

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001 Pàg. 1 / 16

Registre de canvis del document

Revisió	Redactat per	Revisat per	Aprobat per	Data aprovació	Data publicació
Revisió	Apartat	Data Modificació	Motiu del canvi		
1.0	-	23/06/2026	Primera versió del disseny tècnic de la política ctti-error-management 2.0.0.		

RESPONSABLE DEL DOCUMENT: Accenture

Í N D E X

1. INTRODUCCIÓ	2
1.1 Propòsit	2
1.2 Abast	2
1.3 Definicions, acrònims i abreviatures	2
1.4 Referències	3
2. DISSENY DETALLAT	4
2.1 Component ctti-error-management	4
2.1.1 Disseny del component	4
2.1.2 Escenaris i algorismes	5
2.1.3 Altres vistes i informació addicional.....	7
ANNEXOS	11
Annex A — Seguretat i controls	11
Annex B — Consideracions de rendiment	11
Annex C — Versionat i compatibilitat	11
Annex D — Traçabilitat amb codi i tests	12
Annex E — Configuració d'API consumidora i exemples de resposta	14
Annex F — Parametrització i rols	16

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 2 / 16

1. INTRODUCCIÓ

El present document descriu el disseny tècnic detallat de la política ctti-error-management versió 2.0.0, desplegada sobre IBM API Connect 10.0.8.8 i IBM DataPower API Gateway 10.6.0.6 (compatible també amb la versió 10.6.0.9 prevista). La informació aquí recollida ha de ser suficient perquè la fase de construcció no requereixi prendre decisions addicionals més enllà de la pròpia sintaxi.

1.1 Propòsit

El propòsit d'aquesta política és centralitzar la generació de respostes d'error de totes les APIs del catàleg corporatiu. La política s'invoca des del bloc catch de l'acoblament i produeix un cos de resposta coherent, un statusCode HTTP correcte i un Content-Type consistent amb el format configurat (JSON, XML, SOAP 1.1 o SOAP 1.2). D'aquesta manera, totes les APIs del CTTI exposen errors amb la mateixa estructura i vocabulari, simplificant la integració per part dels consumidors.

La versió 2.0.0 amplia la funcionalitat de la versió 1.0.0 amb suport simultani de SOAP 1.1 i SOAP 1.2, generació correcta del faultcode com QName vàlid, normalització del statusCode al format exigít per DataPower, correcció de diversos errors del mapa d'errors, suport de nous noms d'excepció suportats i robustesa addicional en la lectura de l'errorCode procedent d'excepcions controlades.

L'objectiu d'aquest disseny és documentar formalment la nova versió, alineada amb el codi font lliurat i la col·lecció Postman de tests funcionals, perquè pugui ser utilitzada com a base contractual entre l'equip de plataforma i els equips d'aplicació consumidors.

1.2 Abast

Aquest document cobreix els aspectes tècnics següents:

- Arquitectura i ubicació de la política dins del catch de l'acoblament d'una API.
- Estructura del GatewayScript (codi font) i del YAML de definició de política.
- Catàleg d'excepcions DataPower suportades i el seu mapatge a errorCode CTTI.
- Catàleg d'errors estàndard CTTI (mapaErrores) amb el seu httpCode i moreInformation.
- Mecanisme d'errors particulars per API mitjançant la propietat erroresParticularesCtti.
- Mecanisme d'estructura custom del cos d'error mitjançant x-customDefinitions del swagger.
- Generació de la resposta en els quatre formats suportats: json, xml, soap11, soap12.
- Procediment de versionat i migració des de la versió 1.0.0.

Queden fora de l'abast d'aquest document:

- La política nativa d'IBM "catch" inclosa per defecte en API Connect (CTTI conserva la seva pròpia política per obtenir funcionalitat addicional com SOAP, custom schema i errors particulars).
- Altres polítiques CTTI invocades en fluxos nominals (ctti-validate-ip, ctti-validate-request, etc.), que es documenten en dissenys tècnics propis.
- La infraestructura DataPower (clústers, gateways, certificats), responsabilitat de l'equip d'infraestructura.

1.3 Definicions, acrònims i abreviatures

Terme	Descripció
API Connect	Producte d'IBM per a la gestió, publicació i seguretat d'APIs REST i SOAP. Versió actual: 10.0.8.8.
DataPower	Gateway d'aplicació d'IBM (DataPower API Gateway) que executa les polítiques d'API Connect. Versió actual: 10.6.0.6, prevista 10.6.0.9.
GatewayScript	Llenguatge basat en JavaScript ES5/V8 que s'executa dins del motor de DataPower. La política està escrita a GatewayScript v2.0.
Catch	Bloc de l'acoblament d'una API que captura els errors generats per les polítiques prèvies i s'encarrega de transformar-los en resposta cap al client.

