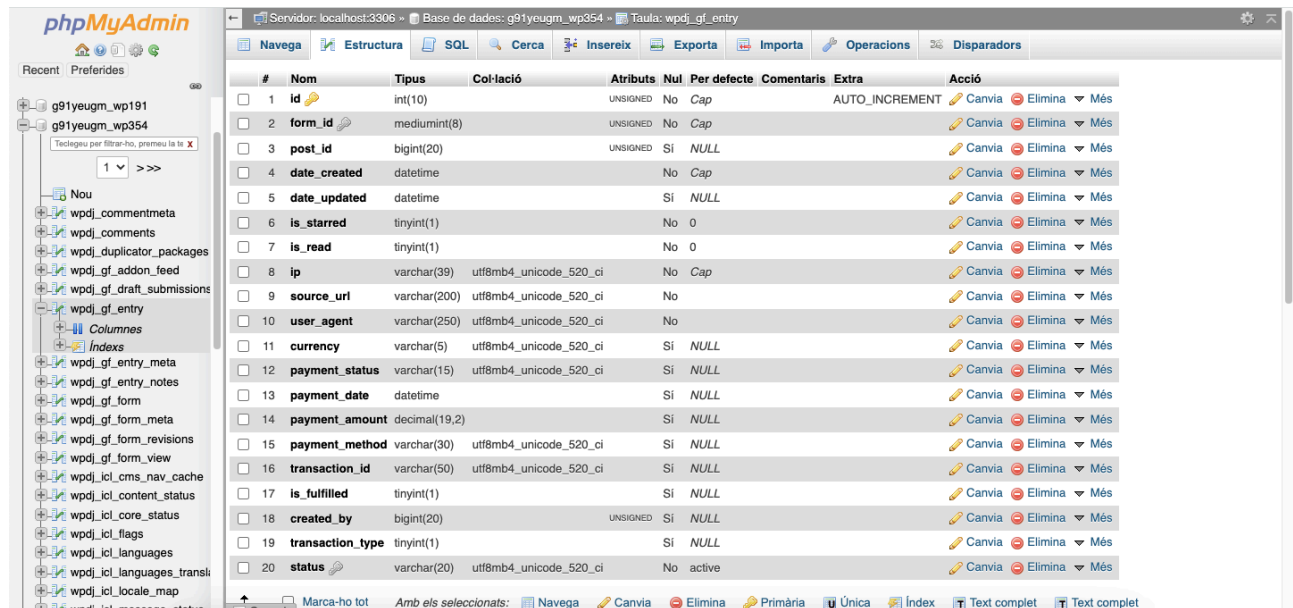


Estructura de Dades.

Actualment el Gravity Forms utilitza les taules que comencen per wpdj\_gf.

Les que contenen les dades són:

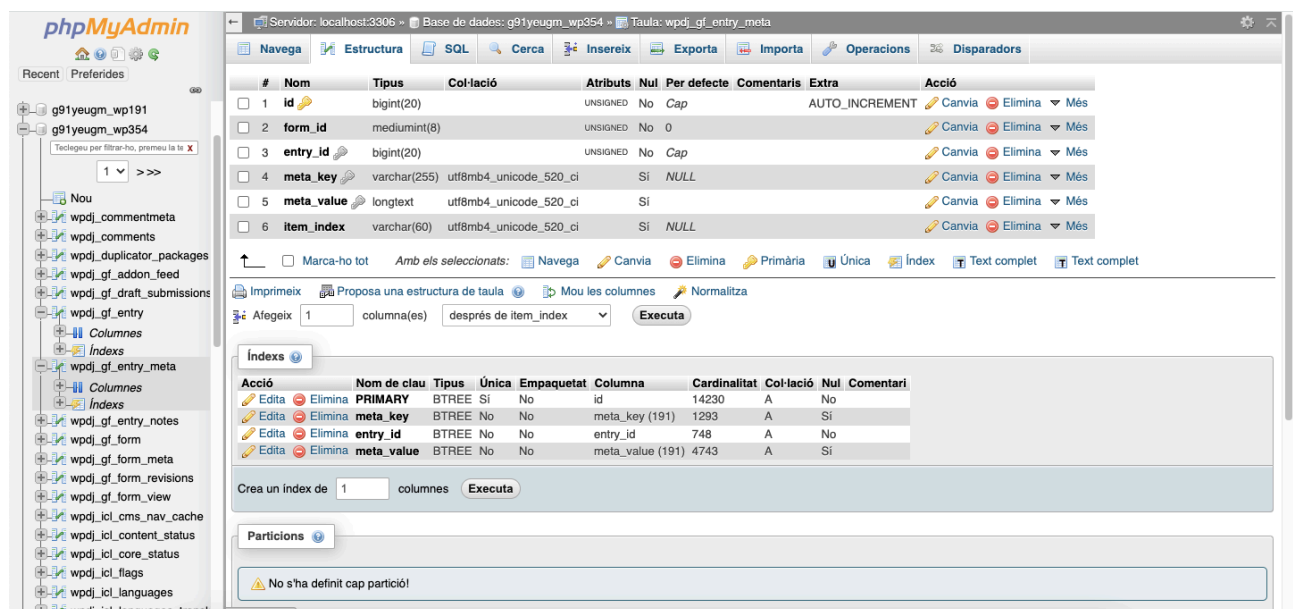
wpdj\_gf\_entry Una entrada per inscripció al formulari. Serien uns 300 registres.



phpMyAdmin - Servidor: localhost:3306 - Base de dades: g91yeugm\_wp354 - Taula: wpdj\_gf\_entry

#	Nom	Tipus	Col·locació	Atributs	Nul	Per defecte	Comentaris	Extra	Acció
1	id	int(10)		UNSIGNED	No	Cap		AUTO_INCREMENT	Canvia Elimina Més
2	form_id	mediumint(8)		UNSIGNED	No	Cap			Canvia Elimina Més
3	post_id	bigint(20)		UNSIGNED	Si	NULL			Canvia Elimina Més
4	date_created	datetime			No	Cap			Canvia Elimina Més
5	date_updated	datetime			Si	NULL			Canvia Elimina Més
6	is_starred	tinyint(1)			No	0			Canvia Elimina Més
7	is_read	tinyint(1)			No	0			Canvia Elimina Més
8	ip	varchar(39)	utf8mb4_unicode_520_ci		No	Cap			Canvia Elimina Més
9	source_url	varchar(200)	utf8mb4_unicode_520_ci		No				Canvia Elimina Més
10	user_agent	varchar(250)	utf8mb4_unicode_520_ci		No				Canvia Elimina Més
11	currency	varchar(5)	utf8mb4_unicode_520_ci		Si	NULL			Canvia Elimina Més
12	payment_status	varchar(15)	utf8mb4_unicode_520_ci		Si	NULL			Canvia Elimina Més
13	payment_date	datetime			Si	NULL			Canvia Elimina Més
14	payment_amount	decimal(19,2)			Si	NULL			Canvia Elimina Més
15	payment_method	varchar(30)	utf8mb4_unicode_520_ci		Si	NULL			Canvia Elimina Més
16	transaction_id	varchar(50)	utf8mb4_unicode_520_ci		Si	NULL			Canvia Elimina Més
17	is_fulfilled	tinyint(1)			Si	NULL			Canvia Elimina Més
18	created_by	bigint(20)		UNSIGNED	Si	NULL			Canvia Elimina Més
19	transaction_type	tinyint(1)			Si	NULL			Canvia Elimina Més
20	status	varchar(20)	utf8mb4_unicode_520_ci		No	active			Canvia Elimina Més

i la wpdj\_gf\_entry\_meta. Una entrada per cada camp omplert del formulari per cada inscripció. Si comptem que en cada inscripció del formulari hi ha un mínim de 24 camps plens (els opcionals no solen estar massa complimentats) hi haurien 300 inscrits x 24 dades/inscrit = 7.200 registres per el formulari de inscripcions al simpo.



phpMyAdmin - Servidor: localhost:3306 - Base de dades: g91yeugm\_wp354 - Taula: wpdj\_gf\_entry\_meta

