

**Jordi Jiménez Orrios**

**Configuració, integració i posta en marxa d'un PILOT de bessó digital  
(Digital Twin) al Consorci d'Aigües de Tarragona**

**TREBALL DE FI DE GRAU**

**dirigit per Sr. Xavier Vilanova**

**Grau de Enginyeria Electrònica Industrial i Automàtica**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2021**

# 1 Índex.

1	Índex.....	2
2	Memòria Descriptiva.....	5
2.1	Objectiu del TFG.....	5
2.2	Justificació del TFG.....	5
2.3	Abast del TFG.....	5
2.4	Concepte del Bessó digital.....	6
2.5	El Consorci d'Aigües de Tarragona.....	8
2.6	Arquitectura de control i monitorització del CAT.....	9
2.6.1	<i>Xarxa de distribució. Front End, concentradores i remotes.</i> .....	10
2.6.2	<i>ETAP</i> .....	12
2.6.3	<i>Analitzadors i altres equips</i> .....	12
2.6.4	<i>SCADA</i> .....	13
2.6.5	<i>Bases de dades</i> .....	14
2.7	Implementació.....	15
2.7.1	<i>Arquitectura inicial</i> .....	15
2.7.2	<i>Estudi de l'arquitectura desitjada</i> .....	16
2.7.2.1	<i>Simulador hidràulic</i> .....	16
2.7.2.2	<i>Connectivitat OpenFlows WaterSight amb CitectSCADA</i> .....	20
2.7.3	<i>Arquitectura desitjada</i> .....	23
3	Memòria de Càlcul.....	25
3.1	OpenFlows WaterSight.....	25
3.2	Base de dades del Historitzador.....	25
3.2.1	<i>Estudi i disseny de les taules del Historitzador</i> .....	25
3.2.2	<i>Qualitat de les dades</i> .....	32
3.2.2.1	<i>Taules de configuració dels tags</i> .....	33
3.2.2.2	<i>Taules de dades dels tags</i> .....	38
3.2.2.3	<i>Taules de dades manuals</i> .....	42
3.2.2.4	<i>Taules de valors calculats</i> .....	44
3.3	Implementació del registre dades desde CitectSCADA.....	45
3.3.1	<i>Historitzador</i> .....	45
3.3.2	<i>Registre de dades Historian en servidors CitectSCADA</i> .....	45
3.4	Implementació registre dades en SQL Server.....	52
3.4.1	<i>Tasca Ct_Manteniment sentencies scada01/02 + inet</i> .....	53
3.4.2	<i>Tasca Ct_Sincronitzar sentencies scada01/02 + execució inet</i> .....	54
3.5	Diagrama funcional de procés.....	55

3.6	Tasques de comprovació del registre historitzador .....	56
4	Plànols.....	58
4.1	Localització de la ETAP del CAT.....	58
5	Pressupost.....	60
5.1	Equips servidors Citect SCADA, SQL Server i OpenFlows WaterSight. ....	60
5.2	Equips OpenFlows WaterSight by Bentley.....	62
5.3	Equips clients .....	63
5.4	Programació i posta en marxa del bessó digital .....	64
5.5	Pressupost total.....	64
6	Plec de condicions.....	65
6.1	Objectiu .....	65
6.2	Materials.....	65
6.3	Condicions Generals .....	65
6.4	Condicions Econòmiques.....	65
6.5	Condiciones de Seguretat .....	65
7	Annexes.....	66
7.1	Codi de la implementació de l'historitzador en CitectSCADA.....	66
7.1.1	<i>HIST_Evt_GuardaLecturesSQL</i> .....	66
7.1.2	<i>_HIST_GuardaLecturesSQL</i> .....	67
7.1.3	<i>HIST_GuardarSentenciaSQL</i> .....	71
7.1.4	<i>HIST_ObrirConnexioSQL</i> .....	72
7.1.5	<i>HIST_TancarConnexioSQL</i> .....	73
7.1.6	<i>TRN_LeerValorTrend</i> .....	74
7.1.7	<i>CTSubs_ValorTag</i> .....	75
7.1.8	<i>HIST_ExecutarSentenciaSQL</i> .....	77
7.1.9	<i>Funcions auxiliars</i> .....	78
7.1.9.1	<i>PS_GetActive_RptSvr</i> .....	78
7.1.9.2	<i>PS_Prog_Log</i> .....	79
7.1.9.3	<i>Email_RunEnviarEmail</i> .....	81
7.2	Codi de la implementació l'historitzador al SQL Server .....	82
7.2.1	<i>Codi d'inserció i tractament de dades</i> .....	82
7.2.1.1	<i>sp_Insert_Sentencia</i> .....	82
7.2.1.2	<i>sp_sync_Ct_Sentencias</i> .....	83
7.2.1.3	<i>sp_exec_Ct_Sentencias</i> .....	85
7.2.1.4	<i>sp_exec_Ct_Sentencia</i> .....	86
7.2.1.5	<i>HIST_sp_AfegirDada</i> .....	87

7.2.2	<i>Codi de manteniment de les bases de dades.....</i>	90
7.2.2.1	<i>sp_Mant_Sentencies_scada01_02.....</i>	90
7.2.2.2	<i>sp_mant_Ct_Sentencies.....</i>	91
7.2.3	<i>Codi de comprovació de la integritat de les taules del Historian.....</i>	93
7.2.3.1	<i>sp_Hist_CheckDiario_D.....</i>	93
7.2.3.2	<i>sp_Hist_CheckDiario_I.....</i>	96
7.2.3.3	<i>sp_Hist_CheckDiario_R.....</i>	99
8	Webgrafia.....	102

## **2 Memòria Descriptiva.**

### **2.1 Objectiu del TFG**

L'objectiu del TFG és la configuració, integració i posta en marxa d'un PILOT de bessó digital (Digital Twin) al Consorci d'Aigües de Tarragona.

El sistema de control actual es limita a una arquitectura clàssica servidor-client, en la que el SCADA es comunica amb tots els PLCs/dispositius de camp per tal de monitoritzar i controlar els processos que componen la instal·lació.

L'objectiu és incorporar al sistema un simulador hidràulic, OpenFlows WaterSight, que reproduïx el comportament de la nostra instal·lació, el qual obtindrà les dades actuals i històriques d'una base de dades SQL mitjançant la implementació d'un sistema historitzador.

L'historitzador serà gestionat pel SCADA, CitectSCADA v7.40 spB, el qual s'encarregarà de deixar les dades necessàries en tota la estructura de taules creades a un servidor Microsoft SQL Server 2012.

### **2.2 Justificació del TFG**

El Consorci d'Aigües de Tarragona té la necessitat de realitzar la implementació d'un bessó digital per tal de continuar amb el procés de digitalització necessària per una instal·lació crítica com la nostra.

Un bessó digital ens permetrà en aquest pilot poder prevenir de malfuncionaments de cabalímetres o qualsevol equip de la instal·lació gràcies a la execució conjunta del temps real i la simulació.

La funcionalitat de poder fer córrer el model simulat amb dades antigues del historitzador també permet poder analitzar situacions anòmales o l'estudi de moments concrets de la instal·lació.

Això ens permet disposar d'una eina molt poderosa per la presa de decisions.

### **2.3 Abast del TFG**

L'abast del projecte és tenir definits totes els equips, estructures i programació necessària per a que el bessó digital esdevingui una realitat a la nostra instal·lació i ens serveixi com a punt de partida per a futures implementacions i millores.

Això ens implicarà configurar, programar i tenir connectivitat bidireccional entre els tres aplicatius en que es recolza el nostre bessó digital:

- el simulador hidràulic OpenFlows WaterSight.
- el SCADA CitectSCADA.
- La base de dades Microsoft SQL Server.

## 2.4 Concepte del Bessó digital

Tota l'automatització de processos en els últims 10 anys ha sofert tota una sèrie de canvis que han provocat més que una evolució, una revolució, en quant que els límits del que abastava històricament el control de qualsevol procés productiu s'han vist modificats de manera notable.

Aquests canvis han vingut donats per diferents raons:

- Notables millores de les famílies de PLCs. Els autòmats programables han deixat de ser equips que controlen una sèrie d'entrades i sortides, per ser equips fàcilment programables i escalables, deixant de ser un element limitador en qualsevol arquitectura.
- Gran varietat de protocols de comunicacions, fàcilment integrables, que permeten interconnexionar multitud d'equips de diferents fabricants sense problemes.
- En conseqüència comunicacions més eficients, robustes i segures donat que també la seguretat ha esdevingut un tema clau donada la connectivitat global (Internet, 3G, 4G...)

Tot plegat ha provocat una convergència dels espais de OT i IT i han fet que no puguem ja parlar de sistemes SCADA orientats només al control i visualització del procés.

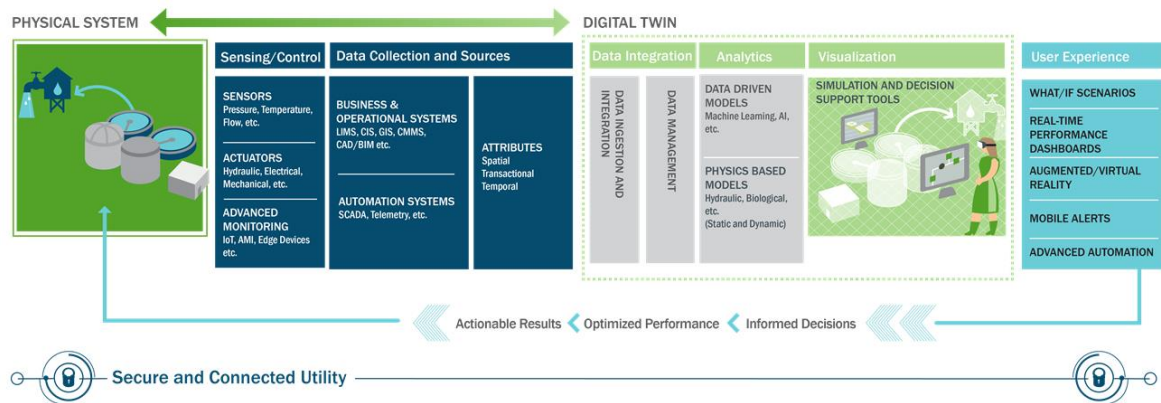
Les necessitats en l'àrea de producció impliquen tenir eines per la optimització i per a poder preveure els comportaments en funció de demanda, del cost energètic, de l'ús de les matèries emprades...

En el àrea de manteniment hem de disposar d'eines preventives dels equips per tal de millorar el seu rendiment, poder preveure les possibles fallades abans que es produeixin per tal de minimitzar els costos derivats de les parades.

Aquesta realitat és la raó per la qual neix la necessitat de la implementació d'un bessó digital en qualsevol procés industrial, per tenir un punt de partida robust per a poder créixer en aquesta direcció.

El bessó digital és una tecnologia disruptiva que proporciona una representació virtual / digital tant dels elements como de la dinàmica d'una planta o sistema. Si s'implementa correctament, un bessó digital pot influir en el disseny, construcció y operació del sistema a al llarg del seu cicle de vida i ajudar a optimitzar la operació a través de informació que ens proporciona. És un model de software dinàmic (model hidràulic + Machine learning) del sistema basat en calibració continua en temps real.

Com veiem el model del bessó digital és composta de una capa física i una virtual:



Il·lustració 1 Arquitectura Bessó digital

La capa física és la que fins ara ha esdevingut l'estàndard en control i monitorització de processos, els sensors i actuadors de la instal·lació són modelats en temps real al sistema SCADA mitjançant PLCs.

El model de la planta ha de ser ajustat al nostre procés, per aquesta raó la importància del machine learning. Simular un sistema amb 30000 variables p.ex. de qualsevol naturalesa física de producció (tractament d'aigües, processos químics, línies de producció...) és un procés costós en temps i recursos. Una primera implementació de tots els paràmetres de la instal·lació potser només ens assegura una fiabilitat de la simulació en un 20-30%.

El bessó digital ha de ser una eina per tenir una retroalimentació al sistema simulat desde el sistema físic per tal de que ens ajudi a ajustar tots els paràmetres de la virtualització per tal que el nostre model s'ajusti realment a la realitat que volem representar.

Una vegada la simulació estigui ajustada a la realitat ja ens podem endinsar en les eines de simulació i suport a les presses de decisions.

Aquest és el punt de partida en el que volem situar el nostre sistema actual.

## 2.5 El Consorci d'Aigües de Tarragona

El Consorci d'Aigües de Tarragona (CAT) és un ens que va néixer per la necessitat històrica d'aigua a la província de Tarragona.

El Consorci d'Aigües de Tarragona és un organisme amb personalitat jurídica pròpia sense afany de lucre. Es va crear l'any 1981 a través de la Llei 18-81 d'actuacions en matèria d'aigües a Tarragona. A partir de l'aprovació d'aquesta llei, l'any 1985 es publicaven al DOGC 593, 27/09/1985 els Estatuts pels quals es regeix el CAT on consten els òrgans de govern propis així com el seu règim jurídic i financer. Un d'aquests òrgans de govern és el Consell d'Administració que es regeix pel seu propi reglament.

Per tal de regular el funcionament intern del Consorci d'Aigües de Tarragona es van redactar les Normes d'explotació, les quals són revisades i aprovades per l'Assemblea General d'usuaris i un cop aprovades per aquesta, són d'obligat compliment per a tots els membres de l'ens.

L'activitat està regulada per la Llei 18/81, que concedeix una autorització administrativa al CAT per fer servir un màxim de 4 m<sup>3</sup>/s d'aigua procedent dels canals dels marges esquerre i dret del riu Ebre.

Un cop captada dels canals, el CAT sotmet l'aigua a un procés de potabilització, en controla la qualitat i finalment la distribueix als consorciats tenint en compte criteris d'eficiència, sostenibilitat i responsabilitat social. La prioritat és la garantia del subministrament: posar a l'abast dels consorciats un recurs segur i de qualitat, ajustat al cost i adequat als principis de la Directiva Marc de l'Aigua, preservant el Medi Ambient i fomentant el retorn al territori. Actualment el formen un total de 63 ajuntaments i 25 indústries.

A dia d'avui la xarxa s'estén al llarg de 402 km de canonades i disposa de 23 estacions de bombament.

Per tal d'abastir a tots els consorciats, diferenciem tres àrees productives ben diferenciades:

- Producció

Compren des de la captació de l'aigua als canals del marge dret i esquerre a Campredó fins a la planta de tractament a L'Ampolla (ETAP). Una vegada l'aigua s'ha sotmès al procés de tractament, és emmagatzemada al Dipòsit d'Aigua Neta (DAN).

- Distribució

La distribució s'inicia a la estació de bombament PAB01 que es troba a la ETAP.

La distribució compren des de aquest punt fins a tots els punts finals de subministre a ajuntaments e indústries de la nostra província de Tarragona.

Hi han 23 estacions de bombament repartides per tot el territori, així mateix es disposen de dipòsits reguladors i finalment els dipòsits de subministre als consorciats.

- Sistema d'alerta

Anomenem sistema d'alerta a tots els analitzadors i equips de mesura de la qualitat de l'aigua que es troben repartits per tota la instal·lació, producció i distribució.

Ens permeten tenir informació en temps real de la qualitat de l'aigua al llarg de tot el procés de producció i distribució.





*Il·lustració 2 Xarxa de producció i distribució del CAT*

## 2.6 Arquitectura de control i monitorització del CAT

El CAT disposa d'un sistema SCADA redundat que obté les dades en temps real de la planta de tractament, tota la xarxa de distribució i tots els equips de mesura de qualitat. El sistema de control es troba a la ETAP a L'Ampolla.

Actualment el SCADA del CAT interroga 50246 tags.

El temps de refresc de les dades al SCADA està fixat a 4000 ms.

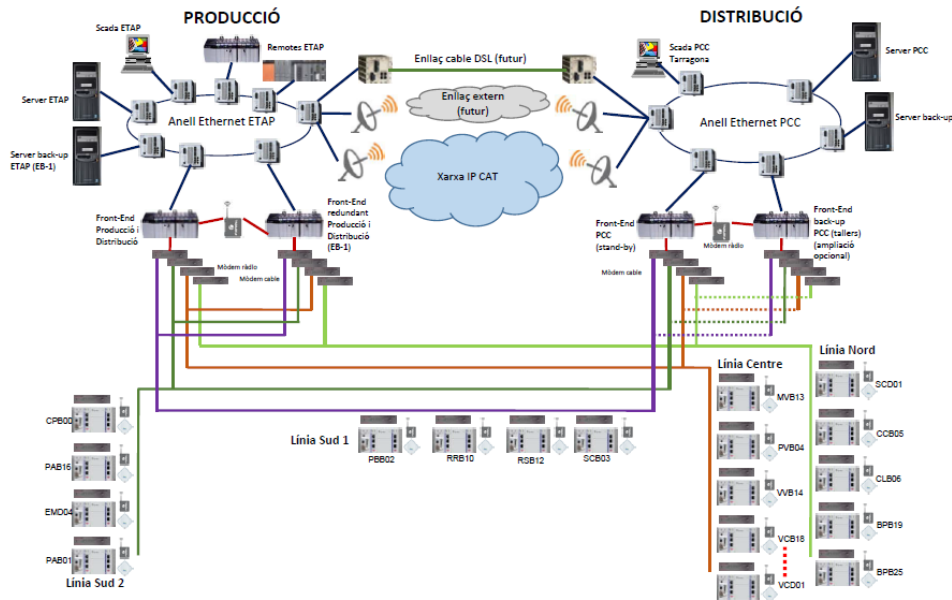
A nivell de PLCs diferenciem tres àrees ben diferenciades:

- **Distribució**  
Englobarien tots els equips que es troben en tota la província per controlar la distribució de l'aigua per tots els consorciats. Les instal·lacions són les canonades, bombaments, dipòsits i terciàries.
- **Producció**  
Per producció entenem la planta de tractament d'aigua potable (ETAP) situada a L'Ampolla. És la instal·lació més semblant al concepte de fàbrica que tenim ja que l'aigua provinent de la captació a Campredó es emmagatzemada aquí i tractada amb successius processos físico-químics fins que compleix els requisits necessaris per l'ús humà i llavors s'inicia la seva distribució per tota la xarxa.
- **Analitzadors i d'altres**  
El CAT disposa d'una sèrie d'equips multi paramètrics per tota la xarxa per assegurar la qualitat de l'aigua en qualsevol punt de la distribució.  
En aquest punt també podem incloure qualsevol altre dispositiu que no podem englobar ni en distribució o producció però que es interrogat pel SCADA, com podrien ser mesuradors elèctrics.

### 2.6.1 Xarxa de distribució. Front End, concentradores i remotes.

A les instal·lacions de la ETAP hi ha un Front-End (FE) Rockwell a que interroga a 18 concentradores repartides per tota la província.

Hi ha una tri redundància de comunicacions entre el FE i les concentradores, ja que el FE pot interrogar a les concentradores per Ethernet (radioenllaç), cable o ràdio.



Il·lustració 3 Sistema de comunicacions

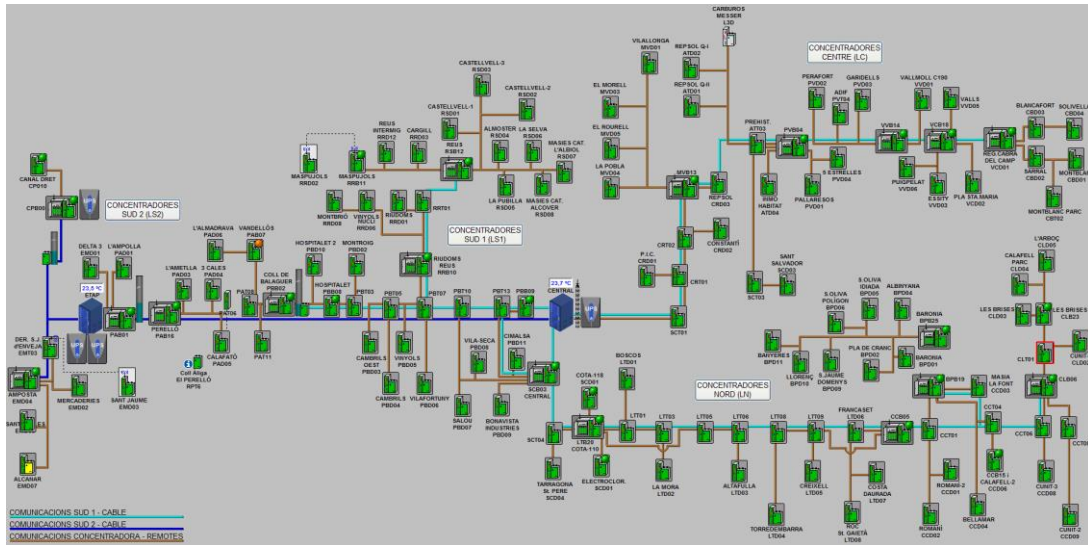
Actualment el FE que només existia a la ETAP, està tri redundat ja que s'han afegit 2 FE més, un a les mateixes instal·lacions de la ETAP però al edifici del bombament PAB01 i un altre a les instal·lacions centrals que hi han a Tarragona.

Això ens permet davant d'una fallida del principal, poder commutar de FE i d'aquesta manera no perdre connectivitat amb totes les concentradores. Tots tres FE estan configurats per poder comunicar per Ethernet, cable o ràdio, tenint una redundància plena de medis.

De cada concentradora poden penjar fins a un màxim de 40 remotes. Actualment hi han configurades 138 remotes. Les remotes tenen una arquitectura Mitsubishi. Les remotes són els PLCs que es troben en els dipòsits, terciàries, bombaments... al llarg de la província.

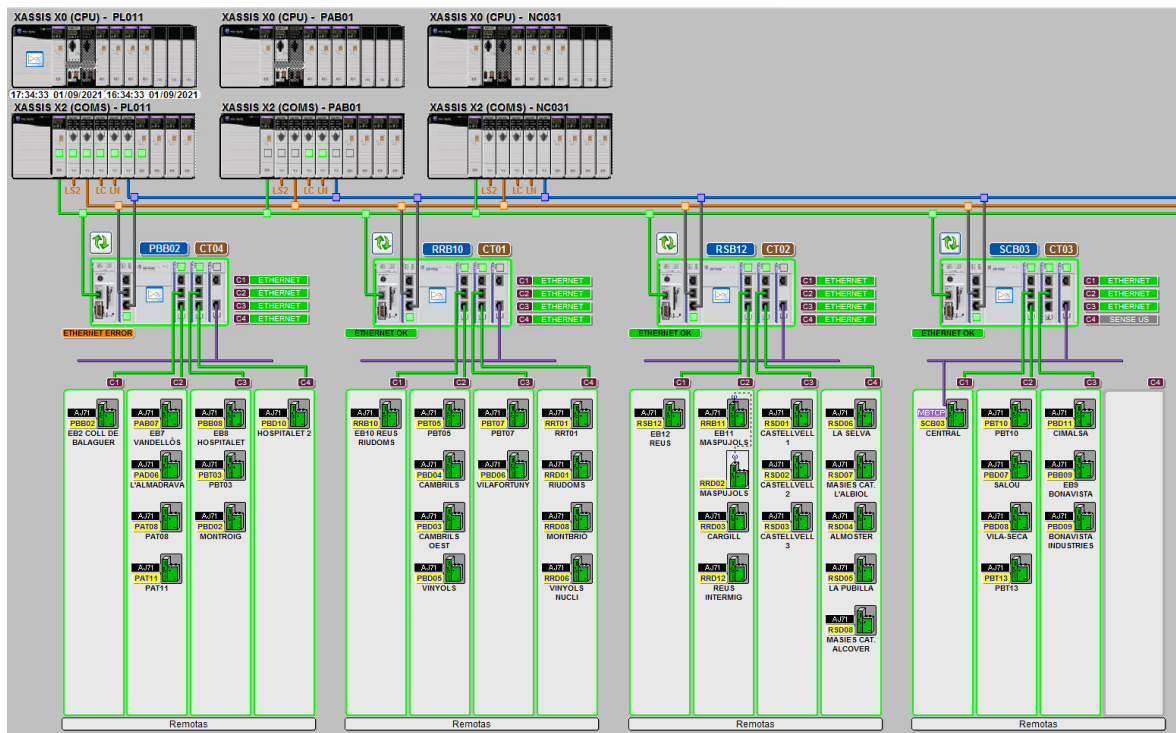
Les comunicacions entre concentradora i remotes es fan per cable mitjançant un protocol AJ71 de Mitsubishi.

En les remotes on hi ha connectivitat Ethernet (ja sigui radioenllaç, fibra òptica, 3G...), la comunicació està redundada, ja que la concentradora es pot comunicar amb elles per protocol MBTCP.



II-Il·lustració 4 Sinòptic de comunicacions amb totes les concentradores i remotes

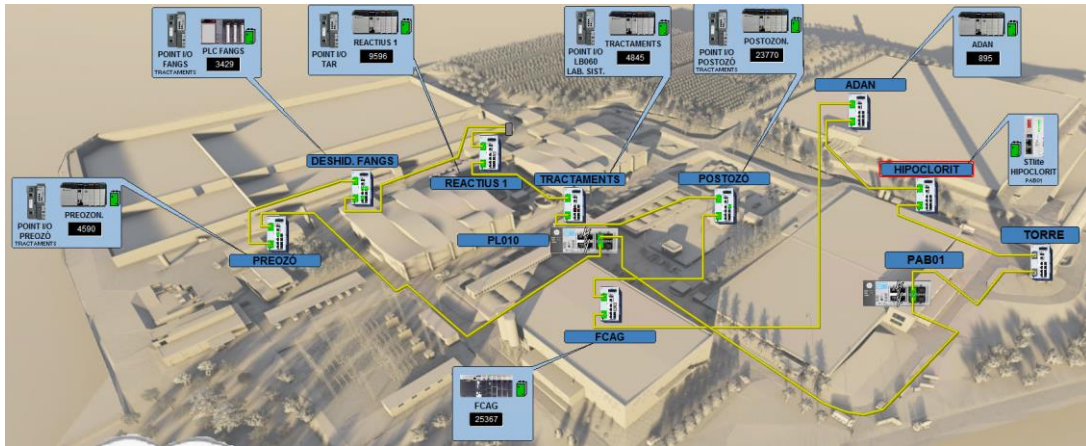
Desde el sistema SCADA es monitoritza en continu l'estat de les comunicacions, en una línia com la que es mostra pengen quatre concentradores i en total 40 remotes:



II-Il·lustració 5 Sinòptic de comunicacions de una de les línies

### 2.6.2 ETAP

Els PLCs que controlen els processos de la ETAP son Rockwell i el SCADA interroga directament als PLCs via protocol Ethernet ABCLX de CitectSCADA, ja que a la ETAP hi ha un anell de fibra òptica.



