Table of Contents

[API de Auditoría para Originación de Tarjetas de Crédito - Bradescard México 1](#_Toc1)

[Tabla de Contenidos 1](#_Toc2)

[Resumen Ejecutivo 1](#_Toc3)

[Beneficios Clave 1](#_Toc4)

[Socios Tecnológicos Clave 1](#_Toc5)

[1. CONTEXTO DEL NEGOCIO Y TAXONOMÍA DE EVENTOS 2](#_Toc6)

[Contexto del Negocio 2](#_Toc7)

[Partners Tecnológicos Identificados 2](#_Toc8)

[Objetivo de la Bitácora 2](#_Toc9)

[Proceso Estándar de Originación de Tarjetas de Crédito 3](#_Toc10)

[Catálogo Completo de Eventos Estándar (51 Eventos) 5](#_Toc11)

[1. Eventos de Inicio y Captura (Partner) 5](#_Toc12)

[2. Eventos de Digital Onboarding Biométrico (Partner Biométrico) 5](#_Toc13)

[3. Eventos de Consultas Externas (Bradescard) 6](#_Toc14)

[4. Eventos de Business Rules Engine y Scoring 6](#_Toc15)

[5. Eventos de Decisión (Partner) 7](#_Toc16)

[6. Eventos de Core Bancario (Bradescard) 7](#_Toc17)

[7. Eventos de Producción y Entrega 8](#_Toc18)

[8. Eventos de Error y Abandono 8](#_Toc19)

[2. ARQUITECTURA DE SOLUCIÓN AZURE 9](#_Toc20)

[Arquitectura Simplificada para Auditoría Financiera 9](#_Toc21)

[Capa de Ingreso 10](#_Toc22)

[Capa de API 10](#_Toc23)

[Capa de Procesamiento 10](#_Toc24)

[Capa de Persistencia 10](#_Toc25)

[Capa de Monitoreo Básico 10](#_Toc26)

[3. MODELO DE DATOS Y COMPLIANCE 11](#_Toc27)

[Esquema de Base de Datos Especializado 11](#_Toc28)

[Tablas Principales en SQL Server On-Premise 12](#_Toc29)

[SP para Gestionar Configuraciones del Sistema 31](#_Toc30)

[SP para Consultar Timeline de Solicitud 32](#_Toc31)

[SP para Consultar Métricas por Socio 33](#_Toc32)

[SP para Búsqueda Avanzada de Eventos 34](#_Toc33)

[SP para Análisis de Logs y Rendimiento del Sistema 35](#_Toc34)

[Configuraciones de Compliance Financiero 39](#_Toc35)

[1. Regulaciones Mexicanas - CNBV 39](#_Toc36)

[2. Encriptación y Seguridad de Datos 39](#_Toc37)

[4. IMPLEMENTACIÓN TÉCNICA 40](#_Toc38)

[Stack Tecnológico Simplificado 40](#_Toc39)

[Backend (.NET 8) 40](#_Toc40)

[Seguridad Esencial 40](#_Toc41)

[API Endpoints Core 40](#_Toc42)

[1. Endpoint Principal - Registro de Eventos de Auditoría 40](#_Toc43)

[2. Endpoints de Consulta por Solicitud 41](#_Toc44)

[3. Endpoints de Consulta por Socio Comercial 41](#_Toc45)

[4. Endpoints de Búsqueda Avanzada 41](#_Toc46)

[5. Endpoints de Configuración y Monitoreo 41](#_Toc47)

[6. Endpoints de Reportes y Analytics 41](#_Toc48)

[5. Endpoints de Salud del Sistema 41](#_Toc49)

[6. Endpoints de Configuración 41](#_Toc50)

[5. MONITOREO OPERACIONAL 42](#_Toc51)

[Monitoreo de API únicamente 42](#_Toc52)

[Application Insights - Métricas Esenciales 42](#_Toc53)

[Alertas Automáticas 42](#_Toc54)

[Logs de Auditoría 42](#_Toc55)

[6. PRESUPUESTO Y ROI 42](#_Toc56)

[Estimación de Costos Mensual - Arquitectura Híbrida 42](#_Toc57)

[Total Estimado Azure: $2,061 - $3,353 USD/mes 42](#_Toc58)

[Infraestructura On-Premise (Bradescard): 42](#_Toc59)

[Ahorro vs. Arquitectura Compleja: 90% menos costo Azure 43](#_Toc60)

[Análisis de ROI - Arquitectura Híbrida 43](#_Toc61)

[Beneficios Cuantificables 43](#_Toc62)

[ROI Proyectado: 320% en el primer año 43](#_Toc63)

[Ventajas del Modelo Híbrido 43](#_Toc64)

[7. PLAN DE IMPLEMENTACIÓN 43](#_Toc65)

[Fases de Implementación Simplificada (10 semanas) 43](#_Toc66)

[Fase 1: Setup Básico (Semanas 1-3) 43](#_Toc67)

[Fase 2: Funcionalidad Core (Semanas 4-6) 43](#_Toc68)

[Fase 3: Integración y Monitoreo (Semanas 7-8) 43](#_Toc69)

[Fase 4: Producción (Semanas 9-10) 44](#_Toc70)

[Criterios de Éxito 44](#_Toc71)

[Técnicos 44](#_Toc72)

[De Negocio 44](#_Toc73)

[Conclusiones 44](#_Toc74)

[6. SCRIPTS DE INICIALIZACIÓN Y CONFIGURACIÓN 44](#_Toc75)

[Script de Creación de Base de Datos Completa 44](#_Toc76)

[Ejemplos de Uso Completos 47](#_Toc77)

[Ejemplo 1: Flujo Completo de Originación con Auditoría 47](#_Toc78)

[Ventajas del Modelo Híbrido: 51](#_Toc79)

[Funcionalidad Core Garantizada: 51](#_Toc80)

# API de Auditoría para Originación de Tarjetas de Crédito - Bradescard México

**Propuesta Técnica Completa**  
*Fecha: 26 de Diciembre, 2024*  
*Versión: 1.0*

## Tabla de Contenidos

1. [Contexto del Negocio y Taxonomía de Eventos](#Xab192b8b18f67caab27f87ec3dd0ad9ef1f62f0)
2. [Arquitectura de Solución Azure](#Xac9380e1bd5a4541168f355e455b8ec359fa871)
3. [Modelo de Datos y Compliance](#X1d56d7c1f422330ed18b434f254d841fe6cbbc3)
4. [Implementación Técnica](#X413c77421a6887d156c7675d9749f523e8aea6a)
5. [Monitoreo y Analytics](#Xed700c985d7b91eb725e9bda9e70a9e42912dca)
6. [Presupuesto y ROI](#X848b5eb465d089ac574865a4df731342b240f09)
7. [Plan de Implementación](#Xbda10ea956173b3bc4e4796f837ac61959d300c)

## Resumen Ejecutivo

Este documento presenta la arquitectura completa para una **API de Auditoría especializada en originación de tarjetas de crédito** para Bradescard México. La solución está diseñada para manejar **10,000+ aplicaciones mensuales** con múltiples partners externos, incorporando **validación biométrica avanzada**, **business rules engine** y **compliance total** con regulaciones financieras mexicanas.

### Beneficios Clave

* ✅ **Reducción del abandono** del 30% al 20% = +$1.2M USD/año
* ✅ **Prevención de multas CNBV** = $500K - $2M USD/año ahorrados
* ✅ **Detección de fraude** en tiempo real con biometría avanzada
* ✅ **Automatización compliance** = -40% esfuerzo manual = $300K USD/año
* ✅ **Integración universal** con partners existentes y futuros
* ✅ **ROI proyectado**: 380%+ en el primer año

### Socios Tecnológicos Clave

* **Partner Biométrico**: Validación biométrica y firma digital de terceros
* **Business Rules Engines**: FICO, SAS, Equifax Veraz, Trans Union México
* **Partners de Originación**: Retailers mexicanos y tiendas departamentales

# 1. CONTEXTO DEL NEGOCIO Y TAXONOMÍA DE EVENTOS

## Contexto del Negocio

**Bradescard México** - Empresa financiera especializada en tarjetas de crédito y departamentales que trabaja con **partners externos** para la originación de créditos. Los partners manejan sus propios procesos tecnológicos y tocan base con Bradescard en puntos críticos del flujo de originación.

### Partners Tecnológicos Identificados

#### **1. Partner de Validación Biométrica**

* **Servicios**: Digital Onboarding, Validación Biométrica, Firma Digital
* **Capacidades**:
  + **Photo ID OCR**: Extracción automática de datos de documentos oficiales
  + **Captura Facial**: Tecnología de reconocimiento facial
  + **3D Liveness**: Pruebas de vida avanzadas anti-spoofing
  + **Facematch**: Comparación facial documento vs. selfie
  + **OTP Services**: One Time Password Services
  + **Firma Digital**: Firma digital certificada con estándares internacionales
* **Integración**: APIs REST para cada componente biométrico

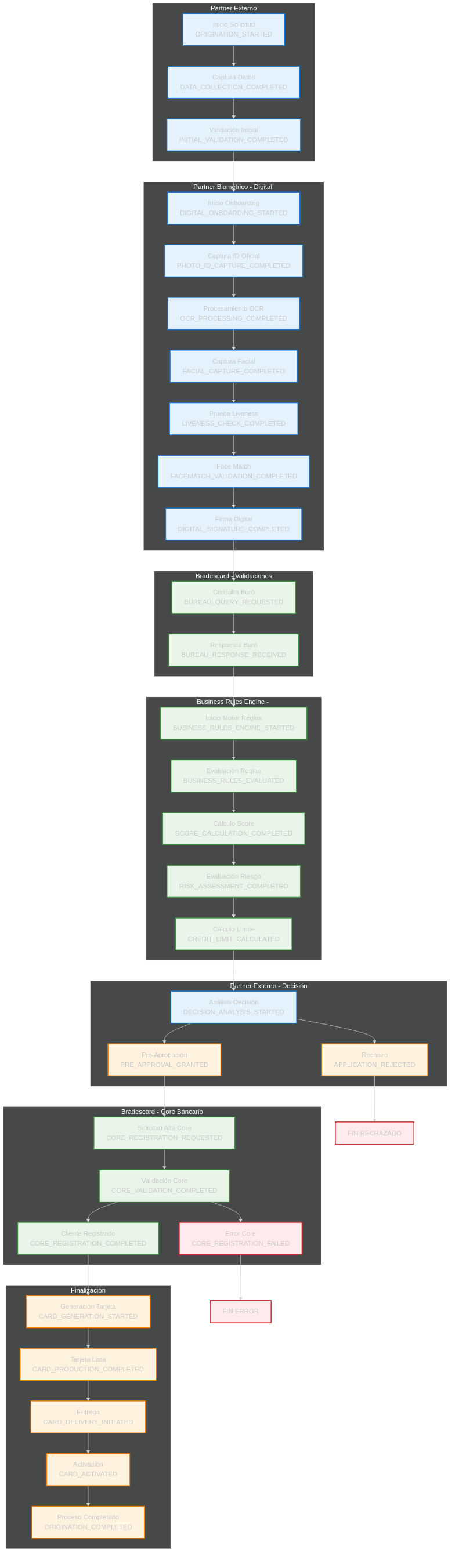
#### **2. Business Rules Engine / Scoring Partners (Típicos en México)**

* **FICO México**: Modelos de scoring crediticio y motor de reglas
* **SAS Risk Management**: Plataforma de gestión de riesgo crediticio
* **Equifax Veraz**: Scoring y business rules locales
* **Trans Union México**: Modelos predictivos y reglas de negocio
* **Providers Locales**: Motores de reglas personalizados para Bradescard

### Objetivo de la Bitácora

* **Auditoría completa** del proceso de originación de tarjetas de crédito
* **Trazabilidad** de los 10,000 casos mensuales aproximadamente
* **Análisis de abandono** del ~30% de casos que no avanzan
* **Compliance** con regulaciones financieras mexicanas (CNBV, Condusef)
* **Interfaz genérica** para múltiples partners actuales y futuros

## Proceso Estándar de Originación de Tarjetas de Crédito



Flujo del Proceso de Originación

*Figura 1: Flujo Completo del Proceso de Originación de Tarjetas de Crédito. Muestra los 51 eventos estandarizados desde la solicitud inicial hasta la activación de la tarjeta, incluyendo integración con Partner Biométrico, Business Rules Engine y Core Bancario.*

## Catálogo Completo de Eventos Estándar (51 Eventos)

### 1. **Eventos de Inicio y Captura (Partner)**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| ORIGINATION\_STARTED | Inicio del proceso de originación | applicationId, partnerId, productType, channel |
| DATA\_COLLECTION\_STARTED | Inicio captura de datos del solicitante | applicationId, step, formType |
| DATA\_COLLECTION\_COMPLETED | Captura de datos completada | applicationId, dataFields, completeness |
| INITIAL\_VALIDATION\_STARTED | Inicio validaciones básicas | applicationId, validationType |
| INITIAL\_VALIDATION\_COMPLETED | Validaciones iniciales completadas | applicationId, validationResult, errors |

### 2. **Eventos de Digital Onboarding Biométrico (Partner Biométrico)**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| DIGITAL\_ONBOARDING\_STARTED | Inicio proceso onboarding digital | applicationId, onboardingSession, deviceInfo, ipAddress |
| PHOTO\_ID\_CAPTURE\_STARTED | Inicio captura documento oficial | applicationId, documentType, captureMethod |
| PHOTO\_ID\_CAPTURE\_COMPLETED | Captura de documento completada | applicationId, documentImages, quality, timestamp |
| OCR\_PROCESSING\_STARTED | Inicio procesamiento OCR | applicationId, ocrProvider, documentType |
| OCR\_PROCESSING\_COMPLETED | OCR procesado exitosamente | applicationId, extractedData, confidence, validationFlags |
| OCR\_PROCESSING\_FAILED | Fallo en procesamiento OCR | applicationId, errorCode, errorReason, retryable |
| FACIAL\_CAPTURE\_STARTED | Inicio captura biométrica facial | applicationId, biometricSession, deviceCapabilities |
| FACIAL\_CAPTURE\_COMPLETED | Captura facial completada | applicationId, biometricTemplate, quality, attempts |
| LIVENESS\_CHECK\_STARTED | Inicio prueba de vida | applicationId, livenessType, challengeType |
| LIVENESS\_CHECK\_COMPLETED | Prueba de vida completada | applicationId, livenessResult, confidence, spoofingDetected |
| LIVENESS\_CHECK\_FAILED | Fallo en prueba de vida | applicationId, failureReason, suspiciousActivity, retryAllowed |
| FACEMATCH\_VALIDATION\_STARTED | Inicio validación facial | applicationId, referenceImage, candidateImage |
| FACEMATCH\_VALIDATION\_COMPLETED | Validación facial completada | applicationId, matchScore, threshold, matchResult |
| DIGITAL\_SIGNATURE\_STARTED | Inicio firma digital | applicationId, documentHash, signatureMethod |
| DIGITAL\_SIGNATURE\_COMPLETED | Firma digital completada | applicationId, signatureData, certificate, timestamp |
| DIGITAL\_ONBOARDING\_COMPLETED | Onboarding digital completado | applicationId, overallScore, fraudIndicators, recommendations |

