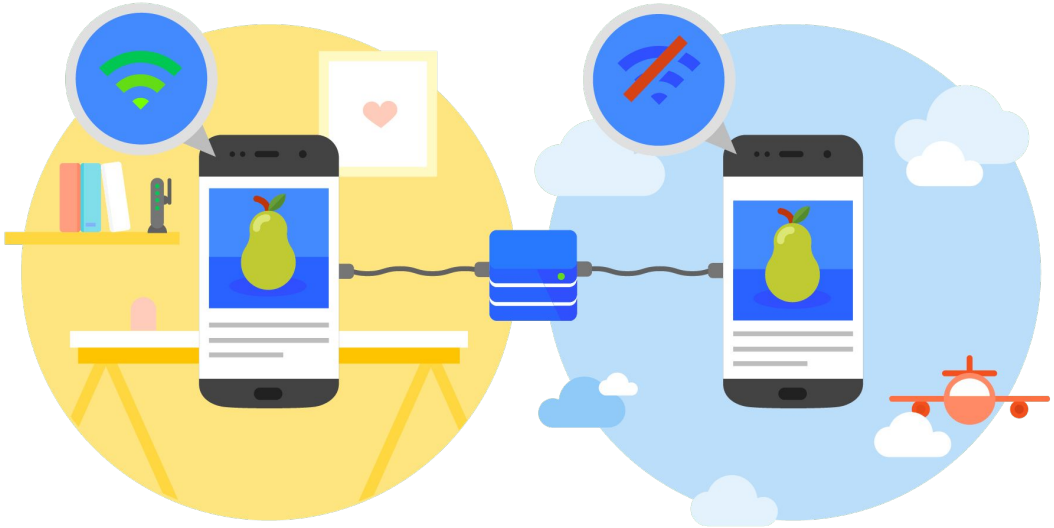
Una PWA es una web “vitaminada” que utiliza las últimas tecnologías disponibles en los “navegadores” para ofrecer una experiencia en móviles lo más parecida a la de una aplicación nativa.

¿Objetivos a perseguir por una PWA?

1. Reliable.
2. Fast.
3. Engaging.

¿Objetivos a perseguir por una PWA?

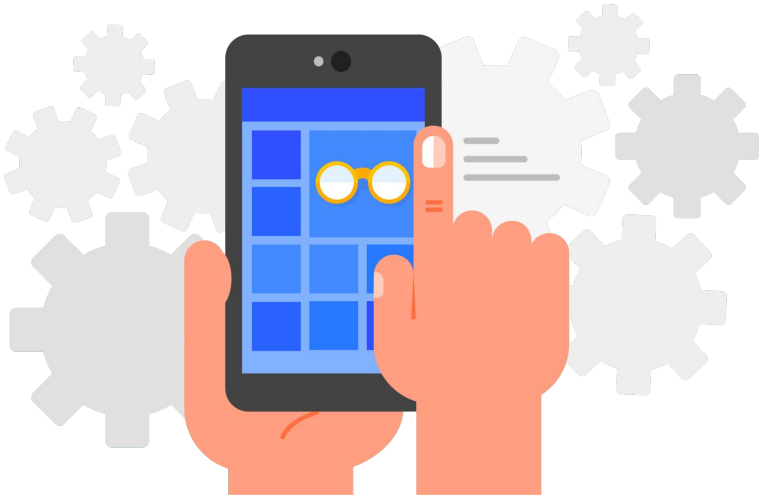
1 **Reliable → Fiable**



Nunca más el dinosaurio → Caching

2

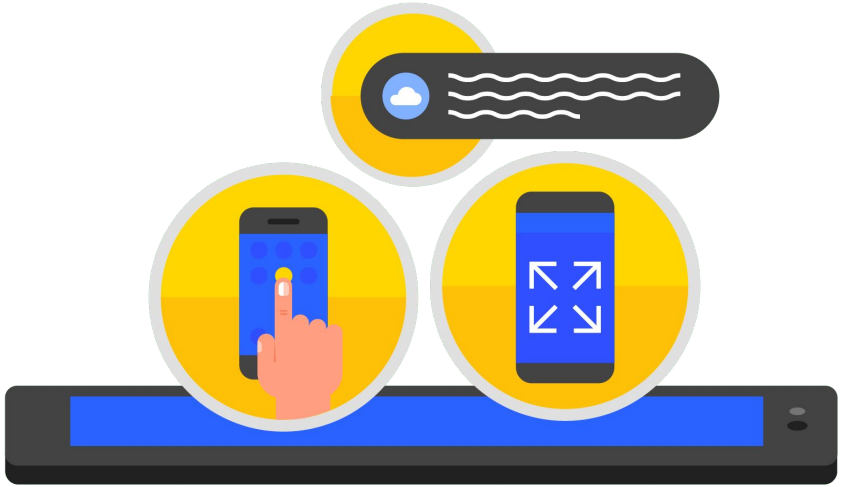
Fast → Rápida



Mejora UX → Mejora Volumen de Negocio

3

Engaging



Icono + Ventana + Notificaciones

¿Objetivos a perseguir por una PWA?

1. Mayor rendimiento y que **cargue** de manera casi **instantánea**.
2. Una **buena interfaz** que provoque una sensación similar a una app nativa.
3. La posibilidad de **trabajar sin conexión**.
4. Poder **enviar notificaciones** a los usuarios, como una app nativa.
5. **Instalable** y ejecutable como una app nativa.

