# Sistema Completo de Anuncios - Documentación de Implementación

## 📋 Resumen de la Implementación

Se ha implementado exitosamente un sistema completo de anuncios con límites diarios por rango de usuario, reseteo automático a medianoche, y funcionalidad mejorada de visualización de anuncios con contador de 10 segundos.

## 🎯 Especificaciones Implementadas

### 1. Sistema de Límites Diarios por Rango de Usuario

Los límites diarios de anuncios que puede ver cada usuario según su rango:

| Rango | Límite Diario | Ubicación en Código |
|---|---|---|
| Registrado | 5 anuncios/día | `hooks/useSimulation.ts:127` |
| Invitado | 10 anuncios/día | `hooks/useSimulation.ts:139` |
| Básico | 10 anuncios/día | `hooks/useSimulation.ts:157` |
| VIP | 10 anuncios/día | `hooks/useSimulation.ts:174` |
| Premium | 15 anuncios/día | `hooks/useSimulation.ts:191` |
| Elite | 20 anuncios/día | `hooks/useSimulation.ts:208` |

**Código modificado:**

```
// hooks/useSimulation.ts - Definición de rangos con límites diarios
dailyAdsLimit: 5, // para registrado
dailyAdsLimit: 10, // para invitado, basico, vip
dailyAdsLimit: 15, // para premium
dailyAdsLimit: 20, // para elite
```

### 2. Sistema de Puntos Actualizado

- **Puntos por anuncio:** 2 puntos (anteriormente 5)
- **Duración del anuncio:** 10 segundos (anteriormente 5)

**Archivos modificados:**

- `components/AdViewPage.tsx:26` - Timer de 10 segundos
- `components/AdViewPage.tsx:64` - Progreso al 100% en 10 segundos
- `components/AdViewPage.tsx:118,195,243` - Mensajes de “2 puntos”
- `hooks/useSimulation.ts:618` - Otorgar 2 puntos al reclamar

## 3. Tracking Diario con Reseteo Automático

**Nueva interface implementada:**

```
// hooks/useSimulation.ts:56-60
export interface DailyAdTracking {
  date: string; // formato YYYY-MM-DD
  adsViewed: number;
  lastResetTime: string;
}
```

**Lógica de reseteo:**
- Se verifica automáticamente al cargar la aplicación
- Si la fecha del tracking no coincide con la fecha actual, se resetea a 0
- Se guarda en `localStorage` bajo la clave `daily_ad_tracking`

**Ubicación en código:**
- `hooks/useSimulation.ts:384-398` - Verificación y reseteo automático
- `hooks/useSimulation.ts:558-650` - Lógica de verificación en claimAdPoints

## 4. Sistema de Mensajes y UI

**Mensajes implementados:**

1. **Contador de anuncios disponibles:**
   - Muestra: “Anuncios disponibles hoy: X / Y”
   - Cambia de color (verde/rojo) según disponibilidad
   - Ubicación: `components/PublicidadSection.tsx:321-363`

2. **Mensaje de límite alcanzado:**
   - Texto: “Límite diario alcanzado”
   - Submensaje: “Los anuncios se habilitarán después de las 00:00”
   - Muestra tiempo restante en formato: “Xh Ym Zs”
   - Ubicación: `components/PublicidadSection.tsx:342-362`

3. **Botones deshabilitados:**
   - Se deshabilita el botón “Ver Anuncio” cuando se alcanza el límite
   - Muestra “Límite diario alcanzado” en rojo
   - Ubicación: `components/PublicidadSection.tsx:445-479`

## 5. Gestión de Anuncios

**Anuncios de muestra implementados:**

Se agregaron 15 anuncios de muestra (antes 5) que cubren diferentes categorías y rangos:

| ID | Categoría | Rango | Usuario |
|---|---|---|---|
| sample_1 | Marketing Digital | VIP | María González |
| sample_2 | Trading/Crypto | Premium | Carlos Mendoza |
| sample_3 | Salud Natural | Básico | Ana Rodriguez |
| sample_4 | Consultoría | Elite | Roberto Silva |
| sample_5 | Fitness | Invitado | Laura Pérez |
| sample_6 | Programación | Premium | Diego Fernández |
| sample_7 | Diseño Gráfico | VIP | Patricia Torres |
| sample_8 | eBook Inversión | Registrado | Javier Morales |
| sample_9 | Fotografía | Básico | Sofía Ramírez |
| sample_10 | E-commerce | Elite | Miguel Ángel Castro |
| sample_11 | Yoga/Meditación | Invitado | Valeria Herrera |
| sample_12 | Influencers | Premium | Fernando Ortiz |
| sample_13 | Software Empresarial | VIP | Camila Vega |
| sample_14 | Idiomas | Básico | Ricardo Domínguez |
| sample_15 | Ciberseguridad | Elite | Daniela Flores |

**Ubicación:** `components/PublicidadSection.tsx:119-365`

## 6. Almacenamiento de Datos

El sistema utiliza el modo de simulación existente con `localStorage`:

**Claves de almacenamiento:**
- `user_simulation_data` - Datos generales del usuario (puntos, rango, etc.)
- `daily_ad_tracking` - Tracking diario de anuncios vistos
- `ad_views` - Historial de anuncios vistos con timestamps
- `user_ads` - Anuncios creados por el usuario

**Sistema transaccional:**
- Utiliza `simStorage` para operaciones atómicas
- Evita condiciones de carrera con sistema de locks
- Preserva datos existentes al actualizar

# 📁 Archivos Modificados

## 1. `hooks/useSimulation.ts`

**Cambios principales:**
- Nueva interface `DailyAdTracking`
- Campo `dailyAdsLimit` en `UserRank`
- Campo `dailyAdTracking` en `SimulationState`
- Actualización de límites diarios en RANKS
- Lógica de verificación en `loadSimulationData`
- Modificación de `claimAdPoints` para verificar límite diario
- Cambio de 5 a 2 puntos por anuncio

