

# API Produccion Interface

## Informacion del Proyecto

 <p><b>Grupo San Pablo</b></p> <p>Cliente</p> <p>Complejo Hospitalario San Pablo</p>	 <p><b>ADG Systems</b> Advance Developer Group</p> <p>Desarrollado por</p> <p><b>ADG Systems</b></p>
---	--

Atributo	Valor
Proyecto	Gestion de Honorarios Medicos
Modulo	API Interface de Produccion
Cliente	Complejo Hospitalario San Pablo
Desarrollador	ADG Systems
Version	2.0

## Descripcion General

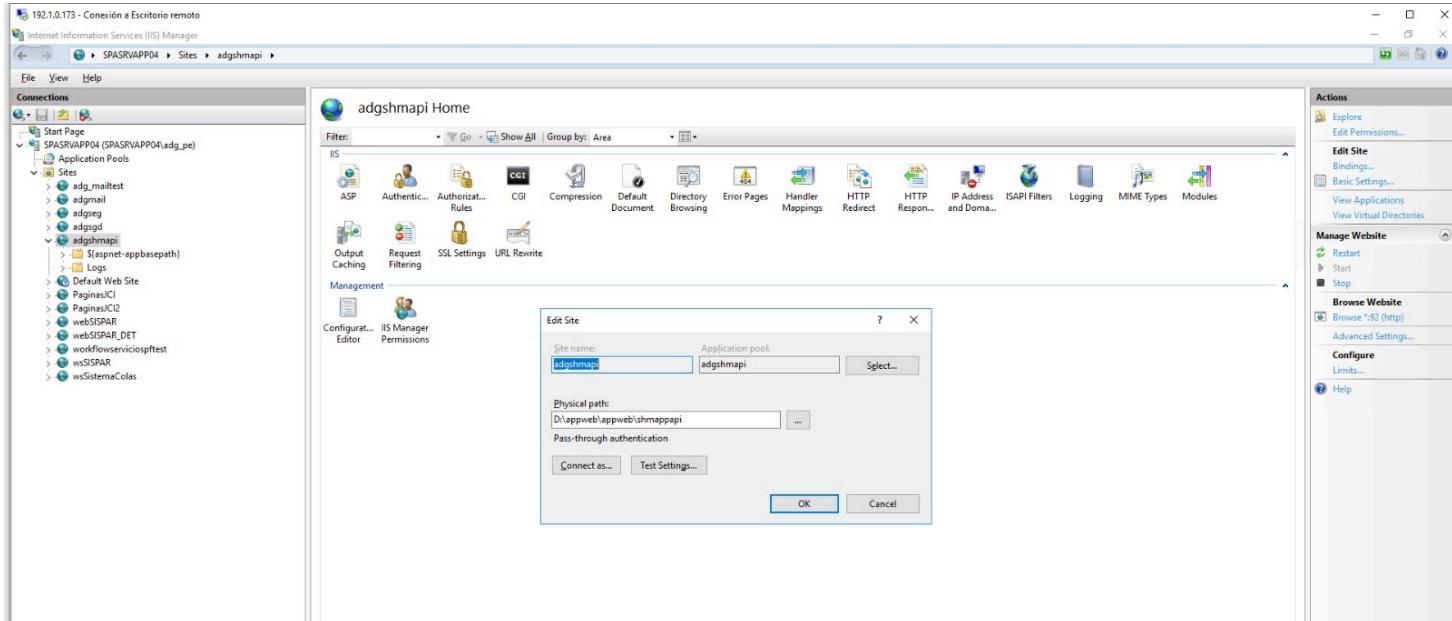
API para el registro masivo de producciones medicas y actualizacion de liquidaciones a traves de una interfaz de integracion. Permite crear multiples registros de produccion y actualizar datos de liquidacion en una sola llamada, validando duplicados y manejando transacciones de forma segura.

