

Guia de bones pràctiques

Desenvolupament a Celonis

Última actualització desembre 2025

Finançat per



Unió Europea
Fons Europeu
Next Generation



GOBIERNO
DE ESPAÑA



**Plan de Recuperación,
Transformación
y Resiliencia**



**Next Generation
Catalunya**

C. del Foc, 57, edifici B
08038 Barcelona
Tel. 93 857 40 00

Finançat per



**Unió Europea
Fons Europeu
Next Generation**



GOBIERNO
DE ESPAÑA



**Plan de Recuperación,
Transformación
y Resiliencia**



**Next Generation
Catalunya**

Índex

1.	Introducció i propòsit	4
2.	Fase de preparació del projecte	4
3.	Fase de execució del projecte	5
3.1	Detalls a tenir en compte	5
3.2	Nomenclatura i altres aspectes d'Arquitectura	5
3.3	Aspectes relacionats amb l'extracció de dades	6
3.4	Aspectes relacionats amb la transformació de dades	7
3.5	Aspectes relacionats amb la carrega del model de dades	8
3.5.1	Aspectes relacionats amb la planificació de Tasks	8
3.6	Aspectes relacionats amb indicadors o dashboards: Anàlisi i vistes	9
3.7	Aspectes relacionats amb Action Flows (AFs)	9
3.7.1	Paquets a utilitzar directament pels usuaris (sense personalitzacions)	10
3.7.2	Paquets a utilitzar directament per equips de desenvolupament (amb personalitzacions)	10
4.	Evolutius	11
5.	Projecte en ús	11
5.1	Back (Data Integration)	11
5.2	Front (Studio) Equips de desenvolupament	12

1. Introducció i propòsit

L'objectiu d'aquest document és guiar els equips de projecte que desenvoluparan una solució de **Process Mining a Celonis** sobre processos de negoci, en tots els aspectes tècnics que han de complir. Es pretén que tant el traspàs com el posterior manteniment siguin tan àgils i senzills com sigui possible.

El document està dividit en les següents seccions:

- **Fase de preparació del projecte**
- **Aspectes a tenir en compte en la fase d'execució del projecte**
 - Nomenclatura a utilitzar i altres aspectes d'Arquitectura
 - Aspectes relacionats amb l'extracció de dades
 - Aspectes relacionats amb la transformació de dades
 - Aspectes relacionats amb la càrrega del model de dades
 - Aspectes relacionats amb indicadors: anàlisi i vistes
 - Aspectes relacionats amb *Action Flows*

2. Fase de preparació del projecte

Després de la presentació de l'equip de desenvolupament i el seu interlocutor, **el Responsable de la Plataforma** crearà per al projecte:

- **Dos grups d'usuaris:** *Analyst* i *Member*.
- **Els usuaris de l'equip de desenvolupament** s'inclouran dins del grup *Analyst*.
- **Els usuaris de negoci** s'inclouran als grups *Analyst* o *Member*, segons ho sol·liciti el responsable del projecte.
- **Un *Data Pool*** per al desenvolupament de la part "back" (ETL). Aquest Data Pool tindrà el sufix "_DES".
- **Dos *Spaces***, PRE i PRO, per al desenvolupament de la part "front".

Tant als espais com també als paquets, hi haurà una organització clara en carpetes per permetre la resta de divisions necessàries.

3. Fase de execució del projecte

Un cop el projecte es doni per iniciat, els **Spaces** i **Data Pools** seran responsabilitat de l'equip de desenvolupament fins que s'acosti la posada en producció del projecte. En aquest moment es començaran a **revocar permisos** a l'equip de desenvolupament, que passaran a ser assumits per el Responsable de la Plataforma.

Si l'equip necessita concedir algun permís internament, sempre s'haurà de fer **a nivell de grup**, mai a nivell d'usuari.

Durant el desenvolupament, els equips podran crear i esborrar paquets de prova, però hauran d'utilitzar els **paquets creats per** Responsable de la Plataforma per desenvolupar el projecte, realitzar les proves necessàries. Sobre aquests paquets, Responsable de la Plataforma restringirà els permisos **DELETE PACKAGE** i **MANAGE PERMISSIONS**.