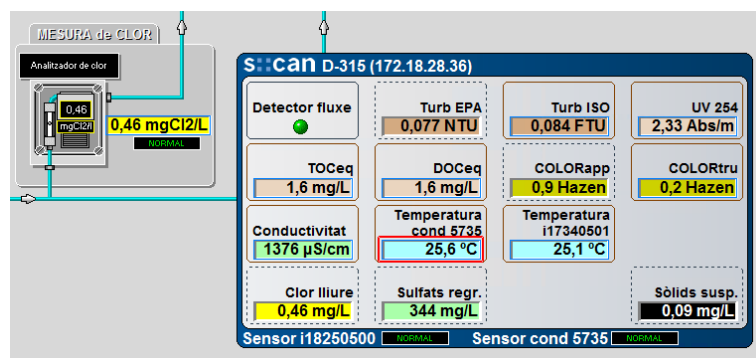
Il·lustració 6 Anell de fibra òptica a la ETAP

### 2.6.3 Analitzadors i altres equips

A banda de lo que és distribució (FE) i producció (ETAP), hi han una sèrie d'equips, com per exemple els multi paramètrics, que es poden trobar per tota la xarxa, que son interrogats via MBTCP.

Els multi paramètrics son equips de mesura de l'aigua que tenen diferents sondes per tenir informació de diferents paràmetres de qualitat de l'aigua tals com la torbesa, el pH, la temperatura, la conductivitat, els índex ultraviolats...

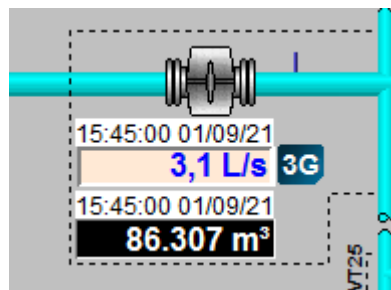
Existeixen altres equips amb particularitats com p.ex. cabalímetres que al trobar-se en llocs on no arriba la escomesa i/o comunicacions estableixen la comunicació amb el SCADA via 3G.



Il·lustració 7 Equip multi paramètric de mesura en continu de la qualitat de l'aigua

ANALITZADOR DE XARXA PWM 3000 TRACTAMENTS					
Intensitat L1	252,0 A	Tensió L1-N	227,0 V	Tensió L1-L2	392,0 V
Intensitat L2	253,3 A	Tensió L2-N	226,5 V	Tensió L2-L3	392,6 V
Intensitat L3	247,8 A	Tensió L3-N	227,1 V	Tensió L3-L1	394,3 V
Intensitat Mitja	251,0 A	Tensió Mitja	226,9 V	Tensió Mitja	393,0 V
P. Act. L1	48,1 Kw	P. React. L1	-31,2 KVAR	P. Aparent. L1	57,4 kVA
P. Act. L2	49,7 Kw	P. React. L2	-27,1 KVAR	P. Aparent. L2	56,7 kVA
P. Act. L3	48,5 Kw	P. React. L3	-27,6 KVAR	P. Aparent. L3	55,5 kVA
P. Act. Total	146,3 Kw	P. React. Total	-85,9 KVAR	P. A. Total	169,6 kVA
P. Factor L1	82,451	Freq. Media	50,0 Hz	Demanda Int.	265,2 A
P. Factor L2	86,374	Freq. Ult. Cicle	50,0 Hz	Demanda Act.	155,9 Kw
P. Factor L3	85,318	I. Desequilibri	2,1 %	Demanda Reac.	-90,9 KVAR
P. Factor Mitja	84,516	T. Desequilibri	0,3 %	Demanda Ap.	160,8 kVA
TOTAL ENERGIA ACTIVA (KWh)			TOTAL ENERGIA REACTIVA (KVARh)		
000011230227,646			000000000008,740		

Il·lustració 8 Analitzador de xarxa elèctrica



Il·lustració 9 Cabalímetre amb un datalogger que comunica via 3G

#### 2.6.4 SCADA

Actualment el sistema SCADA és un CitectSCADA v7.40 spB. Els dos servidors redundants corren cinc tasques simultànies:

- Procés IOServer  
És la tasca encarregada de les comunicacions.
- Procés Report  
És la tasca encarregada dels events.
- Procés Alarm  
És la tasca encarregada de les alarmes.
- Procés Trend  
És la tasca encarregada de les tendències.
- Procés Client  
És la tasca que mostra els sinòptics.
- Procés OPC DA Server  
És la tasca que permet que el CitectSCADA actiu com a OPC DA Server.

Els equips físics que alberguen el sistema SCADA es troben a la ETAP.

Tenim dos servidors redundants i 5 clients a la sala de control que permeten que els operadors puguin controlar tots els processos en temps real. El SCADA engloba totes les dades provinents de la ETAP, distribució i sistema d'alerta.

L'operador desde els sinòtics pot veure i controlar l'estat de tots els elements de la xarxa.

Així mateix el SCADA és l'encarregat de proporcionar dades a les aplicacions que ho puguin necessitar, com p.ex. a bases de dades SQL o mitjançant la component OPC DA Server a altres aplicatius.

### ***2.6.5 Bases de dades***

Tenim un servidor dedicat que corre un Microsoft SQL Server 2012 que emmagatzema tota la informació necessària del nostre sistema per la generació d'informes, consums...

El sistema SCADA s'encarrega de dipositar les dades necessàries (alarmes, registre d'actuacions, nivells, pressions, cabals, analitzadors, estat de vàlvules, estat de bombes...) en unes bases de dades locals al mateixos servidors (també Microsoft SQL Server 2012) o directament sobre aquest mateix servidor SQL principal.

Les dades que son desades en les bases de dades locals del servidors CitectSCADA son traspasades periòdicament al servidor principal per al seu posterior tractament. Això permet al sistema no perdre registres a les taules si hi ha problemes de connectivitat.

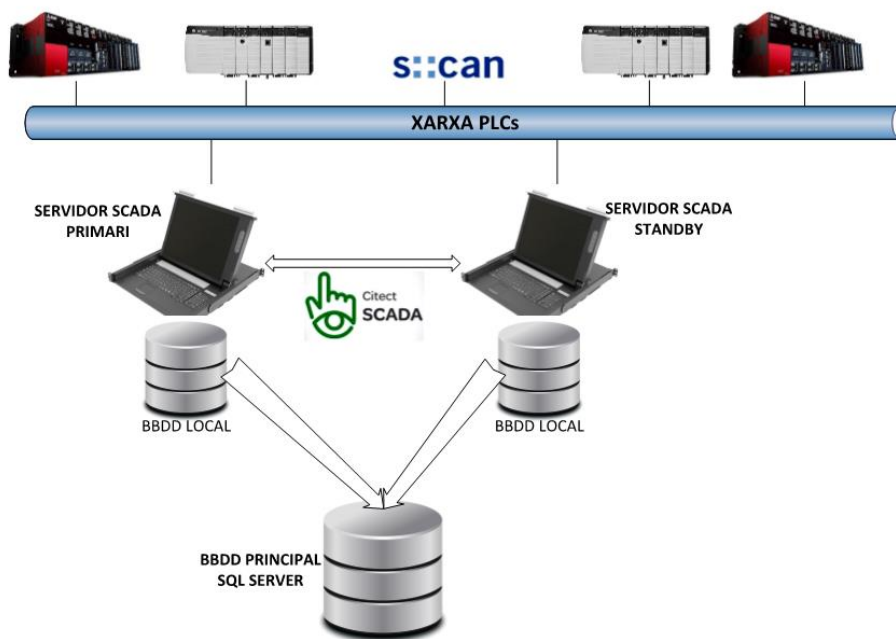
## 2.7 Implementació.

Ens centrarem en la instal·lació que ens trobem abans de la implementació del bessó digital i cap a on hem d'anar per tal de posar en marxa el bessó digital.

### 2.7.1 Arquitectura inicial

Partim d'una arquitectura en la que els dos servidors SCADA es comuniquen amb tots els PLCs de la instal·lació.

Cada servidor SCADA té una base de dades en local (en la mateixa màquina) i les dades emmagatzemades acaben en un únic servidor SQL que és la BDD principal.



*Il·lustració 10 Arquitectura de control inicial*

És una arquitectura de control clàssica on les dades que es reben al servidor de base de dades són proporcionades pel sistema SCADA que corre totes les components del sistema de control, accés a dades i control.

## **2.7.2 Estudi de l'arquitectura desitjada**

Iniciem l'estudi de quina és la arquitectura que necessitem per assolir els objectius proposats.

### **2.7.2.1 Simulador hidràulic**

La necessitat d'un simulador hidràulic de la instal·lació porta al departament de ITE del CAT a fer un estudi dels softwares comercials que hi han al mercat per a poder realitzar la implementació que s'ajusti millor a les particularitats i requeriments del nostre sistema.

Donat que nosaltres disposem d'un SCADA robust i treballat desde fa anys, ens interessa que el simulador hidràulic treballi de forma autònoma amb les dades que pugui obtindre del nostre sistema i que ens retorni la informació que necessitem, principalment al sistema SCADA. Diem principalment ja que la configuració i modelat del sistema de simulació no el realitzem nosaltres i per l'anàlisi el software ha de disposar d'eines pròpies, però si que tenim molt clar que la informació que ens proporciona el model ha de tornar al SCADA per tal que l'operador la pugui consultar en temps real i puguem tancar la realimentació necessària de tots dos sistemes i per poder ajustar alarmes o avisos per tal de detectar situacions que el departament de producció o de manteniment considerin útils i necessàries.

El departament de ITE decideix realitzar la implementació del OpenFlows WaterSight de Bentley.

OpenFlows WaterSight combina SCADA, GIS, modelat hidràulic i informació del client en un únic panel interoperable.

OpenFlows WaterSight permet el càlcul del rendiment actual, històric i previst per a cada actiu del sistema. Els usuaris poden obtindre gràfics detallats del rendiment actual del sistema, així com les conseqüències i millores previstes per a les accions relacionades amb l'activitat operativa i de manteniment. A més a més, es pot provar i simular virtualment les mesures de resposta davant de situacions d'emergència, com incendis, talls de canonades, aturada de bombes o altres esdeveniments en els que el temps es crucial.

Permet controlar fàcilment el flux, la pressió i les condicions de qualitat de l'aigua en cada punt del sistema. Fa càlculs d'auditoria d'aigua en temps real. Millora la eficiència energètica aprofitant l'anàlisi en temps real de cada bomba i dipòsit, amb alertes que apareixen quan el rendiment està fora dels llimars de servei.

Un sistema d'aquestes característiques requereix d'un modelat del model físic en el software. És necessari definir tots els elements que componen el nostre sistema físic. Aquesta feina la ha realitzat el departament de OTiD i ITE del CAT. Suposa definir totes les canonades, amb la longitud i diàmetre correcte i les seves característiques físiques, definir cada un dels elements, bombes, pressòstats, cabalímetres, en la posició exacta on es troben i amb les característiques pròpies de cada element....

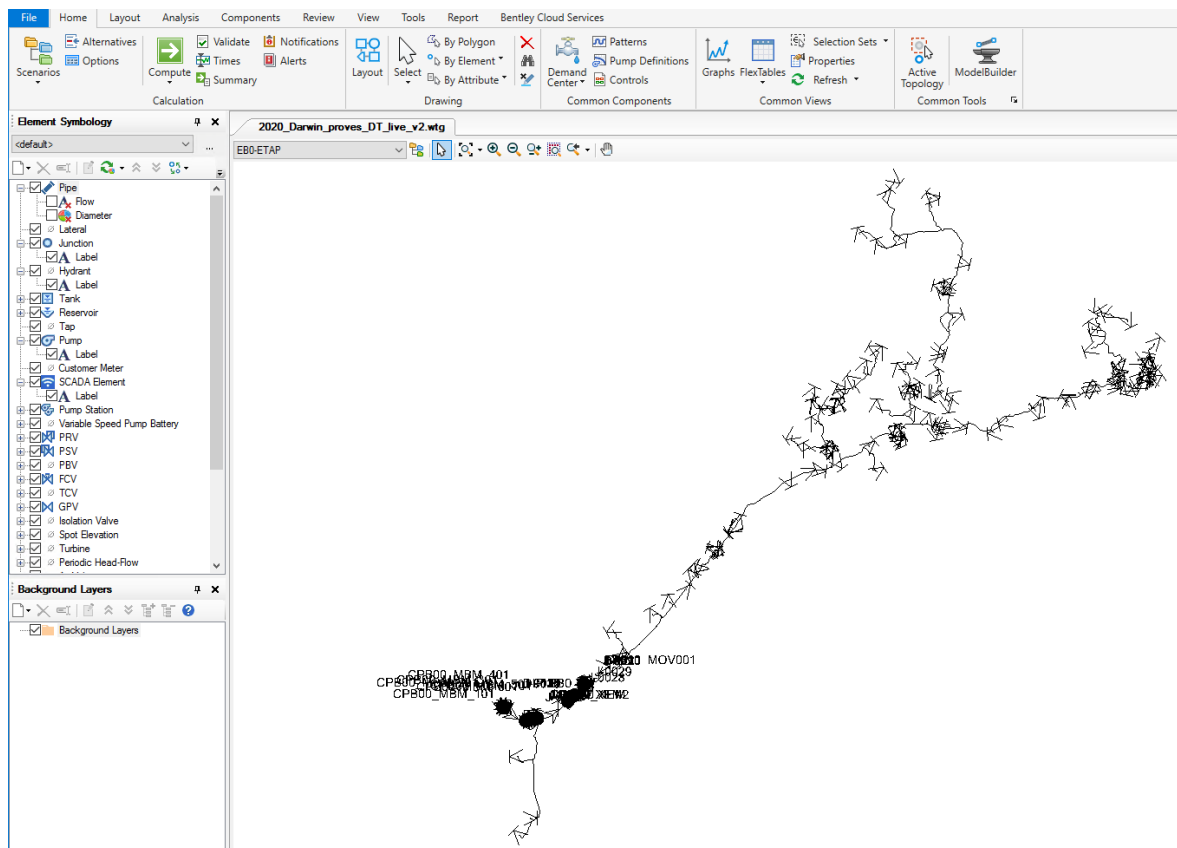


És una feina molt tediosa però necessària per tal que el nostre model digital s'ajusti a la realitat física. No és un procés que s'acabi una vegada s'han definits tots els elements sinó que el sistema s'anirà ajustant progressivament amb les dades que obtindrem del sistema real i la simulació hidràulica fins que el teòricament la simulació sigui del 100%.

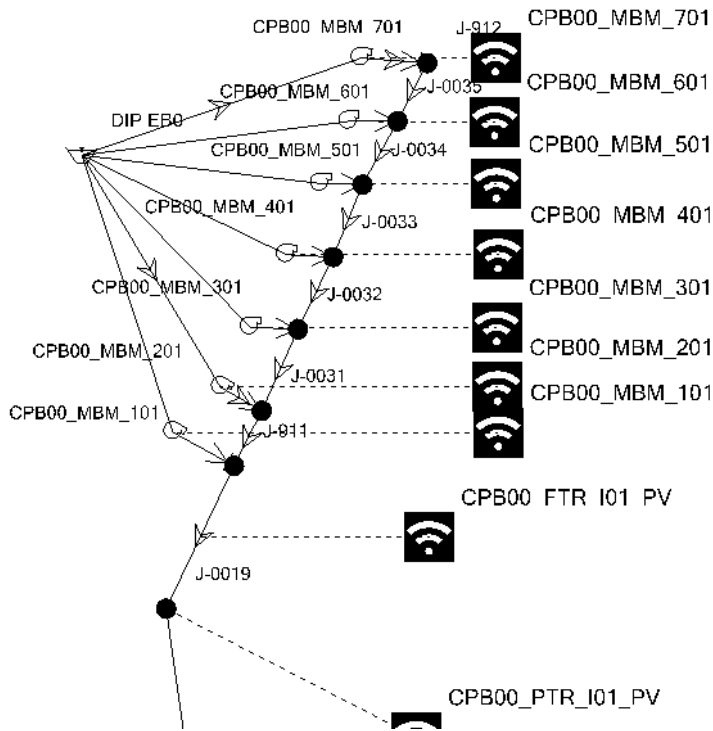
Aquest punt és molt important ja que ens mostra com la connectivitat del sistema físic (representat pel sistema SCADA / Bases de dades) i el simulador hidràulic ens proporciona la informació per ajustar el simulador cada vegada de forma més acurada.

És una característica molt important del bessó digital, l'autoaprenentatge, que ens permetrà convergir la simulació a la realitat física.

El simulador OpenFlows WaterSight de Bentley al nostre sistema constarà d'un PC que correrà on es realitzarà la configuració i modelat de la nostra instal·lació:



*Il·lustració 11 Modelat de la nostra xarxa al OpenFlows WaterSight*

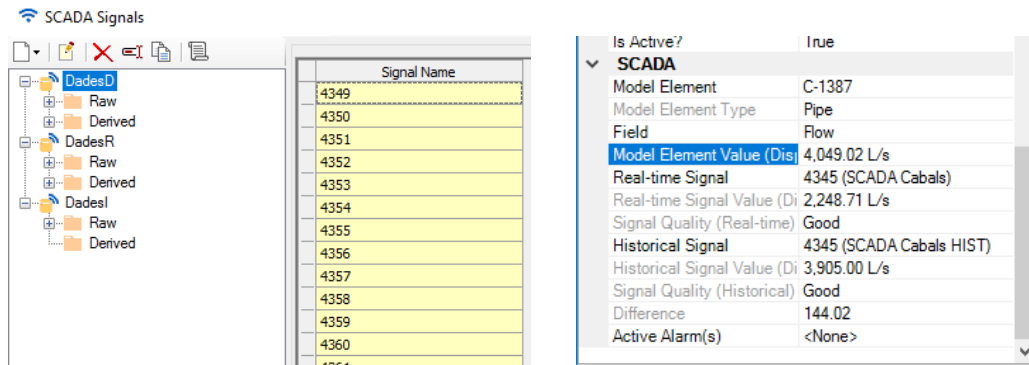


Il·lustració 12 Detall del modelat de la nostra xarxa al OpenFlows WaterSight

Així mateix es configuraran les dades que el hi proporciona el nostre CitectSCADA / Microsoft SQL Server, com més endavant detallarem.

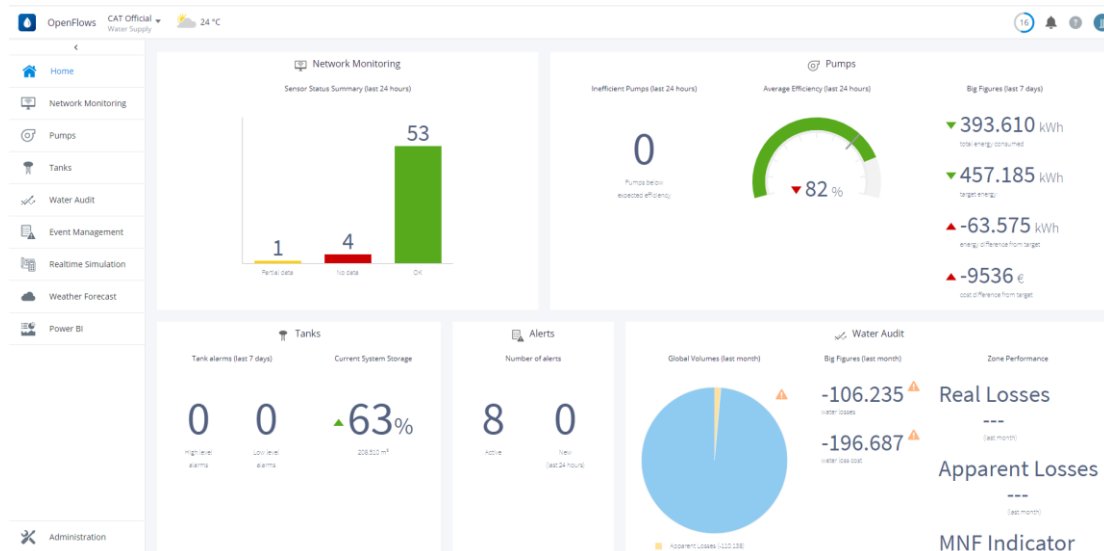
Label	Is Active?	Model Element	Model Element Type	Field	Real-time Signal	Real-time Signal Value (Display)	Signal Quality (Real-time)	Historical Signal	Historical Signal Value (Display)	Signal Quality (Historical)	Model Element Value (Numeric)
S123: EMD07_FTR_T01_P	<input type="checkbox"/>	FCV-028	FCV	Flow Setting		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S124: EMD07_PTR_T01_P	<input type="checkbox"/>	J-0125	Junction	Pressure		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S126: EMD05_FTR_T01_P	<input type="checkbox"/>	FCV-027	FCV	Flow Setting		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S127: EMD05_MOV_U01_	<input type="checkbox"/>	FCV-027	FCV	Valve Status		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S128: EMD04_FTR_T01_P	<input type="checkbox"/>	FCV-026	FCV	Flow Setting		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S129: EMD04_MOV_U01_	<input type="checkbox"/>	FCV-026	FCV	Valve Status		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S130: EMT03_PTR_P01_P	<input type="checkbox"/>	SAN JAUME	Junction	Pressure		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S131: EMT03_FTR_R01_P	<input type="checkbox"/>	SAN JAUME	Junction	Demand		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S132: EMD02_FTR_D01_P	<input type="checkbox"/>	FCV-025	FCV	Flow Setting		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S133: EMD02_FTR_E01_P	<input type="checkbox"/>	FCV-114	FCV	Flow Setting		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S134: EMD02_PTR_T01_P	<input type="checkbox"/>	J-0116	Junction	Pressure		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S135: EMD02_FTR_T01_P	<input type="checkbox"/>	C-121	Pipe	Flow		(N/A)	Bad	<None>	(N/A)	Bad	(N/A)
S136: CPB00_MBM_701	<input checked="" type="checkbox"/>	CPB00_MBM_70	Pump	Pump Status		(N/A)	Uncertain	4369 (DadesD)	(N/A)	Uncertain	(N/A)
S137: CPB00_MBM_601	<input checked="" type="checkbox"/>	CPB00_MBM_60	Pump	Pump Status		(N/A)	Uncertain	4366 (DadesD)	(N/A)	Uncertain	(N/A)
S138: CPB00_MBM_501	<input checked="" type="checkbox"/>	CPB00_MBM_50	Pump	Pump Status		(N/A)	Uncertain	4363 (DadesD)	(N/A)	Uncertain	(N/A)

Il·lustració 13 Taula de tags que llegeix OpenFlows WaterSight de nostre sistema



Il·lustració 14 Detalls de tags de lectura del nostre sistema

El simulador OpenFlows WaterSight de Bentley és un aplicatiu que treballa en el núvol seguint la interfase pel responsable del CAT una pàgina web on podem configurar i programar totes les funcionalitats que desitgem:



Il·lustració 15 Dashboard del OpenFlows WaterSight de Bentley del CAT

### ***2.7.2.2 Connectivitat OpenFlows WaterSight amb CitectSCADA***

El departament de ITE ens informa que el software OpenFlows WaterSight s'adapta a les seves necessitats i per tant el departament de Telecomandament hem de comprovar que s'ajusti al nostre sistema.

Els dos aspectes fonamentals son:

- Com OpenFlows WaterSight obté les dades en temps real o històric del nostre sistema SCADA
- Com OpenFlows WaterSight ens retorna les dades al nostre sistema SCADA.

Ens adrecem al departament tècnic de Bentley per mirar d'analitzar les possibles opcions.

#### ***2.7.2.2.1 Com OpenFlows WaterSight obté les dades en temps real o històric del nostre sistema SCADA***

En aquest primer punt, la obtenció de dades del nostre sistema SCADA per part de OpenFlows WaterSight, és obvi que la lectura de dades en temps real s'aconsegueix si OpenFlows WaterSight interroga al nostre SCADA per qualsevol protocol estàndard, ja sigui ModbusTCP, OPC DA, OPC UA...

El poder aconseguir dades històriques ja ens implica que hauria de demanar les dades a la component de tendències del SCADA (el nostre sistema te una component de tendències que abasta 36 mesos), lo que ens implica haver de programar algun tipus de DLL via API del CitectSCADA, p.ex. i ajustar-lo amb OpenFlows WaterSight. Ràpidament es descarta aquesta solució ja que el cost en temps (programació, proves, posta en marxa...) és massa elevat.

L'altre opció es que OpenFlows WaterSight pugui llegir les dades de una base de dades, ja sigui Microsoft SQL Server, MySQL... Donat que ja disposem de un servidor de Microsoft SQL Server en funcionament i que el SCADA registra en ell diferents taules (alarmes, registres d'operador, valors d'analitzadors, totalitzadors...) ja veiem que la solució ha d'anar en aquesta direcció.

Aquest tipus de aplicatius d'emmagatzematge de dades d'un SCADA en una base de dades es coneixen com a historitzadors (de l'anglès Historian) i n' existeixen múltiples opcions a nivell comercial. Generalment es recolzen sobre un SQL Server i permeten la integració amb un SCADA i major o menor configuració i programació del seu funcionament.

#### 2.7.2.2.2 *Historitzador*

Un Historian es un repositori de dades. Les tres característiques principals d'un Historian son:

- **Recollida de dades**  
El sistema ha de ser capaç de fer lectures de tots els tags desitjats en continu.
- **Validació de dades**  
El sistema ha de tenir eines per tal de validar les dades rebudes i poder implementar algoritmes per poder discriminar la qualitat de les dades.
- **Arxivament de les dades**  
El sistema ha de suportar l'emmagatzemament d'un històric de dades prou ample per les nostres necessitats o per la normativa de dades.

El Historian s'encarregarà d'agafar les dades dels servidors SCADA i depositar-les a la BBDD. Llavors en el nostre cas, el simulador hidràulic OpenFlows WaterSight anirà a buscar les dades actuals o històriques a la BDD.

Donat el coneixement que tenim d'alguns historitzador comercials i que ja tenim el propi servidor SQL Server i que el nostre sistema SCADA ja enregistra dades en ell, preferim desenvolupar nosaltres el nostre propi sistema Historian. Els motius principals són:

- **No dependre de software comercial extern.**  
Tenir software de diferents fabricants sempre implica que quan hi han canvis de versions sorgeixen problemes de configuració o pèrdues de funcionalitats.  
Desenvolupant el nostre historitzador sobre el nostre SCADA CitectSCADA ens permet minimitzar aquests riscos donat que la connectivitat del SCADA amb un SQL Server és una connectivitat molt elemental actualment.
- **Ens estalviem els llicències i costos fixes del software historitzador.**  
No ens cal cap contracte, compra de llicències o contractes de manteniment.
- **Aplicació a mida a la nostra realitat**  
El disseny de les taules, estructures a la base de dades... es fa a mida de les nostres necessitats i funcionalitats seguint els estàndards propis del CAT conjuntament amb el departament de IT. Ens permet dissenyar l'arquitectura que s'ajusta a la nostra realitat i no tenir funcionalitat que no ens siguin necessàries.
- **Sanejament i reestructuració de les nostres Bases de dades**  
Tots els registres que actualment fa el nostre SCADA sobre taules al servidor SQL s'han anat afegint a mesura que hi havia una necessitat per part de producció, laboratori o manteniment. El fet de dissenyar ara tota la estructura de dades pel nostre historitzador farà que moltes de aquestes taules antigues desapareguin ja que les mateixes dades hi seran a les noves taules del Historian.

- Creixement a curt i mig termini  
Realitzar el disseny propi ens permet orientar la programació tenint en compte futures funcionalitats i necessitats i ens permetrà un creixement ordenat i suposarà un estalvi en temps i recursos.

El motor de l'historitzador es realitza en CiCode, que és el llenguatge propi de CitectSCADA. És un llenguatge d'alt nivell similar a un VisualBasic o C++.

Està totalment integrat al programa de CitectSCADA permetent que el programador pugui adaptar l'aplicatiu a les seves necessitats reals.