Finançat per



GOBIERNO
DE ESPAÑA



Plan de
Recuperación,
Transformación
y Resiliencia



Next Generation
Catalunya



Generalitat
de Catalunya

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 3 / 16

Terme	Descripció
errorCode CTTI	Codi d' error amb format E0NNN definit per la convenció CTTI. Identifica unívocament la categoria de l' error i s' utilitza tant en logs com en la resposta al client.
SOAP Fault	Estructura XML estandarditzada per representar errors en respostes SOAP. Hi ha dues versions incompatibles (1.1 i 1.2).
QName	Qualified Name. Nom XML compost per un prefix de namespace i un nom local (per exemple SOAP-ENV:Client). És el format requerit per a l' element faultcode de SOAP 1.1.
Badgerfish	Convenció per representar XML en JSON utilitzada pel mòdul json-xml-converter. Cada valor de text s'embolica en un objecte amb la clau "\$".

1.4 Referències

- Documentació corporativa API Connect CTTI (<https://canigo.ctti.gencat.cat/plataformes/apim/documentacio/>).
- IBM API Connect 10.0.8 KC (<https://www.ibm.com/docs/en/api-connect/10.0.8>).
- IBM DataPower Gateway 10.6 KC (<https://www.ibm.com/docs/en/datapower-gateway/10.6>).
- W3C SOAP 1.1 (<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>). W3C SOAP 1.2 (<https://www.w3.org/TR/soap12-part1/>).
- RFC 7807 (<https://datatracker.ietf.org/doc/html/rfc7807>).
- RFC 8259 (<https://datatracker.ietf.org/doc/html/rfc8259>).

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 4 / 16

2. DISSENY DETALLAT

2.1 Component ctti-error-management

2.1.1 Disseny del component

Visió general de la política

ctti-error-management és una política de tipus gatewayscript que s'executa dins del bloc catch de l'acoblament d'una API d'API Connect. La seva responsabilitat és transformar qualsevol excepció produïda durant el processament de la petició (ja sigui generada per DataPower, per una política prèvia o per un backend) en una resposta HTTP coherent amb el format declarat per l'API.

La política produeix tres elements de la resposta: el cos (body) amb la informació de l'error en el format configurat, el statusCode HTTP amb la seva reason phrase corresponent i el Content-Type adequat al format. La resposta resultant s'envia al client sense més processament addicional.

Posició dins de l'assemblatge

La política s'invoca exclusivament des del catch de l'acoblament, mai des del flux nominal. Qualsevol política del flux nominal que necessiti rebutjar la petició ho fa mitjançant context.reject() emetent un error controlat, que serà recollit pel catch i transformat per aquesta política.

Configuració recomanada del catch:

- Un únic bloc catch per a tots els noms d'excepció suportats.
- Dins del catch, una única invocació de ctti-error-management com a política terminal.
- No s'han d'encadenar polítiques addicionals després de l'error-management: la política escriu directament a l'output stream i finalitza la resposta.
- No cal configurar diferents catches per codi d'error, ja que la pròpia política s'encarrega del mapatge intern.

Diagrama de components

Representació textual del flux d'execució de la política, des de la captura de l'excepció fins a l'escriptura de la resposta:

```
[Excepció en flux nominal]
  v (context.reject / error DataPower)
[Catch de l'acoblament de l'API]
  v
[Política ctti-error-management v2.0.0]
  |- Lectura de context ----> FormatoRespuesta, exception, verb, path, statusCode
  |- Lectura del swagger ----> x-ibm-configuration.x-
customPaths.<path>.<verb>.responses.<code>.schema.$ref
  |- Lectura de propietats --> api.properties.errorParticularesCtti
  |- Parseu de detall ----> exception.message (string o JSON)
  |- initCapturaErrores() ----> resol errorCode segons nom d'excepcion
  |   |- JavaScriptError -> errorPropio(detalle)
  |   |- ConnectionError -> E0900
  |   |- BadRequestError -> E0400
  |   |- ValidateError -> E0400
  |   |- SOAPError -> E0600
  |   |- OperationError -> httpCodeToErrorCode(statusCode)
  |   |- TransformError -> E0800
  |   |- PropertyError -> E0800
  |   |- RedactionError -> E0800
  |   |- AssemblyRateLimit -> E0429
  |   |- default -> E0500
  |- estructuraError() ----> aplica schema custom si existeix
  |- createResponse() ----> json | xml | soap11 | soap12
  |- session.output.write() --> cos + statusCode + Content-Type
```

Components i mòduls del GatewayScript

Finançat per



GOBIERNO
DE ESPAÑA



Plan de
Recuperación,
Transformación
y Resiliencia



Next Generation
Catalunya



Generalitat
de Catalunya

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 5 / 16

El codi font de la política s'estructura en vuit seccions lògiques numerades. Cada secció té una responsabilitat única i pot ser auditada de forma independent.