# Instalacion y Despliegue

## Informacion del Servidor

Atributo	Valor
Servidor	192.1.0.173
URL Local	<a href="http://localhost:92">http://localhost:92</a>
URL Swagger	<a href="http://localhost:92/swagger">http://localhost:92/swagger</a>
Carpeta Fisica	D:\appweb\shmappapi
Plataforma	IIS (Internet Information Services)

## Configuracion en IIS



## Swagger UI

Para probar los endpoints de forma interactiva, acceda a la interfaz de Swagger:

**URL:** <http://localhost:92/swagger>



# Estructura de Carpetas

```
D:\appweb\shmappapi\  
├── appsettings.json  
├── appsettings.Production.json  
├── SHM.AppApiHonorarioMedico.dll  
├── SHM.AppApiHonorarioMedico.exe  
├── web.config  
└── ...
```

## Verificacion de Instalacion

Para verificar que el API esta funcionando correctamente:

1. Abra un navegador web
2. Acceda a: <http://localhost:92/api/ProduccionInterface/test>
3. Deberia recibir una respuesta como:

```
{  
  "status": "OK",  
  "message": "API activa",  
  "serverDateTime": "2026-01-31T10:30:00",  
  "serverDateTimeUtc": "2026-01-31T15:30:00Z"  
}
```

## Health Check (Monitoreo de Salud)

El API incluye un endpoint de **Health Check** para monitorear el estado de la aplicacion y sus dependencias (base de datos Oracle).

**URL:** <http://localhost:92/health>

**Response (Healthy):**

```
{
  "status": "Healthy",
  "totalDuration": "125.45 ms",
  "timestamp": "2026-01-31T10:30:00",
  "checks": [
    {
      "name": "oracle-database",
      "status": "Healthy",
      "description": null,
      "duration": "120.32 ms"
    }
  ]
}
```

### Response (Unhealthy):

```
{
  "status": "Unhealthy",
  "totalDuration": "5023.12 ms",
  "timestamp": "2026-01-31T10:30:00",
  "checks": [
    {
      "name": "oracle-database",
      "status": "Unhealthy",
      "description": "Error de conexion a Oracle",
      "duration": "5000.00 ms"
    }
  ]
}
```

Estado	Descripcion
<b>Healthy</b>	Todos los servicios funcionan correctamente
<b>Degraded</b>	Algunos servicios tienen problemas pero el API sigue operativo
<b>Unhealthy</b>	Servicios criticos no disponibles

# Logs (NLog)

El API utiliza **NLog** para el registro de eventos y errores. Los logs se generan diariamente en la carpeta `/Log`.

Atributo	Valor
<b>Carpeta</b>	D:\appweb\shmappapi\Log
<b>Formato archivo</b>	Log_yyyyMMdd.txt
<b>Ejemplo</b>	Log_20260131.txt
<b>Rotacion</b>	Diaria (un archivo por dia)

## Estructura de carpeta:

```
D:\appweb\shmappapi\  
└── Log\  
    ├── Log_20260129.txt  
    ├── Log_20260130.txt  
    └── Log_20260131.txt  
└── appsettings.json  
└── ...
```

## Ejemplo de contenido del log:

```
2026-01-31 10:30:15.123 |INFO| Inicio de creacion masiva de producciones mediante interface  
2026-01-31 10:30:15.456 |INFO| Producciones procesadas: 10, Creados: 8, Obviados: 2  
2026-01-31 10:35:22.789 |WARN| Error de validacion al crear producciones - Sede con codigo 'XXX'  
2026-01-31 10:40:33.012 |ERROR| Error al crear producciones mediante interface - ORA-12541: TNS:
```

## Niveles de log:

Nivel	Descripcion
<b>DEBUG</b>	Informacion detallada para depuracion
<b>INFO</b>	Eventos normales de operacion
<b>WARN</b>	Advertencias (errores de validacion)
<b>ERROR</b>	Errores de sistema o excepciones