3.1 Detalls a tenir en compte

- Durant el traspàs de PRE a PRO, s'han de desactivar tant els *Action Flow* com les *Skills*, de manera que hi hagi dos entorns funcionals simultanis que puguin generar duplicats; serà un "moment cec" que ha de durar el mínim temps possible.
- Que la documentació estigui actualitzada és una condició necessària perquè un procés passi a l'espai productiu.
- És important que, un cop finalitzada una etapa, es netegi l'entorn; cal revisar que totes les peces que componen el procés siguin útils i que no formin part de proves antigues.
- Cada vegada que hi hagi canvis significatius en els paquets, s'ha de publicar una nova revisió, per tal de poder utilitzar el control de versions per tornar enrere davant un error, pèrdua de codi, etc.

3.2 Nomenclatura i altres aspectes d'Arquitectura

A l'hora de crear objectes, s'ha de seguir una nomenclatura clara i coherent al llarg de tot el projecte.

En iniciar un nou projecte, és necessari tenir en compte els requeriments de sistemes de Celonis.

Sempre s'han de consultar les darreres recomanacions de Celonis:

<https://docs.celonis.com/en/system-requirements-of-an-on-premise-extractor-server.html>

<https://docs.celonis.com/en/on-prem-client-system-requirements.html>

3.3 Aspectes relacionats amb l'extracció de dades

S'han de tenir en compte les bones pràctiques indicades a la documentació de Celonis:

<https://docs.celonis.com/en/extraction-task-best-practice.html>

En especial, cal considerar les següents:

- Extreure **només les columnes necessàries**.
- Acordar amb l'usuari l'**històric de dades estrictament necessari** per a les funcionalitats requerides.
- Si s'han de tenir més de **20 activitats**, cal comunicar-ho a l'equip responsable.
- Plantejar mantenir **models separats** per a funcionalitats analítiques i transaccionals si conviuen en el projecte.
- Per defecte, totes les **extraccions diàries han de ser incrementals**.
- Evitar al màxim les **extraccions completes** de taules, tret de les petites o amb baixa actualització.
- Implementar extraccions **incrementals per data + hora** a les taules que tinguin aquestes columnes.
- No utilitzar **OR sobre el mateix atribut**; cal agrupar tots els valors possibles en una sola condició (per exemple, BLART).
- Assegurar-se que s'ha **desactivat "Use Change Log"**, ja que Celonis l'activa per defecte quan la connexió inclou Change Logs.
- Posar **només els filtres necessaris**; massa filtres poden impedir l'ús correcte d'índexs.
- Si un Data Job (DJ) s'utilitza per extraccions Delta, **no afegir filtres de Full** "per si de cas". Si cal una Full, crear un DJ diferent.
- Per eficiència, els filtres Delta s'han de fer amb **FIND_MAX/FIND_MIN** sobre les columnes representatives, no amb *offsets*.

En general, cal dedicar temps a l'optimització de les extraccions, amb un doble objectiu:

- **Reduir el temps de càrrega**
- **Evitar carregar el sistema origen més del necessari**

- **Això vol dir intentar llançar el mínim nombre d'extraccions possible.**
- Recomanacions concretes basades en l'experiència:
 - o **S'evitarà la creació de paràmetres "Text"** en Celonis, excepte quan sigui evident que és un valor únic sense possibilitat de convertir-se en llista.
 - o Per a la resta de paràmetres de text, sempre s'utilitzarà **"Text List"**, encara que actualment només es necessiti un valor.
 - o Per exemple, si l'usuari sol·licita carregar només documents del tipus **"RV"**, es crearà un **Text List** amb aquest únic valor.
 - o Així, si més endavant es necessiten més tipus de document, només caldrà afegir-los a la llista.