### 3. **Eventos de Consultas Externas (Bradescard)**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| BUREAU\_QUERY\_REQUESTED | Solicitud consulta buró de crédito | applicationId, bureauProvider, queryType, curp |
| BUREAU\_RESPONSE\_RECEIVED | Respuesta de buró recibida | applicationId, bureauScore, riskLevel, recommendations |
| BUREAU\_QUERY\_FAILED | Fallo en consulta buró | applicationId, errorCode, providerError, retryable |

### 4. **Eventos de Business Rules Engine y Scoring**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| BUSINESS\_RULES\_ENGINE\_STARTED | Inicio motor reglas de negocio | applicationId, rulesetVersion, inputParameters |
| BUSINESS\_RULES\_EVALUATED | Reglas de negocio evaluadas | applicationId, rulesExecuted, ruleResults, overrides |
| RISK\_VARIABLES\_CALCULATED | Variables de riesgo calculadas | applicationId, variables, dataSourced, weights |
| SCORE\_CALCULATION\_STARTED | Inicio cálculo score crediticio | applicationId, scoreModel, modelVersion, inputData |
| SCORE\_CALCULATION\_COMPLETED | Score crediticio calculado | applicationId, finalScore, scoreComponents, confidence |
| RISK\_ASSESSMENT\_COMPLETED | Evaluación de riesgo completada | applicationId, riskLevel, riskFactors, mitigations |
| CREDIT\_LIMIT\_CALCULATION\_STARTED | Inicio cálculo límite crédito | applicationId, policyRules, baseLimit |
| CREDIT\_LIMIT\_CALCULATED | Límite de crédito calculado | applicationId, recommendedLimit, limitFactors, restrictions |
| POLICY\_RULES\_APPLIED | Reglas de política aplicadas | applicationId, appliedPolicies, exceptions, approvals |
| FRAUD\_SCREENING\_COMPLETED | Screening antifraude completado | applicationId, fraudScore, fraudIndicators, action |

### 5. **Eventos de Decisión (Partner)**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| DECISION\_ANALYSIS\_STARTED | Inicio análisis de decisión | applicationId, decisionEngine, criteria |
| PRE\_APPROVAL\_GRANTED | Pre-aprobación otorgada | applicationId, approvedLimit, productOffered, conditions |
| APPLICATION\_REJECTED | Solicitud rechazada | applicationId, rejectionReason, rejectionCode, appealOption |
| MANUAL\_REVIEW\_REQUIRED | Requiere revisión manual | applicationId, reviewReason, assignedAnalyst |

### 6. **Eventos de Core Bancario (Bradescard)**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| CORE\_REGISTRATION\_REQUESTED | Solicitud alta en core bancario | applicationId, customerData, productConfig |
| CORE\_VALIDATION\_COMPLETED | Validación en core completada | applicationId, validationStatus, customerId |
| CORE\_REGISTRATION\_COMPLETED | Cliente registrado exitosamente | applicationId, customerId, accountNumber, cardNumber |
| CORE\_REGISTRATION\_FAILED | Fallo en registro de core | applicationId, errorCode, errorDescription, retryable |

### 7. **Eventos de Producción y Entrega**

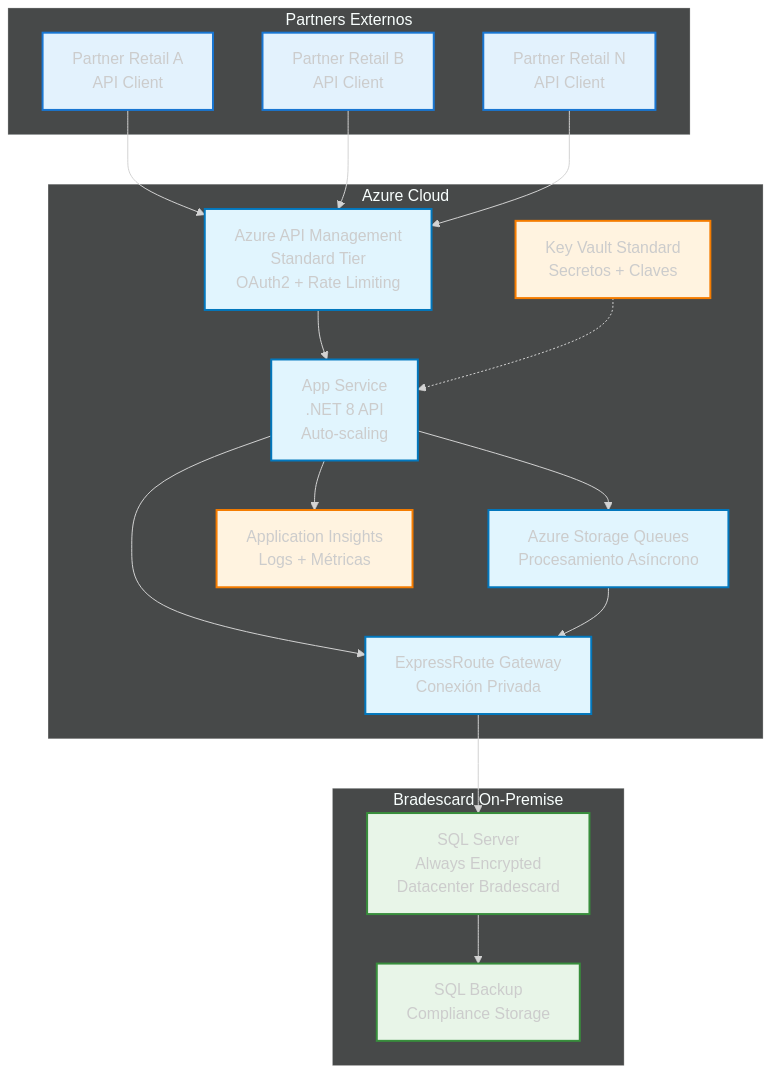
|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| CARD\_GENERATION\_STARTED | Inicio generación de tarjeta | applicationId, customerId, cardType, deliveryAddress |
| CARD\_PRODUCTION\_COMPLETED | Tarjeta producida | applicationId, cardNumber, expirationDate, trackingNumber |
| CARD\_DELIVERY\_INITIATED | Envío de tarjeta iniciado | applicationId, carrier, trackingNumber, estimatedDelivery |
| CARD\_DELIVERED | Tarjeta entregada | applicationId, deliveryDate, recipientName |
| CARD\_ACTIVATED | Tarjeta activada por cliente | applicationId, activationDate, activationChannel |
| ORIGINATION\_COMPLETED | Proceso completado exitosamente | applicationId, completionDate, totalProcessTime |

### 8. **Eventos de Error y Abandono**

|  |  |  |
| --- | --- | --- |
| Evento | Descripción | Datos Requeridos |
| PROCESS\_ABANDONED | Cliente abandonó el proceso | applicationId, lastStep, timeToAbandon, abandonReason |
| SYSTEM\_ERROR | Error de sistema | applicationId, errorType, systemComponent, errorDetails |
| TIMEOUT\_ERROR | Timeout en proceso | applicationId, timeoutStep, maxWaitTime, actualWaitTime |
| VALIDATION\_ERROR | Error de validación | applicationId, validationField, errorMessage, correctionRequired |

# 2. ARQUITECTURA DE SOLUCIÓN AZURE

## Arquitectura Simplificada para Auditoría Financiera



Arquitectura Híbrida Azure + On-Premise

*Figura 2: Arquitectura Simplificada de Solución Azure para Auditoría de Originación. Diseño pragmático enfocado en funcionalidad core con compliance CNBV y escalabilidad gradual.*

La arquitectura implementa un patrón simple y efectivo con componentes esenciales:

### **Capa de Ingreso**

* **Azure API Management (Standard)**: Punto único de entrada con autenticación básica
* **Gestión de Partners**: Rate limiting y políticas por partner
* **Seguridad**: OAuth2 + certificados SSL

### **Capa de API**

* **App Service (.NET 8)**: API REST para eventos de auditoría
* **Auto-scaling**: Escalamiento horizontal basado en demanda
* **Event Validator**: Validación de esquemas JSON

### **Capa de Procesamiento**

* **Azure Storage Queues**: Cola simple para procesamiento asíncrono
* **Background Jobs**: Procesamiento de eventos en lotes
* **Retry Logic**: Manejo de errores y reintentos automáticos

### **Capa de Persistencia**

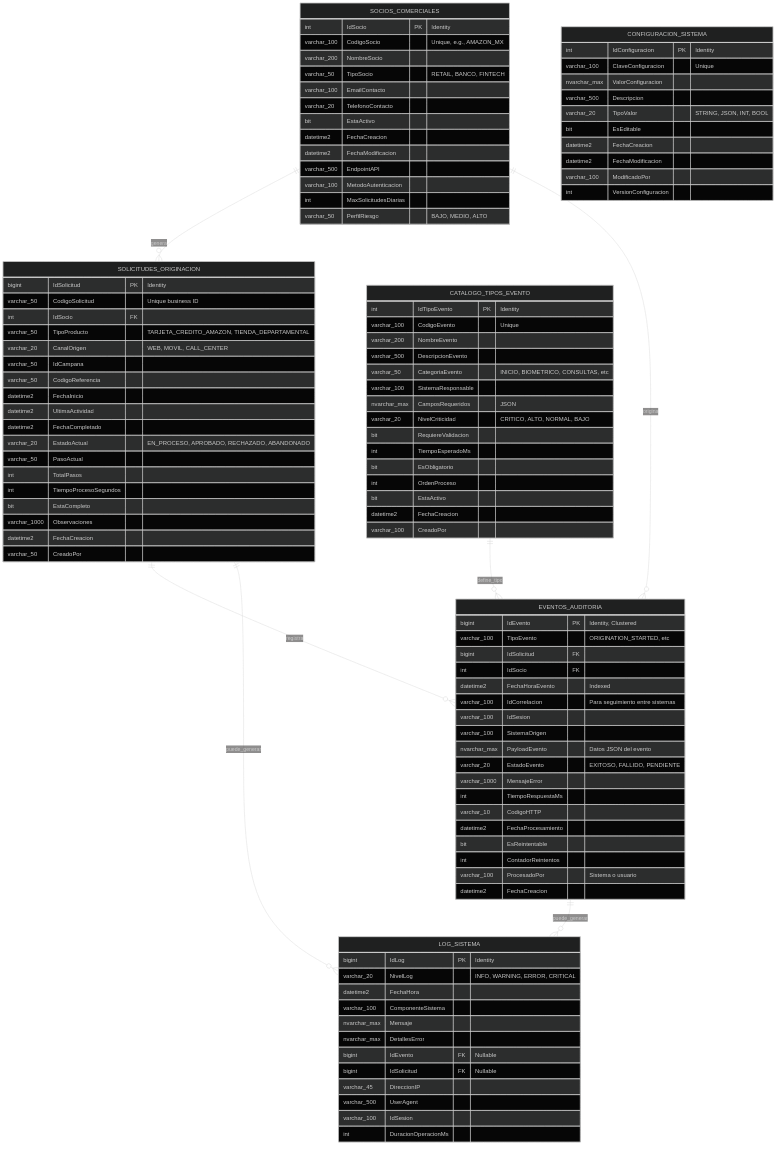
* **SQL Server On-Premise**: Base de datos principal con Always Encrypted
* **Azure ExpressRoute**: Conectividad privada y segura a datacenter Bradescard
* **Blob Storage (Cool)**: Archivo de largo plazo para compliance (opcional)

### **Capa de Monitoreo Básico**

* **Application Insights (Basic)**: Logs y métricas esenciales
* **Azure Monitor**: Alertas básicas de health y performance
* **Key Vault (Standard)**: Gestión de secretos y claves

# 3. MODELO DE DATOS Y COMPLIANCE

## Esquema de Base de Datos Especializado



Modelo de Base de Datos

*Figura 3: Modelo de Datos con Always Encrypted y Particionamiento. Esquema de base de datos optimizado para auditoría financiera con encriptación de datos PII y cumplimiento de regulaciones mexicanas.*

### Tablas Principales en SQL Server On-Premise

#### **Tabla SociosComerciales** - Gestiona información de retailers y partners externos

CREATE TABLE [dbo].[SociosComerciales] (  
 [IdSocio] INT IDENTITY(1,1) NOT NULL, -- Identificador único del socio comercial  
 [CodigoSocio] VARCHAR(100) NOT NULL, -- Código de negocio del socio (ej: RETAIL\_A, RETAIL\_B)  
 [NombreSocio] VARCHAR(200) NOT NULL, -- Razón social del socio comercial  
 [TipoSocio] VARCHAR(50) NOT NULL DEFAULT 'RETAIL', -- Tipo: RETAIL, BANCO, FINTECH  
 [ClaveAPI] VARCHAR(500) NULL, -- Clave API encriptada para autenticación  
 [MaxSolicitudesDiarias] INT NOT NULL DEFAULT 1000, -- Límite diario de solicitudes permitidas  
 [EstaActivo] BIT NOT NULL DEFAULT 1, -- Indica si el socio está activo  
 [FechaCreacion] DATETIME2(3) NOT NULL DEFAULT SYSUTCDATETIME(), -- Fecha de registro del socio  
 [FechaActualizacion] DATETIME2(3) NULL, -- Última fecha de modificación  
 [ContactoEmail] VARCHAR(200) NULL, -- Email de contacto técnico  
 [ContactoTelefono] VARCHAR(20) NULL, -- Teléfono de contacto  
   