Un dels punts més importants de CitectSCADA, a banda de poder comunicar amb un molt gran número de protocols diferents, és que no és un software tancat si no que permet al programador fer-se eines a mida via CiCode, API...

#### *2.7.2.2.3 Com OpenFlows WaterSight ens retorna les dades al nostre sistema SCADA.*

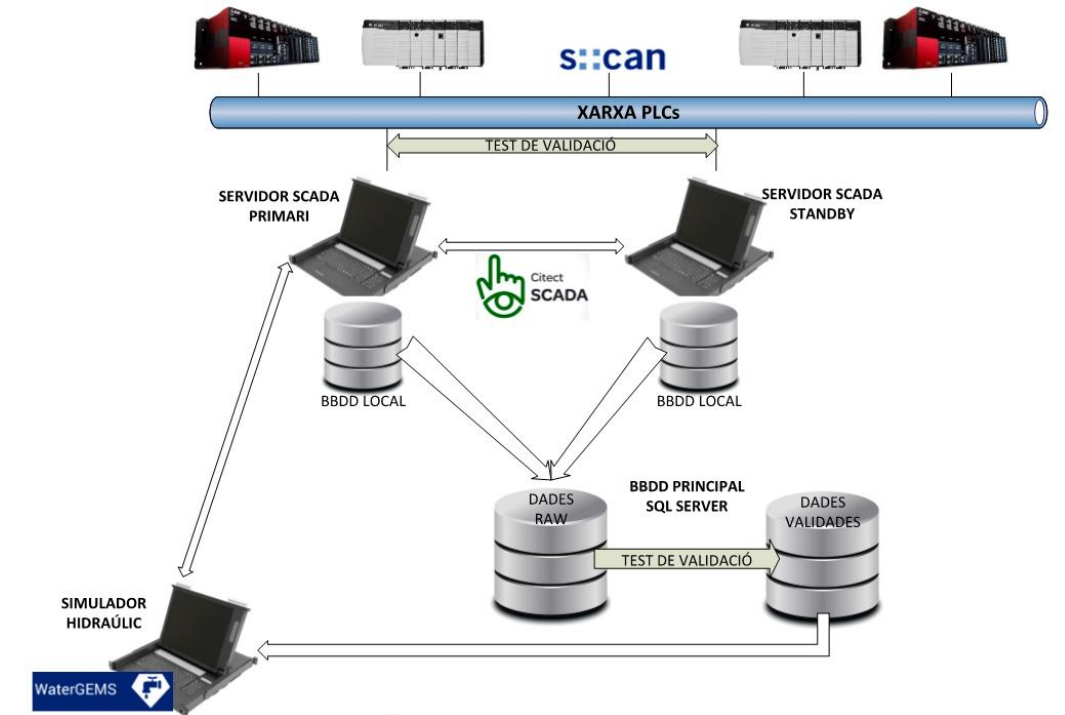
Les dades calculades per OpenFlows WaterSight han de ser retornades al nostre CitectSCADA.

Una de les característiques d'un SCADA és l'adquisició de les dades en temps real. Per aquesta raó sempre mirem d'obtenir les dades mitjançant un protocol estàndard, típicament en el cas del CAT, ModbusTCP, OPC DA, OPC UA... que son protocol estàndards que ja tenim implementats i no ens depenen de cap hardware o software adicional per la seva implementació.

Per a aquesta prova pilot, les dades que obtenim del simulador hidràulic son desades en la base de dades per tal que producció pugui analitzar-les.

### 2.7.3 Arquitectura desitjada

Una vegada hem aclarit les necessitats i els requeriment del nostre sistema, la nova arquitectura producte de la inclusió en el sistema del simulador hidràulic, OpenFlows WaterSight, ens implica la següent nova arquitectura:



*Il·lustració 16 Arquitectura de control final*

El principal canvi a nivell de hardware és la inclusió d'un nou PC (en el nostre cas màquina virtual) a la nostra xarxa de Telecomandament on correrà el OpenFlows WaterSight.

Les noves BBDD del Historitzador s'allotjaran al ja existent servidor Microsoft SQLServer.

I a més hem de aplicar a nivell de software dos procediments de validació de dades per tal de garantir la qualitat de les dades.

El motiu és la de la inclusió d'aquest procediments és la necessitat de poder assegurar la qualitat i el timestamp de les dades.

Qualsevol dada que provingui de camp haurà de passar per dos filtres:

- Quan arriben al CitectSCADA per tal d'assegurar que la dada que veu l'operador és correcta
- Quan arriben a la base de dades SQL Server per tal d'assegurar que la dada emmagatzemada és correcta.

Podem concloure que el bessó digital en el nostre sistema constarà de quatre àrees fàcilment diferenciables:

- Conjunt de PLCs que actuen sobre el nostre procés i ens proporcionen la informació d'estat i comandament.
- Sistema SCADA que engloba totes les senyals provinents de camp.
- Historitzador (BDD) per a poder gestionar l'emmagatzemat i validació de les dades.
- Model hidràulic que simula el procés en temps real o en qualsevol moment desitjat, amb les senyals que li proporciona el SCADA via Historitzador i que permet retornar informació al Sistema SCADA.



## 3 Memòria de Càlcul.

### 3.1 OpenFlows WaterSight

OpenFlows WaterSight necessita de dades en temps real i dades històriques.

Les dades en temps real les pot obtenir sense haver de fer cap modificació al nostre aplicatiu ja que CitectSCADA actua com un OPC DA Server. Es configurarà OpenFlows WaterSight per tal que tingui connectivitat amb aquest protocol.

Per una altra banda, necessita de dades històriques, raó per la qual vam decidir implementar la base de dades SQL Server perquè OpenFlows WaterSight pugui obtenir les dades d'aquestes taules.

Els requeriments per part del fabricant son tenir dades històriques cada 10 minuts de les variables que necessiten per tal de simular el sistema. Són les pressions, nivells, cabals, estat de marxa de les bombes, estat de les vàlvules de tot el sistema. En total la intenció es poder interrogar un total de 1121 senyals.

### 3.2 Base de dades del Historitzador

#### 3.2.1 Estudi i disseny de les taules del Historitzador.

En primer lloc hem de definir les taules del SQL on emmagatzemarem les dades.

Volem enregistrar senyals de tres tipus diferents:

- DIGITALS  
Iniciadors de vàlvules, confirmació de marxa de bombes, motors...
- REALS  
Valors de pressions, cabals, nivells...
- INTEGER  
Valors de totalitzadors, hores de funcionament, actuacions...

Donat que estem parlant de taules de dades que arribaran a milions de registres (un registre per cada lectura) ens interessa que les taules siguin lo més senzilles possibles, es a dir, que tinguin pocs camps i que les cerques es realitzin per camps numèrics. D'aquesta manera les consultes de dades seran ràpides. Per aquesta raó i per alleugerar el tamany de les taules hem d'evitar camps del tipus string.

Això ens porta a definir una taula diferent per cada un dels tipus de dades. Hi ha haurà una taula de valors del historitzador de dades tipus digital, un altre per les tipus real i finalment un altre per les de tipus integer. En funció de quina dada volem consultar, accedirem a una de les tres.

Llavors les taules que omplirà el CitectSCADA i consultarà el OpenFlows WaterSight seran del tipus:

idTag	int	Identificador únic del tag
valor	real/int/bool	Valor del tag
data	datetime	Data/hora del registre en format 2021-02-11 12:10:00
idproced	smallint	Procedència de la dada
qualitat	smallint	Qualitat de la dada

Siguent els camps idTag i data, claus del registre per tal d'evitar duplicats i accelerar les cerques.

La aparició en fase de disseny dels camps de procedència i qualitat responen a una necessitat que analitzarem tot seguit.

Un dels primers problemes que ens plantegem en aquesta fase de disseny és tenir en compte si el valor no es registra o el valor registrat és incorrecte quines accions hem de realitzar.

- Si un PLC està fora de servei, es a dir no hi ha comunicacions entre el CitectSCADA i un PLC (un problema de xarxa, un problema d'alimentació...), se'ns generarà un forat en el registre dels tags que pertanyen a aquest PLC.

La solució a aquest problema ens implica afegir el camp de procedència de la lectura. El valors enregistrats desde CitectSCADA tindran un valor de procedència diferent dels valors que hi puguem afegir manualment a posteriori per tal d'omplir el forat que ens ha aparegut en la sèrie temporal i hem de poder diferenciar l'origen d'aquests dos valors.

Això ens portarà a crear un altre triada de taules, una per cada tipus de dada, per tal de registrar el valor afegit o modificat de les taules de dades. Això ens permetrà tenir una traçabilitat de la dada per poder saber de la procedència de cadascuna.

- Si la dada que rep el CitectSCADA és incorrecta, p.ex, si la sonda d'un nivell analògic te un rang de 0 a 6.12 m i el CitectSCADA rep un valor de 8, el valor és incorrecte i s'haurà de poder detectar i corregir.

La solució a aquest problema ens implica afegir el camp de qualitat de la dada. Els algoritmes de validació de les dades ens permetran revisar la qualitat de la dada i assignar-li el valor de qualitat adient abans de inserir el registre en la taula de dades.

Com el registre de dades es farà en continu cíclicament, haurem de generar una sèrie de events que informin al administrador del sistema del historitzador del forat o dada incorrecta per tal que pugui esmenar la incidència.

La nova dada introduïda, ho serà manualment pel responsable del historitzador que haurà de validar el valor a introduir ja sigui per interpolació o per altres mètodes d'obtenció de la dada o analitzar i solventar el problema en la qualitat de la dada.

Pensant en funcionalitats futures, hem de poder emmagatzemar dades que no siguin lectures de valors de camp, feina que desenvolupa el historitzador en sentit estricte, sinó de valors calculats.

La necessitat ens apareix, per exemple, arran de controlar que els valor totalitzats pel cabalímetres a camp siguin correctes per poder detectar lo abans possible malfuncionaments d'aquests mesuradors, donada la seva importància en el nostre sistema.

Si generem un valor del totalitzat horari a partir del resultant de la integració dels valors de camp del cabal instantani en una hora ens permetria detectar malfuncionaments dels cabalímetres amb antelació i evitar problemes amb la gestió de consums.

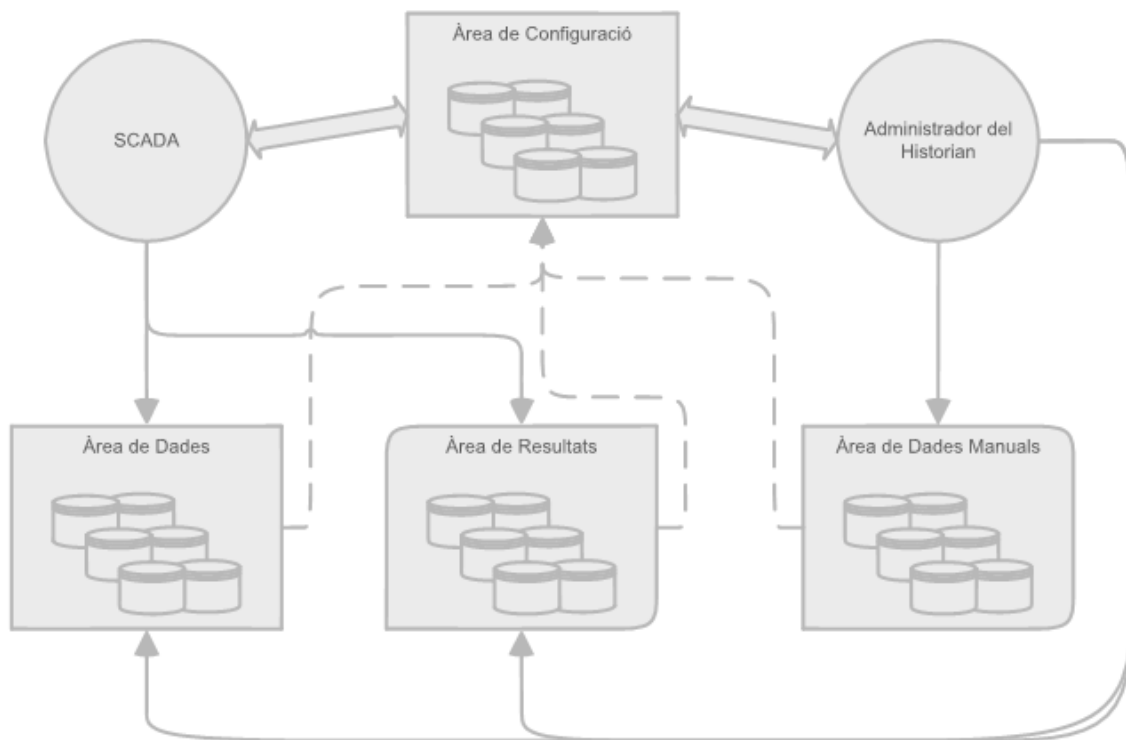
El valor calculat, resultant de la integració dels valors de camp del cabal, seria obtingut igualment pel CitectSCADA però aquest valor al no ser un valor d'una lectura de camp no pot anar a les taules de dades del historitzador. Hem de generar altres taules de valors calculats.

Finalment per motius per exemple, d'una operativa de buidament de canonada, necessitem també de taules en les que introduïrem valors de forma manual. Quan es buida una canonada una part de l'aigua que pugui haver en ella pot no ser totalitzada per cap cabalímetre en el seu ramal. Això implica la necessitat de definir uns totalitzats virtual per tal que el balanç de masses total no es vegi distorsionat.

L'estudi de les necessitats i dels problemes propis de la nostra instal·lació ens porta a definir quatre àrees diferenciades de taules:

- Taules de dades o taules SCADA  
Taules que emmagatzemen els valors dels tags. CitectSCADA escriurà els valors de forma periòdica en aquestes taules. Aniran precedides pel prefix HIST ja que son valors historitzats del sistema, lectures de dades d'equipaments de la instal·lació que en proporciona el sistema SCADA.  
Aquí estarien les tres taules que hem comentat, una per cada tipus de dada (REAL, INTEGER i BOOL).

- **Taules de configuració**  
 Conjunt de taules de definició de tots els tags que historitzem i taules que relacionen i agrupen aquests tags. Aniran precedides pel prefix CFG.  
 Ens permetran realitzar consultes agrupades, p.ex. només consultar valors de nivells analògics de dipòsits o només tags d'un àrea en concret.  
 Necessàries per identificar els registres de les taules de dades tal i com he vist abans mitjançant el camp idTag.
- **Taules de dades manuals.**  
 Taules que emmagatzemen valors que s'introdueixen manualment. Necessitem de taules que puguem omplir nosaltres, ja que les de dades només son omplertes pel CitectSCADA. El prefix vindrà donat segons la utilitat de les taules.
- **Taules de resultats**  
 Taules que emmagatzemen valors calculats. El prefix vindrà donat segons la utilitat de les taules creades.



*Il·lustració 17 Arquitectura de taules necessàries per implementar el historitzador*

Una vegada tenim resolts els requisits podem definir quines seran les taules que hem de crear. Hem de tenir en compte que a banda de les nostres necessitats, definim altres que son d'utilitat al departament de IT i Producció.

A l'àrea de configuració hem de poder abastar la definició i relacions necessàries de tots els tags de la nostra instal·lació. Aquestes taules seran omplertes en primera instància pel administrador del sistema historitzador. Les taules definides són:

- CFG\_Tags  
Llista de tots els tags de la instal·lació, és la taula genèrica i principal on tenim tota la informació relativa a un tag.
- CFG\_TipusTags  
Informació de la mesura del tag, un cabal, una temperatura...
- CFG\_TipusInst  
Ens indica que a quina instal·lació pertany el tag, un bombament, un dipòsit, una electrocloració...
- CFG\_TipusProced  
Ens indica la procedència del valor d'un tag, si és una lectura del CitectSCADA, un valor introduït de forma manual...
- CFG\_Sup  
Necessari pels tags de nivell de dipòsits. Té informació relativa al dipòsit, alçada, superfície..
- CFG\_Zones  
Tot el CAT està dividit en zones, aquesta taula ens permet conèixer a quina zona pertany el tag. Necessari pels totalitzadors.
- CFG\_ZonesTags  
Relaciona tots els tags que pertanyen a una mateixa zona.

L'àrea de taules de dades només tindrà les taules de dades que ens proporciona el CitectSCADA:

- HIST\_DadesD  
Taula de valors de dades de tipus BOOL.
- HIST\_DadesI  
Taula de valors de dades de tipus INTEGER.
- HIST\_DadesR  
Taula de valors de dades de tipus REAL.

A l'àrea de taules de dades calculades definim les taules:

- VC\_DadesR  
Taula de totalitzadors provinents de la validació de cabals.  
La validació dels cabals és el càlcul dels totalitzats dels cabalímetres a partir de la integració dels seus valors de cabal instantani.

A l'àrea de taules de dades manuals ja podem diferenciar les subàrees segons l'àmbit al que pertanyen.

En el cas del Historitzador necessitàvem taules manuals per tal de poder inserir valors que no s'hagin registrat a les taules del Historian o que necessitin ser modificats els valors enregistrats. Les taules necessàries son:

- HIST\_RectifD  
Taula de valors rectificats de dades de tipus BOOL.
- HIST\_RectifI  
Taula de valors rectificats de dades de tipus INTEGER.
- HIST\_RectifR  
Taula de valors rectificats de dades de tipus REAL.

En el cas dels totalitzats virtuals necessaris pel balanç de masses ja que quan es buida una canonada, aquesta aigua no es computada pels cabalímetres també ha de tenir la seva taula de valors manuals definida pel mateix raonament que hem seguit amb les taules de valors del Historian. Definim la taula:

- HIST\_Anomalies

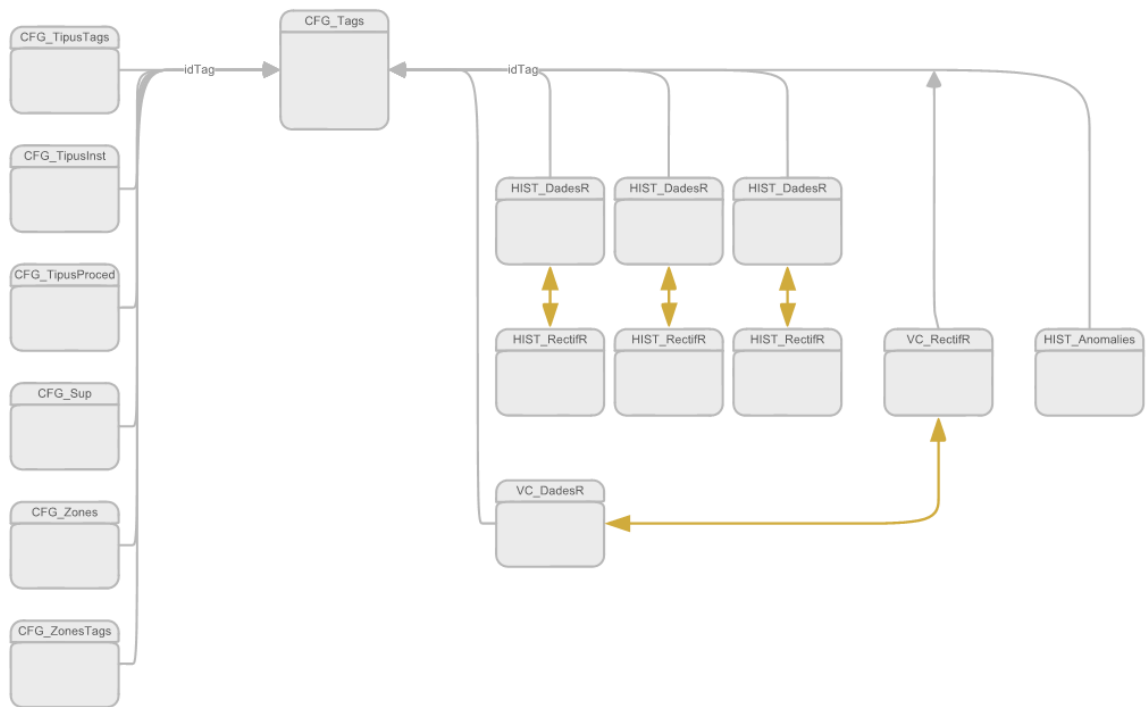
I finalment en el cas de la taula de valors calculats dels totalitzadors a partir dels valors dels cabals instantanis, apliquem la mateixa lògica ja que un valor calculat pot no ser registrat o el seu càlcul erroni. Definim la taula:

- VC\_RectifR  
Taula de valors rectificats de la validació de cabals.

Ara ja tenim conceptualment totes les taules necessàries per tal de poder començar a fer la implementació al SQL Server.

Una vegada les taules estiguin definides ja començarem a realitzar la programació per tal que el CitectSCADA depositi en elles les dades.

El diagrama funcional de les taules és el següent:



*Il·lustració 18 Diagrama de taules de la BBDD*

Abans de començarem a detallar cada una de les taules, camps necessaris i tipus de dades que implementarem, incidirem de nou en el camp de qualitat del tag.

### 3.2.2 Qualitat de les dades

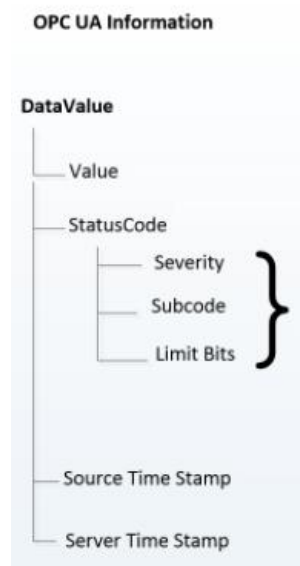
El criteri emprat pel valor de qualitat que s'assignarà a les dades del nostre historitzador s'ajustarà al estàndard OPC UA.

OPC Unified Architecture (OPC UA) és un protocol de comunicacions de control industrial. OPC UA proporciona un arquitectura orientada a serveis (SOA) per a aplicacions industrials, definint interfícies per a serveis similars als serveis web.

Ens interessa ajustar-nos en la mesura del possible a aquest estàndard en aquesta fase pilot ja que és àmpliament implementat i present al nostre entorn industrial.

El tractament de les dades que fa aquest protocol és molt semblant al que volem implementar al nostre historitzador, ja que una dada sempre va associada a quatre components:

- El valor de la dada,
- el StatusCode que nosaltres anomenem qualitat ,
- la marca de temps (timestamp) de la dada i
- la marca de temps del servidor (moment en que es rep la dada)



*Il·lustració 19 Informació d'una dada OPC UA*

Segons la implementació del estàndard OPC UA, la qualitat, que és un valor de 32 bits, es pot dividir en tres grups principals:

- Bona (Good)  
El primer byte te un valor de 0 (0x00)
- Incerta (Uncertain)  
El primer byte te un valor de 64 (0x40)
- Dolenta (Bad).  
El primer byte te un valor de 128 (0x80)



En el nostre cas, en aquesta prova pilot els algorismes de validació de la dada només assignarà quatre d'ells.

Good (0x00000000) The operation succeeded.

Uncertain (0x40000000) The operation was uncertain.

UncertainEngineeringUnitsExceeded (0x40940000)  
The value is outside of the range of values defined for this parameter.

Bad (0x80000000) The operation failed.

A la pàgina web:

<http://www.opcfoundation.org/UA/schemas/1.04/StatusCode.csv>

Podem trobar el llistat complet que defineix el estàndard OPC UA.

### 3.2.2.1 Taules de configuració dels tags

Definirem una sèrie de taules amb el prefixe CFG (configuració) que llistaran la informació dels tags que enregistrem. Aquestes tables mixtes en quan a informació emmagatzemada (hi han dades integers, string, real...) ens permetran tenir tota la informació dels equips i poder interrelacionar-los en funció del nostre interès.

#### 3.2.2.1.1 CFG\_Tags

En la taula CFG\_Tags estaran llistats tots els tags de la instal·lació.

Ens adreçarem a aquesta taula per tenir qualsevol informació d'un tag en concret. És una taula que ens indexarà totes les taules de dades, ja que qualsevol taula de dades com a mínim tindrà el camp idTag, un numèric, que serà referenciat a aquesta taula de tags.

El motiu principal és que donat que les taules de dades poden emmagatzemar milions de registres, les cerques son mes ràpides i eficients si les realitzem via el seu id numèric i no p.ex. pel nom del tag.

CFG_Tags		
<b>idTag</b>	int	Identificador únic del tag que obtindrem via consulta per tal de treballar amb els tags segons nomenclatura CAT
<b>tag</b>	varchar(80)	Identificador de tag en nomenclatura CAT aaaaa_bbb_ccc_ddd aaaaa: Instal·lació (CCB15...) bbb: tipus instrument (FTR, LTR, PTR...) ccc: identificador de l'instrument (M01, M02, 001...) ddd: sufixe del tipus de dada (PV, TOT...)
<b>idTagTip</b>	smallint	<a href="#">Id de tipus de tag (veure taula CFG_TipusTags)</a>

<b>eng_unit</b>	varchar(12)	Unitats d'enginyeria
<b>eng_zero</b>	real	Zero de les unitats d'enginyeria en format REAL
<b>eng_full</b>	real	Fons d'escala de les unitats d'enginyeria en format REAL
<b>descTag</b>	varchar(64)	Descripció del tag
<b>tagOld</b>	varchar(80)	A efectes de tenir compatibilitat amb la notació antiga
<b>dataAlta</b>	datetime	Datetime de la donada d'alta del tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>dataBaixa</b>	datetime	Datetime de la donada de baixa del tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>idInstTip</b>	int	<a href="#">Id de tipus d'instal·lació (veure taula CFG_TipusInst)</a>
<b>idPare</b>	id	Té valor (<> de NULL) Si es un totalitzador i agafa aigua a través de l'intermediari idPare
<b>per10</b>	bit	cert si el totalitzador envia 1 pols per cada 10m3
<b>aporta</b>	bit	cert si el totalitzador aporta aigua a la xarxa de distribució del CAT (EB1 imp., Riudecanyes,...)
<b>xarxa</b>	bit	cert si el totalitzador es un comptador de xarxa del CAT (ex: EB14 entrada inv, EB23 impulsíó,...)
<b>fictici</b>	bit	Cert si es tracta d'algun dels cabals que indiquen compliment/buidatge de canonada. Hi ha un tag de tipus fictici per cada zona
<b>LimMinQ</b>	real	Cabal mínim admès al comptatge (per sota es considera erroni) m3/h - Valor típic 0
<b>LimMaxQ</b>	real	Cabal màxim admès al comptatge (per sobre es considera erroni) m3/h - segons punt
<b>UltAcces</b>	datetime	Datetime de última lectura del tag

Una mostra dels registres que s'emmagatzemen a aquesta taula:

idTag	tag	idTagTip	eng_unit	eng_zero	eng_full	descTag	tagOld	dataAlta	dataBaixa	idInstTip	idPare	per10	aporta	xarxa	fictici	LimMinQ	LimMaxQ	UltAcces
4070	ATD01_FTR_T01_PV	0	L/s	-10	500	Reg. analògic cabal	L7D	2021-11-02 12:00:00.000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-07-31 19:05:14.457
4071	ATD01_LTR_D01_PV	3	m	0	25.5	Reg. analògic nivell	L7D	2021-11-02 12:00:00.000	NULL	NULL	NULL	0	NULL	NULL	NULL	0	16000	2021-07-31 19:05:14.457
4072	ATD01_MOV_T01_Y	4	%	0	100	Reg. analògic % obert v.pri...	NULL	2021-11-02 12:00:00.000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-07-31 19:05:14.457
4073	ATD01_MOV_T02_Y	4	%	0	100	Reg. analògic % obert v.byp.	NULL	2021-11-02 12:00:00.000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-07-31 19:05:14.457
4074	ATD02_FTR_T01_PV	0	L/s	-6	300	Reg. analògic cabal	L2D	2021-11-02 12:00:00.000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-07-31 19:05:14.457
4075	ATD02_LTR_D01_PV	3	m	0	16.32	Reg. analògic nivell	L2D	2021-11-02 12:00:00.000	NULL	NULL	NULL	0	NULL	NULL	NULL	0	16000	2021-07-31 19:05:14.457

Il·lustració 20 Registres a la taula CFG\_Tags

A diferència de la resta de taules de configuració que detallarem tot seguit, que son emplenades manualment pel administrador del Historian, aquesta taula serà emplenada o completada pel administrador del Historian donat que quan CitectSCADA vagi a enregistrar el valor de un tag mitjançant el procediment emmagatzemat de SQL Server d'inserció de dada del historitzador que hem programat i que veurem més endavant, primer es comprovarà si el tag existeix en aquesta taula i en cas que no ho estigui, donarà d'alta el nou tag i li assignarà el seu idTag únic. D'aquesta manera no caldrà fer cap modificació al CitectSCADA quan afegim nous tags al sistema Historian.

