

---

# **Sistema de Seguretat i Confort Ferroviari (SSCF)**

---

**Jordi Carbonell i Trilla**  
**Enginyeria Tècnica en Informàtica de Sistemes**

**Consultor: Jordi Bécares Ferrés**

Data Lliurament: 03/01/2012

*A la Gemma per haver-me animat a reprendre la carrera i per la paciència i el suport de tots aquets anys.*

*A tots els amics que m'han donat un cop de mà quan més ho he necessitat.*

*A la Neula, companya infatigable que ha aguantat al meu costat moltes hores d'estudi.*

### RESUM

El present treball té com a objectiu desenvolupar un sistema que permeti portar un registre per posteriorment consultar-lo dels diferents paràmetres de confort i seguretat que ofereixen els vagons d'un comboi ferroviari als seus usuaris.

Per dur a terme aquest projecte s'han utilitzat uns dispositius anomenats motes (pols intel·ligent o anglès: smartdust) que utilitzant el sistema operatiu TinyOS i el llenguatge de programació nesC proporcionen una xarxa de sensors sense fils, els quals poden captar diversos paràmetres de l'entorn i enviar-los via aèria aprofitant la xarxa en malla que són capaços de crear aquests petits dispositius.

Utilitzant tots els sensors i característiques que ens ofereix la mota obtenim la temperatura, lluminositat, polsador d'emergència, obertura i tancament de la porta de cada vagó. Per visualitzar aquesta informació s'han desenvolupat dues versions de programari en llenguatge Java. Una versió permet monitoritzar en temps real i l'altra, en diferit, els diferents paràmetres que es van rebre a través de les motes. Totes les dades rebudes en qualsevol de les dues versions del programari són emmagatzemades en una base de dades relacional i oberta que permet el seu accés de forma ràpida i fàcil a tota la informació per tal que l'anàlisi estadístic no sigui exclusiu d'aquest projecte, sinó que tercers puguin desenvolupar sistemes per analitzar les dades emmagatzemades.

Paraules Clau: TinyOS, nesC, confort, ferroviari, sensors, comunicació sense fils, USB, node.

Àrea TFC: Sistemes Encastats

## ÍNDIX DE CONTINGUTS

|  |    |
|--|----|
| 1.- INTRODUCCIÓ .....  | 8  |
| 1.1.- Justificació .....   | 8  |
| 1.2.- Descripció .....   | 8  |
| 1.3.- Objectius .....  | 9  |
| 1.4.- Enfocament.....  | 10 |
| 1.5.- Planificació.....  | 11 |
| 1.6.- Recursos emprats .....                                     | 13 |
| 1.6.1.- Maquinari .....  | 13 |
| 1.6.2.- Programari .....   | 13 |
| 1.7.- Productes obtinguts.....                                   | 14 |
| 1.7.1.- Desenvolupats .....                                      | 14 |
| 1.7.2.- No desenvolupats: .....                                  | 14 |
| 1.8.- Descripció altres capítols.....                            | 15 |
| 2.- Antecedents .....  | 16 |
| 2.1.- Estat de l'art.....  | 16 |
| 2.2.- Estudi de mercat.....                                      | 20 |
| 3.- Descripció funcional .....                                   | 23 |
| 3.1.- Sistema total .....  | 23 |
| 3.1.a.- Diagrama de blocs de l'aplicació .....                   | 23 |
| 3.1.b.- Xarxa.....   | 23 |
| 3.1.c.- Inter actuació dels diferents objectes del sistema ..... | 24 |
| 3.2.- PC.....  | 24 |
| 3.2.a.- Diagrama de blocs de l'aplicació .....                   | 24 |
| 3.2.b.- Xarxa.....   | 25 |
| 3.2.c.- Inter actuació dels diferents objectes del sistema ..... | 25 |
| 3.3.- Mota.....  | 25 |
| 3.3.a.- Diagrama de blocs de l'aplicació .....                   | 26 |
| 3.3.b.- Xarxa.....   | 27 |
| 3.3.c.- Interactuació dels diferents objectes del sistema .....  | 27 |

|  |    |
|--|----|
| 4.- Descripció detallada.....          | 28 |
| 4.1.- Sistema total.....               | 28 |
| 4.2.- PC.....                          | 29 |
| 4.2.1.- PC_CAB.....                    | 29 |
| 4.2.2.- PC_EST.....                    | 31 |
| 4.2.3.- Altres.....                    | 32 |
| 4.3.- Mota.....                        | 34 |
| 4.3.1.- MOTA_VAG.....                  | 34 |
| 4.3.2.- MOTA_CAB.....                  | 36 |
| 4.3.3.- MOTA_EST.....                  | 36 |
| 4.3.4.- Gestor Base de Dades.....      | 36 |
| 4.3.5.- Estructura dels missatges..... | 37 |
| 5.- Viabilitat tècnica.....            | 39 |
| 5.1.- Limitacions.....                 | 39 |
| 5.2.- Punts forts.....                 | 39 |
| 6.- Valoració econòmica.....           | 40 |
| 6.1.- Pressupost.....                  | 40 |
| 6.1.1.- Programari.....                | 40 |
| 6.1.2.- Maquinari.....                 | 40 |
| 6.1.3.- Manteniment.....               | 41 |
| 7.- Conclusions.....                   | 42 |
| 7.1.- Conclusions.....                 | 42 |
| 7.2.- Proposta de millores.....        | 43 |
| 7.3.- Autoavaluació.....               | 43 |
| 8.- Glossari.....                      | 45 |
| 9.- Bibliografia.....                  | 47 |
| 10.- Annexos.....                      | 48 |
| 10.1.- Manual d'usuari.....            | 48 |
| 10.1.1.- PC CAB.....                   | 48 |
| 10.1.2.- PC_EST.....                   | 50 |

|                                    |    |
|------------------------------------|----|
| 10.2.- Execució i compilació ..... | 51 |
| 10.2.1.- MOTA_VAG .....            | 51 |
| 10.2.1.- MOTA_CAB .....            | 51 |
| 10.2.1.- MOTA_EST.....             | 51 |
| 10.2.3.- PC_CAB.....               | 53 |
| 10.2.4.- PC_EST .....              | 53 |

## ÍNDEX DE FIGURES

|  |    |
|--|----|
| Il·lustració 1:Tren arribant a l'estació - activació motes - .....                 | 9  |
| Il·lustració 2:Diagrama de Gantt -Planificació inicial- .....                      | 11 |
| Il·lustració 3: Diagrama de Gantt -Planificació final- .....                       | 12 |
| Il·lustració 4: Processador Intel Core 2 Quand .....                               | 13 |
| Il·lustració 5: Mota COU_1_2 24 A2 .....   | 13 |
| Il·lustració 6: Iman .....   | 13 |
| Il·lustració 7 Estructura mota.....  | 20 |
| Il·lustració 8: sistema climatització i compensació de pressió .....               | 20 |
| Il·lustració 9: Sensors fotoelèctrics de barrera.....                              | 21 |
| Il·lustració 10: maneta d'emergència i interfonia.....                             | 21 |
| Il·lustració 11: Tren arribant a l'estació -indicació de les motes- .....          | 23 |
| Il·lustració 12: Diagrama de blocs resumit de tota l'aplicació .....               | 23 |
| Il·lustració 13: Diagrama de blocs del PC Cabina.....                              | 24 |
| Il·lustració 14: Diagrama de blocs del PC Estació .....                            | 24 |
| Il·lustració 15: Diagrama de blocs de la MOTA VAGÓ .....                           | 26 |
| Il·lustració 16: Diagrama de blocs de la MOTA CABINA .....                         | 26 |
| Il·lustració 17: Diagrama de blocs de la MOTA ESTACIÓ.....                         | 26 |
| Il·lustració 18: Situació dels diferents elements al comboi .....                  | 27 |
| Il·lustració 19: Diagrama detallat de funcionament de projecte. ....               | 29 |
| Il·lustració 20: Diagrama de flux del PC de cabina .....                           | 30 |
| Il·lustració 21: Diagrama de flux del PC de l'estació.....                         | 31 |
| Il·lustració 22: Control d'errors de transmissió.....                              | 32 |
| Il·lustració 23: Diagrama de flux de lectura dels sensors a la MOTA_VAG.....       | 34 |
| Il·lustració 24: Diagrama de flux del polsador d'alarma de la MOTA_VAG .....       | 35 |
| Il·lustració 25: Diagrama de flux del sensor Hall de la MOTA_VAG .....             | 35 |
| Il·lustració 26: Pantalla principal PC Cabina .....                                | 48 |
| Il·lustració 27: Pantalla principal PC Cabina -Alarma polsador activada-.....      | 49 |
| Il·lustració 28: Pantalla principal PC Cabina -Alarma il·luminació activada- ..... | 49 |
| Il·lustració 29: Pantalla principal PC Cabina -Anunci enviant dades-.....          | 50 |
| Il·lustració 30: Pantalla principal PC Estació -Anunci rebent dades- .....         | 50 |

## **1.- INTRODUCCIÓ**

### **1.1.- Justificació**

Avui en dia hi ha la necessitat de garantir una certa comoditat i seguretat a les persones en qualsevol situació, lloc o estància. Oferir un alt grau de confort i tranquil·litat ja no és un tret diferencial d'algun servei o classe sinó que és una exigència de totes les persones, usuaris o clients, això obliga a controlar en tot moment els paràmetres de confort i seguretat. Utilitzant tots els materials que em proporciona la UOC TFC-Sistemes Encastats es vol dissenyar un sistema que doni resposta a aquestes necessitats.

Tots els usuaris de tren o metro saben que en funció de l'hora d'utilització d'aquests transports públics poden tenir una temperatura a l'interior massa elevada, com per exemple seria a les vuit del matí causada per l'aglomeració d'usuaris, o per contra arribar a passar fred com seria a les dotze del migdia quan els combois van pràcticament buits.

Cal destacar la importància de no només controlar la temperatura interior de confort sinó també el correcte funcionament de l'enllumenat, que totes les portes estiguin correctament tancades abans d'iniciar la marxa o també que l'usuari disposi d'un polsador d'alarma per si tingués alguna emergència.

Només portant un control i registre de totes les dades podem donar resposta o prendre les mesures oportunes per solucionar aquestes incidències, com per exemple augmentar la sortida d'aire fred en les hores de més afluència d'usuaris i disminuir-la en la de menys, o si en un vagó acostuma a fallar molt l'enllumenat revisar-ho, si en un comboi o una línia de tren/metro el polsador d'alarma es premut més vegades del que seria habitual, comprovar perquè es dona aquesta situació i trobar-hi solució.

És per aquests motius i per aprofitar tot el potencial de la mota/node que sorgeix la idea de portar un control i el seu corresponent registre d'incidències per tal d'augmentar el confort i seguretat dels viatges de tren/metro, tot i que es podria obrir a qualsevol mitjà de transport col·lectiu.

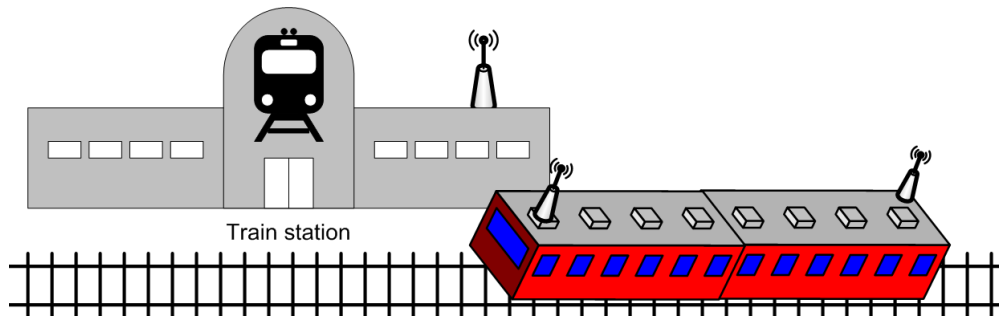
### **1.2.- Descripció**

Aquest projecte pretén establir un sistema de monitorització per registrar els diferents inconvenients de confort que els usuaris de transports ferroviaris poden estar exposats a causa d'una mala climatització o d'incidències remarcables en l'enllumenat. Aprofitant tot el potencial de les motes també podem controlar i oferir paràmetres de seguretat com són el correcte tancament de les portes d'accés als vagons així com un polsador d'alarma per si hi ha algun cas d'emergència.



El projecte desenvolupat no pretén ser un sistema autònom de climatització, enllumenat i seguretat, és a dir, les motes només recullen informació i la transmeten, per després poder ser analitzada, en cap cas una mota pren decisions preestablertes ni actua sobre cap sistema extern, per tant, la idea principal de ser un sistema de monitorització i alarma és la que s'ha dut a terme.

Cal remarcar que en aquest projecte s'ha desenvolupat un prototip de sistema de monitorització i que per tant la seva instal·lació i/o comercialització ha d'anar precedida de l'adaptació física als combois actuals així com d'una bateria exhaustiva de proves sobre el terreny.



Il·lustració 1: Tren arribant a l'estació - activació motes -

### 1.3.- Objectius

El sistema ha de funcionar automàticament i la informació s'ha de transmetre sense la necessitat d'intervenció de cap usuari. Aquestes dades s'han de mostrar i actualitzar automàticament a les pantalles establertes per aquest fi.

- Monitoritzar la temperatura de dins del vagó.
- Monitoritzar la il·luminació de dins del vagó.
- Monitoritzar l'estat de la porta del vagó.
- Controlar el polsador d'alarma.
- Enviament de dades entre els vagons i la cabina (MOTA\_VAG --> MOTA\_CAB).
- Enviament de dades entre la cabina i les estacions (MOTA\_CAB --> MOTA\_EST).
- Interfície gràfica i emmagatzematge d'informació al PC\_CAB i PC\_EST.

Per assegurar el correcte confort dels viatgers cal analitzar constantment la temperatura i la il·luminació de l'interior dels vagons. La seguretat s'ha de garantir posant a la disposició dels usuaris un polsador d'emergència que en cas de ser premut ha d'avisar immediatament al conductor i d'un sistema de control d'estat real de les portes per saber si estan obertes o tancades abans d'iniciar la marxa. S'ha de crear un sistema de comunicació entre les diferents motes perquè la informació es transmeti correctament i arribi a cadascun dels PC's, així tant el conductor com el cap d'estació com qualsevol altre responsable puguin veure i gestionar les dades, per fer-ho es crearan unes interfícies gràfiques adaptades a cada un d'aquets llocs.

### 1.4.- Enfocament

Quan s'inicia un projecte s'ha de comprovar de quins recursos es disposa i de les prestacions de que són capaços de proporcionar-nos, per això i abans de definir el sistema concret a desenvolupar s'ha de recopilar tota la informació possible de diferents fonts d'informació, que principalment són:

- Projectes anteriors de la UOC.
- Projectes anteriors d'altres universitats.
- Pàgines web específiques.
- Wikipedia.
- Manuals de referència dels recursos (plaques, sensors, unitats de control, etc...)

En aquest cas tenim com a recurs específic la mota, de la qual s'ha d'avaluar totes els seves prestacions i també les limitacions.

En la definició prèvia del projecte es marquen unes restriccions, no en el sentit de funcionament ni dels objectius que ha de complir el sistema, sinó que són restriccions del propi plantejament del projecte. En aquest cas, una de les fites autoimposades és que el projecte no hagués estat prèviament desenvolupat en els anys anteriors, perquè ja s'han desenvolupat la majoria de sistemes més clàssics i també d'altres amb una complexitat molt elevada.

Tot i concretar el projecte definint el projecte de forma clara els seus objectius i límits, hem de ser conscients que la falta d'experiència en l'entorn de desenvolupament (sistemes encastats, Java, etc...) i els seus components (TinyOS, nesC, motes, etc...) fa augmentar la probabilitat de trobar complicacions en fases posteriors i per tant hem de deixar marge pels possibles contratemps que puguin sorgir.

