

# Cambios en el Sistema de Anuncios - Implementación Mejorada

---

## Fecha de Implementación

---

10 de Noviembre de 2025

## Resumen Ejecutivo

---

Se ha realizado una refactorización completa del sistema de visualización de anuncios para mejorar la experiencia del usuario. Los cambios principales incluyen:

1. **Eliminación de la ventana flotante/modal** en la página principal
  2. **Implementación de contador en nueva pestaña** con barra de progreso visual
  3. **Integración de Page Visibility API** para pausar/reanudar el contador
  4. **Sistema de reclamación de puntos mejorado** con interfaz visual atractiva
- 

## Cambios Detallados

---

### 1. Nueva Página de Visualización de Anuncios

**Archivo creado:** `/app/ad-view/[adId]/page.tsx`

#### Características Principales:

- **Ruta dinámica:** Utiliza el parámetro `[adId]` para identificar el anuncio
- **Barra de progreso horizontal:** Ocupa todo el ancho de la pantalla en la parte superior
- **Estados visuales dinámicos:** Diferentes colores y mensajes según el estado actual
- **Contador de 10 segundos:** Tiempo requerido para poder reclamar puntos

## Estados de la Barra de Progreso:

Estado	Color	Descripción
Visualizando	Azul-Morado	Contador activo, mostrando tiempo restante
Pausado	Amarillo-Naranja	Pestaña no visible, contador congelado
Listo	Verde	Tiempo completado, botón de reclamar visible
Reclamado	Verde Oscuro	Puntos reclamados exitosamente
Bloqueado	Rojo	Ya visto en últimas 24 horas

## 2. Page Visibility API - Gestión Inteligente del Contador

### Implementación:

```
useEffect(() => {
  const handleVisibilityChange = () => {
    if (document.visibilityState === 'visible') {
      setIsPageVisible(true);
      // Reanudar desde donde se quedó
      startTimeRef.current = Date.now() - elapsedTimeRef.current * 1000;
    } else {
      setIsPageVisible(false);
      // Guardar tiempo transcurrido
      elapsedTimeRef.current = (Date.now() - startTimeRef.current) / 1000;
    }
  };

  document.addEventListener('visibilitychange', handleVisibilityChange);
  return () => document.removeEventListener('visibilitychange', handleVisibilityChange);
}, []);
```

### Beneficios:

- ☒ El contador se **pausa automáticamente** cuando el usuario cambia de pestaña
- ☒ El contador se **reanuda desde donde se quedó** cuando el usuario vuelve
- ☒ Previene el abuso del sistema
- ☒ Mejora la experiencia del usuario con feedback visual claro

## 3. Modificaciones en PublicidadSection.tsx

Archivo modificado: /components/PublicidadSection.tsx

### Cambios Realizados:

#### 1. Eliminada la importación de AdViewPage

```
```typescript
```

```
// ANTES
import AdViewPage from './AdViewPage';

// DESPUÉS
// Importación eliminada
...
```

### 1. Eliminados estados relacionados con el modal

```
``typescript
// ANTES
const [showAdViewPage, setShowAdViewPage] = useState(false);
const [selectedAdForView, setSelectedAdForView] = useState(null);

// DESPUÉS
// Estados eliminados
...
```

### 1. Función handleViewAd refactorizada

```
``typescript
const handleViewAd = (ad: UserAd) => {
  if (ad.userId === 'current_user') return;

  // Verificar límite diario
  if (hasReachedDailyLimit()) {
    alert(Has alcanzado el límite diario...);
    return;
  }

  // Guardar datos del anuncio en sessionStorage
  sessionStorage.setItem( `ad_${ad.id}`, JSON.stringify({
    id: ad.id,
    title: ad.title,
    description: ad.description,
    url: ad.url,
    imageUrl: ad.imageUrl,
  }));

  // Abrir nueva pestaña
  window.open( `ad-view/${ad.id}`, '_blank');
};
...
```

### 2. Eliminado el renderizado del modal AdViewPage

