

# Instrucciones de Testing - Sistema de Límites de Anuncios

## Inicio Rápido

### 1. Instalar Dependencias

```
cd /home/ubuntu/code_artifacts
npm install
```

### 2. Configurar Variables de Entorno

```
# Crear archivo .env basado en .env.example
cp .env.example .env

# Configurar la base de datos (SQLite por defecto)
echo "DATABASE_URL=file:./dev.db" >> .env
```

### 3. Inicializar Base de Datos

```
# Generar cliente de Prisma
npx prisma generate

# Crear las tablas (opcional, ya usa SQLite)
npx prisma db push
```

### 4. Iniciar el Servidor de Desarrollo

```
npm run dev
```

El proyecto estará disponible en: **http://localhost:3000**

**⚠ IMPORTANTE:** Este localhost se refiere al localhost de la computadora donde se está ejecutando el servidor, no tu máquina local. Para acceder localmente o remotamente, necesitarás desplegar la aplicación en tu propio sistema.

## Tests Manuales

### Test 1: Verificar Límites por Rango (10 minutos)

#### Objetivo:

Verificar que los límites diarios se respeten según el rango del usuario.

#### Pasos:

1. Abrir la aplicación: <http://localhost:3000>
2. Navegar a: Sección “Publicidad” > “Ver Anuncios”

**3. Observar el contador:** Debe mostrar “5/5 anuncios disponibles hoy” (para rango Registrado)

**4. Ver primer anuncio:**

- Clic en “Ver Anuncio (2 pts)”
- Esperar 10 segundos
- Clic en “Reclamar 2 Puntos”

**5. Verificar actualización:**

- Contador debe cambiar a: “4/5 anuncios disponibles hoy”
- Puntos incrementan de 0 a 2

**6. Repetir hasta alcanzar límite:**

- Ver anuncios hasta que el contador sea “0/5”

**7. Verificar bloqueo:**

- Todos los botones deben mostrar: “Límite diario alcanzado”
- Debe aparecer alerta roja con temporizador
- Intentar clic en “Ver Anuncio” → No debe funcionar

**Resultado Esperado:**

- Contador actualiza correctamente
  - Botones se deshabilitan al llegar a 0
  - Mensaje de límite alcanzado visible
  - Temporizador hasta medianoche funcionando
- 

## Test 2: Reset Automático a Medianoche (5 minutos)

**Objetivo:**

Verificar que el contador se resetea automáticamente a las 00:00.

**Método 1: Simulación con DevTools**

```
// 1. Abrir consola del navegador (F12)
// 2. Ejecutar este código para simular un nuevo día:

const today = new Date().toISOString().split('T')[0];
const tracking = {
  date: today,
  adsViewed: 0,
  lastResetTime: new Date().toISOString()
};
localStorage.setItem('daily_ad_tracking', JSON.stringify(tracking));
location.reload();
```

**Método 2: Cambiar Hora del Sistema (Avanzado)**

1. Alcanzar límite diario (5/5 anuncios)
2. Cambiar hora del sistema a 23:59:50
3. Esperar 10 segundos hasta medianoche
4. Verificar reset automático

**Resultado Esperado:**

- Contador se resetea a “5/5 anuncios disponibles hoy”
- Página se recarga automáticamente

- Botones se habilitan nuevamente
  - Log en consola: “ Reset automático a medianoche”
- 

## Test 3: Ascenso de Rango (7 minutos)

### Objetivo:

Verificar que los límites aumentan al ascender de rango.

### Pasos:

#### 1. Estado inicial:

- Rango: Registrado
- Límite: 5 anuncios/día
- Contador: “5/5 anuncios disponibles hoy”

#### 2. Ascender a Invitado:

- Ir a sección “Ascender”
- Pagar \$5 USD para ascender
- Verificar notificación de ascenso

#### 3. Verificar nuevo límite:

- Volver a “Publicidad”
- Contador debe mostrar: “10/10 anuncios disponibles hoy”
- Plan debe mostrar: “Plan Invitado”

#### 4. Ver anuncios:

- Ahora puede ver 10 anuncios en total
- Verificar que contador actualiza correctamente

#### 5. Repetir para otros rangos:

- **Básico:** 15 anuncios/día (\$10)
- **VIP:** 20 anuncios/día (\$50)
- **Premium:** 30 anuncios/día (\$400)
- **Elite:** ∞ Ilimitados (\$6000)

### Resultado Esperado:

- Límite aumenta según el rango
  - Contador muestra el nuevo límite
  - Plan actualizado en UI
  - Para Elite: “∞ Ilimitado”
- 

## Test 4: Usuario Elite (3 minutos)

### Objetivo:

Verificar que los usuarios Elite no tienen límites.

