

## 1. Introducció

Cada vegada més, les empreses s'interessen en la col·laboració d'obres socials per tal de millorar la seva imatge de cara als clients. La responsabilitat social corporativa, o també anomenada responsabilitat social empresarial, pot definir-se com a la contribució activa i voluntària de la millora social, econòmica i ambiental per part de les empreses, generalment, amb l'objectiu de millorar la seva situació competitiva i valorativa i el seu valor afegit. Així doncs, aquesta contribució social no deixa de ser un interès per tal d'obtenir els màxims beneficis.

Des d'un punt de vista personal, l'interès per col·laborar en una obra social implica un respecte com a deute en front la societat a més a més d'una realització i satisfacció personals. D'aquesta manera, una petita aportació de qualsevol tipus, no només de caire econòmic, per part de cadascú permet grans beneficis socials i millores en la qualitat de vida de moltes persones que realment necessiten atenció especial.

Les fundacions sense afany de lucre que contribueixen a iniciatives socials disposen de recursos limitats per la quantitat de necessitats i demandes que se l'hi ofereixen. El fet de realitzar un estudi o projecte pot ser una bona oportunitat per col·laborar en un iniciativa social i així poder facilitar el dia a dia que per algunes persones resulta complicat.

L'oferta d'una oportunitat com aquesta, en la qual es veu una utilitat a una feina desenvolupada i, a més a més, que ajudi a les persones més necessitades comporta un gran al·licient a l'hora d'afrontar-lo, treballar-hi i finalitzar-lo amb bons resultats.



## 2. La Fundació

La Fundació privada El Maresme Pro Persones Amb Disminució Psíquica és una institució d'iniciativa social sense afany de lucre que promou i impulsa la integració social de les persones amb disminució psíquica des dels principis de normalització i reconeixement de la diferència.

La Fundació el Maresme va ser constituïda el 22 de desembre de 1994, continuant la tasca iniciada l'any 1966 per l'Associació Patronat Pro Persones Amb Disminució Psíquica del Maresme.

Com a centre comarcal de referència, atén i dona resposta a qualsevol qüestió relacionada amb les persones amb disminució psíquica, ja sigui a nivell informatiu i d'orientació, o d'assistència o tractament.

La Fundació el Maresme procura donar resposta a les necessitats i demandes de la comarca, organitzant una xarxa de serveis que ofereixin una atenció en continuïtat a la persona amb disminució psíquica i la seva família.

Per tal de donar servei a les diferents necessitats, la Fundació es troba distribuïda en departaments:

### Centre de Desenvolupament Infantil i Atenció Precoç del Maresme (CDIAP)

És un servei de diagnòstic i tractament en règim ambulatori, completament gratuït per les famílies, que atén als infants des dels 0 als 6 anys que presenten algun tipus de trastorn, disminució, disfunció o disharmonia en el seu desenvolupament, o que es troben en situació de risc de patir-la.

### Escola d'educació Especial l'Arboç

Atén nens i nenes afectats de paràlisis cerebral o altres lesions amb manifestació motora que interfereixen greument en el procés de desenvolupament i d'aprenentatge.

### Servei Ocupacional d'Inserció (SOI)

El SOI és un Centre Ocupacional que acull persones adultes, a partir dels 18 anys, amb un grau de disminució psíquica igual o superior al 33% i amb un nivell de quocient intel·lectual entre mig i lleuger, que han acabat el període de formació escolar, però que encara no estan integrats dins del Centre Especial de Treball ni dins del Servei de Teràpia Ocupacional.

### Centre Especial de Treball del Maresme (CEO)

El CEO del Maresme té com a objectiu la integració laboral, personal i social de la persona amb discapacitat psíquica, en règim de treball normalitzat, amb edat laboral (a partir dels 16 anys), ja sigui a l'empresa ordinària o dins del propi Centre Especial de Treball.

Dóna resposta laboral a persones amb discapacitat psíquica, a partir dels 16 anys, amb un grau de disminució psíquica igual o superior al 33% i capacitat productiva.

El Centre Especial de Treball és en l'actualitat una empresa que ofereix treball a unes 200 persones amb discapacitat psíquica i que continua la seva tasca de recerca de llocs de treball per donar una possibilitat de sortida laboral a les persones que actualment estan en formació.

### Servei de Teràpia Ocupacional (STO)

Els Serveis de Teràpia Ocupacional (STO) són l'alternativa des de l'àmbit de serveis socials a la integració de les persones amb disminució psíquica que no poden incorporar-se al sistema de treball ordinari per no arribar a un determinat nivell de productivitat.

Atén persones adultes, a partir de 18 anys, amb un grau de disminució del 65% o més, i amb un nivell entre mig i sever de dependència. L'objectiu principal és afavorir al màxim la integració de les persones amb disminució psíquica mitjançant una atenció diürna de tipus rehabilitador integral.

### Servei d'Habitatges

El Servei d'Habitatges té per objecte facilitar, de forma temporal o permanent, una vida independent i equivalent a la pròpia llar a persones adultes, amb una disminució psíquica en grau igual o superior al 33%.

### Servei d'Acompanyament i Suport a la Vida Independent (SASVI)

El Programa de suport a l'autonomia a la pròpia llar pretén proporcionar a aquelles persones amb disminució psíquica amb un nivell alt d'autonomia i capacitats, una sèrie de suports puntuals i intermitents, que els facilitin la seva integració social, els permetin viure de manera autogestionada i portar una vida independent i desinstitucionalitzada al seu domicili particular, ja sigui de lloguer o en propietat, sols, o acompanyats d'altres persones beneficiàries d'aquest programa.

L'atenció i suport en la vida independent fomenta l'autonomia com a eina del desenvolupament personal i social, i actua complementàriament a d'altres mitjans d'intervenció i xarxes de suport com la comunitat, família o amics.

### Residència Llar Sta. Maria

La residència i CAE Llar Sta. Maria Assumpta és un establiment de la Fundació Maresme, adreçat a persones amb disminució psíquica greu en edat adulta, que degut al seu alt grau d'afectació necessiten assistència d'una forma bàsica i continuada i que, per raons familiars i socials, no poden viure a casa seva.



### 3. Proposta de la Fundació el Maresme

Es va proposar un projecte des de la Fundació el Maresme Pro Persones amb Disminució Psíquica a través del Departament del Servei Ocupacional d'Inserció (SOI) de la Fundació i l'Escola Universitària Politècnica de Mataró (EUPMT), on s'especificava el següent:

*En essència, la intenció és que aquest recurs tecnològic faci d'assistent personal d'una persona discapacitada que viu amb un elevat grau d'autonomia en el seu pis, o amb una autonomia més reduïda, en una llar.*

*Com a agenda electrònica pot realitzar recordatoris de cites, prendre medicacions, posar o estendre roba de la rentadora, anar a comprar, etc.*

*Facilitar l'anar a comprar fent-se llistes de la compra senzilles i pràctiques.*

*Que a la vegada, el responsable de vetllar per aquesta persona, l'educador/a, pugui enviar-li missatges entenedors, programar-li recordatoris, etc... des d'un PC*

A més a més, es posa a disposició un Pocket PC, model Movistar TSM 500, que serà l'eina per la qual es notificarà a la persona discapacitada qualsevol tipus de recordatori. Es suposa, per això, que la persona discapacitada anirà sempre acompanyada del Pocket PC.





## 4. Objectius

L'objectiu principal del projecte consisteix en desenvolupar una aplicació que permeti que un administrador extern pugui gestionar l'agenda electrònica d'una persona discapacitada de tal manera que l'agenda alerti dels diferents recordatoris en el moment apropiat.

Apareixen dos subobjectius pel fet que el projecte es divideix en dues parts: una part de comunicació entre l'administrador extern i el Pocket PC, i una part de visualització dels recordatoris de l'agenda en el moment idoni.

Per la part de comunicació, cal establir un vincle entre l'administrador extern i el Pocket PC de la persona discapacitada utilitzant les diferents tecnologies a disposició del programador tenint en compte les possibilitats i/o limitacions que presenta el Pocket PC.

Pel que fa a la part de visualització, cal elaborar un sistema que mostri els diferents recordatoris del Pocket PC en el moment apropiat, de tal manera que la persona discapacitada s'assabenti de les seves notificacions.

Per últim, cal fer una valoració del fet d'incorporar un sistema TTS (Text to Speech) que permeti la notificació del recordatori convertint el text en veu sintètica.



## 5. Alternatives a la proposta de la Fundació

### 5.1. Alternatives de comunicació

#### 5.1.1. Alternativa amb servidor Exchange i mòdul GPRS

Una alternativa a tenir en compte és el fet de configurar un servidor amb Microsoft Exchange Server que emmagatzemi les dades de tots els usuaris del servei i que es comuniqui amb l'exterior a través d'un mòdul GPRS connectat a ell. Un administrador extern és qui gestiona la inserció dels esdeveniments per a cadascun dels usuaris a través d'una connexió entre el servidor i els diferents Pocket PCs.

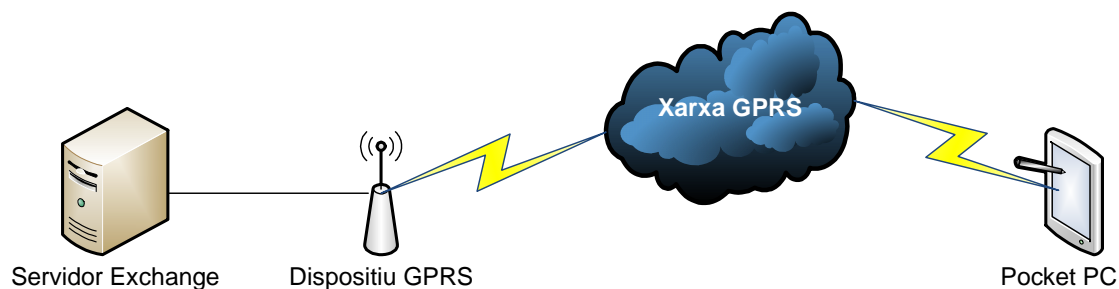


Fig. 5.1.1.1 – Esquema conceptual d'alternativa amb servidor Exchange

Es descarta aquesta alternativa pel fet que és econòmicament inviable, bàsicament pel cost econòmic del mòdul GPRS, i amb una certesa de prosperitat de projecte baixa ja que, per exemple, només hi ha una empresa que utilitza aquesta alternativa a Mataró i és el propi Ajuntament de Mataró.

### 5.1.2. Alternativa amb Google Calendar i Java

Una altra possible alternativa tracta d'utilitzar l'eina web Google Calendar de tal manera que emmagatzemi les dades de tots els usuaris del servei i que es comuniqui amb el Pocket PC via GPRS. Una aplicació instal·lada al Pocket PC és qui gestiona tant la sincronització amb Google Calendar com la notificació dels esdeveniments al dispositiu. Un administrador gestiona la inserció dels esdeveniments per a cadascun dels usuaris al Google Calendar i l'aplicació desenvolupada en Java i instal·lada al Pocket PC s'encarrega d'establir una connexió entre Google Calendar i el dispositiu mòbil.

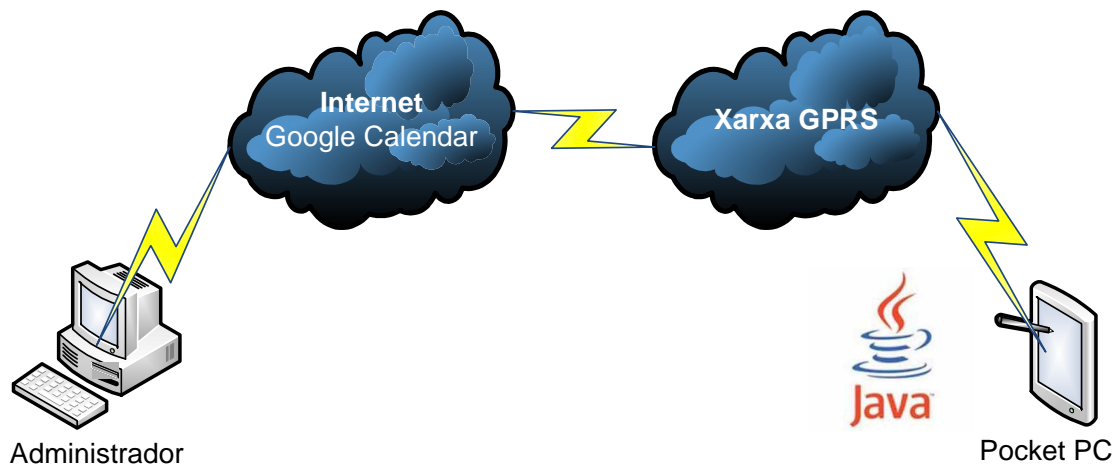


Fig. 5.1.2.1 – Esquema conceptual de l'alternativa amb Google Calendar

Aquesta alternativa té limitacions de comunicació amb el Pocket Outlook de Pocket PC ja que disposa d'un paquet de comunicacions molt reduït. El problema principal ve determinat pel fet que el paquet de comunicacions .net de Java no es troba disponible a la versió de Windows Mobile. Això té un cert sentit ja que aquest paquet de comunicacions té un pes important en espai de disc si tenim en compte les limitacions en quant a hardware d'aquests dispositius. Cal tenir en compte que un programa per a Pocket PC ha de ser de pocs centenars de de KiloBytes, cosa que restringeix, en gran mesura, funcionalitats addicionals.

