



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona

**FIB**

# DISSENY I IMPLEMENTACIÓ D'UN SISTEMA DE CONTROL GENÈRIC PER A EXPOSITORS

Treball realitzat per:  
**Gerard Rosell Cardús**

Dirigit per:  
**Manel Frigola Bourlon**

Màster en:  
**Enginyeria informàtica**

Barcelona, 16/10/2020

Departament ESAII

**TREBALL DE FINAL DE MÀSTER**

# Taula de contingut

1	Glossari.....	4
2	Introducció.....	5
2.1	Motivació .....	6
2.2	Metodologia .....	7
3	Estat de l'art .....	8
3.1	Màquines expenedores de productes envasats .....	8
3.2	Màquines expenedores de cafè.....	8
3.3	Màquines expenedores de productes a granel.....	8
3.4	Altres.....	9
4	Disseny global del sistema .....	10
4.1	Descripció d'entitats.....	11
4.1.1	SOM .....	11
4.1.2	Aplicació de configuració .....	19
4.1.3	Aplicació pel client .....	20
4.1.4	Servidor al núvol .....	20
4.1.5	Hardware.....	21
4.2	Blocs de comunicació .....	23
4.2.1	Bloc 1: Comunicació amb hardware .....	23
4.2.2	Bloc 2: Servei de configuració.....	27
4.2.3	Bloc 3: Aplicació pel client .....	28
4.2.4	Bloc 4: Comunicació amb el servidor de base de dades.....	29
5	Disseny del model de dades .....	31
5.1	Estructura de dades del SOM .....	32
5.1.1	“Col·lecció Configuració”.....	32
5.1.2	Col·lecció “Comunicacions” .....	32
5.1.3	Col·lecció “Usuaris” i relacionades.....	32
5.1.4	Col·lecció “Productes” .....	32
5.1.5	Col·lecció “Serveis” .....	33
5.1.6	Col·lecció “Sessions” .....	33
5.1.7	Col·lecció “Auditoria interfície”.....	33
5.2	Estructura de la base de dades del núvol.....	33
6	Planificació i costos .....	34
6.1	Avaluació de costos .....	37

7	Test i avaluació .....	38
7.1	Comunicació SOM - placa de control del hardware.....	38
7.2	Funcionament de la API .....	39
7.3	Aplicació de configuració .....	40
7.4	Aplicació de client .....	40
7.5	Conjunt de serveis i sistema .....	41
8	Conclusions i treball futur .....	42
8.1	Necessàries per tirar endavant el producte .....	42
8.2	Millores del sistema .....	42
9	Bibliografia .....	44
10	Agraïments .....	45

## Taula de Figures

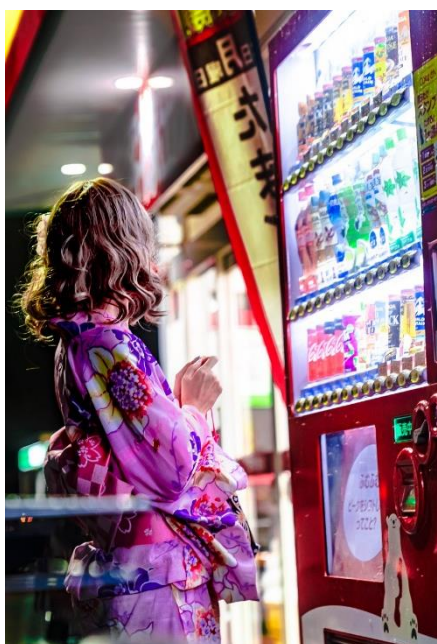
Figura 2-1 Màquina expenedora de begudes envasades .....	5
Figura 2-2 Consumidor interactua amb la màquina a través d'una pantalla tàctil per obtenir un producte .....	6
Figura 3-1 Màquina expenedora de cafè.....	8
Figura 3-2 Sistema expenedor de productes a granel.....	9
Figura 4-1 Esquema intern del sistema per blocs de disseny .....	10
Figura 4-2 SOM Connect Core X8 de Digi .....	11
Figura 4-3 Diagrama de blocs del mòdul.....	12
Figura 4-4 Pàgina d'inici de l'aplicació de configuració .....	19
Figura 4-5 Pàgina principal amb menú de navegació .....	20
Figura 4-6 Esquema del servidor del núvol connectat a les bases de dades de cada màquina.....	21
Figura 4-7 Esquema de distribució de mòduls per una placa de control .....	21
Figura 4-8 Taula de traducció del número d'ubicació a la ubicació física assignant els bits corresponents .....	22
Figura 4-9 Esquema de comunicacions entre el SOM i el hardware de la màquina .....	23
Figura 4-10 Cadena de bytes d'instruccions del SOM per la placa de control de hardware .....	24
Figura 4-11 Exemple de cadena de bytes per la operació initProduct .....	25
Figura 4-12 Cadena de bytes de respostes de la placa de control pel SOM....	26
Figura 4-13 Llista de codis d'error possibles per les respostes de la placa de control al SOM.....	26
Figura 4-14 Esquema de connexió entre aplicació de configuració i el SOM...	28
Figura 4-15 Esquema de connexió entre aplicació pel client i el SOM.....	29
Figura 4-16 Esquema de comunicació entre SOM i servidor del núvol .....	30
Figura 5-1 Estructura de la base de dades al SOM i de la rèplica del núvol ....	31
Figura 6-1 Primera part del diagrama de Gantt de planificació .....	34
Figura 6-2 Segona part del diagrama de Gantt de planificació.....	35
Figura 7-1 Mostra de la taula extreta de la col·lecció "serveis" després d'executar el test de 1000 mostres.....	38
Figura 7-2 Càlcul de temps total d'execució i del temps promig de càlcul i demora de comunicació .....	39

# 1 Glossari

SOM	<i>System on module</i> : Circuit electrònic que conté tot el necessari per executar un sistema operatiu i dur a terme una funcionalitat de servidor completa. Dit d'una altra manera és un ordinador integrat en un sol circuit.
SGBD	Sistema Gestor de Base de Dades.
SO	Sistema Operatiu.
Front End	Conversió de dades a una interfície gràfica perquè l'usuari pugui veure i interactuar amb un sistema.
Back End	Conjunt de funcions que s'engloben al funcionament d'un sistema com pot ser gestió de base de dades o comunicacions amb altres aplicacions.
LAN	<i>Local Area Network</i> : Xarxa de petites dimensions que connecta varis equips en una àrea reduïda.
WLAN	<i>Wireless Local Area Network</i> : LAN amb comunicació dins sense cables dins la xarxa.
WAN	<i>Wide Area Network</i> : Xarxa de grans dimensions que engloba comunicacions de territoris per connectar diferents xarxes.
JSON	<i>JavaScript Object Notation</i> : És un estàndard de text pla en forma d'estructures pensat per l'intercanvi d'informació entre sistemes.
API	<i>Application programming interface</i> : Conjunt de funcions, mètodes i subrutines que ofereix una llibreria per ser utilitzat per un altre software o nivell d'abstracció.
CRC	Codi de Redundància Cíclica: conjunt d'operacions sobre una cadena de caràcters per validar-ne el contingut en un altre punt de la comunicació.
Framework	Entorn de treball que agrupa un conjunt estàndard de conceptes i criteris per resoldre problemes o dissenys de tipus similar.
Punt d'accés WiFi	Dispositiu que genera una xarxa WiFi en la que s'hi permet connectar altres dispositius.

## 2 Introducció

Observant el mercat i la estructura de punts de venda actuals, trobem una diversitat de formats de presentació molt gran. Tot i això, podem observar que hi ha màquines que estandarditzen sistemes de venda a petita escala que permeten oferir productes en llocs on no es podrien posar punts de venda habituals, com ara màquines expenedores. El gran problema que duen implícit és que tot ha d'estar envasat i aïllat.



*Figura 2-1 Màquina expenedora de begudes envasades*

Partint d'aquesta base i canviant de concepte, aquest treball té com a objectiu dissenyar i implementar un sistema de control per expositors adaptable a diferents aplicacions a granel. La màquina resultant serà una combinació de software i hardware ajustable a diferents àmbits del món de les vendes parametrizant configuració que permetrà servir productes personalitzats que s'ajustin a certes característiques de les ubicacions.

En el cas del software, estem parlant d'una sèrie d'eines de configuració i gestió el més estàndards possibles per poder obtenir un ventall de mercat més gran.

A més, el sistema ha d'estar pensat per fer grans desplegaments de màquines, que es puguin gestionar amb una eina al núvol. D'aquesta manera, el control de vendes i reaprovisionament és molt més automàtic i escalable.

## 2.1 Motivació

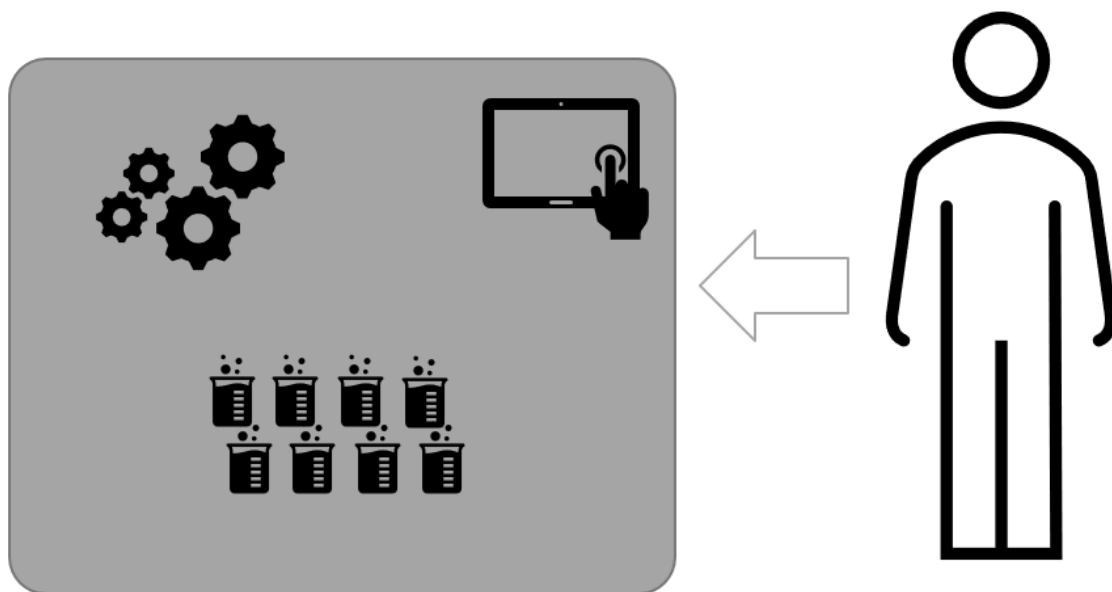
El projecte sorgeix de la observació de necessitats de gestió en certs sectors referents a les vendes. Si fem un cop d'ull al mercat, veiem que la majoria d'establiments depenen en gran percentatge de processos manuals, que requereixen mesures i controls per evitar possibles errors humans.

Partint d'aquesta premissa, com que parlem d'un ventall molt ampli de productes, ens centrem en productes que puguin emmagatzemar-se en recipients de mida fixe i que es puguin aprovisionar a granel (qualsevol tipus de gas o líquid, per exemple).

