

Características de Seguridad - Snackfly Rasca y Gana

Resumen de Mejoras de Seguridad

Esta versión mejorada del juego "Rasca y Gana" incluye múltiples capas de seguridad para prevenir trampas y fraudes, manteniendo la funcionalidad solo en el frontend.

Medidas de Seguridad Implementadas

1. Verificación de Fecha y Hora

Timestamp de Generación

- **Cada premio incluye fecha y hora exacta de generación**
- **Formato:** DD/MM/AAAA HH:MM:SS con timezone
- **Visible en:** Modal de premio y datos de verificación
- **Propósito:** Prevenir capturas de pantalla reutilizadas

Reloj en Tiempo Real

- **Muestra fecha y hora actual constantemente**
- **Actualización:** Cada segundo
- **Detección:** Cambios sospechosos de tiempo del sistema
- **Formato:** Día completo, fecha y hora legible

Validación de Fechas

- **Verificación:** Los premios no pueden ser del futuro
- **Tolerancia:** 1 minuto para diferencias de sincronización

- **Expiración:** Datos más antiguos de 24 horas se invalidan
- **Limpieza:** Datos corruptos se eliminan automáticamente

2. Sistema de Sesiones Únicas

✓ ID de Sesión Único

- **Generación:** Timestamp + número aleatorio + codificación Base64
- **Formato:** 16 caracteres alfanuméricos
- **Propósito:** Identificar cada jugada de forma única
- **Visible en:** Modal de premio y logs del sistema

✓ Información de Sesión

- **User Agent:** Primeros 50 caracteres (por privacidad)
- **Resolución de pantalla:** Ancho x Alto
- **Tiempo de inicio:** Timestamp exacto de inicio
- **Tiempo de finalización:** Timestamp de completado

✓ Detección de Interrupciones

- **Cambio de pestaña:** Durante el raspado
- **Cierre de ventana:** Antes de completar
- **Manipulación:** Intentos de modificar el juego

3. Validación de Integridad de Datos

✓ Hash de Verificación

- **Algoritmo:** Hash personalizado con clave secreta
- **Aplicación:** Todos los datos guardados en localStorage

- **Verificación:** Al cargar datos existentes
- **Acción:** Limpieza automática si los datos están corruptos

✓ Estructura de Datos Segura

JavaScript

```
{
  payload: {
    // Datos del juego
  },
  hash: "abc123..." // Hash de verificación
}
```

✓ Detección de Manipulación

- **localStorage modificado:** Hash no coincide
- **Datos corruptos:** Estructura inválida
- **Fechas inválidas:** Timestamps manipulados
- **Respuesta:** Limpieza y reinicio automático

4. Prevención de Trampas Comunes

✓ Contra Capturas de Pantalla

- **Fecha y hora visible:** En cada premio
- **ID único de sesión:** No reutilizable
- **Información de verificación:** Difícil de falsificar
- **Marca de agua:** ID de sesión en el canvas

✓ Contra Manipulación de Código

- **Ofuscación de claves:** Base64 y nombres complejos
- **Validación cruzada:** Múltiples verificaciones

- **Detección de cambios:** Monitoreo de integridad
- **Logs de seguridad:** Registro de actividades sospechosas

✓ Contra Manipulación de Tiempo

- **Detección de saltos:** Cambios bruscos de tiempo
- **Validación de fechas:** Coherencia temporal
- **Logs de advertencia:** Actividades sospechosas
- **Limpieza automática:** Datos inválidos

5. Interfaz de Usuario Mejorada

✓ Indicadores de Seguridad

- **Badge "Versión Segura":** En el header
- **Reloj en tiempo real:** Siempre visible
- **"Premio Verificado":** En el modal
- **Información de verificación:** Detalles técnicos

✓ Información Detallada

- **Fecha de generación:** Visible en el premio
- **ID de sesión:** Para verificación
- **Última jugada:** Información histórica
- **Total de jugadas:** Contador acumulativo

✓ Mensajes de Seguridad

- **Notas explicativas:** Sobre la verificación
- **Advertencias:** Para actividades sospechosas
- **Confirmaciones:** De integridad de datos

Cómo Funciona la Verificación

Proceso de Generación de Premio

1. Usuario inicia el juego

- Se crea una sesión única con ID
- Se registra fecha/hora de inicio
- Se captura información del navegador

2. Usuario raspa la tarjeta

- Se monitorea el progreso del raspado
- Se detectan cambios de pestaña sospechosos
- Se registra el tiempo de raspado