CODIFICACIÓN

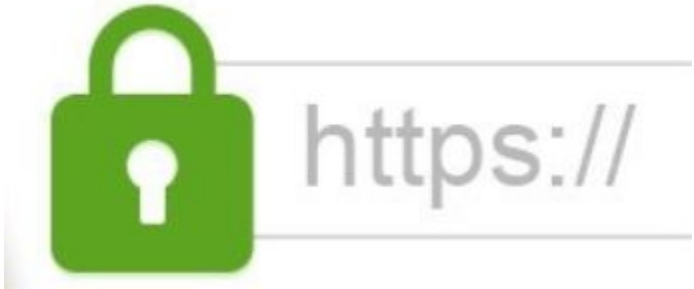
¡Empezamos lo bueno!

Punto de partida:

1. Protocolo HTTPS. (Obligatorio)
2. Responsive Design (Adaptativo). (Recomendable)
3. AppShell (Recomendable)

Punto de partida:

1 **Protocolo HTTPS.** **(Obligatorio)**



¿Por qué es obligatorio?

- **Mejora la seguridad de tu sitio.**
- **Es lo esperado por el usuario.**
- **Mejora rendimiento SEO.**
- **Disminuir soporte para HTTP.**

<< It's time to just be a good web citizen >>

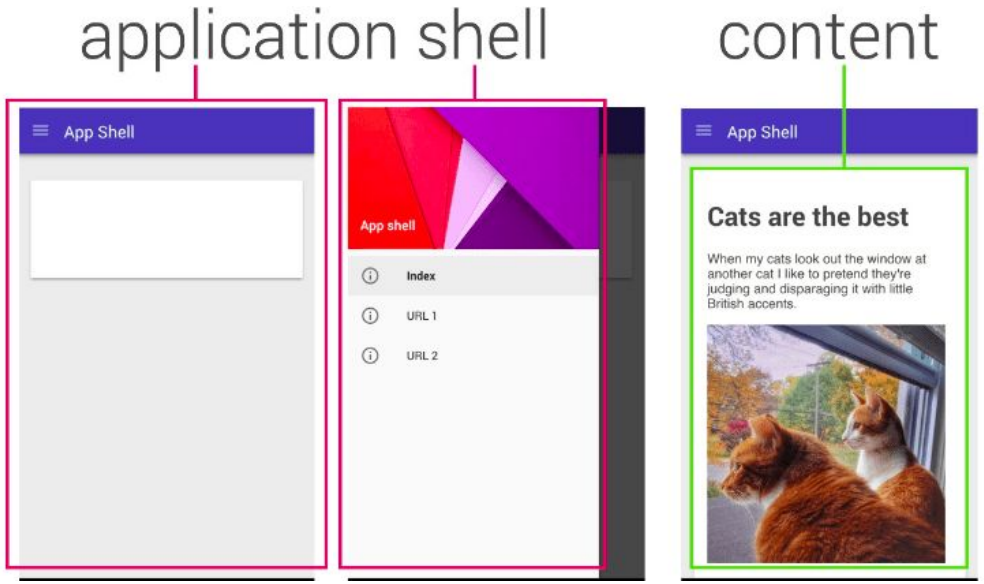
Punto de partida:

2 Responsive Design (Adaptativo). (Recomendable)

- CSS Grid y Flexbox.
- Bootstrap
- Etc.

Punto de partida:

3 **AppShell** **(Recomendable)**



Separar la aplicación entre funcionalidad y contenido y cargarlos por separado ;)

Archivos Necesarios:

index.html

style.css

app.js

manifest.json

serviceworker.js

Archivos Necesarios: manifest.json

Fichero **JSON** que contiene un objeto con una serie de **parámetros** para que el navegador obtenga **información de configuración**.

- Permite **integración** del sitio web con el navegador y con el sistema operativo.

- Mejora la **clasificación** en buscadores de la web.

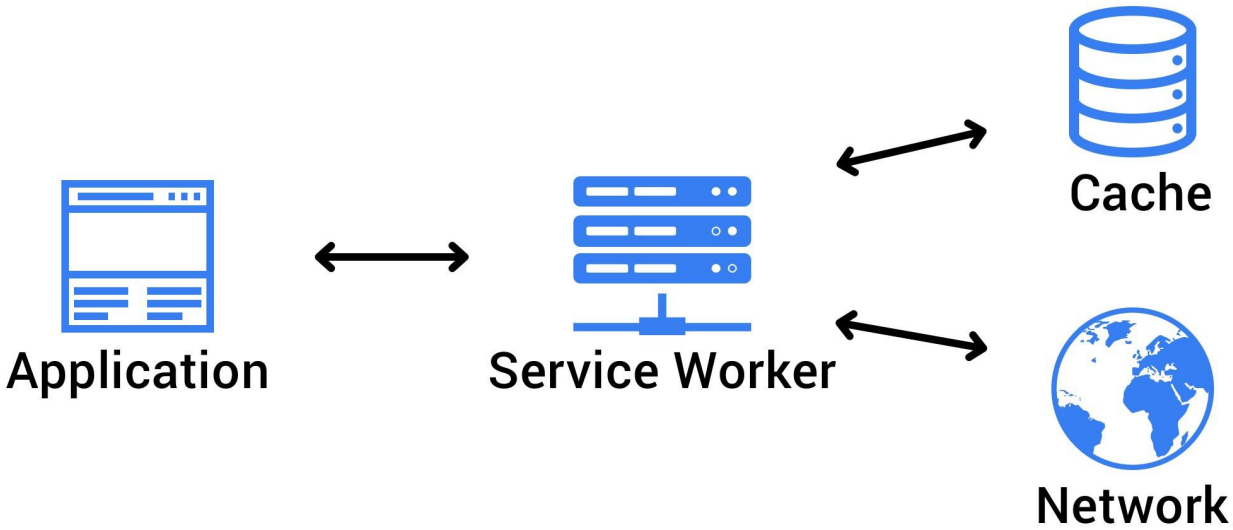
manifest.json

```
{
    "name": "PWAConferencia",          //Nombre de la app
    "short_name": "PWAConferencia",    //Nombre espacio limitado
    "start_url": ".",                  //Directorio base PWA
    "display": "standalone",           //Modo de visualización: fullscreen, standalone,
                                         Minimal-ui, browser
    "background_color": "#2A3443",     //Color mientras se carga la PWA.
    "description": "Una web PWA para demostrar sus capacidades",
    "theme_color": "#2A3443",          //Color navegador
    "icons": [
      {
        "src": "./img/icons/icon-512x512.png",
        "sizes": "512x512",
        "type": "image/png"
      }
    ]
}
```

