UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

ENGINYERIA EN TECNOLOGIES INDUSTRIALS

TREBALL FÍ DE GRAU

Font Granja, Marc

ESTUDI DE LES ETAPES D'IMPLEMENTACIÓ D'UN PROJECTE D'AUTOMATITZACIÓ I ROBÒTICA.

Director del TFG: Delgado Prieto, Miguel

Co-director del TFG: Fernández Sobrino, Ángel

Convocatòria: Juliol, 2020

I declare that,

the work in this Master Thesis / Degree Thesis (*choose one*) is completely my own work,

no part of this Master Thesis / Degree Thesis (*choose one*) is taken from other people's work without giving them credit,

all references have been clearly cited,

I'm authorised to make use of the company's / research group (*choose one*) related information I'm providing in this document (*select when it applies*).

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by *The Universitat Politècnica de Catalunya - BarcelonaTECH*.

Marc Font Granja		18/06/2020
Student Name	Signature	Date

Title of the Thesis : Study of the implementation stages of an automation and robotics project. "Estudi de les etapes d'implementació d'un projecte d'automatització i robótica."

ÍNDEX

ÍNDEX DE FIGURES

ÍNDEX DE TAULES

ÍNDEX DE FÓRMULES

ÍNDEX DE DIAGRAMES

CAPÍTOL 1. INTRODUCCIÓ

1.1. Organització de la memòria

La memòria ha estat desenvolupada de forma que el lector pugui entendre i seguir amb facilitat tots els passos realitzats. Primer és farà referència als aspectes clau per entendre els conceptes que més tard posarem en pràctica.

En aquest primer capítol d'introducció, parlarem sobre la motivació que ha fet possible el projecte i el perquè ens ha portat a fer-ho. Definirem els objectius proposats des del inici i és farà una descripció general del treball per etapes.

En el capítol 2 és contextualitza l'estat de l'art del qual parlem, és posa en coneixement les tecnologies involucrades directa o indirectament en el projecte, que són necessàries per la seva interpretació i realització. Definirem protocols de comunicació, ens endinsarem en l'entorn de les xarxes i com transmeten la informació en ella i també entendrem els tipus de llenguatges de programació per PLCs.

En els següents capítols 3 i 4 és descriu l'arquitectura del projecte desenvolupat, des de les pròpies dades test emulades, una guia docent de tots els softwares utilitzats per els fluxos i emmaagatzematges, fins el programa per a la visualització i anàlisis d'aquesta informació.

El capítol 5 és validant els processos que s'han dut a terme per entendre el funcionament dels sistemes i el treball final, però en el cas de no obtenir els resultats idonis s'esmenta com s'han solucionat.

L'últim capítol parlaré sobre les conclusions, aclariment de conceptes més rellevants e impressions personals o dificultats presentades degut a la situació viscuda. És farà incís en possibles millores o afegits que és poden implementar.

1.2. Motivació

Els avenços tecnològics aconseguits fins ara fa que les indústries agafin avantatge davant competidors i entrin al mercat industrial amb molts coneixements enfront el que fabriquen o el servei que proporcionen.

Cal recalcar la diferència entre l'automatització, que ofereix fiabilitat i excel·lent productivitat quan és tracta de treballar en entorns ben estructurats, i d'altre banda la robòtica té l'avantatge de comptar amb autonomia per a la pressa de decisions que permet treballar en entorns canviants.

La reducció de costos dels dispositius elèctrics i electrònics, l'optimització d'etapes en projectes o l'automatització de processos industrials han desencadenat en una nova metodologia de treball, que les empreses del sector asseguren que incrementa beneficis i rendiments de tots els elements d'aquesta.

Per tant, els futurs integrants tant d'empreses industrials com de models empresarials, s'han de sumar aquest nou rumb que prenen els negocis.

1.3. Objectius

L'objectiu principal és la implementació d'un projecte d'automatització i robòtica.
En el qual ens marquem tres grans directrius:

- Entendre que fa la línea de muntatge i les connexions amb les que treballa.
- Obtindre les dades, emmagatzemar-les en una base de dades i posteriorment visualitzar els resultats.
- Validar el projecte i buscar-ne una aplicació.

1.4. Descripció del treball realitzat

Primerament el projecte és va desenvolupar respecte la cel·la industrial del laboratori Schneider, però degut a les circumstàncies viscudes s'ha hagut d'emular i ajustar gran part de les simulacions i dades que és necessitaven.

De caire més teòric, és va començar fent una cerca dels concepte i aspectes a tenir en compte per entendre el funcionament i les connexions, tant dels elements físics, com dels softwares utilitzats.
Hem continuat profunditzant en els estats dels retenidors de la cel·la i com és veien afectats dintre les etapes del projecte.

A nivell pràctic, s'han creat uns fluxos mitjançant el programa Node Red per emular les dades que havíem de rebre dels dispositius programables del laboratori.
Més tard s'han emmagatzemat en el servidor InfluxDB, per així finalment aconseguir un entorn complet de visualització d'aquests resultats, allunyats dels elements físics que contenia la cel·la industrial.

Per acabar, s'ha dissenyat una esquema per a la futura monitorització del projecte gràcies al software Grafana.

A part s'han fet unes proves extres per tal de complementar la part pràctica, ja que ens hem hagut d'adaptar amb els pocs recursos que disposàvem al no tenir accés directe a la cel·la del laboratori.

Aquest estudi simplement és un pas més per fer del procés un entorn variant, la intenció és que operi com una cèl·lula flexible, prenent decisions autònomes, i en un futur ja s'implantarà la intel·ligència perquè es programi a base d'algoritmes.

CAPÍTOL 2. CONTEXTUALITZACIÓ

2.1. Estat de l'Art

2.1.1. Antecedents

Al llarg de l'historia hi ha hagut diferents processos d'industrialització que han desencadenat grans canvis econòmics i socials, anomenats *Revolucions Industrials.*

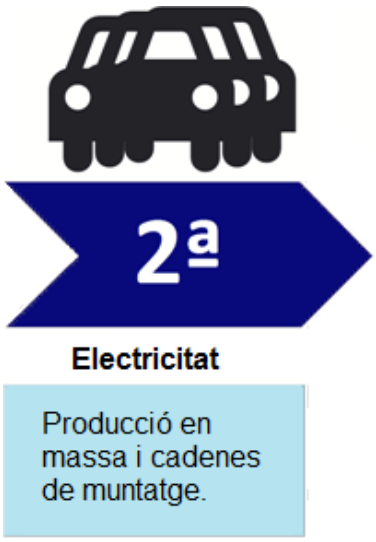
· Primera Revolució Industrial

Lloc: Gran Bretanya - UK

Segle: XVIII (1760 – 1840)

Descripció: Durant aquest període és va viure les transformacions econòmiques, tecnològiques i socials més importants de l'historia de la humanitat. L'economia rural basada fundamentalment en agricultura i comerç, va passar a una economia de caràcter urbà, industrialitzada i mecanitzada.

Figura 1: Primera Revolució Industrial



· Segona Revolució Industrial

Lloc: Europa i Estats Units

Segle: XIX (1870 – 1914)

Descripció: Període de creixement i expansió de noves indústries, com l'acer, el petroli, l'electricitat i l'energia elèctrica per la producció en massa. Els principals avenços tecnològics van ser el telèfon, la bombeta i el motor de combustió interna.

Figura 2: Segona Revolució Industrial

Figura 3: Tercera Revolució Industrial

· Tercera Revolució Industrial

Lloc: Unió Europea, Estats Units i Japó

Segle: XX (1980 – Actualment)

Descripció: Fa referència a l'era de la revolució digital, avenç des dels dispositius electrònics i mecànics analògics, fins la tecnologia disponible avui en dia. Apareixen els primers ordinadors personals, Internet i les comunicacions (TIC).

Figura 4: Quarta Revolució Industrial

· Quarta Revolució Industrial

Lloc: Fira Hannover - Alemanya

Segle: XXI (2011 – Actualment)

Descripció: Caracteritzada per la interconnexió de màquines i sistemes, flux d'intercanvi d'informació amb l'exterior e intel·ligència artificial.
Basant-se en la tecnologia del Internet de les coses, *Big Data* o robòtica, permet que els processos siguin més òptims i eficaços per millorar la productivitat.

2.1.2. Actualitat

"Estem entrant a una revolució tecnològica que modificarà fundamentalment la forma en la que vivim, treballem i ens relacionem. L'escala, l'abast i la complexitat, faran que aquesta transformació sigui diferent a qualsevol altre que l'esser humà hagi experimentat abans**."**

(Klaus Schwab, autor del llibre "La quarta Revolució Industrial" publicat l'any 2016).

Anàlisis DAFO. Indústria 4.0 en la Unió Europea

Fortaleses	**Debilitats**
• Increment en la productivitat, ingressos i eficiència. • Llocs de treball d'alta qualificació i ben remunerats. • Major satisfacció del client amb productes millor personalitzats • Major flexibilitat i control de la producció.	• Capacitat d'adaptació tecnològica: petites disrupcions poden tenir grans impactes. • Dependència de l'èxit: tant de l'estàndard, la coherència de l'entorn, les inversions I+D, etc. • Costos de desenvolupament i posades en marxa. • Pèrdua de potencial de control sobre l'empresa.
Oportunitats	**Amenaces**
• Europa com a líder en la industria de manufacturació i altres sectors. • Desenvolupament de nous mercats punters. • Disminució de les barreres d'entrada per algunes PYMES per participar en noves cadenes de subministrament.	• Seguretat digital, propietat intel·lectual i privacitat de les dades. • Treballadors, economia nacionalista i PYMES. • Volatilitat de les cadenes de valor globals i vulnerables. • Aportació de l'industria 4.0 per part de competidors.

Taula 1: Anàlisis DAFO Indústria 4.0

Font: Smit et al (2006).

2.2. PLC (Programmable Logic Controller)

Definició: És un dispositiu programable que s'encarrega de la gestió automàtica d'un procés industrial, en el nostre cas, el control de la maquinaria d'unes línies de muntatge.

2.2.1. Història:

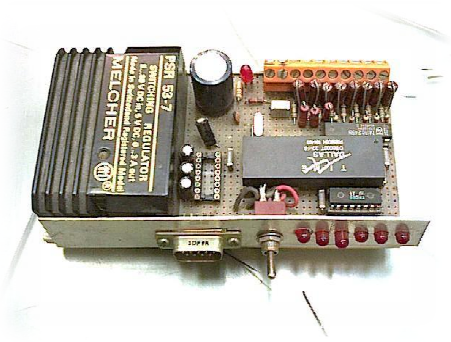


Figura 5: PLC Modicon

1968

És remunta als anys 70 i neix de la mà de Dick Morley, com a resposta a la demanda creixent d'implementar un dispositiu flexible a modificacions, resistent a la contaminació ambiental, vibracions, temperatures extremes i pertorbacions elèctriques, les quals estaven augmentant amb l' incorporació d'electrònica i noves tecnologies.

1969

La empresa americana General Motors, encarregada de la fabricació d'automòbils i motors, va ser la primera en reemplaçar els cablejats inflexibles per PLCs, que es feien servir en les seves línies de producció.

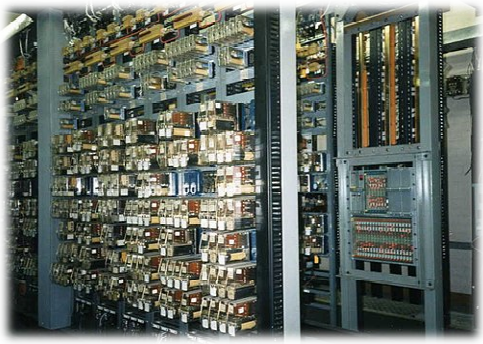


Figura 6: Industrialització dels PLCs

1971

És van expandir a altres indústries perquè aconseguien fer més fàcil la comunicació home-màquina.

En aquells temps els microprocessadors no eren tan ràpids, però eren molt populars, entre els Modicon:

- L'ADM 2901 i el 2903.

1973

Va aparèixer el primer sistema que feia possible comunicar PLCs entre ells, anomenat Modbus.

Figura 7: Connexió Modbus entre PLCs

1974

Se'ls va augmentar la pròpia memòria i gaudien de la possibilitat de tenir entrades i sortides dirigides. Inclús podien estar allunyats de la maquinaria que controlaven, però encara faltava estandardització, degut als canvis constants en la tecnologia e industria.

1980

Al Novembre dels anys 80, és va fer un intent per fer estàndard la comunicació entre PLCs amb el Protocol d'automatització de manufactura de la General Motors.

1985

La mida va disminuir i la seva programació és realitzava a través del PC en comptes de dispositius dedicats només aquell ús.

Figura 8: Connexions elèctriques d'un PLC

1990

És va intentar combinar tots els llenguatges de programació dels PLCs en un sol:

- L'Estàndard IEC 1131-3.

1992

Malauradament, amb noves funcionalitats dels PLCs van anar apareixen nous llenguatges més adequats per realitzar altres tasques, com càlculs complexes o sentencies condicionals, que permetien una reutilització del codi.

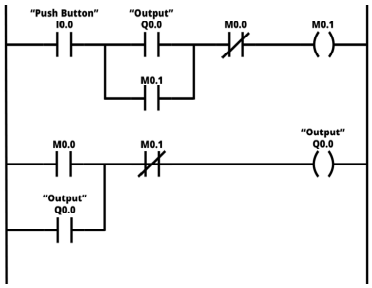


Figura 9: Llenguatge "Ladder"

1993

Això va portar que a finals d'any, la IEC va publicar una revisió de la norma que havien publicat anys anteriors, l'Estàndard 61131, i que avui en dia encara és vigent.

2000

A començaments del segle XXI és quan el món del PLC ha tingut un millor avenç. Reducció de costos en els equips electrònics més complexes i en la seva electrònica.

Actualment

El cicle d'operació d'un PLC té una sèrie d'etapes que funcionen de forma seqüencial i repetitiva, millorant la seva velocitat de processament.

En els últims anys, han sorgit controladors específics amb aplicacions tant interessants com la visió artificial o de seguretat.

Figura 10: PLC controlat per un PC

2.2.2. Parts: L'estructura és divideix en diferents parts, les quals poden estar integrades o per mòduls.

Element	Descripció
Font d'Alimentació	Encarregada de subministrar l'energia elèctrica necessària a la CPU i a altres dispositius del PLC.
CPU	La unitat central de processament, és l'encarregada d'interpretar cada una de les instruccions que té programades el PLC.
Mòduls	Són imprescindibles ja que a través d'ells és possible la connexió física entre la CPU i el sistema a controlar. Existeixen dos tipus: • Entrada: Les quals envien una retroalimentació al PLC perquè pugui processar les dades. Com podrien ser els interruptors. • Sortida: En base a la programació donada al PLC, envia una resposta als actuadors per controlar el procés. • Per exemple: transistors o relés.
Memòria	On és guarda el programa del PLC. Les que més es fan servir són: *RAM, ROM, EAROM*, entre d'altres.
Unitat de programació	És la que permet comunicar l'operari amb el sistema. Les seves funcions són la transferència i modificació de programes, la verificació i la informació del funcionament dels processos.

Taula 2: Elements que conté un PLC

2.3. Llenguatges de Programació

Definició: Són símbols, caràcters i regles d'ús que van ser dissenyades per poder tenir una comunicació entre l'usuari i la màquina. Gràcies aquest vincle, som capaços de controlar el funcionament del nostre procés.

En l'actualitat, els principals fabricants de PLCs han normalitzat els llenguatges de programació, recollits a la norma internacional de l'IEC 61131-3.

Tipus:

- Alt nivell: Categoria on s'utilitza una interfase de símbols per declarar les instruccions de control de manera clara, però la programació està limitada als símbols existents.

  → 2.3.1. Ladder: Era el llenguatge més utilitzat als anys 90. També se li diu diagrama d'escala perquè s'assembla a la estructura que té, ja que conté rails verticals i horitzontals que representen els esglaons.

Característiques:

1) *Els dos rails verticals són l'alimentació: Voltatge i terra.*
2) *Les sortides sempre es col·loquen al costat dret.*
3) *Les instruccions és col·loquen al costat esquerre.*
4) *És poden escriure instruccions i sortides en paral·lel.*
5) *El processador del PLC interpreta les dades de dalt a baix i d'esquerra a dreta.*

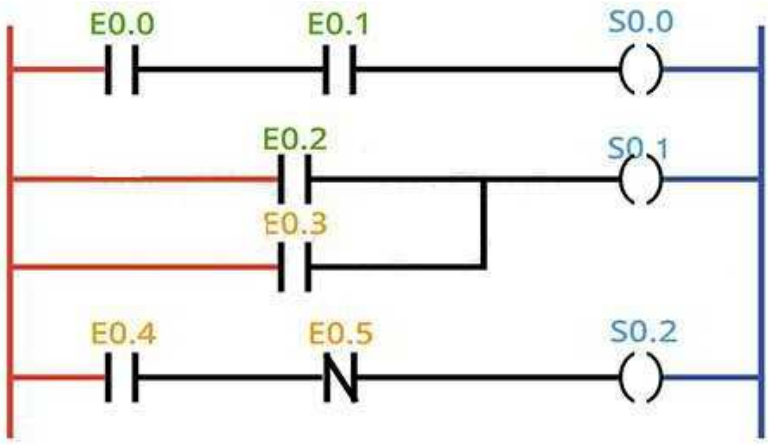


Figura 11: Esquema del Llenguatge Ladder

2.3.2. Diagrama de Blocs: S'utilitzen blocs de símbols lògics, equivalent a les portes lògiques. La sortida està assignada a les sortides dels blocs lògics, pel que no cal incorporar-hi una bobina.

Característiques:

1) *Les sortides dels blocs no es connecten entre elles.*
2) *L'avaluació d'una xarxa s'acaba abans d'iniciar la següent.*

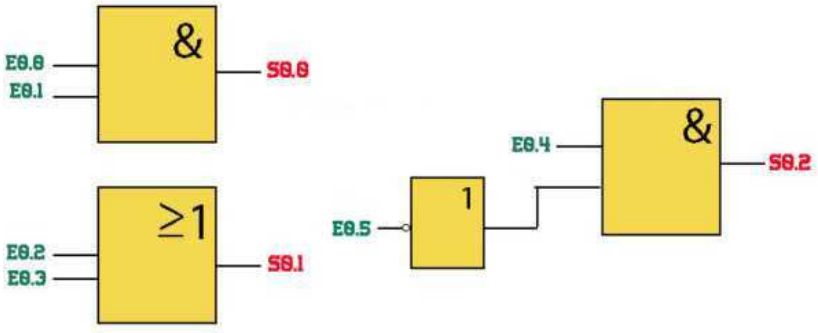


Figura 12: Esquema del Llenguatge de Diagrama de Blocs

- Baix nivell: Categoria on és troben els llenguatges a través de text, ens els que s'utilitzen cadenes de caràcters per indicar les instruccions de control.

  2.3.3. Text estructurat: Està format per una sèrie d'instruccions que s'executen de forma condicionada.
  És un llenguatge molt semblant al *C o C++.*

Característiques:

1) *Tracta de la mateixa manera majuscules i minuscules.*
2) *Suporta instruccions amb aritmetica complexe.*

```
IF ((E0.0 == TRUE) && (E0.1 == TRUE))
{
S0.0 = TRUE;
}
ELSE S0.0= FALSE;

IF ((E0.2 == TRUE) || (E0.3 ==TRUE))
{
S0.1 = TRUE;
}
ELSE S0.1 = FALSE;
```

Figura 13: Esquema del Llenguatge de Text Estructural

2.3.4. Llistat d'Instruccions: És el més antic, ja que és la base de tots els altres llenguatges de programació que existeixen. S'utilitzava quan els ordinadors encara no tenien gràfics.

Característiques:

1) *Tots els llenguatges poden ser traduïts a aquest, però no a l'inrevés.*
2) *La programació és més compacta.*
3) *És el llenguatge més complet de tots.*

```
U E0.0
U E0.1
= S0.0

U E0.1
O E.02
= S0.1

U   E0.3
UN  E0.4
=   S0.2
```

Figura 14: Esquema del Llenguatge de Llistat d'Instruccions

2.4. Xarxes

Definició: És un conjunt de sistemes informàtics independents, connectats entre sí, de tal manera que fan possible el intercanvi de dades.
Necessària tant la connexió física com la connexió lògica de tots els sistemes mitjançant protocols especialitzats. Depenent de la mida o l'abast de la xarxa, trobem diverses escales de xarxes on destacant les següents:

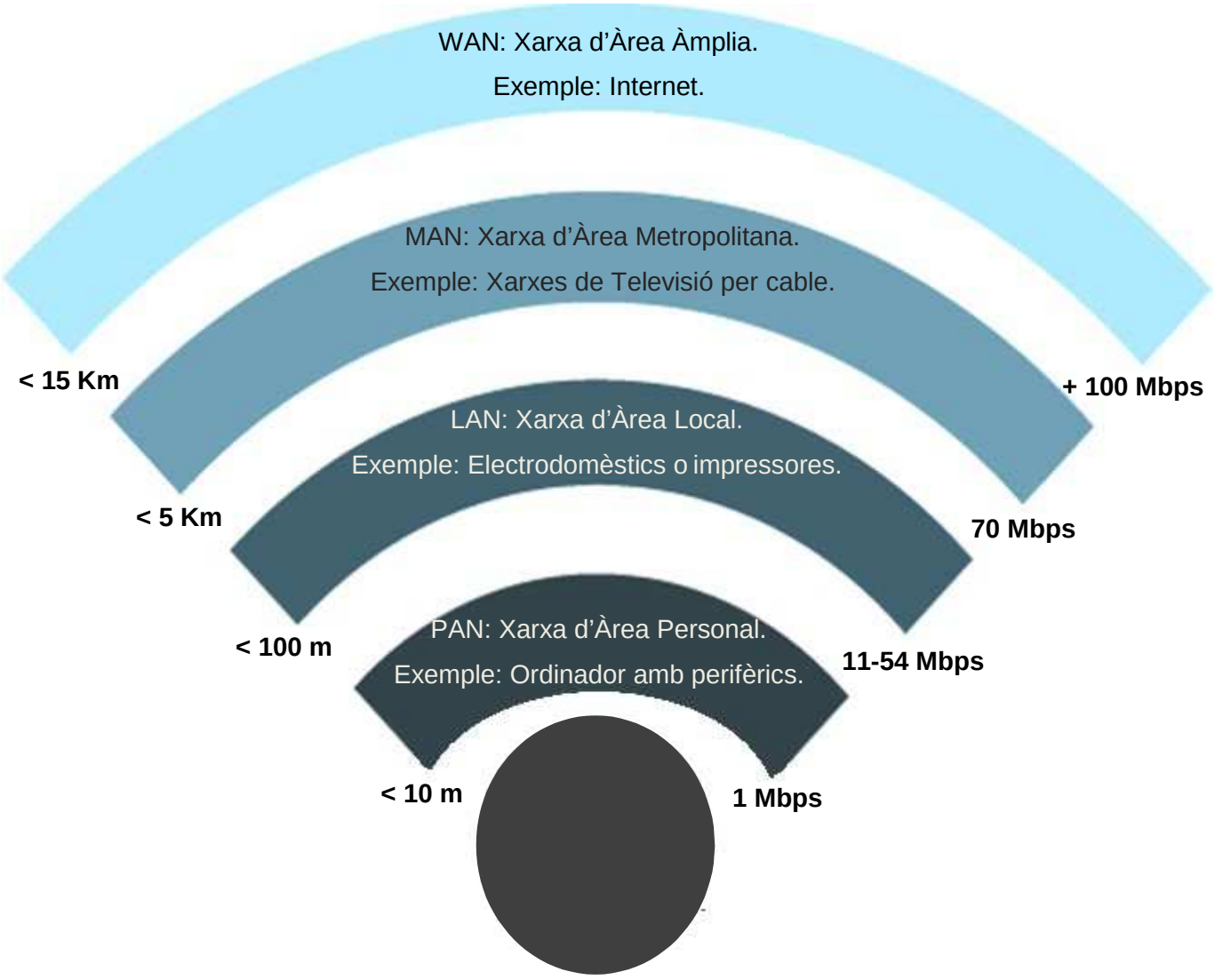


Figura 15: Tipus de Xarxes informàtiques.