#	Nom	Tipus	Col·locació	Atributs	Nul	Per defecte	Comentaris	Extra	Acció
1	id	bigint(20)		UNSIGNED	No	Cap		AUTO_INCREMENT	Canvia Elimina Més
2	form_id	mediumint(8)		UNSIGNED	No	0			Canvia Elimina Més
3	entry_id	bigint(20)		UNSIGNED	No	Cap			Canvia Elimina Més
4	meta_key	varchar(255)	utf8mb4_unicode_520_ci		Si	NULL			Canvia Elimina Més
5	meta_value	longtext	utf8mb4_unicode_520_ci		Si				Canvia Elimina Més
6	item_index	varchar(60)	utf8mb4_unicode_520_ci		Si	NULL			Canvia Elimina Més

Índexs

Acció	Nom de clau	Tipus	Única	Empaquetat	Columna	Cardinalitat	Col·locació	Nul	Comentari
Elimina	PRIMARY	BTREE	Si	No	id	14230	A	No	
Elimina	meta_key	BTREE	No	No	meta_key (191)	1293	A	Si	
Elimina	entry_id	BTREE	No	No	entry_id	748	A	No	
Elimina	meta_value	BTREE	No	No	meta_value (191)	4743	A	Si	

Crea un índex de 1 columnes Executa

Particions

No s'ha definit cap partició!

Per obtenir les dades faig aquesta select:

// Establir i realitzar consulta. La guardem en variable.

```
$consulta = "SELECT entry_id, meta_key, meta_value, date_created, date_updated
FROM wpdj_gf_entry_meta, wpdj_gf_entry
WHERE wpdj_gf_entry_meta.form_id = 1 AND wpdj_gf_entry.id =
wpdj_gf_entry_meta.entry_id AND status = 'active'
ORDER by wpdj_gf_entry_meta.entry_id DESC, wpdj_gf_entry_meta.meta_key";
```

\$resultat = mysqli\_query( \$conexio, \$consulta ) or die ( "Alguna cosa ha anat malament a la consulta a la base de dades");

El resultat d'aquesta SELECT es el següent (Primera pantalla. El id=757 correspon al últim participant)

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, including the 'wp\_gf\_entry\_meta' table. The main area displays the table structure and a list of entries. The table has columns: entry\_id, meta\_key, meta\_value, date\_created, and date\_updated. The data shows various entries with meta\_keys like 'Xavier', 'Si', 'Inea', 'Ewen', '16', '0', 'No', '226', 'Theene Strasse 78', 'Theene', '26624', 'Catalunya', 'Alegida a última hora amb Klaus Weyhing.', 'Pagat ...', 'Pagat', '226', 'Xavier', 'Participant', '1972-10-01', 'D', 'L2FV3MTCC', '(+49) 017148107', 'insa444ewen@web.de', 'Anglès', 'L', 'Europea', '210', 'Si', and 'A'.

I després carrego una taula interna al programa amb una línia per cada participant amb totes les dades del participant. Després l'exploto amb un bucle for. (Com si no, venint del cobol!!! jejeje)

El que proposo es crear una nova taula (única) molt més senzilla amb només les dades que necessitem per les funcionalitats del QR. que es carregaria a a partir de les dades del Gravity Forms (les de les taules i el programa anterior. Seria alguna cosa així:

The screenshot shows the phpMyAdmin interface for the 'wp\_gf\_pagaia\_qr\_sympo2023' table. The table structure is displayed with columns: #, Nom, Tipus, Col·lació, Atributs, Nul, Per defecte, Comentaris, Extra, and Acció. The data includes fields like 'Id', 'Nom', 'Modalitat', 'M\_Arribat', 'M\_Esm\_D1', 'M\_Esm\_D2', 'M\_Esm\_D3', 'M\_Esm\_D4', 'M\_Esm\_D5', 'M\_Esm\_D6', 'M\_Esm\_D7', 'M\_Pic\_D1', 'M\_Pic\_D2', 'M\_Pic\_D3', 'M\_Sopar\_Sympo', 'M\_Sopar\_SP', and 'Data\_Modificat'. Each row has a corresponding 'Acció' column with links for 'Canvia', 'Elimina', and 'Més'.