# Endpoints Disponibles

Metodo	URL	Descripcion
GET	/health	Health Check - Estado del API y BD
GET	/api/ProduccionInterface/test	Prueba de conexion
POST	/api/ProduccionInterface/producciones	Crear producciones
POST	/api/ProduccionInterface/liquidaciones	Actualizar liquidaciones

## Formato de Respuesta Estandarizado

Todos los endpoints devuelven una respuesta con la siguiente estructura:

```
{  
  "isSuccess": true,  
  "message": "Correcto.",  
  "data": { ... },  
  "errors": []  
}
```

Campo	Tipo	Descripcion
isSuccess	boolean	Indica si la operacion fue exitosa
message	string	Mensaje descriptivo del resultado
data	object	Datos de la respuesta (segun endpoint)
errors	array	Lista de errores si los hubiera

# 1. Test de Conexion

## Informacion del Endpoint

Atributo	Valor
URL	GET /api/ProduccionInterface/test
Metodo HTTP	GET
Autenticacion	No requerida

## Response (HTTP 200 OK)

```
{  
  "status": "OK",  
  "message": "API activa",  
  "serverDateTime": "2026-01-31T10:30:00",  
  "serverDateTimeUtc": "2026-01-31T15:30:00Z"  
}
```

# 2. Crear Producciones

## Informacion del Endpoint

Atributo	Valor
URL	POST /api/ProduccionInterface/producciones
Metodo HTTP	POST
Content-Type	application/json
Autenticacion	No requerida (version actual)

## Estructura del Body

El endpoint recibe un **array JSON** de objetos con la siguiente estructura:

```

[
  {
    "codigoSede": "string",
    "codigoEntidad": "string",
    "codigoProduccion": "string",
    "numeroProduccion": "string",
    "tipoProduccion": "string",
    "tipoEntidadMedica": "string",
    "tipoMedico": "string",
    "tipoRubro": "string",
    "descripcion": "string",
    "periodo": "string",
    "fechaProduccion": "string",
    "estadoProduccion": "string",
    "mtoConsumo": 0.00,
    "mtoDescuento": 0.00,
    "mtoSubtotal": 0.00,
    "mtoRenta": 0.00,
    "mtoIgv": 0.00,
    "mtoTotal": 0.00
  }
]

```

## Descripcion de Campos

Campo	Tipo	Obligatorio	Descripcion
codigoSede	string	Si	Codigo de la sede. Se utiliza para obtener el IdSede interno.
codigoEntidad	string	Si	Codigo de la entidad medica. Se utiliza para obtener el IdEntidadMedica interno.
codigoProduccion	string	Si	Codigo unico de la produccion.
numeroProduccion	string	Si	Numero de la produccion.
tipoProduccion	string	Si	Tipo de produccion (ej: "CONS", "PROC").
tipoEntidadMedica	string	Si	Tipo de entidad medica (ej: "MEDICO", "CLINICA").
tipoMedico	string	Si	Tipo de medico (ej: "IN", "EX").

Campo	Tipo	Obligatorio	Descripcion
tipoRubro	string	Si	Tipo de rubro.
descripcion	string	Si	Descripcion de la produccion.
periodo	string	Si	Periodo de la produccion en formato dd/MM/yyyy (ej: "12/01/2026").
fechaProduccion	string	Si	Fecha de produccion en formato dd/MM/yyyy HH:mm:ss (ej: "31/01/2026 14:30:00").
estadoProduccion	string	Si	Estado de la produccion (ej: "ACTIVO", "PENDIENTE").
mtoConsumo	decimal	Si	Monto de consumo.
mtoDescuento	decimal	Si	Monto de descuento.
mtoSubtotal	decimal	Si	Monto subtotal.
mtoRenta	decimal	Si	Monto de renta.
mtoIgv	decimal	Si	Monto de IGV.
mtoTotal	decimal	Si	Monto total.

## Llave Unica (Producciones)

La combinacion de los siguientes campos forma la **Llave unica** para identificar registros duplicados:

- codigoSede (convertido a IdSede)
- codigoEntidad (convertido a IdEntidadMedica)
- codigoProduccion
- numeroProduccion
- tipoEntidadMedica

Si un registro con esta llave ya existe en la base de datos, sera **omitido** (no se creara duplicado).