Una de les parts més importats d'un projecte és la planificació, que aquesta sigui clara, concisa i amb totes les parts i fases detallades i temporitzades correctament pot facilitar enormement la tasca del desenvolupament ja que sempre sabrem on som i on volem arribar i en quin temps s'ha de fer. Una planificació errònia pot forçar un desenvolupament ràpid en fases crítiques o una pèrdua de temps en altres

Per altra banda, tenim l'entorn de desenvolupament que ha d'estar curosament preparat per evitar que falti algun recurs en les fases de desenvolupament, cosa que provocaria l'endarreriment del projecte. A partir de la informació recollida en l'anàlisi i amb tot l'entorn de desenvolupament preparat es creen els diferents components, que en aquest cas són:

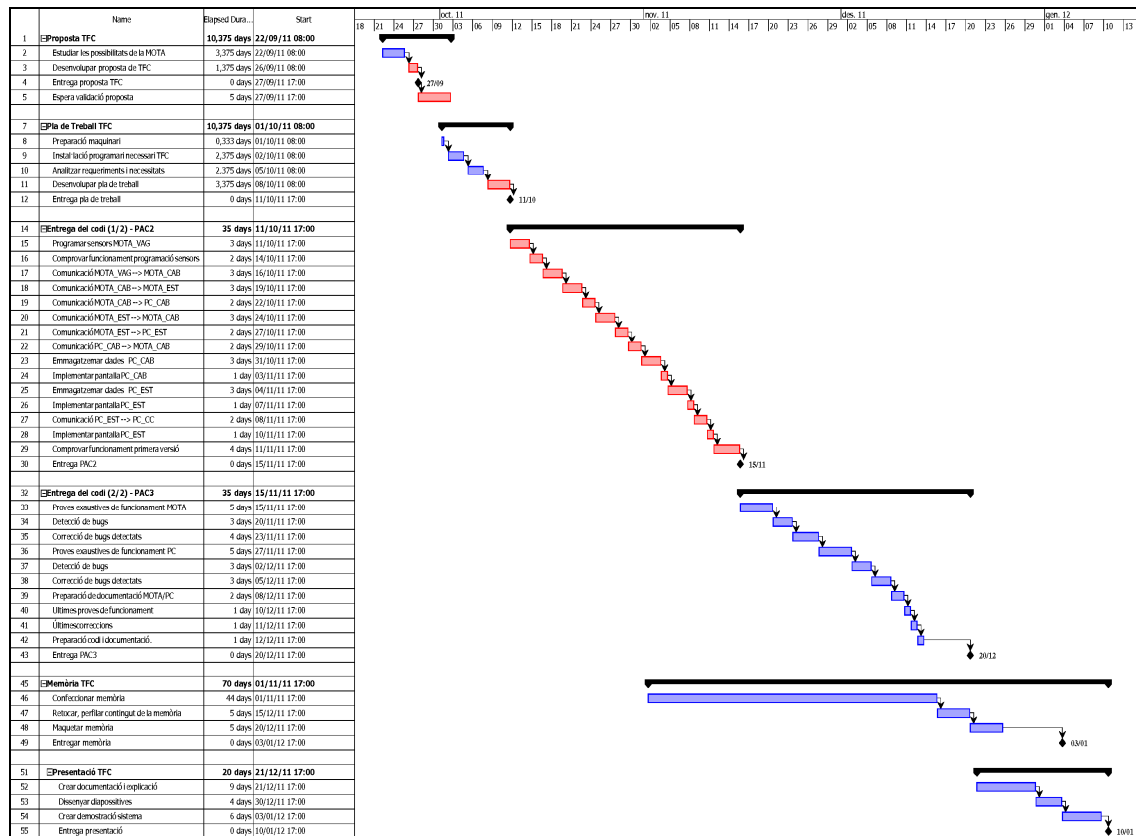
- MOTA\_VAG → Col·locat a cada porta dels vagons.
- PC\_CAB → Col·locat a la cabina del conductor.
- PC\_EST → Col·locat a cada estació.

Tot i tenir la intenció d'utilitzar un sistema de desenvolupament en cascada, és a dir, no començar una fase fins que no s'acabi la seva immediata anterior, a la pràctica va resultar impossible ja que el desconeixement del sistema va fer que l'anàlisi no fos del tot correcte i ocasionà que a la part final s'haguessin de modificar algunes seccions de fases anteriors. Això va causar que no es pogués seguir la planificació i no s'entreguessin tots els mòduls previstos en la primera entrega, però es va corregir la planificació aplicant algunes reduccions per tal de complir amb els altres terminis fixats en les següents entregues.

El desenvolupament de la memòria es va realitzar conjuntament mentre estava desenvolupant tota l'aplicació. A mesura que anava desenvolupant parts del sistema anava apuntat les idees i recursos més importants agrupats per mòdul desenvolupat, sense utilitzar l'estructura definida per a la presentació final. D'aquesta manera disposava d'una valuosa informació per desenvolupar els apartats següents i poder-los enllaçar d'una forma més fàcil.

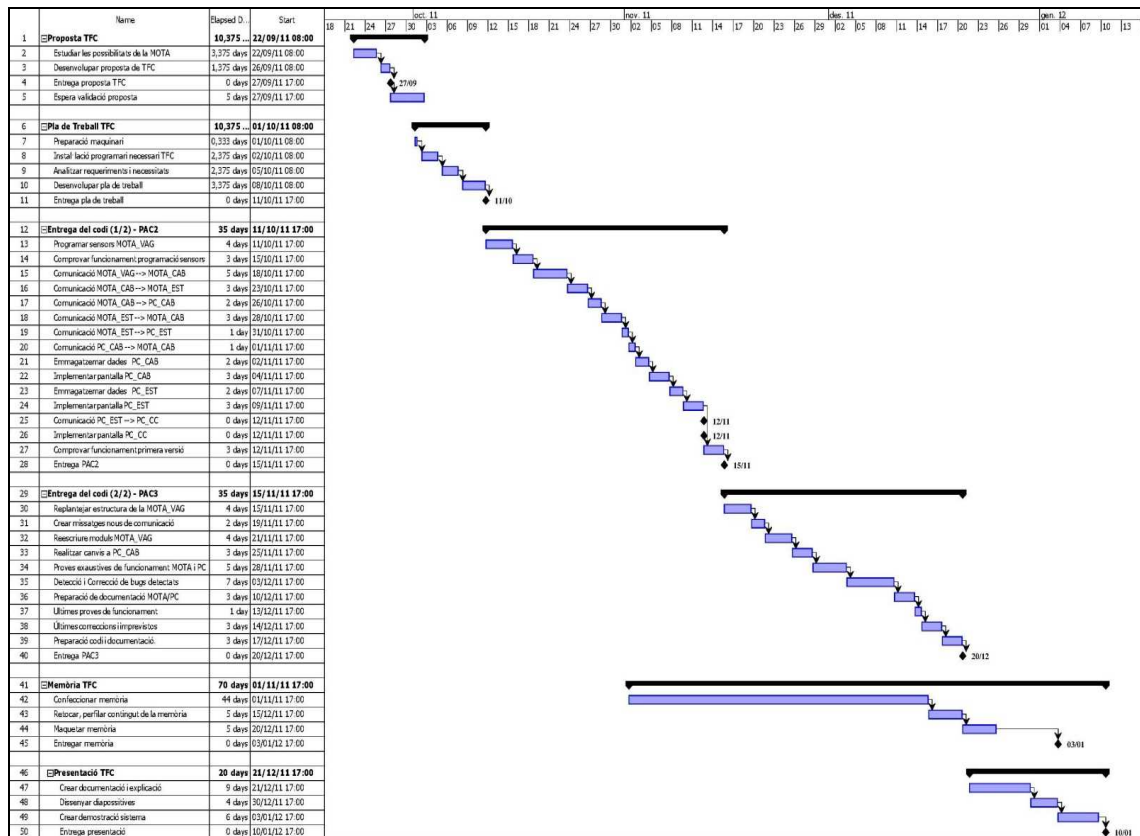
### 1.5.- Planificació

#### Planificació inicial



II-Il·lustració 2: Diagrama de Gantt -Planificació inicial-

Planificació final



II-lustració 3: Diagrama de Gantt -Planificació final-

Justificació

Els punts de variació temporal més importants entre la planificació inicial i la real corresponen al desenvolupament inicial de les motes. En un principi es va suposar que la corba d'aprenentatge per poder programar correctament les motes era menor a la que va ésser en realitat ja que tot i tenir certa experiència el la programació en llenguatge C, les diferències entre aquest i el nesC eren més àmplies de les suposades.

Per altra part, també destacar que el temps extra que es va dedicar en l'aprenentatge de les motes es va poder recuperar gràcies a suprimir l'última part del desenvolupament del programari de gestió de les dades recollides pels sensors de les motes, ja que l'objectiu principal d'aquest és la programació i gestió de les motes i la comunicació d'aquesta amb el programari, no en el tractament i gestió de les dades que proporcionen.

## **1.6.- Recursos emprats**

Per al desenvolupament d'aquest projecte es podien utilitzar un innumerable nombre de recursos, però els escollits per dur-lo a terme són els següents:

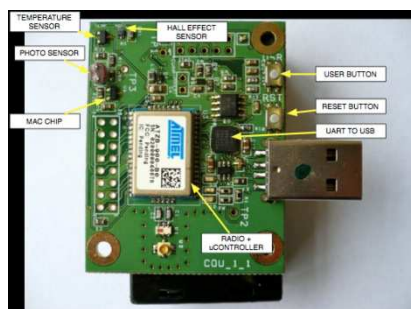
### **1.6.1.- Maquinari**

- Ordinador 1: Intel Core 2 Quad Q6600 a 2,4Ghz, 4Gb RAM amb Windows 7 64 bits



Il·lustració 4: Processador Intel Core 2 Quad

- Mota COU\_1\_2 24 A2



Il·lustració 5: Mota COU\_1\_2 24 A2

- Iman



Il·lustració 6: Iman

### **1.6.2.- Programari**

- Windows 7 64 bits

- MS Office 2007

- OpenProj 1.4

### - Eclipse (Helios)

- Plugin Yeti 2 - TinyOs 2
- Java 1.6.0\_25
- Distribució linux Osian
- TinyOS 2.1.1
- Meshprog 0.1.2
- SerialForwarder
- Java 1.6.0\_20
- Eclipse (Helios)
  - Plugin Yeti 2 - TinyOs 2

## 1.7.- Productes obtinguts

El desenvolupament del projecte ha donat com a resultat la creació de certs productes i se n'han utilitzat d'altres desenvolupats per tercers:

### 1.7.1.- Desenvolupats

MOTA\_VAG: Aplicatiu per a la mota situada en cada porta dels vagons del comboi, s'utilitza per obtenir les dades de temperatura, lluminositat, obertura/tancament de portes i polsador d'alarma.

PC\_CAB: Programa desenvolupat utilitzant el llenguatge de programació Java i situat a la cabina del comboi ferroviari que proporciona informació al conductor i a més a més s'utilitza per emmagatzemar i transferir tota la informació recollida per les MOTA\_VAG.

PC\_EST: Programa desenvolupat utilitzant el llenguatge de programació Java i situat a cada estació que rep les dades de cada PC\_CAB que passi, és a dir, cada comboi ferroviari que passa per l'estació li transfereix les dades que te emmagatzemades i que encara no s'han transferit a cap estació.

### 1.7.2.- No desenvolupats:

BaseStation: Aplicatiu per a la mota que tot i no estar desenvolupat en aquest projecte si que s'utilitza per transferir/rebre dades des d'un PC als altres ordinador o motes. La funcionalitat de l'aplicatiu és que quan es rep una dada a través de l'antena sense fils(via ràdio), s'activa el led de color taronja i es queda encès fins que en rebem un altre pel mateix medi. Aquesta dada rebuda és retransmesa cap al port sèrie (USB).

Quan es rep una dada a través del port sèrie (USB), s'activa el led de color vermell i es queda encès fins que en rebem un altre pel mateix medi. Aquesta dada rebuda és retransmesa cap l'antena sense fils(via ràdio).

### 1.8.- Descripció altres capítols

A continuació hi ha una breu pinzellada sobre els diferents capítols de que consta aquesta memòria, tot indicant les parts més rellevants de cadascun.

#### Capítol 2:

Es mostra un recull de tots els dispositius similars a la nostra mota, les prestacions i la tecnologia emprada per cadascun així com els diferents sensors del mercat i el seu preu. També es mostra el resultat de la cerca d'un producte de similars condicions i les avantatges que ofereix el projecte desenvolupat.

#### Capítol 3:

Breu descripció funcional del projecte desenvolupat, tan des d'una visió general com des d'una visió específica de cada part. S'acompanya d'il·lustracions i de diagrames per fer més entenedor el seu funcionament.

#### Capítol 4:

Descripció detallada, funcional i tècnica de tot el projecte desenvolupat i de cadascuna de les seves parts, així com les decisions més importants que s'han hagut de prendre. També s'inclou una explicació dels sistemes desenvolupats més importants acompanyat per il·lustracions i diagrames del seu funcionament.

#### Capítol 6:

Explicació detallada de la viabilitat del projecte, mancances que puguin ser esmenades abans de la seva comercialització i remarcar els punts forts del projecte.

#### Capítol 7:

Descripció del cost total del projecte, tan de l'apartat no tangible com és el programari i tot el que envolta el seu desenvolupament com del maquinari mínim perquè funcioni el sistema desenvolupat.

#### Capítol 8:

Explicació detallada dels objectius del projecte aclarint els motius dels que no s'han pogut desenvolupar. Un llistat de millores que es podrien incorporar així com d'una avaluació de tot el projecte.

#### Capítol 9:

Llista de termes que per la seva especificitat no són al vocabulari d'ús habitual.

#### Capítol 10:

Fonts d'informació d'on s'ha extret tota la informació per desenvolupar el projecte.

#### Capítol 11:

Altres tipus de documentació com són els manual d'usuari, les instruccions per compilar, instal·lar i executar el projecte.

## 2.- Antecedents

### 2.1.- Estat de l'art

La mota de què disposem per realitzar el TFC engloba un conjunt de diferents tecnologies com són sensors, microcontroladors, comunicació sense fils, leds, etc... per tant primer de tot descriurem el sistema de què disposem i els seus components:

Unitat de control:

Microcontrolador: Atmel Atmega 1281V

Transceptor: AT86RF230 RF (comunicació sense fils: protocol IEEE 802.15.4)

Sensors:

- Temperatura: Microchip MCP9700/9701 (3-Pin SOT-23)
- Lluminositat: Advanced Photonix PDV-P9003-1
- Hall: Rohm BU52001GUL

Altres:

- 2 polsadors
- 3 leds
- Connector USB

Actualment al mercat podem trobar infinitat de productes similars amb una gran variabilitat de preus com podem veure a continuació:

| Tipus                      | Marca             | Model           | Preu aprox. |
|----------------------------|-------------------|-----------------|-------------|
| <b>Sensor Temperatura</b>  | SGS-Thomson       | LM335           | 1,90€       |
|                            | Melexis           | MLX90614        | 15,90€      |
|                            | Maxim             | DS18B20         | 4,90€       |
| <b>Sensor Lluminositat</b> | Metal CdS         | GL5528          | 1,10€       |
|                            | CEBEK             | C-2795          | 0,53€       |
| <b>Sensor Hall</b>         | Honeywell         | SS443A          | 5,13€       |
|                            | Allegro           | A3425LK         | 3,99€       |
| <b>Transceptor</b>         | Aurel             | XTR CYP         | 22,00€      |
|                            | MicroChip         | MRF24J40MA-I/RM | 8,43€       |
|                            | Atmel             | AT86RF212-ZU    | 5,78€       |
| <b>uC</b>                  | MicroChip         | PIC18LF8722     | 9,57€       |
|                            | Atmel             | ATMEGA128L      | 25,48€      |
|                            | Texas Instruments | MSP430F2617     | 20,52€      |
|                            |                   |                 |             |



A la taula anterior podem veure varis tipus de sensors que analitzem seguidament:

Temperatura, n'hi ha dos de digitals (MLX90614; DS18B20) i un de resistència (LM335), és a dir, aplica més o menys resistència a la corrent que passa per ell en funció de la temperatura que detecta. Tots tres tenen un rang de funcionament similar que va des dels -40°C fins als 100°C o 125°C. Cal mencionar que els sensors digitals són més cars.