Això ja ho faré amb una versió del programa php que fa el resum de dades del simpòsium. Pots executar-lo amb la següent URL:

<https://seguridadwebsimpo.pagaia.club/sympo19-resum/>

Amb pwd=nOmEsIAsApstU Atenció, cal ser discret!!!

Lògica de programació:

1.- Generació de correu als participants. Un cop tancada la llista de inscripcions (el formulari) i abans del 31 de març de 2022.

Ho farà en Xavier Tous a partir del excel de les dades de inscripció al formulari. Aquest codi ha de contenir, com a mínim el id del participant. Podria contenir també el Nom per possibles quadres. En principi ho farà amb les Eines de Workspace de Google i ho ficarà en un pdf que s'enviarà al correu dels inscrits. El disseny que proposo seria aquest, que conté tots els camps necessaris. En cas de no funcionar el mòbil, serviria de control d'accès. Recordem que això els hi enviarem i ho poden tenir en paper, al mòbil o als dos. Si no funciona al mòbil, simplement cal mirar el contingut del "carnet".



.- Nou logo del simpòsium

.- — El mateix format al mòbil que al pdf enviat!!!!

.-QR generat amb totes les dades identificatives que calguin. Al menys ha de contenir el ID del participant

.-El nom i cognoms del participant

.-La modalitat d'inscripció: Colaborador, Coach (Ponent?), Participant (Soci/no?), Acompanyant de Ponent, Acompanyant (participant)

.-Les opcions que té contractades/pagades amb la senyal de SI o NO. També es podria imprimir només les contractades.  
Tenir en compte que col·laboradors i ponents i els seus acompanyants ho tenen tot inclòs.

Això es el que s'enviarà a cada participant. S'els hi dirà que ho tinguin disponible al mòbil per ensenyar a l'entrada al Simposium. Si no va bé al mòbil, o per seguretat imprimir-ho en paper. La mida es la de un "carnet" (petita) en el mateix format que el mòbil.

**Afegir imprès el codi de participant!!!**

2.- Creació de la taula wpdj\_pagaia\_qr\_sympo2023 a la BB.DD. del Simpòsium. (la de la tercera còpia de pantalla). Al mateix temps que el correu del primer punt. Es fan versions de proves quan comenci la inscripció a partir del 1 de novembre.

Ho farà en JordiMo utilitzant les taules del Gravity Forms (la primera i segona còpia de pantalla) una versió del programa existent de resum i/o generació d'excel del Sympo 2019.

### 3.-Programa del Servidor.

Ho farà en JordiMo amb php accedint (i actualitzant) la taula wpdj\_pagaia\_qr\_sympo2023.

Aquest programa (per fer proves podem fer servir el indicat en el whatsapp) ha de rebre com a mínim dos dades:

3.1.- "Id" l'identificador del participant. Ve del registre del formulari i és el que està al codi QR. És el camp Id de la taula wpdj\_pagaia\_qr\_sympo2023. Serveix per accedir a les dades del participant.

3.2.- Codi de petició. Podria ser un camp numèric molt senzill que indiqui què es demana.

- 01 - Validació arribada
- 02 - Validació esmorzar 1er. dia (01/04/2023)
- 03 - Validació esmorzar 2on. dia (02/04/2023)
- 04 - Validació esmorzar 3er. dia (03/04/2023)
- 05 - Validació esmorzar 4rt. dia (04/04/2023)
- 06 - Validació esmorzar 5è. dia (05/04/2023)
- 07 - Validació esmorzar 6è. dia (06/04/2023)
- 08 - Validació esmorzar 7è. dia (06/04/2023)
- 09 - Validació picnic 1er. dia (01/04/2023)
- 10 - Validació picnic 2on. dia (02/04/2023)
- 11 - Validació picnic 3er. dia (03/04/2023)
- 12 - Validació sopar simpòsium
- 13 - Validació sopar Setmana Paleig
- 14 - Validació ....