## Respuesta Exitosa (HTTP 201 Created)

```
{  
  "isSuccess": true,  
  "message": "Correcto.",  
  "data": {  
    "cantidadCreados": 8,  
    "cantidadObviados": 2,  
    "totalProcesados": 10  
  },  
  "errors": []  
}
```

## Respuesta de Error - Validacion (HTTP 400 Bad Request)

### Colección vacía:

```
{  
  "isSuccess": false,  
  "message": "Error de validacion.",  
  "data": null,  
  "errors": ["La colección de producciones no puede estar vacía"]  
}
```

### Sede no encontrada:

```
{  
  "isSuccess": false,  
  "message": "Error de validacion.",  
  "data": null,  
  "errors": ["Sede con código 'SEDE001' no encontrada"]  
}
```

### Entidad médica no encontrada:

```
{
  "isSuccess": false,
  "message": "Error de validacion.",
  "data": null,
  "errors": ["Entidad medica con codigo 'ENT001' no encontrada"]
}
```

### **Validacion de campos obligatorios:**

```
{
  "isSuccess": false,
  "message": "Error de validacion.",
  "data": null,
  "errors": [
    "The CódigoSede field is required.",
    "The TipoEntidadMedica field is required."
  ]
}
```

## **Respuesta de Error - Servidor (HTTP 500 Internal Server Error)**

```
{
  "isSuccess": false,
  "message": "Error interno del servidor.",
  "data": null,
  "errors": ["Detalle del error"]
}
```

## **3. Actualizar Liquidaciones**

### **Informacion del Endpoint**

Atributo	Valor
<b>URL</b>	POST /api/ProduccionInterface/liquidaciones
<b>Metodo HTTP</b>	POST

Atributo	Valor
<b>Content-Type</b>	application/json
<b>Autenticacion</b>	No requerida (version actual)

## Estructura del Body

El endpoint recibe un **array JSON** de objetos con la siguiente estructura:

```
[
  {
    "codigoSede": "string",
    "codigoEntidad": "string",
    "codigoProduccion": "string",
    "numeroProduccion": "string",
    "tipoEntidadMedica": "string",
    "numeroLiquidacion": "string",
    "codigoLiquidacion": "string",
    "periodoLiquidacion": "string",
    "estadoLiquidacion": "string",
    "fechaLiquidacion": "string",
    "descripcionLiquidacion": "string"
  }
]
```

## Descripcion de Campos

Campo	Tipo	Obligatorio	Descripcion
codigoSede	string	Si	Codigo de la sede (llave para busqueda).
codigoEntidad	string	Si	Codigo de la entidad medica (llave para busqueda).
codigoProduccion	string	Si	Codigo de la produccion (llave para busqueda).
numeroProduccion	string	Si	Numero de la produccion (llave para busqueda).

Campo	Tipo	Obligatorio	Descripcion
tipoEntidadMedica	string	Si	Tipo de entidad medica (llave para busqueda).
numeroLiquidacion	string	Si	Numero de liquidacion a registrar.
codigoLiquidacion	string	Si	Codigo de liquidacion a registrar.
periodoLiquidacion	string	Si	Periodo de liquidacion (ej: "202601").
estadoLiquidacion	string	Si	Estado de la liquidacion (ej: "PROCESADO", "PENDIENTE").
fechaLiquidacion	string	Si	Fecha de liquidacion en formato dd/MM/yyyy HH:mm:ss .
descripcionLiquidacion	string	Si	Descripcion de la liquidacion.

## Llave de Busqueda (Liquidaciones)

La combinacion de los siguientes campos se usa para **localizar** la produccion a actualizar:

- `codigoSede` (convertido a `IdSede`)
- `codigoEntidad` (convertido a `IdEntidadMedica`)
- `codigoProduccion`
- `numeroProduccion`
- `tipoEntidadMedica`

Si la produccion **no existe**, el registro sera **omitido** (cantidadObviados).