2.5. Topologies:

Defineix com està dissenyada la xarxa, tant de manera física com lògica.
És la manera en que tindrem el cablejat que es connectarà als ordinadors d'una xarxa. Cada topologia és especifica al disseny i en destaquen quatre d'importants:

2.5.1. Anell:

És un tipus de topologia simple, on les estacions de treball, és troben connectats entre ells formant un cercle. La informació viatja en bucle, en un únic sentit pel que si un node deixa de funcionar cau la xarxa o deixa de transmetre informació a la resta d'ordinadors que és troben dins d'aquest anell. Actualment no es fa servir perquè és poc eficaç.

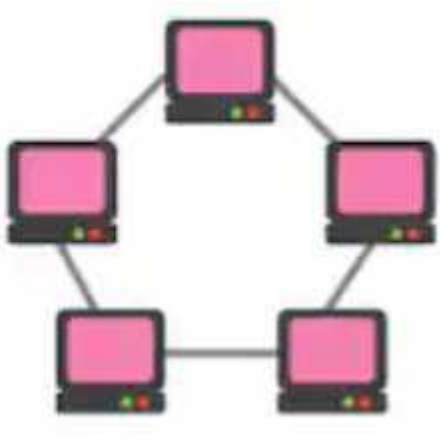


Figura 16: Topologia Anell

2.5.2. Estrella

La distribució de la informació va des d'un punt central, anomenat Host, cap a tots els nodes de la xarxa. El *Host* és el que realitza tota la feina, fa de servidor local que administra la informació que li arriba. A diferència de la topologia d'anell, si un node falla la xarxa continua treballant sense cap inconvenient. Actualment és molt utilitzada gràcies a la seva eficàcia.

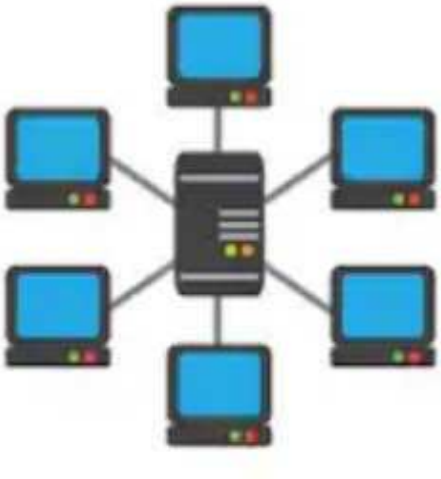


Figura 17: Topologia Estrella

2.5.3. Malla:

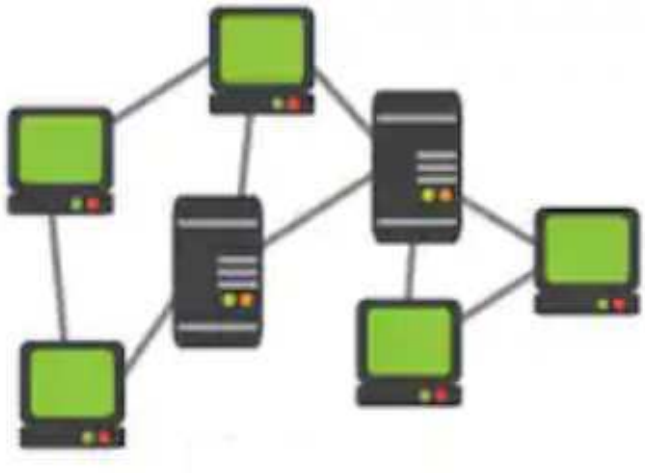


Figura 18: Topologia Malla

Aquesta topologia és defineix amb el nom de disseny de trama. És tracte d'un ajust d'interconnexió entre nodes, dibuixant la figura d'una malla. Important perquè la informació pot viatjar mitjançant diferents camins, de manera que si falla algun node sempre hi haurà una altre manera d'arribar al final. Actualment s'usa entre les xarxes WAN o d'àrea àmplia.

2.5.4. Bus:

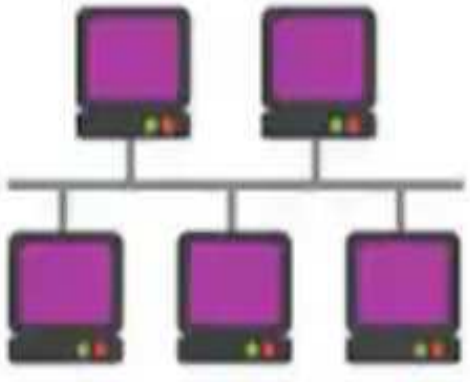


Figura 19: Topologia Bus

Partim d'un cable central, el qual porta la informació a tots els ordinadors en forma de ramificacions, de mode que les dades viatgen de manera seqüencial per tots els nodes de la xarxa. El desavantatge és que degut a la seva distribució, si el cable central és interromput, la xarxa queda inutilitzada.
Actualment és poc utilitzada, tot i que encara s'usa en processos industrials.

També n'existeixen de tipus híbrid que són una combinació de dos o més, de diferents topologies, per adaptar-se a les necessitats del client. Aquestes topologies hibrides tenen infinites varietats i s'ajusten a l'estructura física i als equips on s'implementa la xarxa.

2.6. Transmissió de Dades

Definició: És la transferència d'informació entre dos dispositius.
Aquest mètode de transmissió, defineix la direcció del flux dels senyals entre els dispositius connectats.

Tipus:

2.6.1. Simplex: Com el seu nom indica, és el mode de transmissió més simple. Només envia dades en una sola direcció, és a dir, de l'emissor al receptor.

Figura 20: Comunicació Simplex

2.6.2. Half-Dúplex: Les dades és poden enviar en dues direccions, però no simultàniament. L'emissor envia dades i el receptor ha d'espera a que li arribin, abans de poder contestar.

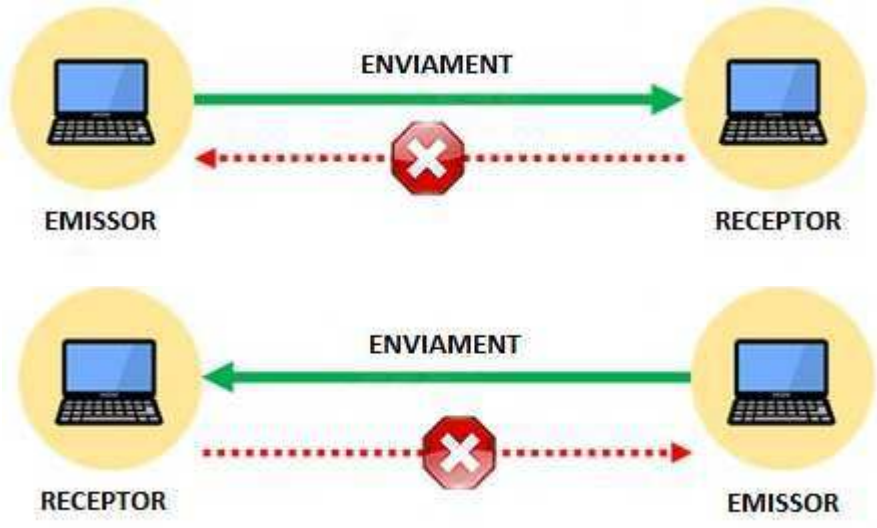


Figura 21: Comunicació Half-Dúplex

2.6.3. Full-Dúplex: En aquest mode complert, la transmissió de l' informació entre l'emissor i el receptor és pot produir simultàniament. S'utilitza quan la comunicació en ambdues direccions és necessària tot el temps.

Figura 22: Comunicació Full-Dúplex

2.7. Model OSI

Definició: És un estàndard desenvolupat per l'organització ISO l'any 1984.
Té com a objectiu principal aconseguir connectar sistemes de diferent procedència perquè aquests puguin intercanviar informació sense ningun tipus d'impediment, degut als protocols amb els que operen de forma pròpia segons el seu fabricant.

El model té diferents nivells o capes, que són el conjunt de funcions especifiques per facilitar la comunicació física, agrupades en una entitat que a la vegada és relaciona amb nivells superiors e inferiors.

Nivells:

- Capa 1: Física
  - Gestiona els procediments a nivell electrònic.
  - Defineix el medi físic de transmissió: cables, ones i fibra òptica.
  - Controla senyals elèctriques i transmet el flux de bits.
  - Defineix les característiques dels materials.
- Capa 2: Dades
  - S'encarrega d'establir les connexions dels elements físics.
  - És construeix la trama de bits, la direcció de dades i la detecció d'errors durant la transmissió.
  - El *switch* rep i envia dades del emissor al receptor.
- Capa 3: Xarxa
  - Identifica si dues o més xarxes tenen el mateix itinerari.
  - És necessari que la capa conegui la tipologia de la xarxa.
  - El protocol més conegut que s'encarrega d'això és la IP.

Capa 4: Transport

- Realitza el transport de dades des de l'origen fins al final.
- És dur a terme de forma independent al tipus de xarxa.
- Treballa amb ports lògics i s'ha de proporcionar la qualitat suficient perquè el missatge és transmeti correctament.

Capa 5: Sessió

- Treballen directament amb les aplicacions que sol·liciten nivells inferiors.
- Controla i manté actiu l'enllaç entre maquines.
- Adequa l' informació perquè sigui comprensible per l'usuari.

Capa 6: Presentació

- S'encarreguen de la representació de l' informació transmesa.
- Assegura que les dades arriben a l'usuari.
- Treballa amb continguts útils que l'usuari entén.

Capa 7: Aplicació

- Última capa, i l'encarregada de permetre als usuaris executar accions i ordres en les pròpies aplicacions.
- Permet també la comunicació entre la resta de capes.

2.7.1. Model TCP/IP

Té els mateixos objectius que el Model OSI, però aquest s'usa per descriure, discutir i comprendre funcions de xarxes individuals.

Tot i que el Model OSI és genèric e independent del protocol que s'utilitzi, ja que tots s'adapten a ell, aquest és basa en protocols desenvolupats per Internet.

Nivells:

Capa 1: Accés
- Capa d'enllaç responsable de col·locar i rebre els paquets TCP/IP en el portador de dades de la xarxa.
- És pot utilitzar per connectar diferents tipus de xarxes.

Capa 2: Internet
- És responsable de les funcions d' empaquetatge i itinerari del *host.*
- La IP agrega la capçalera als paquets per donar-los una direcció.

Capa 3: Transport
- S'encarrega de proporcionar la comunicació a les capes.
- El protocol usat és el TCP, que proporciona un servei de comunicacions individuals, fiables i orientades a la connexió.
- També recupera els paquets perduts durant la transmissió.

Capa 4: Aplicació
- Ofereix la capacitat d'accedir als serveis d'altres capes.
- Defineix els protocols que s'utilitzen per intercanviar dades.

2.8. Comunicacions Industrials

Definició: La comunicació entre dispositius succeeix a través d'una connexió especial a base de normes, que permeten la transferència de dades o informació entre cada una d'ells. A aquests tipus de normes se'ls coneix com *Protocols de Comunicació*.

A continuació descrivim les formes de comunicació més recurrents:

2.8.1. Sèrie – (Bus de comunicacions)

És el sistema bàsic que es proporciona per cada controlador.

Les interfases de comunicació poden estar incorporades a la CPU o en mòduls de processament no integrats. Aquestes interfases s'utilitzen per transferir dades a alta velocitat entre PLC i el dispositiu remot.

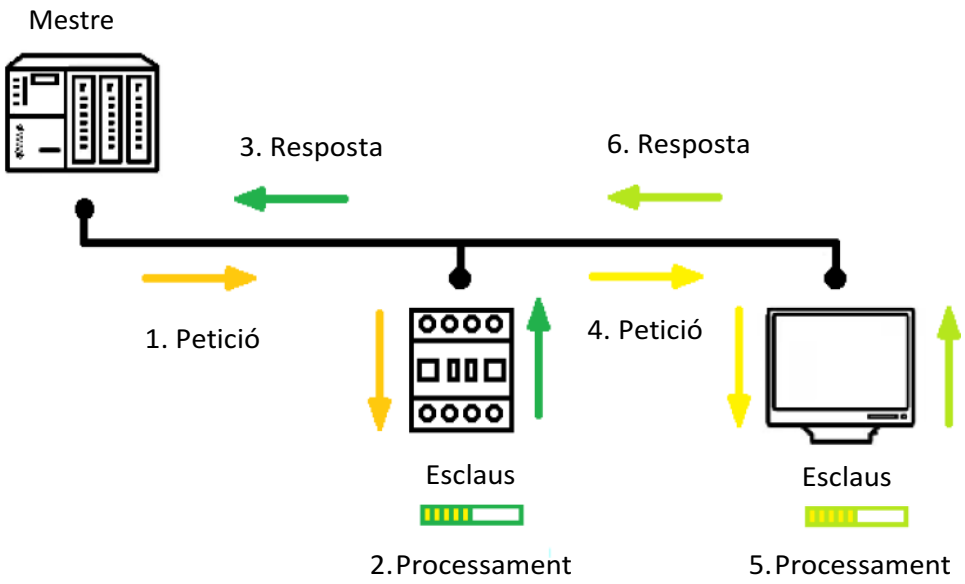


Figura 23: Comunicació en sèrie mestre – esclau.

Parlem d'emissor com a un mestre i dels receptors com a esclaus.

1. *El mestre inicia la comunicació i envia una petició als esclaus.*
2. *L'esclau la rep i la processa.*
3. *Dona una resposta que arriba al mestre i aquest la processa.*
4. *Només tancant el cicle, el mestre pot enviar una altre petició.*

Només el mestre es qui pot iniciar la comunicació, i disposa de dos mètodes per dirigir-se als esclaus:

- Mode de Difusió: L'emissor envia una sol·licitud a tots els receptors. Tots els receptors han d'acceptar el missatge i tenir la mateixa direcció associada, però no el responen, només l'accepten.

- Mode de No difusió: L'emissor es dirigeix únicament a un receptor de forma individual. No cal que els receptors tinguin una única direcció.

2.8.2. Paral·lel – (Xarxa de comunicacions)

El funcionament de les interfases és el mateix que en sèrie.

Canvia el mètode en que l'emissor intercanvia informació amb els receptors.

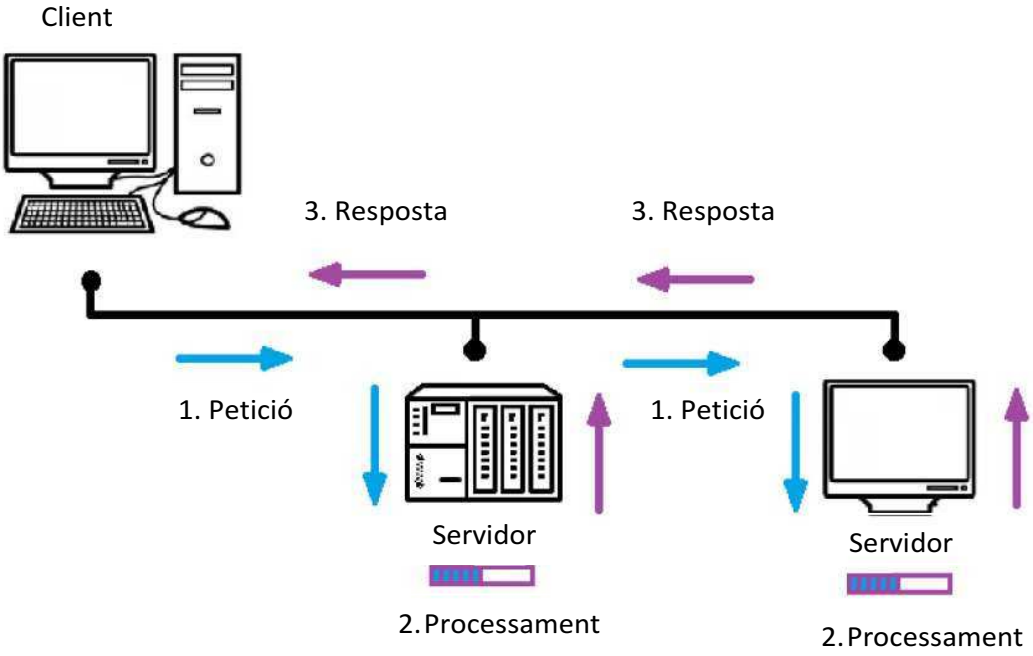


Figura 24: Comunicació en paral·lel client – servidor.

En aquest cas l'emissor és el client i els receptors són els servidors.

5. *El client és capaç d'enviar una o múltiples peticions als receptors.*
6. *Els servidors són capaços de rebre totes les peticions, processar-les i gestionar-les simultàniament.*
7. *El client és capaç de rebre les respostes al mateix temps.*

2.9. UART (Universal Asynchronous Receiver-Transmitter)

Definició: És el dispositiu que controla les comunicacions en sèrie de ports i altres dispositius. S'encarrega de llegir les dades que arriben, generar-les i gestionar interrupcions. També envia dades i controla els temps de bit.

Funcionament: És el mateix en tots els dispositius, ja que sinó fos així, no és podrien comunicar entre ells. Suposem que volem enviar dades a través del port sèrie, en concret el caràcter "b":

Caràcter	Taula ASCII	Codi Binari
b	98	01100010_2

.

Taula 3: Conversió caràcter "b"

1. Un cop traspassat el caràcter a binari, ajuntem tots els bits en una trama.
2. Invertim l'ordre dels bits, l'últim passa a ser el primer i així successivament. Això és degut a que amb aquest protocol el primer bit és el menys significatiu.
3. La línea de transmissió sempre que estigui en repòs estarà a un nivell alt.
4. Per iniciar la comunicació hem d'enviar un bit d'arrencada, que sempre serà un 0 i el qual mantindrem durant un temps determinat anomenat temps de bit.
5. Passat el temps de bit començarem a enviar les dades, bit a bit.
6. Cada bit és manté a la línia de dades durant tot el temps de bit.
7. Finalitzat l'enviament de dades hem d'alliberar la línea, enviant un bit de stop amb valor 1, amb el que indicarem que ja hem acabat.
8. Després la línea quedarà a l'espera en un nivell alt, com teníem a l' inici.

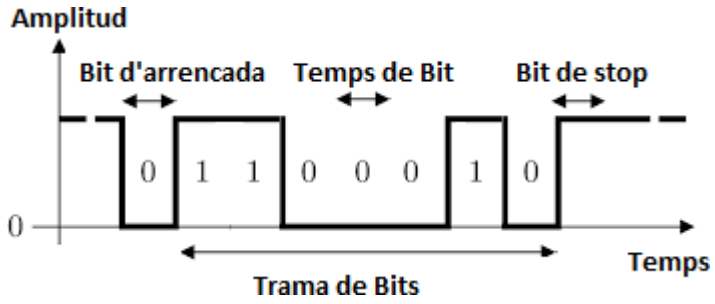


Figura 25: Enviament de la trama de bits

Com hem explicat, el temps de bit no és altre cosa que el temps en el qual mantindrem un bit en la línea de transmissió.
El càlcul és senzill, ja que està relacionat amb la velocitat de transmissió de les dades *("baud rate"),* el qual ens indica el numero de bits que podem enviar per segon. Mitjançant la formula que ens relaciona els dos termes obtenim:

$$\text{Temes de bit} = \frac{1}{\text{baud rate}}$$

Formula 1: Relació entre Temps de bit i velocitat de transmissió

Per exemple; una comunicació en sèrie pot ser a 9600 bauds, és a dir, és poden transmetre fins a 9600 bits en un segon.

$$\text{Temes de bit} = \frac{1}{9600 \text{ bits/segon}} = 104{,}2\ \mu S$$

En conclusió si volem enviar el caràcter "b" a 9600 bauds, hem de mantenir el bit d'arrencada durant 104,2 µSi els bits de la trama, s'aniran enviant també al mateix temps l'un rere l'altre.

En les comunicacions digitals el que tindrem és una línea de massa, i com a conseqüència a aquesta línea hi haurà tensions. Si parlem de capes físiques, direm que els valors 0 és converteixen en 0 volts i els valors 1 compresos entre 1,8 i 5 volts.

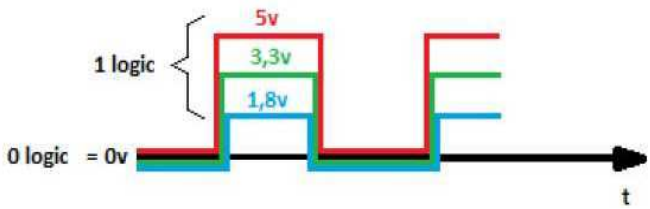
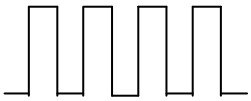
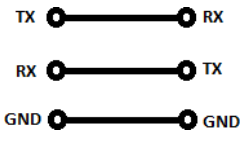
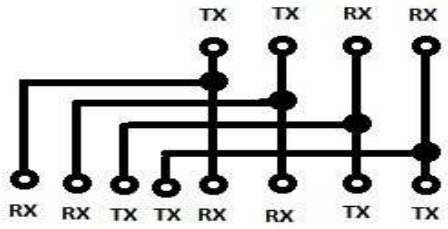


Figura 26: Enviament de dades a través de la capa física

2.10. Ports Sèrie

Definició: Són estàndards físics que normalment vinculant l'ordinador amb els perifèrics, i que permeten enviar i rebre dades bit a bit.
S'implanten amb RS, que significa estàndards recomanats i n'hi ha de diferents tipus adequats a cada necessitat.

Nom	RS-232	RS-422	RS-485
Dispositius	1 Transmissor 1 Receptor	5 Transmissors 50 Receptors (10 x transmissor)	32 Transmissors 32 Receptors
Tipus de Transmissió	Simplex o Full-dúplex	Simplex o Half-Dúplex	Simplex o Half-dúplex
Longitud màxima de cable	15 metres Distancies curtes	1220 metres Distàncies llargues	
Velocitat transferència de dades	19.2 Kbps Vel. baixes	10 Mbps Velocitats altes	
Senyal			
Voltatge mínim d'entrada	+/- 3 V	0.2 V diferencial	
Corrent de sortida	500 mA	150 mA	250 mA
Connexions			

Taula 4: Característiques dels diferents ports sèrie RS

En els ports sèrie tenim dos línies, Tx i Rx. Aquestes línies s'han de creuar per comunicar els dos dispositius, és a dir, Tx del primer dispositiu ha d'estar connectat a Rx del segon dispositiu. La línea Tx del segon s'ha de connectar a Rx del primer. A més els dos controladors han de compartir una massa comuna.
Quan parlem de connectar terra, ens referim a l'abreviatura GND referent a "ground".

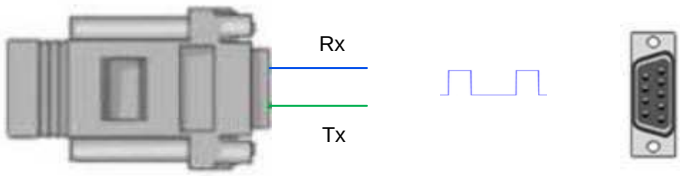


Figura 27: Connexió en sèrie amb estàndard RS485

En altres circumstàncies podem trobar dispositius en els que només existeix Tx. Això és degut a que aquell controlador només envia dades, no necessita rebre-les. Un clar exemple són els sensors o un receptor GPS.
Amb el port sèrie és transmeten els bits un a un, fent llargs temps de transmissió.

Existeixen ports paral·lels que transmeten de 2 a 8 bits simultàniament.
Per aconseguir-ho necessitem que els connectors tinguin més línies.