O, encara més senzill, donat que tenim la data tant al mòbil com al servidor:

- 1 - Validació arribada
- 2 - Validació esmorzars (agafarà el dia del Servidor)
- 3 - Validació pícnic (agafarà el dia del Servidor)
- 4 - Validació sopars (agafarà el dia del Servidor)

El retorn pot ser tant senzill com:

Camp 1.- Un (o dos) dígits de codi de petició Els enviats

Camp 2.- Els literals "OK" o bé "Error"

Camp 3.- Descripció del error. (En blanc si OK) Podrien ser simplement:

- "Identificador no trobat"
- "Esmorzar no contractat/pagat"
- "Esmorzar ja utilitzat"
- "Picnic no contractat/pagat"
- "Picnic ja recollit"
- "Entrada sopar simpòsium ja utilitzat"
- "Entrada sopar Setmana Paleig ja utilitzat"

Enviats/retornats en el format que pactem, amb l'encodificació que diguem, etc...

A més a més de retornar el OK/Error hauria de retornar després totes les dades actualitzades del participant.

Es a dir

Línia 1 : OK/Error + Hash

Línia 2 : T + Text de l'Error

Línia 3 : Dades actualitzades del participant camps separats per ";", encoding utf8

Afegir una opció '00' que retorna les dades de tots els participants format csv

Aquí l'Error és un error tècnic (fallo base de dades etc.)

Línia 1 : OK/Error + Hash

Línia 2 : T + text de l'error

Línia 3...:

Una línia per participant

Camps separats per ";", encoding utf8

#### 4.- Programa del client/mòbil.

Ho farà en Paco utilitzant la càmera per llegir el codi QR generat en el punt 1. Per mòbils Android (al menys) amb els llenguatges DART i les llibreries FLUTTER.

Aquest programa es pot despreocupar de les dades de inscripció. Llegint l'Id del QR ja seria suficient. Amb l'excepció que es vulgui habilitar una BB.DD. local com a Pla de Contingència.

Encara que el Pla de Contingència podria ser simplement mirar en el document generat en el punt 1 (el que també hi ha el QR que s'està intentant llegir) si té pagat/contractat el que s'està controlant.

La pantalla principal de validacions podria ser una cosa així:



Al prémer una de les opcions, s'activaria la càmera i llegiria el QR que conté entre altres coses el Id del participant. Un número sencer ("entero"). Llavors ja es pot cridar al programa del Servidor de Nominalia enviat aquest número i un 1 si ha premut arribada, 2 si Esmorzar, 3 si picnic i 4 si Sopar.

El programa php del Servidor sabrà quin participant accedir amb el Id i quina dada consultar i actualitzar combinat el darrer número amb la data del Servidor. I retornar el resultat segons lo vist en el punt anterior.

Quan pulses l'opció s'ha de obrir el selector de participants.

Es com el que et vaig enviar, pots seleccionar per QR, codi o nom i fer click a la llista.

Quan es QR o codi o tan sols un participant seleccionat salta a la pantalla de validació incidència a partir de la resposta del servidor.

Una vegada processat torna a la pantalla de participants. No cal tornar al menú principal.

