

DOCUMENTO TECNICO (BORRADOR)

BACKEND API SIMAT

APIs REST y Gestión de Respuestas JSON – Backend SIMAT

Fecha de elaboración: 2025-12
Documento en elaboración.
Autor: Julián Andrés Ramírez

Tabla de contenido

Tabla de contenido.....	2
1. Propósito del documento	3
2. Principios de diseño de las respuestas JSON	3
3. Estructura estándar de respuesta exitosa	3
3.2 Descripción de campos	3
4. Descripción detallada de JSON por dominio	4
4.1 Catálogos (Ejemplo: Países)	4
4.2 Estudiantes	4
4.3 Instituciones.....	4
4.4 Matrículas	5
5. Gestión actual de errores en las APIs	5
5.1 Estructura general de error	5
6. Tipos de errores documentados.....	5
6.1 Error 400 – Bad Request	5
6.2 Error 404 – Not Found.....	6
6.3 Error 500 – Internal Server Error	6
7. Mejoras pendientes en la gestión de errores	6
8. Consideraciones para consumo de endpoints	6
9. Estado del documento	6

Estado del documento: Borrador técnico / Evidencia de desarrollo

Proyecto: Backend API SIMAT – Sistema de Integración y APIs para Power BI

Fecha: Diciembre 2025

1. Propósito del documento

Este documento técnico en estado de borrador amplía la descripción de las estructuras JSON generadas por las APIs del Backend SIMAT, incorporando el análisis de los modelos de respuesta actuales y la documentación preliminar de gestión de errores en el consumo de los endpoints.

El documento se presenta como evidencia del desarrollo, sin carácter definitivo ni contractual.

2. Principios de diseño de las respuestas JSON

Las respuestas del API SIMAT fueron diseñadas bajo los siguientes principios:

- Consistencia estructural en todos los endpoints
- Separación clara entre datos y metadatos
- Mensajes descriptivos orientados al consumidor del servicio
- Inclusión de marcas de tiempo para trazabilidad
- Preparación para integración con herramientas BI y sistemas externos

3. Estructura estándar de respuesta exitosa

Todas las respuestas exitosas siguen el siguiente esquema general:

```
{  
  "success": true,  
  "message": "Descripción funcional del resultado",  
  "data": {},  
  "timestamp": "2025-12-17T01:08:15.728Z"  
}
```

3.2 Descripción de campos

- **success:** Indica el resultado lógico de la operación.
- **message:** Mensaje descriptivo orientado a usuarios técnicos.
- **data:** Objeto o arreglo con la información solicitada.
- **timestamp:** Fecha y hora de generación de la respuesta en formato ISO 8601.

4. Descripción detallada de JSON por dominio

4.1 Catálogos (Ejemplo: Países)

```
{
  "id_pais": 1,
  "codigo_iso": null,
  "nombre": "COLOMBIA"
}
```

Consideraciones técnicas: - Los campos opcionales pueden retornar `null`. - Los valores de texto se entregan normalizados. - La estructura es liviana para consumo masivo.

4.2 Estudiantes

```
{
  "id_estudiante": 19262,
  "numero_documento":
  "ANON_53cdce7a39e5b410cc2dfff458f854c0a",
  "fecha_nacimiento": "2018-12-06T00:00:00.000Z",
  "estrato": 1,
  "observaciones": "Documento anonimizado"
}
```

Aspectos clave: - El campo `numero_documento` se entrega anonimizado. - No se expone información personal sensible. - El modelo está preparado para agregaciones estadísticas.

4.3 Instituciones

```
{
  "id_institucion": 51,
  "nombre_institucion": "AMERU",
  "dane_institucion": "291407000263",
  "id_sector": 1,
  "id_jerarquia": 5,
  "id_calendario": 1
}
```

Uso del JSON: - Permite relacionar jerarquías territoriales. - Facilita filtros por sector y calendario académico.

4.4 Matriculas

```
{
  "id_matricula": 1,
  "id_estudiante": 8,
  "id_sede": 6,
  "id_jornada": 1,
  "id_modelo": 1,
  "id_anio": 1,
  "id_estado": 1,
  "pais_origen_id": 1,
  "consecutivo": "19100100001201",
  "fecha_registro": "2025-12-14T12:10:52.000Z"
}
```

Observaciones: - Modelo relacional optimizado. - Preparado para explotación analítica y reportes.

5. Gestión actual de errores en las APIs

Actualmente, el backend maneja errores de forma centralizada, retornando respuestas estructuradas.

5.1 Estructura general de error

```
{
  "success": false,
  "message": "Descripción del error",
  "timestamp": "2025-12-17T01:12:29.955Z",
  "statusCode": 400
}
```

6. Tipos de errores documentados

6.1 Error 400 – Bad Request

Se presenta cuando: - Faltan parámetros obligatorios - Los valores enviados no cumplen validaciones

```
{
  "success": false,
  "error": "Parámetro de búsqueda requerido",
  "statusCode": 400
}
```

6.2 Error 404 – Not Found

Ocurre cuando el recurso solicitado no existe.

```
{  
  "success": false,  
  "error": "Recurso no encontrado",  
  "statusCode": 404  
}
```

6.3 Error 500 – Internal Server Error

Error inesperado durante la ejecución.

```
{  
  "success": false,  
  "message": "Error interno del servidor",  
  "statusCode": 500  
}
```

7. Mejoras pendientes en la gestión de errores

Se identifican las siguientes mejoras en desarrollo:

- Unificación del atributo `message` vs `error`
- Inclusión de códigos de error internos
- Mayor detalle técnico controlado (entorno desarrollo)
- Documentación extendida por endpoint
- Manejo diferenciado de errores de validación y negocio

8. Consideraciones para consumo de endpoints

- Validar siempre el campo `success`
- Manejar correctamente los códigos HTTP
- Preparar lógica de reintento para errores 500
- No asumir que `data` siempre contiene información

9. Estado del documento

Este documento corresponde a un **borrador técnico en evolución**, susceptible de ajustes conforme avance en el desarrollo del backend, la implementación de la gestión de errores y la integración con Power BI. Asimismo, se cuenta con el documento técnico – backend API SIMAT - Sistema de Integración y APIs para

Power BI donde se profundiza el desarrollo de los endpoints hasta el momento de la elaboración del presente documento.

Documento elaborado como evidencia técnica del desarrollo del Backend SIMAT

CONTROL DE DOCUMENTO

Versión	Fecha	Autor	Descripción
	Diciembre 2025	Julian Andres Ramirez 	Documento en elaboración