Secció	Identificador en el codi	Responsabilitat
1	Constants i mapaErrors	Definició del catàleg d'errades estàndard CTTI amb el seu httpCode, httpMessage i moreInformation.
2	Lectura de context	Recull FormatoRespuesta, exception, verb, operationPath, statusCode i reasonPhrase originals.
3	Helpers d'accés al mapa	getHttpCode, getHttpMessage, getMoreInformation, getErrorCode, addErrorMap, httpCodeToErrorCode.
4	Generació de l'objecte d'error	initCapturaErrores (switch sobre el tipus d'excepció) i errorPropio (processa errors controlats estructurats).
5	Estructura custom de l'error	estructuraError, construirObjetoEstructura, limpiarObjeto: aplica el schema custom definit en el swagger.
6	Format de resposta	createResponse (dispatcher), escribirErrorXML (badgerfish), escribirErrorSOAP (1.1 i 1.2), escapeXml.
7	Resolució del schema custom	montarJson: resol els \$ref del swagger per construir l'estructura del cos.
8	Flux principal	context.swagger.readAsJSON amb callback que orquestra totes les seccions anteriors.

Estructura del fitxer YAML de la política

El fitxer ctti-error-management-2.0.0.yaml conté la definició completa de la política. Els seus blocs principals són:

Bloc YAML	Contingut
policy	Versió del schema de política (1.0.0).
info	title (CTTI Error Management), name (ctti-error-management), version (2.0.0), description.
attach	rest, soap. La política aplica a APIs REST i SOAP.
gateways	datapower-api-gateway.
properties	Schema JSON Draft 04 amb la propietat FormatoRespuesta (enum json / xml / soap11 / soap12, default json, requerida).
assembly.execute	Únic element gatewayscript v2.0.0 amb el codi font encastat com a bloc literal indentat.

2.1.2 Escenaris i algorismes

Flux nominal: error controlat emès per una altra política

- En el flux nominal, qualsevol política prèvia de l'acoblament pot detectar una condició d'error i propagar-la al bloc catch mitjançant context.reject(...), ja sigui com a error controlat o com una de les excepcions suportades per la plataforma.
- Un cop capturada l'excepció, la política analitza el seu nom (exception.name) i, quan procedeix, tracta d'interpretar el contingut d'exception.message per recuperar informació estructurada addicional.
- En el cas d'errors controlats emesos com JavaScriptError, si el detall conté un objecte serialitzat amb la informació de negoci de l'error, la funció errorPropio construeix la resposta final a partir de l'errorCode declarat, resolent el seu httpCode, httpMessage i moreInformation segons el mapa d'errors.

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)		N. revisió doc.: 1.0
	Disseny detallat de la solució		
	N. versió aplicació: 2.0.0	Build: 001	Pàg. 6 / 16

- Per a la resta de casuístiques suportades (RedactionError, PropertyError, TransformError, OperationError o altres), la política aplica el tractament específic definit en `initCapturaErrores`, resolent en tots els casos un objecte d'error normalitzat.
- Finalment, es genera el cos en el format configurat i s'escriu la resposta corresponent al consumidor.

Flux d'error de backend: OperationError

- La política `invoke` ha rebut del backend una resposta HTTP 4xx o 5xx.
- El `catch` captura l'excepció amb nom "OperationError"; `context.message.statusCode` conté el codi original del backend.
- `initCapturaErrores` invoca `httpCodeToErrorCode(statusCode)` per construir l'`errorCode` CTTI equivalent (per exemple HTTP 404 → E0404).
- Si l'`errorCode` generat existeix en `mapaErrores`, s'utilitza directament; si no existeix, es cau sobre E0600 (Backend service error).
- Es genera la resposta amb el `httpCode` original del mapa, conservant la coherència amb el codi retornat pel backend.

Flux d'error de connexió

- `DataPower` no ha pogut establir connexió amb el backend (DNS, timeout TCP, certificat, etc.).
- El `catch` captura l'excepció amb nom "ConnectionError".
- `initCapturaErrores` resol directament l'`errorCode` E0900 (Backend connection error) amb `httpCode` 500.
- No s'intenta inspeccionar el detall: la resposta al client és genèrica per no filtrar informació d'infraestructura.

Flux d'error genèric no previst

- Qualsevol excepció el nom de la qual no estigui contemplat en el `switch` (RuntimeError, altres).
- S'entra al `case default` i es resol amb `DEFAULT_ERROR` (E0500) i `httpCode` 500.
- Aquest flux actua com a xarxa de seguretat: cap excepció pot arribar al client sense transformació.

Mapeig d'excepcions DataPower

La política gestiona explícitament els següents noms d' excepció, obtinguts de `exception.name`:

Nom d' excepció	errorCode resultant	httpCode	Origen típic
JavaScriptError	Variable (errorPropio)	Maternal	<code>context.reject</code> des d' una política CTTI amb detall estructurat.
ConnectionError	E0900	500	Fallada de connexió al backend (DNS, TCP, TLS, timeout).
BadRequestError	E0400	400	Body de petició malformat, <code>content-type</code> invàlid.
ValidateError	E0400	400	Validació de <code>schema</code> fallida (<code>validate policy</code>).
SOAPError	E0600	500	El backend ha retornat un SOAP Fault.
OperationError	E0NNN (HTTP del backend) o E0600	Maternal	El backend ha retornat HTTP 4xx o 5xx.
TransformError	E0800	500	Error en política de transformació (<code>map</code> , <code>xslt</code>).
PropertyError	E0800	500	Error de propietat en <code>invoke</code> o <code>set-variable</code> .
RedactionError	E0800	500	Error en política de redacció de camps.

