

# Resumen de Cambios - Sistema de Sorteos y Loterías

## Fecha de Implementación

16 de Noviembre de 2025

## Descripción General

Se implementó un sistema completo de sorteos y loterías para la plataforma YigiCoin, eliminando la sección de libros digitales y agregando tres tipos de sorteos/loterías con restricciones por rangos de usuarios.



## Cambios Principales

### 1. Tienda de Totems ✅

- **MODIFICADO:** Costo de totems reducido de 1,000 puntos a **100 puntos**
- **Funcionalidad mantenida:** Los totems siguen restaurando el temporizador de usuarios
- Al comprar, se suma un totem más a la cuenta del usuario
- Límite máximo: 5 totems por usuario

### 2. Libros Digitales ❌

- **ELIMINADO:** Sección completa de libros digitales removida
- La navegación ya no incluye esta opción
- Código relacionado eliminado de BeneficiosSection

### 3. Sistema de SORTEOS (Pago con Puntos) 🎁

- **Disponible desde:** Rango Invitado en adelante (Invitado, Miembro, VIP, Premium, Élite)
- **Sorteo Semanal:**
  - Premio: 20 USD
  - Costo por boleto: 100 puntos
  - Icono: 🎁
- **Sorteo Mensual:**
  - Premio: 100 USD
  - Costo por boleto: 300 puntos
  - Icono: 🎫
- **Características:**
  - Se realizan los viernes a las 00:00
  - Selección aleatoria del ganador
  - Premio se acumula si no hay participantes

### 4. Sistema de LOTERÍA NORMAL (Pago con Metamask) 🎲

- **Disponible desde:** Rango Miembro (basico) en adelante

- **Lotería Semanal:**

- Premio: 100 USD
- Costo por boleto: 2 USD
- Icono: 🎲

- **Lotería Mensual:**

- Premio: 5,000 USD
- Costo por boleto: 80 USD
- Icono: 💰

- **Características:**

- Pago con Metamask (proceso manual simulado)
- Se realizan los viernes a las 00:00
- Selección aleatoria del ganador
- Premio se acumula si no hay participantes

## 5. Sistema de LOTERÍA VIP (Pago con Metamask)

- **Disponible desde:** Rango VIP en adelante

- **Lotería VIP Semanal:**

- Premio: 6,000 USD
- Costo por boleto: 100 USD
- Icono: 🎉

- **Lotería VIP Mensual:**

- Premio: 10,000 USD
- Costo por boleto: 150 USD
- Icono: 🏆

- **Características:**

- Pago con Metamask (proceso manual simulado)
- Se realizan los viernes a las 00:00
- Selección aleatoria del ganador
- Premio se acumula si no hay participantes

## 6. Visualización de Secciones

Nueva estructura de navegación en Beneficios:

- **Tienda:** Totems y paquetes de anuncios
- **Sorteos:** Sorteos semanales y mensuales (pago con puntos)
- **Loterías:** Loterías normales y VIP (pago con Metamask)
- **Historial:** Boletos comprados y resultados de sorteos anteriores

## 7. Historial de Boletos y Resultados

- Los usuarios pueden ver todos sus boletos comprados
- Estado de cada boleto (Pagado, Ganador, No ganó)
- Historial de resultados de sorteos anteriores
- Información del ganador (si aplica)
- Premios acumulados si no hubo ganador

---



## Archivos Modificados

---

### Esquema de Base de Datos

- `prisma/schema.prisma`
- Agregados enums: `DrawType`, `DrawStatus`, `TicketPaymentType`, `TicketStatus`
- Agregado modelo `Draw` (sorteos/loterías)
- Agregado modelo `Ticket` (boletos)
- Agregado modelo `DrawResult` (resultados)

### Configuración

- `lib/economyConfig.ts`
- Costo de totems cambiado a 100 puntos
- Agregadas configuraciones `RAFFLE_CONFIG` (sorteos con puntos)
- Agregadas configuraciones `LOTTERY_NORMAL_CONFIG` (loterías normales)
- Agregadas configuraciones `LOTTERY_VIP_CONFIG` (loterías VIP)
- Agregadas funciones auxiliares para verificar acceso por rango
- Agregada función `getNextDrawDate()` para calcular fechas de sorteos

