



Microservicio MsDatosGrafos

Documento Versión: 1.0

Gestión de Seguridad Electrónica



Fecha generación:

01/12/2025

CONTROL DE VERSIÓN

VERSIÓN	FECHA DE ACTUALIZACIÓN	CARGO	DESCRIPCIÓN
1.0	24/11/2025	Arquitecto de Software	Versión inicial del documento

Tabla de contenido

1	CONTEXTO	4
2	INTRODUCCIÓN	4
2.1	Justificación de la Arquitectura de Microservicios	4
2.2	Arquitectura de Referencia	4
2.3	Códigos de Respuesta HTTP Estándar	4
3	CATÁLOGO DE MICROSERVICIOS	5
3.1	Microservicio MsDatosGrafos.....	5
3.1.1	Endpoint: Consultar Vínculos Parentesco.....	5

1 CONTEXTO

El Registro Nacional de Identificación y Estado Civil (RENIEC) es el organismo técnico autónomo encargado de la identificación de los peruanos, otorgar el Documento Nacional de Identidad (DNI) y registrar los hechos vitales. En el marco de la modernización y transformación digital del Estado peruano, RENIEC ha desarrollado el DNI Electrónico (DNIE), un documento de identidad que incorpora tecnología de chip y biometría, permitiendo la autenticación electrónica de ciudadanos y facilitando servicios digitales seguros.

2 INTRODUCCIÓN

Este documento describe el **catálogo de microservicios** identificados para la solución de Personalización del DNIE de RENIEC. El objetivo es establecer una arquitectura técnica moderna, escalable y mantenible que reemplace o complemente los sistemas monolíticos actuales mediante una transición ordenada hacia una arquitectura orientada a microservicios.

2.1 Justificación de la Arquitectura de Microservicios

La adopción de microservicios para esta solución responde a necesidades técnicas y operativas concretas:

Escalabilidad Independiente: Componentes con cargas diferenciadas pueden escalar de forma independiente según demanda real, optimizando recursos de infraestructura.

Resiliencia y Tolerancia a Fallos: El fallo de un microservicio no compromete servicios críticos. Los patrones Circuit Breaker y Retry garantizan continuidad operativa.

Agilidad en el Desarrollo: Equipos autónomos pueden desarrollar, probar y desplegar servicios de manera independiente, reduciendo tiempos de entrega.

Mantenibilidad y Evolución Tecnológica: Cada servicio puede evolucionar tecnológicamente sin afectar al ecosistema completo.

Trazabilidad y Observabilidad: Arquitectura distribuida permite implementar logging centralizado, distributed tracing y métricas granulares.

2.2 Arquitectura de Referencia

La solución se estructura en tres capas principales:

Capa de Exposición (API Management Layer): API Manager como punto único de entrada con gestión centralizada de seguridad, throttling y versionado.

Capa de Representación (Microservices Layer): Microservicios de negocio con lógica específica de dominio y responsabilidad única.

Capa de Integración (Integration Layer): Event Streaming para comunicación asíncrona y conectores a sistemas legados.

2.3 Códigos de Respuesta HTTP Estándar

Todos los microservicios implementan un conjunto estandarizado de códigos de respuesta HTTP para garantizar consistencia:

Código	Descripción
200	OK - Operación completada exitosamente
201	Created - Recurso creado exitosamente
400	Bad Request - Parámetros inválidos o datos incompletos
401	Unauthorized - Token JWT inválido, expirado o ausente

403	Forbidden - Sin permisos suficientes para ejecutar la operación
404	Not Found - Recurso no encontrado en el sistema
408	Request Timeout - Tiempo de espera agotado al conectar
409	Conflict - Conflicto con el estado actual del recurso
422	Unprocessable Entity - Datos válidos pero no procesables por reglas de negocio
429	Too Many Requests - Límite de rate limit excedido
500	Internal Server Error - Error interno del servicio (información generalizada al exterior, detalle en logs)
502	Bad Gateway - Servicio externo no disponible o respuesta inválida
503	Service Unavailable - Servicio temporalmente no disponible o Circuit Breaker abierto
504	Gateway Timeout - Servicio externo no respondió en tiempo esperado

3 CATÁLOGO DE MICROSERVICIOS

3.1 Microservicio MsDatosGrafos

El **Microservicio Datos Grafos** es un componente de persistencia especializada dentro de la plataforma SIIRC. Su propósito principal es interactuar exclusivamente con la Base de Datos de Grafos, la cual está optimizada para el almacenamiento y la consulta de datos estructurados como nodos (entidades) y aristas (relaciones).