Un cop dissenyat el sistema, la idea és poder fer desplegaments de màquines controlant els grups de sistemes des d'eines al núvol per oferir un nivell de control molt alt als potencials clients.

A més, com que les necessitats poden ser molt diverses, l'objectiu no serà crear un producte final. Ens centrarem en crear tot l'esquelet i l'estructura amb les eines necessàries per fer desenvolupaments específics per cada producte i client, fent així molt més atractiu el producte.

Tot i això, la estructura de la venda ha de ser similar a una màquina expenedora actual. La veiem a la següent figura.



*Figura 2-2 Consumidor interactua amb la màquina a través d'una pantalla tàctil per obtenir un producte*

## 2.2 Metodologia

Abans de començar amb el disseny, cal veure tots els factors que intervenen en un consum d'un producte: característiques del/la consumidor/a, qualitats físiques del producte, lloc de la venda i consum, etc... Un cop tenim clares totes les variables, hem de buscar punts en comú que poden satisfer totes les necessitats de tots els entorns de venda.

Un cop tenim els diferents blocs i entitats definits, cal dissenyar la estructura de hardware i software necessària per dur-ho a terme.



## 3 Estat de l'art

El mercat actual de les màquines expenedores és molt ampli i admet molts formats de presentació de productes. Per analitzar-ne els pros i contres les agruparem.

### 3.1 Màquines expenedores de productes envasats

Dins d'aquest grup ens referim a productes d'alimentació, begudes refrigerades, objectes envasats i similars. Si analitzem el grup a nivell de presentació dels productes no ens aporten massa beneficis per les nostres necessitats ja que depenen d'envasos individuals per cada producte.

Tot i que l'emmagatzematge es pot gestionar amb espais de mida estàndard (depenent del que s'ofereixi) no és un model que ens serveixi per distribuir les nostres ubicacions i el seu contingut.

### 3.2 Màquines expenedores de cafè

Aquest tipus de màquina ja comporta servir productes a granel (no són unitats envasades) escollint el que es vol obtenir. Tot i que estan pensades per un producte molt concret, poden funcionar amb recipients estàndard pels productes i es pot modificar el mode de servir, proporcions i altres paràmetres. És una de les màquines que pot tenir un parentesc amb el nostre projecte tot i que el control acostuma a ser completament mecànic.



Figura 3-1 Màquina expenedora de cafè

### 3.3 Màquines expenedores de productes a granel

Aquest grup ens encaixa bastant, tot i que acostumen a estar pensades per sòlids com fruits secs, llegums o altres. És una infraestructura que, adaptant-la,

ens podria arribar a servir per satisfer les necessitats del projecte. Tot i això, no hi ha sistemes de màquines que es gestionin automàticament ni que es controlin informàticament personalitzant tant la experiència dels consumidors.



*Figura 3-2 Sistema expenedor de productes a granel*

### 3.4 Altres

Dins d'aquest grup s'inclouen màquines que serveixen productes menys habituals però molt concrets: expenedores de connexió WiFi, expenedores de pizza, expenedores de plantes, etc...

Les màquines d'aquest grup, tot i que pot ser que alguna situació ens servís, acostumen a estar pensades per productes únics i molt específics de manera que no compleix la estandardització que necessitem.

## 4 Disseny global del sistema

Pel disseny del sistema, tenim en compte 5 dades importants:

- El sistema estarà pensat per la venda de productes líquids o gasosos que es puguin emmagatzemar en recipients de mida estàndard (tots els recipients de la màquina iguals).
- El sistema proporcionarà el producte en qüestió sense necessitat d'una persona venedora, interactuant el consumidor directament amb la màquina.
- El sistema ha de tenir tota la informació en un magatzem de dades per ser controlat des de l'exterior (remotament o a través d'eines a un núvol).
- El reaprovisionament de producte ha de ser el més senzill possible.
- El sistema s'ha de poder configurar manualment per controlar-ne el funcionament.

Basant-nos en això, dividirem el disseny en entitats i blocs de comunicació. Les entitats faran referència a dispositius de hardware físics i els blocs faran referència a les comunicacions entre dispositius. La figura 3-1 ens mostra les entitats com a requadres i els blocs de comunicació com a fletxes bidireccionals.

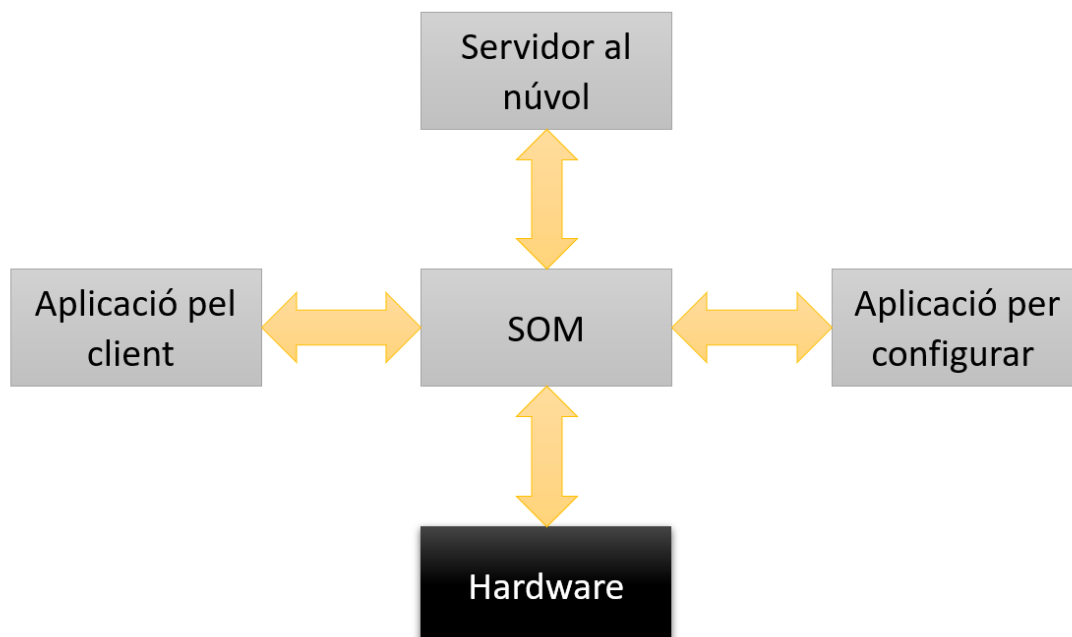


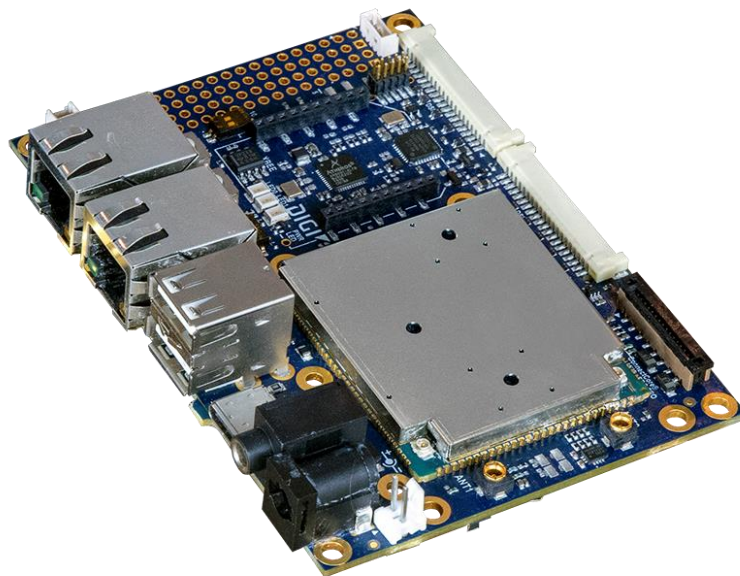
Figura 4-1 Esquema intern del sistema per blocs de disseny

## 4.1 Descripció d'entitats

### 4.1.1 SOM

El SOM és un sistema integrat en un sol mòdul, la placa principal del sistema. La seva funció és gestionar totes les comunicacions de l'expositor i emmagatzemar-ne la configuració i registre d'esdeveniments. Aquest dispositiu serà el controlador de tot el sistema permetent la comunicació entre les diferents parts i proporcionant la informació i els serveis necessaris.

El dispositiu físic que es farà servir és un SOM de la marca Digi anomenat ConnectCore 8X, el veiem a la següent figura integrat dins la seva placa d'avaluació.



*Figura 4-2 SOM Connect Core X8 de Digi*

Els principals avantatges d'aquesta placa són que porta integrats els mòduls sense fils WiFi i Bluetooth a més dels perifèrics físics com els ports Ethernet, USB, pins físics d'entrada i sortida, USB-C i altres (el sistema actual no els farà servir tots però permetrà certa flexibilitat a possibles ampliacions futures). A la següent figura veiem el diagrama de blocs del mòdul, on podrem veure els recursos dels que disposa i la connectivitat disponible.