 CONSTRAINT [PK\_SociosComerciales] PRIMARY KEY CLUSTERED ([IdSocio]),  
 CONSTRAINT [UQ\_SociosComerciales\_Codigo] UNIQUE ([CodigoSocio])  
);

#### **Tabla SolicitudesOriginacion** - Registra cada solicitud de tarjeta de crédito

CREATE TABLE [dbo].[SolicitudesOriginacion] (  
 [IdSolicitud] BIGINT IDENTITY(1,1) NOT NULL, -- Identificador único de la solicitud  
 [CodigoSolicitud] VARCHAR(50) NOT NULL, -- Código de negocio único de la solicitud  
 [IdSocio] INT NOT NULL, -- Referencia al socio comercial que originó  
 [TipoProducto] VARCHAR(50) NOT NULL, -- Tipo de producto: TARJETA\_CREDITO\_RETAIL, DEPARTAMENTAL  
 [CanalOrigen] VARCHAR(20) NOT NULL, -- Canal de origen: WEB, MOBILE, CALL\_CENTER  
 [FechaInicio] DATETIME2(3) NOT NULL, -- Timestamp de inicio del proceso  
 [UltimaActividad] DATETIME2(3) NOT NULL, -- Última actividad registrada  
 [FechaCompletado] DATETIME2(3) NULL, -- Fecha de completado (si aplica)  
 [EstadoActual] VARCHAR(30) NOT NULL DEFAULT 'EN\_PROCESO', -- Estado: EN\_PROCESO, APROBADO, RECHAZADO, ABANDONADO  
 [PasoActual] VARCHAR(100) NULL, -- Describe el paso actual del proceso  
 [TotalPasos] INT NULL, -- Número total de pasos esperados  
 [TiempoProcesoSegundos] INT NULL, -- Tiempo total de procesamiento en segundos  
 [EstaCompleto] BIT NOT NULL DEFAULT 0, -- Indica si el proceso está completado  
 [Observaciones] VARCHAR(1000) NULL, -- Notas adicionales sobre la solicitud  
 [IdCampana] VARCHAR(50) NULL, -- Identificador de campaña de marketing  
 [CodigoReferido] VARCHAR(50) NULL, -- Código de referido si aplica  
   
 CONSTRAINT [PK\_SolicitudesOriginacion] PRIMARY KEY CLUSTERED ([IdSolicitud]),  
 CONSTRAINT [FK\_SolicitudesOriginacion\_SociosComerciales] FOREIGN KEY ([IdSocio])   
 REFERENCES [dbo].[SociosComerciales]([IdSocio]),  
 CONSTRAINT [UQ\_SolicitudesOriginacion\_Codigo] UNIQUE ([CodigoSolicitud])  
);

#### **Tabla EventosAuditoria** - Almacena todos los eventos de trazabilidad del proceso

CREATE TABLE [dbo].[EventosAuditoria] (  
 [IdEvento] BIGINT IDENTITY(1,1) NOT NULL, -- Identificador único del evento  
 [TipoEvento] VARCHAR(100) NOT NULL, -- Tipo de evento (ej: BUREAU\_QUERY\_REQUESTED)  
 [IdSolicitud] BIGINT NOT NULL, -- Referencia a la solicitud de originación  
 [IdSocio] INT NOT NULL, -- Referencia al socio comercial  
 [FechaHoraEvento] DATETIME2(3) NOT NULL, -- Timestamp exacto del evento  
 [IdCorrelacion] VARCHAR(100) NOT NULL, -- ID para correlacionar eventos relacionados  
 [IdSesion] VARCHAR(100) NULL, -- Identificador de sesión del usuario  
 [SistemaOrigen] VARCHAR(100) NOT NULL, -- Sistema que generó el evento  
 [PayloadEvento] NVARCHAR(MAX) NULL, -- Datos del evento en formato JSON  
 [EstadoEvento] VARCHAR(20) NOT NULL DEFAULT 'EXITOSO', -- Estado: EXITOSO, FALLIDO, PENDIENTE  
 [MensajeError] VARCHAR(1000) NULL, -- Mensaje de error si el evento falló  
 [TiempoRespuestaMs] INT NULL, -- Tiempo de respuesta en milisegundos  
 [CodigoHTTP] VARCHAR(10) NULL, -- Código de respuesta HTTP  
 [FechaProcesado] DATETIME2(3) NOT NULL DEFAULT SYSUTCDATETIME(), -- Fecha de procesamiento  
 [EsReintentoble] BIT NOT NULL DEFAULT 0, -- Indica si el evento puede ser reintentado  
 [ContadorReintentos] INT NOT NULL DEFAULT 0, -- Número de reintentos realizados  
 [ProcesadoPor] VARCHAR(100) NULL, -- Sistema o usuario que procesó el evento  
 [VersionEvento] VARCHAR(10) NOT NULL DEFAULT '1.0', -- Versión del esquema del evento  
   
 CONSTRAINT [PK\_EventosAuditoria] PRIMARY KEY CLUSTERED ([IdEvento]),  
 CONSTRAINT [FK\_EventosAuditoria\_SolicitudesOriginacion] FOREIGN KEY ([IdSolicitud])   
 REFERENCES [dbo].[SolicitudesOriginacion]([IdSolicitud]),  
 CONSTRAINT [FK\_EventosAuditoria\_SociosComerciales] FOREIGN KEY ([IdSocio])   
 REFERENCES [dbo].[SociosComerciales]([IdSocio])  
);  
  
-- Índices optimizados para consultas frecuentes  
CREATE NONCLUSTERED INDEX [IX\_EventosAuditoria\_Solicitud\_Fecha]  
ON [dbo].[EventosAuditoria] ([IdSolicitud], [FechaHoraEvento] DESC)  
INCLUDE ([TipoEvento], [EstadoEvento], [TiempoRespuestaMs]);  
  
CREATE NONCLUSTERED INDEX [IX\_EventosAuditoria\_Socio\_Fecha]  
ON [dbo].[EventosAuditoria] ([IdSocio], [FechaHoraEvento] DESC)  
INCLUDE ([IdSolicitud], [TipoEvento], [EstadoEvento]);  
  
CREATE NONCLUSTERED INDEX [IX\_EventosAuditoria\_Correlacion]  
ON [dbo].[EventosAuditoria] ([IdCorrelacion])  
INCLUDE ([IdSolicitud], [FechaHoraEvento], [TipoEvento]);  
  
CREATE NONCLUSTERED INDEX [IX\_EventosAuditoria\_TipoEvento\_Estado]  
ON [dbo].[EventosAuditoria] ([TipoEvento], [EstadoEvento], [FechaHoraEvento] DESC);  
  
#### \*\*Tabla ConfiguracionSistema\*\* - Configuraciones y parámetros del sistema  
```sql  
CREATE TABLE [dbo].[ConfiguracionSistema] (  
 [IdConfiguracion] INT IDENTITY(1,1) NOT NULL, -- Identificador único de configuración  
 [ClaveConfiguracion] VARCHAR(100) NOT NULL, -- Clave única de la configuración  
 [ValorConfiguracion] NVARCHAR(MAX) NOT NULL, -- Valor de la configuración (JSON o texto)  
 [Descripcion] VARCHAR(500) NOT NULL, -- Descripción del propósito de la configuración  
 [TipoValor] VARCHAR(20) NOT NULL DEFAULT 'STRING', -- Tipo: STRING, JSON, INT, BOOL  
 [EsEditable] BIT NOT NULL DEFAULT 1, -- Indica si la configuración puede ser modificada  
 [FechaCreacion] DATETIME2(3) NOT NULL DEFAULT SYSUTCDATETIME(), -- Fecha de creación  
 [FechaModificacion] DATETIME2(3) NULL, -- Última fecha de modificación  
 [ModificadoPor] VARCHAR(100) NULL, -- Usuario que realizó la modificación  
 [VersionConfiguracion] INT NOT NULL DEFAULT 1, -- Versión de la configuración  
   
 CONSTRAINT [PK\_ConfiguracionSistema] PRIMARY KEY CLUSTERED ([IdConfiguracion]),  
 CONSTRAINT [UQ\_ConfiguracionSistema\_Clave] UNIQUE ([ClaveConfiguracion])  
);  
  
-- Insertar configuraciones iniciales del sistema  
INSERT INTO [dbo].[ConfiguracionSistema] ([ClaveConfiguracion], [ValorConfiguracion], [Descripcion], [TipoValor], [EsEditable]) VALUES  
('VERSION\_ESQUEMA\_EVENTOS', '1.0', 'Versión actual del esquema de eventos de auditoría', 'STRING', 0),  
('TIEMPO\_RETENCION\_EVENTOS\_DIAS', '2555', 'Días de retención de eventos (7 años para compliance CNBV)', 'INT', 1),  
('HABILITAR\_VALIDACION\_ESQUEMA', 'true', 'Indica si se debe validar el esquema JSON de eventos', 'BOOL', 1),  
('MAX\_REINTENTOS\_EVENTO', '3', 'Número máximo de reintentos para procesar un evento', 'INT', 1),  
('TIMEOUT\_PROCESAMIENTO\_MS', '30000', 'Timeout en milisegundos para procesamiento de eventos', 'INT', 1),  
('ESQUEMAS\_EVENTOS\_JSON', '{"CONSULTA\_BURO\_SOLICITADA":{"required":["bureauProvider","tipoConsulta"]}}', 'Esquemas JSON para validación de eventos específicos', 'JSON', 1);

#### **Tabla LogSistema** - Registro de eventos del sistema y errores

CREATE TABLE [dbo].[LogSistema] (  
 [IdLog] BIGINT IDENTITY(1,1) NOT NULL, -- Identificador único del log  
 [NivelLog] VARCHAR(20) NOT NULL, -- Nivel: INFO, WARNING, ERROR, CRITICAL  
 [FechaHora] DATETIME2(3) NOT NULL DEFAULT SYSUTCDATETIME(), -- Timestamp del log  
 [ComponenteSistema] VARCHAR(100) NOT NULL, -- Componente que generó el log  
 [Mensaje] NVARCHAR(MAX) NOT NULL, -- Mensaje del log  
 [DetallesError] NVARCHAR(MAX) NULL, -- Stack trace o detalles adicionales del error  
 [IdEvento] BIGINT NULL, -- Referencia al evento relacionado (si aplica)  
 [IdSolicitud] BIGINT NULL, -- Referencia a la solicitud relacionada (si aplica)  
 [DireccionIP] VARCHAR(45) NULL, -- IP del cliente que originó la operación  
 [UserAgent] VARCHAR(500) NULL, -- User agent del cliente  
 [IdSesion] VARCHAR(100) NULL, -- Identificador de sesión  
 [DuracionOperacionMs] INT NULL, -- Duración de la operación en milisegundos  
   
 CONSTRAINT [PK\_LogSistema] PRIMARY KEY CLUSTERED ([IdLog]),  
 CONSTRAINT [FK\_LogSistema\_EventosAuditoria] FOREIGN KEY ([IdEvento])   
 REFERENCES [dbo].[EventosAuditoria]([IdEvento])  
);  
  
-- Índice para consultas de logs por fecha y nivel  
CREATE NONCLUSTERED INDEX [IX\_LogSistema\_Fecha\_Nivel]  
ON [dbo].[LogSistema] ([FechaHora] DESC, [NivelLog])  
INCLUDE ([ComponenteSistema], [Mensaje]);  
  
#### \*\*Tabla CatalogoTiposEvento\*\* - Catálogo maestro de tipos de eventos  
```sql  
CREATE TABLE [dbo].[CatalogoTiposEvento] (  
 [IdTipoEvento] INT IDENTITY(1,1) NOT NULL, -- Identificador único del tipo de evento  
 [CodigoEvento] VARCHAR(100) NOT NULL, -- Código único del evento (ej: SOLICITUD\_INICIADA)  
 [NombreEvento] VARCHAR(200) NOT NULL, -- Nombre descriptivo del evento  
 [DescripcionEvento] VARCHAR(500) NOT NULL, -- Descripción detallada del evento  
 [CategoriaEvento] VARCHAR(50) NOT NULL, -- Categoría: INICIO, BIOMETRICO, CONSULTAS, REGLAS\_NEGOCIO, DECISION, CORE\_BANCARIO, PRODUCCION, ERROR  
 [SistemaResponsable] VARCHAR(100) NOT NULL, -- Sistema que típicamente genera este evento  
 [CamposRequeridos] NVARCHAR(MAX) NULL, -- JSON con los campos requeridos para este evento  
 [NivelCriticidad] VARCHAR(20) NOT NULL DEFAULT 'NORMAL', -- CRITICO, ALTO, NORMAL, BAJO  
 [RequiereValidacion] BIT NOT NULL DEFAULT 0, -- Indica si requiere validación adicional  
 [TiempoEsperadoMs] INT NULL, -- Tiempo esperado de procesamiento en milisegundos  
 [EsObligatorio] BIT NOT NULL DEFAULT 1, -- Indica si es obligatorio en el flujo  
 [OrdenProceso] INT NULL, -- Orden sugerido en el proceso de originación  
 [EstaActivo] BIT NOT NULL DEFAULT 1, -- Indica si el evento está activo  
 [FechaCreacion] DATETIME2(3) NOT NULL DEFAULT SYSUTCDATETIME(), -- Fecha de creación del registro  
 [FechaModificacion] DATETIME2(3) NULL, -- Última fecha de modificación  
 [CreadoPor] VARCHAR(100) NOT NULL DEFAULT 'SISTEMA', -- Usuario que creó el registro  
   
 CONSTRAINT [PK\_CatalogoTiposEvento] PRIMARY KEY CLUSTERED ([IdTipoEvento]),  
 CONSTRAINT [UQ\_CatalogoTiposEvento\_Codigo] UNIQUE ([CodigoEvento])  
);  
  
-- Índice para consultas por categoría y orden  
CREATE NONCLUSTERED INDEX [IX\_CatalogoTiposEvento\_Categoria\_Orden]  
ON [dbo].[CatalogoTiposEvento] ([CategoriaEvento], [OrdenProceso])  
INCLUDE ([CodigoEvento], [NombreEvento], [EstaActivo]);  
  