3.4 Aspectes relacionats amb la transformació de dades

Les transformacions hauran d'estar el més optimitzades possible, parant especial atenció als apartats següents:

- S'utilitzaran taules en lloc de vistes, excepte la final que s'hagi de pujar al model.
- Els camps clau de cada taula s'hauran de col·locar al principi.
- Es recopilaran estadístiques de totes les taules creades en les transformacions.
- S'utilitzarà EXPLAIN per comprovar que s'utilitzen les estadístiques i que els JOINS són el més eficients possible.
- S'evitarà l'ús de DISTINCT, havent de justificar el seu ús en cada cas, ja que si hi ha un bon control sobre les dades presents a les taules, en general es podrien evitar els duplicats.

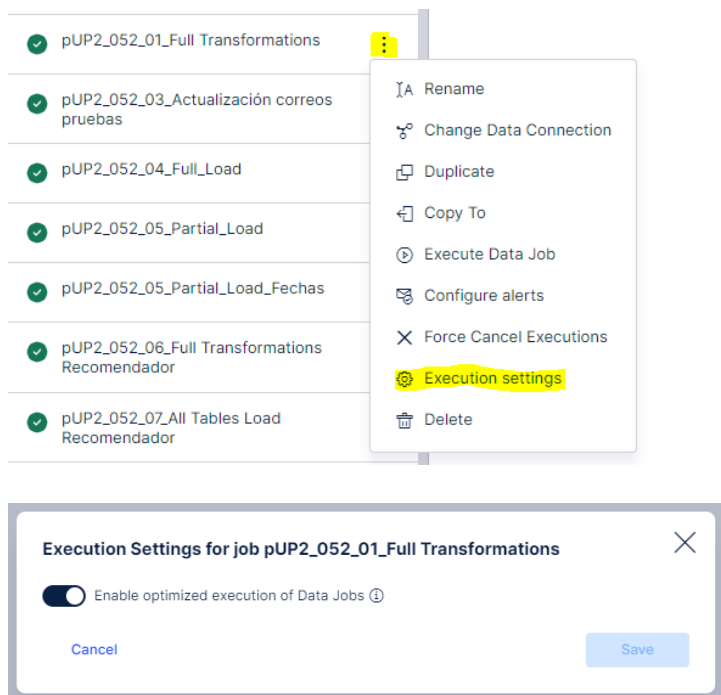
S'adjunten enllaços a les bones pràctiques recomanades per Celonis:

<https://docs.celonis.com/en/transformation-task-best-practice.html>

<https://docs.celonis.com/en/sql-best-practice-checklist.html>

<https://docs.celonis.com/en/vertica-sql-performance-optimization-guide.html>

Es recomana activar el “Smart ETL”, que mitjançant un algoritme intern intenta paral·lelitzar automàticament les transformacions. Per a això, l’habilitarem a les “Execution settings” de cada DJ de transformacions:



Cal plantejar-se les tasques que s’han de fer quan es canviï d’any. Cal modificar algun paràmetre? S’intentarà que les tasques siguin automàtiques en la mesura del possible i les que no ho siguin s’hauran de traspasar documentades a l’equip de suport.

3.5 Aspectes relacionats amb la carrega del model de dades

Els models s’entregaran, en la mesura del possible, sense *warnings* en el procés de càrrega. Per a això, les columnes hauran de tenir les tipologies correctes, la columna EVENTTIME haurà de contenir valors dins del rang admès, etc.

És imprescindible acordar amb l’usuari una hora màxima de finalització de la càrrega dels Data Models. Per defecte, l’hora límit serà les 08:00.

3.5.1 Aspectes relacionats amb la planificació de Tasks

En general, la finestra *batch* s’estableix entre les 20:00 i les 08:00 del dia següent, i s’ha d’evitar l’execució de tasques pesades fora d’aquest horari per no perjudicar el treball dels usuaris. En qualsevol cas, s’hauran de consensuar amb l’usuari de negoci.