Actualment hi han registrats 1407 tags

### 3.2.2.1.2 CFG\_TipusTags

La taula CFG\_TipusTags està referenciada a CFG\_Tags. Per un tag de la taula CFG\_Tags ens informa del tipus de variable del tag:

CFG_TipusTags		
<b>idTagTip</b>	smallint	tipus de tag
<b>descTagTip</b>	varchar(64)	Descripció de tipus de tag
<b>TipusVar</b>	varchar(64)	tipus de dada de la variable: digital, real, long int
<b>SufixTagTip</b>	varchar(16)	Sufixe del tipus de tag

Ens permet conèixer la naturalesa física del senyal del tag.

En aquest cas, el número de registres es fixe o es pot veure incrementat pel administrador del historitzador si no existeix cap registre associat a la variable física que mesura l'instrument.

Els registres que s'emmagatzemen a aquesta taula:

idTagTip	descTagTip	TipusVar	SufixTagTip	ArrelTagTip
0	Valor de cabal	REAL	_PV	FTR
1	Valor de pressió	REAL	_PV	PTR
2	Valor de temperatura	REAL	_PV	TTR
3	Valor de nivell	REAL	_PV	LTR
4	Valor de posició	REAL	_Y	NULL
5	Valor de pes	REAL	_PV	WTR
6	Valor de Humitat	REAL	_PV	NULL
7	Valor de velocitat	REAL	_PV	NULL
8	Valor de volum	REAL	_PV	NULL
9	Voltatge	REAL	_PV	NULL
10	Intensitat	REAL	_PV	NULL
11	Potència	REAL	_PV	NULL
50	Totalitzador de cabal	LONG	_TOT	FTR
51	Totalitzador virtual de cabal	LONG	_TOTV	FTR
52	Temps de funcionament	LONG	_TF	NULL
53	Número d'actuacions	LONG	_ACT	NULL
100	Iniciador de oberta	DIGITAL	_O	NULL
101	Iniciador de tancada	DIGITAL	_T	NULL
102	Condió de marxa	DIGITAL	_CM	NULL

Il·lustració 21 Registres a la taula CFG\_TipusTags

### 3.2.2.1.3 CFG\_TipusInst

La taula CFG\_TipusInst està referenciada a CFG\_Tags. Per un tag de la taula CFG\_Tags ens informa del tipus de instal·lació a la que pertany el tag:

CFG_TipusInst		
<b>idInstTip</b>	smallint	tipus de tag
<b>descInstTip</b>	varchar(64)	Descripció de tipus de instal·lació

Els registres que s'emmagatzemen a aquesta taula:

idInstTip	descInstTip
0	Bombament
1	Dipòsit
2	Terciària
3	Electrocloració
10	ETAP

Il·lustració 22 Registres a la taula CFG\_TipusInst

#### 3.2.2.1.4 CFG\_TipusProced

La taula CFG\_TipusProced està referenciada a les taules de dades del historitzador que veure posteriorment. Ens informa de la procedència de la dada emmagatzemada:

CFG_TipusProced		
<b>idProced</b>	smallint	Identificador de procedència de tag (scada, manual, automàtic, de càlculs,...)
<b>descProced</b>	varchar(128)	descripció de la procedència
		possibles valors del camp idproced:
		0...19 SCADA
		20...29 Valors introduïts de forma manual
		30...59 Valors introduïts de forma automàtica

	idProced	descProced
1	0	Dada SCADA provinent via TagRead
2	1	Dada SCADA provinent via TagSubscribe
3	2	Dada SCADA provinent via Trend Server
4	5	Totalitzat calculat a partir dels valors instantanis de cabal (Validació de cabals)
5	20	Cabals ficticis
6	30	Dada introduïda de forma automàtica (pending...)
7	40	Dada alterada per càlculs UPC

Il·lustració 23 Registres a la taula CFG\_TipusProced

#### 3.2.2.1.5 CFG\_Sup

Esta és una de les taules, conjuntament amb les dos que explicarem tot seguit, CFG\_Zones i CFG\_ZonesTags, que es creen per a cobrir necessitats del departament de IT i de Producció. Les llistem per tal d'entendre que el historitzador és només una part de tota la informació de la base de dades que hem planificat i punt de partida per tal de implementar multitud d'eines necessàries per l'anàlisi de la nostra instal·lació.

La taula CFG\_Sup llista informació dels dipòsits associats al seu tag de nivell analògic.

CFG_Sup		
<b>idTag</b>	int	Identificador de Tag
<b>sup</b>	real	Superfície de la base del dipòsit
<b>nivMax</b>	real	Nivell màxim del dipòsit [m]
<b>Alta</b>	datetime	data des de la que es vàlida la superfície indicada
<b>Baixa</b>	datetime	Data de baixa

idTag	sup	nivMax	Alta	Baixa
4850	19,6	88	1900-01-01 00:00:00.000	NULL
4931	124,7	7,65	2005-01-01 00:00:00.000	NULL
4867	23	4,8	1900-01-01 00:00:00.000	NULL
5095	282	6,1	2018-03-13 00:00:00.000	NULL
4909	62	6	2018-03-14 00:00:00.000	NULL
4503	74	5,4	1900-01-01 00:00:00.000	NULL
5056	680	4,5	1900-01-01 00:00:00.000	NULL

*Il·lustració 24 Registres a la taula CFG\_Sup*

### 3.2.2.1.6 CFG\_Zones

La taula CFG\_Zones defineix els diferents ramals que componen la xarxa de distribució del Consorci d'Aigües de Tarragona.

CFG_Zones		
<b>idZona</b>	smallint	Identificador de zona
<b>descZona</b>	varchar(64)	descripció
<b>LimMinPF</b>	real	Límit mínim error pares-fills acceptable (p.ex. -4000)
<b>LimMaxPF</b>	real	Límit màxim error pares-fills acceptable (p.ex. +4000)

idZona	descZona	LimMinPF	LimMaxPF
1	Ramal Baix Ebre	-25	25
2	Ramal Principal	-500	500
4	Ramal EB10	-50	50
6	Ramal EB12	-25	25
7	Ramal Secundari	-500	500
8	Ramal EB13	-15	15
9	Ramal EB4	-200	200
10	Ramal EB18 i Cabra	-10	10
11	Ramal Litoral	-75	75
13	Ramal EB6	-10	10
14	Ramal Calafell Cunit	-150	150
16	Ramal EB14	-50	50
17	Ramal EB19	-50	50
50	Dipòsits RSD03	-50	50
51	Dipòsits PBB02	-50	50
52	Dipòsits PVB04	-50	50
53	Dipòsits SCB03	-50	50

*Il·lustració 25 Registres a la taula CFG\_Zones*

### 3.2.2.1.7 CFG\_ZonesTags

La taula CFG\_ZonesTags relaciona els tags dels cabalímetres de cada ramal.

CFG_ZonesTags		
<b>idZona</b>	smallint	Identificador de zona
<b>idTag</b>	int	identificador de tag
<b>distrib</b>	bit	Cert si el tag aporta aigua al ramal

idZona	IdTag	distrib
53	20070	0
53	20069	1
17	20012	0
16	20011	0
14	20010	0
13	20009	0
11	20008	0
10	20007	0

Il·lustració 26 Registres a la taula CFG\_ZonesTags

### 3.2.2.2 Taules de dades dels tags

Una vegada ja hem definit les estructures necessàries, ja podem descriure les taules de dades amb el prefixe HIST (historitzador) on emmagatzemarem les dades que ens proporciona el CitectSCADA. Aquestes son les taules que consultarà el OpenFlows WaterSight.

Com ja hem comentat anteriorment definim tres taules diferents segons el tipus de dada que volem guardar. Aquestes taules son:

- HIST\_DadesR  
Emmagatzema valors de tipus REAL
- HIST\_DadesI  
Emmagatzema valors de tipus INTEGER
- HIST\_DadesD  
Emmagatzema valors de tipus DIGITAL

### 3.2.2.2.1 HIST\_DadesD

En aquesta taula s'emmagatzemen els valors de tipus digitals del historitzador.

HIST_DadesD		
<b>idTag</b>	int	Identificador de Tag
<b>valor</b>	bit	Valor del tag DIGITAL
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>idproced</b>	smallint	Procedència. Valor de validació de la dada. Definim un integer per tal de definir l'origen de la dada
<b>qualitat</b>	smallint	Informació de qualitat de la dada

idTag	valor	data	idproced	qualitat
4720	0	2021-08-29 23:20:00.000	2	0
4718	1	2021-08-29 23:20:00.000	1	0
4717	1	2021-08-29 23:20:00.000	1	0
4716	0	2021-08-29 23:20:00.000	1	0
4721	0	2021-08-29 23:20:00.000	2	0
4722	0	2021-08-29 23:20:00.000	2	0
4710	1	2021-08-29 23:20:00.000	1	0
4709	0	2021-08-29 23:20:00.000	1	0
4708	1	2021-08-29 23:20:00.000	1	0
4707	0	2021-08-29 23:20:00.000	1	0
4706	1	2021-08-29 23:20:00.000	1	0
4705	0	2021-08-29 23:20:00.000	1	0
4704	1	2021-08-29 23:20:00.000	1	0
4703	0	2021-08-29 23:20:00.000	1	0
4702	0	2021-08-29 23:20:00.000	1	0
4701	1	2021-08-29 23:20:00.000	1	0
4700	1	2021-08-29 23:20:00.000	1	0
4699	0	2021-08-29 23:20:00.000	1	0

Il·lustració 27 Registres a la taula HIST\_DadesD

### 3.2.2.2.2 HIST\_DadesI

En aquesta taula s'emmagatzemen els valors de tipus integer del historitzador.

HIST_DadesI		
<b>idTag</b>	int	Identificador de Tag
<b>valor</b>	int	Valor del tag INT
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>idproced</b>	smallint	Procedència. Valor de validació de la dada. Definim un integer per tal de definir l'origen de la dada
<b>qualitat</b>	smallint	Informació de qualitat de la dada

idTag	valor	data	idproced	qualitat
5564	70665	2021-08-23 16:00:00.000	0	0
5382	3104788	2021-08-23 15:00:00.000	0	0
5386	5562102	2021-08-23 15:00:00.000	0	0
5395	12846030	2021-08-23 15:00:00.000	0	0
5402	8102161	2021-08-23 15:00:00.000	0	0
5406	3094799	2021-08-23 15:00:00.000	0	0
5409	10503	2021-08-23 15:00:00.000	2	0
5412	98370	2021-08-23 15:00:00.000	0	0
5414	1445773	2021-08-23 15:00:00.000	0	0
5418	2316666	2021-08-23 15:00:00.000	0	0
5419	6786216	2021-08-23 15:00:00.000	0	0
5420	19259	2021-08-23 15:00:00.000	2	0
5421	91977600	2021-08-23 15:00:00.000	0	0
5423	2925102	2021-08-23 15:00:00.000	0	0
5426	2421789	2021-08-23 15:00:00.000	0	0
5428	172923	2021-08-23 15:00:00.000	0	0
5431	4220133	2021-08-23 15:00:00.000	0	0

Il·lustració 28 Registres a la taula HIST\_DadesI



### 3.2.2.2.3 HIST\_DadesR

En aquesta taula s'emmagatzemen els valors de tipus real del historitzador.

HIST_DadesR		
<b>idTag</b>	int	Identificador de Tag
<b>valor</b>	real	Valor del tag en format REAL. En cas de ser idTag fictici pot tenir valor de cabal negatiu (en m3)
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>idproced</b>	smallint	Procedència. Valor de validació de la dada. Definim un integer per tal de definir l'origen de la dada
<b>qualitat</b>	smallint	Informació de qualitat de la dada

idTag	valor	data	idproced	qualitat
5161	0	2021-08-29 13:50:00.000	2	0
5162	43,25	2021-08-29 13:50:00.000	2	0
5163	0	2021-08-29 13:50:00.000	2	0
5164	64,93	2021-08-29 13:50:00.000	2	0
5165	0	2021-08-29 13:50:00.000	2	0
5107	1,83	2021-08-29 13:50:00.000	1	0
5105	2,12	2021-08-29 13:50:00.000	1	0
5106	0	2021-08-29 13:50:00.000	2	0
5108	823,5	2021-08-29 13:50:00.000	2	0
5109	5,12	2021-08-29 13:50:00.000	2	0
5104	1,27	2021-08-29 13:50:00.000	1	0
5103	3470,63	2021-08-29 13:50:00.000	1	0
5101	4,13	2021-08-29 13:50:00.000	1	0
5100	72,05	2021-08-29 13:50:00.000	2	0
5084	2,74	2021-08-29 13:50:00.000	2	0
5085	6,42	2021-08-29 13:50:00.000	2	0
5086	0	2021-08-29 13:50:00.000	2	0
5092	3,58	2021-08-29 13:50:00.000	2	0

Il·lustració 29 Registres a la taula HIST\_DadesR

### **3.2.2.3 Taules de dades manuals**

En les taules manuals les dades son emplenades de forma manual per l'administrador del historitzador. En un futur la introducció d'aquestes dades seria mes correcte poder realitzar-les desde un aplicatiu extern al SQL Server.

Com hem comentat anteriorment, per mantenir la traçabilitat i la integritat de les dades emmagatzemades al historitzador, necessitem crear les taules de valors rectificats del historitzador. D'aquesta manera la consistència de la base dades serà correcta.

Es programaran procediments al SQL Server diàriament per tal que via e-mail es pugui informar al administrador del historitzador de quines son les dades que no s'han registrat o quines s'han registrat però son incorrectes.

Repliquem en conseqüència les tres taules de dades del historitzador dins de l'àrea de taules de dades manuals. Aquestes taules son:

- HIST\_RectifiR  
Emmagatzema rectificacions de valors de tipus REAL
- HIST\_RectifiI  
Emmagatzema rectificacions de valors de tipus INTEGER
- HIST\_RectifiD  
Emmagatzema rectificacions de valors de tipus DIGITAL

Així mateix definim la taula d'anomalies necessària per a poder introduir dades del cabalímetres ficticis.

- HIST\_Anomalies  
Emmagatzema valors de tipus INT dels totalitzats dels cabalímetres ficticis.

### 3.2.2.3.1 HIST\_RectifD

En aquesta taula tenim els valors de tipus digital rectificats dels valors de HIST\_DadesD.

HIST_RectifD		
<b>idTag</b>	int	Identificador de tag
<b>valor</b>	bit	Valor del tag en format DIGITAL
<b>valorant</b>	bit	Valor abans de rectificar, considerat erroni
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>datarecti</b>	datetime	Data en què s'ha fet la rectificació
<b>idproced</b>	smallint	id de procedència abans de la rectificació ( <a href="#">veure taula CFG_TipusProced</a> )
<b>qualitat</b>	smallint	Informació de qualitat de la dada

### 3.2.2.3.2 HIST\_RectifI

En aquesta taula tenim els valors de tipus integer rectificats dels valors de HIST\_DadesI.

HIST_RectifI		
<b>idTag</b>	int	Identificador de tag
<b>valor</b>	int	Valor del tag en format INT
<b>valorant</b>	int	Valor abans de rectificar, considerat erroni
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>datarecti</b>	datetime	Data en què s'ha fet la rectificació
<b>idproced</b>	smallint	id de procedència abans de la rectificació ( <a href="#">veure taula CFG_TipusProced</a> )
<b>qualitat</b>	smallint	Informació de qualitat de la dada

### 3.2.2.3.3 HIST\_RectifR

En aquesta taula tenim els valors de tipus reals rectificats dels valors de HIST\_DadesR.

HIST_RectifR		
<b>idTag</b>	int	Identificador de tag
<b>valor</b>	real	Valor del tag en format REAL
<b>valorAnt</b>	real	Valor abans de rectificar, considerat erroni
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>dataRectif</b>	datetime	Data en què s'ha fet la rectificació
<b>idProced</b>	smallint	id de procedència abans de la rectificació ( <a href="#">veure taula CFG_TipusProced</a> )
<b>qualitat</b>	smallint	Informació de qualitat de la dada

#### 3.2.2.3.4 HIST\_Anomalies

En aquesta taula s'introdueix un valor cada cop que es buida/omple una canonada, es posa el valor del volum afegit o tret en 1 hora en m3.

HIST_Anomalies		
<b>idTag</b>	int	identificador de tag. El tag serà un dels ficticis (hi ha 1 cabal fictici per zona)
<b>valor</b>	int	Valor del tag en format INT
<b>data</b>	datetime	Datetime del valor de tag en format aaaa-dd-mm hh:mm:ss.mmm
<b>comentari</b>	varchar(100)	Comentari sobre si omplim canonada (valor serà positiu) o si buidem canonada (valor serà negatiu)

#### 3.2.2.4 Taules de valors calculats

En les taules de valors calculat les dades poden venir desde el CitectSCADA, el aplicatiu WaterGEMS... i corresponen a valors que son calculats.

### 3.3 Implementació del registre dades desde CitectSCADA

Una vegada hem definit la estructura de les taules necessàries on emmagatzemarem les dades ja ens podem centrar en la programació en el costat SCADA per tal que omplim les taules amb dades de camp.

#### 3.3.1 Historitzador

La implementació del historitzador es realitza en CiCode, ja que ens permet la possibilitat de poder fer tot l'aplicatiu a mida. Ara llistem les rutines emprades per a possibilitar el registre del historitzador en continu.

#### 3.3.2 Registre de dades Historian en servidors CitectSCADA

Via event horari, que s'executa cada 10 minuts es llença la crida a la funció:

[HIST\\_Evt\\_GuardaLecturesSQL](#) (Codi a l'annex)

Aquesta funció obre una tasca:

[HIST\\_GuardaLecturesSQL](#) (Codi a l'annex)

que realitzarà el registre de les dades en la Base de dades.

Els tags que volem enregistrar estan definits en un arxiu dBaseIII:

HIST\_Tags.DBF

El format d'aquest arxiu:

NAME	Nom del tag
TYPE	Tipus de dada
ENG_ZERO	Zero d'escala
ENG_FULL	Rang d'escala
ENG_UNITS	Unitats d'enginyeria
FORMAT	Format numèric
COMMENT	Comentari
TIPO	Tipus de variable
VAR	Indicador de variable
ENABLE	Habilitació

NAME	TYPE	ENG_ZERO	ENG_FULL	ENG_UI	FORMAT	COMMENT	TIPO	VAR	ENABLE
ATD01_FTR_T01_PV	INT	-10.0	500.0	L/s	###.#EU	Reg. analògic cabal	0	0	1
ATD01_LTR_D01_PV	INT	0.0	25.5	m	###.#EU	Reg. analògic nivell	3	0	1
ATD01_MOV_T01_Y	INT	0	100	%	###.#EU	Reg. analògic % obert v.princ.	4	0	1
ATD01_MOV_T02_Y	INT	0	100	%	###.#EU	Reg. analògic % obert v.byp.	4	0	1
ATD02_FTR_T01_PV	INT	-6.0	300.0	L/s	###.#EU	Reg. analògic cabal	0	0	1
ATD02_LTR_T01_Y	INT	0	100	%	###.#EU	Apertura vàlvula reguladora	4	0	1
ATD02_LTR_D01_PV	INT	0	16.32	m	###.#EU	Reg. analògic nivell	3	0	1
ATD02_MOV_T02_Y	INT	0	100	%	###.#EU	Apertura vàlvula seccionadora	4	0	1
ATD04_FTR_G01_PV	INT	-2.0	100.0	L/s	###.#EU	Reg. analògic cabal	0	0	1
ATD04_LTR_D01_PV	INT	0	4.08	m	###.#EU	Reg. analògic nivell	3	0	1
ATD04_PTR_G01_PV	INT	-0.1224	6.12	kg/cm²	###.#EU	Reg. an. pressió canonada	1	0	1
ATT02_FTR_G01_PV	REAL	0	25	L/s	###.#EU	Cabal Carburors Metàlics	0	1	1
ATT03_FTR_T01_D_PV	INT	-20.0	1000.0	L/s	####.#EU	Reg. an. cabal directe	0	0	1
ATT03_FTR_T01_I_PV	INT	-20.0	1000.0	L/s	####.#EU	Reg. an. cabal invers	0	0	1
ATT03_MOV_P01_Y	INT	0	100	%	###.#EU	Reg. analògic % obert v.P01	4	0	1
ATT03_MOV_P02_Y	INT	0	100	%	###.#EU	Reg. analògic % obert v.P02	4	0	1
ATT03_MOV_P03_Y	INT	0	100	%	###.#EU	Reg. analògic % obert v.P03	4	0	1
ATT03_PTR_T01_PV	INT	-0.2	10.0	kg/cm²	###.#EU	Reg. an. pressió canonada	1	0	1

*Il·lustració 30 Tags a enregistrar pel Historian*

Quan la tasca vol emmagatzemar el valor de tag sol·licitat en la base de dades, diferenciem tres tipus de lectures possibles:

- Si el tag pertany a un cabalímetre SOFREL que comunica via 3G cada 8h, CitectSCADA agafa el seu valor del servidor de tendències de 8h abans.
- En cas contrari, si el tag es registrat en el servidor de tendències el sistema obtindrà el seu valor de fa 10 minuts
- Si el tag no te tendència el sistema obtindrà el seu valor actual via TagSubscribe. El tag serà subscrit si no ho era.

El motiu d'aquesta diferenciació és que el cost en temps de llegir una dada del servidor de tendències respecte a una lectura via TagSubscribe, lectura subscrita de la dada del PLC, és molt menor lo que ens permet que la tasca pugui executar-se en menys d'un minut. Així mateix hem de tenir en compte que el número de tags subscrits no pot ser infinit per tant la solució de llegir el valor del servidor de tendències és més escalable.

Hi ha 3 formes d'obtindre el valor de un tag al CitectSCADA:

- Fer una lectura directa del tag  
Implica que fem la petició de lectura d'aquell registre al PLC i el driver de comunicacions gestiona la lectura. En funció del driver de comunicacions, per fer la lectura del registre, demanarà un paquet d'una mida fixada on es troba la dada, el PLC la retornarà i llavors arribarà el valor al CitectSCADA.  
Com es pot veure, es ineficient si hem de llegir dades de diferents posicions de memòria de forma repetitiva.
- Subscriure el tag al CitectSCADA.  
CitectSCADA ens ofereix eines per tal de subscriure tags. El funcionament és que CitectSCADA te un buffer de dades on subscrivim els tags que volem que s'interroguin contínuament. Això permet que CitectSCADA optimitzi les comunicacions per tal de tenir el valor més recent (actual) en tot moment.

- Llegir el valor del tag de servidor de tendències. Aquesta última opció ens permet tenir el valor del tag fent una lectura del valor de servidor de tendències. Com el servidor de tendències registra senyals cada 10 segons podem llegir aquest valor en continu i l'accés és instantani

Actualment el nostre sistema CitectSCADA del CAT està enregistrant 1157 tags al historitzador.

Les dades no s'enregistren directament sobre la base de dades principal, si no que emmagatzemem la sentència SQL a executar en la BBDD local del servidor CitectSCADA que executa la rutina com un string:

```
HIST_sp_AfegirDada 'ATD01_FTR_T01_PV', '-10.0', '500.0', 'L/s', 'Reg. analògic cabal', 0, 101.61, '10-02-21 12:17:00'
```

El procediment emmagatzemat de SQL:

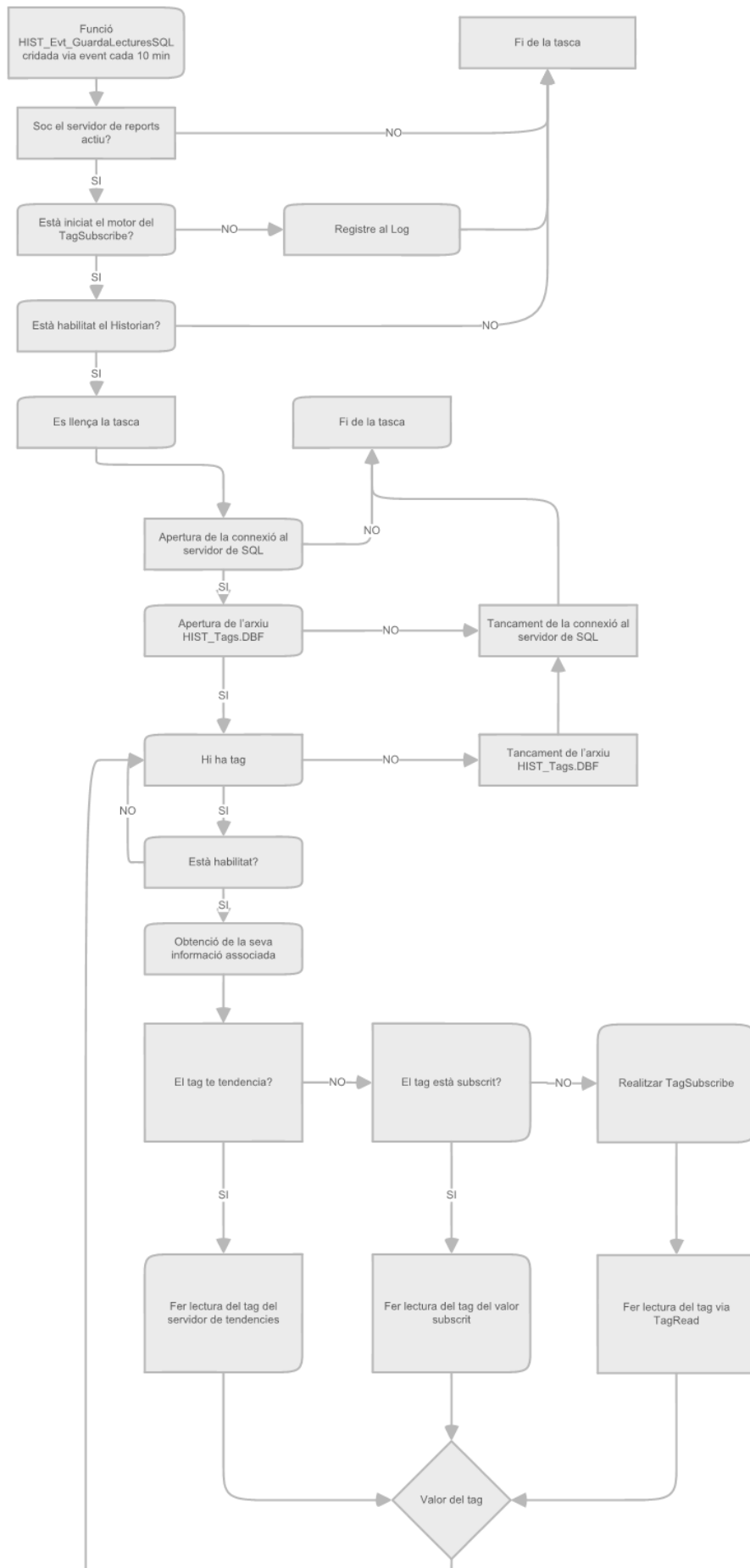
[HIST\\_sp\\_AfegirDada](#) (Codi a l'annex)

és el que posarà la dada a la taula corresponent com analitzarem més endavant.