## 2. `components/AdViewPage.tsx`

**Cambios principales:**
- Timer cambiado de 5 a 10 segundos
- Progreso actualizado para completar en 10 segundos
- Mensajes actualizados de “5 puntos” a “2 puntos”

## 3. `components/PublicidadSection.tsx`

**Cambios principales:**
- Nueva funcionalidad de cálculo de tiempo hasta medianoche
- Funciones `hasReachedDailyLimit()` y `getAdsRemainingToday()`
- Sección de información de límite diario
- Mensaje de límite alcanzado con contador
- Validación en `handleViewAd` para verificar límite
- Botones actualizados con verificación de límite
- 15 anuncios de muestra (antes 5)

# 🎨 Características de la UI

## Indicadores Visuales

1. **Tarjeta de información:**
   - Fondo azul claro/gris oscuro según tema
   - Muestra anuncios disponibles con colores:
     - Verde: Hay anuncios disponibles
     - Rojo: Límite alcanzado
2. **Alerta de límite alcanzado:**
   - Fondo rojo con borde
   - Ícono de reloj
   - Mensaje claro con tiempo restante
   - Actualización en tiempo real cada segundo
3. **Botones de anuncios:**
   - Azul: Anuncio disponible para ver
   - Gris: Anuncio bloqueado 24h (ya visto)
   - Rojo: Límite diario alcanzado

# 🔧 Funcionamiento Técnico

## Flujo de Visualización de Anuncios

1. Usuario hace clic en “Ver Anuncio”
2. Sistema verifica:
   - ¿Es un nuevo día? → Resetear contador
   - ¿Alcanzó el límite diario? → Mostrar mensaje de error
   - ¿Ya vio este anuncio en las últimas 24h? → Mostrar bloqueado
3. Si pasa todas las verificaciones:
   - Abrir anuncio en nueva pestaña
   - Iniciar contador de 10 segundos
   - Mostrar progreso visual
4. Al completar 10 segundos:
   - Habilitar botón “Reclamar 2 Puntos”
   - Usuario hace clic para reclamar
5. Al reclamar:
   - Sumar 2 puntos al usuario
   - Incrementar contador diario (+1)
   - Registrar anuncio como visto
   - Guardar en localStorage

## Sistema de Reseteo Automático

```
// Pseudocódigo del reseteo
function checkAndResetDailyTracking() {
  const today = getTodayDate(); // YYYY-MM-DD
  const savedTracking = loadFromStorage();

  if (savedTracking.date !== today) {
    // Es un nuevo día, resetear
    newTracking = {
      date: today,
      adsViewed: 0,
      lastResetTime: now()
    };
    saveToStorage(newTracking);
  }
}
```

# 🧪 Pruebas Recomendadas

## Pruebas Funcionales

1. **Verificar límites por rango:**
   - Registrarse y verificar límite de 5 anuncios
   - Ascender a cada rango y verificar límites correspondientes

2. **Probar contador de 10 segundos:**
   - Verificar que el progreso dura exactamente 10 segundos
   - Confirmar que el botón se habilita al completar

3. **Probar sistema de puntos:**
   - Ver un anuncio y verificar que otorga 2 puntos
   - Verificar que los puntos se suman correctamente

4. **Probar reseteo a medianoche:**
   - Cambiar fecha del sistema y recargar
   - Verificar que el contador se resetea

5. **Probar mensajes de límite:**
   - Alcanzar el límite diario
   - Verificar mensaje y contador de tiempo restante

### Pruebas de Edge Cases

1. Cambio de rango durante el día
2. Múltiples pestañas abiertas simultáneamente
3. Actualización de página durante visualización de anuncio
4. Cambio de fecha del sistema

## 📊 Métricas y Estadísticas

El sistema permite rastrear:
- Anuncios vistos por día
- Puntos ganados por día
- Anuncios más populares
- Rendimiento por rango de usuario

## 🔒 Seguridad y Validación

**Validaciones implementadas:**
- Verificación de límite diario en frontend y backend
- Prevención de múltiples reclamaciones del mismo anuncio
- Validación de timestamp para evitar manipulación
- Sistema transaccional para evitar pérdida de datos

## 🚀 Próximas Mejoras Sugeridas

1. **Backend real:** Migrar de localStorage a base de datos
2. **Analytics:** Dashboard de estadísticas de anuncios
3. **Moderación:** Sistema de aprobación de anuncios
4. **Reportes:** Sistema de reportes de anuncios inapropiados
5. **Notificaciones:** Alertas cuando se resetea el límite diario
6. **Bonus:** Multiplicadores de puntos en eventos especiales
7. **Gamificación:** Logros por ver cierto número de anuncios

## 📝 Notas de Desarrollo

- El sistema usa modo simulación sin base de datos real
- Los datos se guardan en localStorage del navegador
- El sistema es compatible con temas claro/oscuro

- Responsive design implementado para móviles y escritorio
- Iconos de RemixIcon utilizados para consistencia visual

## 🔄 Control de Versiones

**Commits realizados:**
1. Estado inicial del proyecto
2. Agregar .gitignore
3. Implementar sistema completo de anuncios con límites diarios

**Branch:** master

## 📞 Soporte

Para preguntas o problemas con el sistema de anuncios:
- Revisar este documento primero
- Consultar los comentarios en el código
- Verificar logs en consola del navegador (con `__simStorageDebug = true`)

---

**Fecha de implementación:** Noviembre 10, 2025
**Versión:** 1.0.0
**Estado:** ✅ Completado y funcional