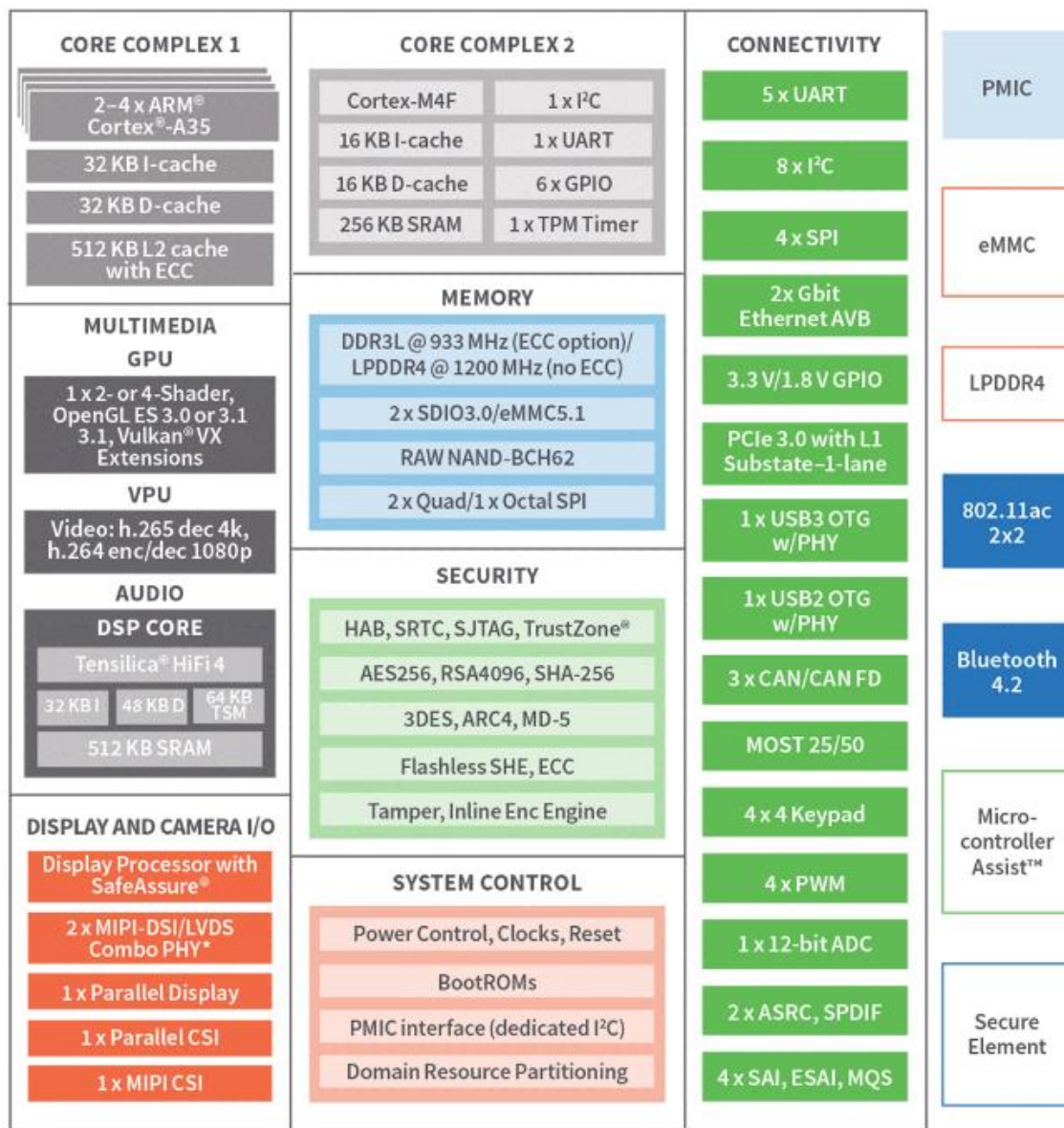


Figura 4-3 Diagrama de blocs del mòdul

Com a recomanació del fabricant, el SOM es gestionarà amb un SO específicament dissenyat i creat només amb les característiques necessàries per optimitzar-ne el funcionament i millorar-ne el rendiment. Aquest sistema operatiu és una compilació de YoctoProject, basat en una distribució Linux, en la que he triat la versió de compilació 2.6 (release 3) per problemes detectats a la versió 3.0 (la última).

Aquest sistema operatiu té una sèrie de paquets i configuracions concretes per cada processador i cada sistema. En el nostre cas, a part de la base proporcionada pel fabricant basat en el processador i perifèrics que porta, hem afegit els paquets corresponents a:

- Servidor ssh: per permetre connexions remotes a nivell de terminal.
- Servidor web Node.js: per aixecar el servei de cada aplicació que conté.
- MongoDB: és el servei pel SGBD que hi haurà dins.

Amb això ja tenim el conjunt a punt per compilar i muntar al SOM. El procés de compilació implica varies hores depenent de l'ordinador que s'utilitzi.

Un cop tenim muntat el SO, cal configurar els serveis perquè s'aixequin automàticament al iniciar el dispositiu. Els serveis configurats necessaris son:

- Punt d'accés WiFi: el SOM genera una xarxa WiFi per proporcionar les aplicacions i serveis web descrits del sistema.
- Servei MongoDB: el servei de base de dades s'ha de posar disponible al iniciar ja que moltes funcions llegiran paràmetres dallà.
- Aplicacions: les pròpies aplicacions sobre el servidor web Node.js.

El SOM allotjarà principalment 3 serveis: la aplicació de configuració, l'aplicació pel client i el servei Back End (que generarà la API per enviar instruccions a la màquina des de les dues primeres).

#### *4.1.1.1 Servei Back End*

Aquest servei serà l'encarregat de proporcionar el grup d'eines disponibles per interactuar amb la màquina des de les aplicacions. És un servei generat amb WebSocket (llenguatge javascript) que rebrà peticions amb missatges en format JSON i interpretarà què s'ha de fer.

En el cas de les peticions dirigides al hardware s'encarregarà de generar la cadena de bytes per enviar per comunicació Sèrie. En cas que les peticions siguin pel propi SOM s'encarregarà d'executar les transaccions necessàries a la base de dades.

#### *4.1.1.2 Descripció de la API*

La API generada pel Back End conté totes les funcions que seran necessàries per controlar la màquina. És un servei programat amb javascript implementant la llibreria WebSocket per proporcionar un punt de connexió i donar el servei necessari.

La funcionalitat la podem dividir en seccions depenent de qui necessitarà cridar cada una de les operacions.

##### *4.1.1.2.1 Funcions per l'aplicació de client*

La API és un conjunt de funcions amb una estructura específica definida en JSON que permet facilitar la comunicació amb la màquina des del Front End.



A continuació, es detallen les operacions disponibles i es defineix la estructura de la petició per cada una.

#### 4.1.1.2.1.1 `initProduct`

Aquesta funció indica a la màquina que ha de servir un producte i ha d'incloure els següents paràmetres:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per `initProduct` op serà igual a 1)
- **id**: identificador del producte
- **mode**: automàtic (0) / a petició (1)

Un cop el SOM rebí la petició s'encarregarà de transformar-ho en un missatge hexadecimal per fer la petició a la placa de control.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:1,  
id:"123",  
mode:0};
```

#### 4.1.1.2.1.2 `stopProduct`

Aquesta funció indica a la màquina que ha d'aturar qualsevol servei que tingui en funcionament i ha d'incloure els següents paràmetres:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per `stopProduct` op serà igual a 2)

Un cop el SOM rebí la petició s'encarregarà de transformar-ho en un missatge hexadecimal per fer la petició a la placa de control.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:2};
```

#### 4.1.1.2.1.3 `initProductMix`

Aquesta funció indica a la màquina que ha de servir un producte després de fer una barreja de productes base i ha d'incloure els següents paràmetres:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per `initProduct Mix` op serà igual a 3)
- **list**: especifica la llista de productes i com s'ha de fer la barreja
  - **id**: identificador del producte
  - **perc**: percentatge de producte
- **mode**: automàtic (0) / a petició (1)

Un cop el SOM rebí la petició s'encarregarà de transformar-ho en un missatge hexadecimal per fer la petició a la placa de control. En aquest cas, el SOM també comprovarà que es compleixin les següents condicions:

- La suma de percentatges ha de ser igual a 100. En cas contrari ha de retornar error.
- No hi pot haver un identificador de producte repetit a la llista. Si es dona la situació ha de retornar error.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:1,  
list:[  
    {id:"24", perc:80},  
    {id:"13", perc:20},  
],  
mode:0};
```



#### 4.1.1.2.1.4 `initSession`

Aquesta funció indica a la màquina que un usuari ha començat una sessió de tast i ha d'incloure els següents paràmetres:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per `initSession` op serà igual a 4)

Si es realitza la crida a la operació `initSession` no hi ha generació de missatge cap a la placa de control, el SOM simplement introduirà aquest inici de sessió a la base de dades per tenir-ne el registre.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:4};
```

#### 4.1.1.2.1.5 `endSession`

Aquesta funció és la contrària a la anterior i indica a la màquina que un usuari ha acabat una sessió de tast. Per tant, ha d'incloure els següents paràmetres:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per `endSession` op serà igual a 5)

Igual que en el cas anterior no hi ha generació de missatge cap a la placa de control i el SOM també introduirà aquest final de sessió a la base de dades per tenir-ne el registre.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:5};
```

#### 4.1.1.2.1.6 install

Aquesta funció indica a la màquina que es vol substituir un recipient per un altre de ple i ha d'incloure els següents paràmetres:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per install op serà igual a 6)
- **location**: ubicació del recipient a substituir

Un cop el SOM rebí la petició s'encarregarà de crear el missatge indicant el recipient a substituir i posarà el producte de la ubicació corresponent a 100% de capacitat dins la base de dades.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:6,  
location:145};
```

#### 4.1.1.2.2 Funcions de manteniment

La API també inclou funcions pensades per realitzar un diagnòstic en cas d'error o simplement fer el manteniment i seguiment de la pròpia màquina.

A continuació, es detallen les operacions de manteniment disponibles i es defineix la estructura de la petició per cada una.

##### 4.1.1.2.2.1 getStatus

Aquesta funció té com a objectiu saber l'estat actual de la màquina per tenir una primera avaluació. Els paràmetres necessaris són els següents:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per getStatus op serà igual a 6)

Quan el SOM rep aquesta petició respondrà directament amb l'estat actual que té emmagatzemat a la base de dades ja que periòdicament va actualitzant el valor i va controlant que tot està correcte.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:6};
```

#### 4.1.1.2.2.2 `getErrorList`

Aquesta funció fa una consulta dels últims errors que s'han registrat de la màquina. Els paràmetres necessaris són els següents:

- **user**: usuari que fa la petició
- **password**: contrasenya de l'usuari
- **op**: operació que estem invocant (per `getStatus` op serà igual a 7)
- **category**: hardware (0) / software (1) / totes (2)
- **qty**: quantitat d'errors que es volen visualitzar (amb data descendent)

Quan el SOM rep aquesta petició respondrà consultant els errors requerits a la base de dades.

Estructura del missatge en JSON:

```
{user:"usuari@domini.com",  
password:"1234",  
op:7,  
category:2,  
qty:20};
```

## 4.1.2 Aplicació de configuració

La aplicació de configuració és de caràcter web i permetrà interactuar amb el sistema per parametritzar la màquina i analitzar-ne el funcionament: donar d'alta i de baixa productes, comprovar el funcionament del hardware, consultar l'estat de la màquina i analitzar problemes o incidències. El servei d'aquesta aplicació estarà ubicat també dins del SOM encara que el client s'executi des de qualsevol dispositiu compatible amb WiFi i navegador (pensat preferiblement per una tableta o un ordinador).

### 4.1.2.1 Descripció de l'aplicació

L'aplicació constarà de 6 pestanyes després del control d'accés: configuració de la màquina, configuració de l'aplicació, configuració de productes, auditoria, prova de producte i informació del sistema.

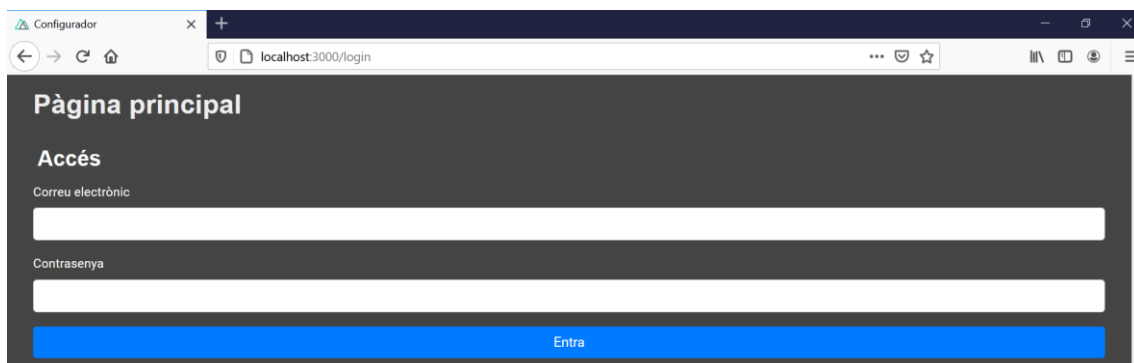


Figura 4-4 Pàgina d'inici de l'aplicació de configuració

Enumerant pestanya a pestanya del menú lateral esquerre, tenim:

- Configuració de la màquina: conté parametrització de comunicacions, connexions i un mode depuració per analitzar la màquina en cas de tenir problemes tècnics.
- Configuració aplicació: conté la gestió d'usuaris, rols i permisos de la aplicació.
- Configuració de productes: conté la informació dels productes assignats a cada ubicació i l'estat en que es troben. En aquesta pantalla és on es permetrà donar d'alta productes per poder ser instal·lats físicament a la màquina. El límit de la versió dissenyada és 256 productes.
- Auditoria: pestanya que permet consultar modificacions i accions que s'han dut a terme en cada una de les dades emmagatzemades.

- Prova de producte: pestanya destinada a manteniment de la màquina que permetrà fer serveis de producte simples utilitzant ubicacions físiques concretes per comprovar-ne el funcionament.
- Informació del sistema: conté informació de drets i identificació de la instal·lació.

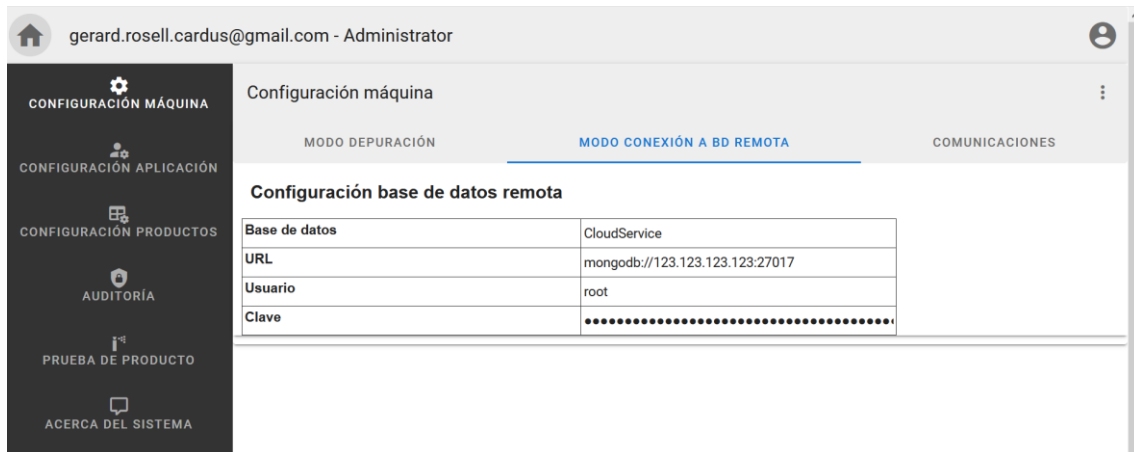


Figura 4-5 Pàgina principal amb menú de navegació

### 4.1.3 Aplicació pel client

És aplicació que utilitzarà el potencial comprador del producte. Igual que en el cas anterior, estem parlant d'una aplicació web que podem executar des de qualsevol dispositiu amb navegador capaç de connectar-se a una xarxa WiFi. L'objectiu d'aquest és utilitzar les funcions del sistema i servir-se el producte.

Tal com s'ha esmentat a la introducció, aquesta aplicació formaria part del desenvolupament específic per cada producte, marca i client. El projecte inclou una aplicació amb les crides bàsiques com a demostració per la creació de la aplicació definitiva.

### 4.1.4 Servidor al núvol

La funció d'aquest dispositiu és emmagatzemar el registre de tots els expositors per monitoritzar-ne el funcionament i l'estat. El dispositiu físic d'aquest no és rellevant ja que consisteix en una màquina virtual accessible des de la xarxa amb els serveis descrits.

Aquesta màquina virtual tindrà un sistema operatiu Ubuntu 18.04 que contindrà un servei MongoDB amb la infraestructura de base de dades corresponent a l'agrupació de màquines.

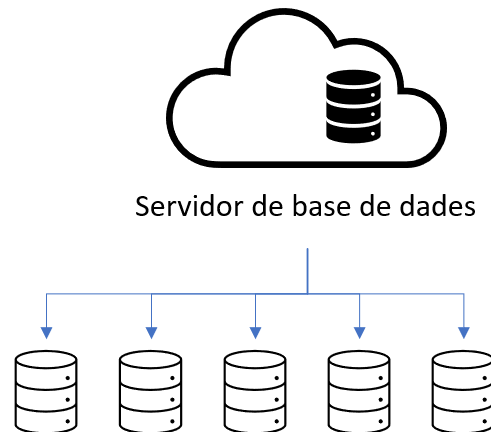


Figura 4-6 Esquema del servidor del núvol connectat a les bases de dades de cada màquina

#### 4.1.5 Hardware

Aquesta entitat inclou tot el conjunt de motors que controlarà el nostre sistema. El disseny i desenvolupament d'aquest, però, no forma part del projecte. Pel nostre sistema el HW serà una caixa negra a la que li enviarem ordres i de la que rebrem informació per registrar.

Encara que no es programi ni es dissenyi aquest apartat, si que serà necessària una distribució dels recipients fixada per fer referència a ubicacions coherents.

La quantitat de recipients disponibles i la seva disposició ve definida. Per fer la màquina escalable en millores futures al projecte, s'han distribuït els recipients en mòduls de 8 posicions agrupats en blocs de 8 mòduls. Per defecte, si tenim 4 blocs ens surten un total de 256 posicions.

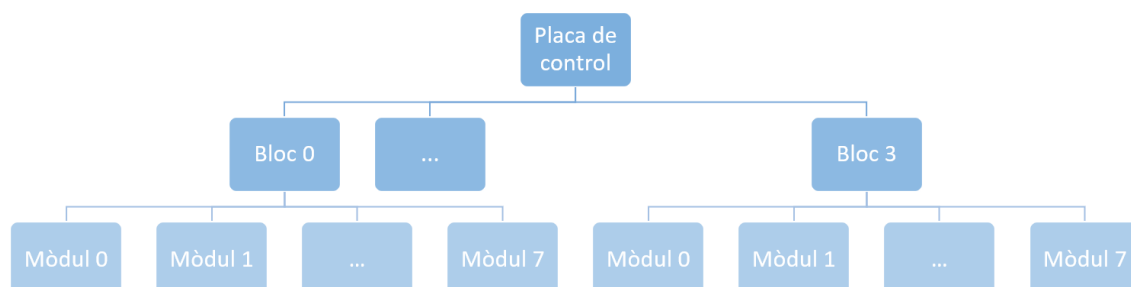


Figura 4-7 Esquema de distribució de mòduls per una placa de control

Si ho representem amb els valors de 0 a 255 traduïnt a cada ubicació final obtenim la figura següent (valors agafats a l'atzar de mostra per no adjuntar totes les combinacions).

ID	Cadena	2 bits	Bloc	3 bits	Mòdul	3 bits	Posició	Ubicació
0	00000000	00	0	000	0	000	0	0
26	00011010	00	0	011	3	010	2	1A
117	01110101	01	1	110	6	101	5	75
161	10100001	10	2	100	4	001	1	DE
255	11111111	11	3	111	7	111	7	FF

Figura 4-8 Taula de traducció del número d'ubicació a la ubicació física assignant els bits corresponents

Partint d'aquest disseny es pot ampliar la màquina afegint plaques de control i bytes a les ubicacions, respectivament.

## 4.2 Blocs de comunicació

El funcionament del sistema el separarem en 4 blocs diferenciats poder afinar en la descripció d'aquest: comunicació amb HW, servei de configuració, Front End d'usuari i comunicació amb servidor.

### 4.2.1 Bloc 1: Comunicació amb hardware

Aquest bloc defineix la interacció entre el SOM i la mecànica de l'expositor. Tal com es mostra a la Il·lustració 3, el flux d'informació d'aquest bloc esdevé sobre una comunicació sèrie i el dividim en els 2 sentits:

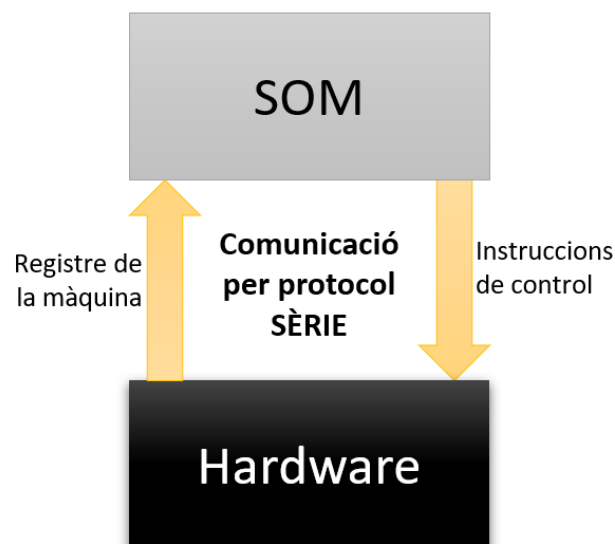


Figura 4-9 Esquema de comunicacions entre el SOM i el hardware de la màquina

#### 4.2.1.1 Instruccions de control de la mecànica

Són el grup definit d'instruccions que pot enviar el SOM a la placa de control de la mecànica que accionarà la funcionalitat de l'expositor. Totes les instruccions que es generin al SOM per ser enviades per aquest canal es dispararan des del servei Back End, mantenint un registre a la base de dades de l'origen del requeriment i una marca temporal per poder analitzar el funcionament.

Hi ha 3 possibles orígens de les instruccions: requeriments de l'aplicació de configuració a través de la API, requeriments de l'aplicació client també a través de la API o generades des del propi SOM per control del sistema. En qualsevol dels 3 casos, el Back End traduirà les peticions en cadenes de bytes interpretables per la placa de control del HW i registrarà a la base de dades del



SOM la petició per mantenir un registre d'activitat i poder fer un manteniment correcte. Seguidament farà l'enviament de la cadena de bytes.

Trobarem el protocol definit a l'apartat "Missatges del SOM a la placa de control".

#### 4.2.1.2 Registre de màquina

El registre és el conjunt d'informació d'estat i historial de funcionament de la mecànica. L'objectiu és monitoritzar tot el que implica estat i instruccions executades. Aquesta informació es generarà automàticament definit en intervals de temps, per monitoritzar l'estat periòdic, o bé quan esdevingui un error per detallar què ha passat i quan. El destí d'aquesta informació serà sempre la base de dades del SOM.

Trobarem el protocol definit a l'apartat "Missatges de la placa de control al SOM".

#### 4.2.1.3 Missatges del SOM a la placa de control

Codi	Descripció	Longitud (bytes)	Dec	Hex	Bin
CI	Caràcter d'inici	1	201	C9	11001001
CF	Caràcter de final	1	202	CA	11001010
CE	Caràcter d'escapament	1	210	D2	11010010
EI	Caràcter d'escapament d'inici	1	211	D3	11010011
EF	Caràcter d'escapament de final	1	212	D4	11010100
EE	Caràcter d'escapament d'un escapament	1	213	D5	11010101
op	Operació de la petició	1			
position	Posició del producte: 2 bits bloc, 3 bits mòdul, 3 bits posició dins del mòdul	1			
mode	Mode automàtic (0) o a petició (1)	1			
perc	Percentatge per una barreja	1			
CRC	Codi de redundància cíclica (bytes de seguretat). Algoritme utilitzat CRC16	2			

Figura 4-10 Cadena de bytes d'instruccions del SOM per la placa de control de hardware

A continuació, partint del significat dels valors dels bytes, veurem la traducció d'un missatge "initProduct" com a exemple:

**Pas 1:** rebre el missatge en format JSON rebut a través de la API:

```
{user:"gerard.rosell.cardus@gmail.com",
password:"0000",
op:1,
id:"ABC123",
mode:0};
```

**Pas 2:** validar usuari i contrasenya contra la base de dades del SOM.

**Pas 3:** Buscar la ubicació del "id" de producte a la col·lecció productes: ABC123 correspon a la ubicació 156 (dit d'una altra manera 10011100 que implica bloc 2, mòdul 3 i posició 4 segons la figura de posicions de l'Annex III).

**Pas 4:** Creació de la cadena amb totes les dades i enviament cap a la placa de control per comunicació sèrie.

	Cl	op	position	mode	perc	CRC	CF
<b>decimal</b>	201	1	156	0	100	47933	202
<b>hexadecimal</b>	C9	1	9C	0	64	BB3D	CA

*Figura 4-11 Exemple de cadena de bytes per la operació initProduct*

**Pas 5 (executat a la placa de control de hardware):** aplicar el mateix càlcul CRC16 per comprovar que el missatge és correcte.

**Pas 6:** executar les accions necessàries sobre el hardware.

**Pas 7:** enviar la resposta corresponent al SOM.

#### 4.2.1.4 Missatges de la placa de control al SOM

Codi	Descripció	Longitud (bytes)	Dec	Hex	Bin
CI	Caràcter d'inici	1	201	C9	11001001
CF	Caràcter de final	1	202	CA	11001010
CE	Caràcter d'escapament	1	210	D2	11010010
EI	Caràcter d'escapament d'inici	1	211	D3	11010011
EF	Caràcter d'escapament de final	1	212	D4	11010100
EE	Caràcter d'escapament d'un escapament	1	213	D5	11010101
result	Resultat de la operació: 0 OK / 1 error	1			
desc	0 sempre que result=0. Si ha error un valor de la llista d'errors	1			
CRC	Codi de redundància cíclica (bytes de seguretat). Algoritme utilitzat CRC16	2			

Figura 4-12 Cadena de bytes de respostes de la placa de control pel SOM

En el cas dels missatges que s'envien des de la placa de control al SOM es genera la cadena de bytes de la mateixa manera que en l'exemple de l'apartat anterior. En aquest cas, s'indica que s'ha executat correctament i, en cas negatiu, quin codi d'error té associat.

#### 4.2.1.5 Llista d'errors

Valor	Descripció
1	paràmetres incorrectes
2	ubicació inexistent
3	ubicació buida
4	timeout mode petició
5	error de recepció de missatge

Figura 4-13 Llista de codis d'error possibles per les respostes de la placa de control al SOM

Si observem la figura anterior tenim com a possibles missatges d'error:

- Paràmetres incorrectes: missatge JSON incoherent en referència a la operació. S'esperaven una estructura de paràmetres diferent.

- Ubicació inexistent: apareix aquest error quan tenim una ubicació sense producte assignat. La màquina no obliga a tenir totes les ubicacions amb productes.
- Ubicació buida: apareix aquest error quan tenim un recipient amb un percentatge disponible igual a 0 o si tenim el producte marcat com a inactiu a la base de dades perquè no interessa mostrar-lo com a disponible.
- Timeout mode petició: aquest error és específic de la operació "initProduct" quan es llença en mode petició. Aquest mode esperarà a que el consumidor interactuï per servir (ja sigui amb un botó, sensor o similar). La màquina marcarà un timeout de 30 segons i retornarà aquest error si no hi ha hagut interacció.
- Error de recepció: aquest error apareix quan el càlcul de CRC no és correcte, quan no hi ha codi inicial o quan no hi ha codi final.

#### 4.2.1.6 Codi de redundància cíclica

Els bytes corresponents al codi de redundància cíclica són el resultat d'un grup d'operacions efectuades a partir dels altres bytes de la cadena que es vol enviar. L'objectiu és poder fer una comprovació al rebre el missatge per assegurar que està complet i no ha tingut interferències o altres problemes de transmissió.

El càlcul d'un CRC és basa resoldre una divisió de la cadena de bytes que volem enviar (passada a binari) i un polinomi conegut que ens donarà l'algoritme escollit. En el cas del CRC16, el polinomi és el següent:

$$P(x) = x^{16} + x^{15} + x^2 + 1$$

O representat en binari: 11000000000000101

Dividint i obtenint el residu (que sempre tindrà una mida de 16 bits com a màxim, igual que el polinomi) l'adjuntem com a CRC i el receptor pot repetir el càlcul per verificar.

#### 4.2.2 Bloc 2: Servei de configuració

Aquest bloc defineix la comunicació entre l'aplicació de configuració i el SOM, que esdevindrà sobre tecnologia TCP/IP. Consistirà en una aplicació web creada amb el framework Nuxt.js (de Vue.js) que es comunicarà amb el SOM a través d'una API anteriorment esmentada. El servei es generarà des del SOM i estarà accessible connectant a la seva direcció IP des del navegador d'un dispositiu a través d'una xarxa WiFi generada pel propi SOM (que funcionarà com a punt d'accés).

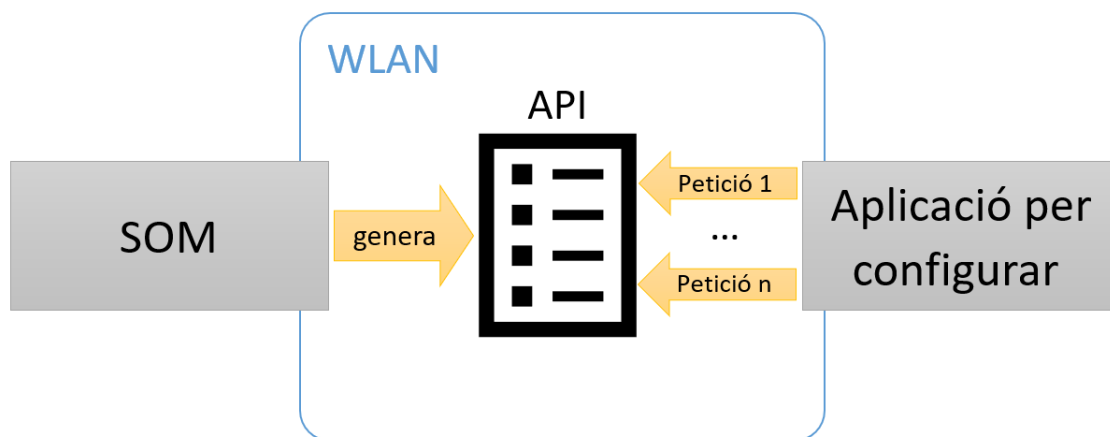


Figura 4-14 Esquema de connexió entre aplicació de configuració i el SOM

L'objectiu d'aquesta aplicació és poder configurar tots els paràmetres de la mecànica de l'expositor com afegir i treure productes o canviar els paràmetres dels motors.

A part d'això, tindrà un apartat de manteniment orientat a anàlisi d'error i depuració del sistema. Aquest apartat només estarà disponible pels desenvolupadors i els encarregats del manteniment del sistema, no està enfocada a perfils no tècnics (tant per simplificar com per evitar possibles problemes).

Tal com està definit al Bloc 2, tota aquesta comunicació va dirigida al servei Back End i a la escriptura d'activitat a la base de dades del SOM.

#### 4.2.2.1 Seguretat

En aquest bloc, per seguretat, hi haurà una validació d'usuari per assignar els permisos que corresponguin i controlar que cada individu pugui gestionar la part de la màquina que toqui.

#### 4.2.3 Bloc 3: Aplicació pel client

Aquest bloc és la part de client del sistema. Igual que l'aplicació de configuració, consistirà en una aplicació web comunicada amb el SOM a través de la API. En aquest cas, la informació i accions que formaran part de l'aplicació són les crides a les instruccions de funcionament que puguem donar-li a l'expositor.

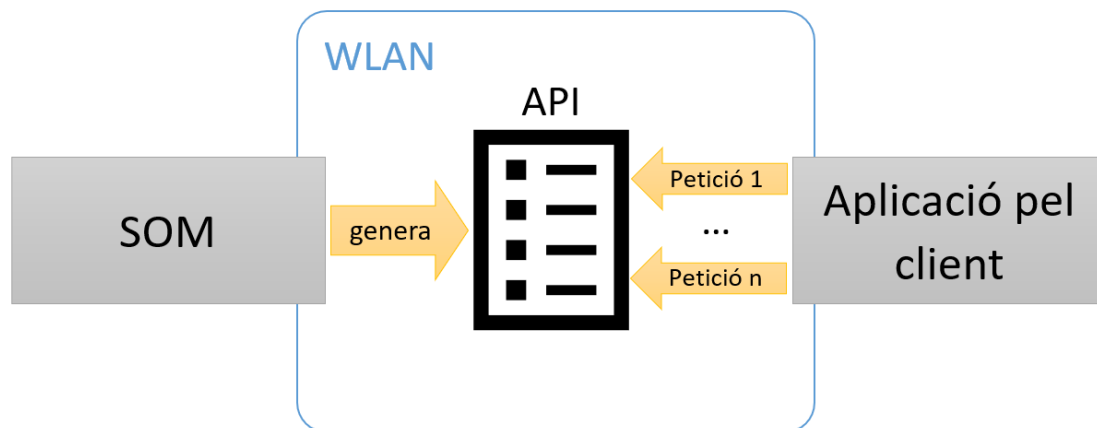


Figura 4-15 Esquema de connexió entre aplicació pel client i el SOM

Aquesta aplicació es desenvoluparà per cada client concret adaptant-se al producte a oferir. En aquest projecte, però, inclourà una petita aplicació que contingui les crides a la funcionalitat bàsica per fer demostracions i ajudar als desenvolupadors que s'encarreguin d'aquesta part.

#### 4.2.4 Bloc 4: Comunicació amb el servidor de base de dades

Aquest bloc és la definició de la comunicació entre el SOM i el servidor de base de dades (situat en un servidor virtual fora de la LAN). Aquest bloc tindrà un flux d'informació programat i periòdic amb varis objectius:

- Realitzar còpies de seguretat: si qualsevol de les màquines té algun problema mecànic s'ha de poder recuperar amb les dades del servidor.
- Monitoritzar les màquines: analitzant la informació s'han de poder extreure estadístiques, ratis i dades rellevants sobre les vendes i consums dels productes. També s'ha de poder consultar l'estat de producte que queda en els recipients calculant els consums periòdicament.
- Descarregar informació: si es volen fer actualitzacions sobre el contingut s'ha de poder fer massivament.

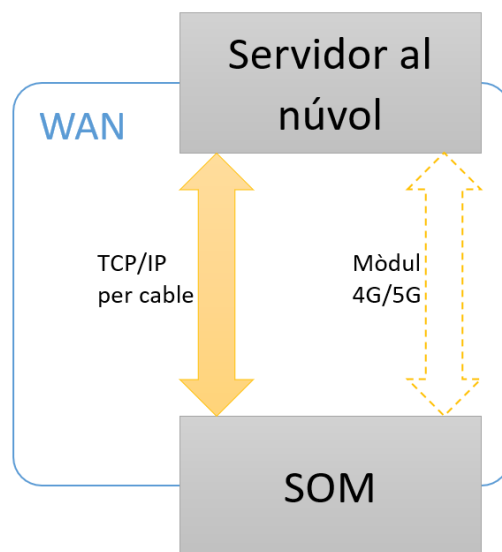


Figura 4-16 Esquema de comunicació entre SOM i servidor del núvol

Mòdul 4G/5G esmentat a accions futures.

#### 4.2.4.1 Seguretat

No estarà inclòs en el projecte però aquest intercanvi d'informació ha d'incorporar el xifrat a través d'un certificat SSL. És una de les millores i accions futures per posar el sistema en producció.

## 5 Disseny del model de dades

Tenint en compte que totes les comunicacions que passen pel SOM són contra la base de dades, podem assumir que serà el nucli del funcionament del sistema. La base de dades funcionarà sobre un servei MongoDB i es comunicarà periòdicament amb una base de dades principal al núvol. Partint d'aquesta base, tindrem una rèplica de les col·leccions de dades que tenim al SOM ubicada al núvol en la que s'aniran fent pujades incrementals. A la següent figura veiem l'esquema simplificat:

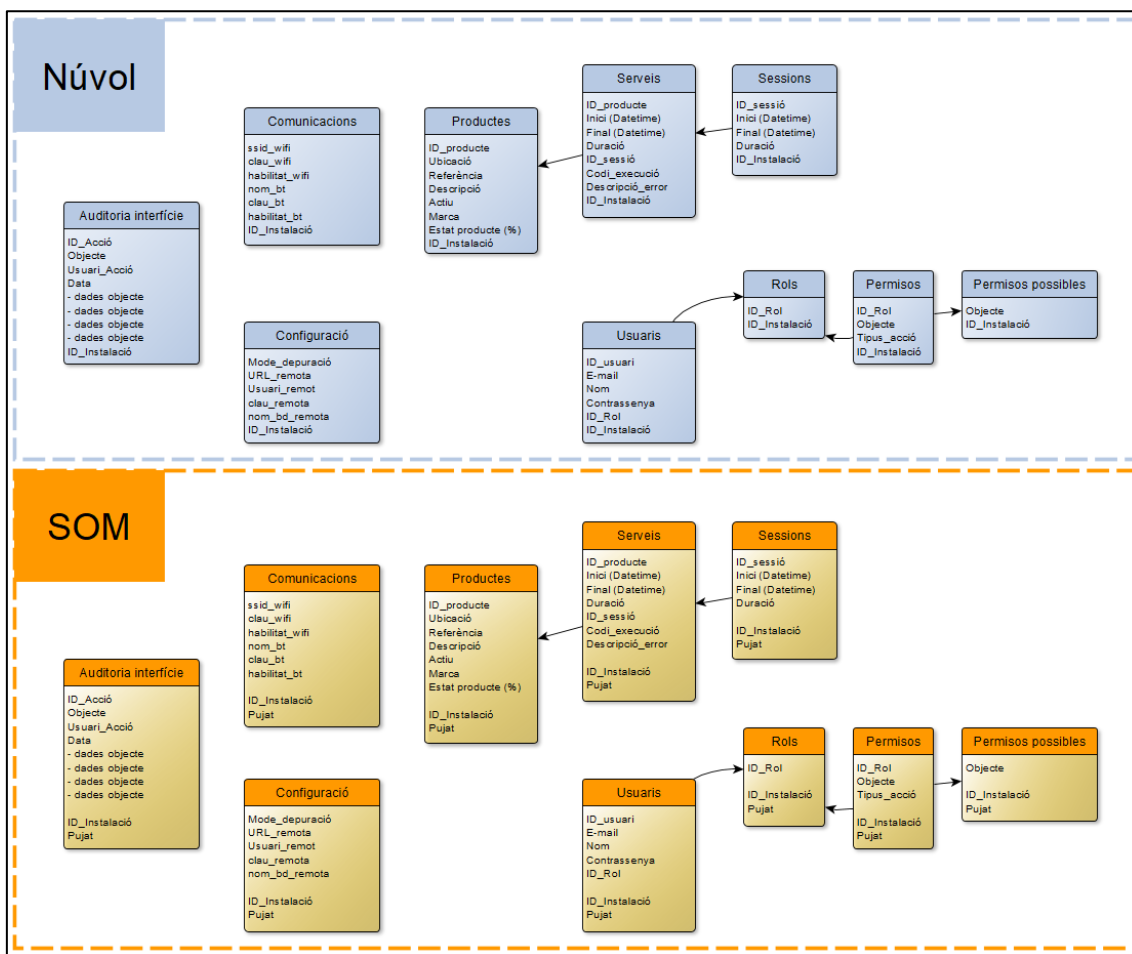


Figura 5-1 Estructura de la base de dades al SOM i de la rèplica del núvol

Basant-nos amb aquest esquema, podem definir les col·leccions i les seves funcionalitats.



## 5.1 Estructura de dades del SOM

### 5.1.1 “Col·lecció Configuració”

Aquesta col·lecció només conté un document, que es va actualitzant a través de l'aplicació de configuració. La seva funció és emmagatzemar paràmetres de connexió a la base de dades del núvol i habilitar o no el mode depuració (aquest últim és un simple valor booleà).

L'objectiu és que la connexió llegeixi les credencials i l'adreça del servei del núvol per poder mantenir-ho configurat fàcilment si es fan canvis.

### 5.1.2 Col·lecció “Comunicacions”

La idea d'aquesta col·lecció és la mateixa que la anterior però amb informació referent a la connectivitat sense cables. També estarà formada per un únic document que contindrà la xarxa i la contrasenya per connectar a la xarxa WiFi que genera el propi SOM i per connectar a través de la interfície Bluetooth.

La interfície Bluetooth no s'utilitzarà durant el projecte però ho he inclòs com a possible desenvolupament posterior en versions de millora del sistema.

### 5.1.3 Col·lecció “Usuaris” i relacionades

Aquesta col·lecció s'utilitza per la gestió d'accessos a la aplicació, permisos a les zones i control d'usuaris. També estan gestionades des de l'aplicació de configuració i tenen la relació següent; un registre a la col·lecció usuaris conté un usuari, el qual té assignat un rol d'accés que, relacionat amb el rol, té assignat un grup de permisos.

Per facilitar la gestió, hi ha la col·lecció de “Permisos possibles” que ens mostraran a l'aplicació de configuració què es pot permetre i/o denegar a cada rol.

### 5.1.4 Col·lecció “Productes”

La col·lecció de productes és la que emmagatzemarà tota la informació del contingut de la màquina i d'on es troba cada producte. Des d'aquí podrem consultar dades com la informació i descripció dels productes, el percentatge que queda al recipient, la ubicació física del recipient i si està actiu o no per ser servit.

El plantejament és gestionar des d'aquí què apareixerà com a disponible per consumir a la aplicació pel client de manera que es tingui un control complet sobre el que es serveix.

### 5.1.5 Col·lecció “Serveis”

La col·lecció serveis ens mostrarà un registre de totes les peticions per servir producte amb informació de si s'ha dut a terme correctament. En base a aquestes dades podrem actualitzar també els percentatges disponibles als recipients de la col·lecció anterior estudiant, en cada cas, les unitats de consum per unitats de temps.

Aquesta és la col·lecció que, analitzada al núvol, ens proporcionarà informació de reaprovisionament aproximada.

### 5.1.6 Col·lecció “Sessions”

És una col·lecció similar a la anterior, destinada al registre, però enfocada a les vegades que es fa servir la màquina. El funcionament és anar registrant sessions (a través de “initSession” i “endSession”, definit a la API de l'Annex I) per analitzar ratis de vendes respecte sessions o temps de sessió.

L'anàlisi d'aquesta també està plantejat al núvol.

### 5.1.7 Col·lecció “Auditoria interfície”

Aquesta col·lecció és la encarregada de registrar tots els canvis de les altres col·leccions guardant una marca temporal i classificant cada esdeveniment. Està destinada només al registre dels esdeveniments de les col·leccions relacionades amb l'aplicació de configuració.

## 5.2 Estructura de la base de dades del núvol

La estructura al núvol és la mateixa que la del SOM tenint en compte que assigna cada registre a un identificador d'instal·lació.

Aquestes col·leccions serviran bàsicament per analitzar les vendes de productes i les necessitats de cada instal·lació. L'objectiu final és gestionar els desplegaments de màquina sense necessitat d'estar in-situ de manera que es puguin enviar recanvis de productes automàticament i que arribin les dades de vendes a direcció directament.

A més, en el cas que alguna màquina tingui algun problema que requereixi que es torni a configurar, tindrem tota la informació disponible per descarregar i tornar a posar en funcionament el sistema més ràpid.

## 6 Planificació i costos

A l'hora de planificar i valorar costos del projecte tindrem en compte dues unitats de desenvolupament del projecte: temps de disseny i temps d'implementació. Fet d'aquesta manera, podrem valorar costos de realitzar una oferta definida i costos de la execució del projecte en una suposada contractació. Les tasques de disseny les veurem en verd i les d'implementació en groc. També tindrem divisions de tasques marcades amb un fons gris per dividir tasques grosses.

Per representar-ho es farà servir un diagrama de Gantt per veure la evolució teòrica de les tasques en blocs de mitja jornada laboral (4 hores de feina).

Dins d'aquests temps només s'inclou el que s'hi ha d'invertir efectiu independentment del temps invertit en aprenentatge o investigació de tecnologies.

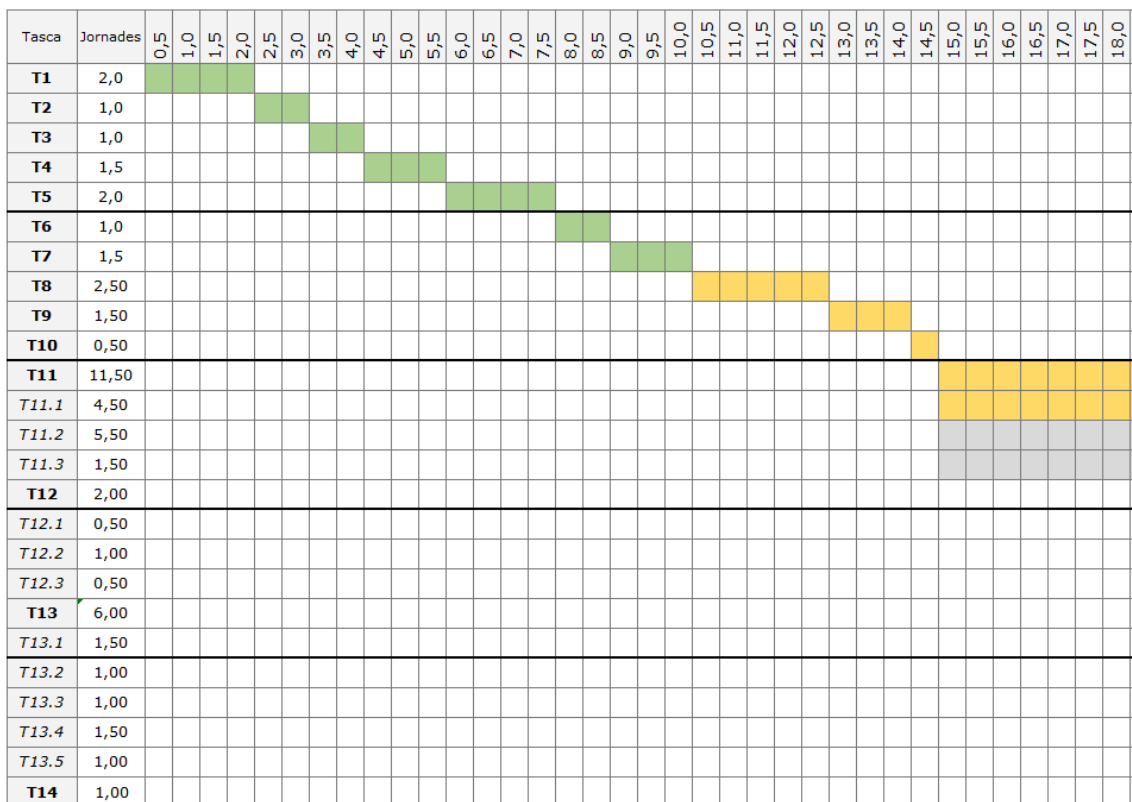


Figura 6-1 Primera part del diagrama de Gantt de planificació

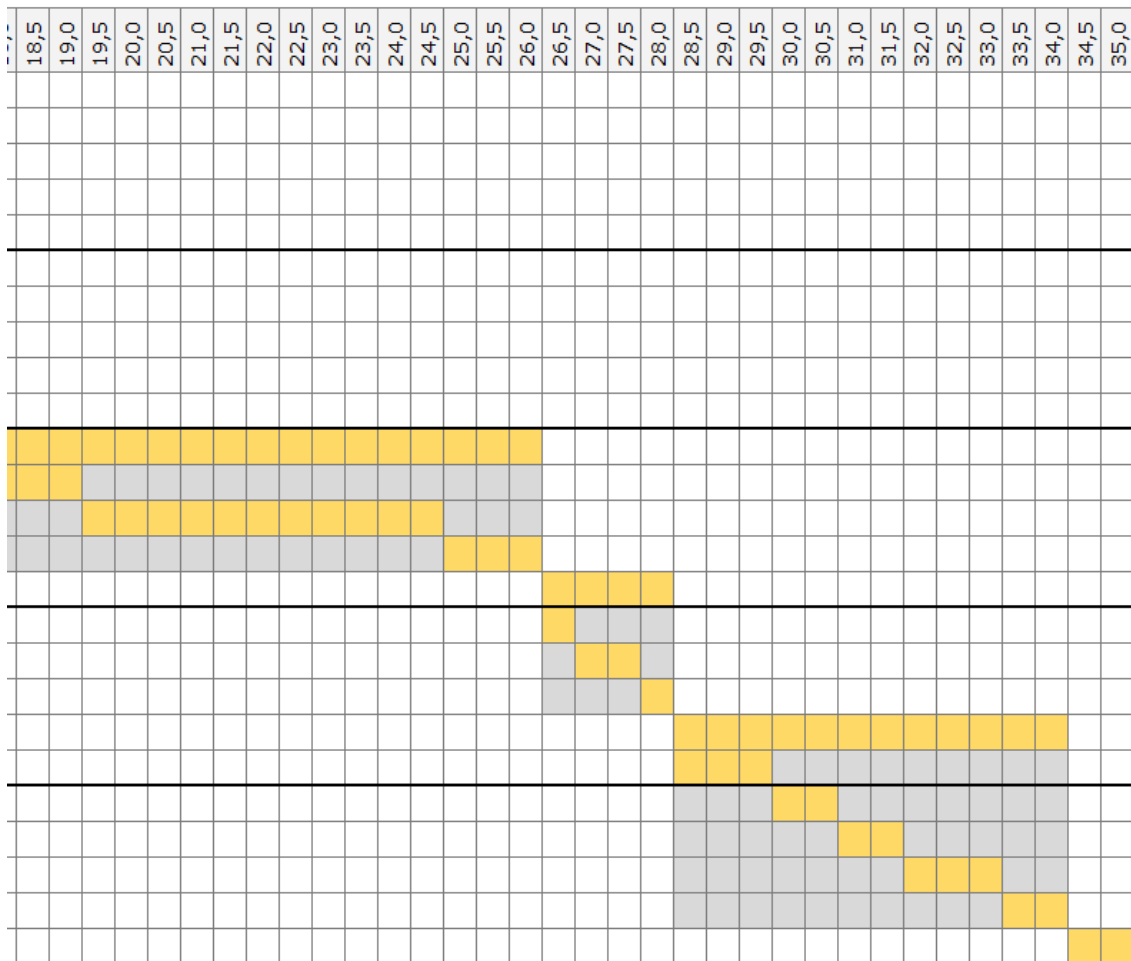


Figura 6-2 Segona part del diagrama de Gantt de planificació

Observant les figures 6-1 i 6-2 i enumerant cada tasca tenim les descripcions de totes les parts del projecte:

- T1 Especificació: fa referència al disseny global del sistema en el que es defineixen aplicacions, serveis, blocs i comunicacions del sistema.
- T2 Escollir SOM físic contactant amb proveïdors: fa referència a triar el circuit electrònic basant-se en recursos necessaris, espai, perifèrics incorporats, qualitat de la assistència tècnica i preu.
- T3 Escollir sistema operatiu pel SOM: un cop tenim SOM definit aquest punt defineix quin sistema operatiu s'utilitzarà entre els compatibles basant-se en necessitats de funcionament, estabilitat i qualitat del suport del fabricant i/o sistema operatiu.
- T4 Model de dades: fa referència a la tria de la tecnologia del model de dades basant-se en recursos necessaris per gestionar-se, possibilitats de disseny i flexibilitat. També inclou el disseny de les col·leccions i relacions que hi haurà.

- T5 Disseny de les operacions de la API i paràmetres: aquesta tasca es refereix a la definició de totes les operacions necessàries de la API de comunicació, la definició de com s'han de cridar aquestes operacions i quins paràmetres seran necessaris per poder dur-la a terme.
- T6 Disseny protocol de comunicació SÈRIE: implica la descripció del funcionament de la traducció de missatges per enviar-los al hardware i el significat de cada un dels bytes que s'enviaran. Aquí s'inclouen els controls de comunicació com ara el codi de redundància cíclica i els caràcters d'escapament.
- T7 Disseny d'entorn del núvol i processos que es duran a terme: fa referència a tot el que implica l'entorn al núvol com el sistema operatiu, ubicació, recursos, etc...
- T8 Configuració i compilació del sistema operatiu: fa referència a la parametrització i configuració de YoctoProject per obtenir el sistema que necessitem. Per fer-ho s'utilitzen paquets, capes i llibreries.
- T9 Configuració de serveis: aquesta tasca engloba el funcionament i generació del punt d'accés WiFi, la comprovació del port sèrie i el funcionament del servei MongoDB. Tot ha d'estar configurat per executar-se al iniciar el SOM (des del boot del sistema).
- T10 Creació del model de dades: fa referència a la creació de col·leccions, vistes i relacions de la base de dades. També s'hi inclouen els temes de seguretat i accés a les dades.
- T11 Programació de la aplicació de configuració (dividida en 3 tasques):
