

**Nombre: Omar Garcia Coronado**

**Matricula: 20608**

**Grupo: 10-B**

**18/05/25**

Actividad 1 – Tipos de Aplicaciones

Aplicaciones Nativas

**¿Cómo funcionan?**  
 Las aplicaciones nativas son desarrolladas específicamente para un sistema operativo, como Android o iOS. Esto significa que se escriben en lenguajes de programación compatibles con la plataforma objetivo: por ejemplo, Kotlin o Java para Android y Swift o Objective-C para iOS. Estas aplicaciones se instalan directamente desde las tiendas de aplicaciones (como Google Play o App Store) y pueden aprovechar al máximo las capacidades del dispositivo (como cámara, GPS, sensores, etc.).

**Ventajas:**

* Alto rendimiento y fluidez.
* Acceso completo a las funcionalidades del dispositivo.
* Mejor experiencia de usuario gracias a la integración con la interfaz nativa.
* Mayor seguridad.

**Desventajas:**

* Mayor costo de desarrollo, ya que se necesita una versión por cada sistema operativo.
* Más tiempo de desarrollo y mantenimiento.
* Necesidad de equipos con conocimientos específicos por plataforma.

Aplicaciones Híbridas

**¿Cómo se desarrollan?**  
 Las aplicaciones híbridas combinan tecnologías web (HTML, CSS y JavaScript) dentro de un contenedor nativo. Se desarrollan una sola vez y luego se adaptan para múltiples plataformas utilizando frameworks como Ionic, Cordova o React Native. Aunque se ejecutan dentro de una “webview” en el dispositivo, pueden acceder a funciones nativas a través de plugins.

**Diferencias frente a las nativas:**

* Menor rendimiento en comparación con las nativas puras, especialmente en aplicaciones que requieren muchos recursos (como juegos).
* Menor integración con el diseño y comportamiento específico de cada sistema operativo.
* Más rápida y económica de desarrollar para múltiples plataformas.

Aplicaciones Web Progresivas (PWA)

**¿Qué son y cómo se comportan?**  
 Las PWAs son aplicaciones web diseñadas para comportarse como aplicaciones nativas. Se accede a ellas desde el navegador, pero pueden instalarse en el dispositivo como una app convencional. Usan tecnologías modernas como Service Workers y Web App Manifests para ofrecer funcionalidades avanzadas como trabajo sin conexión, notificaciones push y carga rápida.

**Beneficios para el usuario:**

* No requieren instalación desde una tienda de aplicaciones.
* Funcionan offline o con mala conexión.
* Actualizaciones automáticas sin intervención del usuario.

**Beneficios para el desarrollador:**

* Un solo desarrollo para todas las plataformas.
* Menor costo de mantenimiento.
* Distribución más sencilla (sin necesidad de pasar por revisiones de tiendas).

Navegadores compatibles con PWA

La mayoría de los navegadores modernos son compatibles con PWA, aunque con diferentes niveles de soporte:

|  |  |
| --- | --- |
| **Navegador** | **Compatibilidad con PWA** |
| Google Chrome | Completa (Android, Windows, macOS, Linux) |
| Microsoft Edge | Completa |
| Firefox | Parcial (no permite instalación directa en escritorio) |
| Safari | Parcial (soporte limitado en iOS) |
| Opera | Buena compatibilidad |

Herramientas de desarrollo para PWAs

* **Lighthouse**: herramienta de auditoría de Google para evaluar el rendimiento y capacidades de una PWA.
* **Workbox**: librería para facilitar la implementación de Service Workers.
* **Webpack**: empaquetador de módulos muy usado en proyectos PWA.
* **Visual Studio Code**: editor de código muy popular con extensiones para desarrollo web y PWA.
* **PWA Builder**: herramienta de Microsoft que convierte sitios web en PWAs de forma sencilla.