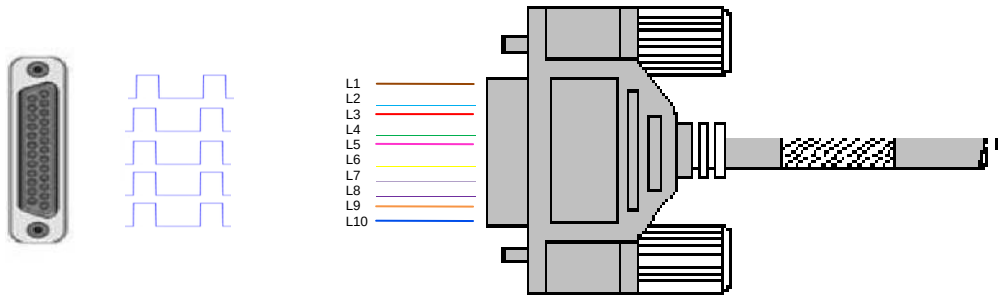


Figura 28: Connexió en paral·lel amb estàndard RS232

2.11. CAN (Controller Area Network)

Definició: És un protocol de comunicació de topologia Bus, desenvolupat per l'empresa alemanya Bosch l'any 1993 sota les restriccions de la norma ISO 11898. Transmet missatges en entorns distribuïts i ofereix solució per a la gestió de comunicacions entre múltiples CPUs.
També aporta beneficis davant d'altres protocols com:

- Habilitat per diagnosticar i reparar errors de dades.
- Protocol normalitzat, és pot connectar subsistemes de diferents fabricants.
- El *Host* delega la càrrega de comunicacions als perifèrics, pel que disposa de més temps per executar les pròpies tasques.
- Redueix el cablejat i elimina les connexions punt a punt.

Trama general Bus CAN:

Identificador	Control	Dades	CRC
12 bytes	6 bytes	8 bytes	16 bytes

Taula 5: Trama general Bus CAN

Identificador: S'inicia amb un bit dominant. En aquest camp s'identifica el node i la prioritat, en l'ordre, de la trama dins la xarxa, és a dir, s'estableix un ordre en el qual les dades seran transmeses. Inclou un bit addicional per determinar si és una trama de dades, valor 0, o de petició.
Control: Els dos bits inicials estan reservats per aplicacions futures, i els darrers 4 defineixen la mida de la trama de dades del darrer camp.
Dades: Inclou la informació necessària per executar les ordres.
CRC: Assegura la integritat de les dades. Acaba amb un bit recessiu pel delimitador CRC, dos bits que actuen com a cel·la de reconeixement i el fi de trama.

2.12. Profibus

Definició: És una xarxa estandarditzada de tipologia bus de camp, que s'encarrega de la comunicació entre sensors i sistemes de controladors basant-se en el funcionament del model OSI.
Desenvolupat l'any 1987 per empreses alemanyes com Siemens i Bosch.
Respecte PLCs i cèl·lules industrials necessiten comunicar-se entre sí, pel que requereixen un gran paquet de dades en diferents i potents funcions de comunicació per ser transferides.

Tipus:

· DP

És la solució d'alta velocitat. Dissenyat específicament per la comunicació entre sistemes d'automatització i equips remots.

Utilitza interfases de comunicació físiques com el RS-485 o la *fibra òptica*.

Actualment en menys de dos minuts pot transmetre 1 Kbyte des de l'entrada fins la sortida.

· PA

És el protocol que millor compleix els requisits d'automatització industrial.

Permet la medició i el control a través d'una línea y dos cables.

Afegeix i elimina estacions de bus sense afectar altres restriccions.

Té comunicació transparent amb els altres dos protocols.

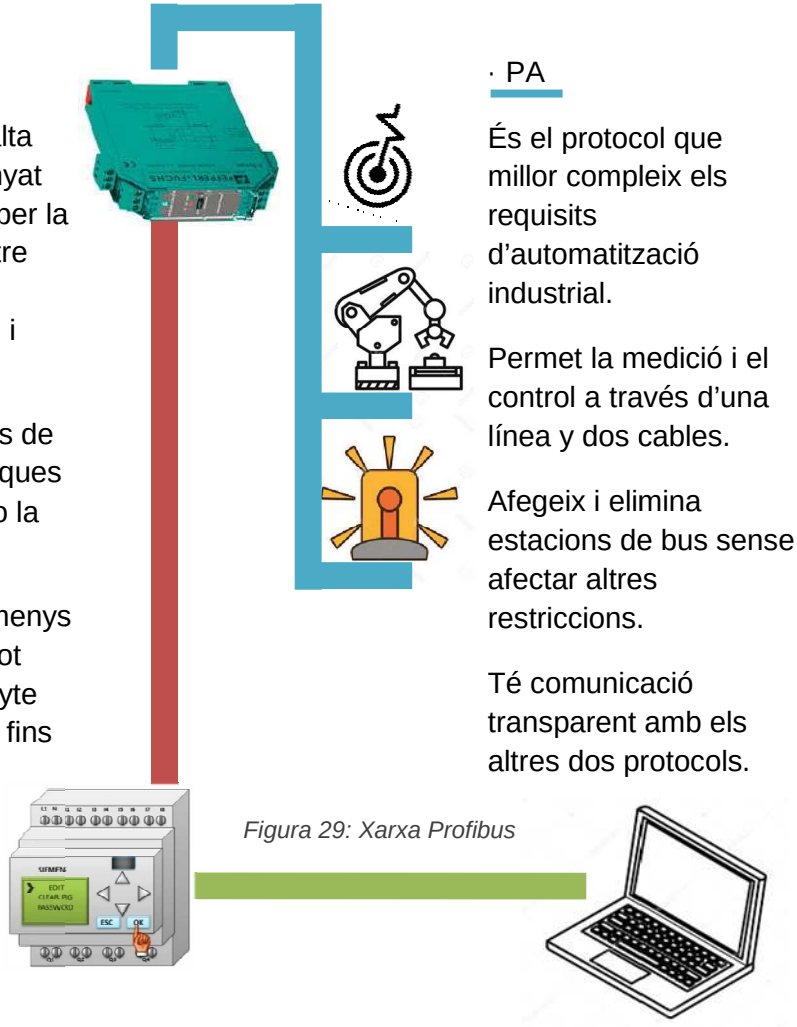


Figura 29: Xarxa Profibus

· FMS

És el primer protocol que és va utilitzar per a la solució de tasques complexes durant les comunicacions entre PLCs i el que conté més funcionalitats cap a l'usuari.

Intercanvia dades entre equips intel·ligents, utilitzats normalment com a nivells de control.

Actualment està essent substituït pel protocol Ethernet.

2.13. Ethernet

Definició: És l'evolució de la comunicació sèrie que te com a base la norma internacional IEE 802.3. És un estàndard de xarxes d'àrea local creades per a la unió de diferents ordinadors a través de cables. Té com a objectiu la connexió entre diferents dispositius sense la necessitat de connexió externa, com Internet. A part de compartir dades entre ordinadors, també permet que els equips és connectin a un mateix perifèric. Pot tenir diferents topologies com d'Anell o Malla i la connexió és realitza a través d'una LAN.

Funcionament: Per aconseguir enviar les dades, és necessari fragmentar-les en petites fraccions denominades paquets de commutació, evitant així que col·lapsi.

Utilitzen la *codificació Manchester*, útil en la transferència de dades binaries en base a senyals analògiques i digitals de alta velocitat o llarga distància.
És un mètode en el qual cada temps de bit hi ha una transferència entre dos nivells de senyal. Aquesta codificació està sincronitzada, ja que a cada bit se li associa un senyal de rellotge, el que fa possible la fluïdesa del flux de dades.

Un inconvenient seria que consumeix el doble d'ample de banda que una transmissió asíncrona. Avui en dia hi ha nombroses codificacions amb la mateixa finalitat que consumeixen menys ample de banda i que obtenen el mateix resultat.

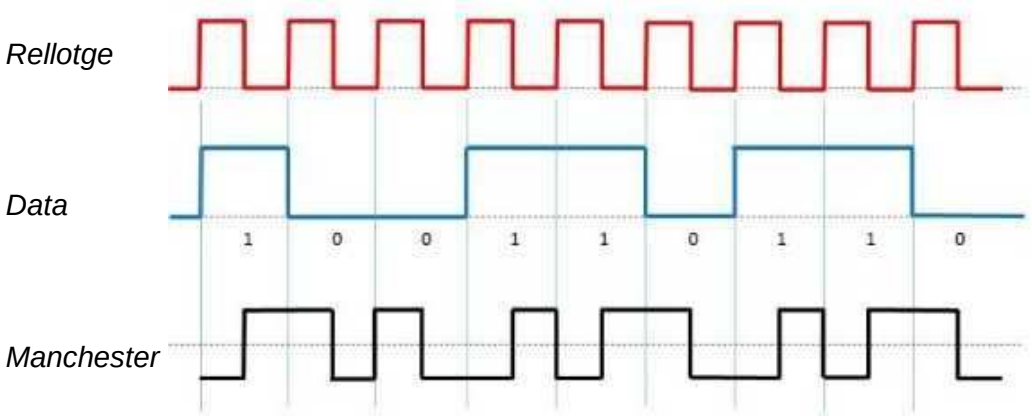
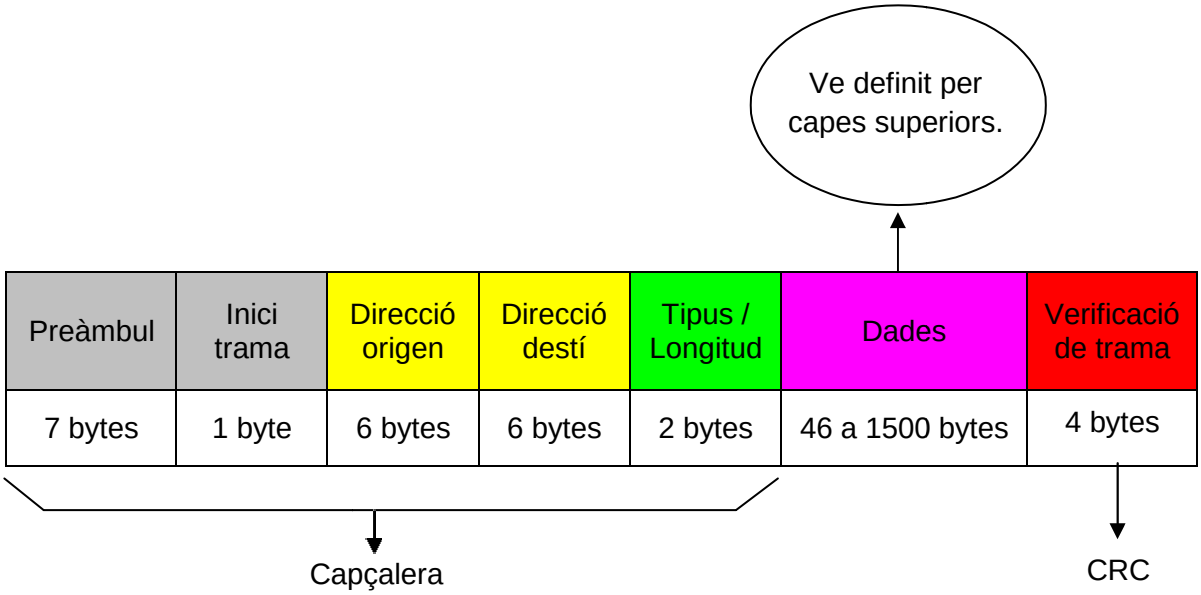


Figura 30: Codificació Manchester

Trama general:



Taula 6: Trama general Ethernet

Preàmbul: Indica a les estacions receptores que una trama és Ethernet mitjançant un patró d'uns i zeros. Inclou un byte addicional equivalent al inici de trama.

Inici de trama: Byte delimitador que finalitza amb la seqüència de dos bits de valor 1. Serveix per sincronitzar els paquets de commutació de totes les estacions.

Direccions d'origen i destí: Inclou les direccions físiques úniques que envia la trama d'inici i el receptor final. La direcció d'origen sempre és una única, mentre que la de destí pot ser enviada a una màquina, a varies o a tots els nodes.

Tipus: Un cop completat el processament de dades Ethernet, s'especifica el protocol amb el que ve de capes superiors.
Longitud: Indica la quantitat de bytes que segueixen aquest camp.

Dades: Inclou la informació enviada en la trama. Si no s'aconsegueix tenir un mínim de 46 bytes, s'han d'incerta fins a completar aquesta mesura.

Verificació trama: Conté una seqüència de valors que el receptor crea per verificar la trama, així com avaluar si la trama està danyada. Quan un paquet és rebut pel destinatari, la capçalera mira si les dades corresponen a un missatge IP i ho traspassa a aquest protocol per després processar-ho.

2.14. Modbus

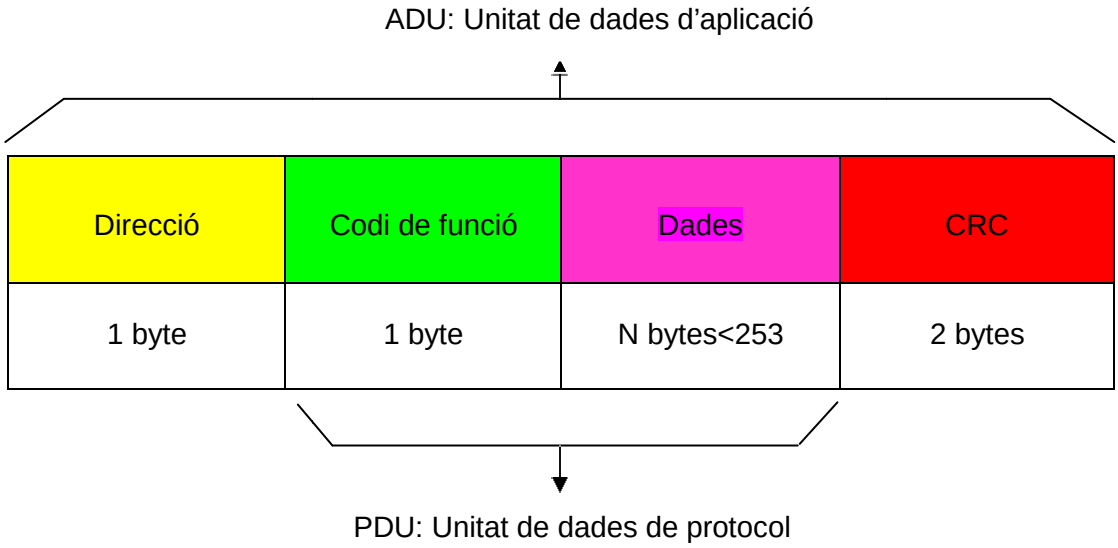
Definició: És de tipus sèrie o bus de comunicació usat principalment per PLCs. Creat per l'empresa Modicon a finals dels anys 70, d'estructura senzilla i que utilitza els nivells 1, 2 i 7 del Model OSI basat en l'arquitectura mestre – esclau o client - servidor.
El seu ús en l'entorn industrial s'ha imposat per sobre a altres protocols de comunicació degut a una sèrie de raons:

- És dissenya tenint en compte la finalitat que tindrà en la industria
- És públic i gratuït
- És fàcil d'implementar, és a dir, no cal un desenvolupament complex
- No suposa restriccions sobre les dades que transporta

Existeixen infinits *mòdems* que accepten aquest protocol i implementació per la connexió, ja sigui per cable, *wireless* o GPS.

Funcionament: Cada dispositiu de la xarxa conté una direcció única. Qualsevol dispositiu pot enviar ordres, tot i que l'habitual és que ho enviï el dispositiu mestre. Cada ordre que rep el mestre sap de quin esclau és. Tots els dispositius reben la trama de dades però només el destinatari l'executa.

Trama general:



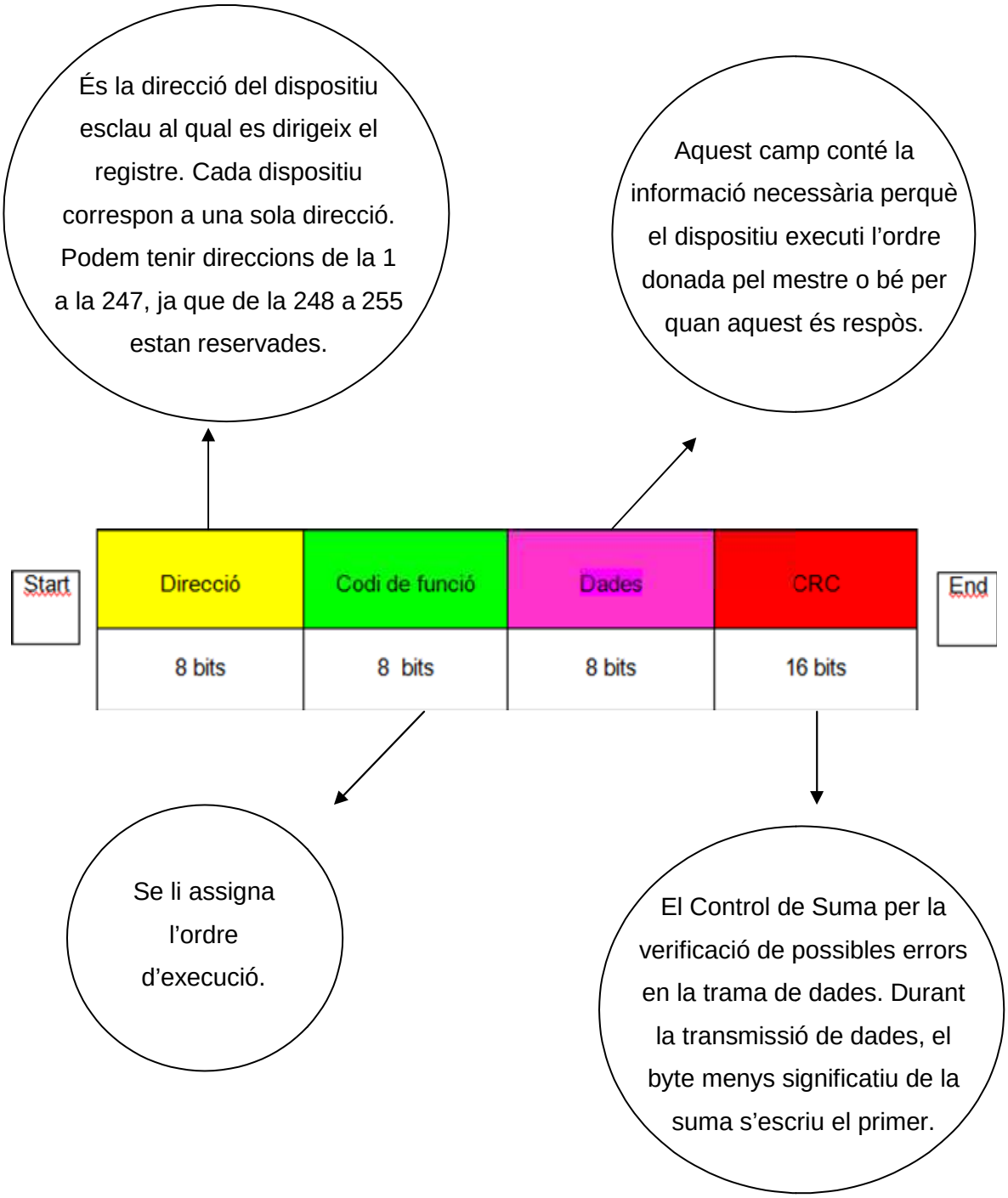
Taula 7: Trama general Modbus

Les ordres bàsiques del Modbus permeten controlar un dispositiu per modificar el valor dels seus registres o bé per veure'ls el contingut.

Tipus:

2.14.1. RTU

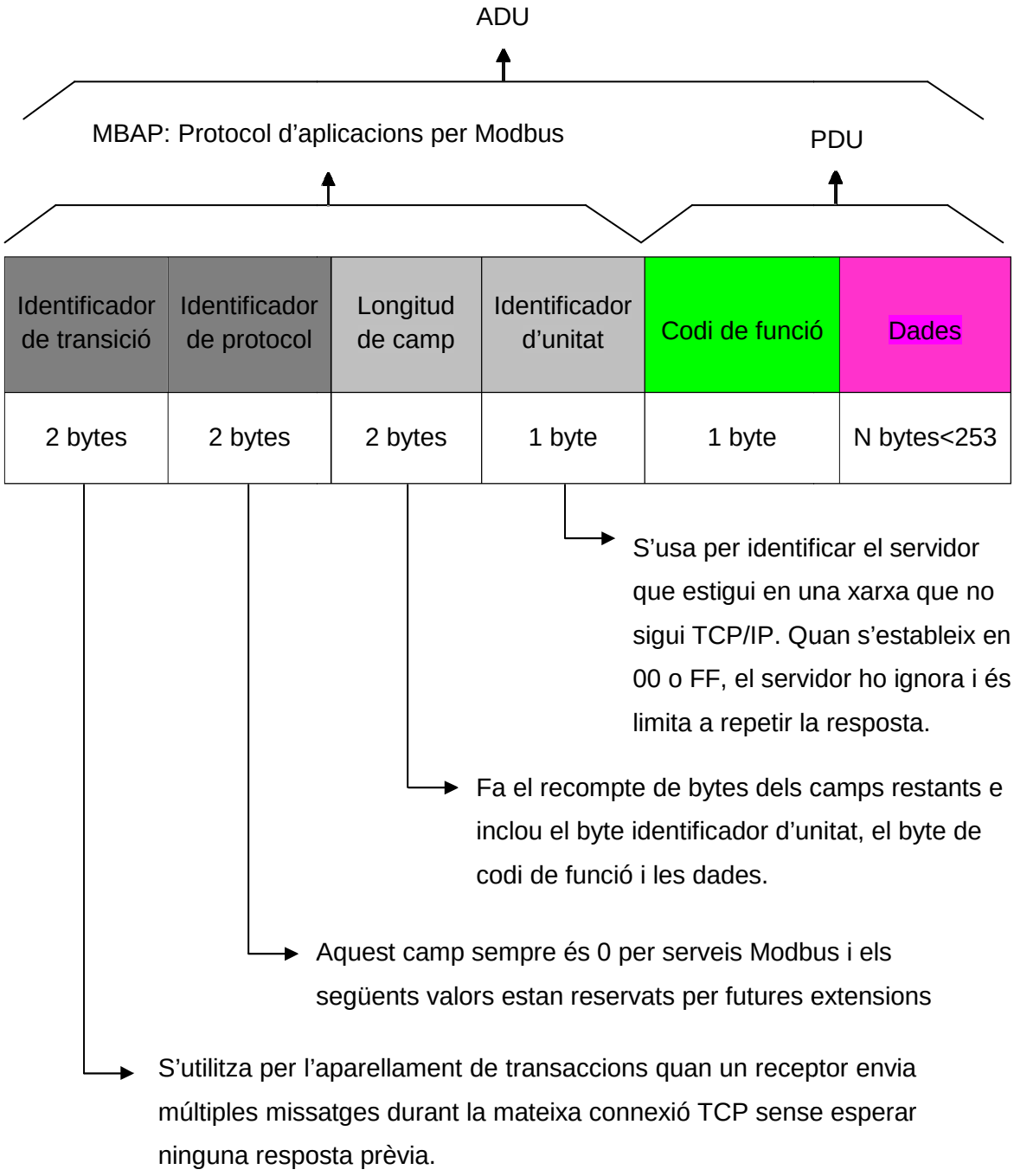
Protocol que permet el control i la supervisió a través d'un bus de camp.



Taula 8: Trama Modbus RTU

2.14.2. TCP/IP

Protocol introduït per Schneider Automation com a variant del RTU i que permet a equips industrials o dispositius físics d'entrada i sortida, el control i la supervisió a través d'Internet.



Taula 9: Trama Modbus TCP/IP

2.15. HTTP (Hypertext Transfer Protocol)

Definició: Protocol que ens permet realitzar una petició de dades o recursos. Dissenyat a principis dels 90, però ampliable amb el llarg dels anys, és transmet sobre el protocol TCP i funciona amb la capa d'aplicació del model OSI.

Es la base de qualsevol intercanvi de dades a la web o en línea. Amb una estructura client – servidor, on el que demana la petició és el navegador web. No només transfereix documents de text (*HTML*), si no que a part s'usa per transmetre imatges o vídeos, enviar continguts al servidor i actualitzar pàgines web.

