

Progressive Web Applications (PWA)

The logo consists of the letters 'PWA' in a bold, sans-serif font. The 'P' and 'A' are dark grey, while the 'W' is a teal color. The letters are closely spaced and have a slight shadow effect.

Temario

Temario

- Concepto y estructura básica
- App manifest.
- Service Workers.
- Estrategias de caching con Service Workers.
- Web Push Notifications.

Introducción a las PWA

PWA

PWA

- Diferencias entre aplicaciones nativas y webs

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas
- Trabajan offline

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas
- Trabajan offline
- Un solo lenguaje

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas
- Trabajan offline
- Un solo lenguaje
- Fácil distribución

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas
- Trabajan offline
- Un solo lenguaje
- Fácil distribución
- Enganchan al usuario

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas
- Trabajan offline
- Un solo lenguaje
- Fácil distribución
- Enganchan al usuario
- Aparecen en Google

PWA

- Diferencias entre aplicaciones nativas y webs
- Ventajas de web y de apps nativas
- Aplicaciones progresivas y multiplataforma
- Aplicaciones web (HTML, CSS, JS)
- Responsive
- Rápidas
- Trabajan offline
- Un solo lenguaje
- Fácil distribución
- Enganchan al usuario
- Aparecen en Google
- Seguras

Entorno de desarrollo



Entorno de desarrollo

- Editor: [Visual Studio Code](#)
 - [Spanish Language Pack for Visual Studio Code](#)
 - [EditorConfig for VSCode](#)
 - [PWA Tools](#)
 - [Live Share](#)
 - Configurar:
 - Format on Paste y Format on Save
- [Git](#)
- [Node.js](#) y npm
- Extensión [ColorZilla](#) para el navegador
- [Android Studio](#) para el emulador de Android
- [Xcode](#) para el emulador de iOS

Git



Comandos básicos

Comandos básicos

- Clonar un repositorio:

```
git clone URL
```

Comandos básicos

- Clonar un repositorio:

```
git clone URL
```

- Descargar última versión del repositorio:

```
git pull origin master
```

Configuración proxy

```
git config --global http.proxy http://username:password@host:port
```

```
git config --global https.proxy http://username:password@host:port
```


Node.js y npm



npm

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles
- Módulos globales y módulos locales

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles
- Módulos globales y módulos locales
- La carpeta `node_modules`

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles
- Módulos globales y módulos locales
- La carpeta `node_modules`
- El archivo `package.json`:

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles
- Módulos globales y módulos locales
- La carpeta `node_modules`
- El archivo `package.json`:
 - Registro de dependencias

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles
- Módulos globales y módulos locales
- La carpeta `node_modules`
- El archivo `package.json`:
 - Registro de dependencias
 - Dependencias de desarrollo y de producción

npm

- Instalar última versión después de instalar Node.js (configurar proxy si es necesario): `npm install -g npm`
- Repositorio de módulos distribuibles
- Módulos globales y módulos locales
- La carpeta `node_modules`
- El archivo `package.json`:
 - Registro de dependencias
 - Dependencias de desarrollo y de producción
 - Versiones (SEMVER)

Comandos npm

Comandos npm

- Instalar un paquete globalmente:
npm install -g paquete

Comandos npm

- Instalar un paquete globalmente:
npm install -g paquete
- Instalar un paquete de producción:
npm install paquete

Comandos npm

- Instalar un paquete globalmente:
npm install -g paquete
- Instalar un paquete de producción:
npm install paquete
- Instalar un paquete de desarrollo:
npm install paquete --save-dev

Comandos npm

- Instalar un paquete globalmente:
npm install -g paquete
- Instalar un paquete de producción:
npm install paquete
- Instalar un paquete de desarrollo:
npm install paquete --save-dev
- Instalar todas las dependencias:
npm install

Comandos npm

- Instalar un paquete globalmente:
`npm install -g paquete`
- Instalar un paquete de producción:
`npm install paquete`
- Instalar un paquete de desarrollo:
`npm install paquete --save-dev`
- Instalar todas las dependencias:
`npm install`
- Instalar las dependencias de producción:
`npm install --production`

Comandos npm

- Instalar un paquete globalmente:
npm install -g paquete
- Instalar un paquete de producción:
npm install paquete
- Instalar un paquete de desarrollo:
npm install paquete --save-dev
- Instalar todas las dependencias:
npm install
- Instalar las dependencias de producción:
npm install --production
- Listar paquetes instalados:
npm list --depth=0 (locales)
npm list -g --depth=0 (globales)

Comandos npm

Comandos npm

- Lanzar el ejecutable de un paquete:
npx ejecutable

(aunque no esté instalado)

Configuración proxy

```
npm config set proxy http://username:password@host:port
```

```
npm config set https-proxy http://username:password@host:port
```

JavaScript



Funciones

Funciones

- Pasar funciones anónimas como parámetros

Funciones

- Pasar funciones anónimas como parámetros
- Funciones callback

Funciones

- Pasar funciones anónimas como parámetros
- Funciones callback
- Funciones arrow

Funciones

- Pasar funciones anónimas como parámetros
- Funciones callback
- Funciones arrow
- `map()` y `filter()`

Promesas

Promesas

- Procesos asíncronos

Promesas

- Procesos asíncronos
- Dos métodos principales: `then()` y `catch()`

Promesas

- Procesos asíncronos
- Dos métodos principales: `then()` y `catch()`
- Encadenado de promesas

Promesas

- Procesos asíncronos
- Dos métodos principales: `then()` y `catch()`
- Encadenado de promesas
- El método `finally()`

Promesas

- Procesos asíncronos
- Dos métodos principales: `then()` y `catch()`
- Encadenado de promesas
- El método `finally()`
- `Promise.all()`

Promesas

- Procesos asíncronos
- Dos métodos principales: `then()` y `catch()`
- Encadenado de promesas
- El método `finally()`
- `Promise.all()`
- `async / await`