-- Índice para consultas por sistema responsable  
CREATE NONCLUSTERED INDEX [IX\_CatalogoTiposEvento\_Sistema]  
ON [dbo].[CatalogoTiposEvento] ([SistemaResponsable], [EstaActivo])  
INCLUDE ([CodigoEvento], [NivelCriticidad]);  
  
#### \*\*Inserts para los 51 Tipos de Eventos Estándar\*\*  
```sql  
-- ===================================================================  
-- INSERTS PARA CATÁLOGO DE TIPOS DE EVENTOS (51 EVENTOS ESTÁNDAR)  
-- ===================================================================  
  
-- Categoría 1: EVENTOS DE INICIO Y CAPTURA (Partner)  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('ORIGINATION\_STARTED', 'Inicio del Proceso de Originación', 'Inicio del proceso de originación de tarjeta de crédito', 'INICIO', 'SISTEMA\_ORIGINACION\_RETAIL', '["applicationId","partnerId","productType","channel"]', 'CRITICO', 1, 100),  
('DATA\_COLLECTION\_STARTED', 'Inicio Captura de Datos', 'Inicio de la captura de datos del solicitante', 'INICIO', 'SISTEMA\_ORIGINACION\_RETAIL', '["applicationId","step","formType"]', 'ALTO', 2, 50),  
('DATA\_COLLECTION\_COMPLETED', 'Captura de Datos Completada', 'Captura de datos del solicitante completada exitosamente', 'INICIO', 'SISTEMA\_ORIGINACION\_RETAIL', '["applicationId","dataFields","completeness"]', 'ALTO', 3, 200),  
('INITIAL\_VALIDATION\_STARTED', 'Inicio Validaciones Básicas', 'Inicio de validaciones iniciales de datos', 'INICIO', 'MOTOR\_VALIDACIONES', '["applicationId","validationType"]', 'NORMAL', 4, 300),  
('INITIAL\_VALIDATION\_COMPLETED', 'Validaciones Iniciales Completadas', 'Validaciones iniciales de datos completadas', 'INICIO', 'MOTOR\_VALIDACIONES', '["applicationId","validationResult","errors"]', 'ALTO', 5, 500);  
  
-- Categoría 2: EVENTOS DE DIGITAL ONBOARDING BIOMÉTRICO (Partner Biométrico)  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('DIGITAL\_ONBOARDING\_STARTED', 'Inicio Onboarding Digital', 'Inicio del proceso de onboarding digital biométrico', 'BIOMETRICO', 'PARTNER\_BIOMETRICO', '["applicationId","onboardingSession","deviceInfo","ipAddress"]', 'CRITICO', 6, 100),  
('PHOTO\_ID\_CAPTURE\_STARTED', 'Inicio Captura Documento', 'Inicio de captura de documento oficial de identificación', 'BIOMETRICO', 'PARTNER\_BIOMETRICO', '["applicationId","documentType","captureMethod"]', 'ALTO', 7, 200),  
('PHOTO\_ID\_CAPTURE\_COMPLETED', 'Captura Documento Completada', 'Captura de documento oficial completada exitosamente', 'BIOMETRICO', 'PARTNER\_BIOMETRICO', '["applicationId","documentImages","quality","timestamp"]', 'ALTO', 8, 300),  
('OCR\_PROCESSING\_STARTED', 'Inicio Procesamiento OCR', 'Inicio del procesamiento OCR del documento capturado', 'BIOMETRICO', 'OCR\_ENGINE', '["applicationId","ocrProvider","documentType"]', 'NORMAL', 9, 1000),  
('OCR\_PROCESSING\_COMPLETED', 'OCR Procesado Exitosamente', 'Procesamiento OCR completado con extracción de datos', 'BIOMETRICO', 'OCR\_ENGINE', '["applicationId","extractedData","confidence","validationFlags"]', 'ALTO', 10, 2000),  
('OCR\_PROCESSING\_FAILED', 'Fallo en Procesamiento OCR', 'Error en el procesamiento OCR del documento', 'BIOMETRICO', 'OCR\_ENGINE', '["applicationId","errorCode","errorReason","retryable"]', 'ALTO', 11, 1000),  
('FACIAL\_CAPTURE\_STARTED', 'Inicio Captura Biométrica Facial', 'Inicio de captura biométrica facial del solicitante', 'BIOMETRICO', 'PARTNER\_BIOMETRICO', '["applicationId","biometricSession","deviceCapabilities"]', 'ALTO', 12, 300),  
('FACIAL\_CAPTURE\_COMPLETED', 'Captura Facial Completada', 'Captura biométrica facial completada exitosamente', 'BIOMETRICO', 'PARTNER\_BIOMETRICO', '["applicationId","biometricTemplate","quality","attempts"]', 'ALTO', 13, 500),  
('LIVENESS\_CHECK\_STARTED', 'Inicio Prueba de Vida', 'Inicio de prueba de vida (liveness) biométrica', 'BIOMETRICO', 'LIVENESS\_ENGINE', '["applicationId","livenessType","challengeType"]', 'CRITICO', 14, 2000),  
('LIVENESS\_CHECK\_COMPLETED', 'Prueba de Vida Completada', 'Prueba de vida biométrica completada exitosamente', 'BIOMETRICO', 'LIVENESS\_ENGINE', '["applicationId","livenessResult","confidence","spoofingDetected"]', 'CRITICO', 15, 3000),  
('LIVENESS\_CHECK\_FAILED', 'Fallo en Prueba de Vida', 'Fallo en la prueba de vida biométrica', 'BIOMETRICO', 'LIVENESS\_ENGINE', '["applicationId","failureReason","suspiciousActivity","retryAllowed"]', 'CRITICO', 16, 2000),  
('FACEMATCH\_VALIDATION\_STARTED', 'Inicio Validación Facial', 'Inicio de validación de coincidencia facial', 'BIOMETRICO', 'FACEMATCH\_ENGINE', '["applicationId","referenceImage","candidateImage"]', 'CRITICO', 17, 1500),  
('FACEMATCH\_VALIDATION\_COMPLETED', 'Validación Facial Completada', 'Validación de coincidencia facial completada', 'BIOMETRICO', 'FACEMATCH\_ENGINE', '["applicationId","matchScore","threshold","matchResult"]', 'CRITICO', 18, 2000),  
('DIGITAL\_SIGNATURE\_STARTED', 'Inicio Firma Digital', 'Inicio del proceso de firma digital de documentos', 'BIOMETRICO', 'SIGNATURE\_ENGINE', '["applicationId","documentHash","signatureMethod"]', 'ALTO', 19, 500),  
('DIGITAL\_SIGNATURE\_COMPLETED', 'Firma Digital Completada', 'Proceso de firma digital completado exitosamente', 'BIOMETRICO', 'SIGNATURE\_ENGINE', '["applicationId","signatureData","certificate","timestamp"]', 'ALTO', 20, 800),  
('DIGITAL\_ONBOARDING\_COMPLETED', 'Onboarding Digital Completado', 'Proceso completo de onboarding digital finalizado', 'BIOMETRICO', 'PARTNER\_BIOMETRICO', '["applicationId","overallScore","fraudIndicators","recommendations"]', 'CRITICO', 21, 1000);  
  
-- Categoría 3: EVENTOS DE CONSULTAS EXTERNAS (Bradescard)  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('BUREAU\_QUERY\_REQUESTED', 'Solicitud Consulta Buró de Crédito', 'Solicitud enviada al buró de crédito para consulta', 'CONSULTAS', 'BURO\_CREDITO\_ADAPTER', '["applicationId","bureauProvider","queryType","curp"]', 'CRITICO', 22, 500),  
('BUREAU\_RESPONSE\_RECEIVED', 'Respuesta de Buró Recibida', 'Respuesta del buró de crédito recibida exitosamente', 'CONSULTAS', 'BURO\_CREDITO\_ADAPTER', '["applicationId","bureauScore","riskLevel","recommendations"]', 'CRITICO', 23, 3000),  
('BUREAU\_QUERY\_FAILED', 'Fallo en Consulta Buró', 'Error en la consulta al buró de crédito', 'CONSULTAS', 'BURO\_CREDITO\_ADAPTER', '["applicationId","errorCode","providerError","retryable"]', 'CRITICO', 24, 1000);  
  
-- Categoría 4: EVENTOS DE BUSINESS RULES ENGINE Y SCORING  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('BUSINESS\_RULES\_ENGINE\_STARTED', 'Inicio Motor Reglas de Negocio', 'Inicio del procesamiento en motor de reglas de negocio', 'REGLAS\_NEGOCIO', 'BUSINESS\_RULES\_ENGINE', '["applicationId","rulesetVersion","inputParameters"]', 'ALTO', 25, 200),  
('BUSINESS\_RULES\_EVALUATED', 'Reglas de Negocio Evaluadas', 'Reglas de negocio evaluadas y procesadas', 'REGLAS\_NEGOCIO', 'BUSINESS\_RULES\_ENGINE', '["applicationId","rulesExecuted","ruleResults","overrides"]', 'ALTO', 26, 800),  
('RISK\_VARIABLES\_CALCULATED', 'Variables de Riesgo Calculadas', 'Cálculo de variables de riesgo completado', 'REGLAS\_NEGOCIO', 'RISK\_ENGINE', '["applicationId","variables","dataSourced","weights"]', 'ALTO', 27, 1000),  
('SCORE\_CALCULATION\_STARTED', 'Inicio Cálculo Score Crediticio', 'Inicio del cálculo de score crediticio', 'REGLAS\_NEGOCIO', 'SCORING\_ENGINE', '["applicationId","scoreModel","modelVersion","inputData"]', 'CRITICO', 28, 300),  
('SCORE\_CALCULATION\_COMPLETED', 'Score Crediticio Calculado', 'Cálculo de score crediticio completado', 'REGLAS\_NEGOCIO', 'SCORING\_ENGINE', '["applicationId","finalScore","scoreComponents","confidence"]', 'CRITICO', 29, 1500),  
('RISK\_ASSESSMENT\_COMPLETED', 'Evaluación de Riesgo Completada', 'Evaluación completa de riesgo finalizada', 'REGLAS\_NEGOCIO', 'RISK\_ENGINE', '["applicationId","riskLevel","riskFactors","mitigations"]', 'CRITICO', 30, 2000),  
('CREDIT\_LIMIT\_CALCULATION\_STARTED', 'Inicio Cálculo Límite Crédito', 'Inicio del cálculo de límite de crédito', 'REGLAS\_NEGOCIO', 'LIMIT\_ENGINE', '["applicationId","policyRules","baseLimit"]', 'ALTO', 31, 200),  
('CREDIT\_LIMIT\_CALCULATED', 'Límite de Crédito Calculado', 'Límite de crédito calculado y asignado', 'REGLAS\_NEGOCIO', 'LIMIT\_ENGINE', '["applicationId","recommendedLimit","limitFactors","restrictions"]', 'ALTO', 32, 800),  
('POLICY\_RULES\_APPLIED', 'Reglas de Política Aplicadas', 'Reglas de política corporativa aplicadas', 'REGLAS\_NEGOCIO', 'POLICY\_ENGINE', '["applicationId","appliedPolicies","exceptions","approvals"]', 'ALTO', 33, 500),  
('FRAUD\_SCREENING\_COMPLETED', 'Screening Antifraude Completado', 'Proceso de screening antifraude finalizado', 'REGLAS\_NEGOCIO', 'FRAUD\_ENGINE', '["applicationId","fraudScore","fraudIndicators","action"]', 'CRITICO', 34, 1200);  
  
-- Categoría 5: EVENTOS DE DECISIÓN (Partner)  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('DECISION\_ANALYSIS\_STARTED', 'Inicio Análisis de Decisión', 'Inicio del análisis para toma de decisión final', 'DECISION', 'MOTOR\_DECISIONES', '["applicationId","decisionEngine","criteria"]', 'CRITICO', 35, 300),  
('PRE\_APPROVAL\_GRANTED', 'Pre-aprobación Otorgada', 'Pre-aprobación de tarjeta de crédito otorgada', 'DECISION', 'MOTOR\_DECISIONES', '["applicationId","approvedLimit","productOffered","conditions"]', 'CRITICO', 36, 500),  
('APPLICATION\_REJECTED', 'Solicitud Rechazada', 'Solicitud de tarjeta de crédito rechazada', 'DECISION', 'MOTOR\_DECISIONES', '["applicationId","rejectionReason","rejectionCode","appealOption"]', 'CRITICO', 37, 300),  
('MANUAL\_REVIEW\_REQUIRED', 'Requiere Revisión Manual', 'Solicitud requiere revisión manual por analista', 'DECISION', 'MOTOR\_DECISIONES', '["applicationId","reviewReason","assignedAnalyst"]', 'ALTO', 38, 100);  
  
-- Categoría 6: EVENTOS DE CORE BANCARIO (Bradescard)  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('CORE\_REGISTRATION\_REQUESTED', 'Solicitud Alta Core Bancario', 'Solicitud de alta de cliente en core bancario', 'CORE\_BANCARIO', 'CORE\_BRADESCARD', '["applicationId","customerData","productConfig"]', 'CRITICO', 39, 500),  
('CORE\_VALIDATION\_COMPLETED', 'Validación Core Completada', 'Validación de datos en core bancario completada', 'CORE\_BANCARIO', 'CORE\_BRADESCARD', '["applicationId","validationStatus","customerId"]', 'CRITICO', 40, 2000),  
('CORE\_REGISTRATION\_COMPLETED', 'Cliente Registrado Exitosamente', 'Cliente registrado exitosamente en core bancario', 'CORE\_BANCARIO', 'CORE\_BRADESCARD', '["applicationId","customerId","accountNumber","cardNumber"]', 'CRITICO', 41, 3000),  
('CORE\_REGISTRATION\_FAILED', 'Fallo Registro Core', 'Error en el registro del cliente en core bancario', 'CORE\_BANCARIO', 'CORE\_BRADESCARD', '["applicationId","errorCode","errorDescription","retryable"]', 'CRITICO', 42, 1000);  
  