### 5.1.3. Alternativa amb Google Calendar i XML

Una altra possible alternativa tracta d'utilitzar l'eina web Google Calendar de tal manera que emmagatzemi les dades de tots els usuaris del servei i que es comuniqui amb el Pocket PC via GPRS. Una aplicació instal·lada al Pocket PC és qui gestiona tant la sincronització amb Google Calendar com la notificació dels esdeveniments al dispositiu. Un administrador gestiona la inserció dels esdeveniments per a cadascun dels usuaris al Google Calendar i l'aplicació desenvolupada en XML i instal·lada al Pocket PC s'encarrega d'establir una connexió entre Google Calendar i el dispositiu mòbil.

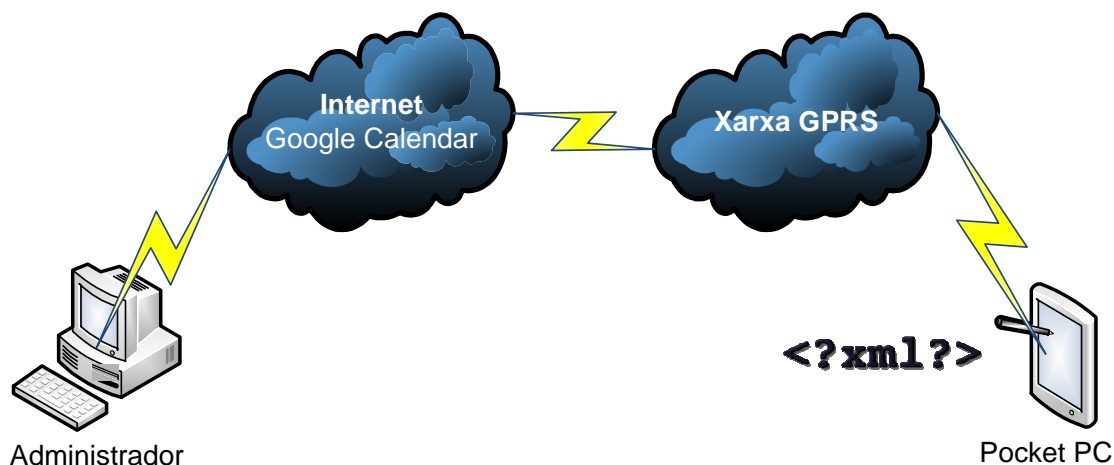


Fig. 5.1.3.1 – Esquema conceptual de l'alternativa amb Google Calendar i XML

Aquesta alternativa també es descarta ja que és difícil estimar si el desenvolupament de la totalitat de l'aplicació correspon a un projecte. Cal tenir en compte que els programes que realitzen només la part de comunicació d'aquest projecte estan desenvolupats en XML i són de pagament, cosa que fa pensar que suposen un volum d'hores important.

XML sembla ser l'únic llenguatge amb el qual es poden extreure dades del Google Calendar.

5.1.4. Alternativa amb Google Calendar i GooSync

Una altra possible alternativa seria utilitzar l'eina web Google Calendar de tal manera que emmagatzemi les dades de tots els usuaris del servei i que es comuniqui amb el Pocket PC via GPRS. Una aplicació externa instal·lada en el Pocket PC és qui gestiona la sincronització de Google Calendar amb el calendari predeterminat del dispositiu. Una aplicació apart extreu les dades del calendari predeterminat del Pocket PC i notifica dels esdeveniments al dispositiu. Un administrador extern és qui gestiona la inserció dels esdeveniments per a cadascun dels usuaris al Google Calendar.

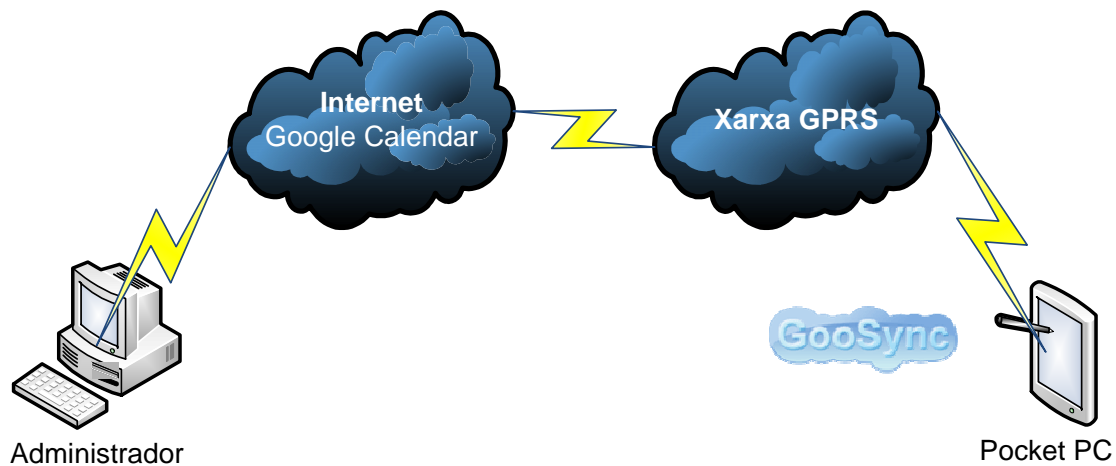


Fig. 5.1.4.1 – Esquema conceptual de l'alternativa amb Google Calendar i GooSync

És la solució escollida ja que facilita la feina de comunicació i permet acotar la durada del projecte de manera que es pugui dur a terme en un temps de dedicació raonable.

La única limitació que pot suposar aquesta alternativa és el fet de no poder afegir alguna possible funcionalitat útil de cara al futur malgrat sembla que les funcionalitats actuals són més que suficients per l'ús que se li pot donar a l'aplicació.

## 5.2. Alternatives de visualització

### 5.2.1. Alternativa en Java

L'alternativa inicial per la part de visualització fa referència a la implementació d'un programa en Java que mostri els recordatoris emmagatzemats al Pocket Outlook del Pocket PC en el moment necessari. Aquest programa es connecta amb la base de dades del Pocket Outlook, n'extreu les dades necessàries i en mostra els resultats en el moment escaient. És l'alternativa inicial ja que és un llenguatge multiplataforma molt utilitzat en l'actualitat i que disposa d'un gran nombre de recursos i APIs de tercers que en primer moment poden semblar una bona solució a la comunicació amb Pocket Outlook.

Apart del problema de comunicació, Java ofereix una API de PIM que permet a les aplicacions Java dels dispositius mòbils llegir i escriure de/a la base de dades d'informació personal emmagatzemada al dispositiu.

El Pocket PC Movistar TSM 500 disposa d'un programa anomenat Midlet Manager que funciona com a emulador de Java que permet executar programes escrits en Java per a telèfons mòbils al Pocket PC.

El problema ve donat quan s'intenta accedir a les dades de Pocket Outlook on, a partir del petit programa següent, es determina que l'API PIM no es troba disponible en aquest dispositiu.

```
import java.io.IOException;

public class PIMInstallTester {
    public static void main(String[] a) throws Exception {
        String version = null;
        version = System.getProperty("microedition.pim.version");
        if (version != null) {
            throw new IOException("PIM optional package is not available."
        }
    }
}
```

### 5.2.2. Alternativa en C++

Una altra alternativa per la part de visualització fa referència a la implementació d'un programa en C++ que mostri els recordatoris emmagatzemats al Pocket Outlook del Pocket PC en el moment necessari. Aquest programa es connecta amb la base de dades del Pocket Outlook, n'extreu les dades necessàries i en mostra els resultats en el moment escaient. És una alternativa a tenir en compte ja que l'eina de la qual se n'extreuen les dades és una eina de Microsoft, el Pocket Outlook, i aquest disposa de molta documentació i exemples en aquest llenguatge a la seva web

És la solució escollida ja que disposava d'exemples i plantilles útils per a desenvolupar l'aplicació, com ara l'API Pocket Outlook Object Model (POOM) que permet la comunicació entre el Pocket Outlook i l'aplicació. La única dificultat és la poca experiència en el desenvolupament d'aplicacions en aquest llenguatge però, en part, és un llenguatge orientat a objectes que utilitza la mateixa estructura conceptual.



### 5.3. Alternativa d'instauració de TTS

L'alternativa d'instauració d'un sistema TTS al dispositiu mòbil respon a la utilitat d'una aplicació que, a més a més de mostrar l'esdeveniment a través de la pantalla del dispositiu, aquest el reproduïx en forma de veu sintètica. Així doncs, TTS respon a les sigles Text to Speech i tracta d'un conversor de text llegible a veu sintètica.

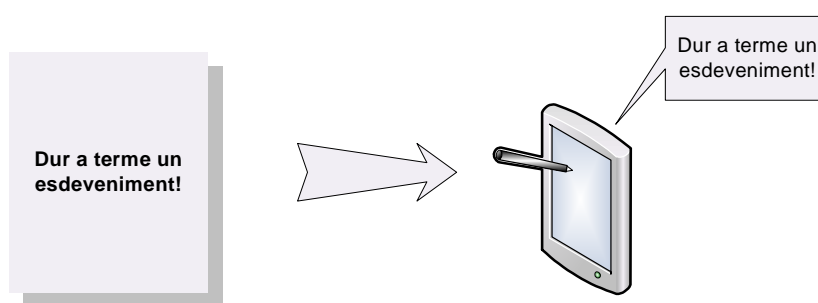


Fig. V – Conversió de text a veu

Aquesta alternativa es pot tenir en compte de cara a l'ús en altres dispositius però, concretament en el Movistar TSM 500, no té cap mena de sentit ja que la distorsió que crea el dispositiu a l'hora d'emetre veu fa intel·ligible el missatge que es dona.

Les proves es poden dur a terme amb l'aplicació Flite, un programa que converteix text inserit a través del teclat a veu per a dispositius mòbils, com per exemple PDAs.

La solució es pot afegir al propi codi de l'aplicació desenvolupada a través de la incorporació de les APIs necessàries que disposa la web del programa.



## 6. Solució adoptada

El projecte es divideix en dues parts ben diferenciades: la part de comunicació i la part de visualització.

### Comunicació

La part de comunicació pretén resoldre la creació d'un vincle entre l'administrador extern i el Pocket PC de la persona discapacitada utilitzant les diferents tecnologies a disposició del programador tenint en compte les possibilitats i/o limitacions que presenta el Pocket PC. Aquest tipus de vincle facilita la sincronització de les dades del calendari del PC de l'administrador amb les dades del calendari del Pocket PC de la persona discapacitada de tal manera que noves dades inserides per l'administrador es reflecteixen al Pocket PC destí. Així doncs, la solució adoptada tracta de comunicar l'eina web Google Calendar amb el calendari del Pocket Outlook del Pocket PC.

Per establir aquesta comunicació, s'utilitza un programa, anomenat GooSync, extern a la pròpia aplicació i configurable de tal manera que la sincronització entre PC i Pocket PC sigui unidireccional, és a dir, que el Pocket PC es sincronitzi amb el PC però no a l'inrevés.

### Visualització

La part de visualització pretén mostrar les dades del calendari del Pocket Outlook per la pantalla del Pocket PC. La solució adoptada consisteix en un programa desenvolupat en llenguatge C++ que extreu les dades del calendari del Pocket Outlook del Pocket PC a través d'APIs com la Pocket Outlook Object Model (POOM) i les mostra per la pantalla d'aquest quan pertoca, és a dir, quan l'hora de l'esdeveniment en qüestió coincideix amb l'hora local actual.