Este microservicio actúa como la capa de abstracción de datos para todos los servicios del SIIRC que requieran análisis complejos de conectividad y parentesco.

3.1.1 Endpoint: Consultar Vínculos Parentesco

Endpoint que permite consultar los vínculos de parentesco directos de un ciudadano específico, con soporte para filtrado por tipo de parentesco, grado y vigencia.

Atributo	Valor
Path	/api/v1/APD/MsDatosGrafos/vinculos-parentesco
API Gateway	Interno
Método HTTP	GET
Protocolo	REST/HTTP
Headers	Authorization String (Bearer token JWT para autenticación del usuario) Content-Type String ("application/json") X-Correlation-ID UUID (Identificador único de correlación para trazabilidad end-to-end) X-User-Role String (Rol del usuario que realiza la operación)
Entrada	<ul style="list-style-type: none"> idCiudadano (string): Identificador del ciudadano utilizado como filtro. gradoMaximo (integer): Grado máximo aplicado al criterio de búsqueda. categoriaParentesco (string): Categoría de parentesco utilizada como filtro. soloVigentes (boolean): Indica si solo deben considerarse registros vigentes. pagina (integer): Número de página para la paginación. registrosPorPagina (integer): Cantidad de registros por página.
Respuesta	{ "success": "boolean", "data": { "ciudadanoConsultado": { ... } } }

```
"idCiudadano": "string",
"nombre": "string",
"fechaNacimiento": "YYYY-MM-DD"
},
"vinculos": [
{
  "idRelacion": "integer",
  "tipoParentesco": {
    "codigo": "string",
    "descripcion": "string",
    "esSimetrico": "boolean"
  },
  "gradoParentesco": "integer",
  "categoriaParentesco": "string",
  "ciudadanoRelacionado": {
    "idCiudadano": "string",
    "nombre": "string",
    "fechaNacimiento": "YYYY-MM-DD"
  },
  "vigencia": {
    "fechaInicio": "YYYY-MM-DD",
    "fechaFin": "YYYY-MM-DD",
    "esVigente": "boolean"
  },
  "sustento": {
    "idActaSustento": "string",
    "idDocumentoSustento": "string"
  },
  "observacion": "string"
}
],
"resumen": {
  "totalVinculos": "integer",
  "vinculosConsanguineos": "integer",
  "vinculosPorAfinidad": "integer",
  "vinculosVigentes": "integer",
  "vinculosNoVigentes": "integer"
},
"paginacion": {
  "paginaActual": "integer",
  "registrosPorPagina": "integer",
  "totalRegistros": "integer",
  "totalPaginas": "integer",
  "tieneAnterior": "boolean",
  "tieneSiguiiente": "boolean"
}
},
"metadata": {
  "timestamp": "YYYY-MM-DDThh:mm:ssZ",
  "correlationId": "string",
  "version": "string",
  "tiempoRespuesta": "string"
}
}
```

```
"error": {
  "tipo": "string",
  "titulo": "string",
  "estado": "integer",
  "errores": [
    {
      "detalleError": "string"
    }
  ]
}
```

Error Response

3.1.1.1 Parámetros de Entrada

Dato	Atributo	Tipo	Obligatorio	Longitud Mínima	Longitud Máxima
ID Ciudadano	idCiudadano	String	Sí	8	20
Grado Máximo	gradoMaximo	Integer	No	1	10
Categoría Parentesco	categoriaParentesco	String	No	3	20
Solo Vigentes	soloVigentes	Boolean	No	NA	NA
Página	página	Integer	No	1	9999
Registros por Página	registrosPorPagina	Integer	No	1	100