Fetch API

Fetch API

- Request y Response

Fetch API

- Request y Response
- La función `fetch()`

Fetch API

- Request y Response
- La función `fetch()`
- Nos interesa del Request:

Fetch API

- Request y Response
- La función `fetch()`
- Nos interesa del Request:
 - `method`

Fetch API

- Request y Response
- La función fetch()
- Nos interesa del Request:
 - method
 - mode

Fetch API

- Request y Response
- La función fetch()
- Nos interesa del Request:
 - method
 - mode
 - destination

Fetch API

- Request y Response
- La función fetch()
- Nos interesa del Request:
 - method
 - mode
 - destination
 - headers

Fetch API

- Request y Response
- La función fetch()
- Nos interesa del Request:
 - method
 - mode
 - destination
 - headers
 - url

Fetch API

- Request y Response
- La función fetch()
- Nos interesa del Request:
 - method
 - mode
 - destination
 - headers
 - url
 - body

Fetch API

Fetch API

- Nos interesa del Response:

Fetch API

- Nos interesa del Response:
 - ok

Fetch API

- Nos interesa del Response:
 - ok
 - status

Fetch API

- Nos interesa del Response:
 - ok
 - status
 - headers

Fetch API

- Nos interesa del Response:
 - ok
 - status
 - headers
 - body

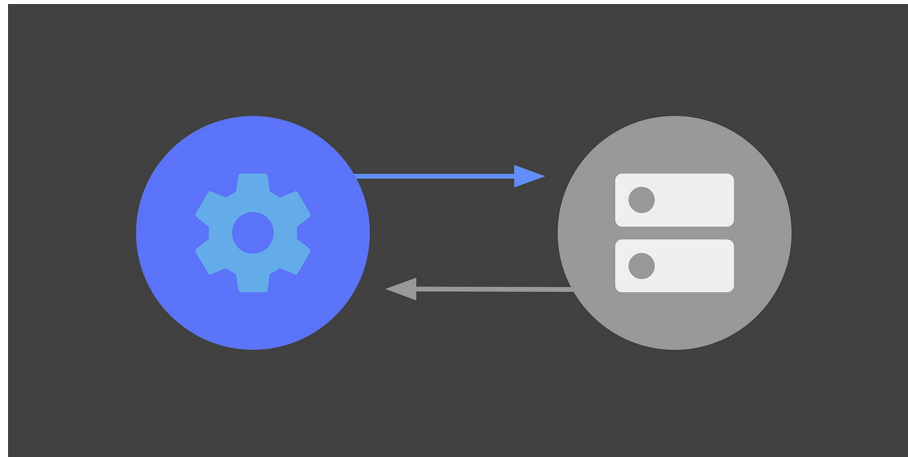
Fetch API

- Nos interesa del Response:
 - ok
 - status
 - headers
 - body
 - `body.text()`

Fetch API

- Nos interesa del Response:
 - ok
 - status
 - headers
 - body
 - `body.text()`
 - `body.json()`

PWA



Lighthouse

Lighthouse

- Integrada en Chrome

Lighthouse

- Integrada en Chrome
- También disponible como [paquete npm](#)

Lighthouse

- Integrada en Chrome
- También disponible como [paquete npm](#)
- Nos guía para saber qué le falta a nuestra PWA

Web manifest

Web manifest

- Información sobre la apariencia de la app instalada

Web manifest

- Información sobre la apariencia de la app instalada
- Archivo JSON

Web manifest

- Información sobre la apariencia de la app instalada
- Archivo JSON
- `<link rel="manifest">`

Web manifest

Web manifest

- Estructura:

Web manifest

- Estructura:

```
1 {
2   "name": "Nombre de la app",
3   "short_name": "Nombre corto",
4   "theme_color": "#2196f3",
5   "background_color": "#2196f3",
6   "display": "standalone",
7   "start_url": "/",
8   "icons": [
9     {
10      "src": "images/icons/icon-192x192.png",
11      "sizes": "192x192",
12      "type": "image/png"
13    },
14    {
15      "src": "images/icons/icon-512x512.png",
16      "sizes": "512x512",
17      "type": "image/png"
18    }
19  ]
20 }
```

Web manifest

Web manifest

- Iconos para web

Web manifest

- Iconos para web

```
<link rel="icon" type="image/png" sizes="32x32" href="<ruta>/favicon-32x32.png">  
<link rel="icon" type="image/png" sizes="16x16" href="<ruta>/favicon-16x16.png">  
<link rel="shortcut icon" href="<ruta>/favicon.ico">
```

Web manifest

Web manifest

- Iconos para Android (mínimo 192x192 y 512x512)

Web manifest

- Iconos para Android (mínimo 192x192 y 512x512)

```
"icons": [  
  {  
    "src": "images/icons/icon-72x72.png",  
    "sizes": "72x72",  
    "type": "image/png"  
  },  
  {  
    "src": "images/icons/icon-96x96.png",  
    "sizes": "96x96",  
    "type": "image/png"  
  },  
  {  
    "src": "images/icons/icon-128x128.png",  
    "sizes": "128x128",  
    "type": "image/png"  
  },  
  {  
    "src": "images/icons/icon-144x144.png",  
    "sizes": "144x144",  
    "type": "image/png"  
  },  
  {  
    "src": "images/icons/icon-192x192.png",  
    "sizes": "192x192",  
    "type": "image/png"  
  },  
  {  
    "src": "images/icons/icon-512x512.png",  
    "sizes": "512x512",  
    "type": "image/png"  
  }  
]
```

Web manifest

Web manifest

- iOS 🍷 (documentación)

