



Enfoques de Desarrollo de Aplicaciones Móviles Multiplataforma



Aplicaciones Web Progresivas - PWA

Esp. Juan Fernández Sosa



Aplicaciones Web Progresivas

- Una PWA utiliza las **últimas tecnologías** disponibles en los **navegadores** para ofrecer una **experiencia** lo más **parecida a la de una aplicación nativa**.
- Combina lo mejor de una app web con lo mejor de las aplicaciones móviles.
- El empleo de éstas tecnologías permite **superar** algunas de las **limitaciones** que subyacen al enfoque que **desarrollo web móvil**.
- Entre dichas tecnologías se encuentran los **Service Workers** que se encargan de realizar **tareas en segundo plano** y el **Manifiesto de la Aplicación Web** que permite **configurar** cómo mostrar la aplicación en **dispositivos móviles**.



Características

- **Desarrollo simple y rápido**, donde un **mismo código fuente** puede ser **ejecutado en dispositivos de diferentes características** (smartphones, tablets y laptops) mientras éstos cuenten con un **navegador web compatible**. (**Portabilidad**)
- Buen **rendimiento** en móviles y **carga instantánea**
- Interfaz lo más parecida posible a una aplicación nativa
- Permitir el **trabajo sin conexión**
- Envío de **notificaciones Push**



Características

- **Instalables** (se pueden agregar a la pantalla de inicio de nuestros dispositivos sin necesidad de ir a la tienda)
- Son seguras (requieren **https** para funcionar)
- No es necesario pasar por la tienda de aplicaciones para actualizarla
- Al ser una aplicación web, la **puesta en producción** y el **mantenimiento** es relativamente **simple**



Desventajas y limitaciones

- ◉ Bajo **rendimiento** en comparación con una aplicación nativa
- ◉ No tiene **acceso** a todos los **recursos del dispositivo** (estamos limitados por el navegador)
- ◉ No permite la **interacción** con otras **aplicaciones** ya que se ejecuta en el entorno cerrado de un navegador web



Componentes de una PWA

App shell

- Esqueleto de la app
- Cantidad mínima de HTML, CSS y JS requerida para activar la interfaz del usuario.
- Los recursos o assets utilizados se pueden colocar en la caché

Service Worker

- Posibilitan el acceso y utilización off-line de la PWA
- Notificaciones PUSH
- Ejecución en segundo plano
- Diferentes estrategias de acceso al contenido

Web App Manifest

- Permite especificar metadatos de la app: nombre, color, ícono, etc.
- Posibilita la instalación en el dispositivo



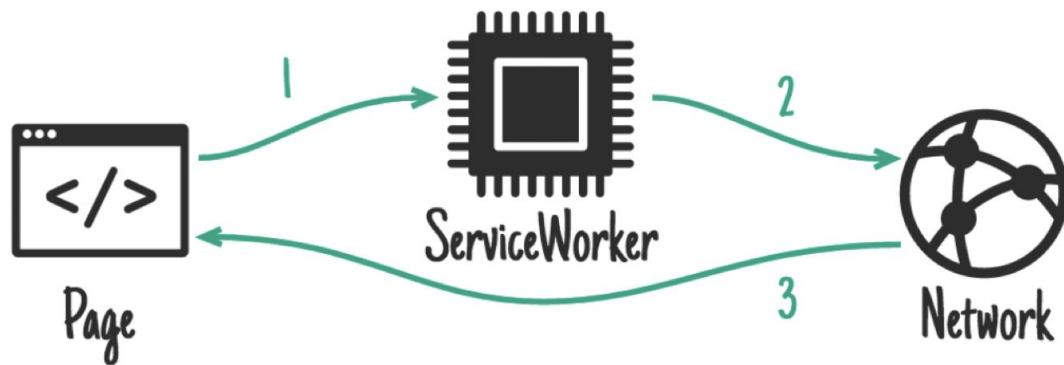
Estrategias de acceso a contenido

- ⦿ Solo red
- ⦿ Solo caché
- ⦿ Primero red, después caché
- ⦿ Primero caché, después red
- ⦿ Contenido estático