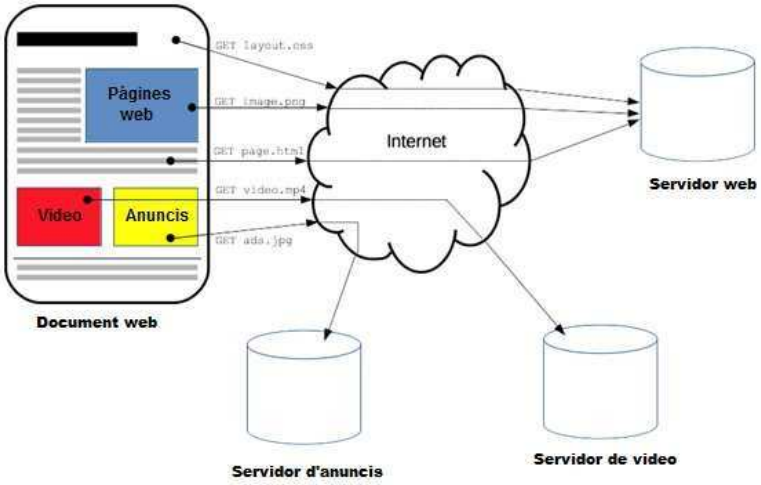


Figura 31: Esquema del funcionament protocol HTTP

Característiques:

- Simple: Pensat i desenvolupat per ser llegit i comprés fàcilment per l'usuari.
- Extensible: Noves versions HTTP/1.0, HTTP/2 i HTTPS, on les dades viatgen encriptades a diferència del protocol inicial.
- Sense estats però amb sessions: No guarda dades entre dos peticions al mateix temps, tot i que les famoses *cookies* permeten crear un context comú per cada sessió de comunicació.
- Connexions: No necessita que un altre protocol en la capa de transport mantingui les connexions continues entre els participants d'una comunicació, només necessita que l'altre protocol sigui fiable i no perdi els missatges, perquè pugui informar en cas de possibles errors.

2.16. Softwares

2.16.1. Node Red

És una eina de desenvolupament basada en el flux per a la programació visual, creada per l'empresa IBM amb la finalitat de facilitar la connexió de dispositius físics com PLC, hardware, *API* i serveis en línia com part del *IoT*.

Mostra visualment les relacions i funcions que permeten a l'usuari programar, sense necessitat d'escriure ningun llenguatge de programació.
És un editor de fluxos, on és poden afegir o eliminar nodes i connectar-los entre ells per mantenir una comunicació.
Existeixen dos tipus de nodes: d'injecció; produeixen un missatge sense necessitat d'entrada i transmeten el missatge al següent node connectat a ell, i de funció; tenen una entrada i realitzant treballs en aquest.

Des del 2013 és un programa de codi obert, és a dir, és poden modificar per a complir les necessitats especifiques de l'usuari.
Gràcies a l'organització *JS Foundation* és manté gratuït des del 2016.

2.16.2. InfluxDB

És una base de dades de sèries temporals de codi obert que guarda series de dades indexades al llarg del temps.
Desenvolupada per *InfluxData* a finals de l'any 2013.

Està programat e llenguatge compilat *"Go"*, semblant al llenguatge C, i optimitza l'emmagatzematge ràpid i d'alta recuperació de dades de sèries temporals en camps com el control de les operacions.

Gràcies al programa podem crear un servidor local o en línea, on guardar aquestes dades.

2.16.3. Grafana

És un software lliure que permet la visualització i la formació de dades mètriques com gràfiques i planells. Permet crear comandaments i gràfiques a partir de múltiples fonts, com InfluxDB o altres bases de dades de series temporals. Desenvolupat per l'empresa Torkel l'any 2014.

A partir d'una sèrie de dades recollides, obtenim un panorama gràfic de la situació en temps real de serveis motoritzats o infraestructures.

En aquest programa també de codi obert, podem crear un entorn de visualització mitjançant "*dashboards*" fent ús d'un ventall d'alternatives per personalitzar les dades obtingudes. Inclou la integració de notificacions d'alertes en temps real, així com l'aplicació de lògica en la informació llegida.

CAPÍTOL 3. DISSENY

3.1. Visió General

Amb la finalitat de poder monotoritzar qualsevol procés industrial que requereixi d'una línia de muntatge amb diferents estacions i que interactuïn entre elles, optem per una arquitectura que serveixi a nivell global per aconseguir-ho.
Mitjançant elements bàsics per a les comunicacions entre dispositius de la cel·la, softwares de flux, gestors i servidors de dades i programes per a la visualització.

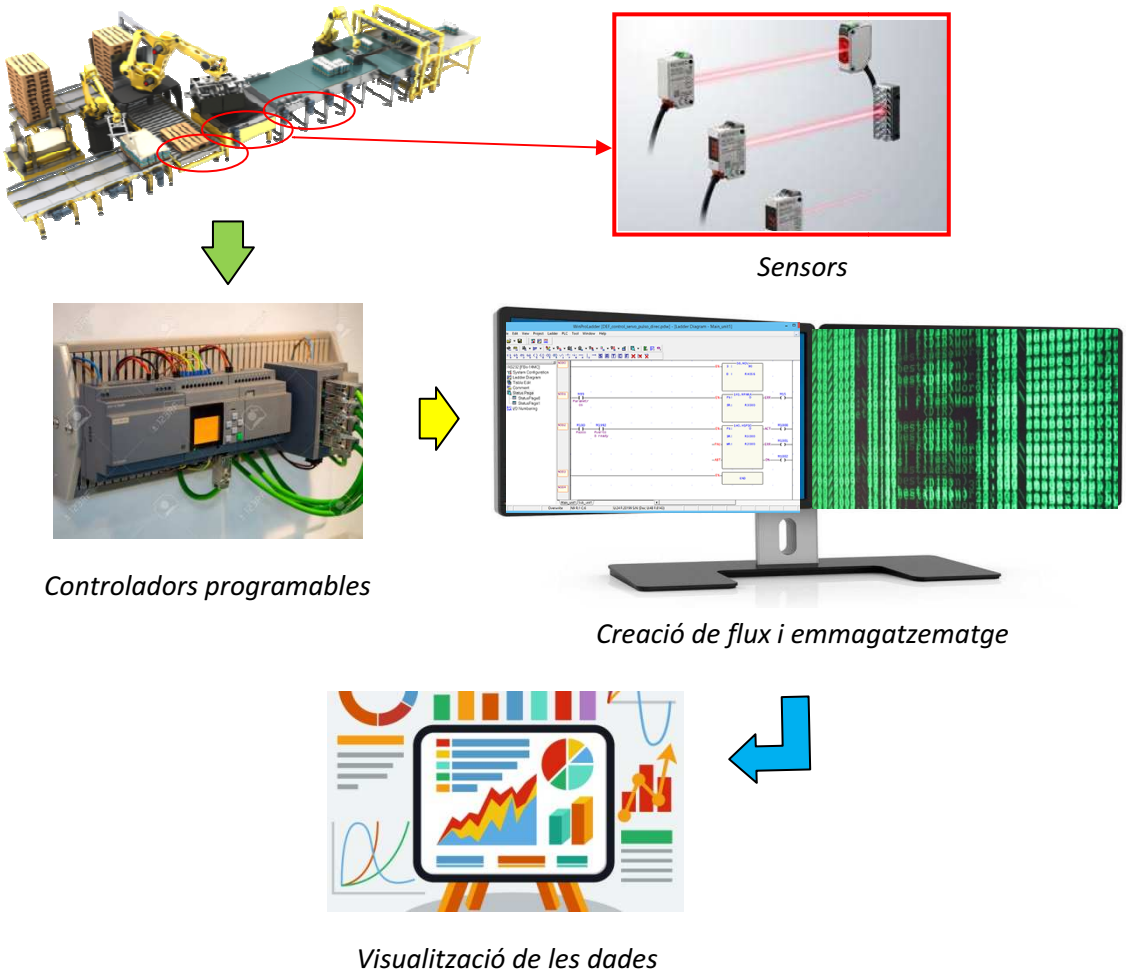


Diagrama 1: Transmissió de dades des de la cel·la industrial fins la visualització per ordinador

Per tal de validar el nostre projecte, ens centrarem a grans trets i d'una manera abstracta en la nostre cel·la d'estudi, tot i no haver-hi tingut accés físic en ella.

3.2. Cel·la Industrial Laboratori Schneider UPC

És una línia de muntatge que, en el nostre cas, ens permet transportar safates a través de corretges accionades per motors. Hi ha uns retenidors que s'utilitzen per parar les safates i unes plataformes per canviar la direcció del flux. Tots aquests elements estan connectats a uns PLCs que permeten gestionar les ordres perquè el flux de safates respongui adequadament durant tot el procés.

Figura 32: Cel·la Industrial Laboratori Schneider UPC

L'equip també compta amb un Pulmó que s'usa com a memòria per l'emmagatzematge temporal d'informació i que més tard és transferirà a altres unitats funcionals. Hi ha quatre tipus de comunicació en els bus de camp de la cel·la: *Profibus, CAN, Ethernet i ASI*, amb els protocols corresponents explicats en el capítol 2, tot i que l'últim no s'ha utilitzat en aquest projecte.

Las línees actives on s'han usat retenidors i plataformes són les de CAN, Profibus i Ethernet, en cap cas s'ha usat cap element de la línea ASI.

La seva distribució sobre la línia de muntatge és la següent:

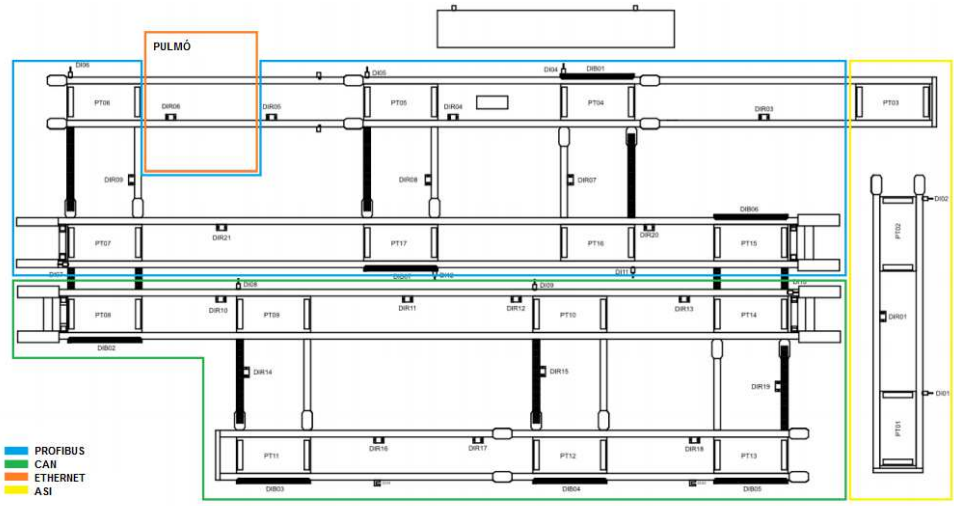


Figura 33: Distribució de Protocols en els bus de camp

Cada secció de la cel·la industrial utilitzada en aquest projecte està formada per diferents illes en la xarxa del bus. Aquesta distribució permet situar els perifèrics d'entrada i sortida a prop del sistema de control i connectar-lo amb el mínim de cables al PLC, utilitzant el protocol adequat en cada cas.
El PLC llegirà les dades d'entrada procedents dels perifèrics i les processarà.
Més tard enviarà el senyal de sortida a aquests perifèrics perquè activin els sensors d'actuació. Amb la distribució feta permetem un estalvi important en manteniment i muntatge degut a la disminució del cablejat.

3.2.1. Elements importants

Sensors

- Inductius: Només detecten objectes metàl·lics.
  Funcionen com un transformador, és a dir, alternen les corrents elèctriques. En la línea industrial existeixen tres tipus:

  1. *Basculants* – detecten l'arribada o sortida de la safata a la plataforma.
  2. *Totals* – al arribar a la plataforma detecten la peça metàl·lica que hi ha a un costat de la safata.
  3. *Amb retenidor* – detecten l'arribada de la safata gràcies a la peça metàl·lica que tenen sobre.

- Fotoelèctrics: És un dispositiu que detecta la presència d'objectes mitjançant llum, visible o no, i que en funció dels valors rebuts per aquesta llum, activa o desactiva el senyal. Aquests detectors tenen una alimentació de 12 i 24 volts en contínua, tot i que no han estat usats durant la simulació en línea.

- Capacitius: És un interruptor electrònic que treballa sense contacte. Aprofita l'efecte que tenen els metalls i no metalls per augmentar la capacitat del sensor en presència d'un camp elèctric.
  La funció més lògica que té és la interacció amb l'usuari en línea, com podria ser informar de la finalització d'un producte.

Plataformes

Són els elements situats enmig de les línies de la cel·la industrial. Permeten el flux correcte de safates en el trajecte. Treballem amb dos tipus de plataformes segons la quantitat d'estats o posicions que tenen.
El primer tipus està compost de dos estats: posició de repòs i pujada, mentre que el segon tipus conté tres estats: posició de repòs, pujada i baixada.
Com el nom indica, la de pujada permet el flux entre diferents altures.
La posició de baixada permet el flux sense retencions.
El funcionament és similar al d'un retenidor, durant el repòs la safata està quieta i és en el moment que s'acciona la baixada que aquest és mou.

Retenidors

Formats per sensors inductius que detecten la presència d'una safata i a través d'un cilindre d'efecte simple amb retorn permeten el pas o l'aturada de la safata. Estan accionats mitjançant electrovàlvules.

Electrovàlvules

Són dispositius dissenyats per controlar el flux d'un fluid. Les electrovàlvules accionen les plataformes o retenidors, denegen o permetent el pas d'aire comprimit, quant està prèviament programat en el PLC.

Completen la línea: polsadors, utilitzats com a *switch*, la frenada d'emergència i els propis motors de la cinta transportadora que permet el moviment.

3.3. Arquitectura

Amb la finalitat d'entendre el flux de dades, representarem el nostre cas en base als elements d'intervenció de la cel·la industrial, protocols de comunicació i els softwares de codi obert utilitzats en cada una de les parts del projecte.

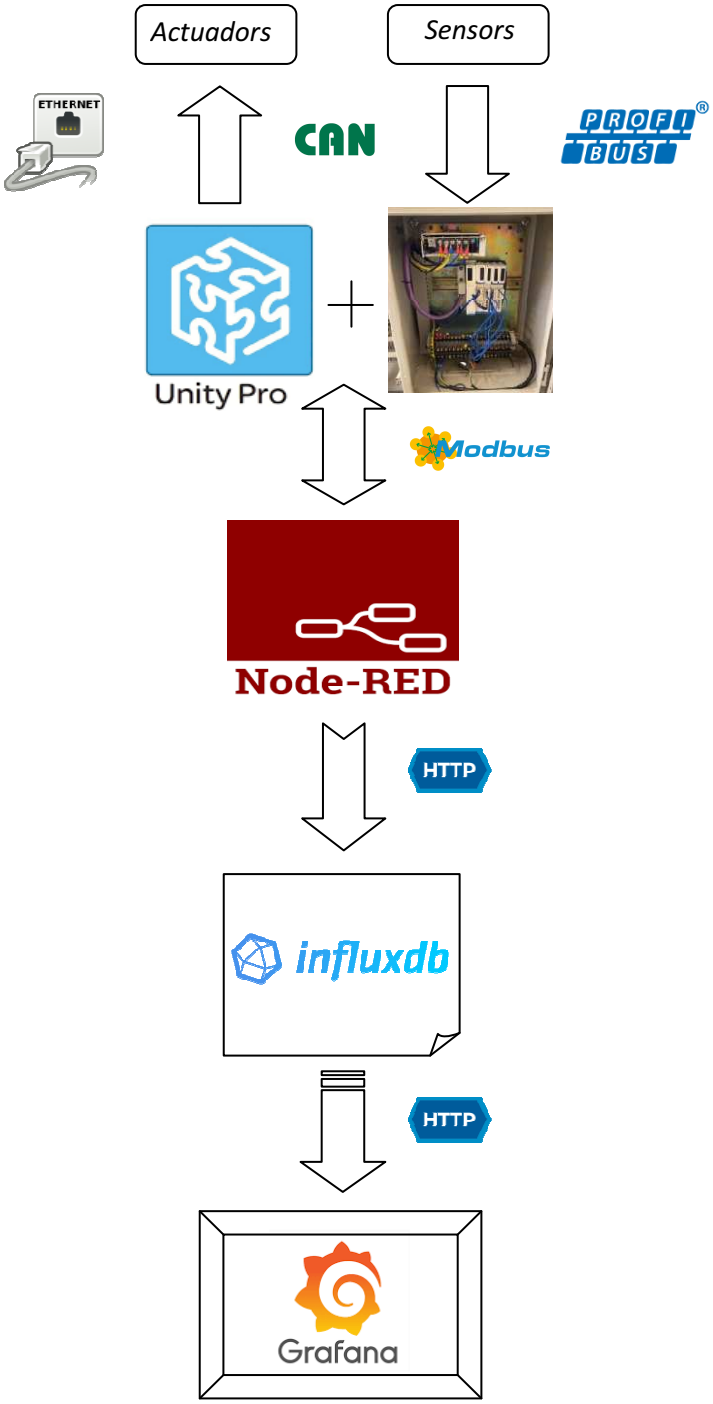
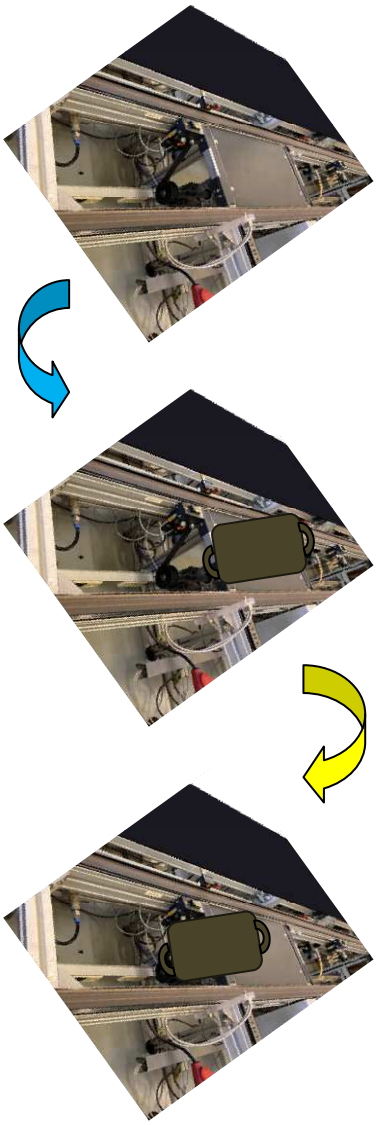


Diagrama 2: Flux de dades PLC – NodeRed- Influxdb - Grafana

3.4. Mecanització dels retenidors

El funcionament bàsic de la línea es basa en la programació realitzada amb el node *"function"* de l'esquema "*flow"* amb NodeRed. Els retenidors i les plataformes de la cel·la poden arribar a tenir 5 combinacions dels 3 estats possibles:

- Estat de repòs – S'activa quan no hi ha cap altre safata a la cinta.
- Estat d'avançament – La safata està circulant per la cinta fins l'estació següent.
- Estat de petició – L'activació d'aquest estat és produeix quan és detecta una safata sobre la cinta.



Combinació 1: Plataforma i retenidor en repòs.

numValue	strValue	Color
1	Repòs	Blau

Combinació 2: Plataforma i retenidor detecten la safata.

numValue	strValue	Color
2	Petició	Groc

Combinació 3: La safata és comença a moure gràcies a l'activació del retenidor i la plataforma.

numValue	strValue	Color
3	Avançament	Verd

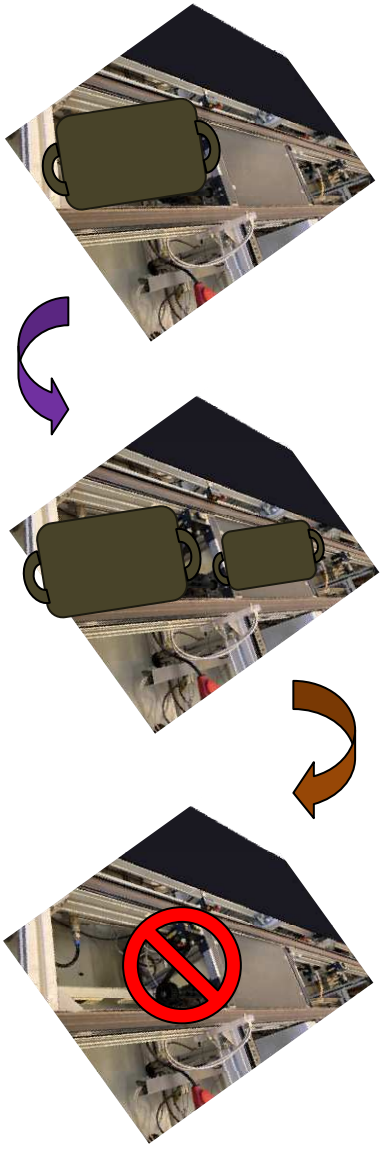


Diagrama 3: Combinacions d'estats dels retenidors

Combinació 4: La safata està en moviment i la plataforma i retenidor canvien a estat de repòs. És produeix quan el retenidor deixa de detectar la safata i està activat l'estat d'avançament.

numValue	strValue	Color
4	Repòs + Avançament	Lila

Combinació 5: La combinació dels altres estats permet l'entrada d'una altre safata a la línea. El sensor activa l'estat de petició quan és detecta la nova safata.
No és passa a l'estat d'avançament fins que la safata antiga no arribi a l'estació següent.

numValue	strValue	Color
5	Petició + Avançament	Marró

Combinació 6: Si la safata surt de la cinta, si els sensors fallen o qualsevol estat que no tingui relació amb cap de les combinacions anteriors, està programat perquè és doni un missatge d'error i es vegi des de on prové.

numValue	strValue	Color
6	Error	Vermell

Els colors referenciats estan assignats a Grafana i mitjançant un nou *"dashboard"* amb el *"plugin flowcharting" i el "plugin imagelt"* mostren l'últim registre d'algun dels 4 retenidors de proba.

Editat de forma fàcil mitjançant blocs i símbols per fer més entenedora la cel·la d'estudi.

CAPÍTOL 4. PROGRAMACIÓ

4.1. NodeRed

4.1.1. Descàrrega

Els tres programes s'executen des de qualsevol ordinador, ja sigui al laboratori Schneider de la UPC connectat mitjançant cable Ethernet als dispositius, o personal, que en el nostre cas i degut als problemes d'accessibilitat a la universitat és el que utilitzarem. Però comentar que podríem treballar remotament a distancia on la connexió és faria mitjançant *Wifi*, connectant-te a la xarxa local del laboratori a través del *router*.

1) *Comencem descarregant-nos a través d'Internet el paquet Node.js amb la versió v6.10.2 recomanada.*

Figura 34: Descàrrega Node.js per Windows

2) *Un cop descarregat necessitem obrir el terminal cmd de l'ordinador per poder integrar-lo. Escrivim l'ordre:*

```
npm install -g --unsafe-perm node-red
```

3) *Windows reconeixerà el software i posteriorment l'executarem mitjançant l'ordre:*

```
C:>node-red
```

Tardarà uns dos minuts en acabar de compilar el programa.

