# Funcionament de local/remot

El programa del terminal manté una copia de :

* Productes (wpdj\_pagaia\_qr\_sympo2023\_productes)
* Serveis (wpdj\_pagaia\_qr\_sympo2023\_serveis)
* Participants (wpdj\_pagaia\_qr\_sympo2023)
* Modalitats (wpdj\_pagaia\_qr\_sympo2023\_modalitats)

La taula de participants es divideix en el moment de la càrrega en :

* Participants
* Contractacions (una línia per servei amb el seu estat)

Les taules es carreguen automàticament amb les operacions :

* productes
* serveis
* modalitats
* participants

de forma automàtica al :

* Iniciar el programa
* Passar de background a foreground (activar programa ja executant-se)

A mes a mes es pot llegir directament una dada amb la operació I una id <> 0

El concepte de producte serveix per agrupar varis serveis a una sola compra. Ex. Si tan sols podem comprar esmorzars de simposi o de setmana aleshores tots els esmorzars de simposi i tots els de setmana s’associen als productes esmorzar simposi i esmorzar pagaia respectivament i quan es compra un dels productes es marquen com a pagats tots els serveis associats.

També serveix per mantenir un diari de compres per quadrat caixa.

Sobre aquestes dades es poden fer les següents operacions :

* registrar : marca com arribat un participant
* consumir : consumeix un servei
* comprar : compra un producte (marca pagat tots els serveis associats i guarda un registre de compra tan sols a la base de dades).

Es fan servir les dades locals per temes de presentació (seleccionar quins serveis es poden consumir, dades com codi I nom del participant, descripció del servei / producte, etc.)

Les operacions s’envien al servidor amb una id que en el cas de registrar es la id del participant, a consumir es (id participant \* 100) + id servei i a comprar es (id participant \* 100) + id producte. Aix+o es tableix una limitació de 99 productes/serveis. Si volem ho podem \* per 1000 però no sembla necessari.

L’us en local de les dades serveix per una primera validació de l’existència del participant/servei/producte però no dels estats. Es a dir s’envia al servidor encara que l’estat local sigui incorrecte.

En principi el funcionament es el següent :

* Check existència. Si error dona un missatge de error
* Check de correcció (ja registrat per comprar etc.)
  + Si OK executa la operació en local ienvia al srrvidor la operació. Al rebre resposta actualitza de nou les dades locals.
  + Si NO mostra missatge breument però envia al servidor la operació doncs no passa res si enviem consumir varies vegades. Al rebre les resposta actualitza les dades.
* Si la operació al servidor no s’executa (error de connexió) la resposta local es mostra I la operació es guarda a una llista I es segueix treballant normalment però indicant que NO estem conectats.
* Quan s’executa una operació correcte es procedeix a enviar al servidor totes les operacions pendents.
* Hi ha una pantalla per consultar les operacions pendents al servidor I forçar l’enviament.
* Quan una operació NO es pot executar per raons resolubles (ex. No està registrat o no ha comprat aquets serveis) surt a la resposta un botó per resoldre l’incidència (botó de registrar-se o botó de comprar).
* Si es resol l’incidència (es fa el registra o la compra) es reintenta l’operació.

Aquest funcionament, si hi ha connexió permet tants terminals actius com vulguem.

Si falla la conexió evidentment un terminal no s’entera de lo que fa un altre però la funcionalitat bàsica NO es veu afectada I les accions que es facin s’actualitzen automàticament al servidor quan es recupera la conexió.

Els terminal guarden a la memòria no volàtil totes les operacions I dades encara que no es puguin passar al servidor.

Les dades adicionals com les dates dels serveis I el producte dels serveis permeten presentar sempre una interficie que intenta limitar al màxim imformació errònia. De fet els botons del consumir tan sols surten si estàn a les dates correctes I la proposta de compra es de únicament del producte corresponent al servei que s’està intentant consumir.