Finançat per



GOBIERNO DE ESPAÑA



Plan de Recuperación, Transformación y Resiliencia



Next Generation Catalunya



Generalitat de Catalunya

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 7 / 16

Nom d' excepció	errorCode resultant	httpCode	Origen típic
AssemblyRateLimitError	E0429	429	Límit de peticions per unitat de temps superat.
RuntimeError / default	E0500	500	Qualsevol altre tipus d' excepció no classificat.

Generació de la resposta per format

La funció createResponse despatxa sobre el valor de FormatoRespuesta i produeix el cos de la resposta juntament amb el Content-Type apropiat. Els quatre formats suportats són:

Format	Content-Type	Generador	Observacions
json	application/json	JSON.stringify(errorDevolver)	Per defecte. Format més utilitzat en el catàleg.
xml	application/xml	escribirErrorXML (badgerfish)	Si el cos té diverses claus arrel, s'embolica en <errorResponse>.
soap11	text/xml	escribirErrorSOAP(' 1.1')	Genera SOAP-ENV:Envelope amb faultcode QName. HTTP forçat a 500.
soap12	application/soap+xml	escribirErrorSOAP(' 1.2')	Genera env:Envelope con env:Code/Value. HTTP forçat a 500.

Detalls de la generació SOAP

Les respostes SOAP segueixen les regles següents, comunes a les dues versions de l' estàndard:

- El faultcode (SOAP 1.1) o env:Code/Value (SOAP 1.2) s'omple amb un QName vàlid: SOAP-ENV:Client per a errors HTTP 4xx i SOAP-ENV:Server per a errors HTTP 5xx. A SOAP 1.2 els valors equivalents són env:Sender i env:Receiver.
- El faultstring (1.1) o env:Reason/Text (1.2) inclou el moreInformation seguit de l'errorCode CTTI entre parèntesis, escapat XML per evitar injeccions.
- El detail inclou l'errorCode CTTI dins de l'element <errorCode>, també escapat XML.
- El statusCode HTTP de la resposta SOAP sempre és 500, independentment de l'errorCode resolt, per compatibilitat amb clients SOAP estrictes que esperen 500 davant de qualsevol fault.

2.1.3 Altres vistes i informació addicional

Integracions i dependències — Mòduls DataPower utilitzats

Mòdul	Mètode utilitzat	Finalitat
apim	getPolicyProperty(name)	Lectura de la propietat FormatoRespuesta declarada al YAML.
json-xml-converter	toXML(' badgerfish', obj)	Conversió de l' objecte JS d' error a XML per al format xml.
context	get(key) / set(key, val)	Lectura de error, request.verb, api.operation.path, properties; escriptura de l' estatus code i reason.
context.message	header.set / statusCode / reasonPhrase	Assignació del Content-Type, estatus code i reason de la resposta.
context.swagger	readAsJSON(callback)	Lectura asíncrona del swagger de l' API per resoldre x-customPaths.
session.output	write(body)	Esctura del cos de la resposta al flux de sortida.

Finançat per



GOBIERNO
DE ESPAÑA



Plan de
Recuperación,
Transformación
y Resiliencia



Next Generation
Catalunya



Generalitat
de Catalunya

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 8 / 16

Propietats consumides — FormatoRespuesta (propietat de la política)

Nom de la propietat	FormatoRespuesta
Tipus	string (enum)
Valors suportats	json, xml, soap11, soap12
Valor per defecte	json
Obligatorietat	Obligatòria (amb default aplicat)
Descripció	Determina el format del cos de la resposta, el Content-Type i, en cas de SOAP, força HTTP statusCode 500.
Origen	Es defineix com una propietat de la política i s'estableix en temps de disseny des de la configuració d'aquesta política en l'assembly de l'API, a través del desplegable disponible en el seu bloc de paràmetres.

Propietats consumides — erroresParticularesCtti (propietat d'API)

Nom de la propietat	erroresParticularesCtti
Tipus	string (JSON serialitzat)
Obligatorietat	Opcional
Descripció	Permet a una API afegir errors particulars al mapaErrores, o sobre escriure errors estàndard. El valor és un JSON amb la mateixa estructura de mapaErrores: { "E0NNN": { "httpCode": N, "httpMessage": "...", "moreInformation": "..." } }.
Origen	api.properties.erroresParticularesCtti en el swagger de l'API.
Exemple	{ "E0410": { "httpCode": 410, "httpMessage": "Gone", "moreInformation": "El recurs ja no està disponible" } }

Propietats consumides — x-customPaths del swagger (estructura custom)

Ruta al swagger	x-ibm-configuration.x-customPaths.<path>.<verb>.responses.<statusCode>.schema.\$ref
Tipus	JSON Pointer a una definició de x-customDefinitions
Obligatorietat	Opcional
Descripció	Permet a una API definir un esquema custom per al cos d'error. La política resol el \$ref i construeix un objecte amb els camps declarats, substituint els valors tipus "valor=XYZ" per la cadena XYZ i els valors que coincideixin amb camps de l'errorEstandar pel seu valor.
Ús típic	APIs SOAP que requereixen un fault personalitzat, o APIs que necessiten exposar errors en un format compatible amb un sistema legacy.