Archivos Necesarios: serviceworker.js

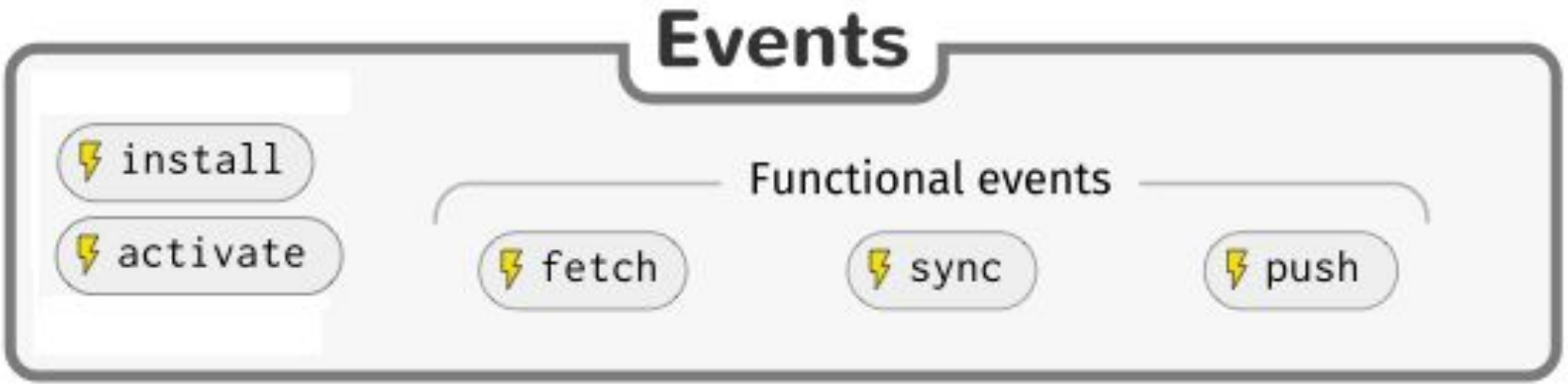
Actúa como un **proxy de red** ejecutándose en el navegador: **intercepta las peticiones HTTP** que salen de nuestro sitio web hacia la red y puede contestar con cualquier tipo de contenido.