4) *L'accés a NodeRed és realitza a través d'una direcció IP i una pestanya del navegador d'Internet. Aquesta interfase utilitza el llenguatge JavaScript i on les dades estan codificades en format obert.*

5) *Al final de l'execució de l'ordre, el terminal cmd ens donarà la direcció IP que haurem d'introduir en el navegador d'Internet. Normalment si utilitzem el mateix ordinador amb el que executem Node Red accedim utilitzant la direcció URL:*

< http://127.0.0.1:1880 o http://localhost:1880 >

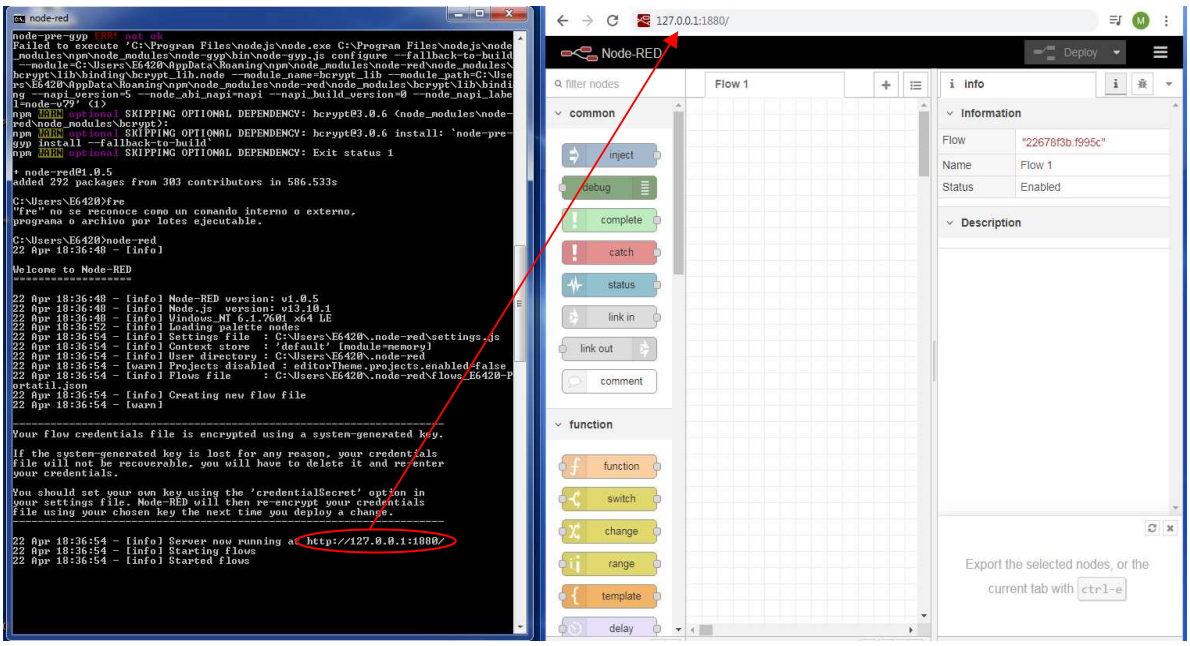


Figura 35: Execució Node Red a través del servidor cmd i la direcció IP

4.1.2. Biblioteca

Dins Node Red tenim l'escriptori de nodes a la part esquerra i a la dreta un desplegable amb les característiques que li vulguem donar a les variables.

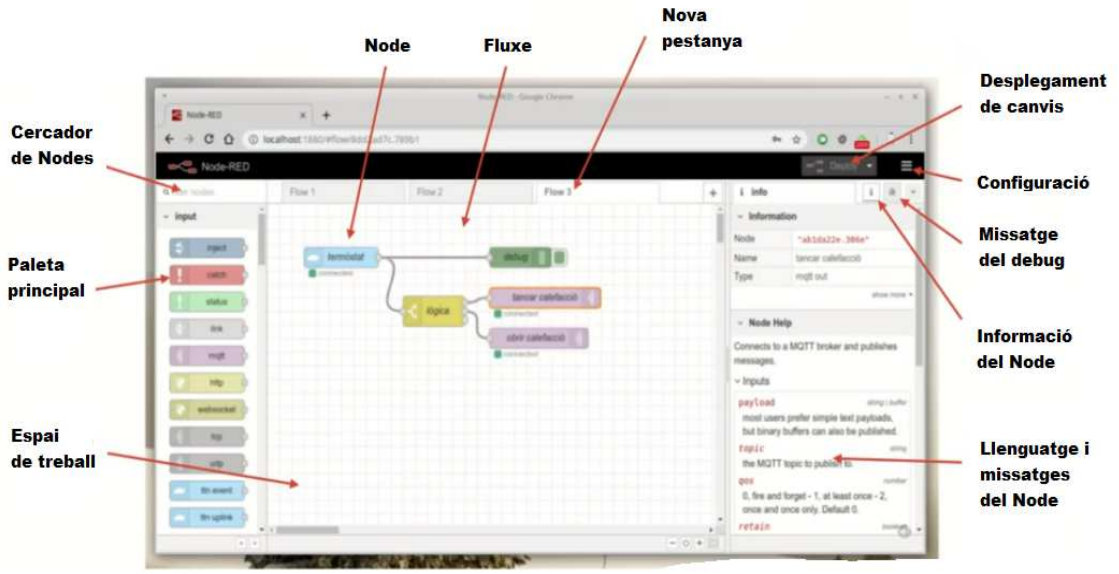
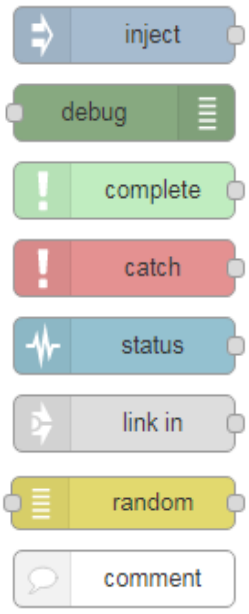


Figura 36: Escriptori Node Red amb les funcionalitats bàsiques

A continuació una breu descripció dels nodes més comuns de la paleta inicial:



Injecta un missatge en un flux manualment o a intervals regulars.

Mostra les propietats del missatge de manera comprensiva.

Activa un flux quan l'altre node finalitza el tracta d'un missatge.

Captura els errors llançats pels nodes a la mateixa pestanya.

Informa dels missatges d'estat d'altres nodes.

Creen cables virtuals entre fluxos.

Genera un valor aleatori entre un valor baix i alt.

Serveix per afegir comentaris a altres nodes o fluxos.

Figura 37: Paleta de nodes comuns

El programa està ple de nodes amb múltiples funcionalitats, estan tots molt ben detallats per tant ens enfoquem amb els que realment hem utilitzat en el projecte:

Figura 38: Node Function

Function:

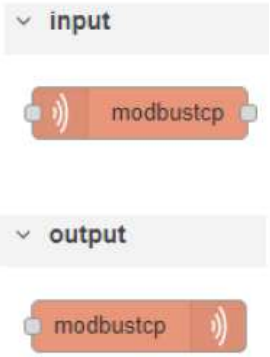
Permet programar un node mitjançant Javascript.

Els missatges es passen com a objecte anomenat msg.

Hem hagut de descarregar nodes que no venien implementats en el software inicial. Pel node Modbus-TCP introduïm al servidor cmd el següent comandament:

```
npm install node-red-contrib-modbustcp
```

Reiniciant el NodeRed, i tornarem a entrar al navegador amb la direcció URL.



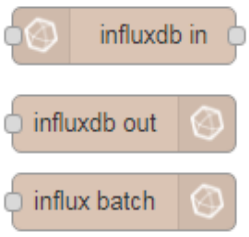
Modbus-TCP:

Accedeix als registres del PLC amb entrada o sortida del Modbus. Diferencia entre bits i texts i retorna les dades en forma de matriu. Hauríem utilitzat un codi de funció que ens permeti llegir o escriure posicions de memòria %MW.

Figura 39: Nodes Modbus -TCP

El mateix passa amb el node InfluxDB, on introduirem al servidor cmd l'ordre:

```
npm install node-red-contrib-influxdb
```



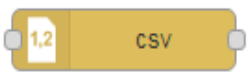
Influxdb:

Permet fer consultes i emmagatzemar dades temporals a la base de dades d'Influxdb.

Consta de tres nodes, dos de lectura i el darrer per escriure. Quan expliquem el desenvolupament ja ens centrarem en les propietats i la seva configuració.

Figura 40: Nodes Influxdb

Degut a les circumstàncies actuals produïdes per la pandèmia del Covid-19 i al no poder recollir les dades dels PLCs del laboratori de la UPC, hem optat per simular l'estat dels retenidors amb nodes csv/txt/alasql.



Csv:

Converteix en matriu la combinació entre una cadena de dades .csv i la seva representació en JavaScript en qualsevol direcció .

Figura 41: Node Csv

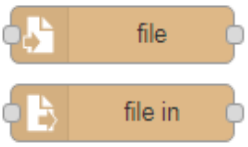


Figura 42: Nodes File

File:

Escriu el debug (msg.payload) en un arxiu, ja sigui agregant o reemplaçant el contingut al final. També llegeix un arxiu com a cadena binaria o *buffer*.

Pel node xcl, que equival a una taula Excel del paquet office, escrivim l'ordre:

```
>npm install node-red-contrib-alasql_
```

.

Figura 43: Node Alasql

Alasql:

Aquest node converteix l'entrada del debug (msg.payload) en un format Excel i ho escriu en un arxiu local.

4.1.3. Creació de fluxos

Mitjançant exemples s'assimilarà el funcionament per arribar a l'objectiu final.

Ex. 1: Per entendre el comportament de com els nodes transmeten un missatge, realitzem un primer exemple senzill amb l'ajuda de tres *"injects"* a l'entrada: el primer ens descriu data i hora en que s'insereix, és a dir, de tipus "*timestamp*". El segon és de tipus "*number*", mentre que el tercer és tipus "*boolean"* verdader o fals. Tots tres missatges seran visualitzats en el *"debug"* de sortida cada X interval de temps.

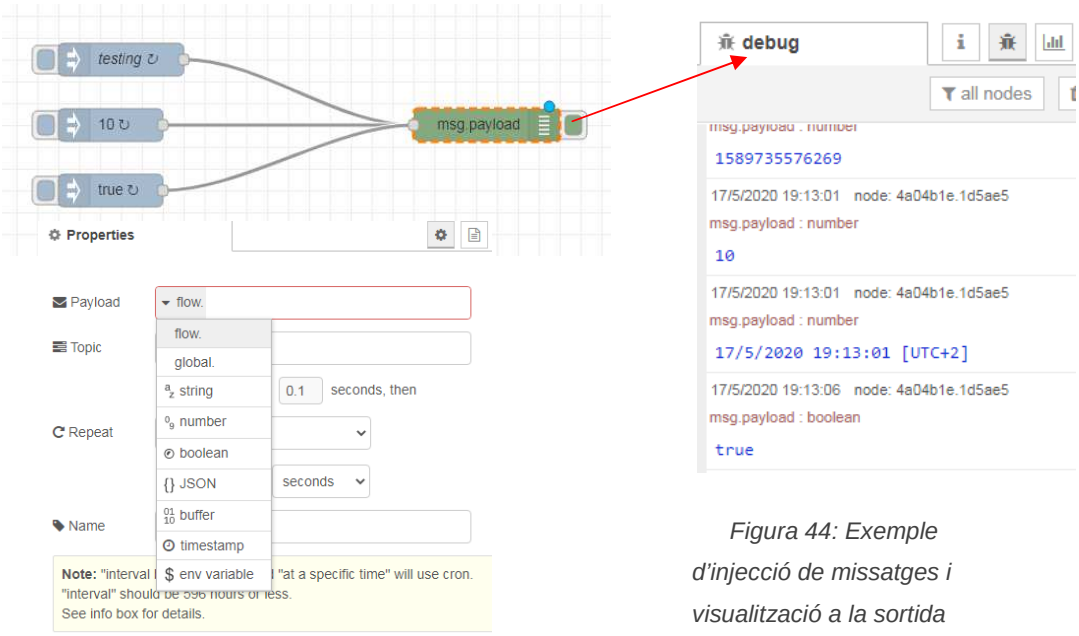


Figura 44: Exemple d'injecció de missatges i visualització a la sortida

Ex. 2: Un segon exemple ha estat la visualització dels missatges a una pàgina d'Internet. Ens permet introduir un missatge a través del node "*alasql*" que permet convertir un arxiu local a codi *Javascript* perquè posteriorment és visualitzi en una nova pestanya.

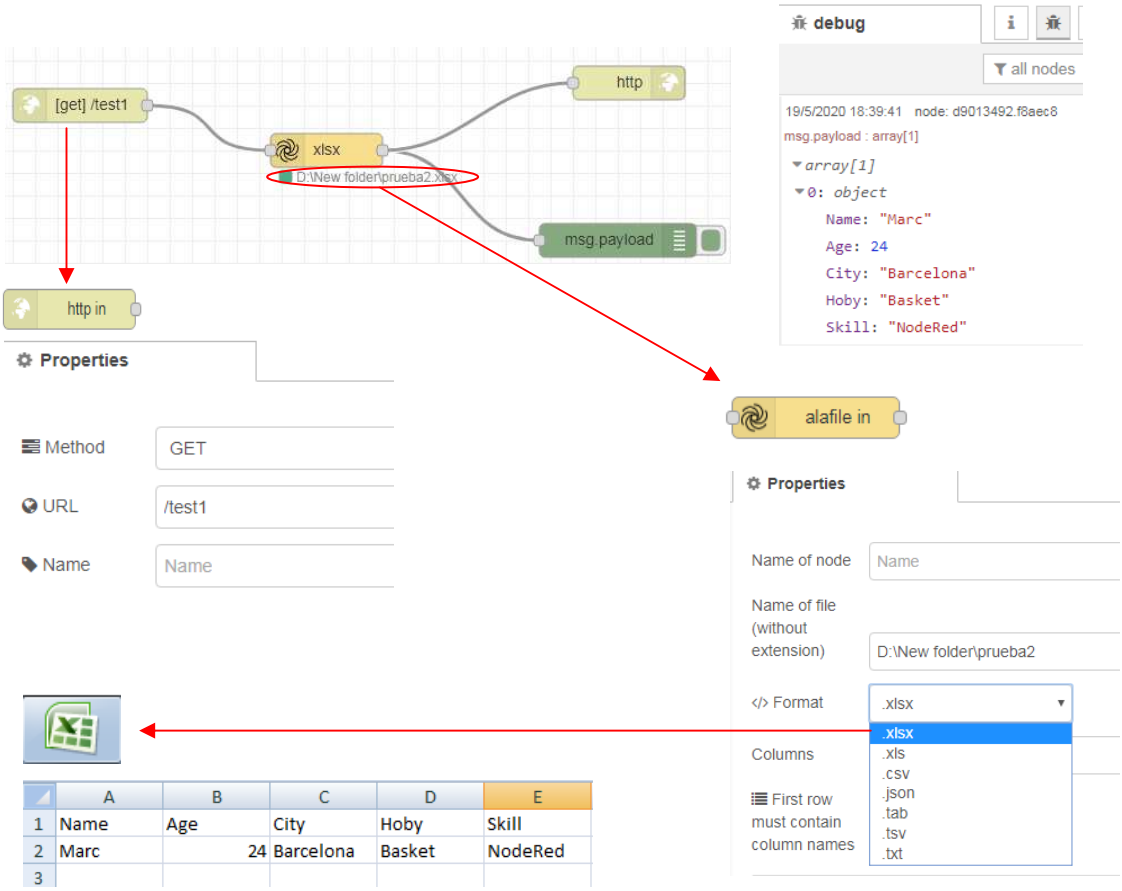


Figura 45: Exemple de conversió d'arxius locals a format Javascript

Amb l'ajuda d'aquest node de sortida, ens quedarà emmagatzemada la informació a aquesta nova direcció URL.

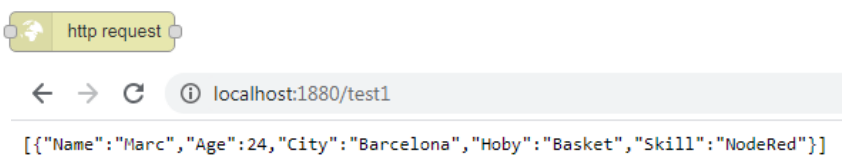


Figura 46: Node Http request i visualització a través d'una direcció URL

Ex. 3: El software també ens permet crear arxius .csv o .xcl, tan sols ens cal un node *"function"* i un *"file"* a la sortida que ens generarà un arxiu que després trobarem normalment al nostre disc local D.

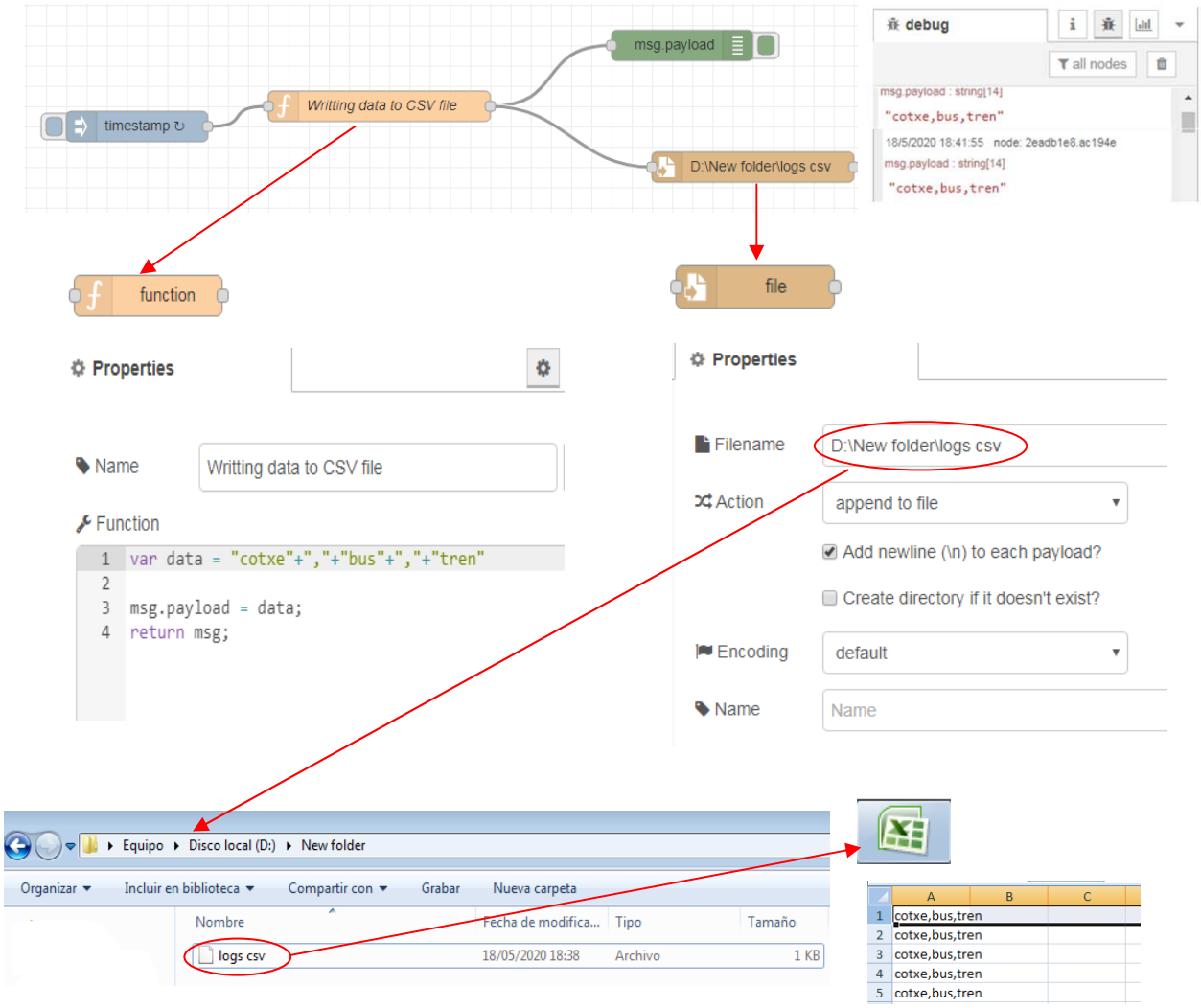


Figura 47: Exemple creació d'arxius i descàrrega al propi disc local

Ex. 4: Per introduir el gestor de dades Influx, afegim el node corresponent perquè s'emmagatzemin els valors i poder-los visualitzar posteriorment.

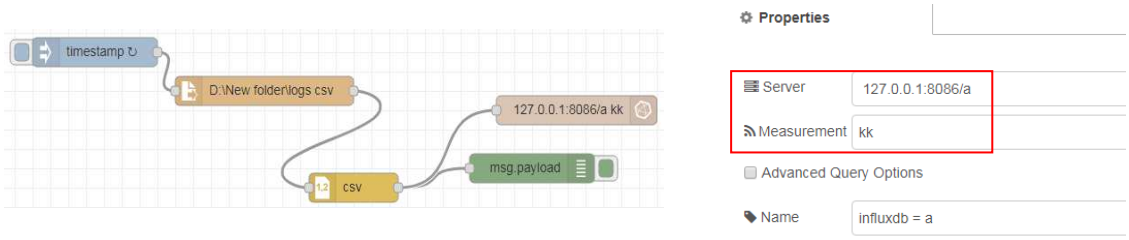


Figura 48: Configuració i propietats del node InfluxDB

4.2. InfluxDB

4.2.1. Descàrrega

InfluxDB s'executa des de la terminal del propi programa. Com hem explicat al capítol 2, el software emmagatzema infinites dades seqüencials que processa i permet consultar a temps real. Passos a seguir per a la correcta instal·lació:

1) Descàrrega a través de la pàgina oficial d'Influxdb el següent document comprimit: `https://dl.influxdata.com/influxdb/releases/influxdb-1.8.0_windows_amd64.zip`

És el començament per a la creació d'una base de dades.

2) En el .zip tenim 5 fichers, hem d'obrir el que porta per nom: Influxd. S'executaran totes les credencials que té i automàticament ja ho tindrem instal·lat al nostre ordinador.

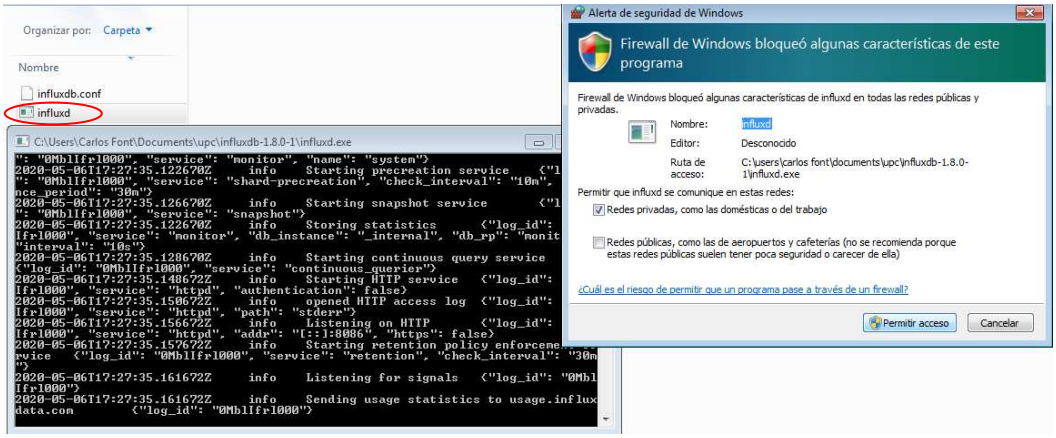


Figura 49: Instal·lació base de dades Influx

3) Per començar a crear hem d'obrir el influx.exe, sense tancar el .influxd.

4.2.2. Comandaments

· **Database**: És el contenidor lògic que conté les series temporals.

`> create database a` **:** per crear una base de dades.

· **Measurement:** És l'estructura amb la qual s'emmagatzemen les dades.

· **Use**

```
> use a
Using database a
```

: per indicar la base de dades que s'utilitza.

· **Timestamp:** És la data i hora en la que les dades es guarden a InfluxDB.

· **Select * from**

```
> select * from a
```

: permet seleccionar i visualitzar el *measurement* de la base de dades seleccionada.

· **Field:** Són els valors emmagatzemats i associats a un *timestamp.*

· **Drop measurement**

```
> drop measurement a
```

: elimina totes les dades del *measurement.*

4.2.3. Implementació

Partint de l'últim exemple de Node Red, on havíem creat un "*flow*" per fer arribar la informació que injectàvem a través del node d'InfluxDB, el primer pas serà crear una nova *"database"* que vincularem al nostre node i s'usarà.