Dependències amb altres polítiques CTTI

La política és el destí final del catch de totes les APIs del catàleg. Les dependències són:

Política dependent	Tipus de dependència	Notes
ctti-validate-ip	Emissor (fort)	Emet errors controlats amb tipusError "controlled" i errorCodes E0400 / E0401 / E0500.
ctti-validate-request	Emissor (fort)	Emet errors controlats E0400 quan la validació de la petició falla.
ctti-validate-response	Emissor (fort)	Emet errors controlats E0406 quan la validació de la resposta falla.
ctti-header-arch	Emissor (feble)	Pot emetre errors controlats ocasionals.

Finançat per



 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001

Pàg. 9 / 16

Política dependent	Tipus de dependència	Notes
ctti-get-variables	Emissor (feble)	Pot emetre errors controlats E0500 si el fitxer de variables no es pot carregar.
ctti-invoke-log / ctti-custom-log	Cap	No emeten errors que arribin al catch.
global-policy-posterror	Cohabitant	Política global que s'executa després del catch; pot llegir les variables d' error sense modificar-les.

Gestió d'errors — Catàleg d'errors suportats

El mapaErrors conté els següents errors estàndard CTTI. Qualsevol errorCode no present en el mapa es resol sobre l'entrada E0XXX (httpCode 500, moreInformation genèric).

errorCode	httpCode	httpMessage	moreInformation
E0400	400	Bad Request	Validation error in the input fields
E0401	401	Unauthorized	You are not authorized to run the service
E0403	403	Forbidden	You do not have the appropriate privileges to run the service.
E0404	404	Not Found	Resource not found
E0405	405	Method Not Allowed	HTTP method not found
E0406	406	Not Acceptable	Validation error in the output fields
E0408	408	Request Timeout	Request timeout expired
E0409	409	Conflict	A conflict has been detected in the resource
E0413	413	Payload Too Large	The request body exceeds the allowed size
E0415	415	Unsupported Media Type	Unsupported content type
E0422	422	Unprocessable Entity	The request is syntactically valid but not processable
E0429	429	Too Many Requests	Too many requests have been detected for a resource
E0500	500	Internal Server Error	Internal Server Error
E0502	502	Bad Gateway	The server was unable to complete your request.
E0503	503	Service Unavailable	Service temporarily unavailable
E0600	500	Internal Server Error	Backend service error
E0800	500	Internal Server Error	User policy error
E0900	500	Internal Server Error	Backend connection error
E0XXX	500	Internal Server Error	Backend service error (fallback)

Gestió d'errors — Resolució de l'errorCode

La funció getErrorCode normalitza qualsevol codi d' entrada al format canònic E0NNN, essent robusta enfront dels següents casos:

Valor d' entrada	Valor normalitzat	Comportament
"E0400"	E0400	Sense canvi. El codi ja està en format canònic.
"400"	E0400	S'anteposa el prefix "E0".
"E400"	E0400	S'insereix el "0" després de l'E per produir el format canònic. Correcció respecte a v1, que produïa "E0E400".

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001 Pàg. 10 / 16

Valor d'entrada	Valor normalitzat	Comportament
undefined / null / ""	E0500	Es retorna DEFAULT_ERROR.
"abc"	E0abc	S'anteposa "E0"; el resultat no serà trobat en mapaErrores i caurà a E0XXX.

Gestió d'errors — Errors particulars per API

Una API pot declarar errors particulars mitjançant la propietat `erroresParticularesCtti`. La funció `addErrorMap` els integra en el `mapaErrores` abans de processar l'excepció: per a cada entrada del JSON, es normalitza la clau amb `getErrorCode` i s'assigna el valor (objecte amb `httpCode`, `httpMessage`, `moreInformation`) al `mapaErrores`. Això permet:

- Afegir errors que no estan en el catàleg estàndard (per exemple E0410 Gone per a APIs que ho necessitin).
- Sobreescrivre errors estàndard (per exemple redefinir el `moreInformation` d'E0404 per a una API concreta).
- Mantenir el contracte general del CTTI mentre es permet la flexibilitat necessària en casos particulars.