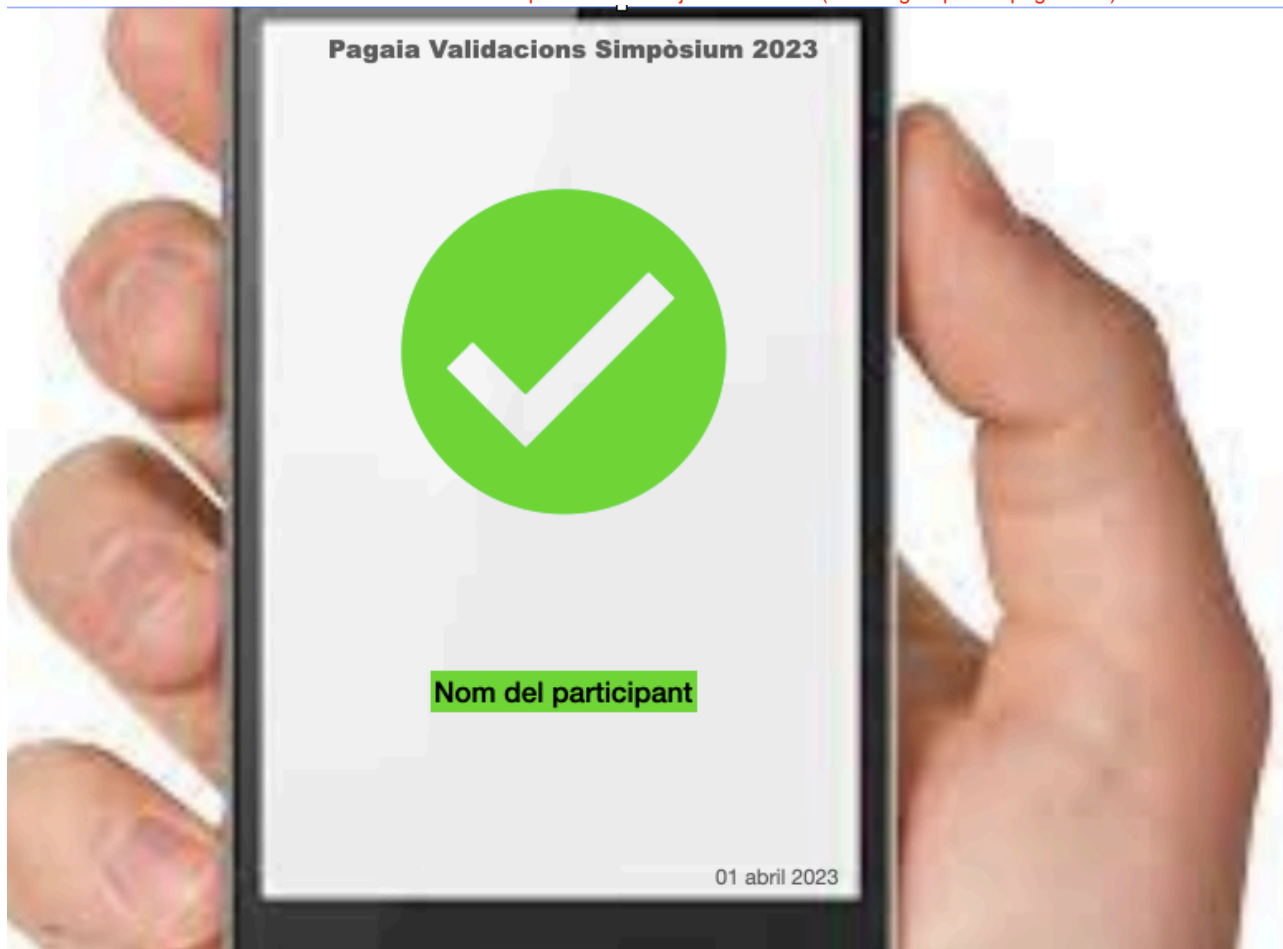
La pantalla de Registre / Arribada hauria de servir també per mostrar les dades del participant (nom i serveis pagats) posteriorment. Es a dir si el participant JA està registrat mostrarà les dades.

El tema comentat de pagament de serveis NO contractes a la cantina, es podria fer amb una opció a la resposta de NO Contractat. El botó enviarà una nova operació al servidor amb el codi de participant i opció contractada i guardaria localment una llista per poder quadrar caixa després. (Tot i que el servidor hauria de tenir les dades).