### Acciones del Servidor

- `app/actions/draws.ts` (NUEVO)
- `buyRaffleWeeklyTicket()` - Compra boleto sorteo semanal
- `buyRaffleMonthlyTicket()` - Compra boleto sorteo mensual
- `buyLotteryWeeklyTicket()` - Compra boleto lotería semanal
- `buyLotteryMonthlyTicket()` - Compra boleto lotería mensual
- `buyLotteryVIPWeeklyTicket()` - Compra boleto lotería VIP semanal
- `buyLotteryVIPMonthlyTicket()` - Compra boleto lotería VIP mensual
- `getUserTickets()` - Obtiene boletos del usuario
- `getDrawHistory()` - Obtiene historial de sorteos
- `getActiveDraw()` - Obtiene sorteo activo actual

### Componentes de UI

- `components/RaffleSection.tsx` (NUEVO)
- Componente para mostrar sorteos con pago en puntos
- Cards para sorteos semanal y mensual
- Verificación de rango de usuario
- Compra de boletos con puntos
- `components/LotteriesSection.tsx` (NUEVO)
- Componente para mostrar loterías con pago en Metamask
- Cards para loterías normales y VIP
- Modal de pago con Metamask
- Verificación de rango de usuario
- `components/DrawHistorySection.tsx` (NUEVO)
- Componente para mostrar historial

- Tabs: “Mis Boletos” y “Resultados”
- Lista de boletos con estado y detalles
- Historial de sorteos completados
- `components/BeneficiosSection.tsx` (MODIFICADO)
  - Eliminada sección de libros digitales
  - Eliminada sección de loterías antiguas
  - Actualizada navegación con nuevas secciones
  - Integración de nuevos componentes

## Script de Ejecución de Sorteos

- `lib/executeDraws.ts` (NUEVO)
- `executeScheduledDraws()` - Ejecuta todos los sorteos pendientes
- `executeDraw()` - Ejecuta un sorteo específico
- Selección aleatoria de ganadores
- Acumulación de premios
- Creación automática de próximos sorteos
- `scripts/execute-draws.ts` (NUEVO)
  - Script CLI para ejecutar sorteos manualmente
  - Uso: `npx ts-node scripts/execute-draws.ts`

## Migración de Base de Datos

- `prisma/migrations/20251116_add_draws_system/migration.sql` (NUEVO)
- Creación de enums para el sistema
- Creación de tabla `Draw`
- Creación de tabla `Ticket`
- Creación de tabla `DrawResult`
- Índices para optimizar consultas

## Instrucciones de Instalación

### 1. Aplicar Migración de Base de Datos

```
cd /home/ubuntu/code_artifacts/referral_project
npx prisma migrate deploy
```

O si estás en desarrollo:

```
npx prisma migrate dev
```

### 2. Generar Cliente de Prisma

```
npx prisma generate
```

### 3. Instalar Dependencias (si es necesario)

```
npm install
```

### 4. Ejecutar el Proyecto

```
npm run dev
```



## Configuración de Sorteos Programados

Los sorteos deben ejecutarse automáticamente cada viernes a las 00:00. Hay dos opciones para configurar esto:

### Opción 1: Cron Job (Linux/Mac)

Editar crontab:

```
crontab -e
```

Agregar línea (ejecutar todos los viernes a las 00:00):

```
0 0 * * 5 cd /ruta/al/proyecto && npx ts-node scripts/execute-draws.ts >> /var/log/draws.log 2>&1
```

### Opción 2: Tarea Programada de Windows

1. Abrir “Programador de tareas”
2. Crear tarea básica
3. Configurar trigger: Semanal, viernes, 00:00
4. Acción: Iniciar programa
5. Programa: cmd.exe
6. Argumentos: /c cd /d "C:\ruta\al\proyecto" && npx ts-node scripts/execute-draws.ts

### Opción 3: Script Node.js con node-cron

Crear archivo server/cron-job.js :

```
const cron = require('node-cron');
const { executeScheduledDraws } = require('../lib/executeDraws');

// Ejecutar cada viernes a las 00:00
cron.schedule('0 0 * * 5', async () => {
  console.log('Ejecutando sorteos programados...');
  await executeScheduledDraws();
});
```

Luego ejecutar en segundo plano:

```
node server/cron-job.js &
```

## Opción 4: Ejecución Manual

Para pruebas o ejecución manual:

```
npx ts-node scripts/execute-draws.ts
```

O agregar script a package.json:

```
{
  "scripts": {
    "draws:execute": "ts-node scripts/execute-draws.ts"
  }
}
```

Y ejecutar:

```
npm run draws:execute
```

## Verificación de Cambios

### 1. Verificar Migración de Base de Datos

```
npx prisma studio
```

Verificar que existen las tablas:

- Draw
- Ticket
- DrawResult

### 2. Probar Compra de Totems

1. Iniciar sesión en la aplicación
2. Navegar a Beneficios > Tienda
3. Verificar que el costo de totems es 100 puntos
4. Comprar un totem
5. Verificar que se descuentan 100 puntos y se suma 1 totem

### 3. Probar Sorteos (Puntos)

1. Asegurarse de tener al menos rango Invitado
2. Navegar a Beneficios > Sorteos
3. Verificar que se muestran sorteos semanal y mensual
4. Comprar un boleto con puntos
5. Verificar en Historial que aparece el boleto

### 4. Probar Loterías (Metamask)

1. Asegurarse de tener al menos rango Miembro para loterías normales
2. Asegurarse de tener rango VIP para loterías VIP

3. Navegar a Beneficios > Loterías
4. Intentar comprar un boleto
5. Se mostrará modal de Metamask (simulado)
6. Confirmar compra
7. Verificar en Historial que aparece el boleto

## 5. Probar Ejecución de Sorteos

```
# Ejecutar manualmente
npx ts-node scripts/execute-draws.ts

# Verificar logs
tail -f /var/log/draws.log # Si configuraste cron job
```



## Restricciones por Rangos

Sección	Rango Mínimo	Contenido Disponible
Tienda	Invitado	Totems (100 pts), Paquetes de anuncios
Sorteos	Invitado	Sorteos semanal y mensual (puntos)
Loterías	Miembro	Loterías normales semanales y mensuales
Loterías VIP	VIP	Loterías VIP semanales y mensuales



## Notas Importantes

1. **Totems:** El costo cambió de 1,000 a 100 puntos según requerimientos.
2. **Premios Acumulados:** Si un sorteo no tiene participantes, el premio se acumula para el siguiente sorteo del mismo tipo.
3. **Metamask Simulado:** La integración con Metamask está simulada para desarrollo. En producción, se debe integrar con la biblioteca Web3.js o Ethers.js real.
4. **Sorteos Programados:** Se recomienda usar un cron job o tarea programada para ejecutar los sorteos automáticamente cada viernes.
5. **Números de Boleto:** Se generan automáticamente números únicos de 8 dígitos para cada boleto.
6. **Notificaciones:** Los ganadores reciben notificaciones automáticamente en el sistema.

## Seguridad

- Todas las transacciones de puntos se realizan en el servidor
- Validación de rango de usuario antes de permitir compras
- Validación de puntos suficientes antes de procesar compra
- Números de boleto únicos por sorteo
- Transacciones atómicas para garantizar consistencia

## Compatibilidad

-  Diseño responsive para móviles y desktop
-  Soporte para tema claro y oscuro
-  Compatible con todos los navegadores modernos
-  Optimizado para carga rápida

## Solución de Problemas

### Error: “Tabla Draw no encontrada”

**Solución:** Ejecutar la migración de Prisma

```
npx prisma migrate deploy
npx prisma generate
```

### Error: “Cannot find module ‘@/lib/economyConfig’”

**Solución:** Verificar que el archivo existe y recompilar

```
npm run build
```

### Los sorteos no se ejecutan automáticamente

**Solución:** Verificar configuración de cron job o tarea programada

```
crontab -l # Ver cron jobs actuales
```

### Error en compra con Metamask

**Solución:** La integración está simulada. Para producción, implementar integración real con Web3.

## Soporte

Para cualquier problema o pregunta sobre la implementación, consultar:

- Este documento

- Código fuente en `/home/ubuntu/code_artifacts/referral_project`
  - Logs del sistema
- 

## Checklist de Implementación

---

- [x] Actualizar esquema de Prisma
  - [x] Crear migración de base de datos
  - [x] Actualizar economyConfig
  - [x] Implementar acciones del servidor
  - [x] Crear componentes de UI
  - [x] Eliminar sección de libros digitales
  - [x] Actualizar BeneficiosSection
  - [x] Implementar historial de boletos
  - [x] Crear script de ejecución de sorteos
  - [x] Documentar cambios
  - [ ] Aplicar migración a base de datos de producción
  - [ ] Configurar cron job para sorteos automáticos
  - [ ] Probar en entorno de staging
  - [ ] Desplegar a producción
- 

**Versión:** 1.0

**Fecha:** 16 de Noviembre de 2025

**Autor:** YigiCoin Development Team