### Pasos:

#### 1. Ascender a Elite:

- Balance mínimo requerido: \$6000

- Ir a "Ascender" > Elite
- Pagar \$6000

## 2. Verificar UI:

- Contador debe mostrar: "∞ Ilimitado"
- Plan: "Plan Elite"
- Color morado en el badge

## 3. Ver anuncios sin límite:

- Ver más de 30 anuncios
- Verificar que nunca se bloquea
- Puntos siguen acumulándose

## 4. Verificar en código:

```
javascript
// En consola del navegador:
const tracking = JSON.parse(localStorage.getItem('daily_ad_tracking'));
console.log(tracking);
// Debe mostrar adsViewed > 30 sin problemas
```

## Resultado Esperado:

- UI muestra "∞ Ilimitado"
  - Puede ver anuncios sin restricción
  - Contador de anuncios vistos sigue incrementando
  - Nunca se bloquea
- 

## Test 5: Validación de Puntos (5 minutos)

### Objetivo:

Verificar que no se otorgan puntos después de alcanzar el límite.

### Pasos:

#### 1. Puntos iniciales:

- Anotar puntos actuales (ej: 0 puntos)

#### 2. Ver anuncios hasta el límite:

- Ver 5 anuncios (rango Registrado)
- Cada anuncio: +2 puntos
- Total esperado: 10 puntos

#### 3. Intentar ver anuncio 6:

- Clic en "Ver Anuncio" → Debe estar deshabilitado
- Intentar reclamar puntos → No debe funcionar

#### 4. Verificar puntos:

- Puntos deben seguir en 10
- No deben incrementar después del límite

#### 5. Verificar en consola:

```
javascript
```

```
// Debe aparecer este log:  
// [useSimulation] ! Límite diario alcanzado: 5/5
```

### Resultado Esperado:

- Puntos se otorgan correctamente (2 por anuncio)
  - No se otorgan puntos después del límite
  - Log de advertencia en consola
  - Total de puntos: límite\_diario × 2
- 

## Test 6: Persistencia de Datos (3 minutos)

### Objetivo:

Verificar que el tracking persiste al recargar la página.

### Pasos:

#### 1. Ver 3 anuncios:

- Contador: "2/5 anuncios disponibles hoy"
- Puntos: 6

#### 2. Recargar página (F5)

#### 3. Verificar estado:

- Contador debe seguir en: "2/5 anuncios disponibles hoy"
- Puntos deben seguir en: 6
- Anuncios vistos: 3

#### 4. Cerrar y abrir navegador:

- Cerrar pestaña completamente
- Abrir nueva pestaña en http://localhost:3000
- Verificar que el estado persiste

### Resultado Esperado:

- Datos persisten al recargar
  - localStorage mantiene el tracking
  - Estado consistente entre sesiones
- 



## Debugging

### Habilitar Logs de Debug:

```
// En consola del navegador:  
window.__simStorageDebug = true;  
  
// Ahora verás logs detallados:  
// [useSimulation] 📺 Puntos reclamados de anuncio: {...}  
// [useSimulation] 🚀 Ascenso de rango: {...}
```

## Inspeccionar localStorage:

```
// Ver tracking diario:  
console.log(JSON.parse(localStorage.getItem('daily_ad_tracking')));  
  
// Ver vistas de anuncios:  
console.log(JSON.parse(localStorage.getItem('ad_views')));  
  
// Ver datos de usuario:  
console.log(JSON.parse(localStorage.getItem('user_simulation_data')));
```

## Resetear Estado Manualmente:

```
// ⚠ CUIDADO: Esto borra todos los datos  
localStorage.clear();  
location.reload();
```

## Cambiar Fecha Manualmente:

```
// Simular día anterior (para testing de reset):  
const yesterday = new Date();  
yesterday.setDate(yesterday.getDate() - 1);  
const tracking = {  
    date: yesterday.toISOString().split('T')[0],  
    adsViewed: 5,  
    lastResetTime: yesterday.toISOString()  
};  
localStorage.setItem('daily_ad_tracking', JSON.stringify(tracking));  
location.reload();  
  
// Ahora al cargar, debe detectar nuevo día y resetear
```