  - T11.1 Desenvolupament del Front End: fa referència al disseny i programació de la part gràfica juntament amb el comportament de la aplicació al navegador.
  - T11.2 Desenvolupament del Back End de la aplicació: fa referència a la programació de les funcions de crida a la base de dades i a la API, la lògica de la aplicació.
  - T11.3 Depuració: és la comprovació d'una correcta experiència d'usuari i funcionament per comprovar els missatges d'avís, alertes i solucionar errors.
- T12 Programació de la aplicació de client de demostració (dividida en 3 tasques):
  - T12.1 Desenvolupament del Front End: igual que en l'aplicació anterior, fa referència al disseny i programació de la part gràfica de l'aplicació.
  - T12.2 Desenvolupament del Back End de la aplicació: igual que en l'aplicació anterior, fa referència a la programació de les funcions de crida a la base de dades i a la API.
  - T12.3 Depuració: de nou, és la comprovació d'una correcta experiència d'usuari i funcionament de l'aplicació.

- T13 Programació de la lògica del Back End del sistema (dividida en 5 tasques):
  - T13.1 Implementació WebSocket i operacions: fa referència a la implementació de les operacions de la API amb funcions corresponents i els missatges de resposta que es necessitin.
  - T13.2 Validacions contra la base de dades: aquesta tasca implica a totes les consultes o modificacions de la base de dades des del servi de Back End.
  - T13.3 Càlculs CRC16 de comprovació de caràcters d'escapament: fa referència al controls de missatgeria de la comunicació amb el hardware per evitar problemes de soroll o errors de transmissió.
  - T13.4 Funcions de còpies incrementals al núvol: aquesta tasca engloba les funcions de comunicació amb el núvol i els controls de còpies de seguretat incrementals.
  - T13.5 Depuració: fa referència a la comprovació del funcionament com els missatges d'avís, el control d'errors i les alertes. També inclou la solució de possibles errors.
- T14 Proves i test: aquesta tasca fa referència a les proves del sistema complet per comprovar-ne el funcionament i buscar els possibles problemes o millores.

## 6.1 Avaluació de costos

Un cop analitzada la planificació, podem estimar els costos aproximats del projecte sencer. Partint d'un sou exemple de 25000€ bruts anuals, comptant amb un factor 1,3 per incloure costos de treballador i comptant amb unes 1750h laborables anuals tenim el següent cost aproximat per hora:

$$\frac{25000€ * 1,3}{1750h} = 18,57€/h$$

Si tenim en compte que el disseny implica unes 10 jornades i el desenvolupament unes altres 25 obtenim el següent preu total aproximat:

$$(35j) * \frac{8h}{1j} * \frac{18,57€}{1h} = 5200€$$

## 7 Test i avaluació

El test del sistema consisteix en fer 4 comprovacions per separat i una de conjunt. Les 4 comprovacions agruparan tots els blocs de disseny de manera que, si funcionen totes només caldrà provar la màquina sencera.

### 7.1 Comunicació SOM - placa de control del hardware

Aquest test es durà a terme amb el SOM i una segona màquina (en el meu cas un ordinador amb Windows 10) connectats per sèrie a través d'un port habilitat del sistema. Com que les proves es faran amb un convertidor USB a sèrie el port per defecte serà el ttyLP0.

La validació de funcionament serà enviar cada una de les possibles cadenes de bytes que corresponen a les funcions que haurà de satisfer la placa de control. Els resultats els podem observar al registre de la pròpia base de dades (així validem tot el tractament de dades i registre). També veurem, des d'un terminal, que les respostes arriben correctament amb el format i la estructura que toca.

A nivell de rendiment, executarem la funció "initProduct" 1000 vegades consecutivament en mode automàtic (en serveis de 3 segons) per observar el temps que perdem en comunicacions i gestions de processat. Com que els serveis es registren a la base de dades només haurem de veure el resultat en format taula tal com veiem a la figura següent.

ID producte	ID sessió	inici	final	duració en segons
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:00:00.000Z	2020-10-13T16:00:03.122Z	3,122
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:00:10.000Z	2020-10-13T16:00:13.117Z	3,117
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:00:20.000Z	2020-10-13T16:00:23.116Z	3,116
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:00:30.000Z	2020-10-13T16:00:33.109Z	3,109
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:00:40.000Z	2020-10-13T16:00:43.131Z	3,131
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:00:50.000Z	2020-10-13T16:00:53.107Z	3,107
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:01:00.000Z	2020-10-13T16:01:03.120Z	3,12
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:01:10.000Z	2020-10-13T16:01:13.109Z	3,109
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:01:20.000Z	2020-10-13T16:01:23.104Z	3,104
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:01:30.000Z	2020-10-13T16:01:33.129Z	3,129
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:01:40.000Z	2020-10-13T16:01:43.119Z	3,119
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:01:50.000Z	2020-10-13T16:01:53.110Z	3,11
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:02:00.000Z	2020-10-13T16:02:03.114Z	3,114
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:02:10.000Z	2020-10-13T16:02:13.125Z	3,125
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:02:20.000Z	2020-10-13T16:02:23.120Z	3,12
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:02:30.000Z	2020-10-13T16:02:33.129Z	3,129
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:02:40.000Z	2020-10-13T16:02:43.111Z	3,111
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:02:50.000Z	2020-10-13T16:02:53.110Z	3,11
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:03:00.000Z	2020-10-13T16:03:03.129Z	3,129
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:03:10.000Z	2020-10-13T16:03:13.112Z	3,112
2	acef8ba1-4a77-4295-9694-baaa5266308e	2020-10-13T16:03:20.000Z	2020-10-13T16:03:23.102Z	3,102

Figura 7-1 Mostra de la taula extreta de la col·lecció "serveis" després d'executar el test de 1000 mostres

Després de fer els càlculs amb l'ajuda d'Excel obtenim els resultats de la figura següent.

Temps d'execució total (segons)	3116,909
Temps promig de retard de gestió (segons)	0,116909

*Figura 7-2 Càlcul de temps total d'execució i del temps promig de càlcul i demora de comunicació*

Podem aproximar el temps de gestió del Back End comptant la demora de la comunicació via sèrie a 0,117 segons amb una velocitat de comunicació de 115200bps.

És interessant la mesura d'aquest paràmetre per ajustar correctament els consums de producte que es van calculant per cada registre de servei. Per la resta del sistema, la velocitat de comunicació no serà un factor crític si ens movem pels mateixos valors ja que són demores molt petites per la percepció humana.

## 7.2 Funcionament de la API

Per dur a terme aquesta avaluació, necessitarem la aplicació de configuració i l'aplicació de client de mostra.

Igual que en el cas anterior, el test consistirà en fer totes les peticions disponibles a la API per fer dues comprovacions:

- Back End: hem de comprovar que totes les peticions a la API arriben i es validen correctament al Back End. També haurem de comprovar que les peticions es tradueixen a les cadenes de bytes correctes, perquè tingui coherència amb el funcionament de la comunicació de l'apartat anterior.
- Aplicacions: per altra banda, haurem de comprovar que les crides obtenen resposta i es tracten des de les aplicacions correctament. També comprovarem que es fa un us de la API amb la funcionalitat correcte i que s'eviten tots els possibles errors de coherència com, per exemple, cridar un "endSession" sense haver llançat un "initSession" abans.