## Respuesta Exitosa (HTTP 200 OK)

```
{  
  "isSuccess": true,  
  "message": "Correcto.",  
  "data": {  
    "cantidadCreados": 5,  
    "cantidadObviados": 1,  
    "totalProcesados": 6  
  },  
  "errors": []  
}
```

**Nota:** En este endpoint, `cantidadCreados` representa la cantidad de registros **actualizados** exitosamente.

## Respuesta de Error - Validacion (HTTP 400 Bad Request)

### Coleccion vacia:

```
{  
  "isSuccess": false,  
  "message": "Error de validacion.",  
  "data": null,  
  "errors": ["La colección de liquidaciones no puede estar vacía"]  
}
```

### Formato de fecha invalido:

```
{  
  "isSuccess": false,  
  "message": "Error de validacion.",  
  "data": null,  
  "errors": ["Formato de fecha invalido para FechaLiquidacion: '2026-01-31'. Use formato dd/MM/"]  
}
```

# Comportamiento Transaccional

Ambos endpoints implementan un comportamiento **transaccional** para garantizar la integridad de los datos:

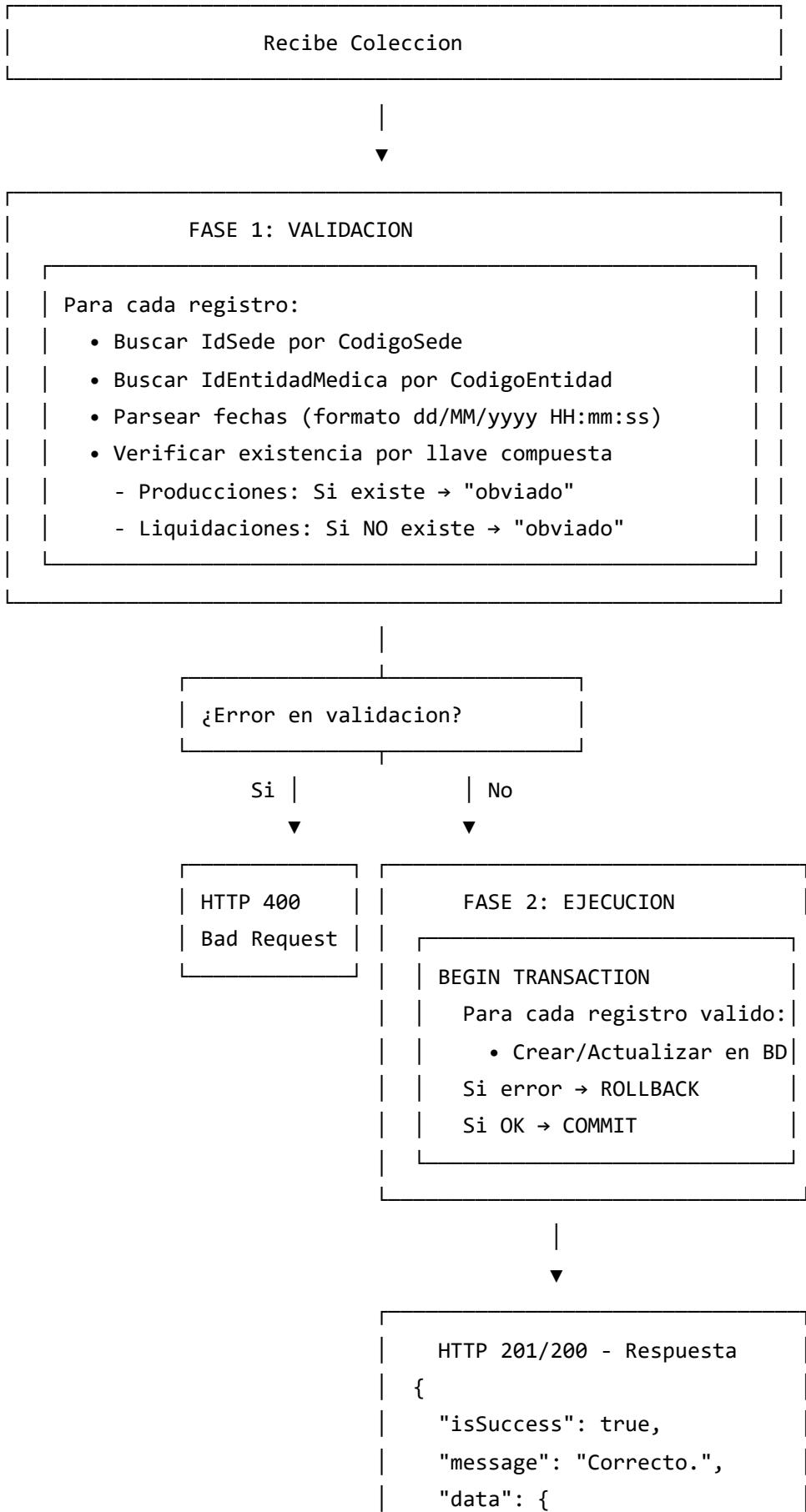
1. **Fase de Validacion:** Se validan todos los registros antes de crear/actualizar cualquiera.