## 6.1. Recursos Tecnològics

### Pocket PC

El Pocket PC és un ordinador de butxaca, també anomenat PDA, sigles de Personal Digital Assistant. Es tracta d'un petit ordinador, dissenyat per ocupar el mínim espai i ser fàcilment transportable, que executa el sistema operatiu Windows CE de Microsoft entre d'altres, el qual li proporciona capacitats similars als PCs d'escriptori.

Qualsevol dispositiu que sigui qualificat com a Pocket PC ha de:

- Executar el sistema operatiu Microsoft Windows CE o Windows Mobile
- Tenir un conjunt d'aplicacions en ROM
- Incloure una pantalla sensible al tacte
- Incloure un dispositiu apuntador anomenat stylus o estilet.
- Incloure un conjunt de botons de hardware per activar aplicacions
- Estar basat en un processador compatible amb el StrongARM

Algunes de les aplicacions que s'inclouen en aquests dispositius son versions reduïdes de Microsoft Outlook, Internet Explorer, Word, Excel, Windows Media Player, etc.

### GPRS

El General Packet Radio Service (GPRS) o Servei General de Paquets Via Radio és una extensió del Sistema Global de Comunicacions Mòbils o GSM, sigles de Global System for Mobile Communications, per a la transmissió no commutada de dades, o per paquets. Permet velocitats de transferència de 56 a 114 kbps.

GPRS es pot utilitzar per a serveis com el WAP, sigles de Wireless Application Protocol, el servei de missatges curts o SMS, el servei de missatgeria multimèdia o MMS, Internet i per als serveis de comunicació, com el correu electrònic i el World Wide Web. La transferència de dades es cobra per volum d'informació transmesa, en kilobytes o megabytes, mentre que la comunicació de dades a través de commutació de circuits

tradicionals es factura per minut de temps de connexió, independentment de si l'usuari utilitza tota la capacitat del canal o està en un estat d'inactivitat.

El GPRS dona un millor rendiment a la commutació de paquets de serveis, en contraposició a la commutació de circuits, on una certa qualitat de servei, també anomenat QoS, està garantida durant la connexió. Per aquest motiu, es considera més adequada la connexió commutada per a serveis com la veu que requereixen un ample de banda constant durant la transmissió, mentre que els serveis de paquets com GPRS s'orienten al trànsit de dades.

### Bluetooth

Bluetooth és una especificació industrial per Xarxes Inalàmbriques d'Àrea Personal que possibilita la transmissió de veu i dades entre diferents dispositius mitjançant un enllaç per radiofreqüència segura i globalment lliure a 2,4 GHz.

Els principals objectius que es pretén aconseguir amb aquesta norma són facilitar les comunicacions entre equips mòbils i fixes, eliminar cables i connectors entre aquests i oferir la possibilitat de crear petites xarxes inalàmbriques i així facilitar la sincronització de dades entre equips personals.

Les dispositius que solen utilitzar aquesta tecnologia són els dels sectors de les telecomunicacions i la informàtica personal, com les PDAs, telèfons mòbils, ordinadors portàtils, ordinadors personals, impressores i càmeres digitals.

## 6.2. Entorn de treball

L'aplicació està desenvolupada en el llenguatge C++, un llenguatge de programació creat amb la intenció d'estendre el llenguatge de programació C amb mecanismes que permetin la manipulació d'objectes.

Per poder dur a terme el desenvolupament d'una aplicació, en aquest cas C++, cal un kit de desenvolupament de software específic pel llenguatge. Un SDK, sigles de Software Development Kit, o kit de desenvolupament de software és, generalment, un conjunt d'eines de desenvolupament que permeten a un programador crear aplicacions per a un sistema concret, per exemple, certs paquets de software, marcs de treball, plataformes de hardware, ordinadors, sistemes operatius, etc. És tant senzill com una interfície de programació d'aplicacions o API (de l'anglès Application Programming Interface) creada per tal de permetre l'ús d'un cert llenguatge de programació. Les eines més comunes inclouen suport per a la detecció d'errors de programació, codis d'exemple i documentació de suport per ajudar a clarificar certs punts del material de referència primari.

L'SDK utilitzat és el que inclou el Microsoft eMbedded Visual C++ 4.0. Per tal de poder treballar amb un marc de treball específic per a la programació per a Windows Mobile 2003 fa falta un SDK anomenat "Pocket PC 2003 SDK" que s'integra perfectament a l'SDK del Microsoft eMbedded Visual C++ 4.0. Aquestes eines de treball són descarregables gratuïtament des de la pàgina de Microsoft, on, fins i tot, s'hi facilita una gran quantitat d'informació i exemples per incentivar i promoure el desenvolupament d'aquest llenguatge de programació.

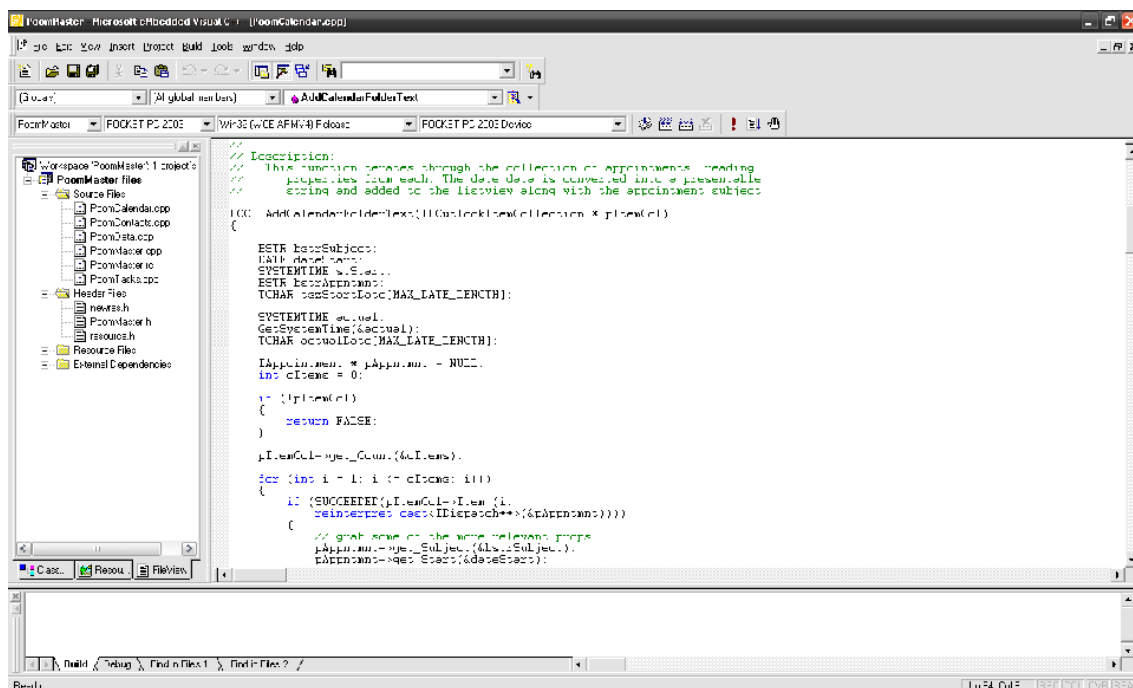


Fig. 6.2.1 – SDK de Microsoft eMbedded Visual C++

Un altre programa de gran importància que comunica el PC amb el Pocket PC és el Microsoft ActiveSync. Microsoft ActiveSync és un programa de sincronització de dades desenvolupat per Microsoft que utilitza la línia de sistemes operatius de Microsoft Windows. Permet als usuaris de Microsoft Windows transferir documents, calendaris, llistes de contactes i email mitjançant un PC i un dispositiu mòbil, com un Pocket PC, un telèfon mòbil o qualsevol altre dispositiu portable que suporti el protocol ActiveSync. ActiveSync està disponible com a descàrrega gratuïta a la web de Microsoft.

És una eina molt còmode a l'hora de fer transferències d'arxius ja que suporta, a més de comunicació per USB, comunicació Bluetooth. És també necessària, o més ben dit, indispensable en el moment d'instal·lar alguns programes.

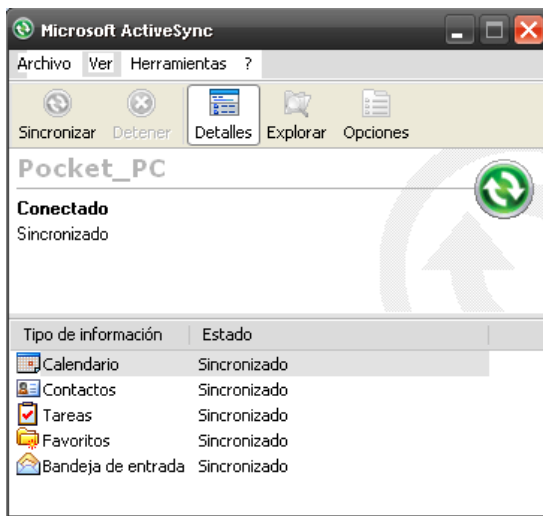


Fig. 6.2.2 – Microsoft ActiveSync per PC

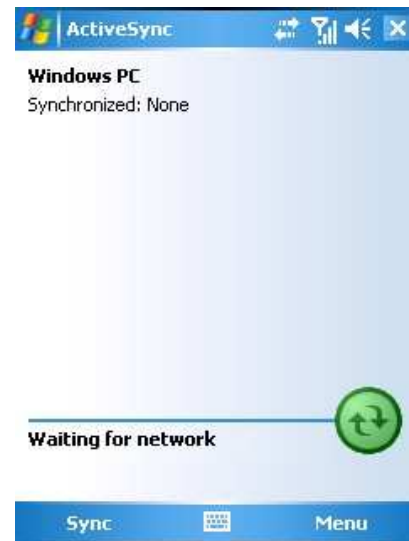


Fig. 6.2.3 – Microsoft ActiveSync per Pocket PC

Per últim, només fa falta anomenar l'aplicació per la qual es pot executar un programa desenvolupat en C++, tant en un PC com al Pocket PC, i que, per tant, conté les llibreries necessàries per tal que aquest funcioni. Aquesta aplicació s'anomena .NET Framework 2.0 i executa programes en C++ d'escriptori. La versió per a Windows Mobile 2003 s'anomena .NET Compact Framework 2.0 i necessita de l'aplicació ActiveSync per tal de poder ser instal·lat al Pocket PC.



### 6.3. Inserció de recordatoris amb Google Calendar

Google Calendar és una agenda i calendari electrònic desenvolupat per Google. Encara que els usuaris no estan obligats a tenir un compte de Gmail, sí que han de disposar d'un compte de Google per a poder utilitzar el software. Així doncs, per poder utilitzar l'eina Google Calendar, primerament, l'administrador s'ha de donar d'alta a Google obrint un compte.

Fins fa poc, per obtenir un compte Google, feia falta que un usuari ja d'un compte Google enviés una invitació per correu electrònic a la persona destinatària, però en aquests moments, un usuari es pot donar d'alta al servei a través de la pàgina d'inici de Gmail ([www.gmail.com](http://www.gmail.com)).