Web manifest

- iOS 🍏 (documentación)

```
<meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0,
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-touch-fullscreen" content="yes">
<meta name="apple-mobile-web-app-title" content="appTitle">
<meta name="apple-mobile-web-app-status-bar-style" content="default">
<link rel="apple-touch-icon" sizes="180x180" href="<ruta>/apple-touch-icon.png">
<link rel="apple-touch-startup-image" media="(device-width: 375px) and (device-height: 812px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-375x812.png">
<link rel="apple-touch-startup-image" media="(device-width: 375px) and (device-height: 667px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-375x667.png">
<link rel="apple-touch-startup-image" media="(device-width: 414px) and (device-height: 736px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-414x736.png">
<link rel="apple-touch-startup-image" media="(device-width: 320px) and (device-height: 568px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-320x568.png">
<link rel="apple-touch-startup-image" media="(device-width: 768px) and (device-height: 1024px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-768x1024.png">
<link rel="apple-touch-startup-image" media="(device-width: 834px) and (device-height: 1194px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-834x1194.png">
<link rel="apple-touch-startup-image" media="(device-width: 1024px) and (device-height: 1366px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-1024x1366.png">
```


Service workers

Service workers

- JavaScript que corre en un hilo paralelo

Service workers

- JavaScript que corre en un hilo paralelo
- Proxy entre la app y la red

Service workers

- JavaScript que corre en un hilo paralelo
- Proxy entre la app y la red
- Registro de un SW

Service workers

- JavaScript que corre en un hilo paralelo
- Proxy entre la app y la red
- Registro de un SW
- Ciclo de vida de un SW

Service workers

Service workers

- Condiciones para que la PWA sea instalable:

Service workers

- Condiciones para que la PWA sea instalable:
 - Manifest con:
 - name o short_name
 - start_url
 - display: "standalone" | "fullscreen" | "minimal-ui"
 - icons: 512 y 192

Service workers

- Condiciones para que la PWA sea instalable:
 - Manifest con:
 - name o short_name
 - start_url
 - display: "standalone" | "fullscreen" | "minimal-ui"
 - icons: 512 y 192
 - Service Worker con evento fetch

Service workers

- Condiciones para que la PWA sea instalable:
 - Manifest con:
 - name o short_name
 - start_url
 - display: "standalone" | "fullscreen" | "minimal-ui"
 - icons: 512 y 192
 - Service Worker con evento fetch
 - HTTPS

Service workers

Service workers

- El prompt de instalación

Service workers

- El prompt de instalación
- El evento `beforeinstallprompt`:

Service workers

- El prompt de instalación
- El evento `beforeinstallprompt`:
 - `preventDefault()` para que no aparezca el de Android

Service workers

- El prompt de instalación
- El evento `beforeinstallprompt`:
 - `preventDefault()` para que no aparezca el de Android
 - Lanzar el prompt con un gesto del usuario

Service workers

- El prompt de instalación
- El evento `beforeinstallprompt`:
 - `preventDefault()` para que no aparezca el de Android
 - Lanzar el prompt con un gesto del usuario
 - Leer `userChoice`

Service workers

- El prompt de instalación
- El evento `beforeinstallprompt`:
 - `preventDefault()` para que no aparezca el de Android
 - Lanzar el prompt con un gesto del usuario
 - Leer `userChoice`
 - Comprobar si se está abriendo la app instalada

Service workers

```
1 if (window.matchMedia('(display-mode: standalone)').matches
2   console.log('Desde la app instalada');
3 }
```

Service workers

```
1 let eventoInstall;
2
3 window.addEventListener('beforeinstallprompt', e => {
4     eventoInstall = e;
5     e.preventDefault();
6     document.querySelector('.install').removeAttribute('hidden');
7 })
8
9 document.querySelector('.install').addEventListener('click', () => {
10     eventoInstall.prompt();
11     eventoInstall.userChoice.then(ch => {
12         if (ch.outcome === 'accepted') {
13             console.log('El usuario ha aceptado instalar');
14         } else {
15             console.log('El usuario no ha aceptado instalar');
16         }
17         document.querySelector('.install').setAttribute('hidden', true);
18     });
19 });
20
21 window.addEventListener('appinstalled', () => console.log("El usuario ha instalado la aplicac
```

Caché

Caché

- CacheStorage API:

Caché

- CacheStorage API:
 - open()

Caché

- CacheStorage API:
 - open()
 - addAll()

Caché

- CacheStorage API:
 - open()
 - addAll()
 - put()

Caché

- CacheStorage API:
 - open()
 - addAll()
 - put()
 - match()