Lluminositat, els dos sensors actuen per resistència tot i que la resistència lumínica és deu vegades més gran en el sensor C-2795 que en el GL5528 ambdós disminueixen en la mateixa proporció. La temperatura de funcionament és de -30°C a +75°C


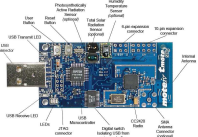

Hall, els dos sensors tenen com a objectiu detectar un camp magnètic hi ho fan de manera similar, la diferència entre els dos a part del voltatge de funcionament que el SS443A va des de 3,8V fins a 30V i el sensor A3425LK va des de 3,3V fins a 18V és que aquest últim incorpora dos canals de sortida on cada canal compara independentment de l'altre l'efecte Hall.

Transceptor, els tres transceptors tot i compartir freqüències o estàndards són incompatibles entre sí. El XTR CYP funciona a 2,4 Ghz i el sistema de modulació utilitzat és GFSK que és una versió millorada de la modulació per freqüència (FSK), i ofereix una velocitat de transferència de 62,5 kbits/s a una distància de 50m. El transceptor MRF24J40MA segueix l'estàndard IEEE 802.15.4 /ZigBee treballant a una freqüència de 2,4 Ghz que tot i funcionar a la mateixa freqüència que l'anterior no són compatibles a causa de la diferent modulació que utilitzen, la velocitat de transferència és de 250kbits/s amb una cobertura d'uns 122 metres. Per últim, tenim AT86RF212-ZU que compleix l'estàndard IEEE 802.15.4 i pot utilitzar el mateix sistema de modulació que l'anterior transceptor, però no són compatibles ja que la freqüència de treball és de 863 a 870 Mhz i no de 2,4 Ghz. la velocitat de transmissió és de fins a 250kbits/s.

µC, tenim una mostra de diferents tecnologies i per tant diferents prestacions, el PIC18LF8722 ens ofereix un xip de baix cost amb arquitectura de 8 bits a una velocitat de 40Mhz i dotat amb interfícies EUSART, I2C, PSP, SPI, per altra banda, tenim el xip ATMEGA128L també de 8 bits a una velocitat de funcionament de 8Mhz i ofereix les interfícies JTAG, SPI, USART tot i que no és tan econòmic, per últim tenim el xip MSP430F2617 de 16 bits i una velocitat de 16Mhz oferint les interfícies I2C, IrDA, JTAG, SPI, UART, USCI. De tots tres el millor dotat de memòria és el PIC amb 128K per programar i 3936K + 1024K per dades i també és el que te més entrades/sortides amb un total de 70.

Tal com podem comprovar la mota és una agrupació de diferents components que treballen conjuntament, cal destacar que la paraula motes prové de l'anglès (**remote sensing**), tot i que el nom inicial és (smart dust) que significa pols intel·ligent.

A continuació podem veure una taula amb un recull dels diferents fabricants i característiques de motes:

|                        | <b>BTnode rev3.2</b>  | <b>Tmote Sky</b>   | <b>Imote 2</b>  |
|------------------------|---|--|---|
|                        |  |  |  |
| <b>uC</b>              |   |  |   |
| Model                  | Atmel Atmega128L  | MSP430F  | PXA 271   |
| Arquitectura           | 8 bits  | 16 bits  | 32 bits   |
| Velocitat              | 7,3728 Mhz  | 8 MHz  | 13 Mhz - 416 Mhz  |
| <b>Memòria</b>         |   |  |   |
| Programació            | 128 kBytes  | 48 kBytes  | 256 kBytes  |
| Dades                  | 64 kBytes   | 10 kBytes  | 256 kBytes  |
| Externa                | SRAM 180 kBytes   | 1024 kBytes  | 32 Mbytes   |
| IO Extern              | 40  | 16   | 26  |
| <b>Comunicació</b>     |   |  |   |
| Model                  | Zeevo ZV4002 //<br>Chipcon CC1000   | TI CC2420  | TI CC2420   |
| Tecnologia             | Bluetooth /<br>UHF RF   | IEEE 802.15.4  | IEEE 802.15.4   |
| Freqüència             | 2,4 Ghz //<br>315 - 976 Mhz   | 2,4 Ghz  | 2,4 Ghz   |
| Velocitat              | 723,2 kbps //<br>38,4 kbps  | 250 kbps   | 250 kbps  |
| Distància<br>cobertura | 30m - 50m //<br>30m - 100 m   | 50m - 125m   | 50m - 125m  |
| <b>Sistema</b>         |   |  |   |
| Sistema<br>Operatiu    | BTnut, TinyOS   | Contiki, TinyOS,<br>SOS, MantisOS  | Linux, TinyOS,<br>SOS   |
| Programació            | C / nesC  | C / nesC   | Microsoft .NET<br>Micro, C, nesC  |
| <b>Consum</b>          |   |  |   |
| Volts                  | 0,5 - 4,4   | 2,1 - 3,6  | 3,2 - 4,5   |
| mA                     | 3 - 60  | 19,5 - 23  | 31 - 66   |
| <b>Dimensions</b>      |   |  |   |
| Ampl. x llarg.         | 33mm x 58mm   | 33mm X 66mm  | 36mm x 48mm   |
| <b>Sensors</b>         |   |  |   |
| Integrats              | No  | Temperatura,<br>humitat, lluminositat  | No  |
| Adicionals             | Si  | Si   | Si  |

A causa de la gran diversitat de les motes que hi ha al mercat i la impossibilitat d'estudiar-les totes, se n'ha extret les tres més rellevants i que utilitzen una tecnologia més diferenciada una de l'altra.

Les unitats de control són molt variades, es poden trobar des de les més clàssiques de 8 bits a uns 8Mhz de velocitat fins a les més potents de 32 bits a 416Mhz de velocitat passant per les de 16 bits. L'augment de bits s'ha de tenir en compte ja que està directament relacionat amb la memòria màxima que pot utilitzar la unitat de control, és a dir, a més bits més memòria podrà gestionar i de forma més ràpida.

Cal destacar tal i com veiem, que la majoria de motes utilitzen el protocol de comunicació IEEE 802.15.4 - ZigBee que proporciona nombrosos avantatges com oferir un consum elèctric reduït, permetre la tipologia de xarxa en malla i la seva facilíssima integració que permet crear un node amb molt poca electrònica. També podem trobar motes amb altres sistemes de comunicació com els basats en tipologia Bus ASK/FSK, Bluetooth,... i també diferents freqüències de funcionament com 433Mhz; 868Mhz; 916Mhz; 2,4Ghz, ... però l'estàndard que sembla imposar-se al mercat és el protocol IEEE 802.15.4

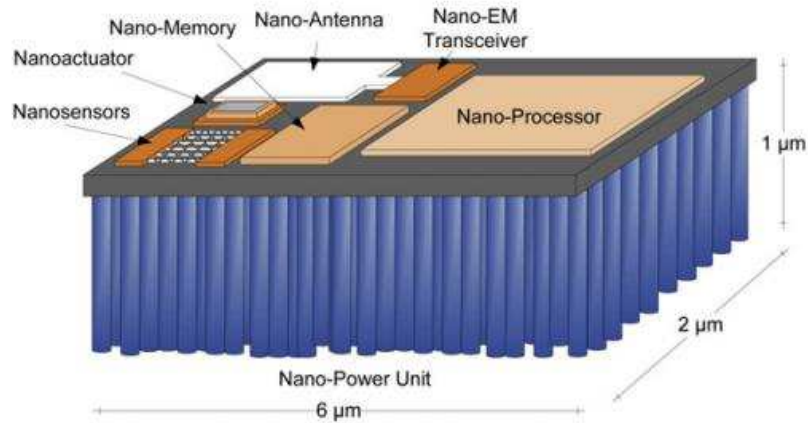
Si ens fixem en el sistema operatiu emprat es pot veure que n'hi ha una gran varietat com SOS Operating System, MantisOS, Microsoft .NET Micro, TinyOS, però també podem veure que els fabricants es decanten per utilitzar preferentment el TinyOS tot i que també en suporten d'altres.

Per altra banda tenim el llenguatge de programació que en aquest apartat predomina el llenguatge 'C' tot i que també trobem variants d'aquest com és el 'nesC' que també està molt acceptat i fins hi tot en trobem de propietaris com el Microsoft .NET Micro. La raó per la qual els fabricants de motes utilitzen preferentment el llenguatge C de programació es pel seu potencial i la mida dels programes compilats que genera. S'ha de tenir en compte que les motes tenen un maquinari molt limitat.

Si ens fixem en l'apartat elèctric tots els dispositius accepten aproximadament el mateix rang de voltatge cosa que no passa amb consum, on aquest i en funció de la tecnologia de comunicació o unitat de procés pot variar substancialment ja que utilitzant sistemes de comunicació no pensats exclusivament per a les motes o uC amb molta potència és disparen els mA necessaris per fer funcionar la mota.

Per últim tenim la quantitat de sensors que dur integrat la mota en la seva pròpia placa. Aquest apartat pot ser molt variant en funció del fabricant i del mercat al qual va dirigida, per aquest motiu només n'hi ha una que dugui sensors instal·lat. Però cal tenir en compte que totes permeten la implementació de varis tipologies d'entrada de dades, ja siguin sensors analògics, digitals, càmeres, etc...

Centrant-nos exclusivament en les motes podem veure que des dels seus inicis al 1998 on el Dr. Kristofer Pister de la Universitat de Califòrnia va desenvolupar els primers prototips fins avui en dia han evolucionat molt, no només en la seva mida que avui en dia poden ser com un gra de sorra sinó en el potencial de la unitat central, la distància de la comunicació sense fils, el nombre i tipus de sensors que pot incorporar, etc... cada dia són més petites i més potents.

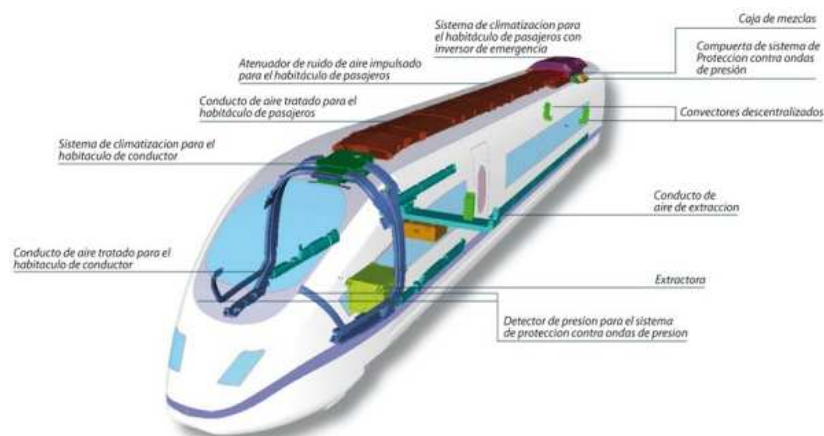


Il·lustració 7 Estructura mota

## 2.2.- Estudi de mercat

Actualment els mitjans de transport ferroviaris van equipats amb diferents mesures de confort i seguretat que funcionen de forma autònoma, és a dir, sense estar integrats en un sistema únic ni de gestió ni de supervisió.

Fins ara els sistemes de climatització funcionaven mitjançant un termòstat que regula la temperatura interior de cada vagó, però els sistemes més moderns ja no és limiten només a aquesta funció, sinó que també tenen en compte la quantitat de CO2 i la diferència de pressió que es crea quan un comboi entra en un túnel a gran velocitat i que contraresten activant els mecanismes necessaris per compensar-ho [8]. Tot i la incorporació de varis sistemes autònoms de funcionament i la creació de programari per gestionar-ho, aquest continua essent individual pel sistema de climatització, sense que estigui integrat amb cap altre sistema.



Il·lustració 8: sistema climatització i compensació de pressió

Tots els combois ferroviaris moderns no poden arrancar si una de les portes de qualsevol vagó resta oberta, per tan totes les portes han d'estar tancades abans de prosseguir o iniciar el viatge. Aquest control es realitza mitjançant uns sensors [9] instal·lats a cada porta d'accés i que indiquen al conductor de que totes les portes estan tancada o no, tot i que el maquinista amb els sistemes actuals ja rep l'estat de les portes, aquesta informació no es desa i per tan no es pot saber a posteriori si hi ha hagut algun contratemps.



Il·lustració 9: Sensors fotoelèctrics de barrera

També podem veure instal·lat en els combois més moderns un interfon al costat del fre d'emergència [10] que ens permet parlar amb el conductor o amb algun centre d'emergència associat.



Il·lustració 10: maneta d'emergència i interfonia

Tal i com podem veure els combois actuals ja disposen de pràcticament tots els dispositius, tot i això no s'ha trobat al mercat cap sistema que pogués monitoritzar-los de forma conjunta i oferir un registre d'informació fiable i una ràpida distribució d'aquesta, sinó que cada sistema és independent un de l'altre i moltes vegades no està preparat per interaccionar amb sistemes d'informació extern.

El projecte s'ha ideat des del punt de vista de poder oferir un registre de dades dels diferents valors ambientals de confort i els sistemes de seguretat. Actualment no s'ha trobat cap sistema que ofereixi un registre d'aquest valors i encara menys que ho ofereix de forma conjunta, la qual cosa, en cas de voler realitzar un estudi o d'analitzar algun error resulta impossible, però amb el sistema presentat es realitza de forma fàcil i còmoda ja que totes les dades queden emmagatzemades com a mínim per duplicat, una a la cabina del comboi i l'altra a l'estació.

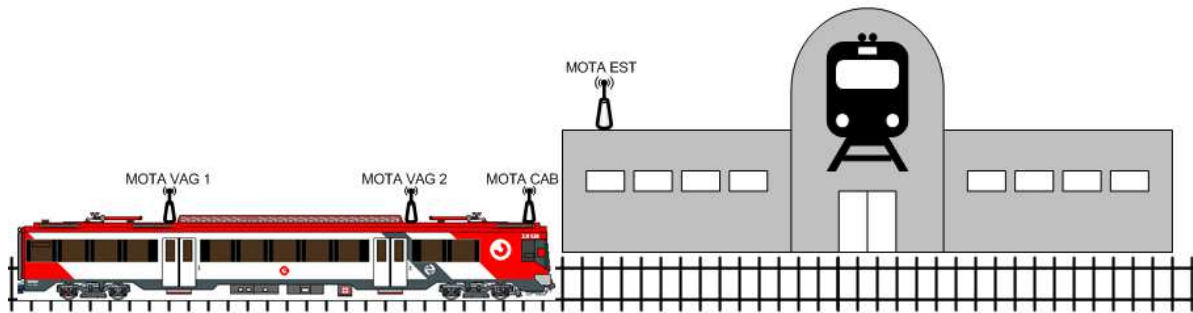
Es cert que el projecte no ofereix un augment de confort ni de seguretat, ja que els sistemes instal·lats actualment compleixen totes les normatives, però sí que ofereix una informació que pot resultar molt valuosa en cas de tenir que realitzar una investigació per algun incident o en altres casos per analitzar com funcionen els diferents sistemes ja instal·lats.