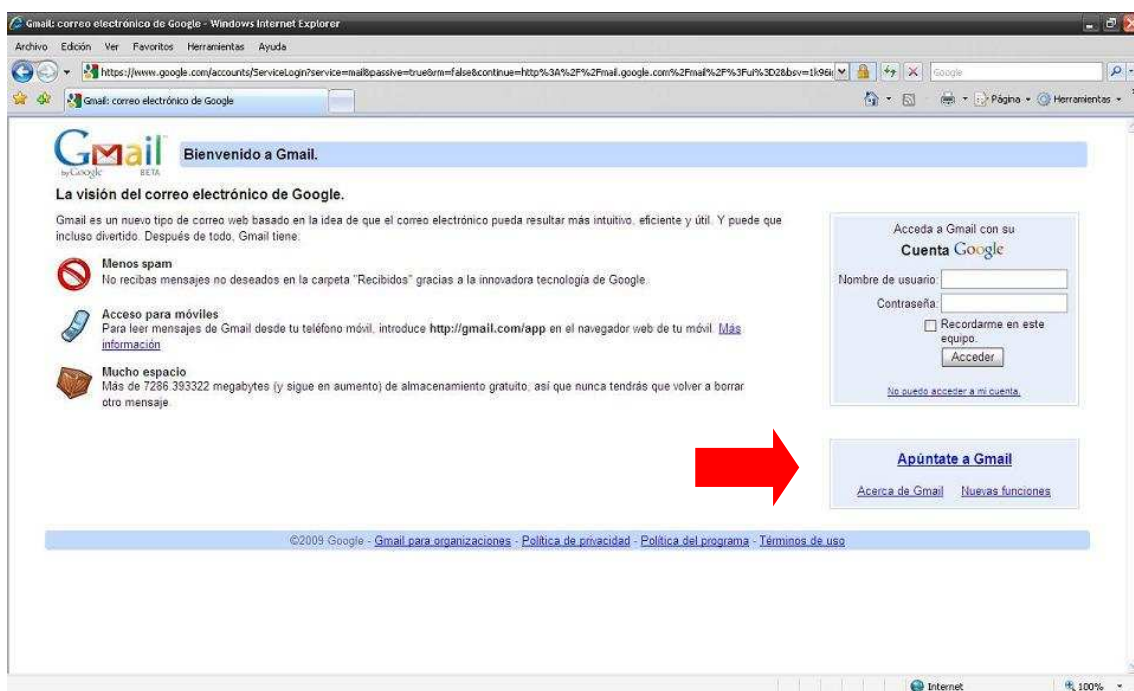


Fig. 6.3.1 – Pantalla d'inici de Gmail

El model conceptual de la comunicació amb el Google Calendar es mostra a la figura següent:

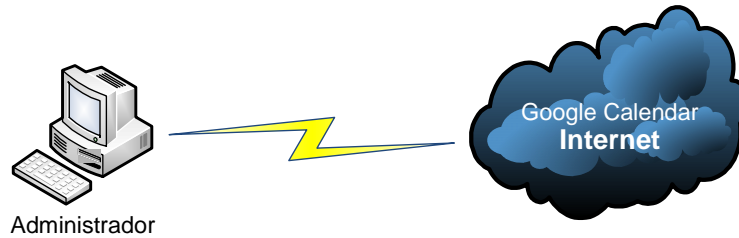


Fig. 6.3.2 – Comunicació administrador – Google Calendar

Un cop l'usuari és donat d'alta, pot accedir a l'eina Google Calendar sempre i quan, òbviament, es loguegi a la pàgina d'inici de Gmail inserint el nom d'usuari i la contrasenya personals. El Google Calendar queda a la part superior esquerra de la pantalla que apareix una vegada l'usuari ha entrat al seu compte Google.

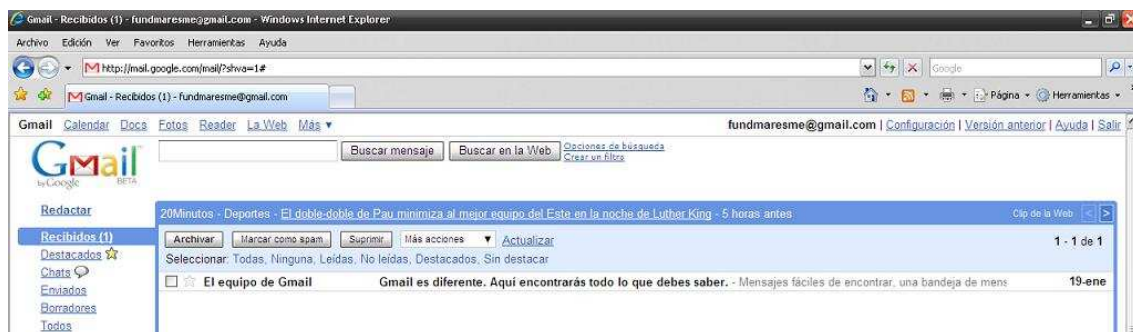


Fig. 6.3.3 – Accés a Google Calendar

La interfície del Google Calendar és semblant a altres utilitats de calendari per escriptori com el Microsoft Outlook o l'iCal per a Mac OS X. Permet als usuaris veure, agregar i moure esdeveniments d'una data a una altra sense recarregar la pàgina. Ofereix una varietat de vistes tals com setmanal, mensual i agenda. Els esdeveniments s'emmagatzemen online, el que significa que el calendari pot ser vist des de molt llocs. En cas de que un usuari experimenti una fallada de disc dur, també significa que no es perd cap tipus d'informació.

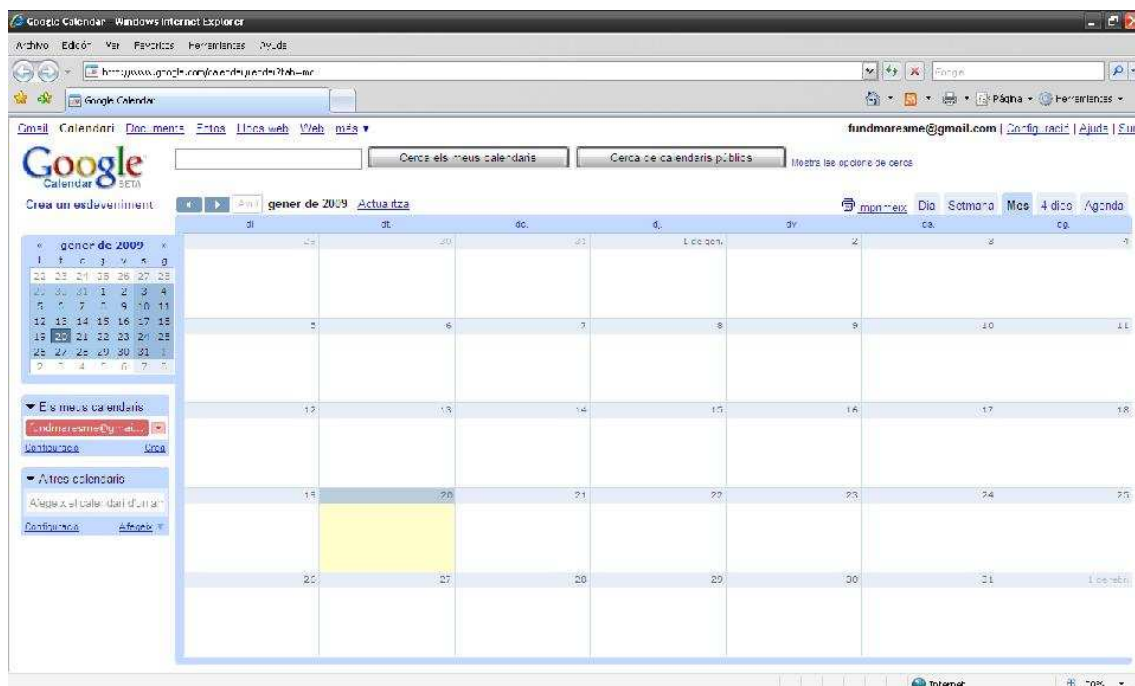


Fig. 6.3.4 – Pantalla d’inici de Google Calendar

Donat que Google Calendar és una aplicació basada en la web, corre virtualment en qualsevol sistema operatiu, sempre i quan el SO tingui un navegador que suporti les tecnologies web necessàries. Degut a aquestes característiques recents del navegador, la compatibilitat de navegador inclou Microsoft Internet Explorer 6+, Mozilla Firefox 1.0+, Opera 9 i Safari 2.0.3.

Per tal d’inserir un nou esdeveniment al calendari, l’usuari pot fer-ho a través de l’enllaç situat a la part esquerra de la pantalla, sota el logotip de Google Calendar, o seleccionant el dia de la quadrícula del calendari que apareix al centre de la pantalla. Si s’opta per a la segona opció, cal pressionar sobre “Edita els detalls de l’esdeveniment” a la finestra emergent que apareix per tal de poder configurar lliurement l’esdeveniment.

En aquest punt es mostra una pantalla per crear un nou esdeveniment amb diferents opcions.

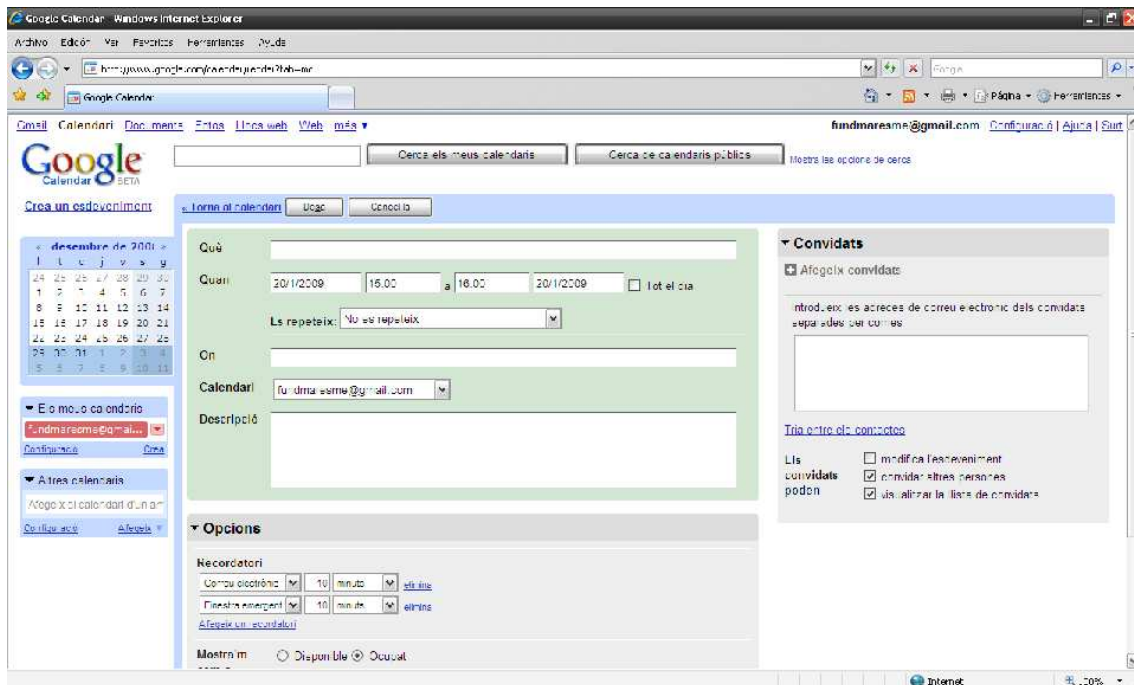


Fig. 6.3.5 – Pantalla de creació d'un nou esdeveniment

Entre les diferents opcions apareix el Què, el Quan, l'On, el Calendari, la Descripció i els recordatoris.

En el Què cal escriure el que és pròpiament l'esdeveniment. És el missatge que apareixerà posteriorment com a recordatori al Pocket PC, així que ha de ser breu, clar i concís.

En el Quan s'ha de desseleccionar la opció de "Tot el dia" i especificar el dia i l'hora en el que el recordatori ha de ser mostrat. L'hora d'inici de l'esdeveniment serà l'hora en què serà notificat i, per tant, l'hora de fi de l'esdeveniment és poc rellevant en aquest sistema. Cal tenir en compte que si es vol fer un recordatori que es vagi mostrant durant un dia, aquest s'ha d'inserir una vegada per cadascun dels avisos que es desitgen.

L'On tampoc té rellevància ja que si és necessari designar algun lloc dins l'esdeveniment, cal fer-ho en el Què.

En el Calendari apareix la llista de calendaris que disposa el compte de Google. La versió gratuïta del programa de sincronització entre Google Calendar i el Pocket PC, GooSync, no suporta més d'un calendari per compte, però és solucionable subscriuint-se al servei Premium i així sincronitzar fins a quatre dispositius per a cada compte Google.

A la Descripció s'han d'incloure les llistes que pugui tenir l'esdeveniment. Un exemple seria una llista de la compra on cadascun dels productes a comprar s'ha d'afegir en una línia diferent.

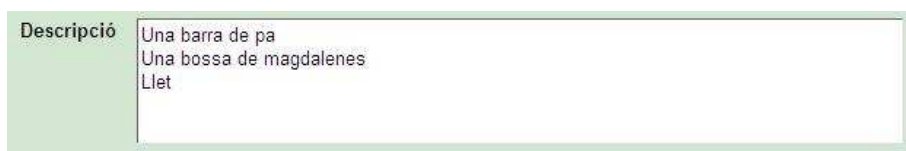


Fig. 6.3.6 – Exemple de llista

Els recordatoris s'han d'eliminar ja que no comporten cap utilitat respecte l'aplicació que s'està desenvolupant.