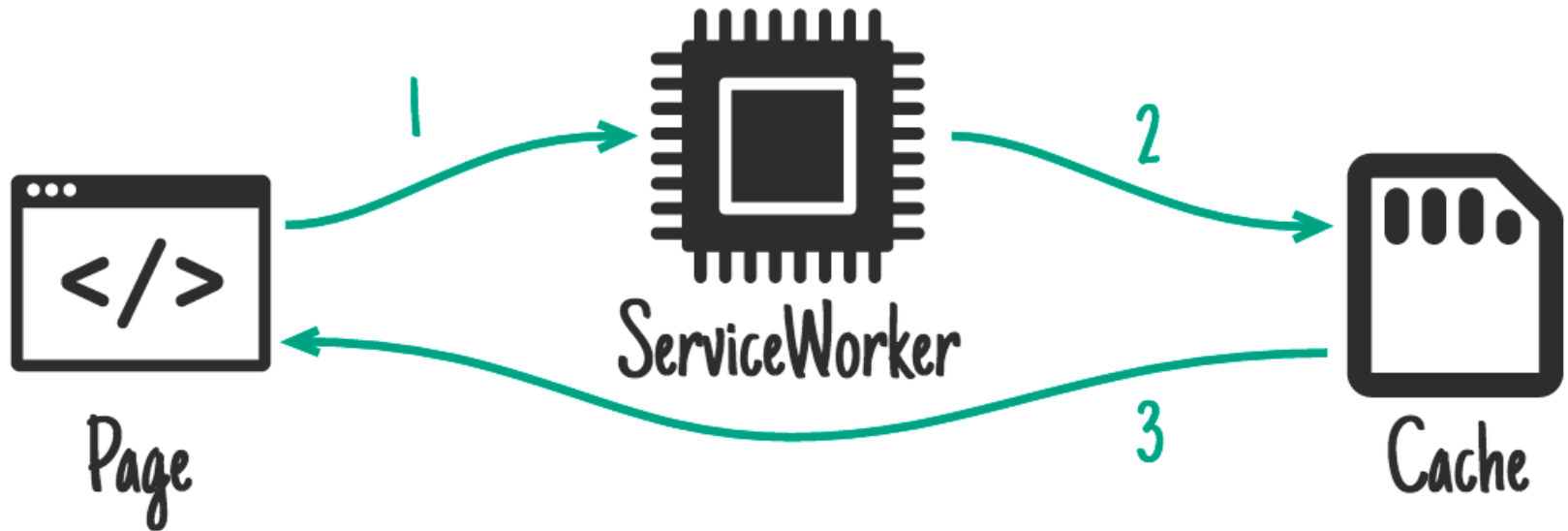
Caché

- CacheStorage API:
 - open()
 - addAll()
 - put()
 - match()
 - keys()