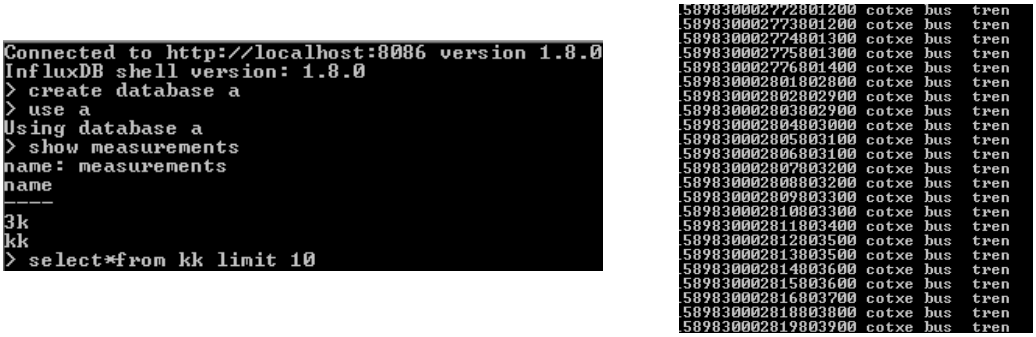


Figura 50: Creació d'una database i execució del servidor Influx

Per defecte el servidor d'Influx ve predestinat en la direcció: < *127.0.0.1:8086* >, a continuació tants sols em d'escriure el nom de la *"database"* creada.
Un cop seleccionat el *"Deploy"* a Node Red, el programa començarà a funcionar sol.
Al gestor Influx escrivim l'ordre: **select*from** + els measurements que haguem seleccionat, i ens visualitzarà tots els valors registrats durant aquest temps.

Ex. 5: Una altre aplicació ha estat emmagatzemar al gestor un document .xcl, on al *"debug"* és formarà una matriu amb tantes columnes com n'hi hagi al Excel.

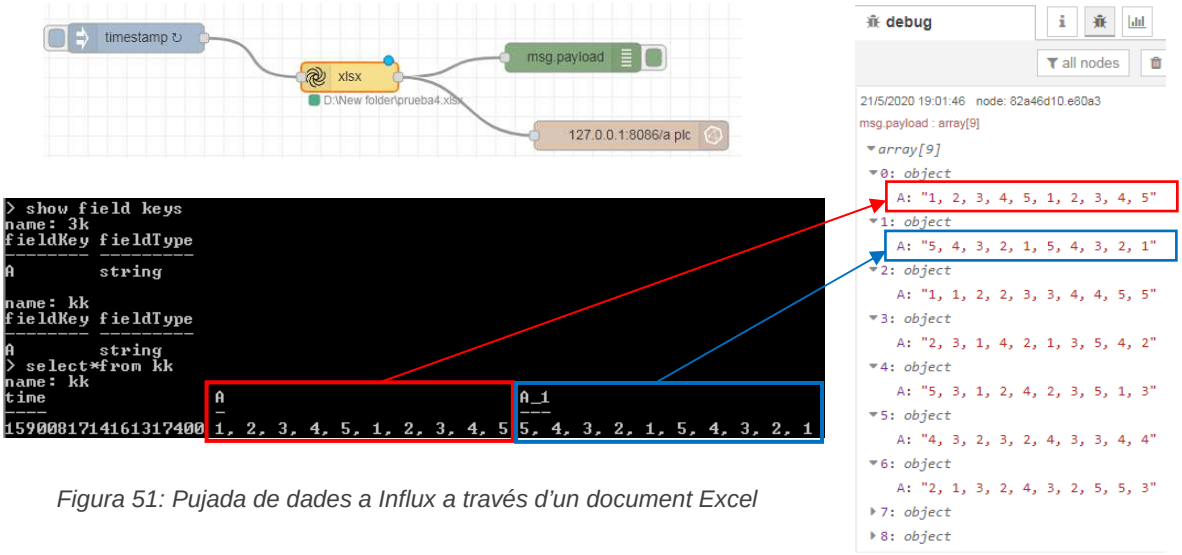


Figura 51: Pujada de dades a Influx a través d'un document Excel

Amb la instrucció: **show field keys,** el programa ens informa de com estan emmagatzemades les dades al servidor, és a dir, el tipus de dada que és i l'hora assignada a aquesta.

Ex. 6: Afegim un sensor i li donem forma amb una funció.

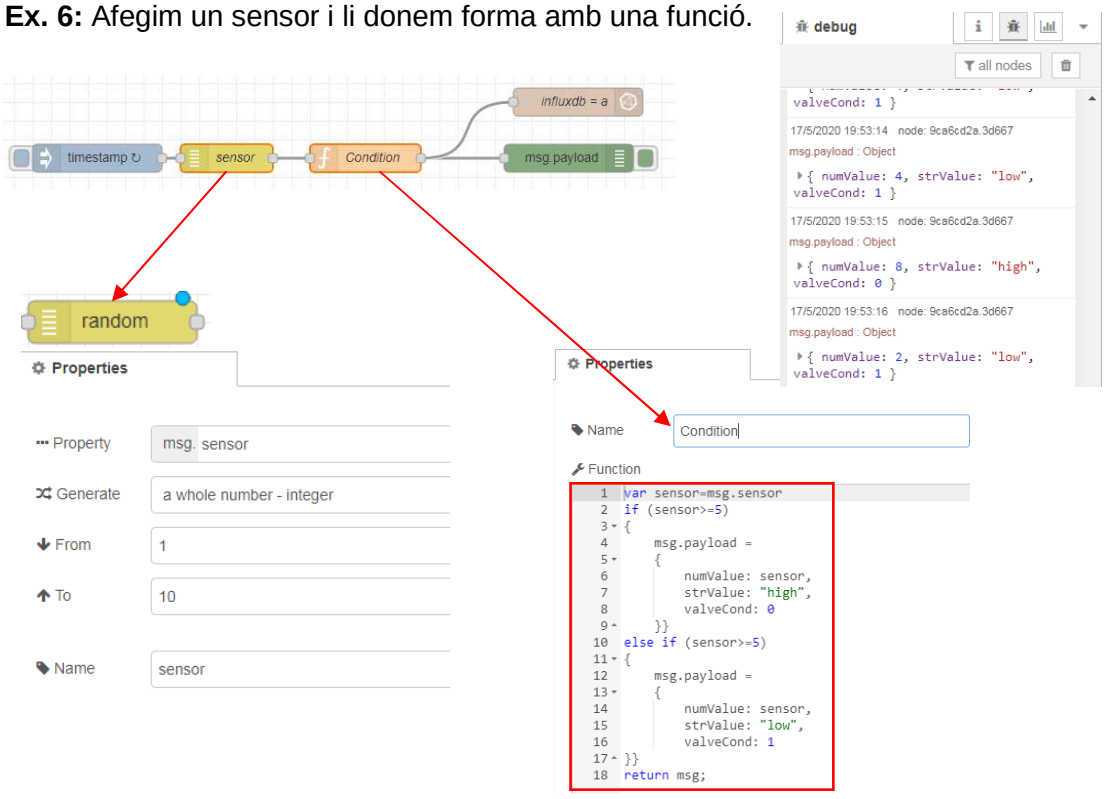


Figura 52: Exemple amb nodes de funció i sensor per posterior pujada a Influx

En aquest nou *"flow"* introduïm la variable del sensor que serà el missatge que és veurà al *"debug".* Amb la funció fem que a part de variables en format número també n'hi hagi en format paraula, *"float and string".* Seguint el mateix procediment que en l'altre exemple i configurant el node Influx de la mateixa manera, obtenim el següent:

```
> show field keys
name: kk
fieldKey   fieldType
--------   ---------
numValue   float
strValue   string
value      float
valveCond  float
```

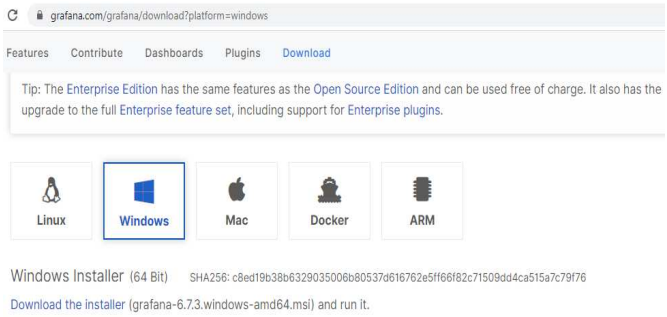
```
> select*from kk
name: kk
time                numValue strValue valveCond
----                -------- -------- ---------
1590340009871397400 1        low      1
1590340010791450000 1        low      1
1590340011792507300 5        high     0
1590340012267534400 10       high     0
1590340012792564500 2        low      1
1590340013795621800 3        low      1
1590340014799679300 5        high     0
1590340015794736200 4        low      1
1590340016796793500 9        high     0
1590340017793850500 8        high     0
1590340018794907800 4        low      1
1590340019795965000 9        high     0
1590340020800022500 9        high     0
1590340021801079700 1        low      1
1590340022803137000 9        high     0
```

Figura 53: Emmagatzematge de les dades en diferents tipologies de valors

4.3. Grafana

4.3.1. Descàrrega

A diferència dels altres dos softwares, Grafana és descàrrega de manera més senzilla. Permet visualitzar tota la informació emmagatzemada en la base de dades d'InfluxDB. Ofereix milers de "*dashboards/plugins*" i també permet crear-ne de nous.



1) Entrem a Internet, busquem la pàgina oficial de Grafana i ens instal·lem el software que sigui compatible amb el nostre ordinador.

Figura 54: Descàrrega Grafana per a Windows

2) A la mateix pàgina ens indiquen que al ser un programa de codi obert, necessitem l'URL del navegador que ens farà d'interfase amb el programa. En aquest cas és:

< http://localhost:3000 >

3) Per defecte, la primera vegada et fa entrar amb un usuari i password determinat (la paraula "admin" en ambdues caselles).

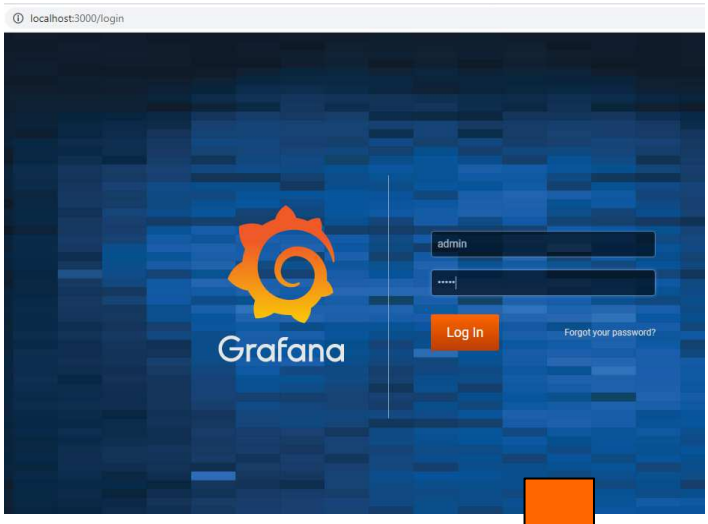


Figura 55: Portal executor Grafana

4) Un cop registrat, has de canviar la contrasenya, i ja podrem accedir-hi sempre, sense necessitat d'executar-ho al servidor cmd.

Figura 56: Pantalla per canviar contrasenya Grafana

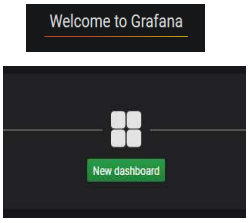
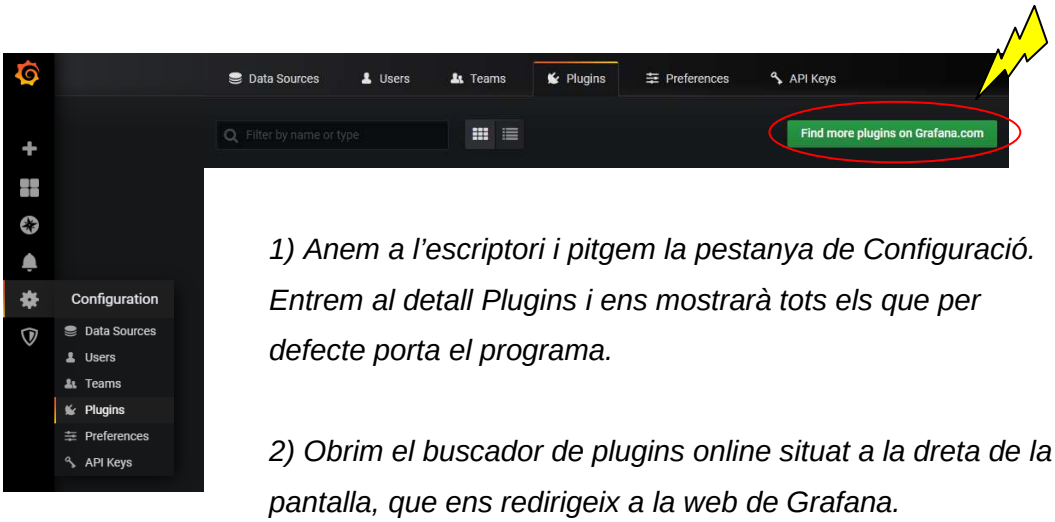


Figura 57: Perfils "dashboard"

5) Per començar amb el programa, és indispensable crear un nou "dashboard" on crearem l'entorn que volem visualitzar i afegirem una font de dades des d' on importarem les dades.

4.3.2. Plugins

Són la manera en que volem visualitzar les dades, ja sigui en gràfics, comptadors, diagrames, fluxos,etc. Per tal de representar el cas concret de la cel·la industrial hem hagut d'instal·lar-ne de nous, perquè implantessin el que volíem transmetre.



1) Anem a l'escriptori i pitgem la pestanya de Configuració. Entrem al detall Plugins i ens mostrarà tots els que per defecte porta el programa.

2) Obrim el buscador de plugins online situat a la dreta de la pantalla, que ens redirigeix a la web de Grafana.

Figura 58: Escriptori Grafana

3) *Busquem els que ens interessen, en aquest cas: On la pestanya Instal·lació i seguim les indicacions.*

Figura 59: Plugin ImageIt i Flowcharting

4) Hi ha dues maneres de descarregar-ho, la primera i més fàcil és introduint les següents ordres al servidor cmd:

```
>cd "C:\Program Files\GrafanaLabs\grafana\bin"
```

Sempre ve donat per aquest link, per tindre el programa localitzat.

```
C:\Program Files\GrafanaLabs\grafana\bin>grafana-cli plugins install pierosavi-i
mageit-panel
```

Aquesta segona ordre ens instal·larà correctament el plugin escollit, però perque s'actualitzi al software Grafana cal que reiniciem l'ordinador i un cop tornem a iniciar el < portalhost:3000 >, ja tindrem els plugins agregats.

5) La segona manera d'agregar plugins, és des del mateix escriptori de Windows. Surt l'adreça on trobarem el programa que un cop baixat el .zip, descomprimit i copiat també haurem de reiniciar l'ordinador i tornar a executar l' URL per Grafana.

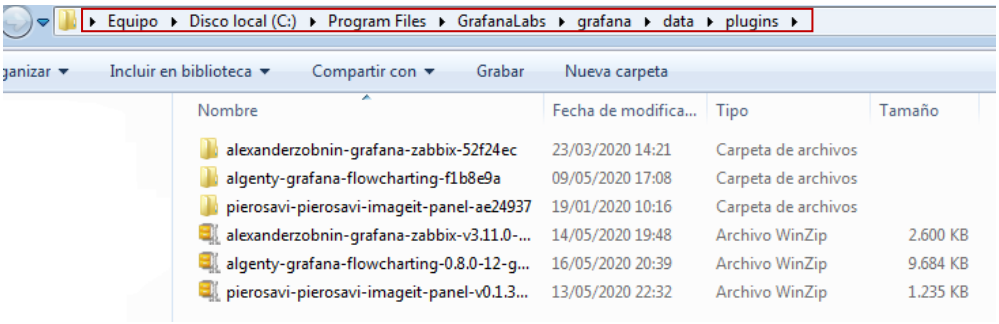


Figura 60: Disc local amb les carpetes corresponents on s'emmagatzemen els plugins

4.3.3. Data Source

La base de qualsevol visualització amb Grafana és fa a través del import d'una font de dades, que en el nostre cas estan emmagatzemades al gestor InfluxDB.

Figura 61: Configuració font de dades Influx i posterior visualització amb Grafana

Per defecte la URL és la que ens dona el propi programa i l'únic camp que hem d'omplir és el nom que li haguem donat a la nostre *database.* Així doncs seguint l'exemple 6 i visualitzant els valors *"float"* del flux, obtenim la gràfica de la figura 61.

El software ens brinda multitud d'opcions alhora de triar gràfics, com més ens convingui plasmar les dades i visualitzar tants gràfics com volguem a la mateixa plantilla. A continuació diferentes opcions:

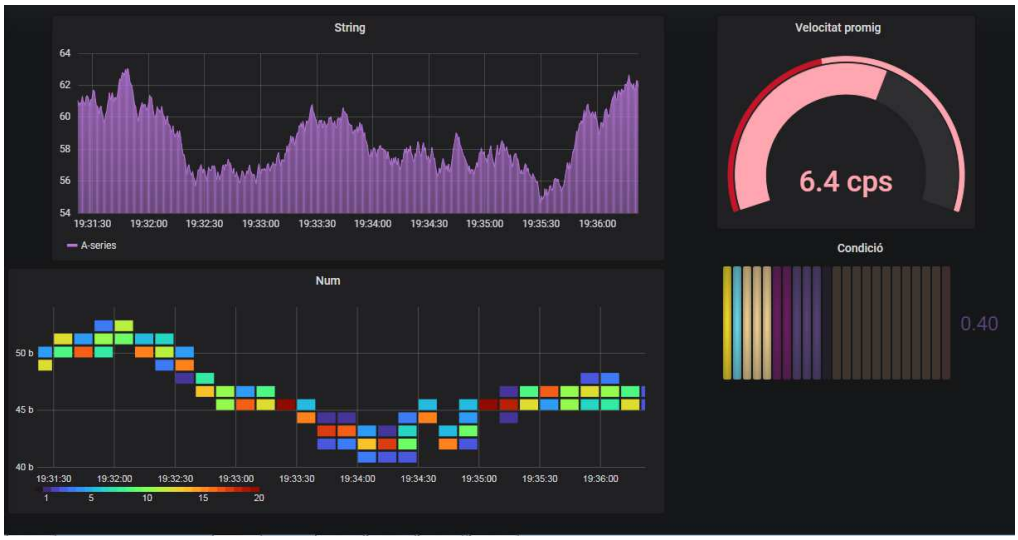
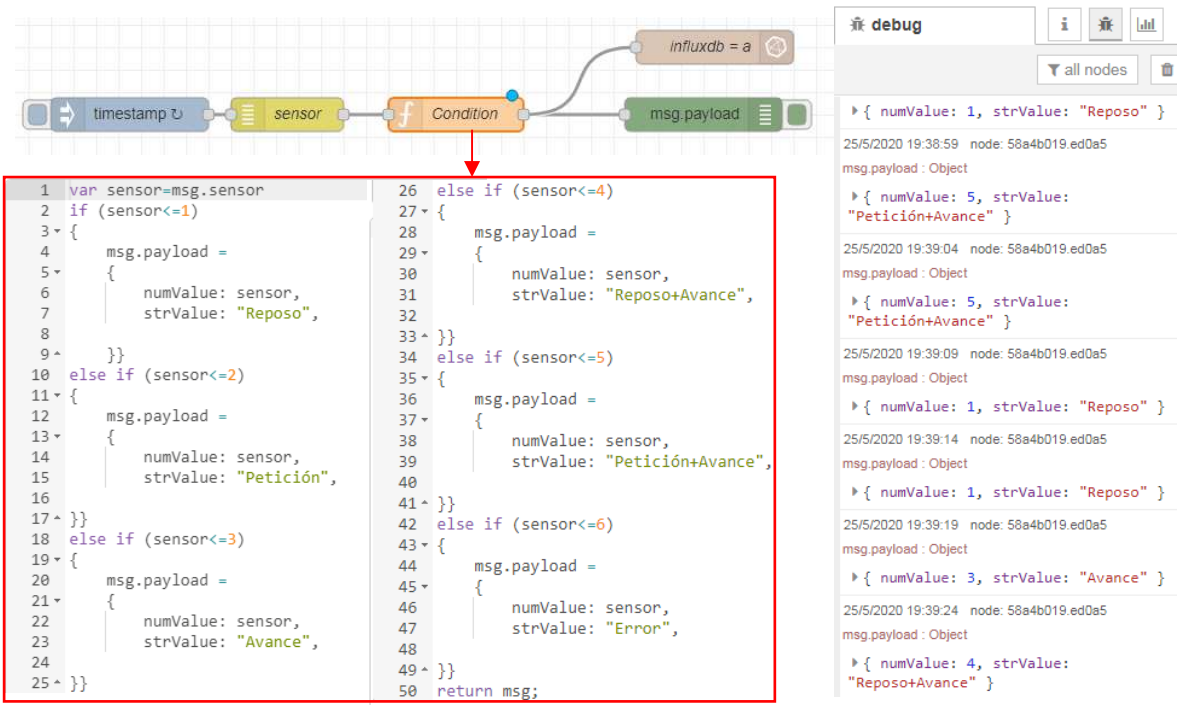


Figura 62: Exemple diferents gràfics representant variables emmagatzemades a la font de dades

Ex. 7: Amb la finalitat de poder monitoritzar la cel·la del laboratori i marcant-nos els sensors com a elements clau, arribem a crear un flux que ens permet conèixer l'estat d'aquests en tot moment i representar-lo, aconseguint entendre com funciona.




```
var sensor=msg.sensor
if (sensor<=1)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Reposo",

    }}
else if (sensor<=2)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Petición",

}}
else if (sensor<=3)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Avance",

}}
else if (sensor<=4)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Reposo+Avance",

}}
else if (sensor<=5)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Petición+Avance",

}}
else if (sensor<=6)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Error",

}}
return msg;
```

Figura 63: Flux per saber l'estat del sensor

Amb el servidor d'Influx emmagatzemem aquestes variables d'estat. Seran aleatòries ja que venen donades pel node *"inject"* cada un cert instant de temps.

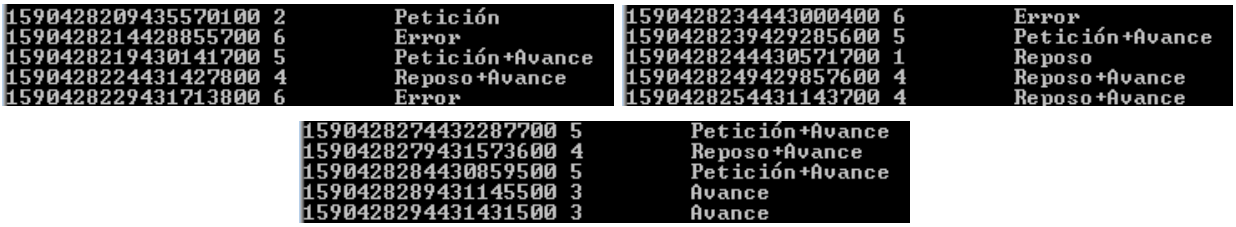


Figura 64: Emmagatzematge de les dades amb diferents tipus d'estat

Finalment visualitzem les dades que ens interessant i ens ajudaran a veure com és el comportament del sensor.

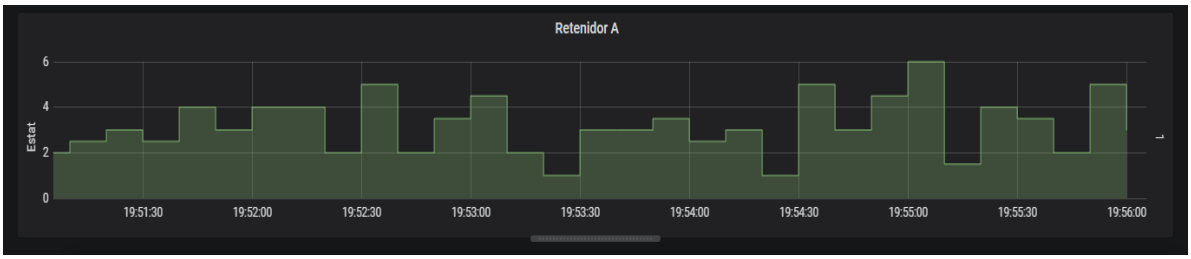


Figura 65: Visualització dels diferents estats en un gràfic

Tot i que un sol no te gaire rellevància i el tipus de gràfic no ens dona massa joc, per això en l'apartat de validació implementarem quatre retenidors situats en diferents punts de la cel·la, per obtenir una visió més amplia i complexa del correcte funcionament.