El projecte inicialment s'ha desenvolupat per instal·lar-se en mitjans de transport ferroviari com el tren, el metro o el tramvia, tot i que també, es pot instal·lar en qualsevol mitjà de transport col·lectiu com poden ser autobusos urbans o interurbans. Però abans de la seva instal·lació seria necessari una petita adaptació del dispositiu ja que el desenvolupat no deixa de ser un prototip i per tant no és d'aplicació directa sinó que falta comprovar el seu funcionament un cop instal·lat en el lloc.

### 3.- Descripció funcional

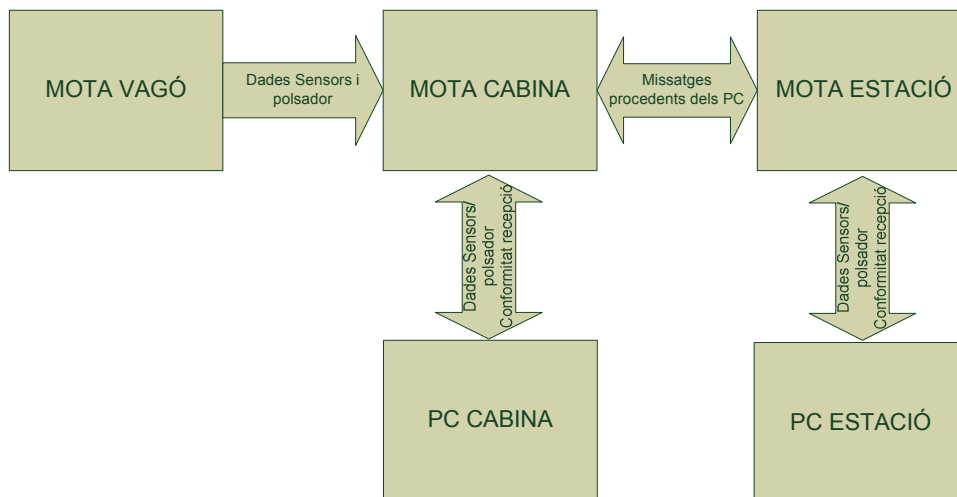
#### 3.1.- Sistema total

Cada porta d'accés als vagons porta una mota instal·lada que envia tota la informació recollida pels sensors a la mota de la cabina del conductor, que és l'encarregada de recollir tota la informació que li arriba de les motes del seu comboi [11]. La mota de la cabina està connectada a un PC que ofereix la visualització tota la informació recollida a través d'una pantalla. Quan el comboi arriba a una estació envia tota la informació que han recollit les motes a la mota instal·lada a l'estació, la qual també està connectada a un PC que mostra la informació subministrada per tots els combois.



Il·lustració 11: Tren arribant a l'estació -indicació de les motes-

#### 3.1.a.- Diagrama de blocs de l'aplicació



Il·lustració 12: Diagrama de blocs resumit de tota l'aplicació

#### 3.1.b.- Xarxa

La xarxa està composta per un mínim de tres motes i dos PC's, les tres motes estan situades; una al vagó de passatgers, l'altra a la cabina del conductor del comboi i l'última a l'estació. Aquestes es comuniquen via aèria mitjançant el protocol IEEE 802.15.4 / ZigBee que permet una tipologia de xarxa en malla. Els dos PC's es connecten mitjançant cable USB un a la mota de l'estació i l'altre a la mota de la cabina del conductor.

### 3.1.c.- Inter actuació dels diferents objectes del sistema

Podem definir cinc objectes diferents que es comuniquen entre sí, aquests objectes són la mota del vagó, la mota de la cabina, la mota de l'estació, el PC de la cabina i el PC de l'estació.

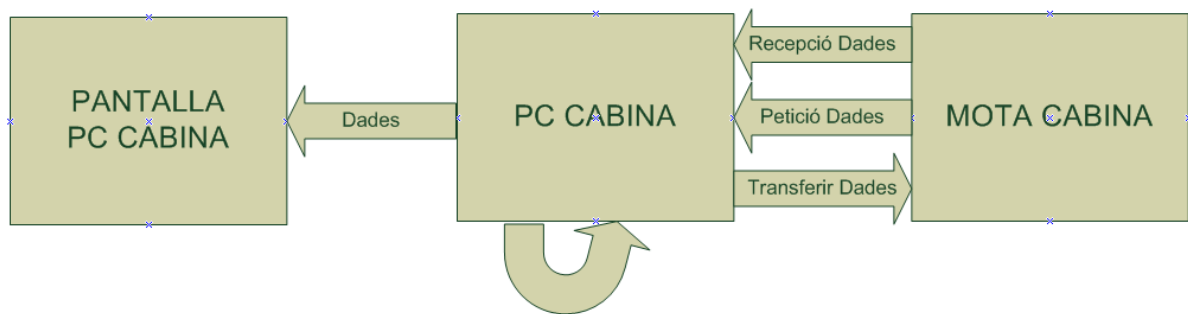
- La mota del vagó es comunica de forma unidireccional amb la mota de la cabina.
- La mota de la cabina rep dades de la mota del vagó i es comunica de forma bidireccional amb el PC de la cabina i amb la mota de l'estació.
- El PC de la cabina es comunica de forma bidireccional amb la mota de la cabina.
- La mota de l'estació es comunica de forma bidireccional amb el PC de l'estació i amb la mota de la cabina.
- El PC de l'estació es comunica de forma bidireccional amb la mota de l'estació.

### 3.2.- PC

El sistema desenvolupat disposa de dos PC's amb interfícies gràfiques un és el situat a la cabina del comboi i l'altre és el situat al despatx del cap d'estació anomenat. El PC de la cabina mostra totes les dades del comboi que li són subministrades per la mota que te connectada i el PC de l'estació mostra totes les dades de tots els combois que passen per l'estació i descarreguen les dades a la mota que te connectat.

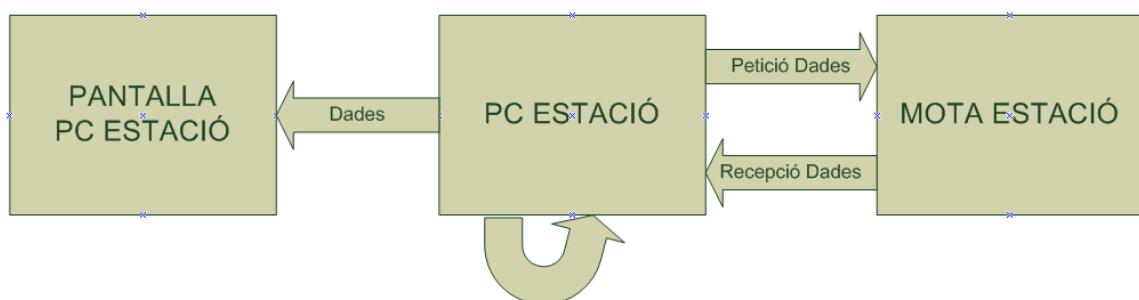
#### 3.2.a.- Diagrama de blocs de l'aplicació

##### PC CABINA



Emmagatzemar Dades  
II-lustració 13: Diagrama de blocs del PC Cabina

##### PC ESTACIÓ



Emmagatzemar Dades  
II-lustració 14: Diagrama de blocs del PC Estació



### 3.2.b.- Xarxa

L'única comunicació que tenen els PC's i que afecta al sistema és la que tenen amb les motes a través d'un cable USB, el PC de la cabina de conductor es connecta mitjançant USB amb la mota ubicada al mateix lloc, i per altra banda el PC de l'estació es connecta mitjançant USB amb la mota ubicada a l'estació.

### 3.2.c.- Inter actuació dels diferents objectes del sistema

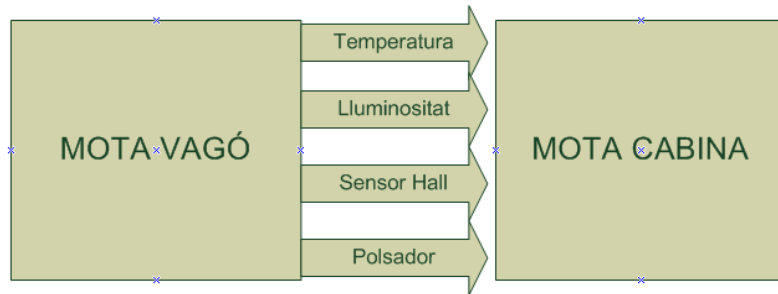
- El PC de la cabina del conductor del comboi rep i envia dades a través de la mota que es troba ubicada en la mateixa cabina. Quan rep dades les emmagatzema i les mostra per pantalla. També es dona el cas que el PC de la cabina pot rebre una petició de dades i al rebre-la comença a enviar totes les dades que no ha enviat abans.
- El PC de l'estació rep i envia dades a través de la mota que es troba ubicada en la mateixa cabina. El PC de l'estació està permanentment enviant peticions de dades i quan un PC cabina ho detecta envia les dades. Quan les rep les emmagatzema i les mostra per pantalla.

### 3.3.- Mota

El sistema desenvolupat disposa d'un mínim de 3 motes per funcionar, una situada a cada porta d'accés al comboi, una altra a la cabina del comboi i l'última a cada estació. Totes les motes tenen la funcionalitat de rebre i enviar dades programada, però només una és la que a més a més també té programat la lectura de les dades que li proporcionen els sensors.

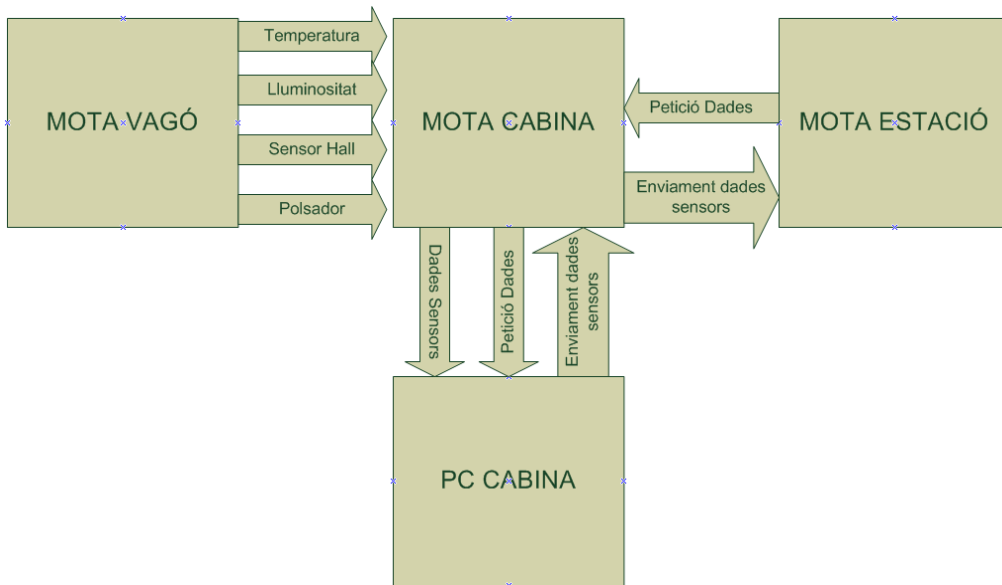
3.3.a.- Diagrama de blocs de l'aplicació

**MOTA VAGÓ**



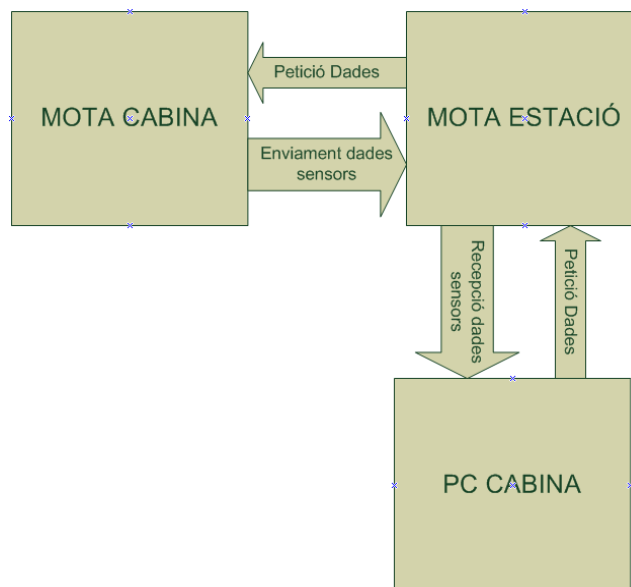
Il·lustració 15: Diagrama de blocs de la MOTA VAGÓ

**MOTA CABINA**



Il·lustració 16: Diagrama de blocs de la MOTA CABINA

**MOTA ESTACIÓ**



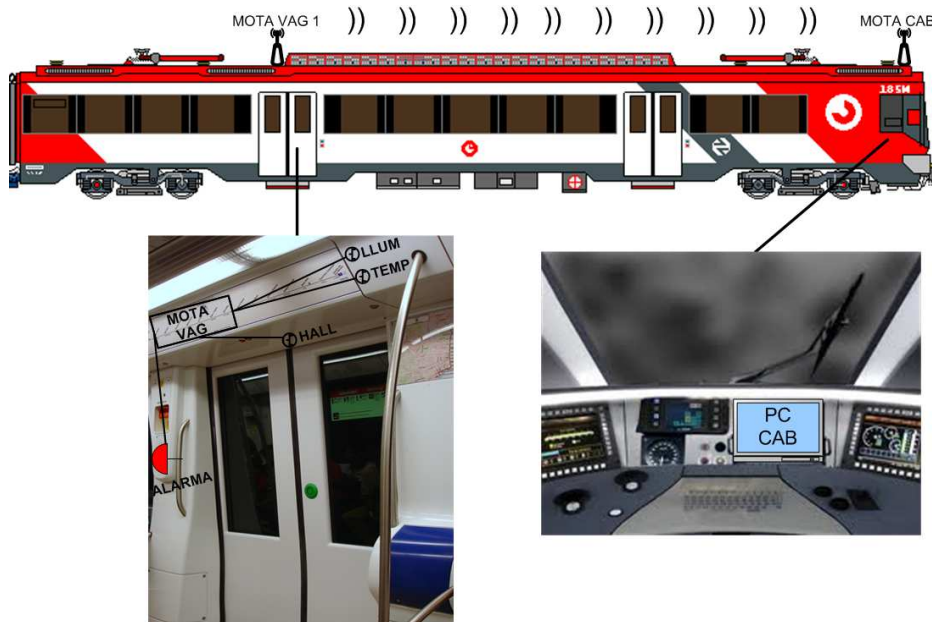
Il·lustració 17: Diagrama de blocs de la MOTA ESTACIÓ

### 3.3.b.- Xarxa

Les motes són la part principal de la comunicació del projecte, és a dir, sense la comunicació via aèria no hi hauria projecte. La mota que dóna sentit al projecte és la mota situada al vagó ja que és la que recull totes les dades i les envia via aèria a la mota situada a la cabina del comboi que les rep i les transfereix al PC de la cabina.

### 3.3.c.- Interactuació dels diferents objectes del sistema

- La mota situada al vagó recull la informació que li proporcionen els sensors i l'envia a la mota situada a la cabina del conductor [18].
- La mota situada a la cabina del conductor fa la funció de pont, és a dir, tot el que rep via aèria que pot ser de la mota del vagó o de la mota de l'estació ho envia via USB al PC de la cabina, i tot el que rep via USB ho retransmet per via aèria cap a la mota de l'estació.
- La mota situada a l'estació igual que la situada a la cabina del conductor fa la funció de pont, però en aquest cas només es comunica amb la mota de la cabina, és a dir, tot el que rep per via aèria ho transmet via USB al PC de l'estació al qual està connectat i tot el que rep via USB des del PC ho retransmet via aèria cap a la mota de la cabina de comboi.