Archivos Necesarios: serviceworker.js

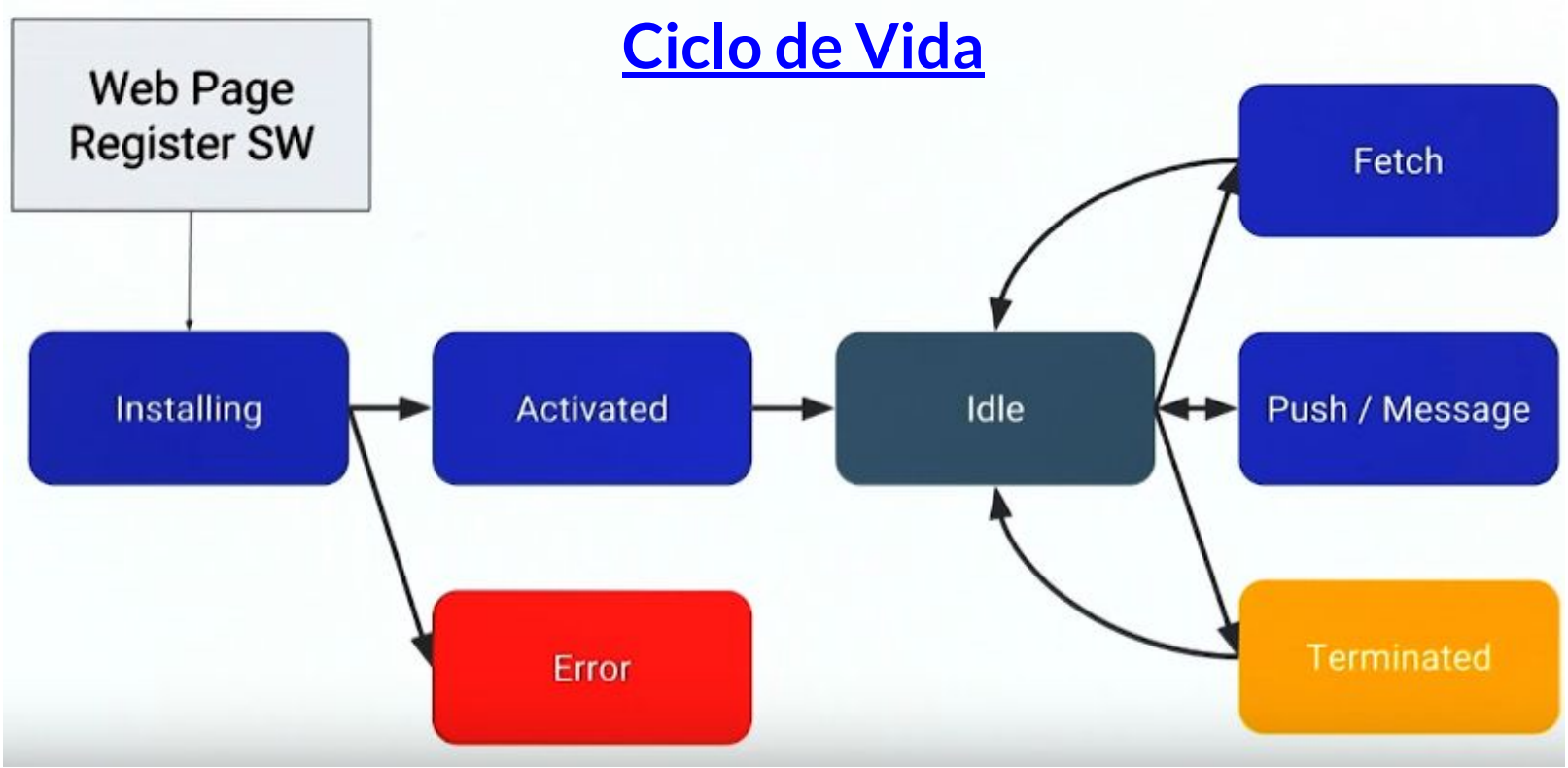


Archivos Necesarios: serviceworker.js

- Se ejecuta en **background**.
- Su ejecución es orientada a **eventos.**

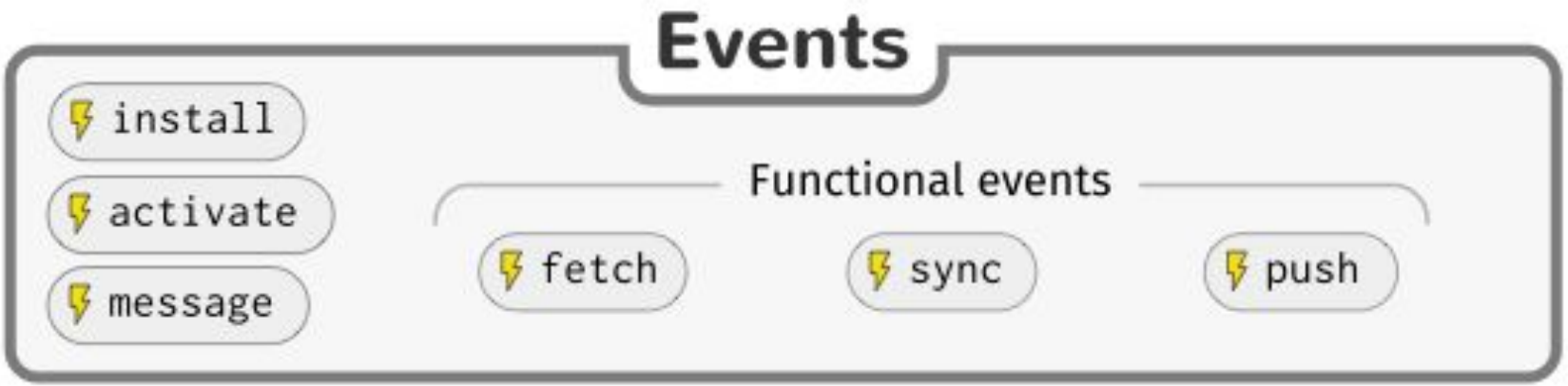


Archivos Necesarios: serviceworker.js



Archivos Necesarios: serviceworker.js

- Se ejecuta en **background**.
- Su ejecución es orientada a **eventos.**



Archivos Necesarios: serviceworker.js

INSTALL

- Se ejecuta tras la solicitud de registro.
- Penaliza el tiempo de ejecución la 1ª vez.
- Se realiza el “caching” de contenido estático.

Operaciones:
1. Abrir caché.
2. Guardar contenido.