CAPÍTOL 5. VALIDACIÓ

Aquest apartat del projecte consta de diferents assajos realitzats durant la programació de l'automatització i monitorització de la cel·la industrial, amb l'objectiu de validar el disseny realitzat. Els assajos és divideixen en dues categories: funcionals i operatives. Les proves funcionals és realitzen per validar diferents parts del sistema a nivell de configuració, mentre que les proves operatives és refereixen a la comprovació del correcte funcionament de tot el conjunt. Avaluem els exemples proposats en el capítol 4, així com l'esquema final de l'automatització del projecte. Pels dos assajos hem elaborat una sèrie de pautes per la correcta identificació dels procediments emprats.

i. Descripció de la prova: que és farà i per a què?
ii. Resultats esperats: en que ens hem fixat per decidir si el funcionament és correcte?
iii. Resultats obtinguts: Obtenim el que esperàvem?
iv. Modificacions: en el cas de no obtenir l'espera't o el no funcionament.

5.1. Proves funcionals

Prova 1

i. El primer exemple ha estat l'enviament de dades mitjançant nodes *"inject"* amb Node Red, configurant-los amb diferents tipus de variables i la seva visualització al *"debug".*

ii. S'espera que retorni els valors escrits en els nodes *"inject"* d'entrada.

iii. Efectivament visualitzem els mateixos resultats sense modificar-ne l'ordre.

Prova 2

i. El segon exemple ha estat la visualització d'un arxiu local en una nova pàgina de sortida, mitjançant nodes *"alasql"* i http tant d'entrada com sortida.

ii. S'espera que és visualitzin els valors escrits en l'arxiu local convertit a llenguatge Javascript en una nova direcció URL.

iii. És visualitza l'arxiu però el format podria ser millor si eliminéssim les cometes.

Prova 3

i. En el tercer exemple hem creat i descarregat un arxiu Excel a través dels nodes *"file"* i *"function"* configurant-los pel correcte desenvolupament en el software.

ii. S'esperava que és generés un arxiu en la direcció i carpeta seleccionada.

iii. L'arxiu s'ha creat, però per defecte ve amb unes propietats desconegudes, i hem tingut que obrir-lo a través del propi Excel.

Prova 4

i. El quart exemple consta d'un arxiu creat que volem emmagatzemar-lo al gestor de dades InfluxDB a través del node *"influxbatch".*

ii. S'esperava visualitzar els valors dintre la base de dades.

iii. En el node *"debug"* es rep la informació correctament però en el node d'Influx surt un missatge d'error, degut a que no tenim la base de dades creada, ni els dos servidors oberts simultàniament.

iv. Hem hagut d'obrir els dos servidors, tant Influxdb com el propi gestor Influx on donem les ordres i mostrar-ne la *"database"* que estem usant, perquè el programa Node Red ho detecti i sàpiga on enviar-ho.

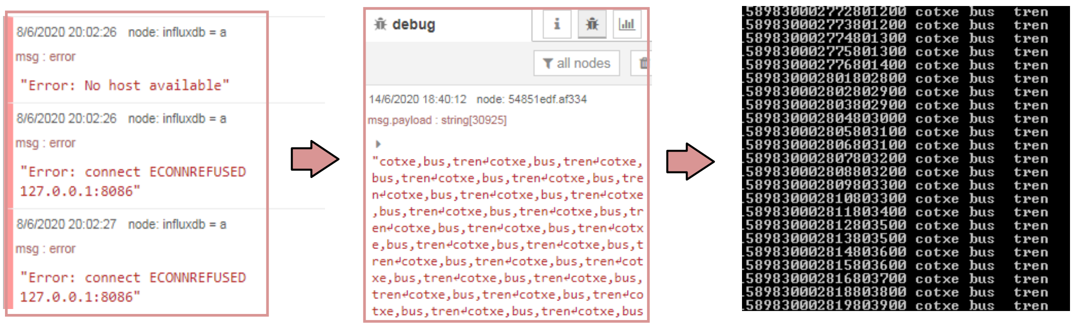


Figura 66: Solucionant problemàtica amb l'enviament de dades al servidor

Prova 5

i. En un cinquè exemple hem emmagatzemat els valors *"string and float"* d'un arxiu Excel al gestor Influx.

ii. S'esperava visualitzar els valors dintre la base de dades.

iii. La visualització no ha estat com desitjàvem, ja que el format no és l'idoni. Es repetien les dades de la primera fila en tots els registres degut a una mala estructura del document.

iv. Hem buscat dins la guia del programa Node Red per veure com s'emmagatzema la informació donada en files, columnes i cel·les del Excel.

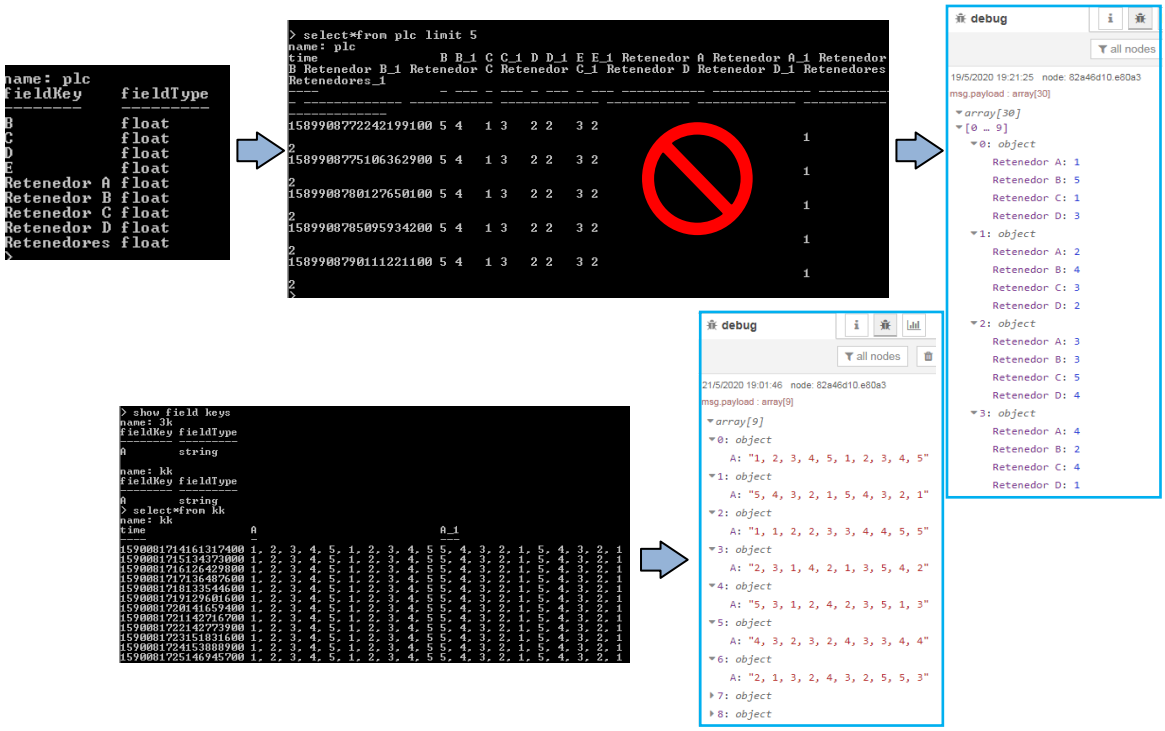


Figura 67: Solucionant problemàtica amb el format d'arxius que enviem a Influx

Prova 6

i. El sisè exemple ha estat introductori pel software Grafana. Hem simulat un procés en el qual el sensor rebia el missatge, sortia pel node *"debug"*, s'emmagatzemava a Influx i posteriorment el visualitzàvem al software.

ii. S'esperava que es completes el curs i obtinguéssim els gràfics de prova.

iii. Al ser la primera vegada, no teníem programat de quina font de dades s'havia d'importar la informació ni de quina manera es configuraven els gràfics. Així que en el resultat ens donava un missatge a Grafana de *"NO DATA"*.

iv. Amb l'ajuda de tutorials a la pagina oficial hem pogut configurar-ho.

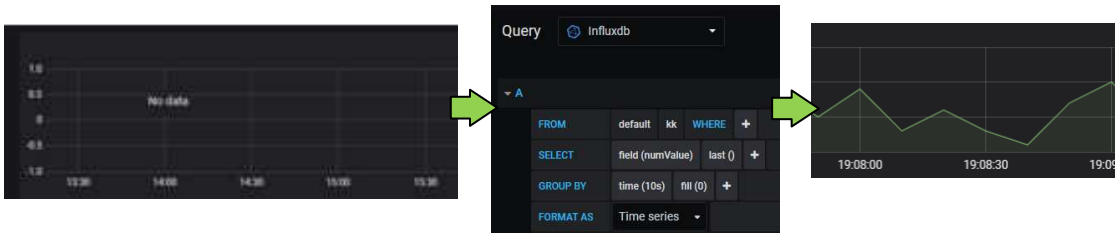


Figura 68: Solucionant problemàtica amb la visualització de dades a Grafana

Prova 7

i. L'últim exemple, ja encaminat en el procés de monitorització que busquem, hem creat un flux que ens permet saber l'estat del sensor en cada moment, s'introdueix, s'emmagatzema i finalment es visualitza.

ii. S'esperava que es completes el curs i obtinguéssim els gràfics de prova.

iii. El primer resultat ha estat un desastre ja que hi havia errors per tot arreu.

iv. L'important en aquesta darrera prova era implementar tots els coneixements adquirits per simplificar-ne el funcionament. El més complex era buscar la funció que fes possible la visualització de tots els estats del sensor i alhora que no donés cap missatge d'error. Després de diversos intents ho hem aconseguit, modificant la funció, posant marges d'entrada de valors de l'1 al 6 i buscant la visualització correcta perquè els valors no fessin sobresalts.

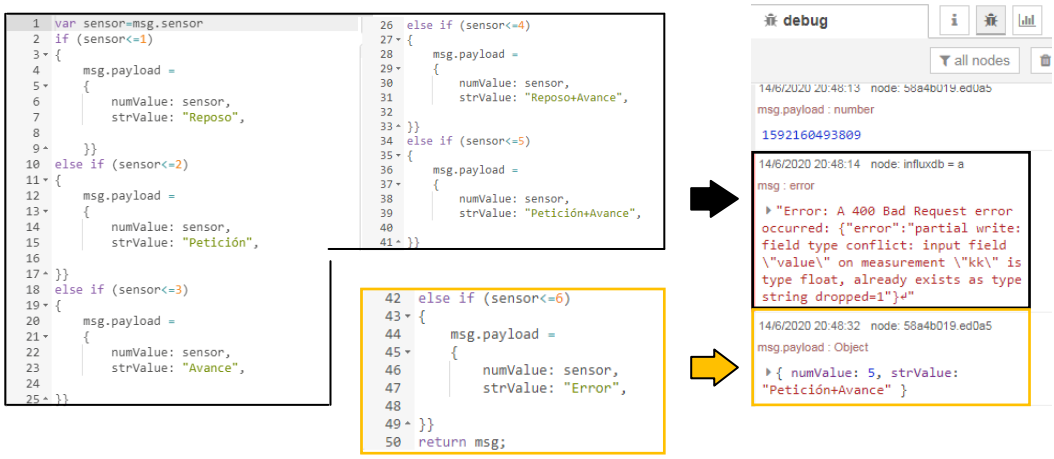


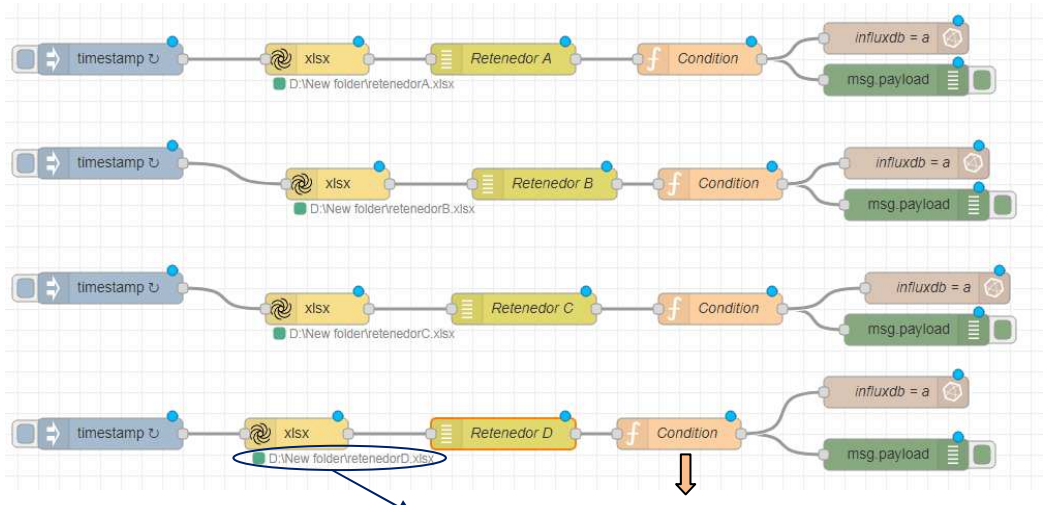
Figura 69: Solucionant problemàtica amb la funció del sensor per reconèixer variables dins el rang establert

5.2. Proves operacionals

A partir d'aquí validarem el projecte principal, al qual hem arribat després de totes aquestes proves, i de l'aprenentatge i la familiarització amb els softwares.

El primer pas hagués estat l'enviament de les dades del mateix PLC del laboratori a Node Red però degut a la situació viscuda hem emulat aquestes dades d'estat a través d'un arxiu Excel, on ens hem centrat en quatre retenidors i el seu funcionament dins la cel·la. Per això amb Node Red hem creat un flux per cadascun.

L'estructura amb 6 nodes per flux, s'inicia amb la injecció del missatge i la correcta lectura des del propi arxiu Excel, per que el node funció pugui llegir la informació i classificar-la, i finalment es visualitzi en el *"debug"* i s'emmagatzemi a Influx.



	A	B	C	D
1	1	5	1	2
2	2	4	3	2
3	3	3	5	3
4	4	2	4	3
5	5	1	2	4
6	1	5	1	4
7	2	4	3	5
8	3	3	5	5
9	4	2	2	1
10	5	1	4	1
11	1	5	3	2
12	2	4	3	2
13	3	3	4	3
14	4	2	5	3
15	5	1	2	5
16	1	5	1	5
17	2	4	3	4
18	3	3	4	4
19	4	2	2	2
20	5	1	5	2
21	1	5	3	1
22	2	4	2	3
23	3	3	1	3
24	4	2	2	2
25	5	1	2	1
26	1	5	3	1
27	2	4	4	5
28	3	3	4	4
29	4	2	5	3
30	5	1	3	2

```
var sensor=msg.sensor
if (sensor<=1)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Reposo",

    }}
else if (sensor<=2)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Petición",

}}
else if (sensor<=3)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Avance",

}}
else if (sensor<=4)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Reposo+Avance",

}}
else if (sensor<=5)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Petición+Avance",

}}
else if (sensor<=6)
{
    msg.payload =
    {
        numValue: sensor,
        strValue: "Error",

}}
return msg;
```

Figura 70: Flux dels quatre retenidors per a la visualització dels seus estats

Oberta la base de dades i el gestor *.db*, s'enregistrarà tants valors com vulguem fins que ho considerem oportú de cara a fer uns gràfics on visualitzar aquests estats.

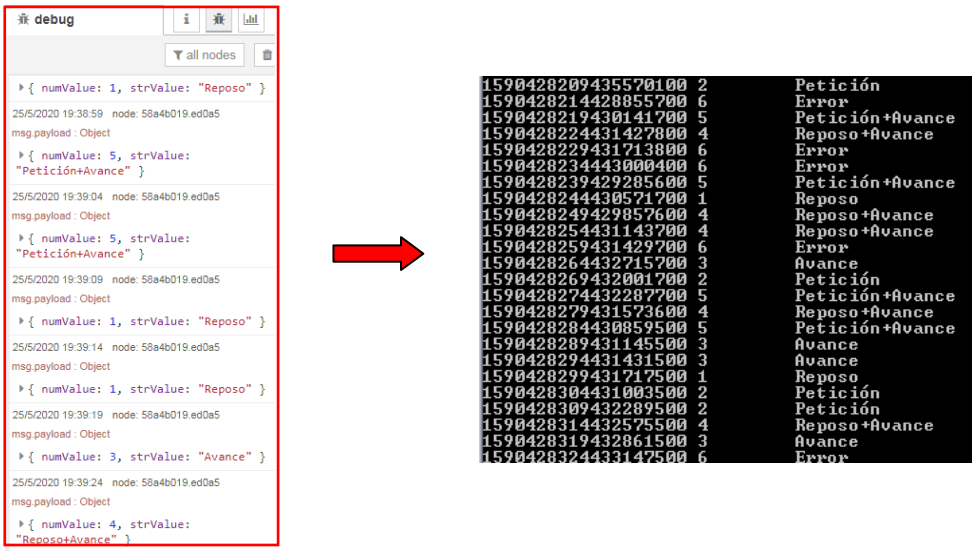


Figura 71: Emmagatzematge de les dades d'estat des de Node Red fins Influx

L'últim pas és la importació de la font de dades creada amb Influx fins al visualitzador Grafana, on mostrem les combinacions dels retenidors en el temps, cadascun d'un color, així com l'últim estat enregistrat i la mitjà d'aquests en diferents gràfics.

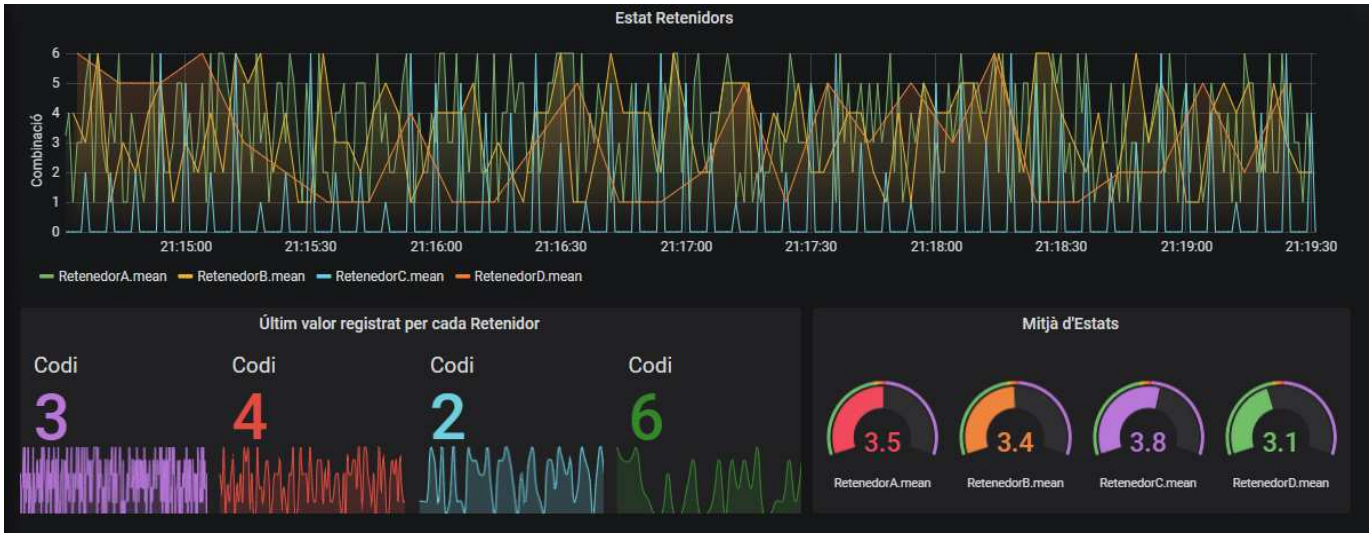


Figura 72: Visualització amb Grafana dels estats dels quatre retenidors

5.2.1. Dashboards

Amb el flux creat per a cada retenidor i el correcte emmagatzematge de les variables, només ens queda donar-li forma al projecte instaurant diferents gràfics per representar la cel·la d'una forma més real i complexa.
En el capítol anterior ens hem descarregat una sèrie de *"plugins"* que ens han ajudat en aquesta tasca d'intentar plasmar el que se'ns demana.

En primer lloc hem optat per la plantilla *"ImageIt"* en la qual ens permet la superposició d'una foto, en aquest cas la de la cel·la d'estudi, amb diferents sensors que li afegim perquè ens determini l'estat de cada retenidor.

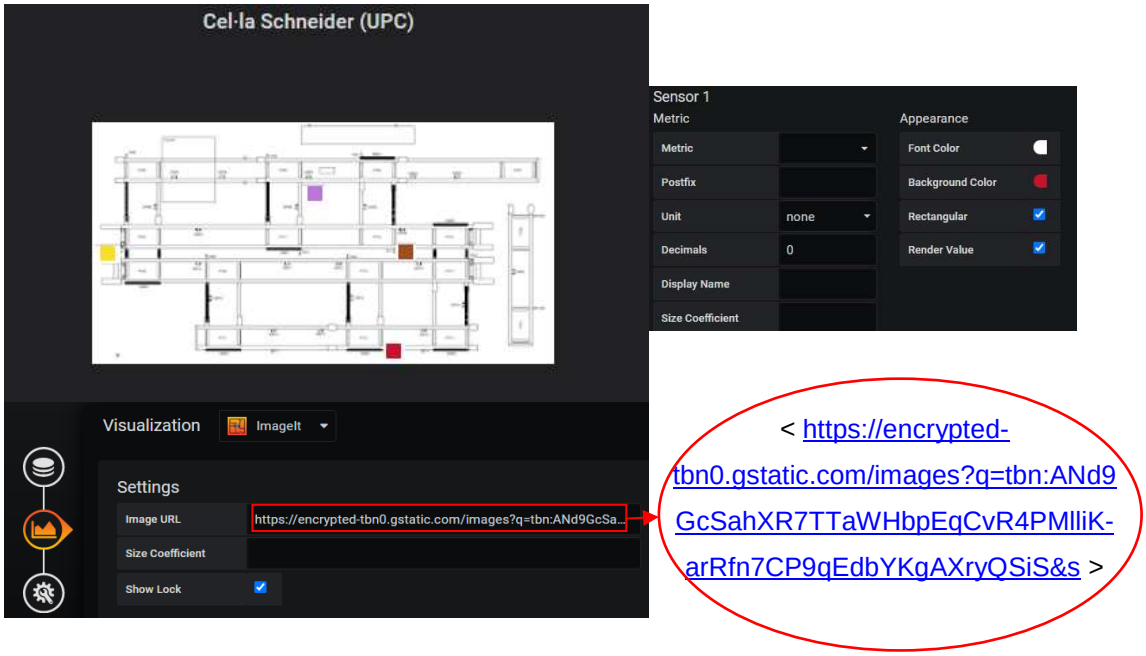


Figura 73: Configuració plugin ImageIt per a la visualització de la cel·la

Com es pot apreciar la resolució és molt baixa degut a que la fotografia és d'Internet però el resultat és el que esperàvem ja que cada rectangle ens informa d'un retenidor. Tot i això hem decidit partir de zero, és a dir, dibuixar la nostre cel·la i visualitzar amb més precisió les variables d'estat de cada un dels retenidors.
Per això hem fet ús del segon "*plugin*", el "*Flowcharting*" que ens ha permès transmetre l'essència del nostre projecte d'una manera molt visual.

Podem editar dins el propi programa en una altre pestanya o també ens deixa fer-ho des de l'escriptori de l'ordinador en una pantalla apart. El menú és molt simple ja que disposa de tot tipus de formes, text, escriptures i colors per fer-se al programa de manera agradable e idoni per representar variables d'una font de dades Influx.