En aquest punt, comprovarem també la comunicació amb el núvol és correcte executant les crides a la càrrega de dades incremental. La validació la farem comprovant que les anotacions als documents pujats són correctes i, des de la base de dades del propi núvol, mirarem si apareix tota la informació.



### 7.3 Aplicació de configuració

Lligat amb una part del punt anterior, haurem de passar un test a la aplicació de configuració. Les proves consistiran en executar totes les funcions de cada pantalla per comprovar que es guarda correctament a la base de dades, que els refrescos són correctes i que l'aplicació no té errors inesperats o incontrolats.

Entrant en detall, anirem a comprovar totes les restriccions establertes en el disseny, que podrien provocar un funcionament erroni del hardware o una mala experiència d'usuari.

Llistant els punts crítics tenim:

- Dades de configuració: comprovarem que les dades de configuració i comunicació es gestionen i s'actualitzen correctament.
- Gestió de seguretat: comprovarem que els permisos per perfils s'apliquen correctament i que es poden gestionar sense problema.
- Gestió de productes: comprovarem que els productes es poden modificar i actualitzar correctament amb les restriccions correctes (no es poden assignar dos productes a una ubicació concreta, els camps obligatoris s'omplen correctament, etc...)
- Auditoria: en aquest punt, simplement comprovarem que es va generant correctament el registre. És una pantalla de lectura.
- Prova de producte: aquesta pantalla s'haurà provat al punt anterior així que ja la tindrem validada ja que només implica una crida a la API generada pel Back End.
- Informació del sistema: simplement comprovar que es visualitza correctament.
- Dinàmica global: per últim haurem de comprovar que els moviments de pestanyes són correctes i que els refrescos quan es guardin els paràmetres funcionen correctament.

### 7.4 Aplicació de client

Aquest apartat és el més senzill ja que durant la validació de la API haurem comprovat tot el funcionament. Simplement comprovarem que es visualitza tot correctament i tota la funcionalitat està disponible.

## 7.5 Conjunt de serveis i sistema

Aquest punt és un dels més crítics i importants ja que, si no funcionen correctament l'inici del sistema i els serveis, la resta no funcionarà tampoc.

Les comprovacions a fer, però, són simples. Partint de la màquina apagada i sense alimentació, la connectem a la corrent i la iniciem. Seguidament, comprovarem que després de 2 min (en principi triga menys però per assegurar que ha acabat el procés d'inici) la llista de serveis estan aixecats i funcionant:

- Punt d'accés WiFi: comprovar que la xarxa WiFi generada pel SOM està disponible, permet connexions i permet comunicació.
- MongoDB: provarem de connectar al servei de base de dades des de fora per validar que està aixecat.
- Aplicacions i Back End: comprovarem que estan disponibles totes les aplicacions llançant alguna instrucció i provarem la API per veure si el Back End està aixecat.

Si validem tots aquests punts a consciència ja només ens faltaria fer les proves de hardware i el funcionament de la màquina a punt per anar a instal·lar-la en un punt de venda.

## 8 Conclusions i treball futur

Després de dissenyar i desenvolupar tot l'entorn d'aquest sistema he pogut tocar un gran ventall de tecnologies noves i comunicacions. Un projecte d'aquestes dimensions et fa veure els punts crítics dels sistemes i la importància de la sincronització entre dispositius.

També, parlant de sistemes integrats, he vist que hi ha un alt nivell de complexitat en la optimització de recursos físics ja que la mida reduïda encareix molt el dispositiu si hi volem un hardware molt potent. Sorprenentment, vaig haver d'invertir una gran quantitat de temps en l'elecció, configuració i compilació del sistema operatiu precisament per aquest motiu.

A part de les complicacions tècniques el projecte m'ha requerit un nivell de planificació molt alt en el disseny, sobretot en tenir controlades totes les restriccions, controls, missatges, errors i problemes que podien sorgir.

Al llarg de la memòria he fet alguna referència a treball futur per millorar el sistema com a producte per poder ser implementat. Les millores i treballs futurs les dividiré en 2 grups: necessàries per posar el sistema en producció i millores del sistema.

### 8.1 Necessàries per tirar endavant el producte

Bàsicament fa referència a mesures de seguretat entre les quals trobem validació entre sistema i núvol amb certificat digital i/o comprovar comunicacions per la xarxa amb adreces IP públiques validades.

També t'ha d'incloure, encara que sigui més a nivell físic i de circuits, un procés de certificació d'emissions i funcionament per comprovar que és el sistema emet radiacions dins dels marges segurs i legals. Passant certificació evitaríem possibles problemes o incidències amb el producte.

### 8.2 Millores del sistema

Seria interessant aplicar un parell de millores tecnològiques i desenvolupar més infraestructura de software.

A nivell de tecnologies, considero interessant la integració d'un mòdul 4G/5G amb una targeta SIM per permetre la comunicació amb el núvol sense necessitat de xarxa física. En algunes instal·lacions concretes com aeroports o similars tenen restriccions de comunicacions amb l'exterior de manera que un mòdul de dades ens resoldria el problema sense haver de buscar alternatives de funcionament.

Per altra banda, ja que el SOM que s'ha triat ho porta incorporat, crec que seria interessant generar la API a través d'una connexió Bluetooth per permetre desenvolupar aplicacions natives per tabletas o smartphones còmodament al client que ho desitgés.

Per últim, trobaria interessant desenvolupar un grup de panells de control al provi servidor del núvol que ens fessin un anàlisi gràfic de les dades. Al mercat hi ha moltes eines per analitzar dades i donar-ne un grup d'informes fet podria ser atractiu per clients. També es podria optar modelar les dades del núvol (considero que es necessitaria un volum important de dades previ) per Machine learning, que ens proporcionin dades rellevants que ens puguin ajudar optimitzar les vendes dels productes.

## 9 Bibliografia

- GitHub, Inc. (2020). *GitHub*. Recollit de <https://github.com/>
- IPAR Vending. (31 / 1 / 2018). *IPAR Vending*. Recollit de <https://iparvendinggroup.com/tipos-maquinas-vending/>
- nuxtjs.org. (sense data). *NUXT.js*. Recollit de <https://nuxtjs.org/>
- Sánchez, J. (2018). *pinterest.es*. Recollit de <https://www.pinterest.es/pin/845339792537531572/>
- stackoverflow.com. (sense data). *Stackoverflow*. Recollit de <https://stackoverflow.com/>
- vencafesa. (30 / 7 / 2019). *vencafesa.com*. Recollit de <https://www.vencafesa.com/images/1.cafes.png>
- W3Schools. (sense data). *W3Schools*, Suport al desenvolupament. Recollit de <https://www.w3schools.com/>
- www.digi.com. (2020). *Digi Embedded documentation*. Recollit de [https://www.digi.com/resources/documentation/digidocs/embedded/dey/2.6/cc8x/yocto-gs\\_index.html](https://www.digi.com/resources/documentation/digidocs/embedded/dey/2.6/cc8x/yocto-gs_index.html)
- www.digi.com. (8 / 2020). *Hardware Reference Manual*. Recollit de <https://www.digi.com/resources/documentation/digidocs/pdfs/90002295.pdf>

## 10 Agraïments

En primer lloc m'agradaria agrair al tutor del treball, al Manel, la orientació, les facilitats i la flexibilitat a l'hora de desenvolupar al projecte en un any complicat com aquest.

En segon lloc voldria agrair el suport i els recursos de hardware que m'ha proporcionat la empresa on treballa, *DigiProces, S.A.*

Per últim, voldria agrair als companys del departament d'I+D de l'empresa pel suport sobretot a nivell de comunicacions de hardware.