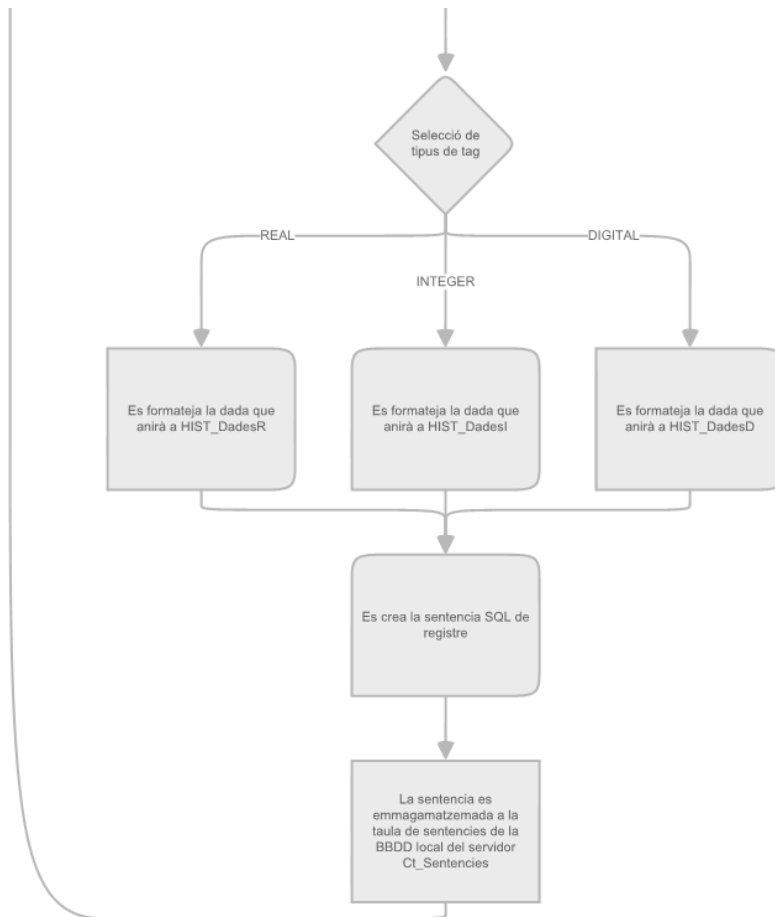
Els motius pels que enregistrem la sentència complerta en local i no directament sobre el servidor SQL principal:

- La execució massiva de les sentències en xarxa desde el servidor CitectSCADA és ineficient ja que col·lapsa la xarxa.
- El registre de la sentència en la base de dades local, que està allotjada al mateix equip que el servidor CitectSCADA és gairebé instantània.
- En cas de fallada de la connectivitat amb el servidor SQL, CitectSCADA continua enregistrant les sentències en local amb lo que no perdrem dades.
- Una vegada son tractades queden com a registre històric per a poder comprovar si hi ha algun tipus de problema.

Lògicament si el cost en temps de executar el registre de la sentència directament sobre la base de dades principal no és assumible, el cost d'executar la sentència directament sobre el servidor SQL principal ho és encara menys.







*Il·lustració 31 Diagrama d'execució de la generació de la sentència del Historitzador*

La sentència és emmagatzemada en el servidor local de SQL via la funció de CiCode:

[HIST GuardarSentenciaSQL](#)

*(Codi a l'annex)*

Aquesta enregistra la sentència SQL al servidor SQL (SCADA01\SQLEXPRESS o SCADA02\SQLEXPRESS) mitjançant el Stored Procedure:

[sp\\_Insert\\_Sentencia](#)

*(Codi a l'annex).*

D'aquesta manera anem omplint la taula sentencies de la BD local de cada servidor CitectSCADA:

sentencies		
<b>ID</b>	numeric(18, 0)	Identificador del registre
<b>Data</b>	datetime	Data/hora del registre
<b>idOp</b>	smallint	Identificador de la operació de la sentencia
<b>Sentencia</b>	varchar(254)	Sentencia a executar en format SQL
<b>trasp</b>	smallint	Flag per indicar si la sentencia ha sigut tractada

En aquesta taula s'emmagatzemen les sentencies que s'han d'executar o que ja han sigut executades.

ID	Data	idOp	sentencia	trasp
97292053	2021-08-01 10:30:46.000	2	HIST_sp_AfegirDada 'VVD06_FTR_G01_PV', '0', '50', 'L/s', 'Cabal entrada', 0, 2, 0.00, '01-08-21 10:20:00'	0
97292054	2021-08-01 10:30:46.000	2	HIST_sp_AfegirDada 'VVD06_LTR_D01_PV', '0', '6.12', 'm', 'Nivell dipòsit D01', 3, 2, 3.01, '01-08-21 10:20:00'	0
97292055	2021-08-01 10:30:46.000	1	HIST_sp_AfegirDada 'VVD06_MOV_G01_O', ' ', ' ', ' ', 'Vàlv. seccionament oberta', 100, 1, 0, '01-08-21 10:30:00'	0
97292056	2021-08-01 10:30:46.000	1	HIST_sp_AfegirDada 'VVD06_MOV_G01_T', ' ', ' ', ' ', 'Vàlv. seccionament tancada', 101, 1, 1, '01-08-21 10:30:00'	0
97292057	2021-08-01 10:30:46.000	2	HIST_sp_AfegirDada 'VVD06_MOV_G01_Y', '0', '100', '%', '% Obertura vàlv. Seccionament', 4, 2, 0.00, '01-08-21 10:20:00'	0
97292058	2021-08-01 10:30:46.000	1	HIST_sp_AfegirDada 'VVD06_MOV_G02_O', ' ', ' ', ' ', 'Vàlv. MOV-G02 oberta', 100, 1, 0, '01-08-21 10:30:00'	0
97292059	2021-08-01 10:30:46.000	1	HIST_sp_AfegirDada 'VVD06_MOV_G02_T', ' ', ' ', ' ', 'Vàlv. MOV-G02 tancada', 101, 1, 1, '01-08-21 10:30:00'	0
97292060	2021-08-01 10:30:46.000	2	HIST_sp_AfegirDada 'VVD06_PTR_G01_PV', '0', '16.32', 'kg/cm²', 'Reg. analògic presión llegada tubería', 1, 2, 7.58, '01-08-21 10:20:00'	0
97292061	2021-08-01 10:30:46.000	2	HIST_sp_AfegirDada 'VVT02_FTR_G01_PV', '0', '25', 'L/s', 'Cabal Plana de Sant Miquel', 0, 2, 0.22, '31-07-21 21:30:00'	0
97292005	2021-08-01 10:30:45.000	2	HIST_sp_AfegirDada 'VCB18_PTR_I01_PV', '0', '40', 'kg/cm²', 'Presión impulsió', 1, 2, 30.96, '01-08-21 10:20:00'	0
97292006	2021-08-01 10:30:45.000	2	HIST_sp_AfegirDada 'VCB18_PTR_T01_PV', '0', '2.55', 'kg/cm²', 'Presio ambada als dipòsits', 1, 2, 0.62, '01-08-21 10:20:00'	0
97292007	2021-08-01 10:30:45.000	2	HIST_sp_AfegirDada 'VCD01_FTR_T01_D_PV', '-2.0', '100.0', 'L/s', 'Reg. analògic cabal directe', 0, 2, 60.20, '01-08-21 10:20:00'	0
97292008	2021-08-01 10:30:45.000	2	HIST_sp_AfegirDada 'VCD01_FTR_T01_I_PV', '-2.0', '100.0', 'L/s', 'Reg. analògic cabal invers', 0, 2, -0.00, '01-08-21 10:20:00'	0
97292009	2021-08-01 10:30:45.000	2	HIST_sp_AfegirDada 'VCD01_FTR_T02_PV', '-2.0', '100.0', 'L/s', 'Reg. analògic cabal sortida', 0, 2, 14.37, '01-08-21 10:20:00'	0

*Il·lustració 32 Registres en la taula sentencies*

Les sentencies que ja han sigut executades son mogudes a la taula sentencies\_trasp periòdicament via el procediment emmagatzemat del servidor principal SQL:

[sp\\_Mant\\_Sentencies\\_scada01\\_02](#)

(Codi a l'annex)

per tal de que la taula sentencies no es faci massa gran.

Les sentencies a la taula sentencies\_trasp romanen aquí com a registre històric de sentencies fins que periòdicament s'esborren.

Sentencies_trasp		
<b>ID</b>	numeric(18, 0)	Identificador del registre
<b>Data</b>	datetime	Data/hora del registre
<b>idOp</b>	smallint	Identificador de la operació de la sentencia
<b>Sentencia</b>	varchar(254)	Sentencia a executar en format SQL
<b>trasp</b>	smallint	Flag per indicar si la sentencia ha sigut tractada

El camp idOp ens dona informació de quin tipus de sentència és. Es poden anar afegint funcionalitats i ens facilita saber quin procediment volem executar:

0: Hist\_SQL via TagRead

1: Hist\_SQL via TagSubscribe

2: Hist\_SQL via Trend

...

5: Validació de cabals

...

El camp trasp ens dona informació, de en quin punt es troba la sentència registrada:

0: Per defecte No tractada

2: Ready Marcada pel servidor INET per a poder ser copiada

4: Copiada Copiada en el servidor INET

La importància d'aquest camp es poder saber quines sentències no han sigut traspassades al servidor principal de SQL per al seu tractament.

### 3.4 Implementació registre dades en SQL Server

Ara ja tenim les sentències SQL en la taula sentències de tots dos servidors CitectSCADA.

Al servidor principal inet per la gestió de les sentències definim la taula sentències .En aquesta taula rebrem les sentències que tenen els servidors SCADA scada01/scada02 per la seu posterior tractament.

sentències		
<b>ID</b>	numeric(18, 0)	Identificador del registre
<b>Data</b>	datetime	Data/hora del registre
<b>idOp</b>	smallint	Identificador de la operació de la sentència
<b>Sentència</b>	varchar(254)	Sentència a executar en format SQL
<b>trasp</b>	smallint	Flag per indicar si la sentència ha sigut tractada

El camp trasp ens dona informació, de en quin punt es troba la sentència registrada:

0: Per defecte No tractada

2: Ready Marcada pel servidor INET per a poder ser executada

4: Copiada Executada en el servidor INET

ID	data	idOp	sentència	trasp
91254272	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'EMD05_MOV_U01_O', ' ', ' ', 'V.uraniego oberta', 100, 1, 1, '30-08-21 10:30:00'	4
91254273	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'EMD05_MOV_U01_T', ' ', ' ', 'V.uraniego tancada', 101, 1, 0, '30-08-21 10:30:00'	4
91254274	2021-08-30 10:30:34.000	2	HIST_sp_AfegirDada 'EMD07_FTR_T01_PV', '-1', '50.0', 'L/s', 'Reg. analògic cabal', 0, 2, -0.03, '30-08-21 10:20:00'	4
91254275	2021-08-30 10:30:34.000	2	HIST_sp_AfegirDada 'EMD07_LTR_D01_PV', '0', '6.12', 'm', 'Reg. analògic nivell', 3, 2, 2.93, '30-08-21 10:20:00'	4
91254276	2021-08-30 10:30:34.000	2	HIST_sp_AfegirDada 'EMD07_PTR_T01_PV', '-0.204', '10.2', 'kg/cm²', 'Reg. analògic pressió canonada', 1, 2, 2.20, '30-08-21 10:20:00'	4
91254277	2021-08-30 10:30:34.000	2	HIST_sp_AfegirDada 'EMT03_FTR_R01_PV', '2.0', '100.0', 'L/s', 'Reg. analògic cabal', 0, 2, -0.10, '30-08-21 10:20:00'	4
91254278	2021-08-30 10:30:34.000	2	HIST_sp_AfegirDada 'EMT03_LTR_D01_PV', '0', '4.08', 'm', 'Reg. an. nivell dip1 S.Jaume', 3, 2, 2.42, '30-08-21 10:20:00'	4
91254279	2021-08-30 10:30:34.000	2	HIST_sp_AfegirDada 'EMT03_LTR_E01_PV', '0', '4.08', 'm', 'Reg. an. nivell dip2 S.Jaume', 3, 2, 0.00, '30-08-21 10:20:00'	4
91254192	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'CPB00_MOV_301_O', ' ', ' ', 'V.impulsió oberta g3', 100, 1, 1, '30-08-21 10:30:00'	4
91254193	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'CPB00_MOV_301_T', ' ', ' ', 'V.impulsió tancada G3', 101, 1, 0, '30-08-21 10:30:00'	4
91254194	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'CPB00_MOV_401_O', ' ', ' ', 'V.impulsió oberta g4', 100, 1, 1, '30-08-21 10:30:00'	4
91254195	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'CPB00_MOV_401_T', ' ', ' ', 'V.impulsió tancada g4', 101, 1, 0, '30-08-21 10:30:00'	4
91254196	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'CPB00_MOV_501_O', ' ', ' ', 'V.impulsió oberta g5', 100, 1, 1, '30-08-21 10:30:00'	4
91254197	2021-08-30 10:30:34.000	1	HIST_sp_AfegirDada 'CPB00_MOV_501_T', ' ', ' ', 'V.impulsió tancada g5', 101, 1, 0, '30-08-21 10:30:00'	4

Il·lustració 33 Conjunt de registres de la taula sentències

Definim un altra taula, Ct\_sentències\_fin on mourem les sentències que ja hagin sigut executades:

Sentències_fin		
<b>Data</b>	datetime	Data/hora del registre
<b>idOp</b>	smallint	Identificador de la operació de la sentència
<b>Sentència</b>	varchar(254)	Sentència a executar en format SQL
<b>error</b>	varchar(1024)	Registre d'error

Data	idOp	sentencia	error
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_FTR_G01_D_PV', '0', '1000', 'L/s', 'Reg. an. cabal entrada directe', 0, 2, 136.50, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_FTR_G01_I_PV', '0', '1000', 'L/s', 'Reg. an. cabal entrada invers', 0, 2, 0.00, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_FTR_G02_PV', '0', '1000', 'L/s', 'Reg. an. cabal sortida', 0, 2, 482.00, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_LTR_D01_PV', '0', '6.12', 'm', 'Reg. analògic nivell', 3, 2, 4.45, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_LTR_E01_PV', '0', '6.12', 'm', 'Reg. analògic nivell E01', 3, 2, 4.38, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	1	HIST_sp_AfegirDada 'SCD01_MOV_G01_O', ' ', ' ', ' ', 'Vàlvula MOV-G01 oberta', 100, 1, 1, '01-07-21 06:50:00'	0
2021-07-01 06:50:49.000	1	HIST_sp_AfegirDada 'SCD01_MOV_G01_T', ' ', ' ', ' ', 'Vàlvula MOV-G01 cerrada', 101, 1, 0, '01-07-21 06:50:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_PTR_G01_PV', '0', '6', 'kg/cm²', 'Reg. an. pressió entrada', 1, 2, 0.29, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD01_PTR_I01_PV', '0', '10.2', 'kg/cm²', 'Pressió dossificació entrada dip. PTR-I01', 1, 2, 1.67, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD03_FTR_T01_PV', '-1.5', '75.0', 'L/s', 'Reg. analògic cabal', 0, 2, 0.00, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD03_LTR_D01_PV', '0', '4.08', 'm', 'Reg. analògic nivell d1', 3, 2, 0.01, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD03_LTR_E01_PV', '0', '4.08', 'm', 'Reg. analògic nivell d2', 3, 2, 0.01, '01-07-21 06:40:00'	0
2021-07-01 06:50:49.000	2	HIST_sp_AfegirDada 'SCD03_MOV_T01_Y', '0', '100', '%', 'Reg. analògic % obert v.secc.', 4, 2, 0.00, '01-07-21 06:40:00'	0

*Il·lustració 34 Conjunt de registres de la taula sentencies\_fin*

Per a la gestió de les sentències dels servidors CitectSCADA i el servidor principal SQL Server, el Server agent del servidor principal inet corre dos tasques:

**3.4.1 Tasca Ct\_Manteniment sentencies\_scada01/02 + inet**

S'executa cada dia a les 07:00

Consta de dos passos:

- Pas 1 EXEC [sp\\_Mant\\_Sentencies\\_scada01\\_02](#) (Codi a l'annex)

Tasca de manteniment de taula sentencies i sentencies\_trasp de scada01/scada02.

Mou els registres de més de 60 dies de la taula [telec].sentencies de scada01 i scada02 a la taula local [telec].sentencies\_trasp

Elimina els registres de més de 180 dies de la taula [telec].sentencies\_trasp de scada01 i scada02

- Pas 2 EXEC [sp\\_Mant\\_Ct\\_Sentencies](#) (Codi a l'annex)

Mou els registres de més de 60 dies de la taula [exp].Ct\_sentencies de inet a la taula [exp].Ct\_sentencies\_fin

Elimina els registres de més de 180 dies de la taula [exp].Ct\_sentencies\_fin de inet

L'objectiu d'aquesta tasca és limitar el tamany de les taules que emmagatzemen les sentències SQL. D'una banda anem traspasant les que ja estan executades a una taula històrica per tal que el tractament de la taula principal sigui més eficient i per una altra banda eliminem les més antigues de la taula històrica. D'aquesta manera podem anar limitant el tamany de les taules a la base de dades

És important mantindre les sentències un temps en la taula de històric per a poder analitzar possibles problemes que ens puguin aparèixer. Aquest registre ens permetrà saber quina és la sentència que es va executar i en cas de fallada podem analitzar quin va ser el problema que va fer que l'execució fallés.

#### **3.4.2 Tasca Ct\_Sincronitzar sentències scada01/02 + execució inet**

S'executa cada 10 minuts al minut 5, es a dir, a 00:05, 00:15, 00:25...

Consta de dos passos:

- Pas 1 EXEC sp\_sync\_Ct\_sentències

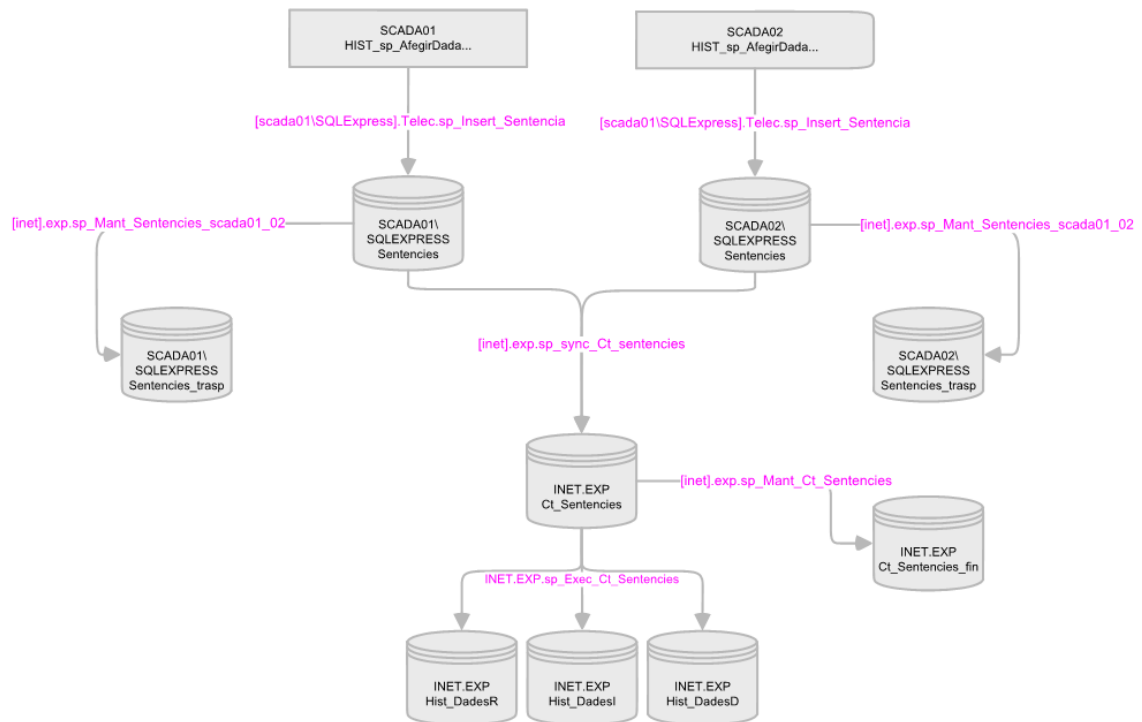
Copiem fins a 5000 sentències de la taula sentències scada01/scada02 a taula Ct\_sentències del servidor principal.

Les sentències copiades quedaran marcades amb camp trasp igual a quatre a les taules sentències de tots dos servidors CitectSCADA.

- Pas 2 EXEC sp\_Exec\_Ct\_Sentències

Executa fins a un màxim de 5000 registres, les sentències SQL emmagatzemades a la taula Ct\_sentències del servidor principal. Una vegada han sigut executades queden marcades amb el camp trasp igual a quatre.

### 3.5 Diagrama funcional de procés



Il·lustració 35 Diagrama funcional del registre de dades del historitzador

### 3.6 Tasques de comprovació del registre historitzador

Donat que el registre és en continu les 24 hores del dia, l'administrador del sistema historitzador necessita d'eines per tal de poder rebre informació diàriament de l'estat del registre.

Si una senyal es deixa de registrar, p.ex. per estar el seu PLC fora de línia, ens generarà un forat en la línia temporal d'aquella dada. Lògicament l'administrador necessitarà saber on tenim el forat i llavors prendre les mesures necessàries per tal de completar les dades.

La comprovació de la consistència de les dades els farem mitjançant procediments emmagatzemats del SQL Server. La informació d'aquests procediments, les dades incorrectes, els forats a la línia temporal... els hi farem arribar via e-mail al administrador del sistema.

El SQL Server agent correrà diàriament una tasca:

- Tasca HIST\_CheckDadesDiaHistorian

Cada dia a les 10:00

Testeja les dades emmagatzemades al Historian a les taules:

HIST\_DadesD

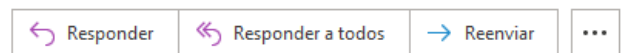
HIST\_DadesI

HIST\_DadesR

en busca de pèrdua de dades (forats) en les taules.

Si hi trobat dades que falten, envia un e-mail al administrador del sistema historitzador informant dels registres que falten per tal que es pugui analitzar el perquè de la pèrdua de dades i manualment es puguin introduir els valors adients.

sp\_Hist\_CheckDiario\_R



data	idTag	tag
2021-07-06 14:30:00.000	20083	CPB00_PM1000_KVAM_PV
2021-07-06 14:30:00.000	20084	CPB00_PM1000_KVARM_PV
2021-07-06 14:30:00.000	20085	CPB00_PM1000_KWMM_PV

(3 filas afectadas)

*Il·lustració 36 E-mail diari que informa al administrador del Historian d'incidències a la taula HIST\_Dades\_R*



La tasca consta de tres passos, cadascun dels quals analitza les tres taules de dades del historitzador.

Pas 1 EXEC exp.dbo.sp\_Hist\_CheckDiario\_D

Tasca de comprovació en la taula de tipus digital

Pas 2 EXEC exp.dbo.sp\_Hist\_CheckDiario\_I

Tasca de comprovació en la taula de tipus integer

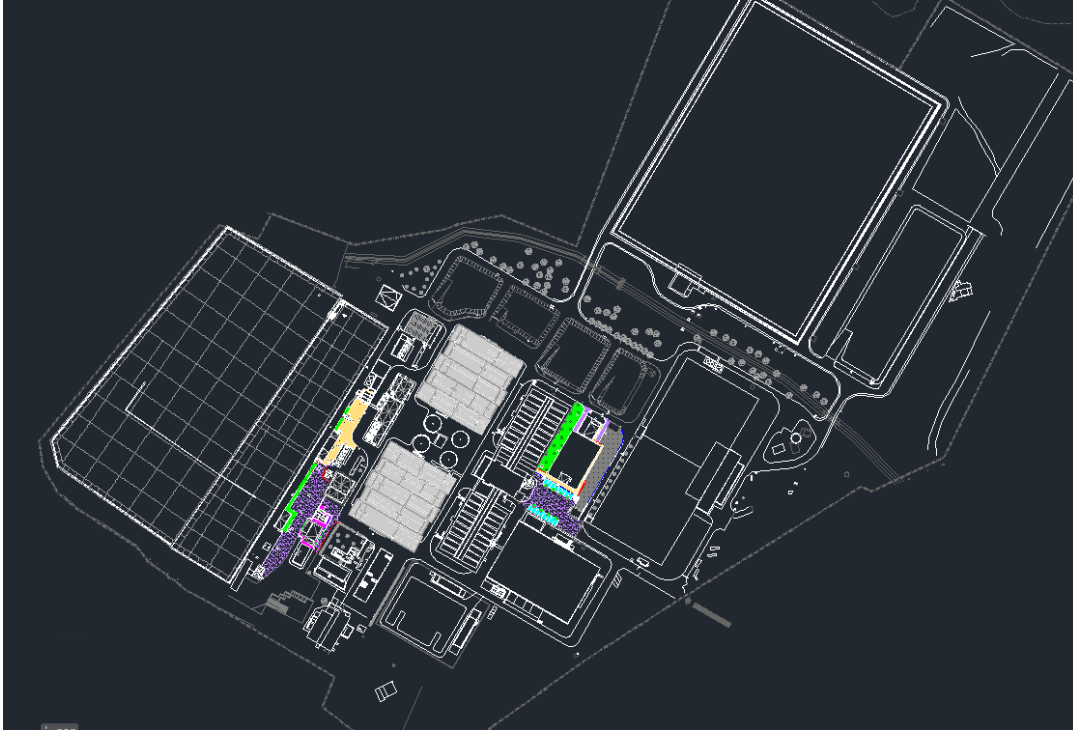
Pas 3 EXEC exp.dbo.sp\_Hist\_CheckDiario\_R

Tasca de comprovació en la taula de tipus real

El codi de tots procediments estan llistats al annex

## 4 Plànols.

### 4.1 Localització de la ETAP del CAT



Il·lustració 37 Instal·lacions de la ETAP



Il·lustració 38 Localització de la ETAP del CAT

Les instal·lacions de la ETAP es troben a la Ctra. N-340, km 1094 al terme municipal de L'Ampolla (Baix Ebre).