- Se verifica que exista cada `codigoSede`
- Se verifica que exista cada `codigoEntidad`
- Se parsean las fechas con el formato requerido
- Se identifican los registros duplicados/inexistentes

2. **Fase de Ejecucion:** Los registros validos se procesan dentro de una transaccion.

- Si ocurre un error durante la operacion, **se revierte toda la operacion**
- Ningún registro se guarda si hay un fallo

# Diagrama de Flujo



```
|     "cantidadCreados": N,  
|     "cantidadObviados": M,  
|     "totalProcesados": N+M  
| },  
|   "errors": []  
| }  
| }
```

## Ejemplos de Uso

### Ejemplo 1: Crear Producciones

Request:

```
POST /api/ProduccionInterface/producciones HTTP/1.1
```

```
Host: localhost:5001
```

```
Content-Type: application/json
```

```
[  
  {  
    "codigoSede": "01",  
    "codigoEntidad": "0994",  
    "codigoProduccion": "20251017",  
    "numeroProduccion": "20251039",  
    "tipoProduccion": "02",  
    "tipoEntidadMedica": "1",  
    "tipoMedico": "02",  
    "tipoRubro": "03",  
    "descripcion": "SEGUNDA SEMANA DE OCTUBRE 2025",  
    "periodo": "09/10/2025",  
    "fechaProduccion": "09/10/2025 10:30:00",  
    "estadoProduccion": "ACTIVO",  
    "mtoConsumo": 1368.46,  
    "mtoDescuento": 794.23,  
    "mtoSubtotal": 574.23,  
    "mtoRenta": 0,  
    "mtoIgv": 103.36,  
    "mtoTotal": 677.59  
  },  
  {  
    "codigoSede": "01",  
    "codigoEntidad": "0995",  
    "codigoProduccion": "20251055",  
    "numeroProduccion": "20251055",  
    "tipoProduccion": "01",  
    "tipoEntidadMedica": "1",  
    "tipoMedico": "02",  
    "tipoRubro": "03",  
    "descripcion": "OCTUBRE 2025",  
    "periodo": "10/10/2025",  
    "fechaProduccion": "10/10/2025 14:00:00",  
    "estadoProduccion": "ACTIVO",  
    "mtoConsumo": 200,  
    "mtoDescuento": 0,  
    "mtoSubtotal": 200,  
    "mtoRenta": 16,  
    "mtoIgv": 0,  
    "mtoTotal": 200  
  }]
```

```
        "mtoTotal": 184
    }
]
```

## Response (201 Created):

```
{
  "isSuccess": true,
  "message": "Correcto.",
  "data": {
    "cantidadCreados": 2,
    "cantidadObviados": 0,
    "totalProcesados": 2
  },
  "errors": []
}
```