3. Premio se revela (70% raspado)

- Se genera el premio con timestamp exacto
- Se asigna ID único al premio
- Se incluye información de verificación

4. Datos se guardan de forma segura

- Se calcula hash de integridad
- Se almacena en localStorage cifrado
- Se registra información de sesión

Proceso de Verificación

1. Al cargar el juego

- Se verifica integridad de datos existentes

- Se valida coherencia de fechas
- Se detectan manipulaciones

2. Si ya jugó hoy

- Se muestra información de la última jugada
- Se incluyen detalles de verificación
- Se permite demo pero no premio real

3. Si hay datos corruptos

- Se limpian automáticamente
- Se registra la actividad sospechosa
- Se permite jugar como usuario nuevo



Detección de Actividades Sospechosas

Indicadores de Trampa

- **Manipulación de localStorage:** Hash inválido
- **Cambio de fecha del sistema:** Saltos temporales
- **Modificación de código:** Comportamiento anómalo
- **Capturas reutilizadas:** IDs duplicados
- **Interrupciones sospechosas:** Cambios de pestaña

Respuestas Automáticas

- **Limpieza de datos:** Eliminación automática
- **Logs de advertencia:** Registro en consola
- **Reinicio del juego:** Estado limpio

- **Bloqueo temporal:** (Opcional, configurable)



Información de Verificación Mostrada

En el Modal de Premio

Plain Text

🎉 ¡FELICIDADES! 🎉

🔒 Premio Verificado

[ICONO] [TEXTO DEL PREMIO]

[DESCRIPCIÓN DEL PREMIO]

📋 Información de Verificación

📅 Generado: 23/07/2025 a las 14:30:25

🔒 ID Sesión: A1B2C3D4E5F6G7H8

🕒 Timezone: America/Bogota

🆔 ID Premio: X9Y8Z7W6V5U4T3S2

🔒 Este premio incluye verificación de fecha, hora y sesión para prevenir fraudes

En la Información de Última Jugada

Plain Text

🕒 ¡Ya jugaste hoy!

Última jugada: 23/07/2025 a las 14:30:25

Último premio: discount

Total de jugadas: 5

🔄 Probar Demo



Configuración de Seguridad

Variables Configurables

JavaScript






```
const CONFIG = {  
  REVEAL_THRESHOLD: 70,           // % para revelar automáticamente  
  BRUSH_SIZE: 30,                 // Tamaño del pincel  
  STORAGE_KEY: "...",            // Clave ofuscada  
  INTEGRITY_KEY: "SF2025_SECURE", // Clave de integridad  
  ONE_DAY_MS: 24 * 60 * 60 * 1000 // Milisegundos en un día  
};
```

Personalización de Seguridad




- **Cambiar clave de integridad:** Modificar `INTEGRITY_KEY`
- **Ajustar tolerancia de tiempo:** Modificar validaciones
- **Personalizar logs:** Agregar más detalles
- **Configurar respuestas:** A actividades sospechosas

Limitaciones y Consideraciones

Lo que SÍ Previene

-  Capturas de pantalla reutilizadas
-  Manipulación básica de localStorage
-  Jugar múltiples veces el mismo día
-  Modificación simple de fechas
-  Uso de datos corruptos

Lo que NO Puede Prevenir Completamente

-  Usuarios muy técnicos con conocimiento avanzado
-  Modificación del código JavaScript completo
-  Uso de herramientas de desarrollo avanzadas

-  Manipulación del reloj del sistema operativo

Recomendaciones Adicionales

- **Para máxima seguridad:** Implementar backend
- **Para producción:** Ofuscar más el código
- **Para analytics:** Agregar tracking de eventos
- **Para soporte:** Logs más detallados



Mantenimiento y Actualizaciones

Monitoreo Recomendado

- **Logs de consola:** Revisar actividades sospechosas
- **Patrones de uso:** Detectar comportamientos anómalos
- **Feedback de usuarios:** Reportes de problemas

Actualizaciones de Seguridad

- **Cambiar claves:** Periódicamente
- **Actualizar algoritmos:** Según necesidades
- **Mejorar detección:** Basado en nuevas amenazas
- **Optimizar rendimiento:** Sin comprometer seguridad

Nota: Esta implementación proporciona un nivel de seguridad significativamente mayor que la versión original, pero para aplicaciones críticas se recomienda implementar un backend con validación del servidor.