-- Categoría 7: EVENTOS DE PRODUCCIÓN Y ENTREGA  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('CARD\_GENERATION\_STARTED', 'Inicio Generación Tarjeta', 'Inicio del proceso de generación física de tarjeta', 'PRODUCCION', 'SISTEMA\_PRODUCCION\_TARJETAS', '["applicationId","customerId","cardType","deliveryAddress"]', 'ALTO', 43, 300),  
('CARD\_PRODUCTION\_COMPLETED', 'Tarjeta Producida', 'Tarjeta física producida exitosamente', 'PRODUCCION', 'SISTEMA\_PRODUCCION\_TARJETAS', '["applicationId","cardNumber","expirationDate","trackingNumber"]', 'ALTO', 44, 1800000),  
('CARD\_DELIVERY\_INITIATED', 'Envío Tarjeta Iniciado', 'Envío de tarjeta física iniciado con courier', 'PRODUCCION', 'SISTEMA\_LOGISTICA', '["applicationId","carrier","trackingNumber","estimatedDelivery"]', 'NORMAL', 45, 600000),  
('CARD\_DELIVERED', 'Tarjeta Entregada', 'Tarjeta física entregada al cliente', 'PRODUCCION', 'SISTEMA\_LOGISTICA', '["applicationId","deliveryDate","recipientName"]', 'ALTO', 46, 259200000),  
('CARD\_ACTIVATED', 'Tarjeta Activada', 'Tarjeta activada por el cliente', 'PRODUCCION', 'SISTEMA\_ACTIVACION', '["applicationId","activationDate","activationChannel"]', 'CRITICO', 47, 300),  
('ORIGINATION\_COMPLETED', 'Proceso Completado Exitosamente', 'Proceso completo de originación finalizado con éxito', 'PRODUCCION', 'SISTEMA\_ORIGINACION\_RETAIL', '["applicationId","completionDate","totalProcessTime"]', 'CRITICO', 48, 100);  
  
-- Categoría 8: EVENTOS DE ERROR Y ABANDONO  
INSERT INTO [dbo].[CatalogoTiposEvento]   
([CodigoEvento], [NombreEvento], [DescripcionEvento], [CategoriaEvento], [SistemaResponsable], [CamposRequeridos], [NivelCriticidad], [OrdenProceso], [TiempoEsperadoMs]) VALUES  
('PROCESS\_ABANDONED', 'Cliente Abandonó Proceso', 'Cliente abandonó el proceso de originación', 'ERROR', 'SISTEMA\_ORIGINACION\_RETAIL', '["applicationId","lastStep","timeToAbandon","abandonReason"]', 'ALTO', 49, 100),  
('SYSTEM\_ERROR', 'Error de Sistema', 'Error técnico del sistema durante el proceso', 'ERROR', 'MONITOR\_SISTEMA', '["applicationId","errorType","systemComponent","errorDetails"]', 'CRITICO', 50, 100),  
('TIMEOUT\_ERROR', 'Error de Timeout', 'Timeout en proceso por exceso de tiempo de espera', 'ERROR', 'MONITOR\_SISTEMA', '["applicationId","timeoutStep","maxWaitTime","actualWaitTime"]', 'ALTO', 51, 100),  
('VALIDATION\_ERROR', 'Error de Validación', 'Error en validación de datos o reglas de negocio', 'ERROR', 'MOTOR\_VALIDACIONES', '["applicationId","validationField","errorMessage","correctionRequired"]', 'NORMAL', 52, 100);  
  
-- Verificación de inserción  
SELECT   
 CategoriaEvento,  
 COUNT(\*) AS TotalEventos,  
 MIN(OrdenProceso) AS PrimerOrden,  
 MAX(OrdenProceso) AS UltimoOrden  
FROM [dbo].[CatalogoTiposEvento]  
GROUP BY CategoriaEvento  
ORDER BY MIN(OrdenProceso);  
  
PRINT '51 tipos de eventos insertados exitosamente en el catálogo';

#### **Stored Procedures para Gestión del Catálogo de Eventos**

-- Procedimiento para validar tipo de evento contra el catálogo  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ValidarTipoEvento]  
 @CodigoEvento VARCHAR(100),  
 @EsValido BIT OUTPUT,  
 @InfoEvento NVARCHAR(MAX) OUTPUT  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 DECLARE @IdTipoEvento INT;  
 DECLARE @NombreEvento VARCHAR(200);  
 DECLARE @CategoriaEvento VARCHAR(50);  
 DECLARE @EstaActivo BIT;  
 DECLARE @CamposRequeridos NVARCHAR(MAX);  
   
 -- Buscar el evento en el catálogo  
 SELECT   
 @IdTipoEvento = IdTipoEvento,  
 @NombreEvento = NombreEvento,  
 @CategoriaEvento = CategoriaEvento,  
 @EstaActivo = EstaActivo,  
 @CamposRequeridos = CamposRequeridos  
 FROM [dbo].[CatalogoTiposEvento]  
 WHERE [CodigoEvento] = @CodigoEvento;  
   
 IF @IdTipoEvento IS NOT NULL AND @EstaActivo = 1  
 BEGIN  
 SET @EsValido = 1;  
 SET @InfoEvento = JSON\_QUERY((  
 SELECT   
 @IdTipoEvento AS idTipoEvento,  
 @CodigoEvento AS codigoEvento,  
 @NombreEvento AS nombreEvento,  
 @CategoriaEvento AS categoria,  
 JSON\_QUERY(@CamposRequeridos) AS camposRequeridos  
 FOR JSON PATH, WITHOUT\_ARRAY\_WRAPPER  
 ));  
 END  
 ELSE  
 BEGIN  
 SET @EsValido = 0;  
 SET @InfoEvento = JSON\_QUERY((  
 SELECT   
 'ERROR' AS status,  
 CASE   
 WHEN @IdTipoEvento IS NULL THEN 'Tipo de evento no encontrado en catálogo'  
 WHEN @EstaActivo = 0 THEN 'Tipo de evento desactivado'  
 ELSE 'Error desconocido'  
 END AS mensaje,  
 @CodigoEvento AS codigoEvento  
 FOR JSON PATH, WITHOUT\_ARRAY\_WRAPPER  
 ));  
 END  
END;  
  
-- Procedimiento para consultar catálogo de eventos por categoría  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ConsultarCatalogoEventos]  
 @CategoriaEvento VARCHAR(50) = NULL,  
 @SistemaResponsable VARCHAR(100) = NULL,  
 @SoloActivos BIT = 1  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 SELECT   
 cte.[IdTipoEvento],  
 cte.[CodigoEvento],  
 cte.[NombreEvento],  
 cte.[DescripcionEvento],  
 cte.[CategoriaEvento],  
 cte.[SistemaResponsable],  
 cte.[NivelCriticidad],  
 cte.[OrdenProceso],  
 cte.[TiempoEsperadoMs],  
 cte.[EsObligatorio],  
 cte.[RequiereValidacion],  
 cte.[EstaActivo],  
 -- Estadísticas de uso (si existen eventos)  
 ISNULL(stats.[TotalEventos], 0) AS TotalEventosRegistrados,  
 ISNULL(stats.[EventosUltimos30Dias], 0) AS EventosUltimos30Dias,  
 ISNULL(stats.[TiempoPromedioMs], 0) AS TiempoPromedioRealMs  
 FROM [dbo].[CatalogoTiposEvento] cte  
 LEFT JOIN (  
 SELECT   
 ea.[TipoEvento],  
 COUNT(\*) AS TotalEventos,  
 COUNT(CASE WHEN ea.[FechaHoraEvento] >= DATEADD(DAY, -30, GETDATE()) THEN 1 END) AS EventosUltimos30Dias,  
 AVG(CAST(ea.[TiempoRespuestaMs] AS FLOAT)) AS TiempoPromedioMs  
 FROM [dbo].[EventosAuditoria] ea  
 WHERE ea.[TiempoRespuestaMs] IS NOT NULL  
 GROUP BY ea.[TipoEvento]  
 ) stats ON cte.[CodigoEvento] = stats.[TipoEvento]  
 WHERE   
 (@CategoriaEvento IS NULL OR cte.[CategoriaEvento] = @CategoriaEvento)  
 AND (@SistemaResponsable IS NULL OR cte.[SistemaResponsable] = @SistemaResponsable)  
 AND (@SoloActivos = 0 OR cte.[EstaActivo] = 1)  
 ORDER BY cte.[CategoriaEvento], cte.[OrdenProceso];  
END;  
  
-- Procedimiento para obtener flujo completo de eventos por solicitud  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ObtenerFlujoCompletoSolicitud]  
 @IdSolicitud BIGINT  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 -- Eventos ejecutados para esta solicitud  
 SELECT   
 'EJECUTADOS' AS TipoSet,  
 ea.[IdEvento],  
 ea.[TipoEvento] AS CodigoEvento,  
 cte.[NombreEvento],  
 cte.[CategoriaEvento],  
 cte.[OrdenProceso] AS OrdenEsperado,  
 ROW\_NUMBER() OVER (ORDER BY ea.[FechaHoraEvento]) AS OrdenReal,  
 ea.[FechaHoraEvento],  
 ea.[EstadoEvento],  
 ea.[TiempoRespuestaMs],  
 cte.[TiempoEsperadoMs],  
 CASE   
 WHEN ea.[TiempoRespuestaMs] > (cte.[TiempoEsperadoMs] \* 2) THEN 'LENTO'  
 WHEN ea.[TiempoRespuestaMs] > cte.[TiempoEsperadoMs] THEN 'DEMORADO'  
 ELSE 'NORMAL'  
 END AS RendimientoTiempo,  
 ea.[SistemaOrigen],  
 ea.[MensajeError]  
 FROM [dbo].[EventosAuditoria] ea  
 INNER JOIN [dbo].[CatalogoTiposEvento] cte ON ea.[TipoEvento] = cte.[CodigoEvento]  
 WHERE ea.[IdSolicitud] = @IdSolicitud  
   
 UNION ALL  
   
 -- Eventos faltantes (obligatorios que no se ejecutaron)  
 SELECT   
 'FALTANTES' AS TipoSet,  
 NULL AS IdEvento,  
 cte.[CodigoEvento],  
 cte.[NombreEvento],  
 cte.[CategoriaEvento],  
 cte.[OrdenProceso] AS OrdenEsperado,  
 NULL AS OrdenReal,  
 NULL AS FechaHoraEvento,  
 'NO\_EJECUTADO' AS EstadoEvento,  
 NULL AS TiempoRespuestaMs,  
 cte.[TiempoEsperadoMs],  
 'FALTANTE' AS RendimientoTiempo,  
 cte.[SistemaResponsable] AS SistemaOrigen,  
 'Evento obligatorio no ejecutado' AS MensajeError  
 FROM [dbo].[CatalogoTiposEvento] cte  
 WHERE cte.[EsObligatorio] = 1   
 AND cte.[EstaActivo] = 1  
 AND NOT EXISTS (  
 SELECT 1 FROM [dbo].[EventosAuditoria] ea   
 WHERE ea.[IdSolicitud] = @IdSolicitud   
 AND ea.[TipoEvento] = cte.[CodigoEvento]  
 )  
   
 ORDER BY TipoSet DESC, OrdenEsperado, OrdenReal;  
END;

#### **Script de Ejemplo: Uso del Catálogo de Eventos**

-- ===================================================================  
-- EJEMPLOS DE USO DEL CATÁLOGO DE EVENTOS  
-- ===================================================================  
  
-- 1. Consultar todos los eventos por categoría  
EXEC [dbo].[sp\_ConsultarCatalogoEventos] @CategoriaEvento = 'BIOMETRICO';  
  
-- 2. Consultar eventos de un sistema específico  
EXEC [dbo].[sp\_ConsultarCatalogoEventos] @SistemaResponsable = 'PARTNER\_BIOMETRICO';  
  
-- 3. Validar si un evento existe en el catálogo  
DECLARE @EsValido BIT, @InfoEvento NVARCHAR(MAX);  
EXEC [dbo].[sp\_ValidarTipoEvento]   
 @CodigoEvento = 'ORIGINATION\_STARTED',  
 @EsValido = @EsValido OUTPUT,  
 @InfoEvento = @InfoEvento OUTPUT;  
  
SELECT @EsValido AS EventoValido, @InfoEvento AS InformacionEvento;  
  
-- 4. Obtener estadísticas del catálogo  
SELECT   
 CategoriaEvento,  
 COUNT(\*) AS TotalEventos,  
 COUNT(CASE WHEN EsObligatorio = 1 THEN 1 END) AS EventosObligatorios,  
 COUNT(CASE WHEN RequiereValidacion = 1 THEN 1 END) AS EventosQueRequierenValidacion,  
 COUNT(CASE WHEN NivelCriticidad = 'CRITICO' THEN 1 END) AS EventosCriticos,  
 AVG(CAST(TiempoEsperadoMs AS FLOAT)) AS TiempoPromedioEsperadoMs,  
 MIN(OrdenProceso) AS PrimerOrden,  
 MAX(OrdenProceso) AS UltimoOrden  
FROM [dbo].[CatalogoTiposEvento]  
WHERE EstaActivo = 1  
GROUP BY CategoriaEvento  
ORDER BY MIN(OrdenProceso);  
  
-- 5. Consultar eventos con tiempos de respuesta más lentos que lo esperado  
SELECT   
 cte.CodigoEvento,  
 cte.NombreEvento,  
 cte.TiempoEsperadoMs,  
 AVG(CAST(ea.TiempoRespuestaMs AS FLOAT)) AS TiempoPromedioReal,  
 COUNT(\*) AS TotalEjecuciones,  
 CAST((AVG(CAST(ea.TiempoRespuestaMs AS FLOAT)) / cte.TiempoEsperadoMs \* 100) AS DECIMAL(5,1)) AS PorcentajeDelEsperado  
FROM [dbo].[CatalogoTiposEvento] cte  
INNER JOIN [dbo].[EventosAuditoria] ea ON cte.CodigoEvento = ea.TipoEvento  
WHERE ea.TiempoRespuestaMs IS NOT NULL  
 AND cte.TiempoEsperadoMs IS NOT NULL  
 AND ea.FechaHoraEvento >= DATEADD(DAY, -30, GETDATE())  
GROUP BY cte.CodigoEvento, cte.NombreEvento, cte.TiempoEsperadoMs  
HAVING AVG(CAST(ea.TiempoRespuestaMs AS FLOAT)) > cte.TiempoEsperadoMs  
ORDER BY PorcentajeDelEsperado DESC;  
  
-- 6. Obtener el flujo completo de una solicitud específica  
-- (Cambiar 123456789 por un ID de solicitud real)  
-- EXEC [dbo].[sp\_ObtenerFlujoCompletoSolicitud] @IdSolicitud = 123456789;

## Stored Procedures Principales  
  
### \*\*SP para Insertar Eventos de Auditoría\*\*  
```sql  
-- Procedimiento principal para registrar eventos de auditoría con validaciones  
CREATE OR ALTER PROCEDURE [dbo].[sp\_InsertarEventoAuditoria]  
 @IdSolicitud BIGINT,  
 @IdSocio INT,  
 @TipoEvento VARCHAR(100),  
 @FechaHoraEvento DATETIME2(3),  
 @IdCorrelacion VARCHAR(100),  
 @SistemaOrigen VARCHAR(100),  
 @PayloadEvento NVARCHAR(MAX) = NULL,  
 @IdSesion VARCHAR(100) = NULL,  
 @TiempoRespuestaMs INT = NULL,  
 @CodigoHTTP VARCHAR(10) = NULL  
AS  
BEGIN  
 SET NOCOUNT ON;  
 SET XACT\_ABORT ON;  
   