Solo red

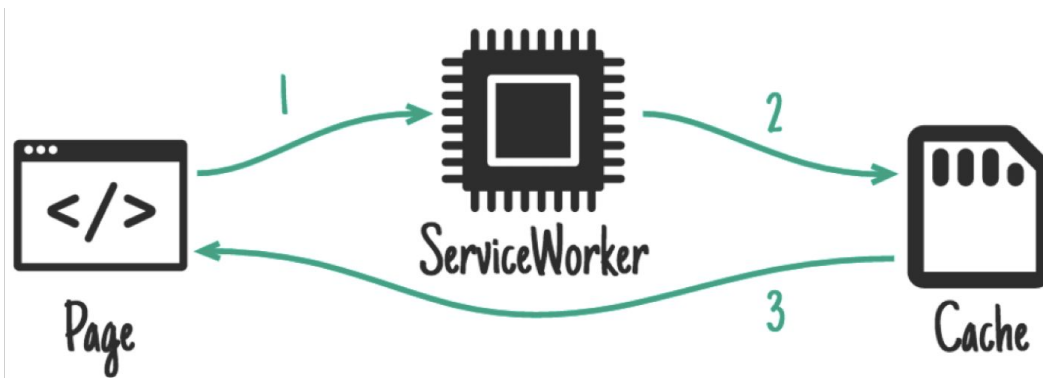
- El contenido es buscado en la red y presentado al usuario.
- Es utilizado para elementos que no tengan equivalente sin conexión





Solo caché

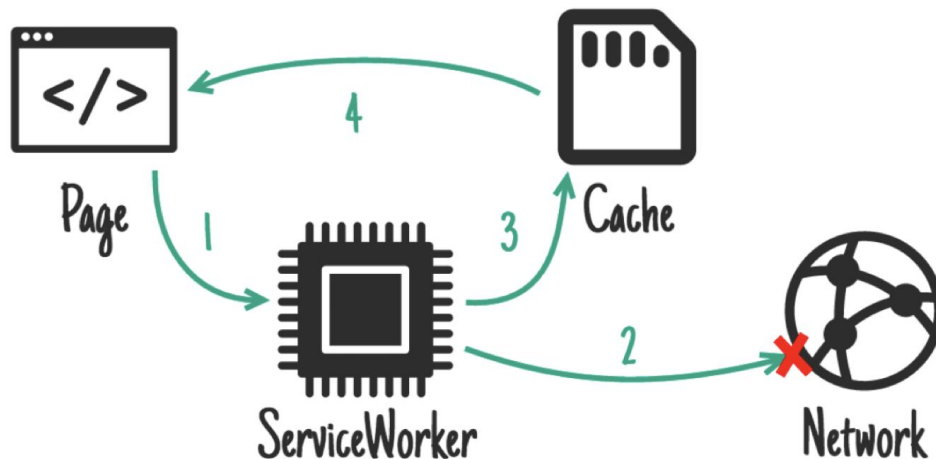
- Este patrón responde a todas las peticiones con archivos de **Cache Storage**.
- Si no lo encuentra fallará.





Primero red, después caché

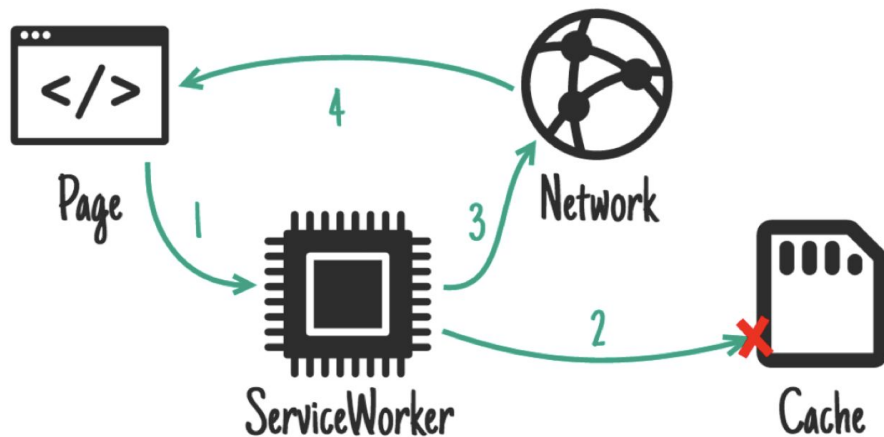
- Este patrón responde a todas las peticiones con el contenido de la **red**.
- Si falla intenta responder con el contenido del **Cache Storage**.