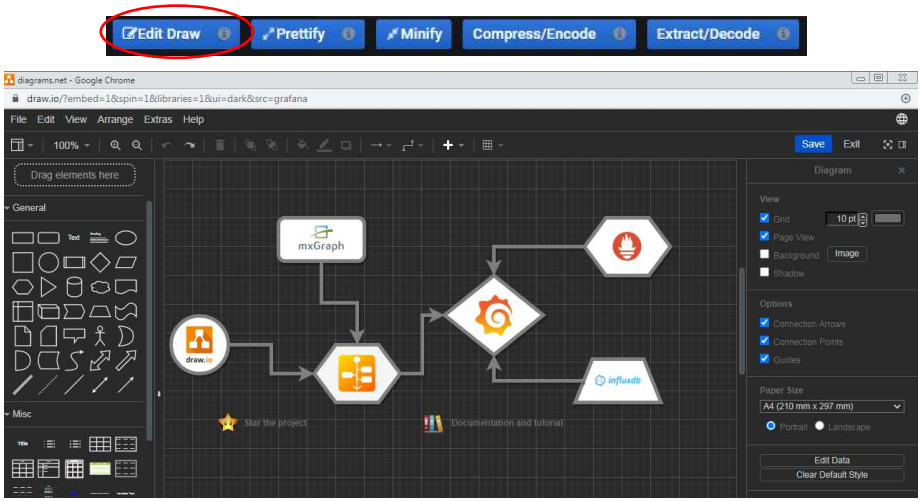


Figura 74: Menú principal per dibuixar i crear amb el plugin Flowcharting

Un cop dibuixat l'esquema de la cel·la, afegim els quatre retenidors i els vinculem amb les variables d'estat donades a Node Red i Influx.
Cada vegada que s'actualitza la pàgina de Grafana ens mostrarà l'últim registre emmagatzemat al servidor.

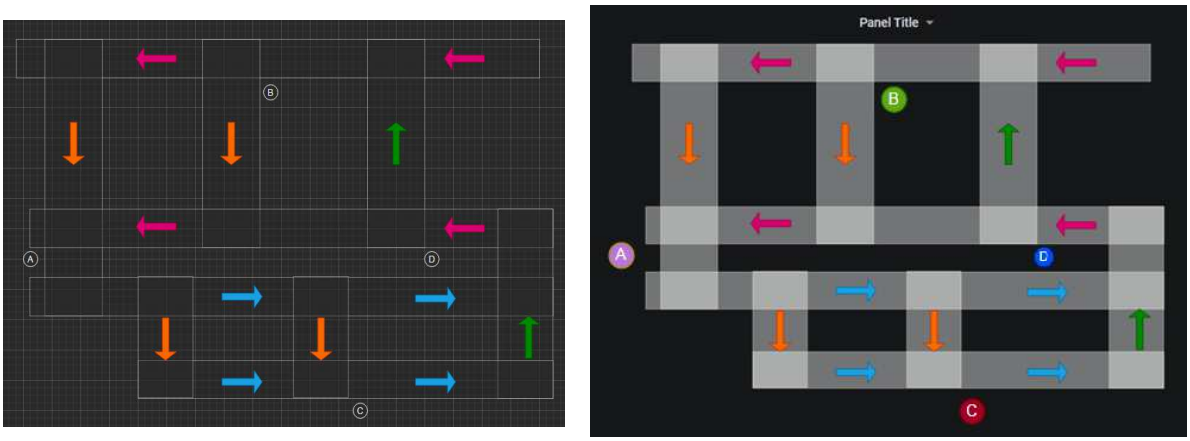


Figura 75: Dashboard de la cel·la d'estudi amb l'estat de cada retenidor

El resultat es el que esperàvem i com veiem en la imatge tenim les combinacions:

- Retenidor A – Color Lila = Codi de Repòs + Avançament
- Retenidor B – Color Verd = Codi d'Avançament
- Retenidor C – Color Vermell = Codi d'Error
- Retenidor D – Color Blau = Repòs

A part s'ha triat aquesta direcció dins la cel·la, ja que és l'òptim per minimitzar els temps de procés, on les fletxes indiquen la direcció i sentit del flux.

5.3. Alternatives

Se'ns demanava des d'un principi la seqüència dels softwares: Node Red – Influx – Grafana, però amb el primer software ho podríem haver implementat tot. Gràcies als propis *"dashboards"* de Node Red on també es pot crear un entorn amb gràfics. Mitjançant uns darrers exemples senzills veurem possibles alternatives de menys cost, poc temps però de baixa qualitat de cara a millores en el procés.

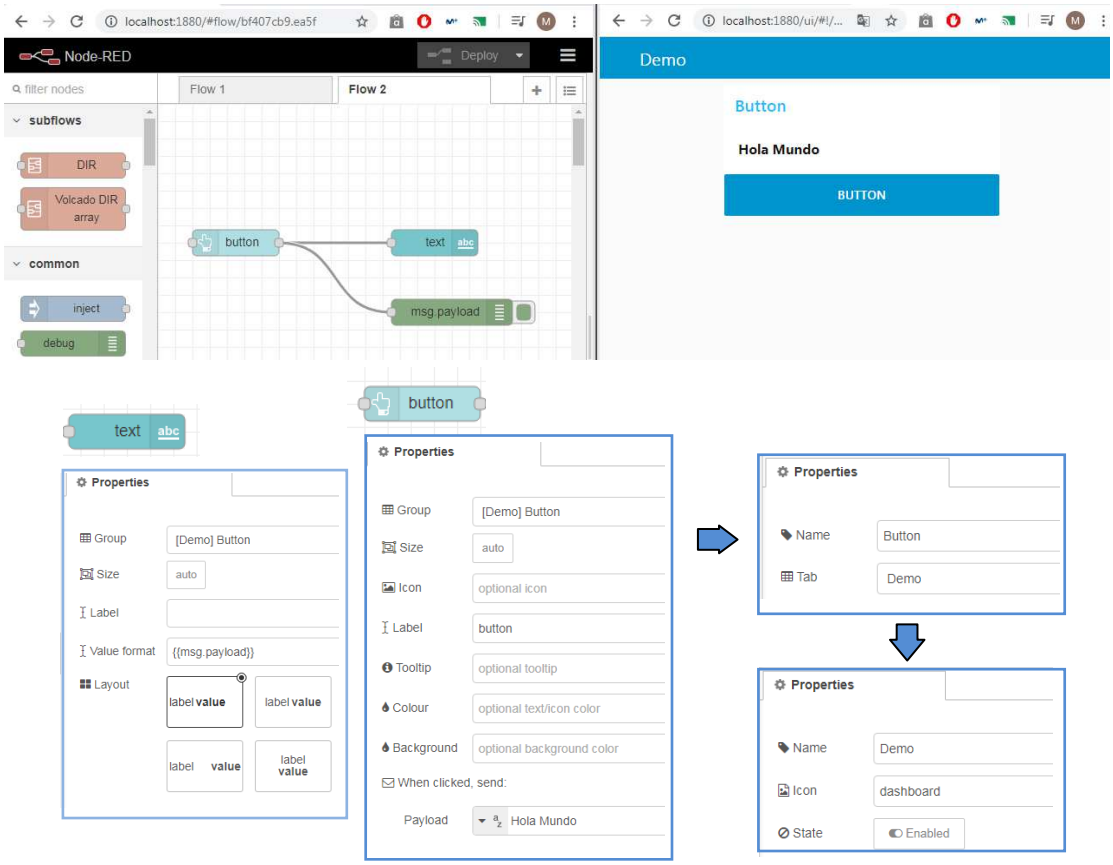
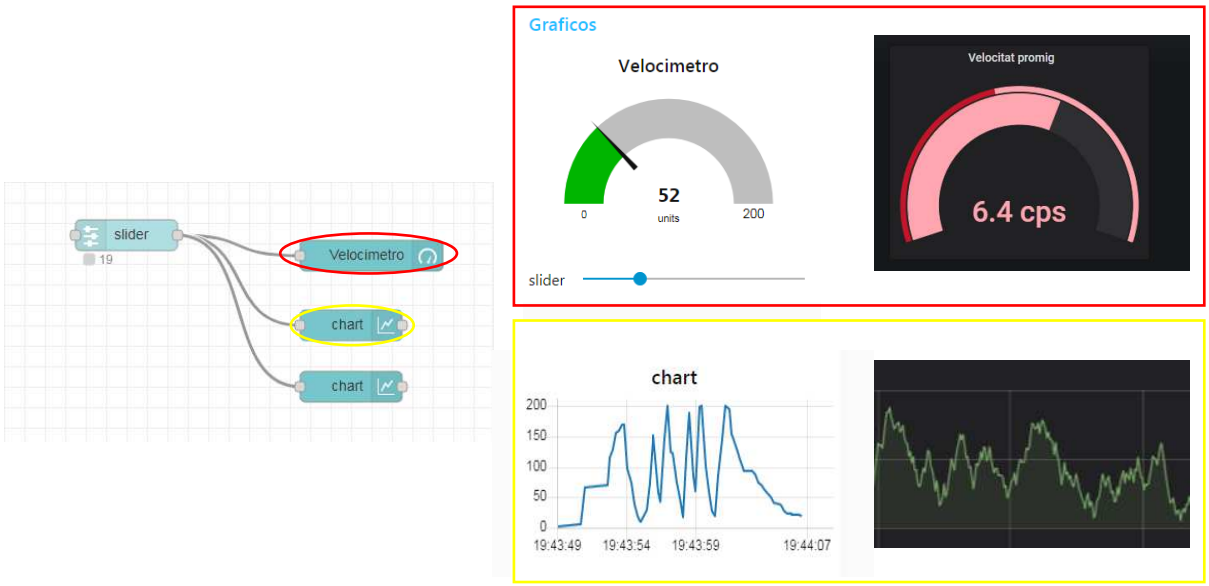


Figura 76: Traspassar un missatge en un dashboard de Node Red

Amb només dos nodes i un *"debug"* podem plasmar el missatge *Hola Mundo* a la nova direcció URL que per defecte ens dona el software: < http://localhost:1880/ui >

Una mostra de la mateixa aplicació tant en Grafana com el propi Node Red és la manera de visualitzar de manera clara gràfics en el temps.



Dashboard amb Node Red - Dashboard amb Grafana

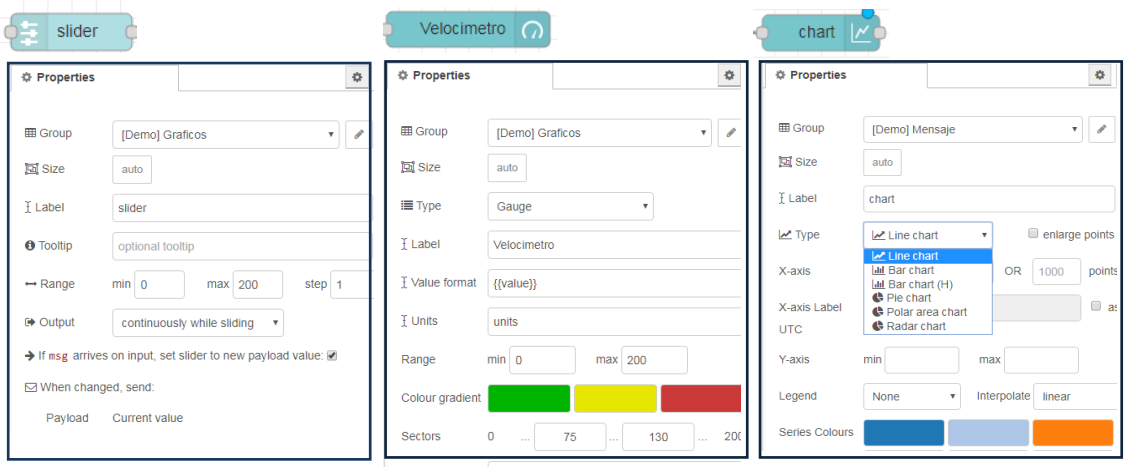


Figura 77: Comparació del mateix gràfic amb dashboards de Node Red vs Grafana

Per últim esmentar la gran possibilitat de *"dashboards"* que es poden arribar fer amb els dos programes, tot i que creiem que hem aplicat tot l'aprés sobre aquests programes i el resultat es prou satisfactori amb el problema proposat inicialment.

CAPÍTOL 6. CONCLUSIONS

Amb el projecte finalitzat és pot afirmar que la realització ha estat millor del que esperàvem, tot i les circumstancies viscudes durant aquest quadrimestre.

Hem fet un recull d'informació de tots els coneixements previs que s'havien d'assolir per entendre les diferents comunicacions i protocols que havien de transmetre's. Passant per una vessant entre connexions dels elements de la cel·la i veient superficialment els tipus de llenguatges empleats en aquests dispositius.

El primer objectiu era entendre el comportament de la cel·la i motoritzar les variables importants, com han sigut les d'estat dels retenidors, on hem acabat veient els registres que tenien durant el seu funcionament.

Un segon objectiu que s'ha vist alterat per la COVID-19, era la connexió entre els dispositius programables i els softwares per aconseguir crear un flux d'unió i emmagatzematge, que hem reinventat a traves de dades emulades.

El tercer objectiu ha estat un èxit, ja que hem dissenyat la nostre pròpia cel·la d'estudi i visualitzat les pròpies operacions d'aquesta.

Per últim s'han validat detalladament totes les proves i el projecte final on també hem donat alternatives de cara a noves monitoritzacions remotes.

Els tres grans punts que han fet tot això possible són: la recopilació de dades d'estat amb Node Red, la transmissió d'aquestes dades a través de xarxes i protocols, l'emmagatzematge i anàlisis d'aquestes a una base de dades com Influx i posteriorment la visualització i presentació dels resultats amb Grafana.

Altre tema a tractar són les possibles millores que és podrien fer com: la recopilació de variables de temps, ja sigui en ús dels retenidors, els seus temps d'espera o infinitat de d'àmbits, ja havent fet l'automatització de la cel·la. També podríem afegir més retenidors o buscar altres aplicacions com pot ser el transport de peces en una cadena de muntatge d'automòbils o un escorxador on els productes carnis són transportats.

A l'allunyar-nos del laboratori hem comprés la importància que te avui en dia l'automatització de línies de muntatge, ja que és la manera més fàcil de tenir sota control un procés industrial com l'estudia't.

Per concloure recordar que vivim en una era tecnològica en constant moviment on les empreses intentant ser pioneres en canvis i millores tan productives com empresarials, per això apostar per l'automatització i via remota dels processos, ajuda a reduir costos, temps de producció i augmentar el rendiment i benestar per a les empreses.

SUMARI

API: És un conjunt de funcions, procediments i subrutines que ofereix una certa llibreria per ser utilitzada per un altre software com a capa d'abstracció. Utilitzades generalment en llibreries de programació.

Big Data: Descriu el gran volum de dades, tant estructurats com no, que inunden el món cada dia. El que importa no és la quantitat de dades si no com estan organitzades. S'analitza per obtindre idees que ens facin prendre millors decisions o per moviments de negoci estratègics.

C++: És un llenguatge de programació que fou creat com a predecessor del C. Les principals característiques són la integració de les classes i és considera el primer llenguatge orientat a objectes.

Cookies: És un terme que fa referència a una petita informació enviada per un lloc web i emmagatzemada en el navegador, de manera que el servidor pot consultar l'activitat prèvia de navegador.

Fibra òptica: És un filament flexible de secció circular fet d'un tipus de vidre o plàstic capaç de transportar feixos de llum en el seu interior. Funciona com a una guia d'ones per al rang de freqüències compreses entre 10^{14} Hz i 10^{15} Hz. És molt important generar la freqüència de llum adequada segons el tipus de fibra amb què es treballa.

Go: És llenguatge de programació gratuït i de codi obert creat a Google el 2007.

HMI: Panell de control dissenyat per aconseguir una comunicació interactiva entre l'operador i el procés, amb la funció de transmetre ordres, visualitzar gràficament resultats i obtindre una situació de la màquina a temps real.

Host: És una computadora o un altre dispositiu connectat a una xarxa informàtica. Un servidor de xarxa pot oferir recursos, serveis i aplicacions d'informació als usuaris o altres nodes de la xarxa.

HTML: És un llenguatge de marcat que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext.

Iot: L' Internet de les coses fa referència a una interconnexió digital d'objectes quotidians amb Internet.

Modicon: Va ser el primer PLC produït comercialment.

Router: És un dispositiu de xarxa que reenvia paquets de dades entre xarxes de computadors. Aquest pertany al de nivell 3 del model OSI.

Switch: És un dispositiu que permet la connexió d'ordinadors i perifèrics a la xarxa perquè puguin comunicar-se entre ells i amb altres.

Wireless: Significa sense cables o sense fil i com el nom indica està vinculat a un tipus de comunicació que no requereix d'un mitjà de propagació físic. S'utilitza per nombrar les comunicacions en tecnologies informàtiques.

BIBLIOGRAFÍA

Treballs Finals de Grau

[1]. **Titulació:** Grau en Enginyeria Electrònica Industrial i Automàtica.

Títol: Supervisió i control d'un procés de fabricació sobre una cèl·lula automatitzada.

Autor: Borrellas Flaque, Ferran. Consultat a data: 01/06/2020. En línea a: https://upcommons.upc.edu/bitstream/handle/2117/172004/Mem%c3%b2riaTFGFerranFlaqueBorrellas.pdf?sequence=1&isAllowed=y

[2]. **Titulació:** Grau en Enginyeria Elèctrica.

Títol: Estudi de les etapes de disseny i desenvolupament d'una arquitectura ciberfísica per la monitorització d'una cel·la de producció.

Autor: Guerra Coll, Albert. Consultat a data: Juny 2020. En línea a: https://upcommons.upc.edu/bitstream/handle/2117/185193/TFG.ANEXOSAlbertCollGuerra.pdf

WEBGRAFÍA

Estat de l'Art

[1]. Antecedents de la revolució industrial. Consultat a data: 24/04/2020.
Vídeo en línea: https://www.youtube.com/watch?v=3LQAnFEADl4

[2]. Actualitat en autòmats programables. Revista digital Interempresas.
Consultat a data: 26/04/2020. En línea a:
https://www.interempresas.net/Robotica/Articulos/108924-El-automata-programable-un-universo-de-posibilidades-en-constante-evolucion.html

Historia PLC

[3]. Evolució dels PLC's. Presentació digital. Consultat a data: 01/04/2020.
En línea a: https://prezi.com/0td9urxrdbvn/evolucion-de-los-plcs/

[4]. Evolució del PLC en el temps. Blog digital. Consultat a data: 02/04/2020.
En línea a: https://www.timetoast.com/timelines/evolucion-del-plc

[5]. PLC: La evolució d'un petit gegant. Revista digital ElectroIndustria.
Consultat a data: 03/04/2020. En línea a:
http://www.emb.cl/electroindustria/articulo.mvc?xid=1131

Parts

[6]. Que és i per a què serveix un PLC. Revista digital Mecafenix.
Consultat a data: 07/04/2020. En línea a:
https://www.ingmecafenix.com/automatizacion/que-es-un-plc/

Llenguatges

[7]. Llenguatges per a la programació d'un PLC. Revista digital Mecafenix.
Consultat a data: 05/04/2020. En línea a:
https://www.ingmecafenix.com/automatizacion/lenguajes-programacion-plc/

[8]. Informe sobre el futur dels PLC y PAC: Reptes i tendències. Article digital. Consultat a data: 06/04/2020. En línea a: http://www.automaticaeinstrumentacion.com/es/notices/2019/11/informe-sobre-el-futuro-de-los-plcs-y-pacs-retos-y-tendencias-45979.php#.XooUYOozaM9

Xarxes informàtiques

[9]. Les xarxes més conegudes. Revista digital IONOS. Consultat a data: 22/04/2020. En línea a: https://www.ionos.es/digitalguide/servidores/know-how/los-tipos-de-redes-mas-conocidos/

Topologies

[10]. Topologies de xarxa. Pàgina web Culturacion. Consultat a data: 23/04/2020. En línea a: http://culturacion.com/topologia-de-red-malla-estrella-arbol-bus-y-anillo/

Transmissió dades

[11]. Comparació entre Simplex, Half-Dúplex i Full-Dúplex. Entrevista digital. Consultat a data: 17/04/2020. En línea a: https://networkinterview.com/comparison-of-simplex-half-duplex-and-full-duplex/

Model OSI

[12]. Model OSI: que és i per a que s'utilitza. Article digital. Consultat a data: 13/04/2020. En línea a: https://www.profesionalreview.com/2018/11/22/modelo-osi/13

[13]. Capes físiques. Pàgina oficial Wikipedia. Consultat a data: 14/04/2020. En línea a: https://es.wikipedia.org/wiki/Modelo_OSI#N-PDU

Model TCP/IP

[14]. Diferències entre Model OSI i Model TCP. Revista digital Medium. Consultat a data: 15/04/2020. En línea a: https://medium.com/@xxxamin1314/cu%C3%A1l-es-la-diferencia-entre-modelo-osi-y-modelo-tcp-ip-83829bbd484d

Comunicacions Industrials

[15]. Què són les xarxes de comunicació industrial?. Aula en línea.
Consultat a data: 08/04/2020. En línea a:
https://www.cursosaula21.com/que-son-las-redes-de-comunicacion-industrial/

[16]. Controlador Lògic Programable. Pàgina oficial Wikipedia.
Consultat a data: 09/04/2020. En línea a:
https://ca.wikipedia.org/wiki/ControladorlC3B2gicprogramableInterfC3ADcied'usuari

Comunicació sèrie (UART)

[17]. Com funciona un port sèrie. Pàgina web Rincón infenieril.
Consultat a data: 18/04/2020. En línea a:
https://www.rinconingenieril.es/funciona-puerto-serie-la-uart/

Codificació Manchester

[18]. Codificació Manchester. Pàgina oficial Wikipedia. Consultat a data: 21/04/2020.
En línea a: https://es.wikipedia.org/wiki/Codificaci%C3%B3n_Manchester

Ports

[19]. Que és un port sèrie. Article digital. Consultat a data: 16/04/2020.
En línea a: https://www.virtual-serial-port.org/es/article/what-is-serial-port/

Bus Can

[20]. Que és el protocol CAN?. Article digital. Consultat a data: 25/04/2020.
En línea a: https://gpstotal.org/es/sensor-automotriz/que-es-canbus

Profibus

[21]. Que és el protocol Profibus?. Aula en línea. Consultat a data: 19/04/2020.
En línea a: https://www.cursosaula21.com/que-es-profibus/

Ethernet

[22]. Que és el Ethernet. Revista digital Locura informatica.
Consultat a data: 20/04/2020. En línea a:
https://www.locurainformaticadigital.com/2018/04/06/que-es-ethernet/

Modbus

[23]. Protocol Modbus. Pàgina oficial Wikipedia. Consultat a data: 10/04/2020.
En línea a: https://es.wikipedia.org/wiki/Modbus

[24]. Trama de bits protocol Modbus. Revista digital.
Consultat a data: 11/04/2020. En línea a:
https://www.ni.com/es-es/innovations/white-papers/14/the-modbus-protocol-in-depth.html1112/04

[25]. Capes físiques. Pàgina web EEYMUC. Consultat a data: 12/04/2020.
En línea a: https://www.eeymuc.co/31-protocolo-modbus/#t%C3%ADtulo2

Node Red

[26]. Instal·lació Node Red. Article digital. Consultat a data: 27/04/2020.
En línea a: https://www.toptal.com/nodejs/programacion-visual-con-node-red-conectando-el-internet-de-las-cosas-con-facilidad

[27]. Llibreria de nodes. Pàgina oficial Node Red. Consultat a data: Maig 2020.
En línea a: https://flows.nodered.org/search?term=csv

InfluxDB

[28]. Per a que serveix Influxdb. Pàgina oficial Wikipedia. Consultat a data: 28/04/2020.
En línea a: https://en.wikipedia.org/wiki/InfluxDB

Grafana

[29]. Com funciona el software Grafana. Blog digital. Consultat a data: 29/04/2020.
En línea a: https://pandorafms.com/blog/es/que-es-grafana/

[30]. Funcionament dashboards de Grafana. Consultat a data: Maig 2020.
Vídeo en línea a: https://www.youtube.com/watch?v=X8ustpkAJ-U&t=185s

[31]. Imatge plugin Imagelt. Recuperada en:
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSahXR7TTaWHbpEqCvR4PMlliK-arRfn7CP9qEdbYKgAXryQSiS&s

Programació amb els softwares

[32]. Node Red – InfluxDB – Grafana. Recuperat en:
https://archive.org/details/ttncat-20190919-720

[33]. Node Red – InfluxDB – Grafana. Consultat a data: Maig 2020.
Vídeo en línea a: https://www.youtube.com/watch?v=g8I9i1dKqYI

Bibliografia

[34]. Referències bibliogràfiques. Pàgina per citar fonts. Consultat a data: 20/06/2020.
En línea a: https://www.citethisforme.com/es/cite/website/autocite