Gestió d'errors — Estructura custom del cos d'error

El cos d'error per defecte té quatre camps: `httpCode`, `httpMessage`, `errorCode` i `moreInformation`. Una API pot definir una estructura diferent declarant un schema en `x-ibm-configuration.x-customPaths.<path>.<verb>.responses.<code>.schema`, que la política resol mitjançant `montarJson` i aplica amb `estructuraError`. Les regles són:

- Cada propietat del schema amb un value que coincideixi amb un camp d'errorEstandar (`httpCode`, `httpMessage`, `errorCode`, `moreInformation`) s'omple amb aquest valor.
- Cada propietat la value de la qual tingui el prefix "valor=" s'omple amb el text que segueix al prefix (constant).
- Les propietats anidades i els arrays es processen.
- Si el `detalleExcepcion` és un objecte, els seus camps sobreescrivien els de l'errorEstandar abans d'aplicar l'estructura (comportament heretat de v1.0.0 conservat per compatibilitat).

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001 Pàg. 11 / 16

ANNEXOS

Annex A — Seguretat i controls

La política participa en la postura de seguretat del catàleg evitant la fuga d'informació tècnica al client i normalitzant les respostes d'error.

Controls aportats

- Normalització del missatge d'error: el client mai rep stack traces, missatges interns de DataPower o informació del backend més enllà del moreInformation declarat.
- Escapat XML del faultstring i del detail a SOAP, prevenint injecció XML en les respostes.
- Statusing coherent: el statusCode HTTP i el reasonPhrase s'assignen sempre a partir del mapaErrores, evitant inconsistències entre el cos i la capçalera.
- Limitació del catàleg d'excepcions: només els noms coneguts es mapegen a errors específics; la resta cau a E0500, evitant filtració accidental de tipus interns.
- Robustesa davant valors indefinits: getErrorCode tracta undefined/null/cadena buida com E0500, assegurant que la política mai produeixi una resposta invàlida.

Limitacions conegudes

- La política no implementa rate limiting d'errors ni detecta patrons d'explotació. Aquests controls s'han d'aportar per polítiques dedicades (rate-limit).
- El cos d'error conté errorCode CTTI, que és un identificador intern conegut. Si el client pot inferir informació sensible a partir del catàleg d'errorCodes, s'ha de fer servir el mecanisme d'estructura custom per emascarar-los.
- En format SOAP, totes les respostes tenen HTTP statusCode 500 per compatibilitat. Això pot dificultar el monitoring basat en codis HTTP; s'ha de complementar amb monitoring del cos (errorCode CTTI).
- La política no signa ni xifra el cos de la resposta d'error. Si la resposta ha de transitar per canals no fiables, s'ha de complementar amb TLS i/o signatura a nivell d'aplicació.

Annex B — Consideracions de rendiment

La política només s'executa quan es produeix una excepció, no en el flux nominal de cada petició. El seu cost computacional és:

- Lectura del swagger (context.swagger.readAsJSON): asíncrona, una lectura per invocació. DataPower manté el swagger en caixet.
- Parseu de l'exception.message (JSON.parse): O(L) sobre la longitud de l'string.
- Resolució de l'errorCode i construcció de l'objecte: O(1) més O(N) en el merge d'errors particulars (N = nombre d'entrades en erroresParticularesCtti, normalment < 10).
- Generació del cos: O(K) sobre la mida de l'objecte d'error (normalment 4-10 camps).
- Escripció a l'output: O(L) sobre la mida del cos serialitzat.

Recomanacions de configuració

- Mantenir erroresParticularesCtti acotat en nombre d'entrades (recomanat < 20). El merge es realitza en cada invocació.
- No abusar de x-customPaths amb estructures profundes i anidades: cada nivell de recursió multiplica el cost de montarJson.
- Per a APIs amb alt volum, monitoritzar el percentatge de respostes que passen pel catch i, si supera el 5%, investigar la causa arrel (probable bug en backend o en polítiques prèvies).

Annex C — Versionat i compatibilitat

Versions de DataPower suportades

Finançat per



GOBIERNO
DE ESPAÑA



Plan de
Recuperación,
Transformación
y Resiliencia



Next Generation
Catalunya



Generalitat
de Catalunya

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 12 / 16

Versió DataPower	Estat de compatibilitat	Notes
10.6.0.6	Compatible (versió actual de CTTI)	Validada amb la col·lecció Postman de tests funcionals desplegada al toolkit local.
10.6.0.9	Compatible (versió prevista post-upgrade)	Sense canvis esperats en el motor GatewayScript v2.0. Re-execució de la col·lecció Postman recomanada després de l'upgrade.
10.5.x	No suportada	Carència de funcionalitats del motor V8 utilitzades (notablement algunes opcions d'Object.keys amb objectes no plain).
v5 (legacy)	No suportada	Arquitectura completament diferent.

Política de versionat semàntic

La política segueix el versionat semàntic MAJOR. MINOR. PATCH:

- **MAJOR:** canvis incompatibles en el contracte (per exemple, nou format del cos d'error per defecte).
- **MINOR:** noves funcionalitats retrocompatibles (per exemple, nous formats de resposta).
- **PATCH:** correccions de bugs sense alteració del contracte.