#### 6.4. Comunicació Google Calendar – Pocket Outlook amb GooSync

GooSync és un servei online que permet la sincronització dels esdeveniments de Google Calendar cap als calendaris integrats dels dispositius mòbils. Pot treballar en qualsevol dispositiu que sigui compatible amb serveis SyncML que inclouen els últims mòbils Nokia, Sony Ericsson i Motorola. També funciona en Pocket PCs i Smatphones amb Windows, Palm OS, Treo BlackBerry i a l'iPhone d'Apple instal·lant un client SyncML.

D'aquesta manera, GooSync permet la sincronització del calendari, els contactes i les tasques entre un dispositiu mòbil i GooSync en ambdues direccions. El dispositiu mòbil es pot sincronitzar amb GooSync inalàmbriquem utilitzant GPRS o Wifi.

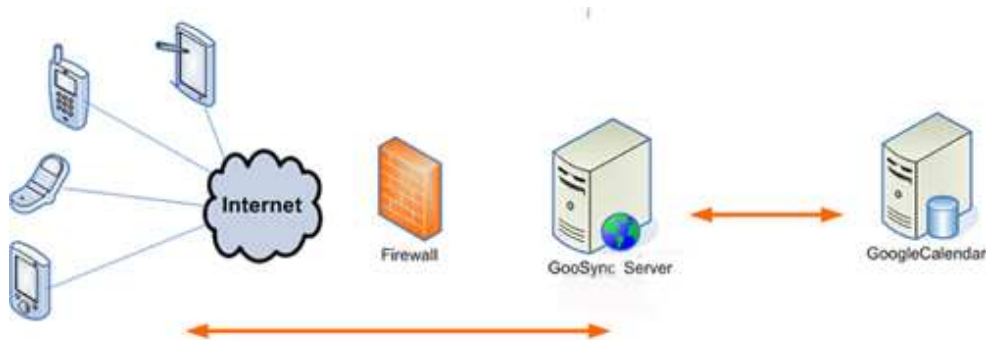


Fig. 6.4.1 – Funcionament de GooSync

Conceptualment, estableix una comunicació com es mostra a la figura següent.

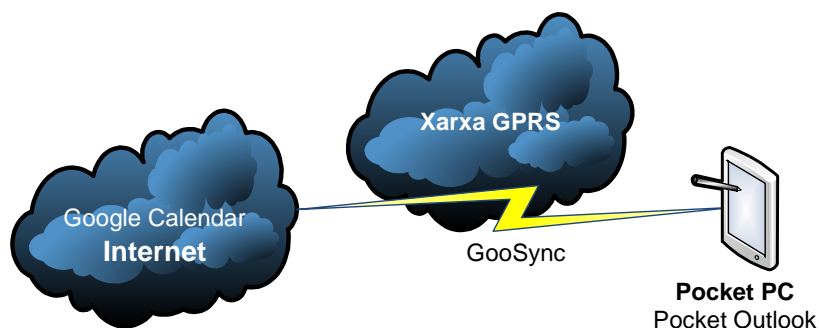


Fig. 6.4.2 – Comunicació Google Calendar – Pocket Outlook

GooSync disposa d'una versió gratuïta i una versió de pagament. Aquesta versió de pagament disposa d'utilitats que la fan indispensable pel tipus d'aplicació desenvolupada. Permet la sincronització amb Google de fins a quatre dispositius utilitzant un sol compte Google, una sistema d'autosincronització pel client del dispositiu Windows Mobile o PalmOS, la configuració d'un interval de sincronització d'esdeveniments de +/- 365 dies i una utilitat que permet seleccionar i sincronitzar esdeveniments de diversos Google Calendars.

Per poder utilitzar el servei, cal registrar-se a la web [www.goosync.com](http://www.goosync.com).

Un cop el compte estigui activat, caldrà seleccionar el tipus de dispositiu mòbil del qual es disposa, en aquest cas, un producte Windows Mobile amb pantalla tàctil.

Posteriorment, la web de [www.goosync.com](http://www.goosync.com) sol·licita l'accés al compte de Google per tal de poder-se connectar amb Google Calendar tot garantint que la web no tindrà accés a la contrasenya ni a la informació personal del compte de Google.

El següent pas tracta de descarregar i instal·lar l'aplicació que facilita la web pel dispositiu. Aquesta instal·lació es pot fer a través del programa Microsoft ActiveSync copiant l'arxiu .cab al dispositiu mòbil i executant-lo per a instal·lar el programa.



Fig. 6.4.3 – Pantalla inicial del GooSync

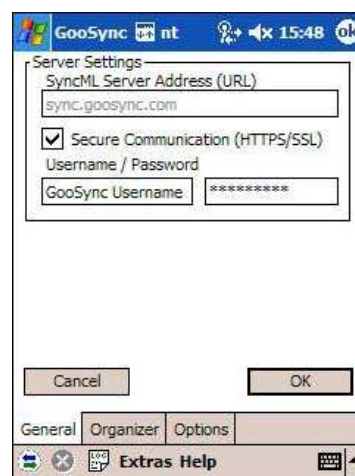


Fig. 6.4.4 – Pantalla d'opcions de GooSync

*Solució adoptada*

Per modificar el perfil opcions, es selecciona la pestanya “Extras” del menú i al menú emergent que apareix, la opció “Opcions”. Caldrà, doncs, introduir el nom d’usuari i la contrasenya utilitzats pel compte GooSync.com.

Per acabar, remarcar que GooSync té diferents modes de sincronització: normal, lenta, actualitzar dispositiu, recarregar dispositiu, actualitzar servidor i recarregar servidor.

La sincronització lenta envia les dades del dispositiu mòbil i del Google Calendar a la web de GooSync i s’envien les dades actualitzades d’un i altra una altra vegada.

La sincronització normal permet que, un cop feta una sincronització completa, poder enviar només els canvis de tal manera que es reenviaran els canvis a ambdós fent que la sincronització sigui més ràpida i utilitzant menys ample de banda. És el mode essencial pels usuaris que sincronitzin per GPRS.

El mode Actualitzar servidor ignora els canvis fets al Google Calendar i insereix en aquest els canvis fets al dispositiu mòbil.

El mode Recarregar servidor borra tot el calendari del dispositiu mòbil i insereix el calendari del Google Calendar.

El mode Actualitzar dispositiu ignora els canvis fets al dispositiu mòbil i insereix en aquest els canvis fets al Google Calendar.

El mode Recarregar servidor borra tot el calendari del Google Calendar i insereix el calendari del dispositiu mòbil.



Tenint en compte el fet de si s'inseriran nous esdeveniments o no des del propi Pocket PC, cal estudiar el millor mode de sincronització per tal que sigui el més econòmic possible. Si es decideix el fet de poder inserir recordatoris directament a través de l'eina Pocket Outlook del Pocket PC, és recomanable el mode normal ja que és el mode bidireccional més econòmic. D'altra banda, si es decideix que no s'ha de contemplar l'actualització de la llista de notificacions a través del Pocket Outlook, l'alternativa més econòmica és el mode Actualitzar dispositiu.

## 6.5. Estructura i funcionament de l'aplicació

L'aplicació està desenvolupada en el llenguatge de programació C++ amb el kit de desenvolupament de software Microsoft eMbedded Visual C++ 4.0 i està destinada a ser executada en Pocket PCs amb el sistema operatiu Windows Mobile 2003.

L'aplicació es troba estructurada en dues parts: la primera part tracta de comunicar el Pocket Outlook del Pocket PC amb la pròpia aplicació, i la segona part pretén notificar els esdeveniments per la pantalla del Pocket PC quan sigui el moment necessari.

Per dur a terme la primera part, la comunicació entre Pocket Outlook i pròpiament l'aplicació, es fa ús de l'API Pocket Outlook Object Model (POOM) que incorpora el llenguatge C++.

Per tal de poder inicialitzar i, per tant, utilitzar l'API POOM, s'han de dur a terme dos passos. El primer pas consisteix en utilitzar el mètode CoCreateInstance per crear una instància de la classe IPOutlookApp. Aquest mètode és l'únic que permet crear instàncies de IPOutlookApp i, pròpiament, inicialitzar POOM. A més a més, aquesta instància IPOutlookApp serà única a l'aplicació. El segon pas consisteix en loguejar la instància IPOutlookApp a l'aplicació a través de la instrucció Logon.

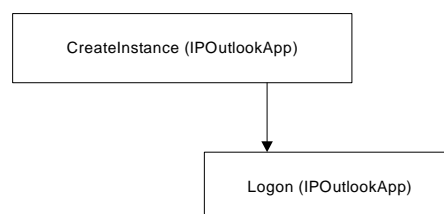


Fig. 6.5.1 – Seqüència d'inici de POOM

La resta de mètodes que utilitza l'API POOM són senzills d'entendre i no són necessaris d'explicar ja que es dedueixen instintivament en el moment de voler realitzar algun consulta a la base de dades de Pocket Outlook.

L'aplicació comença inicialitzada en segon pla, en el moment en que s'encén el Pocket PC, i no es fa veure fins al moment en què un esdeveniment ha de ser notificat. L'aplicació, en iniciar-se, crea una col·lecció de la totalitat d'esdeveniments de la base de dades del Pocket Outlook extraient-ne el Què, la Descripció i la Data d'Inici de cadascun d'ells. La data es converteix a un format String per tal de poder-la comparar d'alguna manera amb la data instantània local. Si aquestes coincideixen, l'element esdeveniment és emmagatzemat a una segona col·lecció. Aquesta comparació es realitza amb cadascun dels esdeveniments existents a la col·lecció i es van afegint a aquesta segona llista un a un segons si la data d'inici de l'esdeveniment és la mateixa que la data i hora locals actuals.

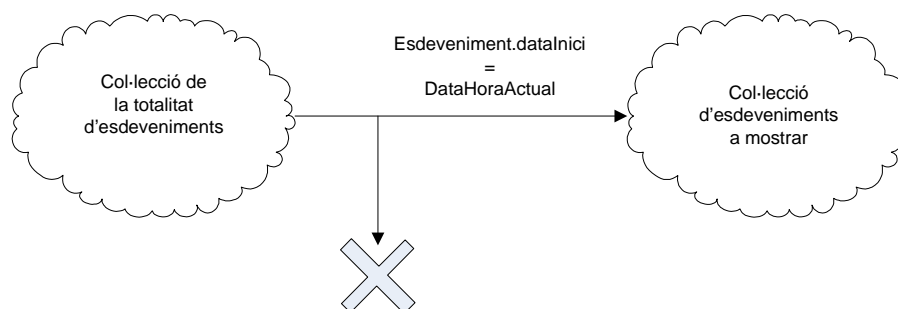


Fig. 6.5.2 – Comparació de dates

Diversos esdeveniments a la segona col·lecció signifiquen un seguit d'esdeveniments que han de ser mostrats a la vegada. Així doncs, segons el nombre d'esdeveniments, l'espai de mostratge per a cadascun d'ells es veurà redimensionat. Si a més a més, aquest esdeveniment disposa de descripció, i per tant, és una llista, la llista es concatena a l'String del Què i també apareix per pantalla. Cal anar en compte amb les llistes ja que si s'ha de mostrar més d'un esdeveniment i un d'ells és una llista, l'espai serà reduït i, segons com, la llegibilitat de la llista es veurà afectada.

Al mateix moment en què apareixi l'esdeveniment per pantalla, s'escoltarà un to per l'audiòfon del Pocket PC de tal manera que alerti a l'usuari d'un nou avís. En tocar qualsevol part de la pantalla, el missatge desapareixerà i l'aplicació es mantindrà en segon pla.



Fig. 6.5.3 - Notificació individual



Fig. 6.5.4 – Notificació de diversos esdeveniments

El refresc de la col·lecció d'esdeveniments a mostrar es farà cada minut, és a dir, cada 60 segons l'aplicació en segon pla farà la comparació entre les dates inici de cadascun dels elements esdeveniment de la col·lecció total i l'hora actual local. Si aquests coincideixen, un nou avís serà mostrat. D'altra banda, l'aplicació es mantindrà en segon pla. El temps de refresc de la col·lecció és degut a que 60 segons és la unitat mínima de diferència temporal entre dos esdeveniments consecutius i, d'aquesta manera, assegura un avís, en quant a temps, molt acurat.