Il·lustració 18: Situació dels diferents elements al comboi

## **4.- Descripció detallada**

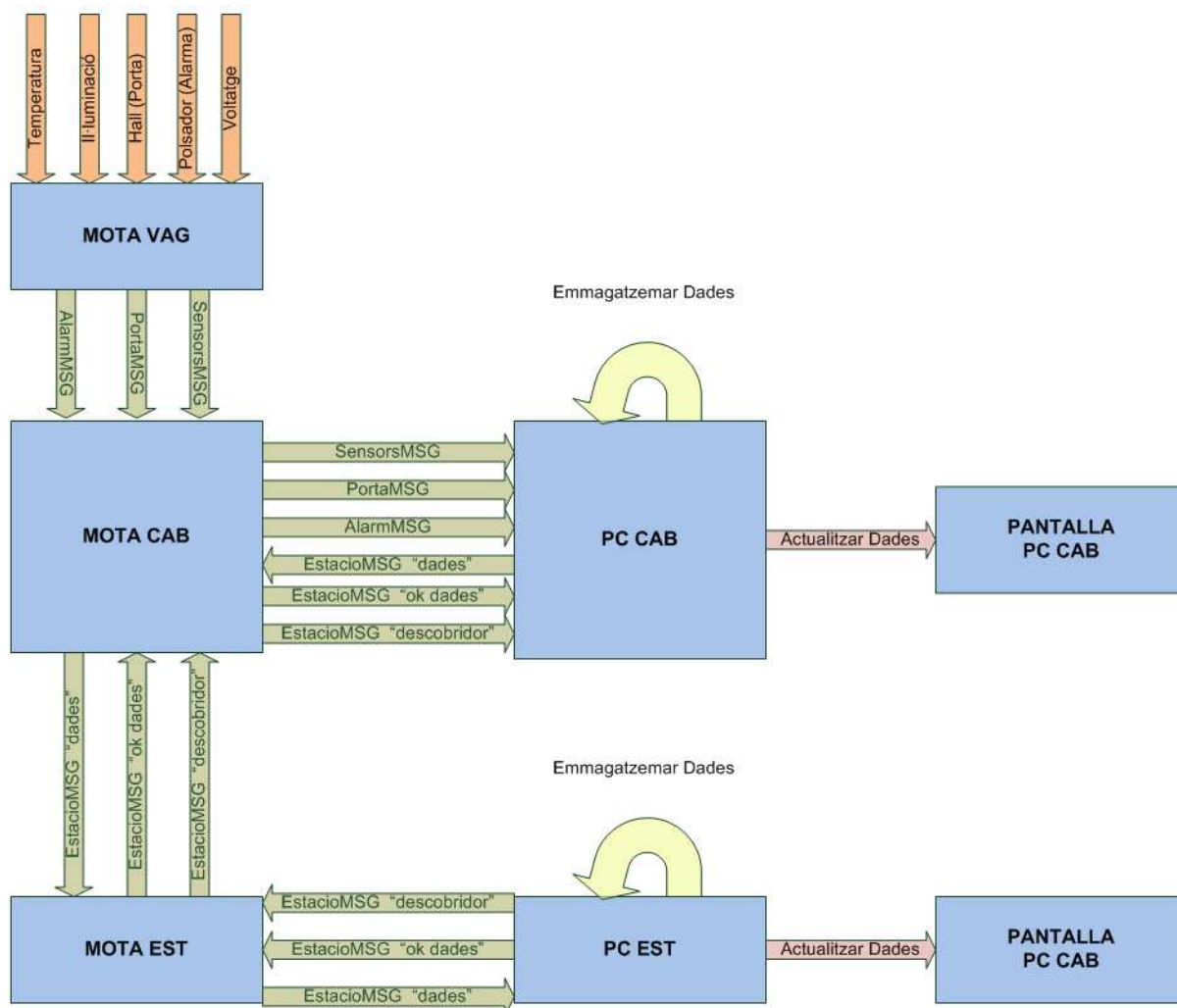
### **4.1.- Sistema total**

El sistema recull les dades dels sensors de temperatura, llum i voltatge de la MOTA\_VAG que està instal·lada al vagó, això ho fa cada trenta segons i són enviades juntament amb l'identificador de la mota cada cop que es fa una lectura de les dades dels sensors, és a dir, és llegeixen les dades i immediatament s'envien a la MOTA\_CAB. Així mateix els valors de l'última lectura dels sensors son emmagatzemats per la mota.

Per altra banda tenim el sensor Hall (obertura/tancament porta) i el polsador (alarma), les dades d'aquests dos sensors són enviades immediatament i només després de que s'activin, ja que se suposa que són dades vitals per la seguretat i no podem esperar al cicle de lectura/enviament de dades automàtic que es produeix cada trenta segons. Cal aclarir que amb el paquet enviat també s'envien les dades de temperatura, lluminositat i voltatge de l'última lectura que hagi fet la MOTA\_VAG. Una restricció a tenir present que té aquesta mota és que quan s'engega o es reinicia és imprescindible que totes les portes estiguin tancades ja que sinó el programa assumeix que un cop iniciada la mota la porta està tancada.

La mota instal·lada a la cabina del conductor (MOTA\_CAB) rep aquestes dades per via aèria (ràdio) i són retransmeses via port sèrie(USB) cap a l'ordinador PC\_CAB.

Quan l'ordinador rep aquestes dades les emmagatzema a la taula corresponent de la base de dades i al mateix temps actualitza la pantalla que pot veure el conductor. També cal destacar que en cas de que es polsi el botó d'alarma o el nivell d'il·luminació sigui inferior a l'establer, es mostrarà un missatge d'alerta.



Il·lustració 19: Diagrama detallat de funcionament de projecte.

## 4.2.- PC

### 4.2.1.- PC\_CAB

Programa desenvolupat utilitzant el llenguatge de programació Java.

- Inicialització: Pantalla gràfica de visualització de dades (Cabina Comboi).
- Funcionament: Quan a través del port sèrie es reben dades, és discerneix quin tipus de dades són (recepció de dades, petició d'enviament de dades).

Recepció de dades: Podem rebre dades utilitzant 3 tipus de paquets diferents, paquet *SensorsMsg* conté dades de temperatura, lluminositat, voltatge i Id mota, el paquet *PortaMsg* conté Id mota, l'obertura/tancament de la porta i els valors dels sensors de temperatura, llum i voltatge de l'última lectura i per últim *AlarmMsg* indicador d'alarma, l'Id mota i els valors dels sensors de temperatura, llum i voltatge de l'última lectura.

Quan rebem els paquets *SensorsMsg* automàticament desmembram la informació a la taula PC\_CAB de la base de dades afegint-hi la data i l'hora de rebuda i un identificador únic.

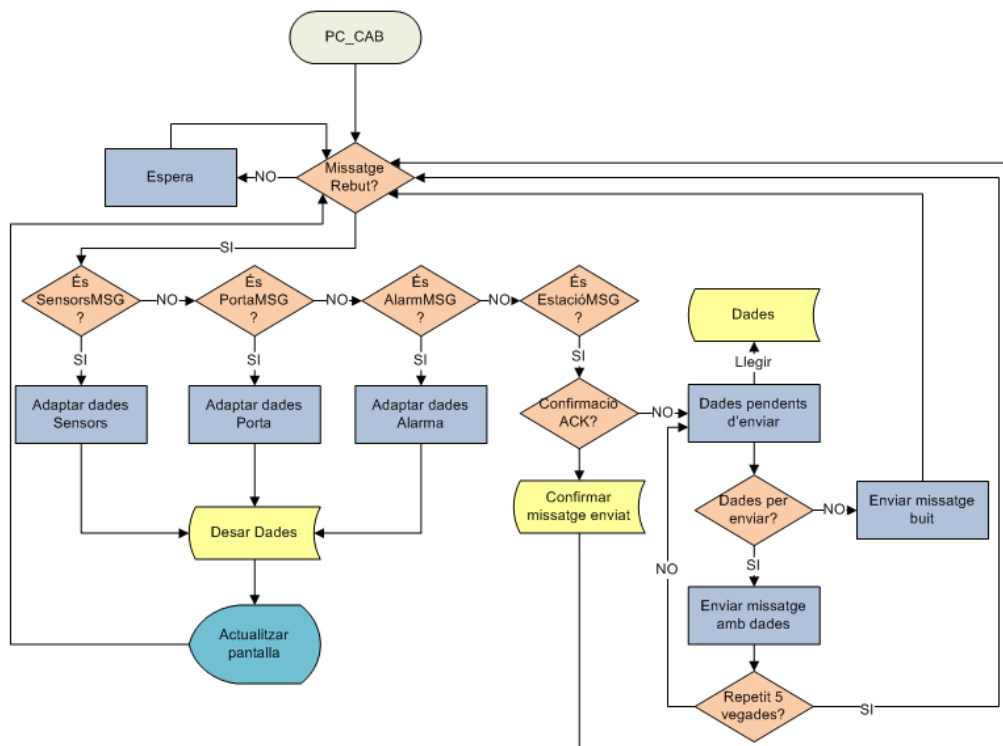
En cas de rebre un paquet *PortaMsg* aquest indicarà si la porta s'ha obert o s'ha informant a cada paquet de l'estat real de la porta, al mateix paquet també hi haurà la informació de l'última lectura dels sensors de la mota i tota aquesta informació serà desada a la taula PC\_CAB de la base de dades, afegint-hi la data i l'hora de rebuda i un identificador únic.

Si rebem un paquet *AlarmMsg* voldrà dir que s'ha premut el polsador d'alarma i ens sortirà un missatge a la pantalla informant d'aquest esdeveniment, al mateix paquet també hi haurà la informació de l'última lectura dels sensors de la mota i tota aquesta informació serà desada a la taula PC\_CAB de la base de dades, afegint-hi la data i l'hora de rebuda i un identificador únic.

També comentar que si al rebre un paquet *SensorsMsg* el valor de lluminositat és inferior a l'establer, ara en 250, es mostrarà un missatge per pantalla informant d'aquesta anomalia.

Enviament de dades: Rebem un paquet del tipus *EstacioMsg* que s'ha creat amb la intensió de dotar d'un paquet de dades el sistema per poder-les enviar entre el PC\_CAB i el PC\_EST, aquest paquet porta l'identificador de l'estació que ens sol·licita la informació. Automàticament es busca a la taula PC\_CAB de la base de dades si tenim registres sense enviar, en cas negatiu informem al PC\_EST mitjançant el mateix paquet que no hi ha dades, pel contrari comencem a enviar tongades de cinc paquets seguits i ens esperem dos segons i així successivament fins que s'acabin els paquets o deixem de rebre peticions. Per controlar el sistema d'enviament i assegurar que els paquets arriben correctament al seu destí, quan el PC\_CAB envia un paquet espera que el PC\_EST li contesti que l'ha rebut, si l'ha rebut el marca com a enviat, sinó el torna a enviar en la següent tongada de cinc paquets i així amb tots els paquets fins que rebí la confirmació.

Quan s'està enviant dades cap al PC\_EST apareix a la pantalla un missatge informant d'aquest esdeveniment.



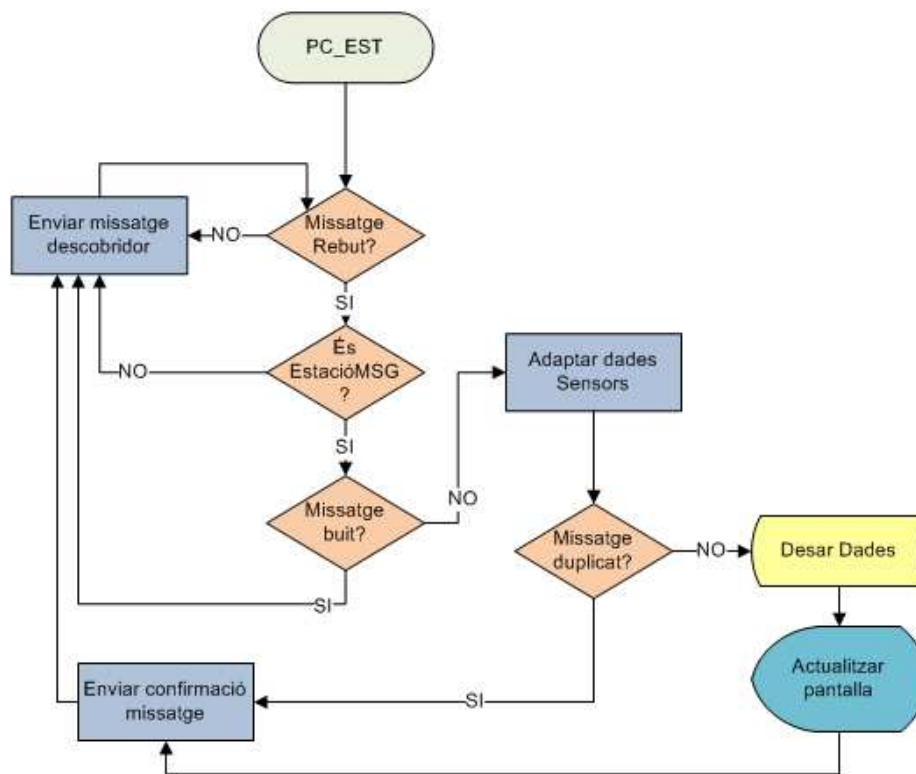
Il·lustració 20: Diagrama de flux del PC de cabina

#### 4.2.2.- PC\_EST

Programa desenvolupat utilitzant el llenguatge de programació Java.

- Inicialització: Pantalla gràfica de visualització de dades (Estació).

- Funcionament: El programa està enviant cada dos segons un missatge "descobridor", és a dir, un paquet *EstacioMsg* només amb l'identificador de l'estació a l'espera que algun PC\_CAB li respongui indicant-li si té dades o no, en el cas que la resposta sigui negativa apareixerà l'Id cab a l'apartat "ID COMBOI" de la pantalla, per indicar que ha arribat un comboi nou a l'estació però l'apartat de dades quedarà en blanc, si per contra, la resposta és afirmativa a la pantalla també apareixerà l'Id cab a l'apartat "ID COMBOI" i a la zona de dades s'anirà actualitzant amb la informació que es vagi rebent fins que no se'n rebi més i aquesta última rebuda quedi fixada fins que arribi el pròxim comboi.