3.6 Aspectes relacionats amb indicadors o dashboards: Anàlisi i vistes

- A l'inici del projecte s'acordarà amb el responsable del projecte l'idioma en què s'expressaran els textos dels *dashboards*. Per defecte, s'utilitzarà el català.
- Per defecte, les anàlisis no es connectaran directament amb un Data Model sinó amb un Knowledge Model. D'aquesta manera, si en el futur cal ajustar un KPI que està present en 3-4 anàlisis, vistes i/o Action Flows, només caldrà modificar-lo una vegada. Igualment, si cal definir un nou KPI que s'apliqui a diversos elements, només s'hauria de crear en un únic lloc. L'objectiu és concentrar en el Knowledge Model tot el codi que sigui possible.

3.7 Aspectes relacionats amb Action Flows (AFs)

A l'hora de desenvolupar AFs des de zero, tingueu en compte que Celonis disposa de molts casos d'ús ja plantejats que poden accelerar el desenvolupament:

<https://docs.celonis.com/en/action-flow-templates.html>

Els AFs hauran de ser reprocesables davant fallades, preferiblement de manera automàtica. És a dir, caldrà incloure mòduls de reprocesament en les activitats finals del flux, de manera que si fallen, Celonis intenti reprocesar i finalitzar correctament almenys una vegada. En cas que també falli el reprocesament automàtic, el disseny del flux estarà preparat perquè es pugui relançar sense repetir el que ja s'hagi completat correctament, processant en aquest segon intent només allò que hagi quedat pendent.

Els AFs tenen un límit de temps d'execució de 60 minuts (MAXIMUM EXECUTION TIMEOUT). Els límits es poden consultar a la documentació:

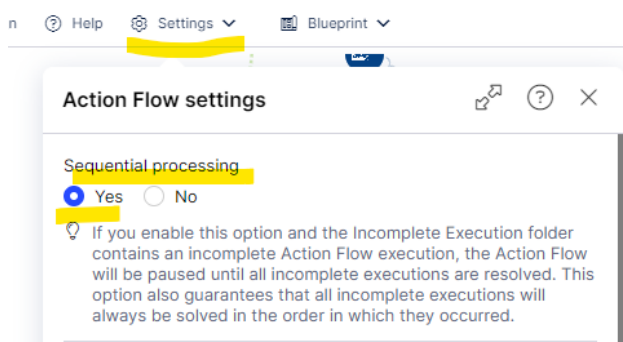
<https://docs.celonis.com/en/action-flows-terms-and-limitations.html>

Un cop superat aquest temps, l'AF retorna un *warning* de tipus 'ExecutionInterruptedError' i s'atura l'execució. Per això no s'han de desenvolupar processos massius que puguin apropar-se a aquests temps. Per al cas concret de l'enviament de correu existeix una arquitectura recomanada, que consisteix a fer un primer AF que calculi tots els correus que s'han d'enviar, i després fer una trucada a un altre AF que faci el tractament per a un únic registre. Amb aquesta arquitectura s'evita el problema del *timeout* i és més senzill reprocesar només l'execució que hagi pogut fallar, en lloc d'haver de començar des del principi.

En el cas que sigui necessari que un AF gravi dades en taules del Backend, per exemple per deixar registre que s'ha enviat, en quina data, etc., no s'utilitzarà l'Skill "Write data to table" excepte si són execucions molt puntuals. Si aquestes gravacions formen part d'un procés massiu, com un enviament de correus, per exemple, s'haurà d'utilitzar un AF que acumuli les gravacions i les envii per paquets, en lloc de fer-ho una a una, com faria l'Skill.

Per accedir a dades d'un Data Model es recomana utilitzar el mòdul Get Rows, en lloc de Query Data, ja que el primer permet accedir a objectes d'un Knowledge Model, podent reutilitzar les fórmules i filtres definits en aquest, evitant haver de repetir codi que després caldria modificar de manera coordinada davant un evolutiu o correctiu. A més, el mòdul Query Data aviat quedarà obsolet.

Quan l'AF truca a una *skill* amb un mòdul de 'Write Data to Table', aquesta no permet treballs en paral·lel, per la qual cosa l'AF s'ha de configurar per executar-se en seqüencial. Des de l'opció 'Sequential processing'.