*Il·lustració 39 Localització de les instal·lacions dels servidors*

Els servidors que implementen el historitzador estan situats en la sala de control de la ETAP.

## 5 Pressupost.

El pressupost el dividirem en el equips necessaris, software i llicències i finalment la programació i posta en marxa del pilot de bessó digital.

El TFG versa sobre una la posta en marxa d'una funcionalitat que afegim al nostre sistema de control i visualització que és existent i ja està funcionant. Tot i això, reflectim al pressupost aquests equips i el seu software, incloent el sistema operatiu i llicències, per tal de tenir un cost real de la implementació si la féssim desde zero.

### 5.1 Equips servidors Citect SCADA, SQL Server i OpenFlows WaterSight.

Necessitem de quatre equips servidors, un per cada servidor CitectSCADA, un més pel servidor SQL i finalment un pel OpenFlows WaterSight by Bentley.

El servidor escollit és un Servidor HP ProLiant DL360 Gen9. Son servidors industrials que van muntats en rack. Això ens permet que els equips vagin connectats a la matriu que hi ha instal·lada als armaris de la sala de ordinadors i podem accedir a ells mitjançant la consola o via escriptori remot.

El hardware de cada servidor es compon de :

Ap. 1.1	EQUIPS	UNITAT	QUANTITAT	€/UNITAT	TOTAL
1.1.1	HPE DL360 Gen9 8SFF CTO Server (55258-B21)	-	2	889,00 €	1.778,00 €
1.1.2	Europe - Multilingual Localization (755258-B21 B19)	-	2	0,00 €	0,00 €
1.1.3	HPE DL360 Gen9 E5-2620v4 FIO Kit (818172-L21)	-	2	401,00 €	802,00 €
1.1.4	HPE 16GB 2Rx4 PC4-2400T-R Kit (836220-B21)	-	4	199,00 €	796,00 €
1.1.5	Factory integrated (836220-B21 0D1)	-	4	0,00 €	0,00 €
1.1.6	HPE 480GB SATA 6G MU SFF SC DS SSD (872344-B21)	-	6	405,00 €	2.430,00
1.1.7	Factory integrated (872344-B21 0D1)	-	6	0,00 €	0,00 €
1.1.8	HP Smart Array P440ar/2G FIO Controller (749974-B21)	-	2	294,00 €	588,00 €
1.1.9	HP 1U SFF Easy Install Rail Kit (734807-B21)	-	2	48,00 €	96,00 €
1.1.10	Factory integrated (734807-B21 0D1)	-	2	0,00 €	0,00 €
1.1.11	HPE 500W FS Plat Ht Plg Pwr Supply Kit (720478-B21)	-	4	128,00 €	512,00 €
1.1.12	Factory integrated (720478-B21 0D1)	-	4	0,00 €	0,00 €
1.1.13	HPE iLO Adv incl 3yr TSU E-LTU (E6U64ABE)	-	2	279,00 €	508,00 €
1.1.14	Factory integrated (E6U64ABE 0D1)	-	2	0,00 €	0,00 €
1.1.15	HPE 3Y Foundation Care NBD Service (H7J32A3)	-	1	0,00 €	0,00 €
1.1.16	HPE ProLiant DL360 Gen9 Support (H7J32A3 TT5)	-	2	193,00 €	386,00 €

El sistema operatiu de cada servidor es compona de :

Ap. 1.2	EQUIPS	UNITAT	QUANTITAT	€/UNITAT	TOTAL
1.2.1	Microsoft Windows Server Std Core 2016 2Lic NL GOV (9EM-00230)	-	1	89,00 €	89,00 €

Els serveis de instal·lació dels servidors:

Ap. 1.3	EQUIPS	UNITAT	QUANTITAT	€/UNITAT	TOTAL
1.3.1	Serveis Professionals: - Acoblament dels diferents dispositius i actualització del firmware dels mateixos a la data d'instal·lació. Configuració de les BIOS. - Instal·lació base de Windows Server 2016 amb actualitzacions a la data i .NET Framework 2.0 i 3.5. Controladors i eines de HPE habilitades. (HTEC01)	-	1	290,00 €	290,00 €

A nivell de llicències necessàries per aquests tres equips:

Ap. 1.4	LLICENCIES	UNITAT	QUANTITAT	€/UNITAT	TOTAL
1.4.1	Microsoft SQL Server 2019 Standard	-	1	775,00 €	775,00 €
1.4.2	CitectSCADA Server License 50000 points	-	2	15.000 €	30.000 €

Resumint:

Ap. 1	EQUIPS	TOTAL	QUANT.	IVA	TOTAL
1.1	Servidors HP ProLiant DL360 Gen9	7.896,00 €	4	21%	38.216,64 €
1.2	S.O. Servidors	89,00 €	4	21%	430,76 €
1.3	Serveis d'instal·lació	290,00 €	4	21%	1403,60 €
1.4	Llicències	30.775,00 €	1	Inclòs	30.775,00 €
		39.228,00 €		-	70.826,00 €

## 5.2 Equips OpenFlows WaterSight by Bentley.

En el cas del hardware/software necessari per al simulador hidràulic OpenFlows WaterSight by Bentley, la implementació d'aquest aplicatiu es fa al núvol, seguint el PC anteriorment ofertat qui fa de nexa entre la nostra base de dades i el sistema CitectSCADA i la seva aplicació al núvol. Bentley ens demana el PC per tal de realitzar la instal·lació del seu software.

Afegim una part de despesa d'un programador per tal de realitzar la integració i el suport a la posta en marxa de Bentley. A nivell tècnic farà de pont entre Bentley i el Consorci d'Aigües de Tarragona. Totes les consultes, recolzament tècnic, programació, proves de les comunicacions, ja siguin via OPC DA Server o desde les taules del SQL Server, i seguiment de la posta en marxa i interconnexió dels components que formen el nostre sistema seran responsabilitat d'aquest tècnic.

Ap. 2.1	EQUIPS	UNITAT	QUANT.	€/UNITAT	TOTAL
2.1.1	OpenFlows WaterSight by Bentley. Configuració i posta en marxa ajustat al nostre sistema.	-	1	50.000,00 €	50.000,00 €

Ap. 2.2	EQUIPS	UNITAT	QUANT.	€/UNITAT	TOTAL
2.2.1	Suport a la posta en marxa del OpenFlows WaterSight by Bentley.	d	10	50,00 €	500,00 €

Resumint:

Ap. 2	EQUIPS	TOTAL	QUANT.	IVA	TOTAL
2.1	Servidors HP ProLiant DL360 Gen9	50.000,00 €	1	Inclòs	50.000,00 €
2.2	S.O. Servidors	500,00 €	1	Inclòs	500,00 €
		50.500,00 €		-	50.500,00 €

### 5.3 Equips clients

Per als equips clients de la sala de control considerem 2 PCs de sobretaula amb 2 pantalles cadascú. Son els equips amb els que els operadors treballaran per tal de tenir control de tota la xarxa.

El hardware de cada client es compon de :

Ap. 3.1	EQUIPS	UNITAT	QUANTITAT	€/UNITAT	TOTAL
3.1.1	PC Dell Optiplex 380 per a dos clients CitectSCADA per la sala de control	-	2	599,37 €	1198,74 €

Els perifèrics necessaris:

Ap. 3.2	EQUIPS	UNITAT	QUANTITAT	€/UNITAT	TOTAL
3.2.1	Monitor - HP 27m, 27" Full HD, IPS, 60 Hz, 5 ms, HDMI, Negre per a clients CitectSCADA de la sala de control	-	4	169,00 €	676,00 €
3.2.2	Ratolí òptic USB HP amb roda de desplaçament per a clients CitectSCADA de la sala de control	-	2	10,89 €	21,78 €
3.2.3	Teclat amb cable HP Pavilion 300 per a clients CitectSCADA de la sala de control	-	2	30,00 €	60,00 €

El sistema operatiu:

Ap. 3.3	SOFTWARE	UNITAT	QUANTITAT	€/UNITAT	TOTAL
3.3.1	Windows 10 Pro	-	2	165,00 €	330,00 €

Les llicències necessàries:

Ap. 3.4	LLICENCIA	UNITAT	QUANTITAT	€/UNITAT	TOTAL
3.1.1	CitectSCADA Client License (unlimited points)	-	2	4800,00 €	9600,00 €

Resumint:

Ap. 3	EQUIPS	TOTAL	QUANT.	IVA	TOTAL
3.1	Clients PC Dell Optiplex 380	1198,74 €	1	Inclòs	1198,74 €
3.2	Perifèrics	757,78 €	1	Inclòs	757,78 €
3.3	S.O. Clients	330,00 €	1	Inclòs	330,00 €
3.4	Llicències	9600,00 €	1	Inclòs	9600,00 €
		11.886,52 €		-	11.886,52 €

#### 5.4 Programació i posta en marxa del bessó digital

En aquesta partida detallem totes les feines de programació, integració i posta en marxa del sistema del bessó digital.

Ap. 4.1	CONCEPTE	UNITAT	QUANTITAT	€/UNITAT	TOTAL
4.1.1	Programació del historitzador en CitectSCADA. Creació de totes les rutines necessàries per al registre SQL. Proves i posta en marxa.	d	20	480,0 €	9.600,00 €

Ap. 4.2	CONCEPTE	UNITAT	QUANTITAT	€/UNITAT	TOTAL
4.2.1	Programació Historian en SQL Server. Creació de totes les taules i procediments necessaris per tenir totes les dades historitzades al SQL Server. Proves i posta en marxa.	d	15	480,0 €	7.200,00 €

Ap. 4.3	CONCEPTE	UNITAT	QUANTITAT	€/UNITAT	TOTAL
4.3.1	Posta en marxa i seguiment comunicacions amb OpenFlows WaterSight	d	5	480,0 €	2.400,00 €

Resumint:

Ap. 4	EQUIPS	TOTAL	QUANT.	IVA	TOTAL
4.1	Programació Historian en CitectSCADA	9.600,00 €	1	Inclòs	9.600,00 €
4.2	Programació Historian en SQL Server	7.200,00 €	1	Inclòs	7.200,00 €
4.3	Posta en marxa i seguiment comunicacions amb OpenFlows WaterSight	2.400,00 €	1	Inclòs	2.400,00 €
		19.200,00 €		-	19.200,00 €

#### 5.5 Pressupost total

Ap.	CONCEPTE	TOTAL
1	Equips servidors Citect SCADA i SQL Server	70.826,00 €
2	Equips OpenFlows WaterSight by Bentley.	50.500,00 €
3	Equips clients	11.886,52 €
4	Programació i posta en marxa del bessó digital	19.200,00 €
		152.412,52 €



## **6 Plec de condicions.**

### **6.1 Objectiu**

L'objectiu d'aquest plec de condicions es definir i establir les limitacions i condicions sobre la distribució, utilització i responsabilitats relatives a aquesta eina de historització de dades.

### **6.2 Materials**

Aquest TFG consisteix en una la programació i posta en marxa de les eines necessàries per la implementació del pilot d'un bessó digital, basat en un sistema SCADA CitectSCADA, una base de dades Microsoft SQL Server i un simulador hidràulic OpenFlows WaterSight. La programació es realitzarà als equips propietat del Consorci d'Aigües de Tarragona a les instal·lacions situades a la ETAP al terme municipal de L'Ampolla .

### **6.3 Condicions Generals**

La eina queda en possessió del Consorci d'aigües de Tarragona com a producte final e indivisible.

### **6.4 Condicions Econòmiques**

Tot el llicenciatge i equips físics els proporciona el Consorci d'Aigües de Tarragona.

### **6.5 Condicions de Seguretat**

La eina ha d'estar correctament instal·lada i configurada per a evitar qualsevol anomalia en el funcionament d'aquesta.

## 7 Annexes.

### 7.1 Codi de la implementació de l'historitzador en CitectSCADA

#### 7.1.1 HIST\_Evt\_GuardaLecturesSQL

S'executa cada 10 minuts. És la encarregada de llençar la tasca que registrarà tots els tags historitzats en la base de dades.

```
/*
** FUNCIO:      HIST_Evt_GuardaLecturesSQL
**
** DESCRIPCIÓ: Cridada via EVENT 'EvWaterGEMS' cada 10m
**
**           Name:      EvWaterGEMS
**           Time:      00:00:30
**           Period:    00:10:00
**           Action:    HIST_Evt_GuardaLecturesSQL()
**           Comment:   Registre Historian en SQL (via TagSubscribe)
**
**
** Aquesta funció emmagatzema a les taules SQL server les cada 10 minut.
** Guardarem les lectures a la BDD exp
**
**           CFG_Tags      Taula de tags de la instal.lació
**
**           HIST_DadesR   Valors REAL
**           HIST_DadesI   Valors INT
**           HIST_DadesD   Valors DIGITAL
*/
INT
FUNCTION HIST_Evt_GuardaLecturesSQL ()
  INT iHnd;

  // Comprovem si som el report server actiu
  IF PS_GetActive_RptSvr () = 1 THEN
    IF NOT CTSubs_giLocVarsInit THEN
      PS_Prog_Log ("-1", "", "HIST_Evt_GuardaLecturesSQL -> Reset
variables TagSubscribe pendent", "No s'executa el codi", "",
102);
      RETURN 0; // Si les variables no han sigut incialitzades
// no executem el codi
    END

    IF HIST_HAB THEN
      iHnd = TaskHnd ("_HIST_GuardaLecturesSQL");
      IF (iHnd > 0) AND ((TimeCurrent () - miTempsInici) > (60 *
10)) THEN
        TaskKill (iHnd);
      ELSE
        TaskNew ("_HIST_GuardaLecturesSQL", "", 8);
      END
    END
  END
END
END
```

### 7.1.2 ***\_HIST\_GuardaLecturesSQL***

Encarregada de realitzar el registre de les dades del historitzador en la Base de dades.

Va llençant les sentències SQL que registren el valor de cada un dels tags que componen el historitzador.

```
/*
** FUNCIO:      _HIST_GuardaLecturesSQL
**
** DESCRIPCIO:
*/
INT
FUNCTION _HIST_GuardaLecturesSQL ()
  INT hDev;
  INT nRec;
  INT iNumRegistres, iNumRegistresEnable;
  INT iNumTagsTrend, iNumTagsSubscribe, iNumTagsTrendSOFREL;

  STRING sNAME;           // Nom del tag al Variable.dbf
  STRING sTYPE;           // Tipus de dada al Variable.dbf
  STRING sENG_ZERO;       // Engineering zero al Variable.dbf
  STRING sENG_FULLL;      // Engineering full al Variable.dbf
  STRING sENG_UNITS;      // Unitats d'enginyeria al Variable.dbf
  STRING sFORMAT;         // Format al Variable.dbf
  STRING sCOMMENT;        // Comentari al Variable.dbf

  INT iTIPO;              // Tipus de dada
  INT iVAR;               // Tipus dada (0: normal / 1:SOFREL)
  INT iENABLE             // Es registra al SQL (1)
  INT iOrigen = -1;       // TagRead(0)/TagSubscribe (1)/Trend (2)
  INT iQualitat = 8;      // Qualitat del valor
  INT iConnexioStatus, iExecucioStatus;

  REAL rValor;

  STRING sValor;
  STRING sSQLExp;         // Expressió de la crida del procediment del SQL
  STRING sDataAux;

  INT iValorsD, iValorsI, iValorsR;
  INT iPos;
  INT iAux;

  ErrSet(1);

  miTempsInici = TimeCurrent (); // Number of seconds since 01/01/1970
  TASK_HIST_EXEC = 1;           // Registre SQL en execució

  //////////////////////////////////////
  // Obrim la connexió al SQL
  //////////////////////////////////////
  iConnexioStatus = HIST_ObrirConnexioSQL(miINET);
  IF iConnexioStatus <> 0 THEN
    RETURN -1;
  END
```

```

////////////////////////////////////
//  Registre de dades
////////////////////////////////////
hDev = DevOpen("HIST_Tags");
IF hDev = -1 THEN
    Message("@(HIST_Tags - Error)", "@(No es pot obrir la base de
dades de tags HIST)", 16);
    RETURN -1;
END

iNumRegistres = 0;
iNumRegistresEnable = 0;
iNumTagsTrend = 0;
iNumTagsTrendSOFREL = 0;
iNumTagsSubscribe = 0;
iValorsD = 0;
iValorsI = 0;
iValorsR = 0;
nRec = DevFirst(hDev);

// Llegim tots els registres
WHILE (nRec = 0) DO

    // Si el programador de tasques està actiu, pausem la execució
    // fins que acabi per tal de no interferir
    WHILE GetmiProgExec() = 1 DO
        PS_Prog_Log("-1", "", "HIST_SQL", "_HIST_GuardaLecturesSQL...
esperant que finalitzi el programador de tasques", "", 103);
        Sleep(30);
    END

    sNAME = DevGetField(hDev, "NAME");           // Nom del tag
    sTYPE = DevGetField(hDev, "TYPE");           // Tipus de dada
    sENG_ZERO = DevGetField(hDev, "ENG_ZERO");   // Engineering zero
    sENG_FULL = DevGetField(hDev, "ENG_FULL");   // Engineering full
    sENG_UNITS = DevGetField(hDev, "ENG_UNITS"); // Unitats d'enginy.
    sFORMAT = DevGetField(hDev, "FORMAT");       // Format
    sCOMMENT = DevGetField(hDev, "COMMENT");     // Comentari
    // Busquem caracter ' al comentari per ser reemplaçat per '' i no
    // no doni problemes la execució del sql
    sCOMMENT = CSV_String_Replace(sCOMMENT, "'", "'");

    // Camp TIPO (0:cabal REAL / 1:pressió REAL / ...)
    iTIPO = StrToInt(DevGetField(hDev, "TIPO"));
    // Camp VAR (0: normal / 1:SOFREL)
    iVAR = StrToInt(DevGetField(hDev, "VAR"));
    // Es registra al SQL (1)
    iENABLE = StrToInt(DevGetField(hDev, "ENABLE"));

    IF iENABLE THEN

        IF _RDB_BuscarTrendTag(sNAME) THEN
            // Trends - Podem fer la lectura dels Trends, sempre i quan
            // estigui trenderitzat
            IF iVAR = 0 THEN
                rValor = TRN_LeerValorTrend(sNAME, msDateT, msTimeT,
                    0, 1); // Lectura de 10 minuts abans
                iNumTagsTrend = iNumTagsTrend + 1;
                iOrigen = 2;
                iQualitat = 0;
                sDataAux = msDTSQLTrend;
            END IF
        END IF
    END IF
END WHILE

```

```

ELSE
    // SOFREL comunica a les 07:45, 15:45 i 23:45.
    // Aquestes son les hores en que es reben dades de
    // cada dispositiu. Com la rutina de lectura és cada
    // 10 minuts, fem la lectura de 8h abans
    rValor = TRN_LeerValorTrend(sNAME, msDateTS,
        msTimeTS, 0, 1);
    iNumTagsTrendSOFREL = iNumTagsTrendSOFREL + 1;
    iOrigen = 2;
    iQualitat = 0;
    sDataAux = msDTSQLTrends;
END
ELSE
    rValor = Round(CTSubs_ValorTag(sNAME), 2);
    // TagSubscribe - Arrodonim a 2 decimals per evitar
    // problemes amb els REAL
    // Si rebem -9999, la lectura ha sigut incorrecta
    // Agafem la lectura actual
    IF rValor = CTSubs_grErrorValue THEN
        rValor = 0.0;
    END
    iNumTagsSubscribe = iNumTagsSubscribe + 1;
    iOrigen = 1;
    iQualitat = 0;
    sDataAux = msDTSQL;
END

SELECT CASE iTIPO
    CASE 0 TO 49 // Valors REAL
        sValor = RealToStr(rValor, 8, 2);
        sValor = StrTrim(sValor);
        iPos = StrSearch(0, sValor, ",");
        sValor = StrLeft(sValor, iPos) + "." + StrMid(sValor,
            iPos + 1, StrLength(sValor) - iPos - 1);
        iValorsR = iValorsR + 1;

        // Els totalitzats els tractem a banda ja que al ser els
        // de distribució dos paraules (_H/_L) no cal colapsar el
        // TagSubscribe. Generem una rutina a banda per als
        // totalitzats
    CASE 50 TO 99 // Valors LONG
        sValor = RealToStr(rValor, 8, 0);
        sValor = StrTrim(sValor);
        iValorsI = iValorsI + 1;

    CASE 100 TO 149 // Valors DIGITAL
        IF (rValor > 0.0) THEN
            rValor = 1.0;
        ELSE
            rValor = 0.0;
        END
        sValor = RealToStr(rValor, 1, 0);
        sValor = StrTrim(sValor);
        iValorsD = iValorsD + 1;

    CASE ELSE
        Email_RunEnviarEmail("jjimenez@ccaait.cat",
            "_HIST_GuardaLecturesSQL - " + sNAME, "Tag sense
            iTIPO vàlid");
END SELECT

```

```

IF miINET THEN// Registre directe sobre BDD SQL Servidor INET
  sSQLExp = " HIST_sp_AfegirDada '" + sNAME + "', '" +
  sENG_ZERO + "', '" + sENG_FULL + "', '" + sENG_UNITS +
  "', '" + sCOMMENT + "', " + IntToStr(iTIPO) + ", " +
  IntToStr(iOrigen) + ", " + IntToStr(iQualitat) + ", " +
  sValor + ", '" + sDataAux + "'";

  iExecucioStatus = HIST_ExecutarSentenciaSQL(sSQLExp);

  sSQLExp = "HIST_sp_AfegirDada '" + sNAME + "', '" +
  sENG_ZERO + "', '" + sENG_FULL + "', '" +
  sENG_UNITS + "', '" + sCOMMENT + "', " +
  IntToStr(iTIPO) + ", " + IntToStr(iOrigen) + ", " +
  IntToStr(iQualitat) + ", " + sValor + ", '" +
  sDataAux + "'";

  iExecucioStatus = HIST_GuardarSentenciaSQL(sSQLExp,
  iOrigen);
END

  iNumRegistresEnable = iNumRegistresEnable + 1;
END
iNumRegistres = iNumRegistres + 1;
nRec = DevNext(hDev);

END
DevClose(hDev);

////////////////////////////////////
// Tanquem la connexió al SQL
////////////////////////////////////
HIST_TancarConnexioSQL();

miTempsInvertit = TimeCurrent() - miTempsInici;
TractaError(0, "Registre Historian a SQL Server acabat - Temps: " +
  IntToStr(miTempsInvertit) + " seg. -> Tags Trend=" +
  IntToStr(iNumTagsTrend) + " + Tags Trend SOFREL=" +
  IntToStr(iNumTagsTrendSOFREL) + " + TagsSubscribe=" +
  IntToStr(iNumTagsSubscribe) + " = " +
  IntToStr(iNumTagsTrend + iNumTagsTrendSOFREL +
  iNumTagsSubscribe) + " -> D/I/R=" + IntToStr(iValorsD) +
  "/" + IntToStr(iValorsI) + "/" + IntToStr(iValorsR));

TASK_HIST_EXEC = 0; // Registre SQL en execució

IF (miTempsInvertit > ((9 * 60) + 45)) THEN
  Email_RunEnviarEmail("jjimenez@ccait.cat",
  "_HIST_GuardaLecturesSQL - " + ServerInfo("Client", 0), "Temps
  execució = " + IntToStr(miTempsInvertit) + " seg.");
END

RETURN 0;
END

```

### 7.1.3 HIST\_GuardarSentenciaSQL

Funció per tal d'emmagatzemar la sentència al servidor local de SQL.

```
/*
** FUNCIO:      HIST_GuardarSentenciaSQL
**
** DESCRIPCIÓ: Guarda la sentència rebuda com a paràmetre a la taula de
**              sentències a la BBDD local.
**
**              Taula sentències
**              [Data]      Data/hora del registre
**              [idOp]      Info de la sentència
**              [sentència] Sentència SQL
**              [trasp]     Flag que indica que inet ha tractat la sentència
**
** [dbo].[sp_Insert_Sentencia]
** @DataReg datetime,      // Data/hora del registre de la sentència
**
** @idOp integer,          // 0: Hist_SQL via TagRead
**                          // 1: Hist_SQL via TagSubscribe
**                          // 2: Hist_SQL via Trend
**                          // ...
**                          // 5: Validació de cabals
**                          // ...
** @Sentencia varchar(254), // Sentència SQL
**
** @flag integer           // Flag de traspàs de info al SQL
*/
INT
FUNCTION HIST_GuardarSentenciaSQL (STRING sSentencia, INT iTipusOp = 0)
    INT iStatus, iError;
    STRING sSQLQuery, sTime;

    ErrSet(1);

    sTime = TimeToStr(TimeCurrent(), 1) + " " + TimeToStr(TimeCurrent(),
        9);

    sSQLQuery = " sp_Insert_Sentencia '" + sTime + "', '" +
        IntToStr(iTipusOp) + "', '" + sSentencia + "', " +
        IntToStr(0);

    // Executem l'stored procedure per emmagatzemar les lectures a l'sql.
    SQLSet(mhSQL, sSQLQuery);
    iStatus = SQLExec(mhSQL, "");
    SQLEnd(mhSQL);
    iError = IsError();
    IF (iError <> 0) THEN
        TractaError(0, SQLErrMsg());
        TractaError(iError, "Error en insertar sentència:" + sSQLQuery);
        RETURN (iError);
    END

    RETURN 0;
END
```

### 7.1.4 HIST\_ObrirConnexioSQL

Funció que obre la connexió SQL per poder fer el registre dels tags.

```
/*
** FUNCIO:      HIST_ObrirConnexioSQL
**
** DESCRIPCIÓ: Obre la connexió SQL pel al registre del tags
*/
INT
FUNCTION HIST_ObrirConnexioSQL(INT iINET = 1)
    INT iErrorSQL; // Control d'errors del SQL

    ErrSet(1);

    // Executem per resetejar si te algun error anterior emmagatzemat la
    // funció IsError
    iErrorSQL = IsError();
    TractaError(iErrorSQL, "HIST_ObrirConnexioSQL - Error = " +
        IntToStr(iErrorSQL));

    // CAS DELS TAGS QUE ES LLEGEIXEN VIA TagSubscribe
    // la data/hora del registre es dd-mm-aa hh:mm:00 ja que es fa cada 10
    // minuts i a efectes de SQL registrem la dd-mm-aa hh:mm quan
    // s'inicia el procés.
    // El valor del tag és el valor del tag en aquell moment via
    // Tagsubscribe
    msDTSQL = Date(2);
    msDTSQL = msDTSQL + " " + TimeHour(TimeCurrent()):#0 + ":" +
        TimeMin(TimeCurrent()):#0 + ":00";
    msDTSQL = StrLeft(msDTSQL, 2) + "-" + StrMid(msDTSQL, 3, 2) + "-" +
        StrMid(msDTSQL, 6, 2) + " " + StrRight(msDTSQL, 8);
    // dd-mm-aa hh:mm:00 actual

    // CAS DELS TAGS QUE ES LLEGEIXEN VIA Trend
    // la data/hora del valor del tag que es llegeix és la obtinguda en
    // el cas del via TagSubscribe i se li restan 10 minuts.
    // Es a dir, volem enregistrar el valor del tag de fa 10 minuts.
    // msDateT = dd/mm/yy      -> (dd/MM/yy actual) - 10 min
    msDateT = GetSubTimesTamp(4, 10, 4);
    // msTimeT = hh:mm        -> (hh:mm actual) - 10 min
    msTimeT = GetSubTimesTamp(12, 10, 4);
    // dd-mm-aa hh:mm:00 actual - 10 min
    msDTSQLTrend = CSV_StrReplace(msDateT, "/", "-") + " " + msTimeT +
        ":00";

    // En el cas dels tags de equips SOFREL, el decalatge ha de ser de 8h
    // per a assegurar dades reals.

    // msDateT = dd/mm/yy      -> (dd/MM/yy actual) - 8h
    msDateTS = GetSubTimesTamp(6, 8, 3);
    // msTimeT = hh:mm        -> (hh:mm actual) - 8h
    msTimeTS = GetSubTimesTamp(14, 8, 3);
    // dd-mm-aa hh:mm:00 actual - 8h
    msDTSQLTrends = CSV_StrReplace(msDateTS, "/", "-") + " " + msTimeTS +
        ":00";

    // Establim la connexio amb la BDD.
    mhSQL = 0;
```



```

IF iINET THEN
    // Registre directe sobre INET
    mhsQL = SQLConnect("DSN=exp;UID=citectSQL;PWD=cat2017-;");
ELSE
    // Registre sobre SQL local (scada01/scada02)
    mhsQL = SQLConnect("DSN=Telec;UID=sa;PWD=ConSORCI09;");
END

IF (mhsQL = -1) THEN
    TractaError(0, "HIST_ObrirConnexioSQL. Error de connexió al SQL
    Server");
    RETURN -1;
END

RETURN 0;
END