Estrategias de caching

- Cache only

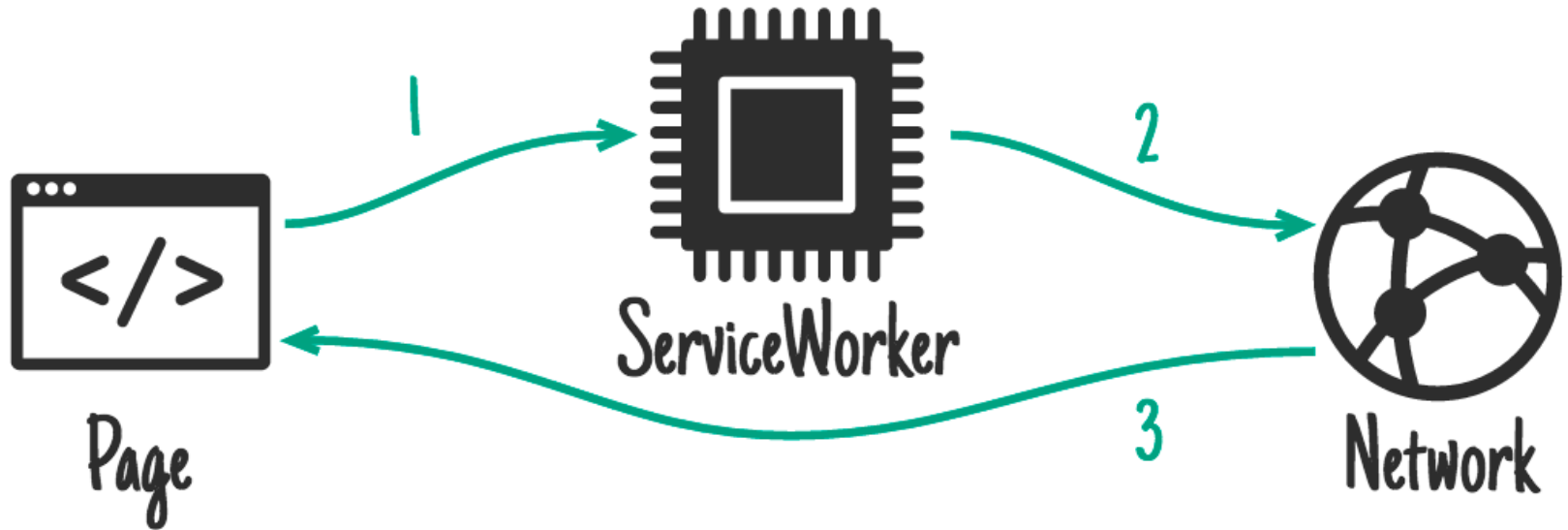
Estrategias de caching



Estrategias de caching

- Cache only
- Network only

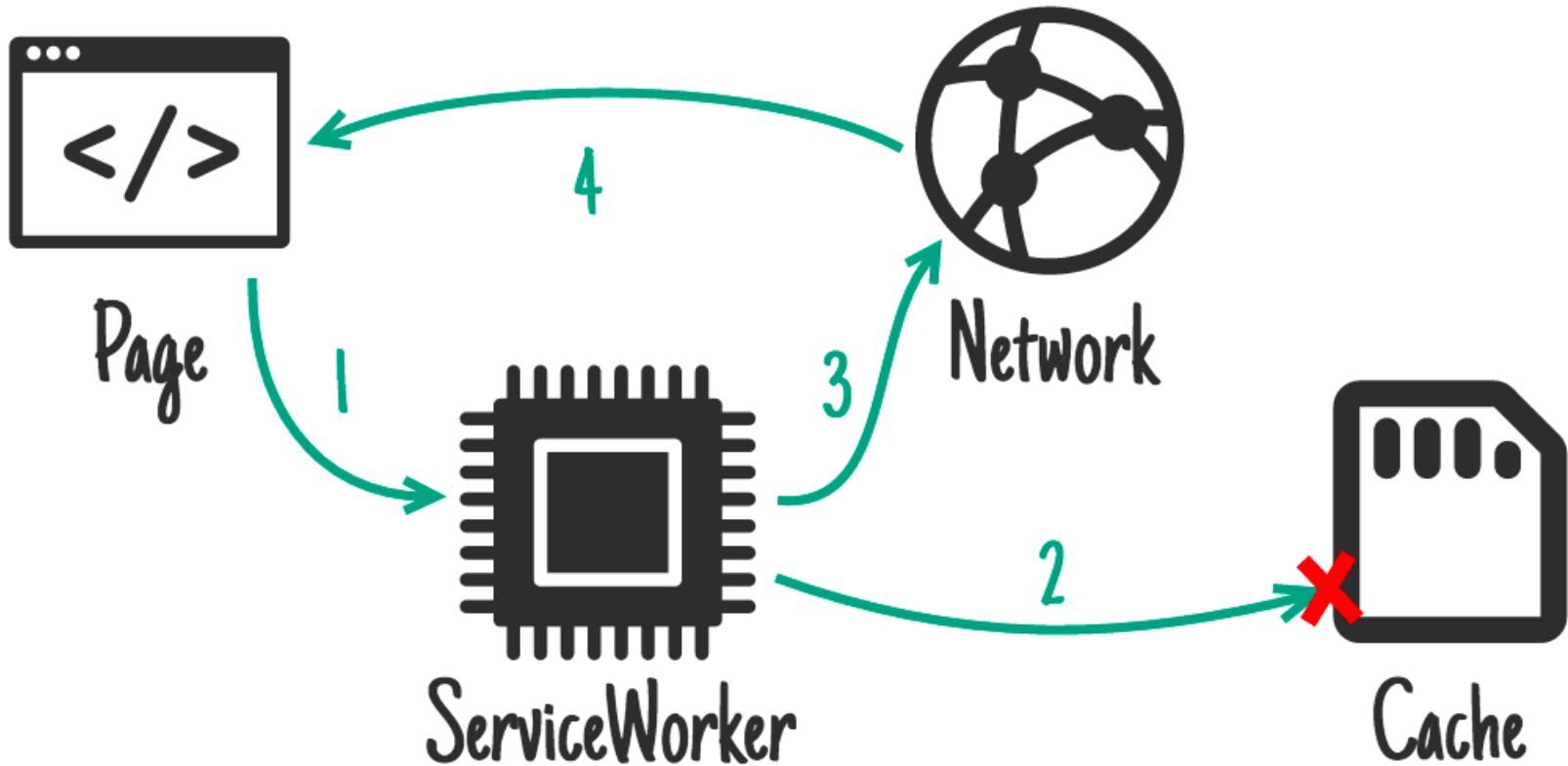
Estrategias de caching



Estrategias de caching

- Cache only
- Network only
- Cache with fallback to network

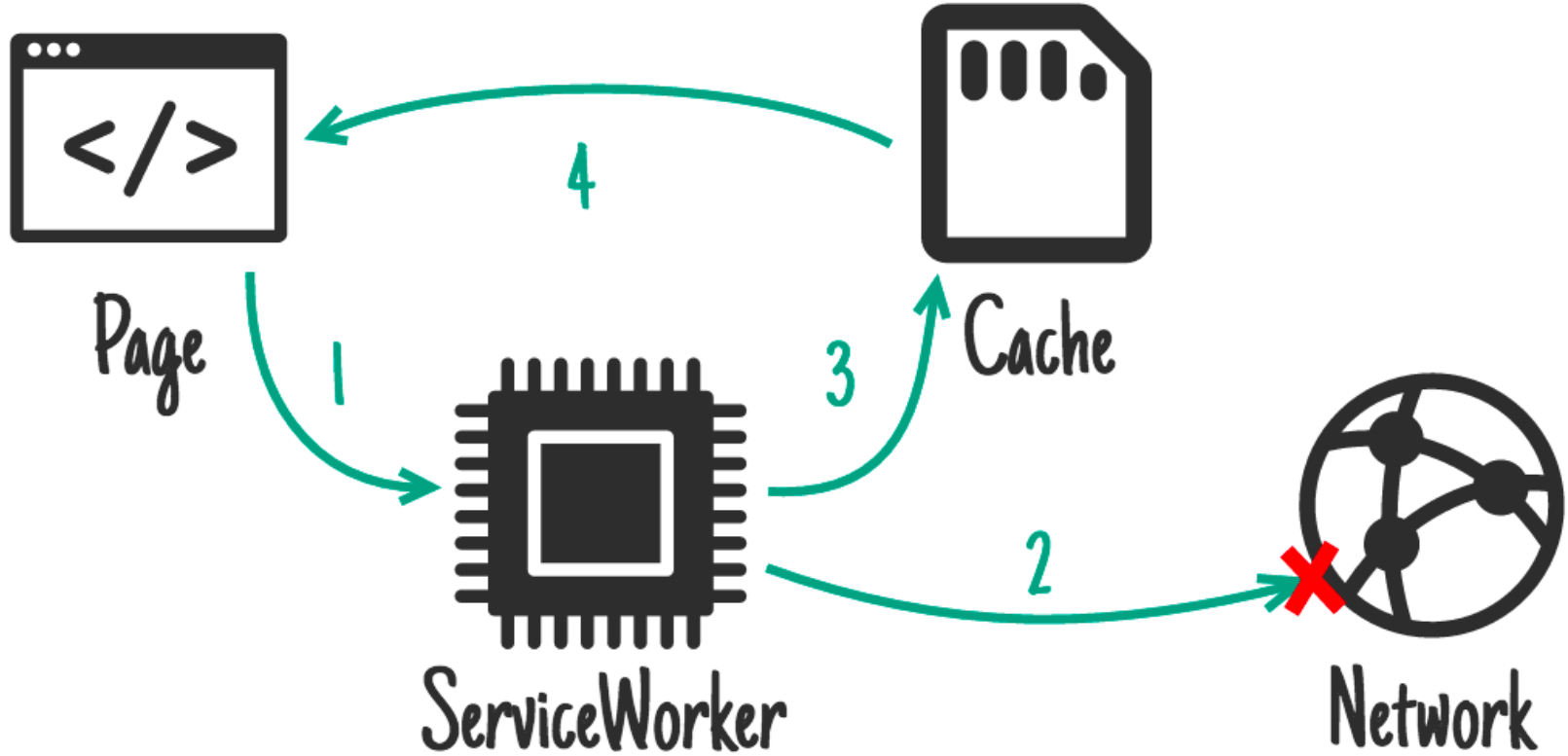
Estrategias de caching



Estrategias de caching

- Cache only
- Network only
- Cache with fallback to network
- Network with fallback to cache