 DECLARE @IdEvento BIGINT;  
   
 DECLARE @EsEventoValido BIT;  
 DECLARE @InfoEvento NVARCHAR(MAX);  
   
 BEGIN TRY  
 BEGIN TRANSACTION;  
   
 -- Validar tipo de evento contra el catálogo  
 EXEC [dbo].[sp\_ValidarTipoEvento]   
 @CodigoEvento = @TipoEvento,  
 @EsValido = @EsEventoValido OUTPUT,  
 @InfoEvento = @InfoEvento OUTPUT;  
   
 IF @EsEventoValido = 0  
 BEGIN  
 RAISERROR('Tipo de evento no válido: %s. Info: %s', 16, 1, @TipoEvento, @InfoEvento);  
 RETURN;  
 END  
   
 -- Validar que la solicitud existe  
 IF NOT EXISTS (SELECT 1 FROM [dbo].[SolicitudesOriginacion] WHERE [IdSolicitud] = @IdSolicitud)  
 BEGIN  
 RAISERROR('La solicitud %d no existe', 16, 1, @IdSolicitud);  
 RETURN;  
 END  
   
 -- Validar que el socio existe y está activo  
 IF NOT EXISTS (SELECT 1 FROM [dbo].[SociosComerciales] WHERE [IdSocio] = @IdSocio AND [EstaActivo] = 1)  
 BEGIN  
 RAISERROR('El socio comercial %d no existe o no está activo', 16, 1, @IdSocio);  
 RETURN;  
 END  
   
 -- Insertar evento de auditoría  
 INSERT INTO [dbo].[EventosAuditoria] (  
 [TipoEvento], [IdSolicitud], [IdSocio], [FechaHoraEvento],  
 [IdCorrelacion], [SistemaOrigen], [PayloadEvento], [IdSesion],  
 [TiempoRespuestaMs], [CodigoHTTP]  
 ) VALUES (  
 @TipoEvento, @IdSolicitud, @IdSocio, @FechaHoraEvento,  
 @IdCorrelacion, @SistemaOrigen, @PayloadEvento, @IdSesion,  
 @TiempoRespuestaMs, @CodigoHTTP  
 );  
   
 SET @IdEvento = SCOPE\_IDENTITY();  
   
 -- Actualizar última actividad de la solicitud  
 UPDATE [dbo].[SolicitudesOriginacion]   
 SET [UltimaActividad] = @FechaHoraEvento  
 WHERE [IdSolicitud] = @IdSolicitud;  
   
 COMMIT TRANSACTION;  
   
 -- Retornar ID del evento creado  
 SELECT @IdEvento AS IdEventoCreado;  
   
 END TRY  
 BEGIN CATCH  
 IF @@TRANCOUNT > 0  
 ROLLBACK TRANSACTION;  
   
 -- Re-lanzar el error  
 DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  
 DECLARE @ErrorSeverity INT = ERROR\_SEVERITY();  
 DECLARE @ErrorState INT = ERROR\_STATE();  
   
 RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);  
 END CATCH;  
END;

### **SP para Gestionar Configuraciones del Sistema**

-- Procedimiento para obtener configuración del sistema  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ObtenerConfiguracion]  
 @ClaveConfiguracion VARCHAR(100)  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 SELECT   
 [ClaveConfiguracion],  
 [ValorConfiguracion],  
 [TipoValor],  
 [Descripcion],  
 [FechaModificacion],  
 [VersionConfiguracion]  
 FROM [dbo].[ConfiguracionSistema]  
 WHERE [ClaveConfiguracion] = @ClaveConfiguracion;  
END;  
  
-- Procedimiento para actualizar configuración del sistema  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ActualizarConfiguracion]  
 @ClaveConfiguracion VARCHAR(100),  
 @NuevoValor NVARCHAR(MAX),  
 @ModificadoPor VARCHAR(100)  
AS  
BEGIN  
 SET NOCOUNT ON;  
 SET XACT\_ABORT ON;  
   
 BEGIN TRY  
 BEGIN TRANSACTION;  
   
 -- Verificar si la configuración existe y es editable  
 IF NOT EXISTS (  
 SELECT 1 FROM [dbo].[ConfiguracionSistema]   
 WHERE [ClaveConfiguracion] = @ClaveConfiguracion AND [EsEditable] = 1  
 )  
 BEGIN  
 RAISERROR('La configuración %s no existe o no es editable', 16, 1, @ClaveConfiguracion);  
 RETURN;  
 END  
   
 -- Actualizar configuración  
 UPDATE [dbo].[ConfiguracionSistema]  
 SET   
 [ValorConfiguracion] = @NuevoValor,  
 [FechaModificacion] = SYSUTCDATETIME(),  
 [ModificadoPor] = @ModificadoPor,  
 [VersionConfiguracion] = [VersionConfiguracion] + 1  
 WHERE [ClaveConfiguracion] = @ClaveConfiguracion;  
   
 -- Registrar cambio en log del sistema  
 INSERT INTO [dbo].[LogSistema]   
 ([NivelLog], [ComponenteSistema], [Mensaje])  
 VALUES   
 ('INFO', 'CONFIGURACION',   
 'Configuración actualizada: ' + @ClaveConfiguracion + ' por ' + @ModificadoPor);  
   
 COMMIT TRANSACTION;  
   
 SELECT 'Configuración actualizada exitosamente' AS Resultado;  
   
 END TRY  
 BEGIN CATCH  
 IF @@TRANCOUNT > 0  
 ROLLBACK TRANSACTION;  
   
 DECLARE @ErrorMsg NVARCHAR(4000) = ERROR\_MESSAGE();  
   
 -- Registrar error  
 INSERT INTO [dbo].[LogSistema]   
 ([NivelLog], [ComponenteSistema], [Mensaje], [DetallesError])  
 VALUES   
 ('ERROR', 'CONFIGURACION', 'Error al actualizar configuración: ' + @ClaveConfiguracion, @ErrorMsg);  
   
 RAISERROR(@ErrorMsg, ERROR\_SEVERITY(), ERROR\_STATE());  
 END CATCH  
END;

### **SP para Consultar Timeline de Solicitud**

-- Procedimiento para obtener la cronología completa de eventos de una solicitud  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ConsultarTimelineSolicitud]  
 @IdSolicitud BIGINT  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 -- Validar que la solicitud existe  
 IF NOT EXISTS (SELECT 1 FROM [dbo].[SolicitudesOriginacion] WHERE [IdSolicitud] = @IdSolicitud)  
 BEGIN  
 RAISERROR('La solicitud %d no existe', 16, 1, @IdSolicitud);  
 RETURN;  
 END  
   
 -- Retornar timeline ordenado cronológicamente  
 SELECT   
 ea.[IdEvento],  
 ea.[TipoEvento],  
 ea.[FechaHoraEvento],  
 ea.[EstadoEvento],  
 ea.[TiempoRespuestaMs],  
 ea.[IdCorrelacion],  
 ea.[SistemaOrigen],  
 ea.[PayloadEvento],  
 LAG(ea.[FechaHoraEvento]) OVER (ORDER BY ea.[FechaHoraEvento]) AS EventoAnterior,  
 DATEDIFF(SECOND,   
 LAG(ea.[FechaHoraEvento]) OVER (ORDER BY ea.[FechaHoraEvento]),   
 ea.[FechaHoraEvento]  
 ) AS TiempoEntreEventosSegundos,  
 ROW\_NUMBER() OVER (ORDER BY ea.[FechaHoraEvento]) AS NumeroSecuencia  
 FROM [dbo].[EventosAuditoria] ea  
 WHERE ea.[IdSolicitud] = @IdSolicitud  
 ORDER BY ea.[FechaHoraEvento] ASC;  
END;

### **SP para Consultar Métricas por Socio**

-- Procedimiento para obtener métricas operacionales de un socio comercial  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ConsultarMetricasSocio]  
 @IdSocio INT,  
 @FechaInicio DATETIME2(3),  
 @FechaFin DATETIME2(3)  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 -- Validar que el socio existe  
 IF NOT EXISTS (SELECT 1 FROM [dbo].[SociosComerciales] WHERE [IdSocio] = @IdSocio)  
 BEGIN  
 RAISERROR('El socio comercial %d no existe', 16, 1, @IdSocio);  
 RETURN;  
 END  
   
 -- Métricas de solicitudes  
 SELECT   
 sc.[CodigoSocio],  
 sc.[NombreSocio],  
 COUNT(so.[IdSolicitud]) AS TotalSolicitudes,  
 COUNT(CASE WHEN so.[EstaCompleto] = 1 THEN 1 END) AS SolicitudesCompletadas,  
 COUNT(CASE WHEN so.[EstadoActual] = 'ABANDONADO' THEN 1 END) AS SolicitudesAbandonadas,  
 COUNT(CASE WHEN so.[EstadoActual] = 'APROBADO' THEN 1 END) AS SolicitudesAprobadas,  
 COUNT(CASE WHEN so.[EstadoActual] = 'RECHAZADO' THEN 1 END) AS SolicitudesRechazadas,  
 AVG(CAST(so.[TiempoProcesoSegundos] AS FLOAT)) AS TiempoPromedioSegundos,  
 MIN(so.[FechaInicio]) AS PrimeraSolicitud,  
 MAX(so.[UltimaActividad]) AS UltimaActividad  
 FROM [dbo].[SociosComerciales] sc  
 LEFT JOIN [dbo].[SolicitudesOriginacion] so ON sc.[IdSocio] = so.[IdSocio]  
 AND so.[FechaInicio] >= @FechaInicio   
 AND so.[FechaInicio] <= @FechaFin  
 WHERE sc.[IdSocio] = @IdSocio  
 GROUP BY sc.[CodigoSocio], sc.[NombreSocio];  
   
 -- Métricas de eventos  
 SELECT   
 ea.[TipoEvento],  
 COUNT(\*) AS TotalEventos,  
 COUNT(CASE WHEN ea.[EstadoEvento] = 'EXITOSO' THEN 1 END) AS EventosExitosos,  
 COUNT(CASE WHEN ea.[EstadoEvento] = 'FALLIDO' THEN 1 END) AS EventosFallidos,  
 AVG(CAST(ea.[TiempoRespuestaMs] AS FLOAT)) AS TiempoRespuestaPromedio  
 FROM [dbo].[EventosAuditoria] ea  
 WHERE ea.[IdSocio] = @IdSocio  
 AND ea.[FechaHoraEvento] >= @FechaInicio   
 AND ea.[FechaHoraEvento] <= @FechaFin  
 GROUP BY ea.[TipoEvento]  
 ORDER BY COUNT(\*) DESC;  
END;

### **SP para Búsqueda Avanzada de Eventos**

-- Procedimiento para búsquedas flexibles de eventos con múltiples filtros  
CREATE OR ALTER PROCEDURE [dbo].[sp\_BuscarEventosAuditoria]  
 @TipoEvento VARCHAR(100) = NULL,  
 @IdSocio INT = NULL,  
 @IdSolicitud BIGINT = NULL,  
 @IdCorrelacion VARCHAR(100) = NULL,  
 @EstadoEvento VARCHAR(20) = NULL,  
 @FechaInicio DATETIME2(3) = NULL,  
 @FechaFin DATETIME2(3) = NULL,  
 @Top INT = 1000  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 -- Validar parámetros  
 IF @Top > 10000 SET @Top = 10000; -- Límite máximo de seguridad  
   
 SELECT TOP (@Top)  
 ea.[IdEvento],  
 ea.[TipoEvento],  
 ea.[IdSolicitud],  
 so.[CodigoSolicitud],  
 ea.[IdSocio],  
 sc.[CodigoSocio],  
 sc.[NombreSocio],  
 ea.[FechaHoraEvento],  
 ea.[IdCorrelacion],  
 ea.[EstadoEvento],  
 ea.[TiempoRespuestaMs],  
 ea.[SistemaOrigen],  
 ea.[MensajeError]  
 FROM [dbo].[EventosAuditoria] ea  
 INNER JOIN [dbo].[SolicitudesOriginacion] so ON ea.[IdSolicitud] = so.[IdSolicitud]  
 INNER JOIN [dbo].[SociosComerciales] sc ON ea.[IdSocio] = sc.[IdSocio]  
 WHERE   
 (@TipoEvento IS NULL OR ea.[TipoEvento] = @TipoEvento)  
 AND (@IdSocio IS NULL OR ea.[IdSocio] = @IdSocio)  
 AND (@IdSolicitud IS NULL OR ea.[IdSolicitud] = @IdSolicitud)  
 AND (@IdCorrelacion IS NULL OR ea.[IdCorrelacion] = @IdCorrelacion)  
 AND (@EstadoEvento IS NULL OR ea.[EstadoEvento] = @EstadoEvento)  
 AND (@FechaInicio IS NULL OR ea.[FechaHoraEvento] >= @FechaInicio)  
 AND (@FechaFin IS NULL OR ea.[FechaHoraEvento] <= @FechaFin)  
 ORDER BY ea.[FechaHoraEvento] DESC;  
END;

