

# JAVASCRIPT

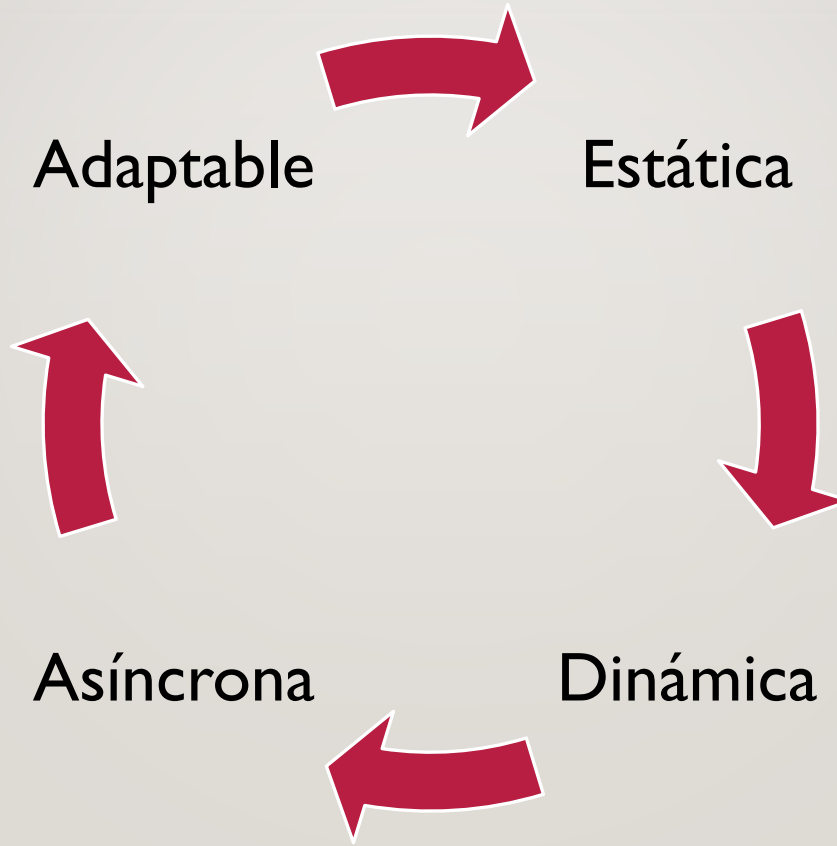
---

PROGRESSIVE WEB APPS



# HISTORIA DE LA WEB

---



# PROBLEMAS SIN RESOLVER

---

- Funcionamiento *on line* exclusivo.
- Menor rendimiento que las apps nativas.
  - Una web necesita la capa extra del navegador.
- Arranque.
  - Una web se arranca en el navegador.



# APLICACIONES PROGRESIVAS

---

- Mitiga estos fallos.
- Pretenden que progresivamente se alcancen los mismos objetivos de las aplicaciones nativas.
- Wikipedia: *“Aplicaciones web que, de manera progresiva, incorporan funcionalidades y tienen un rendimiento similar al de las aplicaciones nativas”*



# ¿POR QUÉ SE LES LLAMA PROGRESIVAS?

---

- El adjetivo *progresiva* tiene que ver con la capacidad de la aplicación de aumentar progresivamente sus capacidades según el dispositivo en el que se ejecute.
  - Si nuestra aplicación se ejecuta en un dispositivo que permita ejecutarla hasta un cierto nivel, esta funcione hasta donde pueda llegar.
  - Si ejecuta en otro que permita expresar todas sus capacidades al máximo la aplicación alcance ahí su máximo rendimiento.



# CARACTERÍSTICAS

---

- Orientadas a dispositivos móviles
- Aspectos innovadores:
  - Funcionamiento off line
  - Notificaciones *push*
  - Son instalables
  - Permiten acceder al hardware del dispositivo gracias a las APIs HTML5
  - Tiempos de carga más rápidos





# PWAPPS VS APPS HÍBRIDAS

---

- No son muchas.
- Ambas se desarrollan con tecnologías web.
- Las apps híbridas se desarrollan en plataformas orientadas a dispositivos móviles.
- Las PWApps forman parte del desarrollo web normal.
- Estas últimas están pensadas para quienes ya conocen una web y la quieren en su versión móvil.



# QUÉ DEBEMOS IMPLEMENTAR

---

- Archivo de configuración ***Manifest.json***.
  - Nombre de la app
  - Características de la *splash screen*
  - Iconos...
- **Service worker**: Fundamental. Se ocupa de hacer todo lo que la PWApp necesita para ser una PWApp.
- Archivo de configuración **ngsw-config.json** para configurar el *Service worker*.





# EL *SERVICE WORKER*

---

- Es un proceso en segundo plano.
- Se ejecuta paralelamente al JavaScript de la aplicación.
- Se “entromete” entre la app y la caché del navegador para actualizar contenidos si hace falta.
- Es como un vigilante que pasa las notificaciones, auto-actualiza la aplicación...
- Cuando la app está en desuso se queda en reposo consumiendo recursos mínimos.
- Todavía tiene que estandarizarse más.



# EL FUTURO

---

- Facilitar que cualquier app pueda ser una PWApp
- Que se acerque a ser un estándar en el desarrollo (muchos frameworks ya lo incorporan)
- Que IOS se abra más al uso de las PWApps
- Se puede testear si nuestro navegador / dispositivo puede albergar PWApps.
  - <https://whatwebcando.today/>