Archivos Necesarios: serviceworker.js

ACTIVATED

IDLE

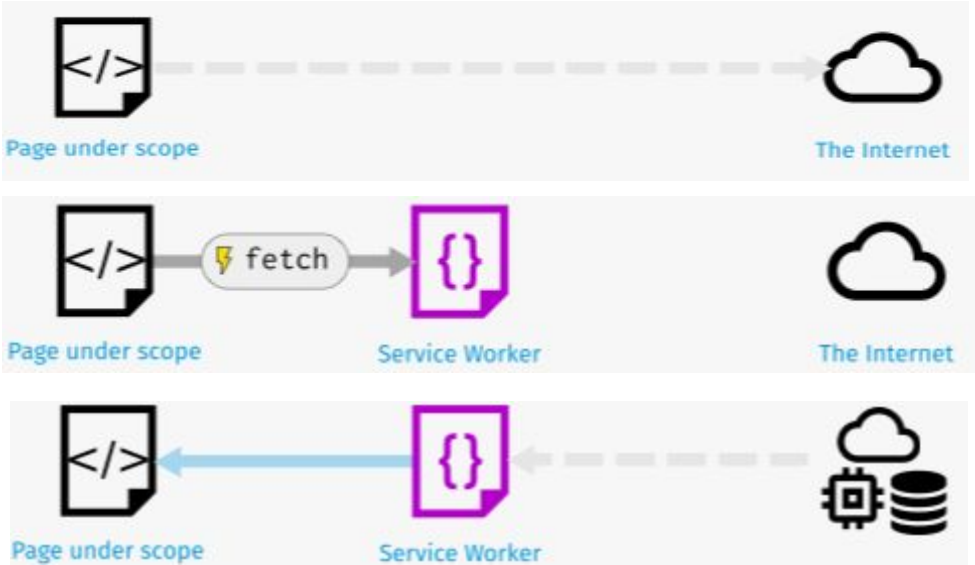
- Una vez instalada se procede a la activación.
- Se recomienda hacer modificaciones en caché (whitelist).

- El estado IDLE es un estado de escucha(ocioso.)

Archivos Necesarios: serviceworker.js

- Se intercepta todas la peticiones http.
- Se establece estrategia de cacheo dinámico.

FETCH



ESTRATEGIAS DE CACHEO

serviceworker.js

Buscar un balance entre:

Velocidad vs **Actualización** vs **Uso Datos**

- **Estrategias:** Network only, Caché only Caché first - network after (refresh), Network first - caché after (fallback), etc.

Archivos Necesarios: serviceworker.js

PUSH

- Se reciben y publican notificaciones push.

SYNC

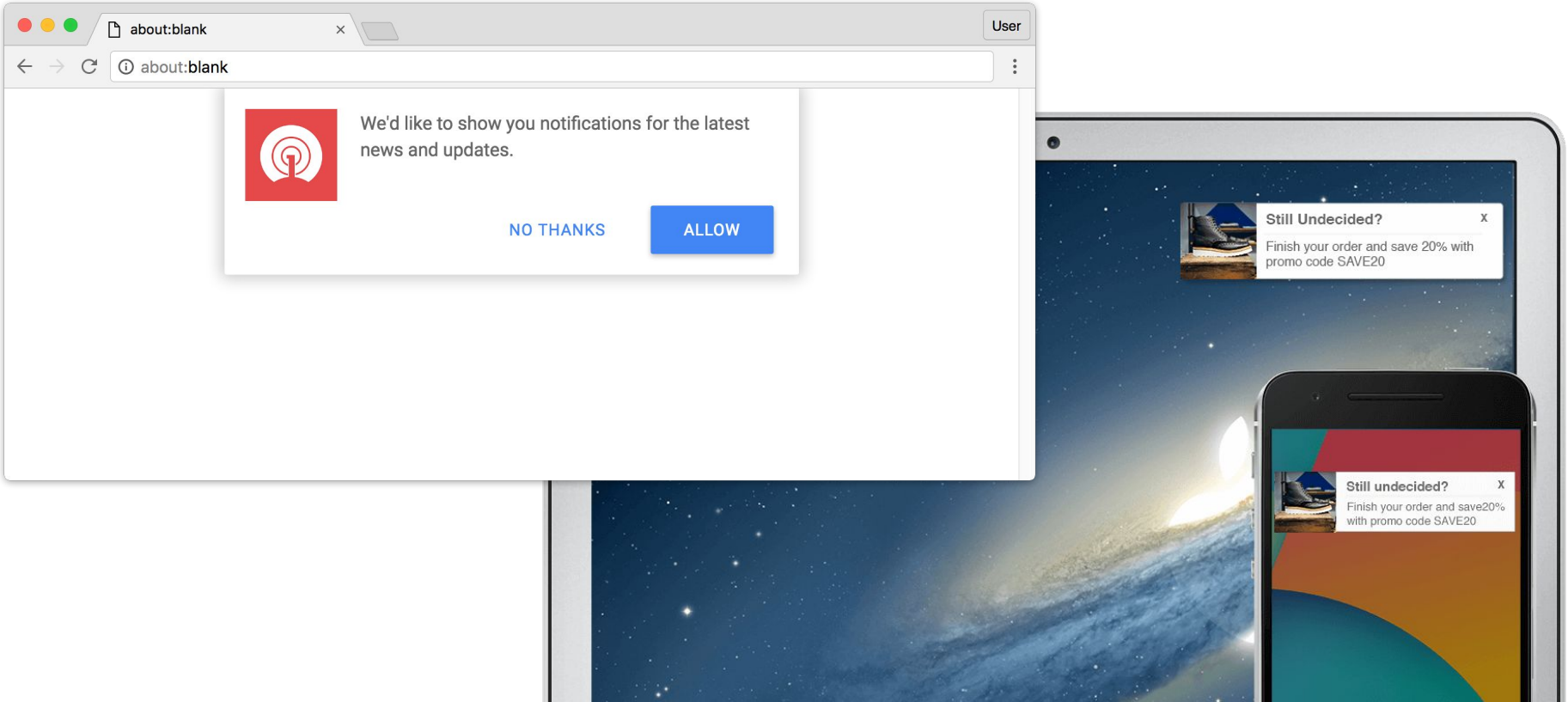
- Background (realiza operaciones suspendidas por mala conectividad) y Periodicas.

serviceworker.js

```
self.addEventListener('install', async event => {
  console.log('Install event');
});
self.addEventListener('activate', async event => {
  console.log('Activate event')
});
self.addEventListener('fetch', async event => {
  console.log('Fetch event')
});
self.addEventListener('sync', async event => {
  console.log('Sync event')
});
self.addEventListener('push', async event => {
  console.log('Push event')
});
```

PUSH NOTIFICATIONS

serviceworker.js



PUSH NOTIFICATIONS

serviceworker.js

2 tipos de notificaciones:

- **Local Notification:** Son generadas por la propia web.

