🔐 Medidas de Seguridad Implementadas

Esta aplicación de finanzas personales maneja datos sensibles y se han implementado múltiples capas de seguridad para proteger la información de los usuarios.

1. Autenticación y Autorización

✅ Contraseñas Seguras

- **Hashing con bcrypt**: Todas las contraseñas se hashean con bcrypt (cost factor 12)
- **Validación de contraseñas**: Mínimo 8 caracteres, al menos una mayúscula, una minúscula y un número
- **No se almacenan contraseñas en texto plano**: Solo se almacena el hash

✅ Gestión de Sesiones

- **NextAuth.js**: Implementación robusta de autenticación
- **Tokens seguros**: JWTs firmados y tokens de sesión en base de datos
- **Expiración de sesiones**: Las sesiones expiran automáticamente
- **Logout seguro**: Eliminación completa de sesiones

✅ Verificación de Autorización

- **Todas las APIs verifican autenticación**: Ninguna operación sin usuario autenticado
- **Verificación de propiedad**: Los usuarios solo pueden acceder a sus propios datos
- **Middleware de autorización**: Capa adicional de protección en cada endpoint

2. Validación y Sanitización de Datos

✅ Validación con Zod

- **Esquemas de validación**: Todos los datos de entrada se validan con Zod
- **Prevención de inyección SQL**: Prisma ORM + validación estricta
- **Prevención de XSS**: Sanitización de entradas HTML/Script
- **Tipos de datos estrictos**: TypeScript + Zod garantizan tipos correctos

✅ Límites de Datos

- **Longitud máxima de strings**: Previene desbordamiento de buffer
- **Rangos numéricos**: Montos limitados a valores realistas
- **Listas limitadas**: Arrays con límites para prevenir ataques de memoria
- **Caracteres permitidos**: Regex para validar formatos (emails, nombres, etc.)

3. Rate Limiting (Limitación de Tasa)

✅ Protección contra Fuerza Bruta

- **Login/Signup**: Máximo 5 intentos cada 15 minutos
- **APIs generales**: 60 requests por minuto
- **Operaciones sensibles**: 10 requests por minuto

- **Headers informativos**: X-RateLimit-Remaining, X-RateLimit-Reset

✅ Identificación por IP

- **Tracking por IP**: Previene múltiples intentos desde la misma fuente
- **Soporte X-Forwarded-For**: Compatible con proxies y load balancers

4. Headers de Seguridad HTTP

✅ Configurados en Next.js

```
X-Frame-Options: DENY                          // Previene clickjacking
X-Content-Type-Options: nosniff                // Previene MIME sniffing
X-XSS-Protection: 1; mode=block               // Protección XSS del navegador
Referrer-Policy: strict-origin-when-cross-origin  // Control de información de refer-
encia
Permissions-Policy: camera=(), microphone=()   // Restringe permisos de API
```

5. Auditoría y Logging

✅ Registro de Actividad

- **Todas las acciones importantes**: Signup, login, eliminación de cuenta
- **Metadata de requests**: IP, User-Agent, timestamp
- **Trazabilidad completa**: Historial de operaciones por usuario
- **Base de datos persistente**: Logs almacenados en PostgreSQL

✅ Tabla AuditLog

- Usuario que realizó la acción
- Tipo de acción (create, update, delete, login, etc.)
- Entidad afectada (transaction, budget, account)
- IP y User-Agent
- Fecha y hora exacta

6. Protección de Datos en Base de Datos

✅ PostgreSQL con Prisma

- **Prepared Statements**: Previene inyección SQL automáticamente
- **Relaciones y cascadas**: Integridad referencial garantizada
- **Índices optimizados**: Para queries rápidas y seguras
- **Eliminación en cascada**: Cuando se elimina usuario, se eliminan todos sus datos

✅ Configuración de Usuario (UserSettings)