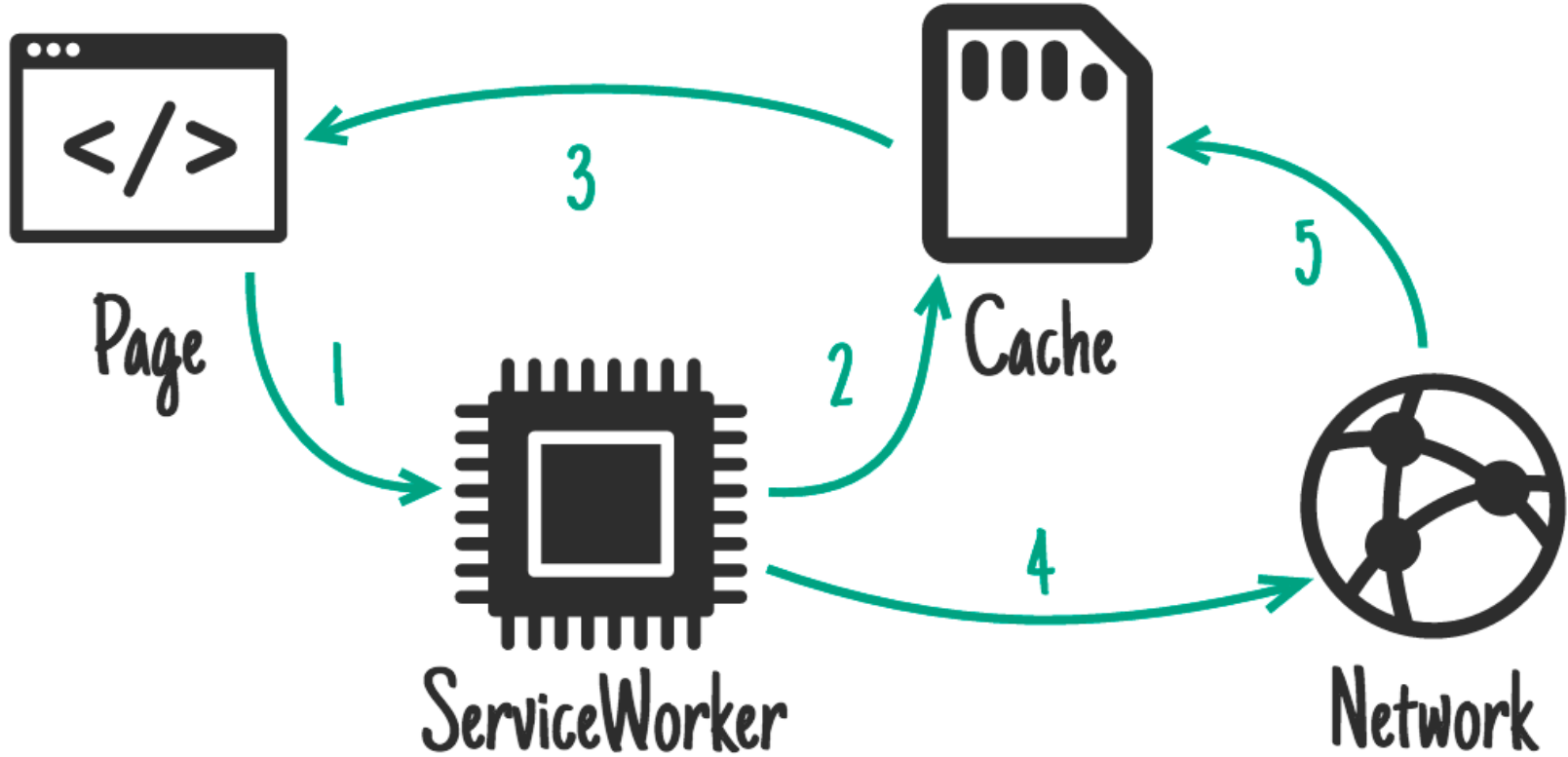
Estrategias de caching



Estrategias de caching

- Cache only
- Network only
- Cache with fallback to network
- Network with fallback to cache
- Stale while revalidate