### **SP para Análisis de Logs y Rendimiento del Sistema**

-- Procedimiento para consultar logs del sistema con filtros  
CREATE OR ALTER PROCEDURE [dbo].[sp\_ConsultarLogsSistema]  
 @FechaInicio DATETIME2(3) = NULL,  
 @FechaFin DATETIME2(3) = NULL,  
 @NivelLog VARCHAR(20) = NULL,  
 @ComponenteSistema VARCHAR(100) = NULL,  
 @IdSolicitud BIGINT = NULL,  
 @MaxRegistros INT = 1000  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 -- Establecer fechas por defecto (últimas 24 horas si no se especifica)  
 IF @FechaInicio IS NULL  
 SET @FechaInicio = DATEADD(DAY, -1, SYSUTCDATETIME());  
   
 IF @FechaFin IS NULL  
 SET @FechaFin = SYSUTCDATETIME();  
   
 SELECT TOP (@MaxRegistros)  
 l.[IdLog],  
 l.[NivelLog],  
 l.[FechaHora],  
 l.[ComponenteSistema],  
 l.[Mensaje],  
 l.[DetallesError],  
 l.[IdEvento],  
 l.[IdSolicitud],  
 l.[DireccionIP],  
 l.[DuracionOperacionMs],  
 -- Información adicional de contexto  
 CASE   
 WHEN l.[IdSolicitud] IS NOT NULL THEN s.[CodigoSolicitud]  
 ELSE NULL   
 END AS CodigoSolicitud,  
 CASE   
 WHEN l.[IdEvento] IS NOT NULL THEN ea.[TipoEvento]  
 ELSE NULL   
 END AS TipoEventoRelacionado  
 FROM [dbo].[LogSistema] l  
 LEFT JOIN [dbo].[SolicitudesOriginacion] s ON l.[IdSolicitud] = s.[IdSolicitud]  
 LEFT JOIN [dbo].[EventosAuditoria] ea ON l.[IdEvento] = ea.[IdEvento]  
 WHERE   
 l.[FechaHora] BETWEEN @FechaInicio AND @FechaFin  
 AND (@NivelLog IS NULL OR l.[NivelLog] = @NivelLog)  
 AND (@ComponenteSistema IS NULL OR l.[ComponenteSistema] = @ComponenteSistema)  
 AND (@IdSolicitud IS NULL OR l.[IdSolicitud] = @IdSolicitud)  
 ORDER BY l.[FechaHora] DESC;  
END;  
  
-- Procedimiento para estadísticas de rendimiento del sistema  
CREATE OR ALTER PROCEDURE [dbo].[sp\_EstadisticasRendimientoSistema]  
 @FechaInicio DATETIME2(3) = NULL,  
 @FechaFin DATETIME2(3) = NULL  
AS  
BEGIN  
 SET NOCOUNT ON;  
   
 -- Establecer fechas por defecto (últimas 24 horas)  
 IF @FechaInicio IS NULL  
 SET @FechaInicio = DATEADD(DAY, -1, SYSUTCDATETIME());  
   
 IF @FechaFin IS NULL  
 SET @FechaFin = SYSUTCDATETIME();  
   
 -- Estadísticas por componente del sistema  
 SELECT   
 'ESTADISTICAS\_POR\_COMPONENTE' AS TipoMetrica,  
 l.[ComponenteSistema],  
 COUNT(\*) AS TotalOperaciones,  
 SUM(CASE WHEN l.[NivelLog] = 'ERROR' THEN 1 ELSE 0 END) AS TotalErrores,  
 CAST(AVG(CAST(l.[DuracionOperacionMs] AS FLOAT)) AS DECIMAL(10,2)) AS DuracionPromedioMs,  
 MAX(l.[DuracionOperacionMs]) AS DuracionMaximaMs,  
 MIN(l.[DuracionOperacionMs]) AS DuracionMinimaMs,  
 CAST((100.0 \* SUM(CASE WHEN l.[NivelLog] != 'ERROR' THEN 1 ELSE 0 END) / COUNT(\*)) AS DECIMAL(5,2)) AS PorcentajeExito  
 FROM [dbo].[LogSistema] l  
 WHERE   
 l.[FechaHora] BETWEEN @FechaInicio AND @FechaFin  
 AND l.[DuracionOperacionMs] IS NOT NULL  
 GROUP BY l.[ComponenteSistema]  
   
 UNION ALL  
   
 -- Estadísticas de eventos por tipo  
 SELECT   
 'EVENTOS\_POR\_TIPO' AS TipoMetrica,  
 ea.[TipoEvento] AS ComponenteSistema,  
 COUNT(\*) AS TotalOperaciones,  
 SUM(CASE WHEN ea.[EstadoEvento] = 'FALLIDO' THEN 1 ELSE 0 END) AS TotalErrores,  
 CAST(AVG(CAST(ea.[TiempoRespuestaMs] AS FLOAT)) AS DECIMAL(10,2)) AS DuracionPromedioMs,  
 MAX(ea.[TiempoRespuestaMs]) AS DuracionMaximaMs,  
 MIN(ea.[TiempoRespuestaMs]) AS DuracionMinimaMs,  
 CAST((100.0 \* SUM(CASE WHEN ea.[EstadoEvento] = 'EXITOSO' THEN 1 ELSE 0 END) / COUNT(\*)) AS DECIMAL(5,2)) AS PorcentajeExito  
 FROM [dbo].[EventosAuditoria] ea  
 WHERE   
 ea.[FechaHoraEvento] BETWEEN @FechaInicio AND @FechaFin  
 AND ea.[TiempoRespuestaMs] IS NOT NULL  
 GROUP BY ea.[TipoEvento]  
   
 ORDER BY TipoMetrica, TotalOperaciones DESC;  
END;  
  
-- Procedimiento para limpieza automática de logs antiguos  
CREATE OR ALTER PROCEDURE [dbo].[sp\_LimpiezaLogsAutomatica]  
AS  
BEGIN  
 SET NOCOUNT ON;  
 SET XACT\_ABORT ON;  
   
 DECLARE @FechaLimite DATETIME2(3);  
 DECLARE @DiasRetencion INT;  
 DECLARE @RegistrosEliminados INT = 0;  
   
 BEGIN TRY  
 -- Obtener días de retención de la configuración  
 SELECT @DiasRetencion = CAST([ValorConfiguracion] AS INT)  
 FROM [dbo].[ConfiguracionSistema]  
 WHERE [ClaveConfiguracion] = 'TIEMPO\_RETENCION\_EVENTOS\_DIAS';  
   
 IF @DiasRetencion IS NULL  
 SET @DiasRetencion = 2555; -- 7 años por defecto  
   
 SET @FechaLimite = DATEADD(DAY, -@DiasRetencion, SYSUTCDATETIME());  
   
 BEGIN TRANSACTION;  
   
 -- Eliminar logs del sistema antiguos (mantener solo errores críticos por más tiempo)  
 DELETE FROM [dbo].[LogSistema]  
 WHERE [FechaHora] < @FechaLimite   
 AND [NivelLog] NOT IN ('CRITICAL', 'ERROR');  
   
 SET @RegistrosEliminados = @@ROWCOUNT;  
   
 -- Registrar la operación de limpieza  
 INSERT INTO [dbo].[LogSistema]   
 ([NivelLog], [ComponenteSistema], [Mensaje])  
 VALUES   
 ('INFO', 'MANTENIMIENTO',   
 'Limpieza automática completada. Registros eliminados: ' + CAST(@RegistrosEliminados AS VARCHAR(10)));  
   
 COMMIT TRANSACTION;  
   
 SELECT @RegistrosEliminados AS RegistrosEliminados, @FechaLimite AS FechaLimiteUtilizada;  
   
 END TRY  
 BEGIN CATCH  
 IF @@TRANCOUNT > 0  
 ROLLBACK TRANSACTION;  
   
 DECLARE @ErrorMsg NVARCHAR(4000) = ERROR\_MESSAGE();  
   
 INSERT INTO [dbo].[LogSistema]   
 ([NivelLog], [ComponenteSistema], [Mensaje], [DetallesError])  
 VALUES   
 ('ERROR', 'MANTENIMIENTO', 'Error en limpieza automática', @ErrorMsg);  
   
 RAISERROR(@ErrorMsg, ERROR\_SEVERITY(), ERROR\_STATE());  
 END CATCH  
END;

# 6. SCRIPTS DE INICIALIZACIÓN Y CONFIGURACIÓN

## Script de Creación de Base de Datos Completa

-- Script completo de inicialización de la base de datos BradesCard Auditoría  
-- Ejecutar en SQL Server 2019/2022 en datacenter Bradescard  
  
USE master;  
GO  
  
-- Crear la base de datos  
IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = 'BradescardAuditoria')  
BEGIN  
 CREATE DATABASE [BradescardAuditoria]  
 ON (  
 NAME = 'BradescardAuditoria\_Data',  
 FILENAME = 'C:\DatabaseFiles\BradescardAuditoria.mdf',  
 SIZE = 1GB,  
 MAXSIZE = 100GB,  
 FILEGROWTH = 256MB  
 )  
 LOG ON (  
 NAME = 'BradescardAuditoria\_Log',  
 FILENAME = 'C:\DatabaseFiles\BradescardAuditoria.ldf',  
 SIZE = 256MB,  
 MAXSIZE = 10GB,  
 FILEGROWTH = 64MB  
 );  
END;  
GO  
  
USE [BradescardAuditoria];  
GO  
  
-- Habilitar Always Encrypted para datos sensibles  
ALTER DATABASE [BradescardAuditoria] SET ENCRYPTION ON;  
GO  
  
-- Crear esquemas de organización  
CREATE SCHEMA [auditoria] AUTHORIZATION [dbo];  
CREATE SCHEMA [configuracion] AUTHORIZATION [dbo];  
CREATE SCHEMA [reportes] AUTHORIZATION [dbo];  
GO  
  
-- Crear función para generar códigos únicos  
CREATE FUNCTION [dbo].[fn\_GenerarCodigoSolicitud]()  
RETURNS VARCHAR(20)  
AS  
BEGIN  
 DECLARE @Codigo VARCHAR(20);  
 DECLARE @Timestamp VARCHAR(10) = FORMAT(GETDATE(), 'yyyyMMdd');  
 DECLARE @Random VARCHAR(6) = RIGHT('000000' + CAST(ABS(CHECKSUM(NEWID())) % 1000000 AS VARCHAR(6)), 6);  
   
 SET @Codigo = 'BC-' + @Timestamp + '-' + @Random;  
   
 RETURN @Codigo;  
END;  
GO  
  
-- Crear trigger para generar códigos automáticamente  
CREATE TRIGGER [trg\_SolicitudesOriginacion\_GenerarCodigo]  
ON [dbo].[SolicitudesOriginacion]  
INSTEAD OF INSERT  
AS  
BEGIN  
 INSERT INTO [dbo].[SolicitudesOriginacion] (  
 [CodigoSolicitud], [IdSocio], [FechaInicio], [EstadoActual], [TiempoProcesoSegundos]  
 )  
 SELECT   
 [dbo].[fn\_GenerarCodigoSolicitud](),  
 i.[IdSocio],  
 i.[FechaInicio],  
 i.[EstadoActual],  
 i.[TiempoProcesoSegundos]  
 FROM inserted i;  
END;  
GO  
  
-- Jobs de mantenimiento automático  
-- Job para limpieza de logs cada domingo a las 2 AM  
IF NOT EXISTS (SELECT job\_id FROM msdb.dbo.sysjobs WHERE name = 'BradescardAuditoria\_LimpiezaLogs')  
BEGIN  
 EXEC msdb.dbo.sp\_add\_job  
 @job\_name = 'BradescardAuditoria\_LimpiezaLogs',  
 @enabled = 1,  
 @description = 'Limpieza automática de logs antiguos del sistema de auditoría',  
 @category\_name = 'Database Maintenance';  
   
 EXEC msdb.dbo.sp\_add\_jobstep  
 @job\_name = 'BradescardAuditoria\_LimpiezaLogs',  
 @step\_name = 'Ejecutar\_Limpieza',  
 @command = 'EXEC [BradescardAuditoria].[dbo].[sp\_LimpiezaLogsAutomatica]',  
 @database\_name = 'BradescardAuditoria';  
   
 EXEC msdb.dbo.sp\_add\_schedule  
 @schedule\_name = 'Semanal\_Domingo\_2AM',  
 @freq\_type = 8, -- Weekly  
 @freq\_interval = 1, -- Sunday  
 @freq\_recurrence\_factor = 1,  
 @active\_start\_time = 020000; -- 2:00 AM  
   
 EXEC msdb.dbo.sp\_attach\_schedule  
 @job\_name = 'BradescardAuditoria\_LimpiezaLogs',  
 @schedule\_name = 'Semanal\_Domingo\_2AM';  
   
 EXEC msdb.dbo.sp\_add\_jobserver  
 @job\_name = 'BradescardAuditoria\_LimpiezaLogs';  
END;  
GO  
  
PRINT 'Base de datos BradescardAuditoria inicializada correctamente';  
PRINT 'Tablas creadas: SociosComerciales, SolicitudesOriginacion, EventosAuditoria, ConfiguracionSistema, LogSistema';  
PRINT 'Stored Procedures creados: 8 procedimientos principales';  
PRINT 'Jobs de mantenimiento: Configurados para ejecución automática';

## Ejemplos de Uso Completos

### Ejemplo 1: Flujo Completo de Originación con Auditoría

// C# - Ejemplo de implementación del cliente API  
using System;  
using System.Net.Http;  
using System.Text.Json;  
using System.Threading.Tasks;  
  
public class BradescardAuditoriaClient  
{  
 private readonly HttpClient \_httpClient;  
 private readonly string \_apiKey;  
 private readonly int \_socioId;  
   
 public BradescardAuditoriaClient(string baseUrl, string apiKey, int socioId)  
 {  
 \_httpClient = new HttpClient { BaseAddress = new Uri(baseUrl) };  
 \_apiKey = apiKey;  
 \_socioId = socioId;  
   
 \_httpClient.DefaultRequestHeaders.Add("Authorization", $"Bearer {apiKey}");  
 \_httpClient.DefaultRequestHeaders.Add("X-Socio-ID", socioId.ToString());  
 }  
   