## Ejemplo 2: Envio de Liquidaciones

### Request:

```
POST /api/ProduccionInterface/liquidaciones HTTP/1.1
Host: localhost:5001
Content-Type: application/json
```

```
[
  {
    "codigoSede": "01",
    "codigoEntidad": "0994",
    "codigoProduccion": "20251017",
    "numeroProduccion": "20251039",
    "tipoEntidadMedica": "1",
    "numeroLiquidacion": "20251017",
    "codigoLiquidacion": "20251017",
    "periodoLiquidacion": "10/10/2025",
    "estadoLiquidacion": "AUTORIZADO",
    "fechaLiquidacion": "31/01/2026 18:00:00",
    "descripcionLiquidacion": "CREDITO / 3RA SEMANA DICIEMBRE 2025, PROD:20251227"
  }
]
```

## Response (200 OK):

```
{  
  " isSuccess": true,  
  " message": "Correcto.",  
  " data": {  
    " cantidadCreados": 1,  
    " cantidadObviados": 0,  
    " totalProcesados": 1  
  },  
  " errors": []  
}
```

## Ejemplo 3: Crear Producciones con cURL

```
curl -X POST "http://localhost:92/api/ProduccionInterface/producciones" \  
-H "Content-Type: application/json" \  
-d '['  
  {  
    "codigoSede": "001",  
    "codigoEntidad": "ENT001",  
    "codigoProduccion": "PROD2026001",  
    "numeroProduccion": "0001",  
    "tipoProduccion": "HONORARIO",  
    "tipoEntidadMedica": "MEDICO",  
    "tipoMedico": "STAFF",  
    "tipoRubro": "CONSULTA",  
    "descripcion": "Consulta ambulatoria cardiologia",  
    "periodo": "202601",  
    "fechaProduccion": "15/01/2026 09:00:00",  
    "estadoProduccion": "PROCESADO",  
    "mtoConsumo": 200.00,  
    "mtoDescuento": 0.00,  
    "mtoSubtotal": 200.00,  
    "mtoRenta": 20.00,  
    "mtoIgv": 36.00,  
    "mtoTotal": 216.00  
  }  
'
```

## Ejemplo 4: Actualizar Liquidaciones con cURL

```
curl -X POST "http://localhost:92/api/ProduccionInterface/liquidaciones" \
-H "Content-Type: application/json" \
-d '[
{
    "codigoSede": "001",
    "codigoEntidad": "ENT001",
    "codigoProduccion": "PROD2026001",
    "numeroProduccion": "0001",
    "tipoEntidadMedica": "MEDICO",
    "numeroLiquidacion": "LIQ-2026-00001",
    "codigoLiquidacion": "LQ001",
    "periodoLiquidacion": "202601",
    "estadoLiquidacion": "LIQUIDADO",
    "fechaLiquidacion": "31/01/2026 23:59:59",
    "descripcionLiquidacion": "Liquidacion mensual enero 2026 - Cardiologia"
}
]'
```

## Ejemplo 5: Producciones con Registros Duplicados

Cuando se envian registros que ya existen en la base de datos:

### Request:

```
[
  { "codigoSede": "001", "codigoEntidad": "ENT001", "codigoProduccion": "PROD001", "numeroProduccion": "0001", "periodoLiquidacion": "202601", "descripcionLiquidacion": "Liquidacion mensual enero 2026 - Cardiologia" },
  { "codigoSede": "001", "codigoEntidad": "ENT001", "codigoProduccion": "PROD002", "numeroProduccion": "0002", "periodoLiquidacion": "202601", "descripcionLiquidacion": "Liquidacion mensual enero 2026 - Cardiologia" },
  { "codigoSede": "001", "codigoEntidad": "ENT001", "codigoProduccion": "PROD003", "numeroProduccion": "0003", "periodoLiquidacion": "202601", "descripcionLiquidacion": "Liquidacion mensual enero 2026 - Cardiologia" }
]
```