La versió 2.0.0 és MAJOR respecte a 1.0.0 perquè, tot i que la majoria de canvis són compatibles, introdueix dos canvis visibles en la resposta: l'estructura del SOAP Fault és diferent (faultcode passa a ser QName) i l'HTTP statusCode de les respostes SOAP sempre és 500 (versus el codi real de l'error en v1.0.0).

Migració des de la versió 1.0.0

La migració requereix actualitzar la referència de la política en el swagger de cada API. Els consumidors han de tenir en compte els canvis visibles següents:

Aspecte	Versió 1.0.0	Versió 2.0.0
Formats suportats	json, xml, soap	json, xml, soap11, soap12 (soap com àlies retro-compatible de soap11)
faultcode SOAP 1.1	errorCode CTTI (no QName)	SOAP-ENV:Client (4xx) o SOAP-ENV:Server (5xx)
HTTP statusCode a SOAP	httpCode de l'error	Sempre 500
Escape XML en faultstring	No	Sí (escapeXml)
mapaErros: E0503	httpCode 500 (incorrecte)	httpCode 503 (correcte)
mapaErros: nous codis	N/A	E0408, E0413, E0415, E0422 afegits
Normalització errorCode "E400"	Produeix "E0E400" (bug)	Produeix "E0400" (correcte)
Robustez getErrorCode	Falla si errorCode undefined	Retorna DEFAULT_ERROR
statusCode emès a DataPower	Númeric vs string (inconsistent)	Assignat per context.set("message.status.code", N)

Els clients que només consumeixin respostes JSON o XML no es veuran afectats pels canvis de SOAP. Els clients SOAP han de ser revisats per confirmar que accepten el nou faultcode QName i el statusCode 500 fix.

Annex D — Traçabilitat amb codi i tests

Mapeig codi font – requisits tècnics

Finançat per



GOBIERNO
DE ESPAÑA



Plan de
Recuperación,
Transformación
y Resiliencia



Next Generation
Catalunya



Generalitat
de Catalunya

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)		N. revisió doc.: 1.0
	Disseny detallat de la solució		
	N. versió aplicació: 2.0.0	Build: 001	Pàg. 13 / 16

Logotip tècnic	Funció / Bloc en el codi	Test funcional (carpeta)	Postman
Catàleg d'errades estàndard	mapaErrores	Carpeta A (12 casos)	
Lectura propietat FormatoRespuesta	apim.getPolicyProperty normalizarFormato	+	Carpets A, C, D, E (cobertura per format)
Mapeig d'excepcions DataPower	initCapturaErrores (switch)		Carpeta H (10 casos)
Errors controlats (JavaScriptError)	errorPropio		Carpeta A (cobertura indirecta)
Errors particulars per API	addErrorMap		Carpeta F (2 casos)
Estructura custom del cos	estructuraError construirObjetoEstructura	+	Carpeta G (2 casos)
Generació JSON	createResponse case json		Carpeta A (12 casos)
Generació XML (badgerfish)	createResponse case xml escribirErrorXML	+	Carpeta A (12 casos)
Generació SOAP 1.1	createResponse case soap11 escribirErrorSOAP	+	Carpeta D, E (4 casos)
Generació SOAP 1.2	createResponse case soap12 escribirErrorSOAP	+	Carpeta D, E (4 casos)
Escape XML del faultstring	escapeXml		Carpets D, E (cobertura indirecta)
Resolució \$ref del swagger	montarJson		Carpeta G (cobertura indirecta)
Normalització errorCode "E400"	getErrorCode		Carpeta B (3 casos)
HTTP statusCode 500 a SOAP	flux principal (httpCodeFinal)		Carpets D, E (casos SOAP)

Col·lecció Postman de tests funcionals

La col·lecció Postman ctti-error-management compta amb 51 requests agrupades en 9 carpetes temàtiques i s'executa contra una API de prova real desplegada en el toolkit local de DataPower o en l'entorn real. No s'utilitzen shims ni simulacions de mòduls: els tests validen el comportament real de la política sobre el motor GatewayScript. El resultat esperat és la validació satisfactòria de tots els escenaris definits. L'execució pot llançar-se des de la interfície de Postman o mitjançant Newman (runner oficial de Postman per a línia de comandaments), facilitant la integració en pipelines CI/CD: `newman run ctti-error-management_2_0_0_tests.postman_collection.json -e entorno_toolkit_local.postman_environment.json --reporters cli,junit`.

La col·lecció Postman de proves està organitzada en **9 carpetes (A-I)** amb un total de **51 requests**. Les vuit primeres carpetes (A-H) contenen **proves unitàries** que validen el comportament de la política provocant els errors de forma forçada des d'un GatewayScript de test que executa `context.reject(<ErrorType>, '<missatge>')`, la qual cosa permet exercitar de forma determinista cadascuna de les branques del switch de l'error-handler.