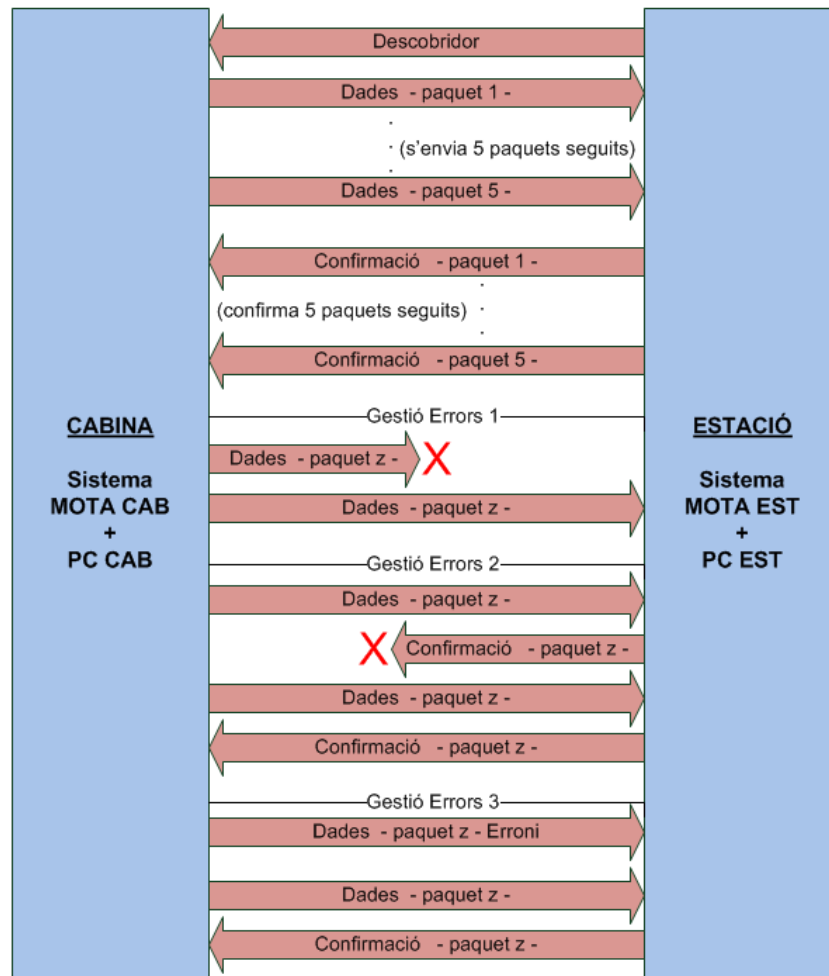
II-lustració 21: Diagrama de flux del PC de l'estació

Quan es rep un paquet d'informació es comprova si està repetit o és defectuós, si està repetit només s'envia un paquet a la PC\_CAB informant que s'ha rebut correctament sense emmagatzemar ni fer res més, per contra si no està repetit ni és defectuós s'emmagatzema a la taula PC\_EST de la base de dades afegint-hi la data i l'hora de rebuda i un identificador únic i a continuació s'envia el paquet confirmant la correcta rebuda del mateix.

Quan s'està enviant dades cap al PC\_CAB apareix a la pantalla un missatge informant d'aquest esdeveniment.

### 4.2.3.- Altres

#### Sistema control d'errors de transmissió



Il·lustració 22: Control d'errors de transmissió

El conjunt Cabina funciona gestionant tot el control el PC CAB i la MOTA CAB només reenvia els missatges que rep a través de l'USB cap a la via aèria i a l'inversa. Igualment fa el conjunt Estació amb els seus respectius sistemes, per això i per descriure'n el funcionament ens referirem a Cabina i a Estació.

Cas normal de funcionament: L'estació va enviant contínuament un missatge *descobridor* a l'espera que alguna cabina el rebí. Si el rep la cabina comença a enviar dades, a relació de cinc paquets de dades per cada paquet descobridor rebut. L'estació cada vegada que rep un paquet de dades envia una confirmació indicant que el paquet s'ha rebut correctament, llavors la cabina el marca com a enviat correctament.

Gestió Errors 1: La cabina envia un paquet de dades però l'estació no el rep mai i per tant no envia la confirmació. La cabina al no rebre la confirmació al proper missatge *descobridor* tornarà a enviar el paquet de dades.



Gestió Errors 2: L'estació rep un paquet i de forma immediata envia una confirmació de recepció que no arriba a rebre mai la cabina, al proper missatge *descobridor* tornarà a enviar el paquet de dades no confirmat, l'estació al rebre'l comprovarà que ja l'havia rebut i tornarà a enviar la confirmació de rebut, un cop la cabina rebí la confirmació el marcarà com a enviat correctament.

Gestió Errors 3: L'estació rep un paquet i detecta algun error en les dades rebudes, immediatament el descarta i no envia la confirmació, la cabina al no rebre la confirmació no el marca com a enviat correctament i al proper missatge *descobridor* el torna a enviar. En aquest cas la estació el rep correctament i envia tot seguit el missatge de confirmació, la cabina rep el missatge de confirmació i marca les dades com a enviades correctament.

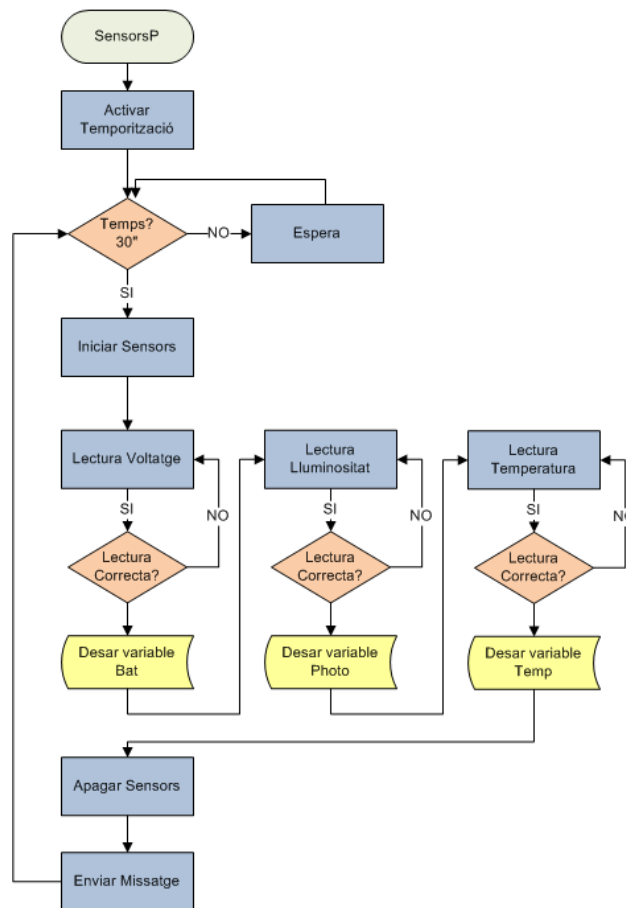
### 4.3.- Mota

#### 4.3.1.- MOTA\_VAG

- Inicialització: Abans de la seva posada en funcionament s'activa una seqüència de leds verd --> verd+taronja --> verd+taronja+vermell -->tot apagat, aquesta seqüència es repeteix 3 vegades fins que al final només queda encesa la llum verda com indicador de funcionament correcte.

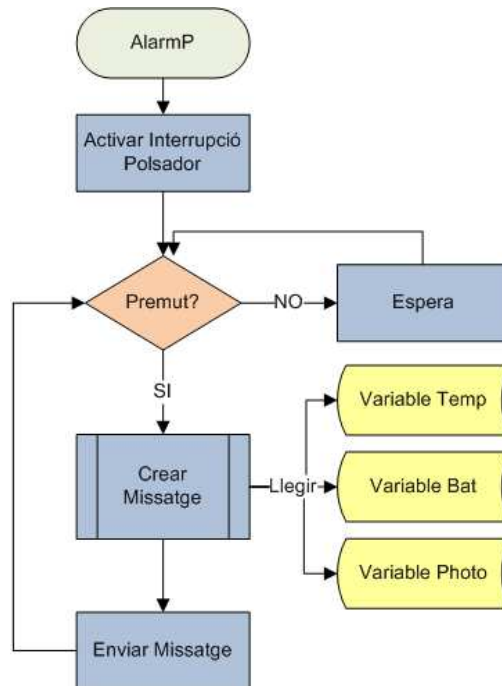
- Funcionament: Cada 30 segons es llegeixen els valors de temperatura, llum i voltatge i s'emmagatzemen només els últims valors a la mota, també i de forma immediata s'envien les lectures juntament amb l'identificador que nosaltres li hem assignat a la mota a la MOTA\_CAB, utilitzant el missatge "SensorsMsg".

Quan la mota envia una dada s'encén el led taronja durant mig segon per indicar que enviem una dada.



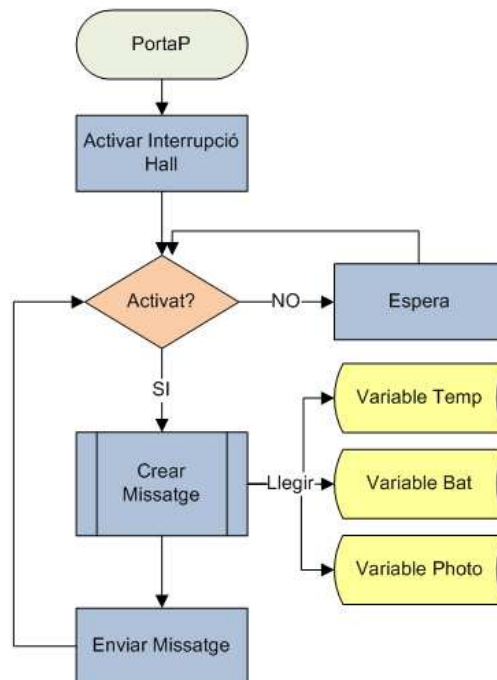
Il·lustració 23: Diagrama de flux de lectura dels sensors a la MOTA\_VAG

Quan accionem el polsador (Alarma) s'activen els leds taronja i vermell durant mig segon i envia un paquet a la MOTA\_CAB amb l'identificador de la mota i el camp "activat" amb el valor 1 i els valors dels sensors de temperatura, llum i voltatge de l'última lectura, utilitzant el missatge "AlarmMsg".



Il·lustració 24: Diagrama de flux del polsador d'alarma de la MOTA\_VAG

Quan activem el sensor Hall (Obertura/Tancament porta) s'activen els leds taronja i vermell durant mig segon i envia un paquet a la MOTA\_CAB amb l'identificador de la mota i el camp "seqNum" amb el valor 1 i els valors dels sensors de temperatura, llum i voltatge de l'última lectura, utilitzant el missatge "PortaMsg".



Il·lustració 25: Diagrama de flux del sensor Hall de la MOTA\_VAG

### **4.3.2.- MOTA\_CAB**

Porta implementat el programa "*BaseStation*".

- Inicialització: Cap.

- Funcionament: Quan es rep una dada a través de l'antena sense fils(via ràdio), s'activa el led de color taronja i es queda encès fins que en rebem un altre pel mateix medi. Aquesta dada rebuda és retransmesa cap al port sèrie (USB).

Quan es rep una dada a través del port sèrie (USB), s'activa el led de color vermell i es queda encès fins que en rebem un altre pel mateix medi. Aquesta dada rebuda és retransmesa cap l'antena sense fils(via ràdio).

### **4.3.3.- MOTA\_EST**

Porta implementat el programa "*BaseStation*".

- Inicialització: Cap.

- Funcionament: (vegi'n MOTA\_CAB Funcionament)

### **4.3.4.- Gestor Base de Dades**

- El gestor de base de dades utilitzat és Firebird Classic Server 2.5 amb la base de dades Info.fdb i les taules PC\_CAB i PC\_EST;

- L'estructura de les taules és la següent:

#### Taula PC\_CAB

id integer,  
enviat char(1),  
data date,  
hora time,  
motacabid integer,  
motavagid integer,  
temp decimal(5,2),  
llum integer,  
volt decimal(4,2),  
porta char(1),  
alarma char(1)

#### Taula PC\_EST

id integer,  
enviat char(1),  
dataest date,  
horaest time,  
motaestid integer,  
data date,  
hora time,  
motacabid integer,  
motavagid integer,  
temp decimal(5,2),  
llum integer,  
volt decimal(4,2),  
porta char(1),  
alarma char(1)

### 4.3.5.- Estructura dels missatges

Paquet EstacioMsg:

- L'identificador del paquet és 43 i la seva estructura és la següent:

|             |           |            |          |               |            |
|-------------|-----------|------------|----------|---------------|------------|
| estaciold   | hihadades | datahora   | cabinald | vagold        | countsTemp |
| 8 bits      | 1 bit     | 8 bits * 6 | 8 bits   | 8 bits        | 16 bits    |
| countsPhoto | countsBat | porta      | alarma   | vrefmilivolts | counter    |
| 16 bits     | 16 bits   | 1 bit      | 1 bit    | 16 bits       | 16 bits    |

- estaciold ..... Identificador de l'estació
- hihadades ..... Si hi ha dades 1 sinó 0
- datahora[6]..... Dia,mes,any,hora,minut,segon
- cabinald ..... Identificador de la cabina del comboi
- vagold ..... Identificador del vagó
- countsTemp ..... Valor de temperatura
- countsPhoto ..... Valor de la il·luminació
- countsBat ..... Valor del voltatge de la corrent elèctrica
- porta ..... 1 si la porta és oberta 0 si és tancada
- alarma ..... 1 si s'ha premut l'alarma, 0 si no s'ha premut
- vrefmilivolts ..... Referència conversió voltatge
- counter ..... Comptador de paquets

Paquet PortaMsg:

- L'identificador del paquet és 41 i la seva estructura és la següent:

|        |        |            |             |           |               |
|--------|--------|------------|-------------|-----------|---------------|
| moteld | seqNum | countsTemp | countsPhoto | countsBat | vrefmilivolts |
| 8 bits | 1 bit  | 16 bits    | 16 bits     | 16 bits   | 16 bits       |

- moteld ..... Identificador del vagó
- seqNum ..... 0 (porta tancada) // 1 (porta oberta)
- countsTemp ..... Valor de temperatura
- countsPhoto ..... Valor de la il·luminació
- countsBat ..... Valor del voltatge de la corrent elèctrica
- vrefmilivolts ..... Referència conversió voltatge

Paquet AlarmMsg:

- L'identificador del paquet és 42 i la seva estructura és la següent:

| moteld | activat | countsTemp | countsPhoto | countsBat | vrefmilivolts |
|--------|---------|------------|-------------|-----------|---------------|
| 8 bits | 1 bit   | 16 bits    | 16 bits     | 16 bits   | 16 bits       |

moteld..... Identificador del vagó  
activat ..... 1 (alarma activada)  
countsTemp..... Valor de temperatura  
countsPhoto..... Valor de la il·luminació  
countsBat..... Valor del voltatge de la corrent elèctrica  
vrefmilivolts ..... Referència conversió voltatge

### **5.- Viabilitat tècnica**

El projecte s'ha desenvolupat com a part del treball de fi de carrera en un entorn molt controlat i amb unes condicions climàtiques estables que proporcionen a les motes un espai òptim pel seu funcionament.

La implementació real en mitjans de transport ferroviari aniria precedida per realitzar uns tests d'estrès a les motes i als seus programes com seria sotmetre'ls a altes / baixes temperatures, diferents graus d'humitat, vibracions, interferències electromagnètiques, variació de voltatges, talls de subministrament elèctric repetits, etc... tot i que algunes d'aquestes condicions ja s'han comprovat en el projecte i per tant podem afirmar que les motes funcionarien correctament, no s'ha realitzat de forma continua durant un període llarg de temps.

#### **5.1.- Limitacions**

S'hauria d'adaptar físicament la mota perquè els sensors integrats a la pròpia mota fa que sigui poc precisa la lectura dels sensors i inviàble el polsador d'emergència, per això s'ha d'instal·lar un polsador més adaptat als usuaris, els sensors s'haurien de poder posar en els llocs més adequats per la tasca que han de realitzar i a més a més s'hauria d'habilitar una caixa estanca per evitar els agents externs (aigua, fred, pols, etc...) per tan no es pot realitzar una implementació directa i això provoca que s'han de realitzar algunes adaptacions menors.

#### **5.2.- Punts forts**

Per altra banda un cop realitzat les tasques descrites a l'apartat anterior, el sistema és totalment funcional i per tant la seva implantació és viable, la qual cosa dotaria als transports ferroviaris d'un control exhaustiu de totes les condicions en que viatgen els passatgers, en temps real pel conductor del comboi i diferit en uns minuts per les estacions de la línia.

El sistema permet tolerar certs errors de comunicació i funcionament. En cas d'error en la comunicació s'ha establert un control d'errors que detecta els missatges duplicats o la seva pèrdua i actua per corregir l'anomalia, a més a més, també pot detectar en certes ocasions quan un paquet arribat ho fa amb dades errònies.

Les motes permeten una reparació fàcil i ràpida en cas d'avaría, simplement s'ha d'agafar una mota nova, carregar-li el programari adequat en funció de la tasca que ha de desenvolupar i substituir la mota avariada per la que funciona correctament.