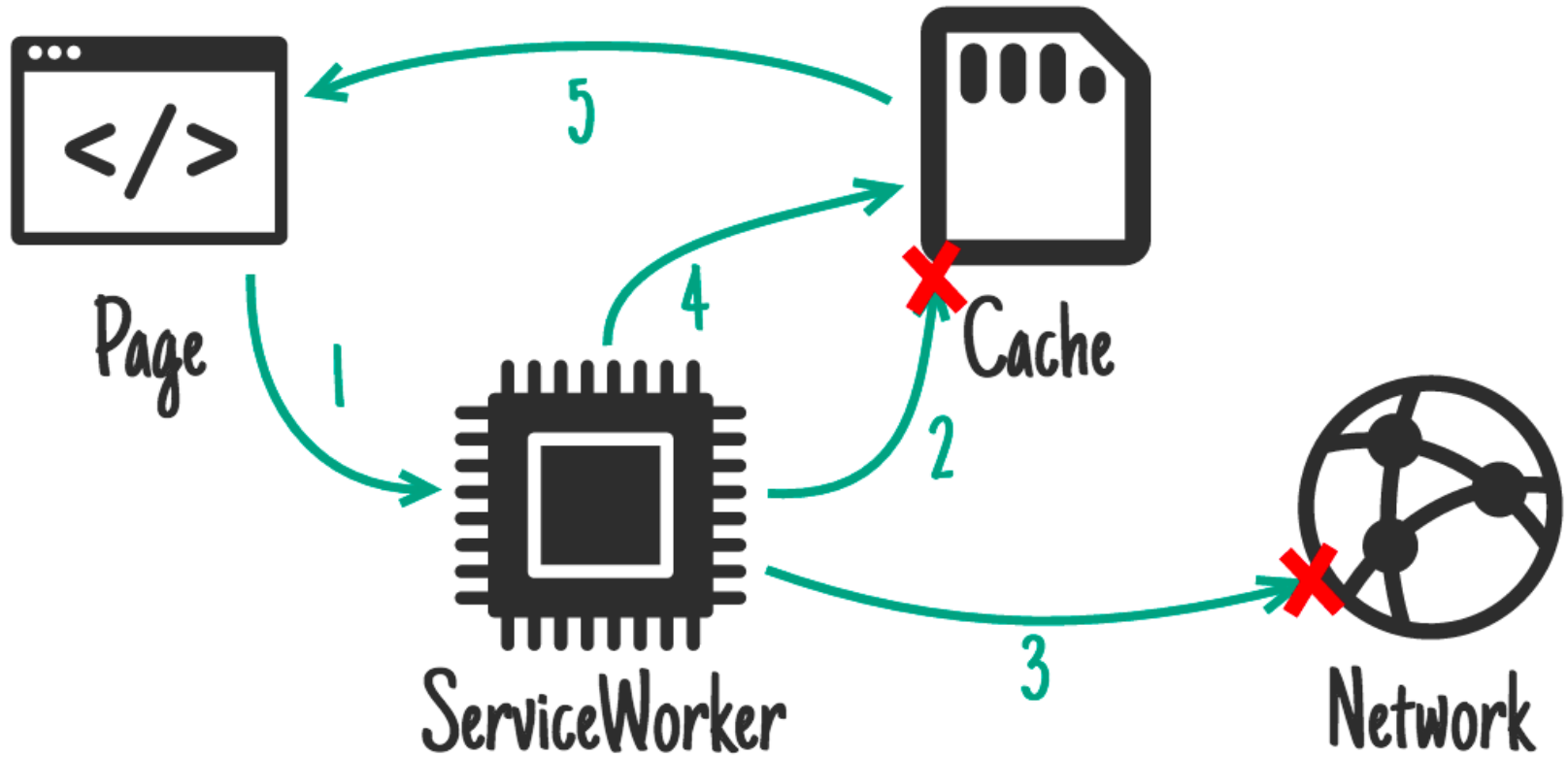
Estrategias de caching



Estrategias de caching

- Cache only
- Network only
- Cache with fallback to network
- Network with fallback to cache
- Stale while revalidate
- Generic fallback

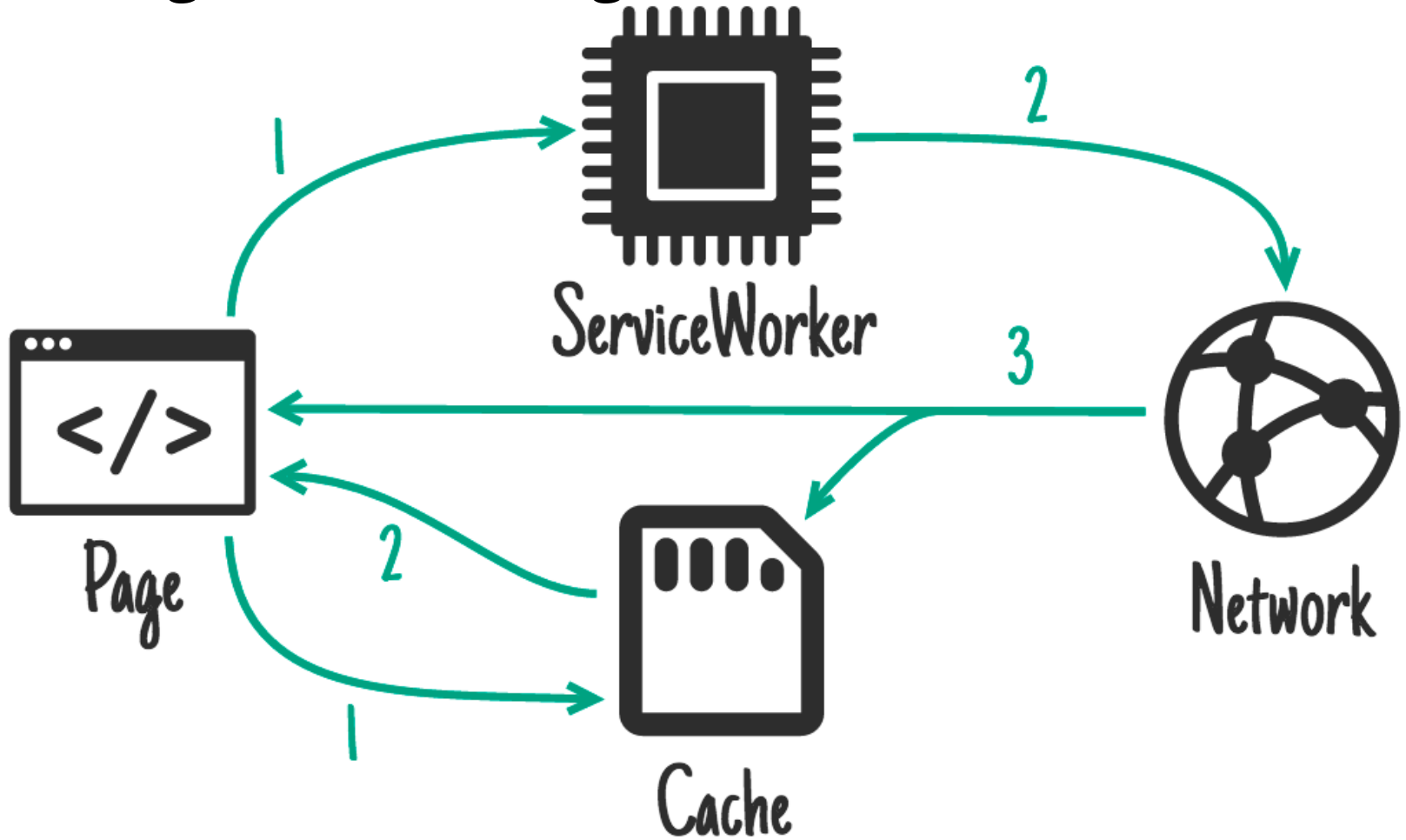
Estrategias de caching



Estrategias de caching

- Cache only
- Network only
- Cache with fallback to network
- Network with fallback to cache
- Stale while revalidate
- Generic fallback
- Cache then network