**Response (201 Created): (si PROD001 y PROD002 ya existian)**

```
{  
  "isSuccess": true,  
  "message": "Correcto.",  
  "data": {  
    "cantidadCreados": 1,  
    "cantidadObviados": 2,  
    "totalProcesados": 3  
  },  
  "errors": []  
}
```

## Ejemplo 6: Error por Sede No Encontrada

**Request:**

```
[  
  {  
    "codigoSede": "SEDE_INVALIDA",  
    "codigoEntidad": "ENT001",  
    "codigoProduccion": "PROD001",  
    ...  
  }  
]
```

**Response (400 Bad Request):**

```
{  
  "isSuccess": false,  
  "message": "Error de validacion.",  
  "data": null,  
  "errors": ["Sede con codigo 'SEDE_INVALIDA' no encontrada"]  
}
```

## Ejemplo 7: Liquidaciones con Produccion No Encontrada

Cuando se intenta actualizar liquidacion de una produccion que no existe:

**Request:**

```
[
  {
    "codigoSede": "001",
    "codigoEntidad": "ENT001",
    "codigoProduccion": "PROD_NO_EXISTE",
    "numeroProduccion": "001",
    "tipoEntidadMedica": "MEDICO",
    "numeroLiquidacion": "LIQ001",
    ...
  }
]
```

**Response (200 OK):** (el registro se omite, no genera error)

```
{
  "isSuccess": true,
  "message": "Correcto.",
  "data": {
    "cantidadCreados": 0,
    "cantidadObviados": 1,
    "totalProcesados": 1
  },
  "errors": []
}
```

## Codigos de Estado HTTP

Codigo	Descripcion
<b>200 OK</b>	Operacion exitosa (liquidaciones).
<b>201 Created</b>	Operacion exitosa (producciones).
<b>400 Bad Request</b>	Error de validacion: campos faltantes, sede/entidad no encontrada, formato de fecha invalido, o colección vacia.
<b>500 Internal Server Error</b>	Error interno del servidor.

# Consideraciones Importantes

- Idempotencia:** El API de producciones es seguro para reintentos. Si se envia la misma colección multiples veces, los registros duplicados serán omitidos automáticamente.
- Atomicidad:** Si ocurre un error durante la creación/actualización de cualquier registro, **ningún registro se guarda**. La operación es todo o nada.
- Validación previa:** Todos los códigos de sede y entidad médica son validados antes de iniciar la operación, evitando operaciones parciales.
- Formato de fechas:** Las fechas deben enviarse en formato dd/MM/yyyy HH:mm:ss (ej: "31/01/2026 14:30:00").
- Rendimiento:** Para grandes volúmenes de datos, considere enviar lotes de máximo 1000 registros por llamada.

## Información Técnica

Atributo	Valor
Autor	ADG Antonio
Fecha de Creación	2026-01-19
versión	2.0
Proyecto	SHM.AppApiHonorarioMedico

## Changelog

versión	fecha	descripción
1.0	2026-01-19	versión inicial del API
2.0	2026-01-31	- Nueva ruta /producciones para crear producciones

<b>Version</b>	<b>Fecha</b>	<b>Descripcion</b>
		<ul style="list-style-type: none"> <li>- Nuevo endpoint /liquidaciones para actualizar liquidaciones</li> </ul>
		<ul style="list-style-type: none"> <li>- Nueva llave compuesta de 5 campos</li> </ul>
		<ul style="list-style-type: none"> <li>- Formato de respuesta estandarizado (isSuccess, message, data, errors)</li> </ul>
		<ul style="list-style-type: none"> <li>- Campos agregados: numeroProduccion, tipoEntidadMedica, fechaProduccion</li> </ul>
		<ul style="list-style-type: none"> <li>- Campos removidos: fechaCreacion, concepto</li> </ul>
		<ul style="list-style-type: none"> <li>- Formato de fecha: dd/MM/yyyy HH:mm:ss</li> </ul>