La compatibilitat del programari PC és molt alta gràcies al seu desenvolupament en llenguatge Java que permet que es pugui utilitzar tan en sistemes operatius Windows com Linux sense cap tipus de limitació. Igualment el gestor de base de dades Firebird també pot funcionar indistintament en qualsevol dels dos sistemes operatius. Això permet integrar el projecte al sistema de l'usuari ja siguin Windows, Linux o una combinació d'ambdós sense que això afecti al seu correcte funcionament.

El disseny en la distribució de les motes fa que si en una estació no funciona la mota (MOTA\_EST) o la mota de la cabina del comboi (MOTA\_CAB) no té temps de descarregar totes les dades, aquestes puguin ser transferides a la mota de la propera estació o posteriors.

## **6.- Valoració econòmica**

El cost econòmic d'un mateix projecte pot variar molt en funció de la qualitat i prestacions dels components, l'experiència dels professionals que el desenvolupen, un correcte anàlisi, una identificació clara dels requeriments... però no només varia el cost sinó també el temps que s'ha de dedicar a cada apartat, per això el pressupost s'ha realitzat agafant com a referència components de gamma mitja i tenint en compte els coneixements i sobretot l'experiència de què dispo en aquest àmbit, per aquest motiu el cost de les hores és reduït en comparació amb un treballador sènior ja que el cost aplicat és el corresponent a un treballador júnior, és a dir, amb poca experiència.

S'ha de tenir en compte que el cost del programari de desenvolupament és zero, també ho és el programari necessari perquè funcioni el nostre sistema. El projecte no té cap cost en llicències si el fem funcionar sobre el sistema operatiu Linux o qualsevol altre de gratuït que hi sigui compatible, en qualsevol altre cas s'hauria de sumar el cost de les llicències.

### **6.1.- Pressupost**

#### **6.1.1.- Programari**

|                     | <b>Dies</b> | <b>Hores/dia</b> | <b>Total hores</b> | <b>Preu/hora</b> | <b>Total</b>  |
|---------------------|-------------|------------------|--------------------|------------------|---------------|
| Disseny/Estudi      | 10          | 5                | 50                 | 30€              | 1.500€        |
| Desenvolupament     | 50          | 4                | 200                | 20€              | 4.000€        |
| Proves              | 5           | 5                | 25                 | 15€              | 375€          |
| Instal·lació        | 2           | 5                | 10                 | 25€              | 250€          |
| Formació            | 2           | 5                | 10                 | 20€              | 200€          |
| Manteniment (1 mes) | 4           | 2                | 8                  | 15€              | 120€          |
| <b>TOTAL</b>        |             |                  |                    |                  | <b>6.445€</b> |

#### **6.1.2.- Maquinari**

|                             | <b>Unitats</b> | <b>Preu/unitat</b> | <b>Total</b>     |
|-----------------------------|----------------|--------------------|------------------|
| Mota COU_1_2 24 A2          | 3              | 32€                | 96€              |
| Lenovo ThinkCenter M81 5049 | 2              | 659,63€            | 1.319,26€        |
| Cable extensió USB          | 3              | 2,75€              | 8,25€            |
| Transformador USB 5V 500mA  | 2              | 8,76€              | 17,52€           |
| <b>TOTAL</b>                |                |                    | <b>1.441,03€</b> |

**TOTAL** (programari + maquinari)= 6.445€ + 1.441,03€ = **7.886,03€**



### **6.1.3.- Manteniment**

Pel manteniment s'ha calculat una dedicació aproximada d'unes dues hores setmanals però tenint en compte que hi haurà setmanes que no serà necessari realitzar cap intervenció, o bé d'altres, que s'hi haurà de dedicar més hores. Aquest pressupost de manteniment no inclou les peces o components que s'hagin de substituir, sent aquest cost a càrrec del client, excepte si el component està en garantia que en aquest cas el client no haurà d'abonar cap import.

|                     | <b>Dies</b> | <b>Hores/dia</b> | <b>Total hores</b> | <b>Preu/hora</b> | <b>Total</b> |
|---------------------|-------------|------------------|--------------------|------------------|--------------|
| Manteniment (1 mes) | 4           | 2                | 8                  | 15               | 120          |
|                     |             |                  |                    |                  |              |

## **7.- Conclusions**

### **7.1.- Conclusions**

Els objectius inicials per aquest projecte eren que el sistema ha de funcionar automàticament i la informació s'ha de transmetre sense la necessitat d'intervenció de cap usuari. Aquestes dades s'han de mostrar i actualitzar automàticament a les pantalles establertes per aquest fi.

- Monitoritzar la temperatura de dins del vagó.
  - Comprovar a quina temperatura es troba l'interior dels vagons, així sabrem si es garanteix el confort dels viatgers.
- Monitoritzar la il·luminació de dins del vagó.
  - Cal analitzar constantment si la il·luminació de l'interior dels vagons és correcta o no, és a dir, si el sistema lumínic dels vagons funciona correctament i si no és així arreglar-ho.
- Monitoritzar l'estat de la porta del vagó.
  - És primordial saber l'estat de la porta de vagó, si està oberta o tancada. Controlant aquest apartat el conductor pot assegurar-se abans d'iniciar la marxa que totes les portes estan correctament tancades.
- Controlar el polsador d'alarma.
  - Els usuaris tenen a la seva disposició un polsador d'alarma que en cas de necessitat o emergència donarà un avís immediat al conductor el qual podrà prendre les mesures necessàries.
- Enviament de dades entre els vagons i la cabina (MOTA\_VAG --> MOTA\_CAB).
  - Establir un sistema de comunicació entre cada una de les motes dels vagons amb la mota de la cabina, la qual rebrà tota la informació.
- Enviament de dades entre la cabina i les estacions (MOTA\_CAB --> MOTA\_EST).
  - Establir un sistema de comunicació bidireccional entre la mota de la cabina dels combois i la de les estacions per tal d'intercanviar informació.
- Enviament de dades entre les estacions i el centre de control (PC\_EST --> PC\_CC).
  - Utilitzar la xarxa LAN ja sigui amb connexió física o sense fils per enviar tota la informació que s'ha recollit a les estacions a un node central al Centre de control.
- Interfície gràfica i emmagatzematge d'informació al PC\_CAB, PC\_EST i PC\_CC.
  - Per que tan el conductor com el cap d'estació com el responsable al centre de control puguin veure i gestionar les dades es crearan unes interfícies gràfiques adaptades a cada un d'aquets llocs.

En el transcurs del desenvolupament hi va haver algunes modificacions que no han afectat a l'essència del projecte.

En primer lloc trobem la comunicació MOTA\_CAB --> MOTA\_EST. Aquesta comunicació en un principi havia d'estar íntegrament controlada per les motes, essent elles les encarregades de controlar els errors de comunicació, de si les dades s'han rebut correctament de detectar-se entre si, i dels PC's només demanar o entregar paquets d'informació ja validats, però a la pràctica i després de vàries proves vaig comprovar que el temps de desenvolupament s'allargaria de forma considerable i no podria complir amb els terminis de presentació fixats, per aquest motiu es va decidir passar tot aquest control al programari del PC ja que el desenvolupament de tots aquets apartats en Java és més ràpid i es podria complir amb les dates d'entrega fixades.

Per altra banda tenim l'enviament de dades al PC\_CC i el seu desenvolupament, el qual va ser desestimant ja que la part central del projecte són les motes i la seva comunicació amb els diferents dispositius i no pas la visualització més o menys sofisticada de les dades.

### 7.2.- Proposta de millores

El llistat de millores i funcionalitats podria ser molt ampli ja que sempre hi ha alguna cosa a millorar, perfeccionar o una funcionalitat nova que reclama el mercat, però de totes elles les més rellevants són:

- El control d'errors i pèrdua de paquets s'hauria d'implementar el seu funcionament directament a les motes.
- Control d'errors i confirmació de recepció entre la MOTA\_VAG i la MOTA\_CAB.
- Control d'errors i confirmació de recepció entre la MOTA\_CAB i la MOTA\_EST gestionat des de les pròpies motes.
- Un entorn gràfic més acurat i amb més opcions de consulta, cerca i gestió de les dades.
- Caixa hermètica per protegir la mota dels agents externs.
- Els sensors no haurien d'estar soldats a la placa, sinó que haurien d'estar connectats per mitjà d'un cable que permetés instal·lar el sensor al lloc més adequat.

### 7.3.- Autoavaluació

A l'iniciar aquest projecte tenia un coneixement nul de les motes i del seu funcionament així que abans de començar el desenvolupament vaig haver de dedicar moltes hores per conèixer els dispositius, el sistema operatiu (TinyOS), el llenguatge que utilitzen (nesC) i tot l'entorn de desenvolupament que s'havia de preparar al PC (Linux, eclipse, meshprog, ...). Tot aquest temps dedicat a l'aprenentatge relacionat amb els dispositius el vaig trobar a faltar a la part final, malgrat que era del tot necessari per poder començar i planificar correctament el projecte.

La programació de les motes tot i ser en un llenguatge basat en el 'C' el nesC i que per tant no hi hauria d'haver cap entrebanc en la programació, a la pràctica ha estat més complicat del que s'havia plantejat inicialment sobretot per evitar barrejar els diferents conceptes de que consta el llenguatge com són interfícies, mòduls, events, tasques, missatges, etc... i la relació que tenen entre si, també en temes d'execució les motes tenen particularitats com l'execució síncrona o asíncrona.

Durant el projecte he pogut anar veient que les motes no són uns dispositius amb unes aplicacions limitades, sinó tot al contrari, el que en un petit dispositiu (actualment 1 mm<sup>3</sup>) pogués haver-hi una unitat de procés, uns sensors i connexió via ràdio fa que el seu potencial sigui enorme, de fet, els usos que se li podrien donar actualment ja són molt amplis, com aplicacions militars, l'observació de condicions climàtiques arreu del món, etc... i en un futur fins i tot en podríem tenir dins del nostre cos perquè comprovessin el nostre estat de salut. A part de les possibles aplicacions de les motes, he constatat que el sistema de programació és complex, no es tracta només de llegir les dades dels sensors i després enviar-les via ràdio sinó que podem operar amb aquestes dades i qualsevol altres que rebem, ja sigui via ràdio o via sèrie. En definitiva, era un apartat que desconeixia totalment i del que es fa poca publicitat respecte al potencial que pot arribar a tenir.

### 8.- Glossari

- **mota / motes**: Acrònim de les paraules en anglès remote sensing. Petit dispositiu de mida molt reduïda amb capacitat computacional i normalment dotat de sensors que funciona de forma autònoma però capaç de crear una xarxa sense fils si es troba varis dispositius similars o iguals.
- **Mota COU\_1\_2 24 A2**: mota que ens proporciona la UOC per realitzar el projecte dotat d'un microprocessador ATMEL, varis sensors i comunicació sense fils utilitzant el protocol ZigBee.
- **Smart dust**: també conegut coma pols intel·ligent, és una xarxa sense fils de motes.
- **TinyOS**: Sistema operatiu que utilitzen les motes i que està especialment dissenyat per aquets dispositius.
- **nesC**: Llenguatge de programació derivat del C i optimitzat per adaptar-se a les prestacions i limitacions de les motes.
- **C**: llenguatge de programació potent, ràpid i eficaç que s'utilitza normalment per desenvolupar sistemes operatius o aplicacions que necessiten un accés pròxim al maquinari.
- **C#**: llenguatge de programació orientat a objectes derivat del C que combina el potencial del C i la velocitat de programació d'altres llenguatges d'alt nivell
- **sensors**: dispositiu capaç de mesurar una magnitud exterior i transformar-la en impulsos elèctrics. Ex. temperatura, il·luminació, humitat...
- **comunicació sense fils / comunicació aèria / comunicació via ràdio**: Intercanvi d'informació entre dos o més dispositius sense la necessitat d'una connexió física, sinó que utilitzen ones electromagnètiques que es propaguen per l'espai.
- **USB**: port de connexió universal utilitzat per connectar dispositius externs de diferents tipus.
- **Sistemes Encastats**: sistema computacional de baix cost on tots els seus components estan integrats i desenvolupat per realitzar algunes tasques en concret.
- **Microcontrolador**: xip amb capacitat computacional, memòria i unitats d'entrada/sortida.
- **Transceptor**: dispositiu de comunicació que permet la recepció i l'enviament d'informació en un mateix dispositiu.
- **Sensor Hall**: sensor que s'activa al detecta un camp magnètic.
- **802.15.4**: estàndard de comunicació que defineix el nivell físic i l'accés al medi de xarxes sense fils domèstiques de baixa velocitat de transmissió.
- **ZigBee**: conjunt de protocols basats en l'estàndard 802.15.4, per les comunicacions sense fils amb velocitats de transmissió i consum reduïts.
- **Bluetooth**: especificació industrial per xarxes sense fils domèstiques que permet connectar diferents tipus de dispositius sense necessitat de cables.

- **Firebird**: sistema de gestió de base de dades relacional de codi obert.
- **JayBrid**: interfície de programació que permet la comunicació entre una aplicació desenvolupada en Java i una base de dades Firebird.
- **SerialForwarder**: eina(programari) de comunicació entre el PC a través del port sèrie (usb) i una mota.
- **meshprog**: eina (programari) que permet transferir un codi font compilat a una mota a través del port sèrie (usb).
- **Eclipse**: entorn de desenvolupament escrit principalment en Java i basat en codi obert que proporciona un entorn de programació per a Java i altres llenguatges.

## **9.- Bibliografia**

- 1.- Wiki UOC. Informació sobre la mota i com programar-la.  
<http://cv.uoc.edu/app/mediawiki14/wiki/>
- 2.- TinyOS web  
<http://www.tinyos.net/>
- 3.- Eclipse web. Entorn de programació  
<http://www.eclipse.org/>
- 4.- Informació sobre la mota  
<http://en.wikipedia.org/wiki/Smartdust>
- 5.- Llistat de motes  
[http://en.wikipedia.org/wiki/List\\_of\\_wireless\\_sensor\\_nodes](http://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes)
- 6.- Manual de programació TinyOS  
<http://ants.dif.um.es/rm/apuntes/Tutorial-TinyOS.pdf>
- 7.- Manual programació NesC  
<http://nesc.sourceforge.net/papers/nesc-ref.pdf>
- 8.- Futur de les motes  
<http://futuretechtrends.com/2011/07/data-carrying-capacity-of-wireless-nanosensor-internet/>
- 9.- Informació sobre el protocol 802.15.4  
[http://en.wikipedia.org/wiki/IEEE\\_802.15.4](http://en.wikipedia.org/wiki/IEEE_802.15.4)
- 10.- Informació sobre el protocol ZigBee  
<http://en.wikipedia.org/wiki/ZigBee>
- 12.- Comparativa dels protocols 802.15.4 i ZigBee  
<http://sensor-networks.org/index.php?page=0823123150>
- 13.- Informació sobre el microprocessador de la mota COU\_1\_2 24 A2  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc8226.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8226.pdf)
- 14.- Botiga productes electrònics  
<http://www.bricogeek.com/shop/23-sensores>
- 15.- Informació sobre combois ferroviaris  
<http://www.ferropedia.es/wiki/Portada>  
<http://www.todotrenes.com/>
- 16.- Informació sobre tipus de motes, sensors, microprocessadors, ...  
<http://www.snm.ethz.ch/Main/HomePage>
- 17.- Sistemes de climatització ferroviària  
<http://www.merak-hvac.com>

## 10.- Annexos

### 10.1.- Manual d'usuari

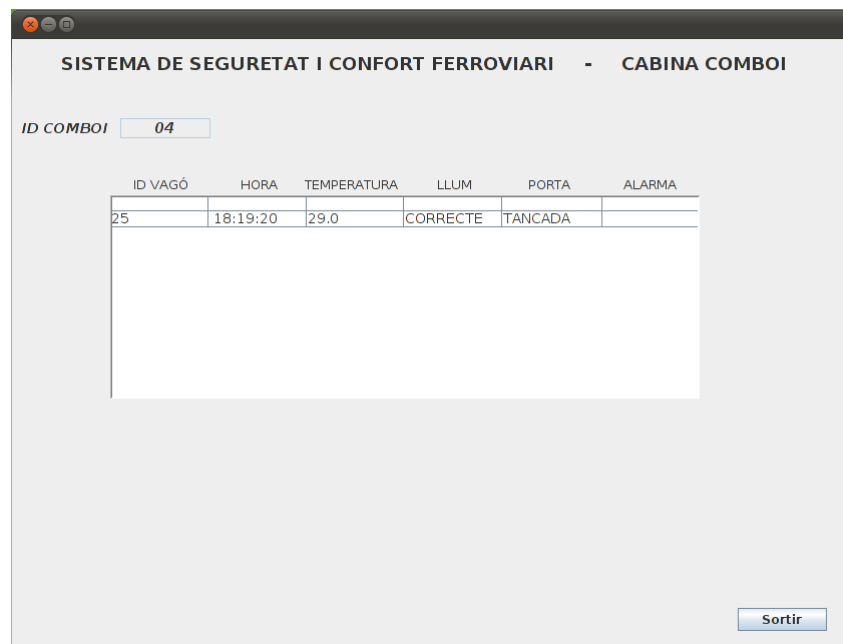
A continuació podem veure les pantalles de funcionament dels aplicatius desenvolupats en JAVA, la funcionalitat és la de mostrar per la pantalla les dades que es van rebre i si hi ha algun error amb la il·luminació o es prem el polsador d'alarma s'activa una alerta i mostra un missatge.