## Checklist de Testing

### Funcionalidades Básicas:

- [ ] Sistema carga correctamente
- [ ] UI muestra contador de anuncios
- [ ] Botones de “Ver Anuncio” funcionan
- [ ] Contador actualiza al ver anuncios
- [ ] Puntos se otorgan correctamente (2 por anuncio)

### Límites por Rango:

- [ ] Registrado: 5 anuncios/día
- [ ] Invitado: 10 anuncios/día
- [ ] Básico: 15 anuncios/día
- [ ] VIP: 20 anuncios/día
- [ ] Premium: 30 anuncios/día
- [ ] Elite: Ilimitados

## Bloqueo al Límite:

- [ ] Botones se deshabilitan al llegar a 0
- [ ] Mensaje “Límite diario alcanzado” visible
- [ ] Temporizador hasta medianoche funciona
- [ ] No se otorgan puntos después del límite
- [ ] Log de advertencia en consola

## Reset Automático:

- [ ] Reset al cargar datos (useSimulation)
- [ ] Reset en tiempo real (cada segundo)
- [ ] Reset al reclamar anuncios
- [ ] Página se recarga automáticamente
- [ ] Log de confirmación en consola

## UI/UX:

- [ ] Contador muestra “X/Y anuncios disponibles hoy”
- [ ] Elite muestra “∞ Ilimitado”
- [ ] Alerta roja cuando se alcanza límite
- [ ] Temporizador actualiza cada segundo
- [ ] Animaciones y transiciones suaves

## Persistencia:

- [ ] Datos persisten al recargar página
- [ ] Estado consistente entre sesiones
- [ ] localStorage actualizado correctamente

## Validación:

- [ ] Validación frontend funciona
- [ ] API backend responde correctamente
- [ ] No hay errores en consola
- [ ] No hay warnings de TypeScript

## Casos Edge a Probar

### 1. Cambio de Hora del Sistema:

- ¿Qué pasa si el usuario cambia la hora del sistema?
-  El sistema detecta el cambio de fecha y resetea

### 2. Multiple Tabs:

- ¿Qué pasa si se abre la aplicación en múltiples pestañas?
-  El último que guarda en localStorage gana (expected)

### 3. Sin Internet:

- ¿El sistema funciona offline?
-  Sí, todo está en localStorage

## 4. Caché del Navegador:

- ¿El sistema funciona con caché agresivo?
- Sí, usa localStorage que no depende de caché

## 5. Ascenso de Rango Durante el Día:

- ¿El límite aumenta inmediatamente?
- Sí, el límite se actualiza al instante
- Contador muestra el nuevo límite



## Reporte de Bugs

Si encuentras un bug, reporta con esta información:

### ### 🐞 Bug Report

#### \*\*Descripción:\*\*

[Descripción breve del problema]

#### \*\*Pasos para reproducir:\*\*

1. [Paso 1]
2. [Paso 2]
3. [Paso 3]

#### \*\*Resultado esperado:\*\*

[Qué debería pasar]

#### \*\*Resultado actual:\*\*

[Qué está pasando]

#### \*\*Logs de consola:\*\*

[Pegar logs aquí]

```
**localStorage:**  
```json  
{  
  "daily_ad_tracking": {...},  
  "ad_views": [...],  
  "user_simulation_data": {...}  
}
```

### Entorno:

- Navegador: [Chrome/Firefox/Safari/Edge]
  - Versión: [Ej: Chrome 119]
  - OS: [Windows/Mac/Linux]
  - Rango de usuario: [registrado/invitado/etc]
- ```

## Resultados Esperados (Resumen)

Al completar todos los tests, debes poder confirmar:

1.  **Límites funcionan** para todos los rangos
2.  **Reset automático** a medianoche operativo
3.  **UI actualiza** correctamente
4.  **Puntos se otorgan** solo dentro del límite
5.  **Elite tiene acceso ilimitado**
6.  **Datos persisten** entre sesiones
7.  **Validación dual** (frontend + backend)
8.  **Sin errores** en consola

---

## Próximo Paso

Una vez completados todos los tests:

1.  Marcar todos los checkboxes
2.  Documentar cualquier bug encontrado
3.  Desplegar a producción

---

**Fecha:** 13 de Noviembre, 2025

**Versión:** 1.0.0

**Testing Status:**  Listo para testing