

# Instrucciones de Ejecución del Sistema de Anuncios

---

## Requisitos Previos

---

- Node.js 18+ instalado
- npm o yarn instalado
- Git (opcional)

## Instalación

---

### 1. Navegar al directorio del proyecto

```
cd /home/ubuntu/code_artifacts/sistema-anuncios
```

### 2. Instalar dependencias

```
npm install  
# o  
yarn install
```

### 3. Configurar base de datos

```
# Aplicar migraciones de Prisma  
npx prisma migrate dev  
  
# Generar cliente de Prisma  
npx prisma generate
```

### 4. (Opcional) Poblar base de datos con datos de prueba

```
npx prisma db seed
```

## Ejecución en Modo Desarrollo

---

```
npm run dev  
# o  
yarn dev
```

La aplicación estará disponible en: `http://localhost:3000`

## Ejecución en Modo Producción

```
# Compilar el proyecto
npm run build

# Ejecutar en producción
npm start
```

## Estructura del Proyecto

```
sistema-anuncios/
├── app/
│   ├── ad-view/
│   │   ├── [adId]/
│   │   │   └── page.tsx # ★ Página del anuncio (MODIFICADO)
│   │   ├── api/
│   │   │   ├── login/
│   │   │   └── page.tsx
│   │   └── ...
│   ├── components/
│   │   ├── PublicidadSection.tsx # ★ Sección de anuncios (MODIFICADO)
│   │   ├── AdProgressBar.tsx
│   │   └── ...
│   ├── hooks/
│   │   └── useSimulation.ts # Hook de simulación
│   └── prisma/
│       ├── schema.prisma # ★ Esquema de DB (MODIFICADO)
│       └── migrations/
│           └── 20251112_add_ad_claims/ # ★ Nueva migración
└── ...
```

## Probar el Sistema de Anuncios

### Paso 1: Acceder a la aplicación

1. Abrir navegador en `http://localhost:3000`
2. Iniciar sesión o registrarse (si es necesario)

### Paso 2: Navegar a la sección de Publicidad

1. Hacer clic en la pestaña “Publicidad” en el menú
2. Ver la lista de anuncios disponibles

### Paso 3: Ver un anuncio

1. Hacer clic en el botón “Ver Anuncio (2 pts)” de cualquier anuncio
2. Se abrirá una nueva pestaña con:
  - Barra de progreso en la parte superior
  - Contador de 10 segundos al lado derecho
  - Contenido del anuncio en iframe

### Paso 4: Observar el contador

1. El contador inicia automáticamente al cargar la página
2. La barra de progreso se llena gradualmente

3. El botón muestra: “10s”, “9s”, “8s”... hasta “0s”
4. Cuando termina, el botón cambia a “Reclamar Puntos”

## Paso 5: Reclamar puntos

1. Hacer clic en el botón “Reclamar Puntos”
2. Se suman 2 puntos a la cuenta
3. Aparece mensaje de confirmación
4. El botón cambia a “✓ Reclamado”

## Paso 6: Verificar sistema diario

1. Intentar ver el mismo anuncio nuevamente
2. El botón debe mostrar “Reclamado Hoy”
3. El mensaje debe indicar cuándo estará disponible (medianoche)

## Paso 7: Probar pausa del contador

1. Ver un anuncio nuevo
2. Durante el contador, cambiar a otra pestaña
3. Regresar a la pestaña del anuncio
4. Verificar que el contador continúa desde donde quedó

## Solución de Problemas Comunes

---

### Error: “Cannot find module ‘prisma/client’”

#### Solución:

```
npx prisma generate
```

### Error: Base de datos no existe

#### Solución:

```
npx prisma migrate dev
```

### Error: Puerto 3000 en uso

#### Solución:

```
# Cambiar puerto en package.json o usar:  
PORT=3001 npm run dev
```

## Iframe no carga contenido

**Causa:** Algunos sitios web bloquean ser cargados en iframes (política CORS)

**Solución:** Es normal, el contador seguirá funcionando

## Verificación de Cambios

---

### 1. Verificar barra de progreso

- [ ] La barra está en la parte superior

- ☐ Se llena gradualmente
- ☐ Tiene colores distintos según el estado

## 2. Verificar botón lateral

- ☐ Está al lado derecho de la barra
- ☐ Muestra el tiempo restante (10s, 9s, ...)
- ☐ Cambia a “Reclamar Puntos” cuando termina
- ☐ Se puede hacer clic cuando está listo