 // Registrar evento de auditoría  
 public async Task<long?> RegistrarEventoAsync(EventoAuditoriaRequest evento)  
 {  
 var json = JsonSerializer.Serialize(evento);  
 var content = new StringContent(json, System.Text.Encoding.UTF8, "application/json");  
   
 var response = await \_httpClient.PostAsync("/api/v1/auditoria/eventos", content);  
 response.EnsureSuccessStatusCode();  
   
 var result = await JsonSerializer.DeserializeAsync<EventoResponse>(  
 await response.Content.ReadAsStreamAsync());  
   
 return result.IdEventoCreado;  
 }  
   
 // Obtener timeline completo de una solicitud  
 public async Task<TimelineResponse> ObtenerTimelineAsync(long idSolicitud)  
 {  
 var response = await \_httpClient.GetAsync($"/api/v1/auditoria/solicitudes/{idSolicitud}/timeline");  
 response.EnsureSuccessStatusCode();  
   
 return await JsonSerializer.DeserializeAsync<TimelineResponse>(  
 await response.Content.ReadAsStreamAsync());  
 }  
   
 // Ejemplo de uso del cliente  
 public async Task EjemploFlujoBuroCredito(long idSolicitud)  
 {  
 try  
 {  
 // 1. Solicitud iniciada  
 await RegistrarEventoAsync(new EventoAuditoriaRequest  
 {  
 TipoEvento = "SOLICITUD\_INICIADA",  
 IdSolicitud = idSolicitud,  
 FechaHoraEvento = DateTime.UtcNow,  
 IdCorrelacion = $"corr\_inicio\_{idSolicitud}",  
 SistemaOrigen = "SISTEMA\_ORIGINACION\_RETAIL",  
 PayloadEvento = JsonSerializer.Serialize(new { canal = "web", dispositivo = "desktop" })  
 });  
   
 // 2. Validación de datos  
 await RegistrarEventoAsync(new EventoAuditoriaRequest  
 {  
 TipoEvento = "DATOS\_VALIDADOS",  
 IdSolicitud = idSolicitud,  
 FechaHoraEvento = DateTime.UtcNow,  
 IdCorrelacion = $"corr\_validacion\_{idSolicitud}",  
 SistemaOrigen = "MOTOR\_VALIDACIONES",  
 PayloadEvento = JsonSerializer.Serialize(new {   
 camposValidados = new[] { "CURP", "RFC", "telefono", "email" },  
 resultadoValidacion = "EXITOSO"  
 })  
 });  
   
 // 3. Consulta a Buró de Crédito  
 var inicioBuro = DateTime.UtcNow;  
 // Simular llamada a Buró...  
 await Task.Delay(500); // Simular latencia  
 var finBuro = DateTime.UtcNow;  
   
 await RegistrarEventoAsync(new EventoAuditoriaRequest  
 {  
 TipoEvento = "CONSULTA\_BURO\_COMPLETADA",  
 IdSolicitud = idSolicitud,  
 FechaHoraEvento = finBuro,  
 IdCorrelacion = $"corr\_buro\_{idSolicitud}",  
 SistemaOrigen = "BURO\_CREDITO\_ADAPTER",  
 TiempoRespuestaMs = (int)(finBuro - inicioBuro).TotalMilliseconds,  
 PayloadEvento = JsonSerializer.Serialize(new {  
 bureauProvider = "CIRCULO\_CREDITO",  
 score = 720,  
 segmento = "BAJO\_RIESGO"  
 })  
 });  
   
 // 4. Decisión final  
 await RegistrarEventoAsync(new EventoAuditoriaRequest  
 {  
 TipoEvento = "DECISION\_TOMADA",  
 IdSolicitud = idSolicitud,  
 FechaHoraEvento = DateTime.UtcNow,  
 IdCorrelacion = $"corr\_decision\_{idSolicitud}",  
 SistemaOrigen = "MOTOR\_DECISIONES",  
 PayloadEvento = JsonSerializer.Serialize(new {  
 decision = "APROBADO",  
 lineaCredito = 50000,  
 motivoDecision = "PERFIL\_CREDITICIO\_FAVORABLE"  
 })  
 });  
   
 Console.WriteLine($"Flujo de originación completado para solicitud {idSolicitud}");  
   
 }  
 catch (Exception ex)  
 {  
 // Registrar error en auditoría  
 await RegistrarEventoAsync(new EventoAuditoriaRequest  
 {  
 TipoEvento = "ERROR\_PROCESAMIENTO",  
 IdSolicitud = idSolicitud,  
 FechaHoraEvento = DateTime.UtcNow,  
 IdCorrelacion = $"corr\_error\_{idSolicitud}",  
 SistemaOrigen = "SISTEMA\_ORIGINACION",  
 PayloadEvento = JsonSerializer.Serialize(new { error = ex.Message })  
 });  
   
 throw;  
 }  
 }  
}  
  
// Modelos de datos  
public class EventoAuditoriaRequest  
{  
 public string TipoEvento { get; set; }  
 public long IdSolicitud { get; set; }  
 public DateTime FechaHoraEvento { get; set; }  
 public string IdCorrelacion { get; set; }  
 public string SistemaOrigen { get; set; }  
 public string PayloadEvento { get; set; }  
 public string IdSesion { get; set; }  
 public int? TiempoRespuestaMs { get; set; }  
}  
  
public class EventoResponse  
{  
 public long IdEventoCreado { get; set; }  
}  
  
public class TimelineResponse  
{  
 public long IdSolicitud { get; set; }  
 public string CodigoSolicitud { get; set; }  
 public List<EventoTimeline> Eventos { get; set; }  
}  
  
public class EventoTimeline  
{  
 public string TipoEvento { get; set; }  
 public DateTime FechaHora { get; set; }  
 public string Estado { get; set; }  
 public int? TiempoRespuesta { get; set; }  
}

## Configuraciones de Compliance Financiero

### 1. **Regulaciones Mexicanas - CNBV**

La solución implementa compliance automático con las principales regulaciones:

* **Artículo 115 CNBV**: Expediente completo del cliente
* **Artículo 116 CNBV**: Identificación oficial obligatoria
* **Artículo 117 CNBV**: Verificación de capacidad jurídica
* **Ley PLD**: Validaciones Anti-Lavado de Dinero automáticas

### 2. **Encriptación y Seguridad de Datos**

{  
 "dataEncryption": {  
 "alwaysEncrypted": {  
 "enabled": true,  
 "keyVaultUrl": "https://bradescard-kv.vault.azure.net/",  
 "encryptedColumns": [  
 "CustomerData.CURP", "CustomerData.RFC",   
 "CustomerData.PhoneNumber", "CustomerData.Email",  
 "CustomerData.FullName", "CustomerData.DateOfBirth"  
 ]  
 }  
 },  
 "dataRetention": {  
 "auditEvents": "7 years",   
 "customerData": "10 years",  
 "complianceLog": "15 years"  
 }  
}

# 4. IMPLEMENTACIÓN TÉCNICA

## Stack Tecnológico Simplificado

### Backend (.NET 8)

* **API Framework**: ASP.NET Core 8 con Web API controllers
* **ORM**: Entity Framework Core con Always Encrypted
* **Queue Processing**: Azure Storage SDK para colas
* **Background Jobs**: Hosted Services para procesamiento asíncrono
* **Logging**: ILogger nativo + Application Insights

### Seguridad Esencial

* **Autenticación**: API Keys + certificados SSL por partner
* **Autorización**: Middleware customizado por partner
* **Encriptación**: Always Encrypted para datos PII en SQL Server
* **Secrets**: Azure Key Vault estándar
* **Network**: ExpressRoute para tráfico privado a SQL Server

## API Endpoints Core

### **1. Endpoint Principal - Registro de Eventos de Auditoría**

POST /api/v1/auditoria/eventos  
Content-Type: application/json  
Authorization: Bearer {api\_key}  
X-Socio-ID: {codigo\_socio}  
  
Body Example:  
{  
 "tipoEvento": "CONSULTA\_BURO\_SOLICITADA",  
 "idSolicitud": 123456789,  
 "fechaHoraEvento": "2024-12-26T10:30:45.123Z",  
 "idCorrelacion": "corr\_app\_98765432101",  
 "sistemaOrigen": "SISTEMA\_ORIGINACION\_RETAIL",  
 "payloadEvento": { "bureauProvider": "CIRCULO\_CREDITO", "tipoConsulta": "REPORTE\_COMPLETO" },  
 "idSesion": "session\_abc123def456",  
 "tiempoRespuestaMs": 250  
}

### **2. Endpoints de Consulta por Solicitud**

GET /api/v1/auditoria/solicitudes/{idSolicitud}/eventos  
GET /api/v1/auditoria/solicitudes/{idSolicitud}/timeline  
GET /api/v1/auditoria/solicitudes/{idSolicitud}/estado  
GET /api/v1/auditoria/solicitudes/{codigoSolicitud}/eventos

### **3. Endpoints de Consulta por Socio Comercial**

GET /api/v1/auditoria/socios/{idSocio}/solicitudes  
GET /api/v1/auditoria/socios/{idSocio}/metricas?fechaInicio={date}&fechaFin={date}  
GET /api/v1/auditoria/socios/{codigoSocio}/estadisticas

### **4. Endpoints de Búsqueda Avanzada**

GET /api/v1/auditoria/eventos/buscar?tipoEvento={tipo}&fechaDesde={date}&fechaHasta={date}  
GET /api/v1/auditoria/eventos/buscar?idSocio={id}&idSolicitud={appId}  
GET /api/v1/auditoria/eventos/buscar?idCorrelacion={corrId}  
GET /api/v1/auditoria/eventos/buscar?estadoEvento={estado}&sistemaOrigen={sistema}

### **5. Endpoints de Configuración y Monitoreo**

GET /api/v1/sistema/configuracion/{claveConfiguracion}  
PUT /api/v1/sistema/configuracion/{claveConfiguracion}  
GET /api/v1/sistema/logs?fechaInicio={date}&fechaFin={date}&nivel={level}  
GET /api/v1/sistema/rendimiento/estadisticas  
GET /api/v1/sistema/salud/check  
POST /api/v1/sistema/mantenimiento/limpiar-logs

### **6. Endpoints de Reportes y Analytics**

GET /api/v1/reportes/socios/{idSocio}/dashboard  
GET /api/v1/reportes/eventos/consolidado?fechaInicio={date}&fechaFin={date}  
GET /api/v1/reportes/rendimiento/componentes  
GET /api/v1/reportes/compliance/cnbv  
GET /api/v1/reportes/exportar/{formato}?filtros={query}

### **5. Endpoints de Salud del Sistema**

GET /api/v1/salud  
GET /api/v1/salud/basedatos  
GET /api/v1/salud/conectividad

### **6. Endpoints de Configuración**

GET /api/v1/configuracion/socios  
GET /api/v1/configuracion/tipos-eventos  
GET /api/v1/configuracion/esquemas-eventos/{tipoEvento}

# 5. MONITOREO OPERACIONAL

## Monitoreo de API únicamente

### **Application Insights - Métricas Esenciales**

* **Uptime de API**: Disponibilidad > 99%
* **Response Time**: Latencia promedio < 2 segundos
* **Error Rate**: Tasa de error < 5%
* **Throughput**: Volumen de requests por minuto

### **Alertas Automáticas**

* **API Down**: Notificación inmediata si API no responde
* **High Latency**: Alerta si response time > 5 segundos
* **Error Spike**: Notificación si error rate > 10%
* **Queue Backlog**: Alerta si cola de procesamiento se acumula

### **Logs de Auditoría**

* **Event Processing**: Success/failure de procesamiento de eventos
* **Database Connectivity**: Estado de conexión ExpressRoute/SQL Server
* **Partner Activity**: Logs de actividad por partner

# 6. PRESUPUESTO Y ROI

## Estimación de Costos Mensual - Arquitectura Híbrida

|  |  |  |  |
| --- | --- | --- | --- |
| Componente | SKU/Tier | Justificación | Costo Mensual (USD) |
| **ExpressRoute** | 50 Mbps Standard | Conectividad privada on-premise | $55 |
| **API Management** | Standard (1 unit) | Gestión partners, OAuth2 | $252 |
| **App Service** | Standard S2 (1-3 instances) | API .NET 8, auto-scaling | $146 - $438 |
| **Storage Queues** | Standard + LRS | Procesamiento asíncrono simple | $25 |
| **Application Insights** | Basic 5GB/month | Monitoreo esencial | $58 |
| **Key Vault** | Standard + transactions | Secretos y claves | $25 |
| **External Services** | Partner Biométrico + BRE | Costos por transacción | $1,500 - $2,500 |

### **Total Estimado Azure: $2,061 - $3,353 USD/mes**

### **Infraestructura On-Premise (Bradescard):**

* SQL Server (licencias y hardware existente)
* Storage para compliance (infraestructura existente)
* Backup y DR (procesos actuales)

# 7. PLAN DE IMPLEMENTACIÓN

## Fases de Implementación Simplificada (10 semanas)

### **Fase 1: Setup Básico**

* ✅ Configuración Azure recursos básicos
* ✅ Setup Azure SQL Database con Always Encrypted
* ✅ Implementación API core con .NET 8
* ✅ Configuración Storage Queues
* ✅ Integración con partner piloto

### **Fase 2: Funcionalidad Core**

* ✅ Implementación eventos de auditoría principales
* ✅ Background processing con Storage Queues
* ✅ Testing básico y validación funcional

### **Fase 3: Integración y Monitoreo**

* ✅ Integración partners biométricos y BRE
* ✅ Configuración Application Insights
* ✅ Alertas básicas de salud del sistema
* ✅ Documentación de APIs

### **Fase 4: Producción**

* ✅ Despliegue a producción
* ✅ Onboarding partners iniciales
* ✅ Monitoreo operacional
* ✅ Go-live y soporte inicial

### 

### **Funcionalidad Core Garantizada:**

* ✅ **51 eventos de auditoría** completos
* ✅ **Compliance CNBV** automático
* ✅ **Always Encrypted** para datos PII
* ✅ **APIs de consulta** por aplicación, partner, correlación
* ✅ **Monitoreo operacional** esencial

Esta solución permite a Bradescard obtener **máximo valor con mínima inversión**, manteniendo control total sobre datos críticos mientras aprovecha las ventajas de APIs cloud modernas.

**Contacto del Proyecto**  
*Equipo de Arquitectura*  
📧 arquitectura@bradescard.mx  
📞 +52 55 1234 5678  
📅 Implementación Q1 2025