#### 10.1.1.- PC CAB

Per poder executar el programa PC\_CAB és necessari tenir prèviament activat i funcionant el *SerialForwarder*, el PC\_CAB espera la connexió pel port 9002. Per iniciar aquest connector ho podem fer:

```
java net.tinyos.sf.SerialForwarder -comm serial@/dev/ttyUSB0:19200
```

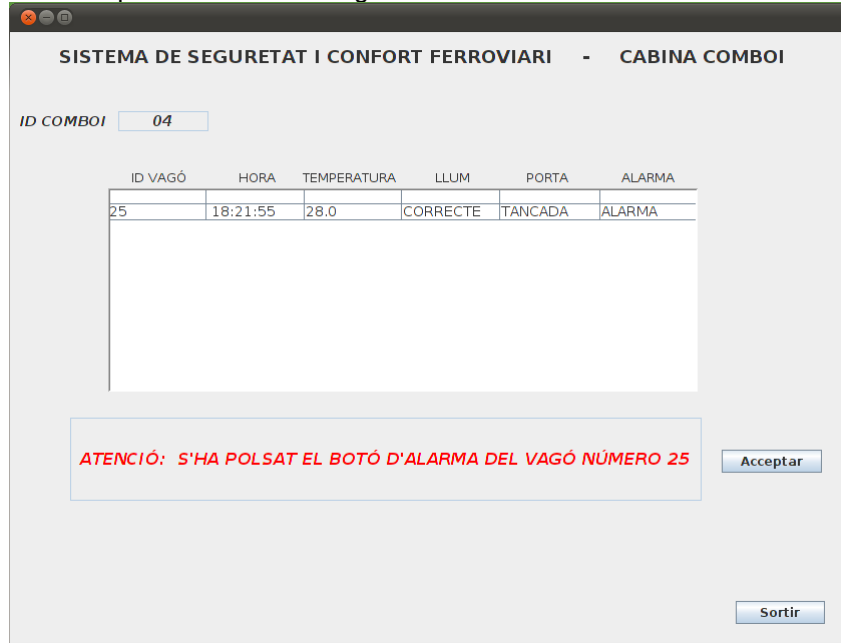
Captura pantalla recepció de dades de la mota del vagó número 25:



Il·lustració 26: Pantalla principal PC Cabina

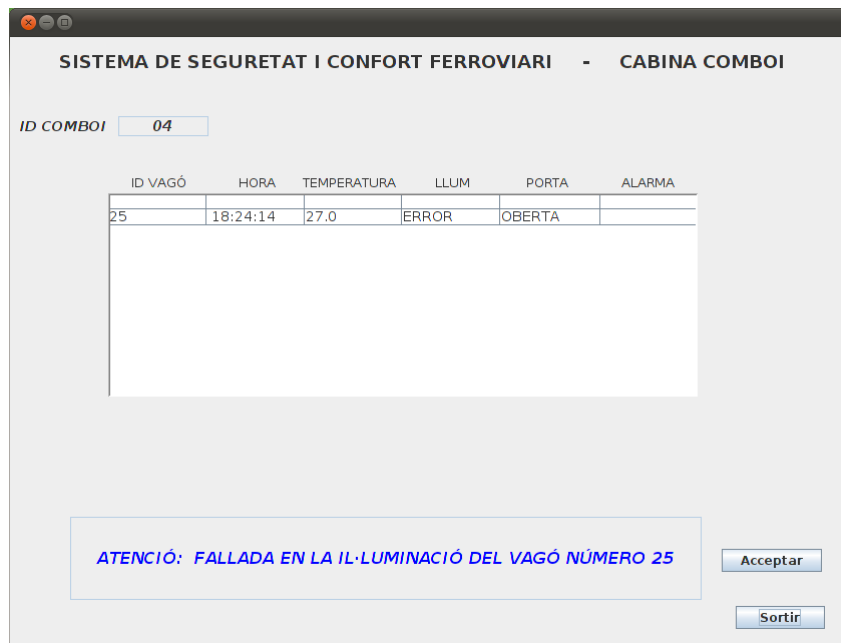


Captura pantalla recepció d'alarma del vagó número 25:



Il·lustració 27: Pantalla principal PC Cabina -Alarma polsador activada-

Captura pantalla recepció de fallada d'il·luminació i porta oberta al vagó número 25:



Il·lustració 28: Pantalla principal PC Cabina -Alarma il·luminació activada-

Captura pantalla enviament de dades a una estació:



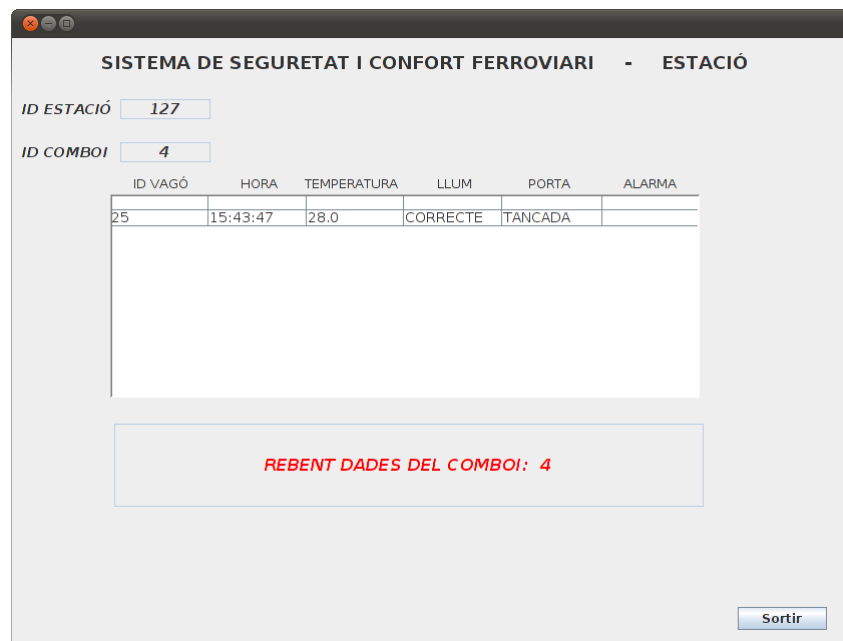
Il·lustració 29: Pantalla principal PC Cabina -Anunci enviant dades-

### 10.1.2.- PC\_EST

Per poder executar el programa PC\_EST igual que a l'apartat anterior és necessari tenir prèviament activat i funcionant el *SerialForwarder*. A diferència del PC\_CAB el PC\_EST espera la connexió pel port 9004. Per iniciar aquest connector ho podem fer:

```
java net.tinyos.sf.SerialForwarder -comm serial@/dev/ttyUSB1:19200
```

Captura pantalla recepció de dades del comboi número 4:



Il·lustració 30: Pantalla principal PC Estació -Anunci rebent dades-

## **10.2.- Execució i compilació**

La compilació i l'execució ho separarem per mòduls per tal que sigui més entenedor:

### **10.2.1.- MOTA\_VAG**

Primer hem de compilar el programa que anirà dins de la mota que anomenarem MOTA\_VAG. Per fer-ho anirem dins de la carpeta MOTA\_VAG/src i utilitzarem la següent comanda:

```
make cou24
```

Ens crearà unes subcarpetes a les quals hi haurem d'entrar que són MOTA\_VAG/src/build/cou24. Un cop estiguem al directori hem de carregar el programa compilat que es troba a l'arxiu "*main.srec*" a la mota, per fer-ho utilitzarem la comanda:

```
meshprog -t /dev/ttyUSB0 -f main.srec
```

Una vegada s'executi la comanda anterior ens sortiran uns punts a la pantalla, i en aquest moment s'ha de prémer el botó reset de la mota per carregar el programa. Un cop premut el botó ens avisarà mentre s'estigui transferint la informació i quan acabi.

Un cop acabat ja tenim la mota amb el programari MOTA\_VAG i ja està llesta per funcionar. Cal destacar que el sistema mostra un sistema d'inicialització utilitzant els leds, que és una seqüència que es repeteix 3 vegades: verd --> verd+taronja --> verd+taronja+vermell --> apagats, quedant al final la llum verda encesa permanent i la taronja només s'encén durant mig segon quan s'envien dades i la vermella també només mig segon quan es prem el polsador (Alarma) o s'activa el sensor Hall (Obertura/Tancament porta).

### **10.2.1.- MOTA CAB**

Primer hem de compilar el programa que anirà dins de la mota que anomenarem MOTA\_CAB. Per fer-ho anirem dins de la carpeta BaseStation/src i utilitzarem la següent comanda:

```
make cou24
```

Ens crearà unes subcarpetes a les quals hi haurem d'entrar que són BaseStation/src/build/cou24. Un cop estiguem al directori hem de carregar el programa compilat que es troba a l'arxiu "*main.srec*" a la mota, per fer-ho utilitzarem la comanda:

```
meshprog -t /dev/ttyUSB0 -f main.srec
```

Una vegada s'executi la comanda anterior ens sortiran uns punts a la pantalla, i en aquest moment s'ha de prémer el botó reset de la mota per carregar el programa. Un cop premut el botó ens avisarà mentre s'estigui transferint la informació i quan acabi.

Un cop acabat ja tenim la mota amb el programari BaseStation que ens servirà per comunicar-nos via sense fils (radio) amb les MOTA\_EST.

### **10.2.1.- MOTA EST**

Les instruccions són les mateixes que la MOTA\_CAB substituint tot el que fa referència a les motes anteriors per MOTA\_EST.

### Instal·lació del FireBird 2.5 Classic

La versió de Linux que s'utilitza és la "OSIAN" basada en la distribució coneguda Ubuntu.

Primer de tot actualitzem el sistema amb al comanda:

```
sudo apt-get update
```

I tot seguit es descarrega i s'instal·la el FireBird amb la següent instrucció:

```
sudo apt-get install firebird2.5-classic
```

Ara ja tenim instal·lat el sistema de base de dades FireBird 2.5 Classic.

### Creació de la BBDD i les Taules

En primer lloc s'ha de crear el fitxer que s'utilitzarà com a base de dades, aquest el crearem a la carpeta /BBDD que no hem d'oblidar de donar els permisos corresponents perquè si pugui accedir sense cap restricció. Per fer-ho ens posarem dins la carpeta BBDD que hem creat i utilitzarem la següent comanda per accedir al sistema que ens permetrà crear la base de dades:

```
isql-fb
```

Un cop estiguem dins hem d'escriure la següent instrucció:

```
CREATE DATABASE './Info.fdb' page_size 4196 user 'SYSDBA' password 'masterkey';
```

Automàticament se'ns crearà una base de dades al fitxer 'Info.fdb' amb una mida de paginació de 4196 i amb l'usuari 'SYSDBA' que se li assigna la contrasenya 'masterkey'. A continuació només ens queda connectar-nos a la nova base de dades amb la comanda:

```
CONNECT './Info.fdb' 'SYSDBA' password 'masterkey';
```

i seguidament procedirem a crear les dues taules necessàries la PC\_CAB i la PC\_EST, ho farem amb les següents instruccions:

```
CREATE TABLE PC_CAB (  
    id integer not null,  
    enviat char(1),  
    data date,  
    hora time,  
    motacabid integer,  
    motavagid integer,  
    temp decimal(5,2),  
    llum integer,  
    volt decimal(4,2),  
    porta char(1),  
    alarma char(1)  
);
```

```
CREATE TABLE PC_EST (  
    id integer not null,  
    enviat char(1),  
    dataest date,  
    horaest time,  
    motaestid integer,  
    data date,  
    hora time,  
    motacabid integer,  
    motavagid integer,  
    temp decimal(5,2),  
    llum integer,  
    volt decimal(4,2),  
    porta char(1),  
    alarma char(1)  
);
```

I per últim utilitzem la instrucció 'commit;' i les taules ja quedaran creades i validades.

### Instal·lació connector JDBC JayBird-2.1.6

Baixar el fitxer "*Jaybird-2.1.6JDK\_1.6.zip*" de la web de "*firebirdsql.org*" ([http://sourceforge.net/projects/firebird/files/firebird-jca-jdbc-driver/2.1.6-release-jdk16/Jaybird-2.1.6JDK\\_1.6.zip/download](http://sourceforge.net/projects/firebird/files/firebird-jca-jdbc-driver/2.1.6-release-jdk16/Jaybird-2.1.6JDK_1.6.zip/download)). Descomprimir-lo i desar-lo al disc, en aquest cas concret s'ha desat al director "*/opt/Jaybird*" i importar la llibreria a l'editor Java que s'estigui utilitzant mitjançant el fitxer "*jaybird-full-2.1.6.jar*"

### **10.2.3.- PC CAB**

Perquè funcioni el PC\_CAB hi ha uns requeriments previs, vegeu:

- Instal·lació FireBird 2.5 Classic.
- Instal·lació connector JDBC JayBird-2.1.6
- Creació de la BBDD i les Taules.

Llibreries especials utilitzades:

- TinyOS 2.1.1
- Jaybird 2.1.6
- javax.swing / java.awt (aparat gràfic)
- Classes: missatges.PortaMsg; missatges.AlarmMsg; missatges.SensorsMsg; missatges.EstacioMsg

Cal recordar que tal i com s'ha dit anteriorment el fitxer de la base de dades (*Info.fdb*) ha d'estar disponible i amb tots els permisos necessaris dins la carpeta "*/BBDD*", la taula que utilitza s'anomena PC\_CAB.

Per compilar el PC\_CAB s'ha utilitzat l'editor Eclipse versió "Helios" amb totes les llibreries i classes anteriorment descrites incorporades, cal tenir en compte que tan per la classe TinyOS com per la (JDBC) JayBrid s'han d'incloure al "classpath", ja que sinó l'aplicatiu no podrà trobar les classes i funcionarà erròniament.

### **10.2.4.- PC EST**

Per poder-lo compilar és necessari les mateixes llibreries i classes que hem descrit en l'apartat anterior, a més a més podem aprofitar la mateixa base de dades si executem les dues pantalles al mateix ordinador, amb la única peculiaritat que la taula que utilitza s'anomena PC\_EST.