Propuesta Técnica: Onboarding Interactivo con Lottie

Fabian Campoverde

Contexto

El onboarding es uno de los puntos más críticos en la experiencia de usuario de una aplicación. En este proyecto se propone un sistema de onboarding animado usando Lottie, enfocado en rendimiento, accesibilidad y capacidad de personalización sin necesidad de actualizar la aplicación desde las tiendas.

La solución está pensada para entornos reales donde se requiere flexibilidad visual, control técnico y mitigación de riesgos comunes como mareos, fallos de red o consumo excesivo de recursos.

Flujo de Trabajo Completo (After Effects → Producción)

El flujo propuesto garantiza una integración fiel al diseño original sin recodificar animaciones manualmente.

1. **Diseño y animación (After Effects):** El diseñador crea la animación utilizando únicamente vectores y transformaciones compatibles con Lottie. Esto evita errores de renderizado y reduce el peso final.

2. **Exportación (Bodymovin):** Se utiliza el plugin Bodymovin para exportar la animación en formato `.json` o `.lottie`, conservando fidelidad visual pixel-perfect.

3. **Optimización:** El archivo se sube a la plataforma LottieFiles para reducir precisión decimal, validar framerate y eliminar capas innecesarias.

4. **Despliegue en CDN:** El archivo final se aloja en un CDN (Amazon S3 o Cloudflare). Esto permite modificar la animación sin recompilar ni redistribuir la app.

5. **Integración en el cliente:** La aplicación consume la animación desde la URL usando librerías nativas como `lottie-web` o `lottie-react-native`.

Este flujo separa claramente diseño, optimización y despliegue, reduciendo errores y facilitando iteraciones rápidas.

Personalización Dinámica: A/B Testing por Red

En lugar de usar una única animación estática, se implementa una carga dinámica basada en segmentación de usuarios.

Caso concreto:

- **Grupo A (usuarios nuevos):** reciben una animación tutorial paso a paso.
- **Grupo B (usuarios recurrentes):** reciben una animación de bienvenida con promociones.

Implementación técnica:

- Al iniciar la app se consulta un endpoint de configuración.
- El backend responde con la URL del archivo Lottie correspondiente.
- El cliente descarga y renderiza esa animación específica.

Esta estrategia permite realizar A/B testing real, campañas estacionales y cambios visuales sin actualizar la aplicación en las tiendas.

Eficiencia e Interacción

Caching

Las animaciones Lottie pueden pesar entre 50 KB y 200 KB, por lo que descargarlas en cada inicio sería ineficiente.

Estrategia aplicada:

- Cache-First mediante headers `ETag` y `Cache-Control`.
- Uso de `cacheKey` en aplicaciones móviles.
- Invalidación automática solo cuando cambia la versión del archivo.

Esto reduce consumo de datos y mejora tiempos de carga.

Interacción y Control de Animación

La animación no se reproduce como un video pasivo. El usuario controla el progreso mediante gestos.

Ejemplo:

- Onboarding de 3 pantallas.
- Animación de 300 frames.
- Cada gesto controla un segmento específico de la animación.

Esto mejora la comprensión y da sensación de control directo al usuario.

Riesgos Reales y Mitigación

- **Accesibilidad / Mareos:** Se detecta `prefers-reduced-motion`. Si está activo, se desactiva el autoplay o se muestra solo un frame estático.
- **Fallo de red o JSON corrupto:** Se captura el evento de error y se reemplaza inmediatamente la animación por una imagen local de respaldo.
- **Peso excesivo y bajo rendimiento:** Se utiliza formato `.lottie` con compresión y se establece un límite máximo de peso en el pipeline CI/CD.

Estas mitigaciones aseguran estabilidad incluso en condiciones adversas.

Evidencia Técnica (Snippet)

```
async function initSmartOnboarding(userType) {
  const animUrl =
    userType === "NEW"
      ? "https://cdn.app.com/tutorial.json"
      : "https://cdn.app.com/promo.json";

  const reduceMotion = window.matchMedia(
    "(prefers-reduced-motion: reduce)"
  ).matches;

  try {
    const anim = lottie.loadAnimation({
      container: document.getElementById("lottie"),
      renderer: "svg",
      loop: false,
      autoplay: !reduceMotion,
      path: animUrl,
    });

    document.getElementById("slider")
      .addEventListener("input", e =>
        anim.goToAndStop(
          (e.target.value / 100) * anim.totalFrames,
          true
        )
      );
  } catch {
    document.getElementById("lottie")
      .innerHTML = "<img src='fallback.png'>";
  }
```

```
}
```

Conclusión

La propuesta combina diseño, ingeniería y experiencia de usuario de forma equilibrada. Se prioriza accesibilidad, rendimiento y control real, evitando animaciones innecesarias y soluciones rígidas. El enfoque es escalable, mantenible y aplicable a productos reales en producción.

Repositorio del Proyecto

El código fuente y los recursos del onboarding se encuentran disponibles en: `https://github.com/FabianSSJ/examen-dashboard`