3.1.1.2 Parámetros de Respuesta

Nombre	Tipo	Obligatorio	Descripción
success	Boolean	Sí	Indica si la operación fue exitosa
data.ciudadanoConsultado	Object	Sí	Datos del ciudadano consultado
data.ciudadanoConsultado.idCiudadano	String	Sí	DNI del ciudadano consultado
data.ciudadanoConsultado.nombre	String	Sí	Nombre completo del ciudadano
data.ciudadanoConsultado.fechaNacimiento	String (Date)	No	Fecha de nacimiento del ciudadano
data.vinculos	Array	Sí	Lista de vínculos de parentesco encontrados
data.vinculos[].idRelacion	Integer	Sí	ID único de la relación en la tabla GRF_PARENTESCO
data.vinculos[].tipoParentesco	Object	Sí	Información del tipo de parentesco
data.vinculos[].tipoParentesco.codigo	String	Sí	Código del tipo de parentesco
data.vinculos[].tipoParentesco.descripcion	String	Sí	Descripción del tipo de parentesco
data.vinculos[].tipoParentesco.esSimetrico	Boolean	Sí	Indica si la relación es simétrica
data.vinculos[].gradoParentesco	Integer	Sí	Grado de parentesco calculado
data.vinculos[].categoriaParentesco	String	Sí	Categoría: CONSANGUINEO o AFINIDAD
data.vinculos[].ciudadanoRelacionado	Object	Sí	Datos del ciudadano relacionado
data.vinculos[].ciudadanoRelacionado.idCiudadano	String	Sí	DNI del ciudadano relacionado
data.vinculos[].ciudadanoRelacionado.nombre	String	Sí	Nombre del ciudadano relacionado

data.vinculos[].ciudadanoRelacionado.fechaNacimiento	String (Date)	No	Fecha de nacimiento
data.vinculos[].vigencia	Object	Sí	Información de vigencia de la relación
data.vinculos[].vigencia.fechaInicio	String (Date)	Sí	Fecha de inicio de vigencia
data.vinculos[].vigencia.fechaFin	String (Date)	Sí	Fecha de fin de vigencia (FECHA_FIN)
data.vinculos[].vigencia.esVigente	Boolean	Sí	Indica si la relación está vigente actualmente
data.vinculos[].sustento	Object	No	Documentos de sustento de la relación
data.vinculos[].sustento.idActaSustento	String	No	ID del acta de sustento
data.vinculos[].sustento.idDocumentoSustento	String	No	ID del documento de sustento
data.vinculos[].observacion	String	No	Observación de la relación
data.resumen	Object	Sí	Resumen estadístico de los vínculos
data.resumen.totalVinculos	Integer	Sí	Total de vínculos encontrados
data.resumen.vinculosConsanguineos	Integer	Sí	Cantidad de vínculos consanguíneos
data.resumen.vinculosPorAfinidad	Integer	Sí	Cantidad de vínculos por afinidad
data.resumen.vinculosVigentes	Integer	Sí	Cantidad de vínculos vigentes
data.resumen.vinculosNoVigentes	Integer	Sí	Cantidad de vínculos no vigentes
data.paginacion	Object	Sí	Información de paginación
data.paginacion.paginaActual	Integer	Sí	Página actual
data.paginacion.registrosPorPagina	Integer	Sí	Registros por página
data.paginacion.totalRegistros	Integer	Sí	Total de registros disponibles
data.paginacion.totalPaginas	Integer	Sí	Total de páginas
data.paginacion.tieneAnterior	Boolean	Sí	Indica si existe página anterior
data.paginacion.tieneSiguiente	Boolean	Sí	Indica si existe página siguiente
metadata.timestamp	String (Date ISO 8601)	Sí	Timestamp de generación de la respuesta
metadata.correlationId	String	Sí	Identificador único de correlación
metadata.version	String	Sí	Versión del API
metadata.tiempoRespuesta	String	Sí	Tiempo total de respuesta
error	Object	No	Objeto de error (presente solo en caso de error)
error.tipo	String	No	Tipo o categoría del error

error.titulo	String	No	Título o descripción del error
error.estado	Integer	No	Código HTTP del error
error.errores	Array	No	Lista de errores específicos
error.errores[].detalleError	String	No	Detalle específico del error

3.1.1.3 Valores para el atributo statusCode

Código	Respuesta	Descripción
200	OK	Vínculos de parentesco consultados exitosamente
400	Bad Request	Parámetros de consulta inválidos o formato incorrecto
401	Unauthorized	Token JWT inválido, expirado o ausente
403	Forbidden	Usuario sin permisos para consultar vínculos de parentesco
404	Not Found	Ciudadano no encontrado en el sistema
422	Unprocessable Entity	ID de ciudadano válido pero sin registros de parentesco
429	Too Many Requests	Límite de rate limit excedido
500	Internal Server Error	Error interno del servidor
503	Service Unavailable	Servicio de base de datos temporalmente no disponible