- **Push Notification:** Son generadas por un servidor.

PUSH NOTIFICATIONS

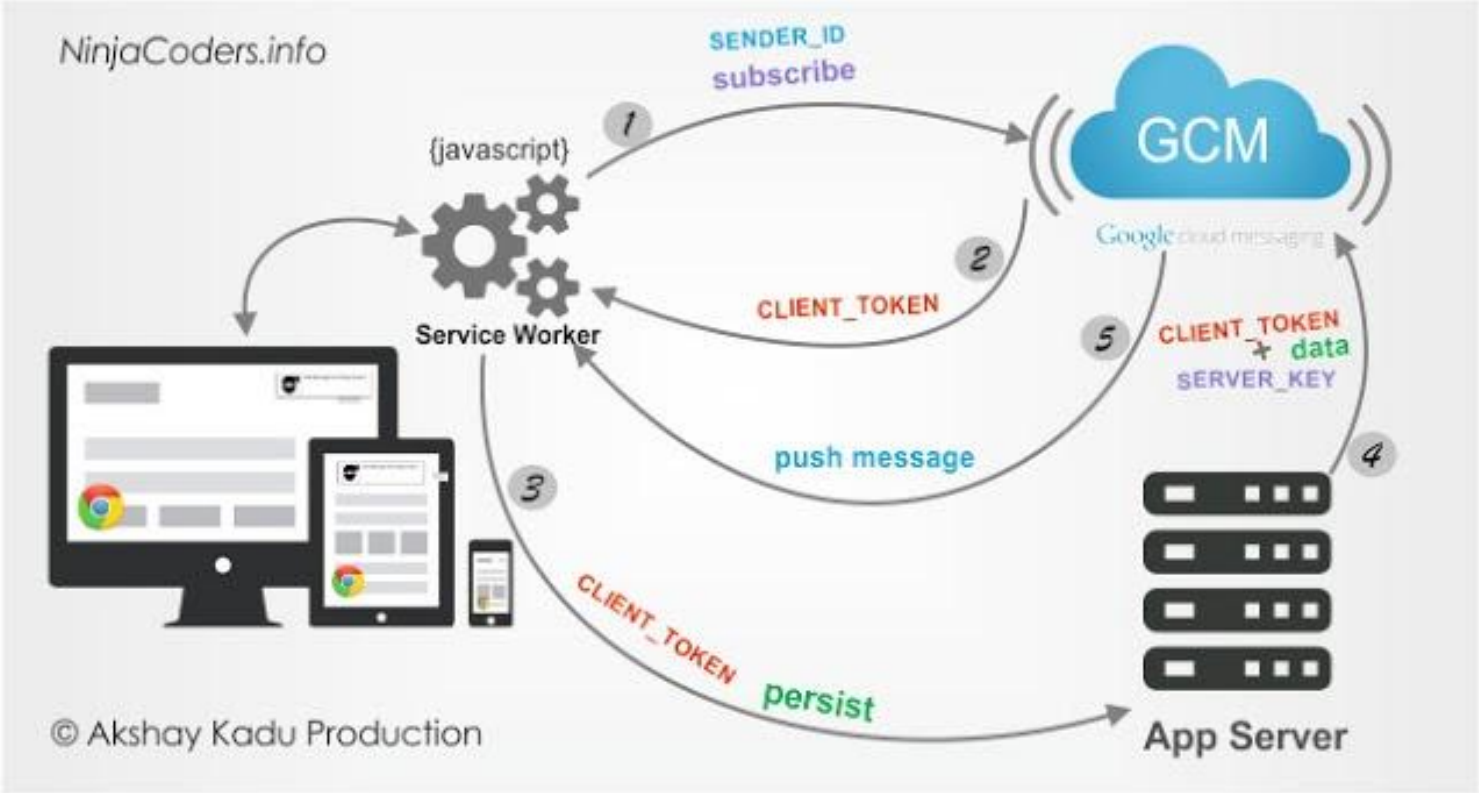
serviceworker.js

2 APIs a utilizar:

- **Notification API:** Se encarga de mostrar las notificaciones al usuario.

- **Push API:** Encargada de gestionar las notificaciones con el servidor.