El refresc de la col·lecció d'esdeveniments total es farà en el moment d'encendre el Pocket PC, o el que és el mateix, el moment en inicialitzar l'aplicació i, a partir de llavors, cada 60 minuts. Aquest valor és agafat aleatòriament però amb l'objectiu de no sobrecarregar el Pocket PC amb temps de procés sense massa sentit. A més a més, deixa un marge suficientment gran a l'administrador per tal de mantenir els esdeveniments d'última hora registrats a la base de dades del Pocket PC

Al moment d'apagar el Pocket PC, l'aplicació es tancarà.

## 7. Pressupost

El pressupost es divideix en dues parts: l'adquisició de les llicències de GooSync i la tarifa de les connexions GPRS.

Les llicències de GooSync contemplen fins a quatre dispositius vinculats a Google Calendar i són interessants i necessàries ja que habiliten a l'aplicació la funcionalitat d'autosincronització. Es poden adquirir a la pròpia web de [www.goosync.com](http://www.goosync.com) i tenen assignats els preus de la taula següent.

Llicència	Cost
Subscripció única	\$39.95 = 30.93€
Subscripció durant 24 mesos	\$29.95 = 23.19€
Subscripció durant 12 mesos	\$19.95 = 15.45€

Taula 7.1 – Taula de costos de les llicències de GooSync

En aquest aspecte, cal meditar l'obtenció de les llicències de Subscripció única segons el nombre d'usuaris del servei o decidir un període de prova de, per exemple, 12 mesos per tal de testejar el sistema i veure si realment el servei és útil o queda obsolet.

Per altra banda, cal estudiar si l'ús del servei és viable o no econòmicament. A més a més, GPRS és una tecnologia cara i cal estudiar la millor opció en aquest sentit.

Les operadores espanyoles de telefonia mobil tenen ofertes molt semblants per a GPRS, amb i sense abonaments. Els abonaments permeten obtenir un preu més barat per Kb, però tenen dos problemes: si es consumeix menys, l'abonament es paga igualment i, si es consumeix més, el preu fora de l'abonament és superior. Els preus sense IVA de Telefonica són els que s'indiquen a la taula següent.

Movistar	Quota	Preu MB	Preu KB	Consum mín.
Sense abonament		20,48 €	0,02 €	
Mòdul Dades A	10,00 €	1,54 €	0,0015 €	6,5 MB
Mòdul Dades B	15,00 €	1,02 €	0,001 €	14 MB
Mòdul Dades C	30,00 €	0,51 €	0,0005 €	58 MB

Taula 7.2 – Tarifes del servei GPRS de Telefonica

Si es té en compte que l'abonament mensual més barat té un cost de 10€ de quota més un cost fix de 10,01€ i que, 1MB sense abonament té un cost de 20,48€, el punt d'amortització és 1MB.

És difícil definir el volum de dades que comporta l'enviament d'un esdeveniment a través de la xarxa GPRS i la freqüència amb la que s'utilitzarà el servei, així que tot dependrà d'aquests dos factors i del que després Telefònica realment cobrarà. Si més no, caldrà un període de proves per treure'n conclusions.

## 8. Continuitat del projecte

La continuïtat del projecte només pot suposar la incorporació de noves funcionalitats a l'aplicació o al fet de desenvolupar o integrar a l'aplicació ja existent la part de comunicació.

El desenvolupament de la part de comunicació entre Google Calendar i el Pocket Outlook del Pocket PC és una part d'aquest projecte que és substituïda per una aplicació externa que realitza aquest servei. Aquesta part de projecte reduiria costos de cara a l'adquisició de llicències que són necessàries pel bon funcionament i facilitaria la incorporació de futures funcionalitats que poden ser interessants segons l'ús que se li vulgui donar a l'aplicació.

Per treballar en aquesta funcionalitat fan falta coneixements de XML i una bona planificació del projecte per tal que aquest suposi una inversió i no una despesa.





## 9. Conclusions

L'aplicació desenvolupada satisfà els requeriments especificats en la proposta de la Fundació ja que contempla i soluciona tant la part de comunicació, entre un administrador extern i els diferents usuaris del servei, com la part de visualització, que pretén gestionar la notificació d'esdeveniments en el Pocket PC.

El cost d'utilització del servei és variable ja que depèn del nombre d'esdeveniments per dia i el nombre d'usuaris del servei tenint en compte que la tecnologia de comunicació GPRS utilitzada i, en aquest cas, la única disponible, és sensiblement cara. D'altra banda, el cost econòmic del manteniment del projecte és, però, reduït ja que només cal renovar les llicències del programa de comunicació del Google Calendar amb el Pocket Outlook del dispositiu si és que no es vol fer una inversió inicial i despreocupar-se d'aquesta despesa en aquest sentit.

L'únic però fa referència a la limitació en qualitat de l'audiòfon del Pocket PC Movistar TSM 500 i que restringeix la utilització d'un sistema TTS que afegiria una bona funcionalitat al servei de l'aplicació.

Per últim, només comentar que el fet de poder col·laborar amb una fundació que contribueix en iniciatives socials per a persones discapacitades dóna un aire de contribució en la societat que afegeix un valor personal al projecte.



## Annex I – Glossari de termes

### SyncML

El SyncML és un protocol de la família XML utilitzat per a proveir de sincronització remota als dispositius mòbils. Es troba integrat en molts dispositius mòbils, com telèfons mòbils i PDAs.

SyncML és una eina multiplataforma que permet l'intercanvi i la sincronització de dades a través de xarxes remotes als dispositius mòbils, el que significa que no cal un ordinador a prop per tal de d'actualitzar l'agenda o guia.

També serveix per fer còpies de seguretat de l'agenda del mòbil.

### Objecte

Els objectes són entitats que combinen estat, comportament i identitat:

L'estat està compost per dades. Són un o diversos atributs als que se'ls hi assigna uns valors concrets, les dades.

El comportament està definit pels procediments o mètodes amb els que pot operar l'objecte, és a dir, quines operacions pot realitzar amb ell.

La identitat és una propietat d'un objecte que el diferencia de la resta, dit en altres paraules, és el seu identificador.

### XML

XML, sigles en anglès de Extensible Markup Language (“Llenguatge de Marques Ampliable”), és un metallenguatge extensible d'etiquetes desenvolupat pel World Wide Web Consortium (W3C). És una simplificació i adaptació del SGML i permet definir la

gramàtica de llenguatges específics, de la mateixa manera que HTML és a la vegada un llenguatge definit per SGML. Per tant, XML no és realment un llenguatge en particular, sinó una manera de definir llenguatges per a diferents necessitats.

XML no ha nascut només per la seva aplicació a Internet, sinó que es proposa com a estàndard per a l'intercanvi d'informació estructurada entre diferents plataformes. Es pot utilitzar en bases de dades, editors de text, fulls de càlcul i casi qualsevol cosa imaginable.

XML és una tecnologia senzilla que té al seu voltant altres que la complementen i la fan molt més gran i amb unes possibilitats majors. Té un paper molt important a l'actualitat ja que permet la compatibilitat entre sistemes per compartir la informació d'una manera segura, fiable i fàcil

### TTS

La conversió text-veu és la generació, per mitjans automàtics, d'una veu artificial que genera un so idèntic al produït per una persona en llegir un text quelasevo en veu alta. És a dir, són sistemes que permeten la conversió de textos en veu sintètica. Els conversors de text-veu son coneguts amb les sigles CTV o per les sigles TTS (Text to Speech).

### API

Una Interfície de Programació d'Aplicacions o API, de l'anglès Application Programming Interface, és el conjunt de funcions i procediments, o mètodes si es refereix a la programació orientada a objectes, que ofereix una certa biblioteca per ser utilitzada per un altre software com una capa d'abstracció i així ser utilitzada com a interfície de comunicació entre components software.

Es tracta, doncs, del conjunt de crides a certes biblioteques que ofereixen accés a certs serveis des dels processos i representa un mètode per aconseguir abstracció a al programació, generalment, encara que no necessàriament, entre nivells o capes inferiorsi superiors de software. Un dels principals propòsits d'una API consisteix en proporcionar

un conjunt de funcions d'ús general, per exemple, per dibuixar finestres i icones a la pantalla. D'aquesta forma, els programadors es beneficien dels avantatges de l'API fent ús de la funcionalitat, evitant el treball de programar-ho tot des del principi.

## PIM

PIM, sigles de Personal Information Management, és un tipus d'aplicació software que funciona com a organitzador personal. S'utilitza, principalment, per facilitar la gestió de certs tipus d'informació personal com poden ser notes personals, llistes de tasques, calendaris, llistes de contactes.

A més a més, alguns productes PIM son capaços de sincronitzar dades amb una altra eina PIM a través d'una xarxa informàtica.



## Annex II - Pocket Outlook Object Model (POOM)

El Pocket Outlook Object Model és un reflex del Microsoft Office Outlook Object Model però la seva funcionalitat es redueix al fet de poder albergar les restriccions pràctiques dels dispositius mòbils. POOM és una llibreria que dona accés a les dades i als objectes del Outlook Mobile Personal Information Management (PIM) o Administrador de Dades Personals de Pocket Outlook. Permet un marc de treball orientat a objectes per crear, modificar, i notificar esdeveniments, tasques i contactes i per manipular les carpetes que els contenen.

El conjunt d'APIs POOM que formen part del Windows Embedded CE són anomenades POOM1. A més a més d'aquestes APIs POOM, el Windows Mobile conté un conjunt addicional d'API's POOM que són específiques per a Windows Mobile. Aquestes APIs POOM addicionals són anomenades POOM2.

L'arquitectura de POOM s'organitza de la següent manera:

La interfície primària del Pocket Outlook Object Model (POOM) és l'objecte de l'aplicació Pocket Outlook IPOutlookApp. Aquest és l'únic objecte POOM que el mètode CoCreateInstance pot crear, amb tots els altres objectes derivats d'ell. Després de loguejar-se a l'objecte de l'aplicació utilitzant IPOutlookApp::Logon, es poden crear objectes PIM utilitzant IPOutlookApp::CreateItem.

Un objecte Folder conté un objecte IPOutlookItemCollection, que té una col·lecció d'elements Outlook. Els Folders són essencialment contenidors per les bases de dades de contactes, calendari i tasques. A més a més, hi ha un Folder Infrared per transferir elements a través d'un port d'infrarojos utilitzant IFolder::SendInfrared.

La implementació del Folder POOM és un subconjunt de l'objecte Folder d'Outlook de tal manera que cada tipus d'element es troba associat amb un OIDefaultFolder

L'objecte `IPOutlookItemCollection` s'obté de l'objecte `Folder` utilitzant `IFolder::get_Items`. Una col·lecció d'elements és una col·lecció d'objectes contacte, tasca i esdeveniment. Es poden recuperar elements individuals utilitzant `IPOutlookItemCollection::Item`, esborrar elements individuals utilitzant `IPOutlookItemCollection::Remove`, i crear elements individuals utilitzant `IPOutlookItemCollection::Add`. Amb la col·lecció d'elements, es poden realitzar filtratges sobre una col·lecció d'objectes utilitzant `IPOutlookItemCollection::Restrict`.

Es poden assignar i recuperar propietats de l'element Outlook utilitzant els mètodes `get/put` associats amb cada tipus d'element Outlook.

Els elements tasca i esdeveniment suporten l'objecte `IRecurrencePattern`. Amb aquest objecte, es pot generar una repetició per una tasca o esdeveniment. Els elements esdeveniment suporten la col·lecció `IRecipients`, amb la que es poden especificar `Meeting Recipients`. Un element esdeveniment que té una col·lecció de `Recipients`, s'anomena també `Meeting`.