- **Almacenamiento de preferencias**: Idioma y moneda
- **Datos no sensibles**: Separados de información crítica
- **Sincronización segura**: API dedicada con autenticación

7. Gestión Segura de Eliminación de Cuenta

✅ Proceso GDPR-Compliant

- **Confirmación explícita**: AlertDialog con advertencia clara
- **Eliminación completa**:
- Todas las transacciones
- Todos los presupuestos
- Todas las cuentas bancarias
- Todas las categorías personalizadas
- Usuario y sesiones
- **Logout automático**: Cierre de sesión inmediato post-eliminación
- **Sin recuperación**: Datos eliminados permanentemente

8. Protección de Variables de Entorno

✅ Secrets Management

- **Archivo .env**: Nunca committed a git
- **Variables de NextAuth**:
- `NEXTAUTH_SECRET`: Clave secreta para firmar tokens
- `NEXTAUTH_URL`: URL de callback para OAuth
- **DATABASE_URL**: Conexión encriptada a PostgreSQL
- **Acceso restringido**: Solo el servidor tiene acceso

9. Seguridad en el Cliente

✅ React y Next.js Best Practices

- **CSR y SSR apropiados**: Datos sensibles solo en server-side
- **useSession con validación**: Verificación de sesión en cada uso
- **Sanitización de output**: React escapa automáticamente HTML
- **No datos sensibles en localStorage**: Solo preferencias de UI

✅ HTTPS Recomendado

- **TLS/SSL en producción**: Encriptación end-to-end
- **Certificados válidos**: Let’s Encrypt o similar
- **HSTS**: Strict-Transport-Security header

10. Internacionalización (i18n) Segura

✅ Sin Vulnerabilidades de i18n

- **Traducciones estáticas**: No inyección de código
- **Formato de números**: Locale-aware pero seguro
- **Validación independiente del idioma**: Backend valida en inglés

11. Recomendaciones Adicionales para Producción

⚠️ Implementar en Producción

1. **WAF (Web Application Firewall)**: Cloudflare, AWS WAF
2. **Secrets Management**: HashiCorp Vault, AWS Secrets Manager
3. **Backup automatizado**: De base de datos cada día
4. **Monitoring**: Sentry, DataDog para detectar anomalías
5. **Penetration Testing**: Auditorías de seguridad regulares
6. **2FA (Autenticación de dos factores)**: Para cuentas sensibles
7. **Encriptación en reposo**: Para datos muy sensibles en DB
8. **DDoS Protection**: Cloudflare, AWS Shield
9. **Logging centralizado**: ELK Stack, CloudWatch
10. **Alertas de seguridad**: Notificaciones de actividad sospechosa

12. Compliance y Regulaciones

✅ GDPR (General Data Protection Regulation)

- **Right to deletion**: Implementado con /api/user/delete
- **Data minimization**: Solo se recopilan datos necesarios
- **Consent**: Usuario acepta términos en signup
- **Transparency**: Documentación clara de uso de datos

✅ Best Practices

- **OWASP Top 10**: Mitigadas las vulnerabilidades principales
- **PCI DSS**: No se almacenan números de tarjeta completos
- **SOC 2**: Controles de seguridad implementados

📝 Resumen

Esta aplicación implementa una estrategia de **defensa en profundidad** (Defense in Depth) con múltiples capas de seguridad:

1. ✅ Autenticación y autorización robustas
2. ✅ Validación estricta de todos los datos de entrada
3. ✅ Rate limiting para prevenir ataques
4. ✅ Headers de seguridad HTTP
5. ✅ Auditoría completa de acciones
6. ✅ Protección de datos en base de datos
7. ✅ Eliminación segura de datos (GDPR)
8. ✅ Secrets management apropiado
9. ✅ Mejores prácticas de React/Next.js
10. ✅ Internacionalización segura

Ningún sistema es 100% seguro, pero estas medidas reducen significativamente los riesgos de seguridad comunes.