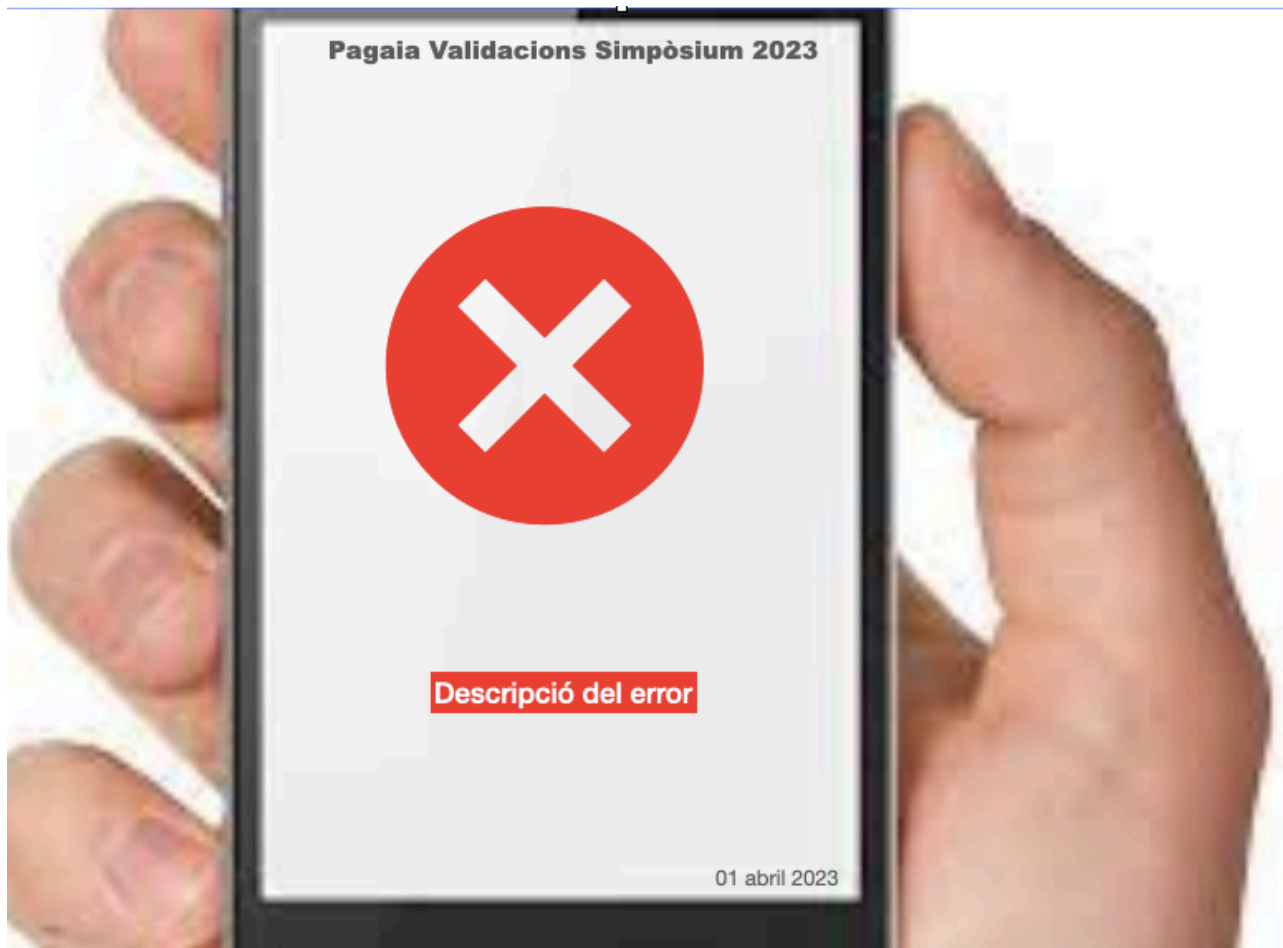


I la de retorn una cosa així si OK:

Hauria de sortir QUE s'ha validat i en el cas de No Pagat si es vol fer la opció de cobrar al moment un botó de "Pagar" amb una pantalla intermitja de validació (Està segur que vol pagar ....?).