No en totes les situacions aplica aquesta solució. Per això també cal tenir en compte la configuració de la crida al Wd2T a la pròpia *skill*:

L'acció de **Write Data To Table** (escriptura de dades a una taula interna) permet 2 opcions de configuració: 1) realitzar l'acció a través d'una *application key* o 2) a través d'una connexió d'usuari.

La configuració amb *application key* permet un tractament massiu de dades molt més eficient.

3.7.1 Paquets a utilitzar directament pels usuaris (sense personalitzacions)

Aquests paquets es descarregaran en un espai concret per tenir-los identificats. **El Responsable de la Plataforma** només donarà suport en la creació de permisos sobre els paquets, però la resta l'haurà de mantenir l'usuari responsable de la seva instal·lació.

3.7.2 Paquets a utilitzar directament per equips de desenvolupament (amb personalitzacions)

Aquest cas es tractarà com un desenvolupament normal, per al qual, un cop es disposi de l'OK de l'usuari, caldrà fer un traspàs a Plataforma perquè s'encarregui del manteniment.

És imprescindible que l'equip de desenvolupament mantingui un document amb la llista exhaustiva de les modificacions realitzades sobre el paquet estàndard, de manera que davant un canvi de versió sigui el més senzill possible tornar a deixar la nova versió com estava l'anterior.

4. Evolutius

Els evolutius es documentaran en el mateix document del projecte, utilitzant el control de versions i fent referència a l'evolutiu, deixant clarament identificat què pertany a l'evolutiu, el codi que s'ha afegit i/o modificat.

Això aplica tant per als desenvolupaments del *back* com del *front*.

5. Projecte en ús

Un cop el projecte està en ús normal per part dels usuaris, l'equip de desenvolupament ja no tindrà accés a modificar cap element del projecte en productiu.

Es contemplen diferents procediments per a la part Back (Data integration) i Front (Studio).

5.1 Back (Data Integration)

Per a qualsevol modificació en els Data Pools de producció se seguirà el procediment següent:

Si és un desenvolupament sobre un Data Pool ja existent a PRO:

1. Quan l'equip de desenvolupament hagi de fer un pas a producció, acordarà internament amb les diferents línies de desenvolupament el millor moment per fer la versió del Data Pool (DP) “_PRO”.
2. Es duplicarà el Data Pool “_PRO”, canviant el sufix per “_PRE”, substituint el que hi hagués creat anteriorment de PRE.
3. Es copiaran tots els canvis corresponents del DP “_DES” al “_PRE”.
4. A PRE s'esborren tots els elements de desenvolupament que no han de passar a PRO i es fa una prova total:
 - Extraccions.
 - Transformacions i Càrregues.
5. Si les proves són correctes:
 - Es copien/modifiquen els objectes o transformacions corresponents a “_PRO”.
 - Si després hi hagués alguna incidència a PRO, es faria un *rollback* a la versió anterior.

Si és un desenvolupament sobre un Data Pool que encara no existeix a PRO:

1. Es duplicarà el Data Pool “_DES”, canviant el sufix per “_PRE”.
2. A PRE s'esborren tots els elements de desenvolupament que no han de passar a PRO i es fa una prova total:
 - a. Extraccions.
 - b. Transformacions i Càrregues.
3. Es duplicarà el Data Pool “_PRE”, canviant el sufix per “_PRO”.

5.2 Front (Studio)

En funció del tipus d'element les accions seran diferents:

- Anàlisis. Es copiarà l'anàlisi des del paquet origen i es modificarà la variable del DM.
- KM i Vistes. Copiant el KM i actualitzant la variable del DM, haurien de funcionar les vistes.
- AFs i Skills. La primera vegada que es passi a producció un element es farà una còpia i caldrà tornar-lo a configurar perquè funcioni en el nou entorn. Les modificacions següents es faran directament sobre l'element, sense copiar-lo des de desenvolupament, per evitar haver-lo de configurar de nou.



Generalitat de Catalunya

Imatge 1: Imatge de la Generalitat de Catalunya