```
``typescript
// ANTES
{showAdViewPage && selectedAdForView && (

)}

// DESPUÉS
// Código eliminado
...
```

## 4. Sistema de Reclamación de Puntos

### Flujo de Reclamación:

1. **Verificación inicial:** Comprueba si el anuncio ya fue visto en las últimas 24h
2. **Contador de 10 segundos:** El usuario debe esperar mientras visualiza el anuncio
3. **Botón de reclamar:** Aparece al completar los 10 segundos
4. **Actualización automática:** Al hacer clic, suma 2 puntos al usuario
5. **Registro en localStorage:** Guarda la fecha de visualización para control de 24h
6. **Pestaña permanece abierta:** El usuario puede seguir navegando

### Código de Reclamación:

```
const handleClaimPoints = () => {
  if (!canClaim || pointsClaimed || isAlreadyClaimed) return;

  const now = new Date();
  const nextClaimTime = new Date(now.getTime() + 24 * 60 * 60 * 1000); // 24 horas

  // Guardar en localStorage
  const adViews = JSON.parse(localStorage.getItem('ad_views') || '[]');
  const newView = {
    adId,
    userId: 'current_user',
    viewedAt: now.toISOString(),
    pointsClaimed: true,
    nextClaimTime: nextClaimTime.toISOString(),
  };

  // Actualizar puntos del usuario (+2 puntos)
  const userData = JSON.parse(localStorage.getItem('user_simulation_data') || '{}');
  userData.points = (userData.points || 0) + 2;
  localStorage.setItem('user_simulation_data', JSON.stringify(userData));

  // Actualizar tracking diario
  const today = new Date().toISOString().split('T')[0];
  userData.dailyAdTracking = {
    date: today,
    adsViewed: (userData.dailyAdTracking?.adsViewed || 0) + 1,
  };
  localStorage.setItem('user_simulation_data', JSON.stringify(userData));







  setPointsClaimed(true);
  alert('¡Has ganado 2 puntos exitosamente! La pestaña permanecerá abierta para que sigas navegando.');
```

## Mejoras en la Experiencia del Usuario

### Antes de los Cambios:

- ❌ Modal flotante bloqueaba la página principal
- ❌ Contador visible en la página principal
- ❌ No había control si el usuario cambiaba de pestaña
- ❌ Experiencia poco intuitiva

## Después de los Cambios:

-  Nueva pestaña dedicada para cada anuncio
-  Barra de progreso visual y atractiva
-  Contador se pausa automáticamente al cambiar de pestaña
-  Feedback visual claro en cada estado
-  Pestaña permanece abierta para seguir navegando
-  Instrucciones claras para el usuario

---

## Características Técnicas

### Tecnologías Utilizadas:

- **React Hooks:** useState, useEffect, useRef
- **Next.js App Router:** Rutas dinámicas con parámetros
- **Page Visibility API:** Control de visibilidad de pestaña
- **LocalStorage & SessionStorage:** Persistencia de datos
- **CSS Gradients & Animations:** Interfaz visual atractiva

### Optimizaciones:

- **Intervalo de 100ms:** Actualización suave del contador
- **Referencias con useRef:** Manejo eficiente del tiempo
- **Cleanup de efectos:** Prevención de memory leaks
- **Validaciones múltiples:** Prevención de reclamaciones duplicadas

---

## Sistema de Validación y Control

### 1. Validación de 24 Horas:

```
// Verifica si el anuncio ya fue visto
const existingView = adViews.find(
  (view) => view.adId === adId && view.userId === 'current_user'
);





if (existingView) {
  const nextClaim = new Date(existingView.nextClaimTime);
  const now = new Date();