PUSH NOTIFICATIONS serviceworker.js



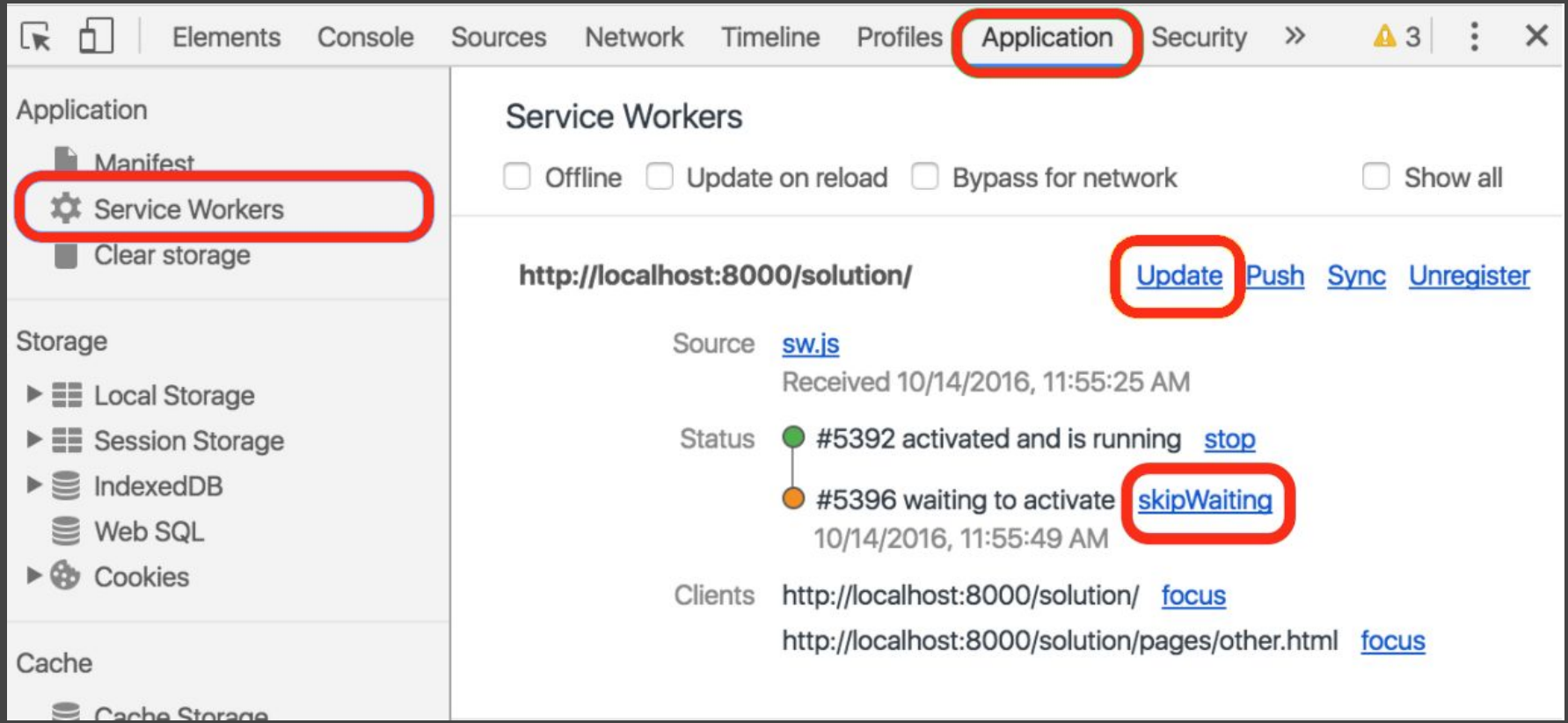
PUSH NOTIFICATIONS

serviceworker.js

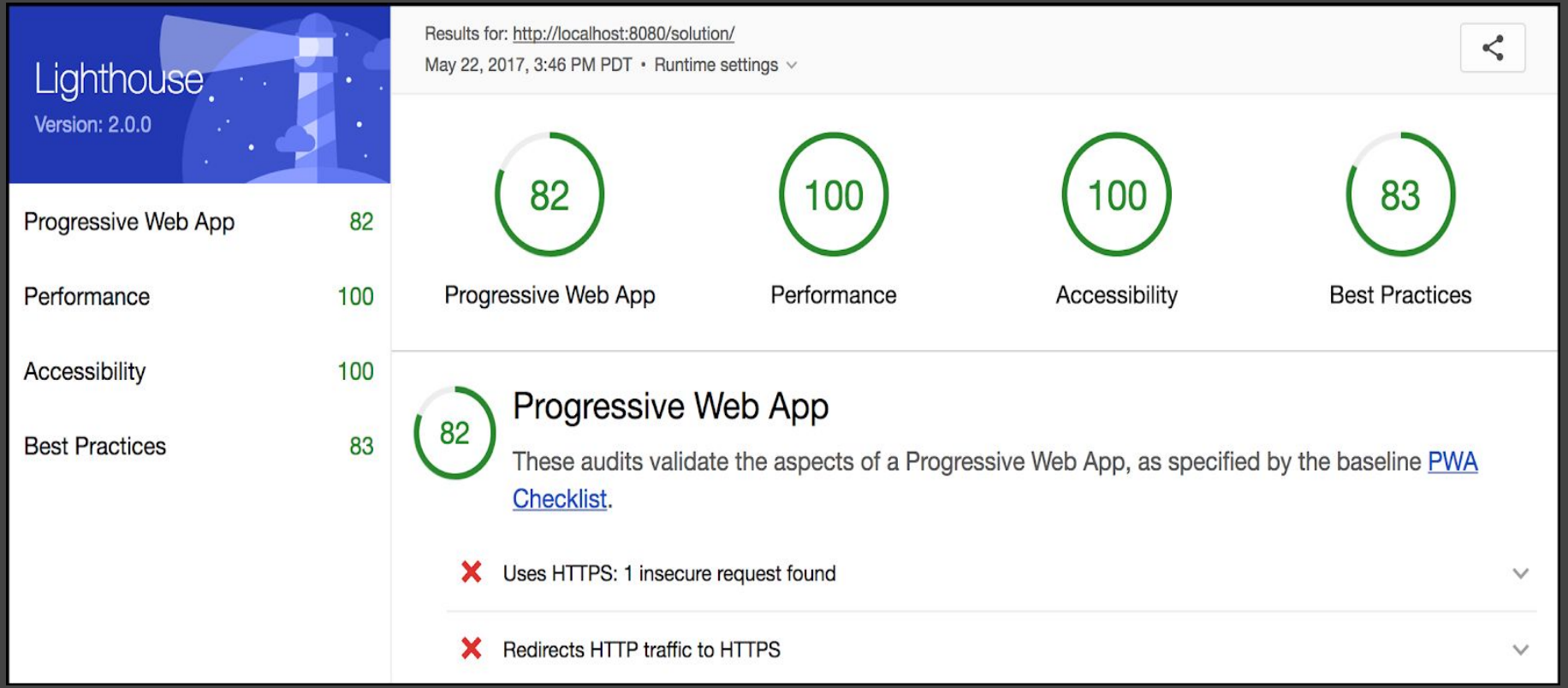
1. Pedir permiso al usuario.
2. Registro en servidor Notificaciones (VAPID)
3. Habilitar a nuestro servidor el registro de Push
4. Nuestro servidor envía al servidor push la notificación.
5. El servidor de Notificaciones envía a la web la notificación.