```

### **7.1.5 HIST\_TancarConnexioSQL**

Funció que tanca la connexió SQL.

```

/*
** FUNCIO:      HIST_TancarConnexioSQL
**
** DESCRIPCIO: Tanca la connexió SQL pel al registre del tags
*/
INT
FUNCTION HIST_TancarConnexioSQL ()
    INT iErrorSQL;

    SQLDisconnect (mhsQL);
    iErrorSQL = IsError ();

    RETURN 0;
END

```

### 7.1.6 TRN\_LeerValorTrend

Permet llegir el valor de un tag del servidor de tendències.

```
/*
**      FUNCIO:      TRN_LeerValorTrend
**
**      DESCRIPCIO: Permet llegir el valor de un tag de la seva tendencia
**
**      TRN_LeerValorTrend("PA01_FTR_001_PV", "22/09/2000", "12:30")
*/
REAL rFileData[1];

REAL
FUNCTION TRN_LeerValorTrend(STRING sTrendTag, STRING sDate, STRING sTime,
                           INT iMessage = 0, INT iOrigen = 0)

    STRING sPeriod = TrnInfo(sTrendTag, 2);
    STRING sEnd_Time;
    INT iSamples;
    INT iError;
    // The actual number of samples read. 0 if an error occurs.
    INT iTrnGetTable;
    STRING sOrigen;

    iSamples = 1;
    sEnd_Time = StrToDate(sDate) + StrToTime(sTime);

    ErrSet(1);
    iError = IsError();
    // Executem per resetejar si te algun error anterior emmagatzemat la
    // funcio IsError
    iTrnGetTable = TrnGetTable(sTrendTag, sEnd_Time, 0, iSamples,
                              rFileData[0], 1);

    iError = IsError();
    ErrSet(0);

    IF (iError <> 0) THEN
        SELECT CASE iOrigen
            CASE 1      sOrigen = "_HIST_GuardaLecturesSQL";
            CASE ELSE  sOrigen = "N/A";
        END SELECT

        Email_RunEnviarEmail("jjimenez@ccaaait.cat", sOrigen + "
            TRN_LeerValorTrend - " + ServerInfo("Client", 0),
            "Err lectura " + sTrendTag + "->Data=" + sDate +
            "-Hora=" + sTime + " TrnGetTable=" +
            IntToStr(iTrnGetTable) + " IsError=" +
            IntToStr(iError));

    END

    IF iMessage THEN
        Message("TRN_LeerValorTrend", RealToStr(rFileData[0], 10, 3),
              64);
    END

    RETURN rFileData[0];
END
```

### 7.1.7 CTSubs\_ValorTag

Permet llegir el valor subscript de un tag.

```
/*
**  FUNCIO:      CTSubs_ValorTag
**
**  DESCRIPCIO: Retorna el valor subscript de un tag.
**              Si el tag encara no ha sigut subscript, el subscribeix i
**              retorna el seu valor mitjançant un TagRead.
**
**              SI LA LECTURA DEL TAG SUBSCRIT VIA
**              SubscriptionGetAttribute ENS DONA ERROR (I PER TANT NO
**              SABEM EL VALOR) LA FUNCIO RETORNA -9999 PER TAL DE PODER
**              TRACTAR L'ERROR
*/
REAL
FUNCTION CTSubs_ValorTag (STRING sTag, INT iMensaje = 0)
  INT id;
  STRING sValor;
  QUALITY qCalidad;
  INT iErrorVariable;

  ErrSet(1);

  CodeSetMode(1, 0);

  id = CTSubs_BuscarHnd(sTag);
  IF (id >= 0) THEN
    sValor = SubscriptionGetAttribute(CTSubs_mhTagSubscripcion[id MOD
      400][id / 400], "Value");

    IF (iMensaje = 1) THEN
      IF (sValor = "") THEN
        iErrorVariable = IsError();
      END
      Message("CTSubs_ValorTag - Subscrito " + sTag, "Valor actual
        = " + sValor + " -> Real = " + RealToStr(sValor, 6,
          2) + " " + TagInfo(CTSubs_msTagSubscripcion[id MOD
            400][id / 400], 1) + "^nError = " +
          IntToStr(iErrorVariable), 64);
      RETURN -2; // És només informatiu
    END

    IF (sValor = "") THEN
      iErrorVariable = IsError();
      SELECT CASE iErrorVariable
        CASE 257 // Value is out of range
          //sValor = TagRead(sTag); // Possibilitat de fer un
          TagRead per obtenir el valor real del tag tot i que
          estigui fora del rang
          CTSubs_Log("CTSubs_ValorTag - Error exec.
            SubscriptionGetAttribute = " +
            IntToStr(iErrorVariable) + " id = " +
            IntToStr(id) + " - Tag: " + sTag + " - Valor = "
            + sValor);

          //CASE 453 // Subscription value is pending -
          // Occurs when a tag has been subscribed to,
          // and attempted to read the value before the
          // initial value has been retrieved from the
          // server.
```

```

CASE ELSE
    CTSubs_Log("CTSubs_ValorTag - Error exec.
        SubscriptionGetAttribute = " +
        IntToStr(iErrorVariable) + " id = " +
        IntToStr(id) + " - Tag: " + sTag + " - Valor = "
        + sValor);
    Email_RunEnviarEmail("jjimenez@ccait.cat",
        "CTSubs_ValorTag - " + ServerInfo("Client", 0),
        "Error execució SubscriptionGetAttribute = " +
        IntToStr(iErrorVariable) + " id = " +
        IntToStr(id) + " - Tag: " + sTag + " - Valor = "
        + sValor);
END SELECT

RETURN RealToStr(CTSubs_grErrorValue, 5, 0);
ELSE
    RETURN sValor;
END

ELSE
    // Chequejem que el tag existeix a la base de dades, si no
    // existeix enviem missatge i continuem
    IF (_RDB_BuscarVariableTag(sTag)) = 0 THEN
        ErrLog("@(CITECT TagSubscribe)" + " - " + "@(Tag inexistent:
            )" + sTag);
        Email_RunEnviarEmail("jjimenez@ccait.cat",
            "CTSubs_SubscribirTag - " + ServerInfo("Client", 0), "Tag
            inexistent: " + sTag);
    ELSE
        CTSubs_SubscribirTag(sTag);
        IF (iMensaje = 1) THEN
            Message("CTSubs_ValorTag - NO Subscrito", "Se realiza el
                TagRead", 64);
        END
        RETURN TagRead(sTag); // Si és la primera vegada que es
            // subscriu, retornem el valor de
            // tag
    END
END

RETURN -1.0;
END

```

### 7.1.8 HIST\_ExecutarSentenciaSQL

Executa la sentència que es rep com a paràmetre.

```
/*
** FUNCIO:      HIST_ExecutarSentenciaSQL
**
** DESCRIPCIO: Executa la sentència rebuda com a paràmetre
*/
INT
FUNCTION HIST_ExecutarSentenciaSQL (STRING sSentencia)
  INT Servidor;
  INT iStatusSQLExec; // Control d' errors de la sentència select
  INT iErrorSQL;      // Control d'errors del SQL

  INT iAux, iAux1, iAux2, iAux3, iAux4;

  ErrSet(1);

  // Executem per resetejar si te algun error anterior emmagatzemat la
  // funció IsError
  iErrorSQL = IsError();

  // Enregistrem la sentència SQL a [DATA]:InforLog.txt
  iAux1 = TractaError(0, sSentencia);
  // Enregistrem la sentència SQL a [DATA]:LogSQL.txt
  iAux2 = LogSQL(sSentencia);

  // Executem l'stored procedure per emmagatzemar les lectures a l'sql.
  iAux = SQLSet(mhSQL, sSentencia);
  iStatusSQLExec = SQLExec(mhSQL, "");
  iErrorSQL = IsError();
  iAux3 = SQLEnd(mhSQL);

  // IsError(); Gets the current error value. The error value is set
  //when any error is detected, and is reset after this function is
  //called
  // 307 - SQL database error -> a general SQL error. Call the
  // SQLErrMsg() FUNCTION FOR details of the detected error.
  IF (iErrorSQL <> 0) THEN
    IF (iErrorSQL = 307) THEN
      TractaError(iErrorSQL, SQLErrMsg());
    END
    TractaError(iErrorSQL, "HIST Error inserció dades SQL -
      mhSQL=" + IntToStr(mhSQL) + " -> via
      HIST_ExecutarSentenciaSQL -> SQLExec = " +
      IntToStr(iStatusSQLExec) + "->" + IntToStr(iAux)
      + "->" + IntToStr(iAux1) + "->" + IntToStr(iAux2)
      + "->" + IntToStr(iAux3));
    RETURN (iErrorSQL);
  END

RETURN 0;
END
```

## 7.1.9 Funcions auxiliars

### 7.1.9.1 PS\_GetActive\_RptSvr

Retorna si qui l'executa és servidor primari de reports.

```
/*
** FUNCIO:      PS_GetActive_RptSvr()
**
** DESCRIPCIO: Retorna 1 si l'equip que crida la funcio es el servidor
** primari de reports (el que esta actiu)
*/
INT
FUNCTION PS_GetActive_RptSvr(INT iMsg = 0)
    STRING sStatus, sNombreMaquina;

    sNombreMaquina = ServerInfo("Client", 0);
    IF (sNombreMaquina = giSrv1Citect) OR (sNombreMaquina = giSrv2Citect)
        THEN // Servidores de CITECT

            sStatus = ServerInfoEx("Report", 0, "Report");

            IF iMsg THEN
                Message("PS_GetActive_RptSvr", sStatus, 64);
            END

            RETURN StrToInt(sStatus);
        ELSE
            IF iMsg THEN
                Message("PS_GetActive_RptSvr", "0", 64);
            END

            RETURN 0;
        END
    END
END
```

### 7.1.9.2 PS\_Prog\_Log

Funció de suport per al registre de dades en un txt per seguiment de l'aplicatiu i anàlisi de fallades.

```
/*
** FUNCIO:      PS_Prog_Log
**
** DESCRIPCIÓ: Registre de funcionament del programador de tasques.
**
**             Les tasques que s'executen es registren al Log_Operador
**             (SQL)
**
**             Es registren les tasques que no s'executen per algun fallo:
** · iError = 1   Fallada de coms amb la remota en un tag de condició
** · iError = 2   Estació en mant. o f.s. en un tag de condició
** · iError = 3   Fallada coms amb el IODevice en un tag de condició
**
** · iError = 9   Fallada de coms amb la remota en un tag d'acció
** · iError = 10  Estació en mant. o fora de servei en un tag d'acció
** · iError = 11  Fallada coms amb el CITECT en un tag d'acció
**
**             Es registra l'inici i el fi de la execució de la tasca del
**             programador:
** · iError = 100 Inici _PS_Programador() Núm. de tags subscrits
** · iError = 101 Fi _PS_Programador()   Temps d'execució de la tasca
**
**             Es registra el reset de les variables que emmagatzemen la
**             suscripció de tags:
** · iError = 102 Reset variables suscripció de tags
**
**             Es registra la espera de HIST_SQL:
** · iError = 103
**
**             Es registra la info dels valors de cada tasca que es executada
** · iError = 110 Tasca condicional
** · iError = 111 Tasca programació horaria
*/
FUNCTION PS_Prog_Log (STRING sId, STRING sIdTasca, STRING sDescTasca,
STRING sCond, STRING sAccio, INT iError)
    INT    iHandle;    // Punter al fitxer d' errors.
    INT    iTamany;    // Tamany del fitxer d' errors.
    INT    iResposta;  // Variable pel control d' errors.
    STRING sLinia;     // Linia a inserir al fitxer d' errors.
    STRING sErrorInfo; // Descripció de l'error
    INT    iDia, iMes, iAnyo;

    ErrSet(1);

    iDia = DateDay(TimeCurrent());
    iMes = DateMonth(TimeCurrent());
    iAnyo = DateYear(TimeCurrent());

    iHandle = FileOpen("[Data]:\PS_ProgLog\PS_ProgLog_" + iAnyo:#0 + "_"
        + iMes:#0 + "_" + iDia:#0 + ".txt", "a");
    IF (iHandle = -1) THEN
        RETURN;
    END
```

```

SELECT CASE iError
  CASE 1, 9   sErrorInfo = "F.estat remota";
              sDescTasca = StrLeft(sDescTasca, 45) + "...";
  CASE 2, 10  sErrorInfo = "F.est.mant./f.s.";
              sDescTasca = StrLeft(sDescTasca, 45) + "...";
  CASE 3, 11  sErrorInfo = "F.com.Citect";
              sDescTasca = StrLeft(sDescTasca, 45) + "...";

  CASE 100   sErrorInfo = "INI TASCA PROG.";
  CASE 101   sErrorInfo = "FIN TASCA PROG.";
  CASE 102   sErrorInfo = "RESET VARIABLES TagSubscribe";
  CASE 103   sErrorInfo = "REG. SQL";

  CASE 110   sErrorInfo = "EXEC.TASCA COND.";
              sDescTasca = StrLeft(sDescTasca, 45) + "...";
  CASE 111   sErrorInfo = "EXEC.TASCA HHmm";
              sDescTasca = StrLeft(sDescTasca, 45) + "...";
END SELECT

// Col.loquem el punter de fitxer al final.
iTamany = FileSize(iHandle);
IF (iTamany = 0) THEN
  sLinia = StrPad("DATA", " ", 8) + "^t" +
           StrPad("HORA", " ", 8) + "^t" +
           StrPad("ERROR", " ", 16) + "^t" +
           StrPad("ID", " ", 4) + "^t" +
           StrPad("IDP", " ", 4) + "^t" +
           StrPad("DESC TASCA", " ", 48) + "^t" +
           StrPad("CONDICIO", " ", 80) + "^t" +
           StrPad("ACCIO", " ", 80);

  iResposta = FileWriteLn(iHandle, sLinia);
  IF (iResposta < (StrLength(sLinia) + 2)) THEN
    RETURN; // S' ha produït un error d' escriptura
  END
END

// Afegim una nova linia.
sLinia = StrPad(Date(), " ", 8) + "^t" +
         StrPad(Time(1), " ", 8) + "^t" +
         StrPad(sErrorInfo, " ", 16) + "^t" +
         StrPad(sId, " ", 4) + "^t" +
         StrPad(sIdTasca, " ", 4) + "^t" +
         StrPad(sDescTasca, " ", 48) + "^t" +
         StrPad(sCond, " ", 80) + "^t" +
         StrPad(sAccio, " ", 80);

iResposta = FileWriteLn(iHandle, sLinia);

FileClose(iHandle);
END

```



### 7.1.9.3 Email\_RunEnviarEmail

Funció de enviament d'e-mails.

```
/*
** FUNCIO:          Email_RunEnviarEmail
**
** DESCRIPCIO:
*/
INT
FUNCTION Email_RunEnviarEmail (STRING sMailDestino, STRING sAsunto, STRING
                               sMensaje, INT iLog = 0)
    INT iError = 0;
    STRING sPath = PathToStr(sPathAssiMail);

    ErrSet(1);

    IF StrLength(sAsunto) > 112 THEN
        sAsunto = StrLeft(sAsunto, 112) + "...";
    END
    IF StrLength(sMensaje) > 124 THEN
        sMensaje = StrLeft(sMensaje, 124) + "...";
    END

    Email_LogTxt(sMailDestino, sMensaje, sAsunto);

    iError = Exec("cmd /c " + sPath + " " + sMailOrigen + ", " +
                 sMailDestino + ", " + sAsunto + ", " + sMensaje + ",
                 " + sSMTP, 6);

    iLog = 1;
    IF iLog THEN
        Email_LogTxt(sMailDestino, sMensaje, sAsunto);
    END

    RETURN iError;
END
```

## 7.2 Codi de la implementació l'històric al SQL Server

### 7.2.1 Codi d'inserció i tractament de dades

#### 7.2.1.1 *sp\_Insert\_Sentencia*

Enregistra la sentència SQL rebuda al servidor SQL local dels servidors CitectSCADA (SCADA01\SQLEXPRESS o SCADA02\SQLEXPRESS)

```
USE [Telec]
GO
/***** Object: StoredProcedure [dbo].[sp_Insert_Sentencia] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Jordi Jiménez
-- Description:  Registre de sentencies del Citect
-- =====
ALTER PROCEDURE [dbo].[sp_Insert_Sentencia]
    @DataReg datetime,
    @idOp integer,
    @Sentencia varchar(254),
    @flag integer
AS
BEGIN

    INSERT INTO sentencies (Data, idOp, sentencia, trasp) VALUES (@DataReg,
    @idOp, @Sentencia, @flag)

END
```

### 7.2.1.2 *sp\_sync\_Ct\_Sentencies*

Mou les sentencies de les bases de dades locals dels servidors CitectSCADA a les taules del servidor principal.

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_sync_Ct_sentencies] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Jordi Jiménez Orrios
-- Description:     Agafa un màxim de 5000 sentencies no traspasades (trasp = 0) -
--                 de la taula sentencies de la BD local de tots dos servidors -
--                 scada i les copia en la taula Ct_sentencies al INET
-- =====
ALTER PROCEDURE [dbo].[sp_sync_Ct_sentencies]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    SET NOCOUNT ON;

    --*****
    --***** SENTENCIES scada01 *****
    --*****

    -- BEGIN TRAN

    -- MARCAMOS UN CONJUNTO QUE QUEREMOS TRASPASAR PONIENDO ready (trasp=2)
    UPDATE TOP(5000) [SCADA01\SQLEXPRESS].Telec.dbo.sentencies
    SET trasp=2 WHERE trasp=0
    IF @@ERROR<>0
        BEGIN
            --ROLLBACK TRAN
            RAISERROR ('Error marcant sentencies del scada01.', 16, 1)
            RETURN
        END

    -- HACEMOS EL INSERT EN INET DE LOS REGISTROS ready (trasp=2)
    INSERT INTO exp.dbo.Ct_sentencies (data, idOp, sentencia)
    SELECT data, idOp, sentencia
    FROM [SCADA01\SQLEXPRESS].Telec.dbo.sentencies SV1
    WHERE SV1.trasp=2
    IF @@ERROR<>0
        BEGIN
            --ROLLBACK TRAN
            RAISERROR('Error copiant sentencies del scada01 al inet.',16,
                1)
            RETURN
        END

    -- MARCAMOS EL CONJUNTO TRASPASADO PONIENDO done (trasp = 4)
    UPDATE [SCADA01\SQLEXPRESS].Telec.dbo.sentencies SET trasp=4 WHERE trasp=2
    IF @@ERROR<>0
        BEGIN
            --ROLLBACK TRAN
            RAISERROR ('Error marcant sentencies traspasades del
                scada01.', 16, 1)
            RETURN
        END
    END
END
```

```

-- COMMIT TRAN

--*****
--***** SENTENCIAS scada02 *****
--*****

-- BEGIN TRAN

-- MARCAMOS UN CONJUNTO QUE QUEREMOS TRASPASAR PONIENDO ready (trasp=2)
UPDATE TOP(5000) [SCADA02\SQLEXPRESS].Telec.dbo.sentencias
SET trasp=2 WHERE trasp=0
IF @@ERROR<>0
    BEGIN
        --ROLLBACK TRAN
        RAISERROR ('Error marcant sentencias del scada02.', 16, 1)
        RETURN
    END

-- HACEMOS EL INSERT EN INET DE LOS REGISTROS ready (trasp=2)
INSERT INTO exp.dbo.Ct_sentencias (data, idOp, sentencia)
SELECT data, idOp, sentencia
FROM [SCADA02\SQLEXPRESS].Telec.dbo.sentencias SV1
WHERE SV1.trasp=2
IF @@ERROR<>0
    BEGIN
        --ROLLBACK TRAN
        RAISERROR ('Error copiant sentencias del scada02 al inet.', 16,
1)
        RETURN
    END

-- MARCAMOS EL CONJUNTO TRASPASADO PONIENDO done (trasp = 4)
UPDATE [SCADA02\SQLEXPRESS].Telec.dbo.sentencias SET trasp=4 WHERE trasp=2
IF @@ERROR<>0
    BEGIN
        --ROLLBACK TRAN
        RAISERROR ('Error marcant sentencias traspasades del
scada02.', 16, 1)
        RETURN
    END

-- COMMIT TRAN

END

```

### 7.2.1.3 sp\_exec\_Ct\_Sentencias

Executa fins un màxim de 5000 sentencies emmagatzemades a Ct\_Sentencias

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Exec_Ct_Sentencias] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jord Jiménez Orrios CAT
-- Description:     Executa un grup de sentencies SQL, màxim de 50000, de la taula
--                  Ct_Sentencias
-- =====
ALTER PROCEDURE [dbo].[sp_Exec_Ct_Sentencias]
    -- Add the parameters for the stored procedure here

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    declare @rows int
    declare @sBD as nvarchar(32)
    declare @sTransaccion as nvarchar(2048)

    -- Ejecución en servidor (local)
    --PRINT '***** EXECALLTRANS[Ejecucion LOCAL]'
    update TOP(50000) exp.dbo.Ct_sentencias set trasp = 2 where trasp is null
    set @rows = @@ROWCOUNT
    set @rows = 1
    IF @rows > 0 BEGIN
        -- *****
        -- Executar todas las trasp = 2
        declare lTrans cursor for
        select sentencia, 'exp' from Ct_sentencias
                                where trasp = 2
                                order by data asc

        open lTrans
        bucl:
        fetch next from lTrans into @sTransaccion, @sBD
        if @@fetch_status = 0
        begin
            exec exp.dbo.sp_Exec_Ct_sentencia @sTransaccion, @sBD
            goto bucl
        end
        close ltrans
        deallocate ltrans
        -- *****
        update exp.dbo.Ct_sentencias set trasp = 4 where trasp = 2
        -- Borrar transacciones terminadas
        if 1=0
            delete from exp.dbo.Ct_sentencias where trasp = 4
    END
END
```

### 7.2.1.4 sp\_exec\_Ct\_Sentencia

Execució d'una sentència SQL.

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Exec_Ct_Sentencia] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Jord Jiménez Orrios CAT
-- Description:     Executa una sentència SQL que es rep per paràmetre
--                  @sTransaccion nvarchar(2000),      Sentència SQL
--                  @sBD nvarchar(32)                  Base de dades
-- =====
ALTER PROCEDURE [dbo].[sp_Exec_Ct_Sentencia]
    @sTransaccion nvarchar(2000),
    @sBD nvarchar(32)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    declare @ssql as nvarchar(2048)
    declare @err as int

    -- SOLUCIÓ PEL PROBLEMA DE LA COMETA
    set @sTransaccion = replace(@sTransaccion, ',', ' ')
    set @sTransaccion = replace(@sTransaccion, ''Ampolla'', ''Ampolla'')
    set @sTransaccion = replace(@sTransaccion, ''Impulsi'', ''Impulsi'')
    set @sTransaccion = replace(@sTransaccion, ''Arrib'', ''Arribada'')
    set @sTransaccion = replace(@sTransaccion, ''equip'', ''equip'')
    set @sTransaccion = replace(@sTransaccion, ''Aldea'', ''Aldea'')
    set @sTransaccion = replace(@sTransaccion, ''Ametlla'', ''Ametlla'')
    set @sTransaccion = replace(@sTransaccion, ''Almadrava'', ''Almadrava'')
    set @sTransaccion = replace(@sTransaccion, ''aire'', ''aire'')

    set @ssql = 'use ' + @sBD + ' ' + 'exec sp_executesql N''' + @sTransaccion +
    '

    begin tran ExecTrans
        exec @err = sp_executesql @ssql
        if @err <> 0
            begin
                insert into Ct_sentencies_Log(data,sentencia,BD,error)
                values(getdate(), @sTransaccion, @sBD, ERROR_MESSAGE())

                rollback tran ExecTrans
            end
        else
            commit tran ExecTrans
    END
```

### 7.2.1.5 HIST\_sp\_AfegirDada

Afegeix la dada rebuda a les taules del historitzador.

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[HIST_sp_AfegirDada *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jordi Jimenez
-- Description:
--     es rep com parametre el nom de tag, el tipus, la lectura i la data
--     si el tag no existeix a la taula HIST_tags, se li afegeix.
--     segons el valor del parametre @tipo afegeix la informacio a una taula o
--     una altra
-- =====
ALTER PROCEDURE [dbo].[HIST_sp_AfegirDada]
    @sNAME varchar(80),
    @sENG_ZERO varchar(12),
    @sENG_FULL varchar(12),
    @sENG_UNITS varchar(12),
    @sCOMMENT varchar(64),
    @tipo integer,      -- segons el tipus s'afegirà en una taula o altra (veure
                        -- CFG_TipusTags):
    -- 0 Valor de cabal                REAL HIST_DadesR
    -- 1 Valor de pressió              REAL HIST_DadesR
    -- 2 Valor de temperatura          REAL HIST_DadesR
    -- 3 Valor de nivell               REAL HIST_DadesR
    -- 4 Valor de posició              REAL HIST_DadesR
    -- 5 Valor de pes                  REAL HIST_DadesR
    -- 6 Valor de Humitat              REAL HIST_DadesR
    -- 7 Valor de velocitat            REAL HIST_DadesR
    -- 50 Totalitzador de cabal        LONG HIST_DadesI
    -- 51 Totalitz. virt. de cabal     LONG HIST_DadesI
    -- 52 Temps de funcionament        LONG HIST_DadesI
    -- 53 Número d'actuacions          LONG HIST_DadesI
    -- 100 Iniciador de oberta         BOOL HIST_DadesD
    -- 101 Iniciador de tancada        BOOL HIST_DadesD
    -- 102 Condiició de marxa         BOOL HIST_DadesD

    @origen integer,
    @sValor real,
    @data datetime

AS
BEGIN

    DECLARE @Reg integer      -- Si TRUE comprova abans a CFG_Tags si el tag és
                            -- existent, si no hi és, el crea.
                            -- Ens permet estalviar-nos la búsqueda en CFG_Tags
                            -- cada vegada que registrem si @Reg = 0
    DECLARE @idTagNew integer -- Obtenció de nou @idTag
    DECLARE @IDReg integer
    DECLARE @Per10 integer

    SET @Reg=1

    IF (@Reg=1) BEGIN
```

```

SET @IDReg=(SELECT idTag FROM CFG_Tags WHERE tag = @sNAME);

-- si no te un valor, afegim el registre a CFG_Tags i li posem el
-- valor a @IDReg
IF (@IDReg IS NULL) BEGIN
    SET @idTagNew = (select exp.dbo.HIST_Get_idTag_CFG_Tags())
    INSERT INTO CFG_Tags (idTag, tag, eng_zero, eng_full, eng_unit,
        descTag, idTagTip, dataAlta)
    VALUES (@idTagNew, @sNAME, @sENG_ZERO, @sENG_FULL, @sENG_UNITS,
        @sCOMMENT, @tipo, GETDATE())
    SET @IDReg = @idTagNew
END
UPDATE CFG_Tags SET UltAcces = GetDate() WHERE idTag = @IDReg
END

-- Si es un totalitzat @tipo=50 hem de comprovar si el totalitzat és per 1 o per 10
-- (camp per10 TRUE)
IF (@tipo = 50) BEGIN
    SET @Per10=(SELECT per10 FROM CFG_Tags WHERE idTag = @IDReg);
    IF @Per10 = 1 BEGIN
        SET @sValor = (@sValor * 10.0)
    END
END

-- Fem insert segons el valor de @tipo
IF (@tipo >= 0) AND (@tipo <= 49)
-- Valor REAL -> HIST_DadesR
    INSERT INTO HIST_DadesR (idTag, valor, data, idproced)
    VALUES (@IDReg, @sValor, @data, @origen);
ELSE
    BEGIN
        IF (@tipo >= 50) AND (@tipo <= 99)
        -- Valor INT -> HIST_DadesI
            INSERT INTO HIST_DadesI (idTag, valor, data, idproced)
            VALUES (@IDReg, @sValor, @data, @origen);
        ELSE
            BEGIN
                IF (@tipo >= 100) AND (@tipo <= 149)
                -- Valor DIGITAL -> HIST_DadesD
                    INSERT INTO HIST_DadesD (idTag, valor, data,
                        idproced)
                    VALUES (@IDReg, @sValor, @data, @origen);
                ELSE
                    BEGIN
                        IF (@tipo >= 150) AND (@tipo <= 199)
                        -- Valor Reserva -> HIST_DadesX
                            INSERT INTO HIST_DadesR (idTag, valor,
                                data, idproced)
                            VALUES (@IDReg, @sValor, @data, @origen);
                        ELSE
                            BEGIN
                                IF (@tipo >= 200) AND (@tipo <= 249)
                                -- Valor Reserva -> HIST_DadesX
                                    INSERT INTO HIST_DadesR (idTag,
                                        valor, data, idproced)
                                    VALUES (@IDReg, @sValor, @data,
                                        @origen);
                                ELSE

```



```

BEGIN
IF (@tipo >= 250) AND (@tipo <= 299)
-- Valor Reserva -> HIST_DadesX
INSERT INTO HIST_DadesR
(idTag, valor, data,
idproced)
VALUES (@IDReg, @sValor,
@data, @origen);
ELSE
-- Tipus incorrecte ho posem en REAL
BEGIN
INSERT INTO HIST_DadesR
(idTag, valor, data,
idproced)
VALUES (@IDReg,
@sValor, @data,
@origen);
END
END
END
END
END
END
END
END

```

## 7.2.2 Codi de manteniment de les bases de dades

Funcions per tal de mantenir el tamany de la base de dades controlat.