## 3. Verificar sistema de puntos

- ☐ Se suman 2 puntos al reclamar
- ☐ Los puntos se actualizan en la interfaz
- ☐ Aparece mensaje de confirmación

## 4. Verificar inicio automático

- ☐ El contador inicia solo al cargar la página
- ☐ No requiere interacción del usuario
- ☐ Funciona incluso si el iframe tiene errores

## 5. Verificar sistema diario

- ☐ Solo se puede reclamar una vez por anuncio por día
- ☐ Muestra tiempo hasta medianoche
- ☐ Se reinicia automáticamente a las 00:00

## Datos de Prueba

---

### Anuncios de Ejemplo

El sistema incluye 15 anuncios de muestra pre-cargados:

- Curso de Marketing Digital
- Plataforma de Trading
- Tienda Online
- Consultoría Empresarial
- Y más...

### Límites por Plan

- **Registrado:** 5 anuncios/día
- **Invitado:** 10 anuncios/día
- **Básico:** 10 anuncios/día
- **VIP:** 10 anuncios/día
- **Premium:** 15 anuncios/día
- **Elite:** 20 anuncios/día

## Base de Datos

---

### Ver datos en la base de datos

```
# Abrir Prisma Studio (interfaz visual)
npx prisma studio
```

### Resetear base de datos

```
# ⚠ CUIDADO: Esto borrará todos los datos
npx prisma migrate reset
```

### Ver reclamos de anuncios

```
-- En Prisma Studio o SQL directamente
SELECT * FROM AdClaim;
```

## Desarrollo

---

### Editar archivos clave

1. **Página del anuncio:** `app/ad-view/[adId]/page.tsx`
2. **Sección de anuncios:** `components/PublicidadSection.tsx`
3. **Hook de simulación:** `hooks/useSimulation.ts`
4. **Esquema de DB:** `prisma/schema.prisma`

### Hot Reload

Los cambios en archivos TypeScript/React se reflejan automáticamente sin reiniciar el servidor.

### Logs y Debugging

- Abrir DevTools del navegador (F12)
- Ver consola para logs del sistema
- Ver Network para llamadas API
- Ver Application > LocalStorage para ver datos almacenados

## Despliegue

---

### Vercel (Recomendado)

```
# Instalar CLI de Vercel
npm i -g vercel

# Desplegar
vercel
```

### Otras plataformas

- **Netlify:** Compatible con Next.js
- **Railway:** Compatible con Next.js + PostgreSQL

- **Heroku:** Compatible con Next.js

## Soporte

---

Para problemas o preguntas:

1. Revisar la documentación en `CAMBIOS_SISTEMA_ANUNCIOS.md`
2. Verificar los logs en la consola del navegador
3. Revisar los archivos modificados
4. Verificar que las migraciones se aplicaron correctamente

## Notas Importantes

---

⚠ **LocalStorage:** El sistema usa localStorage para el tracking diario. Los datos persisten en el navegador del usuario.

⚠ **Iframe Sandbox:** Por seguridad, los iframes tienen permisos limitados. Algunos sitios pueden no cargar correctamente.

⚠ **Reinicio a Medianoche:** El reinicio se calcula basado en la hora local del navegador del usuario.

🌐 **Localhost:** Este localhost (127.0.0.1:3000) se refiere al localhost de la máquina donde se ejecuta el servidor, no tu máquina local. Para acceder desde tu máquina, necesitarás desplegar la aplicación o configurar port forwarding.

## Resultado Esperado

---

✅ Sistema completamente funcional con:

- Barra de progreso horizontal
- Botón lateral con tiempo/reclamar
- 2 puntos por anuncio
- Inicio automático del contador
- Control diario con reinicio a medianoche
- Límites por plan de usuario
- Pausado automático al cambiar de pestaña