HERRAMIENTAS

CONSOLA DESARROLLADOR



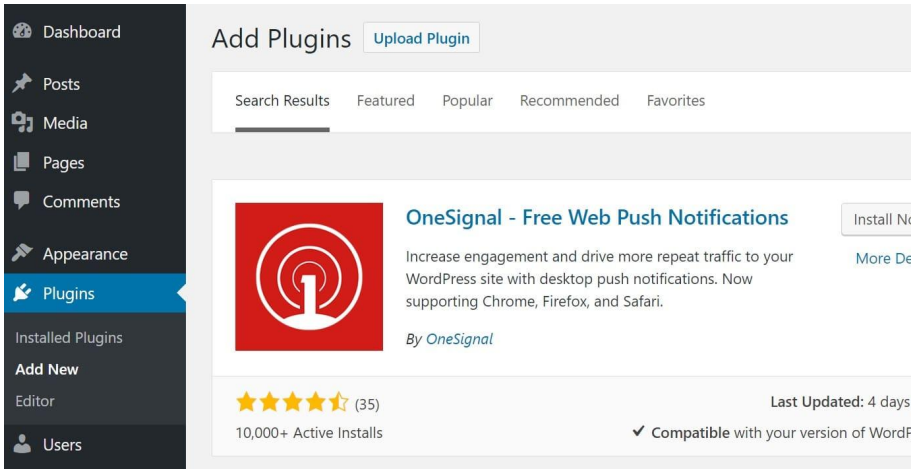
LIGHTHOUSE


Lighthouse
Version: 2.0.0
Progressive Web App 82
Performance 100
Accessibility 100
Best Practices 83
Results for: http://localhost:8080/solution/
May 22, 2017, 3:46 PM PDT • Runtime settings
82
Progressive Web App
100
Performance
100
Accessibility
83
Best Practices
82
Progressive Web App
These audits validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.
Uses HTTPS: 1 insecure request found
Redirects HTTP traffic to HTTPS

¿Y QUÉ PASA CON LOS CMS?

WORDPRESS

Existen multitud de plugins.



También hay soporte para otros CMS.

¿Y QUÉ PASA CON LAS STORES?

STORE Y DESPLIEGUE

Actualmente todas las stores, Google Play, App Store, y Microsoft Store permiten subir PWAs usando los mismos mecanismos que para las aplicaciones híbridas → **WRAPPERS y TRUSTED WEB ACTIVITIES**

DEMO TIME

¿Estás ahí Murphy?

Enlaces: Generadores de PWA

- Web App Manifest Generator
- PWA Builder (asistente)

CONCLUSIONES

CONCLUSIONES

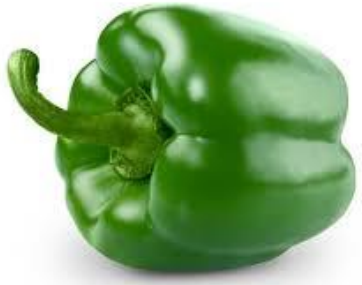
¡PWAs son la leche!

¡Nunca tan poco código ha provocado tanto impacto!

Tenemos webs instalables, se ejecutan como apps nativas, funcionando sin conexión, recibiendo push notifications, sin pedir actualizaciones, distribuibles por las “stores”. ¡Todo con 1 solo código!

CONCLUSIONES

¡PWAs están un poco verdes!



- Homogeneidad por parte de los navegadores.
- Aumentar capacidades (cámara, gps, etc.).
- Mejorar relación usuario-PWA.

- ¿Qué va a pasar con las stores de aplicaciones nativas?

ENLACES

Enlaces: Material didáctico online

- Taller de construcción PWA
- The Offline Cookbook: PWAs
- The Service Worker Cookbook
- The PWA Workshop

Enlaces: Material didáctico online

- Tutorial paso a paso sobre PWAs
- Push Notifications usando VAPID
- Web Push Notifications (Paso a paso)
- Video Tutorial Push Notifications
- Angular PWA to Google Play store, using Trusted Web Activity

Enlaces: Preparar PWA para Stores

- Aplicación para convertir PWA en APK
- Video tutorial convertir PWA en APK
- Wrapper conversión PWA en Android, iOS o Electron

GRACIAS ;)