  if (nextClaim > now) {
    setIsAlreadyClaimed(true);
    // Mostrar tiempo restante
  }
}
```

### 2. Control de Límite Diario:

- Integrado con el sistema existente de `dailyAdsLimit`
- Actualiza automáticamente el contador diario al reclamar
- Muestra tiempo hasta medianoche cuando se alcanza el límite

### 3. Prevención de Abuso:

-  No permite reclamar si ya fue visto en 24h
  -  Pausa el contador cuando la pestaña no está visible
  -  Valida el estado antes de permitir reclamación
  -  Registra timestamp exacto de cada visualización
- 




## Interfaz de Usuario

---

### Componentes de la Barra de Progreso:

1. **Barra Visual (3px altura):**
  - Muestra progreso de 0% a 100%
  - Gradiente animado según el estado
  - Efecto de brillo al completarse
2. **Información del Estado:**
  - Icono animado según el estado
  - Texto descriptivo claro
  - Contador grande y visible
3. **Botón de Reclamación:**
  - Aparece al completar 10 segundos
  - Diseño llamativo con animación
  - Efecto hover para interactividad

### Responsive Design:

-  **Mobile:** Texto adaptado, iconos ajustados
  -  **Desktop:** Layout espacioso, información completa
  -  **Themes:** Compatible con tema claro y oscuro
- 

## Instrucciones para el Usuario

---

La página incluye una sección de instrucciones que explica:

1. El contador se pausará automáticamente si cambias de pestaña
  2. Debes esperar 10 segundos para poder reclamar tus puntos
  3. Una vez reclamados, la pestaña permanecerá abierta para seguir navegando
  4. Podrás ver este anuncio nuevamente en 24 horas
- 

## Testing y Depuración

---

### Modo de Desarrollo:

- Muestra información de debug en la parte inferior de la barra
- Variables visibles: `isVisible` , `progress` , `timeRemaining` , `canClaim`

- Solo visible cuando `NODE_ENV === 'development'`

## Puntos de Prueba:

1. ☒ Abrir anuncio y esperar 10 segundos
  2. ☒ Cambiar de pestaña y verificar pausa
  3. ☒ Volver y verificar reanudación
  4. ☒ Reclamar puntos y verificar suma
  5. ☒ Intentar ver el mismo anuncio inmediatamente
  6. ☒ Verificar límite diario
- 

## Archivos Modificados/Creados

---

### Archivos Creados:

1. `/app/ad-view/[adId]/page.tsx` - Página de visualización de anuncios

### Archivos Modificados:

1. `/components/PublicidadSection.tsx` - Eliminación del modal y actualización del handler

### Archivos No Modificados (pero relacionados):

- `/components/AdViewPage.tsx` - Ya no se usa, puede eliminarse en el futuro
  - `/components/AdProgressBar.tsx` - Ya no se usa, puede eliminarse en el futuro
- 

## Compatibilidad

---

### Navegadores Soportados:

- ☒ Chrome/Edge (>90)
- ☒ Firefox (>88)
- ☒ Safari (>14)
- ☒ Opera (>76)

### Page Visibility API:

- Soportada por todos los navegadores modernos
  - Fallback: Si no está disponible, el contador funciona normalmente
- 

## Próximas Mejoras Sugeridas

---






1. **Analytics:** Agregar tracking de visualizaciones completas
2. **Notificaciones:** Usar toast notifications en lugar de alerts
3. **Progreso persistente:** Guardar progreso del contador en localStorage
4. **Animaciones:** Más efectos visuales en transiciones
5. **Sonidos:** Feedback auditivo al completar o reclamar
6. **Preview del anuncio:** Mostrar miniatura del sitio web

---

## Conclusión

---

Esta implementación mejora significativamente la experiencia del usuario al visualizar anuncios, proporcionando:

-  Mayor claridad visual
-  Mejor control del tiempo de visualización
-  Prevención de abuso del sistema
-  Feedback constante al usuario
-  Integración perfecta con el sistema existente

El sistema es robusto, fácil de usar y mantiene la compatibilidad completa con las funcionalidades existentes de límites diarios y sistema de puntos.

---

## Contacto y Soporte

---

Para cualquier pregunta o problema relacionado con estos cambios, consultar la documentación técnica o contactar al equipo de desarrollo.

**Versión del Sistema:** 2.0.0

**Última Actualización:** 10 de Noviembre de 2025