Estrategias de caching



Workbox

Workbox

- Instalar:
`npm install [-g] [--save-dev] workbox-cli`

Workbox

- Instalar:
`npm install [-g] [--save-dev] workbox-cli`
- Generar configuración:
`workbox wizard --injectManifest`

Workbox

- Instalar:
`npm install [-g] [--save-dev] workbox-cli`
- Generar configuración:
`workbox wizard --injectManifest`
- El SW fuente tiene que tener esta línea:
`workbox.precaching.precacheAndRoute([]);`

Workbox

- Instalar:
`npm install [-g] [--save-dev] workbox-cli`
- Generar configuración:
`workbox wizard --injectManifest`
- El SW fuente tiene que tener esta línea:
`workbox.precaching.precacheAndRoute([]);`
- `workbox injectManifest`

Workbox

```
1 importScripts('https://storage.googleapis.com/workbox-cdn/releases/4.3.1/workbox-sw.js');
2
3 workbox.routing.registerRoute(
4   /\.css$/,
5   new workbox.strategies.CacheFirst();
6 );
7
8 workbox.routing.registerRoute(
9   /\.(?:png|gif|jpg|jpeg|svg)$/,
10  new workbox.strategies.StaleWhileRevalidate({
11    cacheName: '',
12    plugins: [
13      new workbox.expiration.Plugin({
14        maxEntries: 100,
15        maxAgeSeconds: 7 * 24 * 60 * 60,
16      }),
17    ],
18  });
19 );
```

Notificaciones PUSH

Notificaciones PUSH

- Entran en juego:

Notificaciones PUSH

- Entran en juego:
 - La app

Notificaciones PUSH

- Entran en juego:
 - La app
 - El Service Worker

Notificaciones PUSH

- Entran en juego:
 - La app
 - El Service Worker
 - El servicio PUSH del navegador

Notificaciones PUSH

- Entran en juego:
 - La app
 - El Service Worker
 - El servicio PUSH del navegador
 - Un servicio de Google, Firefox...

Notificaciones PUSH

- Entran en juego:
 - La app
 - El Service Worker
 - El servicio PUSH del navegador
 - Un servicio de Google, Firefox...
 - Un servidor PUSH

Notificaciones PUSH

Notificaciones PUSH

- La app:

Notificaciones PUSH

- La app:
 - Puede lanzar el prompt para preguntar al usuario si da permisos para notificaciones:
`Notification.requestPermission()`

Notificaciones PUSH

- La app:
 - Puede lanzar el prompt para preguntar al usuario si da permisos para notificaciones:
`Notification.requestPermission()`
 - Puede preguntar si el usuario ya ha dado permisos o no:
`Notification.permission`
(`'default'`, `'granted'` o `'denied'`)

Notificaciones PUSH

- La app:
 - Puede lanzar el prompt para preguntar al usuario si da permisos para notificaciones:
`Notification.requestPermission()`
 - Puede preguntar si el usuario ya ha dado permisos o no:
`Notification.permission`
(`'default'`, `'granted'` o `'denied'`)
 - Puede pedir una suscripción al servicio PUSH:
`reg.pushManager.subscribe({`
 `userVisibleOnly: true,`
 `applicationServerKey: <llave pública VAPID>`
`});`

Notificaciones PUSH

Notificaciones PUSH

- La app:
 - Cuando obtiene una suscripción del servicio PUSH, la envía a nuestro servidor PUSH, que la almacenará.

Notificaciones PUSH

Notificaciones PUSH

- El servidor PUSH:

Notificaciones PUSH

- El servidor PUSH:
 - Genera las claves VAPID

Notificaciones PUSH

- El servidor PUSH:
 - Genera las claves VAPID
 - Utiliza la librería web-push para enviar notificaciones PUSH.

Notificaciones PUSH

- El servidor PUSH:
 - Genera las claves VAPID
 - Utiliza la librería web-push para enviar notificaciones PUSH.
 - [web-push para varios lenguajes de back](#)

Notificaciones PUSH

Notificaciones PUSH

- El Service Worker:

Notificaciones PUSH

- El Service Worker:
 - Escucha el evento 'push'

Notificaciones PUSH

- El Service Worker:
 - Escucha el evento 'push'
 - Emite una notificación al SO:
`self.registration.showNotification(título, opciones)`

Notificaciones PUSH

- El Service Worker:
 - Escucha el evento 'push'
 - Emite una notificación al SO:
`self.registration.showNotification(título, opciones)`
 - Puede capturar el click en la notificación escuchando al evento 'notificationclick'

Links

- [Tutorial Tu primera PWA](#)
- [Progressive Web App Training \(vídeos\)](#)
- [Estrategias de caché](#)
- [Tablas de compatibilidad en navegadores](#)
- [Biblia de JavaScript](#)
- [Ejemplo de iconos para iOS](#)
- [Generador de iconos](#)
- [Generador de web manifest](#)
- [Generador de splash screen para iOS](#)
- [Librerías para VAPID](#)
- [Función urlBase64ToUint8Array\(\)](#)

mario@mariogl.com
@marioglweb