## Annex III – Codi generat per l'aplicació

```

// *****
// POOMDATA.CPP

// Includes
#include "PoomMaster.h"

// Globals
IPOutlookApp * g_polApp = NULL;

// *****
// Function Name: InitPoom
//
// Purpose: Creates and logs on to the POOM application object
//
// Arguments:
// none
//
// Return Values:
// BOOL
// returns TRUE if POOM was initialized, FALSE otherwise
//
// Description:
// This function creates the POOM application COM object and starts a POOM
// session via Logon. This must be done before any subsequent POOM calls are
// made.
BOOL InitPoom(HWND hwnd)
{
    ;
    BOOL bSuccess = FALSE;

    if (SUCCEEDED(CoInitializeEx( NULL, 0)))
    {
        // Now, let's get the main outlook application
        if (SUCCEEDED(CoCreateInstance(CLSID_Application, NULL,
CLSCTX_INPROC_SERVER,
IID_IPOutlookApp, reinterpret_cast<void **>(&g_polApp))))
        {
            // login to the Pocket Outlook object model
            if(SUCCEEDED(g_polApp->Logon(NULL)))
            {
                // can't login to the app
                bSuccess = TRUE;
            }
        }
    }

    return bSuccess;
}

```

## Annex II – Pocket Outlook Object Model (POOM)

```

// *****
// Function Name: GetPoomFolder
//
// Purpose: Gets the default folder for the specified POOM item type
//
// Arguments:
//   IN int nFolder - POOM folder type to return, olFolderTasks,
//                 olFolderContacts, or olFolderCalendar in this app
//   OUT IFolder **ppFolder - pointer to IFolder interface returned.
//                           Must be released by caller of GetPoomFolder
//
// Return Values:
//   BOOL
//   returns TRUE if the folder interface was retrieved, FALSE otherwise
//
// Description:
//   This function simply encapsulates a call on the global Outlook app interface. The
//   returned pointer is simply passed through this function.
BOOL GetPoomFolder(int nFolder,
                  IFolder **ppFolder)
{
    if (SUCCEEDED(g_polApp->GetDefaultFolder(nFolder, ppFolder)))
    {
        return TRUE;
    }
    else
    {
        return FALSE;
    }
}

// *****
// Function Name: GetPoomApp
//
// Purpose: Gets pointer to POOM application interface
//
// Arguments:
//   OUT IFolder **ppFolder - pointer to IPOutlookApp interface returned.
//                           Must be released by caller of GetPoomApp
//
// Return Values:
//   BOOL
//   returns TRUE if the folder interface was retrieved, FALSE otherwise
//
// Description:
//   This function simply returns a pointer to the global Outlook app interface
//   The returned pointer is simply passed through this function after being AddRef'd.
BOOL GetPoomApp(IPOutlookApp **ppOutApp)
{
    if (g_polApp)
    {
        g_polApp->AddRef();
        *ppOutApp = g_polApp;
        return TRUE;
    }
}

```

```

    else
    {
        return FALSE;
    }
}
// *****
// Function Name: PopulateList
//
// Purpose: Populates listbox with POOM items from given folder
//
// Arguments:
//     IN int nFolder - POOM folder to read/add items from. In this app, either
//                   olFolderTasks, olFolderContacts, or olFolderCalendar
//
// Return Values:
//     BOOL
//     returns TRUE if the items were successfully accessed, FALSE otherwise
//
// Description:
//     This function reads each POOM item from the given category (Tasks, Contacts,
//     or Appointments) and adds each to the main listbox control
BOOL PopulateList(int nFolder)
{
    IFolder * pCurrFldr = NULL;
    IPOutlookItemCollection * pItemCol = NULL;
    // Get the folder and its items, depending on
    // the folder type
    if (GetPoomFolder(nFolder, &pCurrFldr))
    {
        if (SUCCEEDED(pCurrFldr->get_Items(&pItemCol))
        {
            switch (nFolder)
            {
                case olFolderTasks:
                    AddTaskFolderText(pItemCol);
                    break;
            }

            pItemCol->Release();
            pCurrFldr->Release();
            return TRUE;
        }

        pCurrFldr->Release();
    }

    return FALSE;
}
}

```

*Annex III – Codi generat per l'aplicació*

```
// *****  
// Function Name: ShutdownPoom  
//  
// Purpose: Shuts down the current POOM session  
//  
// Arguments:  
//     none  
//  
// Return Values:  
//     none  
//  
// Description:  
// This function logs off the current POOM session and releases the POOM  
// application interface pointer  
void ShutdownPoom()  
{  
  
    if (g_polApp)  
    {  
        g_polApp->Logoff();  
        g_polApp->Release();  
    }  
  
    CoUninitialize();  
  
}
```

*Assistent personal per persones amb necessitats especials*

```

// *****
// POOMCALENDAR.CPP
//
// Includes
#include "PoomMaster.h"
#include <time.h>

// Constants
#define MAX_DATE_LENGTH 28

// Forward Declarations
void FillDateString(LPSYSTEMTIME pst,
                  TCHAR * tszDate,
                  int nLen);

// *****
// Function Name: AddCalendarFolderText
//
// Purpose: Reads and adds existing POOM appointments to the main listbox
//
// Arguments:
// IN IPOutlookItemCollection * pItemCol - ptr to the Calendar's default folder
// collection interface
//
// Return Values:
// Describes the return type and meaning of each value. E.g.:
// BOOL
// returns TRUE if the item was successfully read and added. Otherwise FALSE
//
// Description:
// This function iterates through the collection of appointments, reading
// properties from each. The date data is converted into a presentable
// string and added to the listview along with the appointment subject

BOOL AddCalendarFolderText(IPOutlookItemCollection * pItemCol)
{
    BSTR bstrSubject;
    DATE dateStart;
    SYSTEMTIME stStart;
    BSTR bstrAppntmnt;
    TCHAR tszStartDate[MAX_DATE_LENGTH];

    SYSTEMTIME actual;
    GetSystemTime(&actual);
    TCHAR actualDate[MAX_DATE_LENGTH];

    IAppointment * pAppntmnt = NULL;
    int cItems = 0;

    if (!pItemCol)
    {
        return FALSE;
    }
}

```

## Annex III – Codi generat per l'aplicació

```

pItemCol->get_Count(&cItems);

for (int i = 1; i <= cItems; i++)
{
    if (SUCCEEDED(pItemCol->Item (i,
        reinterpret_cast<IDispatch**>(&pAppntmnt))))
    {
        // grab some of the more relevant props
        pAppntmnt->get_Subject(&bstrSubject);
        pAppntmnt->get_Start(&dateStart);

        VariantTimeToSystemTime(dateStart,
            &stStart);

        // convert numerical date to a formatted string
        FillDateString(&stStart, tszStartDate, MAX_DATE_LENGTH);
        FillDateString(&actual, actualDate, MAX_DATE_LENGTH);

        //allocate a BSTR to store the Appointment information
        bstrAppntmnt = SysAllocStringLen(NULL, SysStringLen(bstrSubject) +
            _tcslen(tszStartDate)+ (2*(sizeof(TCHAR)))));
        memset(bstrAppntmnt, 0, SysStringLen(bstrAppntmnt)*sizeof(TCHAR));

        //copy thing Appointment subject and date into the same string
        _tcscpy(bstrAppntmnt, bstrSubject, SysStringLen(bstrSubject));
        _tcscat(bstrAppntmnt, _T("\n"));
        //_tcscat(bstrAppntmnt, tszStartDate);
        _tcscat(bstrAppntmnt, actualDate);

        time_t atime;
        time(&atime);
        tm *ltime = localtime(&atime);
        _tcscat(bstrAppntmnt, _T("Year::%d Month::%d Day:: %d\n", ltime->tm_year,
            ltime->tm_mon, ltime->tm_mday));

        if(strcmp(tszStartDate,atime) {
            _tcscat(bstrAppntmnt, tszStartDate);
        }

        // add the string to the listview control
        AddToList(bstrAppntmnt);

        // clean up
        pAppntmnt->Release();
        SysFreeString(bstrAppntmnt);
        SysFreeString(bstrSubject);
        bstrAppntmnt = NULL;
        bstrSubject = NULL;
        tszStartDate[0] = _T('\0');
    }

    else
    {
        return FALSE;
    }
}

```

```

    }

    return TRUE;

}

// *****
// Function Name: FillDateString
//
// Purpose: provide readable date data by formatting SYSTEMTIME data
//
// Arguments:
// IN LPSYSTEMTIME - ptr to struct containing date and time information
// OUT TCHAR tszDate - buffer for returned formatted string
//     IN int nLen - length (in chars) of tszDate
//
//
// Return Values:
// void
//
// Description:
// This function simply creates a human-readable date string from a given
// SYSTEMTIME struct. the output should look like: "1/02/2003 - 17:49"
void FillDateString(LPSYSTEMTIME pst,
                  TCHAR * tszDate,
                  int nLen)
{
    TCHAR tmp[5];
    _itot(pst->wMonth, tmp, 10);
    _tcscpy(tszDate, tmp);
    _tcscat(tszDate, _T("/"));

    _itot(pst->wDay, tmp, 10);
    _tcscat(tszDate, tmp);
    _tcscat(tszDate, _T("/"));

    _itot(pst->wYear, tmp, 10);
    _tcscat(tszDate, tmp);
    _tcscat(tszDate, _T(" - "));

    _itot(pst->wHour, tmp, 10);

    _tcscat(tszDate, tmp);
    _tcscat(tszDate, _T(":"));

    if (pst->wMinute < 10)
        _tcscat(tszDate, _T("0"));

    _itot(pst->wMinute, tmp, 10);
    _tcscat(tszDate, tmp);
}

```

## Annex III – Codi generat per l'aplicació

```

// *****
// POOMMASTER.CPP
//
//
// Defines
#define INITGUID
#define MAX_LOADSTRING 100
#define MAX_LISTVIEW_ITEMS 32
#define FT_BOLD 0x1
#define FT_STRIKE 0x2
// Includes
#include <windows.h>
#include <pimstore.h>
#include <aygshell.h>
#include "PoomMaster.h"

// Global Variables:
HINSTANCE          g_hInst;                // The current instance
handle
HWND               g_hwndLV;              // The ListView
control

HFONT              g_hfBaseFont;
HWND               g_hwndMain;
static int listcount = 0;
static SHACTIVATEINFO s_sai;

int g_nCurrFolder = olFolderCalendar;
// Forward declarations of functions included in this code module:

LRESULT CALLBACK  WndProc                  (HWND, UINT, WPARAM, LPARAM);

HWND              CreateListView(HWND hwndParent);
void              RefreshListView(HWND hwndLV);
BOOL              InitInstance(HINSTANCE hInstance, int nCmdShow);
HFONT             BuildFont(HDC hDC, HFONT hBaseFont, DWORD dwFlags);

// *****
// Function Name: WinMain
//
// Purpose: Application entry point
//
// Arguments:
//     IN HINSTANCE hInstance - handle to application instance
//     IN HINSTANCE hPrevInstance - not used
//     IN LPTSTR    lpCmdLine - command line string, not used
//     IN int nCmdShow - not used
//
// Return Values:
//     int
//     Application exit code

```



*Assistent personal per persones amb necessitats especials*

```

//
// Description:
// This function defines the standard application entry point. It starts
// initialization and creates the main application dialog box
int WINAPI WinMain( HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPTSTR lpCmdLine,
                   int nCmdShow)
{
    HACCEL hAccelTable;

    // Perform application initialization:
    INITCOMMONCONTROLSEX comctrex = { 0 };

    comctrex.dwSize = sizeof(INITCOMMONCONTROLSEX);
    comctrex.dwICC = ICC_DATE_CLASSES;
    InitCommonControls();
    InitCommonControlsEx(&comctrex);
    g_hInst = hInstance;
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }

    hAccelTable = LoadAccelerators(hInstance, (LPCTSTR)IDC_POOMMASTER);

    DialogBox(hInstance, MAKEINTRESOURCE(IDD_MAIN), 0, (DLGPROC)WndProc);
    return 0;
}

// *****
// Function Name: InitInstance
//
// Purpose: Checks for existing instance of the app and activates it if existing
//
// Arguments:
//     none
//
// Return Values:
//     BOOL
//     returns TRUE if there is no previously existing instance, FALSE otherwise
//
// Description:
// This function tries to find an existing instance of the application window.
// This is a simple check, looking only for the name "PoomMaster" and does not
// check for the window class. If the window is found, it is brought to the
// foreground
BOOL InitInstance(HINSTANCE hInstance,
                 int nCmdShow)
{
    HWND hWnd = NULL;
    TCHAR szTitle[MAX_LOADSTRING]; // The title bar text

    g_hInst = hInstance; // Store instance handle in our global variable