La **carpeta I** conté **proves d'integració** que repliquen els mateixos tipus d'error de la carpeta H, però **provocats de forma natural per polítiques reals de l'assembly** en lloc de mitjançant `context.reject`. Així, el `TransformError` el genera una política map/xslt amb un mapping invàlid; el `ConnectionError` el genera una política invoke amb timeout o cap a un endpoint inabastable; l'`AssemblyRateLimitError` el genera la pròpia política rate-limit de l'assembly en superar el llindar configurat; i els `OperationError` es provoquen mitjançant invokes a `httpbin.org/status/<codi>` perquè el backend extern retorni un codi d'error HTTP determinat. Aquesta secció valida, per tant, el comportament end-to-end de ctti-error-management en escenaris de producció realistes.

Carpeta	Descripció	Nº requests
1	JSON: codis estàndard (E0400–E0503)	12

Finançat per



 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001 Pàg. 14 / 16

Carpeta	Descripció	Nº requests
2	JSON: casos límit (E9999, '400', 'E400')	3
3	XML (E0401, E0500, E0503, E0400)	4
4	SOAP 1.1 (Client/Server)	4
5	SOAP 1.2 (Sender/Receiver)	4
6	erroresParticularesCtti (custom + override)	2
7	estructuraErrorDevolver (custom schema)	2
8	Tipus d'error de l'assembly (switch)	10
9	Tests d'integració (errors reals)	10

Annex E — Configuració d'API consumidora i exemples de resposta

Configuració d'API consumidora (exemple)

Perquè una API utilitzi la nova versió de la política en el seu catch:

```
x-ibm-configuration:
  x-customPaths:
    /gestion-consulta:
      post:
        responses:
          '200':
            description: success
            schema:
              type: string
          '201':
            description: 201 Create
            schema:
              $ref: '#/x-ibm-configuration/x-customDefinitions/outputData'
          '400':
            description: 400 Bad Request
            schema:
              $ref: '#/x-ibm-configuration/x-customDefinitions/outputError'
  x-customDefinitions:
    inputData:
      type: object
      properties:
        descripcion:
          obligatorio: false
          type: string
          maxLength: 3200
        flag:
          obligatorio: true
          type: boolean
        estado:
          obligatorio: false
          type: string
          list: ['OK','KO']
        arrayDeObjetos:
          $ref: '#/x-ibm-configuration/x-customDefinitions/arrayDeObjetos'
        arrayPersona:
          obligatorio: true
          type: object
          arrayObject:
            documentName:
              obligatorio: true
              type: string
              minLength: 1
            documentApellido:
              obligatorio: true
              type: string
              minLength: 1
```


 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001
		Pàg. 15 / 16

```

    arraySimple:
      obligatorio: true
      array:
        obligatorio: false
        type: number
    arrayDeObjetos:
      type: array
      properties:
        descripcion:
          obligatorio: false
          type: string
          maxLength: 3200
        flag:
          obligatorio: true
          type: boolean
      properties:
        erroresParticularesCtti:
          value: '{"E0410":{"httpCode":410,"httpMessage":"Gone","moreInformation":"Recurso
eliminado"}}'
          encoded: false
    assembly:
      execute:
        # ... polítiques del flux nominal ...
      catch:
        - errors:
          - JavaScriptError
          - ConnectionError
          - BadRequestError
          - ValidateError
          - SOAPError
          - OperationError
          - TransformError
          - PropertyError
          - RedactionError
          - AssemblyRateLimitError
          - RuntimeError
      execute:
        - ctti-error-management:
          version: 2.0.0
          title: ctti-error-management
          FormatoRespuesta: json

```

Exemple de resposta JSON (FormatoRespuesta: json)

HTTP/1.1 401 Unauthorized
Content-Type: application/json

```

{
  "httpCode": 401,
  "httpMessage": "Unauthorized",
  "errorCode": "E0401",
  "moreInformation": "IP no válida"
}

```

Exemple de resposta SOAP 1.1 (FormatoRespuesta: soap11)

HTTP/1.1 500 Internal Server Error
Content-Type: text/xml

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>IP no válida (E0401)</faultstring>
      <detail><errorCode>E0401</errorCode></detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	ctti-error-management (Política de gestió d'errors)	N. revisió doc.: 1.0
	Disseny detallat de la solució	
	N. versió aplicació: 2.0.0	Build: 001 Pàg. 16 / 16

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Exemple de resposta SOAP 1.2 (FormatoRespuesta: soap12)

HTTP/1.1 500 Internal Server Error
Content-Type: application/soap+xml

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <env:Fault>
      <env:Code><env:Value>env:Sender</env:Value></env:Code>
      <env:Reason><env:Text>IP no válida (E0401)</env:Text></env:Reason>
      <env:Detail><errorCode>E0401</errorCode></env:Detail>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Annex F — Parametrització i rols

No aplica a aquesta política: ctti-error-management és un component de gateway que no defineix transaccions, rols ni taules de parametrització pròpies. Es deixa indicat com a pendent / no aplicable.

Transaccions	Descripció
(No aplica)	(No aplica)

Rols	Descripció
(No aplica)	(No aplica)

Taules de parametrització	Descripció
(No aplica)	(No aplica)