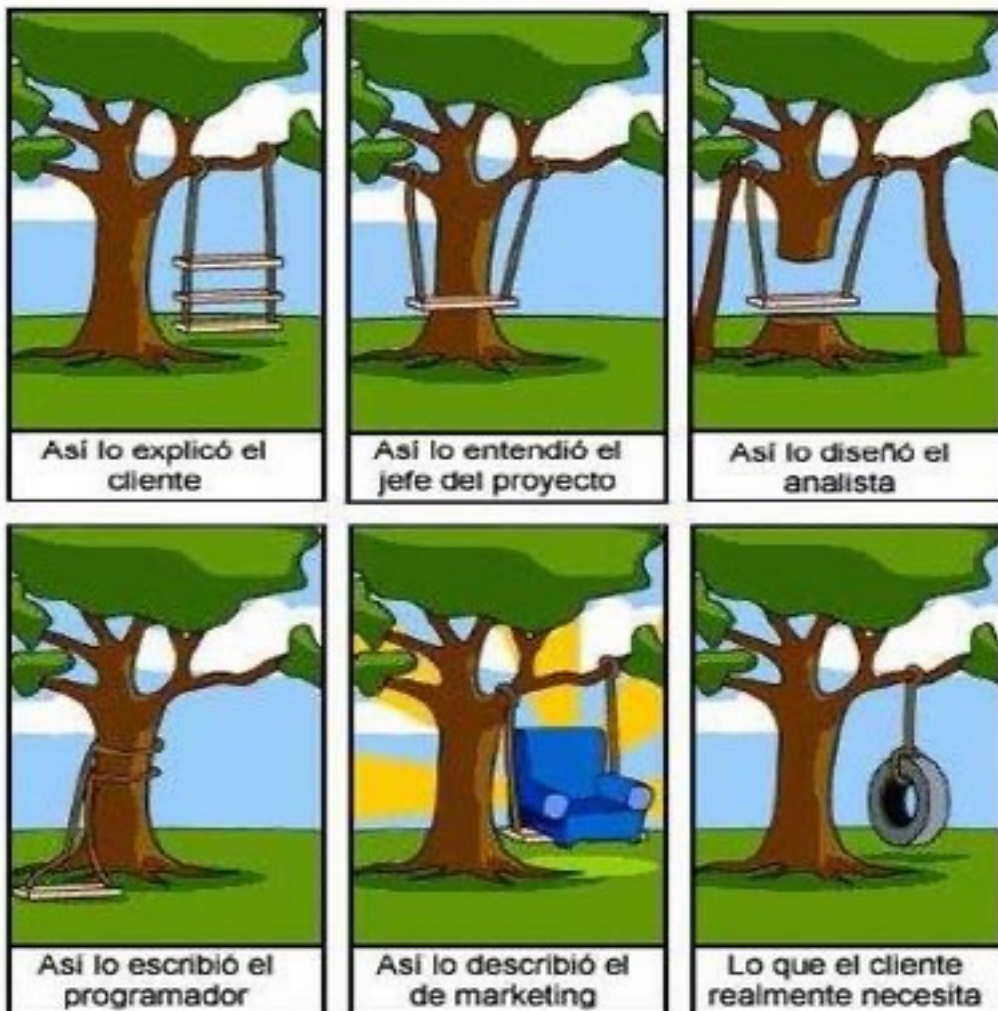
O així si Error:



Dins la meua ignorància sobre les aplicacions de mòbil, crec que això es el més senzill. I ràpid. En qualsevol cas, hem de veure quines son les necessitats. L'Anàlisi Funcional!!!

Tenim temps per parlar-ne.

Voldria evitar que ens passí aquest clàssic:



Flutter

Les aplicacions de Flutter es basen en que el programa manté un "estat" o "model" únic i la pantalla simplement mostre el model.

Es a dir no existeixen els processos de pintar pantalla, e tc.

Ex.:

Tenim un model

que es el

id client  
nom client  
serveis

Quan enviem l'opció al servidor aquest retorna si accepta fer-lo hi la modificació de les dades del client. Automàticament al actualitzar el client la pantalla s'adapta al resultat. Es una mica diferent de la programació tradicional.

De fet "Sempre" has de tenir una estructura de dades que es el teu model de lo que processes.

Evidentment pot ser un sol participant o molts participants. No es un base de dades en el concepte de Oracle o MySql però segueix sent important que estigui normalitzada i no poder repetir informacions. Ha d'haver en qualsevol cas una sola "veritat".

Queda raro lo de "veritat".