### 7.2.2.1 sp\_Mant\_Sentencies\_scada01\_02

Tasca de manteniment de taula sentencies i sentencies\_trasp de scada01/scada02.

Mou els registres de més de 60 dies de la taula [telec].sentencies de scada01 i scada02 a la taula local [telec].sentencies\_trasp

Elimina els registres de més de 180 dies de la taula [telec].sentencies\_trasp de scada01 i scada02

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Mant_Sentencies_scada01_02 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jord Jiménez Orrios CAT
--
-- Description:     Mou les sentencies ja executades de la taula:
--                  sentencies (trasp = 4) a la taula:
--                  sentencies_trasp dels servidors scada01 / scada02
-- =====
ALTER PROCEDURE [dbo].[sp_Mant_Sentencies_scada01_02]

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @DiesMove INTEGER
    DECLARE @DiesDelete INTEGER

    SET @DiesMove=60
    SET @DiesDelete=180

    -- scada01 - Sentencies de més de @DiesMove=60 son passades de taula -
    -- sentencies a taula sentencies_trasp
    INSERT INTO [SCADA01\SQLEXPRESS].telec.dbo.sentencies_trasp (ID, Data, idOp,
        sentencia, trasp)
    SELECT ID, Data, idOp, sentencia, trasp FROM
    [SCADA01\SQLEXPRESS].telec.dbo.sentencies
    WHERE (trasp = 4) AND (Data < (GetDate()-@DiesMove))

    IF @@ROWCOUNT > 0
    BEGIN
        DELETE FROM [SCADA01\SQLEXPRESS].telec.dbo.sentencies
        WHERE (trasp = 4) AND (Data < (GetDate()-@DiesMove))
    END

    -- scada02 - Sentencies de més de @DiesMove=60 son passades de taula -
    -- sentencies a taula sentencies_trasp
```

```

INSERT INTO [SCADA02\SQLEXPRESS].telec.dbo.sentencies_trasp (ID, Data, idOp,
sentencia, trasp)
SELECT ID, Data, idOp, sentencia, trasp FROM
[SCADA02\SQLEXPRESS].telec.dbo.sentencies
WHERE (trasp = 4) AND (Data < (GetDate()-@DiesMove))

IF @@ROWCOUNT > 0
BEGIN
    DELETE FROM [SCADA02\SQLEXPRESS].telec.dbo.sentencies
    WHERE (trasp = 4) AND (Data < (GetDate()-@DiesMove))
END

-- scada01 - Sentencies de més de @DiesDelete=180 son eliminades de taula
-- sentencies_trasp
DELETE FROM [SCADA01\SQLEXPRESS].telec.dbo.sentencies_trasp
WHERE (trasp = 4) AND (Data < (GetDate()-@DiesDelete))
-- scada02 - Sentencies de més de @DiesDelete=180 son eliminades de taula
-- sentencies_trasp
DELETE FROM [SCADA02\SQLEXPRESS].telec.dbo.sentencies_trasp
WHERE (trasp = 4) AND (Data < (GetDate()-@DiesDelete))

END

```

### 7.2.2.2 *sp\_mant\_Ct\_Sentencies*

Tasca de manteniment de taula Ct\_sentencies i Ct\_sentencies\_fin del servidor principal de SQL Server.

Mou els registres de més de 60 dies de la taula [exp].Ct\_sentencies de inet a la taula [exp].Ct\_sentencies\_fin

Elimina els registres de més de 180 dies de la taula [exp].Ct\_sentencies\_fin de inet

```

USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Mant_Ct_Sentencies] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jord Jiménez Orrios CAT
-- Description:     Mou les sentencies ja executades de la taula Ct_sentencies -
--                  (trasp = 4) a la taula Ct_sentencies_fin
-- =====
ALTER PROCEDURE [dbo].[sp_Mant_Ct_Sentencies]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @DiesMove INTEGER
    DECLARE @DiesDelete INTEGER

```

```

SET @DiesMove=60
SET @DiesDelete=180

-- inet - Sentencies de més de @DiesMove=60 son passades de taula
-- Ct_sentencies a taula Ct_sentencies_fin
INSERT INTO Ct_sentencies_fin (data, idOp, sentencia, error)
SELECT Data, idOp, sentencia, '0'
FROM Ct_sentencies WHERE (trasp = 4) AND (Data < (GetDate()-@DiesMove))

IF @@ROWCOUNT > 0
BEGIN
    DELETE FROM Ct_sentencies
    WHERE (trasp = 4) AND (Data < (GetDate()-@DiesMove))
END

-- inet - Sentencies de més de @DiesDelete=180 son eliminades de taula -
-- sentencies_fin
DELETE FROM Ct_sentencies_fin WHERE (Data < (GetDate()-@DiesDelete))

END

```

## 7.2.3 Codi de comprovació de la integritat de les taules del Historian

### 7.2.3.1 sp\_Hist\_CheckDiario\_D

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Hist_CheckDiario_D] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jord Jiménez Orrios CAT
-- Description:     Comprovació del registre historian de les senyals DIGITALS
-- =====
ALTER PROCEDURE [dbo].[sp_Hist_CheckDiario_D]
    @fecha datetime = NULL
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @TagsHIST_DadesD INTEGER
    DECLARE @RegistresD_per_1h INTEGER
    DECLARE @RegistresD_per_1d INTEGER

    DECLARE @NumeroRegistres_1h INTEGER
    DECLARE @NumeroRegistres_1d INTEGER

    DECLARE @aux INTEGER
    DECLARE @forats INTEGER
    DECLARE @FechaIni datetime
    DECLARE @FechaFin datetime

    DECLARE @consulta as varchar(300)
    DECLARE @assumpte as varchar(50)

    SET @TagsHIST_DadesD = 499 -- Es registren 499 DIGITAL tags cada 10 minuts
    SET @RegistresD_per_1h=@TagsHIST_DadesD*6
    SET @RegistresD_per_1d=@RegistresD_per_1h*24

    SET @aux = 0
    SET @forats = 0

    -- Chequejem valor NULL del paràm. d'entrada. @fecha pren la data d'ara
    IF (@fecha IS NULL)
    BEGIN
        SET @fecha = (DATEADD(DAY,-1,SYSDATETIME()))
        SET @fecha = DATETIMEFROMPARTS ( datepart("yy",@fecha),
            datepart("mm",@fecha), datepart("dd",@fecha), '00', '00',
            '00', '00')
    END

    -- Obtenim en número de registres del dia que rebem per paràmetre
    SET @NumeroRegistres_1d = (select TOP 1 count(*)
    from exp.dbo.HIST_DadesD
    where convert(varchar, data,3) = @fecha
```

```
group by convert(varchar, data,3)
order by convert(datetime, convert(varchar, data,3)) desc)
```

```
IF @NumeroRegistres_1d = @RegistresD_per_1d
    BEGIN
        PRINT CAST(@fecha AS NVARCHAR(100)) + ' -> GENERAL DIA ->
CORRECTO ' + CAST(@NumeroRegistres_1d AS NVARCHAR(100)) + '
registros DIGITAL en 1 dia'
    END
ELSE
    BEGIN
        PRINT CAST(@fecha AS NVARCHAR(100)) + ' -> GENERAL DIA -> ERROR
- Falta ' + CAST(@RegistresD_per_1d-@NumeroRegistres_1d AS
NVARCHAR(100)) + ' de ' + CAST(@RegistresD_per_1d AS
NVARCHAR(100)) + ' registros DIGITAL en 1 dia'
        SET @forats = 1

        -- Hem de comprovar a quina franja horaria tenim l'error ->
        -- Chequejem les 24h, una a una
        WHILE @aux < 24
            BEGIN

                SET @FechaIni = DATETIMEFROMPARTS (
                    datepart("yy",@fecha),
                    datepart("mm",@fecha),
                    datepart("dd",@fecha), CAST(@aux as
                    varchar(2)), '00', '00', '00')
                SET @FechaFin = DATETIMEFROMPARTS (
                    datepart("yy",@fecha),
                    datepart("mm",@fecha),
                    datepart("dd",@fecha), '23', '59', '59',
                    '00')

                SET @NumeroRegistres_1h = (select top 1 count(*) from
                    exp.dbo.HIST_DadesD
                    where (data >= @FechaIni) and
                    (data < @FechaFin)
                    group by datepart("hh",data)
                    order by datepart("hh",data))

                IF @NumeroRegistres_1h = @RegistresD_per_1h
                    BEGIN
                        PRINT CAST(@FechaIni AS NVARCHAR(100)) + '
-> CORRECTO ' + CAST(@NumeroRegistres_1h AS
NVARCHAR(100)) + ' registros DIGITAL en 1
hora'
                    END
                ELSE
                    BEGIN
                        PRINT CAST(@FechaIni AS NVARCHAR(100)) + '
-> ERROR - Falta ' +
CAST(@RegistresD_per_1h-@NumeroRegistres_1h
AS NVARCHAR(100)) + ' de ' +
CAST(@RegistresD_per_1h AS NVARCHAR(100)) +
' registros DIGITAL en 1 hora';
                    END

                SET @aux = @aux + 1
            END
    END;
END;
```

```
-- Hem d'enviar info dels registres que falten via email
```

```

IF @forats = 1
    BEGIN
        DELETE FROM exp.dbo.HIST_Temp
        -- Obtenció dels registres que falten en aquesta
        -- hora i envio d'email informant
        SET @FechaIni = DATETIMEFROMPARTS (
            datepart("yy",@fecha),
            datepart("mm",@fecha),
            datepart("dd",@fecha), '00', '00',
            '00' , '00')
        SET @FechaFin = DATETIMEFROMPARTS (
            datepart("yy",@fecha),
            datepart("mm",@fecha),
            datepart("dd",@fecha), '23', '59',
            '59' , '00');

        -- Obtenció dels registres que falten en aquesta
        -- hora i envio d'email informant
        WITH q(s, e, t) AS
            (
                SELECT MIN(data), MAX(data), idTag
                FROM exp.dbo.HIST_DadesD
                WHERE (data>=@FechaIni) and
                (data<@FechaFin)
                GROUP BY idTag
                UNION ALL
                SELECT DATEADD(minute, 10,
                s), e, t
                FROM q
                WHERE s < e
            )

        INSERT INTO exp.dbo.Hist_Temp (a_dt, b_dt, c_int,
            d_varchar80)
        SELECT s, e, t,
        (select tag from exp.dbo.CFG_Tags where q.t=idTag)
        FROM q
        WHERE NOT EXISTS
            (
                SELECT data, idTag
                FROM exp.dbo.HIST_DadesD
                WHERE data = s AND IdTag=t
            )

        ORDER BY t, s
        OPTION (MAXRECURSION 0)

        -- Envio del email informant
        set @assumpte = 'sp_Hist_CheckDiario_D'
        set @consulta = 'SELECT a_dt as data, c_int as
            idTag, d_varchar80 as tag
            FROM exp.dbo.HIST_Temp'

        EXEC msdb.dbo.sp_send_dbmail
        @profile_name = 'perfil INET',
        @recipients = 'jjimenez@ccaait.cat',
        @query = @consulta,
        @subject = @assumpte,
        @attach_query_result_as_file = 0

    END
END
END

```

### 7.2.3.2 *sp\_Hist\_CheckDiario\_I*

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Hist_CheckDiario_I] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jord Jiménez Orrios CAT
-- Description:     Comprovació del registre historian de les senyals INTEGER
-- =====
ALTER PROCEDURE [dbo].[sp_Hist_CheckDiario_I]
    @fecha datetime = NULL
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @TagsHIST_DadesI INTEGER
    DECLARE @RegistresI_per_1h INTEGER
    DECLARE @RegistresI_per_1d INTEGER

    DECLARE @NumeroRegistres_1h INTEGER
    DECLARE @NumeroRegistres_1d INTEGER

    DECLARE @aux INTEGER
    DECLARE @forats INTEGER
    DECLARE @FechaIni datetime
    DECLARE @FechaFin datetime

    DECLARE @consulta as varchar(300)
    DECLARE @assumpte as varchar(50)

    SET @RegistresI_per_1h=221 -- Es registren 221 INTEGER tags cada 1 hora
    SET @RegistresI_per_1d=@RegistresI_per_1h*24

    SET @aux = 0
    SET @forats = 0

    -- Chequejm el valor NULL del paràmetre d'entrada. En aquest cas @fecha pren
    -- la data d'ara
    IF (@fecha IS NULL)
    BEGIN
        SET @fecha = (DATEADD(DAY,-1,SYSDATETIME()))
        SET @fecha = DATETIMEFROMPARTS ( datepart("yy",@fecha),
            datepart("mm",@fecha), datepart("dd",@fecha), '00', '00',
            '00', '00')
    END

    -- Obtenim en número de registres del dia que rebem per paràmetre
    SET @NumeroRegistres_1d = (select TOP 1 count(*)
        from exp.dbo.HIST_DadesI
        where convert(varchar, data,3) = @fecha
        group by convert(varchar, data,3)
        order by convert(datetime, convert(varchar,
            data,3)) desc)
```



```

IF @NumeroRegistres_1d = @RegistresI_per_1d
    BEGIN
        PRINT CAST(@fecha AS NVARCHAR(100)) + ' -> GENERAL DIA ->
CORRECTO ' + CAST(@NumeroRegistres_1d AS NVARCHAR(100)) +
' registros INTEGER en 1 dia'
    END
ELSE
    BEGIN
        PRINT CAST(@fecha AS NVARCHAR(100)) + ' -> GENERAL DIA -> ERROR
- Faltan ' + CAST(@RegistresI_per_1d-@NumeroRegistres_1d
AS NVARCHAR(100)) + ' de ' + CAST(@RegistresI_per_1d AS
NVARCHAR(100)) + ' registros INTEGER en 1 dia'
        SET @forats = 1

        -- Hem de comprovar a quina franja horaria tenim l'error ->
        -- Chequejem les 24h, una a una
        WHILE @aux < 24
            BEGIN

                SET @FechaIni = DATETIMEFROMPARTS (
                    datepart("yy",@fecha), datepart("mm",@fecha),
                    datepart("dd",@fecha), CAST(@aux as varchar(2)),
                    '00', '00', '00')
                SET @FechaFin = DATETIMEFROMPARTS (
                    datepart("yy",@fecha), datepart("mm",@fecha),
                    datepart("dd",@fecha), '23', '59', '59', '00')

                SET @NumeroRegistres_1h = (select top 1 count(*)
                    from exp.dbo.HIST_DadesI
                    where (data >= @FechaIni) and (data
                    < @FechaFin)
                    group by datepart("hh",data) order
                    by datepart("hh",data))

                IF @NumeroRegistres_1h = @RegistresI_per_1h
                    BEGIN
                        PRINT CAST(@FechaIni AS NVARCHAR(100)) + '
-> CORRECTO ' + CAST(@NumeroRegistres_1h AS
NVARCHAR(100)) + ' registros INTEGER en 1
hora'
                    END
                ELSE
                    BEGIN
                        PRINT CAST(@FechaIni AS NVARCHAR(100)) + '
-> ERROR - Faltan ' +
CAST(@RegistresI_per_1h-@NumeroRegistres_1h
AS NVARCHAR(100)) + ' de ' +
CAST(@RegistresI_per_1h AS NVARCHAR(100)) +
' registros INTEGER en 1 hora';
                    END

                SET @aux = @aux + 1
            END;

        -- Hem d'enviar info dels registres que falten via email
        IF @forats = 1
            BEGIN
                DELETE FROM exp.dbo.HIST_Temp
                -- Obtenció dels registres que falten en aquesta
                -- hora i envio d'email informant
            END
    END

```

```

SET @FechaIni = DATETIMEFROMPARTS (
datepart("yy",@fecha), datepart("mm",@fecha),
datepart("dd",@fecha), '00', '00', '00', '00')
SET @FechaFin = DATETIMEFROMPARTS (
datepart("yy",@fecha), datepart("mm",@fecha),
datepart("dd",@fecha), '23', '59', '59', '00');

-- Obtenció dels registres que falten en aquesta -
- hora i envio d'email informant
WITH q(s, e, t) AS
(
SELECT MIN(data), MAX(data),
idTag
FROM exp.dbo.HIST_DadesI
WHERE (data>=@FechaIni) and
(data<@FechaFin)
GROUP BY idTag
UNION ALL
SELECT DATEADD(hour, 1, s),
e, t
FROM q
WHERE s < e
)

INSERT INTO exp.dbo.Hist_Temp (a_dt, b_dt, c_int,
d_varchar80)
SELECT s, e, t,
(select tag from exp.dbo.CFG_Tags where q.t=idTag)
FROM q
WHERE NOT EXISTS
(
SELECT data, idTag
FROM exp.dbo.HIST_DadesI
WHERE data = s AND IdTag=t
)

ORDER BY t, s

OPTION (MAXRECURSION 0)

-- Envio del email informant
set @assumpte = 'sp_Hist_CheckDiario_I'
set @consulta = 'SELECT a_dt as data, c_int as
idTag, d_varchar80 as tag FROM
exp.dbo.HIST_Temp'

EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'perfil INET',
@recipients = 'jjimenez@ccaait.cat',
@query = @consulta,
@subject = @assumpte,
@attach_query_result_as_file = 0

```

END

END

END

### 7.2.3.3 *sp\_Hist\_CheckDiario\_R*

Comprovació diària

```
USE [exp]
GO
/***** Object: StoredProcedure [dbo].[sp_Hist_CheckDiario_R] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Jord Jiménez Orrios CAT
-- Description:     Comprovació del registre historian de les senyals REAL
-- =====
ALTER PROCEDURE [dbo].[sp_Hist_CheckDiario_R]
    @fecha datetime = NULL
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @TagsHIST_DadesR INTEGER
    DECLARE @RegistresR_per_1h INTEGER
    DECLARE @RegistresR_per_1d INTEGER

    DECLARE @NumeroRegistres_1h INTEGER
    DECLARE @NumeroRegistres_1d INTEGER

    DECLARE @aux INTEGER
    DECLARE @forats INTEGER
    DECLARE @FechaIni datetime
    DECLARE @FechaFin datetime

    DECLARE @consulta as varchar(300)
    DECLARE @assumpte as varchar(50)

    -- Es registren 622 REAL tags cada 10 minuts (612 + 10 sofrel)
    SET @TagsHIST_DadesR = 622
    SET @RegistresR_per_1h=@TagsHIST_DadesR*6
    SET @RegistresR_per_1d=@RegistresR_per_1h*24
    SET @aux = 0
    SET @forats = 0

    -- Check valor NULL del paràm. d'entrada. @fecha prendrà la data d'ahir
    IF (@fecha IS NULL)
    BEGIN
        SET @fecha = (DATEADD(DAY,-1,SYSDATETIME()))
        SET @fecha = DATETIMEFROMPARTS ( datepart("yy",@fecha),
            datepart("mm",@fecha), datepart("dd",@fecha), '00', '00',
            '00', '00')
    END

    -- Obtenim en número de registres del dia que rebem per paràmetre
    SET @NumeroRegistres_1d = (select TOP 1 count(*)
    from exp.dbo.HIST_DadesR
    where convert(varchar, data,3) = @fecha
    group by convert(varchar, data,3)
    order by convert(datetime, convert(varchar, data,3)) desc)
```

```

IF @NumeroRegistres_1d = @RegistresR_per_1d
    BEGIN
        PRINT CAST(@fecha AS NVARCHAR(100)) + ' -> GENERAL DIA ->
CORRECTO ' + CAST(@NumeroRegistres_1d AS NVARCHAR(100)) + '
registros REAL en 1 dia'
    END
ELSE
    BEGIN
        PRINT CAST(@fecha AS NVARCHAR(100)) + ' -> GENERAL DIA -> ERROR
- Faltan ' + CAST(@RegistresR_per_1d-@NumeroRegistres_1d AS
NVARCHAR(100)) + ' de ' + CAST(@RegistresR_per_1d AS
NVARCHAR(100)) + ' registros REAL en 1 dia'
        SET @forats = 1

        -- Hem de comprovar a quina franja horaria tenim l'error ->
        -- Chequejem les 24h, una a una
        WHILE @aux < 24
            BEGIN

                SET @FechaIni = DATETIMEFROMPARTS (
                    datepart("yy",@fecha), datepart("mm",@fecha),
                    datepart("dd",@fecha), CAST(@aux as varchar(2)),
                    '00', '00', '00')
                SET @FechaFin = DATETIMEFROMPARTS (
                    datepart("yy",@fecha), datepart("mm",@fecha),
                    datepart("dd",@fecha), '23', '59', '59', '00')

                SET @NumeroRegistres_1h = (select top 1 count(*) from
                    exp.dbo.HIST_DadesR
                    where (data >= @FechaIni) and
                    (data < @FechaFin)
                    group by datepart("hh",data)
                    order by datepart("hh",data))

                IF @NumeroRegistres_1h = @RegistresR_per_1h
                    BEGIN
                        PRINT CAST(@FechaIni AS NVARCHAR(100)) + '
-> CORRECTO ' + CAST(@NumeroRegistres_1h AS
NVARCHAR(100)) + ' registros REAL en 1
hora'
                    END
                ELSE
                    BEGIN
                        PRINT CAST(@FechaIni AS NVARCHAR(100)) + '
-> ERROR - Faltan ' +
CAST(@RegistresR_per_1h-@NumeroRegistres_1h
AS NVARCHAR(100)) + ' de ' +
CAST(@RegistresR_per_1h AS NVARCHAR(100)) +
' registros REAL en 1 hora';
                    END

                SET @aux = @aux + 1
            END;

        -- Hem d'enviar info dels registres que falten via email
        IF @forats = 1
            BEGIN
                DELETE FROM exp.dbo.HIST_Temp
                -- Obtenció dels registres que falten en aquesta
                -- hora i envio d'email informant
            END
    END

```

```

SET @FechaIni = DATETIMEFROMPARTS (
datepart("yy",@fecha), datepart("mm",@fecha),
datepart("dd",@fecha), '00', '00', '00', '00')
SET @FechaFin = DATETIMEFROMPARTS (
datepart("yy",@fecha), datepart("mm",@fecha),
datepart("dd",@fecha), '23', '59', '59', '00');

-- Obtenció dels registres que falten en aquesta
--hora i envio d'email informant
WITH q(s, e, t) AS
(
SELECT MIN(data), MAX(data),
idTag
FROM exp.dbo.HIST_DadesR
WHERE (data>=@FechaIni) and
(data<@FechaFin)
GROUP BY idTag
UNION ALL
SELECT DATEADD(minute, 10,
s), e, t
FROM q
WHERE s < e
)

INSERT INTO exp.dbo.Hist_Temp (a_dt, b_dt, c_int,
d_varchar80)
SELECT s, e, t,
(select tag from exp.dbo.CFG_Tags where q.t=idTag)
FROM q

WHERE NOT EXISTS
(
SELECT data, idTag
FROM exp.dbo.HIST_DadesR
WHERE data = s AND IdTag=t
)

ORDER BY t, s

OPTION (MAXRECURSION 0)

-- Envio del email informant
set @assumpte = 'sp_Hist_CheckDiario_R'
set @consulta = 'SELECT a_dt as data, c_int as
idTag, d_varchar80 as tag FROM exp.dbo.HIST_Temp'

EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'perfil INET',
@recipients = 'jjimenez@ccaait.cat',
@query = @consulta,
@subject = @assumpte,
@attach_query_result_as_file = 0

```

END

END

END

## 8 Webgrafia

<https://reference.opcfoundation.org/>

<https://www.swan-forum.com/digital-twin-h2o-work-group/>

<https://softwaresupport.aveva.com/>

<https://social.msdn.microsoft.com/Forums/es-ES/home?forum=sqlserveres>