Primero caché, después red

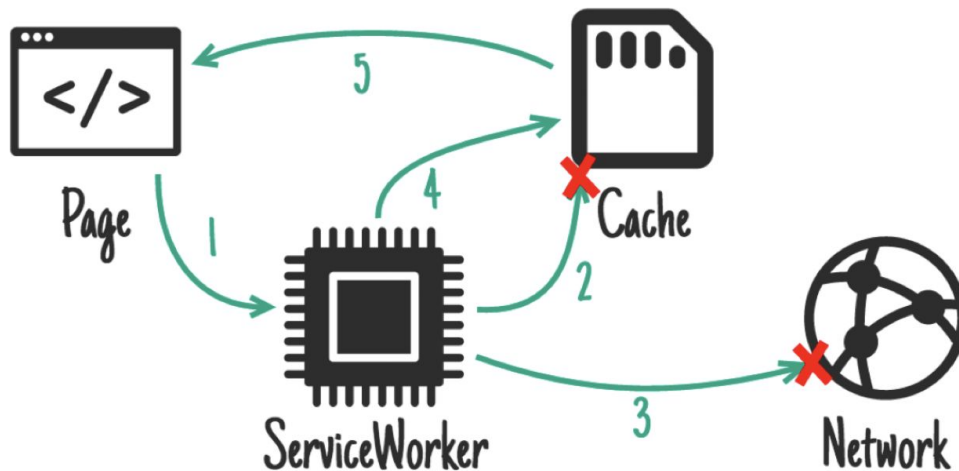
- Este patrón responde a las peticiones con archivos que se encuentran en el **Cache Storage**.
- Si falla la respuesta desde el Cache Storage intenta responder con los archivos desde la **Red**.





Contenido estático

- Si no se puede proporcionar un dato de la caché ni de la red, muestra un contenido estático.





Manifest

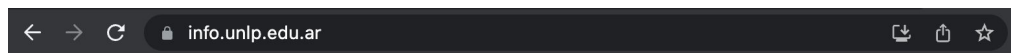
manifest.json

```
{
  "name": "Ionic",
  "short_name": "Ionic",
  "start_url": "index.html",
  "display": "standalone",
  "icons": [{
    "src":
"assets/imgs/logo.png",
    "sizes": "512x512",
    "type": "image/png"
  }],
  "background_color":
"#4e8ef7",
  "theme_color": "#4e8ef7"
}
```

- **Name:** Nombre de la app
- **Short_name:** Nombre corto de la app, se visualiza cuando no hay suficiente espacio en la pantalla
- **Start_url:** URL de acceso a la aplicación
- **Display:** Forma de preferencia de cómo la app se va a visualizar (full screen, portrait, etc)
- **Icons:** Conjunto de íconos de la app, dependiendo de la resolución de la pantalla
- **Background_color:** color de fondo de la app web
- **Theme_color:** color del tema de la app (afecta al color de la barra de notificaciones)



Ejemplo PWA Informática



FACULTAD DE INFORMÁTICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

INSTITUCIONAL ▾ CARRERAS ▾ UNIDADES DE I+D+I POSTGRADO CIYTT EXPO CIENCIA NOVEDADES

LINKS DE INTERÉS



CARRERAS

CARRERAS DE GRADO

Licenciatura en Informática

Licenciatura en Sistemas

Ingeniería en Computación

TITULACIONES DE 3 AÑOS

Analista Programador Universitario

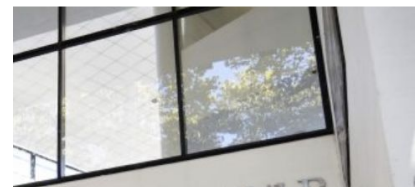
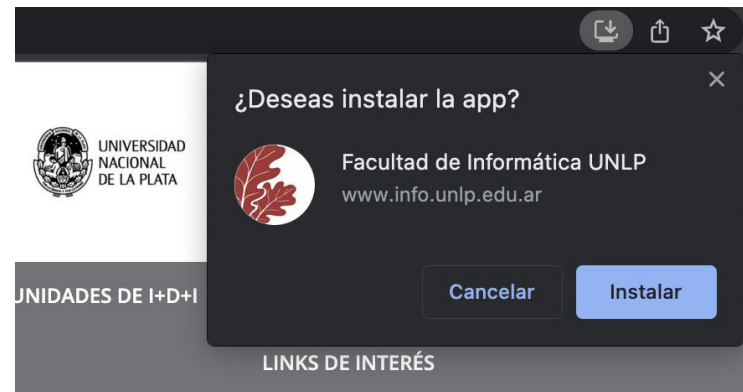
Analista en Tecnologías de la Información

DIPLOMATURAS

Diplomatura en Deportes Electivos

CARRERAS DE POSTGRADO

Postgrado Informática



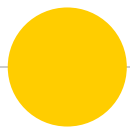
CARRERAS

CARRERAS DE GRADO

Licenciatura en Informática

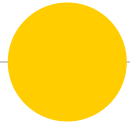
Licenciatura en Sistemas

Utilizando Ionic Framework + Angular para crear una PWA



<https://ionicframework.com/docs/angular/pwa>





Actividad

Seguir la documentación de Ionic para desplegar nuestra app desarrollada en las clases anteriores como una PWA.
<https://ionicframework.com/docs/angular/pwa>