```

## Annex III – Codi generat per l'aplicació

```

// Initialize global strings

LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);

//If it is already running, then focus on the window
hWnd = FindWindow(NULL, szTitle);
if (hWnd)
{
    // set focus to foremost child window
    // The "| 0x01" is used to bring any owned windows to the foreground and
    // activate them.
    SetForegroundWindow((HWND)((ULONG) hWnd | 0x00000001));
    return FALSE;
}

return TRUE;
}

// *****
// Function Name: WndProc
//
// Purpose: Processes messages for the main dialog window
//
// Arguments:
//   IN HWND hDlg - handle to the dialog window
//   IN UINT message - identifier for message to be processed
//   IN WPARAM wParam - message parameter
//   IN LPARAM lParam - message parameter
//
// Return Values:
//   LRESULT
//   The return value is the result of the message processing and depends on
//   the message sent.
//
// Description:
//   This is a standard dialog window message processing function. It handles
//   the following:
//   WM_COMMAND - menu events
//   WM_INITDIALOG - dialog initialization
//   WM_DESTROY - window cleanup
//   WM_ACTIVATE - shell activation notification
//   WM_SETTINGCHANGE - system wide setting change notification
//   WM_DRAWITEM - owner-drawn painting of listview control
//   WM_MEASUREITEM - owner-drawn sizing of listview items
LRESULT CALLBACK WndProc(HWND hWnd,
                          UINT message,
                          WPARAM wParam,
                          LPARAM lParam)
{
    HDC hdc;
    LPDRAWITEMSTRUCT lpdis;
    LPMEASUREITEMSTRUCT lpmi;

    switch (message)
    {
        case WM_COMMAND:

```

## Assistent personal per persones amb necessitats especials

```

// Parse the menu selections:
switch (LOWORD(wParam))
{

case IDOK:
case IDM_QUIT:

    EndDialog(hWnd, 0);
    break;

/* case IDM_NEWITEM:
    switch (g_nCurrFolder)
    {
    case olFolderCalendar:
        if (DialogBox(g_hInst, (LPCTSTR)IDD_NEWAPPNTMNT, hWnd,
(DLGPROC)NewAppntmntProc) == IDOK)
        {
            RefreshListView(g_hwndLV);
        }
        break;

    }

    break;
case IDM_MODE_CALENDAR:
    g_nCurrFolder = olFolderCalendar;
    RefreshListView(g_hwndLV);
    SetWindowText(hWnd, _T("POOM Calendar"));
    break;

*/
    }
    break;
case WM_INITDIALOG:
    SHINITDLGINFO shidi;

    shidi.dwMask = SHIDIM_FLAGS;
    shidi.dwFlags = SHIDIF_DONEBUTTON | SHIDIF_SIPDOWN |
SHIDIF_SIZEDLGFULLSCREEN;
    shidi.hDlg = hWnd;
    SHInitDialog(&shidi);

    g_hwndMain = hWnd;
    // Initialize the shell activate info structure
    memset (&s_sai, 0, sizeof (s_sai));
    s_sai.cbSize = sizeof (s_sai);

    if (InitPoom(hWnd))
    {

        // Create the ListView control
        g_hwndLV = CreateListView(hWnd);
        // populate the main listview display
        RefreshListView(g_hwndLV);
        SetWindowText(hWnd, _T("POOM Calendar"));

        // save our default starting font
        hdc = GetDC(g_hwndLV);
        g_hfBaseFont = (HFONT)GetCurrentObject(hdc, OBJ_FONT);
        ReleaseDC(g_hwndLV, hdc);
    }
}

```

## Annex III – Codi generat per l'aplicació

```

    }
    else
    {
        //POOM could not initialize
        MessageBox(hWnd, _T("Cannot start POOM session"), _T("COM
Error"), MB_OK);
        EndDialog(hWnd, IDOK);
    }
    return TRUE;

case WM_DESTROY:
    //logout of POOM session
    ShutdownPoom();
    CommandBar_Destroy(g_hwndCB);
    break;
case WM_ACTIVATE:
// Notify shell of our activate message
    SHHandleWMActivate(hWnd, wParam, lParam, &s_sai, FALSE);
    break;
case WM_SETTINGCHANGE:
    SHHandleWMSettingChange(hWnd, wParam, lParam, &s_sai);
    break;

case WM_DRAWITEM:
    // redraw a listview item...called for each item
    lpdis = (LPDRAWITEMSTRUCT)lParam;

    if (lpdis->CtlType == ODT_LISTVIEW)
    {
        TCHAR buff[64];
        HBRUSH hbrBkColor;
        HBRUSH hbrSEL;
        COLORREF crOldBkColor;
        COLORREF crOldTextColor;
        COLORREF crRed = RGB(255,0,0);
        DWORD dwFontFlags = 0;
        HFONT hCustomFont = 0;

        hdc = lpdis->hDC;

        // save the current text and background colors
        crOldTextColor = GetTextColor(hdc);
        crOldBkColor = GetBkColor(hdc);

        hbrBkColor = CreateSolidBrush (crOldBkColor);
        hbrSEL = CreateSolidBrush (GetSysColor(COLOR_HIGHLIGHT));
        // make sure standard font is the default
        SelectObject(hdc, g_hfBaseFont);

        // if this item is selected, draw the highlight box
        if ((lpdis->itemAction | ODA_SELECT) && (lpdis->itemState &
ODS_SELECTED))
        {
            SetTextColor(hdc, GetSysColor(COLOR_HIGHLIGHTTEXT));
            FillRect(hdc, &lpdis->rcItem,hbrSEL);
        }
    }

```

## Assistent personal per persones amb necessitats especials

```

// otherwise fill with standard background color and draw a rectangle
around the item
else
{
    RECT r;

    FillRect(hdc, &lpdis->rcItem, hbrBkColor);
    r = lpdis->rcItem;
    Rectangle(hdc, r.left-1, r.top-1, r.right, r.bottom);

}

DeleteObject(hbrSEL);
DeleteObject(hbrBkColor);

ListView_GetItemText(g_hwndLV, lpdis->itemID, 0, buf, 64);
// if viewing tasks, set some font properties
if (g_nCurrFolder == olFolderTasks)
{
    if (GetTaskPriority(lpdis->itemID) == olImportanceHigh)
    {
        // color text red to denote high importance
        SetTextColor(hdc, crRed);
        dwFontFlags |= FT_BOLD;
    }

    if (TaskIsComplete(lpdis->itemID))
    {
        // cross out if task is complete
        dwFontFlags |= FT_STRIKE;
    }

    // Build and select the font
    hCustomFont = BuildFont(hdc, g_hfBaseFont, dwFontFlags);
    SelectObject(hdc, hCustomFont);
}

// Draw the item's text
DrawText(hdc, buf, _tcslen(buf), &lpdis->rcItem, DT_LEFT|
DT_WORDBREAK|DT_TOP);

// Reset the background color and the text color back to their
// original values.
SetTextColor(hdc, crOldTextColor);
SetBkColor(hdc, crOldBkColor);

if (hCustomFont != 0)
{
    DeleteObject(hCustomFont);
}

ReleaseDC(g_hwndLV, hdc);
}

return TRUE;

case WM_MEASUREITEM:

```

## Annex III – Codi generat per l'aplicació

```

        // set the height for each item box
        lpmi = (LPMEASUREITEMSTRUCT)lParam;
        lpmi->itemHeight= 36;

        return TRUE;

    }
    return FALSE;
}

// *****
// Function Name: CreateListView
//
// Purpose: Create a listview control suitable for listing POOM items
//
// Arguments:
//     IN HWND hwnd - handle to parent window
//
// Return Values:
//     HWND
//     handle of created list view control
//
// Description:
// This function creates an owner-drawn listview control with a single column.
// All drawing is handled in the WM_DRAWITEM handler in WndProc. The list items are
// sized via the WM_MEASUREITEM handler.
HWND CreateListView(HWND hwndPrnt)
{
    HWND hwndRet = NULL;

    LV_COLUMN lvColumn;
    RECT mainRect;
    // Get the main rect
    GetClientRect (hwndPrnt, &mainRect);

    // Create ListView
    hwndRet = CreateWindow (WC_LISTVIEW, NULL, WS_CHILD | WS_VISIBLE |
        LVS_REPORT | LVS_SINGLESEL | LVS_EDITLABELS | LVS_OWNERDRAWFIXED |
LVS_NOCOLUMNHEADER,
        0, 0, mainRect.right - mainRect.left, mainRect.bottom - mainRect.top,
        hwndPrnt, NULL, g_hInst, 0);

    ListView_SetItemCount(hwndRet, MAX_LISTVIEW_ITEMS);

    // set column width to screen width
    lvColumn.mask = LVCF_WIDTH;
    lvColumn.cx = mainRect.right - mainRect.left;

    ListView_InsertColumn(hwndRet, 0, &lvColumn);

    return hwndRet;
}

```

*Assistent personal per persones amb necessitats especials*

```

// *****
// Function Name: RefreshListView
//
// Purpose: Create a listview control suitable for listing POOM items
//
// Arguments:
//     IN HWND hwnd - handle to listview control
//
// Return Values:
//     none
//
// Description:
//     This function deletes and repopulates the the listview items with POOM
//     data and sets the first item as selected.
void RefreshListView(HWND hwndLV)
{
    listcount = 0;
    ListView_DeleteAllItems(hwndLV);
    PopulateList(g_nCurrFolder);
    ListView_SetItemState(hwndLV, 0, LVIS_SELECTED, LVIS_SELECTED);
}
// *****
// Function Name: AddToList
//
// Purpose: Add text to a listview item, then insert the item into the list
//
// Arguments:
//     IN TCHAR * tszTxt - ptr to a string containing text to add to listview item
//
// Return Values:
//     none
//
// Description:
//     This function simply creates a listview item with text content, then adds it
//     to the list. It also sets the item order (0 based) and sort parameter (1 based).
//     See LVITEM documentation for details
void AddToList(TCHAR * tszTxt)
{
    LVITEM lv = {0};
    lv.mask = LVIF_TEXT | LVIF_STATE | LVIF_PARAM | LVIF_IMAGE;
    lv.iItem = listcount++;
    lv.lParam = listcount;
    lv.pszText = tszTxt;
    ListView_InsertItem(g_hwndLV, &lv);
}
// *****
// Function Name: GetCtrlText
//
// Purpose: Get the displayed text from a given control
//
// Arguments:
//     IN HWND hDlg - control's parent hwnd
//     IN int nCtrlId - resource ID of the control
//     OUT BSTR * pbstrTxt - length-prefixed buffer for returned string. Must be
//     freed (SysFreeString) by calling code
//
// Return Values:
//     none

```

## Annex III – Codi generat per l'aplicació

```

//
// Description:
// This function retrieves the specified control's window text and stores it in
// an newly allocated BSTR which is returned via output parameter. The allocated
// string must be freed by the calling code with SysFreeString
void GetCtrlBstr(HWND hDlg,
                int nCtrlId,
                BSTR *pbstrTxt)
{
    int nLen = 0;

    nLen = GetWindowTextLength(GetDlgItem(hDlg, nCtrlId));
    (*pbstrTxt) = SysAllocStringLen(NULL, nLen+1);
    GetWindowText(GetDlgItem(hDlg, nCtrlId), *pbstrTxt, nLen+1);
}

HFONT BuildFont(HDC hDC, HFONT hBaseFont, DWORD dwFlags)
{
    LOGFONT lf;
    HFONT hFont = NULL;

    memset(&lf, 0, sizeof(LOGFONT));

    GetObject(hBaseFont, sizeof(lf), (LOGFONT*)&lf);

    if (dwFlags & FT_BOLD)
        lf.lfWeight=FW_BOLD;
    if (dwFlags & FT_STRIKE)
        lf.lfStrikeOut=TRUE;

    return CreateFontIndirect(&lf);
}

```



## 10. Bibliografia

- [1] <http://www.fundmaresme.com/>, Fundació el Maresme Pro Persones Amb Disminució Psíquica
- [2] <http://www.google.com/intl/es/googlecalendar/tour.html>, Google Calendar
- [3] <http://www.goosync.com/>, GooSync Home Page
- [4] <http://msdn.microsoft.com/en-us/library/aa458744.aspx>, *Pocket Outlook Object Model - Documentation for Windows Mobile-Based Pocket PCs*
- [5] [http://code.google.com/intl/es/apis/calendar/docs/2.0/developers\\_guide\\_protocol.html](http://code.google.com/intl/es/apis/calendar/docs/2.0/developers_guide_protocol.html), *Google Calendar APIs and Tools*
- [6] <http://www.speech.cs.cmu.edu/flite/doc/index.html>, *Flite: a small, fast speech synthesis engine*
- [7] <http://www.microsoft.com/downloads/Search.aspx?displaylang=es>, Centre de descàrrega de Microsoft
- [8] <http://es.wikipedia.org/wiki/Wikipedia:Portada>, Wikipedia, la enciclopedia libre

